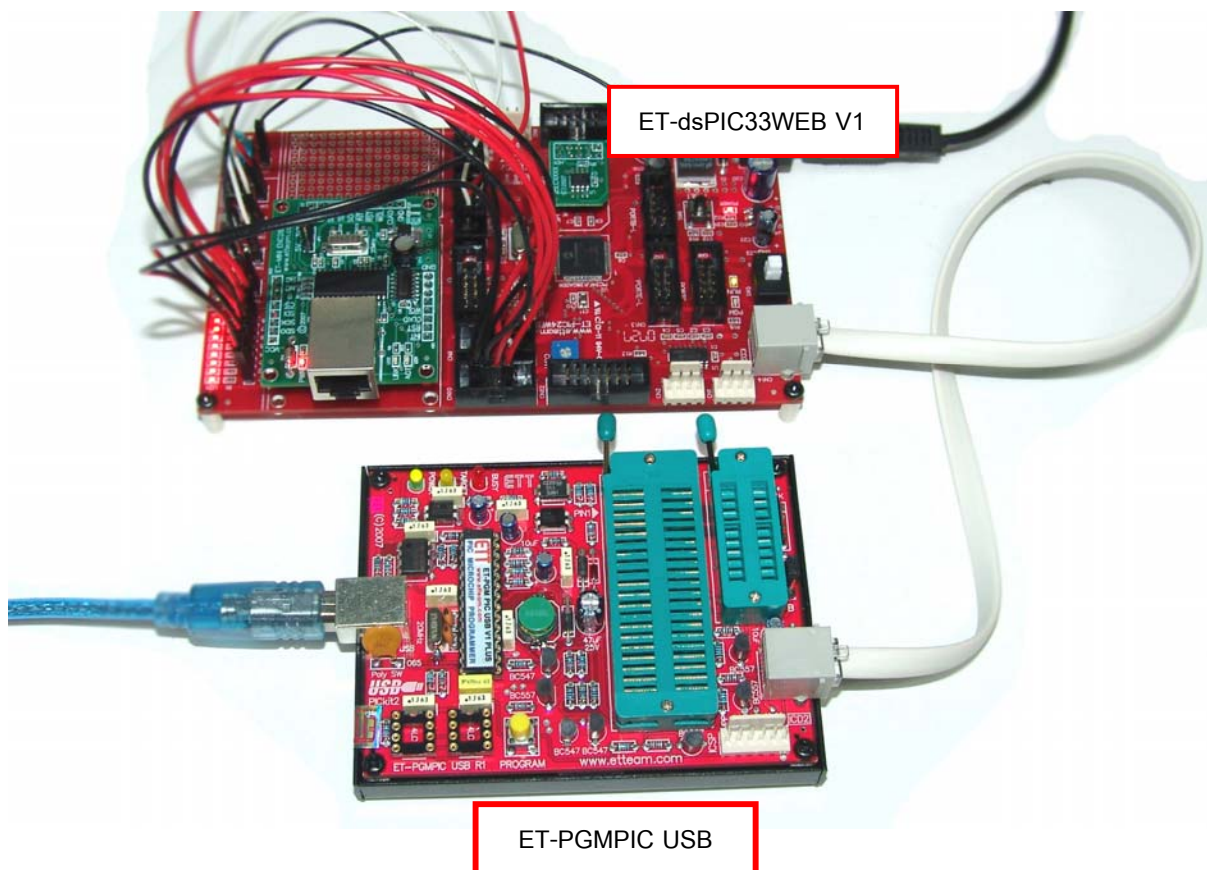


To Develop Program of Board

The method to develop program of Board "ET-dsPIC33WEB V1.0" is divided into 2 parts; development program and download program. In the part of development program of board, it is writing program to control and command MCU to run follows the designed target; in this case, it depends on user or programmer who develops program decides to use which language to develop program. Moreover, user must decides to choose program to compile the command of the selected language into code command that is in the format of Hex File to download into Flash Memory of MCU and it makes user can command MCU to run follows the written program as desired.

In the part of download program or Download Hex File that is developed program for Flash Memory of MCU must use Hardware equipments to develop program. If it is Board "ET-dsPIC33WEB V1.0" must use device to program data into MCU internal board as In-Circuit type through Programmer or In-Circuit Debugger that supports application with MCU No.dsPIC33FJ128GP708. This MCU uses standard Connector "ICD2" of Microchip such as ICD2, and PICKit2. Additionally, ETT has device to support this application that is "ET-PGMPIC USB" as shown in the example below.



To Write Program for Board by MPLAB C30

MPLAB C30 or C30 Tools is C Language Program to compile command of dsPIC MCU that is developed by Microchip; in this case, we do not mention about restrictions and details of writing C Language Program. If user wants to develop program for dsPIC by C Language but user does not know how to write C Language Program, we recommend user to learn and understand the method to write C Language Program from books that mentions about the standard restrictions of "ANSI C" first. Additionally, user can learn and study other minor restrictions of MPLAB C30 from the provided documents and manuals of MPLAB C30 by Microchip. In this case, user can download it from website of Microchips or Folder of **"C:\Program Files\Microchip\MPLAB C30\docs"** that installs Program MPLAB C30. Now, we only mention about the method to set Configuration values in program to use with Board "ET-dsPIC33WEB V1.0". If user wants to use Program MPLAB C30 to write program, user must install 2 programs of Microchips as follows;

- **MPLAB IDE** is the Program Text Editor of Microchips; nowadays (October 2004), it is version 7.21 and user can download data free without any charge from website of Microchips.
- **MPLAB C30** is C Compiler to be command of dsPIC; normally user must purchase this program but user can download the provided demo Program from Microchips as same as MPLAB IDE.

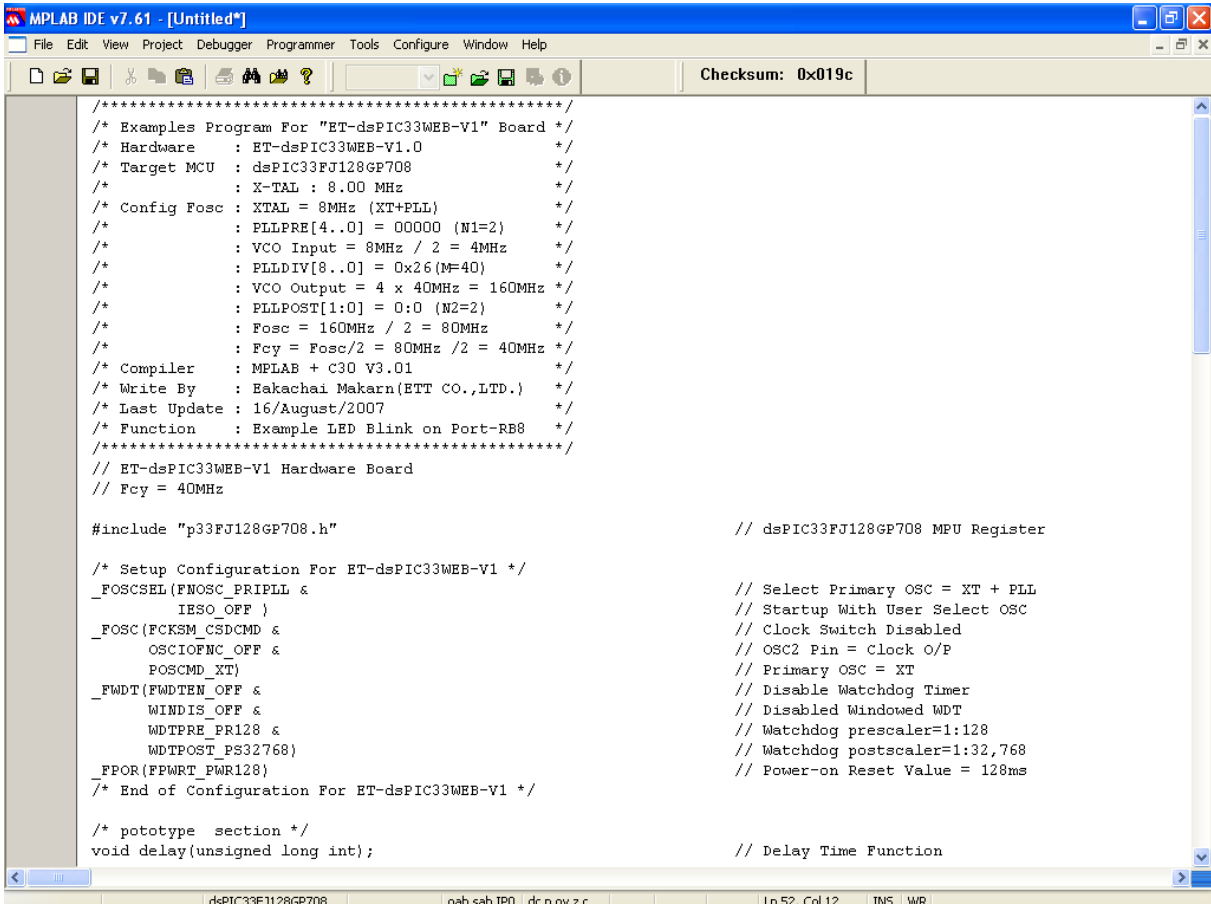
ETT has already downloaded both programs into the CD-ROM that is provided with Board "ET-dsPIC33WEB V1.0". In this case, user must install program in the set of MPLAB IDE first and we recommend user to install program of MPLAB IDE follows the Default values of installed program that is **"C:\Program Files\Microchip\MPLAB IDE\"** because it is the most convenient to use. After user has already installed Program MPLAB IDE successfully, user must restart computer first if it is the first time to use program because it makes Program MPLAB IDE can run without any problem later. Then, user must install Program MPLAB C30; in this case, we recommend user to install program follows the Default values of installation program because it is the most convenient to use, especially setting the connection between MPLAB IDE and MPLAB C30. Now, we will mention about the position to install the program only; if

user install program in the different position, please actually learn and understand the method to modify program and install program.

