SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

DESIGNING WITH MICROCONTROLLERS IN NOISY ENVIRONMENTS

INTRODUCTION

Microcontrollers (MCU) make possible the design of integrated and flexible controls for a constantly decreasing cost. As a result, they are spreading rapidly among most electronic applications and especially noise sensitive equipments such as for power control or automotive use.

An MCU operates with sequential logic, so the control of an application can be lost during a disturbance, as with analog control, but also after a power glitch in the system.

In addition, a modern MCU includes several tens of thousands of transistors switching in the MHz range, potentially radiating interference of high magnitude in a large frequency spectrum. Consequently, noise sensitivity and generation have to be considered as early as possible in MCU based designs.

This Application note presents numerous methods to effectively reduce noise problems. The first part presents a short overview on noise and proposes hardware solutions to increase the equipment immunity to noise. The second part concerns the writing of software more immune to disturbances. The behaviour versus disturbances of an MCU designed for noisy environments, the ST6210, is presented and practical examples and results are shown.

NOISE PREVENTION

The major noise receptors and generators are the tracks and wiring on the Printed Circuit Board (PCB), especially those near the MCU. The first actions to prevent noise problems thus concern the PCB layout and the design of the power supply.

Optimized PCB layout

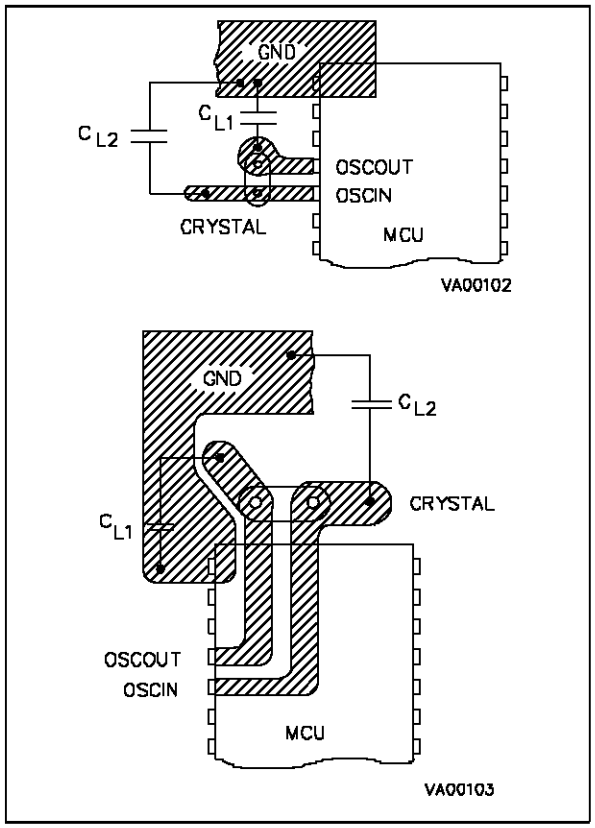
Noise is basically received and transmitted through tracks and components which, once excited, act as antennas. Each loop and track includes parasitic inductances and capacitances which radiate and absorb energy once submitted to a variation of current, voltage or electromagnetic flux.

An MCU chip itself presents high immunity to and low generation of EMI since its dimensions are small versus the wave lengths of EMI signals (typically mm versus 10's of cm for EMI signals in the GHz range). So a single chip solution with small loops and short wires reduces noise problems.

The initial action at the PCB level is to reduce the number of possible antennas. The loops and wires connected to the MCU such as supply, oscillator and I/O should be considered with a special attention (Figure 1). The oscillator loop has to be especially small since it operates at high frequency.

A reduction of both the inductance and the capacitance of a track is generally difficult. Practical experience suggests that in most cases the inductance is the first parameter to be minimized.

Figure 1. PCB Board Oscillator Layout Examples



The reduction of inductance can be obtained by making the lengths and surfaces of the track smaller. This can be obtained by placing the track loops closer on the same PCB layer or on top of one another (Figure 2). The resulting loop area is small and the electromagnetic fields reduce one another.

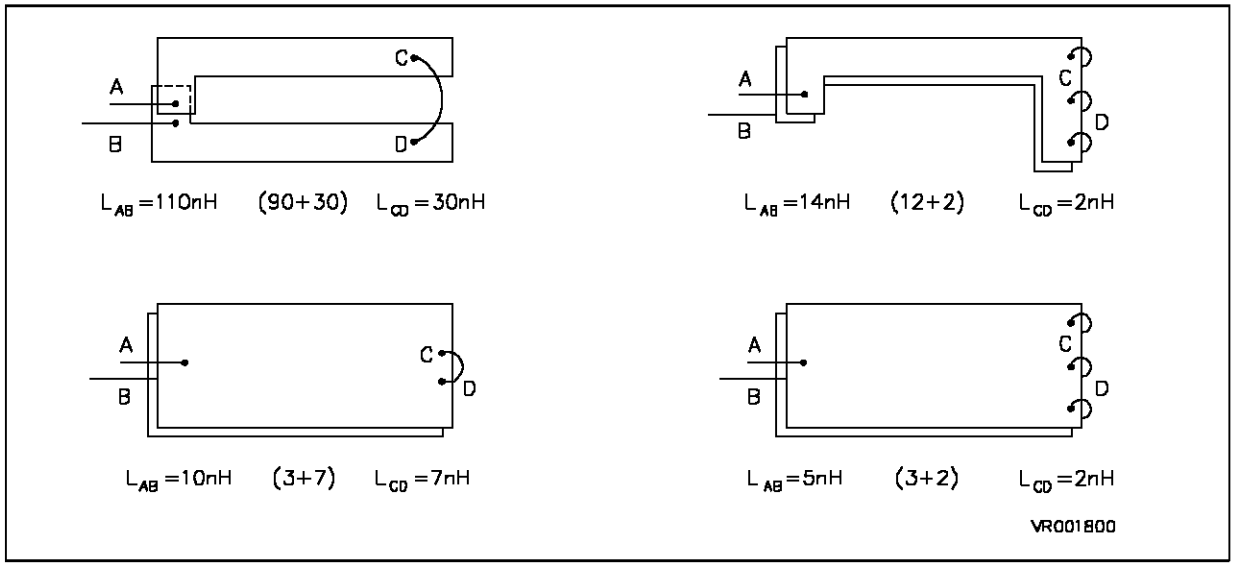
The ratio in order of magnitude relating to the inductance value and the area defined by the wire loop is around 10nH/cm^2. Typical examples of low inductivity wires are coaxial, twisted pair cables or multiple layer PCBs with one ground and one supply layers. The current density in the track can also be smaller due to track enlargement or the paralleling of several small capacitances mounted in the current flow.

In critical cases, the distance between the MCU and the PCB, and therefore the surfaces of the loops between an MCU and its environment, has also to be minimized. This can be achieved by removing any socket between the MCU package and the PCB, by replacing a ceramic MCU package by a plastic one or by using Surface Mounting instead of Dual In Line packages.

