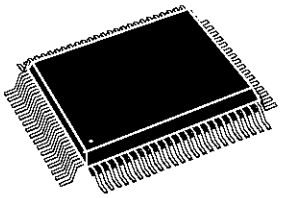
SGS-THOMSON MICROELECTRONICS

ST6240

8-BIT HCMOS MCU WITH LCD DRIVER, EEPROM AND A/D CONVERTER

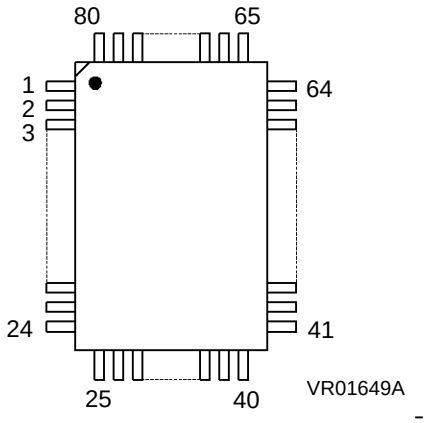
- 3 to 6V supply operating range
- 8.4MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait & Stop Modes
- 5 different interrupt vectors
- Look-up table capability in ROM
- User ROM: 7948 bytes
  Data RAM: 192 bytes
  LCD RAM: 24 bytes
  EEPROM: 128 bytes
- PQFP80 package
- 16 fully software programmable I/O as:
  - Input with/without pull-up resistor
  - Input with interrupt generation
  - Open-Drain or Push-pull outputs
  - Analog Inputs (12 pins)
- 4 I/O lines can sink up to 20mA for direct LED or TRIAC driving, and have SPI alternate functions
- Two 8-bit counters with 7-bit programmable prescalers (Timers 1 and 2)
- Software or hardware activated digital watchdog
- 8-bit A/D converter with up to 12 analog inputs
- 8-bit synchronous Serial Peripheral Interface (SPI)
- LCD driver with 45 segment outputs, 4 back-plane outputs and selectable duty cycle for up to 180 LCD segments direct driving
- 32kHz oscillator for stand-by LCD operation
- Power Supply Supervisor (PSS)
- One external not maskable interrupt
- 9 powerful addressing modes
- The accumulator, the X, Y, V & W registers, the port and peripherals data & control registers are addressed in the data space as RAM locations.
- The ST62E40 is the EPROM version, ST62T40 is the OTP version
- Development tool: ST6240-EMU connected via RS232 to an MS-DOS Personal Computer



PQFP80

(Ordering Information at the end of the datasheet)

Figure 1. 80 Pin Quad Flat Pack (QFP) Package Pinout



ST6240 Pin Description

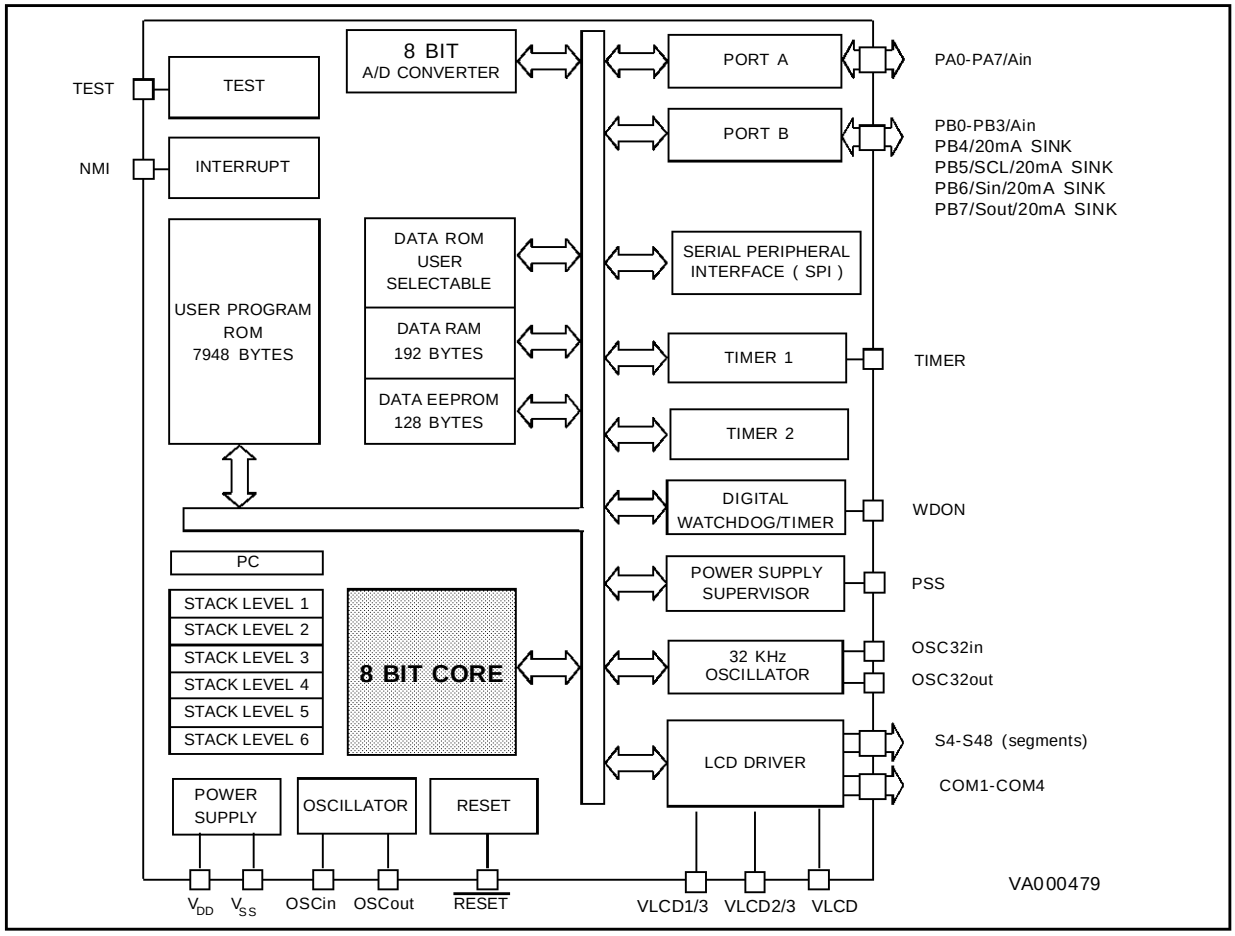
Pin number	Pin name	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
1	S43	25	$\overline{\text{RESET}}$	64	S26	65	S27
2	S44	26	OSCout	63	S25	66	S28
3	S45	27	OSCin	62	S24	67	S29
4	S46	28	WDON	61	S23	68	S30
5	S47	29	NMI	60	S22	69	S31
6	S48	30	TIMER	59	S21	70	S32
7	COM4	31	PB7/Sout (1)	58	S20	71	S33
8	COM3	32	PB6/Sin (1)	57	S19	72	S34
9	COM2	33	PB5/SCL (1)	56	S18	73	S35
10	COM1	34	PB4 (1)	55	S17	74	S36
11	VLCD1/3	35	PB3/Ain	54	S16	75	S37
12	VLCD2/3	36	PB2/Ain	53	S15	76	S38
13	VLCD	37	PB1/Ain	52	S14	77	S39
14	PA7/Ain	38	PB0/Ain	51	S13	78	S40
15	PA6/Ain	39	OSC32out	50	S12	79	S41
16	PA5/Ain	40	OSC32in	49	S11	80	S42
17	PA4/Ain			48	S10		
18	TEST			47	S9		
19	PA3/Ain			46	S8		
20	PA2/Ain			45	S7		
21	PA1/Ain			44	S6		
22	PA0/Ain			43	S5		
23	V_{DD}			42	S4		
24	V_{SS}			41	PSS		

Note 1: 20mA Sink

GENERAL DESCRIPTION

The ST6240 microcontroller is a member of the 8-bit HCMOS ST62xx family, a series of devices oriented to low-medium complexity applications. All ST62xx members are based on a building block approach: a common core is associated with a combination of on-chip peripherals (macrocells). The macrocells of the ST6240 are: a high performance LCD controller/driver with 45 segment outputs and 4 backplanes able to drive up to 180 segments, two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), the digital watchdog (DWD), an 8-bit A/D Converter with up to 12 analog inputs, a Power Supply Supervisor and an 8-bit synchronous Serial Peripheral Interface (SPI). In addition these devices offer 128 bytes of EEPROM for storage of non volatile data. Thanks to these peripherals the ST6240 is well suited for general purpose, automotive, security, appliance and industrial applications. The ST62E40 EPROM version is available for prototypes and low-volume production, an OTP version is also available (see separate datasheet).

Figure 2. ST6240 Block Diagram



Note:
Ain= Analog Input

PIN DESCRIPTION

V_{DD} and **V_{SS}**. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected with the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. OSCin is the input pin, OSCout is the output pin. An external clock signal can be applied to OSCin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to restart the microcontroller at the beginning of its program. The $\overline{\text{RESET}}$ pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TEST. The TEST must be held at V_{SS} for normal operation (an internal pull-down resistor selects normal operating mode if TEST pin is not connected).

NMI. The NMI pin provides the capability for asynchronous applying an external top priority interrupt to the MCU.This pin is falling edge sensitive. The NMI pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TIMER. This is the TIMER 1 I/O pin. In input mode it is connected to the prescaler and acts as external timer clock or as control gate for the internal timer clock. In the output mode the TIMER pin outputs the data bit when a time-out occurs.

WDON. This pin selects the watchdog enabling option (hardware or software). A low level selects the hardware activated option (the watchdog is always active), a high level selects the software activated option (the watchdog can be activated by software, deactivated only by reset, thus enabling STOP mode). An internal pull-up resistance selects the software watchdog option if the WDON pin is not connected.

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured under software control as an input with or without pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output or as analog input for the A/D converter. Port A has a 5mA drive capability in output mode.

PB0-PB3,PB4-PB7. These 8 lines are organized as one I/O port (B). Each line may be configured under software control as an input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output. PB0-PB3 can be programmed as analog inputs for the A/D converter while PB4-PB7 can also sink 20mA for direct LED driving. PB5-PB7 can also be used as respectively Clock, Data in and Data out pins for the on-chip SPI to carry the synchronous serial I/O signals.

COM1-COM4. These four pins are the LCD peripheral common outputs. They are the outputs of the on-chip backplane voltage generator which is used for multiplexing the 45 LCD lines allowing up to 180 segments to be driven.

S4-S48. These pins are the 45 LCD peripheral driver outputs of ST6240. Segments S1-S3 are not connected to any pin.

VLCD. Display voltage supply. It determines the high voltage level on COM1-COM4 and S4-S48 pins.

VLCD1/3, VLCD2/3. Display supply voltage inputs for determining the display voltage levels on COM1-COM4 and S4-S48 pins during multiplex operation.

PSS. This is the Power Supply Supervisor sensing pin. When the voltage applied to this pin is falling below a software programmed value the highest priority (NMI) interrupt can be generated. This pin has to be connected to the voltage to be supervised.

OSC32in and OSC32out. These pins are internally connected with the on-chip 32kHz oscillator circuit. A 32.768kHz quartz crystal can be connected between these two pins if it is necessary to provide the LCD stand-by clock and real time interrupt. OSC32in is the input pin, OSC32out is the output pin.

ST62xx CORE

The core of the ST62xx Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control busses. The in-core communication is arranged as shown in Figure 3; the controller being externally linked to both the reset and the oscillator, while the core is linked to the dedicated on-chip macrocells peripherals via the serial data bus and indirectly for interrupt purposes through the control registers.

Registers

The ST62xx Family core has six registers and three pairs of flags available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs.

Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is addressed in the data space as RAM location at address FFh. Accordingly, the ST62xx instruction set can use the accumulator as any other register of the data space.

Figure 4. ST62xx Core Programming Model

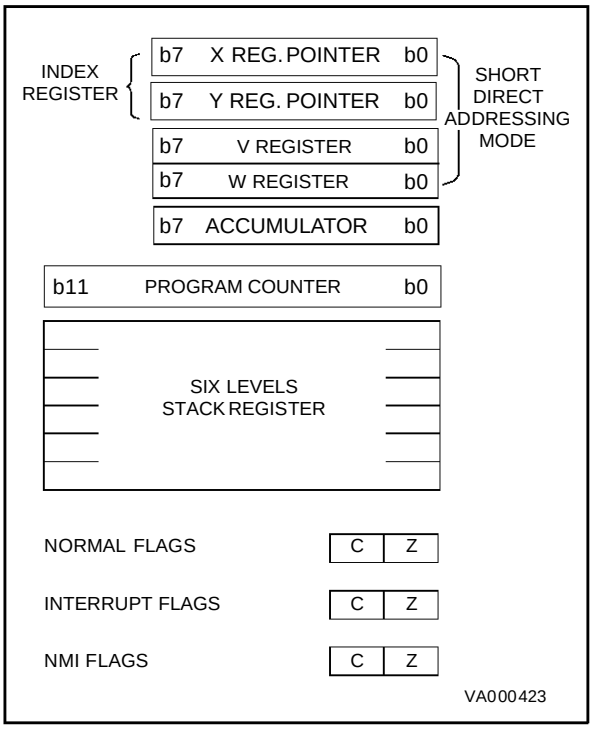
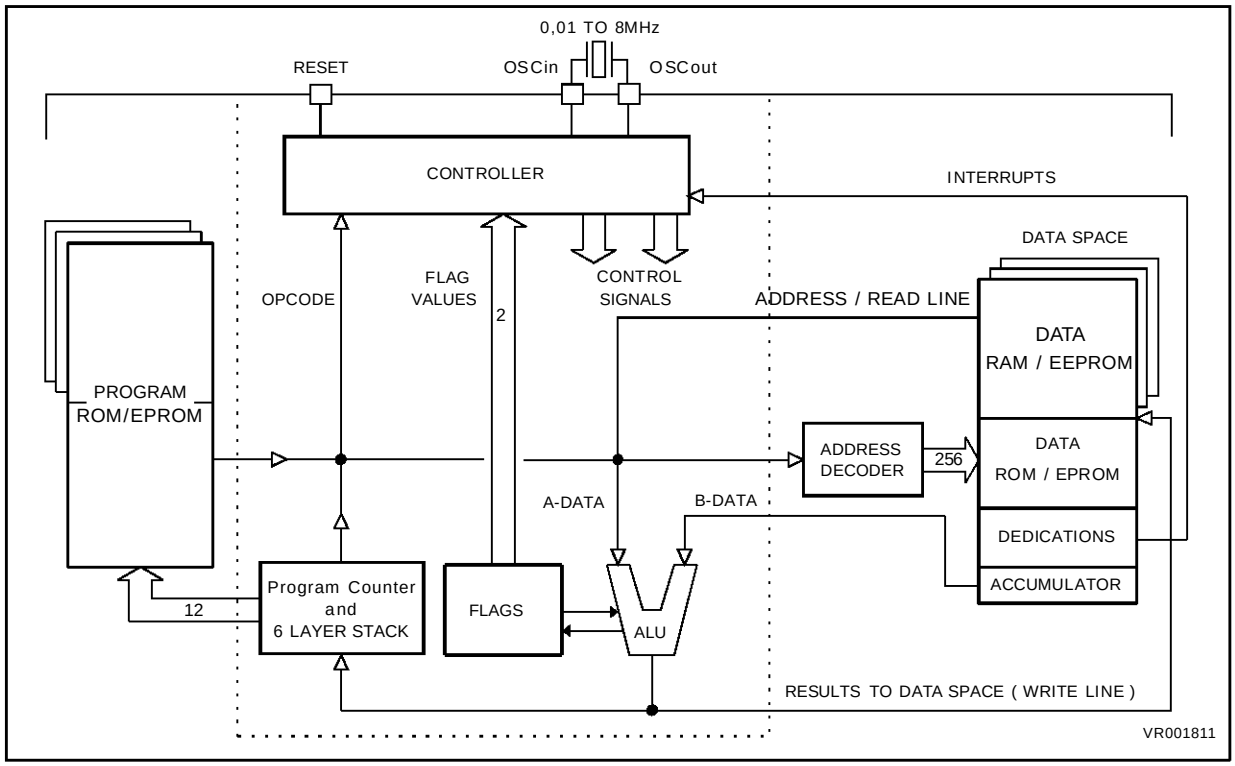


Figure 3. ST62xx Core Block Diagram



ST62xx CORE (Continued)

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in the data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST62xx instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save one byte in short direct addressing mode. These registers can be addressed in the data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST62xx instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or an address of operand. The 12-bit length allows the direct addressing of 4096 bytes in the program space. Nevertheless, if the program space contains more than 4096 locations, as for the ST6240, the further program space can be addressed by using the Program ROM Page register.
The PC value is incremented after it is read from the address of the current instruction. To execute relative jumps the PC and the offset are shifted through the ALU, where they will be added, and the result is shifted back into the PC. The program counter can be changed in the following ways:

- JP (Jump) instruction . . . PC=Jump address
- CALL instruction . . . . PC=Call address
- Relative Branch instructions . . . . . PC=PC ± offset
- Interrupt . . . . . PC=Interrupt vector
- Reset . . . . . PC=Reset vector
- RET & RETI instructions . PC=Pop (stack)
- Normal instruction . . . . PC=PC+1

Flags (C, Z)

The ST62xx core includes three pairs of flags that correspond to 3 different modes: normal mode, interrupt mode and Non-Maskable Interrupt-Mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during normal operation, one pair is used during the interrupt mode (CI, ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

The ST62xx core uses the pair of flags that correspond to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST62xx core uses the interrupt flags (resp. the NMI flags) instead of the normal flags. When the RETI instruction is executed, the normal flags (resp. the interrupt flags) are restored if the MCU was in the normal mode (resp. in the interrupt mode) before the interrupt. It should be observed that each flag set can only be addressed in its own mode (Not-maskable interrupt, normal interrupt or main mode). The flags are not cleared during the context switching and so remain in the state they were at the exit of the last mode switch.

The Carry flag is set when a carry or a borrow occurs during arithmetic operations, otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The Zero flag is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

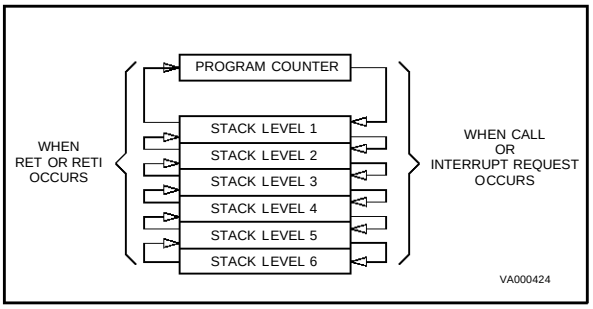
The switching between the three sets of Flags is automatically performed when an NMI, an interrupt or a RETI instruction occurs. As the NMI mode is automatically selected after the reset of the MCU, the ST62xx core uses at first the NMI flags.

ST62xx CORE (Continued)

Stack

The ST62xx core includes a true LIFO hardware stack that eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be lost). When a subroutine or interrupt return occurs (RET or RETI instructions), the first level register is shifted back into the PC and the value of each level is popped back into the previous level. These two operating modes are described in Figure 5. Since the accumulator, as all other data space registers, is not stored in the stack, the handling of these registers should be performed inside the subroutine. The stack pointer will remain in its deepest position if more than 6 calls or interrupts are executed, so that the last return address will be lost. It will also remain in its highest position if the stack is empty and a RET or RETI is executed. In this case the next instruction will be executed.

Figure 5. Stack Operation



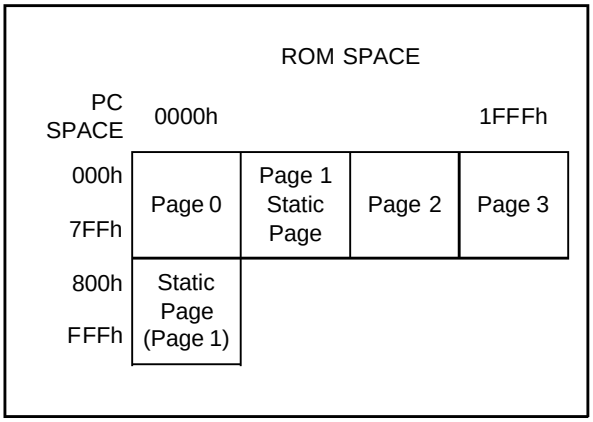
MEMORY SPACES

The MCU operates in three different memory spaces: program space, data space, and stack space. A description of these spaces is shown in the following figures.

Program Space

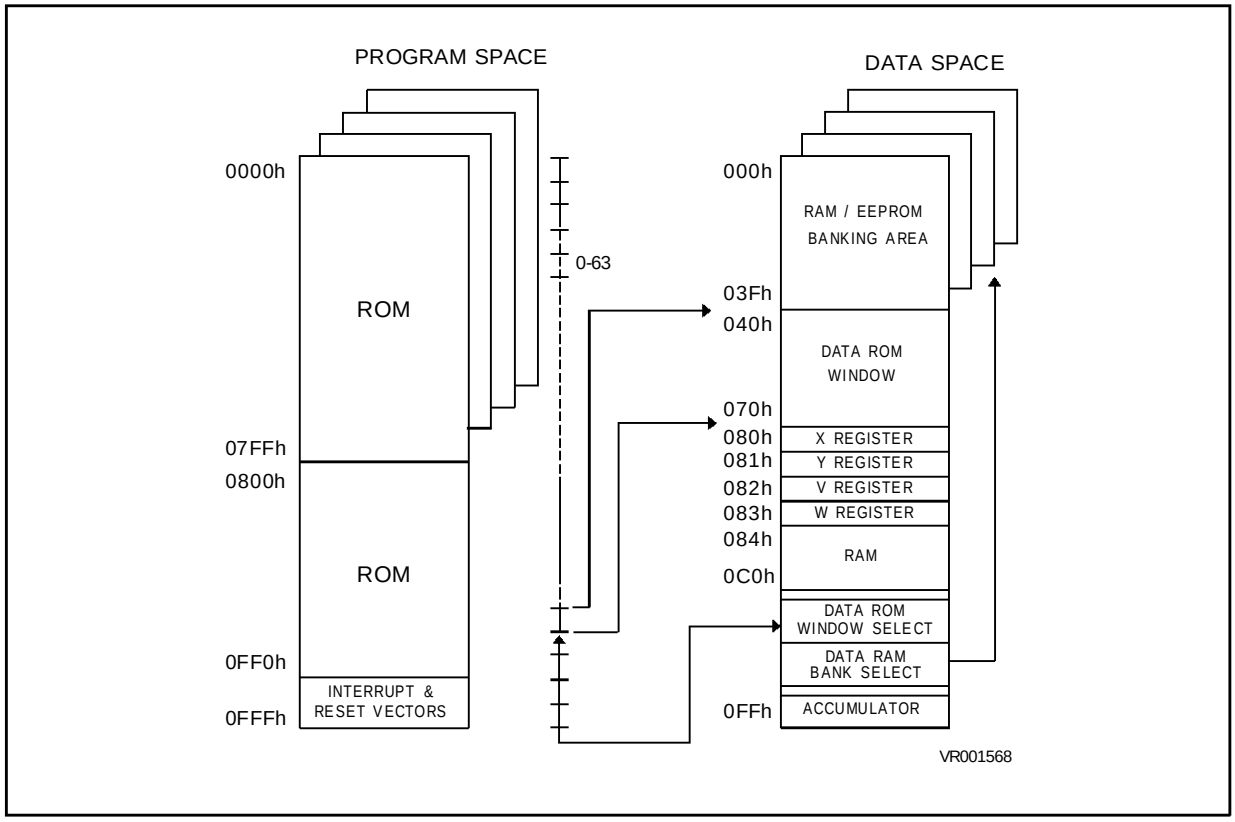
The program space is physically implemented in the ROM memory and includes all the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, the reserved test area and the user vectors. It is addressed by the 12-bit Program Counter register (PC register) and so the ST62xx core can directly address up to 4K bytes of Program Space. Nevertheless, the Program Space can be extended by the addition of 2Kbyte ROM banks as it is shown in the following figure in which the ST6240 8Kbyte memory is described.

Figure 6. ST6240 8Kbytes Program Space Addressing Description



MEMORY SPACES (Continued)

Figure 7. ST62xx Memory Addressing Description Diagram



These banks are addressed in the 000h-7FFh locations of the Program Space by the Program Counter and by writing the appropriate code in the Program ROM Page Register (PRPR register) located at address CAh of the Data Space. Because interrupts and common subroutines should be available all the time, only the lower 2K byte of the 4K program space are bank switched while the upper 2K byte can be seen as static page. Table 2 gives the different codes that allow the selection of the corresponding banks. Note that, from the memory point of view, Page 1 and the Static Page represent the same physical memory: it is only two different ways of addressing the same locations. On the ST6240 a total of 8192 bytes of ROM have been implemented; 7948 are available as user ROM while 244 are reserved for SGS-THOMSON test purposes.

Table 1. ST6240 Program ROM Memory Map

ROM Page	Device Address	Description
Page 0	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 “STATIC”	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	Reserved User ROM
Page 3	0000h-000Fh 0010h-07FFh	Reserved User ROM

MEMORY SPACES (Continued)

Data Space

The instruction set of the ST62xx core operates on a specific space, named Data Space, that contains all the data necessary for the processing of the program. The Data Space allows the addressing of RAM memory, ST62xx core/peripheral registers, and read-only data such as constants and look-up tables.

Data ROM. All the read-only data is physically implemented in the ROM memory in which the Program Space is also implemented. The ROM memory contains consequently the program to be executed, the constants and the look-up tables needed for the program.

The locations of Data Space in which the different constants and look-up tables are addressed by the ST62xx core can be considered as being a 64-byte window through which it is possible to access to the read-only data stored in the ROM memory .

Data RAM/EEPROM. The ST6240 offers 192 bytes of data RAM memory and 128 bytes of EEPROM. 64 bytes of RAM are directly addressed in data space in the range 080h-0BFh (static space). The additional RAM and the EEPROM are addressed using the banks of 64 bytes located between addresses 00h and 3Fh.

Additionally RAM are available in the LCD data map from E0h to F7h and are not banked.

Stack Space

The stack space consists of six 12 bit registers that are used for stacking subroutine and interrupt return addresses plus the current program counter register.

Figure 8. ST6240 Data Memory Space

DATA RAM/EEPROM BANK AREA	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
SPI INT. DISABLE REGISTER	0C2h*
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
RESERVED	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
PROGRAM ROM PAGE REGISTER	0CAh*
DATA RAM/EEPROM BANK REGISTER	0CBh*
PORT A OPTION REGISTER	0CCh
RESERVED	0CDh
PORT B OPTION REGISTER	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER 1 PRESCALER REGISTER	0D2h
TIMER 1 COUNTER REGISTER	0D3h
TIMER 1 STATUS/CONT REGISTER	0D4h
TIMER 2 PRESCALER REGISTER	0D5h
TIMER 2 COUNTER REGISTER	0D6h
TIMER 2 STATUS/CONT REGISTER	0D7h
WATCHDOG REGISTER	0D8h
RESERVED	0D9h
PSS STATUS/CONTROL REGISTER	0DAh
32kHz OSC. CONTROL REGISTER	0DBh
LCD MODE CONTROL REGISTER	0DCh
SPI DATA REGISTER	0DDh
RESERVED	0DEh
EEPROM CONTROL REGISTER	0DFh
LCD RAM	0E0h 0F7h
DATA RAM 7 BYTES	0F8h 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

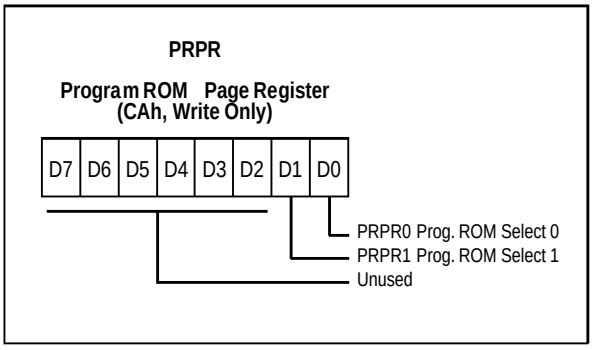
MEMORY SPACES (Continued)

Program ROM Page Register (PRPR)

The PRPR register can be addressed like a RAM location in the Data Space at the address CAh; nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to select the 2-Kbyte ROM bank of the Program Space that will be addressed. The number of the page has to be loaded in the PRPR register. Refer to the Program Space description for additional information concerning the use of this register. The PRPR register is not modified when an interrupt or a subroutine occurs.

Care is required when handling the PRPR register as it is write only. For this reason, it is not allowed to change the PRPR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. This operation may be necessary if common routines and interrupt service routines take more than 2K bytes; in this case it could be necessary to divide the interrupt service routine into a (minor) part in the static page (start and end) and to a second (major) part in one of the dynamic pages. If it is impossible to avoid the writing of this register in interrupt service routines, an image of this register must be saved in a RAM location, and each time the program writes to the PRPR it must write also to the image register. The image register must be written before PRPR, so if an interrupt occurs between the two instructions the PRPR is not affected.

Figure 9. Program ROM Page Register



D7-D2. These bits are not used.

PRPR1-PRPR0. These are the program ROM banking bits and the value loaded selects the corresponding page to be addressed in the lower part of the 4K program address space as specified in Table 2.

Table 2. ST6240 8Kbytes Program ROM Page Register Coding

PRPR1	PRPR0	PC bit 11	Memory Page
X	X	1	Static Page (Page1)
0	0	0	Page 0
0	1	0	Page 1 (Static Page)
1	0	0	Page 2
1	1	0	Page 3

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Note:

Only the lower part of address space is bank switched because interrupt vectors and common subroutines should be available at all times. The reason of this structure is due to the fact that it is not possible to jump from one dynamic page to another except by jumping back to the static page, changing contents of PRPR, and then jumping to a different dynamic page.

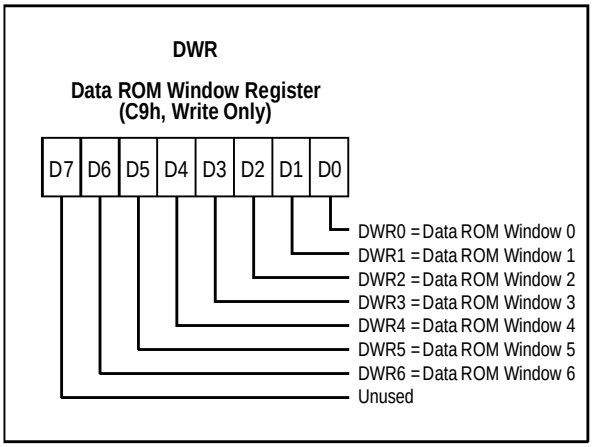
MEMORY SPACES (Continued)

Data Window register (DWR)

The Data ROM window is located from address 040h to address 7Fh in the Data space. It allows the direct reading of 64 consecutive bytes located anywhere in the ROM memory between the addresses 0000h and 1FFFh. All the bytes of the ROM memory can be used to store either instructions or read-only data. Indeed, the window can be moved by step of 64 bytes along the ROM memory in writing the appropriate code in the Write-only Data Window register (DWR register, location C9h).

The DWR register can be addressed like a RAM location in the Data Space at the address C9h, nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to move the 64-byte read-only data window (from the 40h address to 7Fh address of the Data Space) up and down the ROM memory of the MCU in steps of 64 bytes. The effective address of the byte to be read as a data in the ROM memory is obtained by the concatenation of the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits, see Figure 10). So when addressing location 40h of dataspace, and 0 is loaded in the DWR register, the phisycal addressed location in ROM is 00h. The DWR register is not cleared at reset, therefore it must be written to before the first access to the Data ROM window area.

Figure 11. Data ROM Window Register



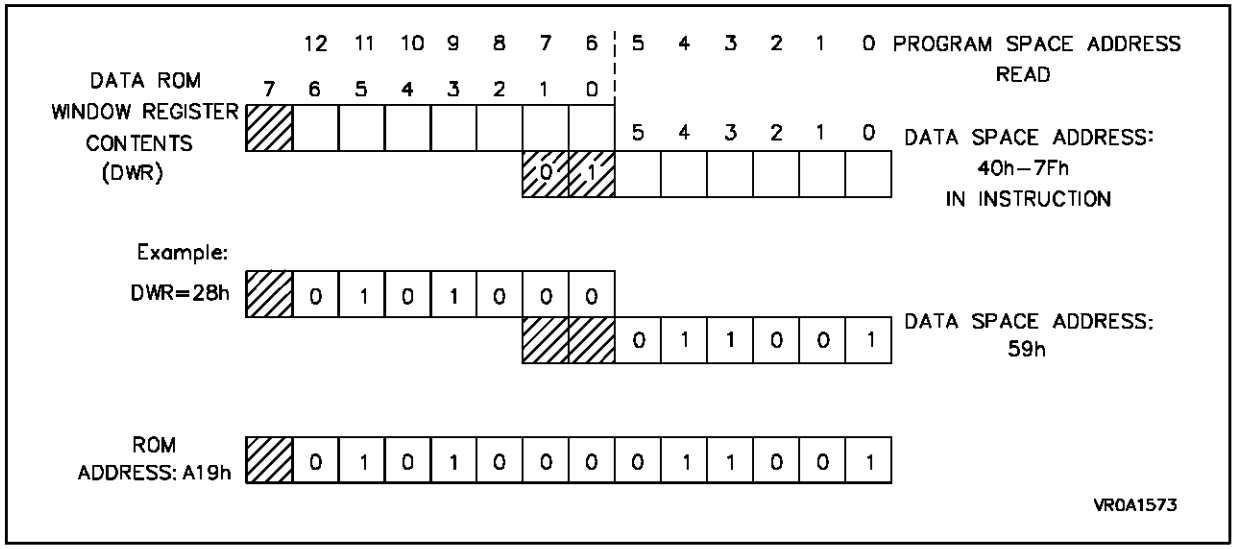
D7. This bit is not used.

DWR6-DWR0. These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Note: Care is required when handling the DWR register as it is write only. For this reason, it is not allowed to change the DWR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in the interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to the DWR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DWR is not affected.

Figure 10. Data ROM Window Memory Addressing



MEMORY SPACES (Continued)

Data RAM/EEPROM Bank Register (DRBR)

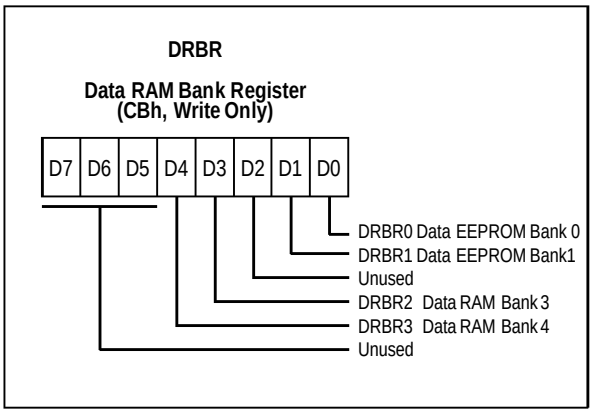
The selection of the bank is made by programming the Data RAM Bank Switch register (DRBR register) located at address CBh of the Data Space. The number of the selected bank is equal to the bit content of the DRBR register. In this way each bank of RAM or EEPROM can be selected 64 bytes at a time. No more than one bank should be set at a time.

The DRBR register can be addressed like a RAM location in the Data Space at the address CBh; nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to select the desired 64-byte RAM/EEPROM bank of the Data Space. The number of the bank has to be loaded in the DRBR register and the instruction has to point to the selected location as if it was in bank 0 (from 00h address to 3Fh address). This register is not cleared during the MCU initialization, therefore it must be written before the first access to the Data Space bank region. Refer to the Data Space description for additional information. The DRBR register is not modified when a interrupt or a subroutine occurs.

Table 3. Data RAM Bank Register Set-up

DRBR Value	Selection
01h	EEPROM Page 0
02h	EEPROM Page 1
08h	RAM Page 1
10h	RAM Page 2

Figure 12. Data RAM Bank Register



The table 3 summarizes how to set the data RAM bank register in order to select the various banks or pages.

D7-D5. These bits are not used.

DRBR4-DRBR3. Each of these bits, when set, will select one RAM page.

D2. This bit is not used.

DRBR1-DRBR0. Each of these bits, when set, will select one EEPROM page.

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Notes:

Care is required when handling the DRBR register as it is write only. For this reason, it is not allowed to change the DRBR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to DRBR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DRBR is not affected.

In DRBR Register, *only 1 bit must be set*. Otherwise two or more pages are enabled in parallel, producing errors.

MEMORY SPACES (Continued)

EEPROM Description

The data space of ST62xx family from 00h to 3Fh is paged as described in Table 4. The ST6240 has 128 bytes of EEPROM located in two pages of 64 bytes (pages 0 and 1).

The EEPROM pages are physically organized in 32 byte modules (2 modules per page) and does not require dedicated instructions to be accessed in reading or writing. The EEPROM is controlled by the EEPROM Control Register (EECTL = DFh). In order to enable access to the EEPROM, bit 6 of this register must be cleared otherwise any access to the EEPROM will be meaningless.

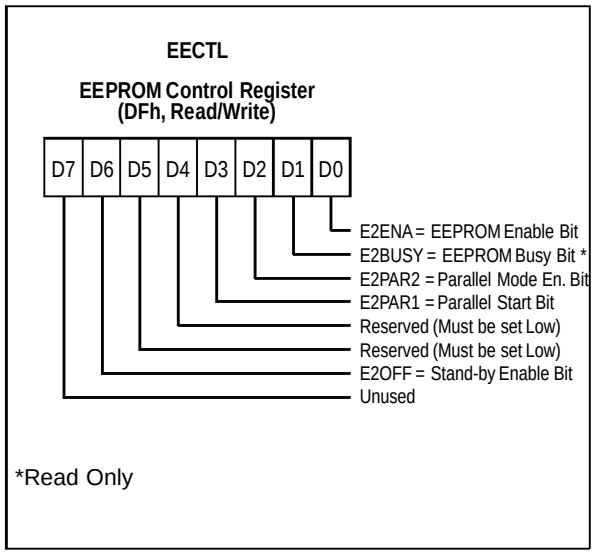
Any EEPROM location can be read just like any other data location, also in terms of access time.

When writing to an EEPROM, the EEPROM is not accessible by the ST62xx. A busy flag can be read to identify the EEPROM status before attempting any access. Writing the EEPROM can work in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). BMODE is the normal way to use the EEPROM and consists in accessing one byte at a time. PMODE consists in accessing 8 bytes per time.

Readout of the EEPROM is made at the same speed as RAM acces.

D7. Not Used

Figure 13. EEPROM Control Register



E2OFF. *WRITE ONLY.* If this bit is set the EEPROM is disabled (any access will be meaningless) and the power consumption of the EEPROM is reduced to the lowest values.

D5, D4. Reserved, must be set to zero.

E2PAR1. *WRITE ONLY.* Once in Parallel Mode, as soon as the user software sets the E2PAR1 bit the parallel writing of the 8 adjacent registers will start. It is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written; after parallel programming the undefined bytes will be unaffected

E2PAR2. *WRITE ONLY.* This bit must be set by the user program in order to perform parallel programming (more than one byte at a time). If E2PAR2 is set and the parallel start bit (E2PAR1) is low, up to 8 adjacent bytes can be written at maximum speed, the contents being stored in volatile registers. These 8 adjacent bytes are considered as a row, whose address lines A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bits. E2PAR2 is automatically reset at the end of any parallel programming procedure. It can be reset by the user software before starting the programming procedure, leaving the EEPROM registers unchanged.

E2BUSY. *READ ONLY.* This bit will be automatically set by the EEPROM control logic when the user program modifies an EEPROM register. The user program must test it before any read or write EEPROM operation; any attempt to access the EEPROM while the busy bit is set will be aborted and the writing procedure in progress completed.

E2ENA. *WRITE ONLY.* This bit MUST be set to one in order to write to any EEPROM register. If the user program attempts to write to the EEPROM when E2ENA = “0”, the involved registers will be unaffected and the BS will not be set.

After RESET the content of EECTL register will be 00h.

Notes:

The data to write has to be written directly at the address that it will have inside the EEPROM space. There is no buffer memory between the data•RAM and the EEPROM spaces.

When the EEPROM is busy (E2BUSY = “1”) EECTL can not be accessed in write mode, it is only possible to read the status of E2BUSY. This implies that as long as the EEPROM is busy, it is not possible to change the status of the EEPROM control register. EECTL bits 4 and 5 are reserved for test purposes, and must never be set to “1”.

MEMORY SPACES (Continued)

Table 4. EEPROM Parallel Write Row Structure

Byte	0	1	2	3	4	5	6	7	Dataspace addresses. Banks 0 and 1.
ROW7									38h-3Fh
ROW6									30h-37h
ROW5									28h-2Fh
ROW4									20h-27h
ROW3									18h-1Fh
ROW2									10h-1Fh
ROW1									08h-0Fh
ROW0									00h-07h

Up to 8 bytes in each row may be programmed at the same time in Parallel Write mode

Additional Notes on Parallel Mode. If the user wishes to perform parallel programming, the first action should be to set the E2PAR2 bit to one. From this time the EEPROM will be addressed in writing, the ROW address will be latched and it will be possible to change it only at the end of the programming procedure or by resetting E2PAR2 without programming the EEPROM. After the ROW address latching the ST62xx can “see” only one EEPROM row (the selected one) and any attempt to write or read other rows will produce errors. Do not read the EEPROM while E2PAR2 is set.

As soon as E2PAR2 bit is set, the 8 volatile ROW latches are cleared. From this moment the user can load data in the whole ROW or in a subset. Setting E2PAR1 will modify the EEPROM registers corresponding to the ROW latches accessed after E2PAR2. For example, if the software sets E2PAR2 and accesses the EEPROM by writing to addresses 18h, 1Ah, 1Bh and then sets E2PAR1, these three registers will be modified at the same time; the remaining bytes will be unaffected. Note that E2PAR2 is internally reset at the end of the programming procedure. This implies that the user must set E2PAR2 bit between two parallel programming procedures. Note that if the user tries to set E2PAR1 while E2PAR2 is not set there will not be any programming procedure and the E2PAR1 bit will be unaffected. Consequently E2PAR1 bit cannot be set if E2ENA is low. E2PAR1 can be affected by the user to set it, only if E2ENA and E2PAR2 bits are also set to one.

TEST MODE

For normal operation the TEST pin must be held low. An on-chip 100kΩ pull-down resistor is internally connected to the TEST pin.

INTERRUPTS

The ST62xx core can manage 4 different maskable interrupt sources, plus one non-maskable interrupt source (top priority level interrupt). Each source is associated with a particular interrupt vector that contains a Jump instruction to the related interrupt service routine. Each vector is located in the Program Space at a particular address.
When a source provides an interrupt request, and the request processing is also enabled by the ST62xx core, then the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction). Finally, the PC is loaded with the address of the Jump instruction and the interrupt routine is processed.

The ST6240 microcontroller has nine different interrupt sources associated to different interrupt vectors as described in Table 5.

Table 5. Interrupt Vectors - Sources Relationship

Interrupt Source	Associated Vector	Vector Address
NMI & PSS Pins	Interrupt Vector #0 (NMI)	(FFCh-FFDh)
SPI Peripheral	Interrupt Vector #1	(FF6h-FF7h)
Port A & B Pins	Interrupt Vector #2	(FF4h-FF5h)
TIMER 1, 2 & 32kHz Oscillator	Interrupt Vector #3	(FF2h-FF3h)
ADC Peripheral	Interrupt Vector #4	(FF0h-FF1h)

Interrupt Vectors Description

The ST62xx core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines in the static page of the Program Space:

– The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at addresses FFCh,FFDh in the Program Space. On ST6240 this vector is associated with the external falling edge sensitive interrupt pin (NMI) and is also connected to the Power Supply Supervisor circuit. The PSS and NMI interrupts are "ORed" together and the discrimination between PSS interrupt and NMI interrupt can be done by reading the interrupt flag (bit 7) of the PSS control register (Address DAh). An on-chip 100kΩ pull-up resistor is internally connected to the NMI pin.

– The interrupt vector located at the addresses FF6h, FF7h is named interrupt vector #1. It is associated with SPI peripheral and can be programmed by software to generate an interrupt request after the falling edge or low level of the eighth external clock pulse according to the code loaded in the Interrupt Option Register (IOR).

– The interrupt vector located at the addresses FF4h, FF5h is named interrupt vector #2. It is associated with Port A and B pins and can be programmed by software either in the falling edge detection mode or in the rising edge detection mode according to the code loaded in the Interrupt Option Register (IOR).

– The interrupt vector located at the addresses FF2h, FF3h is named interrupt vector #3. It is associated with Timer 1, Timer 2 and the 32kHz Oscillator peripherals. All these interrupts are "ORed" together and are connected to interrupt line #3 of the core. Discrimination among the three interrupts must be made by polling the Status/Control registers of Timer 1 (0D4h), Timer 2 (0D7h) and 32kHz oscillator (0DBh).

– The interrupt vector located at the addresses FF0h, FF1h is named interrupt vector #4. It is associated with the A/D converter peripheral.

All the on-chip peripherals (refer to their descriptions for further details) have an interrupt request flag bit (TMZ for timer, EOC for A/D, etc.), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI for timer, EAI for A/D, etc.) that must be set to one to allow the transfer of the flag bit to the Core.

Interrupt Priority

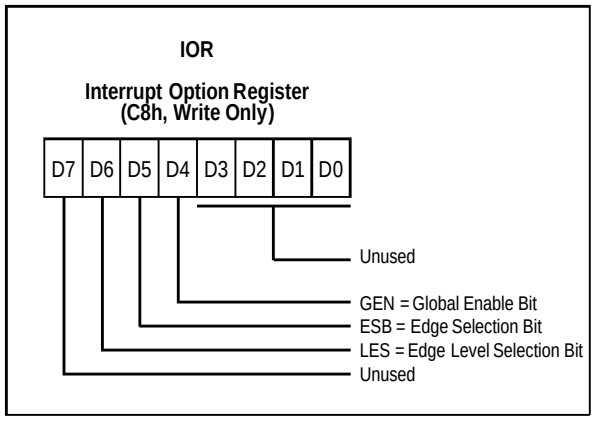
The non-maskable interrupt request has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the four other interrupts cannot interrupt each other. If more than one interrupt request is pending, they are processed by the ST62xx core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

INTERRUPTS (Continued)

Interrupt Option Register

The Interrupt Option Register (IOR register, location C8h) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register can be addressed in the Data Space as RAM location at the address C8h, nevertheless it is a write-only register that cannot be accessed with single-bit operations. The operating modes of the external interrupt inputs associated to interrupt vectors #1 and #2 are selected through bits 5 and 6 of the IOR register.

Figure 14. Interrupt Option Register



D7. This bit is not used.

LES. Level/Edge Selection Bit. When this bit is set to one, the interrupt #1 (SPI) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

ESB. Edge Selection Bit. When this bit is set to one, the interrupt #2 (Port A & B lines) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

GEN. Global Enable Interrupt. When this bit is set to one, all the interrupts are enabled. When this bit is cleared to zero all the interrupts (excluding NMI/PSS) are disabled.

This register is cleared on reset.

Table 6. Interrupt Option Register Description

GEN	SET	Enable all interrupts
	CLEARED	Disable all interrupts
ESB	SET	Rising edge mode on interrupt input #2
	CLEARED	Falling edge mode on interrupt input #2
LES	SET	Level-sensitive mode on interrupt input #1
	CLEARED	Falling edge mode on interrupt input #1
OTHERS	NOT USED	

External Interrupts Operating Modes

The NMI interrupt is associated to the NMI and PSS external pins of the ST6240. The two interrupt requests are "ORed". The highest priority interrupt request will be generated either by a falling edge applied to the NMI pin or when the voltage level applied to the PSS pin goes below the software programmed value. The discrimination between NMI and PSS interrupt can be done by polling the interrupt flag (Bit 7) of the PSS control register (DAh). The NMI interrupt pin signal is latched and is automatically reset by the core at the beginning of the non-maskable interrupt service routine. An on-chip pull-up resistor and a schmitt trigger is available with the NMI pin.

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (SPI vector #1, Ports A and B vector #2,) are connected to two internal latches. Each latch is set when a falling/rising edge occurs and is cleared when the associated interrupt routine is started. So, the occurrence of an external interrupt request is stored: a second interrupt, that occurs during the processing of the first one, will be processed as soon as the first one has been finished (if there is not an higher priority interrupt request). If more than one interrupt occurs during the processing of the first one, these other interrupt requests will be lost.

The storage of the interrupt requests is not available in the level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the core samples the line after the execution of the instructions.

During the end of each instruction the core tests the interrupt lines and if there is an interrupt request the next instruction is not executed and the related interrupt routine is executed.

INTERRUPTS (Continued)

Interrupt Procedure. The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event the user does not know about the context and the time at which it occurred. As a result the user should save all the data space registers which will be used inside the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes which are automatically switched and so these do not need to be saved.

The following list summarizes the interrupt procedure:

ST62xx actions

– Interrupt detection

– The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (or the NMI flags)

– The value of the PC is stored in the first level of the stack

– The normal interrupt lines are inhibited (NMI still active)

– First internal latch is cleared

– The related interrupt vector is loaded in the PC.

User actions

– User selected registers are saved inside the interrupt service routine (normally on a software stack)

– The source of the interrupt is found by polling (if more than one source is associated to the same vector) the interrupt flag of the source.

– Interrupt servicing

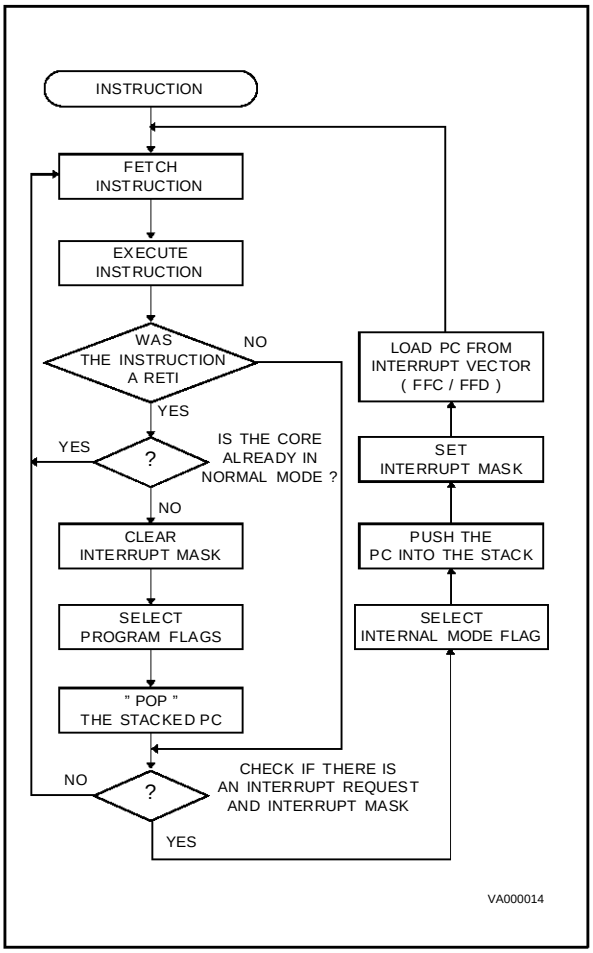
– Return from interrupt (RETI)

ST62xx actions

– Automatically the ST62xx core switches back to the normal flags (or the interrupt flags) and pops the previous PC value from the stack

The interrupt routine begins usually by the identification of the device that has generated the interrupt request (by polling). The user should save the registers which are used inside the interrupt routine (that holds relevant data) into a software stack. After the RETI instruction execution, the core carries out the previous actions and the main routine can continue.

Figure 15. Interrupt Processing Flow-Chart



INTERRUPTS (Continued)

Interrupt request and mask bits

Interrupt Option Register, IOR Location C8h

– GEN. If this bit is set, all the ST62xx interrupts are enabled, if reset all interrupts are disabled (including the NMI).

– ESB. If this bit is set, all the input lines associated to interrupt vector #2 are rising edge sensitive, if reset they are falling edge sensitive.

– LES. If this bit is set, all the inputs lines associated to interrupt vector #1 are low level sensitive, if reset they are falling edge sensitive.

All other bits in this register are not used.

Timer Peripherals, TSCR1 and TSCR2 registers, locations D4h and D7h

– TMZ. A low-to-high transition indicates that the timer count register has decremented to zero. This means that an interrupt request can be generated in relation to the state of ETI bit.

– ETI. This bit, when set, enables the timer interrupt request.

A/D Converter Peripheral, ADCR register location D0h

– EOC. This read only bit indicates when a conversion has been completed, by going to one. An interrupt request can be generated in relation to the state of EAI bit.

– EAI. This bit, when set, enables the A/D converter interrupt request.

PSS Peripheral, PSSCR Register location DAh

– PSSVD. This read only bit indicates when the voltage level applied to the PSS pin goes below the software programmed value. An interrupt request can be generated in relation to the state of PSSEI bit.

– PSSEI. This bit, when set, enables the Power Supply Supervisor interrupt request.

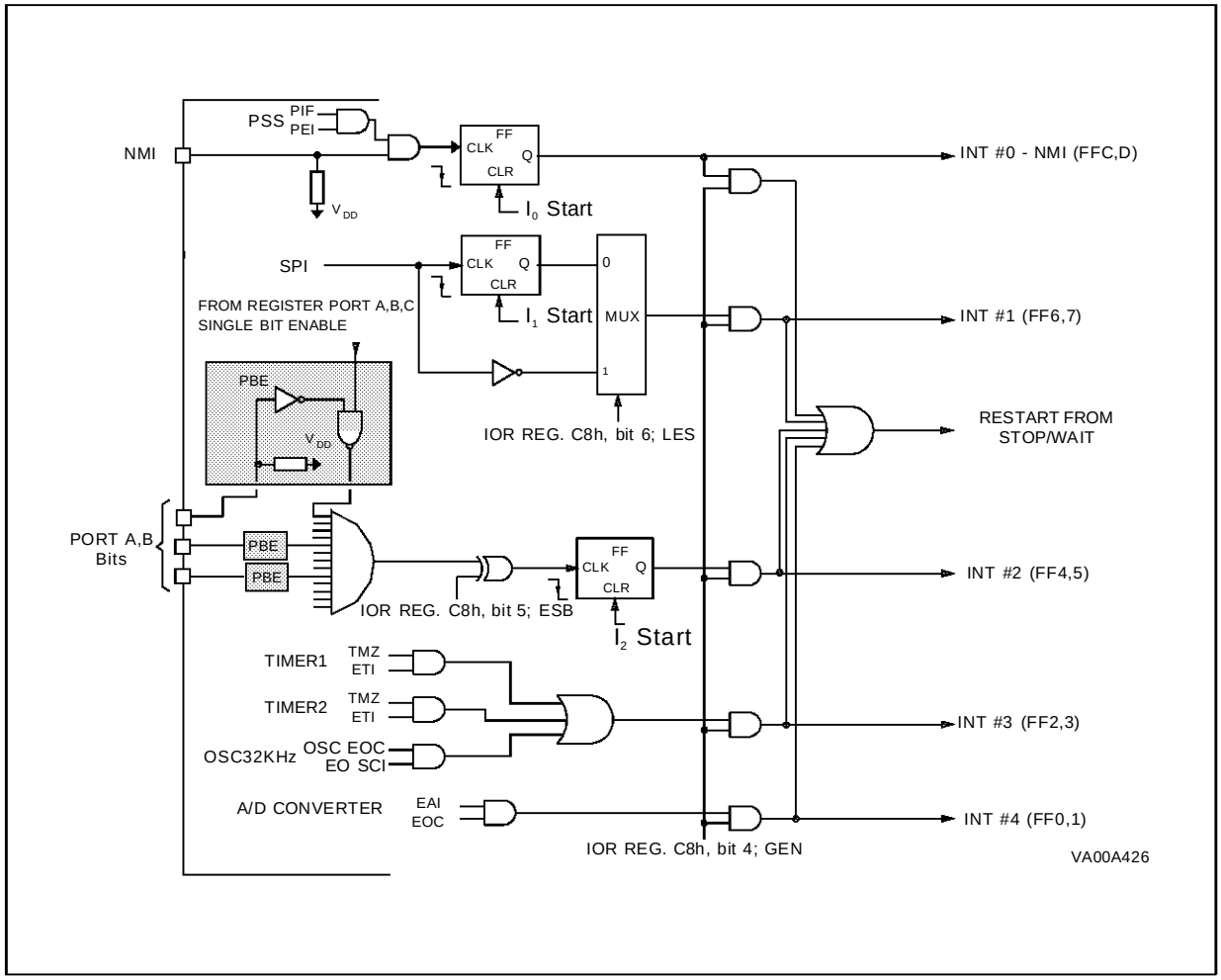
32kHz Oscillator, 320CR register location DBh

– EOSCI. This bit, when set, enables the 32kHz oscillator interrupt request.

– OSCEOC. This read only bit indicates when the 32kHz oscillator has measured a 500ms elapsed time (providing a 32.768kHz quartz crystal is connected to the 32kHz oscillator dedicated pins). An interrupt request can be generated in relation to the state of EOSCI bit.

INTERRUPTS (Continued)

Figure 16. ST6240 Interrupt Circuit Diagram



WARNING. *GEN is the global enable for all interrupts except NMI. If this bit is cleared, the NMI interrupt is accepted when the ST62xx core is in the normal RUN Mode.*
If the ST62xx core is in STOP or WAIT Mode, the NMI is not accepted as a restart is disabled. This state can only be finished by a reset (fromthe Watchdog or an external Reset Signal).
As a consequence the NMI can be masked in STOP and WAIT modes, but not in RUN mode.

RESET

The ST6240 can be reset in three ways: by the external reset input ($\overline{\text{RESET}}$) tied low, by power-on reset and by the digital watchdog/timer peripheral.

RESET Input

The $\overline{\text{RESET}}$ pin can be connected to a device of the application board in order to restart the MCU during its operation. The activation of the $\overline{\text{RESET}}$ pin may occur in the RUN, WAIT or STOP mode. This input has to be used to reset the MCU internal state and provide a correct start-up procedure. The pin is active low. The internal reset signal is generated by adding a delay to the external signal. Therefore even short pulses at the $\overline{\text{RESET}}$ pin will be accepted. This feature is valid providing that V_{DD} has finished its rising phase and the oscillator is correctly running (normal RUN or WAIT modes).

If $\overline{\text{RESET}}$ activation occurs in the RUN or Wait mode, the MCU is configured in the Reset mode for as long as the signal of the $\overline{\text{RESET}}$ pin is low. The processing of the program is stopped (in RUN mode only) and the Input/Outputs are in the High-impedance state with pull-up resistors switched on. As soon as the level on the $\overline{\text{RESET}}$ pin becomes high, the initialization sequence is executed.

If a $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts and all the inputs/outputs are configured in the High-impedance state with pull-up resistors switched on for as long as the level on the $\overline{\text{RESET}}$ pin remains low. When the level of the $\overline{\text{RESET}}$ pin becomes high, a delay is generated by the ST62xx core to wait that the oscillator becomes completely stabilized. Then, the initialization sequence is started.

Power-On Reset (POR)

The function of the POR consists in waking up the MCU during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: every Input/Output port is configured in the input mode (High-impedance state with pull-up) and no instruction is executed. When the power supply voltage becomes sufficient, the oscillator starts to operate, nevertheless the ST62xx core generates a delay to allow the oscillator to be completely stabilized before the execution of the first instruction. The initialization sequence is then executed.

Internal circuitry generates a Reset pulse when V_{DD} is switched on. In the case of fast rising V_{DD} (transition time $\leq 100\mu s$), this reset pulse starts the internal reset procedure without the need of external components at the $\overline{\text{RESET}}$ pin. In cases of slowly or non monotonously rising V_{DD}, an external reset signal must be provided for a proper reset of the MCU.

For as long as the reset pin is kept at the low level, the processor remains in the reset state. The reset will be released after the voltage at the reset pin reaches the high level.

Note:

To have a correct ST62xx start-up, the user should take care that the reset input does not change to the high level before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see recommended operating conditions).

An on-chip counter circuit provides a delay of 2048 oscillator cycles between the detection of the reset high level and the release of the MCU reset.

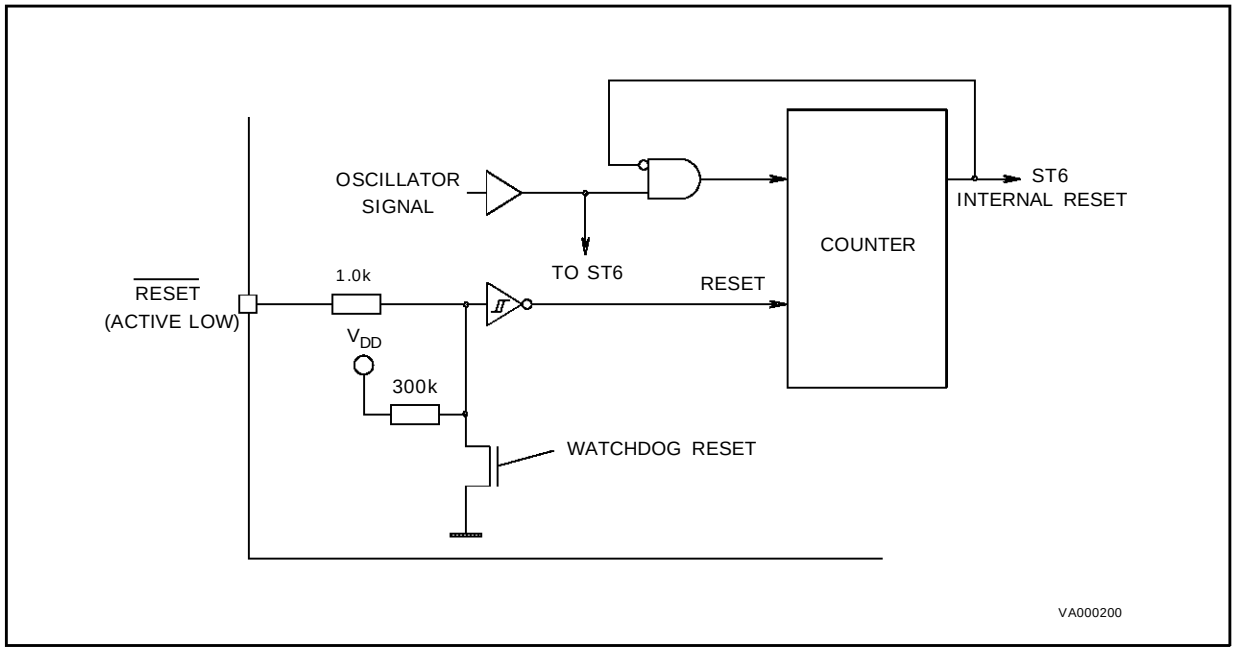
A proper reset signal for slow rising V_{DD}, i.e. the required delay between reaching sufficient operating voltage and the reset input changing to a high level, can be generally provided by an external capacitor connected between the $\overline{\text{RESET}}$ pin and V_{SS}.

RESET (Continued)

Watchdog Reset

The ST6240 provides an on-chip watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed, preventing the end-of-count being reached, an internal circuit pulls down the $\overline{\text{RESET}}$ pin. The MCU will enter the reset state as soon as the voltage at $\overline{\text{RESET}}$ pin reaches the related low level. This also resets the watchdog which subsequently turns off the pull-down and activates the pull-up device at the $\overline{\text{RESET}}$ pin. This causes the positive transition at the $\overline{\text{RESET}}$ pin and terminates the reset state.

Figure 17. Reset Circuit

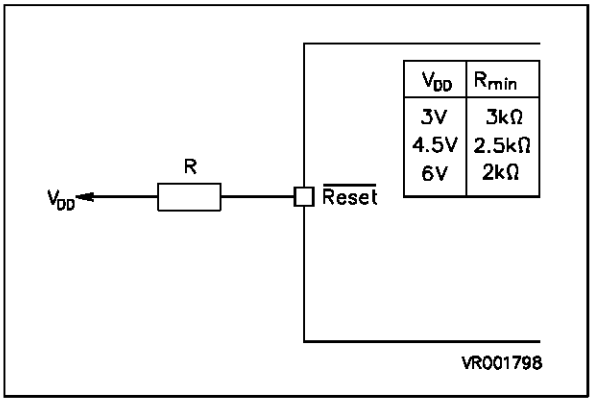


Application Notes

An external resistor between V_{DD} and reset pin is not required because an internal pull-up device is provided. If the user prefers, for any reason, to add an external pull-up resistor its value must comply with the Rmin value defined in Figure 18. If the value is lower than Rmin, the on-chip watchdog pull-down transistor might not be able to pull-down the reset pin resulting in an external deactivation of the watchdog function.

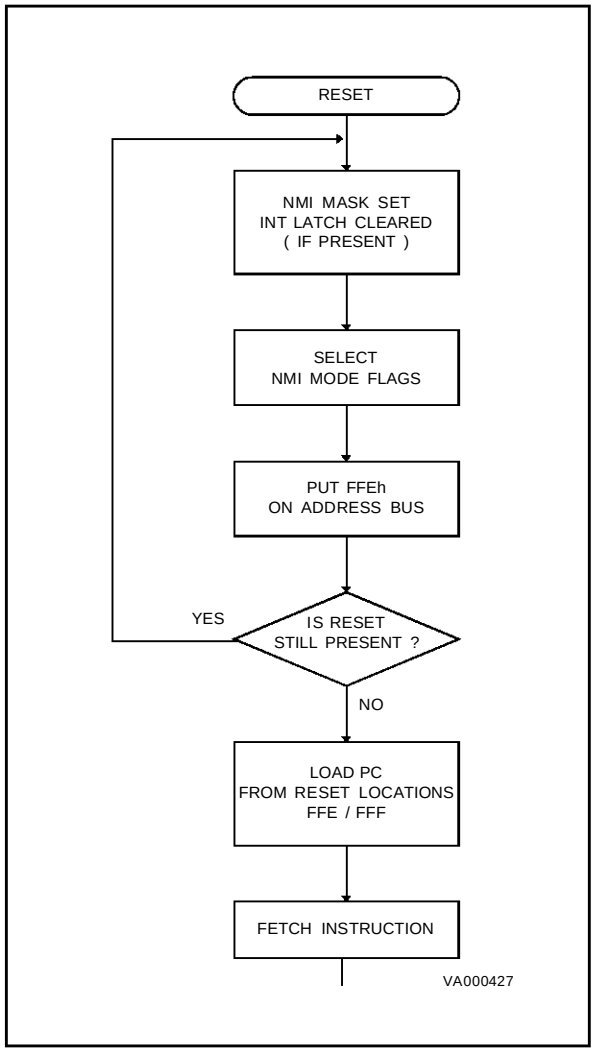
The POR function operates in a dynamic manner in the way that it brings about the initialization of the MCU when it detects a dynamic rising edge of the V_{DD} voltage. The typical detected threshold is about 2 volts, but the actual value of the detected threshold depends on the way in which the V_{DD} voltage rises up. The POR device DOES NOT allow the supervision of a static rising or falling edge of the V_{DD} voltage.

Figure 18. External Reset Resistance



RESET (Continued)

Figure 19. Reset & Interrupt Processing Flow-Chart



MCU Initialization Sequence

When a reset occurs the stack is reset to the program counter, the PC is loaded with the address of the reset vector (located in the program ROM at addresses FFEh & FFFh). A jump instruction to the beginning of the program has to be written into these locations. After a reset the interrupt mask is automatically activated so that the core is in non-maskable interrupt mode to prevent false or ghost interrupts during the restart phase. Therefore the restart routine should be terminated by a RETI instruction to switch to normal mode and enable interrupts. If no pending interrupt is present at the end of the reset routine the ST62xx will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced.

Figure 20. Restart Initialization Program Flow-Chart

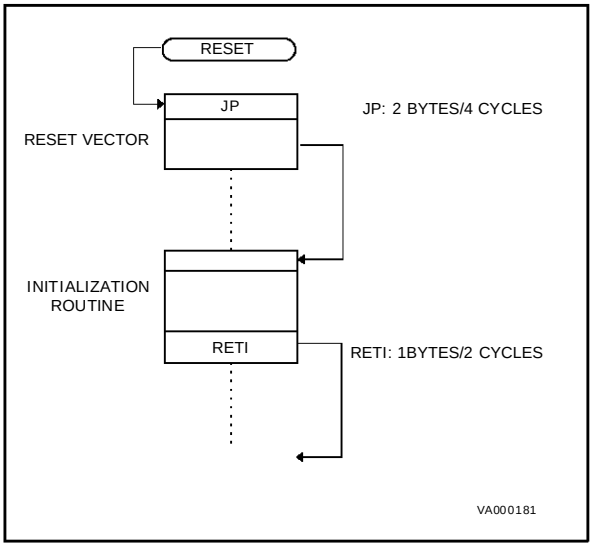


Table 7. Reset Configuration

Input/Output pins	Registers
Input Mode with pull-up and no interrupt	All cleared but A,X,Y,V,W, data RAM, LCD RAM, DWR (C9), PRPR (CA), DRBR (CB). Timers prescaler and TCR are initialized respectively at 7F and FF. Watchdog register DWDR (D8) is set to FEh.

WAIT & STOP MODES

The WAIT and STOP modes have been implemented in the ST62xx core in order to reduce the consumption of the product when the latter has no instruction to execute. These two modes are described in the following paragraphs

WAIT Mode

The configuration of the MCU in the WAIT mode occurs as soon as the WAIT instruction is executed. The microcontroller can also be considered as being in a “software frozen” state where the core stops processing the instructions of the routine, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage but where the peripherals are still working.

The WAIT mode is used when the user wants to reduce the consumption of the MCU when it is in idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide a clock signal to the peripherals. The timer counting may be enabled (writing the PSI bit in TSCR register) and the timer interrupt may be also enabled before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. The above explanation related to the timers applies also to the A/D converter.

If the exit from the WAIT mode is performed with a general RESET (either from the activation of the external pin or by watchdog reset) the MCU will enter a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behavior depends on the state of the ST62xx core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case will be described in the following paragraphs. In any case, the ST62xx core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

If the Watchdog is disabled the STOP mode is available. When in STOP mode the MCU is placed in the lowest power consumption mode. In this operating mode the microcontroller can be considered as being “frozen”, no instruction is executed, the oscillator is stopped, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or Reset activation to output from the STOP state.

If the exit from the STOP mode is performed with a general RESET (by the activation of the external pin) the MCU will enter a normal reset procedure as described in the RESET chapter. The case of an interrupt depends on the state of the ST62xx core before the initialization of the STOP sequence and also of the kind of the interrupt request that is generated.

This case will be described in the following paragraphs. In any case, the ST62xx core generates a delay after the occurrence of the interrupt request in order to wait the complete stabilization of the oscillator before the execution of the first instruction.

Exit from WAIT and STOP Modes

The following paragraphs describe the output procedure of the ST62xx core from WAIT and STOP modes when an interrupt occurs (not a RESET). It must be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) before the start of the WAIT or STOP sequence, but also of the type of the interrupt request that is generated.

Normal Mode. If the ST62xx core was in the main routine when the WAIT or STOP instruction has been executed, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs; the related interrupt routine is executed and at the end of the interrupt service routine the instruction that follows the STOP or the WAIT instruction is executed if no other interrupts are pending.

WAIT & STOP MODES (Continued)

Not Maskable Interrupt Mode. If the STOP or WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs: the instruction that follows the STOP or the WAIT instruction is executed and the ST62xx core is still in the non-maskable interrupt mode even if another interrupt has been generated.

Normal Interrupt Mode. If the ST62xx core was in the interrupt mode before the initialization of the STOP or WAIT sequence, it outputs from the stop or wait mode as soon as any interrupt occurs. Nevertheless, two cases have to be considered:

– If the interrupt is a normal interrupt, the interrupt routine in which the WAIT or STOP was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST62xx core is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance to their priority.

– If the interrupt is a non-maskable interrupt, the non-maskable routine is processed at first. Then the routine in which the WAIT or STOP was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST62xx core remains in the normal interrupt mode.

Notes :

To reach the lowest power consumption the user software must take care of:

– placing the A/D converter in its power down mode by clearing the PDS bit in the A/D control register before entering the STOP instruction.

– switching off the 32kHz oscillator by clearing the oscillator start/stop bit in the 32kHz oscillator control register.

– putting the EEPROM on-chip memory in stand-by mode by setting the E2OFF bit in EEPROM Control Register to one.

The LCD Driver peripheral is automatically switched-off by the STOP instruction when the 32kHz oscillator operation is not selected.

When the hardware activated watchdog is selected or the software watchdog enabled, the STOP instruction is deactivated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

If all the interrupt sources are disabled (including NMI if GEN="0"), the restart of the MCU can only be done by a $\overline{\text{RESET}}$ activation. The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

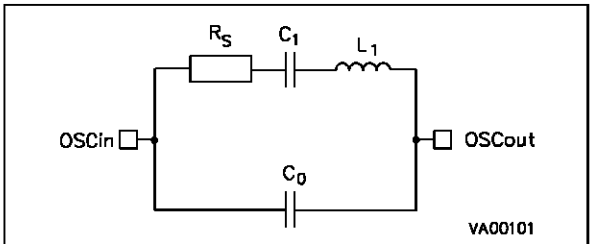
ON-CHIP CLOCK OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. A crystal, a ceramic resonator, or an external signal (provided to the OSCin pin) may be used to generate a system clock with various stability/cost tradeoffs. The different clock generator options connection methods are shown in Figure 22.

One machine cycle takes 13 oscillator pulses; 12 clock pulses are needed to increment the PC while and additional 13th pulse is needed to stabilize the internal latches during memory addressing. This means that with a clock frequency of 8MHz the machine cycle is 1.625µs.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance (CL), IC parameters, ambient temperature, supply voltage. It must be observed that the crystal or ceramic leads and circuit connections must be as short as possible. Typical values for CL1, CL2 are 15-22pF for a 4/8MHz crystal. The oscillator output frequency is internally divided by 13 to produce the machine cycle and by 12 to produce the Timer, the Watchdog and the A/D peripheral clock. A machine cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five machine cycles to be executed.

