AN01054

Smart Card Interface using TDA8007BHL/C2/C3

Rev. 1.1 — 27 March 2012 **Application note**

Document information

Info	Content
Keywords	TDA8007BHL/C2, TDA8007BHL/C3, Smart card interface, Protocol T = 0 and T=1, Double smart card reader, Three smart card reader
Abstract	This application note is a generic documentation given the basic rules to use the TDA8007BHL/C2 or C3 and handles a communication between a system controller and two or three smart cards. The TDA8007B functions are controlled by micro-controller through a parallel interface in both modes, multiplexed and non multiplexed.

Revision history

Rev	Date	Description
1.1	20120327	Fig 9 updated and PRES external resistor calculation added
1.0	20110707	First version

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

The TDA8007B is a low cost card interface for dual smart card readers. Controlled through a parallel bus, it takes care of all ISO 7816, EMV and GSM11-11 requirements. It may be interfaced to the P0/P2 ports of a C51 family micro-controller, and be addressed as a memory through MOVX instructions. It may also be addressed on a non multiplexed 8 bits data bus, by the means of registers addresses AD0, AD1, AD2 and AD3. The integrated ISO UART and the timer counters allow easy use even at high baud rates with no real time constraints. Due to its chip select external I/O and interrupt features, it simplifies a lot the realization of any number of card readers. It gives the cards and the reader a very high level of security, due to its special hardware against ESD, short-circuit, power failure, etc. Its integrated step-up converter allows operation within a supply voltage range of 2.7 to 6V.

The schematic diagram and the layout for dual cards interface is given as example in Annex.

2. General description

2.1 Features

- Control and communication through an 8 bits parallel interface, compatible with multiplexed or non multiplexed memory access.
- Specific ISO UART with parallel access on I/O for automatic convention processing, variable baud rate through frequency or ratio programming, error management at character level for T=0, extra guard time register.
- 1 to 8 characters FIFO in reception mode.
- Automatic activation and deactivation sequence through an independent sequencer.
- 24 bits timers counter for ATR and waiting time processing.
- Current limitation in case of short circuit.
- Supply voltage from 2.7 to 6 V
- Power down mode for reducing current consumption when no activity.
- Manual or automatic character retransmission in case of parity error.

Block diagram of this circuit is presented next page.

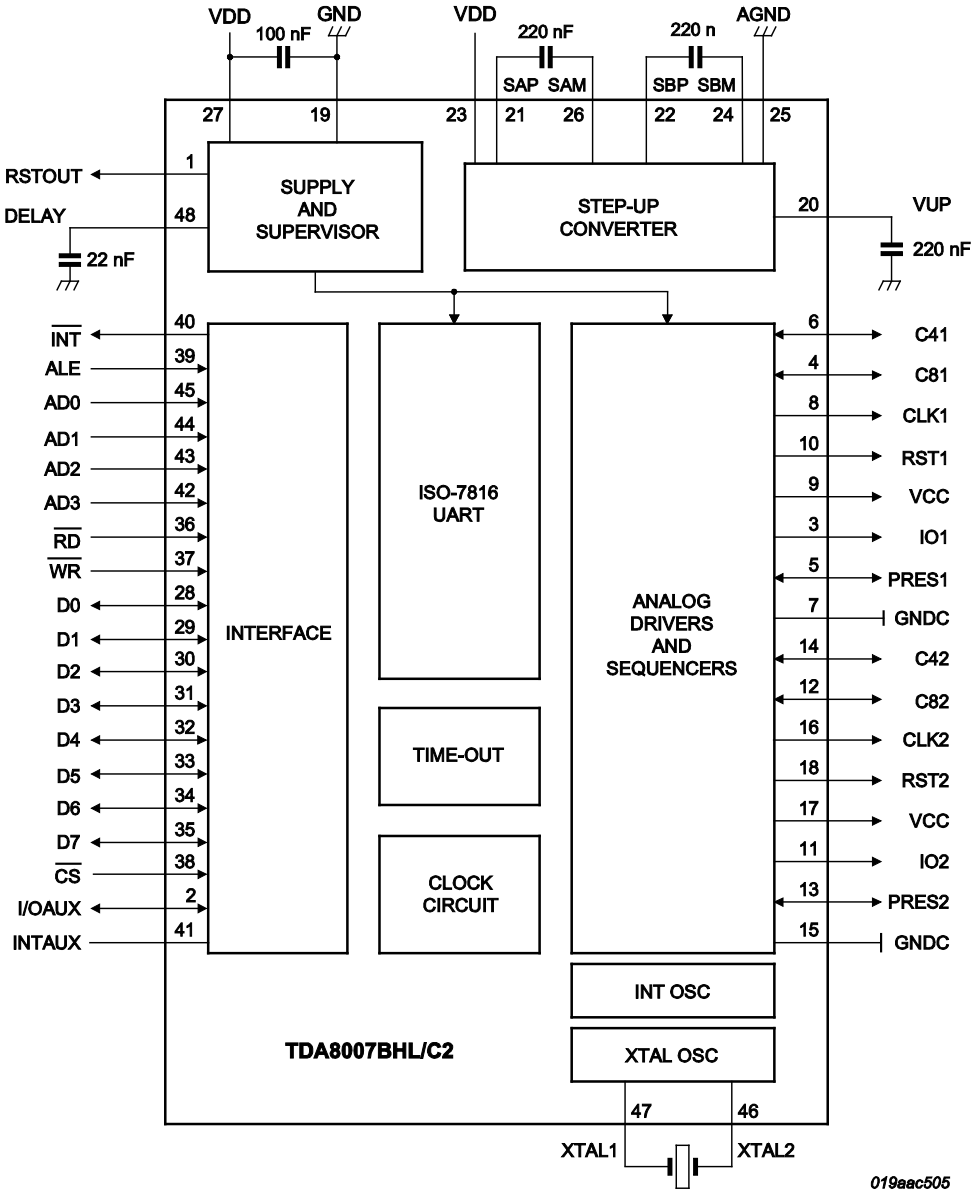


Fig 1. Block diagram

2.2 Differences between TDA8007B/C2 and TDA8007B/C3Heading 3

In order to satisfy the new EMV2000 specifications, the TDA8007B/C2 has been improved.
By the way, the remaining problems seen on the C2 version have been solved.

2.2.1 EMV2000 improvement:

1) Reception at 11,8 etus in T=0, 10,8 in T=1

The TDA8007B/C2 is unable to correctly interpret received characters having a delay between two consecutive leading edges of the start bit of 11,8 etus (10.8 etu in T=1).

This feature is now possible with the TDA8007B/C3.

2) Windows reception during Answer To Reset
The TDA8007B/C2 has a reception window during ATR which is between 384 and 42,000 CLK. This window does not satisfy the new EMV2000 specification (380-42000CLK).

The windows reception has been modified and now it is possible to receive properly an ATR beginning between 380 and 42000 CLK.

2.2.2 TDA8007BC2 problems solved

1) CRED issue

Description:
When CRED bit within the MSR register goes high, 1 (or 2) more card clock cycle should be left to the UART before taking the value into account.

Modification on the TDA8007B/C3:

This issue is solved with the C3 version so when CRED bit is high, the UART is ready.

2) Timers use and card withdrawal

Description:

If during a timer count, the selected card is withdrawn, the timers will be improperly stopped and the next count will not be taken into account.

Workaround:

In the power up function, it is necessary to select the 3rd slot and then stop the timers by writing the TOC to zero before any timers use.

Modification on the TDA8007B/C3:

Timers are now properly stopped when clearing the RIU bit so it is no more necessary to select the 3rd slot for stopping the timers.

3) Spurious PRL1 interrupt in case of card 2 deactivation

Description:

With VDD above 3 volts and C4, C8 pins set to high state, a spurious PRL1 interrupt occurs after a card 2 deactivation.
In this case, deactivation on card 2 may be due either to an emergency deactivation or a software power down.

Workaround:

The PRL1 interrupt has not to be taken into account right after a deactivation.

Modification on the TDA8007B/C3:

The same behavior still exist with the TDA8007B/C3

4) Multiplexed mode

The TDA8007BHL/C2 switches in multiplexed mode when detecting a rising edge of ALE signal with the CS low.
When CS is given by an address bus decoding, it is not possible to have both rising edge of ALE and CS low so it is not possible to switch in non multiplexed mode without external glue.

Workaround:

External glue is necessary in order to get both the rising edge of ALE and CS low.
At power on, RSTOUT pin gives a positive pulse that can be used in combination with the CS.

An example is given hereafter.

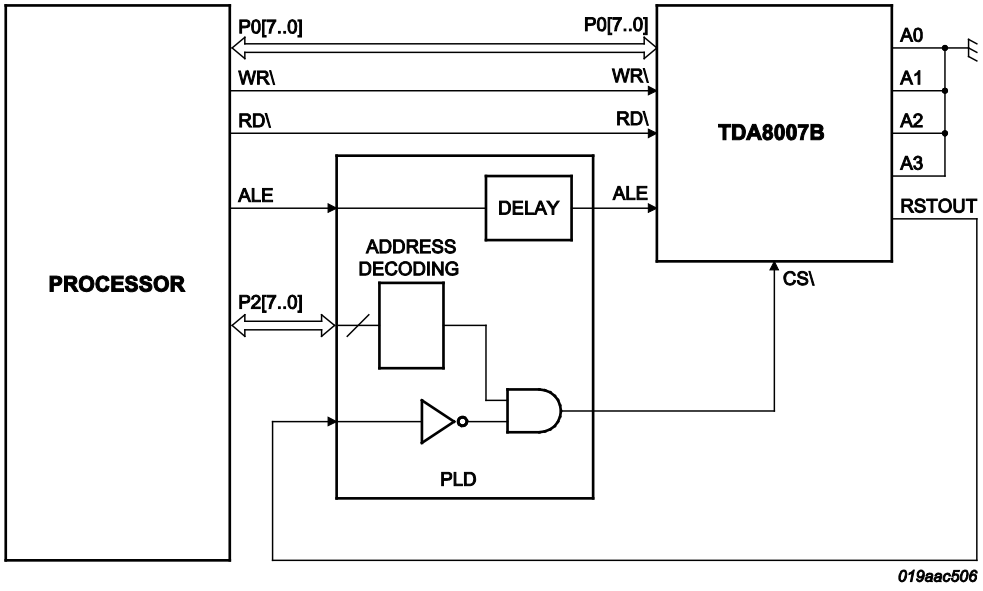


Fig 2. Multiplexed mode with automatic address decoding

Modification on the TDA8007B/C3:

Only the rising edge of ALE is now necessary to switch in multiplexed mode.

2.3 UART Registers

The TDA8007B can be driven with a set of registers, and these registers can be read or written through a parallel bus.
The list of these registers is given below.

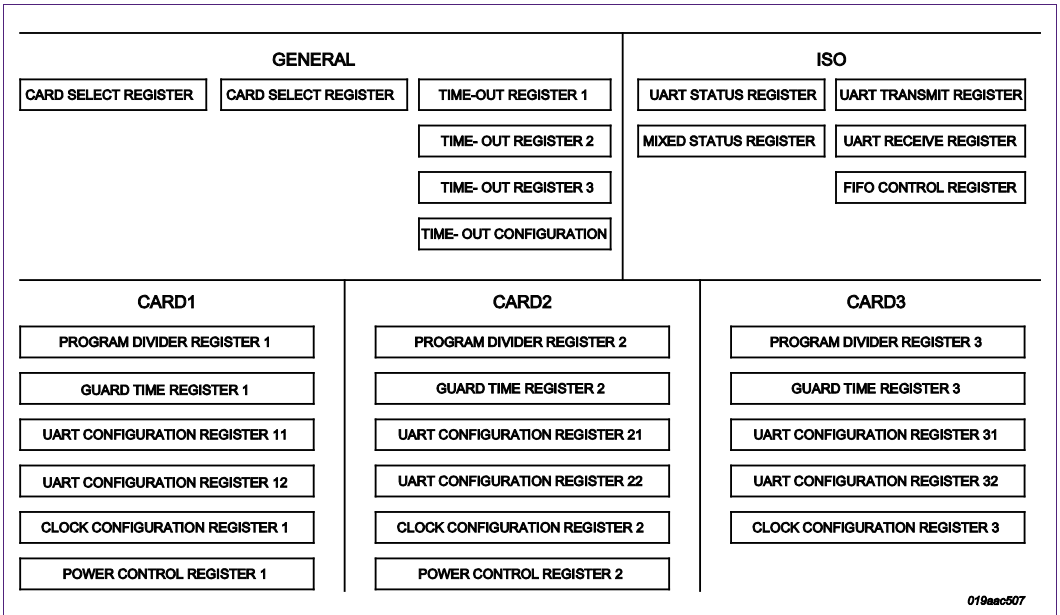


Fig 3. UART registers

2.3.1 Register description

Card Select Register (address 00h)

0	0	1	0	RIU\	SC3	SC2	SC1

This register allows resetting the UART and changes the selected slot.