Power supply filtering

The power supply is used by all parts of the circuit, so it has to be considered with special attention. The supply loops have to be decoupled to make sure that signal levels and power currents do not interfere. These loops can be separated using star

Figure 2. Reduction of PCB Tracks Loop Surfaces



Note: This test is done with a double sided PCB. Insulator thickness is 1.5mm, copper thickness is 0.13mm. The overall board size is 65 x 200mm.

wiring with one node designated as common for the circuit (Figure 3).

The decoupling capacitance should be placed very close to the MCU supply pins to minimize the resultant loop. It should be also large enough to absorb, without significant voltage increase, parasitic currents coming from the MCU via the input protection diodes. The decoupling of the board can be done with electrolytic capacitors (typ. 10µF to 100µF) since the dielectric used in such capacitors provides a high volumic capacitance. However these capacitors behave like inductances at high frequency (typ. above 10MHz) while ceramic or plastic capacitances keep a capacitive behaviour at higher frequency. A ceramic capacitance of, for instance, 0.1µF to 1µF should be used as high frequency supply decoupling for critical chips operating at high frequency.

The supply circuit must be sized in such way that its components can absorb energy peaks during supply overvoltages. For instance, in a power supply done with a capacitor in series between the mains supply and the MCU supply (typically +5V), this capacitance is a short circuit when a voltage spike occurs. The corresponding short circuit current has to be absorbed by a protection zener diode. Depending on the maximum energy to withstand, a standard 0.5W Zener diode (e.g. BZX55C) may have to be replaced by a 1.3W (BZX85C) or 2W (BZV47C) Zener diode (Figure 15). Additional filtering with serial resistance or inductance can be included to reduce the influence of voltage spikes and to absorb transients coming from the input supply line.

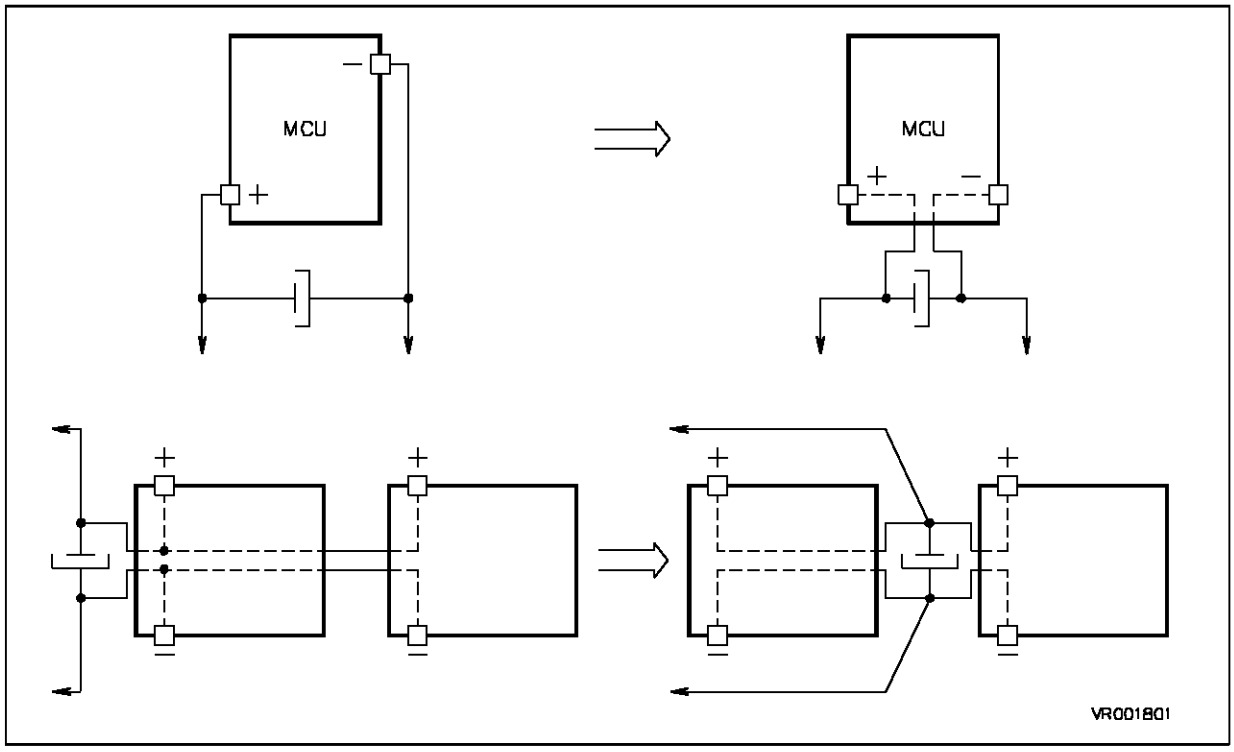
I/O configuration

In general, the smaller the number of components surrounding the MCU, the better the immunity versus noise. A ROMless solution, for instance, is typically more sensitive to and a bigger generator of noise than an embedded circuit.

If the output buffers are embedded in the MCU, their switching speed has to be controlled in order to avoid parasitic oscillations when they are switching. A trade-off between noise and speed has to be found by the MCU designers.

I/O pins which are not used in the application should be preferably grounded or connected via a large impedance (i.e. 100kΩ) to a fixed potential, depending on the MCU reset configuration. Here, the trade-off is between immunity and consumption.

Figure 3. Supply Lay-out Examples



If a current can be forced in an input pin, clamping protection with diodes has to be included in the circuit connected to the pin to divert the current from the MCU structure and to avoid the risk of latch-up (Figure 4). In an MCU such as the ST62, these diodes are integrated inside the chip.

Shielding

Shielding can help in reducing noise reception and emission, but its success depends directly on the material chosen as shield (high permeability, low resistivity) and on its connection to a stable voltage source including a decoupling capacitance via a low serial impedance (low inductance, low resistance).

If the generator of major disturbances is near to the MCU board and can be identified as a strong dV/dt generator (i.e. a transformer or Klystron), the noise is carried mainly by the electrostatic field. The critical coupling between the noise generator and the control board is capacitive. A highly conductive shield (i.e. copper) creating a Faraday cage around the control board may strongly increase the immunity.

If the strongest source of perturbations is a dI/dt generator (i.e. a relay), it is a high source of electromagnetic fields. Therefore, the permeability of the shielding material (i.e. alloy) is crucial to increase the immunity of the board. In addition, the number and size of the holes on the shield should be reduced as much as possible to increase its efficiency.

In critical cases, the implantation of a ground plane below the MCU and the removal of sockets between the device and the PCB can reduce the MCU noise sensitivity. Indeed, both actions lead to a reduction of the apparent surface and loop between the MCU, its supply, its I/O and the PCB.

WRITING SAFER SOFTWARE

The hardware solutions described help inreducing the noise received and radiated by the MCU. Nevertheless if the MCU is disturbed, a modification of a register, for instance the program counter, can occur. In this case, control of the application should not be lost.

