TELKOMNIKA, Vol.11, No.1, January 2013, pp. 371~375
ISSN: 2302-4046

Implementation of Low-level Driver Interface for A CPU Card

Feng Luo*[a], Fengjian Hu[b]
Clean Energy Automotive Engineering Center, College of Automotive Engineering,
Tongji University, Shanghai 201804, China, +86-21-69583892
*corresponding author, e-mail: luo_feng@tongji.edu.cn[a], skyfisch@yahoo.cn[b]

Abstract

In order to develop an automotive digital tachograph, a Central Processing Unit Card (CPU Card) interface IC is designed and implemented. The purpose of this paper is to present a driver proposal on the basis of a CPU card interface IC NCN6804. Under the platform of a Freescale's microcontroller MPC5607B, the low-level driver interface is detailed both in hardware and software. The test result shows that this low-level driver proposal can meet the operation demands of CPU cards, and this method is effective, especially in automotive applications.

Keywords*: Digital Tachograph, CPU Card, low-level driver interface, NCN6804*

1. Introduction

With the rapid development of automotive industry and modern traffic management, for the purpose of lowering traffic accidents, more and more countries are forcing vehicle makers to install digital tachographs or similar devices in vehicles before allowing them to run on roads, especially in commercial vehicles. As shown in Figure 1, in these devices, CPU Cards can be found. With the cards, these devices can identify different operators of vehicles and record different activities of them.

CPU Card is a kind of IC card. What sets it apart is that there is a CPU of computer inside its Printed Circuit Board (PCB). Because of the CPU, there should be some relevant programs which are called Chip Operation System (COS) inside the package. So, a CPU Card should contain a ROM, a RAM and an EEPROM at least [1]. Due to the CPU inside, it has more powerful functions than general IC card, such as better store ability, safer and more flexible secret key management. It is nowadays commonly used in finance, social insurance, traffic police, government agencies, electronic signature, software encryption, etc [2].

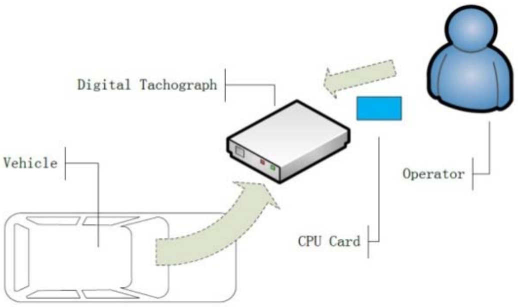


Figure 1. Implementation of CPU Cards

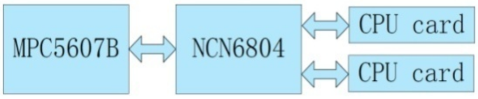


Figure 2. General structure of CPU Card module

Generally, CPU Cards can be divided into two different kinds, the contact cards and the contactless cards. The present work focuses on putting forward a low-level driver interface for contact cards which is fully compatible with ISO 7816 and can be used in digital tachographs or other similar devices.

In order to supervise cards, an ON Semiconductor IC NCN6804 is utilized. The NCN6804 is a dual smart card/ Secure Access Module (SAM) interface IC with 4-wire Serial

Received September 1, 2012; Revised November 28, 2012; Accepted December 10, 2012

Peripheral Interface (SPI) programming bus. It is fully compatible with ISO 7816-3, EMV and GIE-CB standards. With programmable/independent CRD_VCC supply for each smart card, it can handle 1.8V, 3.0V and 5.0V cards. Typically, it is used in Point Of Sales (POS), Automatic Teller Machine (ATM), Transaction Terminals, Set Top Box Decoder, Pay TV, etc.

In this system, a Freescale microcontroller MPC5607B is utilized as the core controller. With up to 64MHz execution speed, MPC5607B is a member of Freescale Qoriwa MPC56xx family of 32-bit microcontroller built on Power Architecture Technology. Having up to 1.5 MB on-chip code flash memory supported with the flash memory controller, and up to 96 KB on-chip SRAM, it is designed for current automotive body applications, which is suitable for the digital tachograph system and could be easy extended if more applications and more functions would be added to the system.

