

Design and Implementation of I/O Modules Based on Dual Redundant Can Bus

Bi Xiaodan

School of Instrument Science and Engineering
Southeast University
Nanjing, China
e-mail: bxdseu@126.com

Abstract—In this research, redundant mechanism for CAN Bus communication is designed and implemented to improve reliability of the I/O modules built with CANopen profile. The STM32F107 hardware platform (Cortex-M3 core) is used in the I/O modules. These modules and main station can combine into a complex control system, and can be developed easily. I/O module is widely used in many fields, but how to improve the communication reliability is the research focus. By solving the problem, the reliability of the I/O module will be improved. The main rule to be met in that type of application is that the system must tolerate at least one arbitrary single channel failure. This implies that a general redundant communication has to be provided to fulfill the requirements of a failure-tolerant system. The hardware and software mechanism are designed and implemented in I/O modules. Experimental results show that the I/O module can work properly in the case of arbitrary single channel fault.

Keywords—I/O module; CANopen; redundancy; CAN bus; communication

I. INTRODUCTION

Today, in the fields of automation, large scale systems may comprise up to thousands of I/O-points, by digital modules and analogue modules. The large amount of data and various system functions makes it necessary to implement the system in form of a hierarchical and modular structured architecture with decentralized module. The CAN (Controller Area Network) field bus has now been widely used in mechatronic systems for distributed measurement and control, attributed a lot to its characteristics such as real time, multicast communication ability, and its performance in heavy network load conditions [1]. Now, there are many CAN bus applications not only in the area of automobile, but also elevator, marine automation, and aerospace [2]-[4]. In common can bus application, only one channel communication is designed, if this channel is fault, the system can not work properly. But in some area the safety demand for the system is very high, e.g. alarm, monitoring and control systems, the real time communication is very important. For this purpose, the research of redundant communication mechanism is necessary.

It is general that each company makes the products on their original standard. The I/O modules are not possible to combine with the product of the other companies, and the development cost will increase [5].

In this paper, the dual redundant hardware and software processing mechanism is designed and implemented for I/O modules. The I/O modules include digital input module, digital output module, analogue input module and analogue output module. The communication profile is based on CANopen standards, the other advantage is these modules can be integrated to other system easily.

This paper is organized into five sections including the introduction. Section II gives the hardware redundancy solution, and Section III presents the software redundancy mechanism. Section IV introduces the implementation of the software based on CANopen stack. Finally, the conclusions are presented in Section V.

II. HARDWARE REDUNDANCY

The module hardware is based on the STM32F107 platform, and divided into the core board and the bottom board. The signal input/output ports, power supply and signal conditioning circuit are on the bottom board. The CPU, JTAG debugger and communication interface are on the core board.

STM32F107xx connectivity line family incorporates the high performance ARM® Cortex™-M3 32-bit RISC core operating at 72 MHz frequency, high speed embedded memories (Flash memory up to 256 Kbytes and SRAM up to 64 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, four general-purpose 16-bit timers plus a PWM timer, as well as standard and advanced communication interfaces: up to two I²Cs, three SPIs, two I²Ss, five USARTs, an USB OTG FS and two CANs [6].

These features make the STM32F107xx connectivity line microcontroller family suitable for a wide range of applications such as motor drives and application control, medical and handheld equipment, industrial applications. It combines very high performance, real-time capabilities, digital signal processing, and low-power, low-voltage operation, while maintaining full integration and ease of development.

The I/O modules developed in this paper are based on the STM32F107VCT6 of STM32F107xx family.

A. Hardware Redundancy Mechanism

According to regulations on CAN nodes architecture in CAN protocol, there can be several redundant solutions in the same node [7]-[9].

The three main kinds of can bus redundancy mechanism include software redundancy, system

redundancy and can controller redundancy. This paper adopts the can controller redundancy mechanism.

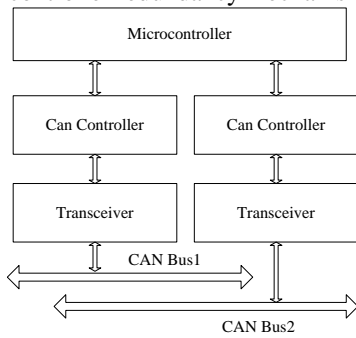


Figure 1. Can Controller Redundancy System Block Diagram

B. The Hardware Implementation of Dual CAN Bus

The CAN bus interface schematic diagram is shown as Fig .2. The two can controllers are embedded in the microcontroller STM32F107VCT6.

