AWERProcedia
Information Technology
&
Computer Science

1 (2012) 952-957

2nd World Conference on Information Technology (WCIT-2011)

Bootloader design application for embedded systems by using controller area network

Ismail Yabanova [a] *, Sezai Taskin [b], Huseyin Ekiz [c], Hasan Cimen [a]

[a]*Afyon Kocatepe University, Electrical Education Department, Afyonkarahisar, Turkey*
[b]*Celal Bayar University, Electronics and Automation Department, Turgutlu-Manisa, Turkey*
[c]*Suleyman Sah University, Faculty of Engineering and Natural Sciences, İstanbul, Turkey*

Abstract

In this study, the software programs that enable programming of microcontrollers on the controller area network (CAN) through the network have been developed. The first developed program is the bootloader that operates on a microcontroller and enables the microcontroller to program itself through the incoming data. The second program operates on the computer and is developed with the LabVIEW™ program. With this program, the Hex file written and compiled by the users is read and sends to the microcontroller through the CAN bus.

Keywords: Bootloader, CAN, Microcontroller, LabVIEW

Selection and peer review under responsibility of Prof. Dr. Hafize Keser.

1. Introduction

Today, microcontrollers are widely used in mechatronics systems. We witness the comprehensive use of microcontroller based mechatronics products in home, office, health, communication, transportation, entertainment, civil and military sectors [1]. In literature, there are many studies made by using microcontrollers. For example, microcontroller-based air conditioner [2], design and implementation of a microcontroller-based fuzzy logic average current-mode control for dc-dc converters [3], a surveillance system for forest environments and protected natural areas [4] and meteorological data-acquisition system [5] are designed. With the developing technology, the microcontrollers incorporate many fundamental features such as communication units, motor control devices, ADC and comparators. For this reason, its use in the industry

*ADDRESS FOR CORRESPONDENCE; İsmail, Yabanova, Afyon Kocatepe University, Electrical Education Department, Afyonkarahisar, Turkey
E-mail address: iyabanova@gmail.com/ Tel.: +90-272-228-1311

increases day by day. The microcontrollers can also change their own program memories. Thanks to this available feature, the users can realize their bootloader operations. The bootloader software is a special program loaded in the microcontroller by means of a programmer. Once this software is loaded, the program memory can be changed by means of the communication modules (RS-232, SPI, CAN, USB etc) in the microcontroller without necessity of any programmer circuit. By using of different communication protocols in microcontrollers, various bootloader systems have been developed. Some of them are; serial bootloader for PIC24F devices [6], I2C bootloader for the PIC16F1XXX [7], high-speed bootloader for PIC16 and PIC18 devices [8], USB and UART bootloader for PIC32 devices [9]. The PIC 18F4580 controller of the Microchip Company was used as a microcontroller. Thanks to the ECAN module in this microcontroller, it can be easily used in CAN bus applications. The ECAN module supports the CAN 1.2, CAN 2.0A, CAN 2.0B passive and CAN 2.0 active versions in the CAN protocol.

In this study, two new software programs have been developed. These programs enable programming of microcontrollers on the CAN through the network. The first developed program is the bootloader that operates on a microcontroller and enables the microcontroller to program itself through the incoming data. The second program operates on the computer and is developed with the LabVIEW™ program. With this program, the Hex extended file written and compiled by the users is read and sends to the microcontroller through the CAN bus. With the developed programs, the program of any microcontroller that exists in the CAN can be changed without taking apart it from the system and necessity of any hardware type programmer. Thanks to this, an infrastructure that can be used in the education of remote accessed embedded systems has been established. In the second section of the article, information about the intel hex file and in the third section, general structure of the system and programs that have been developed for programming process from the CAN bus are given and the article is finalized.

2. Intel Hex File Format

Through representation of binary coded data with ASCII codes, the hex file format has been established. The hexadecimal file is divided in to record blocks that have data, record type, length, memory address and checksum information inside each. Each line consists of six parts;

Start code; it consists of one byte ASCII ':' character.

Byte number: it is in one byte length and determines the length of data in the data area.

Address: it consists of 16 –bit address information that indicates from which address the data in the memory area will be programmed.

