

Research and Design on CAN Network Analyser based on USB Interface

Liping Yuan

School of Information Engineering Wuhan University of Technology Huaxia College, Wuhan, 430000, China

Keywords: CAN Bus; USB Interface; STM32F105RBT6; Driver Development; CANoe.

Abstract. This paper designed a CAN network analyzer based on USB bus technology, whose function including network analysis, protocol parsing and node simulation. The CAN network analyzer based on USB interface in this article is mainly composed of two parts - USB-CAN adapter and host computer interface. This paper designed the hard ware and firm ware of USB-CAN Adapter using STM32F105RBT6 as core controller. Then, this paper designed Windows driver with Driver Studio, and host program with Visual C++ 6.0. At last, visual pure electric vehicle simple model was designed, and its function was tested. Verified by test, the design of CAN network analyzer supports USB2.0 interface at full speed. It can convert between CAN2.0A and CAN2.0B protocol and USB protocol, and support data frame, remote frame type of CAN message bidirectional transmission. Its baud rate is available to support the CAN network work at optional 5 KBPS ~ 1 MBPS. It have those function such as all accept and blocking the filter, transceiver test, data saving.

Introduction

The system consists of USB-CAN adapter and PC. USB-CAN adapter connects a PC via a USB cable USB2.0 interface, while the other end of adapter connects to the CAN bus via twisted pair or coaxial cable. Adapter consists of five parts like USB interface circuit, USB controller, data conversion processing module, CAN controller, CAN interface-level protection circuit. PC needs to provide USB interface driver and host program. System enables transfer of data from the CAN bus to the PC host computer, display, protocol analysis and other tasks.

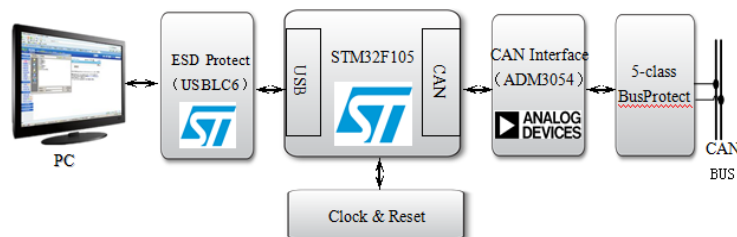


Fig.1: Block diagram of the network analyzer system

USB-CAN Adapter Hardware Design

Core controller circuit

2-bit microcontroller STM32F105, which is based on ARM Cortex-M3 core and its maximum frequency could be 72MHz^[3].

The starting mode of STM32 is set by BOOT0 and BOOT1. System clock oscillator circuit is Pills and crystal clocked is 25MHz. System reset circuit includes a power-on reset and manual reset function.

USB interface circuit

USB interface circuit is shown in Figure 2. The second function of pin 44 (PA11) of STM32F105RBT6 is USB_DM, and the second function of pin 45 (PA12) is USB_DM. In addition, pin 11 (PC3) is distributed as normal IO port used to enable USB_DP via pull-up resistor.

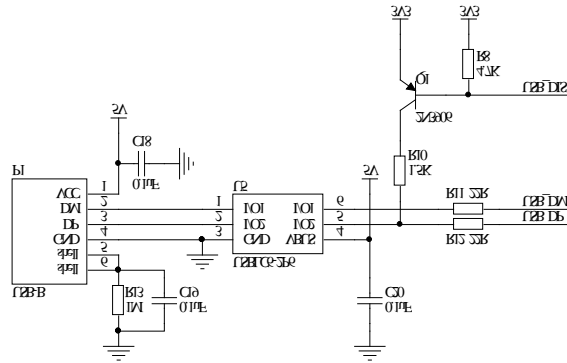


Fig.2: USB Interface Circuit

When USB_DIS pin is low, Q1 turns on and the collector voltage of Q1 is close to 3.3V, then DP bus is pulled. This signal is detected when a USB device is plugged. When USB_DIS pin is high or floating, DP is not a pull.