Writing safe software can prevent most of the problems due to parasitic modifications of registers or program counter. Many register errors can be quickly identified and masked in the program flow without influence on the environment.

The examples and indications given in this section are written for the ST62 MCU family.

The basic precepts for the writing of a safe software are:

-Test only configurations clearly defined in the flow chart

-Regularly check vital data stored in RAM

-Control the program flow

-Fill the unused memory

It is useful for the program to identify to itself that it is following the correct program flow. This can be implemented by using trace points held in several bits of a specific register (Figure 5). If more bit flags are included in the decision than strictly necessary,

Figure 4. Standard MCU I/O Block Protection

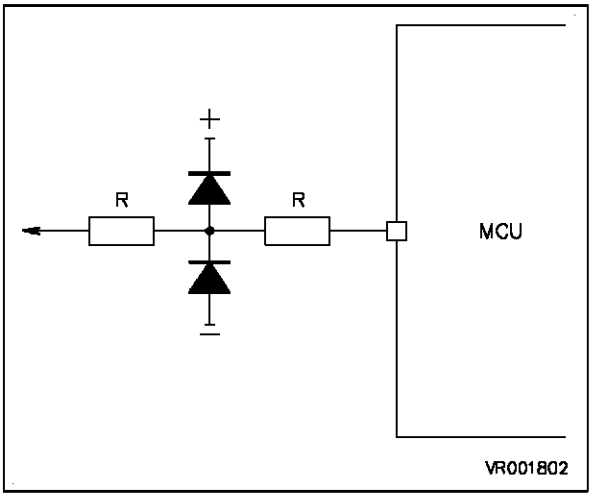
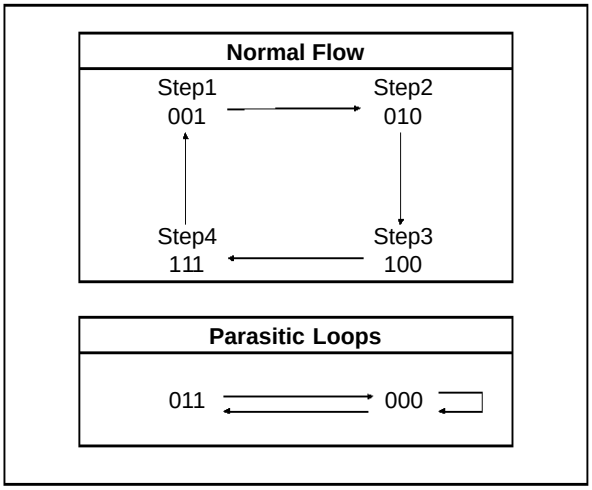


Figure 5. Trace Point of a Program Flow



the program may enter in a mode which it never leaves. Spikes may set an unused bit, so an error can be generated and an exit should be defined for every trace point condition.

The trace points can be used also to control the flow of the tasks. In such a case, a "called" task can be only called by a "calling" task. Such checks can be done at the beginning and at the end of each task.

The program flow can be also monitored by controlling the duration of a subroutine, for instance, by reading a timer value at the beginning and at the end of the task.

The contents of the data RAM may be changed by noise, therefore it is good practice to check regularly the consistency of vital data. For example, the fact that the RAM value is inside a predefined interval, its coherence with a value previously stored, a checksum or a comparison with a copy (inverted or not) can be checked. Constant values can be stored in non-used RAM addresses and regularly checked to make sure that the RAM data are not disturbed. The control registers of the MCU peripherals (used or not) can also be reinitialized regularly inside the main routine.

Unused parts of the program memory can be filled with the NOP instruction plus a reset of the chip at the end of the unused area (LDI WDT,01h). If the program counter is modified after a glitch and sends the program to the unused area, the program will not hang in an endless loop and the core is finally reset.

Example:

unused area

```
04h          ; NOP
04h          ; NOP
....
04h          ; NOP
LDI WDT,01h ; generate reset
             ; of chip
```

end of unused area

The unused part of the program memory can also be filled with the ST62 JP X9X instruction since this instruction has its two hexadecimal bytes identical (hex instruction code X9X9). If an unforseeen disturbance to the program counter sends the program to the unused area, the program immediately jumps to address X9X. This address can be at the beginning of the program (090 for 4kROM, 898 for 2kROM versions) or a reset instruction (LDI WDT,01h).

The Watchdog should be refreshed a minimum of times and in the main loop. Its refresh value can be calculated in order to minimize the reload value and therefore the duration without a potential watchdog reset. In addition, the user has to make sure that the main routine is executed from time to time. The Watchdog should never be used for tests in the main routines, especially not as a timer.

Additional flags and trace points in the subroutines can be used to check that the program path is correct before reloading the watchdog. Instead of refreshing the watchdog with a constant value using a LDI instruction (i.e. LDI WDT,#0FFh), the refresh value can be preselected or calculated depending on the trace point TP, using the accumulator A (i.e. LD A,TP and LD WDT,A).

The program presented in annex 1 has been written for the ST621x/2x. It checks the flags, the trace points and adjusts the watchdog refresh value. It is written in such way that the watchdog is reloaded only in the main loop and not in a subroutine or an interrupt routine. If the watchdog has to be reloaded out of the main loop, the application safety is reduced and this example has to be modified. It can be implemented in applications which can start again after a reset and where the reset configuration of the MCU I/O pins may occur without damage in any step of operation of the equipment.

The ROM content has also to be checked in order to avoid data combinations where the watchdog register may be written unintentionally. This can occur if a byte follows another byte which, read as an instruction, can modify the watchdog, and if the program counter is corrupted. For instance in the ST62, the watchdog address byte (D8) is the same as the JRNZ instruction.

Example:

Initial version:

```
CPI A,#0D    =370D
JRNZ OUTloop =D8nn (0DD8 sequence
                    in program)
```

Modified version:

```
CPI A,#0D     =370D
NOP           =04
JRNZ OUTloop =D8nn
```

The following table lists the critical bytes not to be placed before this byte.

Two Successive Bytes	Equivalent Instructions
0D D8	LDI WDT
2B D8	RES 4,WDT
3B D8	SET 4,WDT
4B D8	RES 2,WDT
5B D8	SET 2,WDT
6B D8	RES 6,WDT
7B D8	SET 6,WDT
9B D8	SET 1,WDT
AB D8	RES 5,WDT
BB D8	SET 5,WDT
CB D8	RES 3,WDT
DB D8	SET 3,WDT
EB D8	RES 7,WDT
FB D8	SET 7,WDT
7F D8	INC WDT
9F D8	LD WDT,A

If the program flow is such that the watchdog register byte address follows one of the critical bytes listed, the watchdog contents can be corrupted. The solution to this problem can be either to modify the first byte i.e. by changing the data RAM location (if used) or to insert a NOP instruction between the 2 critical bytes.

In addition, if possible, the data in the X or Y index registers should never be identical to the WDT address.

Operation may also be disturbed due to noise on input lines. All inputs can be digitally filtered, so that an input (analog or digital) is valid only if it remains constant for a defined time. This reduces the number of passive components.