Record type: it is an area in length of one byte where the type of data area exists. Six different record types have been defined [10]. This record types are;

- '00' Data Record (8,16 or 32 bit format)
- '01' End of File Record (8,16 or 32 bit format)
- '02' Extended Segment Address Record (16 or 32 bit format)
- '03' Start Segment Address Record (16 or 32 bit format)
- '04' Extended Linear Address Record (Only 32 bit format)
- '05' Start Linear Address Record (Only 32 bit format)

Data: it is an area where data array exists.

Chksum: it is an area where total of values of all areas except the start code and CHKSUM areas has complement to 2'. In Figure 1, a model hex file and its areas are given [11].

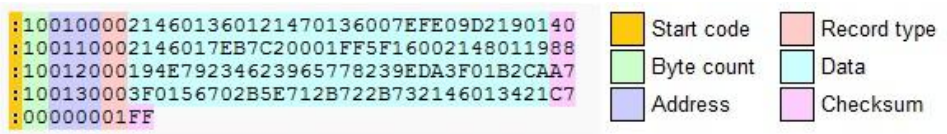

Fig. 1. Hex file example and record areas

3. Programming Microcontrollers Via CAN Bus

For programming of microcontrollers via CAN bus, two programs have been written; one operating on a microcontroller and other operating on a computer. By selecting the program written on a computer and the microcontroller to be programmed, the desired program can be loaded. The bootloader program operating on a microcontroller ensures the program memory to be changed by interpreting the messages from CAN bus. At the start of program memory, 4 Kbyte area has been separated for 18F4580 microcontroller bootloader processes and this area can be adjusted as protected against overwriting by mistake. In Fig. 2, a block diagram of the developed system and the restructured situation of the program memory are given. When the area at start of the program memory is used for bootloader processes, reset vector and addresses of cutting vectors with high and low priority must be redefined in area of program memory.

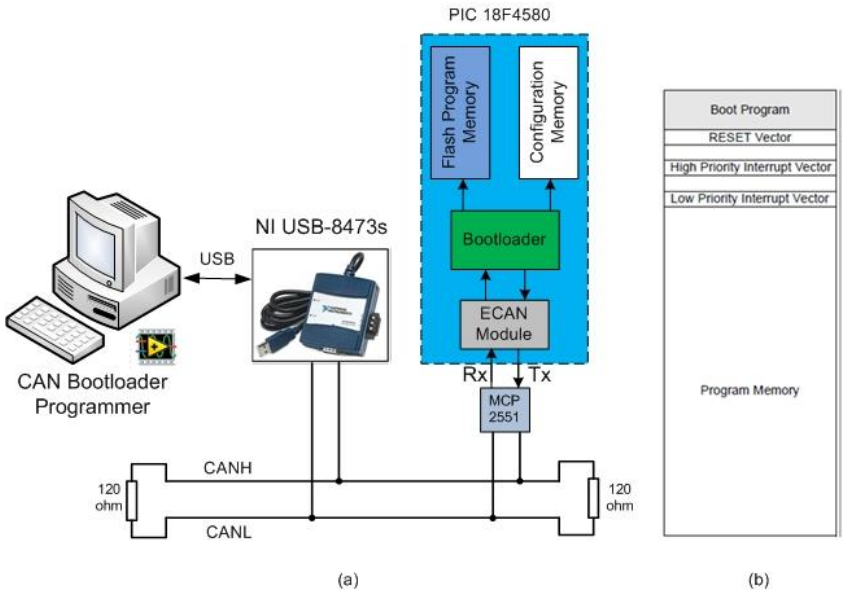

Fig. 2. Block diagram of the developed system and program memory of microcontroller

Combination of the flash technology and robust network communication ability in one device makes programming via network an attractive option [12]. Some points must be considered before realizing the bootloader processes on CAN. For instance, all the nodes on CAN can be programmed simultaneously or only one node program can be changed. As the CAN messages are received by all nodes, when only one node is desired to be programmed, a direct communication with that node must be provided. To make the programming process more productive, some features like self-verification must be given to the device to be programmed. In this way, the message traffic during programming can be decreased to half. The other subject is that which priority will be given to boot loader process when programming will be made in an active system. In case high priority is given to the bootloader process, there may be delays in communications of other nodes when data is begun to be sent to the node to be programmed. In critical systems that require low delay period for data transmitting and receiving, this may hinder operation of the system. In case low priority is given to the bootloader process, as the programmer will require the network to be empty to sent messages, this may cause long programming periods.

