APPLICATION NOTE

Smart Card Reader Application with TDA8029 Mask 03

AN01005

Philips
Semiconductors

PHILIPS

Abstract

This application note describes the software implemented in TDA8029 mask 03 in order to handle a communication between a system controller and an asynchronous smart card using either T=0 or T=1 protocol.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent - or other industrial or intellectual property rights.

APPLICATION NOTE

Smart Card Reader
Application with TDA8029 Mask 03

AN01005

Author(s):
Thierry LEJOSNE
Systems & Applications
Business Line Identification – Product Line RIC
Caen - France

Keywords
TDA8029
Smart card interface
ISO 7816-3&4
E.M.V. 3.1.1

Number of pages : 31

Date : 01/03/07

CONTENTS

1 INTRODUCTION

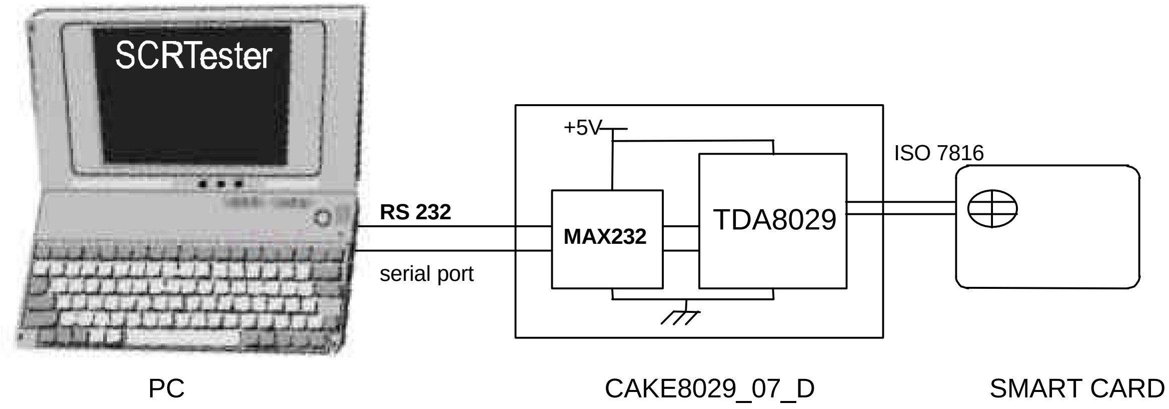
TDA8029 is a smart card coupler providing all the analogue electrical interface signals to the smart card. This coupler is able to manage asynchronous cards due to its specific ISO7816 UART and to its embedded 80C51 microcontroller core ; it can also manage synchronous cards such as I2C cards or prepaid telephone cards.

The software embedded in this device is able to support any ISO7816 asynchronous smart card (T=0 or T=1 protocol) and completely handles the communication layer between the card and the host system.

A specific protocol called «ALPAR» has been defined on the serial interface between TDA8029 and the host system; it uses the APDUs frame types to convey the asynchronous card commands and specific frames for the synchronous cards. A dedicated command has been added to carry TPDUs frames for T=1 protocol only.

A board has been built in order to demonstrate a communication between a smart card and a host system which is here a PC; this board is driven from the PC by means of a software called SCRTester.

The following diagram illustrates this application.



2 HARDWARE AND POWER MANAGEMENT

2.1 Hardware

The board CAKE 8029-07_D (see §2.3, page8) has been made to demonstrate the features of the TDA8029 with Mask 03 software.
This board is supplied under +2.7V to +6V and is connected to the PC by means of the serial port.

Depending on the PC which is used, the communication between the PC and the demoboard can be fixed at different baud rates (from 9600 to 115200 baud).
The default baud rate is 38400 and it can be changed by a special command in SCRTester.