Example:

```
Main1 LDI loop,04h
Main2 JRR 4,PB,Main1; continue flow
                    ; if PB4=0
      DEC loop    ; for 4 successive
                  ; measurements
      JRNZ Main2
Main3 LDI loop,04h
Main4 JRS 4,PB,Main3; continue flow
                    ; if PB=1
```

```
DEC loop    ; for 4 successive
            ; measurements
JRNZ Main4
```

ST62, AN MCU FAMILY DESIGNED FOR NOISE IMMUNITY

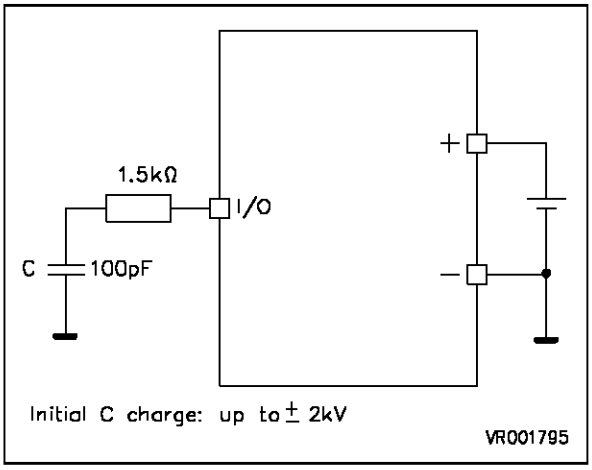
This section presents some technology and design solutions used in the ST62 MCU family to enable safe operation when used in products in disturbed or noise sensitive environments (Figure 6).

High destruction limits

Destruction of an MCU is usually due to Electrostatic discharge (ESD), a peak voltage or latchup which causes uncontrolled current to flow in the chip and to concentrate in some parts of the structure where a high voltage is applied. The common action of the current and voltage is the creation of hot spots which burn the silicon of the device.

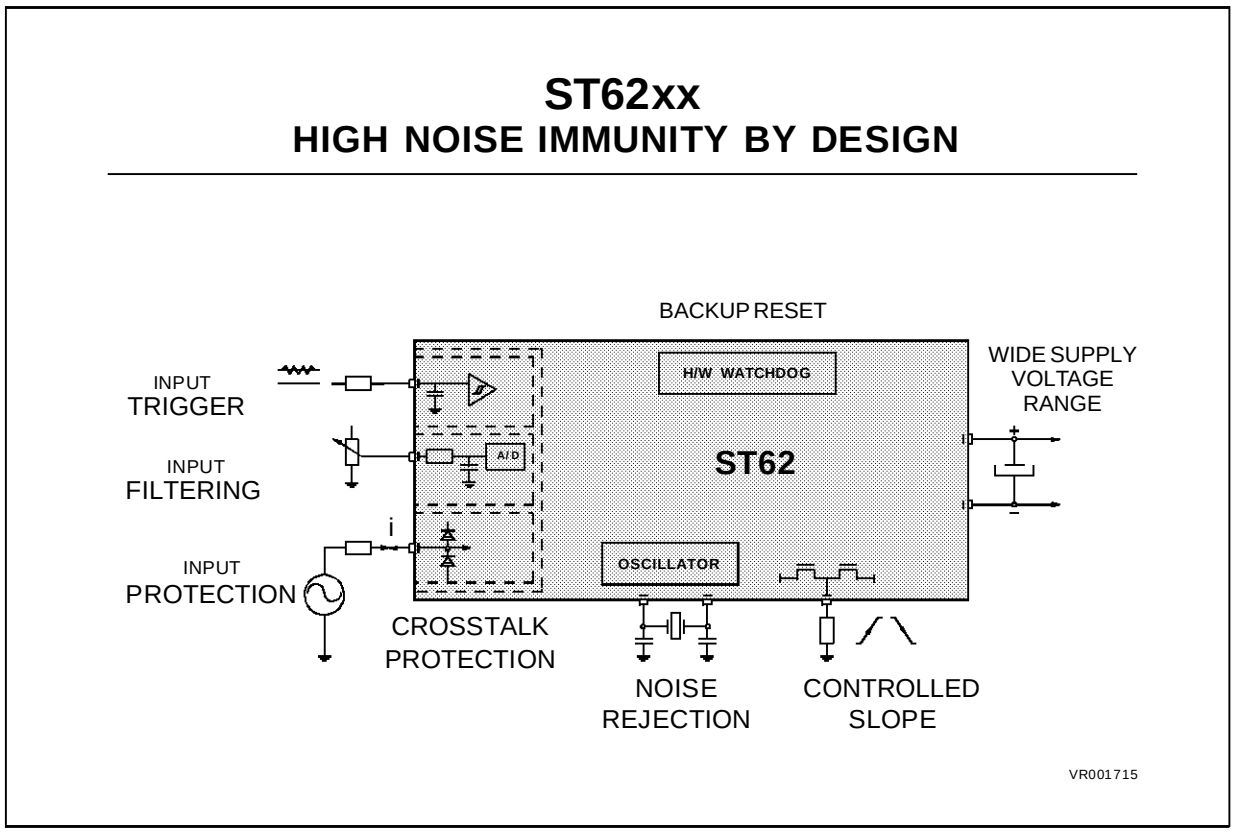
Such defects mechanisms are modelled and corresponding tests are applied on the chips. The ESD test simulates the action of electrostatic energy stored in the parasitic capacitance of a person, which is discharged in the chip. It is modelled by standards such as MIL STD 883.5 (Figure 7).

Figure 7. ESD Test Schematic



The latch-up test determines ruggedness of the device to overvoltage and current injection. An SGS-THOMSON corporate quality specification defines the test procedure. In the first test, an

Figure 6. Major ST62 Features Increasing its Noise Immunity



overvoltage higher than the maximum specified rating is applied on the supply pins. For the second test, a high current pulse is injected in I/O pins of a device supplied normally. In both tests, latch-up is observed by measuring the supply (and I/O pin) current and by making sure that no discontinuity occurs in the current growth (Figure 8).

When the MCU is used inside its specified characteristics with normal handling precautions, such defects mechanism should not occur.

High noise immunity

Several optimization technics have been implemented in the technology and design of the ST62 to minimize its sensitivity to external disturbances.

Voltage potential wells have been integrated between the I/O cells and between other I/O and logic cells, avoiding noisy line influences on other MCU blocks. Protection diodes are included inside each I/O pin, the timer and the NMI cells. If the current in these diodes is limited with external resistances, the diodes can be used functionally, providing that the total current in the supply is also limited. Typical values of the diode current for the ST6210 are 2.5mA per I/O, 0.5mA for NMI/timer and 25mA total.

Schmitt triggers are included in each input to filter noisy signals. The hysteresis levels of comparison on the digital inputs are typically 3.5V for level “1” and 1.5V for level ”0” with a +5V supply.