Figure 21. Crystal Parameters



Crystal parameters: AT-Cut Parallel Resonance Crystal
C0 = Parallel Resonance Capacitance

ON-CHIP CLOCK OSCILLATOR (Continued)

Figure 22. Oscillator Connection

CL1 = CL2 = 12 to 22pF for a 4/8MHz crystal (typical)

INPUT/OUTPUT PORTS

The ST6240 microcontroller has 16 Input/Output lines that can be individually programmed either in the input mode or the output mode with the following options that can be selected by software:

– Input without pull-up and without interrupt
– Input with pull-up and with interrupt
– Input with pull-up without interrupt
– Analog inputs (PA0-PA7, PB0-PB3)
– SPI control signals (PB5-PB7)
– Push-pull output
– Standard Open drain output
– 20mA Open drain output (PB4-PB7)

The lines are organized in two ports (port A,B).

Each port occupies 3 registers in the data space. Each bit of these registers is associated with a particular line (for instance, the bits 0 of the Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The two DATA registers (DRA, DRB), are used to read the voltage level values of the lines programmed in the input mode, or to write the logic value of the signal to be output on the lines configured in the output mode. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by the user software, in conjunction with the related option registers, to select the different input mode options.

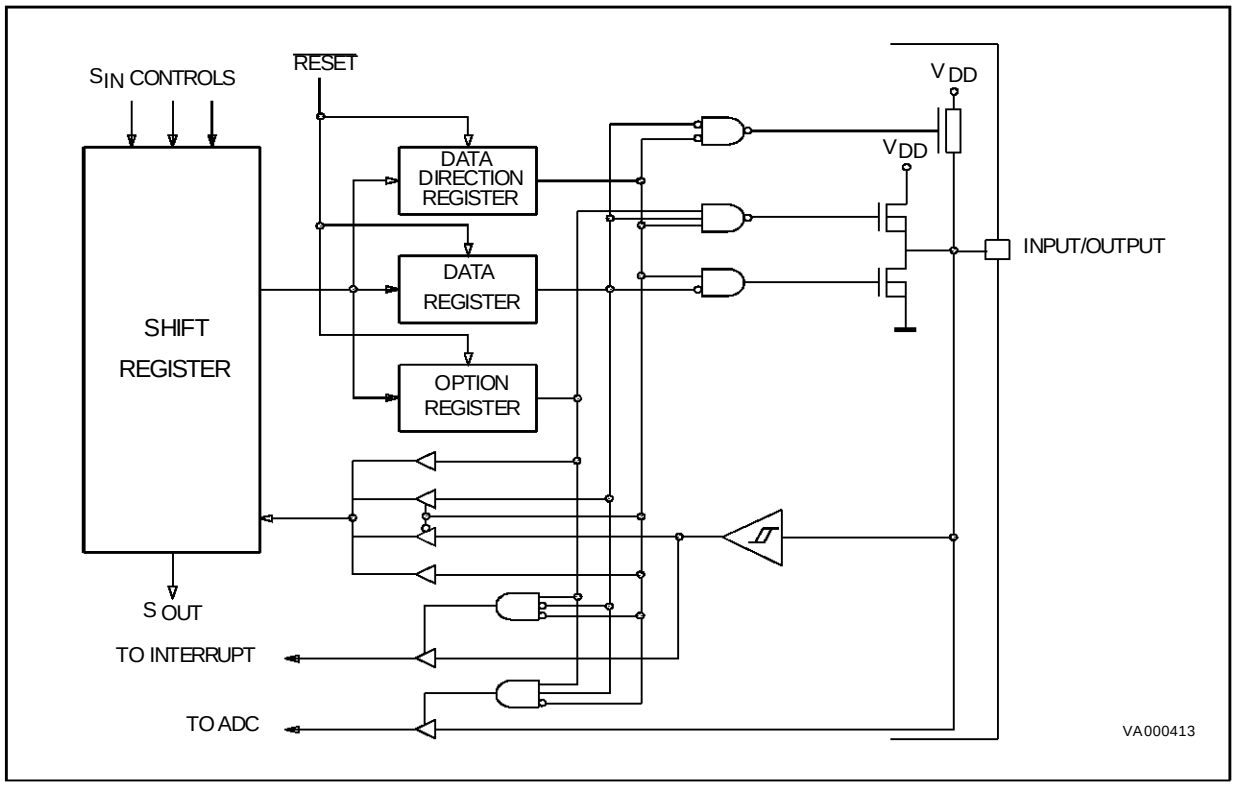
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The two Data Direction registers (DDRA, DDRB) allow the selection of the data direction of each pin (input or output).

The two Option registers (ORA, ORB) are used to select the different port options available both in input and in output mode.

All the I/O registers can be read or written as any other RAM location of the data space, so no extra RAM cell is needed for port data storing and manipulation. During the initialization of the MCU, all the I/O registers are cleared and the input mode with pull-up/no-interrupt is selected on all the pins, thus avoiding pin conflicts.

Figure 23. I/O Port Block Diagram



INPUT/OUTPUT PORTS (Continued)

I/O Pin Programming

Each pin can be individually programmed as input or output with different input and output configurations.

This is achieved by writing to the relevant bit in the data (DR), data direction register (DDR) and option registers (OR). Table 8 shows all the port configurations that can be selected by user software.

Input Option Description

Pull-up, High Impedance Option. All the input lines can be individually programmed with or without an internal pull-up according to the codes programmed in the OR and DR registers . If the pull-up option is not selected, the input pin is in the high-impedance state.

Interrupt Option. All the input lines can be individually connected by software to the interrupt lines of the ST62xx core according to the codes programmed in the OR and DR registers. The pins of Port A and B are "ORed" and are connected to the interrupt associated to the vector #2. The interrupt modes (falling edge sensitive, rising edge sensitive) can be selected by software for each port by programming the IOR register.

Analog Input Option. The twelve PA0-PA7, PB0-PB3 pins can be configured to be analog inputs according to the codes programmed in the OR and DR registers. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as analog input at a time, otherwise the selected inputs will be shorted.

.

Figure 24. I/O Port Data Registers

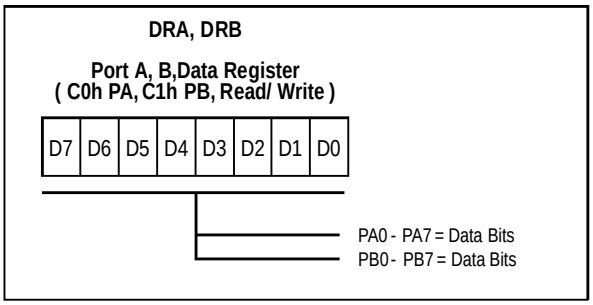


Figure 25. I/O Port Data Direction Registers

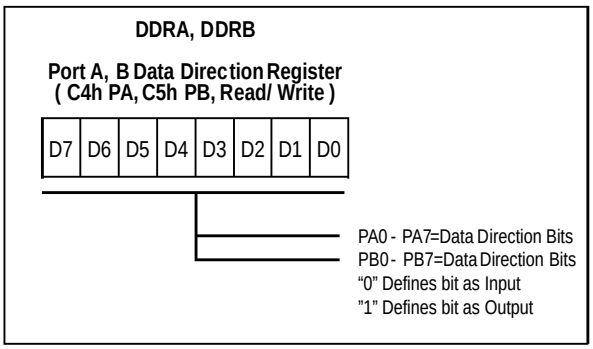
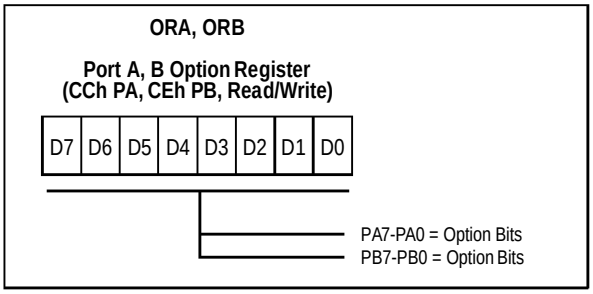


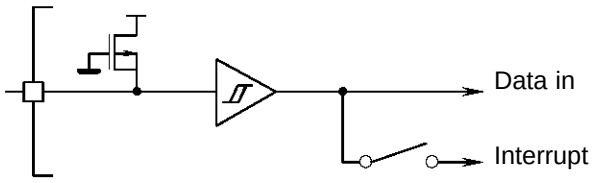
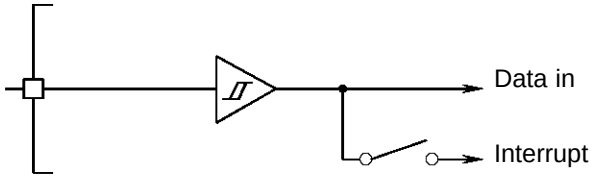
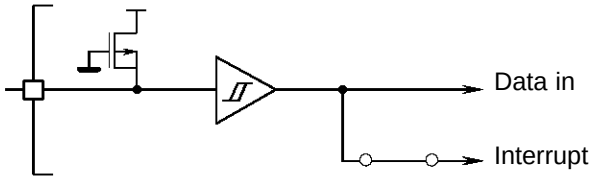
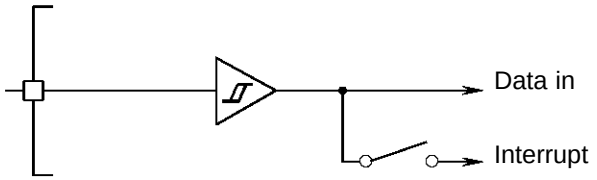
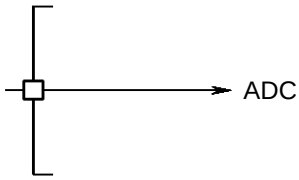
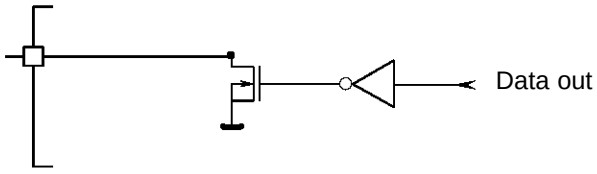
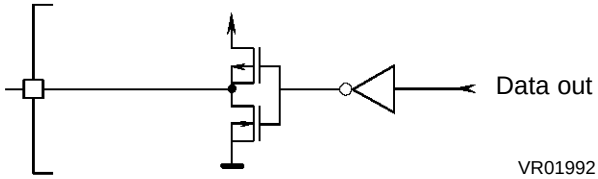
Figure 26. I/O Port Option Registers



Note: For complete coding explanation refer to Table 8.

INPUT/OUTPUT PORTS (Continued)

Table 8. I/O Port Options Selection

DDR	OR	DR	Mode	Option	Schematic
0	0	0	Input	Pull-up No interrupt (RESET state)	
0	0	1	Input	No pull-up No interrupt	
0	1	0	Input	Pull-up Interrupt	
0	1	1	Input	No pull-up No interrupt PB4-PB7	
			Input	Analog input: PA0-PA7 PB0-PB3	
1	0	X	Output	Open drain: 20mA PB4-PB7 Open drain: 5mA PB0-PB3, PA0-PA7	
1	1	X	Output	Push-pull	

INPUT/OUTPUT PORTS (Continued)

SPI alternate function Option. The I/O pins PB5-PB7 are also used by serial peripheral interface SPI. PB5 is connected with the SPI clock input SCL, PB6 is connected with the SPI data input SIN and PB7 is connected with the SPI data output SOUT.

For serial input operation PB5 and PB6 have to be programmed as inputs. For serial output operation PB7 has to be programmed as open-drain output (DDR = "1", OPR = "0"). In this operating mode the output of the SPI shift register instead of the port data register is connected to the port buffer. When PB7 is programmed as push-pull output (DDR = "1", OPR = "1"), the port data register is connected to the port buffer. When the SPI peripheral is not used PB5-PB7 can be used as general purpose I/O lines (provided that PB7 is not selected to be open-drain in output mode).

Notes:

Switching the I/O ports from one state to another should be done in a way that no unwanted side effects can happen. The recommended safe transitions are shown below. All other transitions are risky and should be avoided during change of operation mode as it is most likely that there will be an unwanted side-effect such as interrupt generation or two pins shorted together by the analog input lines.

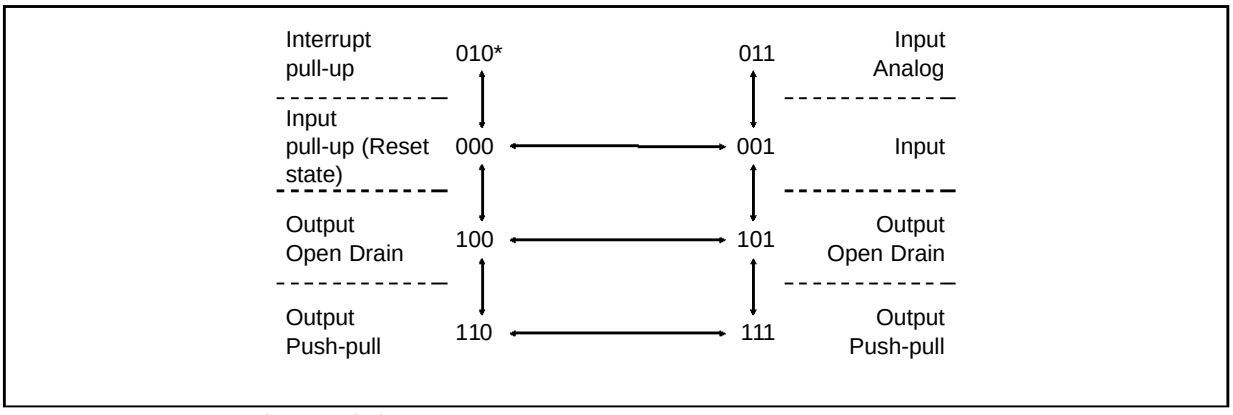
Single bit SET and RES instructions should be used very carefully with Port A and B data registers because these instructions make an implicit read and write back of the whole addressed register byte. In port input mode however data register address reads from input pins, not from data register latches and data register information in input mode is used to set characteristics of the input pin (interrupt, pull-up, analog input), therefore these characteristics may be unintentionally reprogrammed depending on the state of input pins. As general rule is better to use SET and RES instructions on data register only when the whole port is in output mode. If input or mixed configuration is needed it is recommended to keep a copy of the data register in RAM. On this copy it is possible to use single bit instructions, then the copy register could be written into the port data register.

```
SET     bit, datacopy
LD      a, datacopy
LD      DRA, a
```

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user has to take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance in the measurement.

Figure 27. I/O Port StateTransition Diagram for Safe Transitions



Note *. xxx = DDR, OR, DR Bits respectively

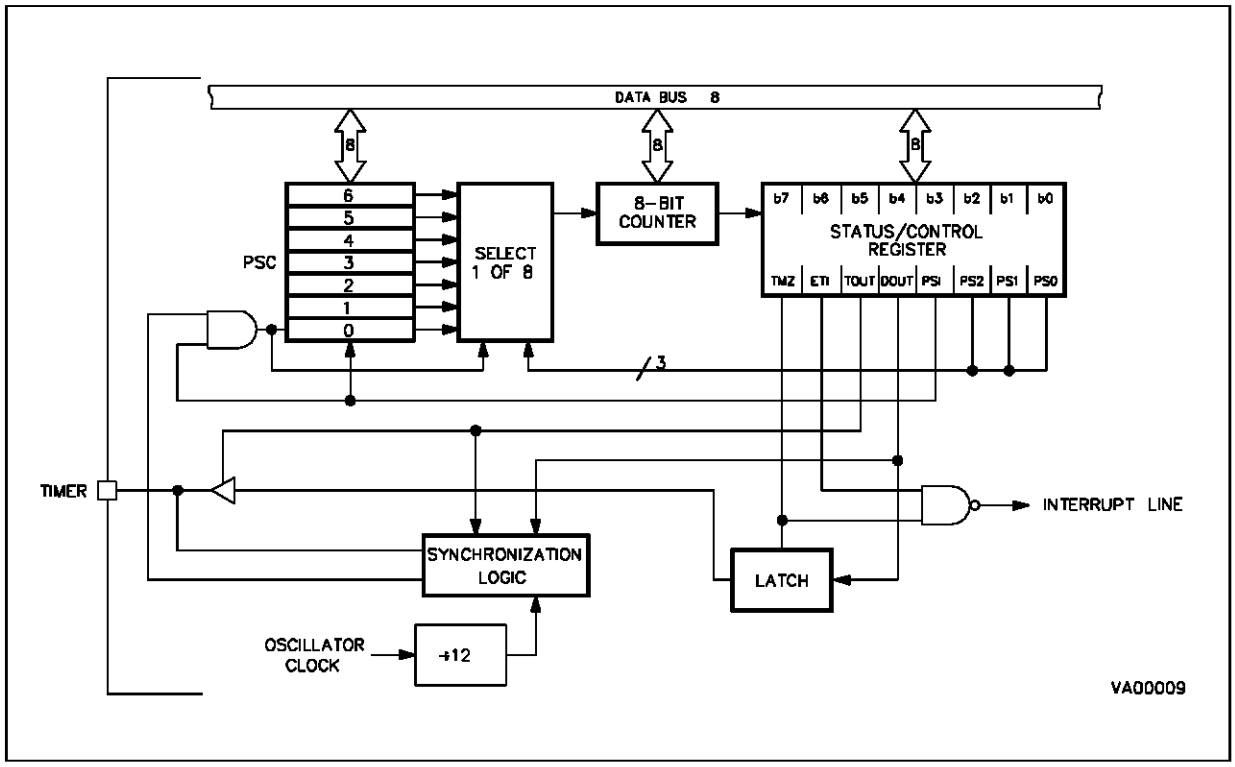
TIMERS

The ST6240 offers two on-chip Timer peripherals named Timer 1 and Timer 2. Each of these timers consists of an 8-bit counter with a 7-bit programmable prescaler, thus giving a maximum count of 2^{15}, and control logic that allows configuring the peripheral in three operating modes. Figure 28 shows the Timer block diagram. Timer 1 only has the external TIMER pin available for the user. The content of the 8-bit counter can be read/written in the Timer/Counter register TCR which is addressed in the data space as a RAM location at addresses D3h or D6h. The state of the 7-bit prescaler can be read in the PSC register at addresses D2h or D5h. The control logic device is managed in the TSCR register (addresses D4h or D7h) as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero) bit in the TSCR is set to one. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set to one an interrupt request, associated to interrupt vector #3, is generated. The interrupt service routine then would determine which timer reached the end of count by poling the TMZ bits. The Timer interrupt can be used to exit the MCU from the WAIT mode.

The prescaler input can be the oscillator frequency divided by 12 or an external clock at TIMER pin (only Timer 1). The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR, the clock input of the timer/counter register is multiplexed to different sources. On division factor 1, the clock input of the prescaler is also that of timer/counter; on factor 2, bit 0 of prescaler register is connected to the clock input of TCR. This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC is connected to clock input of TCR, and so on. The prescaler initialize bit (PSI) in the TSCR register must be set to one to allow the prescaler (and hence the counter) to start. If it is cleared to zero then all of the prescaler bits are set to one and the counter is inhibited from counting. The prescaler can be given any value between 0 and 7Fh by writing to addresses D2h or D5h, if bit PSI in the TSCR register is set to one. The tap of the prescaler is selected using the PS2,PS1,PS0 bits in the control register. Figure 29 shows the Timer working principle.

Figure 28. Timer Peripheral Block Diagram



TIMERS (Continued)

Timer Operating Modes

There are 3 operating modes of the Timer peripheral. They are selected by the bits TOUT and DOUT (see TSCR register). These three modes correspond to the two clock frequencies that can be connected on the 7-bit prescaler (f_{OSC}/12 or TIMER pin signal) and to the output mode. For this reason, only Timer 1 has all three modes, while Timer 2, which does not have a dedicated TIMER pin, must be programmed in Output Mode only.

Clock Input Mode (TOUT = "0", DOUT = "0"). In this mode the TIMER pin is an input and the prescaler is decremented on rising edge. The maximum input frequency that can be applied to the external pin in this mode is 1/8 of the oscillator frequency. This operating mode is not available on Timer 2.

Gated Mode (TOUT = "0", DOUT = "1"). In this mode the prescaler is decremented by the Timer clock input (oscillator divided by 12) but ONLY when the signal at TIMER pin is held high (giving a pulse width measurement potential). This mode is selected by the TOUT bit in TSCR register cleared to "0" (i.e. as input) and DOUT bit set to "1". This operating mode is not available on Timer 2.

Output Mode (TOUT = "1", DOUT = data out). The TIMER pin is connected to the DOUT latch. Therefore the timer prescaler is clocked by the prescaler clock input (f_{OSC}/12).

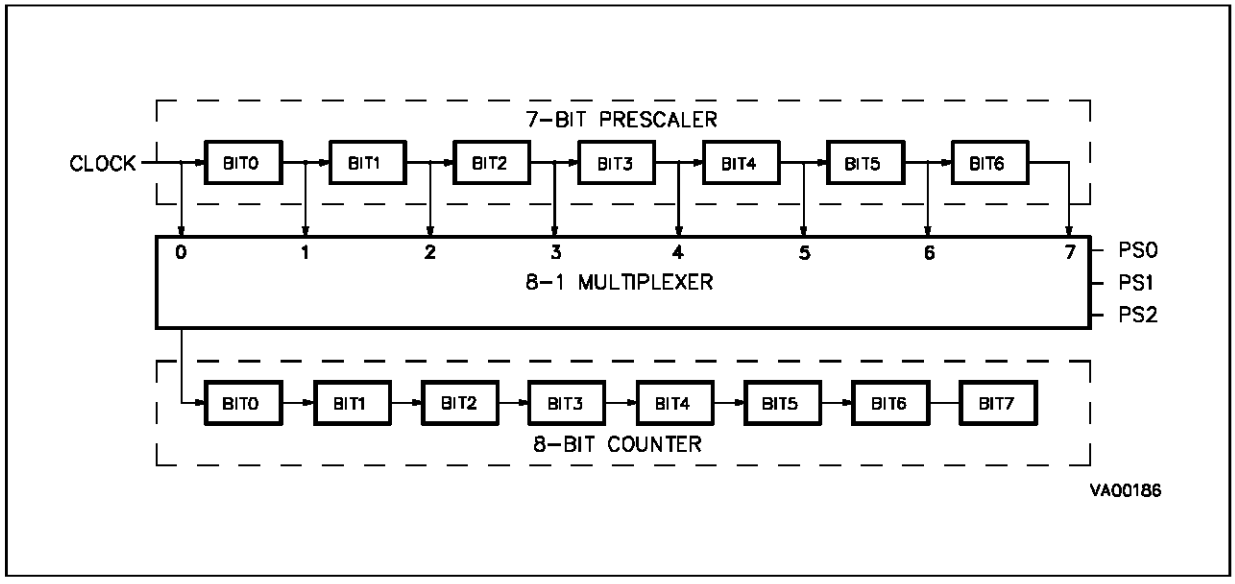
The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The low-to-high TMZ bit transition is used to latch the DOUT bit of the TSCR and pass it to TIMER pin. This operating mode allows external signal generation on the TIMER pin. This is the only operating Mode allowed on Timer 2.

Table 9. Timer Operating Modes

TOUT	DOUT	Timer Pin	Timer Function
0	0	Input	Event Counter (1)
0	1	Input	Input Gated (1)
1	0	Output	Output
1	1	Output	Output

Note 1. Not allowed on Timer 2

Figure 29. Timer Working Principle



TIMERS (Continued)

Timer Interrupt

When the counter register decrements to zero and the software controlled ETI (Enable Timer Interrupt) bit is set to one then an interrupt request associated to interrupt vector #3 is generated. When the counter decrements to zero also the TMZ bit in the TSCR register is set to one.

Since only one interrupt vector is available for the two timers (ORed also with 32kHz oscillator interrupt), the interrupt service routine should determine from which source the interrupt came by polling the TMZ bits (and the OSCEOC bit of the 32kHz oscillator control register).

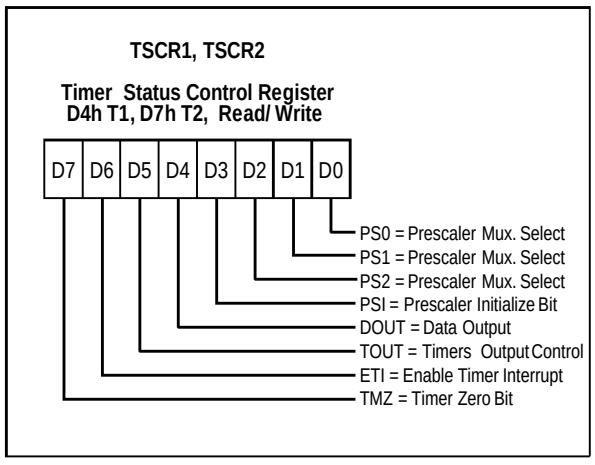
Notes:

TMZ is set when the counter reaches 00h; however, it may be set by writing 00h in the TCR register or setting bit 7 of the TSCR register. TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFh while the 7-bit prescaler is loaded to 7Fh , and the TSCR register is cleared which means that timer is stopped (PSI="0") and the timer interrupt is disabled.

If the Timer is programmed in output mode, DOUT bit is transferred to the TIMER pin when TMZ is set to one (by software or due to counter decrement). When TMZ is high, the latch is transparent and DOUT is copied to the timer pin. When TMZ goes low, DOUT is latched.

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

Figure 30. Timer Status Control Register



TMZ. Low-to-high transition indicates that the timer count register has decremented to zero. This bit must be cleared by user software before starting with a new count.

ETI. This bit, when set, enables the timer interrupt request (vector #3). If ETI="0" the timer interrupt is disabled. If ETI="1" and TMZ="1" an interrupt request is generated.

TOUT. When low, this bit selects the input mode for the TIMER pin. When high the output mode is selected.

DOUT. Data sent to the timer output when TMZ is set high (output mode only). Input mode selection (input mode only). This bit is meaningless for Timer 2.

PSI. Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

PS2, PS1, PS0. These bits select the division ratio of the prescaler register.

Table 10. Prescaler Division Factors

PS2	PS1	PS0	Divided by	PS2	PS1	PS0	Divided by
0	0	0	1	1	0	0	16
0	0	1	2	1	0	1	32
0	1	0	4	1	1	0	64
0	1	1	8	1	1	1	128

Figure 31. Timer Counter Register

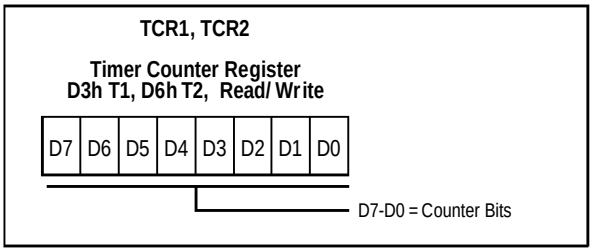
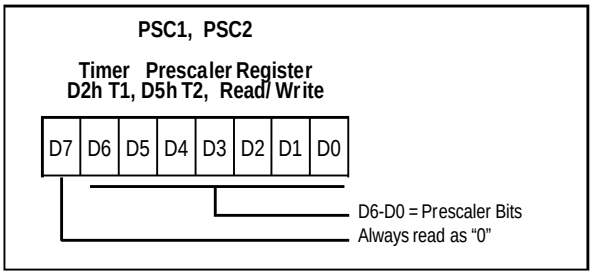


Figure 32. Prescaler Register



DIGITAL WATCHDOG

The digital Watchdog of the ST62 device consists of a down counter that can be used to provide a controlled recovery from a software upset.

The ST6240 watchdog has two watchdog options, the software activated watchdog/timer and the hardware activated watchdog function. The user can select one of the two options by connecting the pin WDON to V_{DD} (software watchdog) or V_{SS} (hardware watchdog). If the pin is kept non connected, an internal pull-up resistance selects the software watchdog option.

Hardware Watchdog Function

The hardware activated digital watchdog function consists of a down counter that is automatically initialized after reset so that this function does not need to be activated by the user program. As the watchdog function is always activated this down counter cannot be used as a timer. The watchdog uses one data space register (DWDR location D8h). The watchdog register is set to FFh on reset and immediately starts to count down, requiring no software start. Similarly the hardware activated watchdog cannot be stopped or delayed by software.

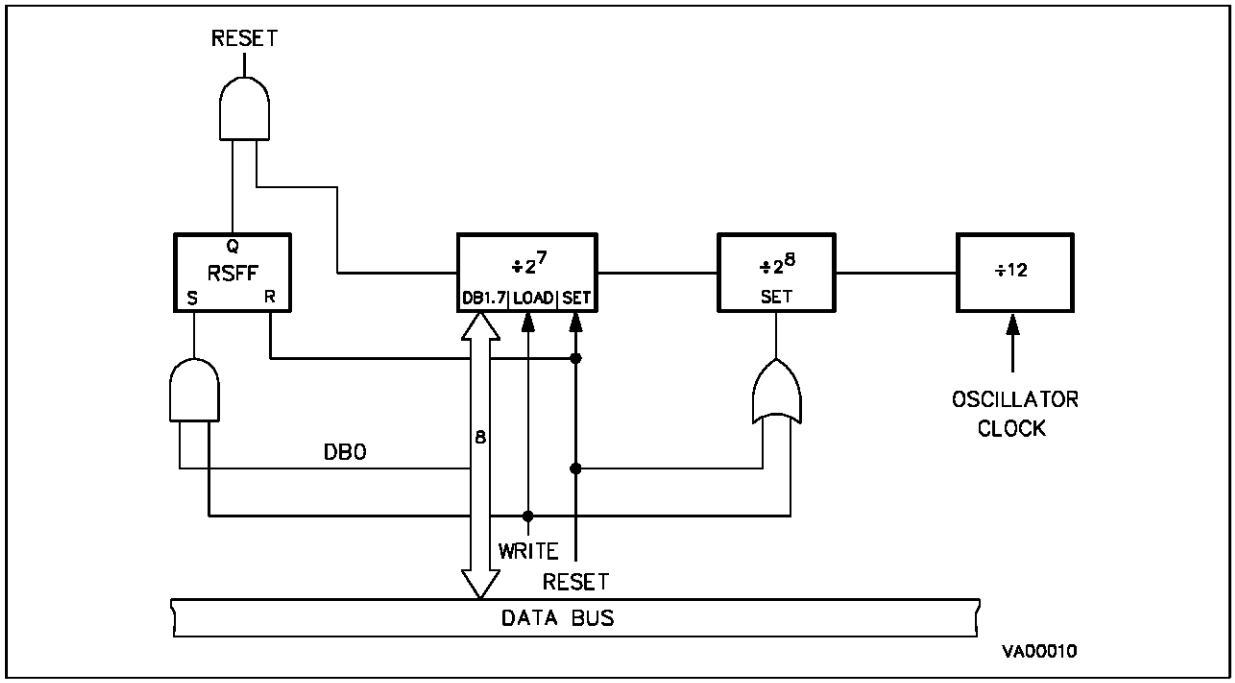
The watchdog time can be programmed using the 6 Most Significant bits in the watchdog register, this gives the possibility to generate a reset in a time between 3072 to 196608 clock cycles in 64 possible steps (with a clock frequency of 8MHz, this means from 384µs to 24.576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones. The check time can be set differently for different routines within the general program. The presence of the hardware watchdog deactivates the STOP instruction and a WAIT instruction is automatically executed instead of a STOP. Bit 1 of the watchdog register (set to one at reset) can be used to generate a software reset if cleared to zero.

Software Watchdog

The software activated digital watchdog consists of a down counter that can be used to provide a controlled recovery from a software upset. The watchdog uses one data space register (DWDR location D8h). The watchdog register is set to FEh after reset and the watchdog function is disabled. The watchdog time can be programmed using the 6 Most Significant Bits in the Watchdog register. The check time can be set differently for different routines within the general program.

After a reset the software Watchdog is in the off-state. The watchdog should be activated inside the Reset restart routine by writing a “1” in watchdog timer register bit 0. Bit one of this register must be

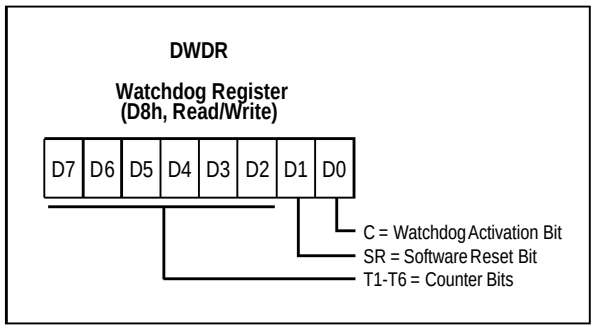
Figure 33. Digital Watchdog Block Diagram



DIGITAL WATCHDOG (Continued)

set to one before programming bit zero as otherwise a reset will be immediately generated when bit 0 is set. This allows the user to generate a reset by software (bit 0 = "1", bit 1 = "0"). Once bit 0 is set, it can not be cleared by software without generating a Reset. The delay time is defined by programming bits 2-7 of the watchdog register. Bit 7 is the Least Significant Bit while bit 2 is the MSB. This gives the possibility to generate a reset in a time between 3072 to 196608 clock cycles in 64 possible steps: (With a clock frequency of 8MHz this means from 384µs to 24.576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones. If the watchdog is active the STOP instruction is deactivated and a WAIT instruction is automatically executed instead of a STOP. If bit 0 of the watchdog register is never set to one then bits 1-7 of the register can be used as a simple 7-bit counter which is decrement every 3072 clock cycles.

Figure 35. Watchdog Register

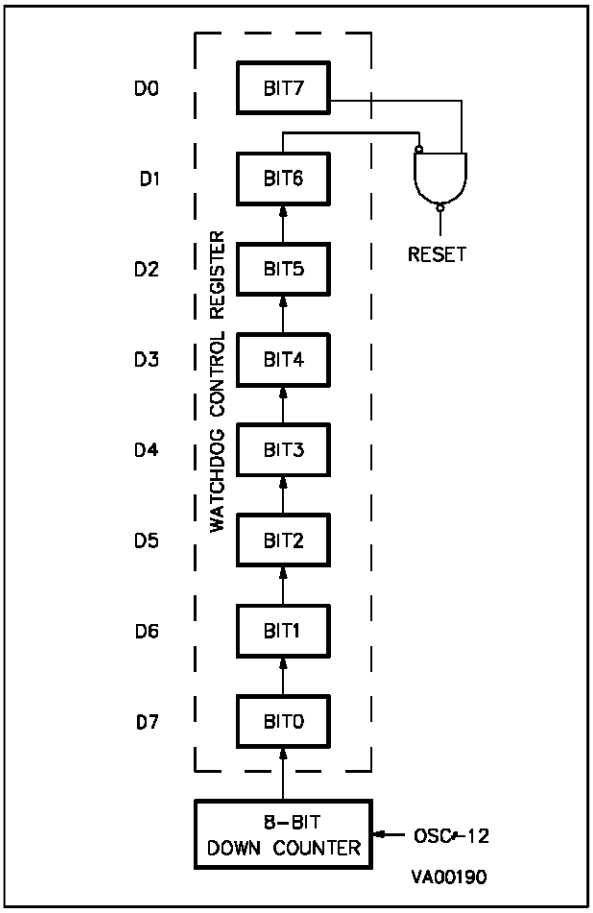


C. This is the watchdog activation bit, that, if set to one, will activate the watchdog function. When cleared to zero it allows the use of the counter as a 7-bit timer. If the user selects the hardware function this bit is automatically set on reset and the user cannot change its state. When the software function is selected, this bit is cleared on reset.

SR. This bit is set to one during the reset and will generate a software reset if cleared to zero. When C = "0" (watchdog disabled software option) it is the MSB of the 7-bit timer.

T1-T6. These are the watchdog counter bits. It should be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter. These bits are in the opposite order to normal.

Figure 34. Watchdog Working Principle



DIGITAL WATCHDOG (Continued)

Application note:

The hardware activation function is very useful when the external circuitry may inject noise on the reset pin, where there is an unstable supply voltage, or RF influence or other similar phenomena.

If the Watchdog software activation is selected and the Watchdog is not used during power-on reset external noise may cause the undesired activation of the Watchdog with a generation of an unexpected reset. To avoid this risk, two additional instructions, that check the state of the watchdog and eventually reset the chip are needed within the first 27 instructions, after the reset. These instructions are:

```
jrx 0, WD, #+3
ldi WD, 0FDH
```

These instructions should be executed at the very beginning of the customer program.

If the Watchdog is used (both hardware or software activated), during power-on reset the Watchdog register may be set to a low value, that could give a reset after 28 instructions earliest. To avoid undesired resets, the Watchdog must be set to the desired value within the first 27 instructions, the best is to put at the very beginning.

Alternatively the normal legal state can be checked with the following short routine:

```
ldi a, 0FEH
and a, WD
cpi a, 0FEH
jrz #+3
ldi WD, 0FDH
```

This sequence is recommended for security applications, where possible stack confusion error loops must be avoided and the Watchdog must only be refreshed after extensive checks.

8-BIT A/D CONVERTER

The A/D converter of ST6240 is an 8-bit analog to digital converter with up to 12 analog inputs (as alternate functions of I/O lines PA0-PA7, PB0-PB3) offering 8-bit resolution with total accuracy ±2 LSB and a typical conversion time of 70µs (clock frequency of 8MHz).

The A/D peripheral converts the input voltage by a process of successive approximations using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, the A/D converter accuracy is decreased.

The selection of the pin signal that has to be converted is done by configuring the related I/O line as analog input through the I/O ports option and data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as analog input at a time. The user must avoid the situation in which more than one I/O pin is selected to be analog input to avoid malfunction of the ST62xx.

The ADC uses two registers in the data space: the ADC data conversion register which stores the conversion result and the ADC control register used to program the ADC functions.

A conversion is started by writing a “1” to the Start bit (STA) in the ADC control register. This automatically clears (resets to “0”) the End Of Conversion Bit (EOC). When a conversion has been finished this EOC bit is automatically set to “1” in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

The STA bit is continually being scanned so that if the user sets it to “1” while a previous conversion is in progress then a new conversion is started before the previous one has been completed.The

8-BIT A/D CONVERTER(Continued)

start bit (STA) is a write only bit, any attempt to read it will show a logical "0".

The A/D converter has a maskable interrupt associated to the end of conversion. This interrupt is associated to the interrupt vector #4 and occurs when the EOC bit is set, i.e. when a conversion is completed. The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. That is achieved when the PDS bit in the ADC control register is cleared to "0". If PDS="1", the A/D is supplied and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow the stabilization of the A/D converter.*This action is needed also before entering the STOP instruction as the A/D comparator is not automatically disabled by the STOP mode*

During reset any conversion in progress is stopped, the control register is reset to all zeros and the A/D interrupt is masked (EAI=0).

Figure 37. A/D Converter Block Diagram

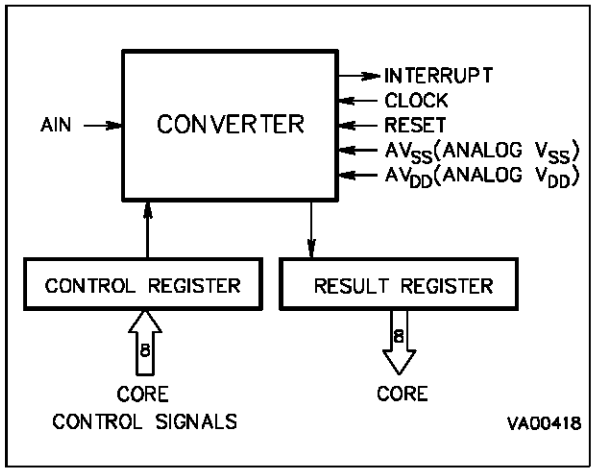
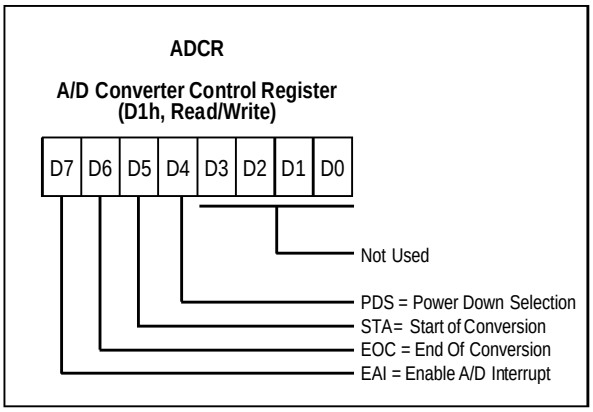


Figure 36. A/D Converter Control Register



EAI. If this bit is set to one the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

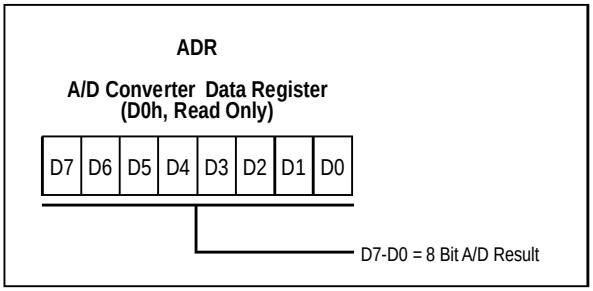
EOC. *Read Only*;This read only bit indicates when a conversion has been completed. This bit is automatically reset to zero when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to one.

STA. *Write Only*; Writing a "1" in this bit will start a conversion on the selected channel and automatically reset to zero the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

PDS. This bit activates the A/D converter if set to 1. Writing a zero into this bit will put the ADC in power down mode (idle mode).

D3-D0. Not used

Figure 38. A/D Converter Data Register



D7-D0. *Read Only*;These are the conversion result bits; the register is read only and stores the result of the last conversion. The contents of this register are valid only when EOC bit in the ADCR register is set to one (end-of-conversion).

8-BIT A/D CONVERTER(Continued)

Notes:

The ST62xx A/D converter does not feature a sample and hold. The analog voltage to be measured should therefore be stable during the conversion time. Variation should not exceed ±1/2 LSB for the best accuracy in measurement.

Since the ADC is on the same chip as the microprocessor the user should not switch heavily loaded output signals during conversion if high precision is needed. This is because such switching will affect the supply voltages which are used for comparisons.

A low pass filter can be used at the analog input pins to reduce input voltage variation during the conversion. For true 8 bit conversions the impedance of the analog voltage sources should be less than 30kΩ while the impedance of the reference voltage should not exceed 2kΩ.

The accuracy of the conversion depends on the quality of the power supply voltages (V_{DD} and V_{SS}). The user must specially take care of applying regulated reference voltage on the V_{DD} and V_{SS} pins (the variation of the power supply voltage must be inferior to 5V/ms).

The converter can resolve the input voltage with an resolution of:

$$\frac{V_{DD} - V_{SS}}{256}$$

So if operating with a supply voltage of 5V the resolution is about 20mV.

The Input voltage (Ain) which has to be converted must be constant for 1µs before conversion and remain constant during the conversion.

The resolution of the conversion can be improved if the power supply voltage (V_{DD}) of the microcontroller becomes lower.

In order to optimize the resolution of the conversion, the user can configure the microcontroller in the WAIT mode because this mode allows the minimization of the noise disturbances and the variations of the power supply voltages due to the switching of the outputs. Nevertheless, it must be take care of executing the WAIT instruction as soon as possible after the beginning of the conversion because the execution of the WAIT instruction may provide a small variation of the V_{DD} voltage (the negative effect of this variation is minimized at the beginning of the conversion because the latter is less sensitive than the end of the conversion when the less significant bits are determined).

The best configuration from a accuracy point of view is the WAIT mode with the Timer and LCD driver stopped. Indeed, only the ADC peripheral and the oscillator are still working. The MCU has to be wake-up from the WAIT mode by the interrupt of the ADC peripheral at the end of the conversion. It must be noticed that the wake-up of the microcontroller could be done also with the interrupt of the TIMER, but in this case, the Timer is working and some noise could disturb the converter in terms of accuracy.

POWER SUPPLY SUPERVISOR DEVICE (PSS)

The Power Supply Supervisor device, described in the Figure 39, permits supervising the crossing of the PSS pin voltage (VPSS) through a programmable voltage (mxV_{DD}/n), where n and m can be chosen by software. This device includes:

– An internal comparator which is connected to the internal INT line to make an interrupt request to the Core.
– 2 resistive voltage dividers that are, respectively, supplied by the PSS pin and the V_{DD} pin. These two voltage dividers are both connected to the two inputs of the internal comparator. They consist of 13 identical resistors. It is possible to select by software 5 voltage rates on the PSS divider ($nxVPSS/13$) and 4 voltage rates on the V_{DD} divider ($mxV_{DD}/13$). The n and m values can be chosen by software. These two voltage dividers are disconnected in STOP mode, and when the PSS device is OFF.
– An internal device that allows the detection with an hysteresis of $V_{DD}/13$.

The PSS device is supplied by an internal connection to V_{DD} supply. The following paragraphs describe the operating mode of the PSS device and the PSS register that permits control over the PSS device. The PSS device is switched off as soon as the Core executes the STOP instruction, but continues to work in the WAIT mode.

Figure 40. PSS Device Operating Modes Description

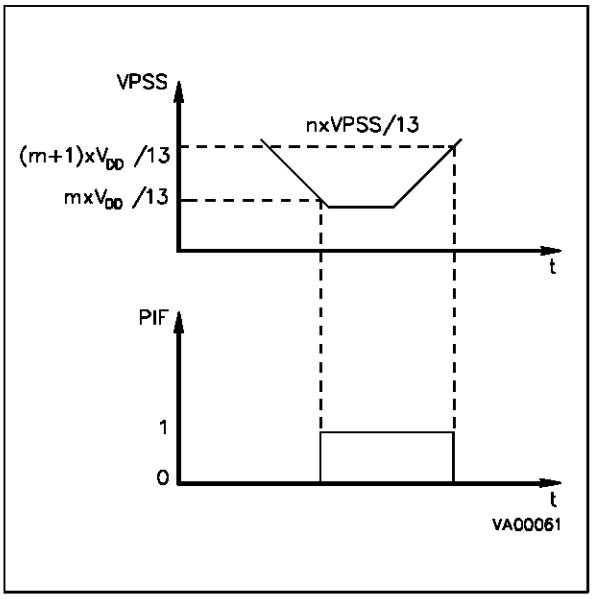
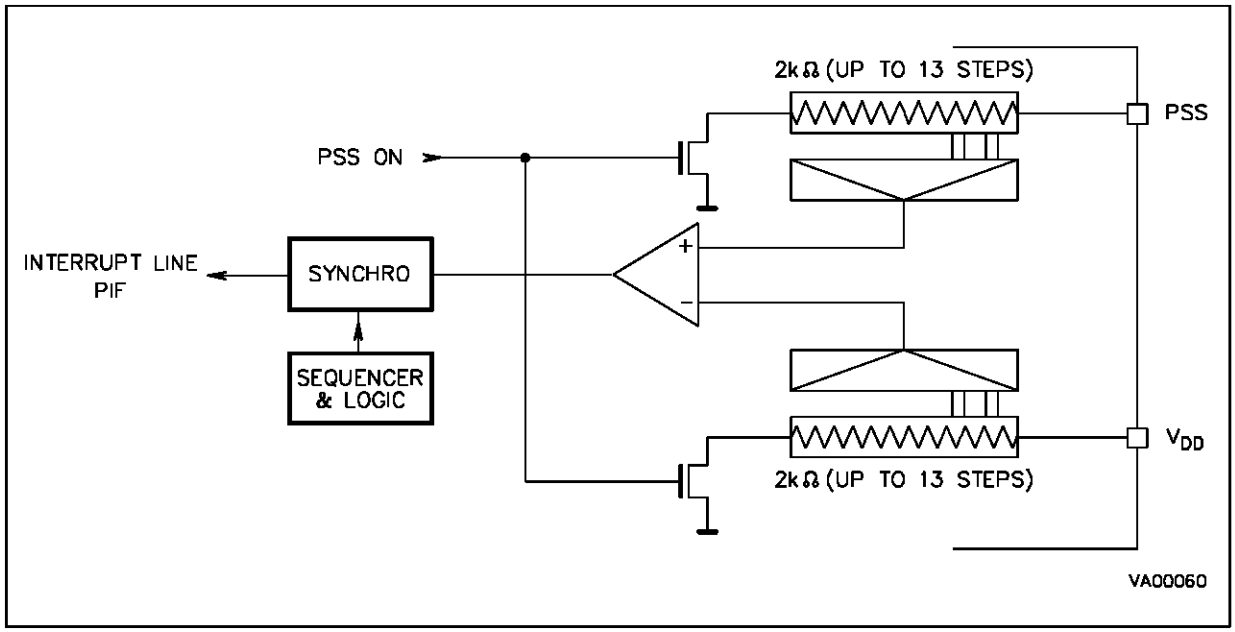


Figure 39. PSS Device Block Diagram



POWER SUPPLY SUPERVISOR(Continued)

PSS Operating Mode Description

The resistive voltage divider connected to the PSS pin provides the internal comparator with the $nxV_{PSS}/13$ voltage. The resistive voltage divider connected to the V_{DD} pin provides the internal comparator with the $mxV_{DD}/13$ voltage. The n and m values are selected with the PSS register. It must be observed that the n and m values must be selected, taking into consideration the following electrical constraints:

$0.5V < nxV_{PSS}/13$ at detection $< V_{DD} - 2V$

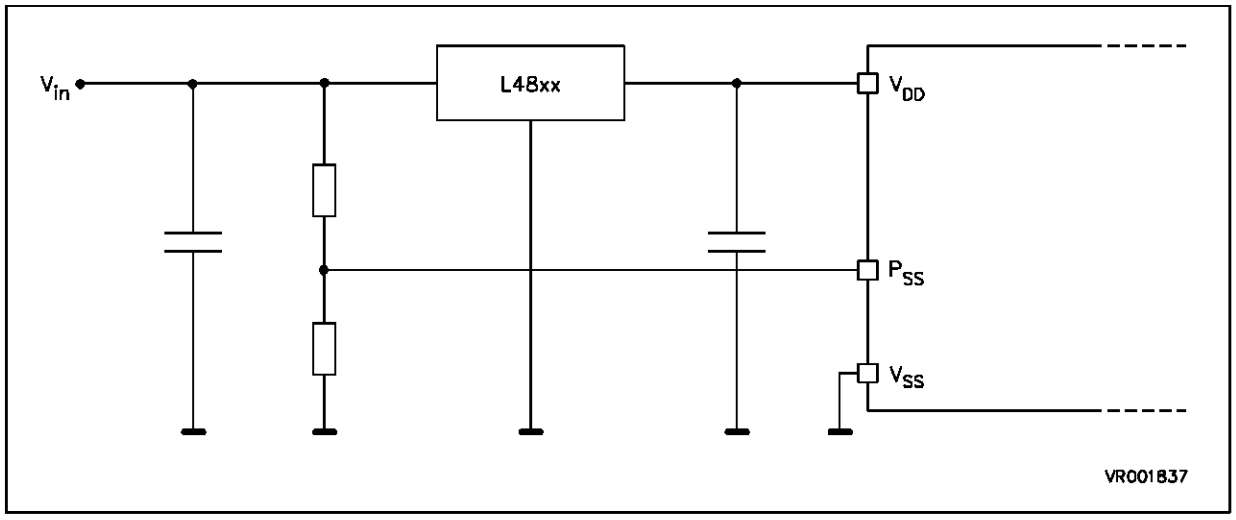
$0.5V < mxV_{DD}/13$ at detection $< V_{DD} - 2$ V

we must also have:

$$\frac{m}{n} V_{DD} \leq V_{PSS} \leq V_{DD}$$

The PIF bit is the interrupt request flag of the PSS device. This bit follows PSS comparator output.

Figure 41. Typical application using the PSS

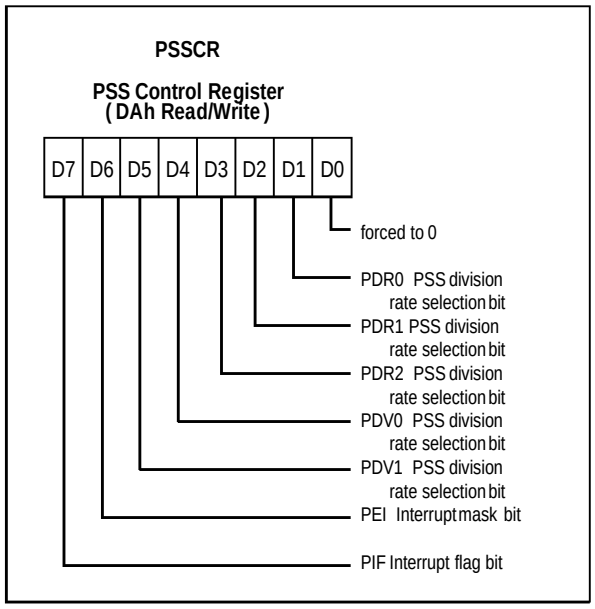


POWER SUPPLY SUPERVISOR(Continued)

PSS Register

The PSS register permits control over the PSS device. The register can be addressed in the data space as a RAM location at DAh. This register is cleared after Reset.

Figure 42. PSS Status Control Register



PIF. This bit is the interrupt flag. This bit is set (resp. cleared) as soon as the equality between nxVPSS and $(m+1)xV_{DD}/13$ (resp. $mxV_{DD}/13$) occurs.

PEI. This bit is the authorization bit of the interrupt request:

– If PEI is set, the interrupt request can reach the Core.

– If PEI is cleared, the interrupt request cannot reach the Core.

PDV1, PDV0. The PDV1/0 bits are used to select the rate of division of the V_{DD} voltage ($mxV_{DD}/13$ or $(m+1)xV_{DD}/13$, according to the hysteresis).

Table 12. V_{DD} Voltage division rate selection bits

PDV1	PDV0	$mxV_{DD}/13$	$(m+1)xV_{DD}/13$
0	0	$3xV_{DD}/13$	$4xV_{DD}/13$
0	1	$5xV_{DD}/13$	$6xV_{DD}/13$
1	0	$6xV_{DD}/13$	$7xV_{DD}/13$
1	1	$7xV_{DD}/13$	$8xV_{DD}/13$

PDR2, PDR1, PDR0. The PDR2/1/0 bits are used to inhibit the PSS device and to select the division rate of the PSS voltage (nxVPSS/13).

The PSS comparator output is valid 8 cycle times after the programming of the PDR2/1/0 bits. It is forced to zero in the meantime.

Table 11. P_{SS} Voltage division rate selection bits

PDR2	PDR1	PDR0	PSS State	nxVPSS/13
0	0	0	IDLE	
0	0	1	BUSY	4xVPSS/13
0	1	0	BUSY	5xVPSS/13
0	1	1	BUSY	6xVPSS/13
1	0	0	BUSY	7xVPSS/13
1	0	1	BUSY	VPSS

32kHz STAND-BY OSCILLATOR

The 32kHz stand-by oscillator allows the ST6240 to generate real time interrupts and to supply the clock to the LCD driver. This enables the ST6240 to provide real time functions with the LCD display capability and lower power consumption. Figure 43 shows the 32kHz oscillator block diagram.

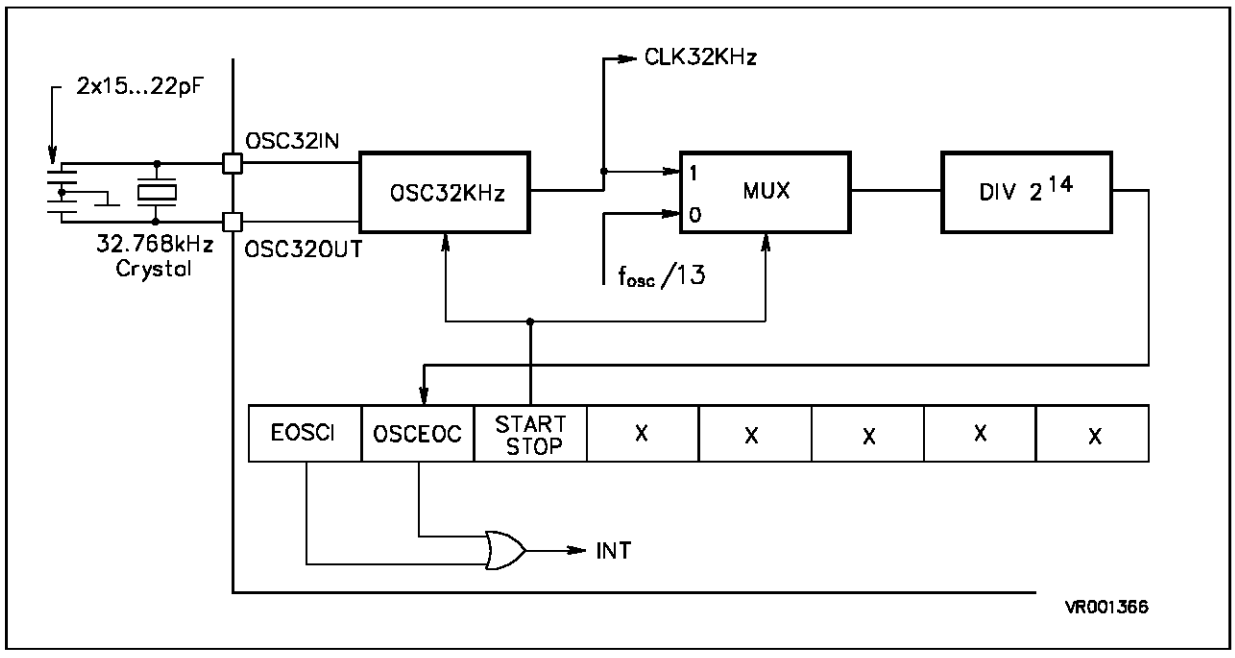
A 32.768kHz quartz crystal must be connected to the OSC32in and OSC32out pins to perform the real time clock operation. Two external capacitors of 15-22pF each must be connected between the oscillator pins and ground. The 32kHz oscillator is managed by the dedicated status/control register located at address 0DBh .

When the 32kHz stand-by oscillator is stopped (bit 5 of the Status/Control register cleared) the divider chain is supplied with a clock signal synchronous with machine cycle ($f_{OSC}/13$), this produces an interrupt request every $13x2^{14}$ clock cycle (i.e. 26.624ms) with an 8MHz quartz crystal.

When the 32kHz stand-by oscillator is enabled (bit 5 of the Status/Control register set to one) the divider chain is directly supplied with the 32kHz oscillator clock. The 32kHz clock from the standby oscillator can also be used as the LCD clock. This allows operation of the LCD in STOP mode. The interrupt output of the 32kHz oscillator peripheral generates an interrupt request every half second (500ms). This can be used to perform a real time clock function when the MCU is in STOP mode.

This interrupt signal is “ORed” with the interrupt request signals of the two on-chip timers and connected to the low level sensitive interrupt input associated to the interrupt vector #3 (FF2h, FF3h). The interrupt request has to be cleared by user software before leaving the interrupt service routine. Discrimination between the three interrupt sources is made by polling the Status/Control registers of Timer 1 (D4h), Timer 2 (D7h) and 32kHz oscillator (DBh).

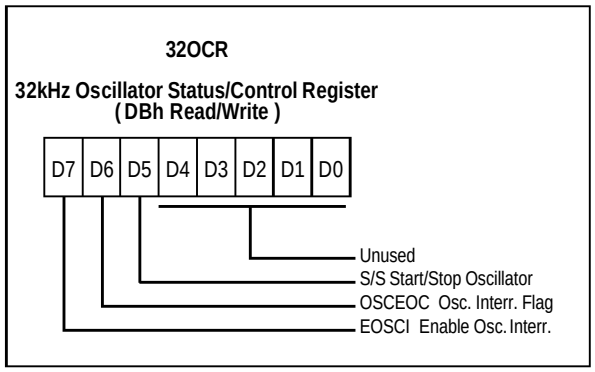
Figure 43. 32kHz Oscillator Block Diagram



32kHz STAND-BY OSCILLATOR (Continued)

32kHz Oscillator Status/Control Register

Figure 44. 32kHz Oscillator Register



EOSCI. Enable Oscillator Interrupt. This bit, when set, enables the 32kHz oscillator interrupt request.

OSCEOC. Oscillator Interrupt Flag. This bit indicates when the 32kHz oscillator has measured a 500ms elapsed time (providing a 32.768kHz quartz crystal is connected to the 32kHz oscillator dedicated pins). An interrupt request can be generated in relation to the state of EOSCI bit. This bit must be cleared by the user program before leaving the interrupt service routine.

START/STOP. Oscillator Start/Stop bit. This bit, when set, enables the 32kHz stand-by oscillator and the free running divider chain is supplied by the 32kHz oscillator signal. When this bit is cleared to zero the divider chain is supplied with the clock signal from the LCD Controller.

This register is cleared during reset.

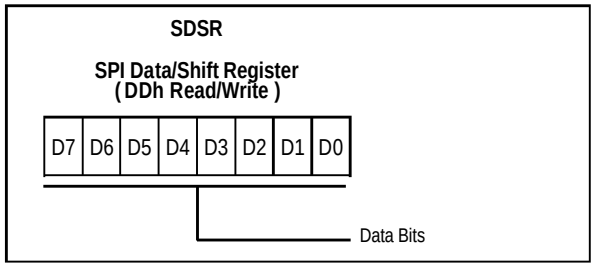
Note:
To achieve minimum power consumption in STOP mode (no system clock), the stand-by oscillator must be switched off (real time function not available) by clearing the Start/Stop bit in the oscillator status/control register.

SERIAL PERIPHERAL INTERFACE (SPI)

The ST6240 SPI is an optimized serial synchronous interface that supports a wide range of industry standard SPI specifications. The ST6240 SPI is controlled by small and simple user software to perform serial data exchange. The serial shift clock can be implemented either by software (using the bit-set and bit-reset instructions), with the on-chip Timer 1 by externally connecting the SPI clock pin to the timer pin or by directly applying an external clock to the SPI.

The peripheral is composed by an 8-bit Data/shift Register (address DDh) and a 4-bit binary counter. The SCL, Sin and Sout SPI data and clock signals are connected to the PB5, PB6 and PB7 I/O lines. With the 3 I/O pins, the SPI can operate in the following operating modes: Software SPI, S-BUS, I^2C-bus and as a standard serial I/O (clock, data, enable). An interrupt request can be generated after eight clock pulses. Figure 46 shows the SPI block diagram.

Figure 45. SPI Data/Shift Register



The PB5/SCL line clocks, on the falling edge, the shift register and the counter. To allow SPI operation the PA5/SCL must be programmed as input, an external clock supplied to this pin will drive the SPI peripheral (slave mode).

If PB5/SCL is programmed as output, a clock signal can be generated by software, setting and resetting the port line by software (master mode).

The SCL clock signal is the shift clock for the SPI data/shift register. The PB6/Sin pin is the serial shift input and PA7/Sout is the serial shift output. These two lines can be tied together to implement two wires protocols (I^2C-bus, etc). When data is serialized, the MSB is the first bit. PA6/Sin has to be programmed as input. For serial output operation PB7/Sout has to be programmed as open-drain output.

After 8 clock pulses (D7..D0) the output $\overline{Q4}$ of the 4-bit binary counter becomes low, disabling the clock from the counter and the data/shift register. Q4 enables the clock to generate an interrupt on the 8th clock falling edge as long as no reset of the counter (processor write into the 8-bit data/shift register) takes place. After a processor reset the interrupt is disabled. The interrupt is active when writing data in the shift register (DDh) and desactivated when writing any data in the register SPI Interrupt Disable (C2h).

SERIAL PERIPHERAL INTERFACE (Continued)

The generation of an interrupt to the Core provides information that new data is available (input mode) or that transmission is completed (output mode), allowing the Core to generate an acknowledge on the 9th clock pulse (I^2C-bus).

Since the SPI interrupt is connected to interrupt #1, the falling edge interrupt option should be selected by clearing to zero bit 6 of the Interrupt Option Register (IOR, C8h).

After power on reset, or after writing the data/shift register, the counter is reset to zero and the clock is enabled. In this condition the data shift register is ready for reception. No start condition has to be detected. Through the user software the Core may pull down the Sin line (Acknowledge) and slow down the SCL, as long as it is needed to carry out data from the shift register.

I^2C-bus Master-Slave, Receiver-Transmitter

When pins Sin and Sout are externally connected together it is possible to use the SPI as a receiver as well as a transmitter. With a simple software routine (by using bit-set and bit-reset on I/O line) a clock can be generated allowing I^2C-bus to work in master mode.

When implementing an I^2C-bus protocol, the start condition can be detected by setting the processor into a “wait for start” condition by simply enabling the interrupt of the PA6/Sin I/O port. This frees the processor from polling the Sin and SCL lines. After the transmission/reception the processor has to poll for the STOP condition.

In slave mode the user software can slow down the SCL clock frequency by simply putting the SCL I/O line in output open-drain mode and writing a zero into the corresponding data register bit.

As it is possible to directly read the Sin pin directly through the port register, the software can detect a difference between internal data and external data (master mode). Similar condition can be applied to the clock.

The typical speed of transmission in I^2C master or slave mode is in the range of 10kHz.

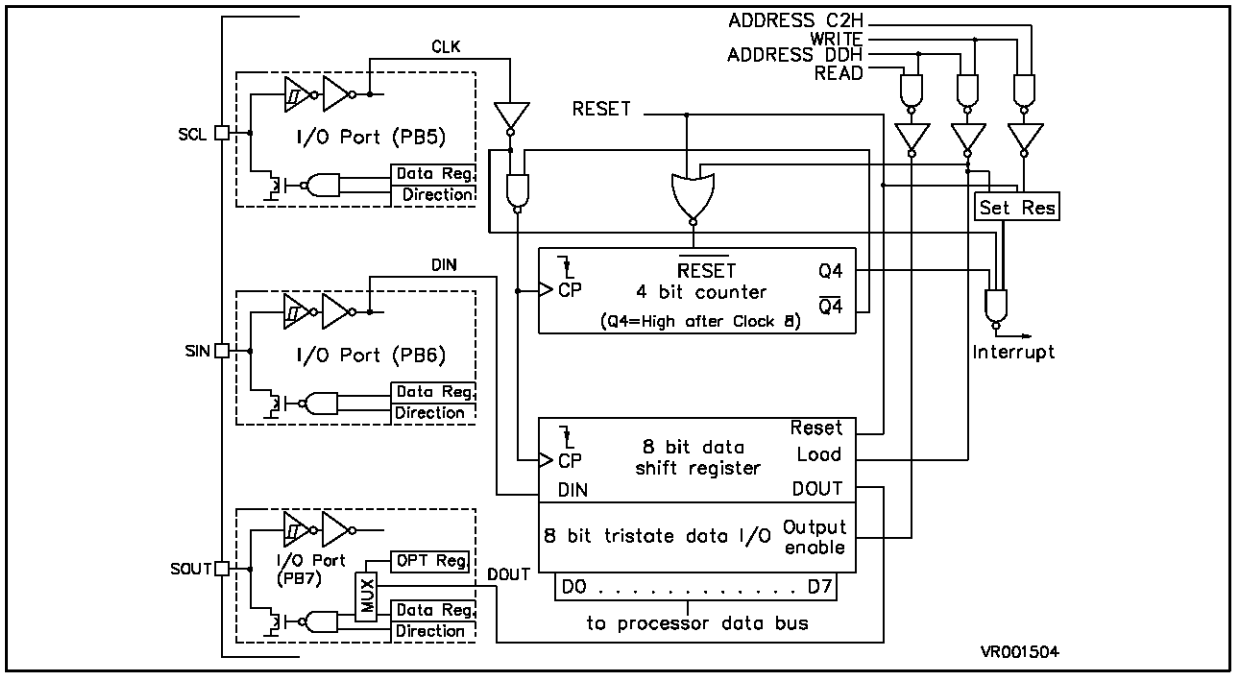
Three (Four) Wire Serial Bus

It is possible to use a single general purpose I/O pin (with the corresponding interrupt enabled) as a “chip enable” pin. SCL acts as active or passive clock pin, Sin as data in and Sout as data out (four wire bus). Sin and Sout can be connected together externally to implement three wire bus.

Note:

When the SPI is not used, the three I/O lines (Sin, SCL, Sout) can be used as normal I/O, with the following limitation: bit Sout cannot be used in open drain mode as this enables the shift register output to the port.
It is recommended, in order to avoid spurious interrupts from the SPI, to disable the SPI interrupt (the default state after reset) i.e. no write must be made to the 8-bit shift register (DDh). An explicit interrupt disable may be made in software by a dummy write to address C2h.

Figure 46. SPI Block Diagram



LCD CONTROLLER-DRIVER

The ST6240 LCD driver consists of a LCD control logic, a programmable prescaler, a 24 bytes wide dedicated LCD RAM, 45 segment and 4 common outputs. This allows a direct driving of up to 180 LCD segments.

The LCD driver is managed by the LCD Mode/Control register located at data RAM address DCh. Different display modes (1/1 duty, 1/2 duty, 1/3 duty and 1/4 duty) are available to cover a wide range of application requirements. The multiplexing display modes are software selectable by programming bits 6 and 7 of the LCD control register. Bits 0-5 are used to select the LCD drive and frame frequency (in relation to the system clock) and to switch off all segments. The LCD Driver can also be supplied by the 32kHz real-time oscillator allowing working in low power conditions and performing real time clock operation.

According to the data in the LCD RAM, the segment and the common drivers generate the segment and common signals which can directly drive an LCD panel.

The LCD control logic reads automatically the data from the LCD RAM independently and without interruption of the processor. The part of the LCD RAM that is not used for displaying can be used as normal data memory.

The scale factor of the clock prescaler can be fixed by software, therefore different frame frequencies can be defined.

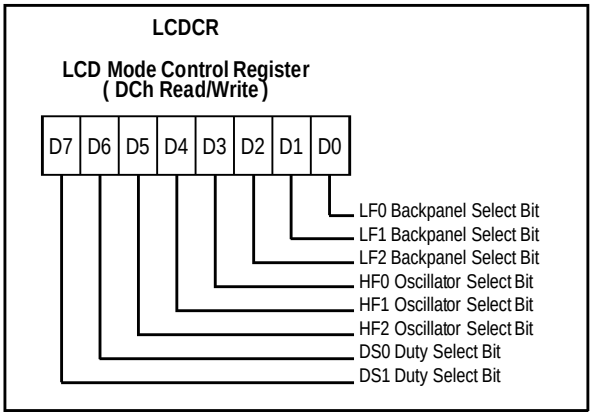
The ST6240 oscillator should operate with a 1.0486, 2.0972, 4.1943, 8.3886MHz frequency quartz crystal. This allows the associated division rates to achieve an internal reference frequency of 32.768kHz. The different division rates can be achieved by programming bits 3, 4, 5 in the LCD control register (see Table 14). It is not recommended to select an internal frequency lower than 32.768kHz as the clock supervisor circuit may switch off the LCD peripheral if the lower frequency is detected.

When the display is turned off, all segment and common outputs are switched to ground, causing all the segments to be switched off regardless of the contents of the LCD RAM.

When the Stand-by oscillator function is selected, the 32kHz stand-by oscillator is selected as clock source for the LCD.

To avoid incomplete frames of the LCD, the mode control bits do not immediately influence the LCD controller when the LCD control register is written. They are stored in a temporary register and change the LCD function only at the end of the frame. Special care must be taken when entering the STOP mode. After switching the LCD clock source from the main oscillator to the 32kHz standby oscillator it must be guaranteed that enough clock pulses are delivered to complete the current frame before entering the STOP mode. Otherwise the LCD function will not be changed and the LCD will be switched OFF after entering the STOP mode. Different LCD frame frequencies for each display mode are selected by bits in the LCD control register (see Table 15).

Figure 47. LCD Mode Control Register



DS0, DS1. Duty cycle select bits. These bits select the number of common backplanes used by the LCD control. This allows different multiplexing conditions.

HF0, HF1, HF2. These bits allow the LCD controller to be supplied with the correct frequency when different high main oscillator frequencies are selected as system clock. Table 14 shows the set-up for different clock crystals.

LF0, LF1, LF2. These bits control the LCD base operational frequency of the LCD common lines. Table 15 shows the set-up to select the different frequencies while table 16 shows the correspond-

Table 13. Duty Cycle Selection

DS1	DS0	Display Mode	Active Blackplanes	Max. Number of Segments Driven
0	0	1/4 duty	COM1, 2, 3, 4	180
0	1	1/1 duty	COM1	45
1	0	1/2 duty	COM1, 2	90
1	1	1/3 duty	COM1, 2, 3	135

LCD CONTROLLER-DRIVER (Continued)

Table 14. High Frequency Select Bits

HF2	HF1	HF0	Function	f_{osc}
0	0	0	Display off	
0	0	1	for stand-by Oscillator	32.768kHz
0	1	0	NOT TO BE USED	
0	1	1	÷ 32 for main oscillator	1.048MHz
1	0	0	÷ 64 for main oscillator	2.097MHz
1	0	1	÷ 128 for main oscillator	4.194MHz
1	1	0	÷ 256 for main oscillator	8.388MHz
1	1	1	NOT TO BE USED	

Notes :

1. The usage f_{osc} values different from those defined in this table cause the LCD to operate at a reference frequency different from 32.768kHz.
2. It is not recommended to select an internal frequency lower than 32.768kHz as the clock supervisor circuit may switch off the LCD peripheral if lower frequency is detected.

Table 15. LCD Frequency Select Bits

LF2	LF1	LF0	f_{LCD} (Hz)
0	0	0	64
0	0	1	85
0	1	0	128
0	1	1	171
1	0	0	256
1	0	1	341
1	1	0	512
1	1	1	Not to be Used

ing frame values with the different multiplexing conditions.

According to the selected LCD drive frequency f_{LCD} the frame frequencies come out as shown in Table 16.

The Figure 54 illustrates the waveforms of the different duty signals.

Table 16. Available Frame Frequencies for LCD

f_{LCD} (Hz)	Frame Frequency f_F (Hz)			
	1/1 duty	1/2 duty	1/3 duty	1/4 duty
512	512	256	171	128
341	341	171	114	85
256	256	128	85	64
171	171	85	57	43
128	128	64	43	32
85	85	43	28	21
64	64	32	21	16

The value of the VLCD voltage can be chosen independently from V_{DD} according to the display requirements. The intermediate VLCD levels 2/3 VLCD, 1/3 VLCD and 1/2 VLCD are generated by an internal resistor network as shown in Figures 52 and 53. The half VLCD level for 1/2 duty cycle is obtained by the external connection of VLCD1/3 and VLCD2/3 pins. All intermediate VLCD levels are connected to pins to enable external capacitive buffering or resistive shunting.