This paper is divided into four major sections. Section one explains the general background and introduces CPU Card, digital tachograph, NCN6804, and MPC5607B. Section two details the configuration of the hardware interface. Section three presents the operating procedures of CPU Card. And a conclusion about the application of this system is made in section four.

2. Configuration of Hardware

The circuit diagram of the interface in this project is shown in Figure 3. It can be divided into three different modules, the CPU Card module, the NCN6804 module and the MPC5607B module.

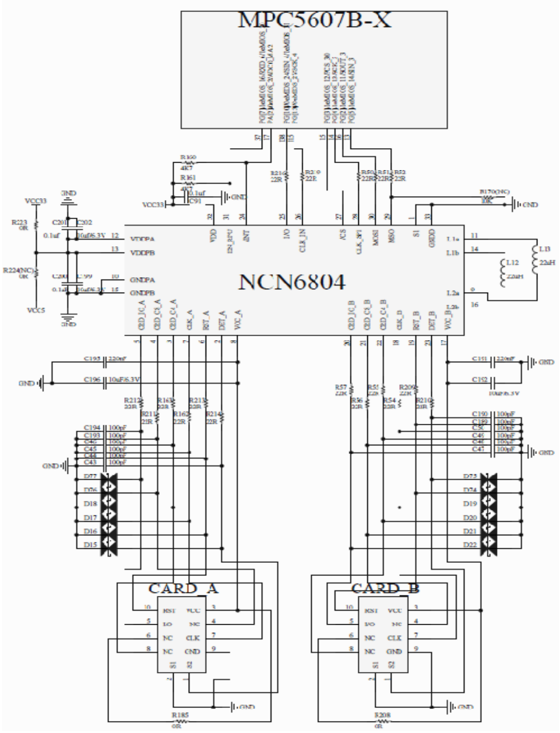
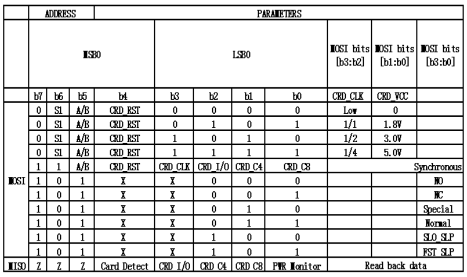


Figure 3. Circuit diagram of interface

Table 1. Electronic signals of CPU Card

Pins	Description
VCC	Power supplied to the IC inside (3.3V in this project)
GND	Common electrical ground (between CPU Card and NCN6804)
RST	Reset line to initiate reset sequence of instructions
VPP	Programming voltage input (not used in this project)
CLK	Clocking or timing signal provided to the IC inside the card
I/O	Input/Output for signal data to/from the IC inside the card (half-duplex)
C4	Reserved for future use
C8	Reserved for future use



	ADDRESS			PARAMETERS							
	MSB0				LSB0				MOSI bits [b3:b2]	MOSI bits [b1:b0]	MOSI bits [b3:b0]
	b7	b6	b5	b4	b3	b2	b1	b0	CRD_CLK	CRD_VCC	
MOSI	0	S1	A/B	CRD_RST	0	0	0	0	Low	0	
	0	S1	A/B	CRD_RST	0	1	0	1	1/1	1.8V	
	0	S1	A/B	CRD_RST	1	0	1	0	1/2	3.0V	
	0	S1	A/B	CRD_RST	1	1	1	1	1/4	5.0V	
	1	1	A/B	CRD_RST	CRD_CLK	CRD_I/O	CRD_C4	CRD_C8			Synchronous
	1	0	1	X	X	0	0	0			NO
	1	0	1	X	X	0	0	1			NC
	1	0	1	X	X	0	1	0			Special
	1	0	1	X	X	0	1	1			Normal
	1	0	1	X	X	1	0	0			SLO_SLP
	1	0	1	X	X	1	0	1			FST_SLP
MISO	Z	Z	Z	Card Detect	CRD I/O	CRD C4	CRD C8	PWR Monitor	Read back data		

Figure 4. MOSI and MISO bits definitions

Firstly, the two cards in this proposal are controlled in a multiplexed mode. And the electronic signals of the cards are described in Table 1.