- SC3, SC2, SC1: allows to respectively select card3, card2 and card1.
- RIU\: this bit reset the ISO UART and needs to be set before using UART.
- The high nibble of this register codes the version number of the chip.

Hardware Status Register (address 0Fh read only)

NU	PRTL2	PRTL1	SUP	PRL2	PRL1	INTAUXL	PTL

This register gives the hardware status of the UART.

- PTL: this bit is set when overheating has occurred.
- INTAUXL: this bit is set each time the INTAUX input has changed.
- PRL1, PRL2: these bits are set when a change has been detected on card reader 1 or on card reader 2.
- SUPL: This bit is set when the supervisor been activated.

- PRTL1, PRTL2: this bit is high when a default has been detected on VCC or RST.

Time-Out Register 1 (address 09h write only)

TOR17	TOR16	TOR1	TOR14	TOR13	TOR1	TOR11	TOR10

This register is used to store the number of etus to count with the timer 1.

Time-Out Register 2 (address 0Ah write only)

TOR27	TOR26	TOR2	TOR24	TOR23	TOR2	TOR21	TOR20

This register is used to store the number of etus to count with the timer 2.

Time-Out Register 3 (address 0Bh write only)

TOR37	TOR36	TOR3	TOR34	TOR33	TOR3	TOR31	TOR30

This register is used to store the number of etus to count with the timer 3.

Time-Out Configuration Register (address 08h)

TOC7	TOC6	TOC5	TOC4	TOC3	TOC2	TOC1	TOC0

This register is used for setting timers mode.

Timer operates in 3 modes: Soft triggered, Start bit, Auto-reload.

- In Soft triggered mode the timer starts counting the value stored in the TORx registers as soon as the TOC is written. It is mandatory to have the timers stopped before starting the timers in soft mode.
- In Start bit mode, the timer starts counting the value stored in the TORx on each start bit detected on IO line.
- In Auto-reload mode the timer starts counting the value stored in the TORx on the first start bit detected on IO line. When the timer reaches its terminal count, an interrupt is given and the timers restart automatically the same count until the timers are stopped.

UART Transmit Register (address 0Dh write only)

UTR7	UTR6	UTR5	UTR4	UTR3	UTR2	UTR1	UTR0

This register is used to transmit a character in direct convention to the selected card. The byte starts on the IO line at the end of the extra guard time.
In case of synchronous card, only bit UTR0 is relevant, and is copied on the IO line.

UART Receive Register (address 0Dh read only)

URR7	URR6	URR5	URR4	URR3	URR2	URR1	URR0

This register is used to read a character from the card. The byte is stored in direct convention.
In T=0 protocol the byte will be stored in the FIFO if no parity error is detected and in T=1 protocol, the bytes will be stored in the UTR even if a parity error is detected.

Mixed Status Register (address 0Ch read only)

CLKS	FE	BGT	CRED	PR2	PR1	INTAUX	TBE/RBF

This register is used for polling purpose.

- TBE/RBF: this bit is set each time a character is sent on the IO line or when the FIFO is full.
- INTAUX: This bit gives the level on the INTAUX pin.
- PR1, PR2: These bits are set when respectively card 1 or card 2 are present.
- CRED: This bit is set when reading or writing is allowed in URR/UTR (delay between two consecutive read or write is 2 CLK), and when writing is allowed in TOC register (delay between two consecutive read or write is 4/31 or 4/32 etus).
  It is advice to use this bit when using fast processor.
  This bit is relevant only when the selected card is activated
- BGT:
  In T=0 protocol this bit is set at 16 etus after each start bit detected on the IO line.
  In T=1 it is the same with 22 etus.
  BGT bit can be used only if a start bit has been detected on IO.
- FE: This bit is set each time the FIFO becomes empty.
- CLKSW: This bit is set when the UART has performed a required clock switch from Fxtal/n to Fint and is reset when the UART has performed a required clock switch from Fint to Fxtal/n.

UART Status Register (address 0Eh read only)

TO3	NU	TO1	EA	PE	OVR	FER	TBE/RBF

This register gives the functional status of the UART.

- TBE/RBF: In transmission, this bit is set when the user writes the next character in the UTR, and in reception when the FIFO is full.
- FER: This bit is set when the IO was not in high-impedance state at 12.21 etus after any start bit.
- OVR: This bit is set if the UART has received a new character whilst the FIFO was full.
- PE: This bit is set if the UART has received a number of naked character equal to the number written in PEC[2..0], or if a transmitted character has been naked by the card a number equal to the number written in the PEC[2..].
- EA: This bit is high if the first start bit on IO during ATR has been detected between 200 and 384 CLK pulses.
- TO1: This bit is set when counter 1 has reached its terminal count.
- T03: Same as T01 with counter 3

FIFO Control Register (address 00h write only)

NU	PEC2	PEC1	PEC0	NU	FL2	FL1	FL0

This register is used to compute both FIFO length and parity error number as follow:

- FL0 to FL2: These bits are coding the FIFO length, 0 means 1 byte, 7 means 8 bytes. When the number of bytes received in the FIFO is matching the FIFO length, the bits TBE/RBF in both MSR and USR are set and an interrupt will be pending if the bit DISTBE/RBF is reset in UCR2.
- PEC0 to PEC2: These bits are coding the number of parity error for getting an interrupt, 0 means 1 error, 7 means 8 errors.
  - In T=0 protocol, if the transmitted character is naked by the card then the UART will automatically retransmit it a number of time equal to the value programmed in PEC[2..0].

Programmable Divider Register (address 02h)

PDR7	PDR6	PDR5	PDR4	PDR3	PDR2	PDR1	PDR0

This register is used to compute the number of clock cycles per ETU.

- if PDR = 12 in decimal and that PSC bit within UCR2 is reset (prescaler = 31), the ETU will last 12 * 31 = 372 card clock cycles

Guard Time Register (address 05h)

GTR7	GTR6	GTR5	GTR4	GTR3	GTR2	GTR1	GTR0

This register is used to compute the number of extra guard time given by the TC1 parameter.
In T=1 protocol value FFh means 11 etus between each leading edge of two consecutive characters in both direction.
In T=0 protocol value FFh means 12 etus between each leading edge of two consecutive characters in both direction.

Clock Configuration Register (address 01h)

NU	NU	SHL	CST	SC	AC2	AC1	AC0

This register is used for setting the clock parameters of the ISO UART.

- AC[2..1]: Gives the divisor factor used to supply the card clock from the xtal qUARTz.
  When switching from Xtal/n to 1/2fint or vice verse it is advised to survey bit CLKSW (MSR bit 7) in order to be sure that the clock switch is effectively done.
- SC: When bit SAN (UCR2 b it 3) is set, this bit gives the level on CLK.
- CST: This bit is used with an asynchronous card for stopping the card clock. The level of the clock is given by the value of the bit SHL.
- SHL: If CST is set, then the CLK is stopped LOW if SHL=0, and HIGH if SHL=1.

UART Configuration Register 1 (address 06h)

NU	FIP	FC	PROT	T/R	LCT	SS	CONV

This register is used for setting the parameters of the ISO UART.

- CONV: This bit is used for setting manually the convention used on IO line. AUTOCONV\ (UCR2 bit 2) needs to be cleared before changing this bit value.
- SS: If sets by software before the first byte to receive in the ATR, the UART will automatically perform a convention detection and early answer detection. This bit is automatically reset by hardware.
- LCT: If set before sending the last character, the UART will automatically change to reception mode at the end of the transmission.
  - In T=0 protocol the UART returns in reception mode just before 12 etus
  - In T=1 protocol the UART returns in reception mode just before 11 etus
- T/R: If sets the UART operates in transmission mode, if resets the UART operates in reception mode. Do not change this bit during character reception or emission.
- PROT: This bit codes the protocol to be used by the UART. Clear this bit if the protocol T=0 needs to be used, and set this bit if the protocol T=1 needs to be used.
- FC: This bit is a test purpose bit and must be left to zero.
- FIP: If set, the UART will NAK a correctly received character, and will transmit characters with wrong parity bits. This feature is useful when using the TDA8007B as card simulator.

UART Configuration Register 2 (address 03h)

NU	DISTBE/R	DISAU	PDW	SAN	AUTOCON	CKU	PSC

This register is used for setting the parameters of the ISO UART.

- PSC: If reset means prescaler equal to 31, if set means prescaler equal to 32.
- CKU: If set, the ETU will have half the duration of the value than with CKU = 0.
- AUTOCONV\: If set, then the convention is set by software using bit CONV in the UCR1 register. If reset, then the convention is automatically detected on the first received character whilst the Start Session (SS) bit is set.
  - AUTOCONV\ shall not be changed during card session.
- SAN: If reset, the UART will operate in Asynchronous mode. If set the UART will operate in Synchronous mode and in this mode the ISO UART is bypassed.
- PDWN: This bit can be set by software in order to stop the crystal oscillator. This mode allows low power consumption.
  - There are 5 ways to get out this mode:
    - Insert or withdraw card 1 or 2.
    - Select the UART by resetting CS\.
    - Change on pin INTAUX.
    - If CS\ is permanently set to Low, reset bit PWDN by software.
  - Except for read operation on HSR, INT\ will be pulled Low.
- DISAUX: If set, any change on INTAUX will not generate an interrupt. In order to disable properly this interrupt, it is mandatory to set this bit on each UCR1 register (UCR11, UCR12, UCR13).
- DISTBE/RBF: If set, the reception or the transmission will not generate an interrupt, only bits TBE/RBF in both MSR and USR will be set if the FIFO is full or if the character has been transmit on IO line.

Power Control Register (address 07h)

NU	NU	C8	C4	1V8	RSTIN	3V/5V	START

- START: When set, the UART starts the activation sequence, and when reset the UART starts the deactivation sequence.
  The activation sequence can be performed only if the selected card is present and that no hardware problem is detected.
- 3V/5V: If this bit is set, then the VCC is 3V. If this bit is reset, then the VCC is 5V.
- RSTIN: When bit start is set, RST is the copy of the value written in RSTIN.

  With synchronous card this bit is used to set or reset the RST pin.
- 1V8: If this bit is set the VCC is 1.8 volt whatever the value of bit 3V/5V.
- C4: C4 contact is the copy of the value written in C4 bit. When this bit is set to logic one, the value of the C4 bit gives the level of C4 contact.
- C8: C8 contact is the copy of the value written in C8 bit. When this bit is set to logic one, the value of the C8 bit gives the level of C8 contact.

3. Hardware implementation

3.1 Functional diagram

The TDA8007B can operate in both multiplexed or non multiplexed mode.

The initial state of the chip is **non-multiplexed** mode. The TDA8007BC2 moves to **multiplexed** mode on the first rising edge of *ALE* with *CS* low, while the TDA8007BC3 switch with only a rising edge of ALE.

3.1.1 Non multiplexed mode

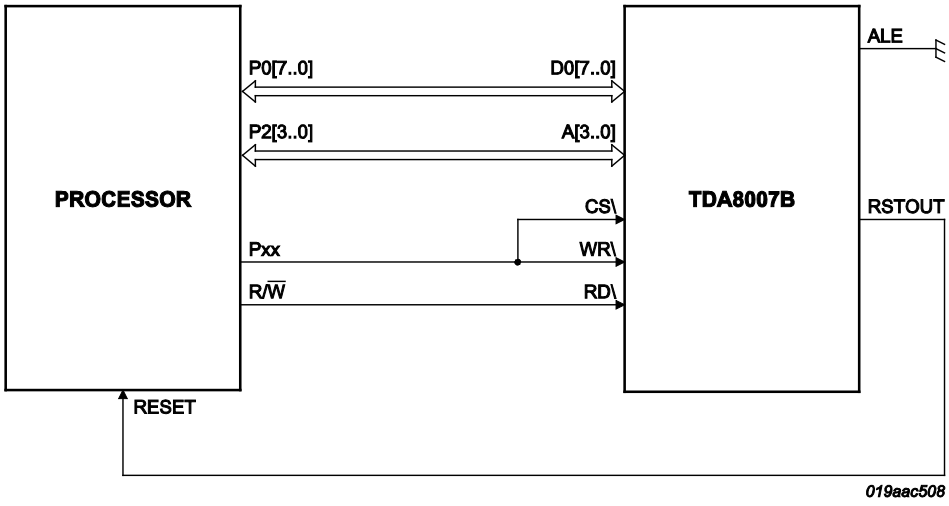


Fig 4. Non multiplexed mode

To avoid any unwanted switch in multiplexed mode, it is recommended to connect ALE to the Ground.