Capacitances are included in the pads (typ. 5pF) to provide a minimum of filtering if an external resistance is connected. These capacitances are internally associated to resistances to avoid capacitive coupling. The A/D converter also includes its own filter to help stabilizing the input signal during the conversion (Figure 9).

The wide supply voltage range between 3V and 6V allows the ST6210 to operate safely inside these limits even if the voltage is not stable, providing that the oscillator frequency is compatible with the voltage (Figure 10).

Figure 8. Latch-up Test Schematics

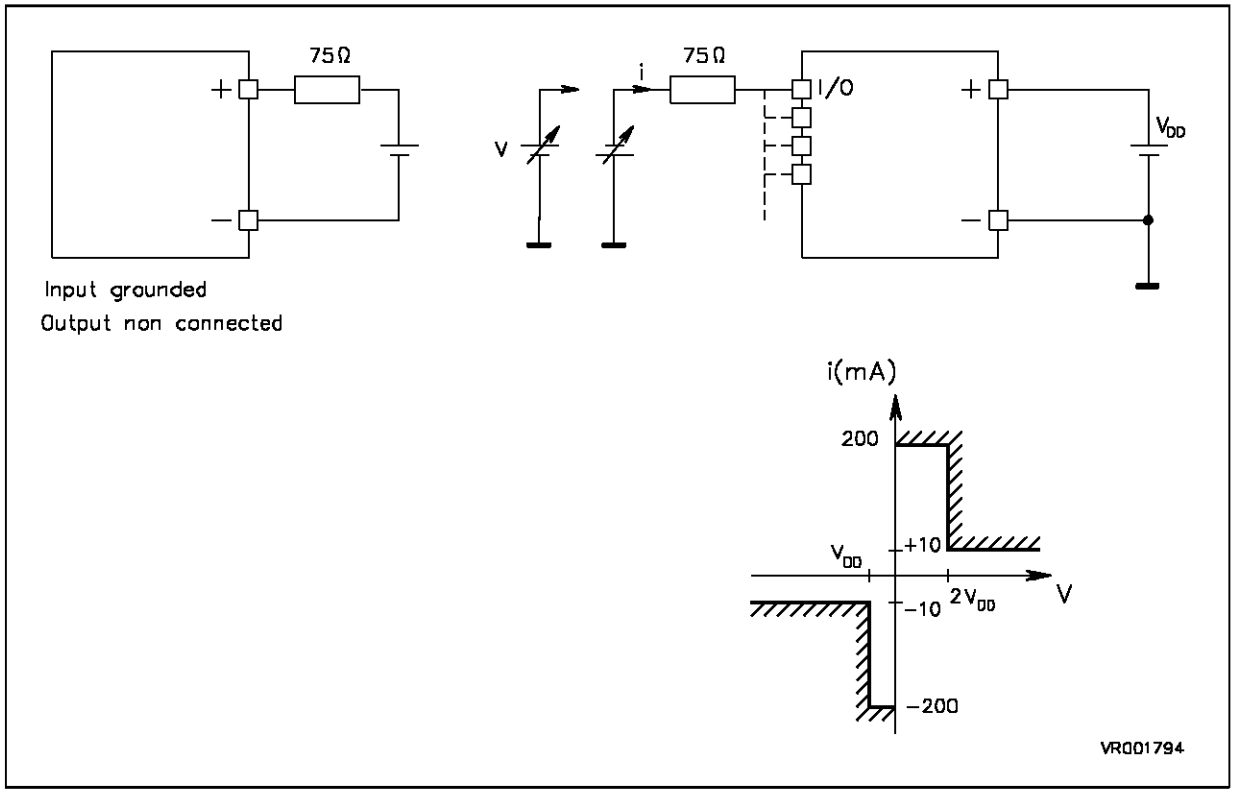


Figure 9. A/D Converter Input Schematic on ST6210

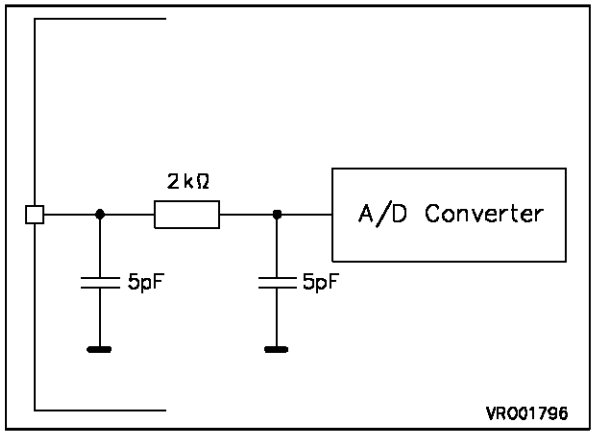


Figure 10. Relation Between Oscillator Frequency and Supply Voltage on ST6210

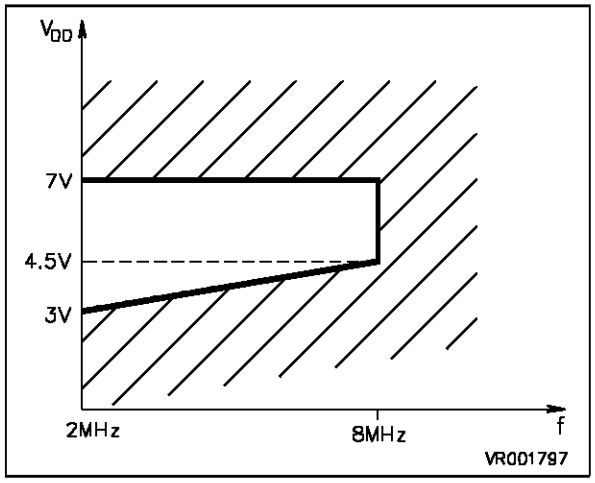
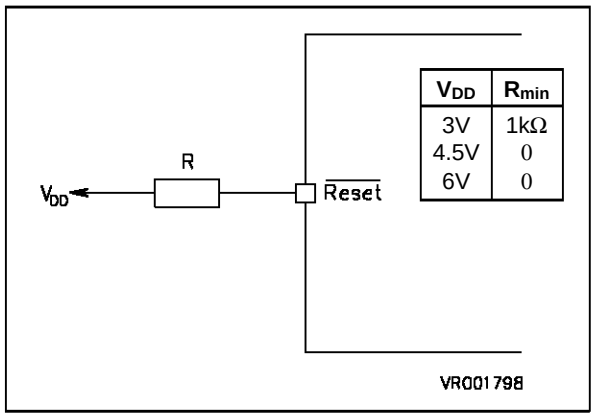


Figure 11. Reset Pin Connection of the ST6210



V_{DD}	R_{min}
3V	1kΩ
4.5V	0
6V	0

The V_{DD}, V_{SS} and oscillator pins of the ST6210 are close to each other. In this way, the surface of the most critical loops is minimized.

