



The Design of RS232 and CAN Protocol Converter

Based on PIC MCU

Xianjun Wang & Wencheng Guo

School of Computer Technology and Automation, Tianjin Polytechnic University

63 Chenglin Road, Hedong District, Tianjin 300160, China

E-mail: wangxianjun1985@163.com

Abstract

CAN bus will be increasingly used in wide range of applications for its superiority, but it couldn't communicate with computer directly. The article describes the design of RS232 and CAN bus protocol converter depending on PIC Microcontroller, which solves the problem that CAN networks can not directly communicate with PC. At present, single-chip microcomputer of 51 series with CAN controller SJA1000 are widely used in domestic. Considering cost and converter size, the paper will talk about PIC18F2580 with integrated CAN microcontroller designed for RS232 and CAN protocol converters to facilitate the direct communication between computers and CAN bus.

Keywords: RS232, CAN, PIC MCU

1. Introduction

CAN bus are being used more and more widely in the areas of automotive, machinery, CNC machine tools, medical devices, smart sensors for its high-performance, high reliability and flexible design features. However, data on CAN bus could not directly communicate with computer.

At present there are more and more MCU with internal CAN controller integration. In this paper, CAN bus and RS232 protocol converter is designed based on Microchip's PIC18F2580 MCU, so that computer can read data on CAN bus directly and also send control commands to CAN bus through the RS232 interface.

2. The introduction of CAN Bus

CAN (Controller Area Network) is one of the most widely used serial communication bus in the world. CAN was first used in the automobile industry at the end of 1980s, which is proposed by the German Bosch company. CAN-bus specification has been developed as international standards the "ISO11898" (high-speed applications) and "ISO11519" (low-speed applications) by ISO. CAN bus optimizes the seven layers of communications architecture of Open System Interconnection model (ISO / OSI model) provided by ISO. And only the physical layer, data link layer and application layer can guarantee the data transmission speed and effectiveness. The basic design specification of CAN protocol not only demands a high bit rate, anti-electromagnetic interference, but also detects any possible errors.

CAN bus have the following main features: (1) Good real-time. CAN bus, which uses CSMA / CD media access method, is a multi-master field bus. This approach is similar with Peer to Peer approach of large-scale network, thus coincides with the open structure. Regardless of master-slave, information can be sent to free bus by any node at any time, which is a flexible means of communication, and this feature can also be used to constitute a (fault-tolerant) multi-machine backup system easily. Each node of the CAN receive all the information in the bus. Packet filtering and shielding mechanism enable each node quickly decide whether current packet enter into the receive buffer or not so that to reduce the information processing time which has nothing to do with the node. (2) Fault isolation is good. CAN uses non-destructive arbitration, when two or more nodes simultaneously send data to the network, the identifier of data frame determines the priority of the data frame, that small identifier means high-priority, large identifier equals to low priority. Nodes which send high-priority information continue to send data, and nodes which send low-priority information take the initiative to exit the bus. For a serious fault node, it will automatically turn off the bus function, which does not affect the work of other nodes but avoids network conflicts or a decrease of arbitration time. In the case of multi-transmission, lost information can be made up for through data fusion technology, which is caused by transmission or equipment failures (3) Long distance communication. The largest rate of CAN communication can be achieved 1Mbps/40m, the maximum transmission distance can reach 10Km/50Kbps. (4) Good anti-mistake transmission design. CAN with short-frame structure (8 effective bytes for each frame) allows short data transmission

time, low probability of interference and short re-sending time. And each frame of information has CRC error checking and other measures to ensure the low error rate of data transmission. (5) Theoretically CAN bus can connect 2000 nodes. In fact less than 110 is appropriate; it can satisfy the needs of majority users. CAN bus can transmit and receive data in several ways such as point-to-point, point-to-multipoint and overall broadcast. CAN bus transmission uses twisted pair and has no special requirement for the transmission medium.

