IGREBOOT bootloader

IGREBOT team

Contents

Abstract

This document describes the IGREBOOT related softwares.

1 Introduction

1.1 Overview

To ease software management on the 2012 robot, the different electronic board microcontrollers are flashed with a bootloader. The bootloader is in charge of updating the board software on demand. This is done from a host PC software. The commands are carried over CAN. Since PCs usually do not have CAN interfaces, a serial to CAN bridge is implemented.

The project source code is temporarly maintained in a GIT repository:

https://github.com/texane/igreloader

The project depends on the following softwares:

- a working LINUX system with standard GNU tools,
- MPLABX version 1.0 .

Note that WINDOWS and MACOSX are not yet supported.

The project tree is organized as follow:

- src/blinker
- src/bridge
- src/device
- src/host

2 Bootloader

The bootloader waits for commands targeted at the specified device. The following commands are implemented:

- CMD_ID_WRITE_PMEM
- CMD_ID_READ_PMEM
- CMD_ID_WRITE_CMEM
- CMD_ID_STATUS
- CMD_ID_GOTO

Figure 1 shows the memory organization for a DSPIC33F128GP802 microcontroller. Refer to the reference manual, figure 4-1 for more information.

The red areas are reserved for the bootloader and should not be used by the user application. To prevent the application from using reserved areas, a modified linker script can be used. Such a linker script can be found in:

igreloader/build/blinker.X/igreboot_p33FJ128GP802.gld

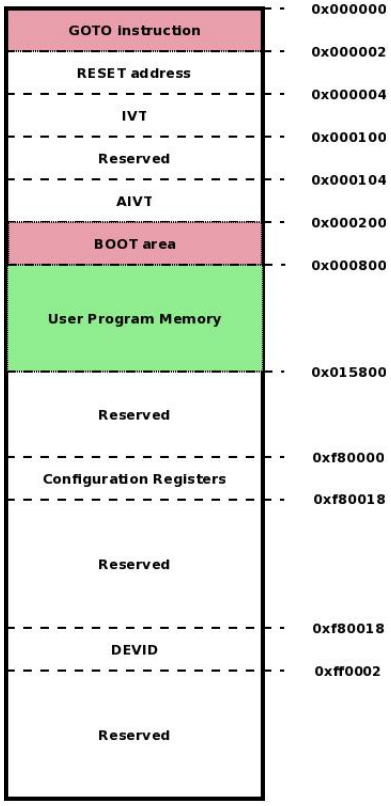

Figure 1: memory map

3 Host PC software

TODO: command line

The following listing contains the commands used in a typical session:

```
# go to the project source tree
texane@dell:~$ cd /home/texane/repo/igreloader/

# compile a sample application
texane@dell:~/repo/igreloader$ cd build/blinker.X/
texane@dell:~/repo/igreloader/build/blinker.X$ make

# return to build directory and compile the host software
texane@dell:~/repo/igreloader/build/blinker.X$ cd ..
texane@dell:~/repo/igreloader/build$ ./do_build_host.sh

# write the application to device flash
./a.out write /dev/ttyUSB0 2 ./blinker.X/dist/default/production/blinker.X.production.hex noconf

# order the device to execute the application
texane@dell:~/repo/igreloader/build$ ./a.out goto /dev/ttyUSB0 2 800
```

4 Serial to CAN Bridge

As PCs usually do not have a CAN interface, a serial to CAN bridge has been implemented. Its purpose is to wrap CAN payloads plus minimal meta information into a serial frame. The wrapping format is described in Figure XXX.

5 Sample applications

TODO: blinker

6 TODOS

- bootloader linker map from 0x200 to 0x800
- add CAN support to bootloader
- implement serial to CAN brigde