3.1. CAN bootloader programmer

The hex-extension files written by users and made ready to be loaded in microcontroller by being compiled are read with the designed programmer and sent to the microcontroller to be programmed via CAN. The hex file

to be loaded is sent to the microcontroller with CAN messages in the least number as much as possible. Besides, controlling of sent messages whether they reached correctly is made by the target microcontroller. After the program file to be loaded and microcontroller is selected, the program to be loaded is sent to the microcontroller by pressing on the “program” button. In Fig. 3, CAN boot loader programmer is given.

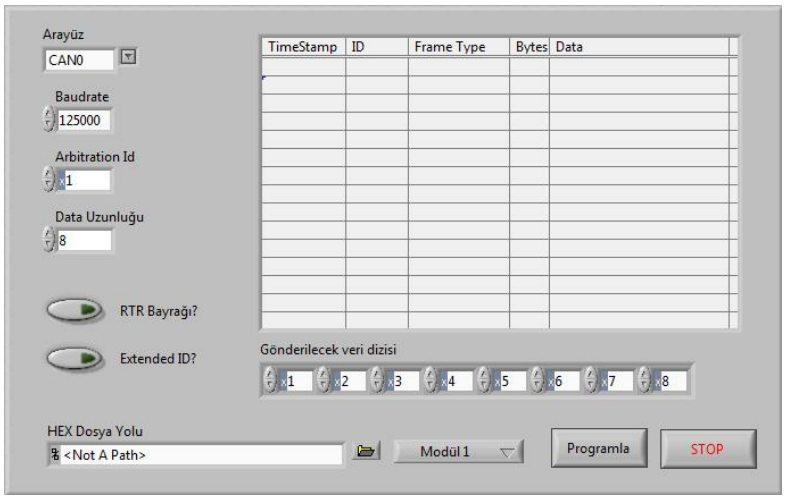

Fig. 3. LabVIEW based CAN Bootloader Program user interface

For passing of microcontroller to the bootloader mode, an event must be defined. This event can be defined with a hardware event such as pressing on a button after the microcontroller reset or with a special data coming via network. In the developed programmer, an ID is defined for passing of each microcontroller to bootloader mode. When the microcontroller to be programmed is selected, the program sends a message with the related ID and it is waited the microcontroller to be programmed will send a message whether it is ready for the bootloader mode. After the message stating that the related microcontroller is ready, the information about how many CAN messages will be used to send data on each line and messages in sufficient number are sent consecutively. After all the line is sent, with a different ID, it is communicated to the microcontroller that sending of line has been completed. If the microcontroller receives all the information correctly, it informs it with a message and the programmer begins to send other lines. In this way, all the lines are sent.

3.2. The bootloader program

The bootloader program is a special program that operates in the microcontroller and ensures loading of new program in the program memory by taking the microcontroller into bootloader mode with the messages coming from the CAN control card programmer. Once the bootloader program is loaded in the microcontroller with a programmer, it becomes ready to use. The bootloader program is loaded from the start of program memory of microcontroller. When a message is sent from the CAN control card programmer with an ID that belongs to the related microcontroller, that controller takes itself to the bootloader mode and sends a message stating that it is ready. The CAN control card programmer sends the data in the hex file line by line and when the data in a line is finished, it communicates it to the microcontroller with a message. The microcontroller records the data sent by messages and when it receives the message stating that all the line is finished, it determines the number of data in the related line, address of program memory the data will be loaded and kind of received data respectively from other data if the first byte begins with ':' and after the data except the checksum and start data are added and their complement to 2 is taken, the obtained data is compared to the data in the checksum area at the end of line. If the data coincide, this means that the data in the transmitted line have been taken correctly and the received data are recorded by the microcontroller in the related areas. After the recording process is completed,

with a message, it is stated that other lines will be sent to the programmer. When all the lines are recorded, the microcontroller is automatically reset and recently loaded program is operated. In Fig. 4, the flowchart of bootloader program that operates in the microcontroller is given.