2.2 Power management

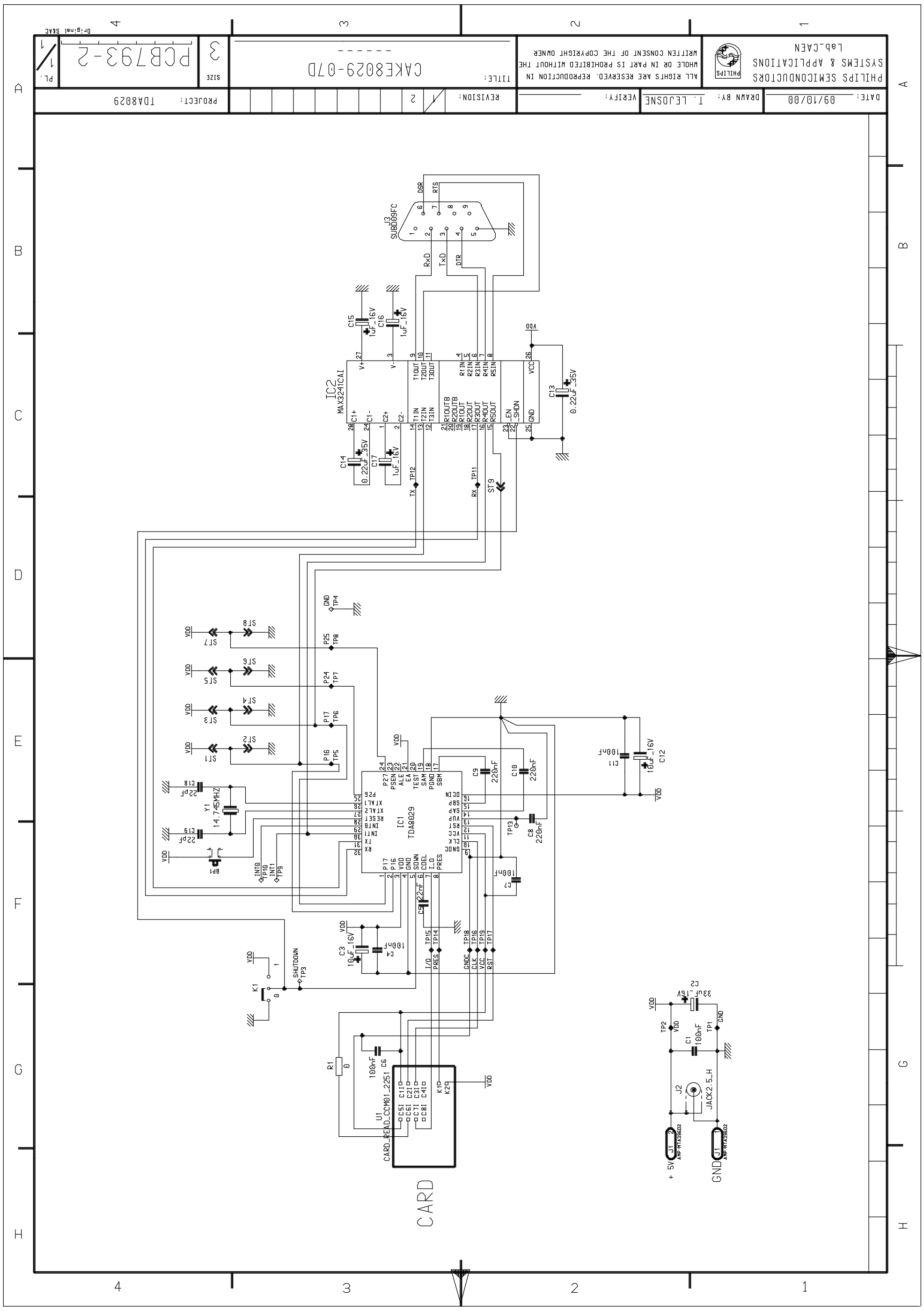
In order to benefit from the low power features of the TDA8029, the Mask 03 software implements a special management of the TDA8029.
To activate this mode P2.6 (i.e. pin 25) has to be connected to V_{DD} or to be left open at the powering on of the board (or at the reset), whereas connecting P2.6 (pin 25) to ground will force the TDA8029 to never enter into this energy saving mode.

In this mode, outside an exchange of commands between the host and the TDA8029, the card clock is either switched off (level high or low) or set to Fint/2 depending on the clock stop mode described in its ATR.
If used, no additional lines between the host controller and the TDA8029 in addition to Rx and Tx are needed.

If the card does not support the clock stop mode or does not specify it, the clock will be set to Fint/2. Note that a command (set_ESM_properties, page 21) allows to force a clock stop mode even if the card does not specify it. Depending on the card used, stop the clock rather than leave it at Fint/2 can provide a power consumption saving about 60%.

In all cases, just after having changed the clock of the card, the TDA8029 microcontroller puts itself in power down mode. A new received command from the host controller or any card event (extraction or insertion, overcurrent on VCC or RST, overheating) will wake it up.

2.3 CAKE8029-7D schematic



3 SOFTWARE ASPECT

This mask has been developed in order to be used either in ISO7816-3 or E.M.V.3.1.1 environment.
Some specific error messages are dedicated to the EMV environment (ATR parameters not allowed).

4 SERIAL INTERFACE

The serial interface between the TDA8029 and the host controller is a full duplex interface using the two lines RX and TX. RX (pin 32) is used to receive data from the host controller, TX (pin 31) is used to send data to the host controller.

No flow control or supplementary line is used (no hand check).

4.1 Data link layer

Serial data format

1	start bit
8	data bits
1	stop bit, no parity

Baud rate	38400	baud (pin 1 (P1.7) in open circuit; default configuration)
	9600	baud (pin 1 (P1.7) connected to GND)
	variable	baud rate selectable by a host command (from 9600 to 115200 baud)

Frame structure :

Data is exchanged between the system controller and TDA8029 in blocks, each made up of binary characters on one byte :

4 header characters
0 to 506 data characters (C-APDU or R-APDU)
1 LRC character

4 bytes	0 to 506 bytes	1 byte
Header	C-APDU or R-APDU	LRC
	Information field	

The 4 header bytes includes:

1st byte								2nd byte	3rd byte	4th byte
A	1	1	0	0	0	0	0			
								Data length to transmit excluding header and LRC		Command byte

A=0 ➔ Acknowledge of the frame (1st byte = 60)
A=1 ➔ Nack of the frame (message with a status error, 1st byte = E0)

LRC byte:

The LRC (Longitudinal Redundancy Check) byte is such that the exclusive-oring of all bytes including LRC is null.

4.2 General dialog structure

The system controller is the master for the transmission ; each command from the master is followed by an answer from TDA8029 including the same command byte as the input command.
The only commands which are outgoing only commands from TDA8029 are the commands related to a card insertion or extraction, a time out detection on Rx line or an automatic emergency deactivation of the card.

4.2.1 Successful command

System to TDA8029

60	XX XX	YY	nnnnnnnnnnnnnnnnnnnn	ZZ
ACK	length	code	Data (C-APDU)	LRC

TDA8029 to System

60	UU UU	YY	mmmmmmmmmmmmmmm	TT
ACK	length	code	Data (R-APDU)	LRC

The same command byte YY is returned in the answer from TDA8029.

4.2.2 Unsuccessful command

System to TDA8029

60	XX XX	YY	nnnnnnnnnnnnnnnnnnnn	ZZ
ACK	length	code	Data (C-APDU)	LRC

TDA8029 to System

E0	UU UU	YY	SS	TT
NACK	length	code	status	LRC

The status is giving the error code information (see error list, page 17).