This design uses a U5 (USBL6) as the USB interface ESD protection chip. The chip consists of two functions, keeping data balance on signal lines (DP and DM) and supplying ESD protection on power line [4].

CAN interface circuit

CAN interface circuit is shown but Figure 3. There are two groups of CAN interface, respectively CAN1 and CAN2. CAN functionality pin is defined as: pin61 (PB8) is CAN1_RX and pin 62 (PB9) is CAN1_TX.

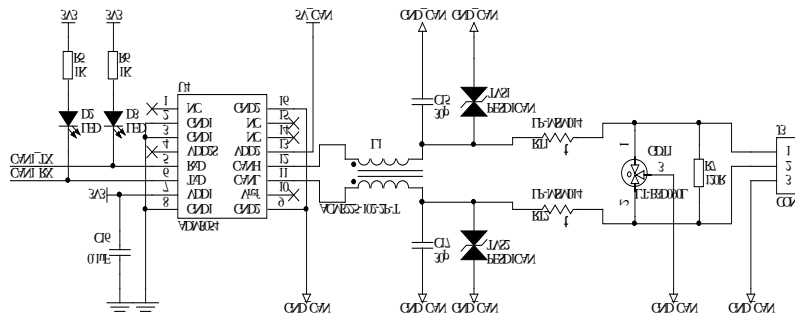


Fig.3: USB Interface Circuit

In order to avoid the occurrence mentioned above, two different measures usually are taken. The first one is electrical isolation and the other one is overvoltage and overcurrent protection on the bus. This system has completed a strict precision bus protection design according to the two ways [4].

The electrical isolation measures taken in this article is using ADI's effective voltage 5kV isolated CAN transceiver device U4 (ADM3054). Taking advantages of surge suppression characteristics of several of devices, this system designs five-class protective structures to prevent interference to damage the transceiver. Gas discharge (lightning tube) GDT1 (LT-B3D090L) is the first one protection to deal with the lightning by discharging its surge of energy. The secondary protection is realized by visitor RT1 (LP-MSM014) and RT2 (LP-MSM014), in μ s-class response time. Visitors can be replaced by resettable fuse SMD014 or metal oxide visitor MOV devices. The third one protection using TVS tube TVS1 (PESD1CAN) and TVS2 (PESD1CAN) can clamp the bus at 6.8V or less. The response time of TVS is PS level. The fourth one protection, as shown in the figure 3, including two 30 pF capacitor C15 and C17, is used to filter out high-frequency electromagnetic radiation and interference on the bus. The last one protection using common mode inductor L1 (ACM3225-102-2P-T) plays a balancing role on the bus.

Power circuit

In this paper, system supplied by the USB interface on PC. In order to get isolated power, DCP01B0505DBP is used on the board and LM1117 circuit is designed to supply 3.3V.

USB-CAN Adapter Firmware Design

Firmware refers to a program running on STM32F105RBT6 inside. The main part of the program includes CAN drivers, USB drivers and real-time data processing algorithms. The fire ware is developed on Real View MDK 4.20 platform.

In this paper, the version of standard interface function fire ware library is STM32F10x_StdPeriph_Lib_V3.5.0. The USB interface function library version is STM32_USB-FS-Device_Lib_V3.2.1. The Overall program flow is shown in figure 5. There are two interrupt function, CAN receive interrupt and USB interrupt, besides the main function in the fire ware.

What is more, in the CAN driver module design, the CAN controller initialization steps are: mapping port; enabling relevant clock; setting the baud rate; setting CAN interrupt; setting filters. CAN messages are sent by calling the function CAN_Transmit (CAN_TypeDef * CANx, CanTxMsg * TxMessage), and CAN messages are received by calling the function CAN_Receive (CAN_TypeDef * CANx, uint8_t FIFONumber, CanRxMsg * RxMessage).

In the USB driver module design, by calling the USB_Init (void) function in usb_int.c, the system complete the initialization of the USB device. USB interface descriptors, including device descriptor, interface descriptor, configuration descriptor, string descriptor, endpoint descriptor, device defined descriptor and other descriptors, are given in us b desk's file as an array. USB transactions are processed in the form of interruption, and USB interrupt handler is the OTG_FS_IRQ Handler function in stm32f10x_it.c, besides STM32_PCD_OTG_ISR_Handler function is called in this function.