LCD CONTROLLER-DRIVER (Continued)

The internal resistive divider network is realized with two parallel dividers. One has high resistivity, the other one low resistivity. The high resistive divider (R_H) is permanently switched on during the LCD operation. The low resistive divider (R_L) is only switched on for a short period of time when the levels of common lines and segment lines are changed. This method combines low source impedance for fast switching of the LCD pixels with high source impedance for low power consumption. Fig. 48 shows the typical current into V_{LCD} pin in dependency of the display voltage V_{LCD}. When the display is switched off (by program or reset) the internal resistor network is also switched off to achieve minimum power consumption. The low resistivity divider is active at each edge of f_{LCD} during 8 clock cycles of F_{32kHz}.

The internal resistor network is implemented with resistive transistor elements to achieve high precision. For display voltages $V_{LCD} < 4.5V$ the resistivity of the divider may be too high for some applications (especially using 1/3 or 1/4 duty display mode). In that case an external resistive divider must be used to achieve the desired resistivity.

Figure 48. Typical Current Consumption on VLCD Pin (25°C, no load, fLCD= 512 Hz, mux=1/3 -1/4)

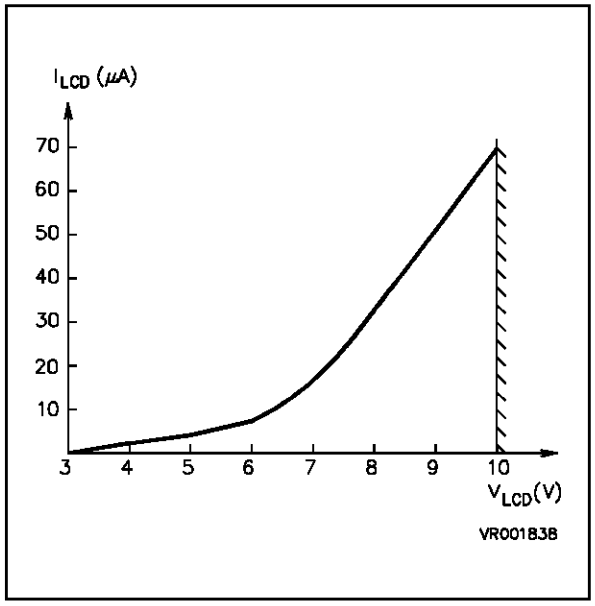
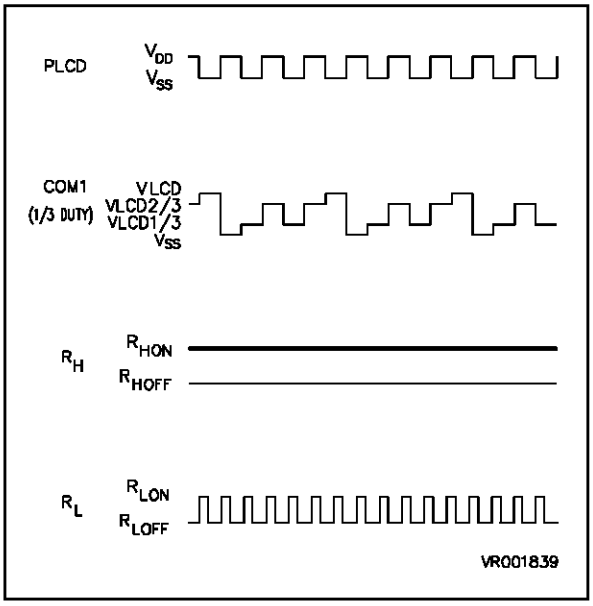


Figure 49. Typical Chronogram of Activation of the V_{LCD} Divider Network



LCD CONTROLLER-DRIVER (Continued)

Typical External resistances values are in the range of 100 kΩ to 150 kΩ. External capacitances in the range of 10 to 47 nF can be added to V_{LCD} 2/3 and V_{LCD} 1/3 pins and to V_{LCD} if the V_{LCD} connection is highly impedant.

When the program is switched off (by program or reset) the internal resistor network is also switched off to achieve minimum power consumption.

Figure 50. Typical Network to connect to V_{LCD} pins if $V_{LCD} \leq 4.5V$

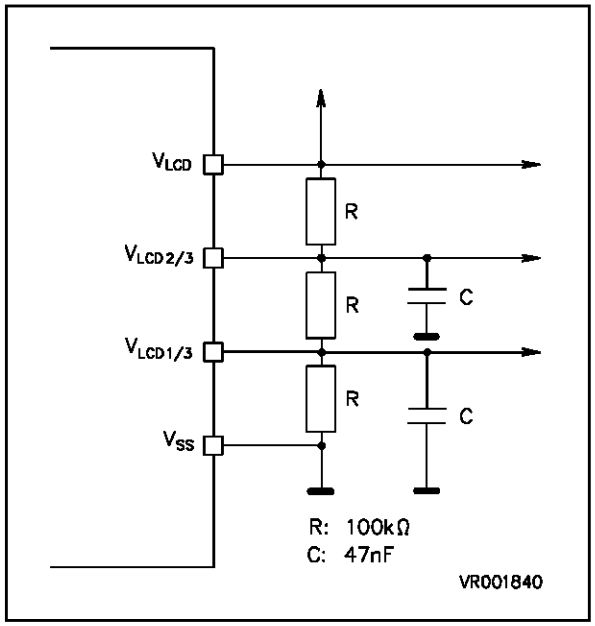


Figure 51. Generation of the 32kHz clock

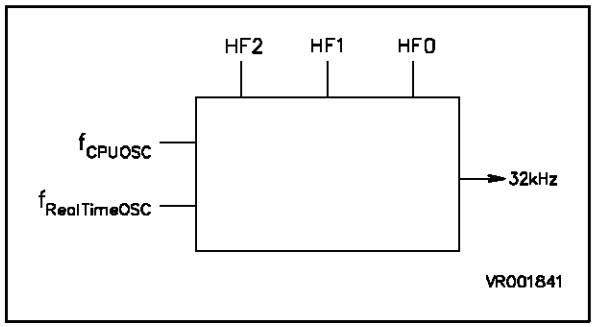


Figure 52. Bias Config for 1/2 Duty

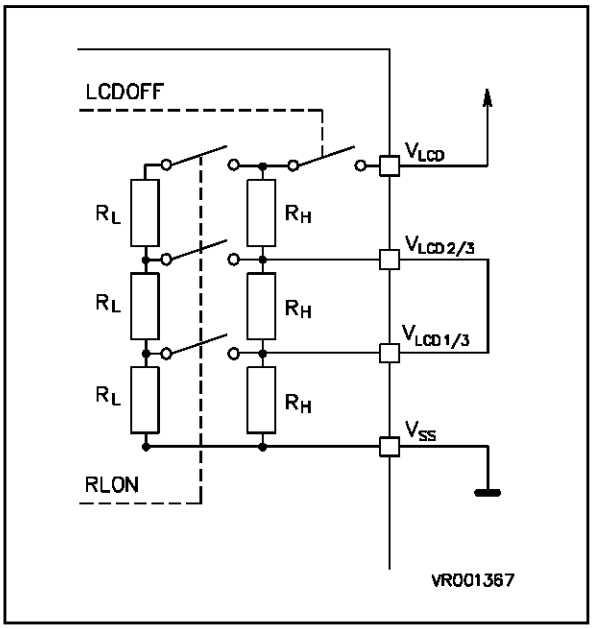
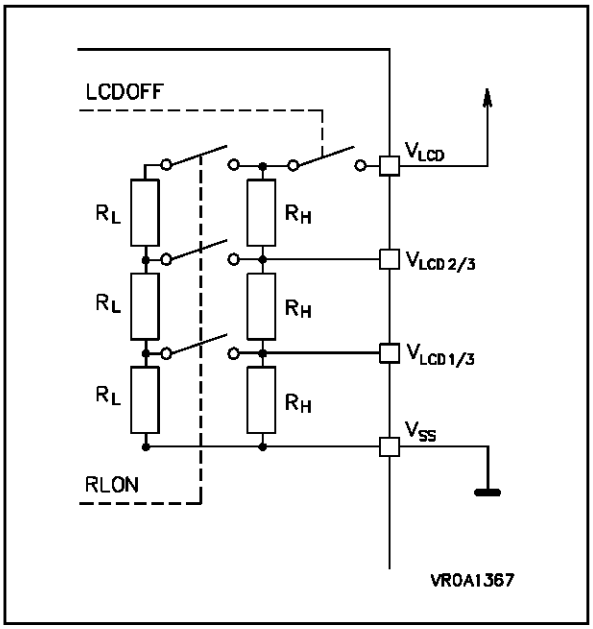


Figure 53. Bias Configuration for 1/1, 1/3 and 1/4 Duty Operation of LCD



LCD CONTROLLER-DRIVER (Continued)

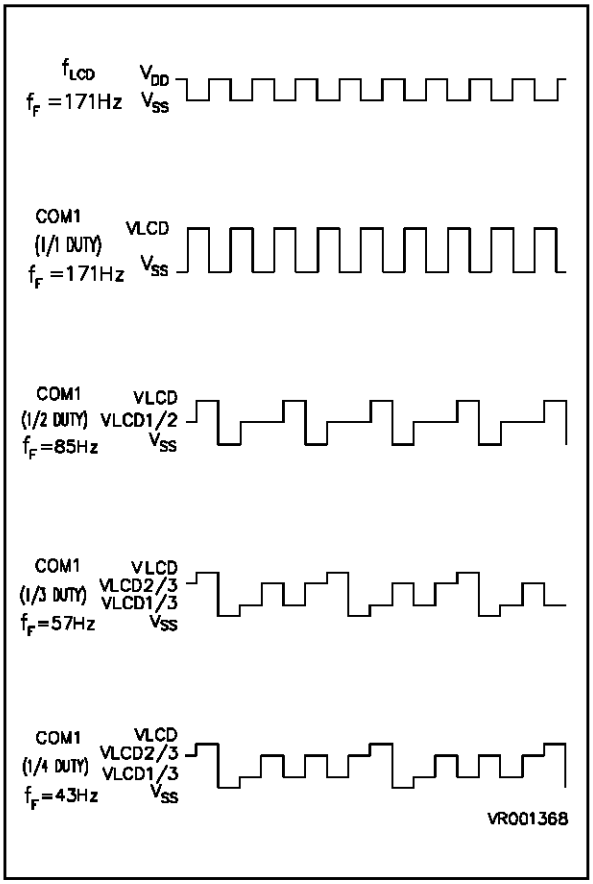
Address Mapping of the Display Segments.

The LCD RAM is located in the ST6240 data space from addresses E0h to F7h. The LCD forms a matrix of 45 segment lines (rows) and up to 4 common lines (columns). Each bit of the LCD RAM is mapped to one element of the LCD matrix, as described in Figure 55. If a bit is set, the corresponding LCD segment is switched on; if it is reset, the segment is switched off. The segments outputs S1, S2 and S3 are not connected to any pin.

When multiplex rates lower than 1/4 are selected, the unused LCD RAM is free for general use. In the 1/2 duty mode, for instance, half of the LCD RAM is available for storing general purpose data. The address range from F8h to FEh can be used as general purpose data RAM, but not for displaying data (it is reserved for future LCD expansion).

After a reset, the LCD RAM is not initializated and contains arbitrary information. As the LCD control register is reset, the LCD is completely switched off.

Figure 54. Common Signal Waveforms



LCD CONTROLLER-DRIVER (Continued)

Figure 55. Addressing Map of the LCD RAM

Data RAM Address	MSB							LSB	
E0	S8	S7	S6	S5	S4	NA	NA	NA	
E1	S16	S15	S14	S13	S12	S11	S10	S9	
E2	S24	S23	S22	S21	S20	S19	S18	S17	
E3	S32	S31	S30	S29	S28	S27	S26	S25	**COM1**
E4	S40	S39	S38	S37	S36	S35	S34	S33	
E5	S48	S47	S46	S45	S44	S43	S42	S41	
E6	S8	S7	S6	S5	S4	NA	NA	NA	
E7	S16	S15	S14	S13	S12	S11	S10	S9	
E8	S24	S23	S22	S21	S20	S19	S18	S17	
E9	S32	S31	S30	S29	S28	S27	S26	S25	**COM2**
EA	S40	S39	S38	S37	S36	S35	S34	S33	
EB	S48	S47	S46	S45	S44	S43	S42	S41	
EC	S8	S7	S6	S5	S4	NA	NA	NA	
ED	S16	S15	S14	S13	S12	S11	S10	S9	
EE	S24	S23	S22	S21	S20	S19	S18	S17	
EF	S32	S31	S30	S29	S28	S27	S26	S25	**COM3**
F0	S40	S39	S38	S37	S36	S35	S34	S33	
F1	S48	S47	S46	S45	S44	S43	S42	S41	
F2	S8	S7	S6	S5	S4	NA	NA	NA	
F3	S16	S15	S14	S13	S12	S11	S10	S9	
F4	S24	S23	S22	S21	S20	S19	S18	S17	
F5	S32	S31	S30	S29	S28	S27	S26	S25	**COM4**
F6	S40	S39	S38	S37	S36	S35	S34	S33	
F7	S48	S47	S46	S45	S44	S43	S42	S41	
F8 - FE	TO BE USED AS GENERAL PURPOSE DATA RAM (NOT FOR DISPLAY DATA)								

Notes:

In STOP mode no clock is available for the LCD controller from the main oscillator. If the 32kHz oscillator is activated the LCD can also operate in STOP mode. If the stand-by oscillator is not active, the LCD controller is switched off when STOP instruction is executed; this mode has to be selected to reach the lowest power consumption.

A missing LCD clock (no oscillator active, broken crystal, etc.) is detected by a clock supervisor circuit that switches all the segments and common lines to ground to avoid destructive DC levels at the LCD.

The LCD function change is only effective at the end of a frame. For this reason special care has to be taken when entering the STOP mode. After switching the LCD clock source from the main oscillator to the 32kHz stand-by oscillator it must be guaranteed that enough clock pulses are delivered to complete the current frame before entering the STOP mode. Otherwise the LCD function will not be changed and the LCD is switched off after entering the STOP mode.

The RAM address F8-FEh are not used for LCD display purposes. So they are available as 7 additional Data RAM registers.

SOFTWARE DESCRIPTION

The ST62xx software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum, in short to provide byte efficient programming capability. The ST62xx core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

Addressing Modes

The ST62xx core has nine addressing modes which are described in the following paragraphs. The ST62xx core uses three different address spaces: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte that is processed by the instruction is stored in the location that follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode . In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) that use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction that follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits that characterize the kind of the test, one bit that determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits that give the span of the branch (0h to Fh) that must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

SOFTWARE DESCRIPTION (Continued)

Instruction Set

The ST62xx core has a set of 40 basic instructions. When these instructions are combined with nine addressing modes, 244 usable opcodes can be obtained. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, bit manipulation. The following paragraphs describe the different types.

All the instructions within a given type are presented in individual tables.

Load & Store. These instructions use one,two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 17. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ . Affected
* . Not Affected

SOFTWARE DESCRIPTION (Continued)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 18. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	*
AND A, (Y)	Indirect	1	4	Δ	*
AND A, rr	Direct	2	4	Δ	*
ANDI A, #N	Immediate	2	4	Δ	*
CLR A	Short Direct	2	4	Δ	Δ
CLR rr	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

SOFTWARE DESCRIPTION (Continued)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 19. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ. Affected
*. Not Affected

Table 20. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
*. Not Affected

Table 21. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated and a WAIT is automatically executed instead of a STOP if the Watchdog function is selected.
Δ. Affected
*. Not Affected

Table 22. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc.12-bit address;
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

Opcode Map Summary. The following table contains an opcode map for the instructions used on the MCU.

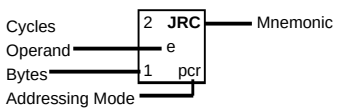
LOW / HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	LOW / HI
0 0000	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b0,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **LD** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b0,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LDI** rr,nn 3 imm	2 **JRC** e 1 pcr	4 **LD** a,(y) 1 ind	0 0000
1 0001	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b0,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** x 1 sd	2 **JRC** e 1 prc	4 **LDI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b0,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** x 1 sd	2 **JRC** e 1 pcr	4 **LD** a,rr 2 dir	1 0001
2 0010	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b4,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **CP** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b4,rr 2 b.d	2 **JRZ** e 1 pcr	4 **COM** a 1 inh	2 **JRC** e 1 pcr	4 **CP** a,(y) 1 ind	2 0010
3 0011	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b4,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,x 1 sd	2 **JRC** e 1 prc	4 **CPI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b4,rr 2 b.d.	2 **JRZ** e 1 pcr	4 **LD** x,a 1 sd	2 **JRC** e 1 pcr	4 **CP** a,rr 2 dir	3 0011
4 0100	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b2,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **ADD** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b2,rr 2 b.d	2 **JRZ** e 1 pcr	2 **RETI** 1 inh	2 **JRC** e 1 pcr	4 **ADD** a,(y) 1 ind	4 0100
5 0101	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b2,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** y 1 sd	2 **JRC** e 1 prc	4 **ADDI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b2,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** y 1 sd	2 **JRC** e 1 pcr	4 **ADD** a,rr 2 dir	5 0101
6 0110	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b6,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **INC** (x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b6,rr 2 b.d	2 **JRZ** e 1 pcr	2 **STOP** 1 inh	2 **JRC** e 1 pcr	4 **INC** (y) 1 ind	6 0110
7 0111	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b6,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,y 1 sd	2 **JRC** e 1 prc	#	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b6,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LD** y,a 1 sd	2 **JRC** e 1 pcr	4 **INC** rr 2 dir	7 0111
8 1000	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b1,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **LD** (x),a 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b1,rr 2 b.d	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 pcr	4 **LD** (y),a 1 ind	8 1000
9 1001	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b1,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** v 1 sd	2 **JRC** e 1 prc	#	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b1,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** v 1 sd	2 **JRC** e 1 pcr	4 **LD** rr,a 2 dir	9 1001
A 1010	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b5,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **AND** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b5,rr 2 b.d	2 **JRZ** e 1 pcr	4 **RLC** a 1 inh	2 **JRC** e 1 pcr	4 **AND** a,(y) 1 ind	A 1010
B 1011	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b5,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,v 1 sd	2 **JRC** e 1 prc	4 **ANDI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b5,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LD** v,a 1 sd	2 **JRC** e 1 pcr	4 **AND** a,rr 2 dir	B 1011
C 1100	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b3,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **SUB** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b3,rr 2 b.d	2 **JRZ** e 1 pcr	2 **RET** 1 inh	2 **JRC** e 1 pcr	4 **SUB** a,(y) 1 ind	C 1100
D 1101	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b3,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** w 1 sd	2 **JRC** e 1 prc	4 **SUBI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b3,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** w 1 sd	2 **JRC** e 1 pcr	4 **SUB** a,rr 2 dir	D 1101
E 1110	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b7,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **DEC** (x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b7,rr 2 b.d	2 **JRZ** e 1 pcr	2 **WAIT** 1 inh	2 **JRC** e 1 pcr	4 **DEC** (y) 1 ind	E 1110
F 1111	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b7,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,w 1 sd	2 **JRC** e 1 prc	#	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b7,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LD** w,a 1 sd	2 **JRC** e 1 pcr	4 **DEC** rr 2 dir	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from:

$Tj = T_A + PD \times RthJA$

Where : T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_{LCD}	Display Voltage	-0.3 to 11.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
I_O	Current Drain per Pin Excluding V_{DD} & V_{SS}	± 10	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PQFP80		70		°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version 6 Suffix Version	0 -40		70 85	°C
V_{DD}	Operating Supply Voltage		3		6	V
V_{LCD}	Display Voltage		3		10	V
V_{DD}	RAM Retention Voltage		2			V

RECOMMENDED OPERATING CONDITIONS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (1)(4)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01 0.01		8.388 2	MHz
I_{INJ+}	Pin Injection Current (positive) Digital Input (2) Analog Input (3)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input (2) Analog Input	V_{DD} = 4.5 to 5.5V			-5	mA

Notes :
1. An oscillator frequency above 1MHz is recommanded for reliable A/D results.
2. A current of ± 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device. For a positive current injected into one pin, a part of this current (~ 10%) can be expected to flow from the neighbouring pins. A current of -5mA can be forced on one input of the analog section at a time (or -2.5mA for all inputs at a time) without affecting the conversion.
3. If a total current of +1mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 1mA, all the conversion is resulting shifted of +1LSB. If a total positive current of +5mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 5mA, all the conversion is resulting shifted of +2LSB.
4. Operation below 0.01 MHz is possible but requires increased supply current.

EEPROM INFORMATION

The ST62xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	RESET, NMI, TIMER, WDON Pin			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	TIMER	$0.80V_{DD}$			V
		RESET, NMI, WDON Pin	$0.70V_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	RESET Pin V_{DD} =5V $V_{IN} = V_{DD}$ (1) $V_{IN} = V_{DD}$ (2) $V_{IN} = V_{SS}$ (5)			10 1 50	µA mA µA
V_{OL}	Low Level Output Voltage	TIMER, I_{OL} = 5.0mA			$0.2V_{DD}$	V
V_{OH}	High Level Output Voltage	TIMER, I_{OL} = -5.0mA	$0.65V_{DD}$			V
R_{PU}	Pull-up Resistor	V_{IN}=0V V_{DD}=5V WDON - NMI	40	100	200	kΩ
		RESET	200	300	500	kΩ

Notes on next page

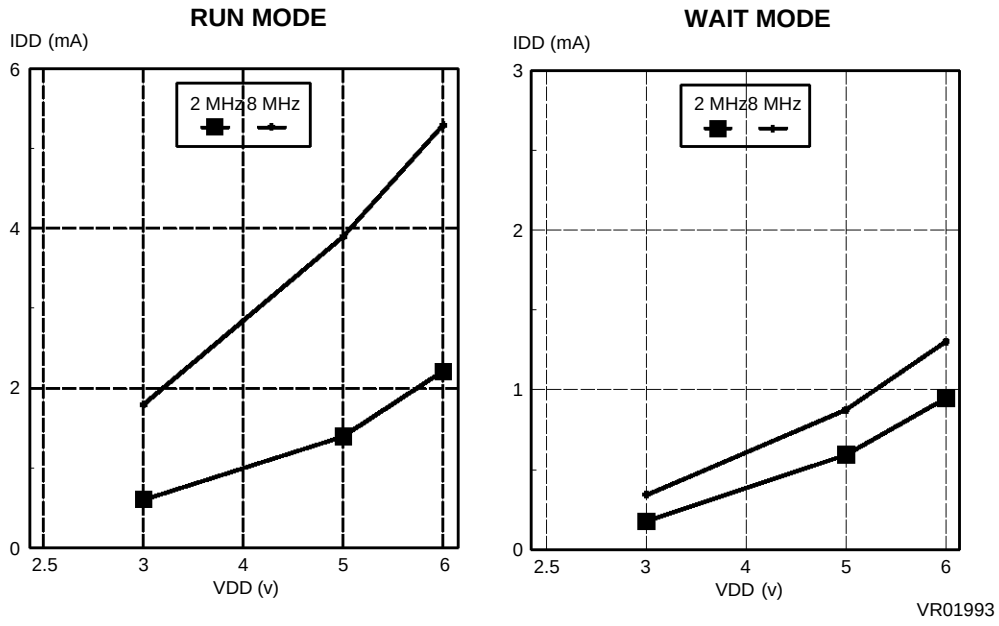
DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{IL} I_{IH}	Input Leakage Current	TIMER $V_{IN} = V_{DD}$ or V_{SS}		0.1	1.0	µA
I_{IL} I_{IH}	Input Leakage Current	NMI VDD = 5.V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			 100 1.0	µA
I_{IL} I_{IH}	Input Leakage Current	WDON VDD = 5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			 100 1.0	µA
I_{DD}	Supply Current RUN Mode	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		4	7	mA
	Supply Current WAIT Mode (4)	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		1	2	mA
	Supply Current RESET Mode	f_{OSC} = 8MHz, $V_{RESET} = V_{SS}$		1	7	mA
	Supply Current STOP Mode (3)(4)	I_{LOAD} = 0mA V_{DD} = 5.0V		1	10	µA

Notes :
1. No Watchdog Reset activated.
2. Reset generated by Watchdog.
3. When the watchdog function is actvated the STOP instruction is deactivated. WAIT instruction is automatically executed.
4. All on-chip peripherals in OFF state
5. Pull-up resistor

CURRENT CONSUMPTION

IDD Current Versus Supply Voltage
Typical Values, Temp : -40/+85°C



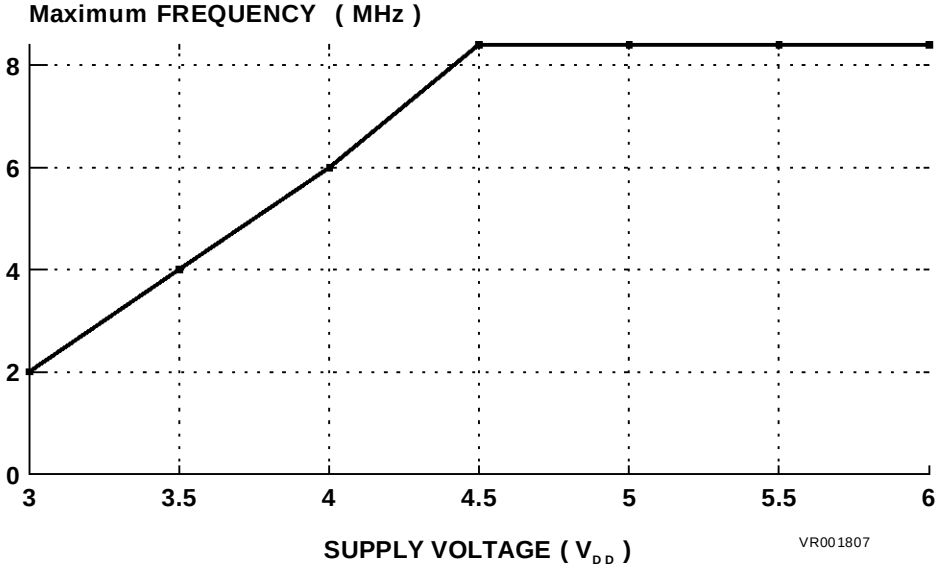
AC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (2)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01		8.388 2	MHz
t_{SU}	Oscillator Start-up Time	$C_{L1} = C_{L2}$ = 22pF - crystal		5	20	ms
t_{SR}	Supply Rise Time	10% to 90%	0.01		100	
t_{REC}	Supply Recovery Time (1)		100			
T_W	Minimum Pulse Width	NMI Pin V_{DD} = 5V	100			ns
		RESET Pin	100			ns
T_{WEE}	EEPROM Write Time	T_A = 25°C One Byte T_A = 85°C One Byte		5 15	10 25	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A L_{OT} Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention	T_A = 55°C	10			years
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:
1. Period for which V_{DD} has to be connected or at 0V to allow internal Reset function at next power-up.
2. Operation below 0.01 MHz is possible but requires increased supply current.

Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



I/O PORTS
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	I/O Pins			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	I/O Pins	$0.7V_{DD}$			V
V_{OL}	Low Level Output Voltage	I/O Pins, I_O = 10µA (sink)			0.1	V
		I/O Pins, $I_{OL} = V_{DD}$x1mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 1.6mA V_{DD} = 3V			0.4	V
	Low Level Output Voltage, PB4-PB7 Only	I/O Pins, $I_{OL} = V_{DD}$x2mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 3.2mA V_{DD} = 3V			0.4	V
		I/O Pins, $I_{OL} = V_{DD}$x4mA V_{DD} = 4.5 to 6V			$0.26xV_{DD}$	V
		I/O Pins, I_{OL} = 6.4mA V_{DD} = 3V			0.8	V
V_{OH}	High Level Output Voltage	I/O Pins, I_O = -10µA (source)	V_{DD}-0.1			V
		I/O Pins, $I_{OL} = -V_{DD}$x1mA V_{DD} = 5.0V	$0.6xV_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	I/O Pins,(1)		0.1	1.0	µA
R_{PU}	Pull-up Resistor	I/O Pins V_{IN} = 0V, V_{DD} = 5.0V	40	100	200	KΩ

Note 1. Pull-up resistor off

SPI ELECTRICAL CHARACTERISTICS
(V_{DD} = 5.0 V, T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
F_{CL}	Clock Frequency	applied on PB5/SCL			1	MHz
t_{SU}	Set-up Time	applied on PB6/Sin		50		ns
t_h	Hold Time	applied on PB6/Sin		100		ns

A/D CONVERTER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution (3)			8		Bit
A_{TOT}	Total Accuracy (3)	f_{OSC} > 1.2 MHz f_{OSC} > 32kHz			± 2 ±4	LSB
t_C(1)	Conversion Time	f_{OSC} = 8MHz		70		µs
V_{AN}	Conversion Range		V_{SS}		V_{DD}	V
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN}(2)	Analog Input Capacitance			2	5	pF
ASI	Analog Source Impedance				30	kΩ
SSI	Analog Reference Supply Impedence				2	kΩ

Notes:
1. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.
2. Excluding Pad Capacitance
3. Noise at V_{DD},V_{SS} ≤ 10mV

TIMER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{12}{f_{OSC}}$			second
f_{IN}	Input Frequency on TIMER Pin				$\frac{f_{OSC}}{8}$	MHz
t_W	Pulse Width at TIMER Pin	$V_{DD} \geq 3V$ $V_{DD} \geq 4.5V$	1 125			µs ns

PSS ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwhise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{PSS}	PSS pin Input Voltage		V_{SS}		V_{DD}	V
I_{PSS}	PSS pin Input Current	PSS RUN PSS STOP V_{PSS} = 5V, T_A = 25°C			350 1	µA

LCD ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

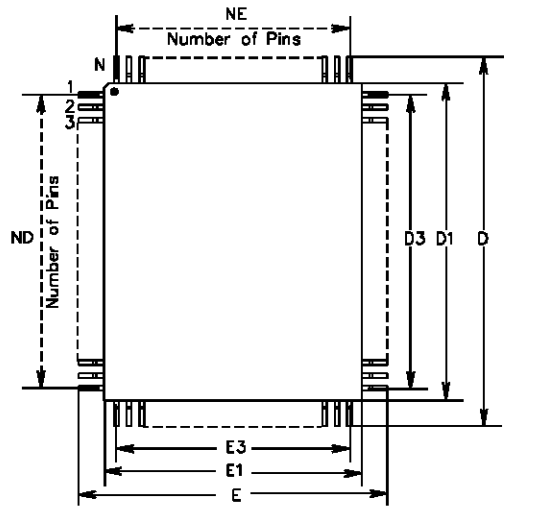
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{FR}	Frame Frequency	1/4 Duty f_{OSC} = 1, 2, 4, 8MHz	16		128	Hz
V_{OS}	DC Offset Voltage(1)	$V_{LCD} = V_{DD}$, no load			50	mV
V_{OH}	COM High Level, Output Voltage	I = 100µA, V_{LCD} = 5V	4.5V			V
V_{OL}	COM Low Level, Output Voltage	I = 100µA, V_{LCD} = 5V			0.5V	V
V_{OH}	SEG High Level, Output Voltage	I = 50µA, V_{LCD} = 5V	4.5V			V
V_{OL}	SEG Low Level, Output Voltage	I = 50µA, V_{LCD} = 5V			0.5V	V
V_{LCD}	Display Voltage	Note 2	3		10	V

Notes :

1. The DC offset voltage refers to all segment and common outputs. It is the difference between the measured voltage value and nominal value for every voltage level. Ri of voltage meter must be greater than or equal to 100MΩ.
2. An external resistances network is required when $V_{LCD} \leq 4.5V$.

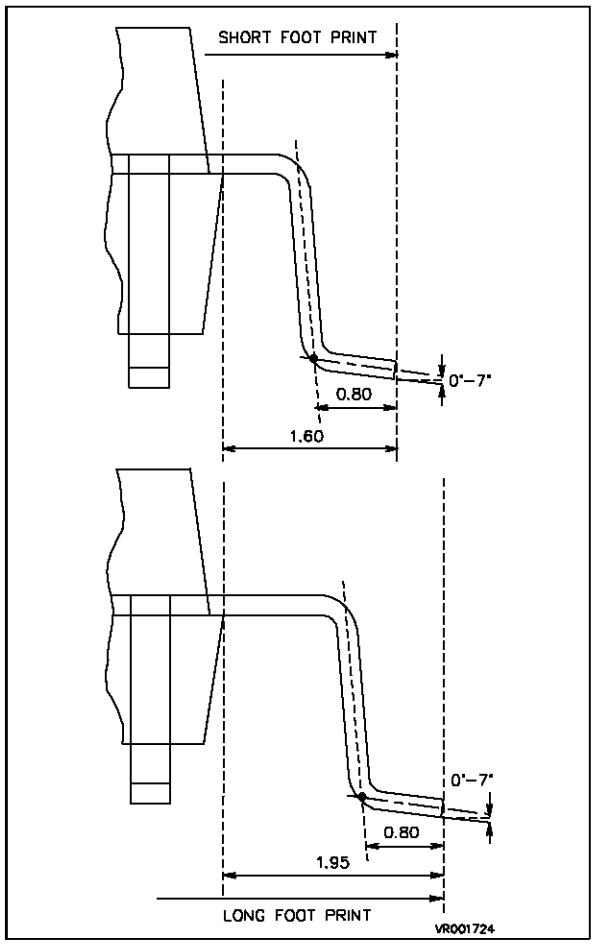
PACKAGE MECHANICAL DATA

Figure 56. ST6240 80 Pin Plastic Quad Flat Pack Package

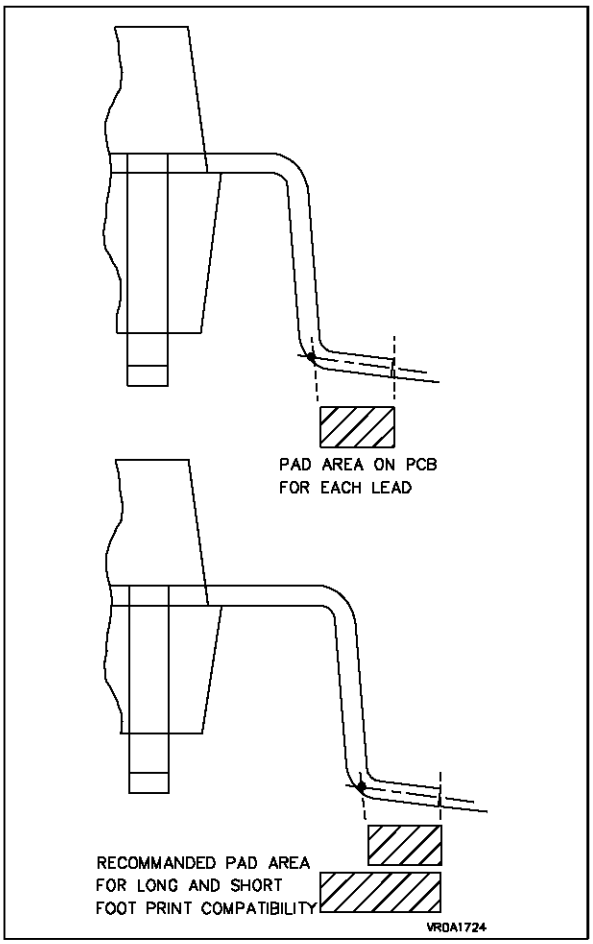


Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.30			0.130
A2	2.55	2.80	3.05	0.100	0.110	0.120
D		23.20			0.913	
D1	19.90	20.00	20.10	0.783	0.787	0.791
D3		18.40			0.724	
E		17.20			0.677	
E1	13.90	14.00	14.10	0.547	0.551	0.555
E3		12.00			0.472	
e		0.80			0.032	
Number of Pins						
ND	24					
NE	16					
N	80					

Short/Long Footprint Measurement

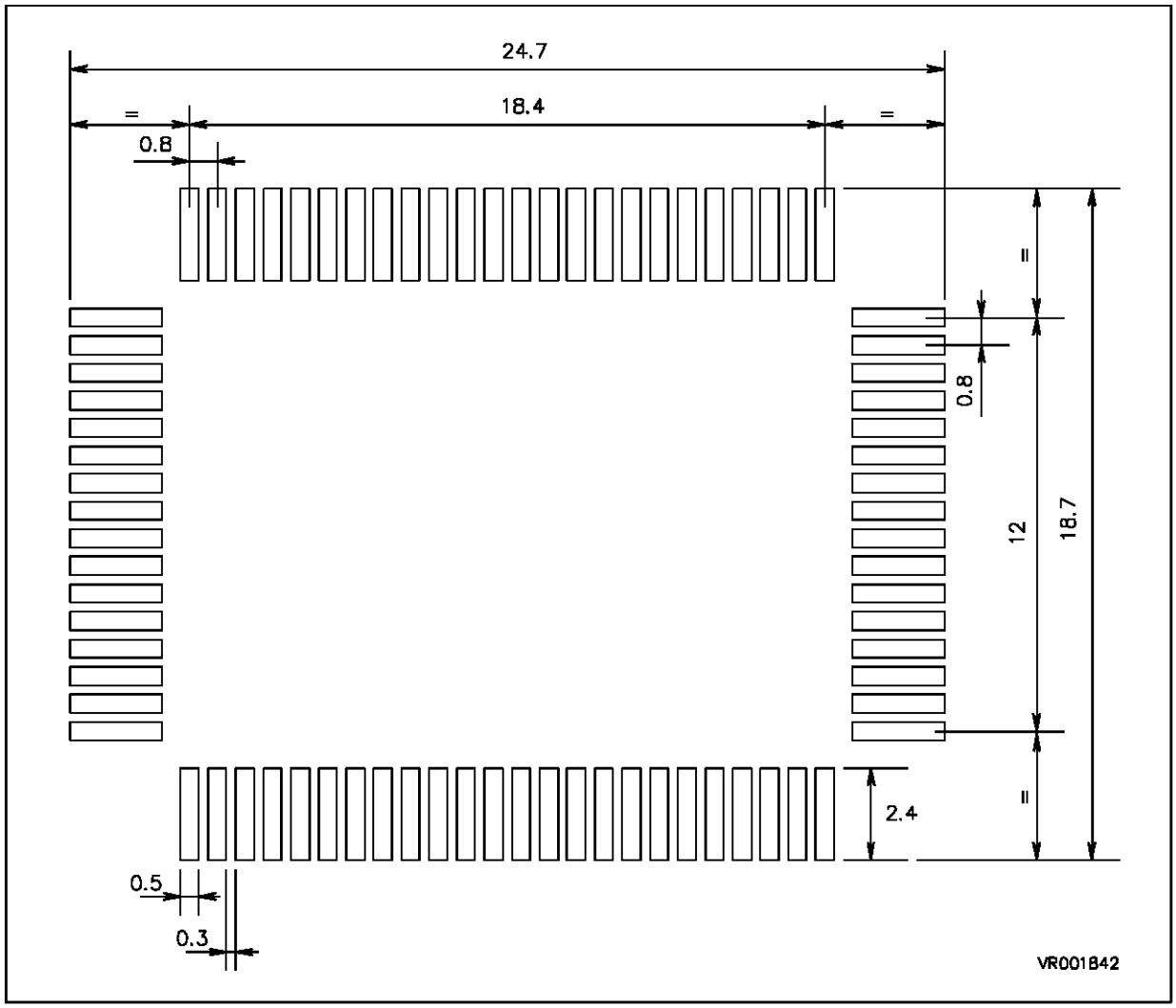


Short/Long Footprint recommended Padding



PACKAGE MECHANICAL DATA (Continued)

Recommended Solder Pad Footprint For QFP80 (in mm)



ORDERING INFORMATION

The following chapter deals with the procedure for transfer the Program/Data ROM codes to SGS-THOMSON.

Communication of the ROM Codes. To communicate the contents of Program/Data ROM memories to SGS-THOMSON, the customer has to send :

– one file in INTEL INTELLEC 8/MDS FORMAT (in an MS-DOS 5" diskette) for the PROGRAM Memory

– one file in INTEL INTELLEC 8/MDS FORMAT (in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)

– a filled Option List form as described in the OPTION LIST paragraph.

The program ROM should respect the ROM Memory Map as in Table 23.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

Table 23. ROM Memory Map

ROM Page	Device Address	Description
Page 0	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	Reserved User ROM
Page 3	0000h-000Fh 0010h-07FFh	Reserved User ROM

Note : EPROM addresses are related to the ROM file to be processed.

Customer EEPROM Initial Contents : Format

a. The content should be written into an INTEL INTELLEC format file.

b. In the case of 128 bytes of EEPROM, the starting address in 000h and the end in 7Fh.

c. Undefined or don't care bytes should have the content FFh.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are compared and a computer listing is generated from them. This listing refers exactly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

ORDERING INFORMATION TABLE

Sales Types	Temperature Range	Package
ST6240Q1/XX	0 to + 70°C	PQFP80
ST6240Q6/XX	-40 to + 85°C	PQFP80

Note : "XX" is the ROM code identifier allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

ST6240 MICROCONTROLLER OPTION LIST

Customer . . . . . .

Address . . . . . .

Contact . . . . . .

Phone No . . . . . .

Reference . . . . . .

SGS-THOMSON Microelectronics references

Device	[ ] ST6240	
Package	[ ] Plastic Quad Flat Package	
Temperature Range	[ ] 0°C to + 70°C	[ ] -40°C to + 85°C

Special Marking [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _"

Authorized characters are Letters, digits, '.', '–', '/' and spaces only.
For marking one line with 10 characters maximum is possible.

Comments :
- Number of LCD segments used :
- Number of LCD backplanes used :
- PSS used:

Note :

Signature

Date

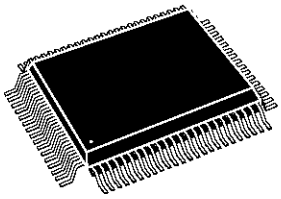
Notes:

SGS-THOMSON MICROELECTRONICS

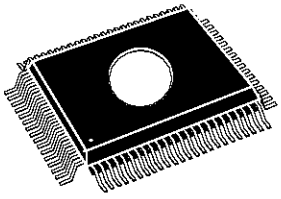
ST62E40
ST62T40

8-BIT OTP/EPROM HCMOS MCUs WITH LCD DRIVER,EEPROM AND A/D CONVERTER

- 3 to 6V supply operating range
- 8.4MHz Maximum Clock Frequency
- Run, Wait & Stop Modes
- 5 different interrupt vectors
- Look-up table capability in EPROM
- User EPROM: 7948 bytes
  Data RAM: 192 bytes
  LCD RAM: 24 bytes
  EEPROM: 128 bytes
- PQFP80 and CQFP80-W packages
- 16 fully software programmable I/O as:
  - Input with/without pull-up resistor
  - Input with interrupt generation
  - Open-Drain or Push-pull outputs
  - Analog Inputs (12 pins)
- 4 I/O lines can sink up to 20mA for direct LED or TRIAC driving and have SPI alternate functions
- Two 8-bit counters and 7-bit programmable prescalers (Timer 1 and 2)
- Software or hardware activated digital watchdog
- 8-bit A/D converter with up to 12 analog inputs
- 8-bit synchronous serial peripheral interface (SPI)
- LCD driver with 45 segment outputs, 4 backplane outputs and selectable duty cycle for up to 180 LCD segments direct driving
- 32kHz oscillator for stand-by LCD operation
- Power Supply Supervisor (PSS)
- One external not maskable interrupt
- 9 powerful addressing modes
- The accumulator, the X, Y, V & W registers, the port and peripherals data & control registers are addressed in the data space as RAM locations.
- The ST62E40 is the EPROM version, ST62T40 is the OTP version, fully compatible with ST6240 ROM version.



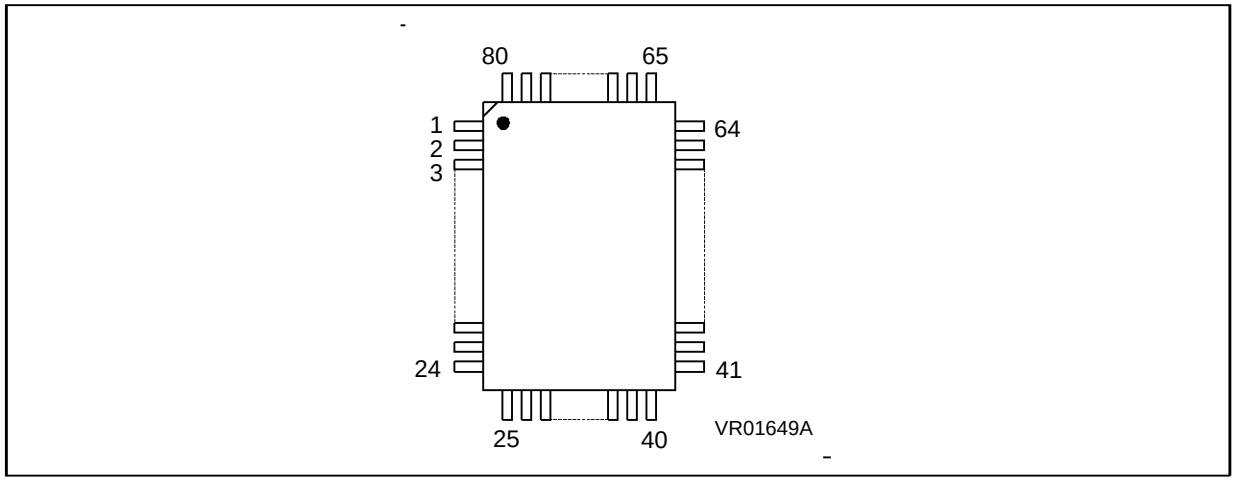
PQFP80



CQFP80-W

(Ordering Information at the end of the datasheet)

Figure 1. 80 Pin Quad Flat Pack (QFP) Package Pinout



ST62E40/T40 Pin Description

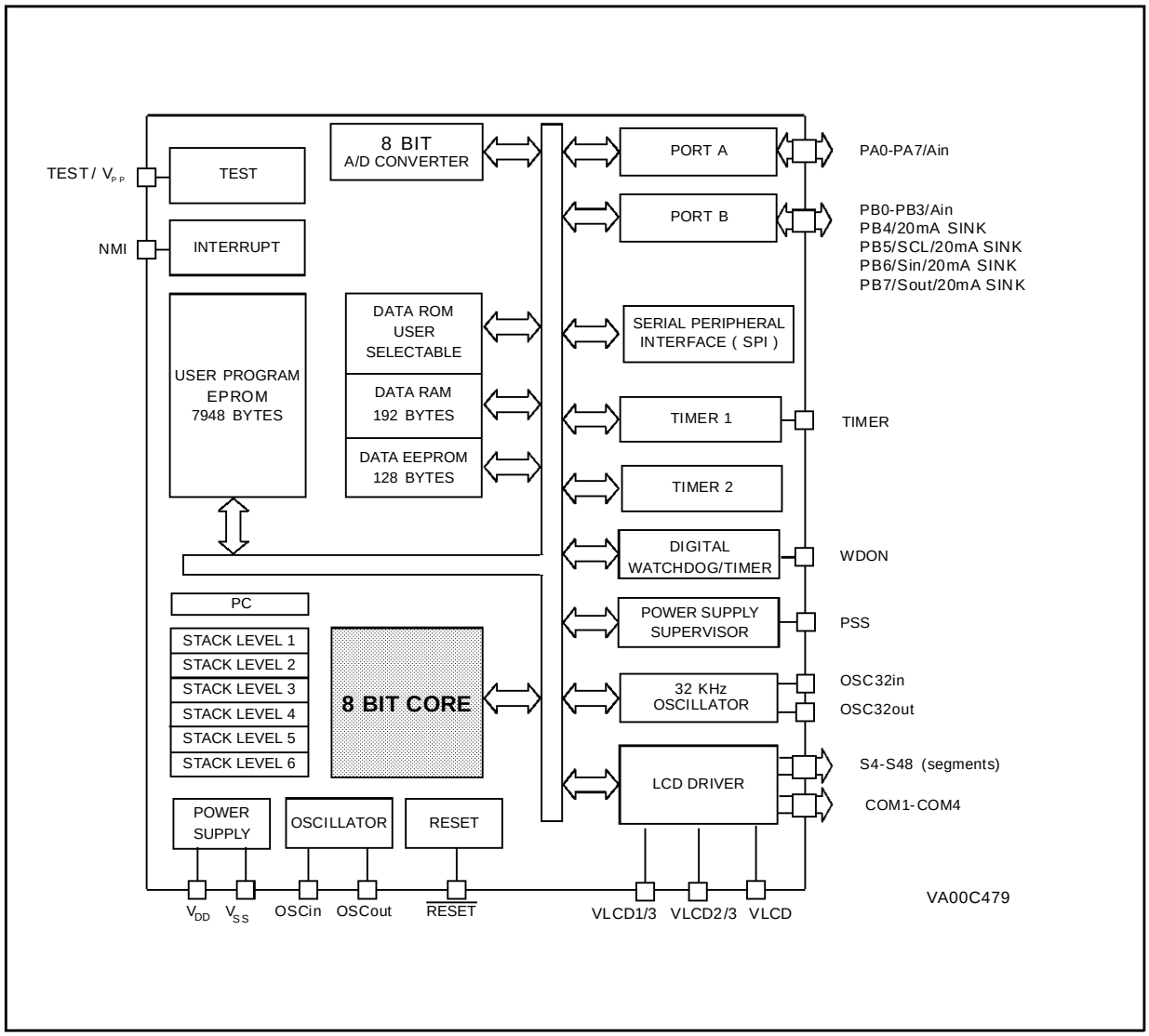
Pin number	Pin name	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
1	S43	25	$\overline{\text{RESET}}$	64	S26	65	S27
2	S44	26	OSCout	63	S25	66	S28
3	S45	27	OSCin	62	S24	67	S29
4	S46	28	WDON	61	S23	68	S30
5	S47	29	NMI	60	S22	69	S31
6	S48	30	TIMER	59	S21	70	S32
7	COM4	31	PB7/Sout (1)	58	S20	71	S33
8	COM3	32	PB6/Sin (1)	57	S19	72	S34
9	COM2	33	PB5/SCL (1)	56	S18	73	S35
10	COM1	34	PB4 (1)	55	S17	74	S36
11	VLCD1/3	35	PB3/Ain	54	S16	75	S37
12	VLCD2/3	36	PB2/Ain	53	S15	76	S38
13	VLCD	37	PB1/Ain	52	S14	77	S39
14	PA7/Ain	38	PB0/Ain	51	S13	78	S40
15	PA6/Ain	39	OSC32out	50	S12	79	S41
16	PA5/Ain	40	OSC32in	49	S11	80	S42
17	PA4/Ain			48	S10		
18	TEST/V_{PP}			47	S9		
19	PA3/Ain			46	S8		
20	PA2/Ain			45	S7		
21	PA1/Ain			44	S6		
22	PA0/Ain			43	S5		
23	V_{DD}			42	S4		
24	V_{SS}			41	PSS		

Note 1: 20mA Sink

GENERAL DESCRIPTION

The ST62E40,T40 microcontrollers are members of the 8-bit HCMOS ST62xx family, a series of devices oriented to low-medium complexity applications. They are the EPROM/OTP versions of the ST6240 ROM device and are suitable for product prototyping and low volume production. All ST62xx members are based on a building block approach: a common core is associated with a combination of on-chip peripherals (macrocells). The macrocells of the ST6240 family are: a high performance LCD controller/driver with 45 segment outputs and 4 backplanes able to drive up to 180 segments, two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), the digital watchdog timer (DWD), an 8-bit A/D Converter with up to 12 analog inputs, a Power Supply Supervisor and an 8-bit synchronous Serial Peripheral Interface (SPI). In addition these devices offer 128 bytes of EEPROM for storage of non volatile data. Thanks to these peripherals the ST6240 family is well suited for general purpose, automotive, security, appliance and industrial applications.

Figure 2. ST62E40 Block Diagram



Note: Ain = Analog Input

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected with the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. OSCin is the input pin, OSCout is the output pin. An external clock signal can be applied to OSCin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to restart the microcontroller at the beginning of its program. The $\overline{\text{RESET}}$ pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TEST/V_{PP}. The TEST must be held at V_{SS} for normal operation (an internal pull-down resistor selects normal operating mode if TEST pin is not connected).

NMI. The NMI pin provides the capability for asynchronous applying an external top priority interrupt to the MCU.This pin is falling edge sensitive. The NMI pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TIMER. This is the TIMER 1 I/O pin. In input mode it is connected to the prescaler and acts as external timer clock or as control gate for the internal timer clock. In the output mode the TIMER pin outputs the data bit when a time-out occurs.

WDON. This pin selects the watchdog enabling option (hardware or software). A low level selects the hardware activated option (the watchdog is always active), a high level selects the software activated option (the watchdog can be activated by software, deactivated only by reset, thus enabling STOP mode). An internal pull-up resistance selects the software watchdog option if the WDON pin is not connected.

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured under software control as an input with or without pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output or as analog input for the A/D converter. Port A has a 5mA drive capability in output mode.

PB0-PB3,PB4-PB7. These 8 lines are organized as one I/O port (B). Each line may be configured under software control as an input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output. PB0-PB3 can be programmed as analog inputs for the A/D converter while PB4-PB7 can also sink 20mA for direct LED driving. PB5-PB7 can also be used as respectively Clock, Data in and Data out pins for the on-chip SPI to carry the synchronous serial I/O signals.

COM1-COM4. These four pins are the LCD peripheral common outputs. They are the outputs of the on-chip backplane voltage generator which is used for multiplexing the 45 LCD lines allowing up to 180 segments to be driven.

S4-S48. These pins are the 45 LCD peripheral driver outputs of ST6240. Segments S1-S3 are not connected to any pin.

VLCD. Display voltage supply. It determines the high voltage level on COM1-COM4 and S4-S48 pins.

VLCD1/3, VLCD2/3. Display supply voltage inputs for determining the display voltage levels on COM1-COM4 and S4-S48 pins during multiplex operation.

PSS. This is the Power Supply Supervisor sensing pin. When the voltage applied to this pin is falling below a software programmed value the highest priority (NMI) interrupt can be generated. This pin has to be connected to the voltage to be supervised.

OSC32in and OSC32out. These pins are internally connected with the on-chip 32kHz oscillator circuit. A 32.768kHz quartz crystal can be connected between these two pins if it is necessary to provide the LCD stand-by clock and real time interrupt. OSC32in is the input pin, OSC32out is the output pin.

ST62E40,T40 EPROM/OTP DESCRIPTION

The ST62E40 is the EPROM version of the ST6240 ROM product. It is intended for use during the development of an application, and for pre-production and small volume production. The ST62T40 OTP has the same characteristics. Both include EPROM memory instead of the ROM memory of the ST6240, and so the program and constants of the program can be easily modified by the user with the ST62E40 EPROM programming board from SGS-THOMSON.

From a user point of view (with the following exception) the ST62E40,T40 products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming of the ST62E40,T40 is described in the User Manual of the EPROM Programming board.

On the ST62E40, all the 8192 bytes of PROGRAM memory are available for the user, as all the EPROM memory can be erased by exposure to UV light. On the ST62T40 (OTP) device) a reserved area for test purposes exists, as for the ST6240 ROM device. In order to avoid any discrepancy between program functionality when using the EPROM, OTP and ROM it is recommended not to use these reserved areas, even when using the ST62E40.

Other than this exception, the ST62E40,T40 parts are fully compatible with the ROM ST6240 equivalent, this datasheet thus provides only information specific to the EPROM based devices.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST6240 ROM-BASED DEVICE FOR FURTHER DETAILS.

EPROM ERASING

The EPROM of the windowed package of the ST62E40 may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST62E40 EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST62E40 package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.

The recommended erasure procedure of the ST62E40 EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST62E40 should be placed within 2.5 cm (1 inch) of the lamp tubes during erasure.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from:

. . . . . Tj = . . T_A + PD x RthJA

Where : . T_A = . . Ambient Temperature.

. . . . . RthJA = Package thermal resistance
. . . . . (junction-to ambient).

. . . . . PD = . Pint + Pport.

. . . . . Pint = . I_{DD} x V_{DD} (chip internal power).

. . . . . Pport = Port power dissipation
. . . . . (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_{LCD}	Display Voltage	-0.3 to 11.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
I_O	Current Drain per Pin Excluding V_{DD} & V_{SS}	± 10	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PQFP80 CQFP80-W		70		°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version 6 Suffix Version	0 -40		70 85	°C
V_{DD}	Operating Supply Voltage		3		6	V
V_{LCD}	Display Voltage		3		10	V

RECOMMENDED OPERATING CONDITIONS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{RM}	RAM Retention Voltage		2			V
f_{OSC}	Oscillator Frequency (1)(4)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01 0.01		8.388 2	MHz
I_{INJ+}	Pin Injection Current (positive) Digital Input (2) Analog Input (3)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input (2) Analog Input	V_{DD} = 4.5 to 5.5V			-5	mA

Notes :

1. An oscillator frequency above 1MHz is recommanded for reliable A/D results.
2. A current of ± 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device. For a positive current injected into one pin, a part of this current (~ 10%) can be expected to flow from the neighbouring pins. A current of -5mA can be forced on one input of the analog section at a time (or -2.5mA for all inputs at a time) without affecting the conversion.
3. If a total current of +1mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 1mA, all the conversion is resulting shifted of +1LSB. If a total positive current of +5mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 5mA, all the conversion is resulting shifted of +2LSB.
4. Operation below 0.01 MHz is possible but requires increased supply current.

EEPROM INFORMATION

The ST62xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	RESET, NMI, TIMER, WDON Pin			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	TIMER	$0.80V_{DD}$			V
		RESET, NMI, WDON Pin	$0.70V_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	RESET Pin V_{DD} =5V $V_{IN} = V_{DD}$ (1) $V_{IN} = V_{DD}$ (2) $V_{IN} = V_{SS}$ (5)			10 1 50	µA mA µA
V_{OL}	Low Level Output Voltage	TIMER, I_{OL} = 5.0mA			$0.2V_{DD}$	V
V_{OH}	High Level Output Voltage	TIMER, I_{OL} = -5.0mA	$0.65V_{DD}$			V

Notes on next page

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{PU}	Pull-up Resistor	V_{IN}=0V V_{DD}=5V WDON - NMI	40	100	200	kΩ
		RESET	200	300	500	kΩ
I_{IL} I_{IH}	Input Leakage Current	TIMER $V_{IN} = V_{DD}$ or V_{SS}		0.1	1.0	µA
I_{IL} I_{IH}	Input Leakage Current	NMI VDD = 5.0V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			 100 1.0	µA
I_{IL} I_{IH}	Input Leakage Current	WDON VDD = 5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			 100 1.0	µA
I_{DD}	Supply Current RUN Mode	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.5V		4	7	mA
	Supply Current WAIT Mode (4)	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		1	2	mA
	Supply Current RESET Mode	f_{OSC} = 8MHz, $V_{RESET} = V_{SS}$		1	7	mA
	Supply Current STOP Mode (3)(4)	I_{LOAD} = 0mA V_{DD} = 5.5V		1	10	µA

Notes :

1. No Watchdog Reset activated.
2. Reset generated by Watchdog.
3. When the watchdog function is actvated the STOP instruction is deactivated. WAIT instruction is automatically executed.
4. All on-chip peripherals in OFF state
5. Pull-up resistor

AC ELECTRICAL CHARACTERISTICS

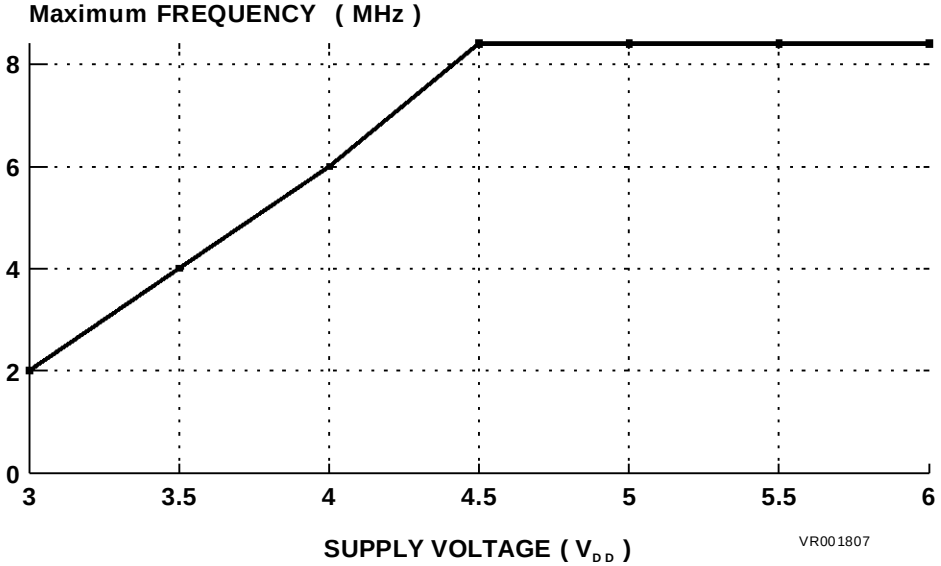
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (2)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01		8.388 2	MHz
t_{SU}	Oscillator Start-up Time	$C_{L1} = C_{L2}$ = 22pF - crystal		5	20	ms
t_{SR}	Supply Rise Time	10% to 90%	0.01		100	
t_{REC}	Supply Recovery Time (1)		100			
T_W	Minimum Pulse Width	NMI Pin V_{DD} = 5V	100			ns
		RESET Pin	100			ns
T_{WEE}	EEPROM Write Time	T_A = 25°C One Byte T_A = 85°C One Byte		5 15	10 25	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A L_{OT} Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention	T_A = 55°C	10			years
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:

1. Period for which V_{DD} has to be connected or at 0V to allow internal Reset function at next power-up.
2. Operation below 0.01 MHz is possible but requires increased supply current.

Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



I/O PORTS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	I/O Pins			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	I/O Pins	$0.7V_{DD}$			V
V_{OL}	Low Level Output Voltage	I/O Pins, I_O = 10µA (sink)			0.1	V
		I/O Pins, $I_{OL} = V_{DD}$x1mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 1.6mA V_{DD} = 3V			0.4	V
	Low Level Output Voltage, PB4-PB7 Only	I/O Pins, $I_{OL} = V_{DD}$x2mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 3.2mA V_{DD} = 3V			0.4	V
		I/O Pins, $I_{OL} = V_{DD}$x4mA V_{DD} = 4.5 to 6V			$0.26xV_{DD}$	V
		I/O Pins, I_{OL} = 6.4mA V_{DD} = 3V			0.8	V
V_{OH}	High Level Output Voltage	I/O Pins, I_O = -10µA (source)	V_{DD}-0.1			V
		I/O Pins, $I_{OL} = -V_{DD}$x1mA V_{DD} = 5.0V	$0.6xV_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	I/O Pins,(1)		0.1	1.0	µA
R_{PU}	Pull-up Resistor	I/O Pins V_{IN} = 0V, V_{DD} = 5.0V	40	100	200	kΩ

Note 1. Pull-up resistor off

SPI ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 V, T_A = -40 to +85°C unless otherwhise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
F_{CL}	Clock Frequency	applied on PB5/SCL			1	MHz
t_{SU}	Set-up Time	applied on PB6/Sin		50		ns
t_h	Hold Time	applied on PB6/Sin		100		ns

A/D CONVERTER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution (3)			8		Bit
A_{TOT}	Total Accuracy (3)	f_{OSC} > 1.2 MHz f_{OSC} > 32kHz			± 2 ±4	LSB
t_C(1)	Conversion Time	f_{OSC} = 8MHz		70		µs
V_{AN}	Conversion Range		V_{SS}		V_{DD}	V
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN}(2)	Analog Input Capacitance			2	5	pF
ASI	Analog Source Impedance				30	kΩ
SSI	Analog Reference Supply Impedence				2	kΩ

Notes:

1. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.
2. Excluding Pad Capacitance
3. Noise at V_{DD},V_{SS} ≤ 10mV

TIMER CHARACTERISTICS
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{12}{f_{OSC}}$			second
f_{IN}	Input Frequency on TIMER Pin				$\frac{f_{OSC}}{8}$	MHz
t_W	Pulse Width at TIMER Pin	$V_{DD} \geq 3V$ $V_{DD} \geq 4.5V$	1 125			µs ns

PSS ELECTRICAL CHARACTERISTICS
(T_A = -40 to +85°C unless otherwhise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{PSS}	PSS pin Input Voltage		V_{SS}		V_{DD}	V
I_{PSS}	PSS pin Input Current	PSS RUN PSS STOP V_{PSS} = 5V, T_A = 25°C			350 1	µA

LCD ELECTRICAL CHARACTERISTICS
(T_A = -40 to +85°C unless otherwise specified)

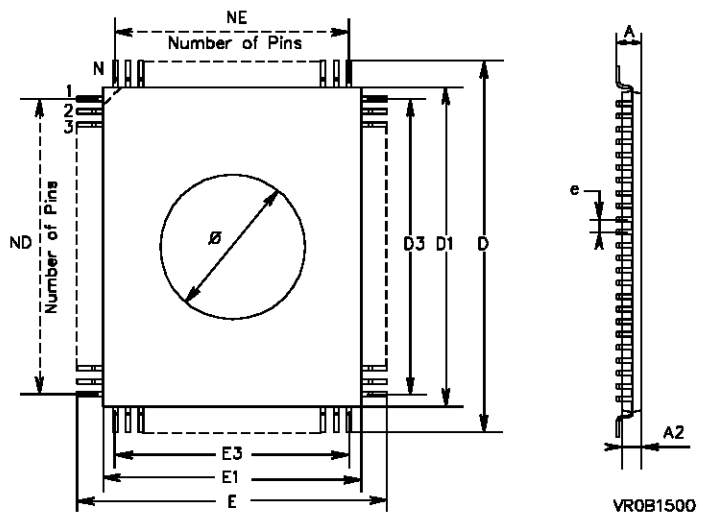
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{FR}	Frame Frequency	1/4 Duty f_{OSC} = 1, 2, 4, 8MHz	16		128	Hz
V_{OS}	DC Offset Voltage(1)	$V_{LCD} = V_{DD}$, no load			50	mV
V_{OH}	COM High Level, Output Voltage	I = 100µA V_{LCD} = 5V	4.5V			V
V_{OL}	COM Low Level, Output Voltage	I = 100µA V_{LCD} = 5V			0.5V	V
V_{OH}	SEG High Level, Output Voltage	I = 50µA V_{LCD} = 5V	4.5V			V
V_{OL}	SEG Low Level, Output Voltage	I = 50µA V_{LCD} = 5V			0.5V	V
V_{LCD}	Display Voltage	Note 2	3		10	V

Notes :

1. The DC offset voltage refers to all segment and common outputs. It is the difference between the measured voltage value and nominal value for every voltage level. Ri of voltage meter must be greater than or equal to 100MΩ.
2. An external resistances network is required when $V_{LCD} \leq 4.5V$.

PACKAGE MECHANICAL DATA

Figure 3. ST62E40 80 Pin Ceramic Quad Flat Package with Window



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		3.55			0.14	
A2		3.40			0.133	
D		23.90			0.941	
D1		20.00			0.787	
D3		18.40			0.724	
E		17.90			0.705	
E1		14.00			0.551	
E3		12.00			0.472	
Ø		7.62			0.3	
e		0.80			0.032	
Number of Pins						
ND	24					
NE	16					
N	80					

ORDERING INFORMATION TABLE

Sales Types	Temperature Range	Package
ST62E40G1	tested at 25°C only	CQFP80-W
ST62T40Q6	-40 to + 85°C	PQFP80

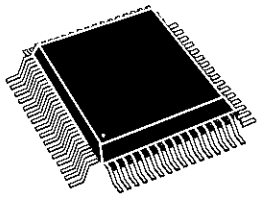
NOTES:

SGS-THOMSON MICROELECTRONICS

ST6242

8-BIT HCMOS MCU WITH LCD DRIVER, AND A/D CONVERTER

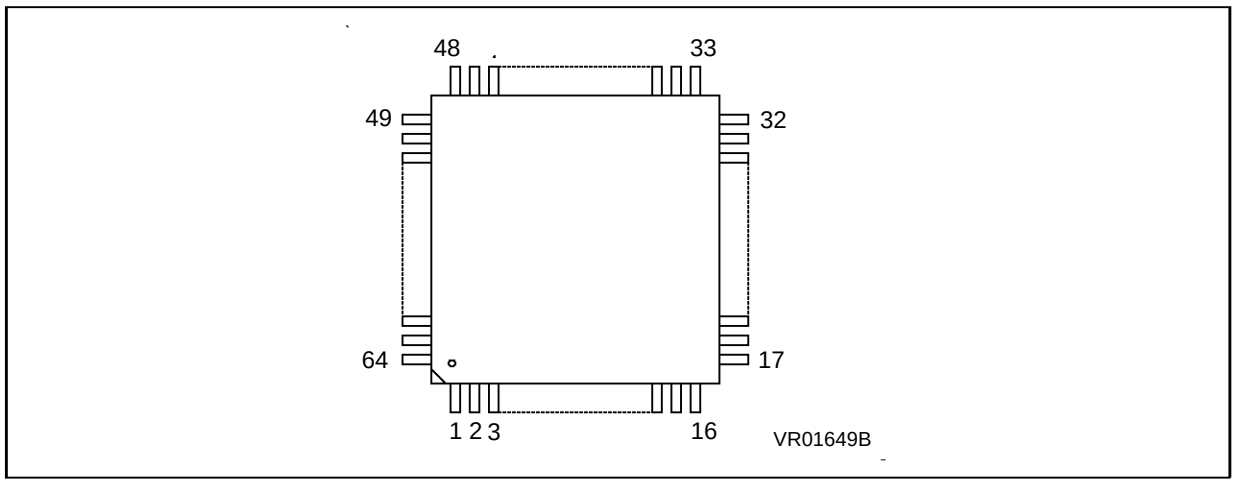
- 3 to 6V supply operating range
- 8.4MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait & Stop Modes
- 5 different interrupt vectors
- Look-up table capability in ROM
- User ROM: 7948 bytes
  Data RAM: 128 bytes
  LCD RAM: 24 bytes
- PQFP64 package
- 10 fully software programmable I/O as:
  - Input with/without pull-up resistor
  - Input with interrupt generation
  - Open-Drain or Push-pull outputs
  - Analog Inputs (6 pins)
- 4 I/O lines can sink up to 20mA for direct LED or TRIAC driving and have SPI alternate functions
- 8-bit counter with 7-bit programmable prescaler
- Software activated digital watchdog
- 8-bit A/D converter with up to 6 analog inputs
- 8-bit synchronous Serial Peripheral Interface (SPI)
- LCD driver with 40 segment outputs, 4 back-plane outputs and selectable duty cycle for up to 160 LCD segments direct driving
- One external not maskable interrupt
- 9 powerful addressing modes
- The accumulator, the X, Y, V & W registers, the port and peripherals data & control registers are addressed in the data space as RAM locations.
- The ST62E42 is the EPROM version, ST62T42 is the OTP version
- Development tool: ST6242-EMU connected via RS232 to an MS-DOS Personal Computer



PQFP64

(Ordering Information at the end of the datasheet)

Figure 1. 64 Pin Quad Flat Pack (QFP) Package Pinout



ST6242 Pin Description

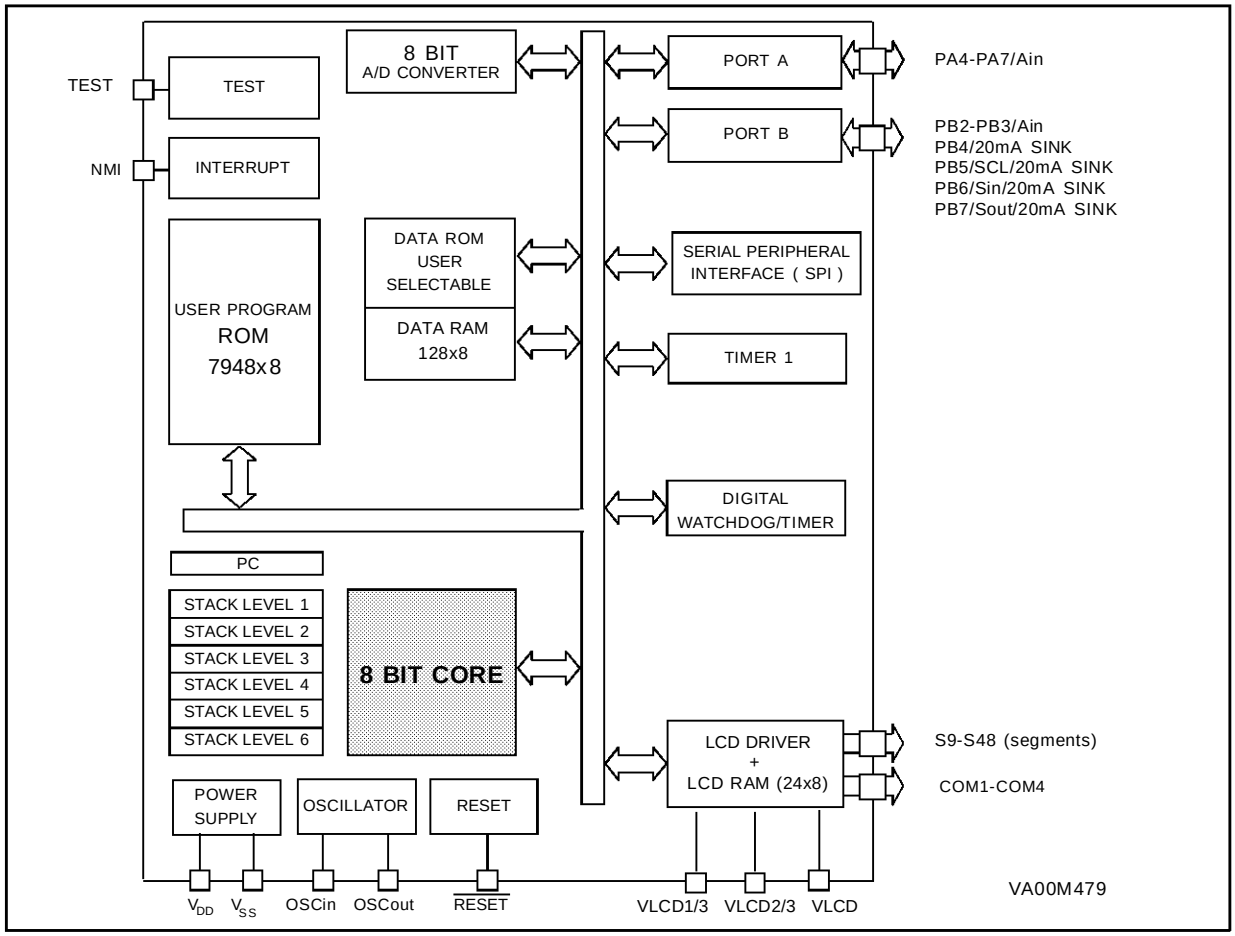
Pin number	Pin name	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
1	S45	17	V_{DD}	33	S13	49	S29
2	S46	18	V_{SS}	34	S14	50	S30
3	S47	19	$\overline{\text{RESET}}$	35	S15	51	S31
4	S48	29	OSCout	36	S16	52	S32
5	COM4	21	OSCin	37	S17	53	S33
6	COM3	22	NMI	38	S18	54	S34
7	COM2	23	PB7/Sout (1)	39	S19	55	S35
8	COM1	24	PB6/Sin (1)	40	S20	56	S36
9	VLCD1/3	25	PB5/SCL (1)	41	S21	57	S37
10	VLCD2/3	26	PB4 (1)	42	S22	58	S38
11	VLCD	27	PB3/Ain	43	S23	59	S39
12	PA7/Ain	28	PB2/Ain	44	S24	60	S40
13	PA6/Ain	29	S9	45	S25	61	S41
14	PA5/Ain	30	S10	46	S26	62	S42
15	PA4/Ain	31	S11	47	S27	63	S43
16	TEST/V_{PP}	32	S12	48	S28	64	S44

Note 1: 20mA SINK

GENERAL DESCRIPTION

The ST6242 microcontroller is a member of the 8-bit HCMOS ST62xx family, a series of devices oriented to low-medium complexity applications. All ST62xx members are based on a building block approach: a common core is associated with a combination of on-chip peripherals (macrocells). The macrocells of the ST6242 are: a high performance LCD controller/driver with 40 segment outputs and 4 backplanes able to drive up to 160 segments. A Timer peripheral including an 8-bit counter with a 7-bit software programmable prescaler (Timer), the digital watchdog timer (DWD), an 8-bit A/D Converter with up to 6 analog inputs, a Power Supply Supervisor and an 8-bit synchronous Serial Peripheral Interface (SPI). Thanks to these peripherals the ST6242 is well suited for general purpose, automotive, security, appliance and industrial applications. The ST62E42 EPROM version is available for prototypes and low-volume production, an OTP version is also available (see separate datasheet).