Low noise generation

High current buffers (20mA typ.) are included in the outputs of the ST62 MCU. The output edges are slowed down in order to avoid over-oscillation at the commutations (typ. 10ns switching time). This is especially useful when inductive loads are driven with the 20mA ports, in parallel or not.

The internal databus of the ST62 CPU is serial, meaning that only few transistors switch at the same time (typically 1/13th versus an MCU with a parallel bus). The radiated spectrum and the noise on the supply are reduced compared to a parallel architecture running at the same oscillator frequency.

Reset modes

A strong glitch or a power line failure may stop or strongly disturb the operation of the program. In such case, the MCU has also to recover safely. This is achieved via a hardware reset. With ST621X such a reset can come from an external pin, the watchdog or the power-on-reset (POR) block.

The reset pin allows the reset from an external component, i.e. a voltage regulator L4947. If this pin is not used and with a +5V supply, the reset pin of the ST6210 can be directly connected to Vdd, providing therefore a high noise immunity on this pin (Figure 11).

After the reset, the I/O configuration has to be checked in order to avoid problems such as short circuits or parasitic drive of external components before the software initialisation. In addition, a system status has to be made to make sure that the program will not restart in a bad step of the process. If necessary, the process can be forced to a clear configuration at the software initialisation phase.

Watchdog. If the program counter is disturbed and the program lost in a loop where the watchdog is not reloaded, the watchdog counts down to zero and resets the MCU in a similar way as the external reset pin. When the program comes out of a loop, an exit condition can be checked and if the condition is not met, the watchdog is activated. An example of reset by watchdog activation is given in Annex 1. In any case, a watchdog reset should never happen in normal operation.

For a safe operation in noisy environment, the user should use a “hardware” watchdog. This circuit is activated when the MCU is supplied with power and when the oscillator runs. Once activated, it can not

be deactivated by any means. A “software” activated watchdog can be chosen when a low power consumption mode is required but it does not provide the same level of safety. This watchdog, once initialized by the software, has the same behaviour as the hardware activated watchdog and can not be deactivated by the program. However, until it is activated there is no watchdog protection.

The two versions of the watchdog (“software” and “hardware” activated) are available on the ST62 MCU.

An embedded counting watchdog can be replaced or doubled with an external analog watchdog designed with a resistance and a capacitor connected to the reset pin. In normal operation, the capacitance is discharged through an I/O port of the MCU at a rhythm defined by the software. The reset occurs if the oscillator stops or the program does not go through the corresponding I/O port drive.

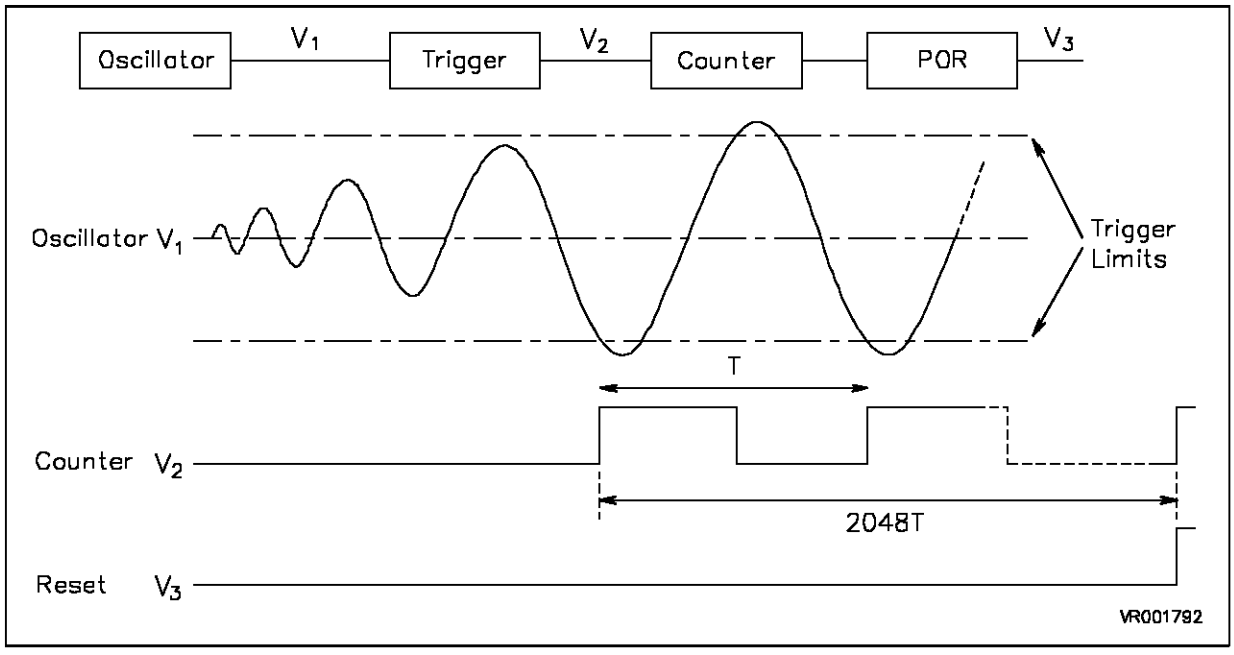
Power On Reset (POR). In the ST6210, both the watchdog and the POR blocks participate to a safe start. When the supply voltage grows above 0.7V to 1V, the oscillator starts. Depending on the type of oscillator (RC, crystal, ceramic resonator), its startup lasts around 2ms to 10ms. Once the oscillator voltage reaches the trigger limits, a clean signal is available and the counter counts 2048 clock periods to ensure a full and valid reset of the ST62. The POR then allows the CPU to exit from the reset state (Figure 12).

Since there is not a precise voltage source inside CMOS technology products and considering that the oscillator startup can vary strongly from one type of oscillator to another, the simplest approach for the user is to make sure that the supply has reached its nominal level 2048 clock periods after the start. In applications supplied directly from the mains, a capacitive supply enables a very fast voltage growth while a resistive supply slows it down.

Disturbances on the supply.

By design, the minimum voltage for watchdog operation is lower than for the CPU (typically 3.5V versus 4V at 8Mhz). So if the supply voltage does

Figure 12. Power-On-Reset (POR) Timing



not decrease below, for instance 3.5V, the watchdog resets the CPU when it counts down to zero. If the supply goes down below 3.5V, both the CPU and the watchdog are stopped. The watchdog restarts when the voltage increases up and resets the CPU when it counts down to zero.

If the supply voltage drops down below 0.7 to 1V, the POR acts when the supply rises again. Then two resets can occur, coming from the POR and the watchdog (Figure 13).

If the supply voltage changes, its speed of variation is normally limited by the decoupling capacitance. If the voltage variations remain inside the limits specified for the given oscillator frequency, the ST6210 CPU operation will not be disturbed.