Example of Creating C Language Program of MPLAB C30

When user has already installed program to connect commands between MPLAB IDE and MPLAB C30 completely, next user can call program MPLAB C30 through Program MPLAB IDE instantly. All setting values will be stored in Configuration of program forever until user wants to change to a new one. In this case, we will mention about an example to write C Language Program to be a guideline for user as follows;

1. Run Program of MPLAB IDE, and create a file to write C Language Source Code through Command **"File → New"**. Next, write C Language Source Code in Work Sheet of Program MPLAB IDE as shown in the example below.



```

MPLAB IDE v7.61 - [Untitled*]
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x019c

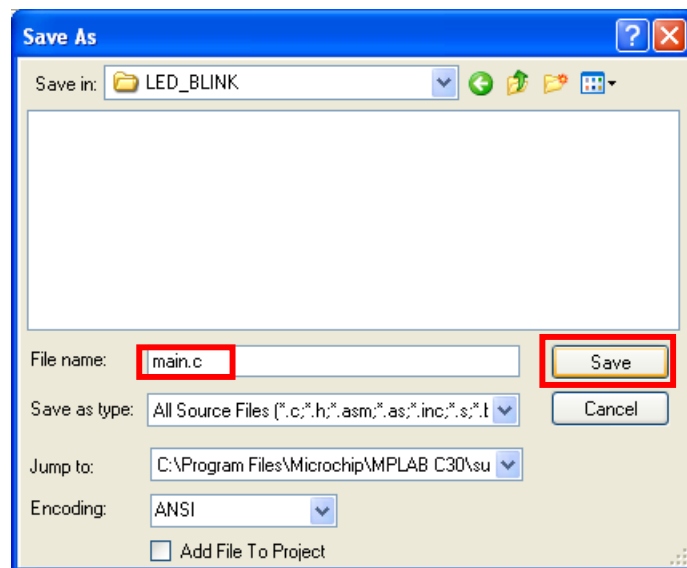
/******
/* Examples Program For "ET-dsPIC33WEB-V1" Board */
/* Hardware      : ET-dsPIC33WEB-V1.0          */
/* Target MCU    : dsPIC33FJ128GP708          */
/*              : X-TAL : 8.00 MHz             */
/* Config Fosc   : XTAL = 8MHz (XT+PLL)        */
/*              : PLLPRE[4..0] = 00000 (N1=2)   */
/*              : VCO Input = 8MHz / 2 = 4MHz   */
/*              : PLLDIV[8..0] = 0x26(M=40)     */
/*              : VCO Output = 4 x 40MHz = 160MHz */
/*              : PLLPOST[1:0] = 0:0 (N2=2)     */
/*              : Fosc = 160MHz / 2 = 80MHz     */
/*              : Fcy = Fosc/2 = 80MHz / 2 = 40MHz */
/* Compiler     : MPLAB + C30 V3.01           */
/* Write By     : Sakachai Makarn(ETT CO.,LTD.) */
/* Last Update  : 16/August/2007              */
/* Function     : Example LED Blink on Port-RB8 */
/******
// ET-dsPIC33WEB-V1 Hardware Board
// Fcy = 40MHz

#include "p33FJ128GP708.h"                // dsPIC33FJ128GP708 MPU Register

/* Setup Configuration For ET-dsPIC33WEB-V1 */
_FOSCSEL(FNOSC_PRIPLL &                // Select Primary OSC = XT + PLL
        IESO_OFF )                    // Startup With User Select OSC
_FOSC(FCKSM_CSDCMD &                  // Clock Switch Disabled
      OSCIOFNC_OFF &                 // OSC2 Pin = Clock O/P
      POSCMD_XT)                      // Primary OSC = XT
_FWDT(FWDTEN_OFF &                   // Disable Watchdog Timer
      WINDIS_OFF &                   // Disabled Windowed WDT
      WDTPRE_PR128 &                  // Watchdog prescaler=1:128
      WDTPOST_PS32768)                // Watchdog postscaler=1:32,768
_FPOR(FPWRT_PWR128)                   // Power-on Reset Value = 128ms
/* End of Configuration For ET-dsPIC33WEB-V1 */

/* pototype section */
void delay(unsigned long int);          // Delay Time Function
  
```

2. Save the written file as C Language Text File; in this case, specify name to be **"main.c"** and then save it in the Folder name **"..\Microchip\MPLAB C30\examples\led_blink\main.c"**. User must use Command **"File → Save As..."**; next, create Folder name **led_blink** under the Folder of examples, specify name to be **"main.c"** and finally, select **"Save"** as shown in picture below.



When user has already saved file as **"main.c"** completely, group of characters that are typed will be divided into groups by using different colors to display the results of each function in each character group; for example, character group of Comment, Command, and Variables. It is an advantage of MPLAB IDE because it can separate and display character groups depend on function through colors, so it makes user can read program easier and more convenient as shown in the picture below.

```

MPLAB IDE v7.61 - [C:\Program Files\Microchip\MPLAB\30\ET-dSPIC_WEB\dsPIC33WEBV1\LED_BLINK\main.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x019c

/* *****
/* Examples Program For "ET-dsPIC33WEB-V1" Board */
/* Hardware      : ET-dsPIC33WEB-V1.0 */
/* Target MCU    : dsPIC33FJ128GP708 */
/* X-TAL : 8.00 MHz */
/* Config Fosc   : XTAL = 8MHz (XT+PLL) */
/* PLLPRE[4..0] = 00000 (N1=2) */
/* VCO Input = 8MHz / 2 = 4MHz */
/* PLLDIV[8..0] = 0x26 (M=40) */
/* VCO Output = 4 x 40MHz = 160MHz */
/* PLLPOST[1:0] = 0:0 (N2=2) */
/* Fosc = 160MHz / 2 = 80MHz */
/* Fcy = Fosc/2 = 80MHz / 2 = 40MHz */
/* Compiler      : MPLAB + C30 V3.01 */
/* Write By      : Sakachai Makarn(ETT CO.,LTD.) */
/* Last Update   : 16/August/2007 */
/* Function      : Example LED Blink on Port-RB8 */
/* *****
// ET-dsPIC33WEB-V1 Hardware Board
// Fcy = 40MHz

#include "p33FJ128GP708.h" // dsPIC33FJ128GP708 MPU Register

/* Setup Configuration For ET-dsPIC33WEB-V1 */
_FOSCSEL(FNOSC_PRIPLL & // Select Primary OSC = XT + PLL
        IESO_OFF) // Startup With User Select OSC
_FOSC(FCKSM_CSDCMD & // Clock Switch Disabled
      OSCIOFNC_OFF & // OSC2 Pin = Clock O/P
      POSCMD_XT) // Primary OSC = XT
_FWDT(FWDTEN_OFF & // Disable Watchdog Timer
      WINDIS_OFF & // Disabled Windowed WDT
      WDTPRE_PR128 & // Watchdog prescaler=1:128
      WDTPOST_PS32768) // Watchdog postscaler=1:32,768
_FPOR(FPWRT_PWR128) // Power-on Reset Value = 128ms
/* End of Configuration For ET-dsPIC33WEB-V1 */

/* pototype section */
void delay(unsigned long int); // Delay Time Function

int main(void)
{
    unsigned char LED; // LED Buffer Display

    /* Start of PLL Config Fcy = 40MIPS */
    CLKDIV &= 0xFFE0; // PLLPRE[4..0] = 00000 (N1=2)
    CLKDIV &= 0xFF3F; // PLLPOST[1:0] = 00 (N2=2)
    PLLFBD = 0x26; // PLLDIV[8..0] = 000100110 (M=40)
    while(!OSCCONbits.LOCK); // Wait PLL to Lock
    OSCCONbits.CLKLOCK = 1; // Disable PLL Modify
    /* End of PLL Config Fly = 40MIPS */

    /* Start Config RB8 = Output */