Figure 2. ST6242 Block Diagram



Note: Ain = Analog Input

PIN DESCRIPTION

V_{DD} and **V_{SS}**. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected with the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. OSCin is the input pin, OSCout is the output pin. An external clock signal can be applied to OSCin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to restart the microcontroller at the beginning of its program. The $\overline{\text{RESET}}$ pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TEST. The TEST must be held at V_{SS} for normal operation (an internal pull-down resistor selects normal operating mode if TEST pin is not connected).

NMI. The NMI pin provides the capability for asynchronous applying an external top priority interrupt to the MCU.This pin is falling edge sensitive. The NMI pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

PA4-PA7. These 4 lines are organized as one I/O port (A). Each line may be configured under software control as an input with or without pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output or as analog input for the A/D converter. Port A has a 5mA drive capability in output mode.

PB0-PB3,PB4-PB7. These 6 lines are organized as one I/O port (B). Each line may be configured under software control as an input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output. PB2-PB3 can be programmed as analog inputs for the A/D converter while PB4-PB7 can also sink 20mA for direct LED driving. PB5-PB7 can also be used as respectively Clock, Data in and Data out pins for the on-chip SPI to carry the synchronous serial I/O signals.

COM1-COM4. These 4 pins are the LCD peripheral common outputs. They are the outputs of the on-chip backplane voltage generator which is used for multiplexing the 40 LCD lines allowing up to 160 segments to be driven.

S9-S48. These pins are the 40 LCD peripheral driver outputs of ST6242. Segments S1-S8 are not connected to any pin.

VLCD. Display voltage supply. It determines the high voltage level on COM1-COM4 and S9-S48 pins.

VLCD1/3, VLCD2/3. Resistor network nodes for determining the intermediate display voltage levels on COM1-COM4 and S9-S48 pins during multiplex operation.

ST62xx CORE

The core of the ST62xx Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control busses. The in-core communication is arranged as shown in Figure 3; the controller being externally linked to both the reset and the oscillator, while the core is linked to the dedicated on-chip macrocells peripherals via the serial data bus and indirectly for interrupt purposes through the control registers.

Registers

The ST62xx Family core has six registers and three pairs of flags available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs.

Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is addressed in the data space as RAM location at address FFh. Accordingly, the ST62xx instruction set can use the accumulator as any other register of the data space.

Figure 4. ST62xx Core Programming Model

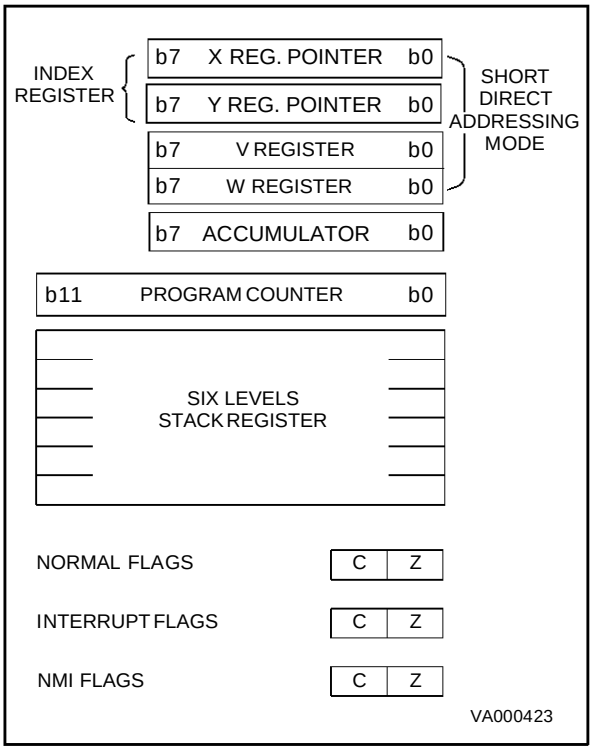
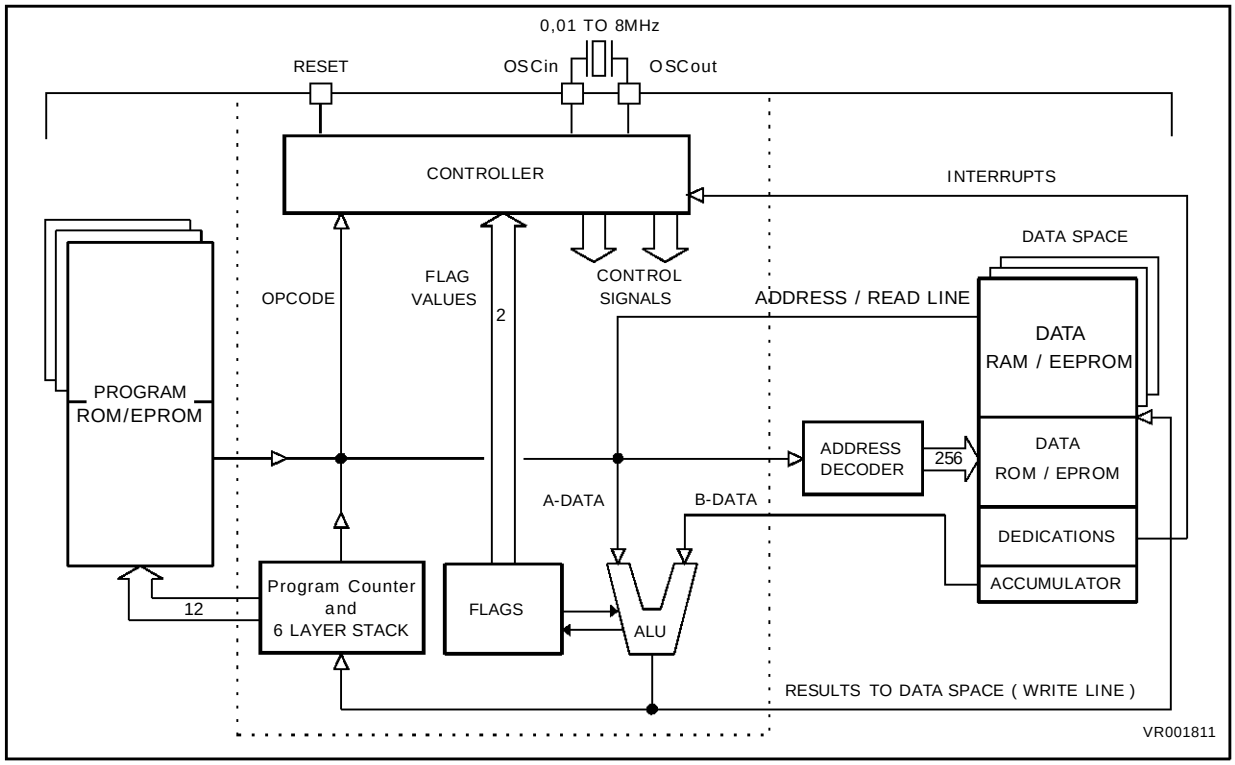


Figure 3.ST62xx Core Block Diagram



ST62xx CORE (Continued)

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in the data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST62xx instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save one byte in short direct addressing mode. These registers can be addressed in the data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST62xx instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or an address of operand. The 12-bit length allows the direct addressing of 4096 bytes in the program space. Nevertheless, if the program space contains more than 4096 locations, as for the ST6242, the further program space can be addressed by using the Program ROM Page register.
The PC value is incremented after it is read from the address of the current instruction. To execute relative jumps the PC and the offset are shifted through the ALU, where they will be added, and the result is shifted back into the PC. The program counter can be changed in the following ways:

- JP (Jump) instruction . . . PC=Jump address
- CALL instruction . . . . PC=Call address
- Relative Branch instructions . . . . . PC=PC ± offset
- Interrupt . . . . . PC=Interrupt vector
- Reset . . . . . PC=Reset vector
- RET & RETI instructions . PC=Pop (stack)
- Normal instruction . . . . PC=PC+1

Flags (C, Z)

The ST62xx core includes three pairs of flags that correspond to 3 different modes: normal mode, interrupt mode and Non-Maskable Interrupt-Mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during normal operation, one pair is used during the interrupt mode (CI, ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

The ST62xx core uses the pair of flags that correspond to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST62xx core uses the interrupt flags (resp. the NMI flags) instead of the normal flags. When the RETI instruction is executed, the normal flags (resp. the interrupt flags) are restored if the MCU was in the normal mode (resp. in the interrupt mode) before the interrupt. It should be observed that each flag set can only be addressed in its own mode (Not-maskable interrupt, normal interrupt or main mode). The flags are not cleared during the context switching and so remain in the state they were at the exit of the last mode switch.

The Carry flag is set when a carry or a borrow occurs during arithmetic operations, otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The Zero flag is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

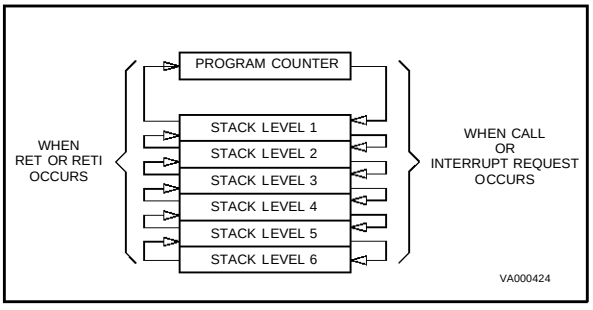
The switching between the three sets of Flags is automatically performed when an NMI, an interrupt or a RETI instruction occurs. As the NMI mode is automatically selected after the reset of the MCU, the ST62xx core uses at first the NMI flags.

ST62xx CORE (Continued)

Stack

The ST62xx core includes a true LIFO hardware stack that eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be lost). When a subroutine or interrupt return occurs (RET or RETI instructions), the first level register is shifted back into the PC and the value of each level is popped back into the previous level. These two operating modes are described in Figure 5. Since the accumulator, as all other data space registers, is not stored in the stack, the handling of these registers should be performed inside the subroutine. The stack pointer will remain in its deepest position if more than 6 calls or interrupts are executed, so that the last return address will be lost. It will also remain in its highest position if the stack is empty and a RET or RETI is executed. In this case the next instruction will be executed.

Figure 5. Stack Operation



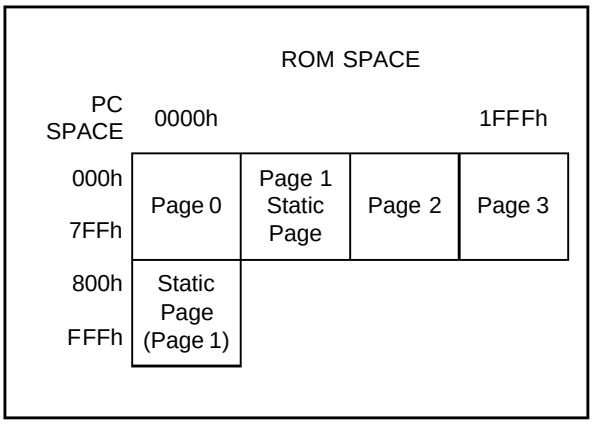
MEMORY SPACES

The MCU operates in three different memory spaces: program space, data space, and stack space. A description of these spaces is shown in the following figures.

Program Space

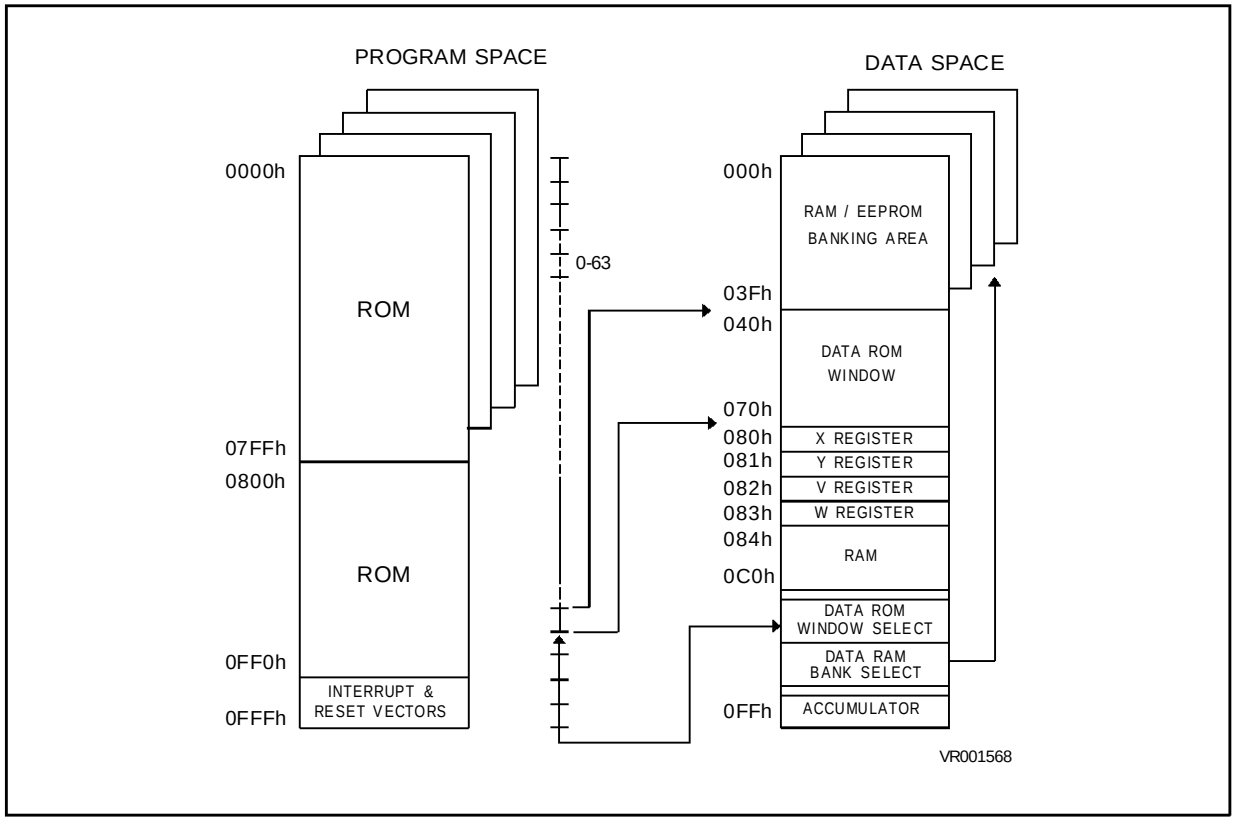
The program space is physically implemented in the ROM memory and includes all the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, the reserved test area and the user vectors. It is addressed by the 12-bit Program Counter register (PC register) and so the ST62xx core can directly address up to 4K bytes of Program Space. Nevertheless, the Program Space can be extended by the addition of 2Kbyte ROM banks as it is shown in the following figure in which the ST6242 8Kbyte memory is described.

Figure 6. ST6242 8Kbytes Program Space Addressing Description



MEMORY SPACES (Continued)

Figure 7. ST62xx Memory Addressing Description Diagram



These banks are addressed in the 000h-7FFh locations of the Program Space by the Program Counter and by writing the appropriate code in the Program ROM Page Register (PRPR register) located at address CAh of the Data Space. Because interrupts and common subroutines should be available all the time, only the lower 2K byte of the 4K program space are bank switched while the upper 2K byte can be seen as static page. Table 2 gives the different codes that allow the selection of the corresponding banks. Note that, from the memory point of view, Page 1 and the Static Page represent the same physical memory: it is only two different ways of addressing the same locations. On the ST6242 a total of 8192 bytes of ROM have been implemented; 7948 are available as user ROM while 244 are reserved for SGS-THOMSON test purposes.

Table 1. ST6242 Program ROM Memory Map

ROM Page	Device Address	Description
Page 0	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 “STATIC”	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	Reserved User ROM
Page 3	0000h-000Fh 0010h-07FFh	Reserved User ROM

MEMORY SPACES (Continued)

Data Space

The instruction set of the ST62xx core operates on a specific space, named Data Space, that contains all the data necessary for the processing of the program. The Data Space allows the addressing of RAM memory, ST62xx core/peripheral registers, and read-only data such as constants and look-up tables.

Data ROM

All the read-only data is physically implemented in the ROM memory in which the Program Space is also implemented. The ROM memory contains consequently the program to be executed, the constants and the look-up tables needed for the program.

The locations of Data Space in which the different constants and look-up tables are addressed by the ST62xx core can be considered as being a 64-byte window through which it is possible to access to the read-only data stored in the ROM memory .

Data RAM Addressing

The ST6242 offers 128 bytes of data RAM memory. 64 bytes of RAM are directly addressed in data space in the range 084h-0BFh (static space). The additional RAM is addressed using the banks of 64 bytes located between addresses 00h and 3Fh.

Additionally 24 bytes of RAM devoted to LCD dat are available from E0h to F7h and are not banked.

Stack Space

The stack space consists of six 12 bit registers that are used for stacking subroutine and interrupt return addresses plus the current program counter register.

Figure 8. ST6242 Data Memory Space

DATA RAM/EEPROM BANK AREA	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
SPI INT. DISABLE REGISTER	0C2h*
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
RESERVED	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
PROGRAM ROM PAGE REGISTER	0CAh*
DATA RAM PAGE REGISTER	0CBh*
PORT A OPTION REGISTER	0CCh
RESERVED	0CDh
PORT B OPTION REGISTER	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER PRESCALER REGISTER	0D2h
TIMER COUNTER REGISTER	0D3h
TIMER STATUS/CONT REGISTER	0D4h
RESERVED	0D5h 0D6h 0D7h
WATCHDOG REGISTER	0D8h
RESERVED	0D9h 0DAh 0DBh
LCD MODE CONTROL REGISTER	0DCh
SPI DATA REGISTER	0DDh
RESERVED	0DEh
WRITE: 40h	0DFh
LCD RAM	0E0h 0F7h
DATA RAM 7 BYTES	0F8h 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

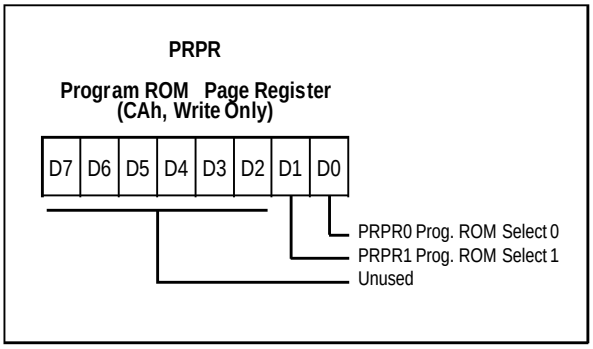
MEMORY SPACES (Continued)

Program ROM Page Register (PRPR)

The PRPR register can be addressed like a RAM location in the Data Space at the address CAh; nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to select the 2-Kbyte ROM bank of the Program Space that will be addressed. The number of the page has to be loaded in the PRPR register. Refer to the Program Space description for additional information concerning the use of this register. The PRPR register is not modified when an interrupt or a subroutine occurs.

Care is required when handling the PRPR register as it is write only. For this reason, it is not allowed to change the PRPR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. This operation may be necessary if common routines and interrupt service routines take more than 2K bytes; in this case it could be necessary to divide the interrupt service routine into a (minor) part in the static page (start and end) and to a second (major) part in one of the dynamic pages. If it is impossible to avoid the writing of this register in interrupt service routines, an image of this register must be saved in a RAM location, and each time the program writes to the PRPR it must write also to the image register. The image register must be written before PRPR, so if an interrupt occurs between the two instructions the PRPR is not affected.

Figure 9. Program ROM Page Register



D7-D2. These bits are not used.

PRPR1-PRPR0. These are the program ROM banking bits and the value loaded selects the corresponding page to be addressed in the lower part of the 4K program address space as specified in Table 2.

Table 2. ST6242 8Kbytes Program ROM Page Register Coding

PRPR1	PRPR0	PC bit 11	Memory Page
X	X	1	Static Page (Page1)
0	0	0	Page 0
0	1	0	Page 1 (Static Page)
1	0	0	Page 2
1	1	0	Page 3

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Note:

Only the lower part of address space is bank switched because interrupt vectors and common subroutines should be available at all times. The reason of this structure is due to the fact that it is not possible to jump from one dynamic page to another except by jumping back to the static page, changing contents of PRPR, and then jumping to a different dynamic page.

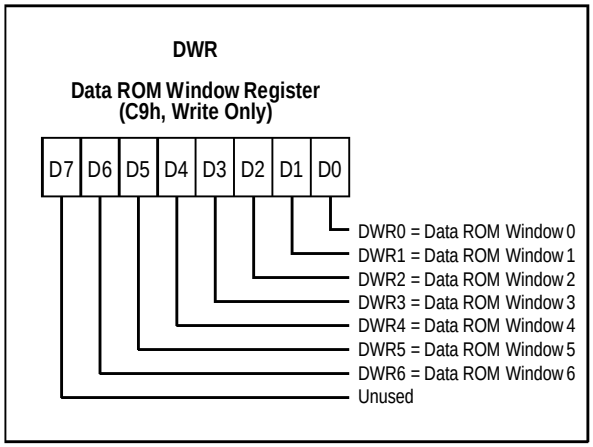
MEMORY SPACES (Continued)

Data Window register (DWR)

The Data ROM window is located from address 040h to address 7Fh in the Data space. It allows the direct reading of 64 consecutive bytes located anywhere in the ROM memory between the addresses 0000h and 1FFFh. All the bytes of the ROM memory can be used to store either instructions or read-only data. Indeed, the window can be moved by step of 64 bytes along the ROM memory in writing the appropriate code in the Write-only Data Window register (DWR register, location C9h).

The DWR register can be addressed like a RAM location in the Data Space at the address C9h, nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to move the 64-byte read-only data window (from the 40h address to 7Fh address of the Data Space) up and down the ROM memory of the MCU in steps of 64 bytes. The effective address of the byte to be read as a data in the ROM memory is obtained by the concatenation of the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits, see Figure 10). So when addressing location 40h of dataspace, and 0 is loaded in the DWR register, the phisycal addressed location in ROM is 00h. The DWR register is not cleared at reset, therefore it must be written to before the first access to the Data ROM window area.

Figure 11. Data ROM Window Register



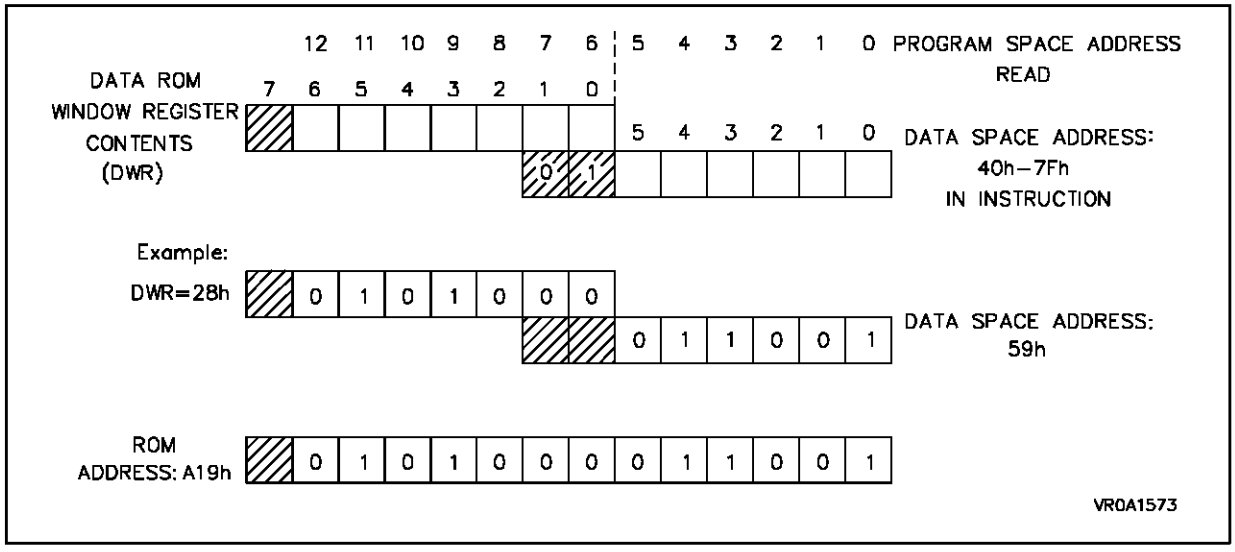
D7. This bit is not used.

DWR6-DWR0. These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Note: Care is required when handling the DWR register as it is write only. For this reason, it is not allowed to change the DWR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in the interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to the DWR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DWR is not affected.

Figure 10. Data ROM Window Memory Addressing



MEMORY SPACES (Continued)

Data RAM Bank Register (DRBR)

The selection of the bank is made by programming the Data RAM Bank Switch register (DRBR register) located at address CBh of the Data Space. The number of the selected bank is equal to the bit content of the DRBR register. In this way each bank of RAM can be selected 64 bytes at a time. No more than one bank should be set at a time.

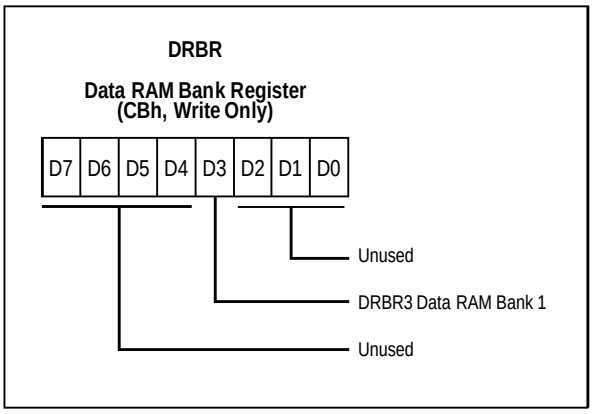
The DRBR register can be addressed like a RAM location in the Data Space at the address CBh; nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to select the desired 64-byte RAM bank of the Data Space. The number of the bank has to be loaded in the DRBR register and the instruction has to point to the selected location as if it was in bank 0 (from 00h address to 3Fh address). This register is not cleared during the MCU initialization, therefore it must be written before the first access to the Data Space bank region. Refer to the Data Space description for additional information. The DRBR register is not modified when a interrupt or a subroutine occurs.

The following table 3 summarizes how to set the data RAM bank register in order to select the various banks or pages.

Table 3. Data RAM Register Set-up

DRBR Value	Selection
08h	RAM Page 1

Figure 12. Data RAM Bank Register



D7-D4. These bits are not used.

DRBR3. This bit, when set, will select the RAM page.

D2-D0. These bits are not used.

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Notes:

Care is required when handling the DRBR register as it is write only. For this reason, it is not allowed to change the DRBR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to DRBR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DRBR is not affected.

In DRBR Register, *only 1 bit must be set*. Otherwise two or more pages are enabled in parallel, producing errors.

TEST MODE

For normal operation the TEST pin must be held low. An on-chip 100kΩ pull-down resistor is internally connected to the TEST pin.

INTERRUPTS

The ST62xx core can manage 4 different maskable interrupt sources, plus one non-maskable interrupt source (top priority level interrupt). Each source is associated with a particular interrupt vector that contains a Jump instruction to the related interrupt service routine. Each vector is located in the Program Space at a particular address.
When a source provides an interrupt request, and the request processing is also enabled by the ST62xx core, then the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction). Finally, the PC is loaded with the address of the Jump instruction and the interrupt routine is processed.

The ST6242 microcontroller has six different interrupt sources associated to different interrupt vectors as described in Table 4.

Table 4. Interrupt Vectors - Sources Relationship

Interrupt Source	Associated Vector	Vector Address
NMI Pins	Interrupt Vector #0 (NMI)	(FFCh-FFDh)
SPI Peripheral	Interrupt Vector #1	(FF6h-FF7h)
Port A & B Pins	Interrupt Vector #2	(FF4h-FF5h)
TIMER	Interrupt Vector #3	(FF2h-FF3h)
ADC Peripheral	Interrupt Vector #4	(FF0h-FF1h)

Interrupts Vectors Description

The ST62xx core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines in the static page of the Program Space:

– The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at addresses FFCh,FFDh in the Program Space. On ST6242 this vector is associated with the external falling edge sensitive interrupt pin (NMI). An on-chip 100kΩ pull-up resistor is internally connected to the NMI pin.

– The interrupt vector located at the addresses FF6h, FF7h is named interrupt vector #1. It is associated with SPI peripheral and can be programmed by software to generate an interrupt request after the falling edge or low level of the eighth external clock pulse according to the code loaded in the Interrupt Option Register (IOR).

– The interrupt vector located at the addresses FF4h, FF5h is named interrupt vector #2. It is associated with Port A and B pins and can be programmed by software either in the falling edge detection mode or in the rising edge detection mode according to the code loaded in the Interrupt Option Register (IOR).

– The interrupt vector located at the addresses FF2h, FF3h is named interrupt vector #3. It is associated with Timer.

– The interrupt vector located at the addresses FF0h, FF1h is named interrupt vector #4. It is associated with the A/D converter peripheral.

All the on-chip peripherals (refer to their descriptions for further details) have an interrupt request flag bit (TMZ for timer, EOC for A/D, etc.), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI for timer, EAI for A/D, etc.) that must be set to one to allow the transfer of the flag bit to the Core.

Interrupt Priority

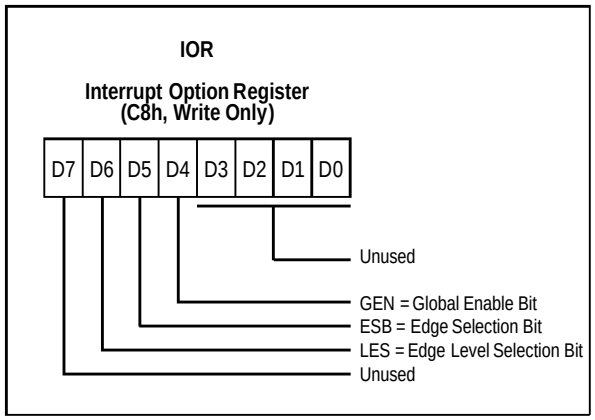
The non-maskable interrupt request has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the four other interrupts cannot interrupt each other. If more than one interrupt request is pending, they are processed by the ST62xx core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

INTERRUPTS (Continued)

Interrupt Option Register

The Interrupt Option Register (IOR register, location C8h) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register can be addressed in the Data Space as RAM location at the address C8h, nevertheless it is a write-only register that cannot be accessed with single-bit operations. The operating modes of the external interrupt inputs associated to interrupt vectors #1 and #2 are selected through bits 5 and 6 of the IOR register.

Figure 13. Interrupt Option Register



D7. This bit is not used.

LES. Level/Edge Selection Bit. When this bit is set to one, the interrupt #1 (SPI) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

ESB. Edge Selection Bit. When this bit is set to one, the interrupt #2 (Port A & B lines) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

GEN. Global Enable Interrupt. When this bit is set to one, all the interrupts are enabled. When this bit is cleared to zero all the interrupts (but NMI) are disabled.

This register is cleared on reset.

Table 5. Interrupt Option Register Description

GEN	SET	Enable all interrupts
	CLEARED	Disable all interrupts
ESB	SET	Rising edge mode on interrupt input #2
	CLEARED	Falling edge mode on interrupt input #2
LES	SET	Level-sensitive mode on interrupt input #1
	CLEARED	Falling edge mode on interrupt input #1
OTHERS	NOT USED	

External Interrupts Operating Modes

The NMI interrupt is associated to the NMI pin of the ST6242. The two interrupt requests are “ORed”. The highest priority interrupt request will be generated by a falling edge applied to the NMI pin. The NMI interrupt pin signal is latched and is automatically reset by the core at the beginning of the non-maskable interrupt service routine. An on-chip pull-up resistor and a schmitt trigger is available with the NMI pin.

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (SPI vector #1, Ports A and B vector #2,) are connected to two internal latches. Each latch is set when a falling/rising edge occurs and is cleared when the associated interrupt routine is started. So, the occurrence of an external interrupt request is stored: a second interrupt, that occurs during the processing of the first one, will be processed as soon as the first one has been finished (if there is not an higher priority interrupt request). If more than one interrupt occurs during the processing of the first one, these other interrupt requests will be lost.

The storage of the interrupt requests is not available in the level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the core samples the line after the execution of the instructions.

During the end of each instruction the core tests the interrupt lines and if there is an interrupt request the next instruction is not executed and the related interrupt routine is executed.

INTERRUPTS (Continued)

Interrupt Procedure. The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event the user does not know about the context and the time at which it occurred. As a result the user should save all the data space registers which will be used inside the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes which are automatically switched and so these do not need to be saved.

The following list summarizes the interrupt procedure:

ST62xx actions

– Interrupt detection
– The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (or the NMI flags)
– The value of the PC is stored in the first level of the stack
– The normal interrupt lines are inhibited (NMI still active)
– First internal latch is cleared
– The related interrupt vector is loaded in the PC.

User actions

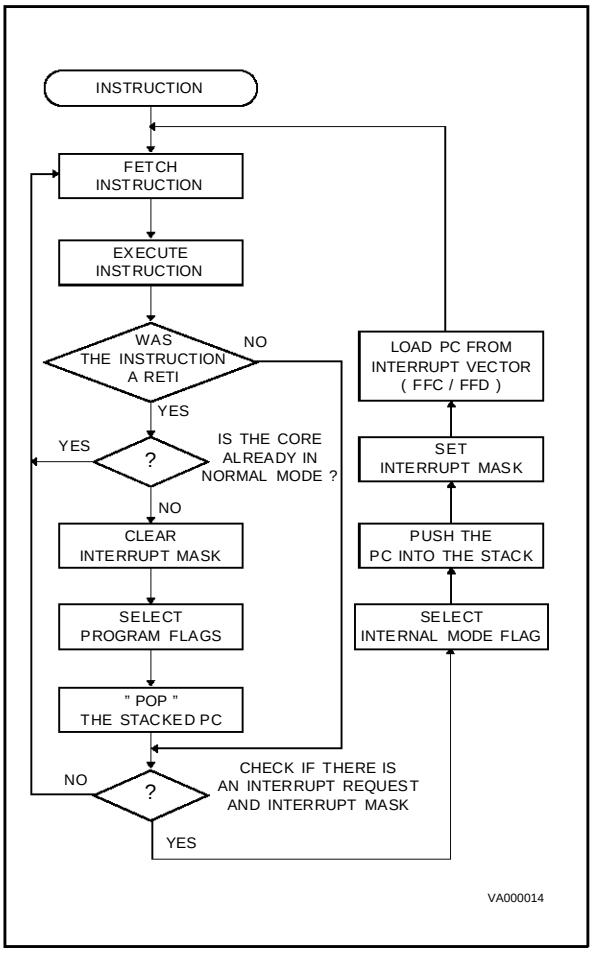
– User selected registers are saved inside the interrupt service routine (normally on a software stack)
– The source of the interrupt is found by polling (if more than one source is associated to the same vector) the interrupt flag of the source.
– Interrupt servicing
– Return from interrupt (RETI)

ST62xx actions

– Automatically the ST62xx core switches back to the normal flags (or the interrupt flags) and pops the previous PC value from the stack

The interrupt routine begins usually by the identification of the device that has generated the interrupt request (by polling). The user should save the registers which are used inside the interrupt routine (that holds relevant data) into a software stack. After the RETI instruction execution, the core carries out the previous actions and the main routine can continue.

Figure 14. Interrupt Processing Flow-Chart



WARNING. *GEN is the global enable for all interrupts except NMI. If this bit is cleared, the NMI interrupt is accepted when the ST62xx core is in the normal RUN Mode.*
If the ST62xx core is in STOP or WAIT Mode, the NMI is not accepted as a restart is disabled. This state can only be finished by a reset (fromthe Watchdog or an external Reset Signal).
As a consequence the NMI can be masked in STOP and WAIT modes, but not in RUN mode.

INTERRUPTS (Continued)

Interrupt request and mask bits

Interrupt Option Register, IOR Location C8h

– GEN. If this bit is set, all the ST62xx interrupts are enabled, if reset all interrupts are disabled (including the NMI).
– ESB. If this bit is set, all the input lines associated to interrupt vector #2 are rising edge sensitive, if reset they are falling edge sensitive.
– LES. If this bit is set, all the inputs lines associated to interrupt vector #1 are low level sensitive, if reset they are falling edge sensitive.

All other bits in this register are not used.

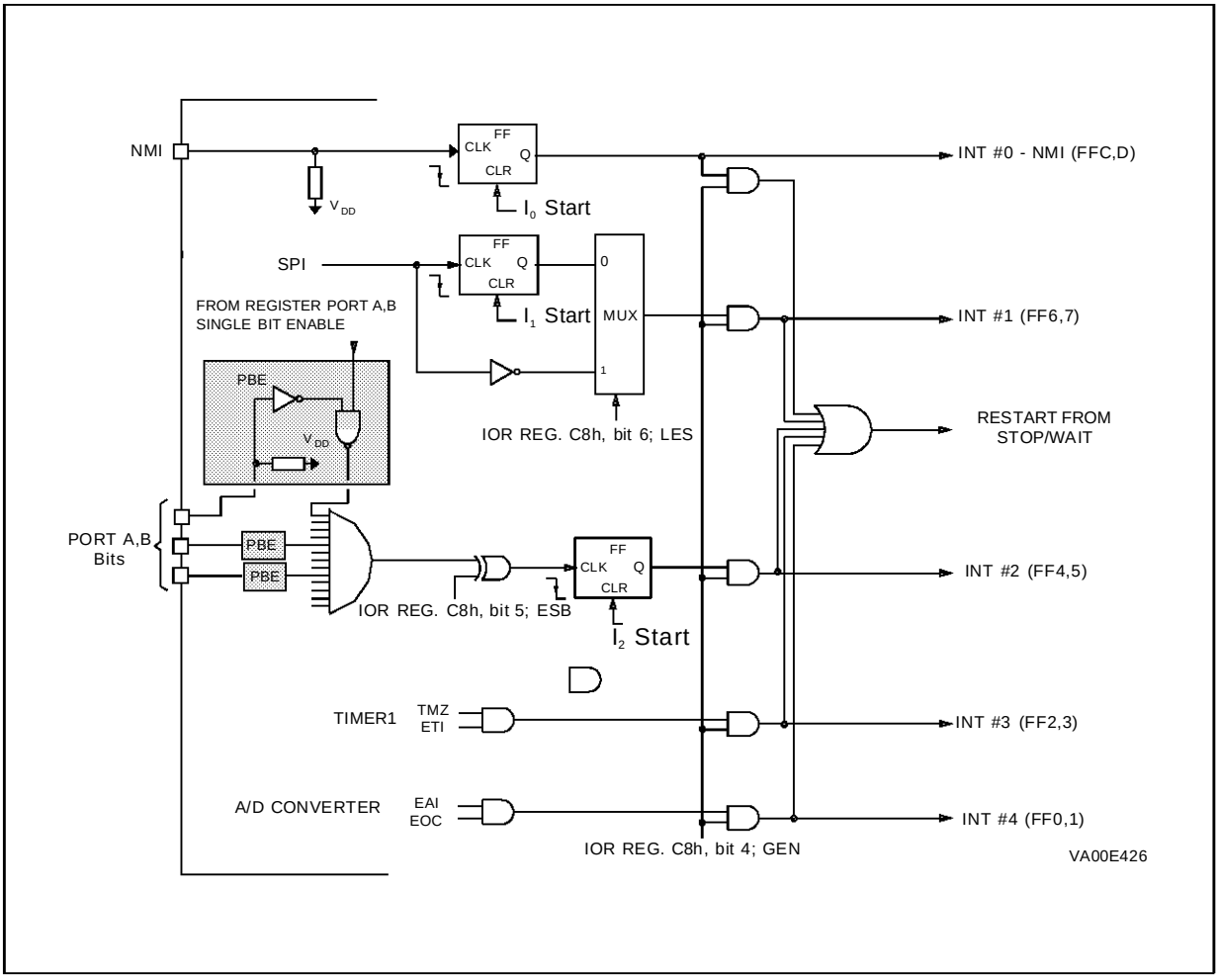
Timer Peripherals, TSCR1 and TSCR2 registers, locations D4h and D7h

– TMZ. A low-to-high transition indicates that the timer count register has decremented to zero. This means that an interrupt request can be generated in relation to the state of ETI bit.
– ETI. This bit, when set, enables the timer interrupt request.

A/D Converter Peripheral, ADCR register location D0h

– EOC. This read only bit indicates when a conversion has been completed, by going to one. An interrupt request can be generated in relation to the state of EAI bit.
– EAI. This bit, when set, enables the A/D converter interrupt request.

Figure 15. ST6242 Interrupt Circuit Diagram



RESET

The ST6242 can be reset in three ways: by the external reset input (RESET) tied low, by power-on reset and by the digital watchdog/timer peripheral.

RESET Input

The RESET pin can be connected to a device of the application board in order to restart the MCU during its operation. The activation of the RESET pin may occur in the RUN, WAIT or STOP mode. This input has to be used to reset the MCU internal state and provide a correct start-up procedure. The pin is active low. The internal reset signal is generated by adding a delay to the external signal. Therefore even short pulses at the RESET pin will be accepted. This feature is valid providing that V_{DD} has finished its rising phase and the oscillator is correctly running (normal RUN or WAIT modes).

If RESET activation occurs in the RUN or Wait mode, the MCU is configured in the Reset mode for as long as the signal of the RESET pin is low. The processing of the program is stopped (in RUN mode only) and the Input/Outputs are in the High-impedance state with pull-up resistors switched on. As soon as the level on the RESET pin becomes high, the initialization sequence is executed.

If a RESET pin activation occurs in the STOP mode, the oscillator starts and all the inputs/outputs are configured in the High-impedance state with pull-up resistors switched on for as long as the level on the RESET pin remains low. When the level of the RESET pin becomes high, a delay is generated by the ST62xx core to wait that the oscillator becomes completely stabilized. Then, the initialization sequence is started.

Power-On Reset (POR)

The function of the POR consists in waking up the MCU during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: every Input/Output port is configured in the input mode (High-impedance state with pull-up) and no instruction is executed. When the power supply voltage becomes sufficient, the oscillator starts to operate, nevertheless the ST62xx core generates a delay to allow the oscillator to be completely stabilized before the execution of the first instruction. The initialization sequence is then executed.

Internal circuitry generates a Reset pulse when V_{DD} is switched on. In the case of fast rising V_{DD} (transition time $\leq$ 100µs), this reset pulse starts the internal reset procedure without the need of external components at the RESET pin. In cases of slowly or non monotonously rising V_{DD}, an external reset signal must be provided for a proper reset of the MCU.

For as long as the reset pin is kept at the low level, the processor remains in the reset state. The reset will be released after the voltage at the reset pin reaches the high level.

Note:

To have a correct ST62xx start-up, the user should take care that the reset input does not change to the high level before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see recommended operating conditions).

An on-chip counter circuit provides a delay of 2048 oscillator cycles between the detection of the reset high level and the release of the MCU reset.

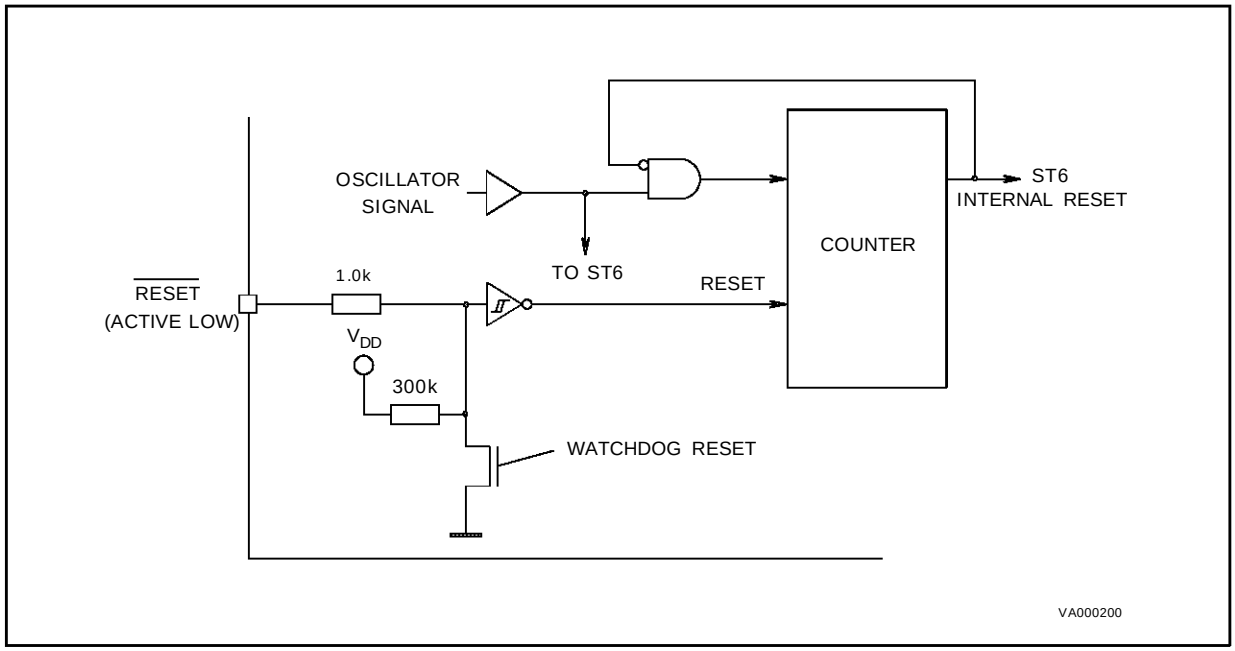
A proper reset signal for slow rising V_{DD}, i.e. the required delay between reaching sufficient operating voltage and the reset input changing to a high level, can be generally provided by an external capacitor connected between the RESET pin and V_{SS}.

RESET (Continued)

Watchdog Reset

The ST6242 provides an on-chip watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed, preventing the end-of-count being reached, an internal circuit pulls down the $\overline{\text{RESET}}$ pin. The MCU will enter the reset state as soon as the voltage at $\overline{\text{RESET}}$ pin reaches the related low level. This also resets the watchdog which subsequently turns off the pull-down and activates the pull-up device at the $\overline{\text{RESET}}$ pin. This causes the positive transition at the $\overline{\text{RESET}}$ pin and terminates the reset state.

Figure 16. Reset Circuit

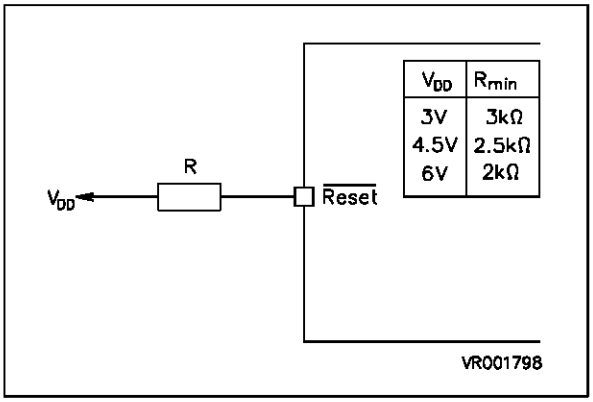


Application Notes

An external resistor between V_{DD} and reset pin is not required because an internal pull-up device is provided. If the user prefers, for any reason, to add an external pull-up resistor its value must comply with the Rmin value defined in Figure 17. If the value is lower than Rmin, the on-chip watchdog pull-down transistor might not be able to pull-down the reset pin resulting in an external deactivation of the watchdog function.

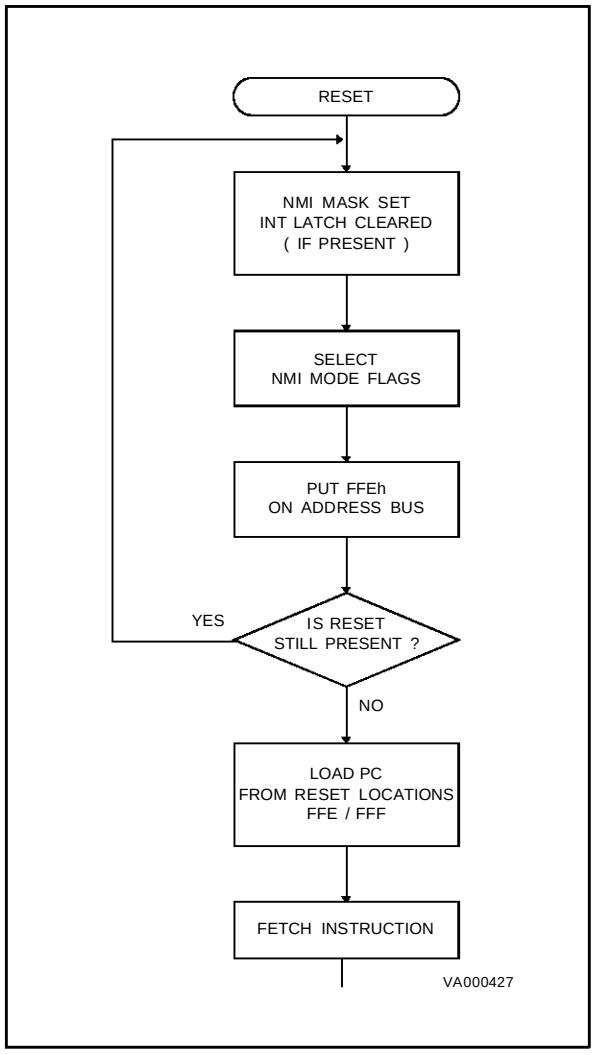
The POR function operates in a dynamic manner in the way that it brings about the initialization of the MCU when it detects a dynamic rising edge of the V_{DD} voltage. The typical detected threshold is about 2 volts, but the actual value of the detected threshold depends on the way in which the V_{DD} voltage rises up. The POR device DOES NOT allow the supervision of a static rising or falling edge of the V_{DD} voltage.

Figure 17. External Reset Resistance



RESET (Continued)

Figure 18. Reset & Interrupt Processing Flow-Chart



MCU Initialization Sequence

When a reset occurs the stack is reset to the program counter, the PC is loaded with the address of the reset vector (located in the program ROM at addresses FFEh & FFFh). A jump instruction to the beginning of the program has to be written into these locations. After a reset the interrupt mask is automatically activated so that the core is in non-maskable interrupt mode to prevent false or ghost interrupts during the restart phase. Therefore the restart routine should be terminated by a RETI instruction to switch to normal mode and enable interrupts. If no pending interrupt is present at the end of the reset routine the ST62xx will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced.

Figure 19. Restart Initialization Program Flow-Chart

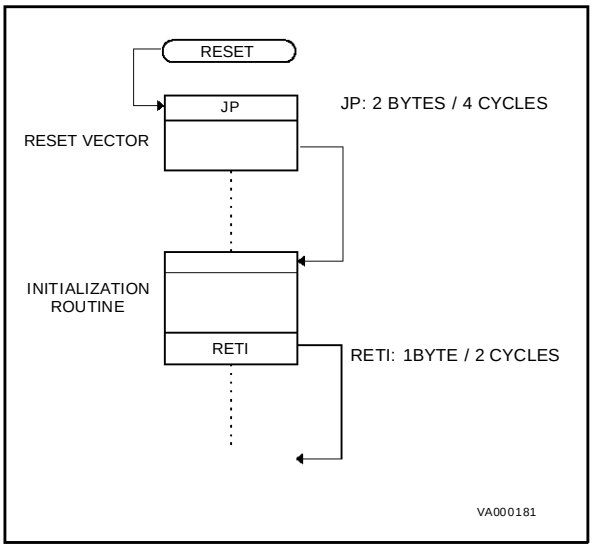


Table 6. Reset Configuration

Input/Output pins	Registers
Input Mode with pull-up and no interrupt	All cleared but A,X,Y,V,W, data RAM, LCD RAM, DWR (C9), PRPR (CA), DRBR (CB). Timers prescaler and TCR are initialized respectively at 7F and FF. Watchdog register DWDR (D8) is set to FEh.

WAIT & STOP MODES

The WAIT and STOP modes have been implemented in the ST62xx core in order to reduce the consumption of the product when the latter has no instruction to execute. These two modes are described in the following paragraphs

WAIT Mode

The configuration of the MCU in the WAIT mode occurs as soon as the WAIT instruction is executed. The microcontroller can also be considered as being in a "software frozen" state where the core stops processing the instructions of the routine, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage but where the peripherals are still working.

The WAIT mode is used when the user wants to reduce the consumption of the MCU when it is in idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide a clock signal to the peripherals. The timer counting may be enabled (writing the PSI bit in TSCR register) and the timer interrupt may be also enabled before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. The above explanation related to the timers applies also to the A/D converter.

If the exit from the WAIT mode is performed with a general RESET (either from the activation of the external pin or by watchdog reset) the MCU will enter a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behavior depends on the state of the ST62xx core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case will be described in the following paragraphs. In any case, the ST62xx core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

If the Watchdog is disabled the STOP mode is available. When in STOP mode the MCU is placed in the lowest power consumption mode. In this operating mode the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or Reset activation to output from the STOP state.

If the exit from the STOP mode is performed with a general RESET (by the activation of the external pin) the MCU will enter a normal reset procedure as described in the RESET chapter. The case of an interrupt depends on the state of the ST62xx core before the initialization of the STOP sequence and also of the kind of the interrupt request that is generated.

This case will be described in the following paragraphs. In any case, the ST62xx core generates a delay after the occurrence of the interrupt request in order to wait the complete stabilization of the oscillator before the execution of the first instruction.

Exit from WAIT and STOP Modes

The following paragraphs describe the output procedure of the ST62xx core from WAIT and STOP modes when an interrupt occurs (not a RESET). It must be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) before the start of the WAIT or STOP sequence, but also of the type of the interrupt request that is generated.

Normal Mode. If the ST62xx core was in the main routine when the WAIT or STOP instruction has been executed, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs; the related interrupt routine is executed and at the end of the interrupt service routine the instruction that follows the STOP or the WAIT instruction is executed if no other interrupts are pending.

WAIT & STOP MODES (Continued)

Not Maskable Interrupt Mode. If the STOP or WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs: the instruction that follows the STOP or the WAIT instruction is executed and the ST62xx core is still in the non-maskable interrupt mode even if another interrupt has been generated.

Normal Interrupt Mode. If the ST62xx core was in the interrupt mode before the initialization of the STOP or WAIT sequence, it outputs from the stop or wait mode as soon as any interrupt occurs. Nevertheless, two cases have to be considered:

– If the interrupt is a normal interrupt, the interrupt routine in which the WAIT or STOP was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST62xx core is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance to their priority.
– If the interrupt is a non-maskable interrupt, the non-maskable routine is processed at first. Then the routine in which the WAIT or STOP was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST62xx core remains in the normal interrupt mode.

Notes:

To reach the lowest power consumption the user software must take care of:

– placing the A/D converter in its power down mode by clearing the PDS bit in the A/D control register before entering the STOP instruction.
– switching off the 32kHz oscillator by clearing the oscillator start/stop bit in the 32kHz oscillator control register.
– putting the EEPROM on-chip memory in standby mode by writing 40h in EEPROM Control Register (address DFh).

The LCD Driver peripheral is automatically switched-off by the STOP instruction when the 32kHz oscillator operation is not selected.

When the watchdog has been enabled, the STOP instruction is deactivated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

If all the interrupt sources are disabled (including NMI if GEN="0"), the restart of the MCU can only be done by a $\overline{\text{RESET}}$ activation. The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

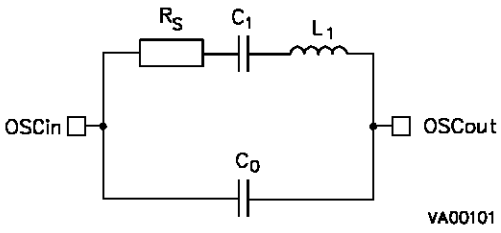
ON-CHIP CLOCK OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. A crystal, a ceramic resonator, or an external signal (provided to the OSCin pin) may be used to generate a system clock with various stability/cost tradeoffs. The different clock generator options connection methods are shown in Figure 21.

One machine cycle takes 13 oscillator pulses; 12 clock pulses are needed to increment the PC while and additional 13th pulse is needed to stabilize the internal latches during memory addressing. This means that with a clock frequency of 8MHz the machine cycle is 1.625µs.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance (CL), IC parameters, ambient temperature, supply voltage. It must be observed that the crystal or ceramic leads and circuit connections must be as short as possible.Typical values for CL1, CL2 are 15-22pF for a 4/8MHz crystal. The oscillator output frequency is internally divided by 13 to produce the machine cycle and by 12 to produce the Timer, the Watchdog and the A/D peripheral clock. A machine cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five machine cycles to be executed.

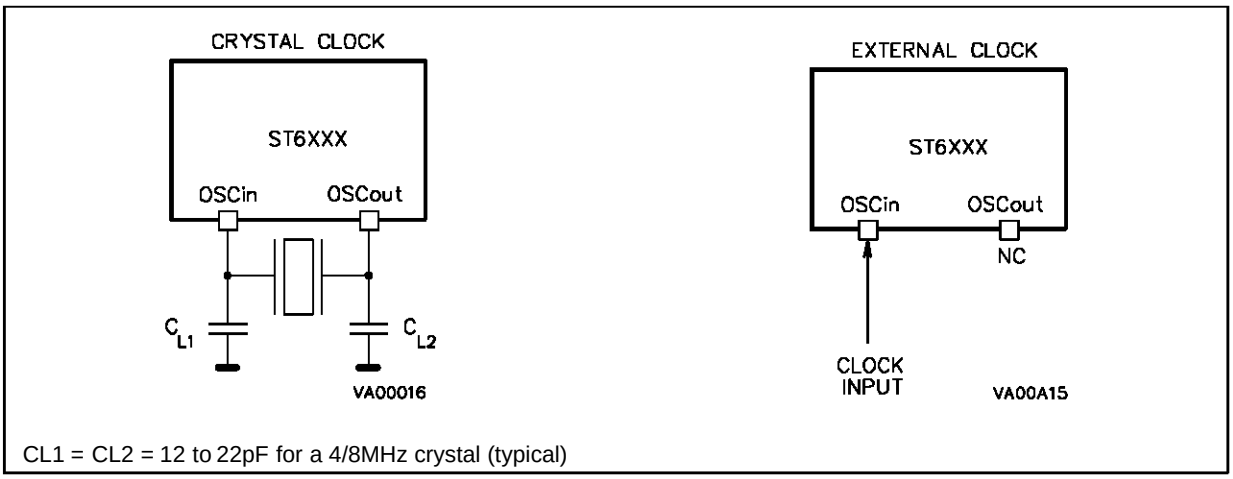
Figure 20. Crystal Parameters



Crystal parameters: AT-Cut Parallel Resonance Crystal
C0 = Parallel Resonance Capacitance

ON-CHIP CLOCK OSCILLATOR (Continued)

Figure 21. Oscillator Connection



CL1 = CL2 = 12 to 22pF for a 4/8MHz crystal (typical)

INPUT/OUTPUT PORTS

The ST6242 microcontroller has 10 Input/Output lines that can be individually programmed either in the input mode or the output mode with the following options that can be selected by software:

– Input without pull-up and without interrupt
– Input with pull-up and with interrupt
– Input with pull-up without interrupt
– Analog inputs (PA4-PA7, PB2-PB3)
– SPI control signals (PB5-PB7)
– Push-pull output
– Standard Open drain output
– 20mA Open drain output (PB4-PB7)

The lines are organized in two ports (port A,B).

Each port occupies 3 registers in the data space. Each bit of these registers is associated with a particular line (for instance, the bits 0 of the Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The two DATA registers (DRA, DRB), are used to read the voltage level values of the lines programmed in the input mode, or to write the logic value of the signal to be output on the lines configured in the output mode. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by the user software, in conjunction with the related option registers, to select the different input mode options.

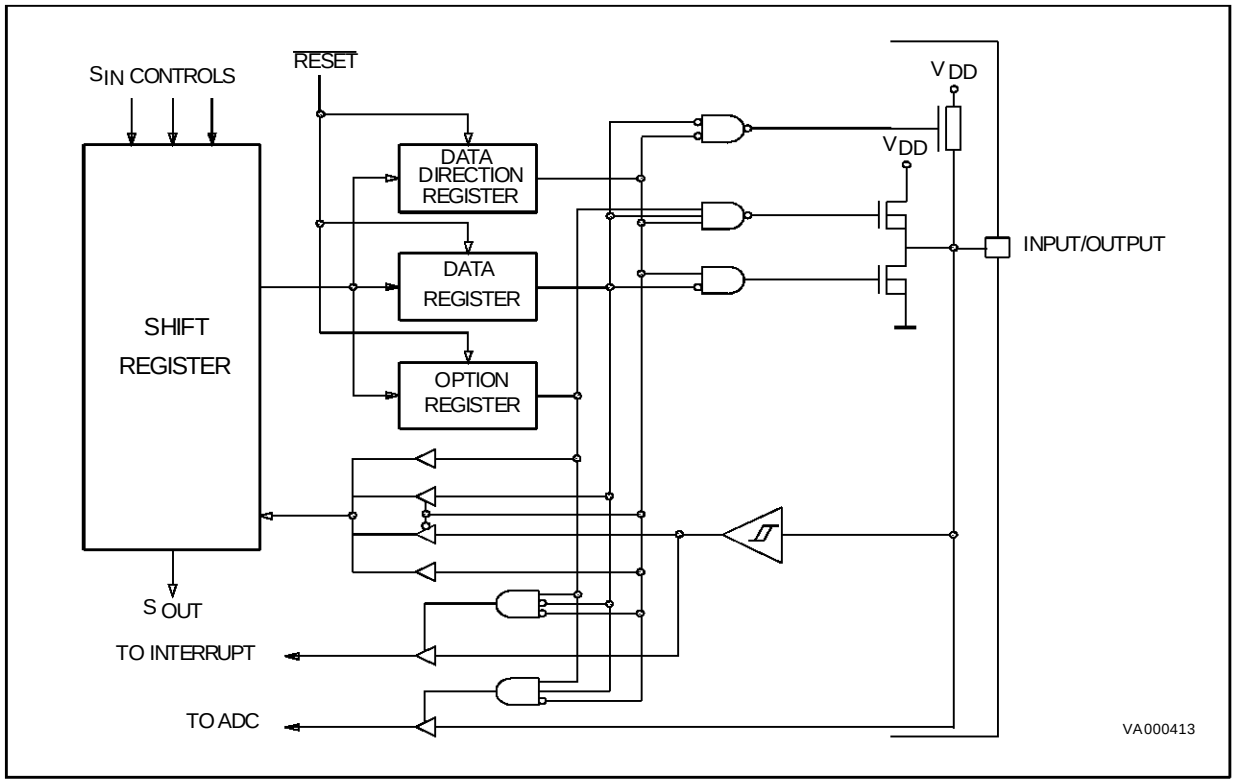
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The two Data Direction registers (DDRA, DDRB) allow the selection of the data direction of each pin (input or output).

The two Option registers (ORPA, ORPB) are used to select the different port options available both in input and in output mode.

All the I/O registers can be read or written as any other RAM location of the data space, so no extra RAM cell is needed for port data storing and manipulation. During the initialization of the MCU, all the I/O registers are cleared and the input mode with pull-up/no-interrupt is selected on all the pins, thus avoiding pin conflicts.

Figure 22. I/O Port Block Diagram



INPUT/OUTPUT PORTS (Continued)

I/O Pin Programming

Each pin can be individually programmed as input or output with different input and output configurations.

This is achieved by writing to the relevant bit in the data (DR), data direction register (DDR) and option registers (OR). Table 7 shows all the port configurations that can be selected by user software.

Input Option Description

Pull-up, High Impedance Option. All the input lines can be individually programmed with or without an internal pull-up according to the codes programmed in the OR and DR registers . If the pull-up option is not selected, the input pin is in the high-impedance state.

Interrupt Option. All the input lines can be individually connected by software to the interrupt lines of the ST62xx core according to the codes programmed in the OR and DR registers. The pins of Port A and B are "ORed" and are connected to the interrupt associated to the vector #2. The interrupt modes (falling edge sensitive, rising edge sensitive) can be selected by software for each port by programming the IOR register.

Analog Input Option. The six PA4-PA7, PB2-PB3 pins can be configured to be analog inputs according to the codes programmed in the OR and DR registers. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as analog input at a time, otherwise the selected inputs will be shorted.

Figure 23. I/O Port Data Registers

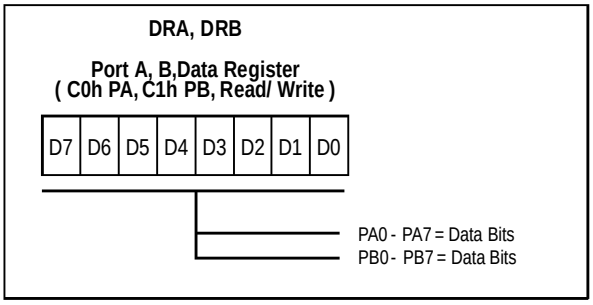


Figure 24. I/O Port Data Direction Registers

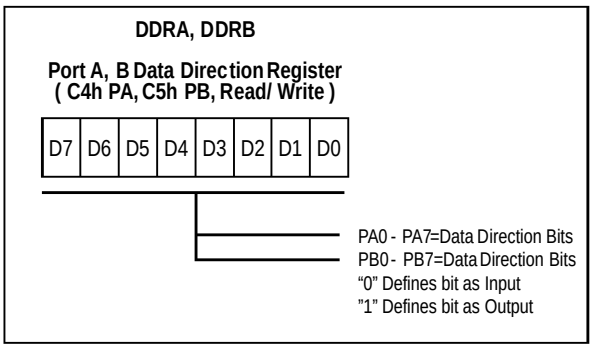
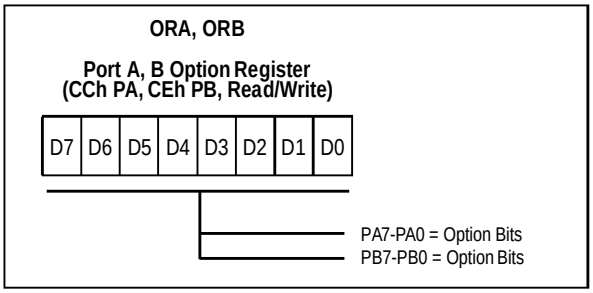


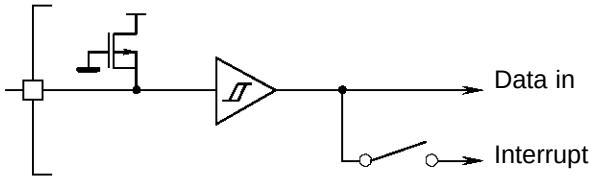
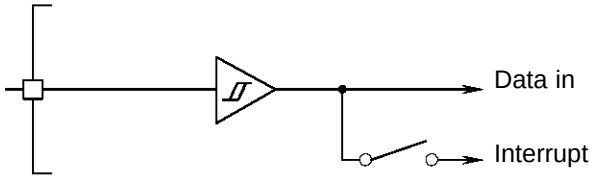
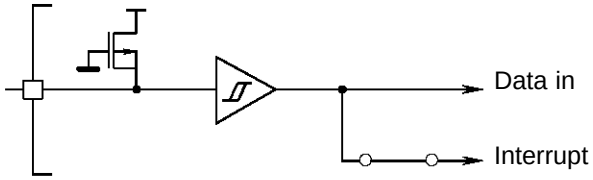
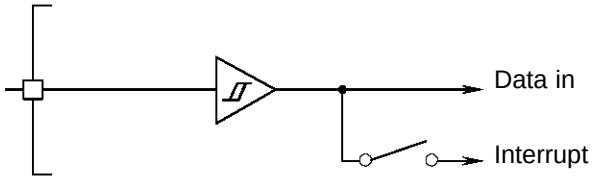
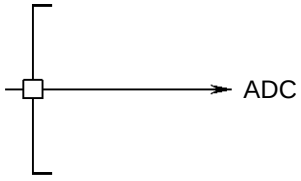
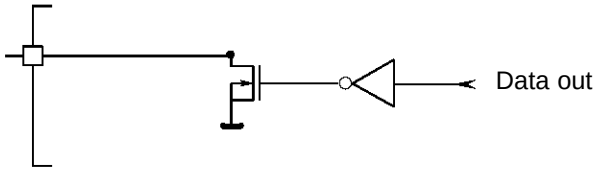
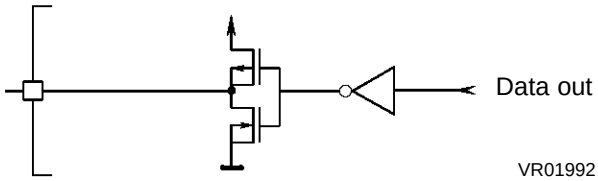
Figure 25. I/O Port Option Registers



Note: For complete coding explanation refer to Table 7.

INPUT/OUTPUT PORTS (Continued)

Table 7. I/O Port Configuration

DDR, OR	DR	Mode	Option	Schematic	
0	0	0	Input	Pull-up No interrupt (RESET state)	
0	0	1	Input	No pull-up No interrupt	
0	1	0	Input	Pull-up Interrupt	
0	1	1	Input	No pull-up No interrupt PB4-PB7	
			Input	Analog input: PA0-PA7 PB2-PB3	
1	0	X	Output	Open drain: 20mA PB4-PB7 Open drain: 5mA PB2-PB3, PA5-PA7	
1	1	X	Output	Push-pull	

INPUT/OUTPUT PORTS (Continued)

SPI alternate function Option. The I/O pins PB5-PB7 are also used by serial peripheral interface SPI. PB5 is connected with the SPI clock input SCL, PB6 is connected with the SPI data input SIN and PB7 is connected with the SPI data output SOUT.

For serial input operation PB5 and PB6 have to be programmed as inputs. For serial output operation PB7 has to be programmed as open-drain output (DDR = "1", OPR = "0"). In this operating mode the output of the SPI shift register instead of the port data register is connected to the port buffer. When PB7 is programmed as push-pull output (DDR = "1", OPR = "1"), the port data register is connected to the port buffer. When the SPI peripheral is not used PB5-PB7 can be used as general purpose I/O lines (provided that PB7 is not selected to be open-drain in output mode).

Notes:

Switching the I/O ports from one state to another should be done in a way that no unwanted side effects can happen. The recommended safe transitions are shown below. All other transitions are risky and should be avoided during change of operation mode as it is most likely that there will be an unwanted side-effect such as interrupt generation or two pins shorted together by the analog input lines.

