Bootloader and Configuration Manager*

Kim Nordlöf
knf12001@student.mdh.se

ABSTRACT

The customer requested that the microcontrollers should be able to download new firmware via the CAN-bus. This is achieved by installing a bootloader in each microcontroller. The bootloader receives CAN messages containing hex data and programs it into the microcontroller's flash memory. The data is sent from a configuration manager that takes the contents of hex files and embeds the data into CAN messages. Since the Naiad AUV has several microcontrollers the configuration manager also keeps track of each microcontroller.

1. INTRODUCTION

The customer gave two requirements for how to handle the firmware.

The first was that the firmware should be downloaded via the CAN-bus. The second stated that the firmware should be handled by a configuration manager.

The customer suggested programming the microcontrollers with a bootloader that can download firmware via the CAN-bus. The AT90CAN128 microcontrollers that are used have good support for bootloaders so that is the method that has been chosen to update the firmware.

2. METHOD

The customer provided resources from a previous project named weRobot [3]. This project was part of the course CDT310 (Foundations of Software Engineering) back in 2008 and a bootloader was created for AT90CAN128 together with a PC client that could send hex files from the computer to the bootloader via CAN.

2.1 Bootloader

The weRobot bootloader is an improvement made upon Atmel's Slim CAN Bootloader [2]. There was a problem with running the weRobot bootloader, so this project uses the original bootloader from Atmel, but with the extra functionalities of the weRobot bootloader reimplemented.

The Slim CAN Bootloader is written in C and it is designed to be compiled with the IAR Embedded Workbench. In this project the workbench used was the AVR Time-limited license v6.30.

2.2 Configuration Manager

The Configuration Manager is built upon the weRobot PC client. The PC client is written in Ada and compiled with GNAT GPL Ada Development Environment 2013. The GUI is made in the GtkGlade GUI builder so it requires the GtkAda [1] library to compile. No information was found specifying in which version of GtkAda it was made, however the deprecated components have been replaced so that it now works in GtkAda 3.4.2 which is the latest version at the time of writing. As of now the client is compiled as an executable on a computer running Windows 8.

*This report was written during the fall of 2013 in an advanced level project course at Mälardalen University, Sweden.

3. IMPLEMENTATION

This section describes the changes that have been made to the bootloader and PC client to fit the requirements.

3.1 Bootloader

3.1.1 EEPROM byte for boot status

EBSB is a byte created in the EEPROM that determines if the bootloader should run or if the flash program should run. If the byte is set to 255 (0xFF) then it will jump to the flash program. Any other value allows the bootloader to run. This byte changes according to the demands of the PC client and since it is written in the EEPROM the value remains the same, even after reset.

3.1.2 EEPROM byte for ID

Each AT90CAN128 with a bootloader is assigned an ID number between 0 and 255. ENNB is a byte created in the EEPROM that stores the ID of the microcontroller. This makes it possible for the bootloader to identify itself when communicating with the PC client.

3.1.3 Message to change EBSB

The bootloader will now listen for one more message. If the bootloader receives a BOOT_STATUS message with CAN ID 11 then it will change the EEPROM byte EBSB into whatever value was sent in the first byte of the CAN message.

3.2 Configuration Manager

3.2.1 New protocol for sending messages

The weRobot PC client was designed to be connected directly to the CAN-bus and send CAN messages. However in the Naiad AUV, it will be connected to the system via a BeagleBone Black through an Ethernet connection. For this reason it was necessary to replace the CAN protocol with a new protocol using TCP sockets instead. This new ethernet protocol is designed to stream the contents of a CAN message over the Ethernet connection and then rebuild it into a CAN message at the receiver.

For more information on where this is used see step 8 in Figure 1.

3.2.2 List of configured nodes

The PC client has a configuration file named card_names.cfg. This file matches the ID of different nodes in the system with a name. The PC client will display all these configured nodes in a list. When the user wants any node to enter the bootloader it will first be selected from this list.

For more information on where this is used see step 1 in Figure 1.

3.2.3 Button to start bootloader

The start bootloader button is used to make the node that has been selected in the list of configured nodes to enter the bootloader. The PC client will first send a mode message with the first byte set to 2 for bootloader mode. This is to tell the whole system that one or more nodes are going to be programmed through the bootloader. After waiting a brief moment it will send a START_BOOTLOADER message. This message signals to the selected node that it should enter the bootloader. The first byte in this message holds the ID of the node. Once completed it will send a DISCOVER message to which all nodes that has entered the bootloader will respond. The PC client will then display all the discovered nodes that replied.

For more information on where this is used see step 2 in Figure 1.

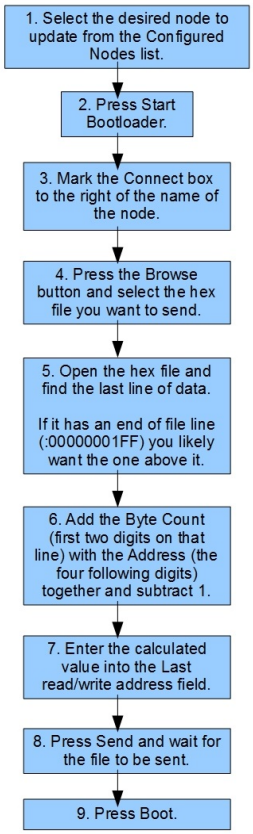

Figure 1: Activity diagram example

4. RESULT

Tests with the bootloader and configuration manager have been successful in programming AT90CAN128 microcontrollers by sending messages from the PC to the BeagleBone Black and then onwards onto the CAN-bus.

There has yet to be any test to see how it works together with the whole system because the necessary hardware was unavailable.

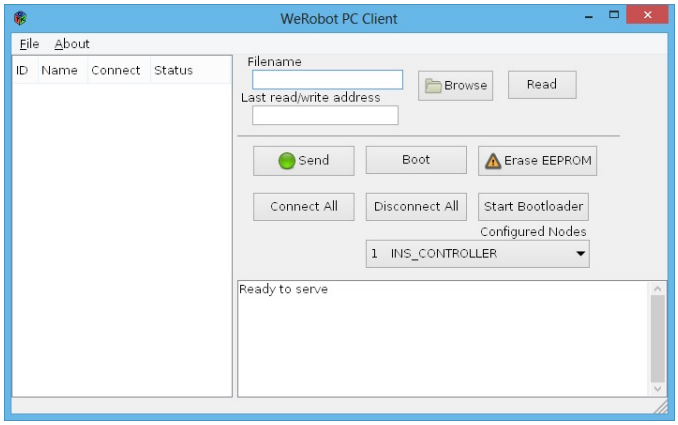

Figure 2: Screenshot of the Configuration Manager

5. CONCLUSION

As mentioned in Section 2, there was a problem with using the weRobot bootloader. This bootloader was adapted into AVR Studio instead of IAR Embedded Workbench. If the problem could be fixed then there is no need to use the IAR Embedded Workbench which requires the user to sign up for a time-limited license which is undesirable.

It should be possible to connect from the PC to the BeagleBone Black with websockets and make a configuration manager in a browser. The good thing about this is that it would remove some of the restrictions on which operating systems that the configuration manager can run on.

6. REFERENCES

[1] Gtkada. `http://libre.adacore.com/tools/gtkada/`. Accessed 2014-01-10.

[2] Avr076: Avr can - 4k boot loader. `http://www.atmel.com/Images/doc8247.pdf`. Accessed 2014-01-07.

[3] Werobot mdh. `http://www.idt.mdh.se/kurser/cdt310/08ht/projects/gr2/`. Accessed 2014-01-09.