```

```

#include "p33FJ128GP708.h" // dsPIC33FJ128GP708 MPU Register

/* Setup Configuration For ET-dsPIC33WEB-V1 */
_FOSCSEL(FNOSC_PRIPLL & // Select Primary OSC = XT + PLL
        IESO_OFF) // Startup With User Select OSC
_FOSC(FCKSM_CSDCMD & // Clock Switch Disabled
      OSCIOFNC_OFF & // OSC2 Pin = Clock O/P
      POSCMD_XT) // Primary OSC = XT
_FWDT(FWDTEN_OFF & // Disable Watchdog Timer
      WINDIS_OFF & // Disabled Windowed WDT
      WDTPRE_PR128 & // Watchdog prescaler=1:128
      WDTPOST_PS32768) // Watchdog postscaler=1:32,768
_FPOR(FPWRT_PWR128) // Power-on Reset Value = 128ms
/* End of Configuration For ET-dsPIC33WEB-V1 */

/* Pototype Section */
void delay(unsigned long int); // Delay Time Function

int main(void)
{
    /* Start of PLL Config Fcy = 40MIPS */
    CLKDIV &= 0xFFE0; // PLLPRE[4..0] = 00000 (N1=2)
    CLKDIV &= 0xFF3F; // PLLPOST[1:0] = 00 (N2=2)
    PLLFBD = 0x26; // PLLDIV[8..0] = 000100110 (M=40)
    while(!OSCCONbits.LOCK); // Wait PLL to Lock
    OSCCONbits.CLKLOCK = 1; // Disable PLL Modify
    /* End of PLL Config Fly = 40MIPS */

    /* Start Config RB8 = Output */

```

```

AD1PCFGLbits.PCFG8 = 1;           // AN8(RB8) = Digital Mode
TRISBbits.TRISB8 = 0;             // RB8 = Output
LATBbits.LATB8 = 0;               // Start = ON LED

while(1)                           // Loop Continue
{
    LATBbits.LATB8 = !LATBbits.LATB8; // Toggle Output RB8
    delay(2000000);                  // Display LED Delay
}

/*****
/* Delay Time Function */
/* 1-4294967296 */
*****/
void delay(unsigned long int count1)
{
    while(count1 > 0) {count1--;}    // Loop Decrease Counter
}

```

Picture displays example Source Code to test operation.

For this example, it commands Port RB8 to be an Output function to drive LED operation; in this case, LED endlessly alternates on and off as blinking light. The method to test operation of this program if using with Board ET-dsPIC33WEB V1.0 is to interface signal from RB8 (PORTB-H) with LED1 Output of board and user will see operation of program; LED1 will continue on and off as blinking light all the time.

The most important and necessary of Program is a part of command to set Configuration Bit value into MCU. This value will be compiled and set operation of Fuse Bit internal MCU in the step of Download Code into MCU. This example is setting MCU to run with external Clock 8 MHz with Phase Lock Loop Circuit (PLL). The example program that is written by ETT will be referred to Configuration as follows;

- Set MCU to run from signal Clock: XTAL = 8MHz (XT+PLL)
- Set operation of Phase Lock Loop Circuit as follows;
 - PLLPRE[4..0] = 00000 (N1=2)
 - VCO Input = 8MHz / 2 = 4MHz
 - PLLDIV[8..0] = 0x26 (M=40)
 - VCO Output = 4 x 40MHz = 160MHz

- o $PLLPOST[1:0] = 0:0$ ($N2=2$)
- o $F_{osc} = 160\text{MHz} / 2 = 80\text{MHz}$
- o $F_{cy} = F_{osc}/2 = 80\text{MHz} / 2 = 40\text{MHz}$

```

/* Setup Configuration For ET-dsPIC33WEB-V1 */
_FOSCSEL(FNOSC_PRIPLL &                               // Select Primary OSC = XT + PLL
        IESO_OFF )                                     // Startup With User Select OSC
_FOSC(FCKSM_CSDCMD &                                   // Clock Switch Disabled
      OSCIOFNC_OFF &                                   // OSC2 Pin = Clock O/P
      POSCMD_XT)                                       // Primary OSC = XT
_FWDT(FWDTEN_OFF &                                     // Disable Watchdog Timer
      WINDIS_OFF &                                     // Disabled Windowed WDT
      WDTPRE_PR128 &                                   // Watchdog prescaler=1:128
      WDTPOST_PS32768)                                 // Watchdog postscaler=1:32,768
_FPOR(FPWRT_PWR128)                                    // Power-on Reset Value = 128ms
/* End of Configuration For ET-dsPIC33WEB-V1 */

```

In the part of command to set operation of Phase Lock Loop to multiply frequency by external signal Clock 8.00MHz, it makes Frequency F_{osc} 80MHz. In this case, CPU can process data with maximum speed 40MIPS ($F_{cy} = 40\text{MHz}$) as shown in example below.

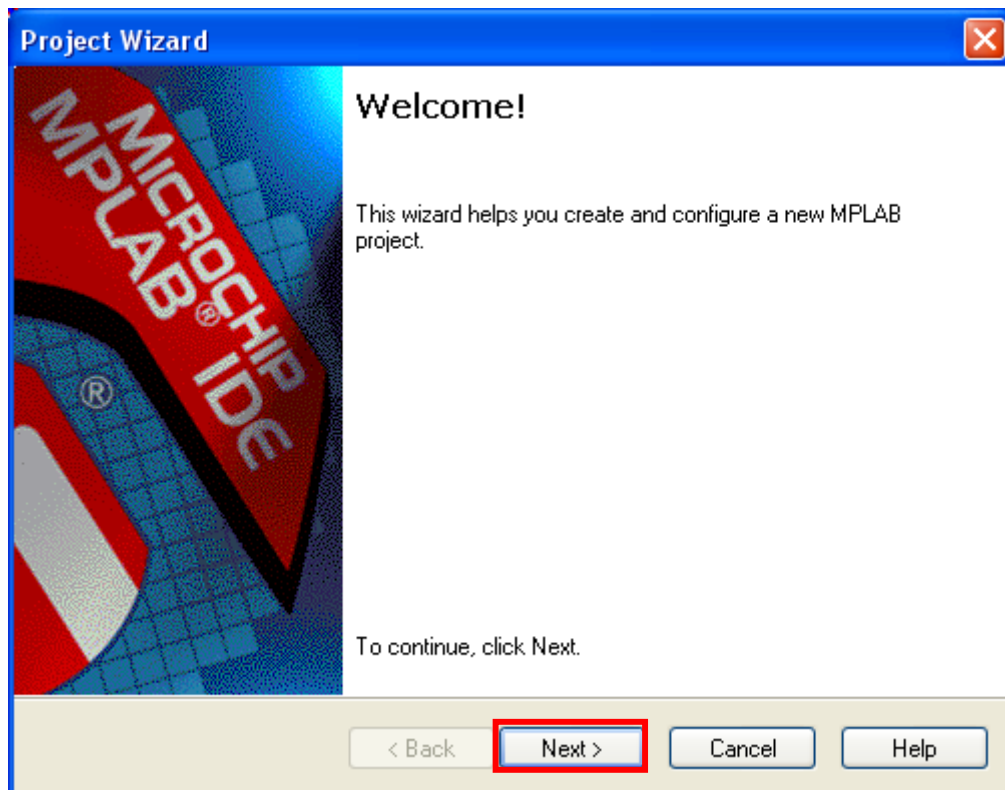
```

/* Start of PLL Config Fcy = 40MIPS */
CLKDIV &= 0xFFE0;                                       // PLLPRE[4..0] = 00000 ( $N1=2$ )
CLKDIV &= 0xFF3F;                                       // PLLPOST[1:0] = 00 ( $N2=2$ )
PLLFBD = 0x26;                                         // PLLDIV[8..0] = 000100110 ( $M=40$ )
while(!OSCCONbits.LOCK);                               // Wait PLL to Lock
OSCCONbits.CLKLOCK = 1;                                // Disable PLL Modify
/* End of PLL Config Fly = 40MIPS */

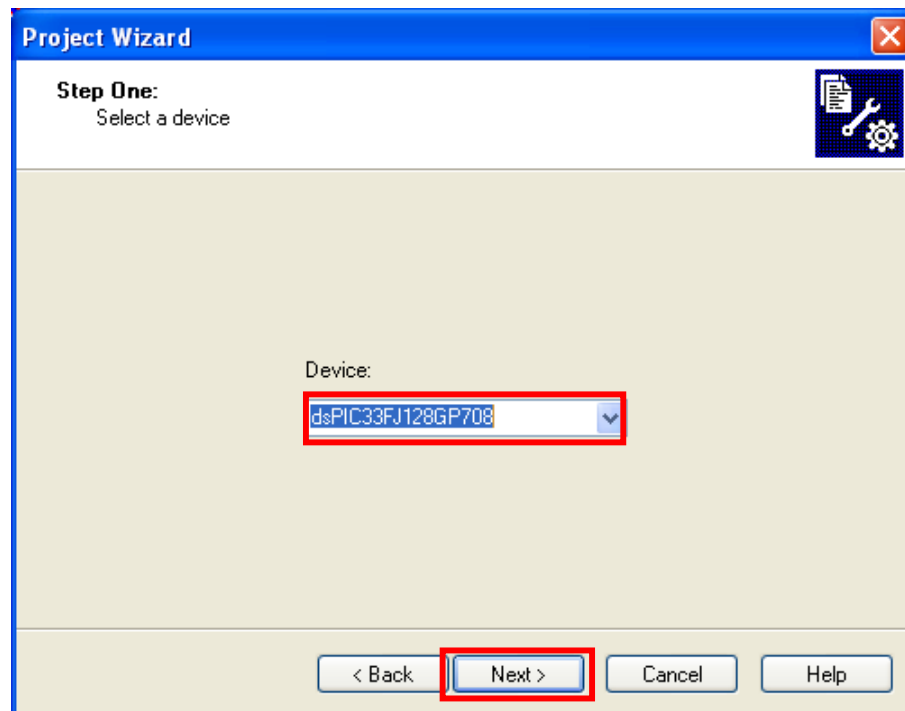
```