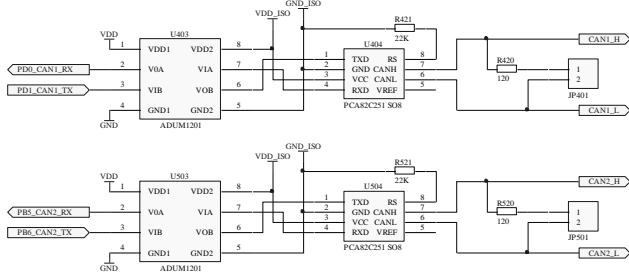


Figure 2. CAN Bus Interface Schematic Diagram

In order to enhance the anti-jamming ability of CAN-bus node, Pin PD1 (CAN1_TX) and Pin PD0 (CAN1_RX) of STM32F107VCT6 are connected to the PCA82C251Y through high-speed digital coupler ADUM1201. Thus electrical isolation is realized between the transceiver and the CPU, the core circuit of I/O module can work safely, and the electrical isolation is also realized among the nodes of CAN bus network.

PCA82C251Y provides differential transmitting capability to the bus and differential receiving capability to CAN controller, and differential driving helps control instantaneous interference in harsh electrical environment.

The CAN2 port, Pin PB6 (CAN2_TX) and Pin PB5 (CAN2_RX) are designed on the same principle as CAN1 port.

When signals on the CAN bus are transmitted in the network, reflection will occur when the signals reach the network endpoint, which will interfere with the transmission of normal signals. To eliminate this interference, two terminal resistances (120Ω) are connected to the two ends of CAN bus network, which can play the dual role of matching bus impedance and eliminating reflection. The neglect of the terminal resistances will greatly reduce the anti-interference ability and reliability of data communication, or even make data communication impossible.

III. SOFTWARE REDUNDANCY MECHANISM

Two CAN controller and channels provide the redundant channels to transmit and receive data, the can bus communication in this paper is based on the CiA Draft

Standard 401 Version 2.0 [10], This document represents the CANopen device profiles for generic digital and analogue input and output modules.

The rule to redundant CANopen network control is based on CiA DSP 307 V1.0, CANopen framework for maritime electronics [6].

A. Outline of CANopen

CANopen is a higher reliability communication protocol of which a physical layer of communication is CAN bus. CANopen was standardized by CiA (CAN in Automation) in 1995. It is used widely in production, with various devices from which high reliability is requested such as vehicles and medical equipment [11][12].

In CANopen, various types of devices have standardized profiles. It means there is uniformity in the standard of hardware, and can be exchange easily for another product.

With CANopen an already widely established and supported standard is available, which provides a very sophisticated solution for implementation of distributed automation systems. It provides standard communication features according to the producer-consumer model as well as the client-server model of communication, network management and system services and a standardized method for the description of devices [13]-[16].

The communication ability of each I/O module in network is monitored continuously by means of the CANopen life guarding mechanism. According to this protocol, the NMT master instance cyclically polls the communication status of each I/O module after expiration of a predefined “guard time”. The module (NMT slave instance) has to respond within the “node life time”. If the module fails to respond within that time, the NMT master will indicate a “node guarding event” to its application. On the other hand, if a module does not receive a status request during its “life time” from the NMT master, the NMT slave issues a “life guarding event” to its application. A typical value for the guard time is 1 second.

Based on CiA DSP 307, the two can bus channels mentioned above are defined as Default CAN line and Redundant CAN line. For identification one CAN line is called the “Default CAN line”, the other is called “Redundant CAN line”. From a technical view there is no difference of the two lines. One of the two CAN lines has the status “active” with respect to the way of message processing on the receiver side.

The different CANopen object types are transmitted and processed by one of the following methods listed in TABLE I.

CANopen defines four kinds of communication object.

PDO (The Process Data Objects), the Process Data Object protocol is used to process real time data among various nodes.

SDO (The Service Data Objects), the SDO protocol is used for setting and for reading values from the object dictionary of a remote device.

NMT (Network Management), the NMT protocols are used to issue state machine change commands.

Emergency Message, emergency messages are triggered by the occurrence of a device internal fatal error situation and are transmitted from the concerned application device to the other devices with high priority.

TABLE I. TRANSMISSION AND PROCESSING OF CANOPEN OBJECT IN MODULE

CANopen object type	Method of transmission and processing
PDO, Emergency Object	Simultaneously transmitted on both CAN lines
Time Stamp Object, Synchronization Object	Simultaneously transmitted on both CAN lines. Only objects received on the active CAN line are processed
SDO	SDO request: Transmitted on one of the two CAN lines which provides the connection to the desired SDO server SDO response: Transmitted on that CAN line on which the SDO request was received
NMT Management Objects, NMT Error Control Protocol	Independently transmitted on both CAN lines and independently processed on both CAN lines

B. Software Framework

According to the CiA DSP 307 profile, the redundant communication mechanism, node guarding mechanism, flying NMT master is specified in detail.

