Using BootLoader

PIC24FJ48GA002 with CCS C Compiler

we make computers do more.

Embedded Systems Design

(FIBO)

Santi Nuratch., Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

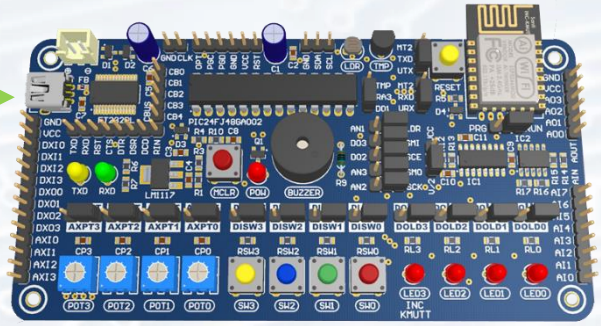
santi.inc.kmutt@gmail.com, santi.nur@kmutt.ac.th

Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**

Bootloader_v1.8.exe & Bootloader.hex

What they are?
How they work?

USB Cable



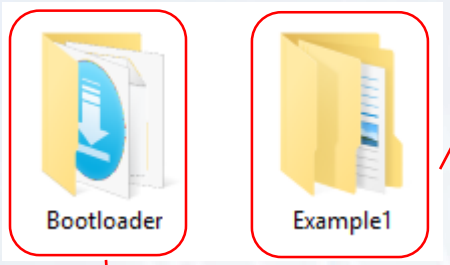
Bootloader_v1.8.exe

The Bootloader_v1.8.exe is a computer application used to send an user application (*.hex) compiled by CCS C compiler to the firmware running in the MCU. Communication between computer and MCU uses RS-232 (UART Protocol).

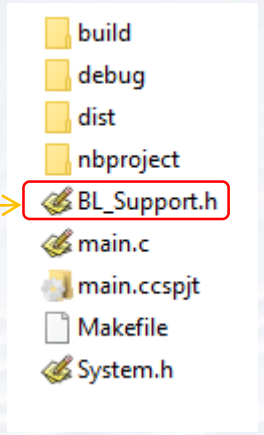
Bootloader.hex

The Bootloader.hex is a firmware that preprogrammed in the bootloader section (0x0000-0x0BFF). The firmware receives an user application (*.hex) from the Bootloader_v1.8.exe running on the computer and writes the *.hex to the user area flash memory starting from 0x0C00

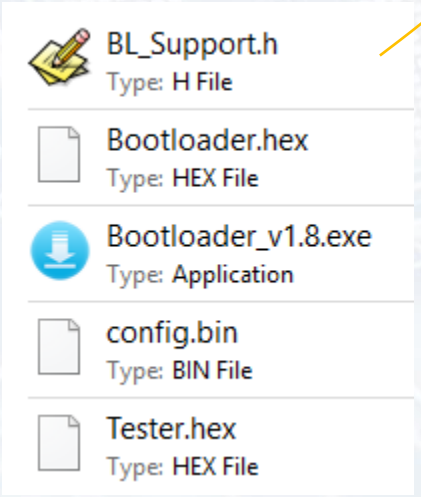
Board_Support



MPLABX Project Example
Example1.X



Bootloader
Supported Files



Bootloader support header file

Firmware working as a bootloader

Bootloader application running on PC

Configuration file of the Bootloader_v1.8

Board tester (user application)

Example1 – main.c

The **BL_Support.h** is required. It **MUST** be included in the source file (main.c)

```
#include "System.h"
#include "BL_Support.h"
#include <stdio.h>
#include <stdlib.h>

unsigned int t1_ticks = 0;
BOOLEAN  t1_isr_flag = FALSE;

#INT_TIMER1
void TIMER1_ISR(void) {
    t1_ticks++;
    t1_isr_flag = TRUE;
}
```

```
void main(void) {
    unsigned int counter = 0;
    disable_interrupts(GLOBAL);
    setup_timer1(TMR_INTERNAL |TMR_DIV_BY_8, 2000);
    enable_interrupts(INT_TIMER1);
    enable_interrupts(GLOBAL);

    printf("System Ready!\r\n");

    while(TRUE) {

        output_toggle(PIN_A2);
        output_toggle(PIN_A4);
        output_toggle(PIN_B2);
        output_toggle(PIN_B3);

        unsigned int t1 = t1_ticks;
        printf("%d : %d\r\n", counter++, t1);
        delay_ms(200);
    }
}
```

System.h

Other preprocessors can be added in this file. **DO NOT** change the read lines.

```
#include <24FJ48GA002.h>

#fuses FRC_PLL, OSCIO
#fuses NOIOL1WAY, NOWDT, NODEBUG, NOWRT, NOPROTECT, NOJTAG
#device *=16 ADC=10
#use delay(clock=16000000)

/* UART1 connection (see in schematic diagram) */
#PIN_SELECT U1RX = PIN_B12
#PIN_SELECT U1TX = PIN_B13

/* UART2 connection (see in schematic diagram) */
#PIN_SELECT U2RX = PIN_B14
#PIN_SELECT U2TX = PIN_B15

/*
 * To map the standard io functions, e.g., printf(), kbhit() and others to
 * the UART1 the UART1 must defined after UART2. The last defined UART will be
 * mapped to the standard io functions.
 */
#use rs232(baud=9600, UART2, stream=ESP)
#use rs232(baud=9600, UART1) // UART1 will be mapped to the standard io functions
```