The ST6210 includes an A/D converter, allowing additional supply voltage monitoring to be achieved using an external Zener diode (Figure 14). The circuit consumption is slightly increased due to the

Figure 14. Supply Monitoring via The A/D Converter

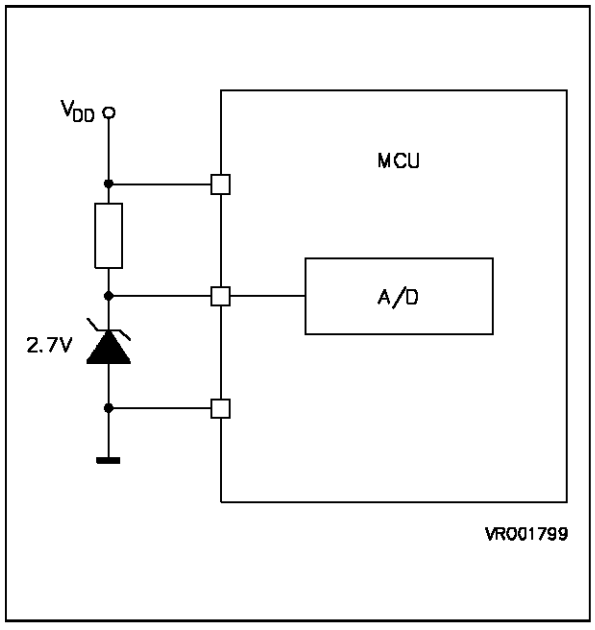
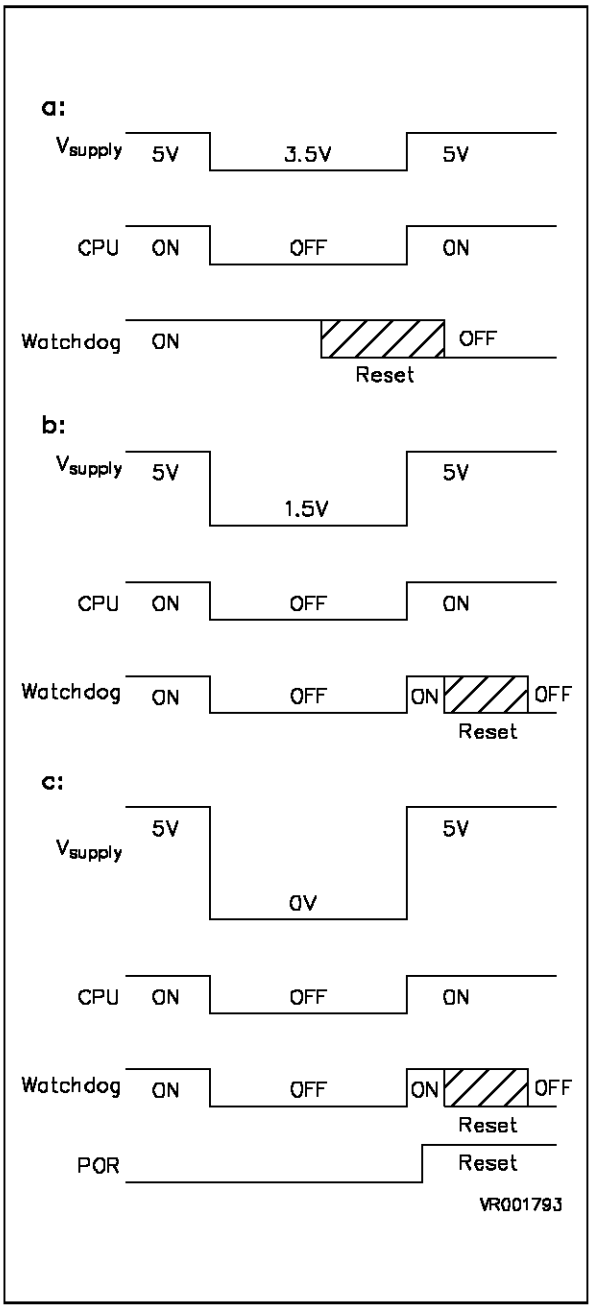


Figure 13. Reset Sequences after Power Disturbances



polarization of the diode. With the A/D converter, the supply level can be accurately measured and a back-up procedure can be decided if the converted value increases above a certain limit.

EXPERIMENTAL RESULTS

The noise immunity of a ST6210 can be tested in a functional manner. One input is forced with high current when the neighbouring input pin is connected to a potentiometer. All other pins are connected in output mode to LEDs. The program converts the analog value to a LED display. When a current is forced in a pin close to the functional pins, no defect appears on the display.

This clamping feature included in the I/O pins can be also used for detection of the mains zero crossing. The mains voltage is directly connected to the ST6210 via a high impedance which limits the current. The internal I/O diodes clamp the signal and the I/O works safely without external diode networks (Figure 15).

Noise generation can be tested using a TEM cell. Such an antenna is a type of coaxial cable with space available inside to put the equipment under test (Figure 16). It can be used either as a noise generator to check the ruggedness of an equip-