The software architecture of the I/O module based on CiA DSP 307 is shown in Fig. 3.

The I/O modules supporting redundant communication shall be able to operate on two independent CAN lines simultaneously.

Two independent NMT slave state machines based on a node state determination mechanism are included in the module.

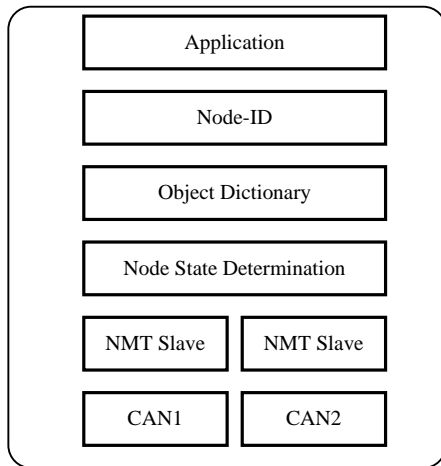


Figure 3. Basic Software Architecture of I/O Module

C. Flying NMT Master and Redundant Communication

In order to apply the flying master principle on a redundant communication system the startup process shown in Fig. 4 has to be followed.

After power-on or reception of NMT “Reset Application” the active CAN line is first determined and then the master negotiation is performed according to CiA DS 302 on the active CAN line. When the device got the mastership, it starts the network on the active CAN line. After this done, it transmits the NMT “Reset Communication” command on the other CAN line in order to get a well defined starting point and starts the network on this CAN line.

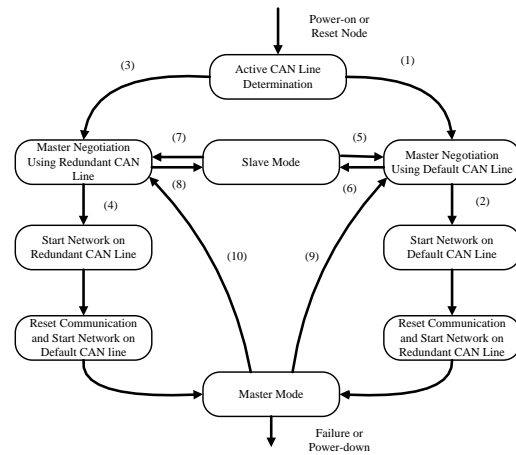


Figure 4. Flying NMT Master Principle on a Redundant Communication System (Only Valid for Modules with Default and Redundant CAN Line Capability)

The condition of state transition among states is as follows.

Condition (1): All devices, which are supporting redundant and default CAN lines function found on the default CAN line, or reception of master negotiation request on the default CAN line, or reception of reset communication command on the default CAN line

Condition (2): Device received mastership

Condition (3): Heartbeat evaluation time 1F60h/02h expired

Condition (4): Device received mastership

Condition (5): Device lost master negotiation

Condition (6): Reception of reset communication on the default CAN line

Condition (7): Device lost master negotiation

Condition (8): Reception of reset communication on the redundant CAN line, or failure of master on both CAN lines

Condition (9): Reception of new master negotiation request on the default CAN line

Condition (10): Reception of new master negotiation request on the redundant CAN line.

D. CANopen Communication Objects and Redundant Communication

1) Process data objects (PDO) in redundant networks

a) Redundant PDO transmission

Each PDO is transmitted on both transmission channels, and the transmission of a PDO is possible within a determined time. A failure of a communication channel therefore can be detected, when the transmission of a PDO is delayed too long.

b) Redundant PDO reception

PDOs are received via two transmission channels (default and redundant channel). It is up to the receiving node (application), how to further use the redundantly received PDOs.

c) Determination of the active CAN Line

If a redundant device detects a missing heartbeat of any other redundant I/O module on the default CAN line and the default CAN line is currently the active CAN line, it transmits the “Indicate Active CAN Line” message on the

redundant CAN line. After power-on or reception of the NMT message “Reset Node” or the NMT message “Reset Communication” the node waits until the reception of the heartbeat messages of all redundant devices or until the “Heartbeat Evaluation Time after Power On or Reset Application” respectively “Heartbeat Evaluation Time after Reset Communication” has elapsed before it applies this mechanism.