Furthermore, as shown in Figure 4, the NCN6804 is configured in asynchronous mode via the instructions transferred through the SPI bus using CS, CLK_SPI, MOSI and MISO pins. Similarly, the RST and the VCC signals can also be provided by the SPI bus. In this mode, the CLK_IN clock input pin is used to provide clock signals to the selected card through CLK_A or CLK_B. The input/output (I/O) pin is used to transfer data signals from/to MPC5607B. When a card is inserted, the DET_A or DET_B is connected to GND and the logic level of INT pin is activated LOW. By reading the value of MISO/b4, the existence of the relevant card could be confirmed.

Besides, the MPC5607B, whose bus frequency is configured as 64MHz, is intended to operate as a Master Controller. The NCN6804 is intended to operate as a Slave Controller, and executes commands coming from the MPC5607B. The basic SPI timings are given in Figure 5 and Figure 6. In this project, the Deserial Serial Peripheral Interface (DSPI) 3 module of MPC5607B is used to communicate with NCN6804.

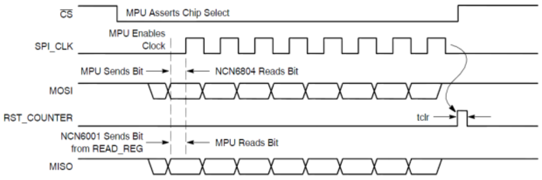


Figure 5. Basic SPI timings and protocol

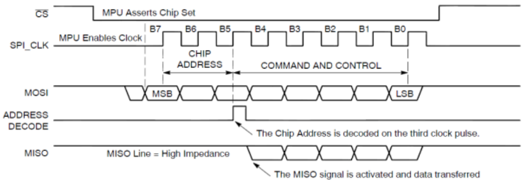


Figure 6. Chip address decoding protocol and MISO sequence

In order to generate the clock signals to the CLK_IN pin, the eMIOS module of MPC5607B is employed. It is worth mentioning that if the eMIOS module cannot provide a suitable frequency of clock signals due to other applications of this module, the DSPI clock signals are also qualified for this work.

Also, in this interface, the Wakeup Unit (WKPU) of MPC5607B is employed to respond to the requests of interrupt functions. The System Timer Module (STM) of MPC5607B is applied to support software timing during the operating procedure of cards.

3. Operating Procedure of CPU Card

The general operating procedure for CPU Card can be divided into five steps as shown in Figure 7 [3-6]:

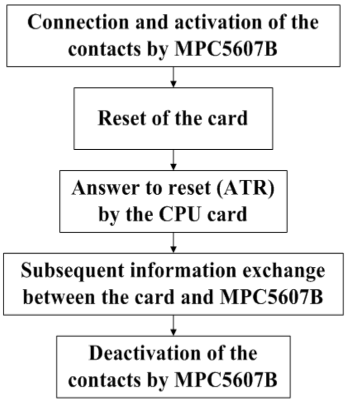

Connection and activation of the contacts by MPC5607B
Reset of the card
Answer to reset (ATR) by the CPU card
Subsequent information exchange between the card and MPC5607B
Deactivation of the contacts by MPC5607B

Figure 7. Operating procedure of CPU Card

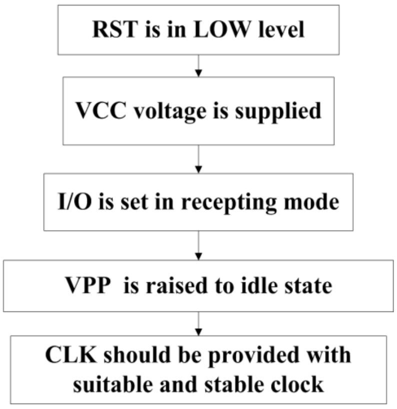

RST is in LOW level
VCC voltage is supplied
I/O is set in recepting mode
VPP is raised to idle state
CLK should be provided with suitable and stable clock