4.2.3 Card removal

TDA8029 to System

60	00 01	A0	00	C1
ACK	Length	code	data	LRC

4.2.4 Card insertion

TDA8029 to System

60	00 01	A0	01	C0
ACK	Length	code	data	LRC

4.2.5 Answer with an acknowledge (power_off, idle_mode, power_down_mode)

System to TDA8029 (example : power_off)

60	00 00	4D	2D
ACK	Length	code	LRC

TDA8029 to System

60	00 00	4D	2D
ACK	Length	code	LRC

In the case where the answer is an acknowledge of the command, the TDA8029 sends back a frame with the same content of the command.

4.3 Dialog structure in case of Energy Saving Mode activated

As it is explained in §2.2, the Mask 03 implements a special management of the TDA8029 for energy savings purpose. This Energy Saving Mode is activated when P2.6 (i.e. pin 25) is tied to V_{DD} of left open at reset of the TDA8029.

Due to this Energy Saving Mode implementation, the serial interface is adapted as follows (see Figure 1, page 14).

4.3.1 Communication initiated by the system controller

To initiate a normal exchange with the TDA8029, the host has to first send a specific frame composed of only one character (0xAA) to wake up the TDA8029. When the TDA8029 is completely waked up, it sends an acknowledged frame back to the host.

System to TDA8029 (Waking up frame)

AA
code

TDA8029 to System

60	00 00	BB	DB
ACK	length	code	LRC

After this synchronisation stage, the operational communication can start as described in paragraph §4.2.

System to TDA8029

60	XX XX	YY	nnnnnnnnnnnnnnnnnnn	ZZ
ACK	length	code	Data (C-APDU)	LRC

TDA8029 to System

60 or E0	XX XX	YY	nnnnnnnnnnnnnnnnnnn	ZZ
ACK or NACK	length	code	Data (C-APDU)	LRC

After having sent its complete frame, the TDA8029 returns in power down.

4.3.2 Communication initiated by the TDA8029

Any card event (extraction or insertion, overcurrent on VCC or RST, overheating) will wake up the TDA8029.
In that case, this is the TDA8029 which initiates the communication.
It first sends the following specific frame to warn the host that it wants to send data. This frame is the same than the one used to acknowledge a received waking-up character (§4.3.1) expected that the length in that case is equal to one (equal to zero in case of a normal acknowledge frame).

TDA8029 to System

60	00 01	BB	01	DB
ACK	length	code	data	LRC

After that, the TDA8029 waits until the host replicates this frame.

System to TDA8029

60	00 01	BB	01	DB
ACK	length	code	data	LRC

Then, the TDA8029 sends the informational frame and returns in power down mode.

TDA8029 to System

60 or E0	XX XX	YY	nnnnnnn	ZZ
ACK or NACK	length	code	Data	LRC

4.3.3 Abnormal communication process

Three different cases can be met :

- time out detected on Rx line (more than 10 ms between the leading edge of two characters from the command frame sent by the host controller). As soon as the time out is detected, the TDA8029 sends back an error frame :

System to TDA8029	0xAA	
TDA8029 to system	0x60 0x00 0x00 0xBB 0xDB	
System to TDA8029	ACK Length Code (Parameters) LRC	(erroneous frame)
TDA8029 to system	NACK 0x00 0x01 Code 0xFF LRC	

Then the TDA8029 goes back to power down mode.

- card extraction detected during a card IO card session : if the TDA8029 detects a card extraction as it is processing an APDU with the card, it returns two consecutive messages back to the host controller

System to TDA8029	0xAA	
TDA8029 to system	0x60 0x00 0x00 0xBB 0xDB	
System to TDA8029	ACK Length 0x00 Parameters LRC	
TDA8029 to system	NACK 0x00 0x01 0x00 0xC0 LRC	(card absent)
TDA8029 to system	ACK 0x00 0x01 0xA0 0x00 LRC	(card extraction)

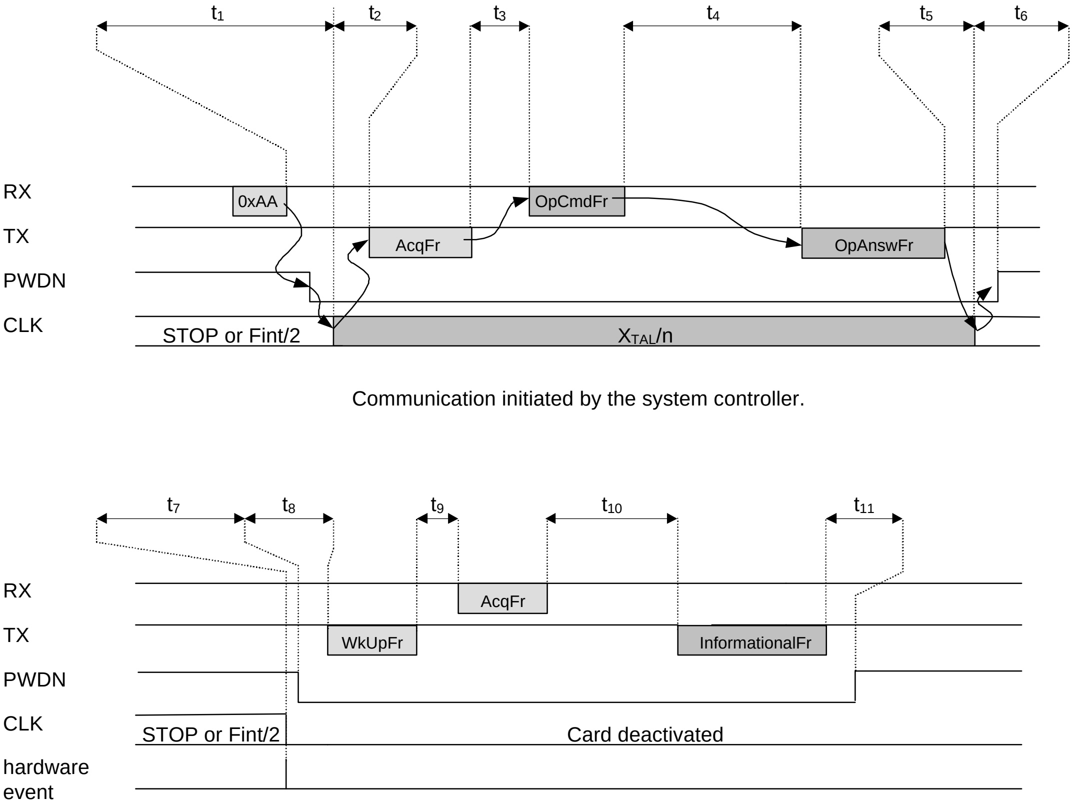
Then the TDA8029 goes back to power down mode.