3. Create Project File to include the related files together because C Language of MPLAB C30 is designed to be flexible to apply, so it creates and separates files into many groups depend on its function, so it makes user call files to run with the written Source Code easily because it reduces time to write program and user does not write all Source Code by self. In this case, user only includes files that are created by MPLAB C30 into the written Source Code together and then compile program and finally user will get the complete file to apply. From the example of the written Source Code above, there is a line of including filename **"p33FJ128GP708.h"**; so it does not waste the time to declare its name and register positions of "dsPIC33FJ128GP708" but user can refer to registers in program instantly. The method to set specifications of Project File is described below;

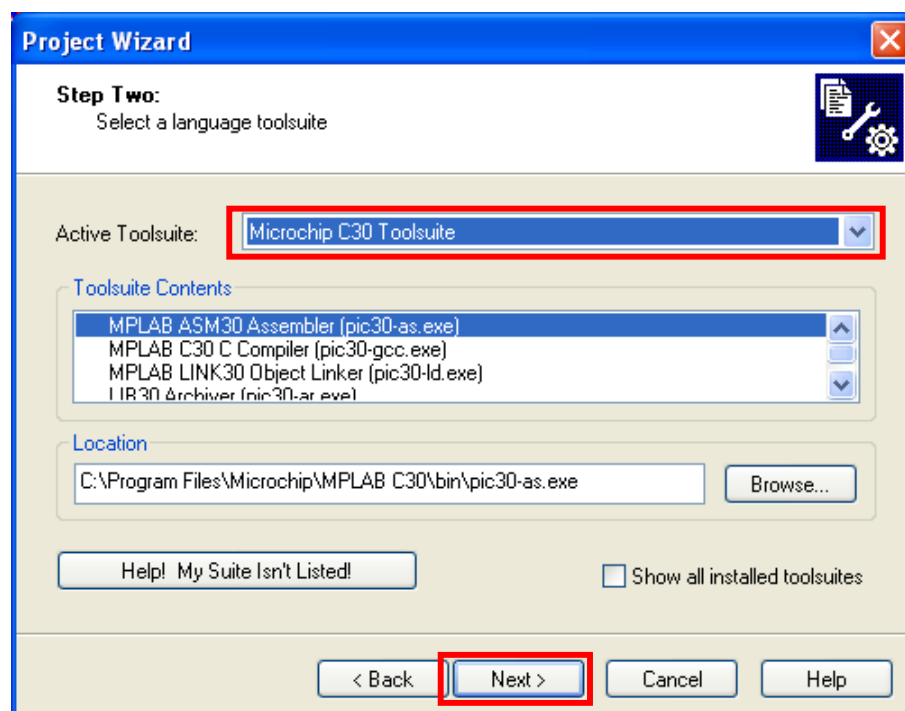
- Set specifications of Project File by using Command **"Project → Project Wizard..."** and its result will be shown as in the example below. Then, select **"Next >"** to go to the next step.



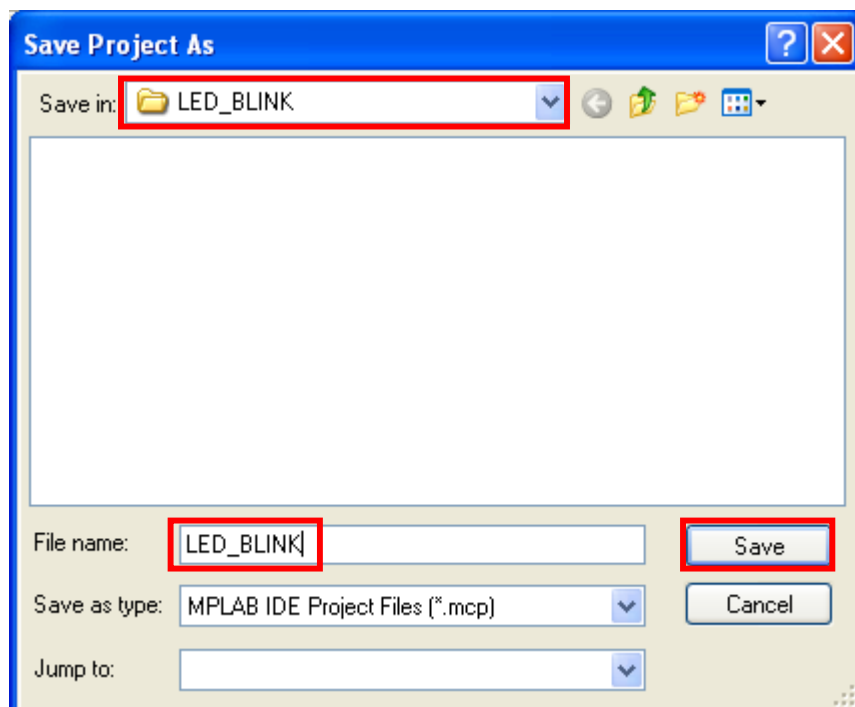
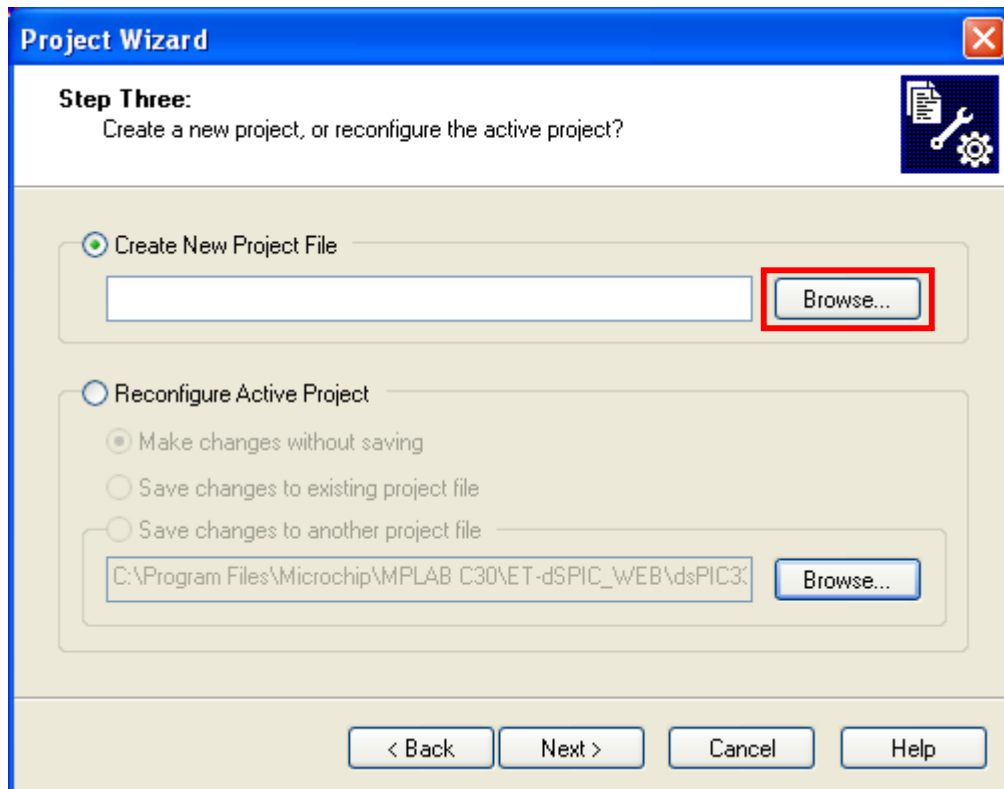
After select **"Next >"**, program will wait for user to set MCU number to use with the written program; in this case, user must select **"dsPIC33FJ128GP708"** and then select **"Next >"** to go to the next step as shown in the picture below.