3.1.2 Multiplexed mode

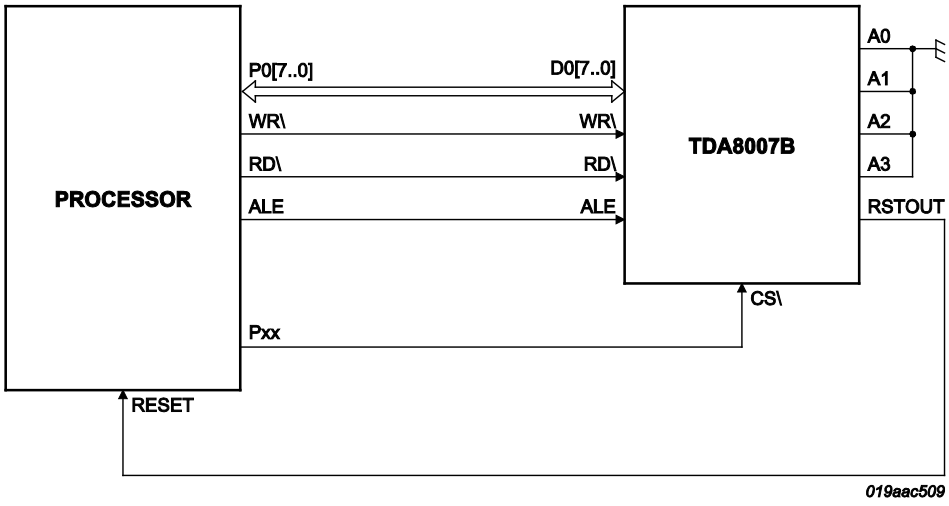


Fig 5. Typical implementation

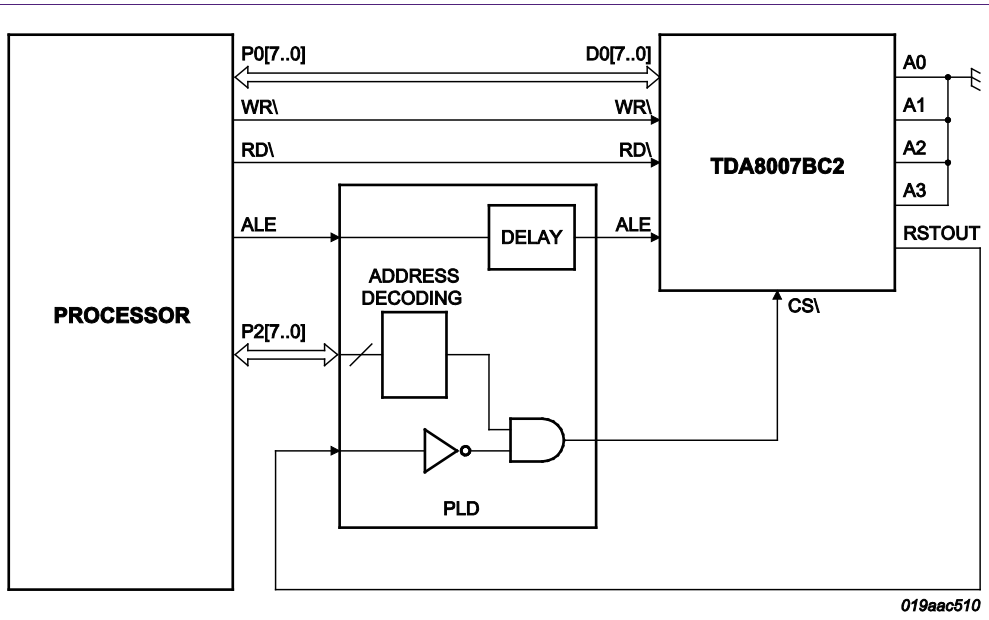


Fig 6. TDA8007BC2 with an automatic address decoding

3.2 Layout

The layout of the TDA8007B has to follow certain rules in order to benefit from the hardware improvements for noise reduction on card signals.

Each card slots has a specific ground line GNDC1 (or GNDC2) to which the capacitor placed on VCC line must be connected. These ground lines must be connected to a common ground point placed close to the AGND pin of the TDA8007B, just after the polarized capacitor (10µF) placed between AGND and AVDD.

The 220nF capacitor used for the step up converter (on VUP, between SAM and SAP, between SBM and SBP) shall be low ESR devices. They must be placed as closed as possible of the respected pin of the TDA8007B to which they are connected. Moreover for connecting the capacitor of the step up converter it is advised not to use metal via (for changing board layer) because they will add some serial parasitic resistor that may create problems.

A 180 Ohms resistor should be added between IOx pin from the TDA8007B and C7 contact in order to prevent glitches during current limitation on IO line.

The CLK line is the most polluting line and shall be routed between GNDC line and eventually another GND line in order to avoid cross coupling with VCC, RST and I/O lines.

A layout possibility is given on the bellow schematic.

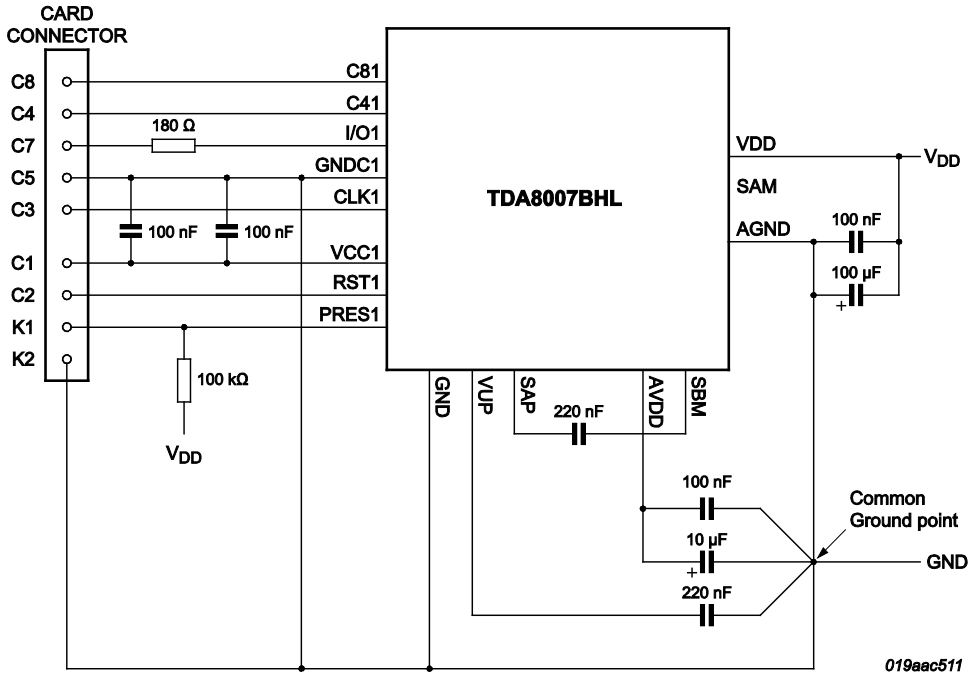


Fig 7. Layout implementation example

3.3 Presence switch implementation

3.3.1 Using normally closed switch

K1, K2 are opened when the card is inserted.

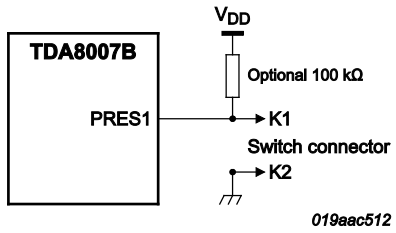


Fig 8. Normally Closed switch

3.3.2 Using Normally opened switch

K1, K2 are opened when the card is inserted.

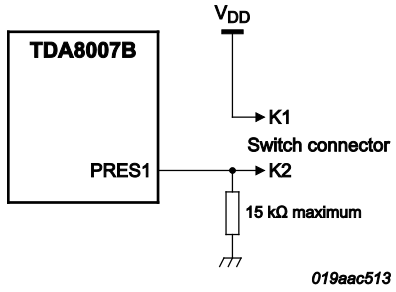


Fig 9. Normally Opened switch

PRES internally embeds a current source to VDD to pull PRES high when the pin is not connected.
In this case, an external pull-down resistor is then required to pull the line low when the connector is opened.

The maximum pull-down resistor can be calculated as follows:
Rmax = Vil max / I max.

Vil max is equal to 0.3.VDD and I max is equal to 55µA.

E.g for VDD = 3.3V, Rmax is 18kΩ (= 15kΩ ± 20%)
A maximum of 15kΩ must then be used for such a design with 3.3V. (with a 20% tolerance)

4. Software aspect

4.1 Initialization

After power-on, the TDA8007B generates a positive pulse on RSTOUT pin whose width is proportional to the value of Cdelay (typically 1ms per 2nF). This pulse may be used as a positive reset pulse by the micro-controller. The processor needs to wait the end of this pulse before any action on the TDA8007B.

It is possible to reset the TDA8007B by shorting Cdelay to GND.

After power-on, bit SUPL in HSR is high and remain high until HSR is readout. All the UART registers are in their default values and all the configuration registers are set to zero so it is mandatory to configure the UART.

After a reset with RIU bit, only ISO UART part is reset so all the configuration of the selected card but the FCR register is staying in the same state (see datasheet).

4.2 Answer To Reset (ATR)

4.2.1 Asynchronous card

When you want to perform Answer To Reset on asynchronous card, you have to respect the following sequences:

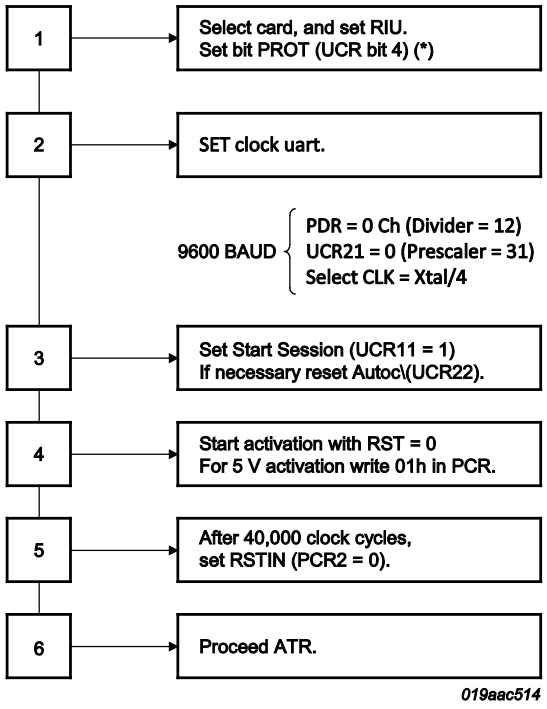


Fig 10. Asynchronous card

Note: Before setting manually bit CONV, don't forget to set bit AUTOC (UCR22).
(*): In order not to "NACK" the IO line in case of parity error, select the protocol T=1 during ATR.

4.2.2 Synchronous card

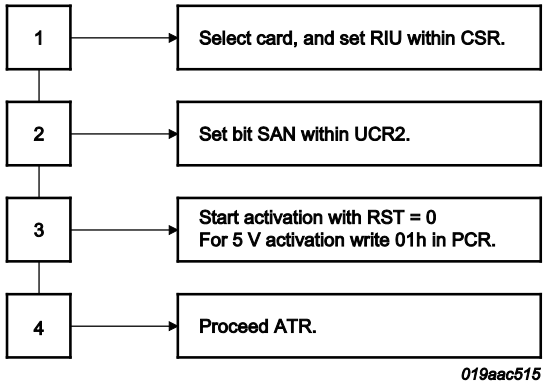


Fig 11. Synchronous card

Note: In case of synchronous card, then the CLK contact is the copy of the value written in SC (CCR3). In reception mode, data from the card is available in URR0; in transmission mode, the I/O line is the copy of the value written in UTR0.

4.3 APDU exchange

Before any APDU exchange the information given in the ATR has to be analyzed and taken into account. You will find hereafter the main parameters to program.

- The protocol bit (UCR1 bit 4) has to be updated with the first protocol offered in the ATR.
- GTR register has to be programmed with the parameter TC1 given in the ATR.
- If the card operates in the specific mode, modify the baud rate with the FiDi given by the TA1 parameter.
- Select the appropriate size for the FIFO (FCR[2..0]).
- Select the number of allowed retransmission in case of parity error detection (FCR[6..4]).
- Compute the WWT value with the TC2 parameter (WI).
- Compute the CWT and BWT values with the TB3 parameter.
- Compute the “IFSC” value with the TA3 parameter.