3. PIC18F2580 introduction and hardware design

3.1 PIC18F2580 CAN module introduction

PIC MCU has advanced RISC-like (Reduced Instruction Set Computer) structure which is reflected in each of efficient and powerful command. Among the command systems, all commands are single-cycle, single-word commands except procedures branch command which is two-cycle, single-word command. There is no cross-functional phase of instructions among these commands, so that all commands are with simplicity thus improve the efficiency of the software coding and reduce the required procedure storage unit, allowing the system with the highest efficiency and outstanding performance, which sets a new performance standard for 8-bit microcontroller market. Advantages of PIC microcontroller are also reflected in following ways: the function of each command is powerful, with high implementation efficiency. Commands digit for Low-grade, mid-grade, high-grade series are 12, 14, 16 respectively, and general command digit are only 33, 35, 58 while upward compatible. However, the CISC (Complex Instruction Set Computer) architecture microcontrollers usually have up to 50-110 multi-byte and multi-cycle commands. Therefore, PIC microcontroller series are easier to study, whose main features are as follows: (1) line structures of command can complete the implementation of a directive and the Fetch operation of next directive in a cycle simultaneously, which improves the efficiency of each internal clock cycle. (2) High-speed instruction execution time achieving 200ns at 20MHz clock or 160ns at 25MHz clock. It can carry out any operation of I / O port bit in a single cycle. Only operations on branch procedure CALL, GOTO and PC do require two-cycle execution time. (3) The system uses byte wide instructions, so that procedure volume is significantly reduced in the same circumstances, and the formation of the code is generally half of the microcontroller.

High-end product PIC18F2580 Single chip is introduced by Microchip of the United States, which applies 16 byte RISC instruction set with short instruction cycle, strong processing ability and high computing power. Data communication can be completed and communication protocols requirements can be satisfied without expanded memory. PIC18F2580 chip integrated single-chip A / D converter, the internal EEPROM memory, output compare, input capture, PWM output, I2C and SPI interfaces, asynchronous serial communication (USART) interface circuit, CAN Bus interface circuit, FLASH memory read / write and etc. Besides, it has powerful chips, strong I / O port drive capability, and can be connected directly with the LCD interface and the design of the circuit is simple and reliable. Addressable Universal Synchronous / Asynchronous Receiver Transmitter (USART) module and the CAN control module can easily achieve data communication in industrial field which has a good prospect.

Its main features include: (1) 1536 bytes RAM; (2) 32Kb FLASH, among which 1Kb is dedicated to USB buffer; (3) 256 bytes EEPROM data memory; (4) supporting RS232, RS485 and EUSART module of LIN serial interface; (5) supporting I2C and master synchronous serial port of SPI communications; (6) 10-bit A / D converter, precision up to ± 1 LSB, which is equipped with up to 13 input channels; (7) two analog comparators; (8) with 16-bit data capture and the resolution capture / compare / PWM module; (9) enhanced capture / compare / PWM module, there is dead-zone control and fault protection input; (10) 4 timers (three 16 bytes timers and one 8 bytes timers); (11) programmable under-voltage reset and low voltage detection circuit; (12) enhanced online debugging features, add up to three hardware breakpoints.

3.2 PIC18F2580 CAN module introduction

The Controller Area Network (CAN) module is a serial interface which is useful for communicating with other peripherals or microcontroller devices. This interface, or protocol, was designed to allow communications within noisy environments.