- unexpected reception detected during a communication process ; the TDA has not finished to process a received command frame (it has not sent completely its answer frame while the host controller sends a new command frame) : in that case, the TDA8029 sends the correct answer to the first received command and then sends the error frame informing the host controller that it has lost at least a command frame.

System to TDA8029	0xAA
TDA8029 to system	0x60 0x00 0x00 0xBB 0xDB
System to TDA8029	ACK Length Code1 (Parameters) LRC
System to TDA8029	ACK Length Code2 (Parameters) LRC
TDA8029 to system	ACK or NACK Length Code1 (Parameters) LRC
TDA8029 to system	NACK 0x00 0x01 Code2 0xF1 LRC

Then the TDA8029 goes back to power down mode.

4.3.4 Timing considerations



Communication initiated by the system controller.

Communication initiated by the TDA8029.
(in this example, card deactivated due to a hardware event)

Figure 1. : Serial interface

- t_1 : Waking up and clock switching time (typically 550 µs).
- t_2 : TDA8029 reaction time (typically 33 µs).
- t_3 : Host-dependant reaction time. (no limit).
- t_4 : Process time (depends on the type of the command frame).
- t_5 : Clock switching time (typically 80 µs).
- t_6 : Power Down setting time (typically 0.3 ms).
- t_7 : Waking up and clock switching time (typically 550 µs).
- t_8 : TDA8029 reaction time (typically 630 µs).
- t_9 : Host-dependant reaction time. (no limit).
- t_{10}: TDA8029 reaction time (typically 630 µs).
- t_{11}: Power Down setting time (typically 0.3 ms).

5 COMMAND BYTES

The following command bytes are available:

Command	*Code*	*Answer from reader*	*(page)*
card_command (APDU)	00_H	Card response (APDU) or error message	*(24)*
process_T=1_command	01_H	T=1 frame or error message	*(24)*
check_pres_card	09_H	Indication of the card presence	*(20)*
send_num_mask	$0A_H$	1 parameter giving the mask number	*(20)*
set_card_baud_rate	$0B_H$	Acknowledge	*(26)*
IFS_request	$0C_H$	Acknowledge or error message	*(25)*
set_serial_baud_rate	$0D_H$	Acknowledge or error message	*(20)*
negotiate (PTS)	10_H	Acknowledge or error message	*(25)*
set_clock_card	11_H	Acknowledge or error message	*(26)*
power_off	$4D_H$	Acknowledge	*(24)*
power_up_iso	69_H	ATR from card + status byte or error message	*(24)*
power_up_3V	$6D_H$	ATR from card + status byte or error message	*(23)*
power_up_5V	$6E_H$	ATR from card + status byte or error message	*(23)*
idle_mode (clock stop low)	A2H	Acknowledge	*(21)*
power_down_mode	A3H	Acknowledge	*(22)*
idle_mode (clock stop high)	A4H	Acknowledge	*(22)*
set_NAD	$A5_H$	Acknowledge or error message	*(27)*
get_card_param	$A6_H$	Fi, Di, CLK, T of the card in use or error message	*(27)*
host_ready	BB_H	Informational message	*(12)*
set_ESM_properties	BC_H	Acknowledge	*(21)*
read_IO	CE_H	Value on the IO pins	*(28)*
set_IO	CF_H	Acknowledge	*(28)*

Outgoing commands (only) :

	Code	*Parameter*
Card_take_off	$A0_H$	00_H
Card_insertion	$A0_H$	01_H

These commands are sent as soon as a card is inserted or extracted without any command coming from the system. These commands use the same operating code but the extra parameter gives the additional information.

These outgoing commands are sent only when the host is waiting for a reply or is in stand by; when the card is extracted whereas the host is sending a frame to TDA8029, the card_take_off message will be sent from TDA8029 only when it has received the complete frame coming from the host controller. This system prevents any conflict on the serial line.

	Code	*Parameter*
Card deactivated	XX_H	$A1_H$
Time_out	XX_H	FF_H
Lost frame	XX_H	$F1_H$

Card deactivated: The card is deactivated due to a hardware problem (short on Vcc, overcurrent)

Time_out: Time out problem on (TDA8029) Rx line
This command is used in order to warn the host controller that the last communication has broken down (time out problem) so that the Rx line of TDA8029 does not remain blocked.
The time out condition is a silence greater than 10 ms in the host command frame.

Lost frame: An unexpected host controller command frame has been received by the TDA8029 while it was busy to process a previous command frame.

In these three commands, the code value is the previous code value used during a normal exchange.

6 ERROR LIST

The error list gives the status code identification and a brief signification of the status error code.