BL_Support.h

DO NOT modify anything in this file.

```
/*********************************************************************/
/*                  BOOTLOADER SUPPORT CONFIGURATION                 */
/*                        (DO NOT CHANGE)                            */
/*********************************************************************/
/* Bootloader */
#define LOADER_PAGES        3
#define ERASE_SIZE          2048
#define LOADER_SIZE         ((LOADER_PAGES * (ERASE_SIZE/2)) - 1)
#define LOADER_END          LOADER_SIZE
#define LOADER_ADDR         0xC00
#build (reset=LOADER_END+1, interrupt=LOADER_END+5)
#org 0, LOADER_END {}
```

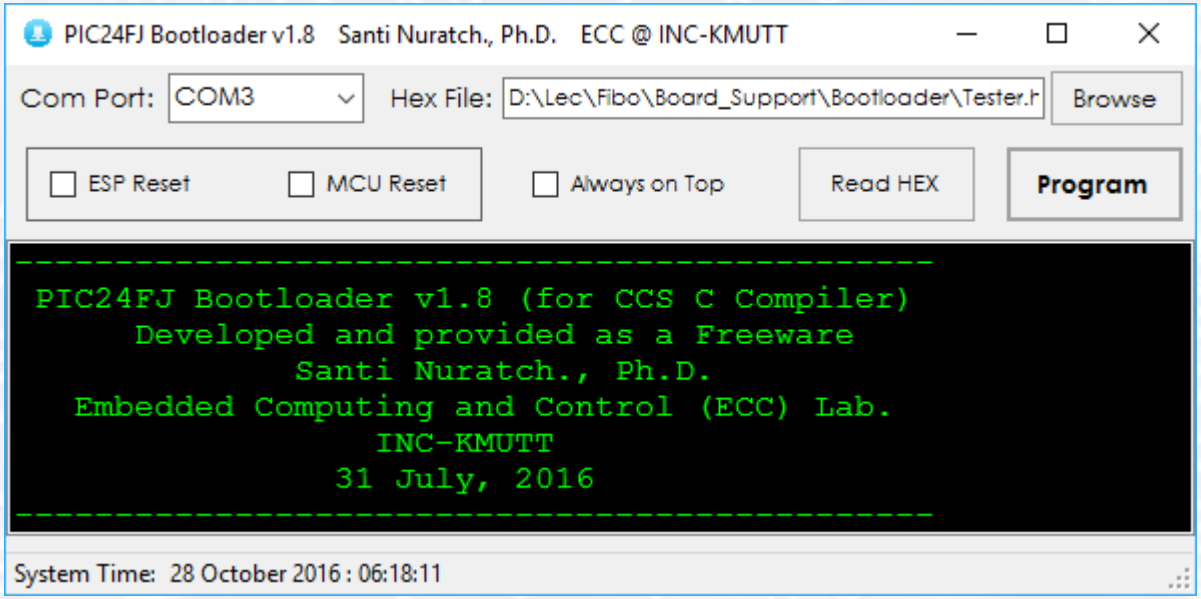
This file tells the compiler how to generate the *.hex of user application. If this file is modified, the user application may not run or the bootloader/firmware my overwritten (destroyed)