A receiving I/O module transmits the “Indicate Active CAN Line” message on the default CAN line after it has received at least 3 heartbeat messages of all redundant nodes on the default CAN line.

d) Emergency messages in redundant networks

An emergency message shall be transmitted over both CAN lines in the same way as defined for redundant PDO transmission. This implies that the object “Inhibit Time EMCY” shall be implemented.

e) Time stamp object in redundant networks

The CANopen module processes received time stamp objects from the active CAN line. In case of a change of the active CAN line a time jitter may occur.

f) Synchronization object in redundant networks

The CANopen module processes received synchronization objects only from the active CAN line. In case of a change of the active CAN line a jitter may occur.

IV. SOFTWARE DESIGN OF REDUNDANT COMMUNICATION

The software is developed on the platform Keil ARM MDK (Microcontroller Development Kit). The hardware is on the STM32F107VCT6 I/O module. The flowchart of the I/O module is shown as Fig .5.

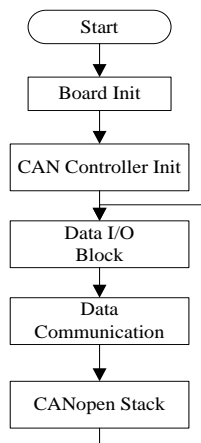


Figure 5. The Flowchart of Main Routine

A. Board Initialization

Board initialization includes setup the microcontroller system, initialize the Embedded Flash Interface, initialize the PLL and update the system frequency variable, etc.

B. CAN Controller Initialization

CAN controller initialization includes the CAN1 and CAN2 pins configuration, baud rate setting, receiving interrupt routine enable, etc., the process and involved function list below.

1) The CAN bus pin remapping

The pins used for CAN bus are not their default function, the remap configuration is needed, including the GPIO redefinition, and the pin function remapping.

```

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_Init(GPIOD, &GPIO_InitStructure);
  
```

The pins for PD0, PB6 and PB5 are configed as above. `RCC_APB1PeriphClockCmd(RCC_APB1Periph_CAN1, ENABLE);`

`RCC_APB1PeriphClockCmd(RCC_APB1Periph_CAN2, ENABLE);`

`GPIO_PinRemapConfig(GPIO_Remap2_CAN1,ENABLE);`

`GPIO_PinRemapConfig(GPIO_Remap_CAN2,ENABLE);`

2) The baud rate setting

The PLL clock is 36 MHz. The can bus baud rate is calculated by (1).

$$Baud_{can} = \frac{RCC_APB1PeriphClock}{(CAN_SJW + CAN_BS1 + CAN_BS2) * CAN_Prescaler} \tag{1}$$

If the baud rate is 125K/s, the parameters in the can controller registers can be set as follows.

`CAN_InitStructure.CAN_SJW = CAN_SJW_1tq;`

`CAN_InitStructure.CAN_BS1 = CAN_BS1_8tq;`

`CAN_InitStructure.CAN_BS2 = CAN_BS2_7tq;`

`CAN_InitStructure.CAN_Prescaler = 18;`

3) The receive interrupt routine enable

In STM32F107, two receive FIFOs are used by hardware to store the incoming messages. Three complete messages can be stored in each FIFO. The FIFOs are managed completely by hardware.

To the I/O module, the can1 and can2 are assigned to FIFO0 and their interrupt enable.

`CAN_ITConfig(CAN1,CAN_IT_FMP0, ENABLE);`

`CAN_ITConfig(CAN2,CAN_IT_FMP0, ENABLE);`

The sending and receiving routine are shown as Fig .6 and Fig .7.

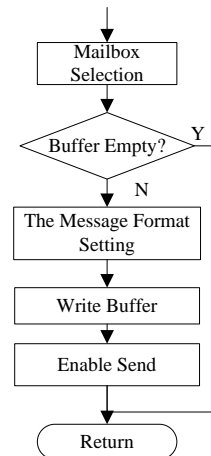


Figure 6. The Flowchart of the Send function

Three transmit mailboxes are provided to the software for setting up messages. The transmission scheduler decides which mailbox has to be transmitted first.

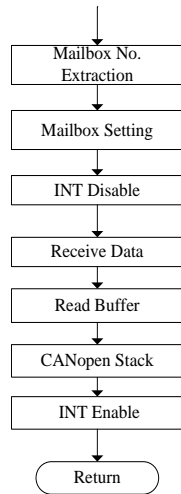


Figure 7. The Receiving Interrupt Serve Routine for CAN

C. Data I/O Block

This block is responsible for data acquisition or control output. Based on the modules type, the block function is divided into digital input, digital output, analogue input, and analogue output.

D. CANopen Stack

The CANopen stack is designed to process the receiving command and send reply. The NMT state transition function is also in this stack.