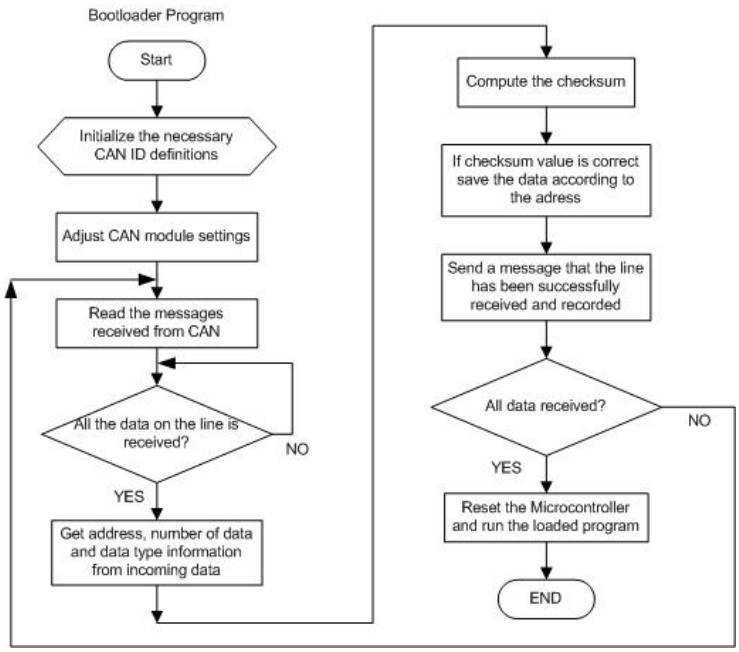

Fig. 4. Flowchart of Bootloader Program

4. Conclusion

In this study, by using the CAN protocol, the software that ensure programming of desired microcontrollers loaded with the bootloader software on the network have been developed. The programmer in the computer has been developed with the LabVIEW program and for communication of the program with the CAN bus, USB-8473 device by NI company has been used. With the developed bootloader structure, the desired microcontroller on the CAN bus can be programmed in place without need of an additional programmer hardware. Besides, by using the web server component of the LabVIEW program, it is possible to make remote accessible microcontroller programming applications.

Acknowledgements

This study was sponsored by a project of Scientific Research Projects Coordination Unit of Afyon Kocatepe University (Project no: 09.TEF.03).

References

[1]. A. H. G. Al-Dhaher, Microprocessors and Microsystems, 25 (7), 317-328 (2001).
[2]. İ. Çolak, R. Bayındır and H. T. Kahraman, 2nd Int. Conf. on Technical and Phsical Problems in Power Engineering, 186-190 (2004).
[3]. D. He, R.M. Nelms, Applied Power Electronics Conference and Exposition, Aubum University, 3, 2000-2006 (2005).
[4]. J.N.Moutinho, F.D.Mesquita, N.Martins, R.E.Araujo, Emerging Technologies and Factory Automation, 2, 191 - 194 (2003).
[5]. S. Ameur, M. Laghrouche, A. Adane, Renewable Energy, 24 (1), 19-30 (2001).
[6]. B. Ivey, A Serial Bootloader for PIC24F Devices, Microchip Technology Inc. (2008).
[7]. C. Toma, An I2C™ Bootloader for the PIC16F1XXX Enhanced Core, Microchip Technology Inc. (2009).
[8]. E. Schlunder, High-Speed Serial Bootloader for PIC16 and PIC18 Devices, Microchip Technology Inc. (2010).
[9]. G. Ramachandra, PIC32 Bootloader, Microchip Technology Inc. (2011).

[10]. Hexadecimal Object File Format Specification, Revision A, January 6, 1988, Intel Corporation (1998).
[11]. Wikipedia Intel HEX, http://en.wikipedia.org/wiki/Intel_HEX (2011).
[12]. Ross M. Fosler, A CAN Bootloader for PIC18F CAN Microcontrollers, Microchip Technology Inc. (2003).