The CAN module in 18F2580 is a communication controller, which is the topic involved in this paper. It implementing the CAN 2.0A or B protocol as defined in the BOSCH specification. The module will support CAN 1.2, CAN 2.0A, CAN 2.0B Passive and CAN 2.0B Active versions of the protocol. The module features are as follows: (1) consistent with the ISO model; (2) Implementation of the CAN protocol CAN 1.2, CAN 2.0A and CAN 2.0B; (3) The CAN module supports the following frame types: Standard Data Frame; Extended Data Frame; Remote Frame; Error Frame; Overload Frame and so on (4) with two priority receive buffer and three priority send buffer; (5) 6 to accept filter: 2 high priority receive buffers, and the remaining 4 low priority receive buffers; (6) with 2 shielding filters, corresponding to two different receiver buffer; (7) with six kinds of operating mode settings: Request Configuration mode; Request Listen Only mode; Request Loop mode; Request Disable mode; Request Normal mode; (8) supporting short-frame structure, Standard and extended data frames, 0-8 bytes data length; (9) Programmable wake-up functionality with

integrated low-pass filter; (10) Signaling via interrupt capabilities for all CAN receiver and transmitter error states; (11) Programmable Loop mode supports self-test operation; (12) Programmable clock source; (13) Programmable link to timer module for time-stamping and network synchronization; (14) Low-power Sleep mode.

The CAN bus module consists of a protocol engine and message buffering and control. The CAN protocol engine automatically handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against filters to see if it should be received and stored in one of the two receive registers. The CAN module uses the RB2/CANTX and RB3/CANRX pins to interface with the CAN bus. In normal mode, the CAN module automatically overrides TRISB<2>. The user must ensure that TRISB<3> is set.

CAN module of PIC18F2580 is configured with three send buffers. If needed, TXBnCON (n is 0, 1, 2) TXPR1, TXPR0 bit can be modified to set 4 different priorities. If the two send buffers have the same priority, then the send buffer with a bigger number has the higher priority.

Messages will be sent in order according to the level of priority. You can start sending messages by sending data to TXBnDm (n is 0, 1, 2, m is 0-8) and then set TXREG bit of TXBnCON. You can determine whether the messages are sent successfully through TXB0IF bit of PIR3.

CAN module of PIC18F2580 is configured of the 2 receive buffers, 6 receive filters and 2 receiver shielding filter. RXB0 corresponds to the receiving filter RXF0, RXF1; shielding filters RXM0, RXB1 correspond to the receiving filter RXF2, RXF3, RXF4, RXF5, shielding filters RXM1. In addition to these two receive buffers; PIC18F2580 is also equipped with a MAB (Memory Allocation Block) which receive all messages from the bus. When the node detects the messages on bus, the messages will be transmitted to MAB and will take its own arbitration and receive filters for comparison. If met, messages will be transmitted to the corresponding receive buffer. Shielding filter will decide which receive filter bit is effective.

3.3 Universal Synchronous Asynchronous Receiver Transmitter (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the three serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs and so on.

3.4 Circuit hardware design

The system mainly includes two parts, the CAN bus driver circuit and the RS232 driver circuit. CAN bus transceiver chip and MCU are isolated by high-speed opt coupler 6N137 to avoid the affecting of CAN bus fluctuations on the normal operation of MCU and to improve the system anti-interference ability. Power and Ground wire on both sides of 6N137 must be isolated too. PCA82C250, which is interface between CAN bus protocol controller and the physical bus, is used as a CAN bus transceiver chip. It provides different send performances for bus and different reception performances for the CAN bus controller and it is fully compatible with the ISO11898 standard. The purpose of the use of PCA82C250 are increasing communication distance, improving the transient system anti-interference ability, protecting the bus, reducing radio frequency interference (RFI) and implementing thermal protection. It should be noted: resistor RS between PCA82C250 No. 8 pin and floor is called slope of resistance and its value determines whether the system is in high-speed working condition or the slope control mode. No. 8 pin should be connected directly with the floor so that the system will be in a high-speed working condition. In this manner, in order to avoid radio frequency interference, it is recommended to use shielded cable for the bus. While at a lower baud rate and short bus, slope control mode will be generally used and slope of the increase and decrease depends on the RS resistance value. CAN bus interface part is as shown in Figure 1.