In order to deal with a large number of high-speed data streams, prevent data channel congestion, and protect the stability and security of data, a buffer of FIFO space for 120 CAN messages is designed. A special algorithm is carried out for the reading and writing of FIFO.

CAN Network Analyzer PC End Software Design

Power circuit

The mission of driver development is using Driver Wizard to establish Driver Studio driver framework, and then build custom requests under Visual C ++ 6.0 environment.

The bus type which should be selected in Driver Wizard is USB. Input VID=03EB and PID=0205, and add 3 points like Ep1Out, Ep2In and Ep3Out for USB devices. The IO request interface parameters address is 0x801, 0x802 and 0x803. The parameter list is shown as figure 4 after driver development finished.

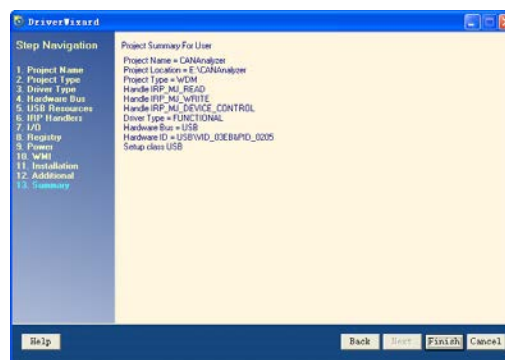


Fig.4: Parameter List

A project is automatically generated by Driver Wizard, and its project tree shown in figure 5.

Some examples such as Driver Entry, Add Device and Unload are included in CAN Analyzer Driver. Cops. Knapp Device, De fault Pap, On Start Device (device start up), On Stop Device (device stop), On Remove Device (device removal) are included in CAN Analyzer Device^[5].

EP1_WRITE_Handler, EP2_READ_Handler, EP3_WRITE_Handler are processed in Device Control. The ports of the three interface function are set mainly calling Build Bulk Transfer function.

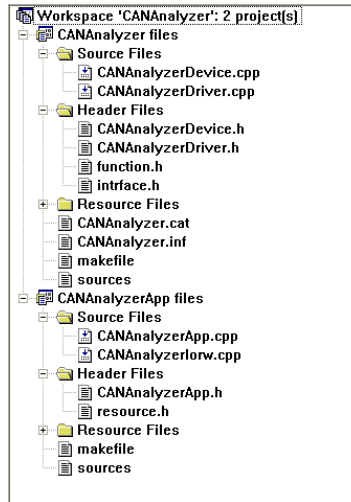


Fig.5: Project Tree

Host program design

This host program is developed by Microsoft Visual C ++ 6.0. The process could be described like: design user interface and program application according to different objects[6]. The interface of USB-CAN mainly includes USB device operation, CAN parameter setting, CAN messages sending, messages display, data saving and so on. The interface on PC is shown as figure 6.

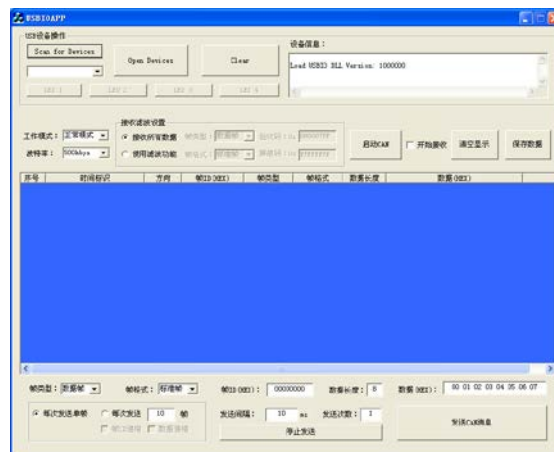


Fig.6: Main Interface on PC

Open the devices by CreateFile function and close the devices by CloseFile function. Some customize operations like data reading and writing could be realized via DeviceIoControl function.