6.1 Exhaustive list of possible error code

Status code	*Meaning*
08_H	Length of the data buffer too short
$0A_H$	3 consecutive errors from the card in T=1 protocol
20_H	Wrong APDU
21_H	Too short APDU
22_H	Card mute now (during T=1 exchange)
24_H	Bad NAD
25_H	Bad LRC
26_H	Resynchronized
27_H	Chain aborted
28_H	Bad PCB
29_H	Overflow from card
30_H	Non negotiable mode (TA2 present)
31_H	Protocol is neither T=0 nor T=1 (negotiate command)
32_H	T=1 is not accepted (negotiate command)
33_H	PPS answer is different from PPS request
34_H	Error on PCK (negotiate command)
35_H	Bad parameter in command
38_H	TB3 absent
39_H	PPS not accepted (no answer from card)
$3B_H$	Early answer of the card during the activation
55_H	Unknown command
80_H	Card mute (after power on)
81_H	Time out (waiting time exceeded)
83_H	4 parity errors in reception
84_H	4 parity errors in transmission
86_H	Bad FiDi
88_H	ATR duration greater than 19200 etus (E.M.V.)
89_H	CWI not supported (E.M.V.)
$8A_H$	BWI not supported (E.M.V.)
$8B_H$	WI (Work waiting time) not supported (E.M.V.)
$8C_H$	TC3 not accepted (E.M.V.)
$8D_H$	Parity error during ATR
90_H	3 consecutive parity errors in T=1 protocol
91_H	SW1 different from 6X or 9X
92_H	Specific mode byte TA2 with b5 byte=1
93_H	TB1 absent during a cold reset (E.M.V.)
94_H	TB1different from 00 during a cold reset (E.M.V.)
95_H	IFSC<10H or IFSC=FFH
96_H	Wrong TDi
97_H	TB2 is present in the ATR (E.M.V.)

98_H	TC1 is not compatible with CWT
$9B_H$	Not T=1 card
$A0_H$	Procedure byte error
$A1_H$	Card deactivated due to a hardware problem
$C0_H$	Card absent
$C3_H$	Checksum error
$C4_H$	TS is neither 3B nor 3F
$C6_H$	ATR not supported
$C7_H$	VPP is not supported
$E1_H$	Card clock frequency not accepted (after a set_clock_card command)
$E2_H$	UART overflow
$E3_H$	Supply voltage drop-off
$E4_H$	Temperature alarm
$E9_H$	Framing error
$F0_H$	Serial LRC error
$F1_H$	At least one command frame has been lost
FF_H	Serial time out

6.2 Error code for each command

COMMAND		POSSIBLE RETURNED ERROR CODE
Power UP 3V, 5V		31h, 35h, 38h, 3Bh, 80h, 85h, 86h, 88h, 89h, 8Ah, 8Bh, 8Ch, 8Dh, 92h, 93h, 94h, 95h, 96h, 97h, 98h, C0h, C3h, C4h, C6h, C7h, E2h, E3h, E4h, E9h, F0h, F1h, FFh
Power_up_iso		31h, 35h, 3Bh, 80h, 96h, C0h, C3h, C4h, C6h, C7h, E2h, E3h, E4h, E9h, F0h, F1h, FFh
Card Command	T=0	08h, 20h, 21h, A1h, 81h 83h, 84h, 91h, A0h, C0h, E2h, E3h, E4h, E9h, F0h, F1h, FFh
	T=1	08h, 22h, 24h, 25h, 26h, 27h, 28h, 29h, A1h, 83h, 90h, C0h, E2h, E3h, E4h, E9h, F0h, F1h, FFh
Negotiate		30h, 31h, 33h, 34h, 35h, 39h, A1h, C0h, E2h, E3h, E4h, E9h, F0h, F1h, FFh
Set Clock Card		C0h, E1h, F0h, F1h, FFh
Set card baud rate		86h, C0h, F0h, F1h, FFh
Set NAD		24h, F0h, F1h, FFh
Get card parameters		A1h, C0h, F0h, F1h, FFh
IFS request		0Ah, A1h, 9Bh, C0h, E2h, E3h, E4h, E9h, F0h, F1h, FFh
Send mask number		F0h, F1h, FFh
Check presence card		F0h, F1h, FFh
Set serial baud rate		F0h, F1h, FFh
Power off		F0h, F1h, FFh
Set ESM properties		F0h, F1h, FFh
Idle mode clock stop low and high		F0h, F1h, FFh
Power down mode		F0h, F1h, FFh

7 COMMANDS DESCRIPTION

7.1 General commands

7.1.1 send_num_mask

This command is used to identify the software version which is masked in TDA8029 ROM.

For example the current software will be coded as : “03 Release 1.0” (14 ASCII characters)
System to TDA8029 :   60   00 00   0A   6A
TDA8029 to System :   60   00 0E   0A   30 33 20 52 65 6C 65 61 73 65 20 31 2E 30   01

7.1.2 check_card_presence

This command is used to check the presence of a card.

System to TDA8029 :   60   00 00   09   69
TDA8029 to System :   60   00 01   09   PRES   LRC

Where PRES indicates the presence of a card (00 if there is no card, 01 if a card is present).

7.1.3 set_serial_baud_rate

This command is used for changing the baud rate onto the serial link between the host and the interface card. The default value is set to 38400 baud or 9600 baud depending on the state of the pin 1.

A parameter has to be transmitted in order to choose the baud rate :

System to TDA8029 :   60   00 01   0D   PAR   LRC
TDA8029 to System :   60   00 00   0D   LRC

Baud rate (Baud)	Parameter
4800	00
9600	01
19200	02
38400	03
57600	04
76800	05
115200	06

After a baud rate change, the new value takes place for the next command sent by the host.