Using the Bootloader_v1.8

1. Compile the source file using MPLABX+CCS to generate a *.hex

2. Connect computer and MCU board using USB cable

3. Run/Open the Bootloader_v1.8

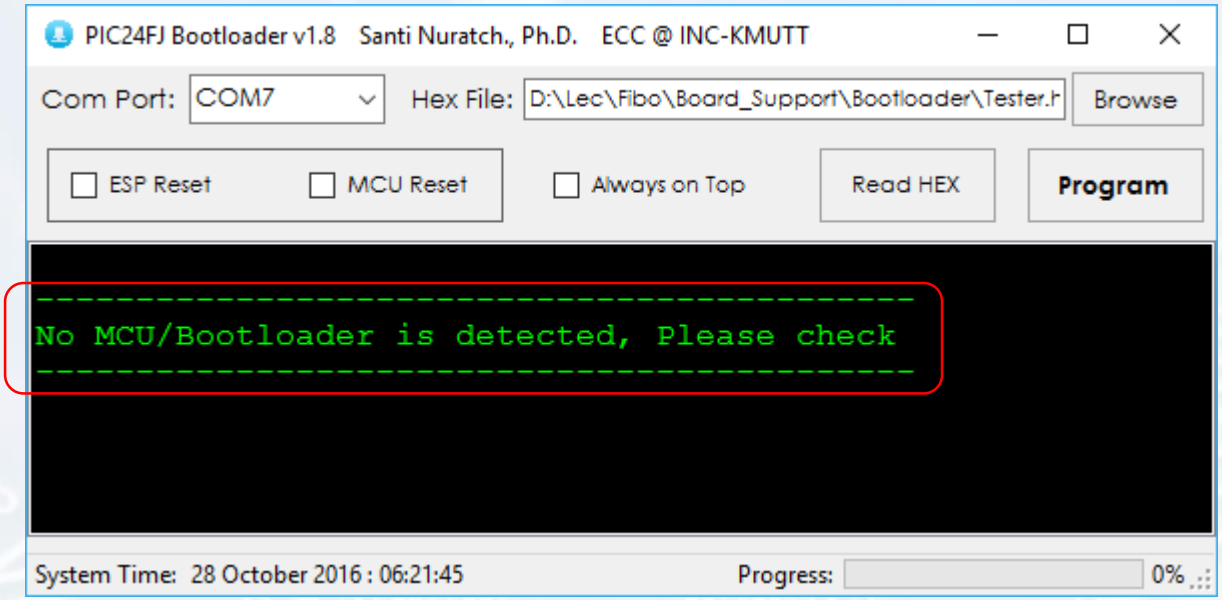


4. Choose a communication port

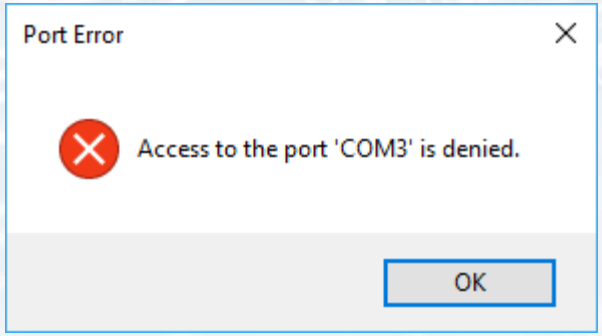
5. Browse to the target *.hex file

6. Click Program button to transfer the *.hex to MCU

Error and Information Messages

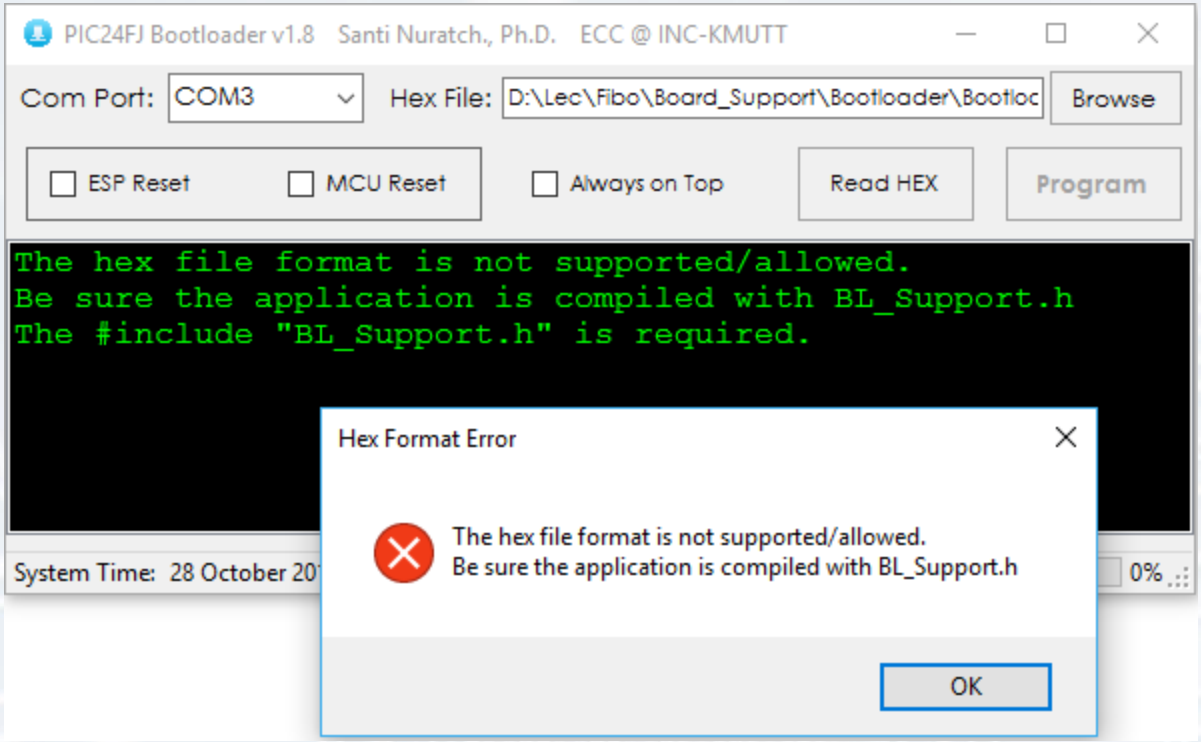


Cannot detect the MCU board, check if the communication port is corrected selected



Cannot open/access the port, be sure the port is not opened by another software

Error and Information Messages



The target *.hex format is not supported/allowed. If the source file may not compiled with the BL_Support.h. Check if the BL_Support.h is included in the source file, i.e., main.c

Thank You

Santi Nuratch, Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

santi.inc.kmutt@gmail.com, santi.nur@kmutt.ac.th

Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**