CAN Network Built Using CANoe Test Platform

CAN network planning

There are vehicle controller, motor controller, battery management system and instrument controller as ECU nodes in CAN network. The speed of whole net is 500 Kbps. CAN network sharing information list is shown in Table 1.

Table 1 CAN Network Sharing Information List

ECU Notes	Variables	CAN sending		CAN receiving	
		Packets	Signal	Packets	Signal
Vehicle Ctrl ECU	Env Driver Switch	Driver Action	Start Key Cmd		
	Env Driver Acc	(0x101)	Acc Power		
MotorCtrlECU		Motor State	Motor Run	Driver Action	Start Key Cmd
		(0x102)	Motor Speed	(0x101)	Acc Power
BatteryMagECU		BmsState	Bms Voltage	MotorState	Motor Run
		(0x103)	Bms Soc	(0x102)	
PanelECU	EnvDispMotorSpeed			MotorState	Motor Run
	EnvDispBMSVoltage			(0x102)	MotorSpeed
	EnvDispBmsSoc			BmsState	Bms Voltage
				(0x103)	BmsSoc

Database and Topology

By using CAN dB ++ Editor Tool under the Vector CANoe to design database, and add signal, message, network node in accordance with Table 1. Generate a network topology designed by the database, as shown in figure 7.

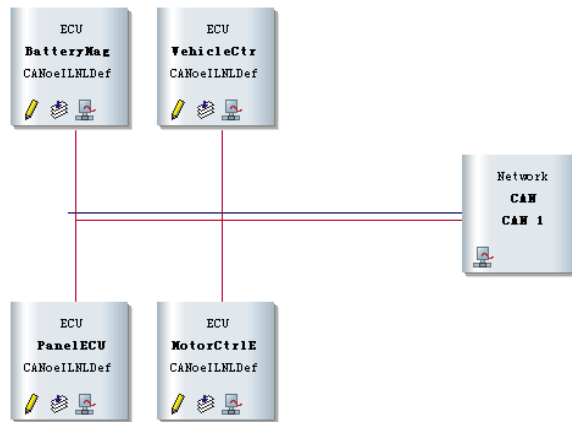


Fig.7: CAN Network Topology

CAPL model of ECU Nodes and Simulation

Design program for communication of each ECU nodes via CAPL language. When the start key is pressed on the panel, vehicle controller sends DriverAction messages as the response of keyboard. VehicleCtrlECU sends DriverAction every 20ms; MotorCtrlECU sends MotorState every 100ms; BatteryMagECU sends BmsState every 1000ms. They are the responses of timer. PanelECU shows them on the panel as soon as it receives MotorState and BmsState. This is the response of CAN messages.

Conclusions

In this paper, a CAN network analyzer has been designed based on USB interface. The CAN network model is tested under CANoe platform after finishing hardware and software design of USB-CAN adapter and the PC program of CAN network analyzer.

References

- [1] Niu J. Design of USB-CAN Controller Based on PIC18F4580[J]. Procedia Engineering, 2012, 29: 329-333.
- [2] Wenfang Xiang, Tao Xing, Binghong Cao. High-speed Data Acquisition System based on USB interface[J]. Beijing: Computer Measurement & Control, 2011, 19(10): 2567-2570.
- [3] Chunhu Liu, Bin Fu, Qinghua Sheng. Development of Micro USB-CAN based on STM32 [J]. Beijing: Computer Measurement & Control, 2013, 21(004): 996-998.
- [4] Wu W, Hong T, Zhu Y, et al. Study on USB Based CAN Bus for Data Measurement System[M]//Intelligent Computing and Information Science. Springer Berlin Heidelberg, 2011: 544-549.
- [5] FAN Y, XIE C, FU L. A Development of USB Device Driver of WDM Based on DriverStudio[J]. Nuclear Electronics & Detection Technology, 2011, 2: 027.
- [6] Mo G M, Guo Y H, Zhang J A. The Design of High-Speed Data-Collection System Based on EZ-USB FX2[J]. Applied Mechanics and Materials, 2013, 373: 1599-1602.