In this step, user must select which program to compile command because MPLAB IDE can be used with many programs; in this case, user should select Program of MPLAB C30 by setting option of **"Active Toolsuite"** to belong to **MPLAB C30** and then set option to be **"Microchip C30 Toolsuite"** as shown in the picture below. Finally, click **"Next >"**.



In this step, user must specify Project name and Folder position to store files that is run by Project; in this case, user must specify its name to be **"LED_BLINK"** as shown in the picture below and then select **"Next >"**.

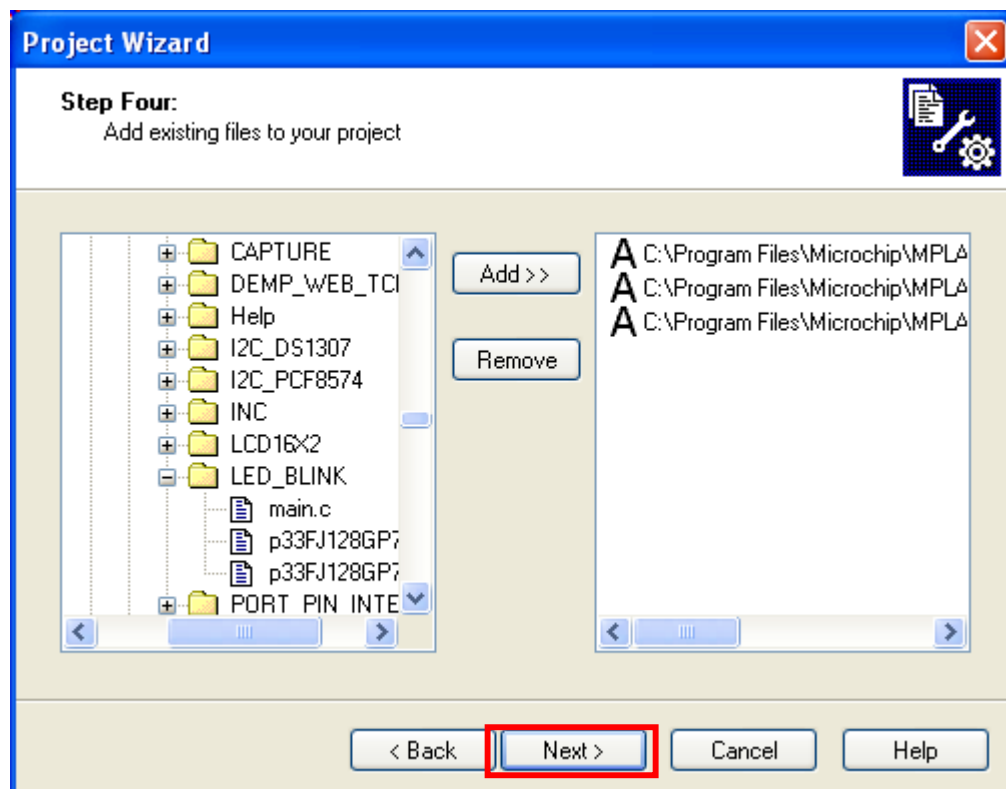
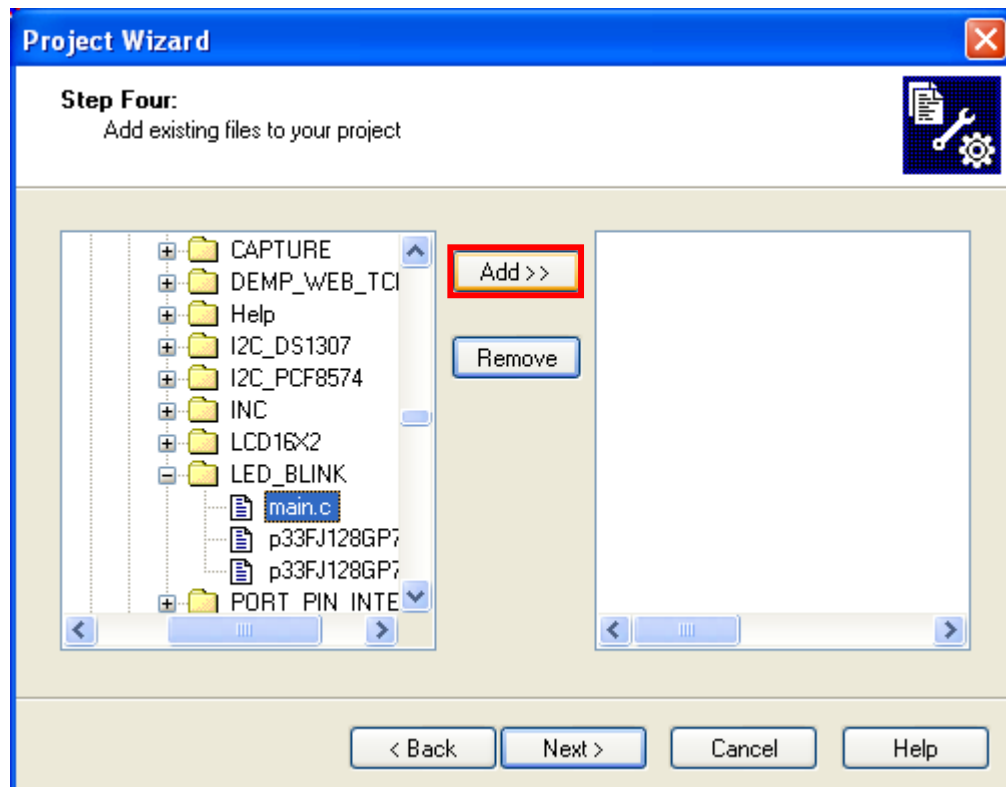


In this step, user must include files together under Project name of **LED_BLINK**; user must include 3 files into the Project as follows;

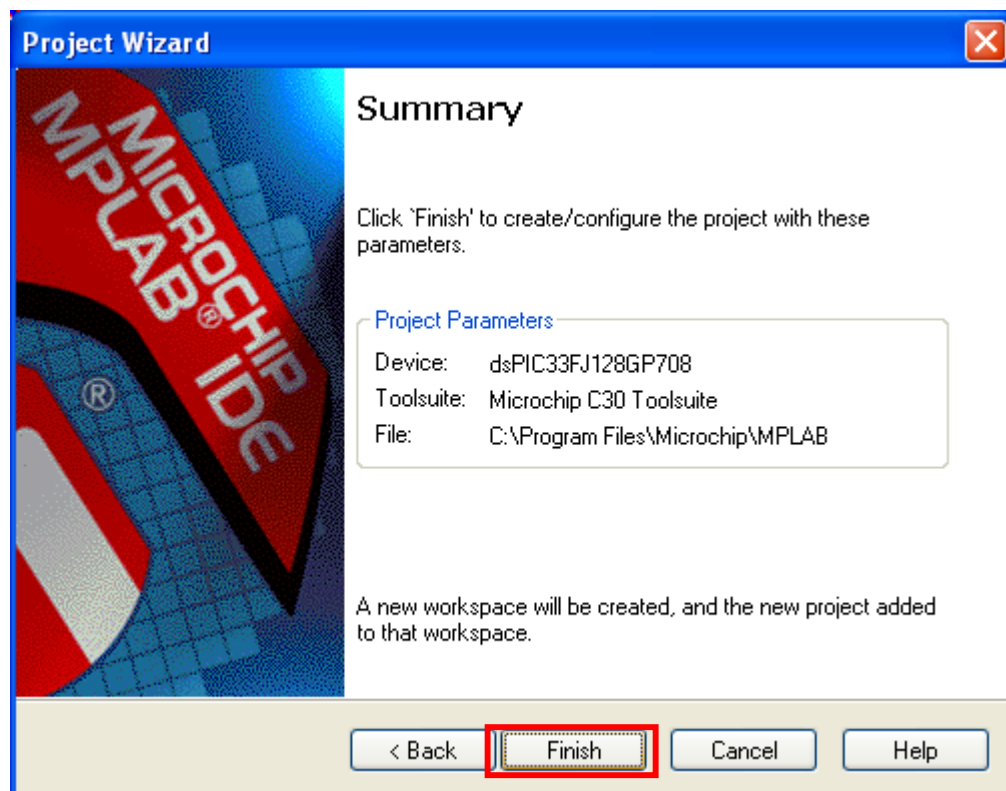
- Include filename **"main.c"** that is the written Source Code and it has already saved in the previous step in **...\LED_BLINK\main.c** completely.
- Include filename **"p33FJ128GP708.h"** that is Header of C Language to refer to the registers of MCU No. "dsPIC33FJ128GP708". If user installs Program C30 follows the Default values, this file will be stored in **"...\MPLAB C30\support\h\"**.
- Include filename **"p33FJ128GP708.gld"** that is Script File of dsPIC33FJ128GP708 that is created by MPLAB C30. If user installs Program follows the Default values, this file will be stored in **"...\MPLAB C30\support\gld\"**.