4.3.1 Asynchronous Reception

4.3.1.1 By polling

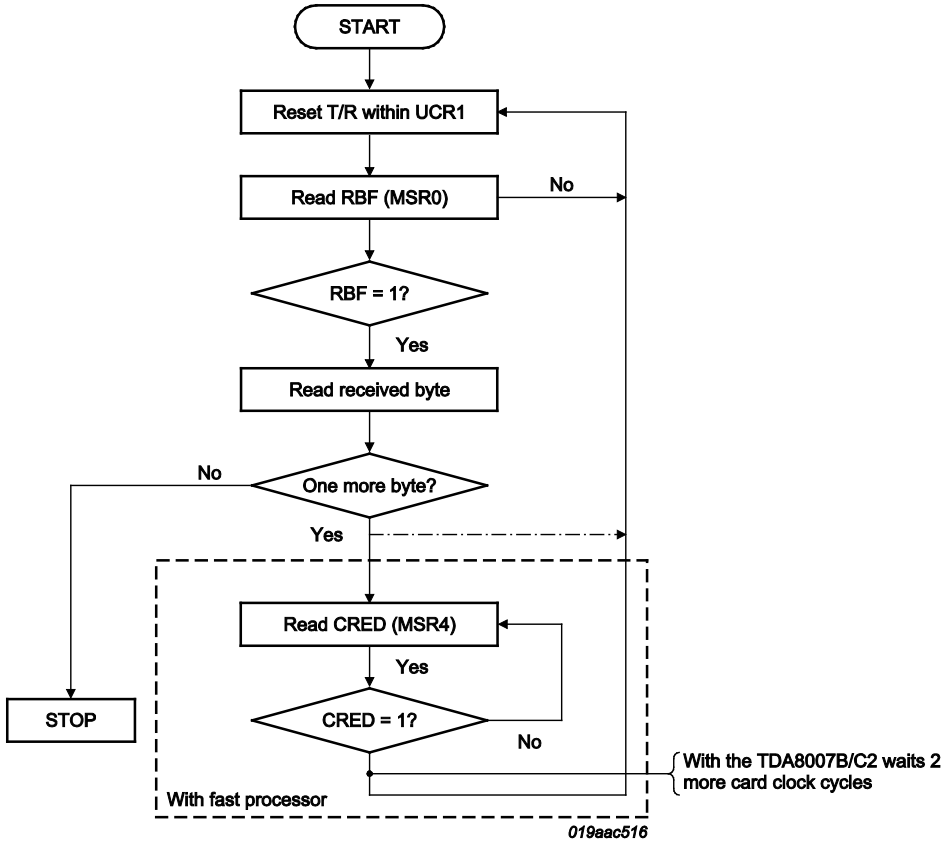


Fig 12. Reception with polling method

4.3.1.2 Under interruption

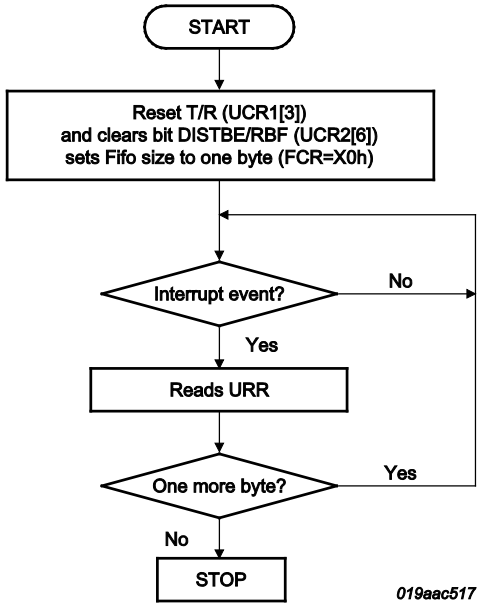


Fig 13. Reception with interrupt method

4.3.1.3 Examples

Reception in polling mode

```
UCHAR user_byte;
UCHAR URR_Value;
UCHAR data_exch_buff[272];

#define  T123STOP 0x00
#define  MSR_CRED   0x10
#define  UART_BASE_ADDRESS 0x0000
#define  TDA_REG  ((UCHAR volatile xdata *) UART_BASE_ADDRESS)
#define  FEuser_byte & 0x40

UCHAR rcv_card(UINT nb,UINT ptr)
{

        for(;nb > 0;nb--,ptr++)    // for all the buffer lenth
        {
                                        // external ISO UART
            do
            {
                ENABLE_UART
                user_byte = TDA_REG[MSR];
                DESABLE_UART
                if (alarm== card_selected)
                {
                    ENABLE_UART
                    while(!(TDA_REG[MSR] & MSR_CRED));
                    TDA_REG[TOC]=T123STOP;
                    DESABLE_UART   // desable ISO UART
                    return(!OK);
                }
            }while(FE)         // wait for one byte in FIFO

            ENABLE_UART
            URR_Value = TDA_REG[URR];    // URR register read
            DESABLE_UART

            data_exch_buff[ptr]= URR_Value;

            while(!(TDA_REG[MSR] & MSR_CRED));   <-- Only with fast processor
            for (i=0; i<Tempo_2CLK; i++);        <-- Only with
        }
        return(OK);          // return "receive successful
}
```

Reception under interruption

```
UCHAR user_byte;
UCHAR URR_Value;
UCHAR data_exch_buff[272];
UCHAR Parity_Error_Flag;
UCHAR Rx_Flag;

#define  T123STOP     0x00
#define  MSR_CRED       0x10
#define  UCR1_TR      0x08
#define  UART_BASE_ADDRESS  0x0000
#define  TDA_REG      ((UCHAR volatile xdata *) UART_BASE_ADDRESS)
#define  TBE_RBF      user_byte & 0x01
#define  PE_DETECTION user_byte & 0x08



UCHAR rcv_card(UINT nb,UINT ptr)
{

        for(;nb > 0;nb--,ptr++)    // for all the buffer lenth
        {
                                              // external ISO UART
            while(!Rx_Flag)
            {
                if (alarm== card_selected)
                {
                    ENABLE_UART
                    while(!(TDA_REG[MSR] & MSR_CRED));
                    TDA_REG[TOC]=T123STOP;
                    DESABLE_UART    // desable ISO UART
                    return(!OK);
                }
        }               // wait for one byte in FIFO
   RX_Flag = 0;
        data_exch_buff[ptr]= URR_Value;

        return(OK);          // return "receive successful
}
```

```
int_TDA8007() interrupt 0 using 1
{
   UCHAR TXRX_Mode;

         ENABLE_UART
   user_byte = TDA_REG[USR];
   DESABLE_UART
   If(TBE_RBF)
   {
      ENABLE_UART
      TXRX_Mode = TDA_REG[UCR1] & UCR1_TR;
      DESABLE_UART
                  if(TXRX_Mode)
                  {
                        //Proceed transmission routine
      }
                  else
                  { // Proceed reception routine
                    ENABLE_UART
                        URR_Value = TDA_REG[UCR1];
                        DESABLE_UART
                        Rx_Flag = 1;
                        Parity_error_Flag = 0;
                        If(PE_DETECTION)
                           Parity_Error_Flag = 1;
                  }
   }
   //proceed the other USR status bits
}
```

4.3.2 Asynchronous Transmission

4.3.2.1 By polling

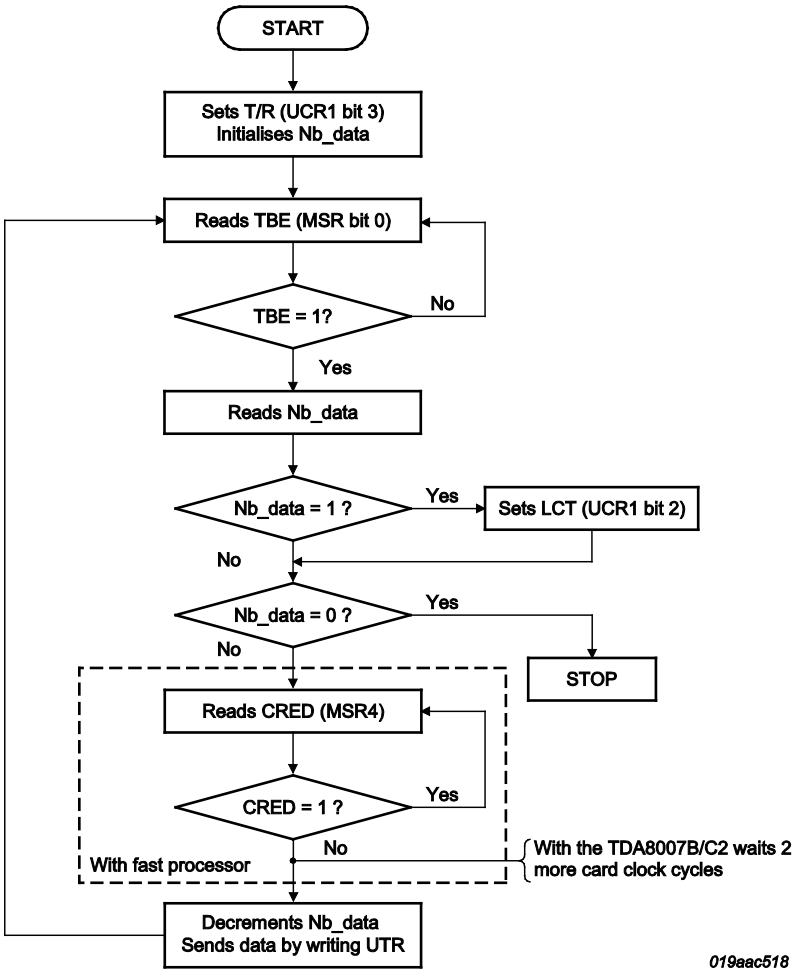


Fig 14. Transmission with polling method

4.3.2.2 Under interruption

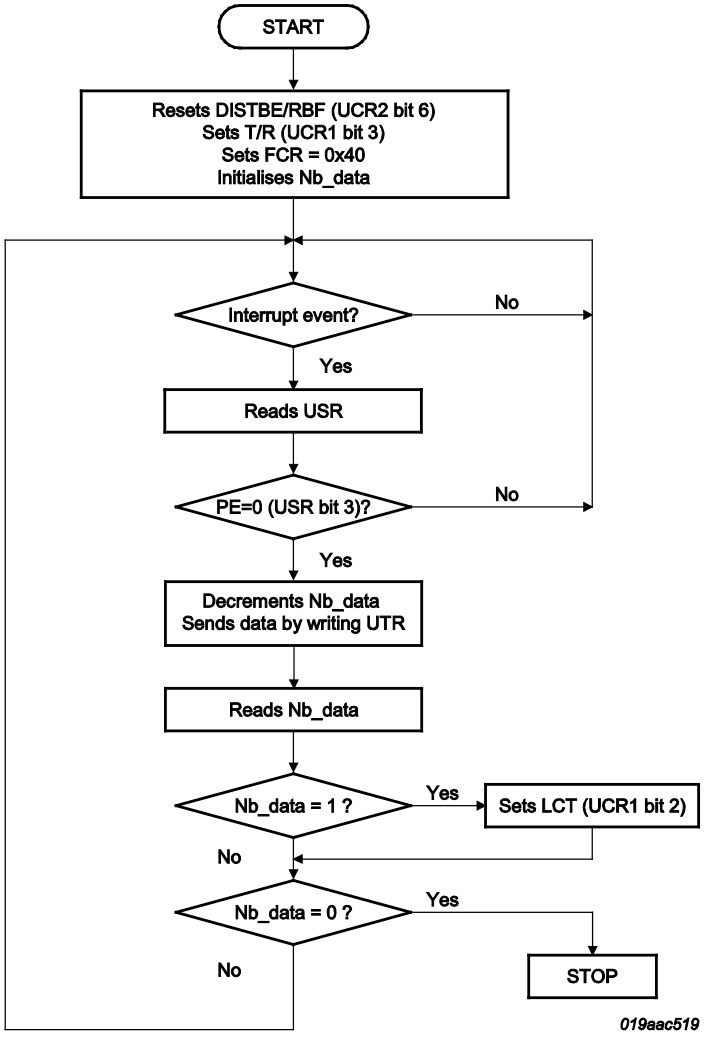


Fig 15. Transmission with interrupt method

4.3.2.3 Examples

Transmission by polling

```
#define  UART_BASE_ADDRESS 0x0000
#define  TDA_REG      ((UCHAR volatile xdata *) UART_BASE_ADDRESS)

UCHAR send_bytes_to_card(UINT nb,UINT ptr)    // by polling
{
  UCHAR ch, TX_mode;
  UINT  CWT_value, Tx_pointer, Tx_counter;
  BOOLEEN GetOneByte = 1;

  Tx_pointer=ptr;
  Tx_counter=nb;

  ENABLE_UART
  TDA_REG[UCR1] |= UCR1_TR;
  DISABLE_UART
  do{
   do{
       if(GetOneByte==1)
       {   // reading byte to send
           ch =  data_exch_buff[Tx_pointer];
           Tx_pointer++;
           Tx_counter--;
           GetOneByte = 0;
       }
       ENABLE_UART
       TX_mode   = TDA_REG[UCR1] & UCR1_TR;
       user_byte   = TDA_REG[MSR];
       DISABLE_UART
       if(alarm==card_selected)
           return(!OK);
   }while(!(MSR_TBE) && TX_mode);

   if(Tx_counter==0)
   {   //writing LCT before sending the last byte
       ENABLE_UART
       TDA_REG[UCR1] |=  UCR1_LCT;
       DISABLE_UART
   }

   ENABLE_UART
   TDA_REG[UTR] = ch;              //transmit one byte
   DISABLE_UART

   while(!(TDA_REG[MSR] & MSR_CRED));
         for (i=0; i<Tempo_2CLK; i++);

   GetOneByte = 1;
  }while(Tx_counter);

  return(OK);

}
```