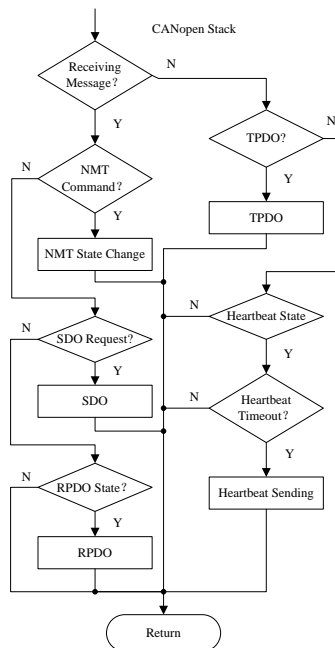


Figure 8. The Flowchart of CANopen Stack

V. CONCLUSION

In this research, the hardware and software mechanism is designed and implemented in I/O modules.

The I/O module developed with CANopen profile in this paper realizes the redundant communication function by the means of redundant can bus, and the module can perform the function of active line judgment, heart beating,

Node/Life guarding, SDO, PDO, etc.. When the default line is fault, the module can change the active line to redundant line and transmit the alarm message.

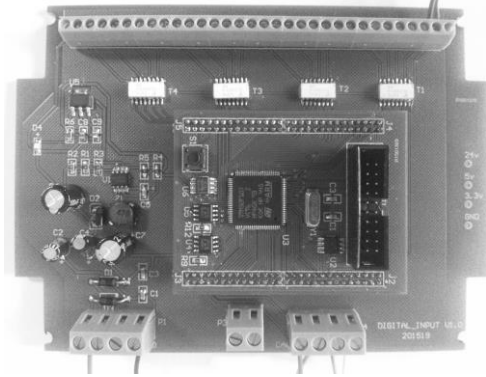


Figure 9. The Digital Input Module

Experimental results show that the I/O module can work properly in the case of arbitrary single channel fault.

REFERENCES

- [1] L.M. Wang, Design and Application of CAN Field BUS. Beijing, China: Electronic Industry Press, 2008
- [2] Jongjun Park, Changwon Lee, Jee-Hun Park, Byeong-cheol Choi, and Jeong Gil Ko, "Exploiting wireless CAN bus bridges for intra-vehicle communications," Vehicular Networking Conference (VNC 2014), IEEE press, Dec. 2014, pp.111-112, doi: 10.1109/VNC.2014.7013320
- [3] Ciardelli, L., Marcenaro, L., Regazzoni, and C.S., "Smart sensing framework for automotive safety based on a dual-camera system and CANbus data," ELMAR, 2011 Proceedings, Sept. 2011, pp.55-58
- [4] Jafary, P., Repo, S., Koivisto, H., "Secure integration of the Home Energy Management System to the battery management system in the customer domain of the smart grid," PES General Meeting | Conference & Exposition, 2014 IEEE press, July 2014, pp.1-5, doi: 10.1109/PESGM.2014.6938986
- [5] Ishiguro, Y., Tanaka, M., Mizukawa, M., Yoshimi, T., Ando, Y., "Design of the redundant RTC-CANopen component," Proc. SICE Annual Conference 2010, Aug. 2010, pp.2812-2817
- [6] STM32F107xx datasheet, STMicroelectronics, March 2014
- [7] CiA DSP 304 V1.0, Framework for safety_relevant communication, January 2011
- [8] CiA DSP 307 V1.0, CANopen framework for maritime electronics, Nov 2002.
- [9] CANaerospace interface specification for airborne CAN applications V1.7. Germany: Stock Flight Systems, 2001
- [10] Wang Li, Guo Lijin, and Liu Zheyuan, "Design of STM32-Based CANopen Motion Control Master in Transfer Robot," Proc. Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2013 Third International Conference on, Sept. 2013, pp.1609-1612, doi: 10.1109/IMCCC.2013.35
- [11] CiA DS 401 Device Profile for Generic I/O Modules Version 2.0, June 2000
- [12] Martinez, D.A.; Poveda, J.D.; Guaqueta, A.F.; Montenegro, D., "Electric vehicle instrumentation by using HMIs," Power Electronics and Power Quality Applications (PEPQA), 2013 Workshop on, vol., no., pp.1.6, 6-7 July 2013
- [13] CiA DS 301 CANopen Communication, Version 4.02, 2002
- [14] CiA DS 302 Framework for Programmable CANopen Devices, Version 3.0, June 2000
- [15] CiA DS 401 CANopen Device Profile for I/O Modules, Version 2.0, December 1999
- [16] CiA DSP 305 Layer Setting Services and Protocol, Version 1.0, May 2000