*****NOTE***** The standard Default value of installation Program Compiler C30; normally, it will be installed at **"C:\Program Files\Microchip\MPLAB C30"**.

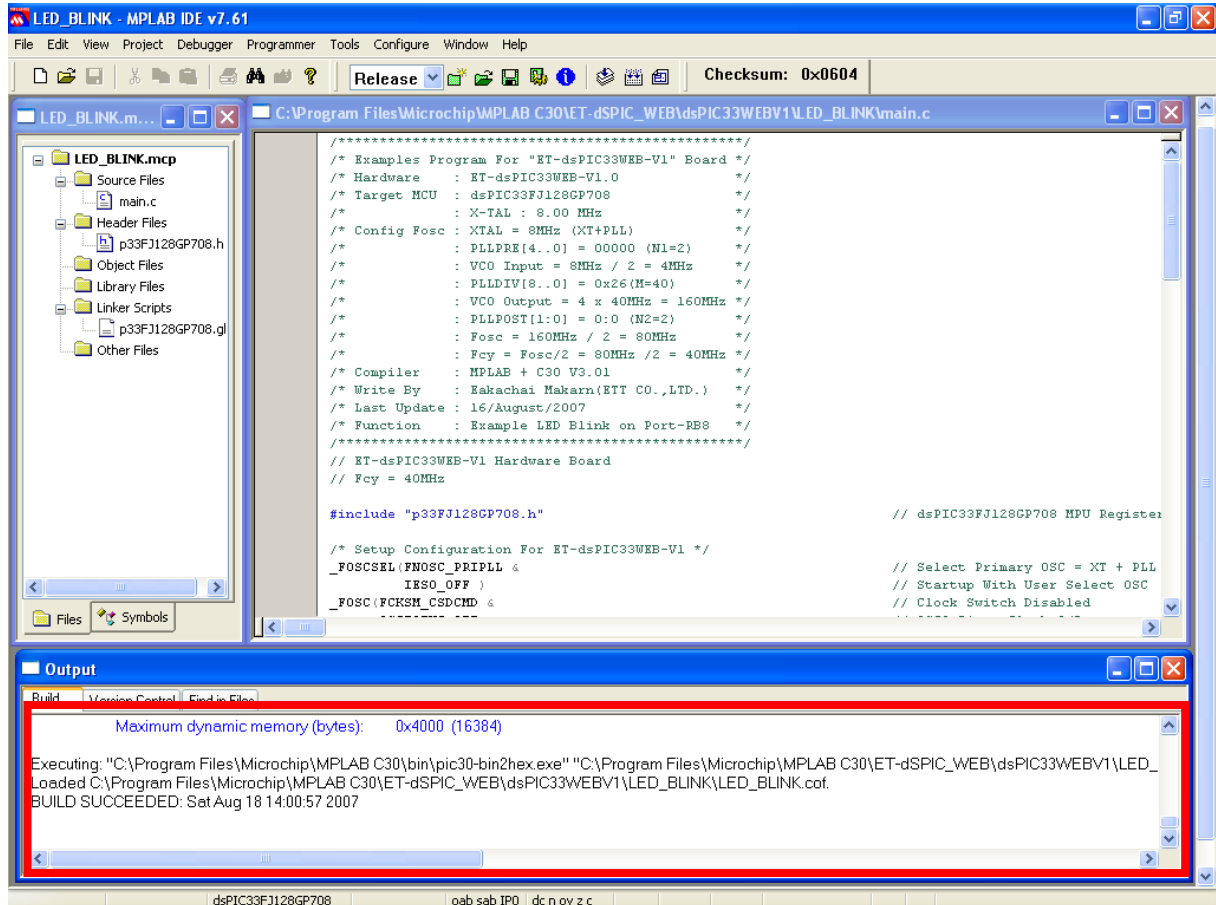
If user wants to include all 3 files together, user must click **"ICON"** of file from the Folder position that is mentioned above. Next, click **"Add>>"** until the desired filename is displayed in the right frame of program. User must follow these instructions until user can command to **"Add"** all 3 files successfully and then click **"Next>"** to go the next step as shown in the picture below.



When user has already included all files into the created Project File completely, program will report result by displaying parameter values as shown in the picture below; user must select **"Finish"** to end the procedure of creating Project File of **BLINK_LED**.



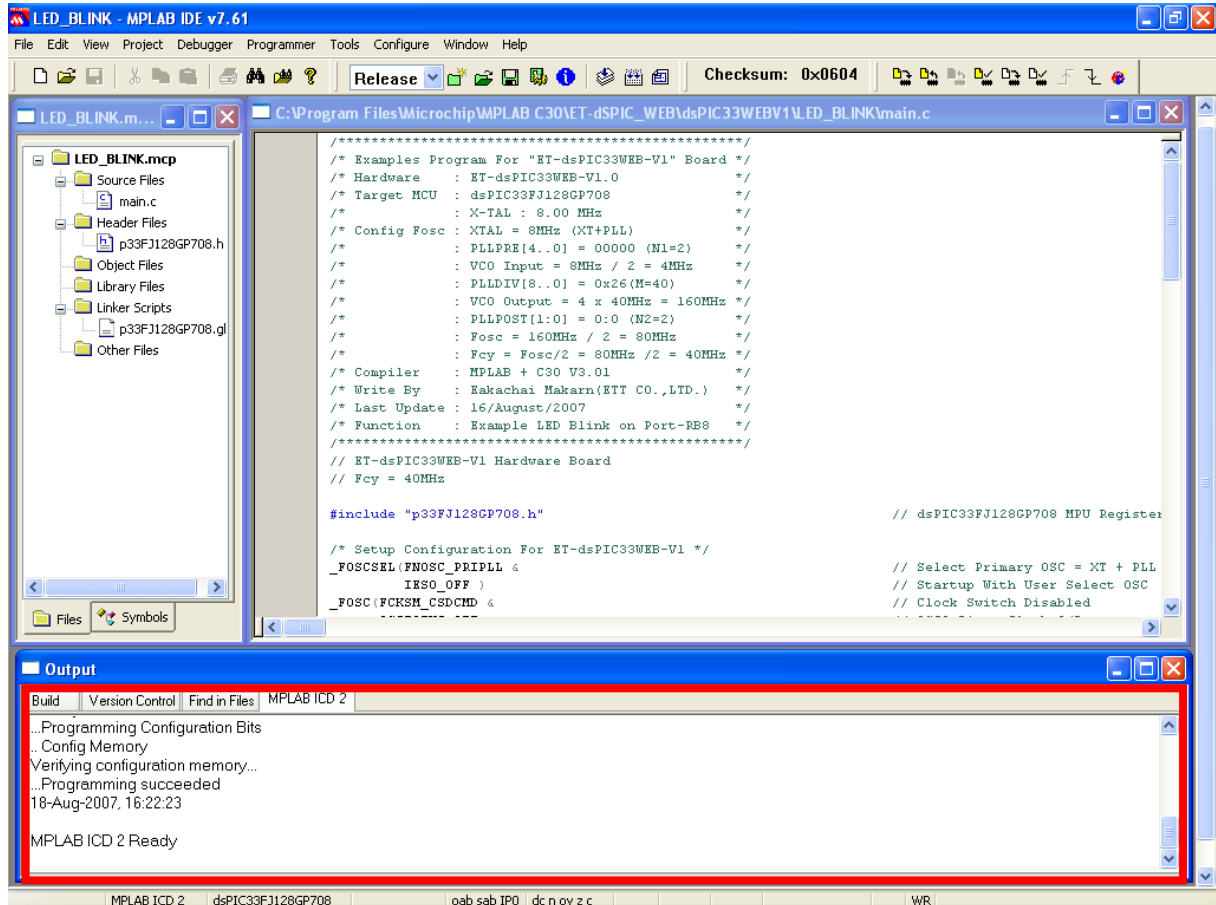
After user has already set values into Project File completely, user can compile command of Program **"main.c"** instantly by using Command **"Project → Build All"**. User will get name of Output file the same as the created Project File but its surname is **HEX**. From the example project, when it has already compiled command successfully without any error, user will get Output filename **"LED_BLINK.HEX"**. This file will be created and stored in Folder position of Project **"..\led_blink\led_blink.hex"**. User can download Hex File name **"LED_BLINK.HEX"** into Board **"ET-dsPIC33WEB V1.0"** to test operation instantly.



Download Code into Board

After user has already written and compiled program successfully and user gets HEX completely, it only remains the step of Download Code into MCU; in this case, there is device to Download Code that maybe ICD2 or Pickit2.