7.1.4 set_ESM_properties

This command is used to fix the behaviour of the TDA8029 if the Energy Saving Mode is activated.
By default, the clock stop information contained in the ATR of the activated card is used during the session with this card to set the clock when the TDA8029 enters in power down mode. Thus, if the card does not explicitly indicate that it supports clock stop mode (High or Low), the card clock will be set to Fint/2.
To save even more energy during these periods, the clock mode can be forced using this command.

```
System to TDA8029 :   60   00 02   BC   STOP STATE   LRC
TDA8029 to System :   60   00 00   BC   DC
```

Where STOP indicates the clock stop request
(00 to leave the clock at Fint/2 , 01 to stop the clock)
STATE is the clock stop level if requested
(00 to stop clock LOW , 01 to stop clock HIGH)

Note 1 : This command can be used only when the Energy Saving Mode is activated. If launched when the Energy Saving Mode is not activated, an *UNKNOWN_COMMAND* error will be returned by the TDA8029.

Note 2 : Once this command has been launched to the TDA8029, all the further activations of cards will use the defined behaviour defined within this command. One has to use this command again to change the behaviour, e.g. to come back to a clock at Fint/2. Furthermore, even if this command is used to force a clock mode, when a card with defined clock stop conditions is encountered, the clock stop mode indicated in the card’s ATR will be used.

7.1.5 time_out

This command is sent from TDA8029 to the host controller if, during a transmission from the host controller to TDA8029, the time interval between 2 characters exceeds 10ms. This timing is calculated between each character of a frame, starts after the first character, and is disabled after the last character of the frame. This feature has been implemented in order to avoid any blocking of the transmission line between the host controller and TDA8029.

```
TDA8029 to System :   E0   00 01   6F   FF   71
```

7.1.6 idle_mode (clock stop low)

This command is used to set the controller in idle mode. The card, if activated, has its clock (CLK) set to low level but is still active.
Any command from the host on the serial line will wake up the device.

```
System to TDA8029 :   60   00 00   A2   C2
TDA8029 to System :   60   00 00   A2   C2
```

Note : This command can be used only when the Energy Saving Mode is not activated. If launched when the Energy Saving Mode is activated, an *UNKNOWN_COMMAND* error will be returned by the TDA8029.

7.1.7 idle_mode (clock stop high)

This command is used to set the controller in idle mode. The card, if activated, has its clock (CLK) set to high level but is still active.
Any command from the host on the serial line will wake up the device.

System to TDA8029 : 60 00 00 A4 C4
TDA8029 to System : 60 00 00 A4 C4

Note : This command can be used only when the Energy Saving Mode is not activated. If launched when the Energy Saving Mode is activated, an *UNKNOWN_COMMAND* error will be returned by the TDA8029.

7.1.8 power_down_mode

This command is used to set the controller in power down mode; if the card is active, it is then deactivated. Exiting this mode is possible with a hardware reset of TDA8029 or an external interruption (INT0, INT1 or Rx).

System to TDA8029 : 60 00 00 A3 C3
TDA8029 to System : 60 00 00 A3 C3

Note : This command can be used only when the Energy Saving Mode is not activated. If launched when the Energy Saving Mode is activated, an *UNKNOWN_COMMAND* error will be returned by the TDA8029.

7.2 Asynchronous card related commands

7.2.1 *power_up commands*

There are three different power up commands (3V, 5V or ISO). Two of them (power_up_3V and power_up_5V) have to be followed by a parameter :

- 00_H indicates that all the parameters of the ATR of the card compliant with ISO7816-3 will be taken into account.
- 01_H indicates that only the ATR of cards whose parameters are inside the E.M.V.3.1.1 specification scope will be taken into account; cards having an ATR which does not comply with EMV3.1.1 requirements will be rejected.

7.2.1.1 power_up_3V

This command allows to activate the card at a VCC of 3V. All the signals going to the card will be referenced to this VCC=3V.
An activation sequence is processed following the ISO7816-3 normalization (VCC is rising, I/O is enabled, CLK is started, and RST is processed). If the card answers to this command, the answer will content all the ATR parameters; these parameters are memorized in TDA8029 and will be taken into account during the whole card session (till the card is deactivated or till a warm reset is processed). The structure of the answer is the following:

System to TDA8029

60	00 01	6E	01	0F
ACK	length	code	EMV	LRC

TDA8029 to System

60	XX XX	6E	nnnnnnnnnnnnnnnnnnn	ZZ
ACK	length	code	ATR parameters	LRC

If the card is in specific mode, TDA8029 will process the next command directly using the new interface parameters of this specific mode. If the card proposes a different Fi/Di in the ATR than the default value (Fi/Di=372), it is up to the application to make a PPS command by using the negotiate command. If the card proposes 2 different protocols in its ATR, it is up to the application to make a PPS command by using the negotiate command.
If the card does not answer to the reset, a status giving an error code is returned to the application.

In the case of EMV compliant power up, if the card is using T=1 protocol, just after having received the ATR, TDA8029 sends an IFSD request to the card indicating that the reader can manage a data buffer of 254 bytes (FEH).

The power_up_3V command can be used to generate a warm reset if the card is already activated.

7.2.1.2 power_up_5V

This command allows to activate the card at a VCC of 5V. Every signal going to the card will be referenced to this VCC=5V.
See power_up_3V for the other characteristics.

7.2.1.3 power_up_iso

This command does not need any argument. The principle consists to activate the card as described in ISO 7816-3 :

- attempt to activate the card at a VCC of 3V, if the cards answers correctly and if it indicates in its ATR that it is a class A or a class AB card (TAi with T=15), then the command is finished and the ATR is returned to the host,
- if in the previous stage, the card did not answer correctly or did not specify in its ATR that it was a class A or a class AB card, a new activation of the card is launched at 5V. If the card does not answer to the reset, a status giving an error code is returned to the application, otherwise the answer contains all the parameters of the card.

See power_up_3V for the other characteristics (when parameter of the command is ISO, not EMV).