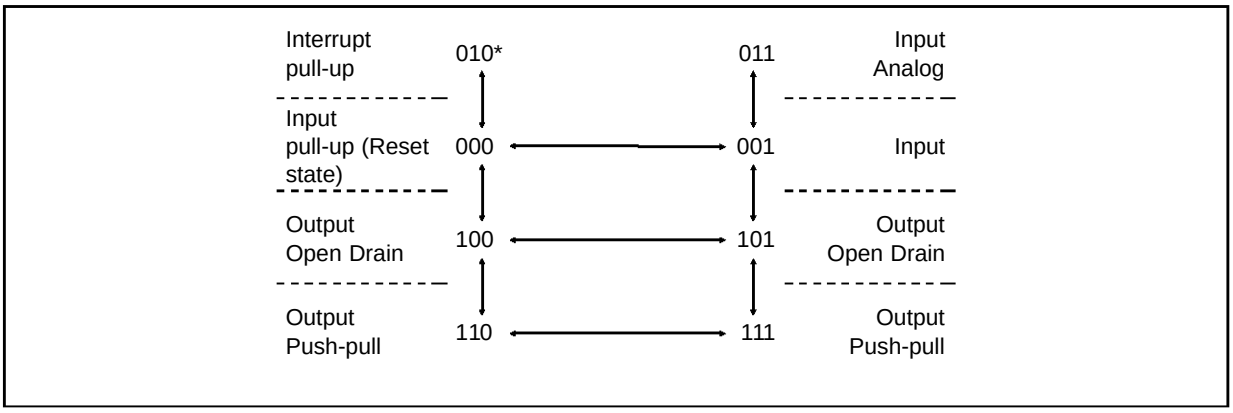
Single bit SET and RES instructions should be used very carefully with Port A and B data registers because these instructions make an implicit read and write back of the whole addressed register byte. In port input mode however data register address reads from input pins, not from data register latches and data register information in input mode is used to set characteristics of the input pin (interrupt, pull-up, analog input), therefore these characteristics may be unintentionally reprogrammed depending on the state of input pins. As general rule is better to use SET and RES instructions on data register only when the whole port is in output mode. If input or mixed configuration is needed it is recommended to keep a copy of the data register in RAM. On this copy it is possible to use single bit instructions, then the copy register could be written into the port data register.

```
SET     bit, datacopy
LD      a, datacopy
LD      DRA, a
```

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user has to take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance in the measurement.

Figure 26. I/O Port State Transition Diagram for Safe



Note *. xxx = DDR, OR, DR Bits respectively

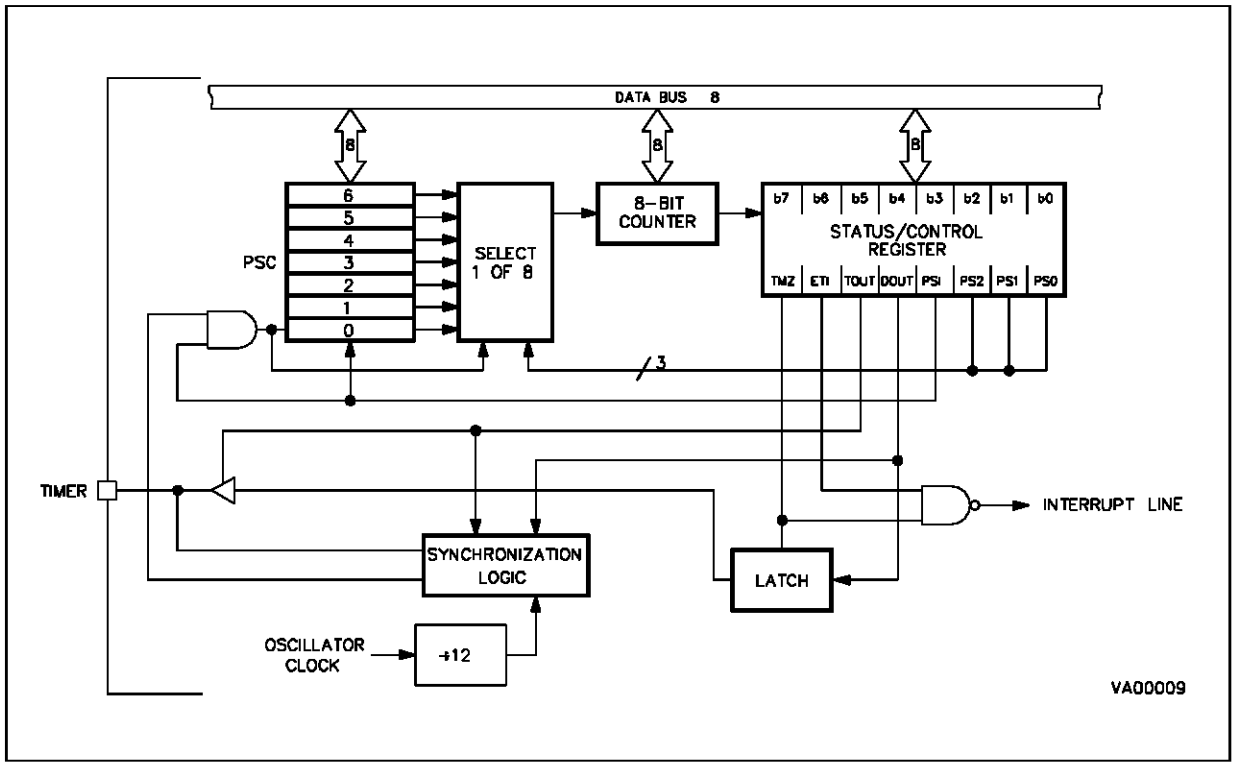
TIMERS

The ST6242 offers an on-chip Timer peripheral consisting of an 8-bit counter with a 7-bit programmable prescaler. This Timer gives a maximum count of 2^{15}, and contains a control logic that allows the configuration of the peripheral in three operating modes. Figure 27 shows the Timer block diagram. The content of the 8-bit counter can be read/written in the Timer/Counter register TCR which is addressed in the data space as a RAM location at addresses D3h. The state of the 7-bit prescaler can be read in the PSC register at address D2h. The control logic device is managed in the TSCR register (address D4h) as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero) bit in the TSCR is set to one. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set to one an interrupt request, associated to interrupt vector #3, is generated. The Timer interrupt can be used to exit the MCU from the WAIT mode.

The prescaler input is the oscillator frequency divided by 12. The prescaler input decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR, the clock input of the timer/counter register is multiplexed to different sources. On division factor 1, the clock input of the prescaler is also that of timer/counter; on factor 2, bit 0 of prescaler register is connected to the clock input of TCR. This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC is connected to clock input of TCR, and so on. On division factor 128, the MSB bit 6 of PSC is connected to the clock input of TCR. The prescaler initialize bit (PSI) in the TSCR register must be set to one to allow the prescaler (and hence the counter) to start. If it is cleared to zero then all of the prescaler bits are set to one and the counter is inhibited from counting. The prescaler can be given any value between 0 and 7Fh by writing to address D2h, if bit PSI in the TSCR register is set to one. The tap of the prescaler is selected using the PS2,PS1,PS0 bits in the control register. Figure 28 shows the Timer working principle.

Figure 27. Timer Peripheral Block Diagram



TIMERS (Continued)

Timer Operating Modes

The Timer has one operating mode. This mode is selected with TOUT= "1" in the Timer status control register TSCR (D4h).

The prescaler is decremented by the timer clock (OSC/12). The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits.

When the TCR count reach 0, it sets the TMZ bit in the TSCR.

Timer Interrupt

When the counter register decrements to zero and the software controlled ETI (Enable Timer Interrupt) bit is set to one then an interrupt request associated to interrupt vector #3 is generated. When the counter decrements to zero also the TMZ bit in the TSCR register is set to one.

Notes:

TMZ is set when the counter reaches 00h; however, it may be set by writing 00h in the TCR register or setting bit 7 of the TSCR register. TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFh while the 7-bit prescaler is loaded to 7Fh , and the TSCR register is cleared which means that timer is stopped (PSI="0") and the timer interrupt is disabled.

If the Timer is programmed in output mode, DOUT bit is transferred to the TIMER pin when TMZ is set to one (by software or due to counter decrement). When TMZ is high, the latch is transparent and DOUT is copied to the timer pin. When TMZ goes low, DOUT is latched.

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

Figure 29. Timer Status Control Register

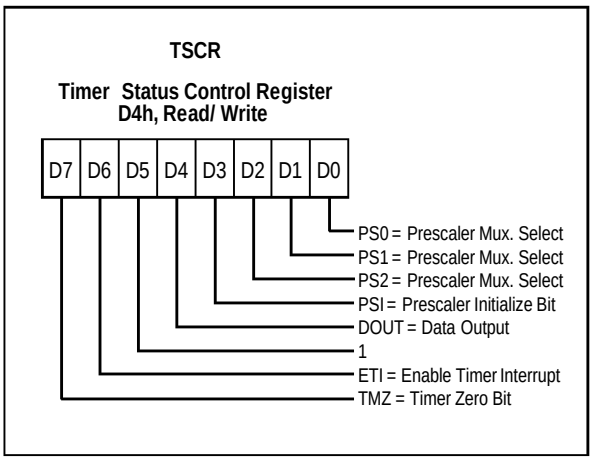
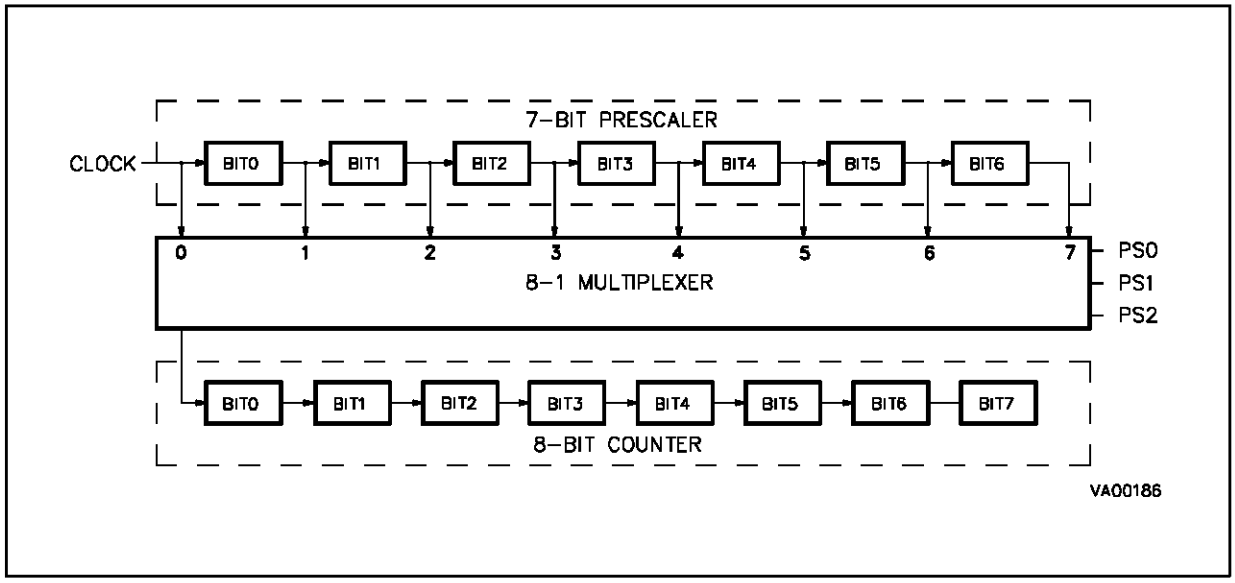


Figure 28. Timer Working Principle



TIMERS (Continued)

TMZ. Low-to-high transition indicates that the timer count register has decremented to zero. This bit must be cleared by user software before starting with a new count.

ETI. This bit, when set, enables the timer interrupt request (vector #3). If ETI="0" the timer interrupt is disabled. If ETI="1" and TMZ="1" an interrupt request is generated.

TOUT. This bit must be set high.

DOUT. Data sent to the timer output when TMZ is set high.

PSI. Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

PS2, PS1, PS0. These bits select the division ratio of the prescaler register (see Table 8).

Table 8. Prescaler Division Factors

PS2	PS1	PS0	Divided by	PS2	PS1	PS0	Divided by
0	0	0	1	1	0	0	16
0	0	1	2	1	0	1	32
0	1	0	4	1	1	0	64
0	1	1	8	1	1	1	128

Figure 30. Timer Counter Register

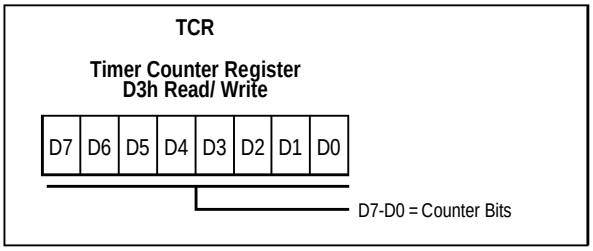
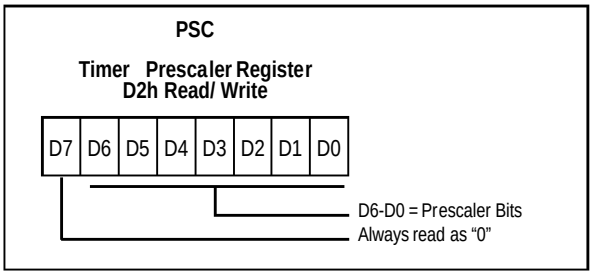


Figure 31. Prescaler Register



DIGITAL WATCHDOG

The ST6242 watchdog is a software activated watchdog.

Figure 32 shows the watchdog block diagram while Figure 34 shows its working principle.

The software activated digital watchdog consists of a down counter that can be used to provide a controlled recovery from a software upset or as a simple 7-bit timer for general purpose counting. The watchdog uses one data space register (DWDR location D8h). The watchdog register is set to FEh after reset and the watchdog function is disabled. The watchdog time can be programmed using the 6 Most Significant Bits in the Watchdog register. The check time can be set differently for different routines within the general program.

After a reset the software Watchdog is in the off-state. The watchdog should be activated inside the Reset restart routine by writing a “1” in watchdog timer register bit 0. Bit one of this register must be set to one before programming bit zero as otherwise a reset will be immediately generated when bit 0 is set. This allows the user to generate a reset by software (bit 0 = “1”, bit 1 = “0”). Once bit 0 is set, it can not be cleared by software without generating a Reset. The delay time is defined by programming bits 2-7 of the watchdog register. Bit 7 is the Least Significant Bit while bit 2 is the MSB. This gives the possibility to generate a reset in a time between 3072 to 196608 clock cycles in 64 possible steps: (With a clock frequency of 8MHz this means from 384µs to 24.576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones. If the watchdog is active the STOP instruction is deactivated and a WAIT instruction is automatically executed instead of a STOP. If bit 0 of the watchdog register is never set to one then bits 1-7 of the register can be used as a simple 7-bit counter which is decrement every 3072 clock cycles.

Figure 33. Watchdog Register

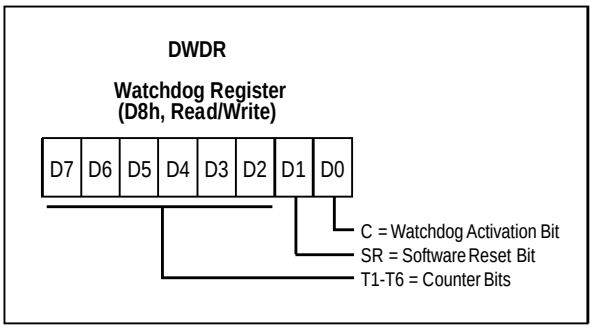
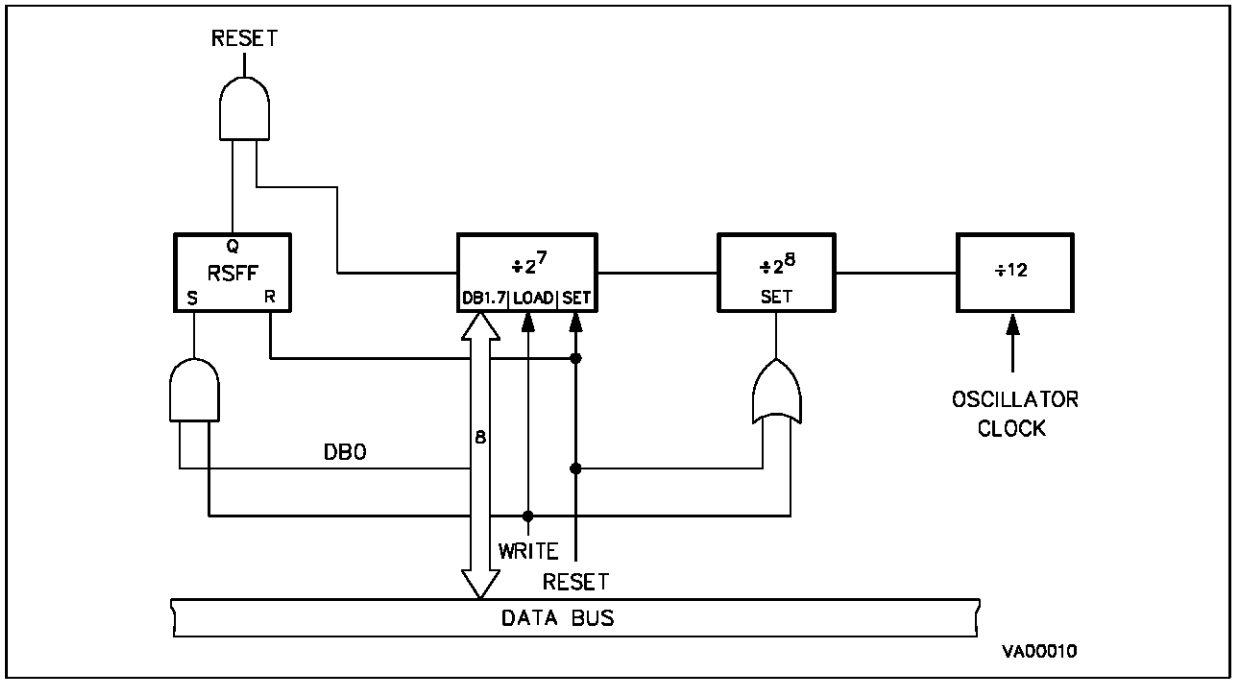


Figure 32. Digital Watchdog Block Diagram



DIGITAL WATCHDOG (Continued)

Watchdog Register

C. This is the watchdog activation bit, that, if set to one, will activate the watchdog function. When cleared to zero it allows the use of the counter as a 7-bit timer. This bit is cleared on reset.

SR. This bit is set to one during the reset and will generate a software reset if cleared to zero. When C = "0" (watchdog disabled software option) it is the MSB of the 7-bit timer.

T1-T6. These are the watchdog counter bits. It should be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter. These bits are in the opposite order to normal.

Application note:

If the Watchdog is not used during power-on reset external noise may cause the undesired activation of the Watchdog with a generation of an unexpected reset. To avoid this risk, two additional instructions, that check the state of the watchdog and eventually reset the chip are needed within the first 27 instructions, after the reset. These instructions are:

```
jrx 0, WD, #+3
ldi WD, 0FDH
```

These instructions should be executed at the very beginning of the customer program.

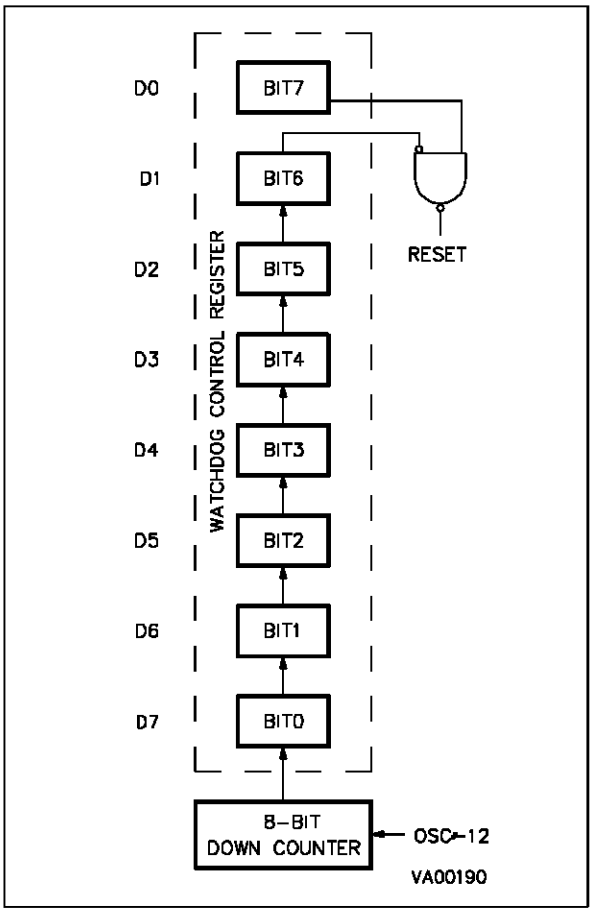
If the Watchdog is used, during power-on reset the Watchdog register may be set to a low value, that could give a reset after 28 instructions earliest. To avoid undesired resets, the Watchdog must be set to the desired value within the first 27 instructions, the best is to put at the very beginning.

Alternatively the normal legal state can be checked with the following short routine:

```
ldi a, 0FEH
and a, WD
cpi a, 0FEH
jrz #+3
ldi WD, 0FDH
```

This sequence is recommended for security applications, where possible stack confusion error loops must be avoided and the Watchdog must only be refreshed after extensive checks.

Figure 34. Watchdog Working Principle



8-BIT A/D C ONVERTER

The A/D converter of ST6242 is an 8-bit analog to digital converter with 6 analog inputs (as alternate functions of I/O lines PA4-PA7, PB2-PB3) offering 8-bit resolution with total accuracy ±2 LSB and a typical conversion time of 70µs (clock frequency of 8MHz).

The A/D peripheral converts the input voltage by a process of successive approximations using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, the A/D converter accuracy is decreased.

The selection of the pin signal that has to be converted is done by configuring the related I/O line as analog input through the I/O ports option and data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as analog input at a time. The user must avoid the situation in which more than one I/O pin is selected to be analog input to avoid malfunction of the ST62xx.

The ADC uses two registers in the data space: the ADC data conversion register which stores the conversion result and the ADC control register used to program the ADC functions.

A conversion is started by writing a "1" to the Start bit (STA) in the ADC control register. This automatically clears (resets to "0") the End Of Conversion Bit (EOC). When a conversion has been finished this EOC bit is automatically set to "1" in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

The STA bit is continually being scanned so that if the user sets it to "1" while a previous conversion is in progress then a new conversion is started before the previous one has been completed.The start bit (STA) is a write only bit, any attempt to read it will show a logical "0".

The A/D converter has a maskable interrupt associated to the end of conversion. This interrupt is associated to the interrupt vector #4 and occurs when the EOC bit is set, i.e. when a conversion is completed. The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. That is achieved when the PDS bit in the ADC control register is cleared to "0". If PDS="1", the A/D is supplied and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow the stabilization of the A/D converter.*This action is needed also before entering the STOP instruction as the A/D comparator is not automatically disabled by the STOP mode*

During reset any conversion in progress is stopped, the control register is reset to all zeros and the A/D interrupt is masked (EAI=0).

Figure 35. A/D Converter Block Diagram

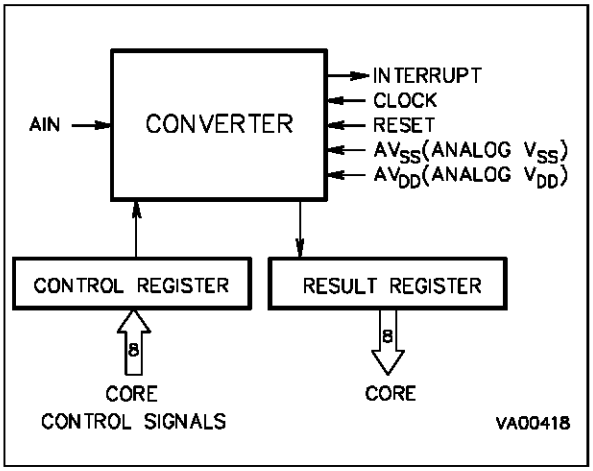


Figure 36. A/D Converter Control Register

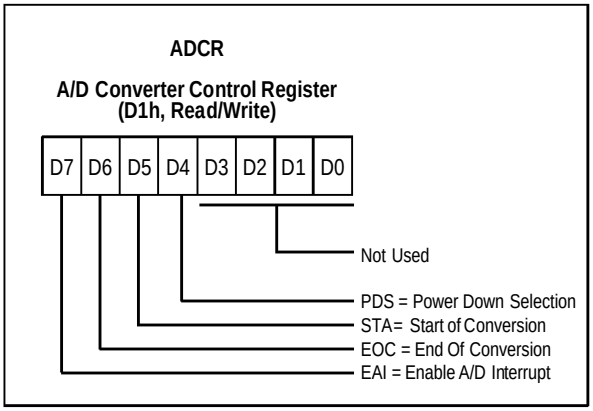
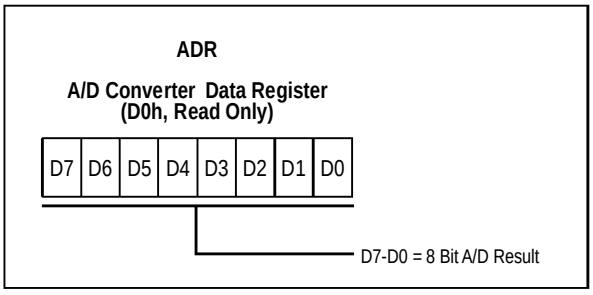


Figure 37. A/D Converter Data Register



8-BIT A/D CONVERTER(Continued)

A/D Converter Control Register

EAI. If this bit is set to one the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

EOC. *Read Only*;This read only bit indicates when a conversion has been completed. This bit is automatically reset to zero when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to one.

STA. *Write Only*; Writing a "1" in this bit will start a conversion on the selected channel and automatically reset to zero the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

PDS. This bit activates the A/D converter if set to 1. Writing a zero into this bit will put the ADC in power down mode (idle mode).

D3-D0. Not used

A/D Converter Data Register

D7-D0. *Read Only*;These are the conversion result bits; the register is read only and stores the result of the last conversion. The contents of this register are valid only when EOC bit in the ADCR register is set to one (end-of-conversion).

Notes:

The ST62xx A/D converter does not feature a sample and hold. The analog voltage to be measured should therefore be stable during the conversion time. Variation should not exceed ±1/2 LSB for the best accuracy in measurement.

Since the ADC is on the same chip as the microprocessor the user should not switch heavily loaded output signals during conversion if high precision is needed. This is because such switching will affect the supply voltages which are used for comparisons.

A low pass filter can be used at the analog input pins to reduce input voltage variation during the conversion. For true 8 bit conversions the impedance of the analog voltage sources should be less than 30kΩ while the impedance of the reference voltage should not exceed 2kΩ.

The accuracy of the conversion depends on the quality of the power supply voltages (V_{DD} and V_{SS}). The user must specially take care of applying regulated reference voltage on the V_{DD} and V_{SS} pins (the variation of the power supply voltage must be inferior to 5V/ms).

The converter can resolve the input voltage with an resolution of:

$$\frac{V_{DD} - V_{SS}}{256}$$

So if operating with a supply voltage of 5V the resolution is about 20mV.

The Input voltage (Ain) which has to be converted must be constant for 1µs before conversion and remain constant during the conversion.

The resolution of the conversion can be improved if the power supply voltage (V_{DD}) of the microcontroller becomes lower.

In order to optimize the resolution of the conversion, the user can configure the microcontroller in the WAIT mode because this mode allows the minimization of the noise disturbances and the variations of the power supply voltages due to the switching of the outputs. Nevertheless, it must be take care of executing the WAIT instruction as soon as possible after the beginning of the conversion because the execution of the WAIT instruction may provide a small variation of the V_{DD} voltage (the negative effect of this variation is minimized at the beginning of the conversion because the latter is less sensitive than the end of the conversion when the less significant bits are determined).

The best configuration from a accuracy point of view is the WAIT mode with the Timer and LCD driver stopped. Indeed, only the ADC peripheral and the oscillator are still working. The MCU has to be wake-up from the WAIT mode by the interrupt of the ADC peripheral at the end of the conversion. It must be noticed that the wake-up of the microcontroller could be done also with the interrupt of the TIMER, but in this case, the Timer is working and some noise could disturb the converter in terms of accuracy.

SERIAL PERIPHERAL INTERFACE (SPI)

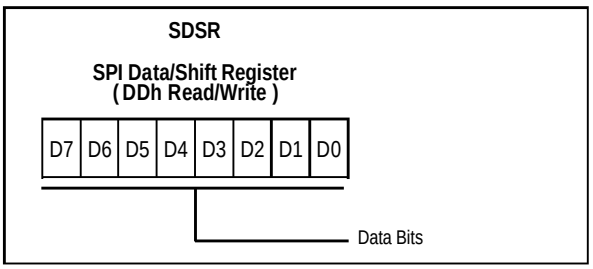
The ST6242 SPI is an optimized serial synchronous interface that supports a wide range of industry standard SPI specifications. The ST6242 SPI is controlled by small and simple user software to perform serial data exchange. The serial shift clock can be implemented either by software (using the bit-set and bit-reset instructions), with the on-chip Timer 1 by externally connecting the SPI clock pin to the timer pin or by directly applying an external clock to the SPI.

The peripheral is composed by an 8-bit Data/shift Register (address DDh) and a 4-bit binary counter. The SCL, Sin and Sout SPI data and clock signals are connected to the PB5, PB6 and PB7 I/O lines. With the 3 I/O pins, the SPI can operate in the following operating modes: Software SPI, S-BUS, I^2C-bus and as a standard serial I/O (clock, data, enable). An interrupt request can be generated after eight clock pulses. Figure 39 shows the SPI block diagram.

The PB5/SCL line clocks, on the falling edge, the shift register and the counter. To allow SPI operation the PB5/SCL must be programmed as input, an external clock supplied to this pin will drive the SPI peripheral (slave mode).

If PB5/SCL is programmed as output, a clock signal can be generated by software, setting and resetting the port line by software (master mode).

Figure 38. SPI Data/Shift Register



The SCL clock signal is the shift clock for the SPI data/shift register. The PB6/Sin pin is the serial shift input and PB7/Sout is the serial shift output. These two lines can be tied together to implement two wires protocols (I^2C-bus, etc). When data is serialized, the MSB is the first bit. PB6/Sin has to be programmed as input. For serial output operation PB7/Sout has to be programmed as open-drain output.

After 8 clock pulses (D7..D0) the output $\overline{Q4}$ of the 4-bit binary counter becomes low, disabling the clock from the counter and the data/shift register. Q4 enables the clock to generate an interrupt on the 8th clock falling edge as long as no reset of the counter (processor write into the 8-bit data/shift register) takes place. After a processor reset the interrupt is disabled. The interrupt is active when writing data in the shift register (DDh) and desactivated when writing any data in the register SPI Interrupt Disable (C2h).

The generation of an interrupt to the Core provides information that new data is available (input mode) or that transmission is completed (output mode), allowing the Core to generate an acknowledge on the 9th clock pulse (I^2C-bus).

Since the SPI interrupt is connected to interrupt #1, the falling edge interrupt option should be selected by clearing to zero bit 6 of the Interrupt Option Register (IOR, C8h).

After power on reset, or after writing the data/shift register, the counter is reset to zero and the clock is enabled. In this condition the data shift register is ready for reception. No start condition has to be detected. Through the user software the Core may pull down the Sin line (Acknowledge) and slow down the SCL, as long as it is needed to carry out data from the shift register.

SERIAL PERIPHERAL INTERFACE (Continued)

I²C-bus Master-Slave, Receiver-Transmitter

When pins Sin and Sout are externally connected together it is possible to use the SPI as a receiver as well as a transmitter. With a simple software routine (by using bit-set and bit-reset on I/O line) a clock can be generated allowing I²C-bus to work in master mode.

When implementing an I²C-bus protocol, the start condition can be detected by setting the processor into a "wait for start" condition by simply enabling the interrupt of the PA6/Sin I/O port. This frees the processor from polling the Sin and SCL lines. After the transmission/reception the processor has to poll for the STOP condition.

In slave mode the user software can slow down the SCL clock frequency by simply putting the SCL I/O line in output open-drain mode and writing a zero into the corresponding data register bit.

As it is possible to directly read the Sin pin directly through the port register, the software can detect a difference between internal data and external data (master mode). Similar condition can be applied to the clock.

The typical speed of transmission in I²C master or slave mode is in the range of 10kHz.

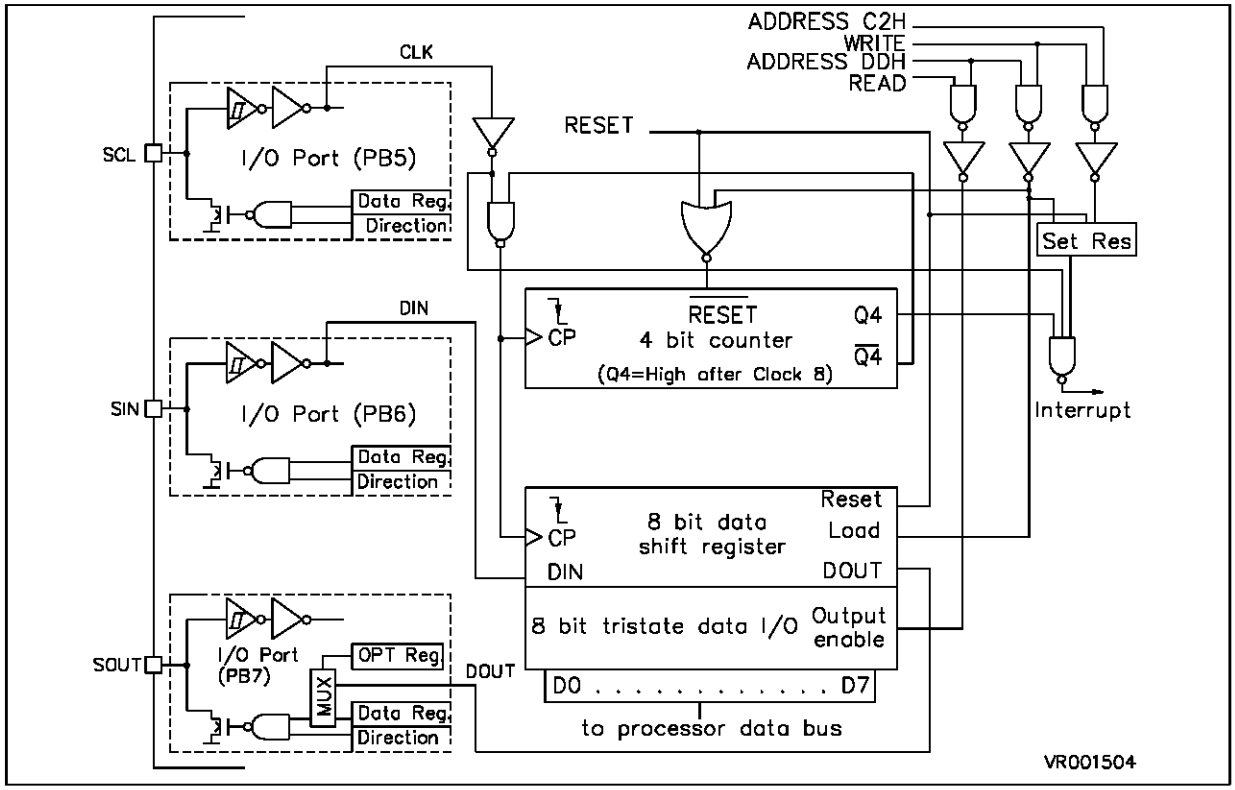
Three (Four) Wire Serial Bus

It is possible to use a single general purpose I/O pin (with the corresponding interrupt enabled) as a "chip enable" pin. SCL acts as active or passive clock pin, Sin as data in and Sout as data out (four wire bus). Sin and Sout can be connected together externally to implement three wire bus.

Note:

When the SPI is not used, the three I/O lines (Sin, SCL, Sout) can be used as normal I/O, with the following limitation: bit Sout cannot be used in open drain mode as this enables the shift register output to the port.
It is recommended, in order to avoid spurious interrupts from the SPI, to disable the SPI interrupt (the default state after reset) i.e. no write must be made to the 8-bit shift register (DDh). An explicit interrupt disable may be made in software by a dummy write to address C2h.

Figure 39. SPI Block Diagram



LCD CONTROLLER-DRIVER

The ST6242 LCD driver consists of a LCD control logic, a programmable prescaler, a 24 bytes wide dedicated LCD RAM, 40 segment and 4 common outputs. This allows a direct driving of up to 160 LCD segments.

The LCD driver is managed by the LCD Mode/Control register located at data RAM address DCh. Different display modes (1/1 duty, 1/2 duty, 1/3 duty and 1/4 duty) are available to cover a wide range of application requirements. The multiplexing display modes are software selectable by programming bits 6 and 7 of the LCD control register. Bits 0-5 are used to select the LCD drive and frame frequency (in relation to the system clock) and to switch off all segments.

According to the data in the LCD RAM, the segment and the common drivers generate the segment and common signals which can directly drive an LCD panel.

The LCD control logic reads automatically the data from the LCD RAM independently and without interruption of the processor. The part of the LCD RAM that is not used for displaying can be used as normal data memory.

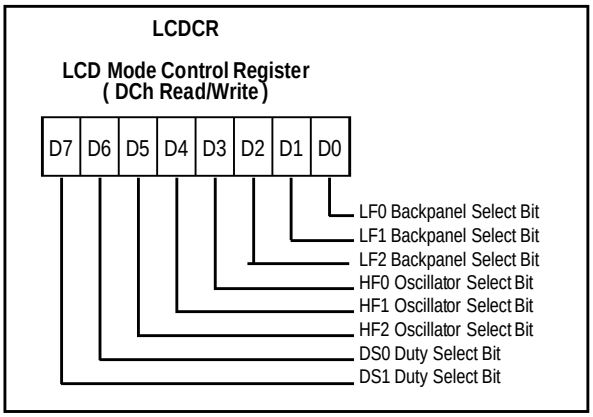
The scale factor of the clock prescaler can be fixed by software, therefore different frame frequencies can be defined.

The ST6242 oscillator should operate with a 1.0486, 2.0972, 4.1943, 8.3886MHz frequency quartz crystal. This allows the associated division rates to achieve an internal reference frequency of 32.768kHz. The different division rates can be achieved by programming bits 3, 4, 5 in the LCD control register (see Table 10). It is not recommended to select an internal frequency lower than 32.768kHz as the clock supervisor circuit may switch off the LCD peripheral if the lower frequency is detected.

When the display is turned off, all segment and common outputs are switched to ground, causing all the segments to be switched off regardless of the contents of the LCD RAM.

To avoid incomplete frames of the LCD, the mode control bits do not immediately influence the LCD controller when the LCD control register is written. They are stored in a temporary register and change the LCD function only at the end of the frame. Different LCD frame frequencies for each display mode are selected by bits in the LCD control register (see Table 11).

Figure 40. LCD Mode Control Register



DS0, DS1. Duty cycle select bits. These bits select the number of common backplanes used by the LCD control. This allows different multiplexing conditions.

HF0, HF1, HF2. These bits allow the LCD controller to be supplied with the correct frequency when different high main oscillator frequencies are selected as system clock. Table 10 shows the set-up for different clock crystals.

LF0, LF1, LF2. These bits control the LCD base operational frequency of the LCD common lines. Table 11 shows the set-up to select the different frequencies while Table 12 shows the corresponding frame values with the different multiplexing conditions.

Table 9. Duty Cycle Selection

DS1	DS0	Display Mode	Active Blackplanes	Max. Number of Segments Driven
0	0	1/4 duty	COM1, 2, 3, 4	180
0	1	1/1 duty	COM1	45
1	0	1/2 duty	COM1, 2	90
1	1	1/3 duty	COM1, 2, 3	135

LCD CONTROLLER-DRIVER (Continued)

Table 10. High Frequency Select Bits

HF2	HF1	HF0	Function	f_{osc}
0	0	0	Display off	
0	0	1	for stand-by Oscillator	32.768kHz
0	1	0	NOT TO BE USED	
0	1	1	÷ 32 for main oscillator	1.048MHz
1	0	0	÷ 64 for main oscillator	2.097MHz
1	0	1	÷ 128 for main oscillator	4.194MHz
1	1	0	÷ 256 for main oscillator	8.388MHz
1	1	1	NOT TO BE USED	

Notes :

1. The usage f_{osc} values different from those defined in this table cause the LCD to operate at a reference frequency different from 32.768kHz.
2. It is not recommended to select an internal frequency lower than 32.768kHz as the clock supervisor circuit may switch off the LCD peripheral if lower frequency is detected.

Table 11. LCD Frequency Select Bits

LF2	LF1	LF0	f_{LCD} (Hz)
0	0	0	64
0	0	1	85
0	1	0	128
0	1	1	171
1	0	0	256
1	0	1	341
1	1	0	512
1	1	1	Not to be Used

Table 12. Available Frame Frequencies for LCD

f_{LCD} (Hz)	Frame Frequency f_F (Hz)			
	1/1 duty	1/2 duty	1/3 duty	1/4 duty
512	512	256	171	128
341	341	171	114	85
256	256	128	85	64
171	171	85	57	43
128	128	64	43	32
85	85	43	28	21
64	64	32	21	16

According to the selected LCD drive frequency f_{LCD} the frame frequencies come out as shown in Table 12.

The Figure 47 illustrates the waveforms of the different duty signals.

The value of the VLCD voltage can be chosen independently from V_{DD} according to the display requirements. The intermediate VLCD levels 2/3 VLCD, 1/3 VLCD and 1/2 VLCD are generated by an internal resistor network as shown in Figures 45 and 46. The half VLCD level for 1/2 duty cycle is obtained by the external connection of VLCD1/3 and VLCD2/3 pins. All intermediate VLCD levels are connected to pins to enable external capacitive buffering or resistive shunting.

LCD CONTROLLER-DRIVER (Continued)

The internal resistive divider network is realized with two parallel dividers. One has high resistivity, the other one low resistivity. The high resistive divider (R_H) is permanently switched on during the LCD operation. The low resistive divider (R_L) is only switched on for a short period of time when the levels of common lines and segment lines are changed. This method combines low source impedance for fast switching of the LCD pixels with high source impedance for low power consumption. Figure 41 shows the typical current into V_{LCD} pin in dependency of the display voltage V_{LCD}.

When the display is switched off (by program or reset) the internal resistor network is also switched off to achieve minimum power consumption. The low resistivity divider is active at each edge of f_{LCD} during 8 clock cycles of F_{32kHz}.

The internal resistor network is implemented with resistive transistor elements to achieve high precision. For display voltages $V_{LCD} < 4.5V$ the resistivity of the divider may be too high for some applications (especially using 1/3 or 1/4 duty display mode). In that case an external resistive divider must be used to achieve the desired resistivity.

Figure 41. Typical Current Consumption on VLCD Pin (25°C, no load, fLCD=512 Hz, mux=1/3-1/4)

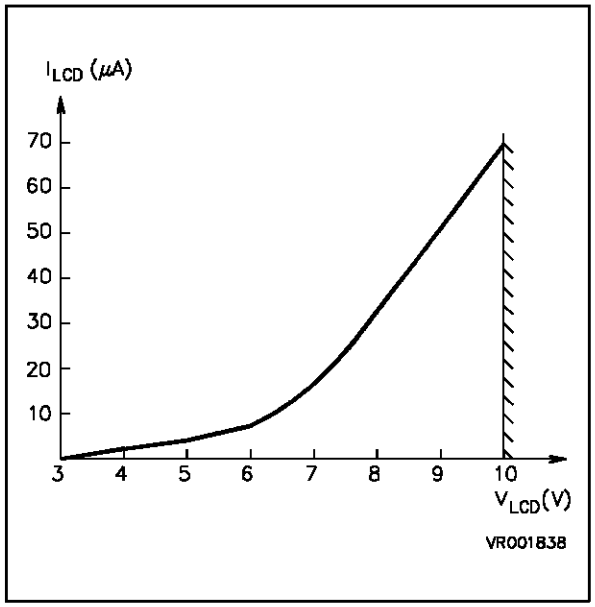
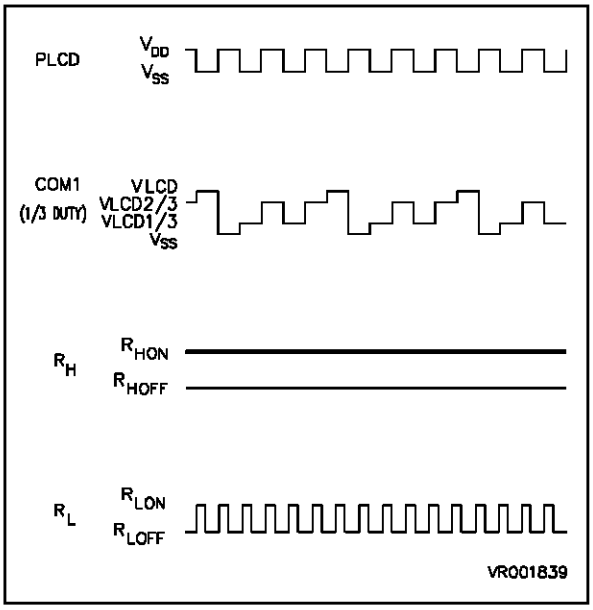


Figure 42. Typical Chronogram of Activation of the V_{LCD} Divider Network



LCD CONTROLLER-DRIVER (Continued)

Typical External resistances values are in the range of 100 kΩ to 150 kΩ. External capacitances in the range of 10 to 47 nF can be added to V_{LCD} 2/3 and V_{LCD} 1/3 pins and to V_{LCD} if the V_{LCD} connection is highly impedant.

When the program is switched off (by program or reset) the internal resistor network is also switched off to achieve minimum power consumption.

Figure 43. Typical Network to connect to V_{LCD} pins if $V_{LCD} \leq 4.5V$

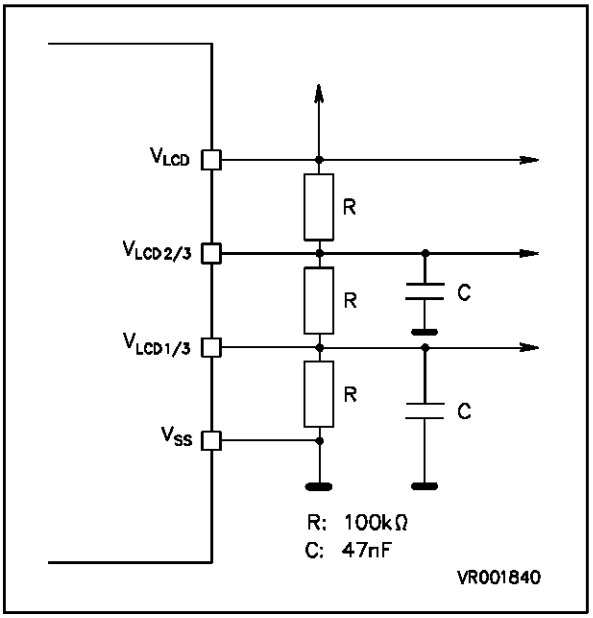


Figure 44. Generation of the 32kHz clock

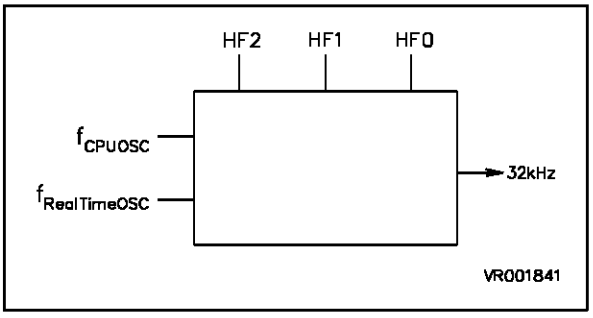


Figure 45. Bias Configuration for 1/2 Duty

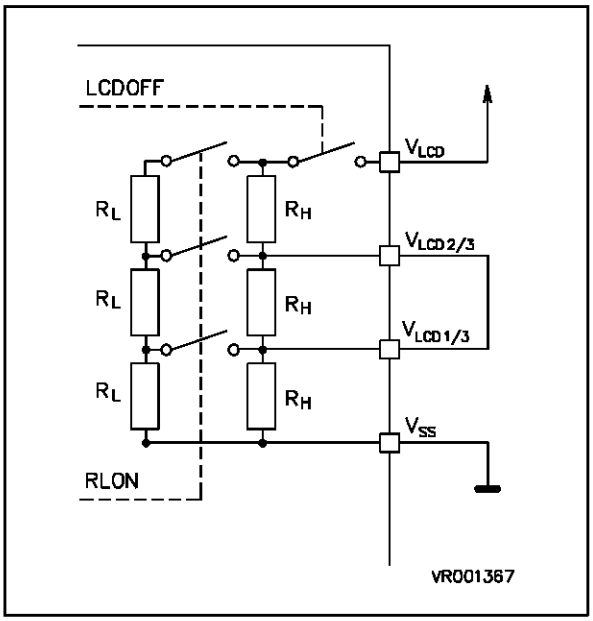
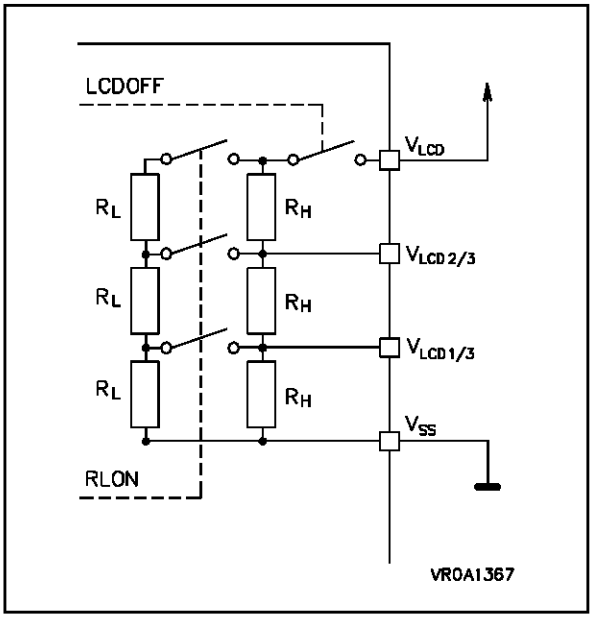


Figure 46. Bias Configuration for 1/1, 1/3 and 1/4 Duty Operation of LCD



LCD CONTROLLER-DRIVER (Continued)

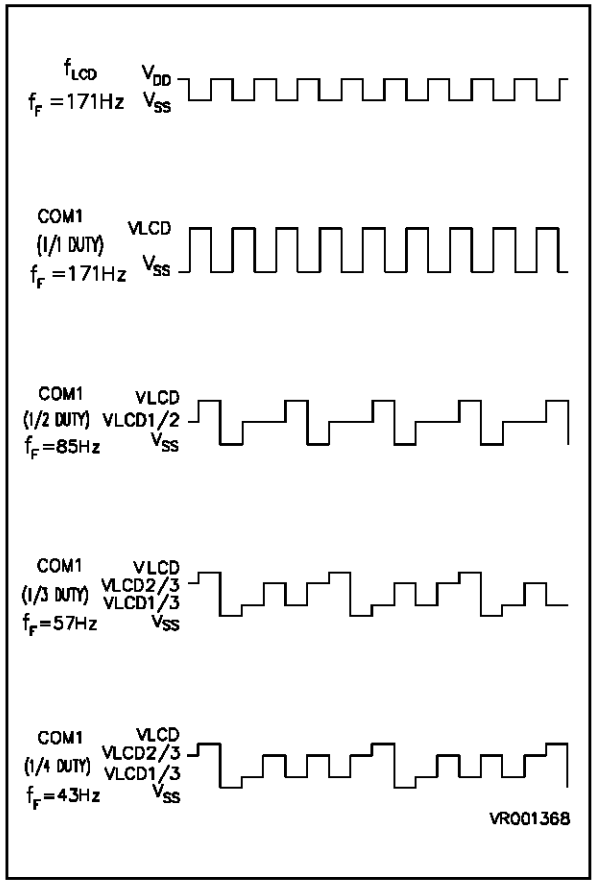
Address Mapping of the Display Segments.

The LCD RAM is located in the ST6242 data space from addresses E0h to F7h. The LCD forms a matrix of 40 segment lines (rows) and up to 4 common lines (columns). Each bit of the LCD RAM is mapped to one element of the LCD matrix, as described in Figure 48. If a bit is set, the corresponding LCD segment is switched on; if it is reset, the segment is switched off. The segments outputs S1to S8 are not connected to any pin.

When multiplex rates lower than 1/4 are selected, the unused LCD RAM is free for general use. In the 1/2 duty mode, for instance, half of the LCD RAM is available for storing general purpose data. The address range from F8h to FEh can be used as general purpose data RAM, but not for displaying data (it is reserved for future LCD expansion).

After a reset, the LCD RAM is not initializated and contains arbitrary information. As the LCD control register is reset, the LCD is completely switched off.

Figure 47. Common Signal Waveforms



LCD CONTROLLER-DRIVER (Continued)

Figure 48. Addressing Map of the LCD RAM

Data RAM Address	MSB							LSB	
E0 *									
E1	S16	S15	S14	S13	S12	S11	S10	S9	
E2	S24	S23	S22	S21	S20	S19	S18	S17	
E3	S32	S31	S30	S29	S28	S27	S26	S25	**COM1**
E4	S40	S39	S38	S37	S36	S35	S34	S33	
E5	S48	S47	S46	S45	S44	S43	S42	S41	
E6 *									
E7	S16	S15	S14	S13	S12	S11	S10	S9	
E8	S24	S23	S22	S21	S20	S19	S18	S17	
E9	S32	S31	S30	S29	S28	S27	S26	S25	**COM2**
EA	S40	S39	S38	S37	S36	S35	S34	S33	
EB	S48	S47	S46	S45	S44	S43	S42	S41	
EC *									
ED	S16	S15	S14	S13	S12	S11	S10	S9	
EE	S24	S23	S22	S21	S20	S19	S18	S17	
EF	S32	S31	S30	S29	S28	S27	S26	S25	**COM3**
F0	S40	S39	S38	S37	S36	S35	S34	S33	
F1	S48	S47	S46	S45	S44	S43	S42	S41	
F2 *									
F3	S16	S15	S14	S13	S12	S11	S10	S9	
F4	S24	S23	S22	S21	S20	S19	S18	S17	
F5	S32	S31	S30	S29	S28	S27	S26	S25	**COM4**
F6	S40	S39	S38	S37	S36	S35	S34	S33	
F7	S48	S47	S46	S45	S44	S43	S42	S41	
F8 - FE *									

Note *. Row to be used as general purpose RAM (not for display data)

Notes:

In STOP mode no clock is available for the LCD controller from the main oscillator. If the 32kHz oscillator is activated the LCD can also operate in STOP mode. If the stand-by oscillator is not active, the LCD controller is switched off when STOP instruction is executed; this mode has to be selected to reach the lowest power consumption.

A missing LCD clock (no oscillator active, broken crystal, etc.) is detected by a clock supervisor circuit that switches all the segments and common lines to ground to avoid destructive DC levels at the LCD.

The LCD function change is only effective at the end of a frame. For this reason special care has to be taken when entering the STOP mode. After switching the LCD clock source from the main oscillator to the 32kHz stand-by oscillator it must be guaranteed that enough clock pulses are delivered to complete the current frame before entering the STOP mode. Otherwise the LCD function will not be changed and the LCD is switched off after entering the STOP mode.

The RAM addresses E6-EC-F2-F8/FE are not used for LCD display purposes. So they are available as 10 additional Data RAM registers.

SOFTWARE DESCRIPTION

The ST62xx software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum, in short to provide byte efficient programming capability. The ST62xx core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

Addressing Modes

The ST62xx core has nine addressing modes which are described in the following paragraphs. The ST62xx core uses three different address spaces: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte that is processed by the instruction is stored in the location that follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode . In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) that use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction that follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits that characterize the kind of the test, one bit that determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits that give the span of the branch (0h to Fh) that must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

SOFTWARE DESCRIPTION (Continued)

Instruction Set

The ST62xx core has a set of 40 basic instructions. When these instructions are combined with nine addressing modes, 244 usable opcodes can be obtained. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, bit manipulation. The following paragraphs describe the different types.

All the instructions within a given type are presented in individual tables.

Load & Store. These instructions use one,two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 13. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ . Affected
* . Not Affected

SOFTWARE DESCRIPTION (Continued)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 14. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	*
AND A, (Y)	Indirect	1	4	Δ	*
AND A, rr	Direct	2	4	Δ	*
ANDI A, #N	Immediate	2	4	Δ	*
CLR A	Short Direct	2	4	Δ	Δ
CLR rr	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

SOFTWARE DESCRIPTION (Continued)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 15. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ . Affected
* . Not Affected

Table 16. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
* . Not Affected

Table 17. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated and a WAIT is automatically executed instead of a STOP if the Watchdog function is selected.
Δ . Affected
* . Not Affected

Table 18. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc.12-bit address;
* . Not Affected

SOFTWARE DESCRIPTION (Continued)

Opcode Map Summary. The following table contains an opcode map for the instructions used on the MCU.

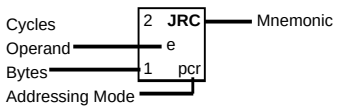
LOW / HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 pcr	4 LD a,(y) 1 ind	0 0000
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 pcr	4 LD a,rr 2 dir	1 0001
2 0010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a 1 inh	2 JRC e 1 pcr	4 CP a,(y) 1 ind	2 0010
3 0011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d.	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 pcr	4 CP a,rr 2 dir	3 0011
4 0100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 pcr	4 ADD a,(y) 1 ind	4 0100
5 0101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 pcr	4 ADD a,rr 2 dir	5 0101
6 0110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 pcr	4 INC (y) 1 ind	6 0110
7 0111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 pcr	4 INC rr 2 dir	7 0111
8 1000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 pcr	4 LD (y),a 1 ind	8 1000
9 1001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 pcr	4 LD rr,a 2 dir	9 1001
A 1010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RLC a 1 inh	2 JRC e 1 pcr	4 AND a,(y) 1 ind	A 1010
B 1011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 pcr	4 AND a,rr 2 dir	B 1011
C 1100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 pcr	4 SUB a,(y) 1 ind	C 1100
D 1101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 pcr	4 SUB a,rr 2 dir	D 1101
E 1110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 pcr	4 DEC (y) 1 ind	E 1110
F 1111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 pcr	4 DEC rr 2 dir	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from:

$Tj = T_A + PD \times RthJA$

Where : T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_{LCD}	Display Voltage	-0.3 to 11.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
I_O	Current Drain per Pin Excluding V_{DD} & V_{SS}	± 10	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PQFP64		70		°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version 6 Suffix Version	0 -40		70 85	°C
V_{DD}	Operating Supply Voltage		3		6	V
V_{LCD}	Display Voltage		3		10	V
V_{DD}	RAM Retention Voltage		2			V

RECOMMENDED OPERATING CONDITIONS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (1)(4)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01 0.01		8.388 2	MHz
I_{INJ+}	Pin Injection Current (positive) Digital Input (2) Analog Input (3)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input (2) Analog Input	V_{DD} = 4.5 to 5.5V			-5	mA

Notes :
1. An oscillator frequency above 1MHz is recommanded for reliable A/D results.
2. A current of ± 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device. For a positive current injected into one pin, a part of this current (~ 10%) can be expected to flow from the neighbouring pins. A current of -5mA can be forced on one input of the analog section at a time (or -2.5mA for all inputs at a time) without affecting the conversion.
3. If a total current of +1mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 1mA, all the conversion is resulting shifted of +1LSB. If a total positive current of +5mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 5mA, all the conversion is resulting shifted of +2LSB.
4. Operation below 0.01 MHz is possible but requires increased supply current.

EEPROM INFORMATION

The ST62xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	RESET, NMI, TIMER, WDON Pin			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	TIMER	$0.80V_{DD}$			V
		RESET, NMI, WDON Pin	$0.70V_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	RESET Pin V_{DD} =5V $V_{IN} = V_{DD}$ (1) $V_{IN} = V_{DD}$ (2) $V_{IN} = V_{SS}$ (5)			10 1 50	µA mA µA
V_{OL}	Low Level Output Voltage	TIMER, I_{OL} = 5.0mA			$0.2V_{DD}$	V
V_{OH}	High Level Output Voltage	TIMER, I_{OL} = -5.0mA	$0.65V_{DD}$			V
R_{PU}	Pull-up Resistor	V_{IN}=0V V_{DD}=5V WDON - NMI	40	100	200	kΩ
		RESET	200	300	500	kΩ

Notes on next page

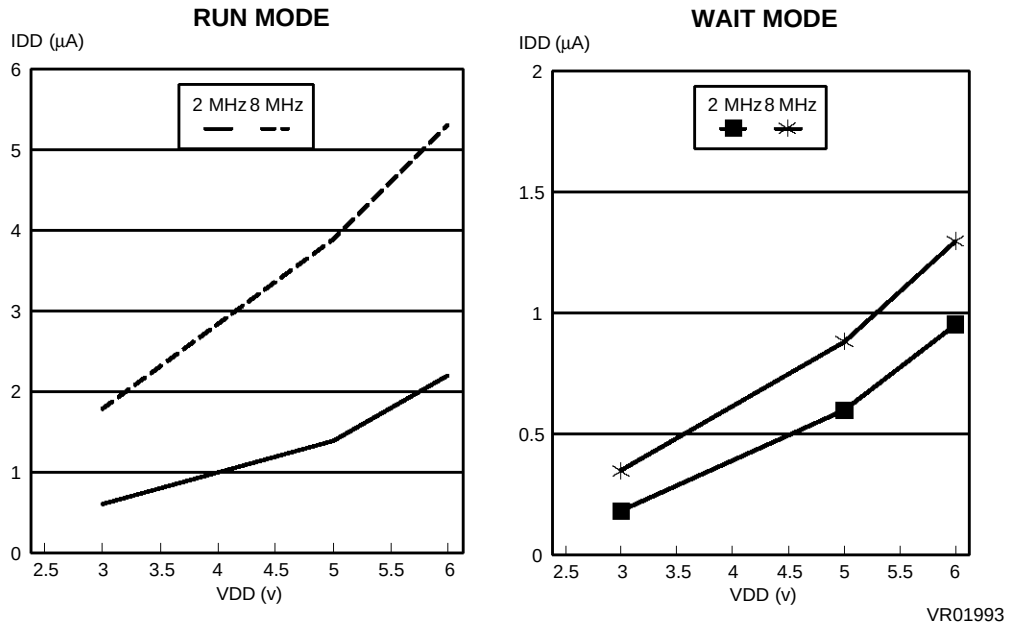
DC ELECTRICAL CHARACTERISTICS (Continued)
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{IL} I_{IH}	Input Leakage Current	TIMER $V_{IN} = V_{DD}$ or V_{SS}		0.1	1.0	µA
I_{IL} I_{IH}	Input Leakage Current	NMI VDD = 5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{IL} I_{IH}	Input Leakage Current	WDON VDD = 5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{DD}	Supply Current RUN Mode	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		4	7	mA
	Supply Current WAIT Mode (4)	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		1	2	mA
	Supply Current RESET Mode	f_{OSC} = 8MHz, $V_{RESET} = V_{SS}$		1	7	mA
	Supply Current STOP Mode (3)(4)	I_{LOAD} = 0mA V_{DD} = 5.0V		1	10	µA

Notes :
1. No Watchdog Reset activated.
2. Reset generated by Watchdog.
3. When the watchdog function is actvated the STOP instruction is deactivated. WAIT instruction is automatically executed.
4. All on-chip peripherals in OFF state
5. Pull-up resistor

CURRENT CONSUMPTION

IDD Current Versus Supply Voltage
Typical Values, Temp : -40/+85°C



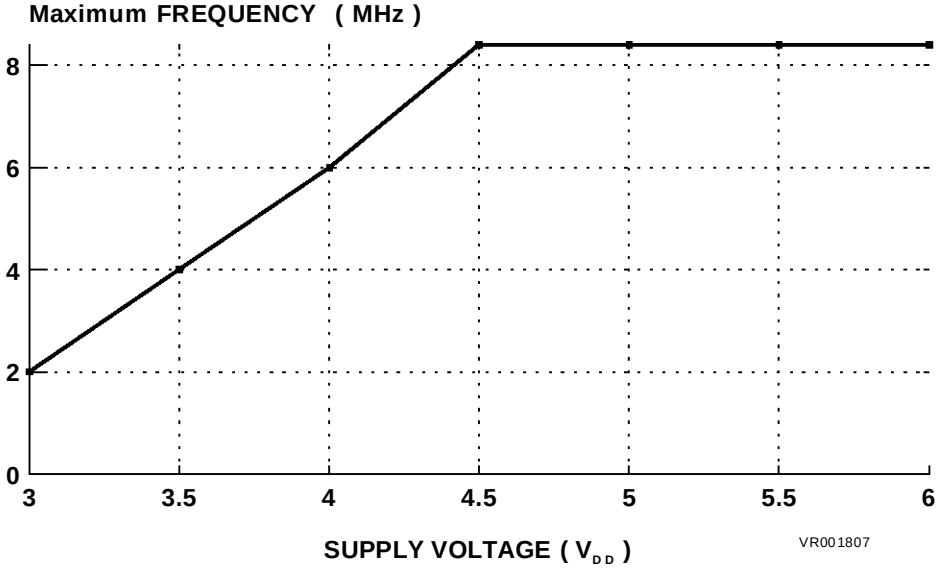
AC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (2)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01		8.388 2	MHz
t_{SU}	Oscillator Start-up Time	$C_{L1} = C_{L2}$ = 22pF - crystal		5	20	ms
t_{SR}	Supply Rise Time	10% to 90%	0.01		100	
t_{REC}	Supply Recovery Time (1)		100			
T_W	Minimum Pulse Width	NMI Pin V_{DD} = 5V	100			ns
		RESET Pin	100			ns
T_{WEE}	EEPROM Write Time	T_A = 25°C One Byte T_A = 85°C One Byte		5 15	10 25	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A L_{OT} Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention	T_A = 55°C	10			years
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:
1. Period for which V_{DD} has to be connected or at 0V to allow internal Reset function at next power-up.
2. Operation below 0.01 MHz is possible but requires increased supply current.

Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



I/O PORTS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	I/O Pins			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	I/O Pins	$0.7V_{DD}$			V
V_{OL}	Low Level Output Voltage	I/O Pins, I_O = 10µA (sink)			0.1	V
		I/O Pins, $I_{OL} = V_{DD}$x1mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 1.6mA V_{DD} = 3V			0.4	V
	Low Level Output Voltage, PB4-PB7 Only	I/O Pins, $I_{OL} = V_{DD}$x2mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 3.2mA V_{DD} = 3V			0.4	V
		I/O Pins, $I_{OL} = V_{DD}$x4mA V_{DD} = 4.5 to 6V			$0.26xV_{DD}$	V
		I/O Pins, I_{OL} = 6.4mA V_{DD} = 3V			0.8	V
V_{OH}	High Level Output Voltage	I/O Pins, I_O = -10µA (source)	V_{DD}-0.1			V
		I/O Pins, $I_{OL} = -V_{DD}$x1mA V_{DD} = 5.0V	$0.6xV_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	I/O Pins,(1)		0.1	1.0	µA
R_{PU}	Pull-up Resistor	I/O Pins V_{IN} = 0V, V_{DD} = 5.0V	40	100	200	KΩ

Note 1. Pull-up resistor off

SPI ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 V, T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
F_{CL}	Clock Frequency	applied on PB5/SCL			1	MHz
t_{SU}	Set-up Time	applied on PB6/Sin		50		ns
t_h	Hold Time	applied on PB6/Sin		100		ns

A/D CONVERTER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution (3)			8		Bit
A_{TOT}	Total Accuracy (3)	f_{OSC} > 1.2 MHz f_{OSC} > 32kHz			± 2 ±4	LSB
t_C(1)	Conversion Time	f_{OSC} = 8MHz		70		µs
V_{AN}	Conversion Range		V_{SS}		V_{DD}	V
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN}(2)	Analog Input Capacitance			2	5	pF
ASI	Analog Source Impedance				30	kΩ
SSI	Analog Reference Supply Impedence				2	kΩ

Notes:
1. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.
2. Excluding Pad Capacitance
3. Noise at V_{DD},V_{SS} ≤ 10mV

TIMER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{12}{f_{OSC}}$			second
f_{IN}	Input Frequency on TIMER Pin				$\frac{f_{OSC}}{8}$	MHz
t_W	Pulse Width at TIMER Pin	$V_{DD} \geq 3V$ $V_{DD} \geq 4.5V$	1 125			µs ns

LCD ELECTRICAL CHARACTERISTICS

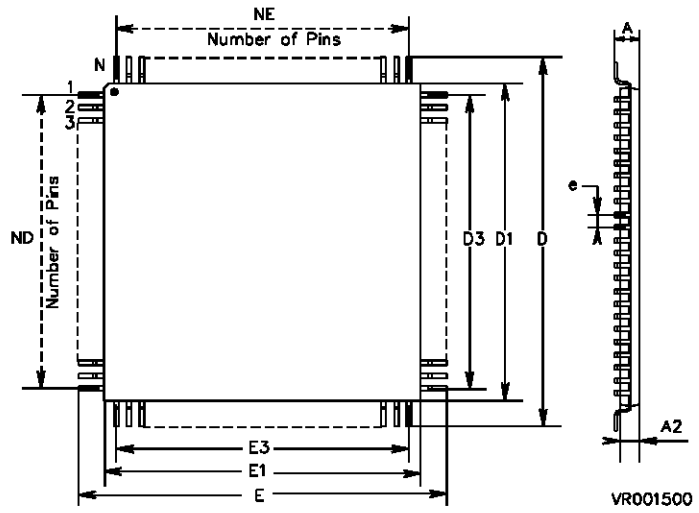
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{FR}	Frame Frequency	1/4 Duty f_{OSC} = 1, 2, 4, 8MHz	16		128	Hz
V_{OS}	DC Offset Voltage(1)	$V_{LCD} = V_{DD}$, no load			50	mV
V_{OH}	COM High Level, Output Voltage	I = 100µA, V_{LCD} = 5V	4.5V			V
V_{OL}	COM Low Level, Output Voltage	I = 100µA, V_{LCD} = 5V			0.5V	V
V_{OH}	SEG High Level, Output Voltage	I = 50µA, V_{LCD} = 5V	4.5V			V
V_{OL}	SEG Low Level, Output Voltage	I = 50µA, V_{LCD} = 5V			0.5V	V
V_{LCD}	Display Voltage	Note 2	3		10	V

Notes :
1. The DC offset voltage refers to all segment and common outputs. It is the difference between the measured voltage value and nominal value for every voltage level. Ri of voltage meter must be greater than or equal to 100MΩ.
2. An external resistances network is required when $V_{LCD} \leq 4.5V$.

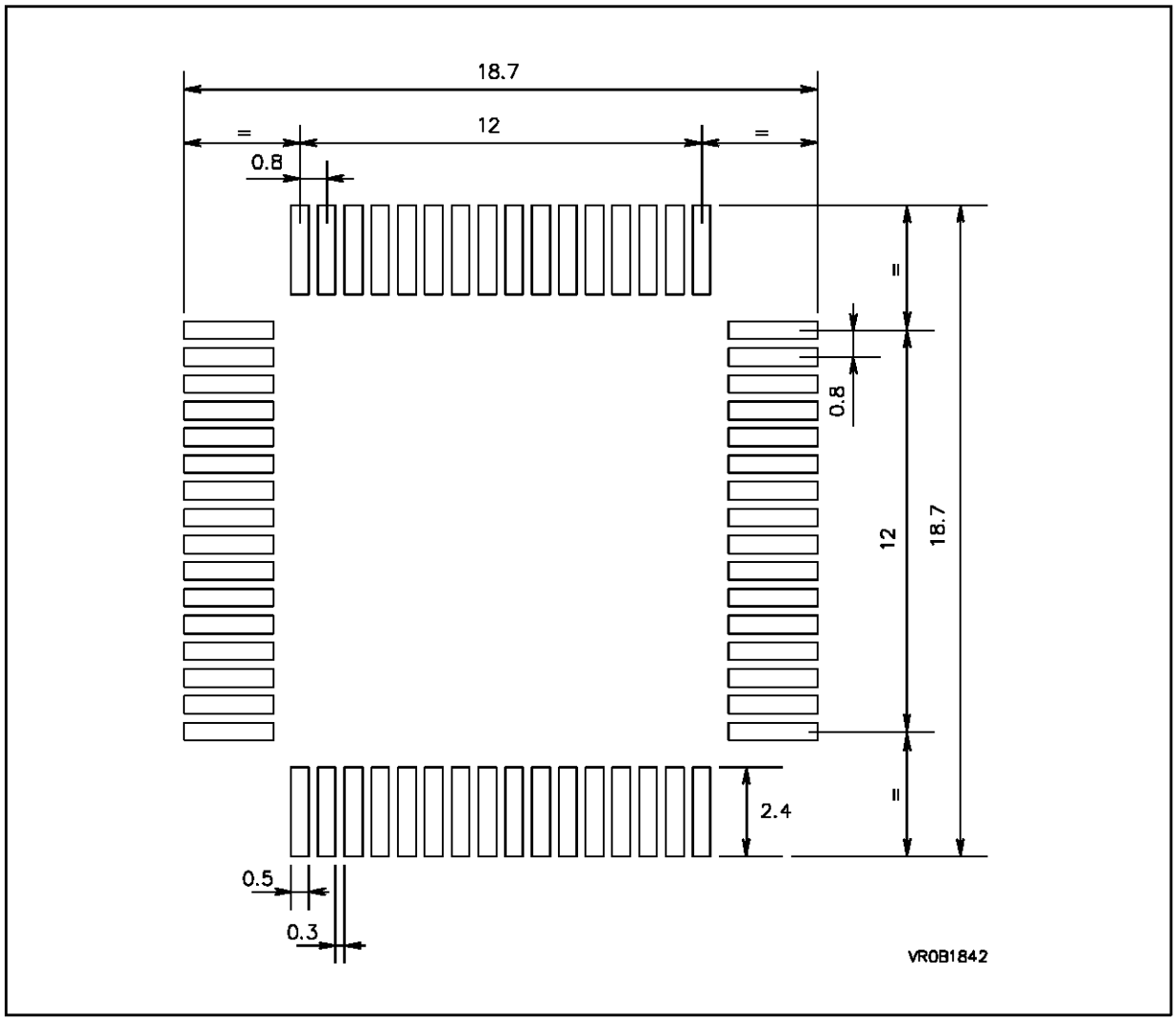
PACKAGE MECHANICAL DATA

Figure 49. ST6242 64 Pin Plastic Quad Flat Pack Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.40			0.134
A2	2.55	2.80	3.05	0.100	0.110	0.120
D	16.95	17.20	17.45	0.667	0.677	0.687
D1	13.90	14.00	14.10	0.547	0.551	0.555
D3		12.00			0.472	
E	16.95	17.20	17.45	0.667	0.677	0.687
E1	13.90	14.00	14.10	0.547	0.551	0.555
E3		12.00			0.472	
e		0.80			0.032	
Number of Pins						
ND	16					
NE	16					
N	64					

Recommended Solder Pad Footprint For QFP64 (in mm)



ORDERING INFORMATION

The following chapter deals with the procedure for transfer the Program/Data ROM codes to SGS-THOMSON.