Only with fast processor

Only with

Transmission under interrupt

```
UCHAR user_byte;
UCHAR data_exch_buff[272];
UCHAR Flag_TX;

#define  UART_BASE_ADDRESS 0x0000
#define  TDA_REG      ((UCHAR volatile xdata *) UART_BASE_ADDRESS)
#define  TBE_RBF      user_byte & 0x01
#define  PE_DETECTION user_byte & 0x08

UINT Tx_counter;

UCHAR send_bytes_to_card(UINT nb,UINT ptr)
{
          UINT Tx_pointer;

    Tx_pointer = ptr;
    Tx_counter = nb;

    byte_to_transmit = data_exch_buff[Tx_pointer++];
    write_bit(TXRX,1);    //start transmit interrupt

    // waiting  for end of transmission or alarm
    do{
        if(Flag_TX)


        {
            Flag_TX = 0;
            byte_to_transmit = data_exch_buff[Tx_pointer++];
        }

    }while(Tx_counter && !alarm);

    // waiting  for UART in reception
    while(read_register(UCR1)&0x08)
    {
        if(alarm==card_selected)  //waitting for UART in reception
            return (!OK);
    }

    if(alarm)
        return(0);
    return(1);
}
```

```
int_TDA8007() interrupt 0 using 1
{
   UCHAR TXRX_Mode;

         ENABLE_UART
   user_byte = TDA_REG[USR];
   DESABLE_UART
   If(TBE_RBF)
   {
      ENABLE_UART
      TXRX_Mode = TDA_REG[UCR1] & UCR1_TR;
      DESABLE_UART
                  if(TXRX_Mode)
                  { //Proceed transmission routine
                        if((Tx_counter--)==1)
                              TDA8007_REG[UCR1] |= 0x04;     //LCT = 1 returning
                        UART after the last byte
                        if(!(PE_DETECTION))
                        {
                              TDA8007_REG[UTR] = byte_to_transmit;
            Flag_TX = 1;
         }
      }
                  else
                  { // Proceed reception routine
                    ENABLE_UART
                        URR_Value = TDA_REG[UCR1];
                        DESABLE_UART
                        Rx_Flag = 1;
                        Parity_error_Flag = 0;
                        If(PE_DETECTION)
                           Parity_Error_Flag = 1;
                  }
   }
   //proceed the other USR status bits
}
```

4.4 Timers use

4.4.1 Definitions

Each counter may be configured in 3 ways:

- software mode: the count starts as soon as the mode is configured in the TOC register and an interrupt is generated at the end of the count.
- start bit mode: the count starts as soon as a start bit on the I/O line has been detected (in transmission or reception mode). An interrupt is generated at the end of the count.
- autoreload mode: the count starts as soon as a start bit on the I/O line has been detected (in transmission or reception mode) and generates an interrupt at the end of the count. Then the count starts again with the initial count value and generates an interrupt each time the count is reached till it is not stopped by writing the appropriate value in the TOC register.

Furthermore, a special mode can be used to stop all the timers after the 12^{th} ETU following the first received start bit detected on the I/O line after this mode has been programmed by writing in the TOC register. This mode is called auto-stop in the following tables. This feature may be very useful during the ATR processing.

4.4.2 Two independent counters

Here are presented the possible TOC register values to work with two independent counters.

TOC	TOR3	TOR2	TOR1
0x00	stopped		stopped
0x05	stopped		start bit autoreload
0x61	soft		stopped
0x65	soft		start bit autoreload
0x85	stopped		start bit autoreload auto-stop
0xE5	soft auto-stop		start bit autoreload auto-stop

4.4.3 One 24-bits counter

Here are presented the possible TOC register values to work with one 24-bits counter.

TOC	TOR3	TOR2	TOR1
0x00	stopped		
0x68	soft		
0x7C	start bit		

4.4.4 Changes rules

The following rules have to be respected to assure a good operation of the counters:

- It is mandatory to stop the concerned counter before restarting a software count.
- The TORs registers can be changed during a count only in case of a single 24-bits start bit counter configuration.
- It is forbidden to write a value in the TOC register when the card clock is not running.

4.4.5 Examples of Use of the counters

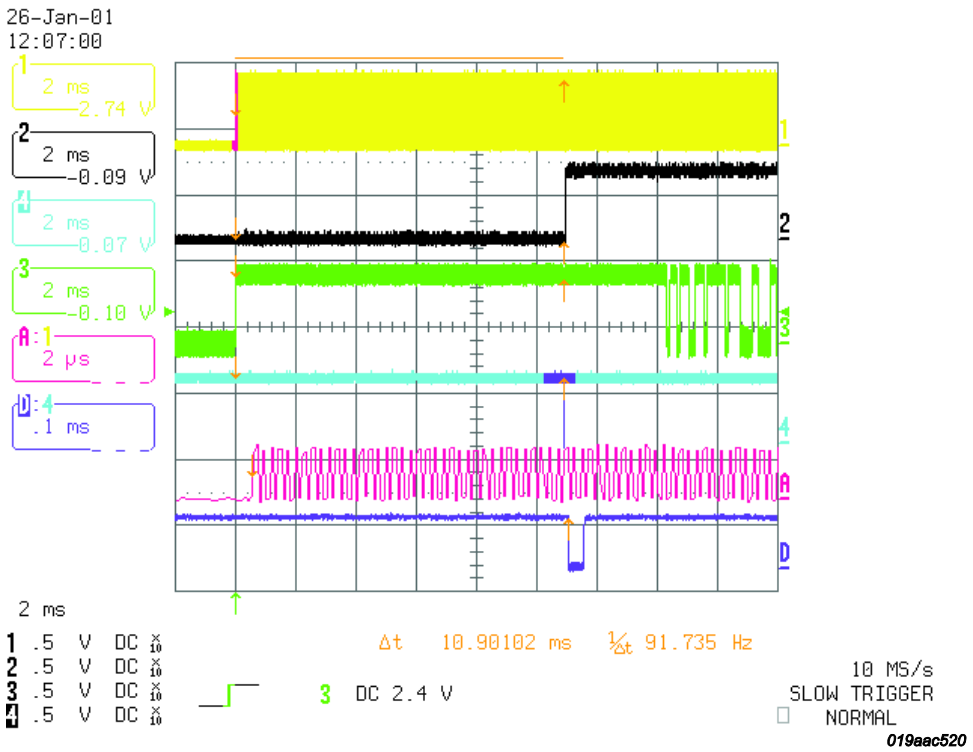
4.4.5.1 During the ATR of the card

The activation of a card is somewhat different in ISO mode or in EMV mode. In particular, the total Answer To Reset duration of the smart card should not exceed 19200 ETUs in EMV mode as it not specified in ISO mode. The two cases are detailed hereafter.

4.4.5.2 Activation in ISO mode

In case of a cold reset, after having applied CLK, the reader has to maintain RST in low state for a period of between 40000 and 45000 clock cycles before to set RST in high state.
This timing control can be achieved using counters 2 + 3 in software mode (it could be done also using one 24-bits counter):

- load TOR3=0x00 and TOR2=0x6C, i.e. 108 ETUs = 40176 clock cycles $\subset$ [40000 ; 45000] clock cycles
- program the Time Out Configuration register (TOC) to work in software mode (T3+T2) : TOC = 0x61
- set START bit in PCR to 1
- wait for a TO3 interrupt
- set RSTIN bit in PCR to 1



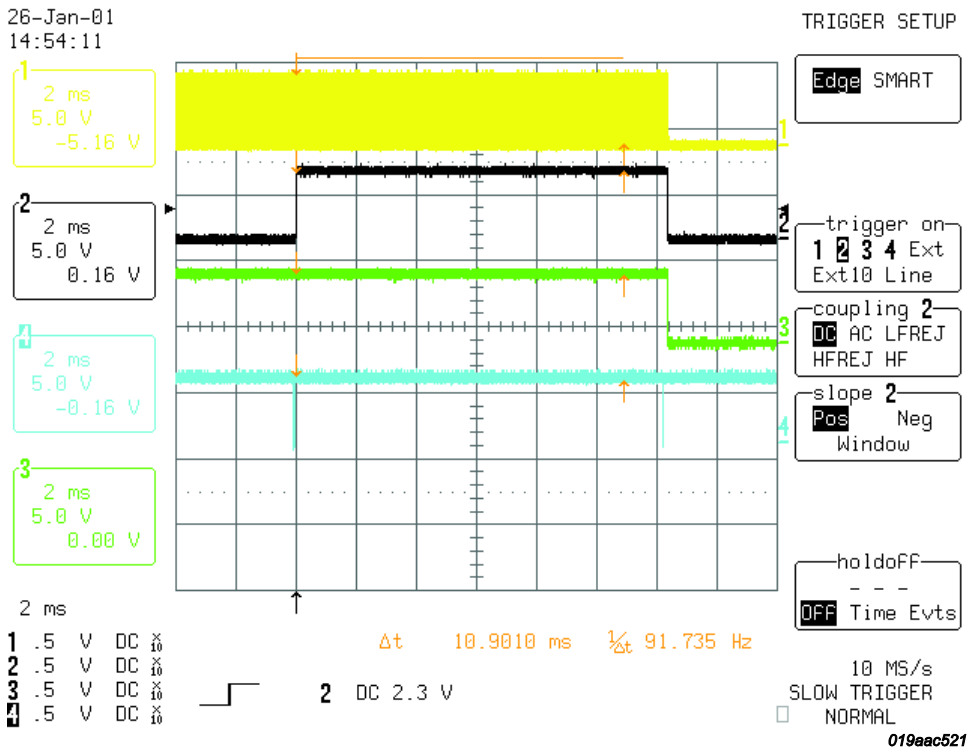
1 (A) CLK
2 RST
3 I/O
4 (D) INT0

Fig 16. ISO activation

This figure shows the TO3 interrupt corresponding to the 108 ETUs count after the first clock cycle. Then the bit RST can be set to 1.

Then, the answer to reset from the card must begin between 400 and 40000 clock cycles. Once more, timer 2 + 3 in software mode may be used to check this timing.

- load TOR3=0x00 and TOR2=0x78, i.e. 120 ETUs. These 120 ETUs include first received character, that means that the time between the TOC write operation and the leading edge of the start bit of the first received character should not exceed 108 ETUs, i.e. 40176 clock cycles. Due to the ETU resolution of the counters, that is the best approximation that can be done to check this timing.
- program the Time Out Configuration register (TOC) to work in software mode (T3 + T2) : TOC=0x61
- if a character reception has not been detected before the end of the count, the card can be considered as mute.



1 CLK
2 RST
3 I/O
4 INT0

Fig 17. Activation – Card mute

This figure shows the TO3 interrupt corresponding to the 120 ETUs count after the bit RST has been set by software. In our case, the card has not given its answer to reset during that time interval ; it can be considered as mute and it is consequently deactivated.

- If a character is received before the end of the count, one have to stop the timers 2 + 3 (TOC=0x00), reload the TOR to check 9600 ETUs between every received character (TOR3=0x25, TOR2=0x80) and at last reload the TOC register with the same value (TOC = 0x61).
- Once the complete ATR has been received, all the timers have to be stopped (TOC = 0x00).

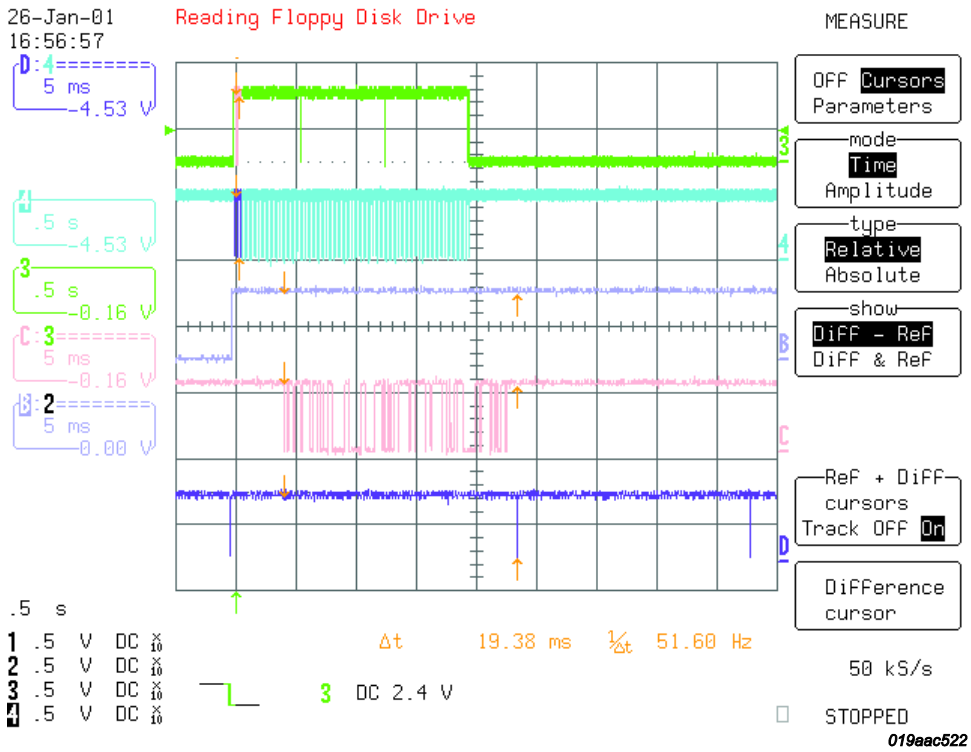
4.4.5.3 Activation in EMV mode

In case of EMV compliant activation, the complete time duration of the complete ATR has to be checked (no more than 19200 ETUs). In that case, the timer 1 may be programmed to give an interrupt every 192 ETUs (0xC0) and then the software has to verify than no more than 100 TO1 interrupts have occurred during the total time of ATR reception. Before receiving the last character of the ATR, a special mode may be used which automatically stops all the timers at 12 ETUs after next the start bit detected on the I/O line. That is what we called soft auto-stop mode in §4.4.2.