If using ICD2, user can download Code from Program MPLAB directly; interfacing cable from "ICD2" with Board "ET-dsPIC33WEB V1.0", shifting Switch MODE to PGM position (Switch position is pressed, red LED PGM is on), and then set the connection with ICD2 by clicking Menu **"Programmer → Select Program → MPLAB ICD 2"**. Next, user can command to Download Code by clicking Menu **"Programmer → Program"**; in this case, program will download Code automatically. When it is complete, it will display message **"MPLAB ICD2 Ready"** as shown in the picture below.



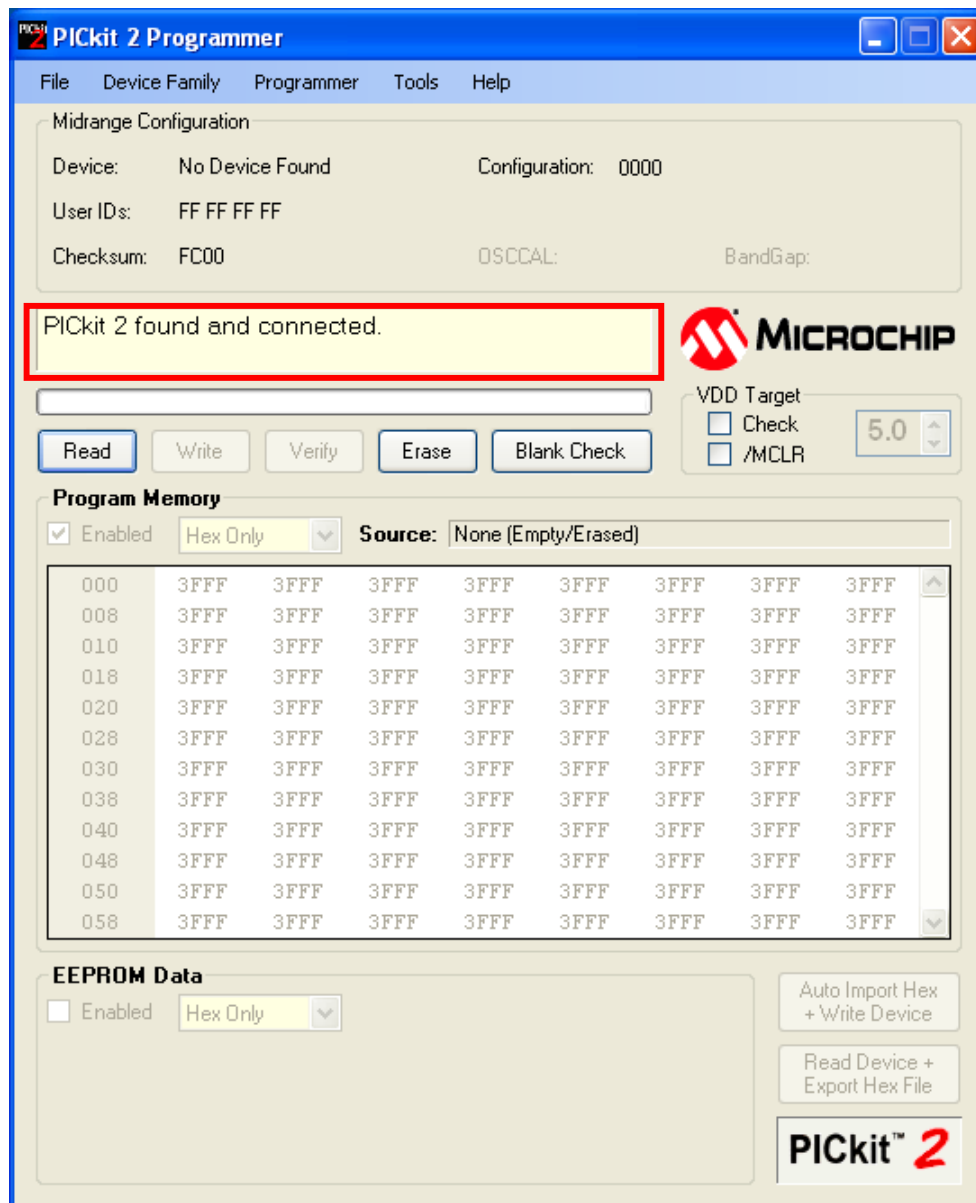
After user has already download Code successfully, shift Switch MODE to RUN position; in this case, position of Switch Mode is not pressed and green LED RUN is on. Next, user must press Switch Reset only one time and board will start running follows the downloaded command instantly.

Download Code by "ET-PGMPIC USB"

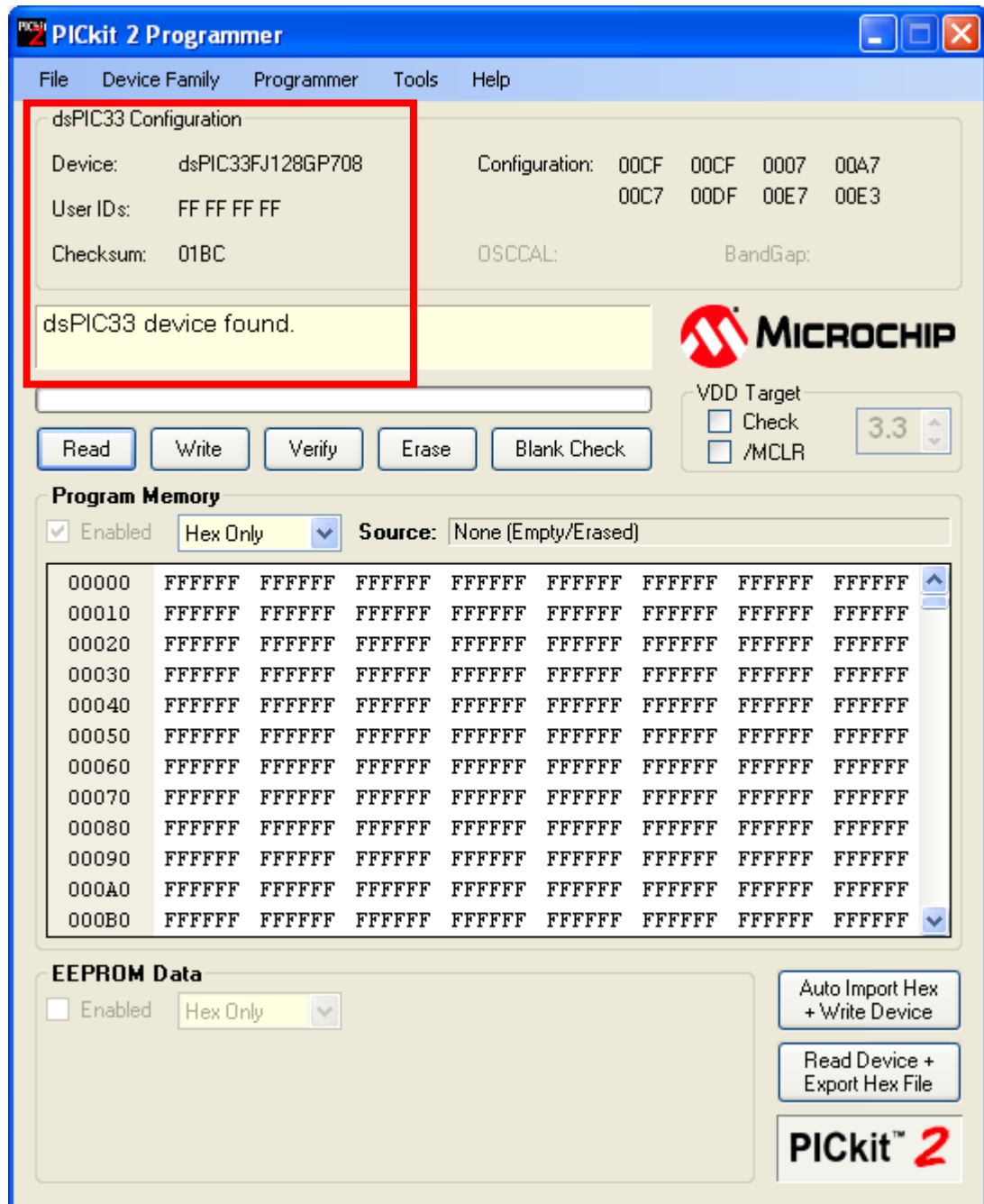
If using Pickit2 Programmer or ET-PGMPIC USB, user can not command to download Code through Program MPLAB the same as ICD2 because nowadays Program MPLAB does not support the connection between command and programmer, so user must use Program Pickit2 to download Code as follows;

1. Interface Cable ICD2 from ET-PGMPIC USB with Board ET-dsPIC33WEB V1.0 and set Jumper "B/T" of ET-PGMPIC USB to "B" position (Target Board) and then press Switch to select Mode of Board "ET-dsPIC33WEB V1.0" to PGM position (red LED PGM is on).