RS232 has been widely applied as a standard computer serial communication interface. However, the transmission distance is short with a maximum 15M. Transfer rate is also relatively low with a maximum of 20Kb/S. Imbalance are used to transmit data, that is, single-ended communication. RS232 protocol uses negative logic instead of TTL-level interface standard. Negative logic: logic "1": -3V ~ -15V, logic "0": +3 V ~ +15 V. MAX232 chip can convert 5V TTL level of single-chip to $\pm 15V$ level which is needed by RS232. The system required only a single +5 V power. The serial ports TX and RX of PIC18F2580 are connected with MAX232 which can carry out full-duplex asynchronous communication with other RS232 interfaces. RS232 interface part is as shown in Figure 2.

4. Software design

In order to improve the operating efficiency of firmware, the main program initializes the system, interrupts it then and makes some analysis and necessary treatment in interrupt service routine and set the corresponding variable signs and

data buffer. The main program queries the variable signs circularly and let appropriate subroutine for processing. This program structure allows the main program be able to deal with various data transfer tasks in foreground, and at the same time can treat bus cases in the background timely through disruption. Initialization of main program includes initializations of I / O port, timers, serial port, CAN controller, interrupt and user signs data. This process gives configuration definition to basic resource of PIC18F2580 and configures reused I/O resource CAN interface and serial port. After setting up a 450ms timer interrupt, defining the serial port and operating parameters of CAN interface, opening the corresponding interrupt source, putting the user signs data the initial value, initialization process ends. MCU enters into the variables circulatory state.

CAN initialization mainly includes configuration of the CAN module, setting send mail, receiving mail identifier and initialization data, setting baud rate and CAN work mode, initializing receive filter and receiving shielding and CAN communication module initialization procedure:

Void CAN init (void)

```
{TRISB = 0x08;          // Set CAN input and output
CANCON = 0X80;        // request to enter CAN configuration mode REQOP = 100
While (CANSTAT & 0X80 == 0) {;} //Wait to enter CAN configuration mode OPMODE = 100
    BRGCON1 = 0X01; // set SJW and BRP, SJW = 1TQ, BRP = 01H
BRGCON2 = 0X90; // set Phase_Seg1 = 3TQ and Prog_Seg = 1TQ
BRGCON3 = 0X42; // set Phase_Seg2 = 3TQ // Set send mail 0 symbols and data
TXB0CON = 0X03; // Sending priority is the highest priority, TXPRI = 11
TXB0SIDH = 0XFF; // Set 0 Standard Identifier for Send Buffer, the procedures use Standard Identifier
TXB0SIDL = 0XE0;
TXB0DLC = 0X08; // Set the data length as 8 bytes
// Set receive mail 0 identifier and initialization data
RXB0SIDH = 0XFF; // set the receive buffer identifier 0
RXB0SIDL = 0XE0;
RXB0CON = 0X20; // just to receive valid information of standard identifier, FILHIT0 = 0 means RXB0 uses filter0
RXB0DLC = 0X08; // set the length of data area for receive buffer 0
RXB0D0 = 0X00; // initialize data from receive buffer 0 data zone
RXB0D1 = 0X00;
RXB0D2 = 0X00;
RXB0D3 = 0X00;
RXB0D4 = 0X00;
RXB0D5 = 0X00;
RXB0D6 = 0X00;
RXB0D7 = 0X00;
// Initialize the receiving filter 0 and receive shielding,
RXF0SIDH = 0XFF;
RXF0SIDL = 0XE0;
RXM0SIDH = 0X00;
RXM0SIDL = 0X00;
// Initialize I / O control register of CAN Module
CIOCON = 0X00;
CANCON = 0X00; // enable CAN enter normal operating mode
While (CANSTAT & 0XE0! = 0) {;}
// Initialize CAN interruption
PIR3 = 0X00; // clear all Interrupt Flag
```

```

PIE3 = 0X01; // interrupt receivers which can receive buffer 0
IPR3 = 0X01; // receiving interruption for receive buffer 0 is at highest priority
}
USART module initialization procedure:
Void USART init (void)
(TRISC = 0x80; // set the serial input and output
SPBRG = 0X4d; // select the baud rate for transmission 9600bps
TXSTA = 0X04; // select asynchronous high-speed 8-bit data transfer
RCSTA = 0X80; // permit synchronous serial port work
TXSTAbits.TXEN = 1; // send permit
RCSTAbits.CREN = 1; // accept the data permit
PIE1bits.RCIE = 1; // Receive Interruption Enable
IPR1bits.RCIP = 0; // receive interruption is at low priority
}