7.2.2 power_off

This command is used to deactivate the card whatever it has been activated for 3V or 5V operation. A deactivation sequence is processed following the ISO 7816-3 normalization in about 100µs.

```
System to TDA8029 :   60   00 00   4D   2D
TDA8029 to System :   60   00 00   4D   2D
```

7.2.3 card_command (APDU)

This command is used to transmit card commands under APDU format from system to TDA8029 whatever T=0 or T=1 protocol are used. Short or extended commands (see limitations in §8.1) can be used.
An answer to such a command is also made in APDU format from TDA8029 to the system.

Example :

```
System to TDA8029 :   60   00 07   00   00 A4 00 00 02 4F 00   8E
TDA8029 to System :   60   00 02   00   90 00   F2
```

7.2.4 process_T=1_command

This command may be used if the application layer provides the complete T=1 frame including prologue, information and epilogue fields. If it is not the case, the above `card_command` opcode shall be used.
This command is used from the application layer in order to send a complete T=1 frame to the card. This command includes the specific framing used in T=1 protocol (Prologue Field, Information Field, Epilogue Field) and will be sent transparently to the card. The answer from the card will be sent as a complete T=1 frame to the application layer. The internal timing of a block (Character Waiting Time) will be handled by TDA8029. The block Waiting Time will also be controlled by TDA8029. In case of Waiting Time Extension request (WTX) from the card, it will be taken into account by the TDA8029.

System to TDA8029	60	XX XX	01	NAD PCB LEN A_1 A_2 …… A_N EDC	LRC
TDA8029 to System	60	00 06	01	NAD PCB LEN SW1 SW2 EDC	LRC

Where A1 A2…..An is information field sent to the card
XX XX is the length of the frame from NAD to EDC

In case of chaining :

System to TDA8029	60	00 XX	01	NAD 20 LEN A_1 A_2 …… A_N EDC	LRC
TDA8029 to System	60	00 04	01	NAD 90 00 EDC	LRC
System to TDA8029	60	00 YY	01	NAD 40 LEN A_{N+1} A_{N+2} … … A_Z EDC	LRC
TDA8029 to System	60	00 ZZ	01	NAD PCB LEN D_1 D_2 … … D_N EDC	LRC

7.2.5 negotiate

This command is used to make a PPS (Protocol and Parameter Selection) to the card, if in its ATR the card proposes a different Fi/Di or 2 different protocols. By using this command a PPS will be made to the card with the Fi or Di and protocol type entered as a parameter (PP). It is up to the host to make the correct Fi/Di submission to the card.

Example :
System to TDA8029 : 60 00 02 10 PP FD LRC
TDA8029 to System : 60 00 00 10 70

Where FD is the ratio Fi/Di given by TA1 parameter of the ATR and PP is the protocol to be used.

If the command is acknowledged, any subsequent exchanges between the card and TDA8029 will be made by using the new parameters.

7.2.6 IFSD request

This command is used to send a S(IFS request) block to the card indicating the maximum length of information field of blocks which can be received by the interface device in T=1 protocol. The initial size following the answer to reset is 32 bytes and this size shall be used throughout the rest of the card session or until a new value is negotiated by the terminal by sending a S(IFS request) block to the card.
In EMV mode, the IFSD size is automatically negotiated to 254 just after the ATR has been received.

System to TDA8029 : 60 00 01 0C PAR LRC
TDA8029 to System : 60 00 00 0C 6C

Where PAR is the IFSD size.

7.2.7 set_clock_card

This command is used for changing the card clock frequency. The default value is set to FXTAL/4 which is 3.68625 MHz.
A parameter has to be transmitted in order to choose the card clock frequency:

System to TDA8029 : 60 00 01 11 PAR LRC

Frequency	Parameter
Fxtal =14.745MHz	00
Fxtal/2=7.37MHz	02
Fxtal/4=3.68MHz	04
Fxtal/8=1.84MHz	06

After a card clock frequency change, all the waiting times are internally set to the new value.

Before applying the requested clock, the compatibility of the frequency with the current Fi used by the card is checked as described in ISO7816-3. For example, if the card has answered in its ATR a Fi parameter of 372 or 558 (fmax ≤ 6MHz), a change of the card clock frequency to Fxtal (14.745MHz) or Fxtal/2 (7.37MHz) will not be processed and an error status will be sent to the application.

7.2.8 card_take_off and card_insertion

These two commands are sent directly to the system processor as soon as a card extraction or insertion has occurred.

```
TDA8029 to System : 60  00 01  A0  00  C1        for a card extraction
                    60  00 01  A0  01  C0        for a card insertion.
```

7.2.9 set_card_baud_rate

This command is used mainly for cards which are not fully ISO 7816-3 compliant with specific and negotiable modes. As a matter of fact some cards are in specific mode but they do not give TA2 parameter in their answer to reset. So the UART has to be set to the right baud rate by means of this specific command which programs the baud rate. For non ISO baud rates there is a possibility to increase the capability of the reader by setting the bit CKU which divides by 2 the number of clock cycles of the etu and thus doubles the baud rate of the ISO UART.

Example :
System to TDA8029 : 60 00 02 0B XX CKU LRC
TDA8029 to System : 60 00 00 0B LRC

Where XX is the value of FiDi
if CKU=0, the baud rate is defined by FiDi
if CKU=1, the baud rate is 2 * the baud rate is defined by FiDi

For an etu of 372 clock cycles : XX=FiDi=0x11
prescaler = 31, divider = 12 → 31 * 12 = 372, CKU=0.

7.2.10 set_NAD

This command is used from the application layer in order to specify a SAD (source address) and a DAD (destination address) for a logical connection using T=1 protocol as defined in ISO7816-3. The default value is 00 and will be kept until the send NAD command has been notified to the TDA8029. Any NAD submission where SAD and DAD are identical (except 00) will be rejected. If bits b4 or b8 of the NAD required are set to 1 (VPP programming) the NAD will be rejected.
The NAD shall be initialised before any information exchange with the card using T=1 protocol, otherwise and error message will be generated.