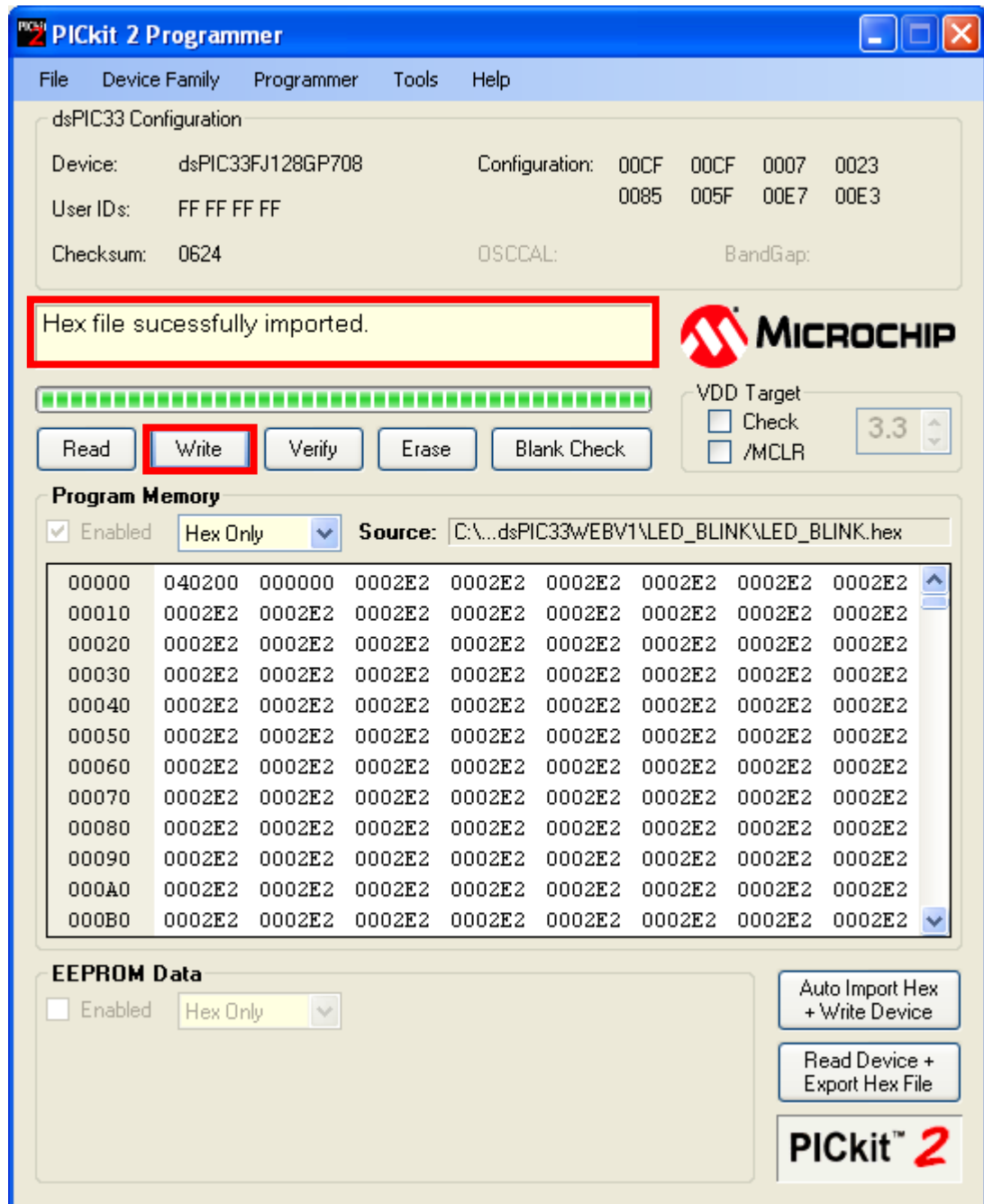
2. Run Program Pickit2, if everything is complete, it will show result as below;



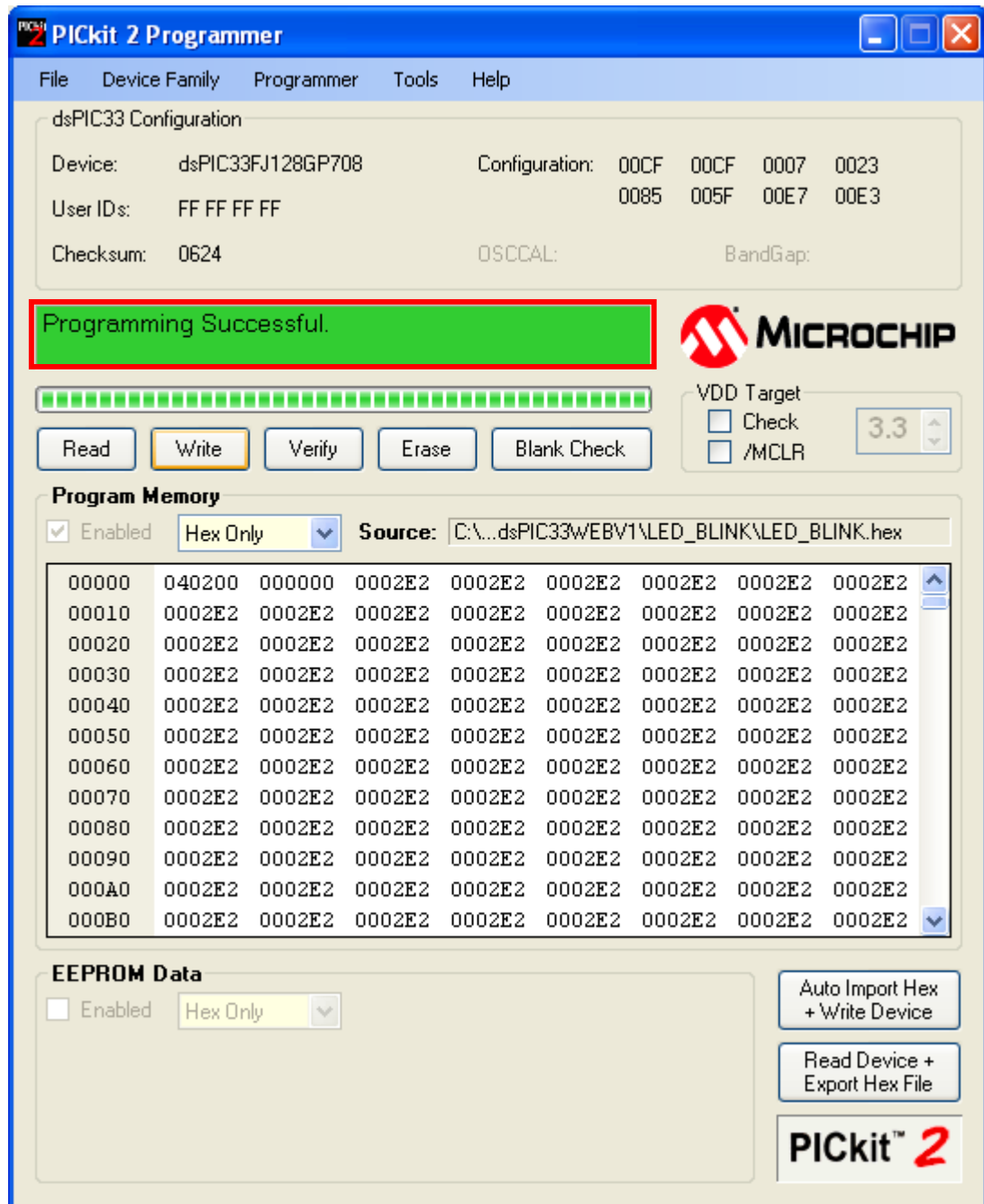
3. Click Menu **"Device Family → dsPIC33"**, it makes programmer connect with MCU and read Configuration values of MCU to display results on screen of program instantly. If using Board "ET-dsPIC33WEB V1.0", if everything is complete, it will display message **"dsPIC33 device found"** and MCU number (device) as **"dsPIC33FJ128GP708"** on the screen of program as shown in the picture below.



- Load Hex File that is compiled by C30, click Menu **"File → Import Hex"** and then select the compiled Hex File by C30. If everything is complete, it will display message **"Hex file successfully imported"**; it means that the method of Load Hex File is successfully. User can command to Download Code from Hex File into memory of MCU instantly, click Button **"Write"** as shown in the picture below.



5. "ET-PGMPIC USB" Programmer starts downloading Code into MCU instantly. When it has already downloaded Code successfully, it will display message **"Programming Successful"** as shown in the picture below.



6. After download Code completely, user must set operation of Switch Mode of Board ET-dsPIC33WEB V1.0 to RUN position. In this case, position of Switch Mode is not pressed and green LED RUN is on; and then user must press Switch Reset only one time, it makes Board start running follow the downloaded command instantly.