Figure 8. Activation procedure of CPU Card

In Figure 8, the activation procedure of CPU Card is described. After activation, if one card is inserted into the card slot, an interrupt will be generated by the INT pin. In order to know the source of the interrupt, the MPC5607B can send commands to NCN6804 to confirm whether the corresponding card is inserted or not. Figure 9 lists all the interrupt functions of NCN6804.

When the card A or card B exists, as shown in Figure 10, the power up CRD_VCC sequence makes sure all the signals related to the card are LOW during the CRD_VCCA/B rising up. Then enable the clock signals. After that, raise the CRD_RST signal. In this way, the card would be reset.

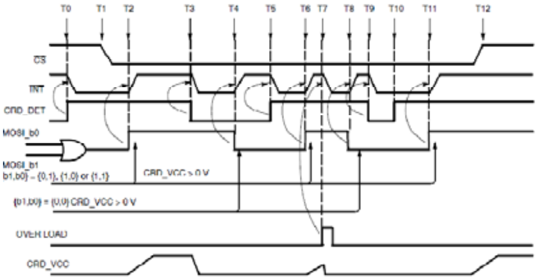


Figure 9. Basic interrupt functions

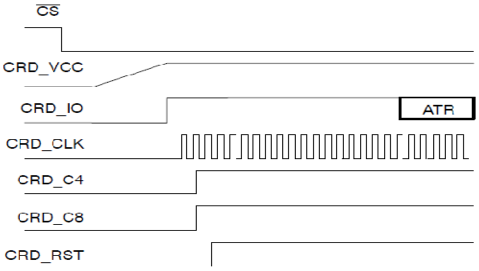


Figure 10. Power up CRD_VCC sequence

After receiving the reset trigger signals, the IC inside the card would respond with a series of signals which contain initial character, format character, interface characters, historical characters, and check character. These characters are called Answer To Reset (ATR). The initial character (TS) implies the forward or reverse protocol of ATR. The format character (T0) encodes in its 4 high-bits the existence of 4 other interface characters, and encodes in its 4 low-order bits the number of historical characters. The interface characters (TAi, TBi, TCi, TDi) encode the parameters and types of communication protocols that the card proposes to use. The historical characters (T1, T2. TK) carry the information about the card makers, card type, version number, the state of the card, etc. The check character (TCK) adds the XOR value from character T0 to the character before TCK to the ATR.

The ISO 7816-3 defines the transmission protocol which is based on an asynchronous half-duplex mode of operation for the interchange of data between the cards and the interface device. This means that the I/O pin must change the transmission direction depending on whether the card or interface device is transmitting data. In brief, the T=0 protocol defines the transmission of bytes while the T=1 protocol contains a block mode of operations.

Usually, the initial Elementary Time Unit (ETU) is 372/Fi. In this project, the Fi is 4MHz, so the initial ETU is 93 μs. If the work ETU need to be changed, for example, changed to 4 μs, according to the formula Work ETU = F/ (D*Fs), the value of F in Table 2 should be 512, and the value of D in Table 3 should be 32. Accordingly, the MPC5607B should send a Protocol Type Selection (PTS) request to the card using the initial ETU 93 μs, and if the baud rate of the card is negotiable and the card has successfully accepted the request, the MPC5607B will receive a PTS response using the new Work ETU value of 4 μs. The content of the PTS response is supposed to be the same as the PTS request's. T=1 protocol is used in this project. So if the request is 0xFF, 0x11, 0x96, 0x78, the response is supposed to be 0xFF, 0x11, 0x96, 0x78, too. After negotiation the new baud rate should remain valid till the card is reset.