System to TDA8029 : 60 00 01 A5 NAD LRC
TDA8029 to System : 60 00 00 A5 LRC

Where NAD is the new value of NAD immediately taken into account.

7.2.11 get_card_param

This command is used from the application level in order to get the Fi and Di parameters of the card in use, the current card clock frequency, and the protocol in use.
FiDi parameter will be given on one byte (FiDi), the card clock frequency on one byte (CC), and the protocol on one byte (TT).
FiDi will give the value of the current Fi Di (Example 11H for Fi=372 and Di=1)
CC will take value 01 H for Fxtal, 02H for FXtal/2, 04H for FXtal/4 and 08H for FXtal/8.
TT will take value 00H for protocol T=0 and value 01H for protocol T=1.

If there is no card in use, an error message will be generated.

System to TDA8029 : 60 00 00 A6 C6
TDA8029 to System : 60 00 03 A6 FiDi CC TT LRC

Where FIDI gives the current FIDI coded as in TA1 parameter,
CC gives the value of the card clock frequency as coded in CCR register of TDA8029,
TT gives the protocol used by the card (00 for protocol T=0, 01 for protocol T=1).

7.3 General purpose IO commands

7.3.1 read_IO

This command is used to read the current state of the four general purpose IO of the TDA8029.

System to TDA8029 : 60 00 00 CE AE
TDA8029 to System : 60 00 01 CE VAL LRC

VAL is coded as follows :

0	0	0	0	P2.7	P2.6	P1.7	P1.6

0 if IO=GND, 1 if IO=V_{DD}

7.3.2 set_IO

This command is used to set one of the general purpose IO of the TDA8029 to a specified logic level.

System to TDA8029 : 60 00 02 CF IO VAL LRC
TDA8029 to System : 60 00 00 CF AF

Where :

- IO is coded as follows,

IO	Port
0x01	P1.6
0x02	P1.7
0x03	P2.6
0x04	P2.7

- VAL is the value to apply to the IO.

VAL	level to apply
0x00	GND
0x01	V_{DD}

8 INFORMATION FIELD FOR ASYNCHRONOUS CARDS

The data buffer has a size of 512 bytes whose 6 bytes located at the end of the buffer are used by the internal library; so the data buffer has a real size of 506 bytes.
The information field that can include up to 506 bytes is composed of APDUs (Application Protocol Data Unit) according to the ISO7816-4 normalization definition.
Different examples are given according to Annex A of the EMV'96 in T = 0.

TAL (System) **TTL (TDA8029)**

Case 1 command

$\{\underbrace{60, 00, 04, 00}_{\text{4 header bytes}}, CLA, INS, P1, P2, LRC\}$ ⇒

⇐ {60, 00, 02, 00, 90, 00, LRC}

Case 2 command

{60, 00, 05, 00, CLA, INS, P1, P2, 00, LRC} ⇒

⇐ {60, Licc+2, 00, [Data (Licc)], 90, 00, LRC}

Case 3 command

{60, Lc+5, 00, CLA, INS, P1, P2, Lc, [data Lc], LRC} ⇒

⇐ {60, 00, 02, 00, 90, 00, LRC}

Case 4 command

{60, Lc+5+1, 00, CLA, INS, P1, P2, Lc, [data Lc], 00, LRC} ⇒

⇐ {60, Licc+2, 00, [data Licc], 90, 00, LRC}

Case 2 command using the 61 and 6C procedure byte

Le = Licc or Le ≥ Licc

{60, 00, 05, 00, CLA, INS,P1, P2, 00, LRC} ⇒

⇐ D1+D2+Dn+2, 00, [data D1+D2+Dn], 90, 00, LRC}

8.1 Extended cases

In T=0 protocol, the extended cases for APDUs are not supported on this mask.

In T=1 protocol, the use of the extended cases for APDUs is transparent from the host point of view as explained below as the TPDUs are identical to the APDUs.

Case 2 extended example :
APDU : CLA INS P1 P2 00 B2 B3 where B2 B3 is the length coded on 2 bytes (from 1 to 65535). With this mask 03 release, B2 B3 shall never exceed 498 bytes.

System to TDA8029 :	60 00 07 00 CLA INS P1 P2 00 B2 B3 LRC
TDA8029 to card :	NAD PCB 07 CLA INS P1 P2 00 B2 B3 EDC
Card to TDA8029 :	NAD PCB LEN1 D1 D2 …. Di EDC Where LEN1 is related to the negotiated data buffer size.
TDA8029 to card :	Rblock for acknowledge.
Card to TDA8029 :	NAD PCB LEN2 Di+1 D1+2 …. Dn SW1 SW2 EDC Where n = B2 B3

For this example it is supposed that only one chaining step is necessary.

TDA8029 to System :	60 B4 B5 00 D1 D2 …….. Dn SW1 SW2 LRC (B4 B5 =n+2)

References: ISO 7816-4 §5.3 and Annex B.

9 CONCLUSION

The following features give the general characteristics of the mask 03 :

- 3V and 5V cards supported
- E.M.V.3.1.1 validated but possibility to switch to full ISO 7816-3
- Optional Energy Saving Mode (hardware selection with P2.6, i.e. pin 25)
- Data buffer up to 506 bytes
- Asynchronous protocols (T=0 and T=1) supported
- Control and communication through a serial interface at different baud rates (from 9600 to 115200)
- Automatic hardware protections in the event of card take off, supply voltage drop short circuit or overheating
- All ISO7816-3 baud rates supported on the I/O line
- Possible selection of card clock frequencies
- Communication with the host made at the APDU level (asynchronous cards) or also possible at TPDU level for protocol T=1.
- Single +2.7V to +6.0V supply voltage.