```

After the completion of all I/O Configuration and CAN initialization subroutine, the main program waits for the connection between MCU and PC. After communications between PC, procedures are ready for data exchange: If the system receive valid CAN data, then the data is sent to the host computer; if the system receive the effective data and then will send it to the CAN bus. The main program flow chart is as shown in figure 3.

5. Results of experiment

This system use serial port of personal computer to receive and transfer data on RS232, and connected to the CAN network through the PIC MCU and the CAN communication module in the MCU. Converters can be two-way communication: on the one hand it can receive data from the CAN bus and transfer it to the computer in standard RS232 format; On the other hand it can be put the data flow that received in RS-232 format into the data flow that compliance with CAN protocol and sent it to the CAN bus. Give full play to the CAN bus communication long distance, node capacity, high reliability, as well as through the host computer to monitor the operation of bus and can be directly sent instructions to CAN bus. PIC18F2580 chip in SOIC package also has greatly narrowed the converter size which makes the industrial applications convenient.

References

- Feng Wu&Xinjian Chen. (2005). *PIC Single Chip C language Entry*. Beijing: Beihang University Press.
- Guoxian Chen. (2004). *PIC Single-chip Microcomputer Principle and Interface Technology*. Electronics Industry Press.
- Microchip Technology Inc. *PIC18F2x8 data sheet*. [Online] Available:www.microchip.com
- Xianhui Yang. (2000). *Field Bus Technology and Applications*. Beijing: Tsinghua University Press.
- Yu Wang. (2006). *PIC Single Chip Entry and Improve*. Beijing: Machinery Industry Press.
- Yuntao Rao, Jijun Zhou&Yongyun Zheng. (2003). *CAN Bus Principle and Application Technology*. Beijing: Beihang University Press.
- Zexiang Liu. (2005). *Field Bus Technology*. Beijing: Machinery Industry Press.