Communication of the ROM Codes. To communicate the contents of Program/Data ROM memories to SGS-THOMSON, the customer has to send :

– one file in INTEL INTELLEC 8/MDS FORMAT (in an MS-DOS 5" diskette) for the PROGRAM Memory

– one file in INTEL INTELLEC 8/MDS FORMAT (in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)

– a filled Option List form as described in the OPTION LIST paragraph.

The program ROM should respect the ROM Memory Map as in Table 19.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

Table 19. ROM Memory Map

ROM Page	Device Address	Description
Page 0	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	Reserved User ROM
Page 3	0000h-000Fh 0010h-07FFh	Reserved User ROM

Note : EPROM addresses are related to the ROM file to be processed.

Customer EEPROM Initial Contents : Format

a. The content should be written into an INTEL INTELLEC format file.

b. In the case of 128 bytes of EEPROM, the starting address in 000h and the end in 7Fh.

c. Undefined or don't care bytes should have the content FFh.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are compared and a computer listing is generated from them. This listing refers exactly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

ORDERING INFORMATION TABLE

Sales Types	Temperature Range	Package
ST6242Q1/XX	0 to + 70°C	PQFP64
ST6242Q6/XX	-40 to + 85°C	PQFP64

Note : "XX" is the ROM code identifier allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

ST6242 MICROCONTROLLER OPTION LIST

Customer . . . . . .

Address . . . . . .

Contact . . . . . .

Phone No . . . . . .

Reference . . . . . .

SGS-THOMSON Microelectronics references

Device	[ ] ST6242	
Package	[ ] Plastic Quad Flat Package	
Temperature Range	[ ] 0°C to + 70°C	[ ] -40°C to + 85°C

Special Marking [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _"

Authorized characters are Letters, digits, '.', '–', '/' and spaces only.
For marking one line with 10 characters maximum is possible.

Comments :
- Number of LCD segments used :
- Number of LCD backplanes used :

Note :

Signature

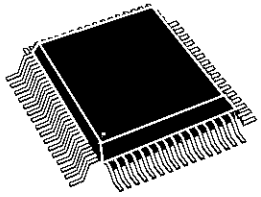
Date

SGS-THOMSON MICROELECTRONICS

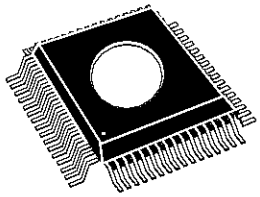
ST62E42
ST62T42

8-BIT OTP/EPROM HCMOS MCUs WITH LCD DRIVER, AND A/D CONVERTER

- 3 to 6V supply operating range
- 8.4MHz Maximum Clock Frequency
- Run, Wait & Stop Modes
- 5 different interrupt vectors
- Look-up table capability in EPROM
- User EPROM: 7948 bytes
  Data RAM: 128 bytes
  LCD RAM: 24 bytes
- PQFP64 and CQFP64-W packages
- 10 fully software programmable I/O as:
  - Input with/without pull-up resistor
  - Input with interrupt generation
  - Open-Drain or Push-pull outputs
  - Analog Inputs (6 pins)
- 4 I/O lines can sink up to 20mA for direct LED or TRIAC driving and have SPI alternate functions
- 8-bit counters and 7-bit programmable prescalers
- Software activated digital watchdog
- 8-bit A/D converter with up to 6 analog inputs
- 8-bit synchronous serial peripheral interface (SPI)
- LCD driver with 40 segment outputs, 4 backplane outputs and selectable duty cycle for up to 160 LCD segments direct driving
- One external not maskable interrupt
- 9 powerful addressing modes
- The accumulator, the X, Y, V & W registers, the port and peripherals data & control registers are addressed in the data space as RAM locations.
- The ST62E42 is the EPROM version, ST62T42 is the OTP version, fully compatible with ST6242 ROM version.



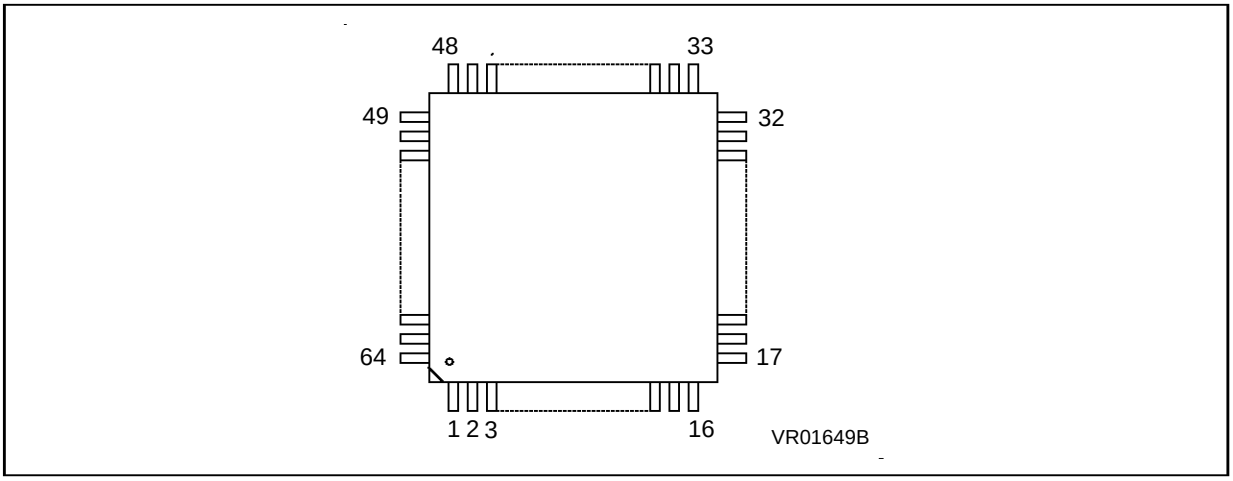
PQFP64



CQFP64-W

(Ordering Information at the end of the datasheet)

Figure 1.Pin Quad Flat Pack (QFP) Package Pinout



ST62E42/T42 Pin Description

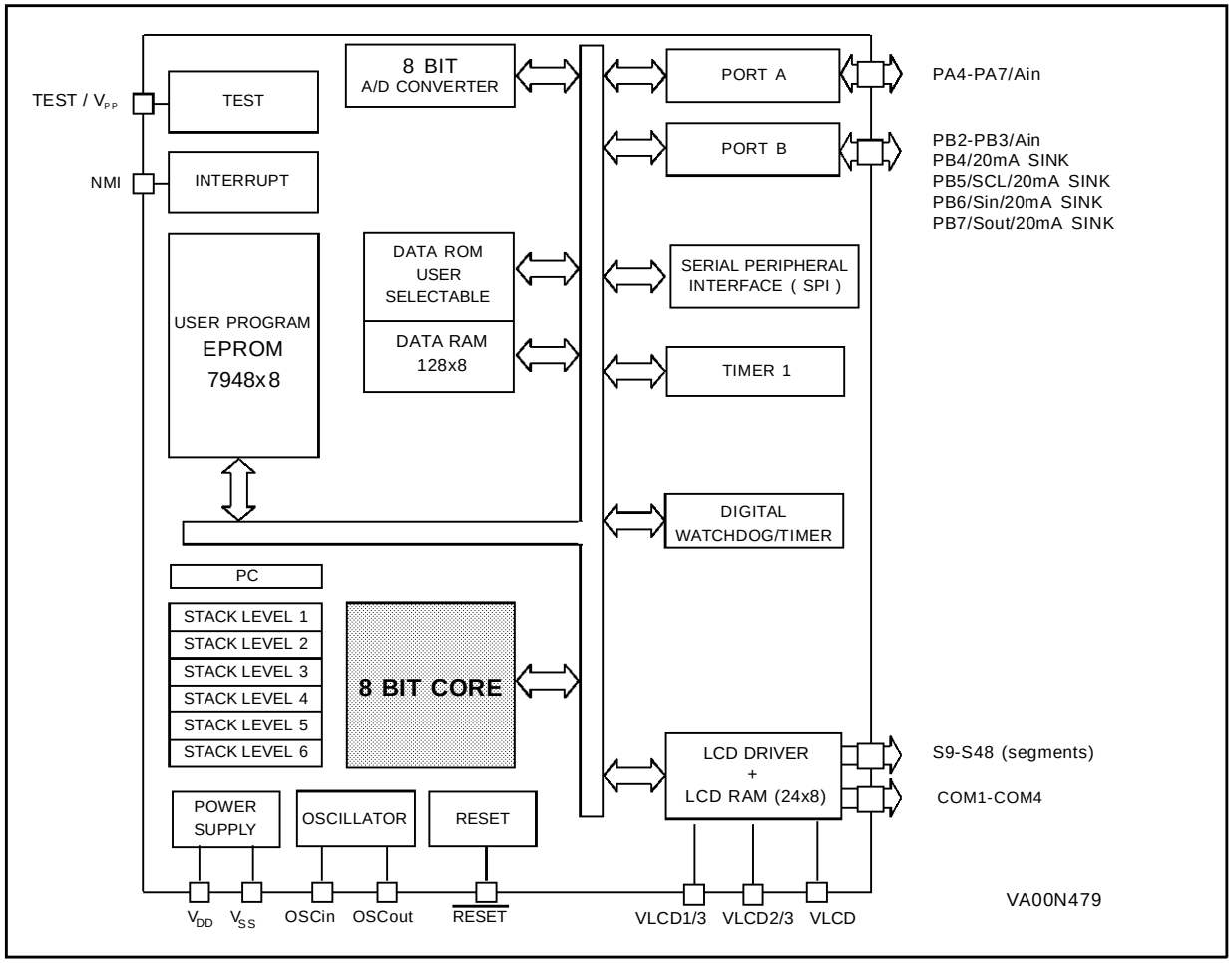
Pin number	Pin name	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
1	S45	17	V_{DD}	33	S13	49	S29
2	S46	18	V_{SS}	34	S14	50	S30
3	S47	19	$\overline{RESET}$	35	S15	51	S31
4	S48	29	OSCout	36	S16	52	S32
5	COM4	21	OSCin	37	S17	53	S33
6	COM3	22	NMI	38	S18	54	S34
7	COM2	23	PB7/Sout (1)	39	S19	55	S35
8	COM1	24	PB6/Sin (1)	40	S20	56	S36
9	VLCD1/3	25	PB5/SCL (1)	41	S21	57	S37
10	VLCD2/3	26	PB4 (1)	42	S22	58	S38
11	VLCD	27	PB3/Ain	43	S23	59	S39
12	PA7/Ain	28	PB2/Ain	44	S24	60	S40
13	PA6/Ain	29	S9	45	S25	61	S41
14	PA5/Ain	30	S10	46	S26	62	S42
15	PA4/Ain	31	S11	47	S27	63	S43
16	TEST/V_{PP}	32	S12	48	S28	64	S44

Note 1: 20mA SINK

GENERAL DESCRIPTION

The ST62E42,T42 microcontrollers are members of the 8-bit HCMOS ST62xx family, a series of devices oriented to low-medium complexity applications. They are the EPROM/OTP versions of the ST6242 ROM device and are suitable for product prototyping and low volume production. All ST62xx members are based on a building block approach: a common core is associated with a combination of on-chip peripherals (macrocells). The macrocells of the ST6242 family are: a high performance LCD controller/driver with 40 segment outputs and 4 backplanes able to drive up to 160 segments, a Timer peripheral including an 8-bit counter with a 7-bit software programmable prescaler, a digital watchdog timer (DWD), an 8-bit A/D Converter with up to 6 analog inputs and an 8-bit synchronous Serial Peripheral Interface (SPI). Thanks to these peripherals the ST6242 family is well suited for general purpose, automotive, security, appliance and industrial applications.

Figure 2. ST62E42 Block Diagram



Note: Ain = Analog Input

PIN DESCRIPTION

V_{DD} and **V_{SS}**. Power is supplied to the ST62E42 using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected with the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. OSCin is the input pin, OSCout is the output pin. An external clock signal can be applied to OSCin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to restart the microcontroller at the beginning of its program. The $\overline{\text{RESET}}$ pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TEST/V_{PP}. The TEST pin is used to place the MCU into special operating mode. TEST must be held at V_{SS} for normal operation (an internal pull-down resistor is present to select normal operating mode if TEST pin is not connected). If this pin is connected to a +12.5V level during the reset phase, the EPROM programming mode is entered.

NMI. The NMI pin provides the capability for asynchronous applying an external top priority interrupt to the ST62E42.This pin is falling edge sensitive. The NMI pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

PA4-PA7. These 4 lines are organized as one I/O port (A). Each line may be configured under software control as an input with or without pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output or as analog input for the A/D converter. Port A has a 5mA drive capability in output mode.

PB2-PB3,PB4-PB7. These 6 lines are organized as one I/O port (B). Each line may be configured under software control as an input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output. PB2-PB3 can be programmed as analog inputs for the A/D converter while PB4-PB7 can also sink 20mA for direct LED driving. PB5-PB7 can also be used as respectively Clock, Data in and Data out pins for the on-chip SPI to carry the synchronous serial I/O signals.

COM1-COM4. These four pins are the LCD peripheral common outputs. They are the outputs of the on-chip backplane voltage generator which is used for multiplexing the 40 LCD lines allowing up to 160 segments to be driven.

S9-S48. These pins are the 40 LCD peripheral driver outputs of ST62E42. Segments S1-S8 are not connected to any pin.

VLCD. Display voltage supply. It determines the high voltage level on COM1-COM4 and S9-S48 pins.

VLCD1/3, VLCD2/3. Display supply voltage inputs for determining the display voltage levels on COM1-COM4 and S9-S48 pins during multiplex operation.

ST62E42,T42 EPROM/OTP DESCRIPTION

The ST62E42 is the EPROM version of the ST6242 ROM product. It is intended for use during the development of an application, and for pre-production and small volume production. The ST62T42 OTP has the same characteristics. Both include EPROM memory instead of the ROM memory of the ST6242, and so the program and constants of the program can be easily modified by the user with the ST62E42 EPROM programming board from SGS-THOMSON.

From a user point of view (with the following exception) the ST62E42,T42 products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming of the ST62E42,T42 is described in the User Manual of the EPROM Programming board.

On the ST62E42, all the 8192 bytes of PROGRAM memory are available for the user, as all the EPROM memory can be erased by exposure to UV light. On the ST62T42 (OTP) device) a reserved area for test purposes exists, as for the ST6242 ROM device. In order to avoid any discrepancy between program functionality when using the EPROM, OTP and ROM it is recommended not to use these reserved areas, even when using the ST62E42.

Notes on programming:

In order to emulate exactly the ST6242 features with the ST62E42 and ST6242, some software precautions have to be taken:

1. Data RAM: The data entered in the Data RAM bank register (CBh) must be 08h.

2. I/O: To prevent floating input or uncontrolled I/O interrrupt on the EPROM/OTP devices, the port bits PA0-PA3, PB0, PB1 must be programmed as push-pull outputs.

3. Timer: The bit "TOUT" of the Timer status control register (D4h) must be set "1" (timer in output mode).

4. Data Memory Space: Write 40h at the address DFh of the Data Memory Space (desabled EE).

5. When programming for the EPROM/OTP parts, it is suggested that the conditional assembly technique is used for controlling the I/O ports in order to disable the appropriate code for the ROM device.

6. Do not access data space locations D5h, D6h, D7h, DAh, DBh.

Other than this exception, the ST62E42,T42 parts are fully compatible with the ROM ST6242 equivalent, this datasheet thus provides only information specific to the EPROM based devices.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST6242 ROM-BASED DEVICE FOR FURTHER DETAILS.

EPROM ERASING

The EPROM of the windowed package of the ST62E42 may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST62E42 EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST62E42 package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.
The recommended erasure procedure of the ST62E42 EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST62E42 should be placed within 2.5 cm (1 inch) of the lamp tubes during erasure.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from:

. . . . . Tj = . . T_A + PD x RthJA

Where : . T_A = . . Ambient Temperature.

. . . . . RthJA = Package thermal resistance
. . . . . (junction-to ambient).

. . . . . PD = . Pint + Pport.

. . . . . Pint = . I_{DD} x V_{DD} (chip internal power).

. . . . . Pport = Port power dissipation
. . . . . (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_{LCD}	Display Voltage	-0.3 to 11.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
I_O	Current Drain per Pin Excluding V_{DD} & V_{SS}	± 10	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PQFP64 CQFP64-W		70 70		°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Condition s	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version 6 Suffix Version	0 -40		70 85	°C
V_{DD}	Operating Supply Voltage		3		6	V
V_{LCD}	Display Voltage		3		10	V
V_{RM}	RAM Retention Voltage		2			V

Note: Refer to ordering information at end of the datasheet.

RECOMMENDED OPERATING CONDITIONS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (1)(4)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01 0.01		8.388 2	MHz
I_{INJ+}	Pin Injection Current (positive) Digital Input (2) Analog Input (3)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input (2) Analog Input	V_{DD} = 4.5 to 5.5V			-5	mA

Notes :

1. An oscillator frequency above 1MHz is recommanded for reliable A/D results.
2. A current of ± 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device. For a positive current injected into one pin, a part of this current (~ 10%) can be expected to flow from the neighbouring pins. A current of -5mA can be forced on one input of the analog section at a time (or -2.5mA for all inputs at a time) without affecting the conversion.
3. If a total current of +1mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 1mA, all the conversion is resulting shifted of +1LSB. If a total positive current of +5mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 5mA, all the conversion is resulting shifted of +2LSB.
4. Operation below 0.01 MHz is possible but requires increased supply current.

EEPROM INFORMATION

The ST62xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	RESET, NMI, TIMER, WDON Pin			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	TIMER	$0.80V_{DD}$			V
		RESET, NMI, WDON Pin	$0.70V_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	RESET Pin V_{DD} =5V $V_{IN} = V_{DD}$ (1) $V_{IN} = V_{DD}$ (2) $V_{IN} = V_{SS}$ (5)			10 1 50	µA mA µA
V_{OL}	Low Level Output Voltage	TIMER, I_{OL} = 5.0mA			$0.2V_{DD}$	V
V_{OH}	High Level Output Voltage	TIMER, I_{OL} = -5.0mA	$0.65V_{DD}$			V

Notes on next page

DC ELECTRICAL CHARACTERISTICS (Continued)
(TA = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{PU}	Pull-up Resistor	V_{IN}=0V V_{DD}=5V WDON - NMI	40	100	200	kΩ
		RESET	200	300	500	kΩ
I_{IL} I_{IH}	Input Leakage Current	TIMER $V_{IN} = V_{DD}$ or V_{SS}		0.1	1.0	µA
I_{IL} I_{IH}	Input Leakage Current	NMI VDD = 5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{IL} I_{IH}	Input Leakage Current	WDON VDD = 5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{DD}	Supply Current RUN Mode	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		4	7	mA
	Supply Current WAIT Mode (4)	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		1	2	mA
	Supply Current RESET Mode	f_{OSC} = 8MHz, $V_{RESET} = V_{SS}$		1	7	mA
	Supply Current STOP Mode (3)(4)	I_{LOAD} = 0mA V_{DD} = 5.0V		1	10	µA

Notes :

1. No Watchdog Reset activated.
2. Reset generated by Watchdog.
3. When the watchdog function is actvated the STOP instruction is deactivated. WAIT instruction is automatically executed.
4. All on-chip peripherals in OFF state
5. Pull-up resistor

AC ELECTRICAL CHARACTERISTICS

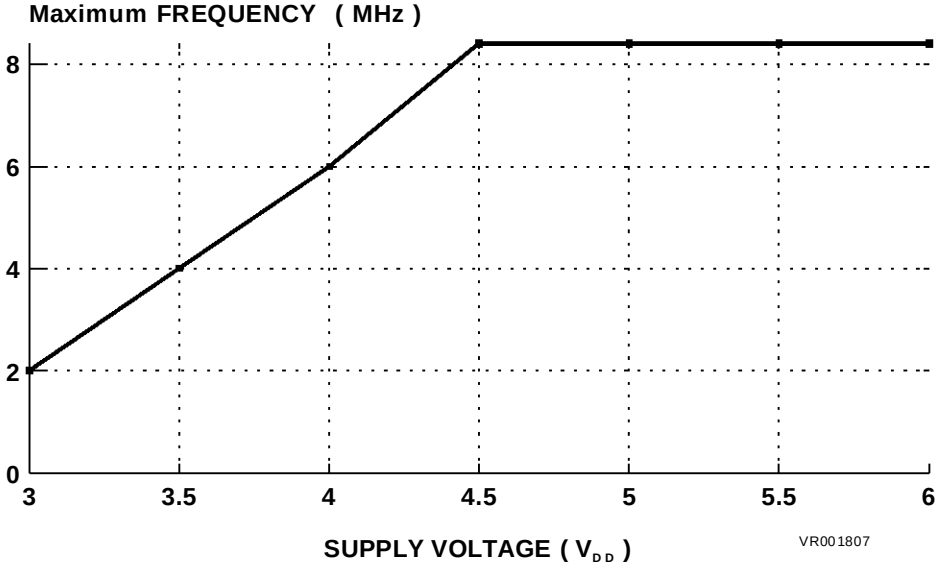
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (2)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01		8.388 2	MHz
t_{SU}	Oscillator Start-up Time	$C_{L1} = C_{L2}$ = 22pF - crystal		5	20	ms
t_{SR}	Supply Rise Time	10% to 90%	0.01		100	
t_{REC}	Supply Recovery Time (1)		100			
T_W	Minimum Pulse Width	NMI Pin V_{DD} = 5V	100			ns
		RESET Pin	100			ns
T_{WEE}	EEPROM Write Time	T_A = 25°C One Byte T_A = 85°C One Byte		5 15	10 25	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A LOT Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention	T_A = 55°C	10			years
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:

1. Period for which V_{DD} has to be connected or at 0V to allow internal Reset function at next power-up.
2. Operation below 0.01 MHz is possible but requires increased supply current.

Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



I/O PORTS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	I/O Pins			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	I/O Pins	$0.7V_{DD}$			V
V_{OL}	Low Level Output Voltage	I/O Pins, I_O = 10µA (sink)			0.1	V
		I/O Pins, $I_{OL} = V_{DD}$x1mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 1.6mA V_{DD} = 3V			0.4	V
	Low Level Output Voltage, PB4-PB7 Only	I/O Pins, $I_{OL} = V_{DD}$x2mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 3.2mA V_{DD} = 3V			0.4	V
		I/O Pins, $I_{OL} = V_{DD}$x4mA V_{DD} = 4.5 to 6V			$0.26xV_{DD}$	V
		I/O Pins, I_{OL} = 6.4mA V_{DD} = 3V			0.8	V
V_{OH}	High Level Output Voltage	I/O Pins, I_O = -10µA (source)	V_{DD}-0.1			V
		I/O Pins, $I_{OL} = -V_{DD}$x1mA V_{DD} = 5.0V	$0.6xV_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	I/O Pins,(1)		0.1	1.0	µA
R_{PU}	Pull-up Resistor	I/O Pins V_{IN} = 0V, V_{DD} = 5.0V	40	100	200	kΩ

Note 1. Pull-up resistor off

SPI ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0V, T_A = -40 to +85°C unless otherwhise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
F_{CL}	Clock Frequency	applied on PB5/SCL			1	MHz
t_{SU}	Set-up Time	applied on PB6/Sin		50		ns
t_h	Hold Time	applied on PB6/Sin		100		ns

A/D CONVERTER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution (3)			8		Bit
A_{TOT}	Total Accuracy (3)	f_{OSC} > 1.2 MHz f_{OSC} > 32kHz			± 2 ±4	LSB
t_C (1)	Conversion Time	f_{OSC} = 8MHz		70		µs
V_{AN}	Conversion Range		V_{SS}		V_{DD}	V
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN} (2)	Analog Input Capacitance			2	5	pF
ASI	Analog Source Impedance				30	kΩ
SSI	Analog Reference Supply Impedence				2	kΩ

Notes:

1. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.
2. Excluding Pad Capacitance
3. Noise at $V_{DD}, V_{SS} \leq 10mV$

TIMER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{12}{f_{OSC}}$			second
f_{IN}	Input Frequency on TIMER Pin				$\frac{f_{OSC}}{8}$	MHz
t_W	Pulse Width at TIMER Pin	$V_{DD} \geq 3V$ $V_{DD} \geq 4.5V$	1 125			µs ns

LCD ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

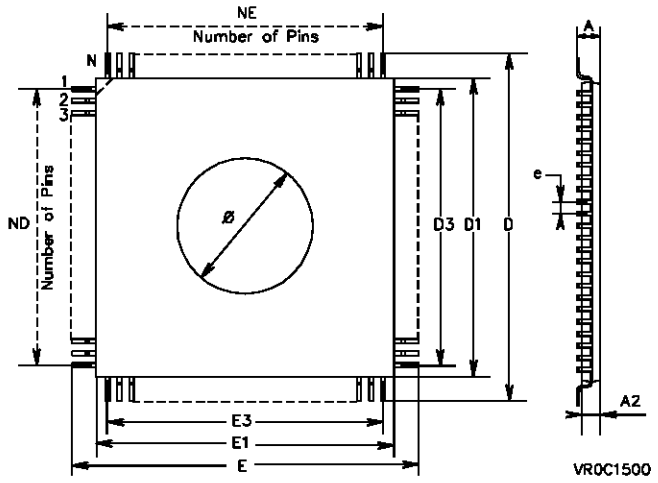
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{FR}	Frame Frequency	1/4 Duty f_{OSC} = 1, 2, 4, 8MHz	16		128	Hz
V_{OS}	DC Offset Voltage(1)	$V_{LCD} = V_{DD}$, no load			50	mV
V_{OH}	COM High Level, Output Voltage	I = 100µA V_{LCD} = 5V	4.5V			V
V_{OL}	COM Low Level, Output Voltage	I = 100µA V_{LCD} = 5V			0.5V	V
V_{OH}	SEG High Level, Output Voltage	I = 50µA V_{LCD} = 5V	4.5V			V
V_{OL}	SEG Low Level, Output Voltage	I = 50µA V_{LCD} = 5V			0.5V	V
V_{LCD}	Display Voltage	Note 2	3		10	V

Notes :

1. The DC offset voltage refers to all segment and common outputs. It is the difference between the measured voltage value and nominal value for every voltage level. Ri of voltage meter must be greater than or equal to 100MΩ.
2. An external resistances network is required when $V_{LCD} \leq 4.5V$.

PACKAGE MECHANICAL DATA

Figure 3. ST62E42 64 Pin Ceramic Quad Flat Package with Window



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		3.55			0.14	
A2		3.40			0.133	
D		17.20			0.677	
D1		14.00			0.551	
D3		12.00			0.472	
E		17.20			0.677	
E1		14.00			0.551	
E3		12.00			0.472	
Ø		7.62			0.3	
e		0.80			0.032	
Number of Pins						
ND	16					
NE	16					
N	64					

ORDERING INFORMATION TABLE

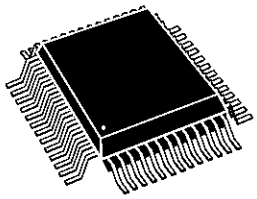
Sales Types	Memory type	Temperature Range	Package
ST62E42G1	8K EPROM	tested at 25°C only	CQFP64-W
ST62T42Q6	8K EPROM	-40 to + 85°C	PQFP64

SGS-THOMSON MICROELECTRONICS

ST6245

8-BIT HCMOS MCU WITH LCD DRIVER, EEPROM AND A/D CONVERTER

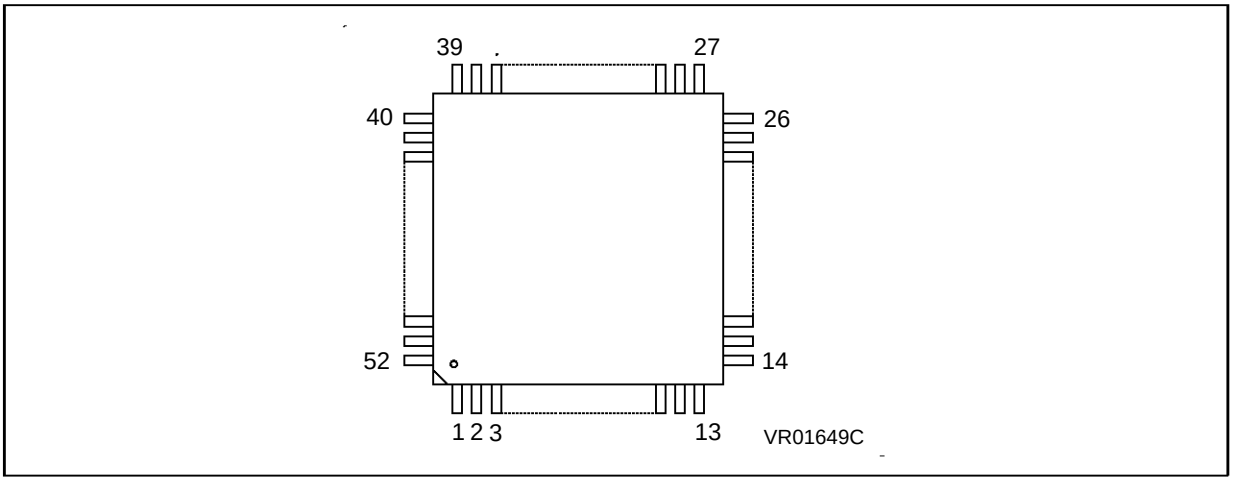
- 3 to 6V supply operating range
- 8.4MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait & Stop Modes
- 5 different interrupt vectors
- Look-up table capability in ROM
- User ROM: 3884 bytes
  Data RAM: 128 bytes
  LCD RAM: 12 bytes
  EEPROM: 64 bytes
- PQFP52 package
- 11 fully software programmable I/O as:
  - Input with/without pull-up resistor
  - Input with interrupt generation
  - Open-Drain or Push-pull outputs
  - Analog Inputs (7 pins)
- 4 I/O lines can sink up to 20mA for direct LED or TRIAC driving, and have SPI alternate functions
- Two 8-bit counters with 7-bit programmable prescalers (Timers 1 and 2)
- Software activated digital watchdog
- 8-bit A/D converter with up to 7 analog inputs
- 8-bit synchronous Serial Peripheral Interface (SPI)
- LCD driver with 24 segment outputs, 4 back-plane outputs and selectable duty cycle for up to 96 LCD segments direct driving
- 32kHz oscillator for stand-by LCD operation
- One external not maskable interrupt
- 9 powerful addressing modes
- The accumulator, the X, Y, V & W registers, the port and peripherals data & control registers are addressed in the data space as RAM locations.
- The ST62E45 is the EPROM version, ST62T45 is the OTP version
- Development tool: ST6245-EMU connected via RS232 to an MS-DOS Personal Computer



PQFP52

(Ordering Information at the end of the datasheet)

Figure 1. 52 Pin Quad Flat Pack (QFP) Package Pinout



ST6245 Pin Description

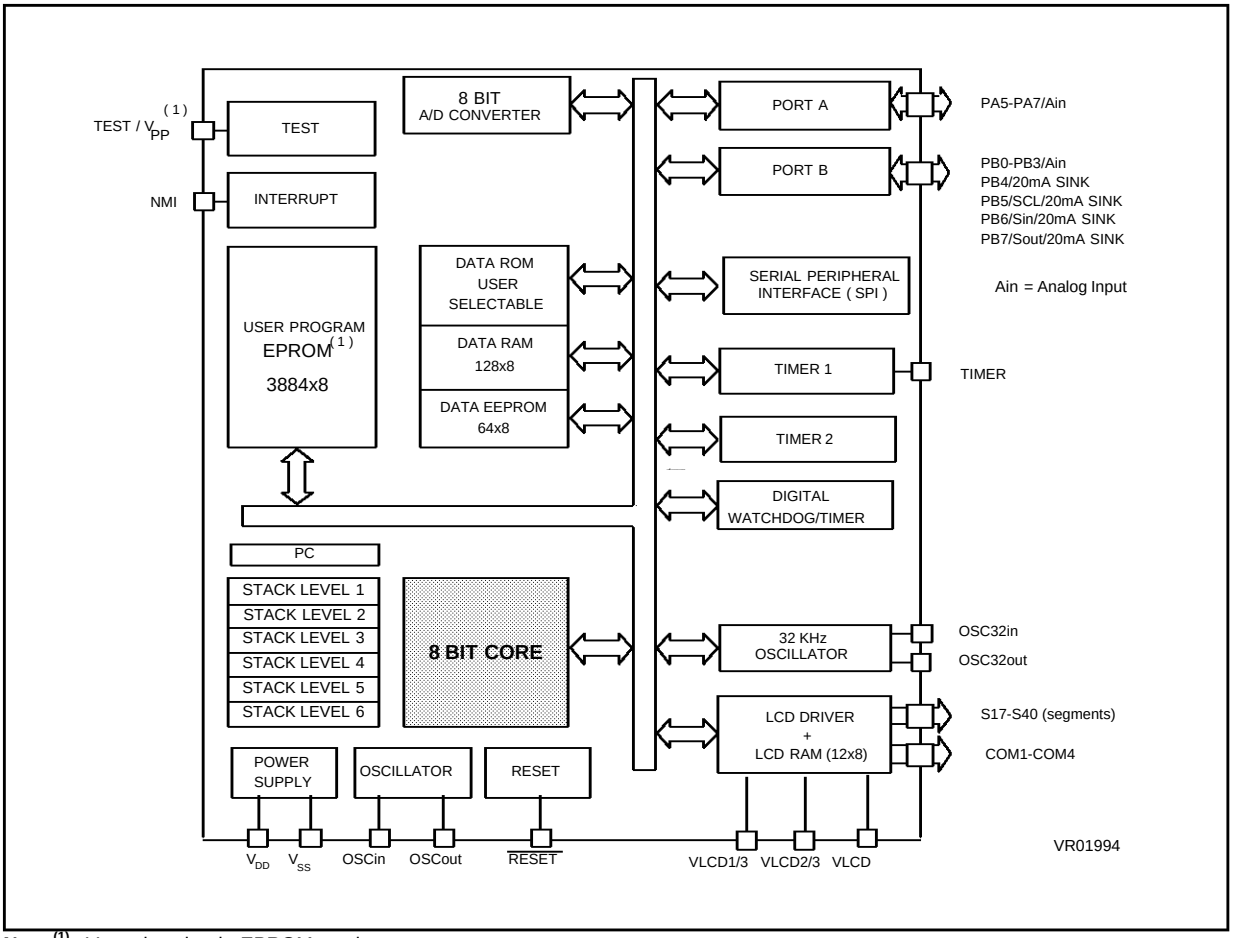
Pin number	Pin name	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
1	COM4	14	$\overline{\text{RESET}}$	27	OSC32out	40	S28
2	COM3	15	OSCout	28	OSC32in	41	S29
3	COM2	16	OSCin	29	S17	42	S30
4	COM1	17	NMI	30	S18	43	S31
5	VLCD1/3	18	TIMER	31	S19	44	S32
6	VLCD2/3	19	PB7/Sout (1)	32	S20	45	S33
7	VLCD	20	PB6/Sin (1)	33	S21	46	S34
8	PA7/Ain	21	PB5/SCL (1)	34	S22	47	S35
9	PA6/Ain	22	PB4 (1)	35	S23	48	S36
10	PA5/Ain	23	PB3/Ain	36	S24	49	S37
11	TEST	24	PB2/Ain	37	S25	50	S38
12	V_{DD}	25	PB1/Ain	38	S26	51	S39
13	V_{SS}	26	PB0/Ain	39	S27	52	S40

Note 1: 20mA SINK

GENERAL DESCRIPTION

The ST6245 microcontroller is a member of the 8-bit HCMOS ST62xx family, a series of devices oriented to low-medium complexity applications. All ST62xx members are based on a building block approach: a common core is associated with a combination of on-chip peripherals (macrocells). The macrocells of the ST6245 are: a high performance LCD controller/driver with 24 segment outputs and 4 backplanes able to drive up to 96 segments, two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), the digital watchdog (DWD), an 8-bit A/D Converter with up to 7 analog inputs and an 8-bit synchronous Serial Peripheral Interface (SPI). In addition these devices offer 64 bytes of EEPROM for storage of non volatile data. Thanks to these peripherals the ST6245 is well suited for general purpose, automotive, security, appliance and industrial applications. The ST62E45 EPROM version is available for prototypes and low-volume production, an OTP version is also available (see separate datasheet).

Figure 2. ST6245 Block Diagram



Note (1): V_{PP} only exists in EPROM version

PIN DESCRIPTION

V_{DD} and **V_{SS}**. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected with the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. OSCin is the input pin, OSCout is the output pin. An external clock signal can be applied to OSCin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to restart the microcontroller at the beginning of its program. The $\overline{\text{RESET}}$ pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TEST. The TEST must be held at V_{SS} for normal operation (an internal pull-down resistor selects normal operating mode if TEST pin is not connected).

NMI. The NMI pin provides the capability for asynchronous applying an external top priority interrupt to the MCU.This pin is falling edge sensitive. The NMI pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TIMER. This is the TIMER 1 I/O pin. In input mode it is connected to the prescaler and acts as external timer clock or as control gate for the internal timer clock. In the output mode the TIMER pin outputs the data bit when a time-out occurs.

PA5-PA7. These 3 lines are organized as one I/O port (A). Each line may be configured under software control as an input with or without pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output or as analog input for the A/D converter. Port A has a 5mA drive capability in output mode.

PB0-PB3,PB4-PB7. These 8 lines are organized as one I/O port (B). Each line may be configured under software control as an input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output. PB0-PB3 can be programmed as analog inputs for the A/D converter while PB4-PB7 can also sink 20mA for direct LED driving. PB5-PB7 can also be used as respectively Clock, Data in and Data out pins for the on-chip SPI to carry the synchronous serial I/O signals.

COM1-COM4. These four pins are the LCD peripheral common outputs. They are the outputs of the on-chip backplane voltage generator which is used for multiplexing the 24 LCD lines allowing up to 96 segments to be driven.

S17-S40. These pins are the 24 LCD peripheral driver outputs of ST6245. Segments S1-S16 and S41-S48 are not connected to any pin.

VLCD. Display voltage supply. It determines the high voltage level on COM1-COM4 and S1-S24 pins.

VLCD1/3, VLCD2/3. Display supply voltage inputs for determining the display voltage levels on COM1-COM4 and S1-S24 pins during multiplex operation.

OSC32in and OSC32out. These pins are internally connected with the on-chip 32kHz oscillator circuit. A 32.768kHz quartz crystal can be connected between these two pins if it is necessary to provide the LCD stand-by clock and real time interrupt. OSC32in is the input pin, OSC32out is the output pin.

ST62xx CORE

The core of the ST62xx Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control busses. The in-core communication is arranged as shown in Figure 3; the controller being externally linked to both the reset and the oscillator, while the core is linked to the dedicated on-chip macrocells peripherals via the serial data bus and indirectly for interrupt purposes through the control registers.

Registers

The ST62xx Family core has six registers and three pairs of flags available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs.

Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is addressed in the data space as RAM location at address FFh. Accordingly, the ST62xx instruction set can use the accumulator as any other register of the data space.

Figure 4. ST62xx Core Programming Model

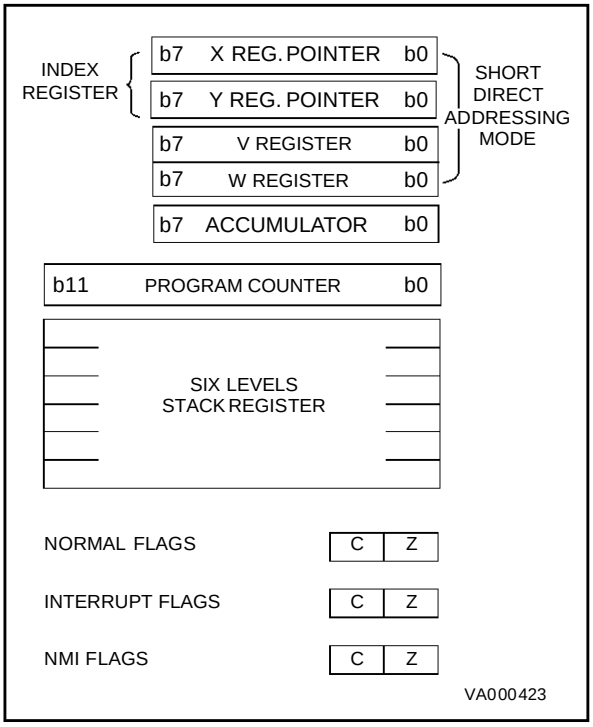
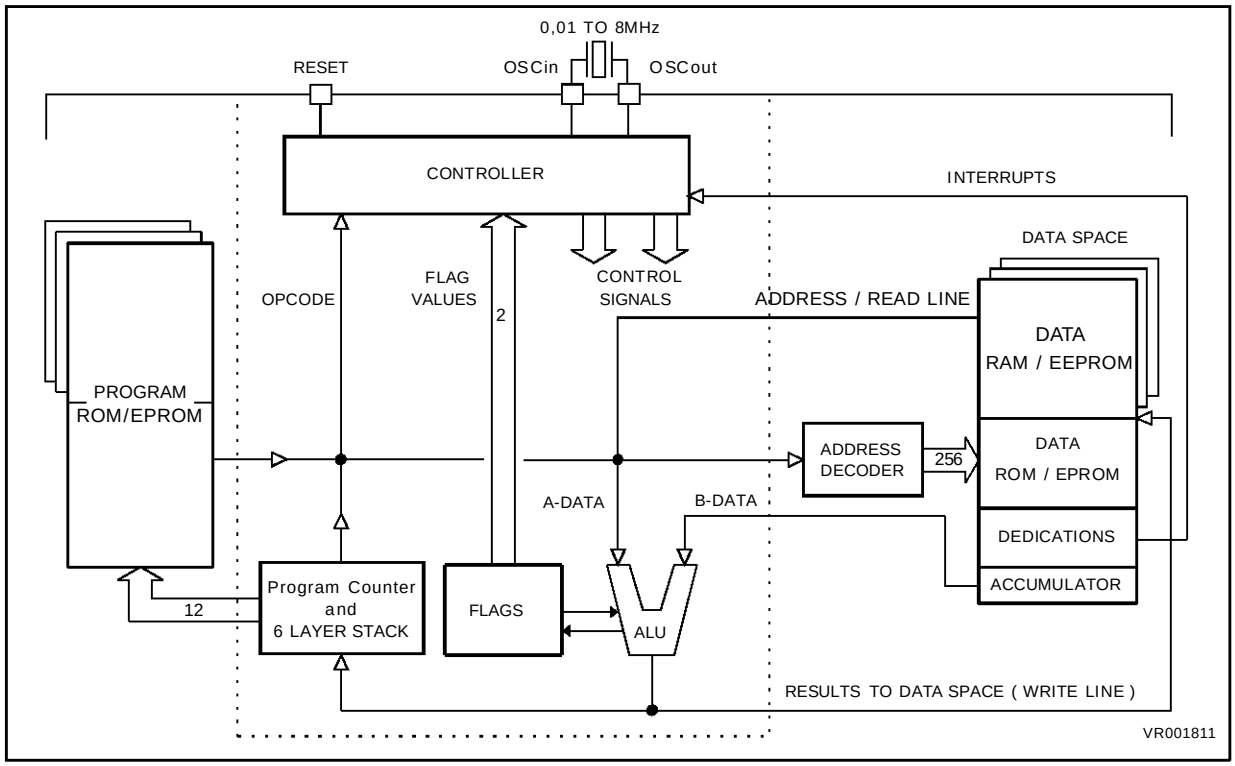


Figure 3. ST62xx Core Block Diagram



ST62xx CORE (Continued)

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in the data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST62xx instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save one byte in short direct addressing mode. These registers can be addressed in the data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST62xx instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or an address of operand. The 12-bit length allows the direct addressing of 4096 bytes in the program space.

The PC value is incremented after it is read from the address of the current instruction. To execute relative jumps the PC and the offset are shifted through the ALU, where they will be added, and the result is shifted back into the PC. The program counter can be changed in the following ways:

- JP (Jump) instruction . . . PC=Jump address
- CALL instruction . . . . PC=Call address
- Relative Branch instructions . . . . . PC=PC ± offset
- Interrupt . . . . . PC=Interrupt vector
- Reset . . . . . PC=Reset vector
- RET & RETI instructions . PC=Pop (stack)
- Normal instruction . . . . PC=PC+1

Flags (C, Z)

The ST62xx core includes three pairs of flags that correspond to 3 different modes: normal mode, interrupt mode and Non-Maskable Interrupt-Mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during normal operation, one pair is used during the interrupt mode (CI, ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

The ST62xx core uses the pair of flags that correspond to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST62xx core uses the interrupt flags (resp. the NMI flags) instead of the normal flags. When the RETI instruction is executed, the normal flags (resp. the interrupt flags) are restored if the MCU was in the normal mode (resp. in the interrupt mode) before the interrupt. It should be observed that each flag set can only be addressed in its own mode (Not-maskable interrupt, normal interrupt or main mode). The flags are not cleared during the context switching and so remain in the state they were at the exit of the last mode switch.

The Carry flag is set when a carry or a borrow occurs during arithmetic operations, otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The Zero flag is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

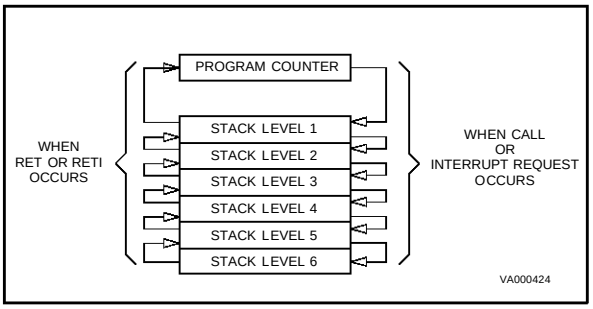
The switching between the three sets of Flags is automatically performed when an NMI, an interrupt or a RETI instruction occurs. As the NMI mode is automatically selected after the reset of the MCU, the ST62xx core uses at first the NMI flags.

ST62xx CORE (Continued)

Stack

The ST62xx core includes a true LIFO hardware stack that eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be lost). When a subroutine or interrupt return occurs (RET or RETI instructions), the first level register is shifted back into the PC and the value of each level is popped back into the previous level. These two operating modes are described in Figure 5. Since the accumulator, as all other data space registers, is not stored in the stack, the handling of these registers should be performed inside the subroutine. The stack pointer will remain in its deepest position if more than 6 calls or interrupts are executed, so that the last return address will be lost. It will also remain in its highest position if the stack is empty and a RET or RETI is executed. In this case the next instruction will be executed.

Figure 5. Stack Operation



MEMORY SPACES

The MCU operates in three different memory spaces: program space, data space, and stack space. A description of these spaces is shown in the following figures.

Program Space

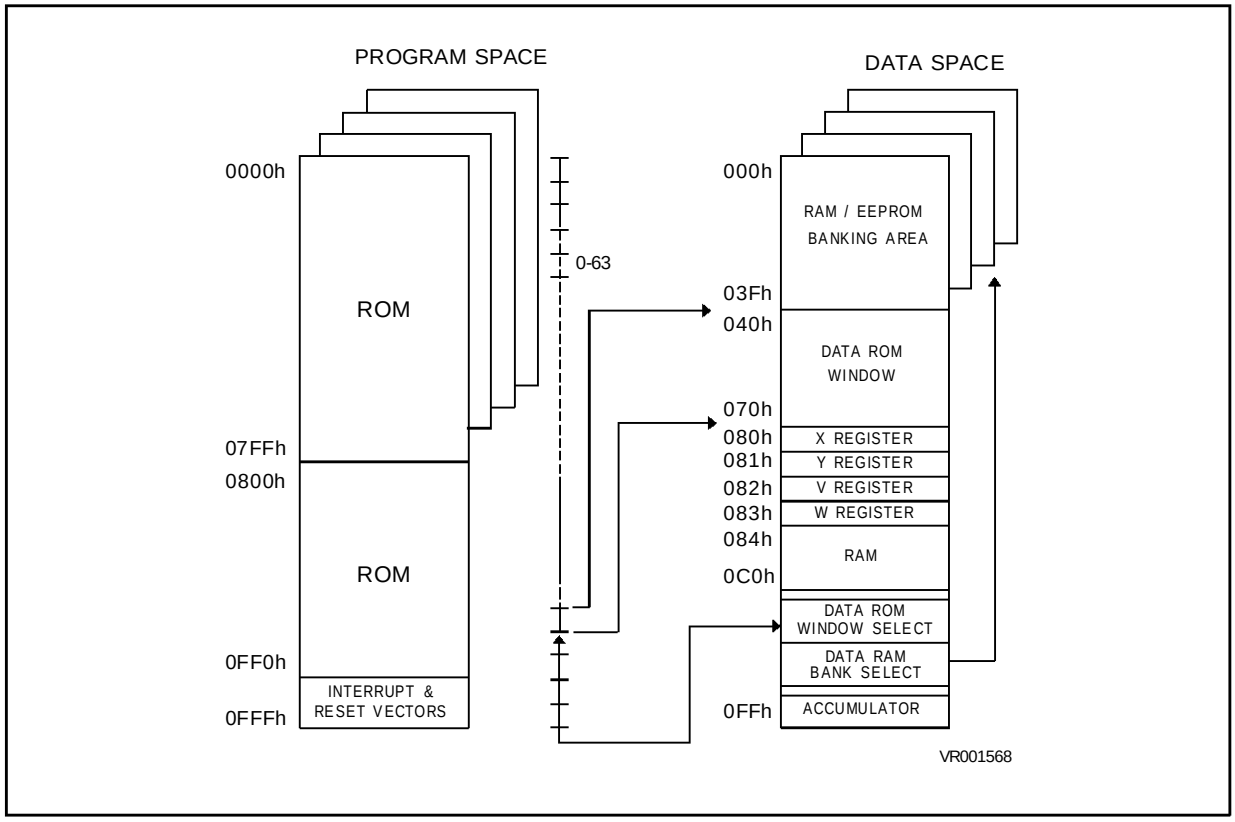
The program space is physically implemented in the ROM memory and includes all the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, the reserved test area and the user vectors. It is addressed by the 12-bit Program Counter register (PC register) and so the ST62xx core can directly address up to 4K bytes of Program Space.

Table 1. ST6245 Program ROM Memory Map

Device Address	Description
0000h-007Fh	Reserved
0080h-0F9Fh	User ROM
0FA0h-0FEFh	Reserved
0FF0h-0FF7h	Interrupt Vectors
0FF8h-0FFBh	Reserved
0FFCh-0FFDh	NMI Vector
0FFEh-0FFFh	Reset Vector

MEMORY SPACES (Continued)

Figure 6. ST62xx Memory Addressing Description Diagram



MEMORY SPACES (Continued)

Data Space

The instruction set of the ST62xx core operates on a specific space, named Data Space, that contains all the data necessary for the processing of the program. The Data Space allows the addressing of RAM memory, ST62xx core/peripheral registers, and read-only data such as constants and look-up tables.

Data ROM. All the read-only data is physically implemented in the ROM memory in which the Program Space is also implemented. The ROM memory contains consequently the program to be executed, the constants and the look-up tables needed for the program.

The locations of Data Space in which the different constants and look-up tables are addressed by the ST62xx core can be considered as being a 64-byte window through which it is possible to access to the read-only data stored in the ROM memory .

Data RAM/EEPROM. The ST6245 offers 128 bytes of data RAM memory and 64 bytes of EEPROM. 64 bytes of RAM are directly addressed in data space in the range 080h-0BFh (static space). The additional RAM are addressed using the banks of 64 bytes located between addresses 00h and 3Fh.

Additionally RAM are available in the LCD data map from E0h to F7h and are not banked.

Stack Space

The stack space consists of six 12 bit registers that are used for stacking subroutine and interrupt return addresses plus the current program counter register.

Figure 7. ST6240 Data Memory Space

DATA RAM/EEPROM BANK AREA	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
SPI INT. DISABLE REGISTER	0C2h*
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
RESERVED	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh*
DATA RAM/EEPROM BANK REGISTER	0CBh*
PORT A OPTION REGISTER	0CCh
RESERVED	0CDh
PORT B OPTION REGISTER	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER 1 PRESCALER REGISTER	0D2h
TIMER 1 COUNTER REGISTER	0D3h
TIMER 1 STATUS/CONT REGISTER	0D4h
TIMER 2 PRESCALER REGISTER	0D5h
TIMER 2 COUNTER REGISTER	0D6h
TIMER 2 STATUS/CONT REGISTER	0D7h
WATCHDOG REGISTER	0D8h
RESERVED	0D9h
RESERVED	0DAh
32kHz OSC. CONTROL REGISTER	0DBh
LCD MODE CONTROL REGISTER	0DCh
SPI DATA REGISTER	0DDh
RESERVED	0DEh
EEPROM CONTROL REGISTER	0DFh
LCD RAM	0E0h 0F7h
DATA RAM 7 BYTES	0F8h 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

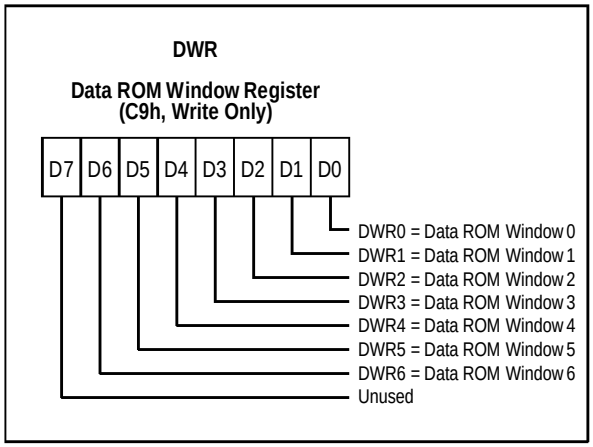
MEMORY SPACES (Continued)

Data Window register (DWR)

The Data ROM window is located from address 040h to address 7Fh in the Data space. It allows the direct reading of 64 consecutive bytes located anywhere in the ROM memory between the addresses 0000h and 1FFFh. All the bytes of the ROM memory can be used to store either instructions or read-only data. Indeed, the window can be moved by step of 64 bytes along the ROM memory in writing the appropriate code in the Write-only Data Window register (DWR register, location C9h).

The DWR register can be addressed like a RAM location in the Data Space at the address C9h, nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to move the 64-byte read-only data window (from the 40h address to 7Fh address of the Data Space) up and down the ROM memory of the MCU in steps of 64 bytes. The effective address of the byte to be read as a data in the ROM memory is obtained by the concatenation of the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits, see Figure 8). So when addressing location 40h of dataspace, and 0 is loaded in the DWR register, the phisycal addressed location in ROM is 00h. The DWR register is not cleared at reset, therefore it must be written to before the first access to the Data ROM window area.

Figure 9. Data ROM Window Register



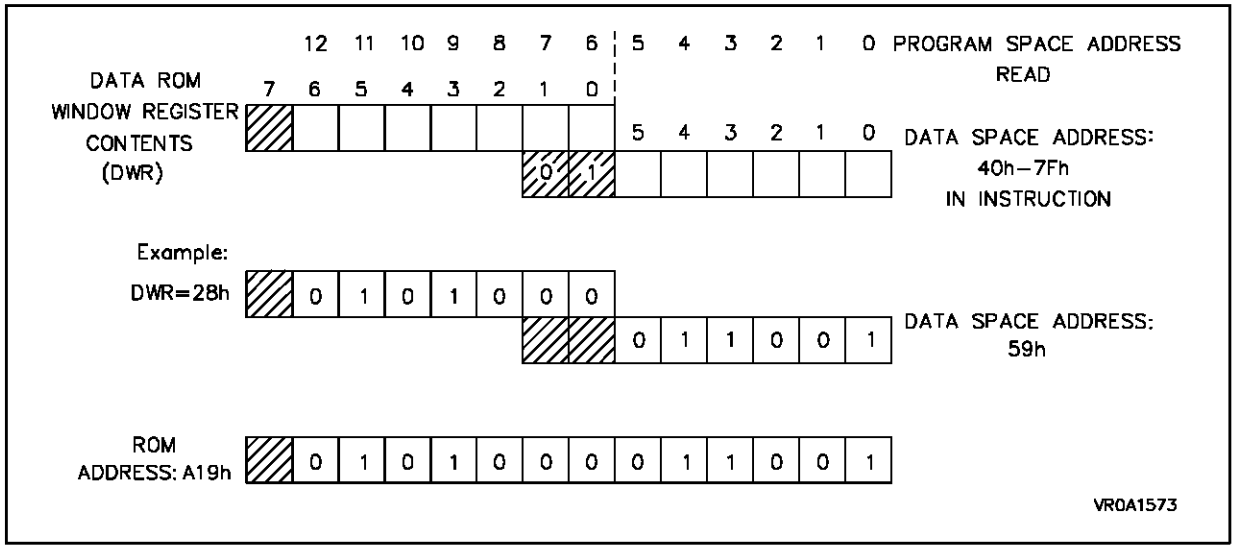
D7. This bit is not used.

DWR6-DWR0. These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Note: Care is required when handling the DWR register as it is write only. For this reason, it is not allowed to change the DWR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in the interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to the DWR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DWR is not affected.

Figure 8. Data ROM Window Memory Addressing



MEMORY SPACES (Continued)

Data RAM/EEPROM Bank Register (DRBR)

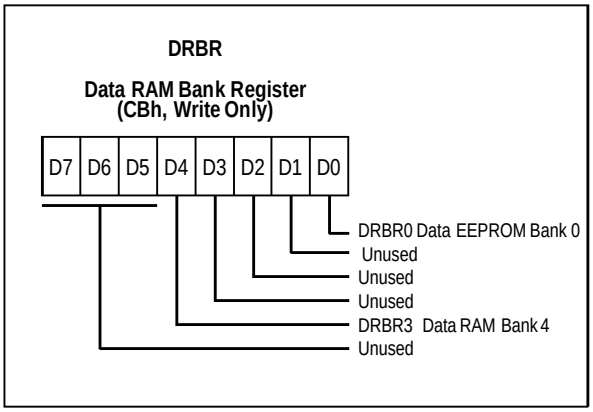
The selection of the bank is made by programming the Data RAM Bank Switch register (DRBR register) located at address CBh of the Data Space. The number of the selected bank is equal to the bit content of the DRBR register. In this way each bank of RAM or EEPROM can be selected 64 bytes at a time. No more than one bank should be set at a time.

The DRBR register can be addressed like a RAM location in the Data Space at the address CBh; nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to select the desired 64-byte RAM/EEPROM bank of the Data Space. The number of the bank has to be loaded in the DRBR register and the instruction has to point to the selected location as if it was in bank 0 (from 00h address to 3Fh address). This register is not cleared during the MCU initialization, therefore it must be written before the first access to the Data Space bank region. Refer to the Data Space description for additional information. The DRBR register is not modified when a interrupt or a subroutine occurs.

Table 2. Data RAM Bank Register Set-up

DRBR Value	Selection
01h	EEPROM
10h	RAM

Figure 10. Data RAM Bank Register



The following Table 2 summarizes how to set the data RAM bank register in order to select the various banks or pages.

D7-D5. These bits are not used.

DRBR3. This bit, when set, will select RAM page.

D3-D1. These bits are not used.

DRBR0. This bit, when set, will select EEPROM page.

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Notes:

Care is required when handling the DRBR register as it is write only. For this reason, it is not allowed to change the DRBR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to DRBR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DRBR is not affected.

In DRBR Register, *only 1 bit must be set*. Otherwise two or more pages are enabled in parallel, producing errors.

MEMORY SPACES (Continued)

EEPROM Description

The data space of ST62xx family from 00h to 3Fh is paged as described in Table 3. The ST6245 has 64 bytes of EEPROM located in one page of 64 bytes (page 0).

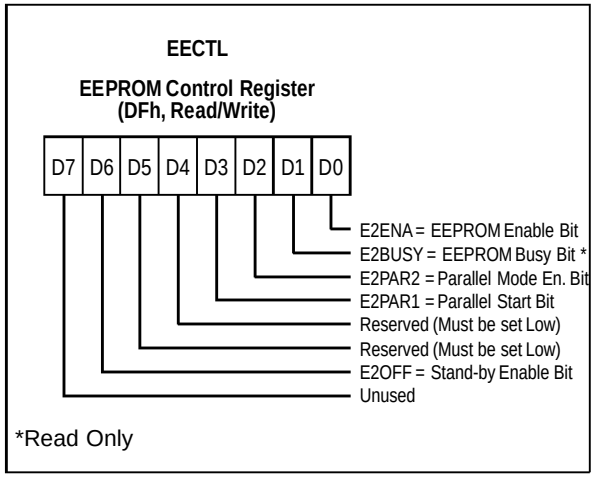
The EEPROM pages are physically organized in 32 byte modules (2 modules per page) and does not require dedicated instructions to be accessed in reading or writing. The EEPROM is controlled by the EEPROM Control Register (EECTL = DFh). In order to enable access to the EEPROM, bit 6 of this register must be cleared otherwise any access to the EEPROM will be meaningless.

Any EEPROM location can be read just like any other data location, also in terms of access time.

When writing to an EEPROM, the EEPROM is not accessible by the ST62xx. A busy flag can be read to identify the EEPROM status before attempting any access. Writing the EEPROM can work in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). BMODE is the normal way to use the EEPROM and consists in accessing one byte at a time. PMODE consists in accessing 8 bytes per time.

Readout of the EEPROM is made at the same speed as RAM acces.

Figure 11. EEPROM Control Register



D7. Not Used

E2OFF. *WRITE ONLY.* If this bit is set the EEPROM is disabled (any access will be meaningless) and the power consumption of the EEPROM is reduced to the lowest values.

D5, D4. Reserved, must be set to zero.

E2PAR1. *WRITE ONLY.* Once in Parallel Mode, as soon as the user software sets the E2PAR1 bit the parallel writing of the 8 adjacent registers will start. It is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written; after parallel programming the undefined bytes will be unaffected

E2PAR2. *WRITE ONLY.* This bit must be set by the user program in order to perform parallel programming (more than one byte at a time). If E2PAR2 is set and the parallel start bit (E2PAR1) is low, up to 8 adjacent bytes can be written at maximum speed, the contents being stored in volatile registers. These 8 adjacent bytes are considered as a row, whose address lines A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bits. E2PAR2 is automatically reset at the end of any parallel programming procedure. It can be reset by the user software before starting the programming procedure, leaving the EEPROM registers unchanged.

E2BUSY. *READ ONLY.* This bit will be automatically set by the EEPROM control logic when the user program modifies an EEPROM register. The user program must test it before any read or write EEPROM operation; any attempt to access the EEPROM while the busy bit is set will be aborted and the writing procedure in progress completed.

E2ENA. *WRITE ONLY.* This bit MUST be set to one in order to write to any EEPROM register. If the user program attempts to write to the EEPROM when E2ENA = “0”, the involved registers will be unaffected and the BS will not be set.

After RESET the content of EECTL register will be 00h.

Notes:

The data to write has to be written directly at the address that it will have inside the EEPROM space. There is no buffer memory between the data•RAM and the EEPROM spaces.

When the EEPROM is busy (E2BUSY = “1”) EECTL can not be accessed in write mode, it is only possible to read the status of E2BUSY. This implies that as long as the EEPROM is busy, it is not possible to change the status of the EEPROM control register. EECTL bits 4 and 5 are reserved for test purposes, and must never be set to “1”.

MEMORY SPACES (Continued)

Table 3. EEPROM Parallel Write Row Structure

Byte	0	1	2	3	4	5	6	7	Dataspace addresses. Banks 0 and 1.
ROW7									38h-3Fh
ROW6									30h-37h
ROW5									28h-2Fh
ROW4									20h-27h
ROW3									18h-1Fh
ROW2									10h-1Fh
ROW1									08h-0Fh
ROW0									00h-07h

Up to 8 bytes in each row may be programmed at the same time in Parallel Write mode

Additional Notes on Parallel Mode. If the user wishes to perform parallel programming, the first action should be to set the E2PAR2 bit to one. From this time the EEPROM will be addressed in writing, the ROW address will be latched and it will be possible to change it only at the end of the programming procedure or by resetting E2PAR2 without programming the EEPROM. After the ROW address latching the ST62xx can “see” only one EEPROM row (the selected one) and any attempt to write or read other rows will produce errors. Do not read the EEPROM while E2PAR2 is set.

As soon as E2PAR2 bit is set, the 8 volatile ROW latches are cleared. From this moment the user can load data in the whole ROW or in a subset. Setting E2PAR1 will modify the EEPROM registers corresponding to the ROW latches accessed after E2PAR2. For example, if the software sets E2PAR2 and accesses the EEPROM by writing to addresses 18h, 1Ah, 1Bh and then sets E2PAR1, these three registers will be modified at the same time; the remaining bytes will be unaffected. Note that E2PAR2 is internally reset at the end of the programming procedure. This implies that the user must set E2PAR2 bit between two parallel programming procedures. Note that if the user tries to set E2PAR1 while E2PAR2 is not set there will not be any programming procedure and the E2PAR1 bit will be unaffected. Consequently E2PAR1 bit cannot be set if E2ENA is low. E2PAR1 can be affected by the user to set it, only if E2ENA and E2PAR2 bits are also set to one.

TEST MODE

For normal operation the TEST pin must be held low. An on-chip 100kΩ pull-down resistor is internally connected to the TEST pin.

INTERRUPTS

The ST62xx core can manage 4 different maskable interrupt sources, plus one non-maskable interrupt source (top priority level interrupt). Each source is associated with a particular interrupt vector that contains a Jump instruction to the related interrupt service routine. Each vector is located in the Program Space at a particular address.
When a source provides an interrupt request, and the request processing is also enabled by the ST62xx core, then the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction). Finally, the PC is loaded with the address of the Jump instruction and the interrupt routine is processed.

The ST6245 microcontroller has eight different interrupt sources associated to different interrupt vectors as described in Table 4.

Table 4. Interrupt Vectors - Sources Relationship

Interrupt Source	Associated Vector	Vector Address
NMI Pin	Interrupt Vector #0	(FFCh-FFDh)
SPI Peripheral	Interrupt Vector #1	(FF6h-FF7h)
Port A& B Pins	Interrupt Vector #2	(FF4h-FF5h)
TIMER 1, 2 & 32kHz Oscillator	Interrupt Vector #3	(FF2h-FF3h)
ADC Peripheral	Interrupt Vector #4	(FF0h-FF1h)

Interrupt Vectors Description

The ST62xx core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines in the static page of the Program Space:

– The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at addresses FFCh,FFDh in the Program Space. On ST6245 this vector is associated with the external falling edge sensitive interrupt pin (NMI). An on-chip 100kΩ pull-up resistor is internally connected to the NMI pin.

– The interrupt vector located at the addresses FF6h, FF7h is named interrupt vector #1. It is associated with SPI peripheral and can be programmed by software to generate an interrupt request after the falling edge or low level of the eighth external clock pulse according to the code loaded in the Interrupt Option Register (IOR).

– The interrupt vector located at the addresses FF4h, FF5h is named interrupt vector #2. It is associated with Port A and B pins and can be programmed by software either in the falling edge detection mode or in the rising edge detection mode according to the code loaded in the Interrupt Option Register (IOR).

– The interrupt vector located at the addresses FF2h, FF3h is named interrupt vector #3. It is associated with Timer 1, Timer 2 and the 32kHz Oscillator peripherals. All these interrupts are “ORed” together and are connected to interrupt line #3 of the core. Discrimination among the three interrupts must be made by polling the Status/Control registers of Timer 1 (0D4h), Timer 2 (0D7h) and 32kHz oscillator (0DBh).

– The interrupt vector located at the addresses FF0h, FF1h is named interrupt vector #4. It is associated with the A/D converter peripheral.

All the on-chip peripherals (refer to their descriptions for further details) have an interrupt request flag bit (TMZ for timer, EOC for A/D, etc.), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI for timer, EAI for A/D, etc.) that must be set to one to allow the transfer of the flag bit to the Core.

Interrupt Priority

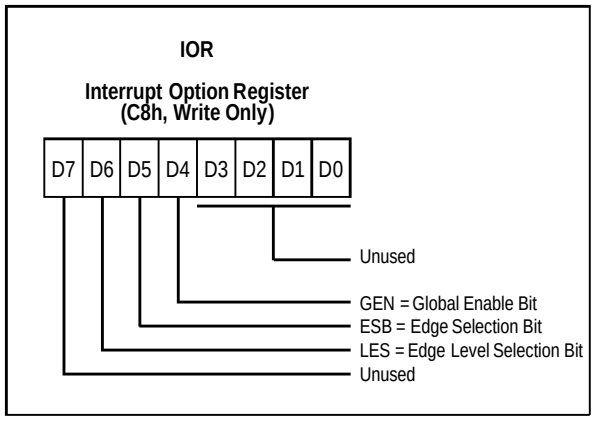
The non-maskable interrupt request has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the four other interrupts cannot interrupt each other. If more than one interrupt request is pending, they are processed by the ST62xx core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

INTERRUPTS (Continued)

Interrupt Option Register

The Interrupt Option Register (IOR register, location C8h) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register can be addressed in the Data Space as RAM location at the address C8h, nevertheless it is a write-only register that cannot be accessed with single-bit operations. The operating modes of the external interrupt inputs associated to interrupt vectors #1 and #2 are selected through bits 5 and 6 of the IOR register.

Figure 12. Interrupt Option Register



D7. This bit is not used.

LES. Level/Edge Selection Bit. When this bit is set to one, the interrupt #1 (SPI) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

ESB. Edge Selection Bit. When this bit is set to one, the interrupt #2 (Port A & B lines) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

GEN. Global Enable Interrupt. When this bit is set to one, all the interrupts are enabled. When this bit is cleared to zero all the interrupts (but NMI) are disabled.

This register is cleared on reset.

Table 5. Interrupt Option Register Description

GEN	SET	Enable all interrupts
	CLEARED	Disable all interrupts
ESB	SET	Rising edge mode on interrupt input #2
	CLEARED	Falling edge mode on interrupt input #2
LES	SET	Level-sensitive mode on interrupt input #1
	CLEARED	Falling edge mode on interrupt input #1
OTHERS	NOT USED	

External Interrupts Operating Modes

The NMI interrupt is associated to the NMI pin of the ST6245. The highest priority interrupt request will be generated by a falling edge applied to the NMI pin. The NMI interrupt pin signal is latched and is automatically reset by the core at the beginning of the non-maskable interrupt service routine. An on-chip pull-up resistor and a schmitt trigger is available with the NMI pin.

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (SPI vector #1, Ports A and B vector #2,) are connected to two internal latches. Each latch is set when a falling/rising edge occurs and is cleared when the associated interrupt routine is started. So, the occurrence of an external interrupt request is stored: a second interrupt, that occurs during the processing of the first one, will be processed as soon as the first one has been finished (if there is not an higher priority interrupt request). If more than one interrupt occurs during the processing of the first one, these other interrupt requests will be lost.

The storage of the interrupt requests is not available in the level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the core samples the line after the execution of the instructions.

During the end of each instruction the core tests the interrupt lines and if there is an interrupt request the next instruction is not executed and the related interrupt routine is executed.

INTERRUPTS (Continued)

Interrupt Procedure. The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event the user does not know about the context and the time at which it occurred. As a result the user should save all the data space registers which will be used inside the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes which are automatically switched and so these do not need to be saved.

The following list summarizes the interrupt procedure:

ST62xx actions

– Interrupt detection

– The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (or the NMI flags)

– The value of the PC is stored in the first level of the stack

– The normal interrupt lines are inhibited (NMI still active)

– First internal latch is cleared

– The related interrupt vector is loaded in the PC.

User actions

– User selected registers are saved inside the interrupt service routine (normally on a software stack)

– The source of the interrupt is found by polling (if more than one source is associated to the same vector) the interrupt flag of the source.

– Interrupt servicing

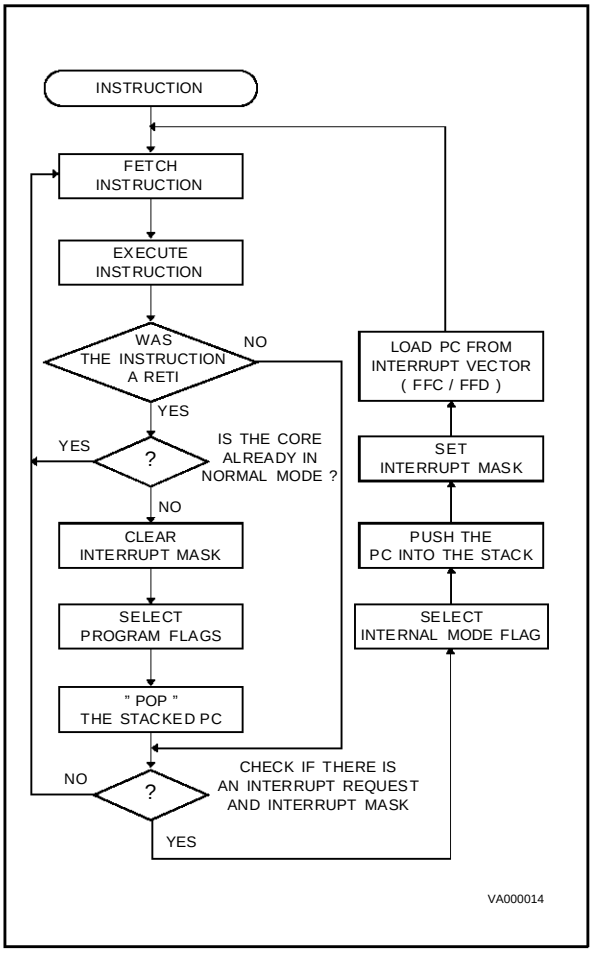
– Return from interrupt (RETI)

ST62xx actions

– Automatically the ST62xx core switches back to the normal flags (or the interrupt flags) and pops the previous PC value from the stack

The interrupt routine begins usually by the identification of the device that has generated the interrupt request (by polling). The user should save the registers which are used inside the interrupt routine (that holds relevant data) into a software stack. After the RETI instruction execution, the core carries out the previous actions and the main routine can continue.