Consequently, the sequence described above is changed as follows:

- load TOR3=0x00 and TOR2=0x6C, i.e. 108 ETUs = 40176 clock cycles $\subset$ [40000 ; 45000] clock cycles
- program the Time Out Configuration register (TOC) to work in software mode (T3+T2) : TOC = 0x61
- set START bit in PCR to 1
- wait for a TO3 interrupt
- set RSTIN bit in PCR to 1
- check [400, 42000] clock cycles ATR beginning : TOR3=0x00, TOR2=0x78, TOR1=0xC0 and TOC=0x65
  Timer1 is also used to begin the 19200 ETUs count after the first received character
- after every received character except the one before the last one, reload the timers to check 9600 ETUs and 19200 ETUs :

  TOC=0x05 *Counter 1 continues to count in autoreload mode*
  TOR3=0x25, TOR2=0x80, TOR1=0xC0
  TOC=0x65 *Counters 2+3 begin a new software count*



(B) RST
3 (C) I/O
4 (D) INT0

Fig 18. EMV activation

This figure shows the TO3 interrupt (the first one on channel D) corresponding to the 108 ETUs count to set the RST line.
The following interrupts correspond to the repetitive 192 ETUs count after the first start bit detected on the I/O line. In our case, the card has not given its complete answer to reset before the 100th interrupt and it is consequently deactivated.

- before receiving the last character of the ATR, use the special End of ATR mode (auto-stop) :
  TOC=0x05 *Counter 1 continues to count in autoreload mode*
  TOR3=0x25, TOR2=0x80, TOR1=0xC0
  TOC=0xE5 *Counters 2+3 begin a new software count*
  All the counters will be automatically stopped 12 ETUs after the next start bit detected.
  In case of an ATR which length is very close to 19200 ETUs, the time to manually stop the timers could be too long and therefore a fake alarm could be detected.

4.4.5.4 During an exchange in T=0 protocol:

In T=0 protocol, as the extra guard time used to send characters from the interface device to the card may be automatically managed by the ISO-UART hardware by the means of the GTR register, the only timing to check is the Work Waiting Time (WWT).

- During a transmission phase to the card, timers may not be used. It is nevertheless advised to load the TOR with the values that will be used during the reception phase (WWT) to gain time
- Before to set the UART in reception mode, the timers have to be programmed to check the WWT. The value of the WWT (960*WI*D) is loaded in the TOR321 and the timer mode is programmed to be a single 24 bits counter, starting counting on the first start bit detected on I/O (TOC=0x7C)
- If a TO3 interrupt occurs during the reception, that means that a time-out error has been detected and the card has to be deactivated
- Else after having received the last character from the card, the timers have to be stopped (TOC=0x00).

4.4.5.5 During an exchange in T=1 protocol:

In T=1 protocol, several timings have to be checked:

- Character Waiting Time (CWT) defined as the maximum delay between the leading edges of two consecutive characters in the same block,
- Block Waiting Time (BWT) defined as the maximum delay between the leading edges of the last character sent by the TDA8007B and the first character received from the card,
- Block GuardTime (BGT) defined as the minimum delay between two consecutive characters in opposite directions. This timing control can be done using the BGT bit in MSR register specifically designed for that.

The following process can be achieved to check CWT and BWT:

- Before a transmission phase to the card, Timers may be programmed for checking the BWT respect between transmission and reception.
  For that, TOC must be programmed to work in a single 24 bits counter starting on start bit (TOC=0x7C) and TOR321 must be loaded with the current BWT value. Thus, the Timers will restart to count BWT on each consecutive start bit sent to the card and in particular on the last character sent.
- If a BWT time-out occurs before having received the first character from the card, a time-out interrupt will be generated and TO3 will be set in the USR register.
  As the TDA8007B is the master of the exchange with the smart card, the CWT is not checked during transmission.
- Immediately after having received each character from the card Timers shall be stopped then started in software mode by writing 0x61 in the TOC register. In-between, TOR2 and TOR3 need to be loaded with the CWT value.
  In the same way, if a CWT time out occurs during character reception from the card, a time-out interrupt will be generated and TO3 will be set in the USR register.
- When receiving the last character from the card, Timers have to be stopped (TOC=0x00)

Timer management can be illustrated as follows:

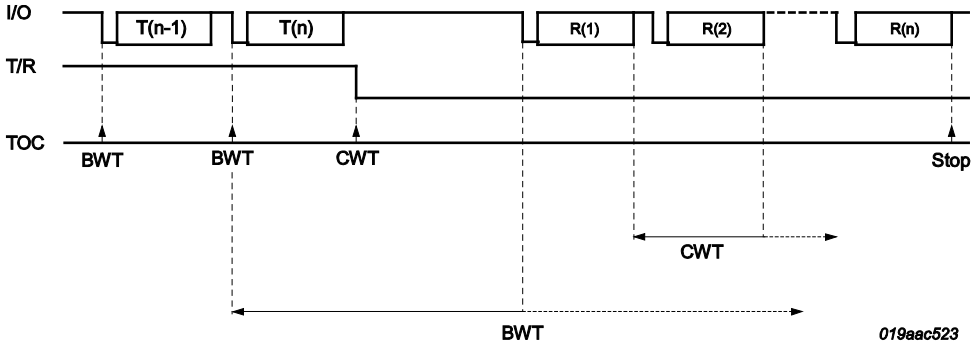


Fig 19. Timers management

4.4.5.6 Possible other applications

Others counters applications are imaginable.

The only restriction is, as mentioned before, to keep in mind that they only can be used when the card is powered, i.e. with a running clock card.

For instance, for cards supporting clock stop, ISO 7816 specifies:

- CLK can be stopped only 1860 clock cycles after the last received character from the card,
- The first character sent by the TDA80007B after the restart of the clock should not be send before 700 clock cycles

Depending on the current ETU duration, the clock cycles counts may be approximated using counters in software mode.

4.5 Parity error management

The parity error interrupt occurs at 10.5 etus in T=0 reception and T=1 emission/reception, and at 11.5 etus in T=0 transmission.

4.5.1 Manual mode

The TDA8007B operates in manual mode as soon as the value written in FCR[6..4] is zero.
In this mode, an interruption is generated and bit PE is set within USR at the first parity error for both reception and transmission mode.

In T=0 reception, the I/O line is “NACKED” at 10.5 etus and the character is not enter into the FIFO, in T=1 the character with error is enter into the FIFO and the I/O line is not “NACKED”.

In T=0 transmission, the transmitted character with error has to be manually retransmitted and will start on I/O line after a delay of at least 13.5 etus.
In transmission for both T=0 or T=1 protocol, as soon as a parity error is detected the UART stays in transmission mode whatever the LCT bit is set or not.

4.5.2 Automatic mode

The TDA8007B operates in automatic mode as soon as the value written in FCR[6..4] is not zero.
In this mode an interruption is generated and bit PE is set within USR when the UART has reached the number of allowed transmission or reception. The parity error interrupt occurs at 11,5 etus in T=0 and 10,5 etus in T=1.

In T=0 reception, the I/O line is “NACKED” at 10.5 etus and the character is not enter into the FIFO.

In T=0 transmission, the transmitted character with will be transmitted again automatically, after a delay of at least 15 etus until the number of allowed transmission is not reached. At the end of allowed retransmission the UART stays in transmission mode whatever the LCT bit is set or not.

In T=1 the character with error is enter into the FIFO and the I/O line is not “NACKED”.

Remark: when operating under interrupt transmission mode, as soon as a parity error as been detected, bit T/R within UCR1 need to be manually reset in order to reset the interrupt and return in reception mode.

4.6 Using slot 3

The TDA8007B allows managing up to three card readers by mean of the third slot. In this case, one analogue interface like TDA8004, TDA8002 or TDA8020 is requested to drive the third slot.

In order to connect an analogue interface to the TDA8007B, two dedicated pins are available:

- I/OAUX is used to drive the input/output line of the analogue interface, connects this pin to the Input/output line of the analogue interface;
- INTAUX is use to drive the hardware event (presence detection, hardware problems...) so connects this pin to the interrupt output of the analogue interface.

Possible schematic that shows the interconnection between the TDA8007B and the analogue interface are given as example on next pages.