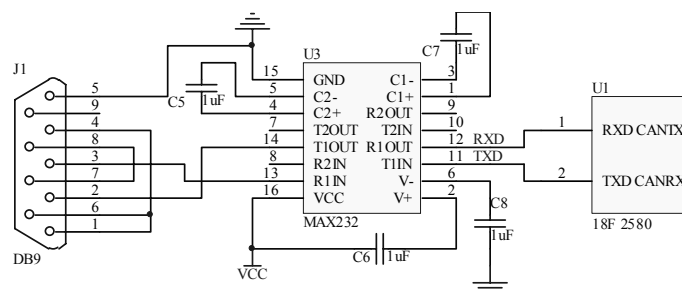


Figure 1. CAN Bus Interface Circuit

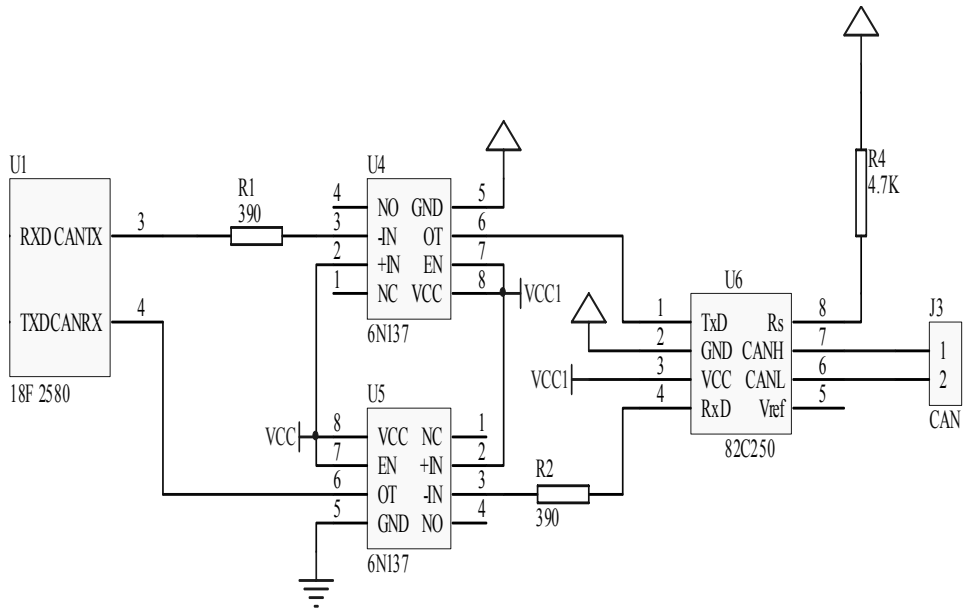


Figure 2. RS232 interface circuit

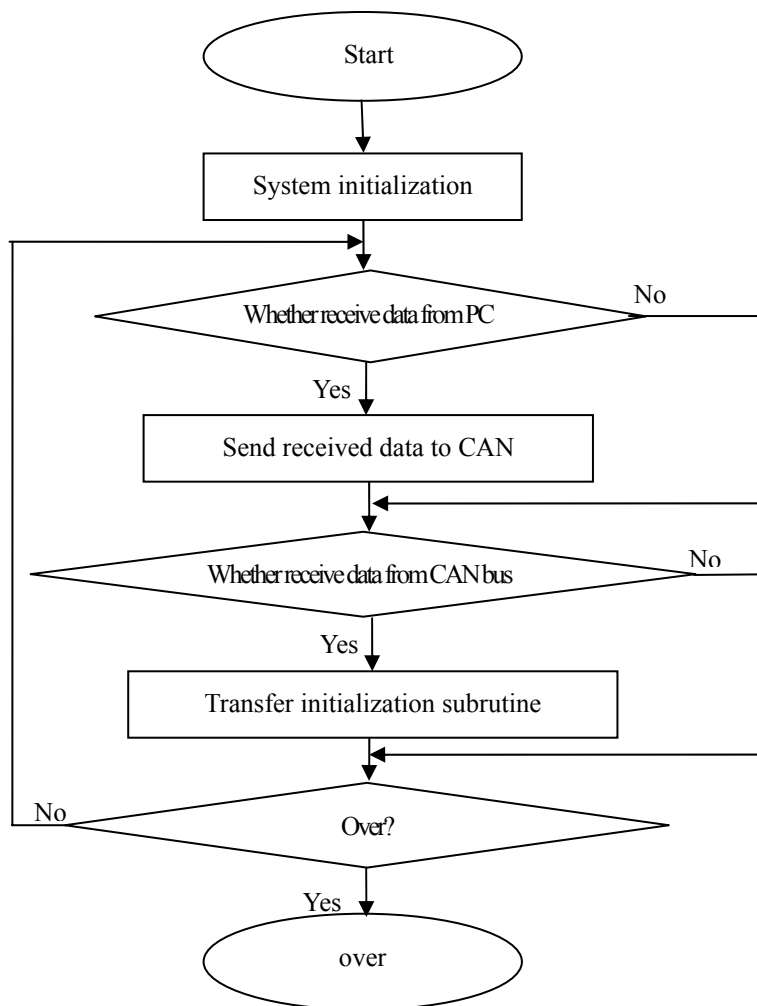


Figure 3. main program flow char