Table 3. Bit rate adjustment factor

DI	0000	0001	0010	0011	0100	0101	0110	0111
D	RFU	1	2	4	8	16	32	RFU
DI	1000	1001	1010	1011	1100	1101	1110	1111
D	12	20	1/2	1/4	1/8	1/16	1/32	1/64

Table 2. Clock rate conversion factor

FI	0000	0001	0010	0011	0100	0101	0110	0111
F	Internal clock	372	558	744	1116	1488	1860	RFU
Fs（max）MHz	N/A	5	6	8	12	16	20	N/A
FI	1000	1001	1010	1011	1100	1101	1110	1111
F	RFU	512	768	1024	1536	2048	RFU	RFU
Fs （max）MHz	N/A	5	7.5	10	15	20	N/A	N/A

All the subsequent information exchanged between the card and the MPC5607B is timed by STM module of MPC5607B. The character frame is shown in Figure 11. When the

MPC5607B transmits data to cards, in order to accurately define the bit time, some codes should be programmed with assembly language. As shown in Figure 12, while receiving frames from card, the MPC5607B should sample the middle point of every bit, a quite near point before it and a quite near point after it. If the three points are measured to be at the same level, the relevant bit is regarded as this level. If not, the data exchange procedure should be retried.

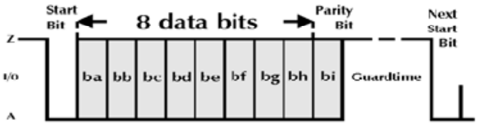


Figure 11. Character frame

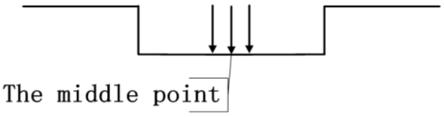


Figure 12. Sampling point of every bit

By default, the Interface Device (IFD) can receive 32 character frames. If more than 32 character frames need to be received by IFD, the IFD should send an IFS request to the card to indicate a new IFSD it can support. For example, if the request is 0x00, 0xC1, 0x01, 0xFE, 0x3E, the response should be 0x00, 0xE1, 0xFE, 0x1F if the request is successfully accepted.

When the information exchanged between the card and the MPC5607B is terminated, the card should be deactivated. The procedure of deactivation is described in Figure 13.

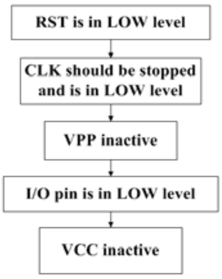


Figure 13. Deactivation procedure of CPU Card

4. Conclusions

In this paper, an implementation of a low-level driver interface for CPU Card was presented. This proposal could be applied in digital tachographs or similar devices. Specifically, the physical level and the data link level interfaces were described from hardware to software. It turns out to be that this driver interface was effective after many practical experiments.

References

[1] International Organization for Standardization/ International Electrotechnical Commission. ISO/IEC 7816-3. *Identification cards –Integrated circuit cards –Part 3: Cards with contacts – Electrical interface and transmission protocols* . Switzerland. International Organization for Standardization. 2006.

[2] Aiying WANG. Smart Card Technology—IC Card. Beijing: Tsinghua University Press. 2000: 47-66.

[3] Congwen ZHANG, Ming LING. Design of IC card Interface on the basis of SEP4020. *Computer Knowledge and Technology*. 2009; 5(16): 4337-4339.

[4] Jinghe ZHANG, Jinpei WU, Qishan ZHANG. Analysis and Implementation of T=0 Communication Protocol of CPU Card. *Application of Electronic Technique*. 2002; 28(9).

[5] Zhenye YANG. Technology and Application of IC Card. Beijing: Science Press. 2006: 126-143.

[6] Feng CHEN, Han YIN. The Interface Characteristics, Communication Protocol and Read-Write Programs Design of CPU Card. *Microcontroller & Embedded System*. 2002; (1).