4.6.1 Three cards reader using TDA8007B and TDA8004

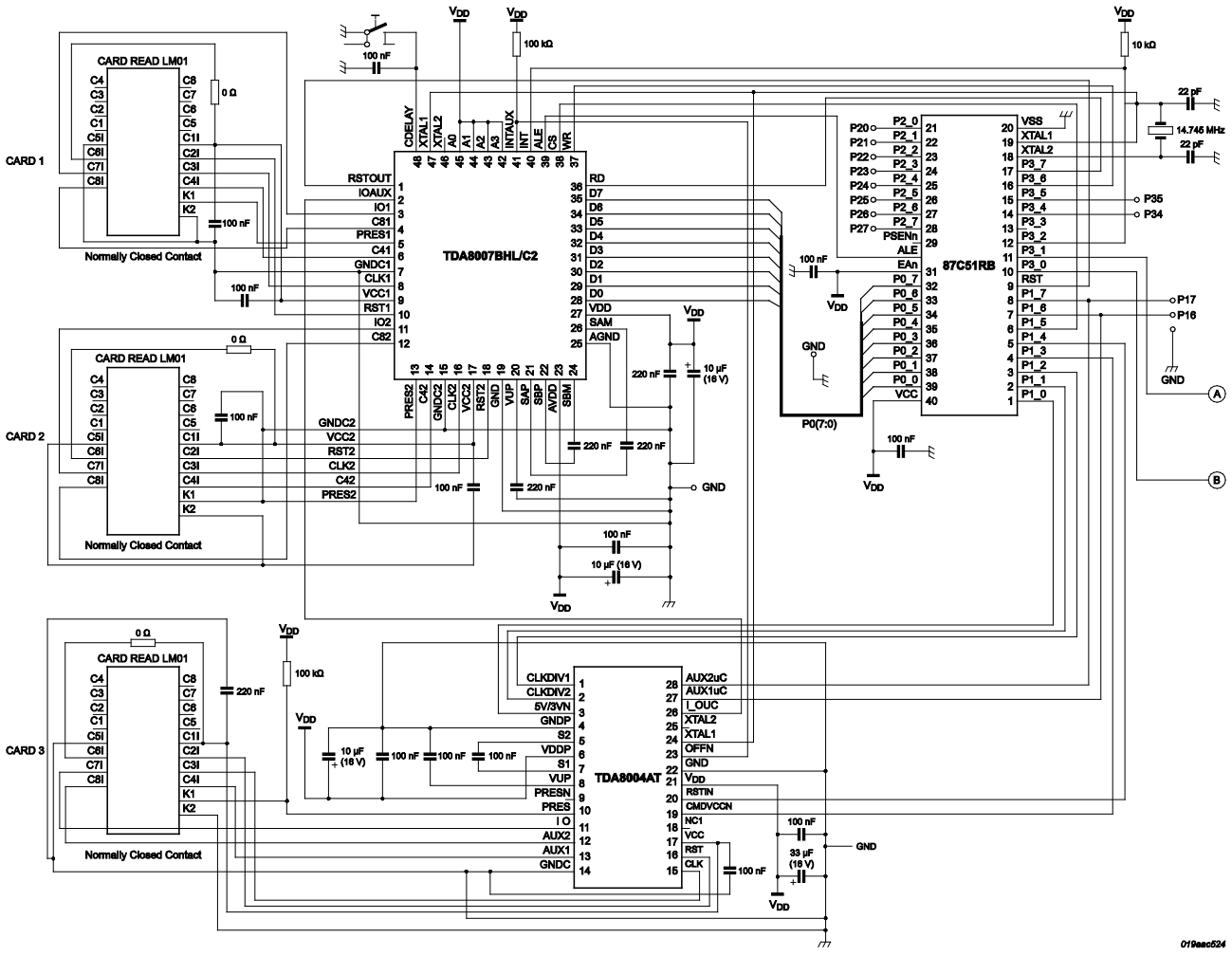


Fig 20. Reference application schematics – 3 slots, part 1

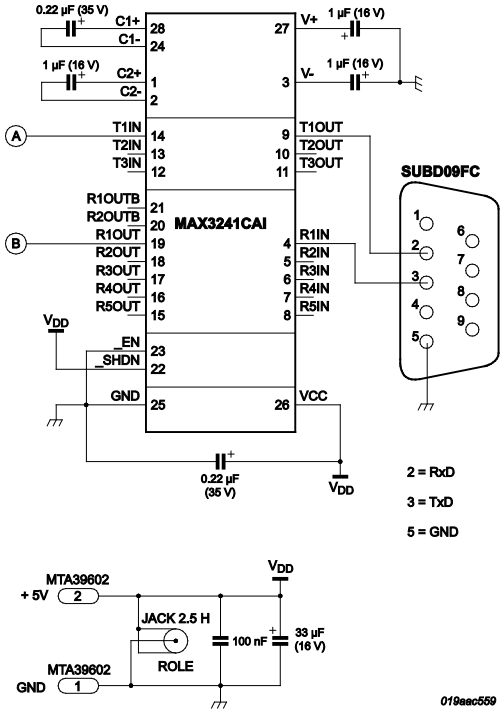


Fig 21. Reference application schematics – 3 slots, part 2

4.6.2 Four cards reader using TDA8007B and TDA8020

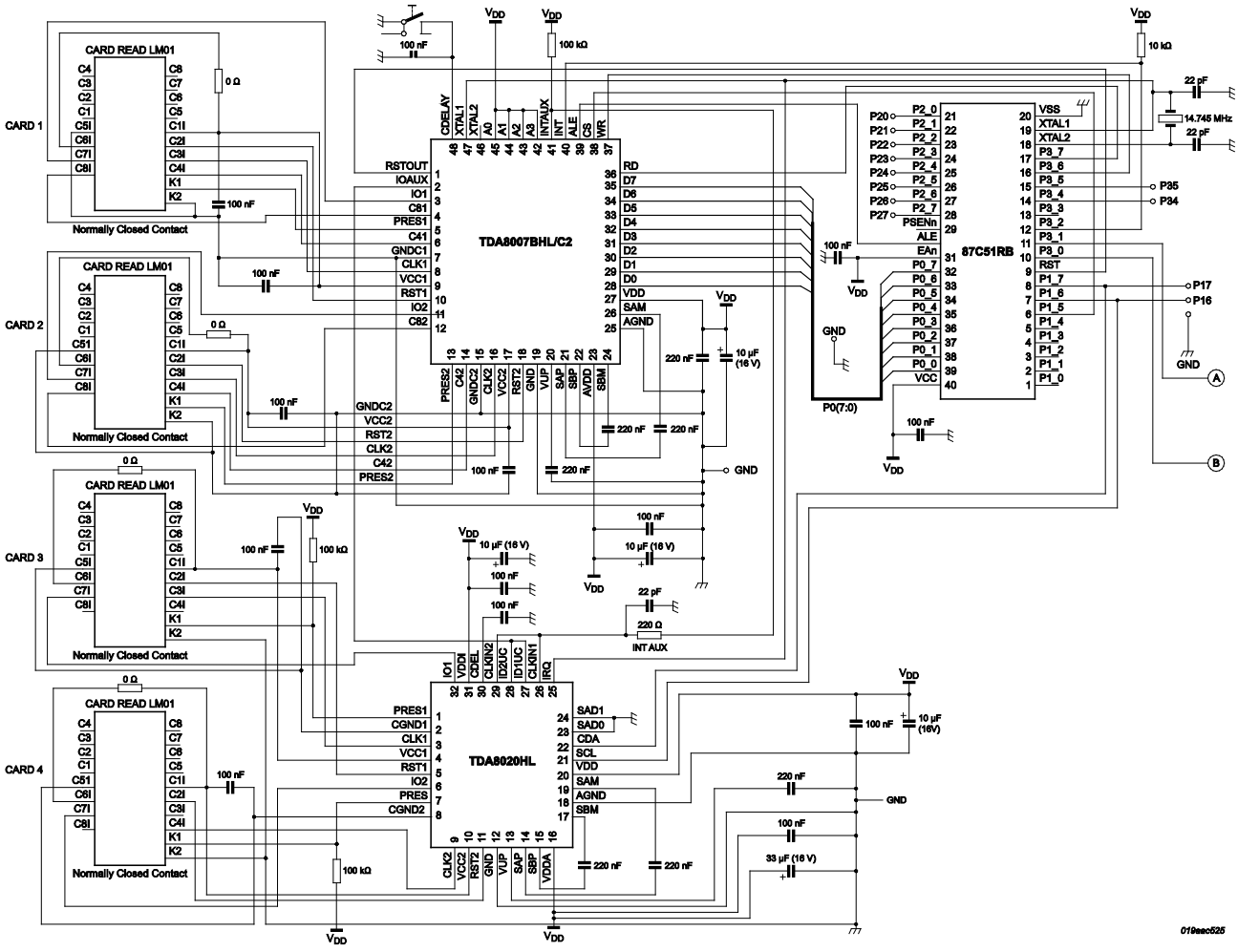


Fig 22. Reference application schematics – 4 slots, part 1

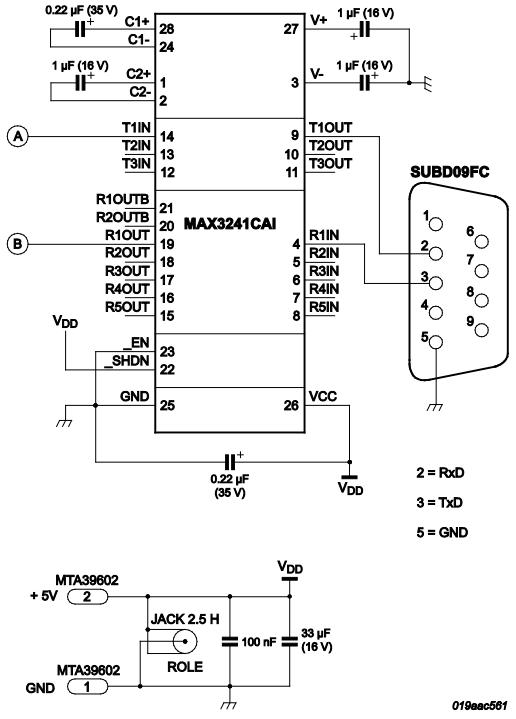


Fig 23. Reference application schematics – 3 slots, part 2

5. Conclusions

The following features give the characteristics of this reader developed for TDA8007B application:

- Double card reader with card selection
- Two or three cards activated at the same time whatever the protocol and the VCC value is
- 1.8V, 3V and 5V cards supported.
- EMV3.1.1 (tested) and ISO 7816-3 supported.
- Asynchronous protocol T=0 & T=1 supported.
- Synchronous smart cards supported (I2C, S=9, S=10, GPM896, GAM226, Eurochip).
- Automatic retransmission in T=0 protocol if parity error is detected.
- Automatic hardware deactivation, in the event of card take off, supply voltage drop, short circuit or overheating.
- Most ISO7816 baud rates supported.
- Selection of the card clock frequency possible.
- Supply voltage from 2.7V up to 6V.