Figure 13. Interrupt Processing Flow-Chart



WARNING. *GEN is the global enable for all interrupts except NMI. If this bit is cleared, the NMI interrupt is accepted when the ST62xx core is in the normal RUN Mode.*
If the ST62xx core is in STOP or WAIT Mode, the NMI is not accepted as a restart is disabled. This state can only be finished by a reset (fromthe Watchdog or an external Reset Signal).
As a consequence the NMI can be masked in STOP and WAIT modes, but not in RUN mode.

INTERRUPTS (Continued)

Interrupt request and mask bits

Interrupt Option Register, IOR Location C8h

– GEN. If this bit is set, all the ST62xx interrupts are enabled, if reset all interrupts are disabled (including the NMI).

– ESB. If this bit is set, all the input lines associated to interrupt vector #2 are rising edge sensitive, if reset they are falling edge sensitive.

– LES. If this bit is set, all the inputs lines associated to interrupt vector #1 are low level sensitive, if reset they are falling edge sensitive.

All other bits in this register are not used.

Timer Peripherals, TSCR1 and TSCR2 registers, locations D4h and D7h

– TMZ. A low-to-high transition indicates that the timer count register has decremented to zero. This means that an interrupt request can be generated in relation to the state of ETI bit.

– ETI. This bit, when set, enables the timer interrupt request.

A/D Converter Peripheral, ADCR register location D0h

– EOC. This read only bit indicates when a conversion has been completed, by going to one. An interrupt request can be generated in relation to the state of EAI bit.

– EAI. This bit, when set, enables the A/D converter interrupt request.

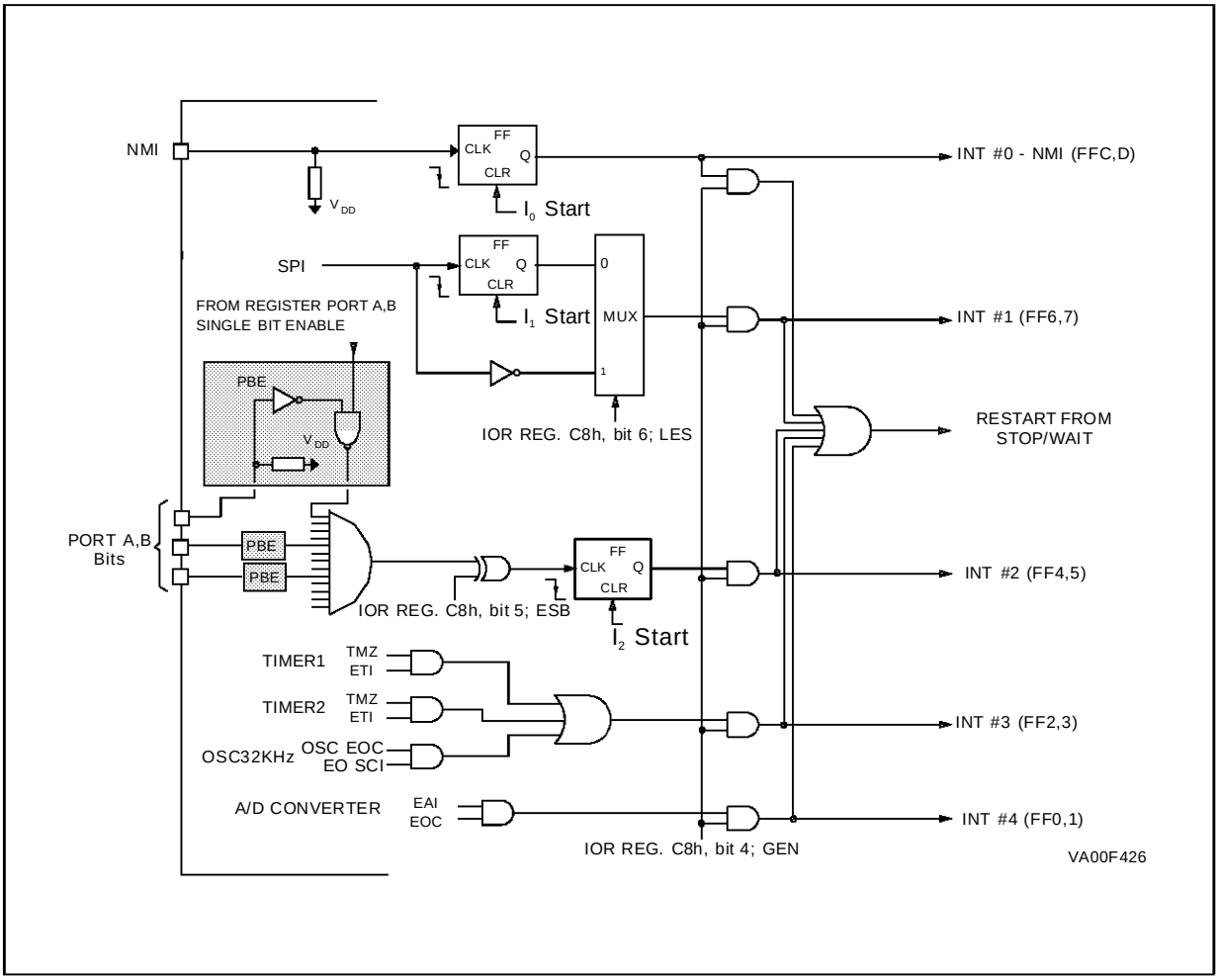
32kHz Oscillator, 320CR register location DBh

– EOSCI. This bit, when set, enables the 32kHz oscillator interrupt request.

– OSCEOC. This read only bit indicates when the 32kHz oscillator has measured a 500ms elapsed time (providing a 32.768kHz quartz crystal is connected to the 32kHz oscillator dedicated pins). An interrupt request can be generated in relation to the state of EOSCI bit.

INTERRUPTS (Continued)

Figure 14. ST6245 Interrupt Circuit Diagram



RESET

The ST6245 can be reset in three ways: by the external reset input ($\overline{\text{RESET}}$) tied low, by power-on reset and by the digital watchdog/timer peripheral.

RESET Input

The $\overline{\text{RESET}}$ pin can be connected to a device of the application board in order to restart the MCU during its operation. The activation of the $\overline{\text{RESET}}$ pin may occur in the RUN, WAIT or STOP mode. This input has to be used to reset the MCU internal state and provide a correct start-up procedure. The pin is active low. The internal reset signal is generated by adding a delay to the external signal. Therefore even short pulses at the $\overline{\text{RESET}}$ pin will be accepted. This feature is valid providing that V_{DD} has finished its rising phase and the oscillator is correctly running (normal RUN or WAIT modes).

If $\overline{\text{RESET}}$ activation occurs in the RUN or Wait mode, the MCU is configured in the Reset mode for as long as the signal of the $\overline{\text{RESET}}$ pin is low. The processing of the program is stopped (in RUN mode only) and the Input/Outputs are in the High-impedance state with pull-up resistors switched on. As soon as the level on the $\overline{\text{RESET}}$ pin becomes high, the initialization sequence is executed.

If a $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts and all the inputs/outputs are configured in the High-impedance state with pull-up resistors switched on for as long as the level on the $\overline{\text{RESET}}$ pin remains low. When the level of the $\overline{\text{RESET}}$ pin becomes high, a delay is generated by the ST62xx core to wait that the oscillator becomes completely stabilized. Then, the initialization sequence is started.

Power-On Reset (POR)

The function of the POR consists in waking up the MCU during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: every Input/Output port is configured in the input mode (High-impedance state with pull-up) and no instruction is executed. When the power supply voltage becomes sufficient, the oscillator starts to operate, nevertheless the ST62xx core generates a delay to allow the oscillator to be completely stabilized before the execution of the first instruction. The initialization sequence is then executed.

Internal circuitry generates a Reset pulse when V_{DD} is switched on. In the case of fast rising V_{DD} (transition time $\leq 100\mu s$), this reset pulse starts the internal reset procedure without the need of external components at the $\overline{\text{RESET}}$ pin. In cases of slowly or non monotonously rising V_{DD}, an external reset signal must be provided for a proper reset of the MCU.

For as long as the reset pin is kept at the low level, the processor remains in the reset state. The reset will be released after the voltage at the reset pin reaches the high level.

Note:

To have a correct ST62xx start-up, the user should take care that the reset input does not change to the high level before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see recommended operating conditions).

An on-chip counter circuit provides a delay of 2048 oscillator cycles between the detection of the reset high level and the release of the MCU reset.

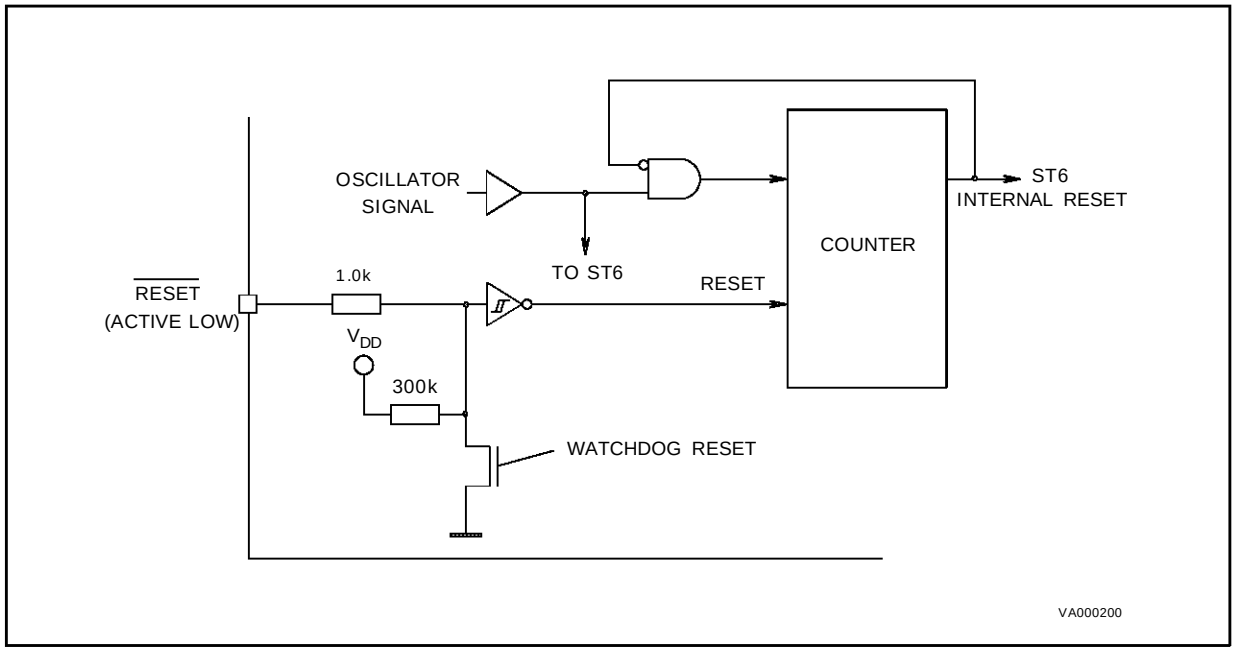
A proper reset signal for slow rising V_{DD}, i.e. the required delay between reaching sufficient operating voltage and the reset input changing to a high level, can be generally provided by an external capacitor connected between the $\overline{\text{RESET}}$ pin and V_{SS}.

RESET (Continued)

Watchdog Reset

The ST6245 provides an on-chip watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed, preventing the end-of-count being reached, an internal circuit pulls down the $\overline{\text{RESET}}$ pin. The MCU will enter the reset state as soon as the voltage at $\overline{\text{RESET}}$ pin reaches the related low level. This also resets the watchdog which subsequently turns off the pull-down and activates the pull-up device at the $\overline{\text{RESET}}$ pin. This causes the positive transition at the $\overline{\text{RESET}}$ pin and terminates the reset state.

Figure 15. Reset Circuit

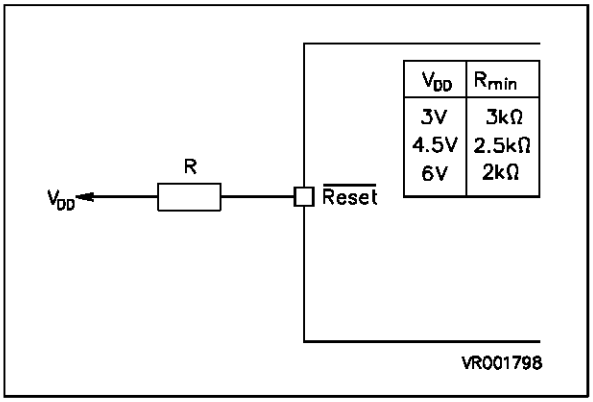


Application Notes

An external resistor between V_{DD} and reset pin is not required because an internal pull-up device is provided. If the user prefers, for any reason, to add an external pull-up resistor its value must comply with the Rmin value defined in Figure 16. If the value is lower than Rmin, the on-chip watchdog pull-down transistor might not be able to pull-down the reset pin resulting in an external deactivation of the watchdog function.

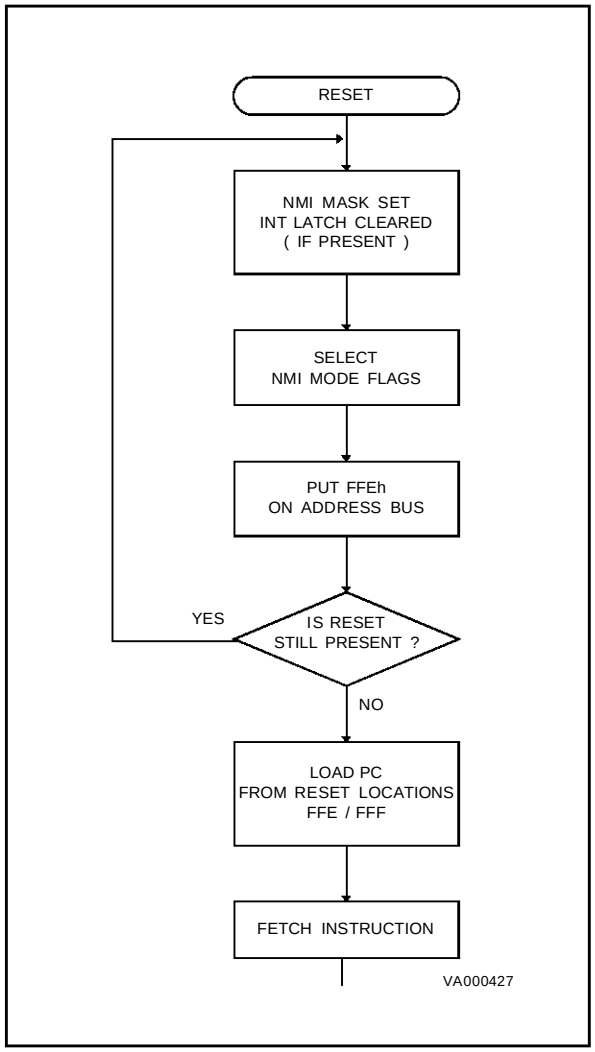
The POR function operates in a dynamic manner in the way that it brings about the initialization of the MCU when it detects a dynamic rising edge of the V_{DD} voltage. The typical detected threshold is about 2 volts, but the actual value of the detected threshold depends on the way in which the V_{DD} voltage rises up. The POR device DOES NOT allow the supervision of a static rising or falling edge of the V_{DD} voltage.

Figure 16. External Reset Resistance



RESET (Continued)

Figure 17. Reset & Interrupt Processing Flow-Chart



MCU Initialization Sequence

When a reset occurs the stack is reset to the program counter, the PC is loaded with the address of the reset vector (located in the program ROM at addresses FFEh & FFFh). A jump instruction to the beginning of the program has to be written into these locations. After a reset the interrupt mask is automatically activated so that the core is in non-maskable interrupt mode to prevent false or ghost interrupts during the restart phase. Therefore the restart routine should be terminated by a RETI instruction to switch to normal mode and enable interrupts. If no pending interrupt is present at the end of the reset routine the ST62xx will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced.

Figure 18. Restart Initialization Program Flow-Chart

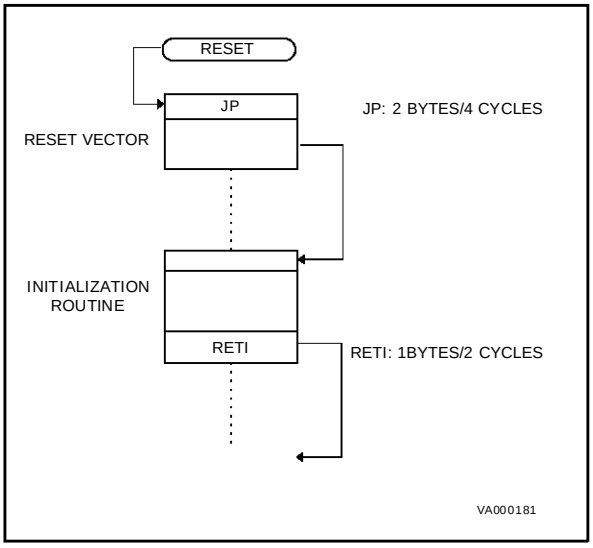


Table 6. Reset Configuration

Input/Outpu t pins	Registers
Input Mode with pull-up and no interrupt	All cleared but A,X,Y,V,W, data RAM, LCD RAM, DWR (C9), PRPR (CA), DRBR (CB). Timers prescaler and TCR are initialized respectively at 7F and FF. Watchdog register DWDR (D8) is set to FEh.

WAIT & STOP MODES

The WAIT and STOP modes have been implemented in the ST62xx core in order to reduce the consumption of the product when the latter has no instruction to execute. These two modes are described in the following paragraphs

WAIT Mode

The configuration of the MCU in the WAIT mode occurs as soon as the WAIT instruction is executed. The microcontroller can also be considered as being in a “software frozen” state where the core stops processing the instructions of the routine, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage but where the peripherals are still working.

The WAIT mode is used when the user wants to reduce the consumption of the MCU when it is in idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide a clock signal to the peripherals. The timer counting may be enabled (writing the PSI bit in TSCR register) and the timer interrupt may be also enabled before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. The above explanation related to the timers applies also to the A/D converter.

If the exit from the WAIT mode is performed with a general RESET (either from the activation of the external pin or by watchdog reset) the MCU will enter a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behavior depends on the state of the ST62xx core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case will be described in the following paragraphs. In any case, the ST62xx core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

If the Watchdog is disabled the STOP mode is available. When in STOP mode the MCU is placed in the lowest power consumption mode. In this operating mode the microcontroller can be considered as being “frozen”, no instruction is executed, the oscillator is stopped, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or Reset activation to output from the STOP state.

If the exit from the STOP mode is performed with a general RESET (by the activation of the external pin) the MCU will enter a normal reset procedure as described in the RESET chapter. The case of an interrupt depends on the state of the ST62xx core before the initialization of the STOP sequence and also of the kind of the interrupt request that is generated.

This case will be described in the following paragraphs. In any case, the ST62xx core generates a delay after the occurrence of the interrupt request in order to wait the complete stabilization of the oscillator before the execution of the first instruction.

Exit from WAIT and STOP Modes

The following paragraphs describe the output procedure of the ST62xx core from WAIT and STOP modes when an interrupt occurs (not a RESET). It must be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) before the start of the WAIT or STOP sequence, but also of the type of the interrupt request that is generated.

Normal Mode. If the ST62xx core was in the main routine when the WAIT or STOP instruction has been executed, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs; the related interrupt routine is executed and at the end of the interrupt service routine the instruction that follows the STOP or the WAIT instruction is executed if no other interrupts are pending.

WAIT & STOP MODES (Continued)

Not Maskable Interrupt Mode. If the STOP or WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs: the instruction that follows the STOP or the WAIT instruction is executed and the ST62xx core is still in the non-maskable interrupt mode even if another interrupt has been generated.

Normal Interrupt Mode. If the ST62xx core was in the interrupt mode before the initialization of the STOP or WAIT sequence, it outputs from the stop or wait mode as soon as any interrupt occurs. Nevertheless, two cases have to be considered:

– If the interrupt is a normal interrupt, the interrupt routine in which the WAIT or STOP was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST62xx core is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance to their priority.

– If the interrupt is a non-maskable interrupt, the non-maskable routine is processed at first. Then the routine in which the WAIT or STOP was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST62xx core remains in the normal interrupt mode.

Notes :

To reach the lowest power consumption the user software must take care of:

– placing the A/D converter in its power down mode by clearing the PDS bit in the A/D control register before entering the STOP instruction.

– switching off the 32kHz oscillator by clearing the oscillator start/stop bit in the 32kHz oscillator control register.

– putting the EEPROM on-chip memory in stand-by mode by setting the E2OFF bit in EEPROM Control Register to one.

The LCD Driver peripheral is automatically switched-off by the STOP instruction when the 32kHz oscillator operation is not selected.

When the hardware activated watchdog is selected or the software watchdog enabled, the STOP instruction is deactivated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

If all the interrupt sources are disabled (including NMI if GEN="0"), the restart of the MCU can only be done by a $\overline{\text{RESET}}$ activation. The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

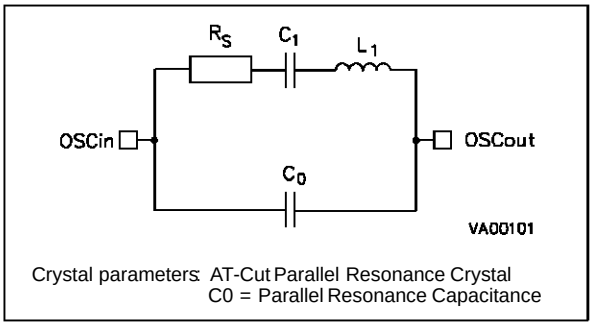
ON-CHIP CLOCK OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. A crystal, a ceramic resonator, or an external signal (provided to the OSCin pin) may be used to generate a system clock with various stability/cost tradeoffs. The different clock generator options connection methods are shown in Figure 20.

One machine cycle takes 13 oscillator pulses; 12 clock pulses are needed to increment the PC while and additional 13th pulse is needed to stabilize the internal latches during memory addressing. This means that with a clock frequency of 8MHz the machine cycle is 1.625µs.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance (CL), IC parameters, ambient temperature, supply voltage. It must be observed that the crystal or ceramic leads and circuit connections must be as short as possible. Typical values for CL1, CL2 are 15-22pF for a 4/8MHz crystal. The oscillator output frequency is internally divided by 13 to produce the machine cycle and by 12 to produce the Timer, the Watchdog and the A/D peripheral clock. A machine cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five machine cycles to be executed.

Figure 19. Crystal Parameters



Crystal parameters: AT-Cut Parallel Resonance Crystal
C0 = Parallel Resonance Capacitance

ON-CHIP CLOCK OSCILLATOR (Continued)

Figure 20. Oscillator Connection

CL1 = CL2 = 12 to 22pF for a 4/8MHz crystal (typical)

INPUT/OUTPUT PORTS

The ST6245 microcontroller has 11 Input/Output lines that can be individually programmed either in the input mode or the output mode with the following options that can be selected by software:

– Input without pull-up and without interrupt
– Input with pull-up and with interrupt
– Input with pull-up without interrupt
– Analog inputs (PA5-PA7, PB0-PB3)
– SPI control signals (PB5-PB7)
– Push-pull output
– Standard Open drain output
– 20mA Open drain output (PB4-PB7)

The lines are organized in two ports (port A,B).

Each port occupies 3 registers in the data space. Each bit of these registers is associated with a particular line (for instance, the bits 0 of the Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The two DATA registers (DRA, DRB), are used to read the voltage level values of the lines programmed in the input mode, or to write the logic value of the signal to be output on the lines configured in the output mode. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by the user software, in conjunction with the related option registers, to select the different input mode options.

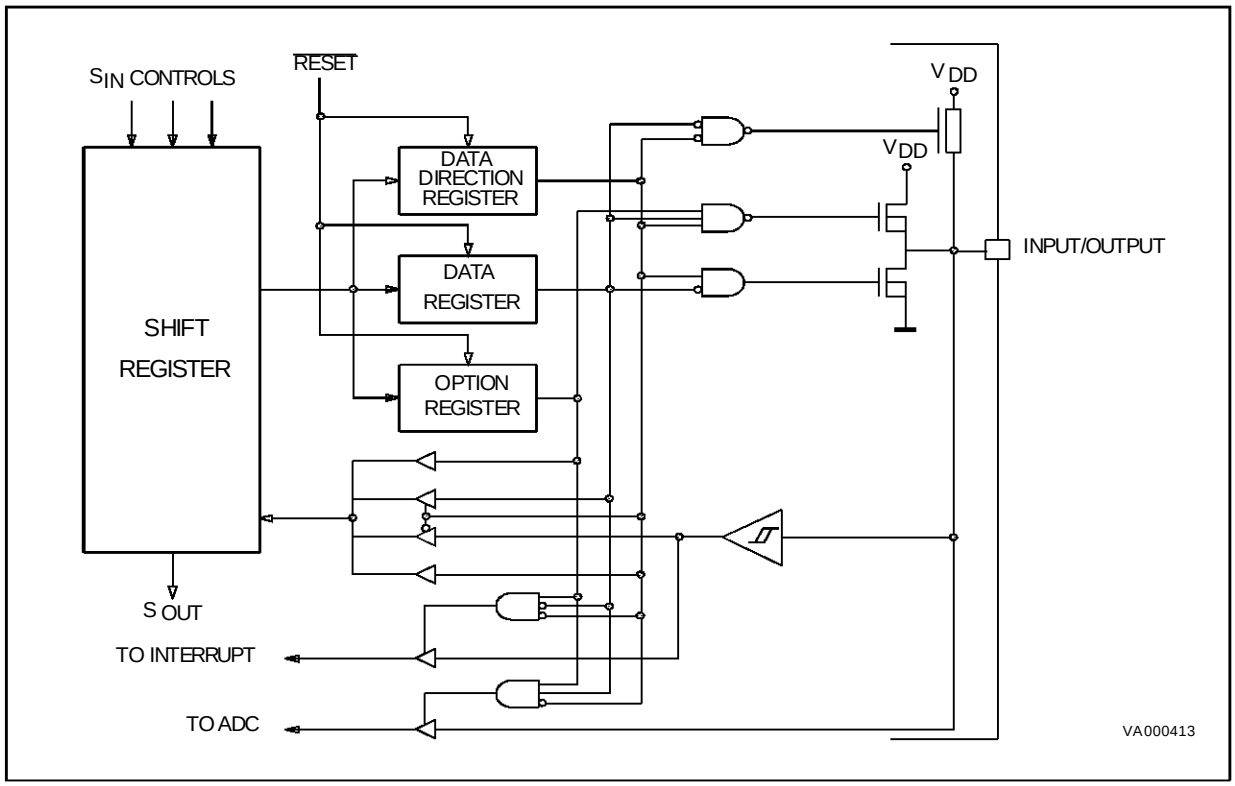
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The two Data Direction registers (DDRA, DDRB) allow the selection of the data direction of each pin (input or output).

The two Option registers (ORPA, ORPB) are used to select the different port options available both in input and in output mode.

All the I/O registers can be read or written as any other RAM location of the data space, so no extra RAM cell is needed for port data storing and manipulation. During the initialization of the MCU, all the I/O registers are cleared and the input mode with pull-up/no-interrupt is selected on all the pins, thus avoiding pin conflicts.

Figure 21. I/O Port Block Diagram



INPUT/OUTPUT PORTS (Continued)

I/O Pin Programming

Each pin can be individually programmed as input or output with different input and output configurations.

This is achieved by writing to the relevant bit in the data (DR), data direction register (DDR) and option registers (OR). Table 7 shows all the port configurations that can be selected by user software.

Input Option Description

Pull-up, High Impedance Option. All the input lines can be individually programmed with or without an internal pull-up according to the codes programmed in the OR and DR registers . If the pull-up option is not selected, the input pin is in the high-impedance state.

Interrupt Option. All the input lines can be individually connected by software to the interrupt lines of the ST62xx core according to the codes programmed in the OR and DR registers. The pins of Port A and B are "ORed" and are connected to the interrupt associated to the vector #2. The interrupt modes (falling edge sensitive, rising edge sensitive) can be selected by software for each port by programming the IOR register.

Analog Input Option. The seven PA5-PA7, PB0-PB3 pins can be configured to be analog inputs according to the codes programmed in the OR and DR registers. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as analog input at a time, otherwise the selected inputs will be shorted.

Figure 22. I/O Port Data Registers

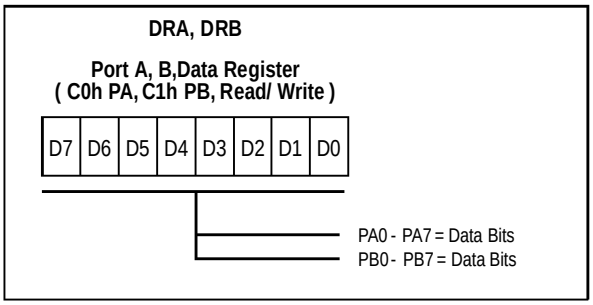


Figure 23. I/O Port Data Direction Registers

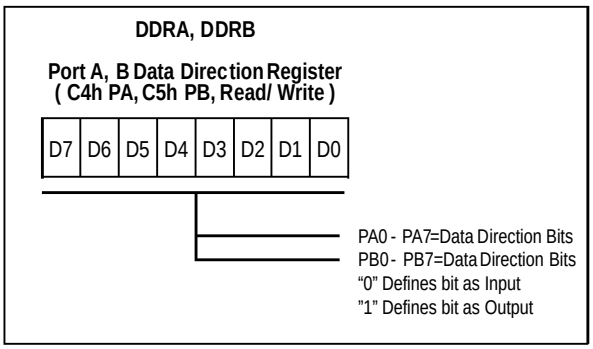
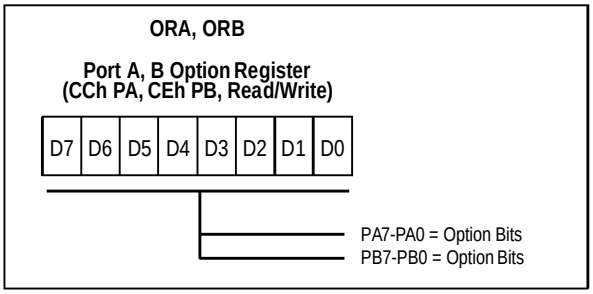


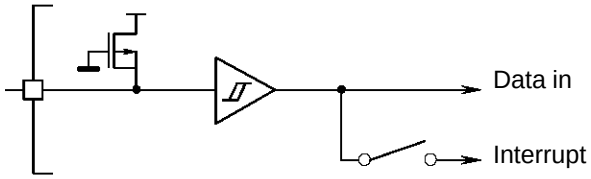
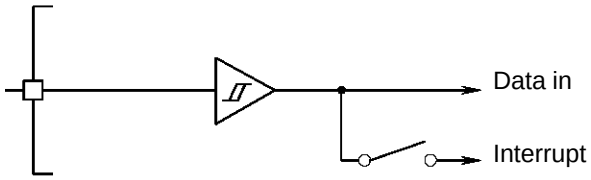
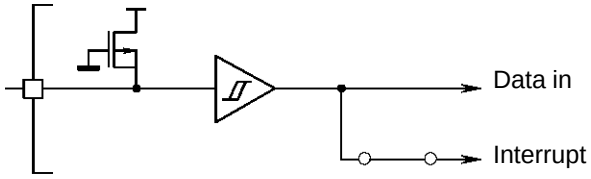
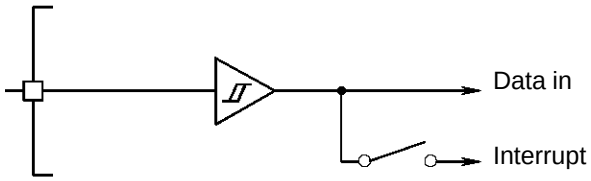
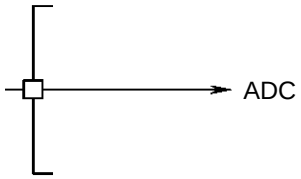
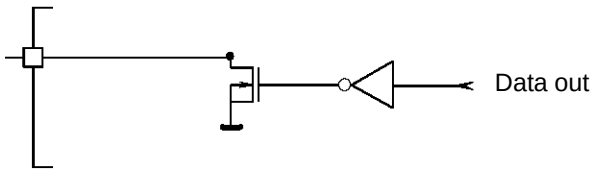
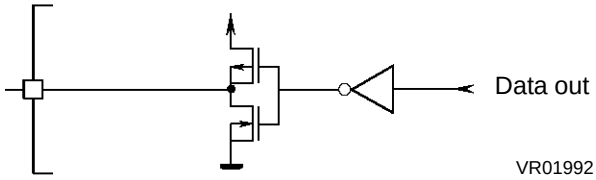
Figure 24. I/O Port Option Registers



Note: For complete coding explanation refer to Table 7.

INPUT/OUTPUT PORTS (Continued)

Table 7. I/O Port Configuration

DDR, OR	DR	Mode	Option	Schematic	
0	0	0	Input	Pull-up No interrupt (RESET state)	
0	0	1	Input	No pull-up No interrupt	
0	1	0	Input	Pull-up Interrupt	
0	1	1	Input	No pull-up No interrupt PB4-PB7	
			Input	Analog input: PA0-PA7 PB2-PB3	
1	0	X	Output	Open drain: 20mA PB4-PB7 Open drain: 5mA PB2-PB3, PA5-PA7	
1	1	X	Output	Push-pull	

INPUT/OUTPUT PORTS (Continued)

SPI alternate function Option. The I/O pins PB5-PB7 are also used by serial peripheral interface SPI. PB5 is connected with the SPI clock input SCL, PB6 is connected with the SPI data input SIN and PB7 is connected with the SPI data output SOUT.

For serial input operation PB5 and PB6 have to be programmed as inputs. For serial output operation PB7 has to be programmed as open-drain output (DDR = "1", OPR = "0"). In this operating mode the output of the SPI shift register instead of the port data register is connected to the port buffer. When PB7 is programmed as push-pull output (DDR = "1", OPR = "1"), the port data register is connected to the port buffer. When the SPI peripheral is not used PB5-PB7 can be used as general purpose I/O lines (provided that PB7 is not selected to be open-drain in output mode).

Notes:

Switching the I/O ports from one state to another should be done in a way that no unwanted side effects can happen. The recommended safe transitions are shown below. All other transitions are risky and should be avoided during change of operation mode as it is most likely that there will be an unwanted side-effect such as interrupt generation or two pins shorted together by the analog input lines.

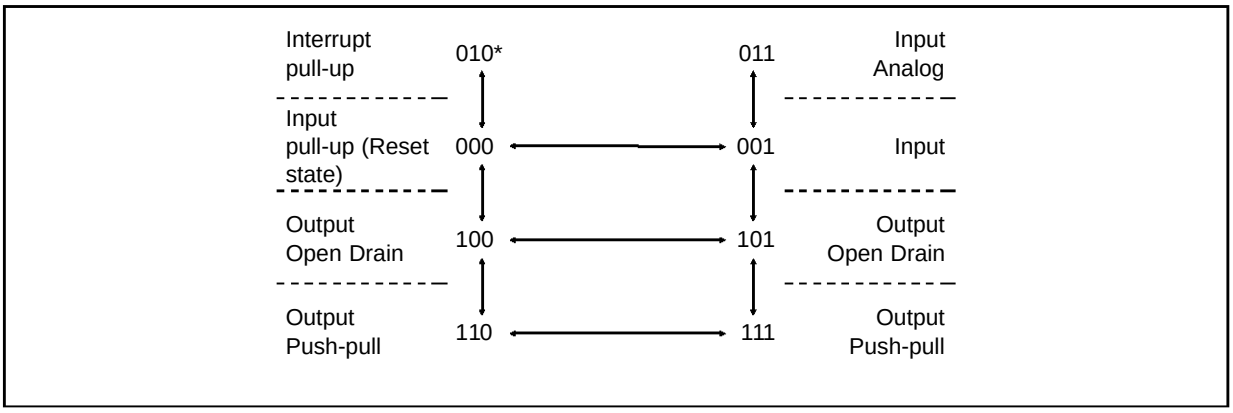
Single bit SET and RES instructions should be used very carefully with Port A and B data registers because these instructions make an implicit read and write back of the whole addressed register byte. In port input mode however data register address reads from input pins, not from data register latches and data register information in input mode is used to set characteristics of the input pin (interrupt, pull-up, analog input), therefore these characteristics may be unintentionally reprogrammed depending on the state of input pins. As general rule is better to use SET and RES instructions on data register only when the whole port is in output mode. If input or mixed configuration is needed it is recommended to keep a copy of the data register in RAM. On this copy it is possible to use single bit instructions, then the copy register could be written into the port data register.

```
SET     bit, datacopy
LD      a, datacopy
LD      DRA, a
```

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user has to take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance in the measurement.

Figure 25. I/O Port StateTransition Diagram for Safe Transitions



Note *. xxx = DDR, OR, DR Bits respectively

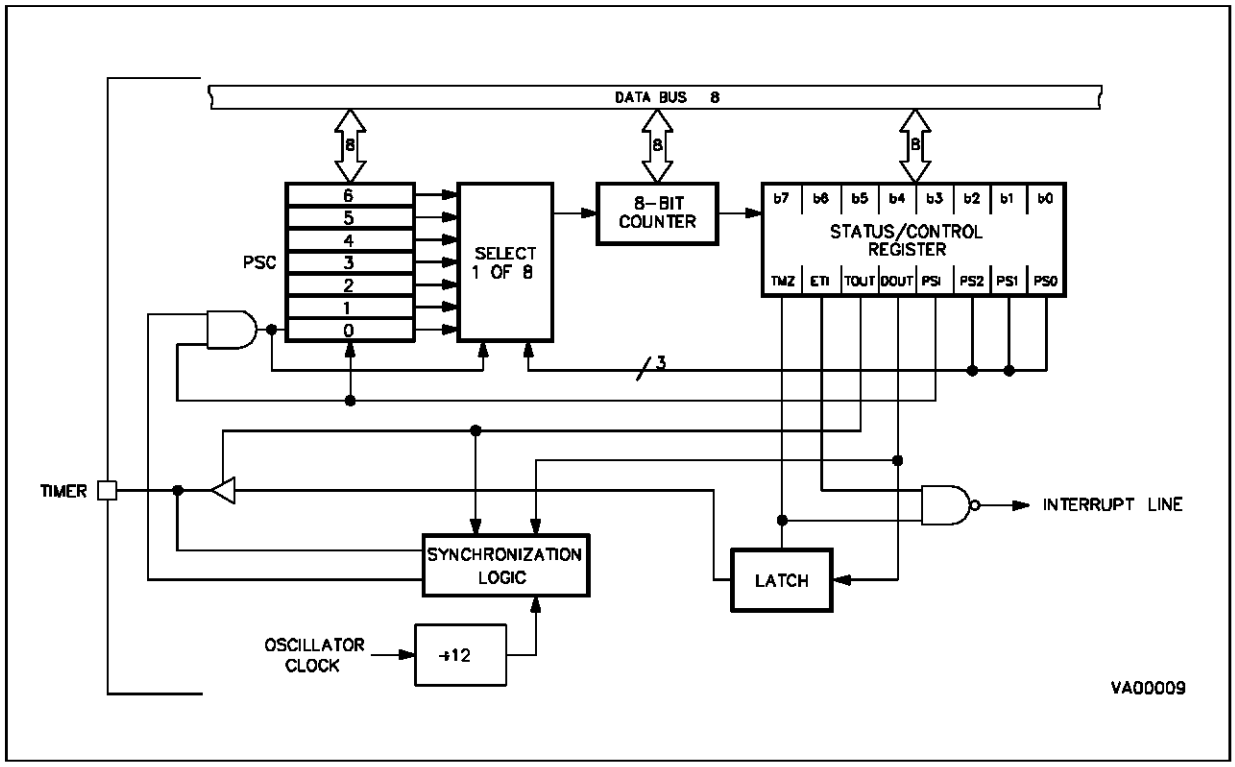
TIMERS

The ST6245 offers two on-chip Timer peripherals named Timer 1 and Timer 2. Each of these timers consists of an 8-bit counter with a 7-bit programmable prescaler, thus giving a maximum count of 2^{15}, and control logic that allows configuring the peripheral in three operating modes. Figure 26 shows the Timer block diagram. Timer 1 only has the external TIMER pin available for the user. The content of the 8-bit counter can be read/written in the Timer/Counter register TCR which is addressed in the data space as a RAM location at addresses D3h or D6h. The state of the 7-bit prescaler can be read in the PSC register at addresses D2h or D5h. The control logic device is managed in the TSCR register (addresses D4h or D7h) as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero) bit in the TSCR is set to one. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set to one an interrupt request, associated to interrupt vector #3, is generated. The interrupt service routine then would determine which timer reached the end of count by poling the TMZ bits. The Timer interrupt can be used to exit the MCU from the WAIT mode.

The prescaler input can be the oscillator frequency divided by 12 or an external clock at TIMER pin (only Timer 1). The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR, the clock input of the timer/counter register is multiplexed to different sources. On division factor 1, the clock input of the prescaler is also that of timer/counter; on factor 2, bit 0 of prescaler register is connected to the clock input of TCR. This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC is connected to clock input of TCR, and so on. The prescaler initialize bit (PSI) in the TSCR register must be set to one to allow the prescaler (and hence the counter) to start. If it is cleared to zero then all of the prescaler bits are set to one and the counter is inhibited from counting. The prescaler can be given any value between 0 and 7Fh by writing to addresses D2h or D5h, if bit PSI in the TSCR register is set to one. The tap of the prescaler is selected using the PS2,PS1,PS0 bits in the control register. Figure 27 shows the Timer working principle.

Figure 26. Timer Peripheral Block Diagram



TIMERS (Continued)

Timer Operating Modes

There are 3 operating modes of the Timer peripheral. They are selected by the bits TOUT and DOUT (see TSCR register). These three modes correspond to the two clock frequencies that can be connected on the 7-bit prescaler ($f_{OSC}/12$ or TIMER pin signal) and to the output mode. For this reason, only Timer 1 has all three modes, while Timer 2, which does not have a dedicated TIMER pin, must be programmed in Output Mode only.

Clock Input Mode (TOUT = "0", DOUT = "0"). In this mode the TIMER pin is an input and the prescaler is decremented on rising edge. The maximum input frequency that can be applied to the external pin in this mode is 1/8 of the oscillator frequency. This operating mode is not available on Timer 2.

Gated Mode (TOUT = "0", DOUT = "1"). In this mode the prescaler is decremented by the Timer clock input (oscillator divided by 12) but ONLY when the signal at TIMER pin is held high (giving a pulse width measurement potential). This mode is selected by the TOUT bit in TSCR register cleared to "0" (i.e. as input) and DOUT bit set to "1". This operating mode is not available on Timer 2.

Output Mode (TOUT = "1", DOUT = data out). The TIMER pin is connected to the DOUT latch. Therefore the timer prescaler is clocked by the prescaler clock input ($f_{OSC}/12$).

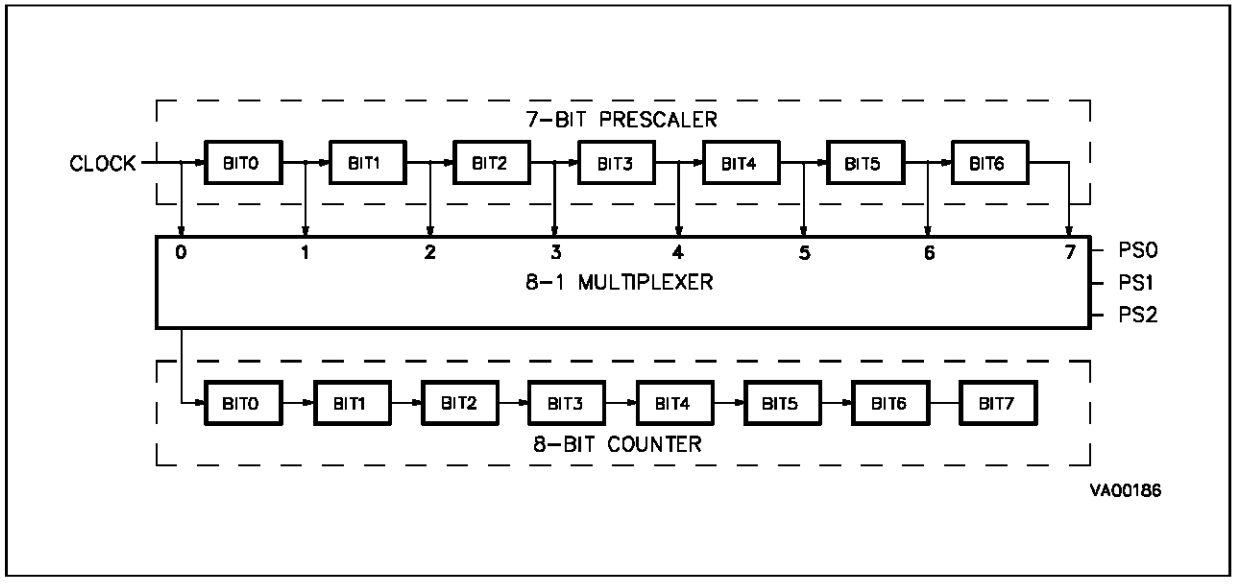
The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The low-to-high TMZ bit transition is used to latch the DOUT bit of the TSCR and pass it to TIMER pin. This operating mode allows external signal generation on the TIMER pin. This is the only operating Mode allowed on Timer 2.

Table 8. Timer Operating Modes

TOUT	DOUT	Timer Pin	Timer Function
0	0	Input	Event Counter (1)
0	1	Input	Input Gated (1)
1	0	Output	Output
1	1	Output	Output

Note 1. Not allowed on Timer 2

Figure 27. Timer Working Principle



TIMERS (Continued)

Timer Interrupt

When the counter register decrements to zero and the software controlled ETI (Enable Timer Interrupt) bit is set to one then an interrupt request associated to interrupt vector #3 is generated. When the counter decrements to zero also the TMZ bit in the TSCR register is set to one.

Since only one interrupt vector is available for the two timers (ORed also with 32kHz oscillator interrupt), the interrupt service routine should determine from which source the interrupt came by polling the TMZ bits (and the OSCEOC bit of the 32kHz oscillator control register).

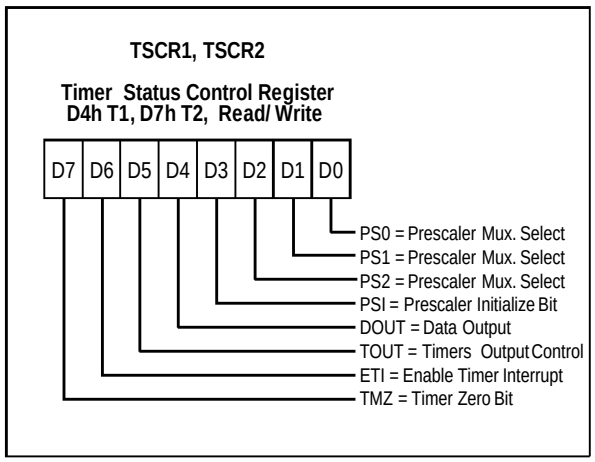
Notes:

TMZ is set when the counter reaches 00h; however, it may be set by writing 00h in the TCR register or setting bit 7 of the TSCR register. TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFh while the 7-bit prescaler is loaded to 7Fh , and the TSCR register is cleared which means that timer is stopped (PSI="0") and the timer interrupt is disabled.

If the Timer is programmed in output mode, DOUT bit is transferred to the TIMER pin when TMZ is set to one (by software or due to counter decrement). When TMZ is high, the latch is transparent and DOUT is copied to the timer pin. When TMZ goes low, DOUT is latched.

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

Figure 28. Timer Status Control Register



TMZ. Low-to-high transition indicates that the timer count register has decremented to zero. This bit must be cleared by user software before starting with a new count.

ETI. This bit, when set, enables the timer interrupt request (vector #3). If ETI="0" the timer interrupt is disabled. If ETI="1" and TMZ="1" an interrupt request is generated.

TOUT. When low, this bit selects the input mode for the TIMER pin. When high the output mode is selected.

DOUT. Data sent to the timer output when TMZ is set high (output mode only). Input mode selection (input mode only). This bit is meaningless for Timer 2.

PSI. Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

PS2, PS1, PS0. These bits select the division ratio of the prescaler register.

Table 9. Prescaler Division Factors

PS2	PS1	PS0	Divided by	PS2	PS1	PS0	Divided by
0	0	0	1	1	0	0	16
0	0	1	2	1	0	1	32
0	1	0	4	1	1	0	64
0	1	1	8	1	1	1	128

Figure 29. Timer Counter Register

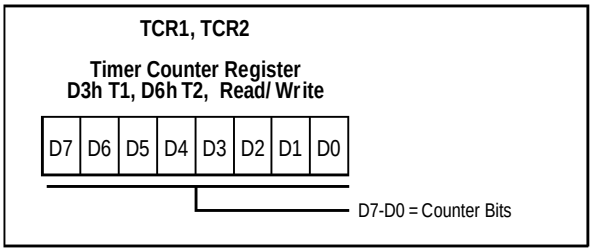
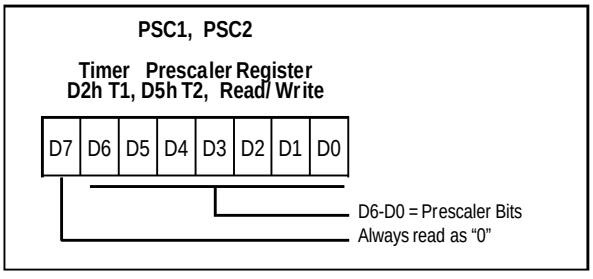


Figure 30. Prescaler Register



DIGITAL WATCHDOG

The digital Watchdog of the ST62 device consists of a down counter that can be used to provide a controlled recovery from a software upset.

The ST6245 watchdog is a the software activated watchdog.

The software activated digital watchdog consists of a down counter that can be used to provide a controlled recovery from a software upset. The watchdog uses one data space register (DWDR location D8h). The watchdog register is set to FEh after reset and the watchdog function is disabled. The watchdog time can be programmed using the 6 Most Significant Bits in the Watchdog register. The check time can be set differently for different routines within the general program.

After a reset the software Watchdog is in the off-state. The watchdog should be activated inside the Reset restart routine by writing a "1" in watchdog timer register bit 0. Bit one of this register must be set to one before programming bit zero as otherwise a reset will be immediately generated when bit 0 is set. This allows the user to generate a reset by software (bit 0 = "1", bit 1 = "0"). Once bit 0 is set, it can not be cleared by software without generating a Reset. The delay time is defined by programming bits 2-7 of the watchdog register. Bit 7 is the Least Significant Bit while bit 2 is the MSB.

This gives the possibility to generate a reset in a time between 3072 to 196608 clock cycles in 64 possible steps: (With a clock frequency of 8MHz this means from 384µs to 24.576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones. If the watchdog is active the STOP instruction is deactivated and a WAIT instruction is automatically executed instead of a STOP. If bit 0 of the watchdog register is never set to one then bits 1-7 of the register can be used as a simple 7-bit counter which is decrement every 3072 clock cycles.

Figure 32. Watchdog Register

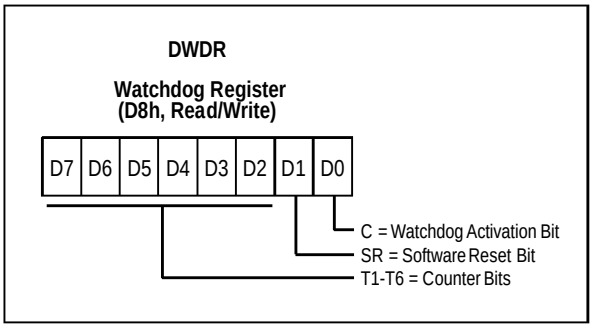
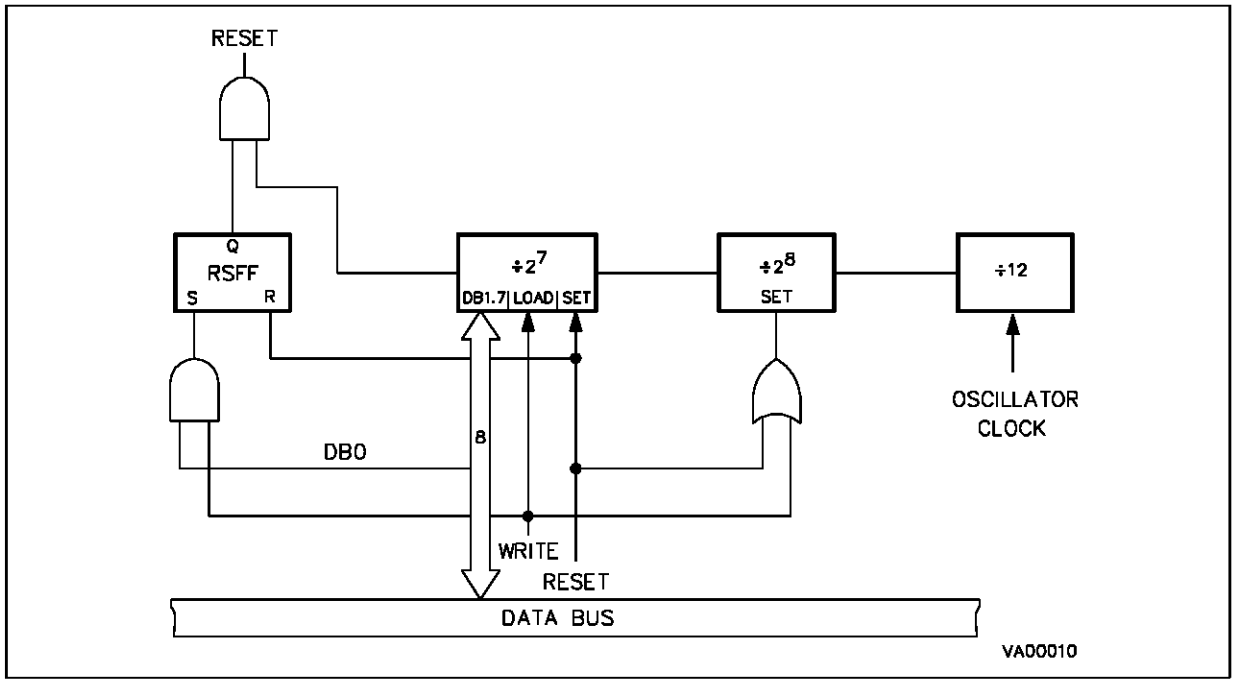


Figure 31. Digital Watchdog Block Diagram



DIGITAL WATCHDOG (Continued)

Watchdog Register

C. This is the watchdog activation bit, that, if set to one, will activate the watchdog function. When cleared to zero it allows the use of the counter as a 7-bit timer. This bit is cleared on reset.

SR. This bit is set to one during the reset and will generate a software reset if cleared to zero. When C = "0" (watchdog disabled) it is the MSB of the 7-bit timer.

T1-T6. These are the watchdog counter bits. It should be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter. These bits are in the opposite order to normal.

Application note:

If the Watchdog is not used during power-on reset external noise may cause the undesired activation of the Watchdog with a generation of an unexpected reset. To avoid this risk, two additional instructions, that check the state of the watchdog and eventually reset the chip are needed within the first 27 instructions, after the reset. These instructions are:

```
jrx 0, WD, #+3
ldi WD, 0FDH
```

These instructions should be executed at the very beginning of the customer program.

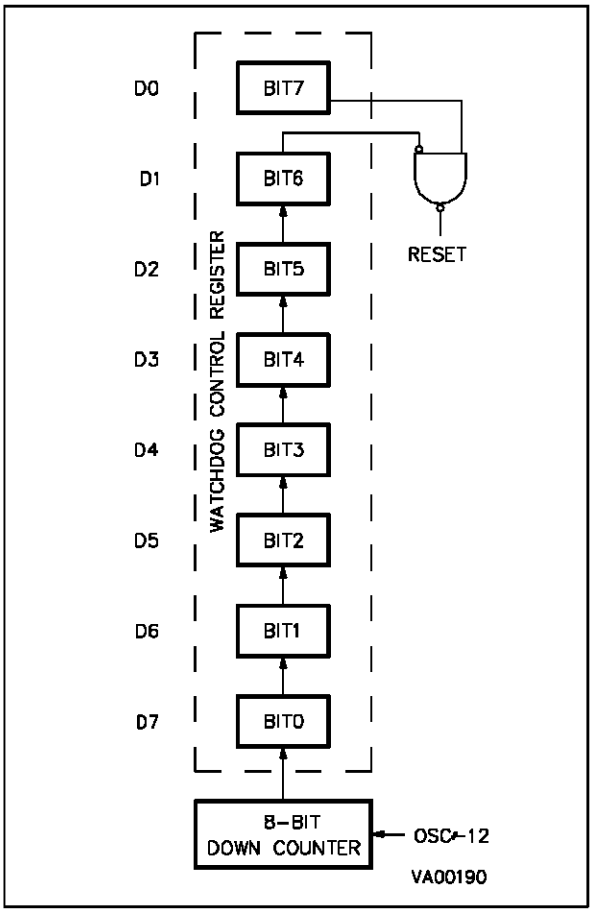
If the Watchdog is used during power-on reset the Watchdog register may be set to a low value, that could give a reset after 28 instructions earliest. To avoid undesired resets, the Watchdog must be set to the desired value within the first 27 instructions, the best is to put at the very beginning.

Alternatively the normal legal state can be checked with the following short routine:

```
ldi a, 0FEH
and a, WD
cpi a, 0FEH
jrz #+3
ldi WD, 0FDH
```

This sequence is recommended for security applications, where possible stack confusion error loops must be avoided and the Watchdog must only be refreshed after extensive checks.

Figure 33. Watchdog Working Principle



8-BIT A/D CONVERTER

The A/D converter of ST6245 is an 8-bit analog to digital converter with up to 7 analog inputs (as alternate functions of I/O lines PA5-PA7, PB0-PB3) offering 8-bit resolution with total accuracy ±2 LSB and a typical conversion time of 70µs (clock frequency of 8MHz).

The A/D peripheral converts the input voltage by a process of successive approximations using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, the A/D converter accuracy is decreased.

The selection of the pin signal that has to be converted is done by configuring the related I/O line as analog input through the I/O ports option and data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as analog input at a time. The user must avoid the situation in which more than one I/O pin is selected to be analog input to avoid malfunction of the ST62xx.

The ADC uses two registers in the data space: the ADC data conversion register which stores the conversion result and the ADC control register used to program the ADC functions.

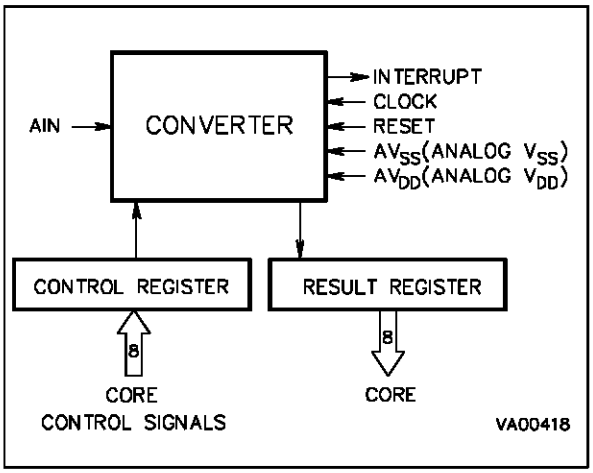
A conversion is started by writing a "1" to the Start bit (STA) in the ADC control register. This automatically clears (resets to "0") the End Of Conversion Bit (EOC). When a conversion has been finished this EOC bit is automatically set to "1" in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

The STA bit is continually being scanned so that if the user sets it to "1" while a previous conversion is in progress then a new conversion is started before the previous one has been completed.The start bit (STA) is a write only bit, any attempt to read it will show a logical "0".

The A/D converter has a maskable interrupt associated to the end of conversion. This interrupt is associated to the interrupt vector #4 and occurs when the EOC bit is set, i.e. when a conversion is completed. The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. That is achieved when the PDS bit in the ADC control register is cleared to "0". If PDS="1", the A/D is supplied and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow the stabilization of the A/D converter.*This action is needed also before entering the STOP instruction as the A/D comparator is not automatically disabled by the STOP mode*

Figure 34. A/D Converter Block Diagram



During reset any conversion in progress is stopped, the control register is reset to all zeros and the A/D interrupt is masked (EAI=0).

Notes:

The ST62xx A/D converter does not feature a sample and hold. The analog voltage to be measured should therefore be stable during the conversion time. Variation should not exceed ±1/2 LSB for the best accuracy in measurement.

Since the ADC is on the same chip as the microprocessor the user should not switch heavily loaded output signals during conversion if high precision is needed. This is because such switching will affect the supply voltages which are used for comparisons.

A low pass filter can be used at the analog input pins to reduce input voltage variation during the conversion. For true 8 bit conversions the impedance of the analog voltage sources should be less than 30kΩ while the impedance of the reference voltage should not exceed 2kΩ.

The accuracy of the conversion depends on the quality of the power supply voltages (V_{DD} and V_{SS}). The user must specially take care of applying regulated reference voltage on the V_{DD} and V_{SS} pins (the variation of the power supply voltage must be inferior to 5V/ms).

The converter can resolve the input voltage with a resolution of:

$$\frac{V_{DD} - V_{SS}}{256}$$

8-BIT A/D CONVERTER(Continued)

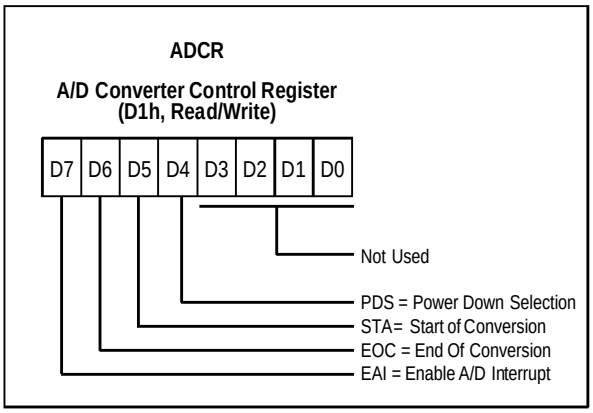
The Input voltage (Ain) which has to be converted must be constant for 1µs before conversion and remain constant during the conversion.

The resolution of the conversion can be improved if the power supply voltage (V_{DD}) of the microcontroller becomes lower.

In order to optimize the resolution of the conversion, the user can configure the microcontroller in the WAIT mode because this mode allows the minimization of the noise disturbances and the variations of the power supply voltages due to the switching of the outputs. Nevertheless, it must be take care of executing the WAIT instruction as soon as possible after the beginning of the conversion because the execution of the WAIT instruction may provide a small variation of the V_{DD} voltage (the negative effect of this variation is minimized at the beginning of the conversion because the latter is less sensitive than the end of the conversion when the less significant bits are determined).

The best configuration from a accuracy point of view is the WAIT mode with the Timer and LCD driver stopped. Indeed, only the ADC peripheral and the oscillator are still working. The MCU has to be wake-up from the WAIT mode by the interrupt of the ADC peripheral at the end of the conversion. It must be noticed that the wake-up of the microcontroller could be done also with the interrupt of the TIMER, but in this case, the Timer is working and some noise could disturb the converter in terms of accuracy.

Figure 35. A/D Converter Control Register



EAI. If this bit is set to one the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

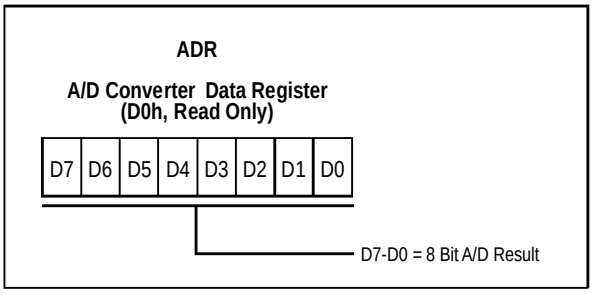
EOC. *Read Only*;This read only bit indicates when a conversion has been completed. This bit is automatically reset to zero when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to one.

STA. *Write Only*; Writing a "1" in this bit will start a conversion on the selected channel and automatically reset to zero the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

PDS. This bit activates the A/D converter if set to 1. Writing a zero into this bit will put the ADC in power down mode (idle mode).

D3-D0. Not used

Figure 36. A/D Converter Data Register



D7-D0. *Read Only*;These are the conversion result bits; the register is read only and stores the result of the last conversion. The contents of this register are valid only when EOC bit in the ADCR register is set to one (end-of-conversion).

32kHz STAND-BY OSCILLATOR

The 32kHz stand-by oscillator allows the ST6245 to generate real time interrupts and to supply the clock to the LCD driver. This enables the ST6245 to provide real time functions with the LCD display capability and lower power consumption. Figure 37 shows the 32kHz oscillator block diagram.

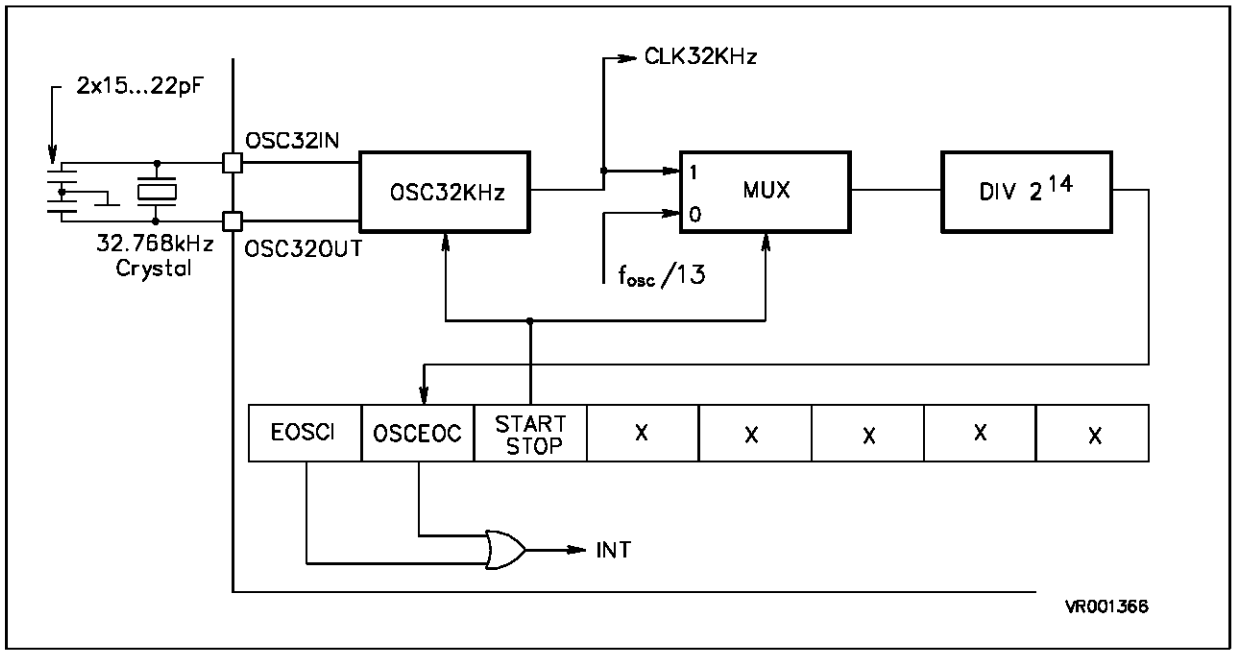
A 32.768kHz quartz crystal must be connected to the OSC32in and OSC32out pins to perform the real time clock operation. Two external capacitors of 15-22pF each must be connected between the oscillator pins and ground. The 32kHz oscillator is managed by the dedicated status/control register located at address 0DBh .

When the 32kHz stand-by oscillator is stopped (bit 5 of the Status/Control register cleared) the divider chain is supplied with a clock signal synchronous with machine cycle ($f_{OSC}/13$), this produces an interrupt request every $13x2^{14}$ clock cycle (i.e. 26.624ms) with an 8MHz quartz crystal.

When the 32kHz stand-by oscillator is enabled (bit 5 of the Status/Control register set to one) the divider chain is directly supplied with the 32kHz oscillator clock. The 32kHz clock from the standby oscillator can also be used as the LCD clock. This allows operation of the LCD in STOP mode. The interrupt output of the 32kHz oscillator peripheral generates an interrupt request every half second (500ms). This can be used to perform a real time clock function when the MCU is in STOP mode.

This interrupt signal is “ORed” with the interrupt request signals of the two on-chip timers and connected to the low level sensitive interrupt input associated to the interrupt vector #3 (FF2h, FF3h). The interrupt request has to be cleared by user software before leaving the interrupt service routine. Discrimination between the three interrupt sources is made by polling the Status/Control registers of Timer 1 (D4h), Timer 2 (D7h) and 32kHz oscillator (DBh).

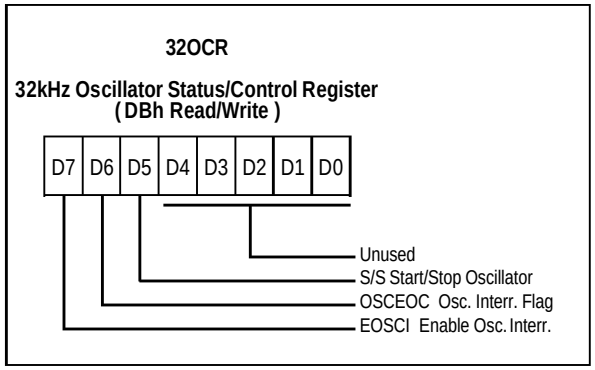
Figure 37. 32kHz Oscillator Block Diagram



32kHz STAND-BY OSCILLATOR (Continued)

32kHz Oscillator Status/Control Register

Figure 38. 32kHz Oscillator Register



EOSCI. Enable Oscillator Interrupt. This bit, when set, enables the 32kHz oscillator interrupt request.

OSCEOC. Oscillator Interrupt Flag. This bit indicates when the 32kHz oscillator has measured a 500ms elapsed time (providing a 32.768kHz quartz crystal is connected to the 32kHz oscillator dedicated pins). An interrupt request can be generated in relation to the state of EOSCI bit. This bit must be cleared by the user program before leaving the interrupt service routine.

START/STOP. Oscillator Start/Stop bit. This bit, when set, enables the 32kHz stand-by oscillator and the free running divider chain is supplied by the 32kHz oscillator signal. When this bit is cleared to zero the divider chain is supplied with the clock signal from the LCD Controller.

This register is cleared during reset.

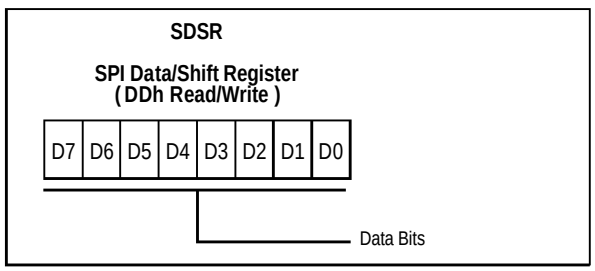
Note:
To achieve minimum power consumption in STOP mode (no system clock), the stand-by oscillator must be switched off (real time function not available) by clearing the Start/Stop bit in the oscillator status/control register.

SERIAL PERIPHERAL INTERFACE (SPI)

The ST6245 SPI is an optimized serial synchronous interface that supports a wide range of industry standard SPI specifications. The ST6245 SPI is controlled by small and simple user software to perform serial data exchange. The serial shift clock can be implemented either by software (using the bit-set and bit-reset instructions), with the on-chip Timer 1 by externally connecting the SPI clock pin to the timer pin or by directly applying an external clock to the SPI.

The peripheral is composed by an 8-bit Data/shift Register (address DDh) and a 4-bit binary counter. The SCL, Sin and Sout SPI data and clock signals are connected to the PB5, PB6 and PB7 I/O lines. With the 3 I/O pins, the SPI can operate in the following operating modes: Software SPI, S-BUS, I^2C-bus and as a standard serial I/O (clock, data, enable). An interrupt request can be generated after eight clock pulses. Figure 40 shows the SPI block diagram.

Figure 39. SPI Data/Shift Register



The PB5/SCL line clocks, on the falling edge, the shift register and the counter. To allow SPI operation the PB5/SCL must be programmed as input, an external clock supplied to this pin will drive the SPI peripheral (slave mode).

If PA5/SCL is programmed as output, a clock signal can be generated by software, setting and resetting the port line by software (master mode).

The SCL clock signal is the shift clock for the SPI data/shift register. The PB6/Sin pin is the serial shift input and PB7/Sout is the serial shift output. These two lines can be tied together to implement two wires protocols (I^2C-bus, etc). When data is serialized, the MSB is the first bit. PB6/Sin has to be programmed as input. For serial output operation PB7/Sout has to be programmed as open-drain output.

After 8 clock pulses (D7..D0) the output $\overline{Q4}$ of the 4-bit binary counter becomes low, disabling the clock from the counter and the data/shift register. Q4 enables the clock to generate an interrupt on the 8th clock falling edge as long as no reset of the counter (processor write into the 8-bit data/shift register) takes place. After a processor reset the interrupt is disabled. The interrupt is active when writing data in the shift register (DDh) and desactivated when writing any data in the register SPI Interrupt Disable (C2h).

SERIAL PERIPHERAL INTERFACE (Continued)

The generation of an interrupt to the Core provides information that new data is available (input mode) or that transmission is completed (output mode), allowing the Core to generate an acknowledge on the 9th clock pulse (I^2C-bus).

Since the SPI interrupt is connected to interrupt #1, the falling edge interrupt option should be selected by clearing to zero bit 6 of the Interrupt Option Register (IOR, C8h).