Figure 15. Power Control Using ST6210 and Snubberless Triac

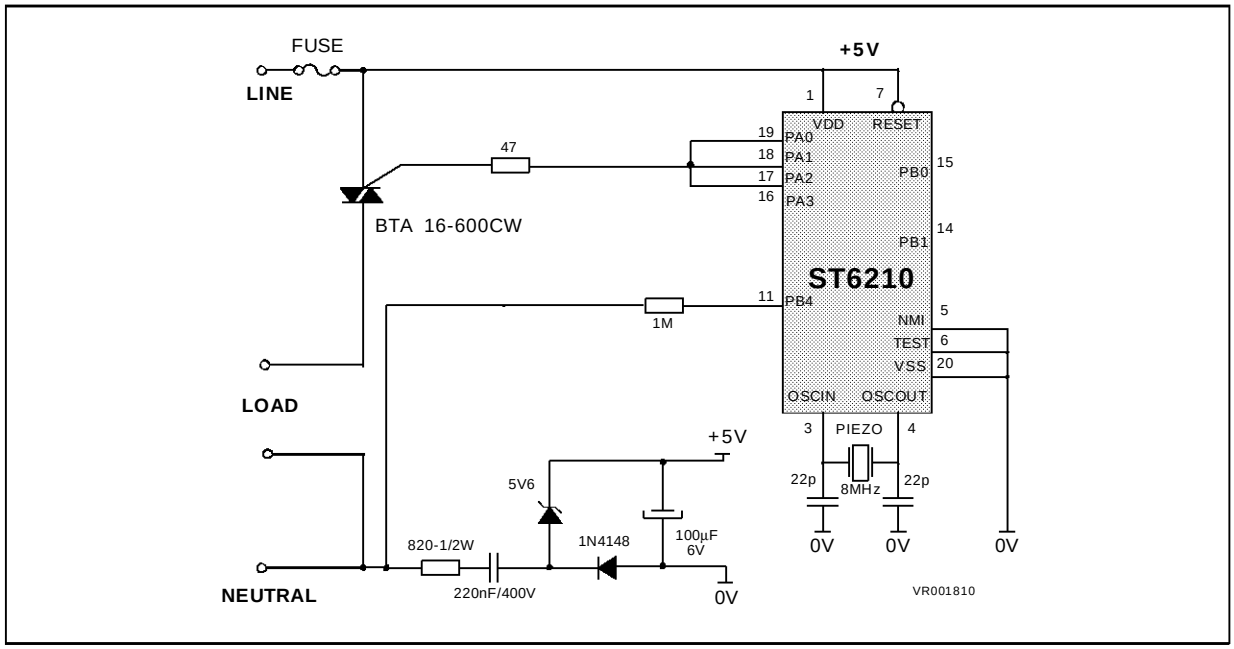
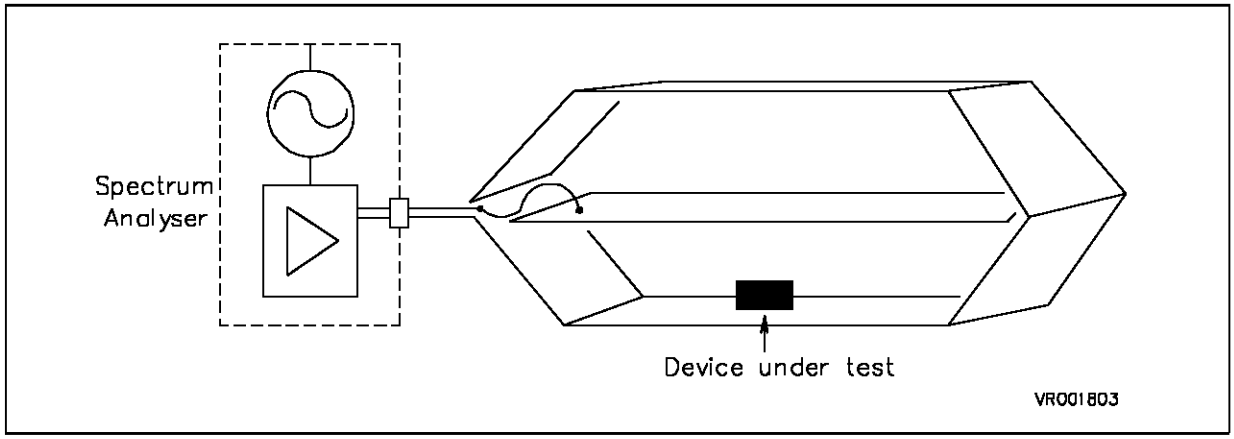


Figure 16. The TEM Cell in Receiver Mode



ment versus EMI or as a receiver to measure the EMI generated by an equipment. Such test equipment is much less expensive than an anechoic chamber.

SUMMARY

Microcontrollers (MCU) are spreading out from applications well protected against noise such as telecom or computer, to noisy environment such as automotive and power control.

Protection against noise goes through the choice of an adapted MCU. The ST62 family, for instance, has been designed to operate safely in disturbed or noise sensitive environment. Its major design characteristics concern the I/O design, the reset modes and the supply voltage range.

Even when using an MCU designed for noisy environment, special care has to be taken during the design on the circuitry, on the PCB lay-out and on the writing of the software. This article presents some concrete solutions applicable to these fields.

With caution on these points, the designer can use MCU's for applications such as motor control, battery charger, light dimming or alarm. And here the real advantages of an MCU can be taken: fast time to market, high flexibility with a minimum of components on the board and treatment of relatively complex algorithms.

Bibliography:

US patent: An integrated Controlled FET switch
(J.Stockinger/SGS-THOMSON Microelectronics)

Power semiconductors and EMI reduction
(L.Perier/Powertechnics 1-91)

Environmental design rules for Mosfet
(B.Maurice/Power transistor manual 91/
SGSTHOMSON)

Program Example

ANNEX 1

This program checks the flags, the trace points and adjusts the watchdog refresh value. It is organized in such way that the watchdog is reloaded only in the main loop and not in a subroutine or an interrupt routine. It is written for the ST621x/2x MCU. It can be implemented in applications which can start again after a reset and where the reset configuration of the MCU I/O pins may occur without damage in any step of operation of the equipment.

```
.def        WDT  0d8h     ; WDT = watchdog timer address
.def        WDM  090h     ; WDM = watchdog trace mask in RAM e.g. reg 090h
      ; WDM is also the WDT refresh value in normal mode main program loop:
            ...
            ...
            LD  A,WDM                ; read last trace mask
            CPI A, last tracepoint   ; check last tracepoint value
            JRZ  cont1          ; continue if correct, else reset chip
            LDI WDT,01h         ; reset chip if the test fails
            set 1,WDM           ; set bit 1 on WDM (WDM=2d)
cont1:      RET                 ; RET will work like a NOP, if executed in
            RET                 ; the main loop it is used to be sure
            RET                 ; that stack is in the top position.
            RET                 ; The stack has 6 levels hardware stack
            RET                 ; so unnecessary RET are seen as NOP.
            RET
            RETI                ; switches normal flags back
                                ; RETI would cancel the interrupt
                                ; So to be sure not to be
                                ; in interrupt mode
            JRNZ contm          ; check that Zero flag still the same
                                ; JRZ is used in alternance with JRNZ for jump
                                ; to detect if Zero flag is stuck in one level
            LDI WDT,01h         ; reset chip if the test fails
contm:      JRNC contn          ; check that Carry flag still cleared
            LDI WDT,01h         ; reset chip if the test fails
contn:      LD A,WDM            ; load refresh value to WD
            LDI WDM,01h         ; set trace register to initial
                                ; reset position (WDM=1d)
```

Program example (Continued)

```
            CPI A,WDMASK      ; WDMASK stored in ROM for double check
            JRZ conto         ; continue if stored and calculated
                              ; values identical
            LDI WDT,01h       ; reset chip if test fails

conto:      LD  WDT,A         ; refresh the watchdog only here
                              ; the last tracepoint before the normal
                              ; watchdog refresh is the next value of
                              ; the watchdog timer itself
            ...               ; continue with normal program flow

            ...
            LD A,WDM          ; first tracepoint in the program flow
                              ; may be in a subroutine or
                              ; interrupt routine
            CPI A,01h         ; test initial value
            JRZ  cont1        ; reset if not valid
            LDI WDT,01h       ; reset chip if the test fails
cont1:      SET 2,WDM         ; set bit 2 (WDM=5d)
            ...               ; continue with normal program flow
            ...
            LD A,WDM          ; second tracepoint in the program flow
                              ; may be in a subroutine or
                              ; interrupt routine
            CPI A,05h         ; test preceding value
            JRZ cont2         ; reset if not valid
            LDI WDT,01h       ; reset chip
cont2:      SET 3,WDM         ; set next or other bit of WDM (WDM=13d)
                              ; SET,RES combinations for generation
                              ; of binary codes for more than 6 tracepoints
                              ; may be used
            ...               ; continue to normal program flow
```

THE SOFTWARE INCLUDED IN THIS NOTE IS FOR GUIDANCE ONLY. SGS-THOMSON SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM USE OF THE SOFTWARE.