6. Annex Cake 714D

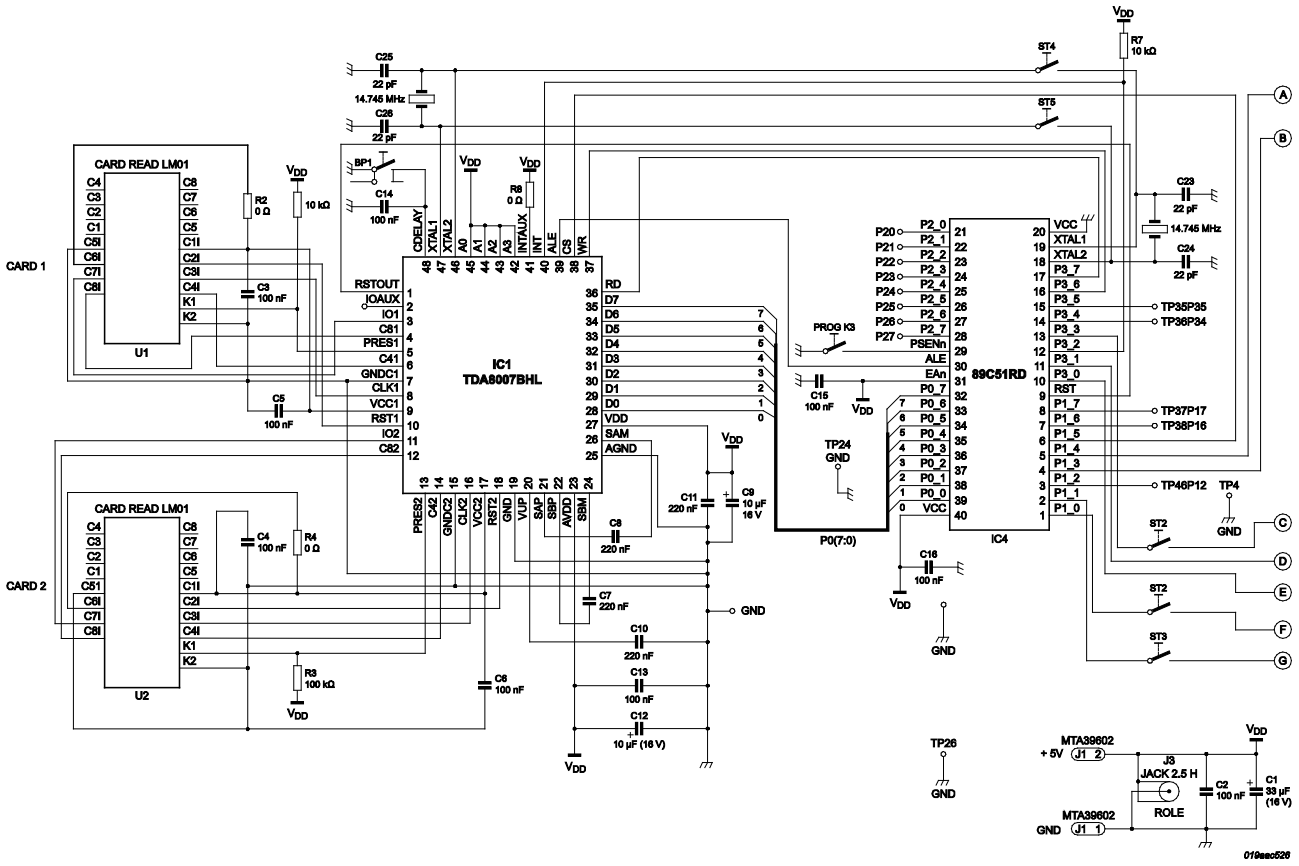


Fig 24. Cake714D schematics, part 1

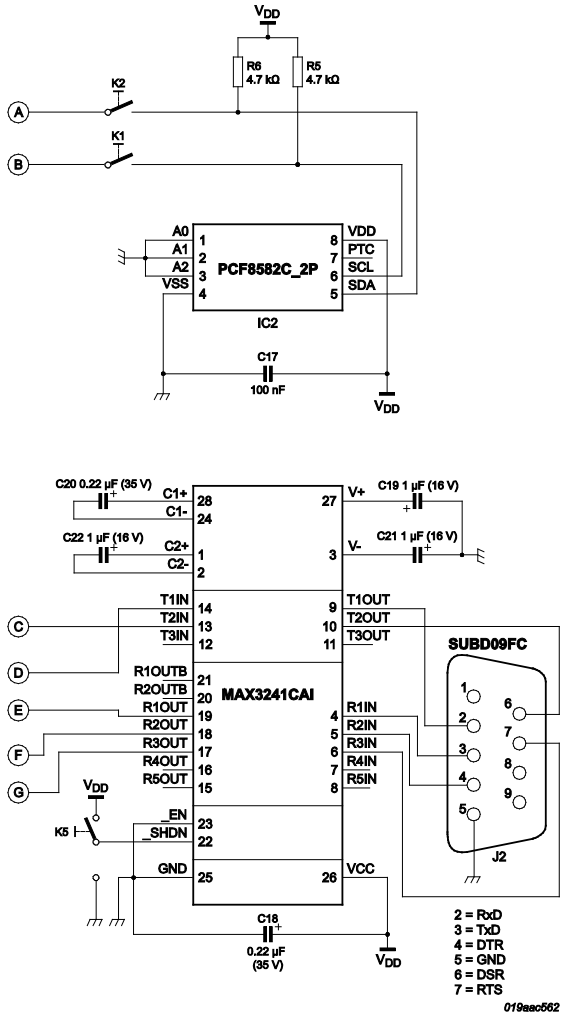


Fig 25. Cake714D schematics, part 2

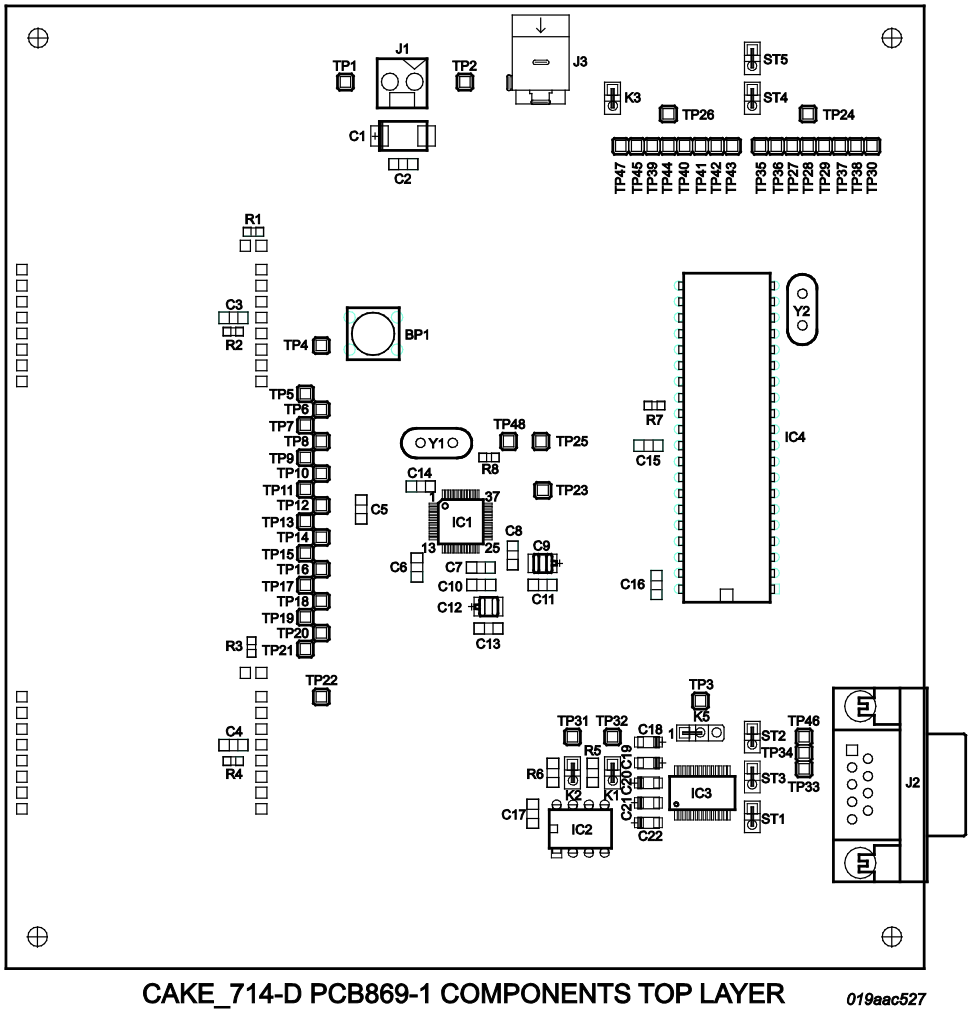


Fig 26. Cake7014D Components placement - Top

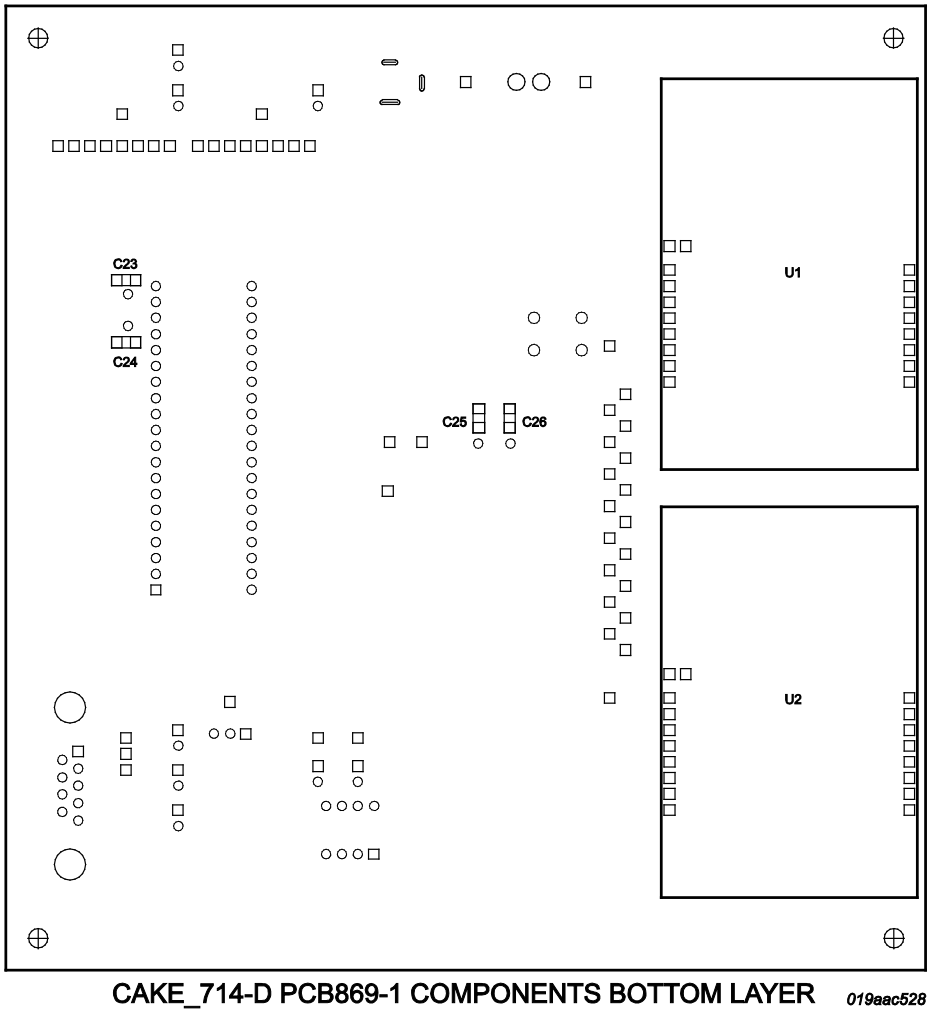


Fig 27. Cake7014D Components placement - Bottom

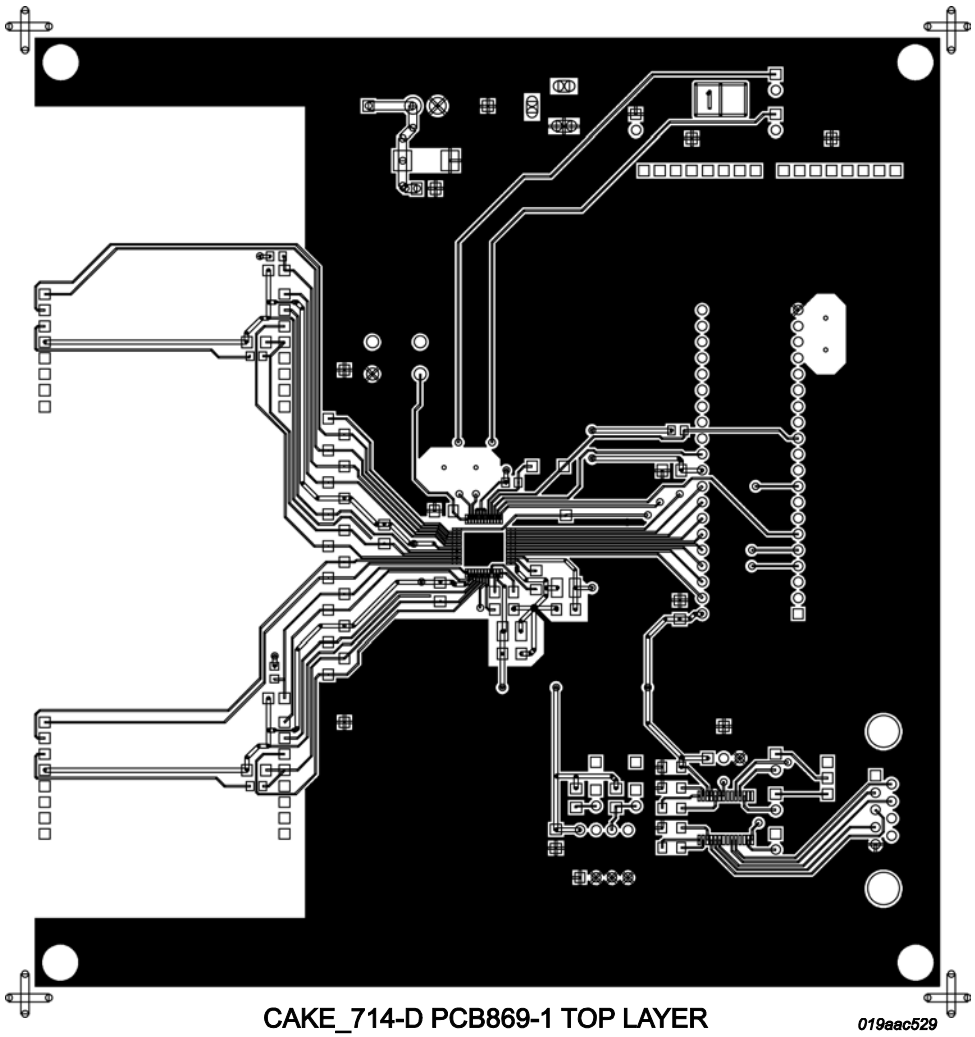


Fig 28. Cake7014D Layout - Top

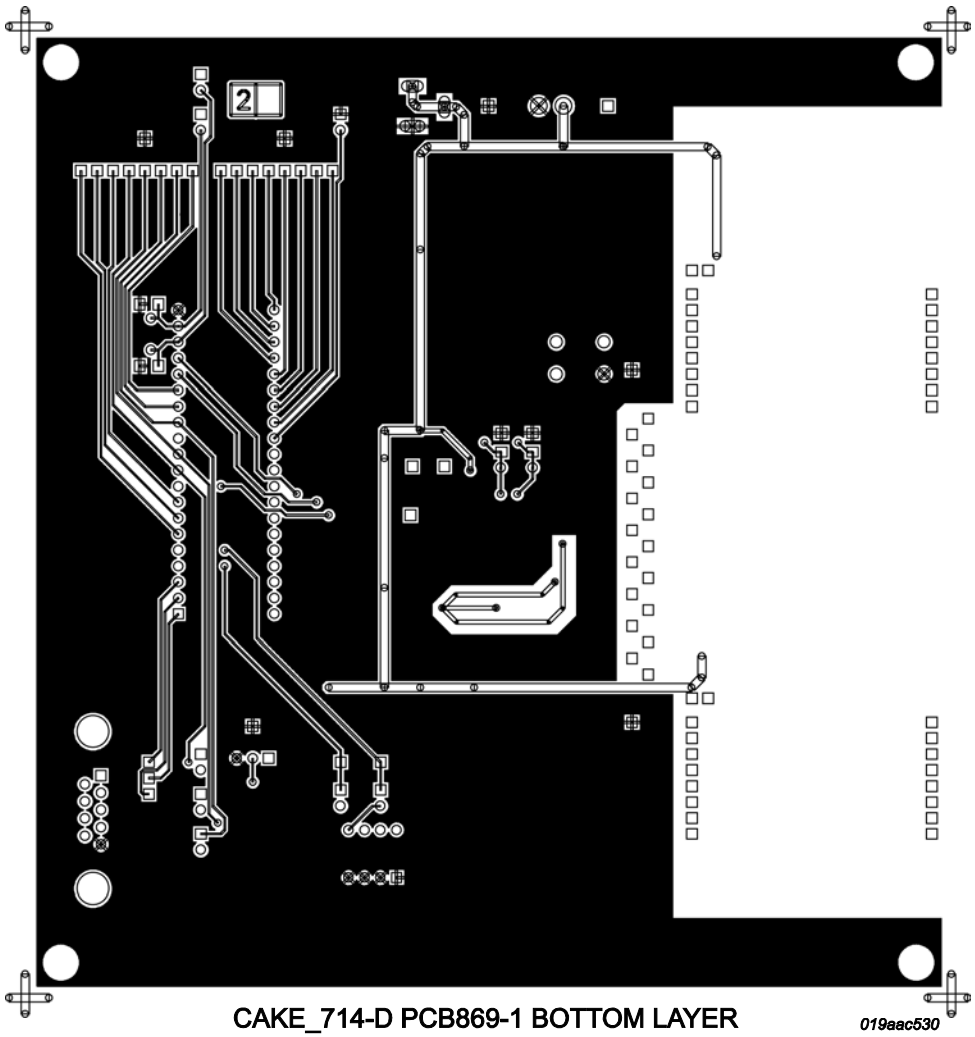


Fig 29. Cake7014D Layout - Bottom

7. Legal information

7.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

7.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

8. List of figures

9. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 27 March 2012
Document identifier: AN01054