After power on reset, or after writing the data/shift register, the counter is reset to zero and the clock is enabled. In this condition the data shift register is ready for reception. No start condition has to be detected. Through the user software the Core may pull down the Sin line (Acknowledge) and slow down the SCL, as long as it is needed to carry out data from the shift register.

I^2C-bus Master-Slave, Receiver-Transmitter

When pins Sin and Sout are externally connected together it is possible to use the SPI as a receiver as well as a transmitter. With a simple software routine (by using bit-set and bit-reset on I/O line) a clock can be generated allowing I^2C-bus to work in master mode.

When implementing an I^2C-bus protocol, the start condition can be detected by setting the processor into a “wait for start” condition by simply enabling the interrupt of the PA6/Sin I/O port. This frees the processor from polling the Sin and SCL lines. After the transmission/reception the processor has to poll for the STOP condition.

In slave mode the user software can slow down the SCL clock frequency by simply putting the SCL I/O line in output open-drain mode and writing a zero into the corresponding data register bit.

As it is possible to directly read the Sin pin directly through the port register, the software can detect a difference between internal data and external data (master mode). Similar condition can be applied to the clock.

The typical speed of transmission in I^2C master or slave mode is in the range of 10kHz.

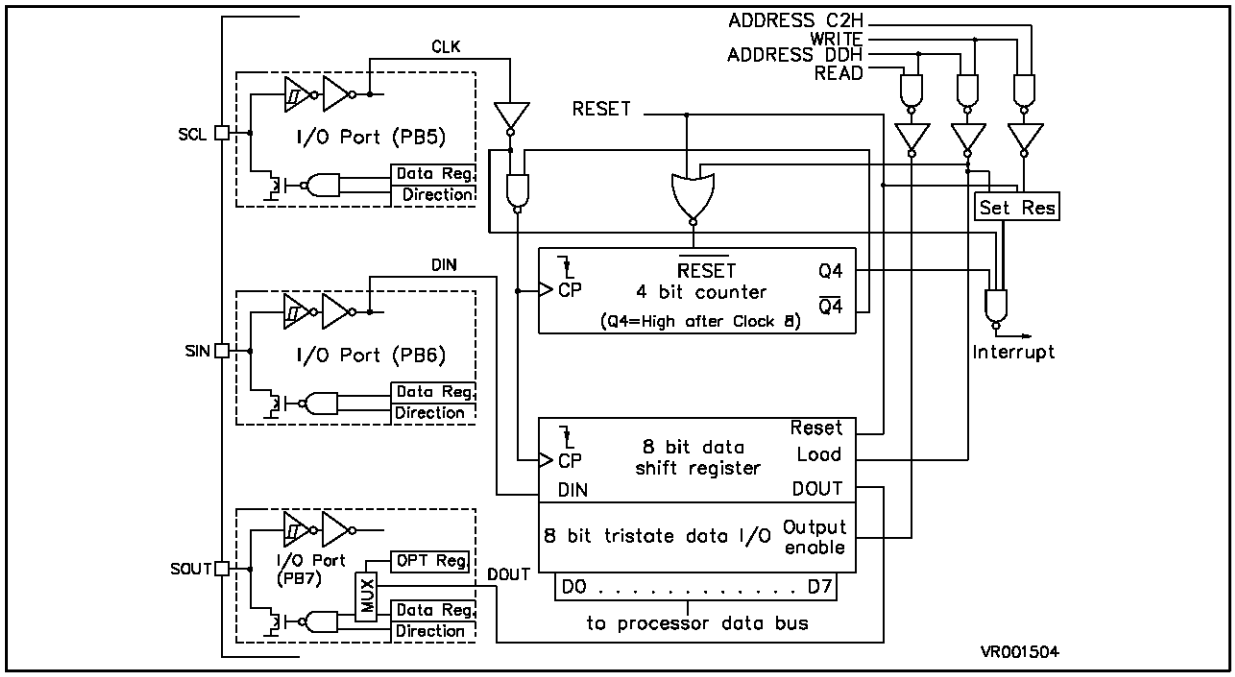
Three (Four) Wire Serial Bus

It is possible to use a single general purpose I/O pin (with the corresponding interrupt enabled) as a “chip enable” pin. SCL acts as active or passive clock pin, Sin as data in and Sout as data out (four wire bus). Sin and Sout can be connected together externally to implement three wire bus.

Note:

When the SPI is not used, the three I/O lines (Sin, SCL, Sout) can be used as normal I/O, with the following limitation: bit Sout cannot be used in open drain mode as this enables the shift register output to the port.
It is recommended, in order to avoid spurious interrupts from the SPI, to disable the SPI interrupt (the default state after reset) i.e. no write must be made to the 8-bit shift register (DDh). An explicit interrupt disable may be made in software by a dummy write to address C2h.

Figure 40. SPI Block Diagram



LCD CONTROLLER-DRIVER

The ST6245 LCD driver consists of a LCD control logic, a programmable prescaler, a 12 bytes wide dedicated LCD RAM, 24 segment and 4 common outputs. This allows a direct driving of up to 96 LCD segments.

The LCD driver is managed by the LCD Mode/Control register located at data RAM address DCh. Different display modes (1/1 duty, 1/2 duty, 1/3 duty and 1/4 duty) are available to cover a wide range of application requirements. The multiplexing display modes are software selectable by programming bits 6 and 7 of the LCD control register. Bits 0-5 are used to select the LCD drive and frame frequency (in relation to the system clock) and to switch off all segments. The LCD Driver can also be supplied by the 32kHz real-time oscillator allowing working in low power conditions and performing real time clock operation.

According to the data in the LCD RAM, the segment and the common drivers generate the segment and common signals which can directly drive an LCD panel.

The LCD control logic reads automatically the data from the LCD RAM independently and without interruption of the processor. The part of the LCD RAM that is not used for displaying can be used as normal data memory.

The scale factor of the clock prescaler can be fixed by software, therefore different frame frequencies can be defined.

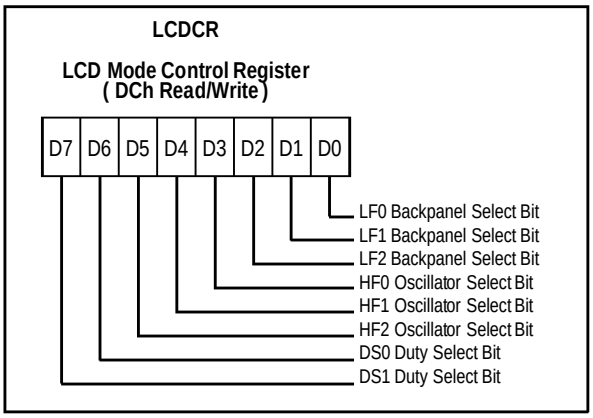
The ST6245 oscillator should operate with a 1.0486, 2.0972, 4.1943, 8.3886MHz frequency quartz crystal. This allows the associated division rates to achieve an internal reference frequency of 32.768kHz. The different division rates can be achieved by programming bits 3, 4, 5 in the LCD control register (see Table 11). It is not recommended to select an internal frequency lower than 32.768kHz as the clock supervisor circuit may switch off the LCD peripheral if the lower frequency is detected.

When the display is turned off, all segment and common outputs are switched to ground, causing all the segments to be switched off regardless of the contents of the LCD RAM.

When the Stand-by oscillator function is selected, the 32kHz stand-by oscillator is selected as clock source for the LCD.

To avoid incomplete frames of the LCD, the mode control bits do not immediately influence the LCD controller when the LCD control register is written. They are stored in a temporary register and change the LCD function only at the end of the frame. Special care must be taken when entering the STOP mode. After switching the LCD clock source from the main oscillator to the 32kHz standby oscillator it must be guaranteed that enough clock pulses are delivered to complete the current frame before entering the STOP mode. Otherwise the LCD function will not be changed and the LCD will be switched OFF after entering the STOP mode. Different LCD frame frequencies for each display mode are selected by bits in the LCD control register (see Table 12).

Figure 41. LCD Mode Control Register



DS0, DS1. Duty cycle select bits. These bits select the number of common backplanes used by the LCD control. This allows different multiplexing conditions.

HF0, HF1, HF2. These bits allow the LCD controller to be supplied with the correct frequency when different high main oscillator frequencies are selected as system clock. Table 11 shows the set-up for different clock crystals.

LF0, LF1, LF2. These bits control the LCD base operational frequency of the LCD common lines. Table 12 shows the set-up to select the different

Table 10. Duty Cycle Selection

DS1	DS0	Display Mode	Active Blackplanes	Max. Number of Segments Driven
0	0	1/4 duty	COM1, 2, 3, 4	180
0	1	1/1 duty	COM1	45
1	0	1/2 duty	COM1, 2	90
1	1	1/3 duty	COM1, 2, 3	135

LCD CONTROLLER-DRIVER (Continued)

Table 11. High Frequency Select Bits

HF2	HF1	HF0	Function	f_{osc}
0	0	0	Display off	
0	0	1	for stand-by Oscillator	32.768kHz
0	1	0	NOT TO BE USED	
0	1	1	÷ 32 for main oscillator	1.048MHz
1	0	0	÷ 64 for main oscillator	2.097MHz
1	0	1	÷ 128 for main oscillator	4.194MHz
1	1	0	÷ 256 for main oscillator	8.388MHz
1	1	1	NOT TO BE USED	

Notes :

1. The usage f_{osc} values different from those defined in this table cause the LCD to operate at a reference frequency different from 32.768kHz.
2. It is not recommended to select an internal frequency lower than 32.768kHz as the clock supervisor circuit may switch off the LCD peripheral if lower frequency is detected.

Table 12. LCD Frequency Select Bits

LF2	LF1	LF0	f_{LCD} (Hz)
0	0	0	64
0	0	1	85
0	1	0	128
0	1	1	171
1	0	0	256
1	0	1	341
1	1	0	512
1	1	1	Not to be Used

Table 13. Available Frame Frequencies for LCD

f_{LCD} (Hz)	Frame Frequency f_F (Hz)			
	1/1 duty	1/2 duty	1/3 duty	1/4 duty
512	512	256	171	128
341	341	171	114	85
256	256	128	85	64
171	171	85	57	43
128	128	64	43	32
85	85	43	28	21
64	64	32	21	16

frequencies while Table 13 shows the corresponding frame values with the different multiplexing conditions.

According to the selected LCD drive frequency f_{LCD} the frame frequencies come out as shown in Table 13.

The Figure 48 illustrates the waveforms of the different duty signals.

The value of the VLCD voltage can be chosen independently from V_{DD} according to the display requirements. The intermediate VLCD levels 2/3 VLCD, 1/3 VLCD and 1/2 VLCD are generated by an internal resistor network as shown in Figures 46 and 47. The half VLCD level for 1/2 duty cycle is obtained by the external connection of VLCD1/3 and VLCD2/3 pins. All intermediate VLCD levels are connected to pins to enable external capacitive buffering or resistive shunting.

LCD CONTROLLER-DRIVER (Continued)

The internal resistive divider network is realized with two parallel dividers. One has high resistivity, the other one low resistivity. The high resistive divider (R_H) is permanently switched on during the LCD operation. The low resistive divider (R_L) is only switched on for a short period of time when the levels of common lines and segment lines are changed. This method combines low source impedance for fast switching of the LCD pixels with high source impedance for low power consumption. Figure 42 shows the typical current into V_{LCD} pin in dependency of the display voltage V_{LCD}.

When the display is switched off (by program or reset) the internal resistor network is also switched off to achieve minimum power consumption. The low resistivity divider is active at each edge of f_{LCD} during 8 clock cycles of F_{32kHz}.

The internal resistor network is implemented with resistive transistor elements to achieve high precision. For display voltages $V_{LCD} < 4.5V$ the resistivity of the divider may be too high for some applications (especially using 1/3 or 1/4 duty display mode). In that case an external resistive divider must be used to achieve the desired resistivity.

Figure 42. Typical Current Consumption on VLCD Pin (25°C, no load, fLCD= 512Hz, mux= 1/3-1/4)

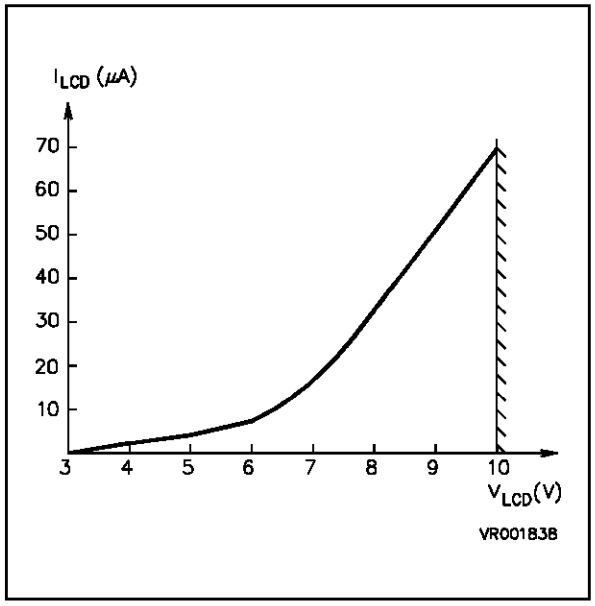
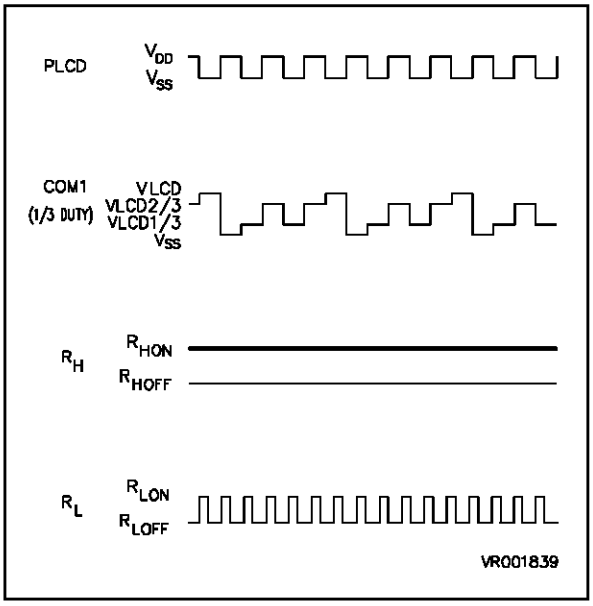


Figure 43. Typical Chronogram of Activation of the V_{LCD} Divider Network



LCD CONTROLLER-DRIVER (Continued)

Typical External resistances values are in the range of 100 kΩ to 150 kΩ. External capacitances in the range of 10 to 47 nF can be added to V_{LCD} 2/3 and V_{LCD} 1/3 pins and to V_{LCD} if the V_{LCD} connection is highly impedant.

When the program is switched off (by program or reset) the internal resistor network is also switched off to achieve minimum power consumption.

Figure 44. Typical Network to connect to V_{LCD} pins if $V_{LCD} \leq 4.5V$

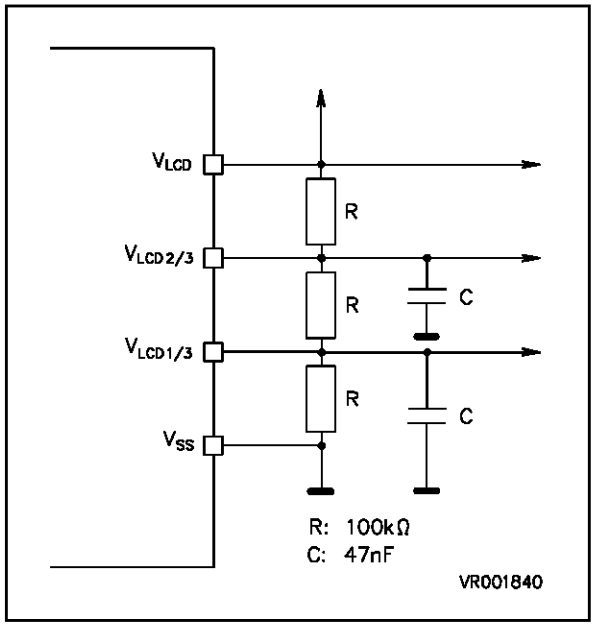


Figure 45. Generation of the 32kHz clock

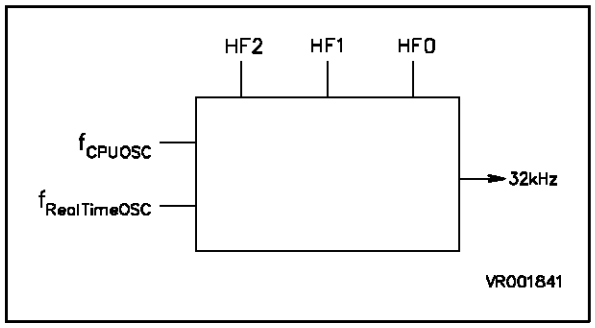


Figure 46. Bias Config for 1/2 Duty

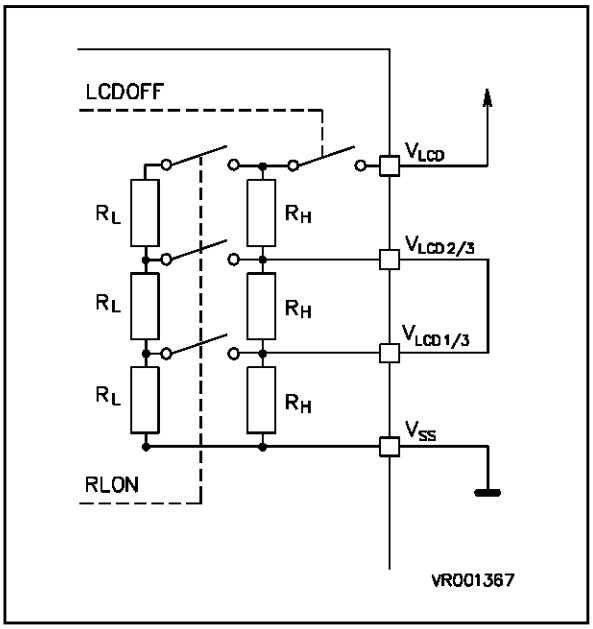
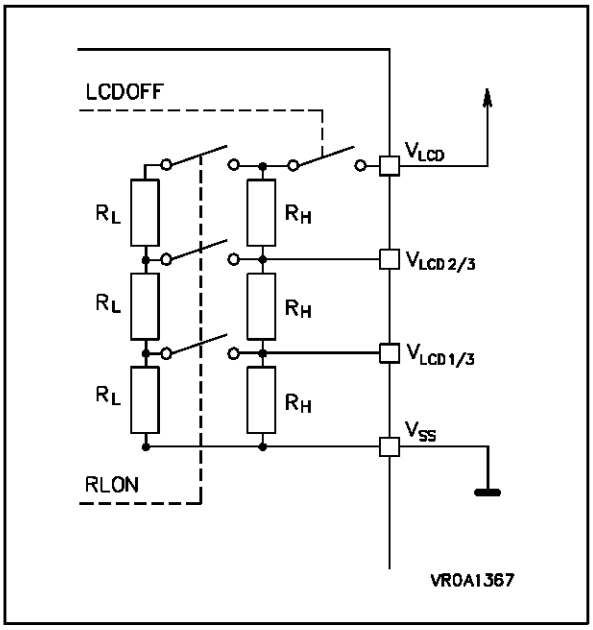


Figure 47. Bias Configuration for 1/1, 1/3 and 1/4 Duty Operation of LCD



LCD CONTROLLER-DRIVER (Continued)

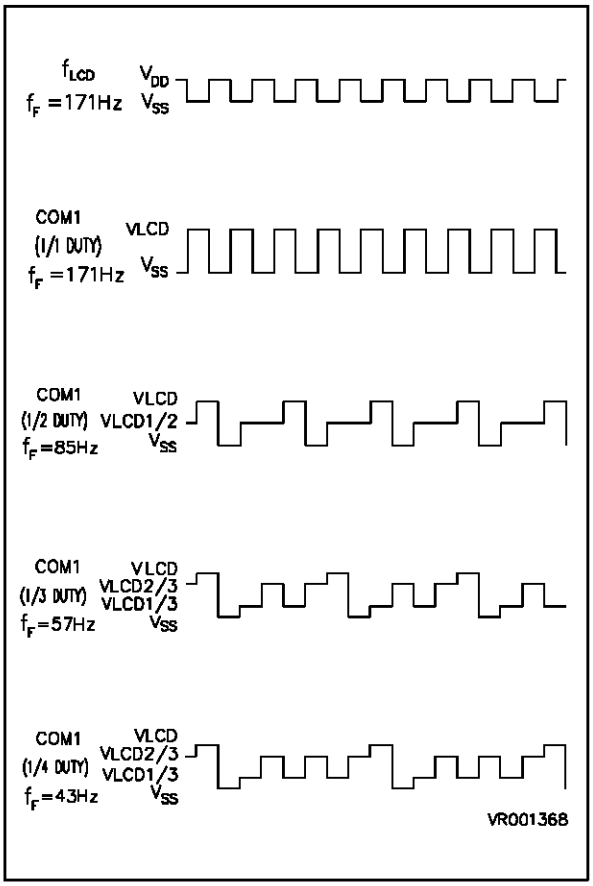
Address Mapping of the Display Segments.

The LCD RAM is located in the ST6245 data space from addresses E0h to F7h. The LCD forms a matrix of 24 segment lines (rows) and up to 4 common lines (columns). Each bit of the LCD RAM is mapped to one element of the LCD matrix, as described in Figure 49. If a bit is set, the corresponding LCD segment is switched on; if it is reset, the segment is switched off. The segments outputs S1-S16 and S41-S48 are not connected to any pin.

When multiplex rates lower than 1/4 are selected, the unused LCD RAM is free for general use. In the 1/2 duty mode, for instance, half of the LCD RAM is available for storing general purpose data. The address range from F8h to FEh can be used as general purpose data RAM, but not for displaying data (it is reserved for future LCD expansion).

After a reset, the LCD RAM is not initializated and contains arbitrary information. As the LCD control register is reset, the LCD is completely switched off.

Figure 48. Common Signal Waveforms



LCD CONTROLLER-DRIVER (Continued)

Figure 49. Addressing Map of the LCD RAM

Data RAM Address	MSB							LSB	
E0* E1* E2 E3 E4 E5*	S24 S32 S40	S23 S31 S39	S22 S30 S38	S21 S29 S37	S20 S28 S36	S19 S27 S35	S18 S26 S34	S17 S25 S33	**COM1**
E6* E7* E8 E9 EA EB*	S24 S32 S40	S23 S31 S39	S22 S30 S38	S21 S29 S37	S20 S28 S36	S19 S27 S35	S18 S26 S34	S17 S25 S33	**COM2**
EC* ED* EE EF F0 F1*	S24 S32 S40	S23 S31 S39	S22 S30 S38	S21 S29 S37	S20 S28 S36	S19 S27 S35	S18 S26 S34	S17 S25 S33	**COM3**
F2* F3* F4 F5 F6 F7*	S24 S32 S40	S23 S31 S39	S22 S30 S38	S21 S29 S37	S20 S28 S36	S19 S27 S35	S18 S26 S34	S17 S25 S33	**COM4**
F8 - FE*									

Note: *. Row to be used as general purpose data RAM (not for display data)

Notes:

In STOP mode no clock is available for the LCD controller from the main oscillator. If the 32kHz oscillator is activated the LCD can also operate in STOP mode. If the stand-by oscillator is not active, the LCD controller is switched off when STOP instruction is executed; this mode has to be selected to reach the lowest power consumption.

A missing LCD clock (no oscillator active, broken crystal, etc.) is detected by a clock supervisor circuit that switches all the segments and common lines to ground to avoid destructive DC levels at the LCD.

The LCD function change is only effective at the end of a frame. For this reason special care has to be taken when entering the STOP mode. After switching the LCD clock source from the main oscillator to the 32kHz stand-by oscillator it must be guaranteed that enough clock pulses are delivered to complete the current frame before entering the STOP mode. Otherwise the LCD function will not be changed and the LCD is switched off after entering the STOP mode.

The RAM address F8-FEh are not used for LCD display purposes. So they are available as 7 additional Data RAM registers.

SOFTWARE DESCRIPTION

The ST62xx software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum, in short to provide byte efficient programming capability. The ST62xx core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

Addressing Modes

The ST62xx core has nine addressing modes which are described in the following paragraphs. The ST62xx core uses three different address spaces: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte that is processed by the instruction is stored in the location that follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode . In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) that use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction that follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits that characterize the kind of the test, one bit that determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits that give the span of the branch (0h to Fh) that must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

SOFTWARE DESCRIPTION (Continued)

Instruction Set

The ST62xx core has a set of 40 basic instructions. When these instructions are combined with nine addressing modes, 244 usable opcodes can be obtained. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, bit manipulation. The following paragraphs describe the different types.

All the instructions within a given type are presented in individual tables.

Load & Store. These instructions use one,two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 14. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ . Affected
* . Not Affected

SOFTWARE DESCRIPTION (Continued)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 15. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	*
AND A, (Y)	Indirect	1	4	Δ	*
AND A, rr	Direct	2	4	Δ	*
ANDI A, #N	Immediate	2	4	Δ	*
CLR A	Short Direct	2	4	Δ	Δ
CLR rr	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

SOFTWARE DESCRIPTION (Continued)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 16. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ. Affected
*. Not Affected

Table 17. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
*. Not Affected

Table 18. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated and a WAIT is automatically executed instead of a STOP if the Watchdog function is selected.
Δ. Affected
*. Not Affected

Table 19. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc.12-bit address;
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

Opcode Map Summary. The following table contains an opcode map for the instructions used on the MCU.

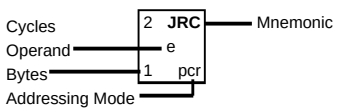
LOW / HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	LOW / HI
0 0000	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b0,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **LD** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b0,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LDI** rr,nn 3 imm	2 **JRC** e 1 pcr	4 **LD** a,(y) 1 ind	0 0000
1 0001	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b0,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** x 1 sd	2 **JRC** e 1 prc	4 **LDI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b0,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** x 1 sd	2 **JRC** e 1 pcr	4 **LD** a,rr 2 dir	1 0001
2 0010	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b4,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **CP** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b4,rr 2 b.d	2 **JRZ** e 1 pcr	4 **COM** a 1 inh	2 **JRC** e 1 pcr	4 **CP** a,(y) 1 ind	2 0010
3 0011	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b4,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,x 1 sd	2 **JRC** e 1 prc	4 **CPI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b4,rr 2 b.d.	2 **JRZ** e 1 pcr	4 **LD** x,a 1 sd	2 **JRC** e 1 pcr	4 **CP** a,rr 2 dir	3 0011
4 0100	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b2,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **ADD** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b2,rr 2 b.d	2 **JRZ** e 1 pcr	2 **RETI** 1 inh	2 **JRC** e 1 pcr	4 **ADD** a,(y) 1 ind	4 0100
5 0101	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b2,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** y 1 sd	2 **JRC** e 1 prc	4 **ADDI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b2,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** y 1 sd	2 **JRC** e 1 pcr	4 **ADD** a,rr 2 dir	5 0101
6 0110	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b6,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **INC** (x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b6,rr 2 b.d	2 **JRZ** e 1 pcr	2 **STOP** 1 inh	2 **JRC** e 1 pcr	4 **INC** (y) 1 ind	6 0110
7 0111	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b6,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,y 1 sd	2 **JRC** e 1 prc	#	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b6,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LD** y,a 1 sd	2 **JRC** e 1 pcr	4 **INC** rr 2 dir	7 0111
8 1000	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b1,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **LD** (x),a 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b1,rr 2 b.d	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 pcr	4 **LD** (y),a 1 ind	8 1000
9 1001	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b1,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** v 1 sd	2 **JRC** e 1 prc	#	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b1,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** v 1 sd	2 **JRC** e 1 pcr	4 **LD** rr,a 2 dir	9 1001
A 1010	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b5,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **AND** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b5,rr 2 b.d	2 **JRZ** e 1 pcr	4 **RLC** a 1 inh	2 **JRC** e 1 pcr	4 **AND** a,(y) 1 ind	A 1010
B 1011	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b5,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,v 1 sd	2 **JRC** e 1 prc	4 **ANDI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b5,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LD** v,a 1 sd	2 **JRC** e 1 pcr	4 **AND** a,rr 2 dir	B 1011
C 1100	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b3,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **SUB** a,(x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b3,rr 2 b.d	2 **JRZ** e 1 pcr	2 **RET** 1 inh	2 **JRC** e 1 pcr	4 **SUB** a,(y) 1 ind	C 1100
D 1101	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b3,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **INC** w 1 sd	2 **JRC** e 1 prc	4 **SUBI** a,nn 2 imm	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b3,rr 2 b.d	2 **JRZ** e 1 pcr	4 **DEC** w 1 sd	2 **JRC** e 1 pcr	4 **SUB** a,rr 2 dir	D 1101
E 1110	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRR** b7,rr,ee 3 bt	2 **JRZ** e 1 pcr	#	2 **JRC** e 1 prc	4 **DEC** (x) 1 ind	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **RES** b7,rr 2 b.d	2 **JRZ** e 1 pcr	2 **WAIT** 1 inh	2 **JRC** e 1 pcr	4 **DEC** (y) 1 ind	E 1110
F 1111	2 **JRNZ** e 1 pcr	4 **CALL** abc 2 ext	2 **JRNC** e 1 pcr	5 **JRS** b7,rr,ee 3 bt	2 **JRZ** e 1 pcr	4 **LD** a,w 1 sd	2 **JRC** e 1 prc	#	2 **JRNZ** e 1 pcr	4 **JP** abc 2 ext	2 **JRNC** e 1 pcr	4 **SET** b7,rr 2 b.d	2 **JRZ** e 1 pcr	4 **LD** w,a 1 sd	2 **JRC** e 1 pcr	4 **DEC** rr 2 dir	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from:

$Tj = T_A + PD \times RthJA$

Where : T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_{LCD}	Display Voltage	-0.3 to 11.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
I_O	Current Drain per Pin Excluding V_{DD} & V_{SS}	± 10	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PQFP52		70		°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Condition s	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version 6 Suffix Version	0 -40		70 85	°C
V_{DD}	Operating Supply Voltage		3		6	V
V_{LCD}	Display Voltage		3		10	V
V_{DD}	RAM Retention Voltage		2			V

RECOMMENDED OPERATING CONDITIONS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (1)(4)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01 0.01		8.388 2	MHz
I_{INJ+}	Pin Injection Current (positive) Digital Input (2) Analog Input (3)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input (2) Analog Input	V_{DD} = 4.5 to 5.5V			-5	mA

Notes :
1. An oscillator frequency above 1MHz is recommanded for reliable A/D results.
2. A current of ± 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device. For a positive current injected into one pin, a part of this current (~ 10%) can be expected to flow from the neighbouring pins. A current of -5mA can be forced on one input of the analog section at a time (or -2.5mA for all inputs at a time) without affecting the conversion.
3. If a total current of +1mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 1mA, all the conversion is resulting shifted of +1LSB. If a total positive current of +5mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 5mA, all the conversion is resulting shifted of +2LSB.
4. Operation below 0.01 MHz is possible but requires increased supply current.

EEPROM INFORMATION

The ST62xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	RESET, NMI, TIMER, WDON Pin			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	TIMER	$0.80V_{DD}$			V
		RESET, NMI, WDON Pin	$0.70V_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	RESET Pin V_{DD} =5V $V_{IN} = V_{DD}$ (1) $V_{IN} = V_{DD}$ (2) $V_{IN} = V_{SS}$ (5)			10 1 50	µA mA µA
V_{OL}	Low Level Output Voltage	TIMER, I_{OL} = 5.0mA			$0.2V_{DD}$	V
V_{OH}	High Level Output Voltage	TIMER, I_{OL} = -5.0mA	$0.65V_{DD}$			V
R_{PU}	Pull-up Resistor	V_{IN}=0V V_{DD}=5V WDON - NMI	40	100	200	kΩ
		RESET	200	300	500	kΩ

Notes on next page

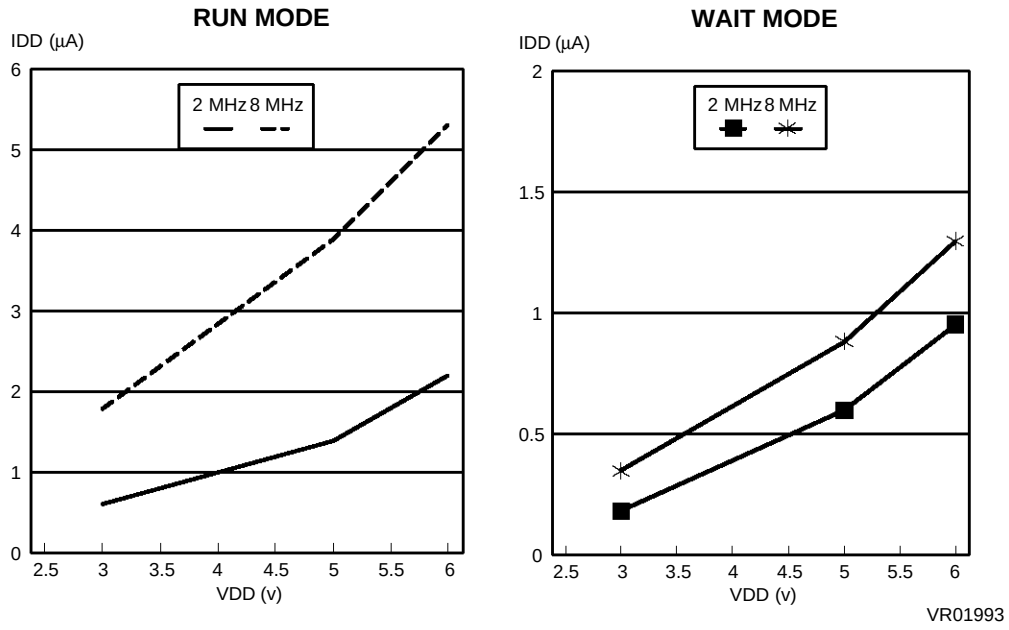
DC ELECTRICAL CHARACTERISTICS (Continued)
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{IL} I_{IH}	Input Leakage Current	TIMER $V_{IN} = V_{DD}$ or V_{SS}		0.1	1.0	µA
I_{IL} I_{IH}	Input Leakage Current	NMI VDD = 5.V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{IL} I_{IH}	Input Leakage Current	WDON VDD = 5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{DD}	Supply Current RUN Mode	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		4	7	mA
	Supply Current WAIT Mode (4)	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		1	2	mA
	Supply Current RESET Mode	f_{OSC} = 8MHz, $V_{RESET} = V_{SS}$		1	7	mA
	Supply Current STOP Mode (3)(4)	I_{LOAD} = 0mA V_{DD} = 5.0V		1	10	µA

Notes :
1. No Watchdog Reset activated.
2. Reset generated by Watchdog.
3. When the watchdog function is actvated the STOP instruction is deactivated. WAIT instruction is automatically executed.
4. All on-chip peripherals in OFF state
5. Pull-up resistor

CURRENT CONSUMPTION

IDD Current Versus Supply Voltage
Typical Values, Temp : -40/+85°C



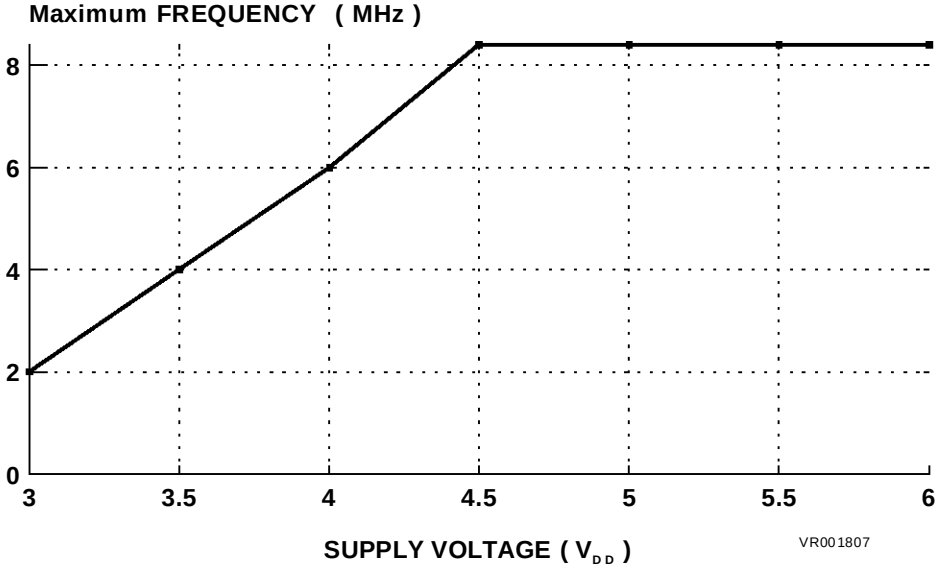
AC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (2)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01		8.388 2	MHz
t_{SU}	Oscillator Start-up Time	$C_{L1} = C_{L2}$ = 22pF - crystal		5	20	ms
t_{SR}	Supply Rise Time	10% to 90%	0.01		100	
t_{REC}	Supply Recovery Time (1)		100			
T_W	Minimum Pulse Width	NMI Pin V_{DD} = 5V	100			ns
		RESET Pin	100			ns
T_{WEE}	EEPROM Write Time	T_A = 25°C One Byte T_A = 85°C One Byte		5 15	10 25	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A L_{OT} Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention	T_A = 55°C	10			years
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:
1. Period for which V_{DD} has to be connected or at 0V to allow internal Reset function at next power-up.
2. Operation below 0.01 MHz is possible but requires increased supply current.

Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



I/O PORTS
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	I/O Pins			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	I/O Pins	$0.7V_{DD}$			V
V_{OL}	Low Level Output Voltage	I/O Pins, I_O = 10µA (sink)			0.1	V
		I/O Pins, $I_{OL} = V_{DD}$x1mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 1.6mA V_{DD} = 3V			0.4	V
	Low Level Output Voltage, PB4-PB7 Only	I/O Pins, $I_{OL} = V_{DD}$x2mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 3.2mA V_{DD} = 3V			0.4	V
		I/O Pins, $I_{OL} = V_{DD}$x4mA V_{DD} = 4.5 to 6V			$0.26xV_{DD}$	V
		I/O Pins, I_{OL} = 6.4mA V_{DD} = 3V			0.8	V
V_{OH}	High Level Output Voltage	I/O Pins, I_O = -10µA (source)	V_{DD}-0.1			V
		I/O Pins, $I_{OL} = -V_{DD}$x1mA V_{DD} = 5.0V	$0.6xV_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	I/O Pins,(1)		0.1	1.0	µA
R_{PU}	Pull-up Resistor	I/O Pins V_{IN} = 0V, V_{DD} = 5.0V	40	100	200	KΩ

Note 1. Pull-up resistor off

SPI ELECTRICAL CHARACTERISTICS
(V_{DD} = 5.0V, T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
F_{CL}	Clock Frequency	applied on PB5/SCL			1	MHz
t_{SU}	Set-up Time	applied on PB6/Sin		50		ns
t_h	Hold Time	applied on PB6/Sin		100		ns

A/D CONVERTER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution (3)			8		Bit
A_{TOT}	Total Accuracy (3)	f_{OSC} > 1.2 MHz f_{OSC} > 32kHz			± 2 ±4	LSB
t_C(1)	Conversion Time	f_{OSC} = 8MHz		70		µs
V_{AN}	Conversion Range		V_{SS}		V_{DD}	V
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN}(2)	Analog Input Capacitance			2	5	pF
ASI	Analog Source Impedance				30	kΩ
SSI	Analog Reference Supply Impedence				2	kΩ

Notes:
1. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.
2. Excluding Pad Capacitance
3. Noise at V_{DD},V_{SS} ≤ 10mV

TIMER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{12}{f_{OSC}}$			second
f_{IN}	Input Frequency on TIMER Pin				$\frac{f_{OSC}}{8}$	MHz
t_W	Pulse Width at TIMER Pin	$V_{DD} \geq 3V$ $V_{DD} \geq 4.5V$	1 125			µs ns

LCD ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

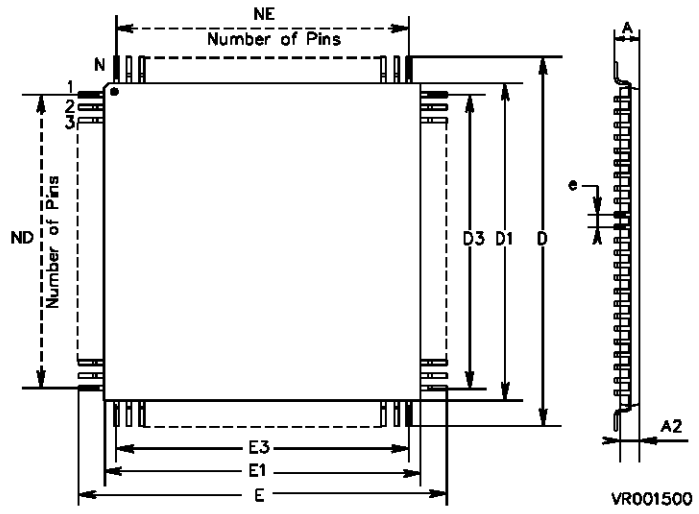
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{FR}	Frame Frequency	1/4 Duty f_{OSC} = 1, 2, 4, 8MHz	16		128	Hz
V_{OS}	DC Offset Voltage(1)	$V_{LCD} = V_{DD}$, no load			50	mV
V_{OH}	COM High Level, Output Voltage	I = 100µA, V_{LCD} = 5V	4.5V			V
V_{OL}	COM Low Level, Output Voltage	I = 100µA, V_{LCD} = 5V			0.5V	V
V_{OH}	SEG High Level, Output Voltage	I = 50µA, V_{LCD} = 5V	4.5V			V
V_{OL}	SEG Low Level, Output Voltage	I = 50µA, V_{LCD} = 5V			0.5V	V
V_{LCD}	Display Voltage	Note 2	3		10	V

Notes :

1. The DC offset voltage refers to all segment and common outputs. It is the difference between the measured voltage value and nominal value for every voltage level. Ri of voltage meter must be greater than or equal to 100MΩ.
2. An external resistances network is required when $V_{LCD} \leq 4.5V$.

PACKAGE MECHANICAL DATA

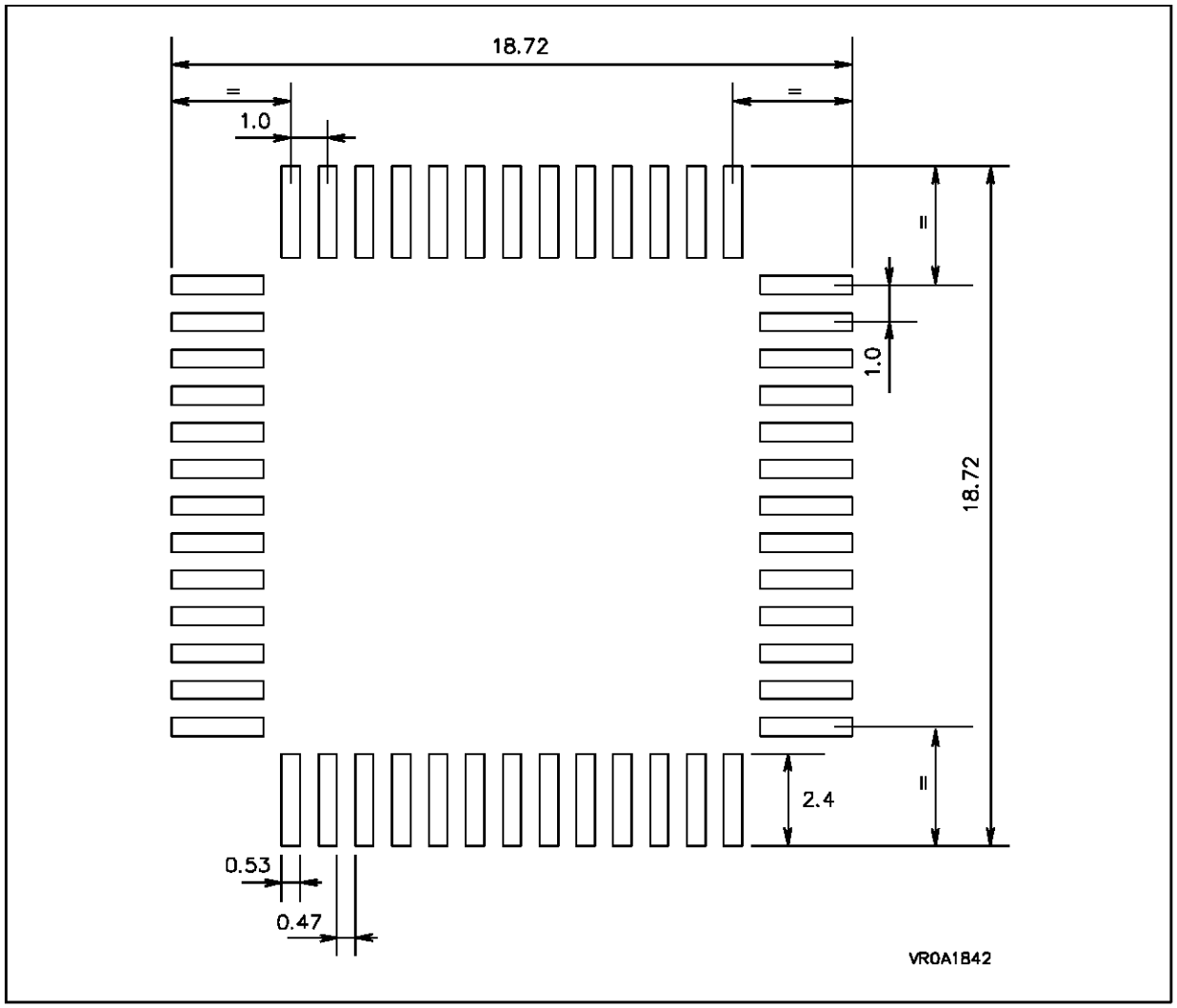
Figure 50. ST6245 52 Pin Plastic Quad Flat Pack Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.40			0.134
A2	2.55	2.80	3.05	0.100	0.110	0.120
D	16.95	17.20	17.45	0.667	0.677	0.687
D1	13.90	14.00	14.10	0.547	0.551	0.555
D3		12.00			0.472	
E	16.95	17.20	17.45	0.667	0.677	0.687
E1	13.90	14.00	14.10	0.547	0.551	0.555
E3		12.00			0.472	
e		1.00			0.039	
Number of Pins						
ND	13					
NE	13					
N	52					

PACKAGE MECHANICAL DATA (Continued)

Recommanded Solder Pad Footprint For QFP52 (in mm)



ORDERING INFORMATION

The following chapter deals with the procedure for transfer the Program/Data ROM codes to SGS-THOMSON.

Communication of the ROM Codes. To communicate the contents of Program/Data ROM memories to SGS-THOMSON, the customer has to send :

– one file in INTEL INTELLEC 8/MDS FORMAT (in an MS-DOS 5" diskette) for the PROGRAM Memory

– one file in INTEL INTELLEC 8/MDS FORMAT (in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)

– a filled Option List form as described in the OPTION LIST paragraph.

The program ROM should respect the ROM Memory Map as in Table 20.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

Table 20. ROM Memory Map

ROM Page	Device Address	Description
Page 0	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector

Note : EPROM addresses are related to the ROM file to be processed.

Customer EEPROM Initial Contents : Format

a. The content should be written into an INTEL INTELLEC format file.

b. In the case of 128 bytes of EEPROM, the starting address in 000h and the end in 7Fh.

c. Undefined or don't care bytes should have the content FFh.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are compared and a computer listing is generated from them. This listing refers exactly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

ORDERING INFORMATION TABLE

Sales Types	Memory Type	Temperature Range	Package
ST6245Q1/XX	4K ROM 64 bytes EEPROM	0 to + 70°C	PQFP52
ST6245Q6/XX	4K ROM 64 bytes EEPROM	-40 to + 85°C	PQFP52

Note : "XX" is the ROM code identifier allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

ST6245 MICROCONTROLLER OPTION LIST

Customer . . . . . .

Address . . . . . .

Contact . . . . . .

Phone No . . . . . .

Reference . . . . . .

SGS-THOMSON Microelectronics references

Device	[ ] ST6245	
Package	[ ] Plastic Quad Flat Package	
Temperature Range	[ ] 0°C to + 70°C	[ ] -40°C to + 85°C

Special Marking [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _"

Authorized characters are Letters, digits, '.', '–', '/' and spaces only.
For marking one line with 10 characters maximum is possible.

Comments :
- Number of LCD segments used :
- Number of LCD backplanes used :

Note :

Signature

Date

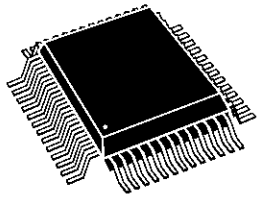
NOTES:

SGS-THOMSON MICROELECTRONICS

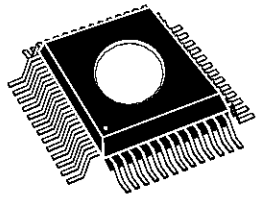
ST62E45
ST62T45

8-BIT OTP/EPROM HCMOS MCUs WITH LCD DRIVER,EEPROM AND A/D CONVERTER

- 3 to 6V supply operating range
- 8.4MHz Maximum Clock Frequency
- Run, Wait & Stop Modes
- 5 different interrupt vectors
- Look-up table capability in EPROM
- User EPROM: 3884 bytes
  Data RAM: 128 bytes
  LCD RAM: 12 bytes
  EEPROM: 64 bytes
- PQFP52 and CQFP52-W packages
- 11 fully software programmable I/O as:
  - Input with/without pull-up resistor
  - Input with interrupt generation
  - Open-Drain or Push-pull outputs
  - Analog Inputs (7 pins)
- 4 I/O lines can sink up to 20mA for direct LED or TRIAC driving and have SPI alternate functions
- Two 8-bit counters and 7-bit programmable prescalers (Timers 1 and 2)
- Software activated digital watchdog
- 8-bit A/D converter with up to 7 analog inputs
- 8-bit synchronous serial peripheral interface (SPI)
- LCD driver with 24 segment outputs, 4 backplane outputs and selectable duty cycle for up to 96 LCD segments direct driving
- 32kHz oscillator for stand-by LCD operation
- One external not maskable interrupt
- 9 powerful addressing modes
- The accumulator, the X, Y, V & W registers, the port and peripherals data & control registers are addressed in the data space as RAM locations.
- The ST62E45 is the EPROM version, ST62T45 is the OTP version, fully compatible with ST6245 ROM version.



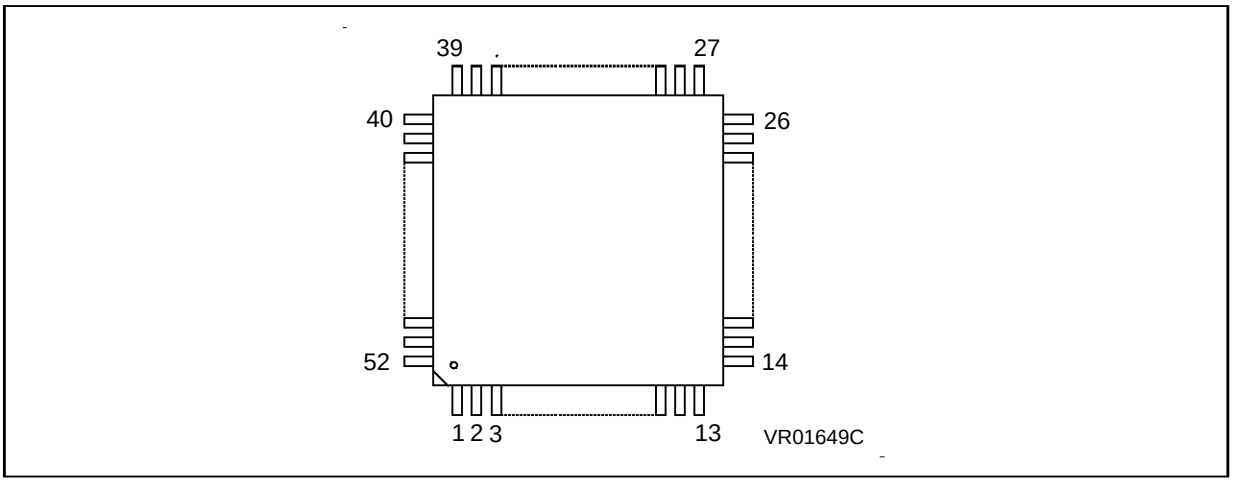
PQFP52



CQFP52-W

(Ordering Information at the end of the datasheet)

Figure 1. 52 Pin Quad Flat Pack (QFP) Package Pinout



ST62E45/T45 Pin Description

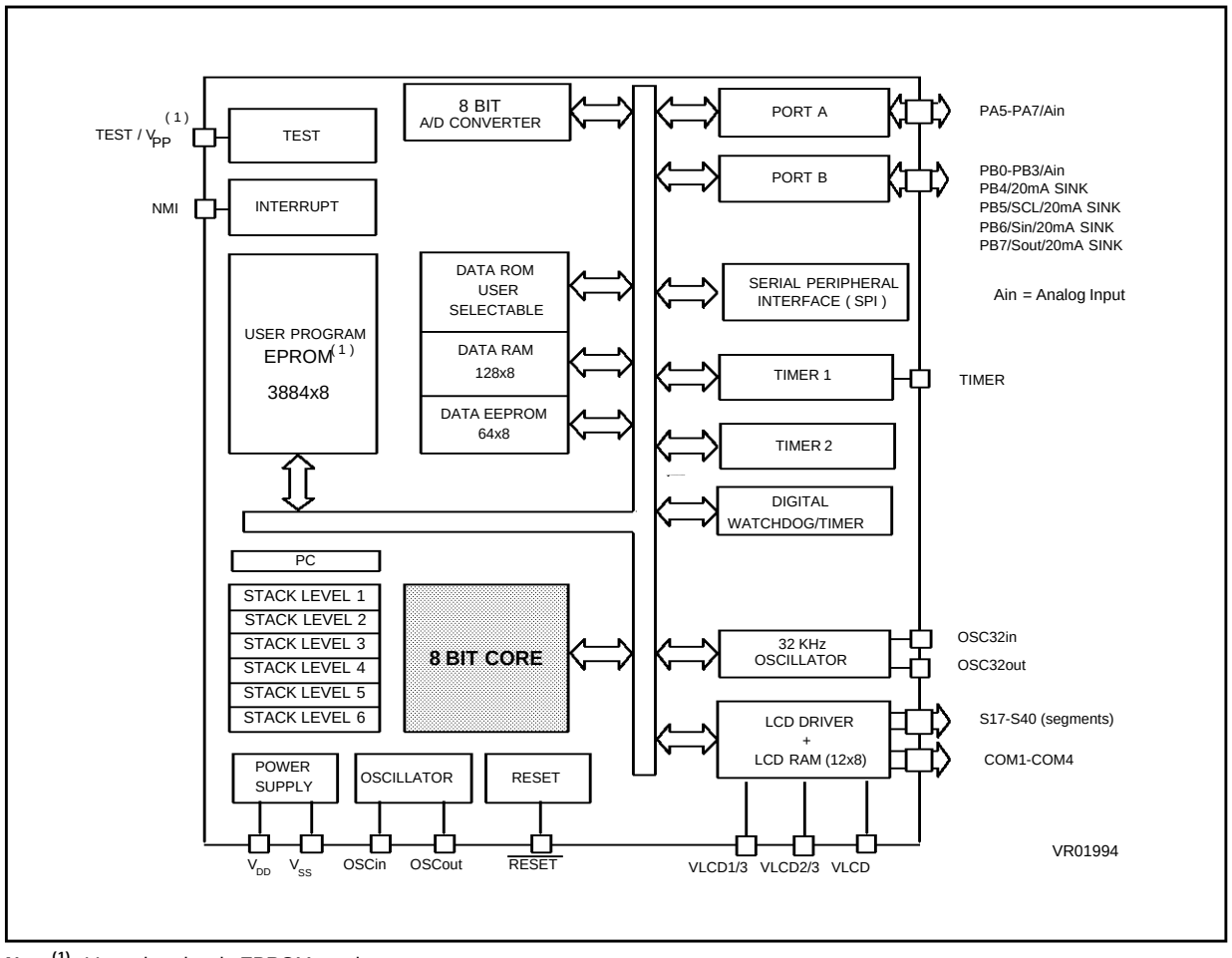
Pin number	Pin name	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
1	COM4	14	$\overline{\text{RESET}}$	27	OSC32out	40	S28
2	COM3	15	OSCout	28	OSC32in	41	S29
3	COM2	16	OSCin	29	S17	42	S30
4	COM1	17	NMI	30	S18	43	S31
5	VLCD1/3	18	TIMER	31	S19	44	S32
6	VLCD2/3	19	PB7/Sout (1)	32	S20	45	S33
7	VLCD	20	PB6/Sin (1)	33	S21	46	S34
8	PA7/Ain	21	PB5/SCL (1)	34	S22	47	S35
9	PA6/Ain	22	PB4 (1)	35	S23	48	S36
10	PA5/Ain	23	PB3/Ain	36	S24	49	S37
11	TEST	24	PB2/Ain	37	S25	50	S38
12	V_{DD}	25	PB1/Ain	38	S26	51	S39
13	V_{SS}	26	PB0/Ain	39	S27	52	S40

Note 1: 20mA SINK

GENERAL DESCRIPTION

The ST62E45,T45 microcontrollers are members of the 8-bit HCMOS ST62xx family, a series of devices oriented to low-medium complexity applications. They are the EPROM/OTP versions of the ST6245 ROM device and are suitable for product prototyping and low volume production. All ST62xx members are based on a building block approach: a common core is associated with a combination of on-chip peripherals (macrocells). The macrocells of the ST6245 family are: a high performance LCD controller/driver with 24 segment outputs and 4 backplanes able to drive up to 96 segments, two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), the digital watchdog timer (DWD), an 8-bit A/D Converter with up to 7 analog inputs and an 8-bit synchronous Serial Peripheral Interface (SPI). In addition these devices offer 64 bytes of EEPROM for storage of non volatile data. Thanks to these peripherals the ST6245 family is well suited for general purpose, automotive, security, appliance and industrial applications.

Figure 2. ST62E45 Block Diagram



Note (1) : V_{PP} only exists in EPROM version

PIN DESCRIPTION

V_{DD} and **V_{SS}**. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected with the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. OSCin is the input pin, OSCout is the output pin. An external clock signal can be applied to OSCin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to restart the microcontroller at the beginning of its program. The $\overline{\text{RESET}}$ pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TEST/V_{PP}. The TEST must be held at V_{SS} for normal operation (an internal pull-down resistor selects normal operating mode if TEST pin is not connected).

NMI. The NMI pin provides the capability for asynchronous applying an external top priority interrupt to the MCU.This pin is falling edge sensitive. The NMI pin is provided with an on-chip pull-up resistor and schmitt trigger input characteristics.

TIMER. This is the TIMER 1 I/O pin. In input mode it is connected to the prescaler and acts as external timer clock or as control gate for the internal timer clock. In the output mode the TIMER pin outputs the data bit when a time-out occurs.

PA5-PA7. These 3 lines are organized as one I/O port (A). Each line may be configured under software control as an input with or without pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output or as analog input for the A/D converter. Port A has a 5mA drive capability in output mode.

PB0-PB3,PB4-PB7. These 8 lines are organized as one I/O port (B). Each line may be configured under software control as an input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output. PB0-PB3 can be programmed as analog inputs for the A/D converter while PB4-PB7 can also sink 20mA for direct LED driving. PB5-PB7 can also be used as respectively Clock, Data in and Data out pins for the on-chip SPI to carry the synchronous serial I/O signals.

COM1-COM4. These four pins are the LCD peripheral common outputs. They are the outputs of the on-chip backplane voltage generator which is used for multiplexing the 24 LCD lines allowing up to 96 segments to be driven.

S17-S40. These pins are the 24 LCD peripheral driver outputs of ST62E45. Segments S1-S16 and S41-S48 are not connected to any pin.

VLCD. Display voltage supply. It determines the high voltage level on COM1-COM4 and S1-S24 pins.

VLCD1/3, VLCD2/3. Display supply voltage inputs for determining the display voltage levels on COM1-COM4 and S1-S24 pins during multiplex operation.

OSC32in and OSC32out. These pins are internally connected with the on-chip 32kHz oscillator circuit. A 32.768kHz quartz crystal can be connected between these two pins if it is necessary to provide the LCD stand-by clock and real time interrupt. OSC32in is the input pin, OSC32out is the output pin.

ST62E45,T45 EPROM/OTP DESCRIPTION

The ST62E45 is the EPROM version of the ST6245 ROM product. It is intended for use during the development of an application, and for pre-production and small volume production. The ST62T45 OTP has the same characteristics. Both include EPROM memory instead of the ROM memory of the ST6245, and so the program and constants of the program can be easily modified by the user with the ST62E45 EPROM programming board from SGS-THOMSON.

From a user point of view (with the following exception) the ST62E45,T45 products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming of the ST62E45,T45 is described in the User Manual of the EPROM Programming board.

On the ST62E45, all the 3884 bytes of PROGRAM memory are available for the user, as all the EPROM memory can be erased by exposure to UV light. On the ST62T45 (OTP) device) a reserved area for test purposes exists, as for the ST6245 ROM device. In order to avoid any discrepancy between program functionality when using the EPROM, OTP and ROM it is recommended not to use these reserved areas, even when using the ST62E45.

Notes on programming:

In order to emulate exactly the ST6245 features with the ST62E45 and ST6245, some software precautions have to be taken:

1. I/O: To prevent floating input or uncontrolled I/O interrrupt on the EPROM/OTP devices, the port bits PA0-PA4 must be programmed as push-pull outputs.

2. When programming for the EPROM/OTP parts, it is suggested that the conditional assembly technique is used for controlling the I/O ports in order to disable the appropriate code for the ROM device.

3. Do not access data space locations CAh, DAh .

Other than this exception, the ST62E45,T45 parts are fully compatible with the ROM ST6245 equivalent, this datasheet thus provides only information specific to the EPROM based devices.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST6240 ROM-BASED DEVICE FOR FURTHER DETAILS.

EPROM ERASING

The EPROM of the windowed package of the ST62E45 may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST62E45 EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST62E45 package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.
The recommended erasure procedure of the ST62E45 EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST62E45 should be placed within 2.5 cm (1 inch) of the lamp tubes during erasure.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from:

. . . . . Tj = . . T_A + PD x RthJA

Where : . T_A = . . Ambient Temperature.

. . . . . RthJA = Package thermal resistance
. . . . . (junction-to ambient).

. . . . . PD = . Pint + Pport.

. . . . . Pint = . I_{DD} x V_{DD} (chip internal power).

. . . . . Pport = . Port power dissipation
. . . . . (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_{LCD}	Display Voltage	-0.3 to 11.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
I_O	Current Drain per Pin Excluding V_{DD} & V_{SS}	± 10	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PQFP52 CQFP52-W		70 70		°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Condition s	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version 6 Suffix Version	0 -40		70 85	°C
V_{DD}	Operating Supply Voltage		3		6	V
V_{LCD}	Display Voltage		3		10	V
V_{RM}	RAM Retention Voltage		2			V

RECOMMENDED OPERATING CONDITIONS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (1)(4)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01 0.01		8.388 2	MHz
I_{INJ+}	Pin Injection Current (positive) Digital Input (2) Analog Input (3)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input (2) Analog Input	V_{DD} = 4.5 to 5.5V			-5	mA

Notes :

1. An oscillator frequency above 1MHz is recommanded for reliable A/D results.
2. A current of ± 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device. For a positive current injected into one pin, a part of this current (~ 10%) can be expected to flow from the neighbouring pins. A current of -5mA can be forced on one input of the analog section at a time (or -2.5mA for all inputs at a time) without affecting the conversion.
3. If a total current of +1mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 1mA, all the conversion is resulting shifted of +1LSB. If a total positive current of +5mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 5mA, all the conversion is resulting shifted of +2LSB.
4. Operation below 0.01 MHz is possible but requires increased supply current.

EEPROM INFORMATION

The ST62xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	RESET, NMI, TIMER, WDON Pin			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	TIMER	$0.80V_{DD}$			V
		RESET, NMI, WDON Pin	$0.70V_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	RESET Pin V_{DD} =5V $V_{IN} = V_{DD}$ (1) $V_{IN} = V_{DD}$ (2) $V_{IN} = V_{SS}$ (5)			10 1 50	µA mA µA
V_{OL}	Low Level Output Voltage	TIMER, I_{OL} = 5.0mA			$0.2V_{DD}$	V
V_{OH}	High Level Output Voltage	TIMER, I_{OL} = -5.0mA	$0.65V_{DD}$			V

Notes on next page

DC ELECTRICAL CHARACTERISTICS (Continued)
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{PU}	Pull-up Resistor	V_{IN}=0V V_{DD}=5V WDON - NMI	40	100	200	kΩ
		RESET	100	300	500	kΩ
I_{IL} I_{IH}	Input Leakage Current	TIMER $V_{IN} = V_{DD}$ or V_{SS}		0.1	1.0	µA
I_{IL} I_{IH}	Input Leakage Current	NMI VDD=5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{IL} I_{IH}	Input Leakage Current	WDON VDD=5V $V_{IN} = V_{SS}$ (5) $V_{IN} = V_{DD}$			100 1.0	µA
I_{DD}	Supply Current RUN Mode	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		4	7	mA
	Supply Current WAIT Mode (4)	f_{OSC} = 8MHz, I_{LOAD} = 0mA V_{DD} = 5.0V		1	2	mA
	Supply Current RESET Mode	f_{OSC} = 8MHz, $V_{RESET} = V_{SS}$		1	7	mA
	Supply Current STOP Mode (3)(4)	I_{LOAD} = 0mA V_{DD} = 5.0V		1	10	µA

Notes :

1. No Watchdog Reset activated.
2. Reset generated by Watchdog.
3. When the watchdog function is actvated the STOP instruction is deactivated. WAIT instruction is automatically executed.
4. All on-chip peripherals in OFF state
5. Pull-up resistor

AC ELECTRICAL CHARACTERISTICS

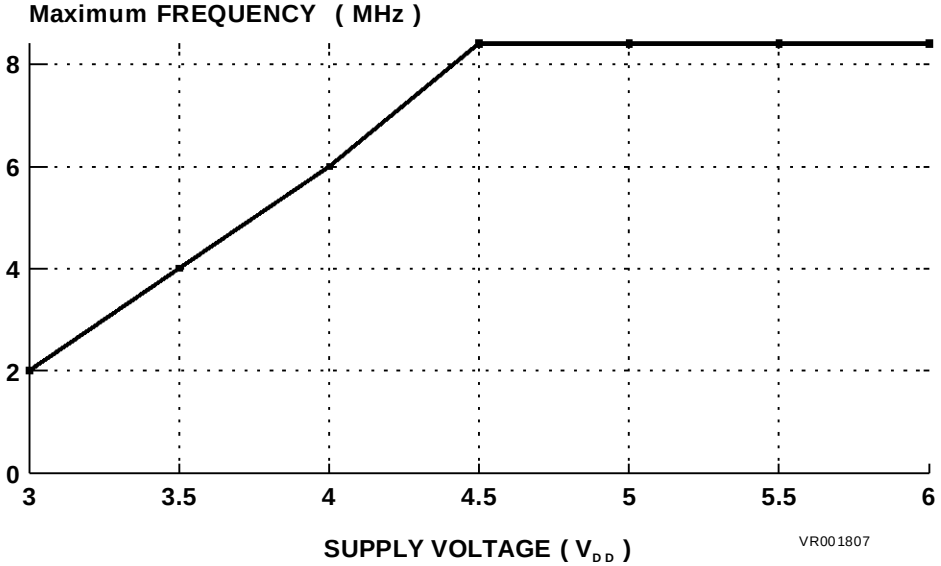
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency (2)	$V_{DD} \geq 4.5V$ $V_{DD} \geq 3V$	0.01 0.01		8.388 2	MHz
t_{SU}	Oscillator Start-up Time	$C_{L1} = C_{L2}$ = 22pF - crystal		5	10	ms
t_{SR}	Supply Rise Time	10% to 90%	0.01		100	
t_{REC}	Supply Recovery Time (1)		100			
T_W	Minimum Pulse Width	NMI Pin V_{DD} = 5V	100			ns
		RESET Pin	100			ns
T_{WEE}	EEPROM Write Time	T_A = 25°C One Byte T_A = 85°C One Byte		5 15	10 25	ms ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A LOT Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention	T_A = 55°C	10			years
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:

1. Period for which V_{DD} has to be connected or at 0V to allow internal Reset function at next power-up.
2. Operation below 0.01 MHz is possible but requires increased supply current.

Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



I/O PORTS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	I/O Pins			$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage	I/O Pins	$0.7V_{DD}$			V
V_{OL}	Low Level Output Voltage	I/O Pins, I_O = 10µA (sink)			0.1	V
		I/O Pins, $I_{OL} = V_{DD}$x1mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 1.6mA V_{DD} = 3V			0.4	V
	Low Level Output Voltage, PB4-PB7 Only	I/O Pins, $I_{OL} = V_{DD}$x2mA V_{DD} = 4.5 to 6V			$0.16xV_{DD}$	V
		I/O Pins, I_{OL} = 3.2mA V_{DD} = 3V			0.4	V
		I/O Pins, $I_{OL} = V_{DD}$x4mA V_{DD} = 4.5 to 6V			$0.26xV_{DD}$	V
		I/O Pins, I_{OL} = 6.4mA V_{DD} = 3V			0.8	V
V_{OH}	High Level Output Voltage	I/O Pins, I_O = -10µA (source)	V_{DD}-0.1			V
		I/O Pins, $I_{OL} = -V_{DD}$x1mA V_{DD} = 5.0V	$0.6xV_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	I/O Pins,(1)		0.1	1.0	µA
R_{PU}	Pull-up Resistor	I/O Pins V_{IN} = 0V, V_{DD} = 5.0V	40	100	200	kΩ

Note 1. Pull-up resistor off

SPI ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0V, T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
F_{CL}	Clock Frequency	applied on PB5/SCL			1	MHz
t_{SU}	Set-up Time	applied on PB6/Sin		50		ns
t_h	Hold Time	applied on PB6/Sin		100		ns

A/D CONVERTER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution (3)			8		Bit
A_{TOT}	Total Accuracy (3)	f_{OSC} > 1.2 MHz f_{OSC} > 32kHz			± 2 ±4	LSB
t_C (1)	Conversion Time	f_{OSC} = 8MHz		70		µs
V_{AN}	Conversion Range		V_{SS}		V_{DD}	V
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN} (2)	Analog Input Capacitance			2	5	pF
ASI	Analog Source Impedance				30	kΩ
SSI	Analog Reference Supply Impedence				2	kΩ

Notes:

1. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.
2. Excluding Pad Capacitance
3. Noise at $V_{DD}, V_{SS} \leq 10mV$

TIMER CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{12}{f_{OSC}}$			second
f_{IN}	Input Frequency on TIMER Pin				$\frac{f_{OSC}}{8}$	MHz
t_W	Pulse Width at TIMER Pin	$V_{DD} \geq 3V$ $V_{DD} \geq 4.5V$	1 125			µs ns

LCD ELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

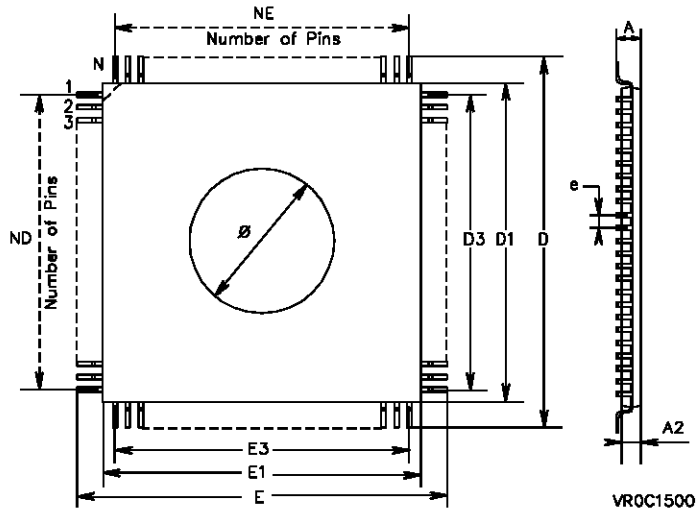
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{FR}	Frame Frequency	1/4 Duty f_{OSC} = 1, 2, 4, 8MHz	16		128	Hz
V_{OS}	DC Offset Voltage(1)	$V_{LCD} = V_{DD}$, no load			50	mV
V_{OH}	COM High Level, Output Voltage	I = 100µA V_{LCD} = 5V	4.5V			V
V_{OL}	COM Low Level, Output Voltage	I = 100µA V_{LCD} = 5V			0.5V	V
V_{OH}	SEG High Level, Output Voltage	I = 50µA V_{LCD} = 5V	4.5V			V
V_{OL}	SEG Low Level, Output Voltage	I = 50µA V_{LCD} = 5V			0.5V	V
V_{LCD}	Display Voltage	Note 2	3		10	V

Notes :

1. The DC offset voltage refers to all segment and common outputs. It is the difference between the measured voltage value and nominal value for every voltage level. Ri of voltage meter must be greater than or equal to 100MΩ.
2. An external resistances network is required when $V_{LCD} \leq 4.5V$.

PACKAGE MECHANICAL DATA

Figure 3.ST62E45 52 Pin Ceramic Quad Flat Package with Window



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		3.55			0.14	
A2		3.40			0.133	
D		17.20			0.677	
D1		14.00			0.551	
D3		12.00			0.472	
E		17.20			0.677	
E1		14.00			0.551	
E3		12.00			0.472	
Ø		7.62			0.3	
e		1.00			0.039	
Number of Pins						
ND	13					
NE	13					
N	52					

ORDERING INFORMATION TABLE

Sales Types	Memory type	Temperature Range	Package
ST62E45G1	4K EPROM	tested at 25°C only	CQFP52-W
ST62T45Q6	4K EPROM	-40 to + 85°C	PQFP52