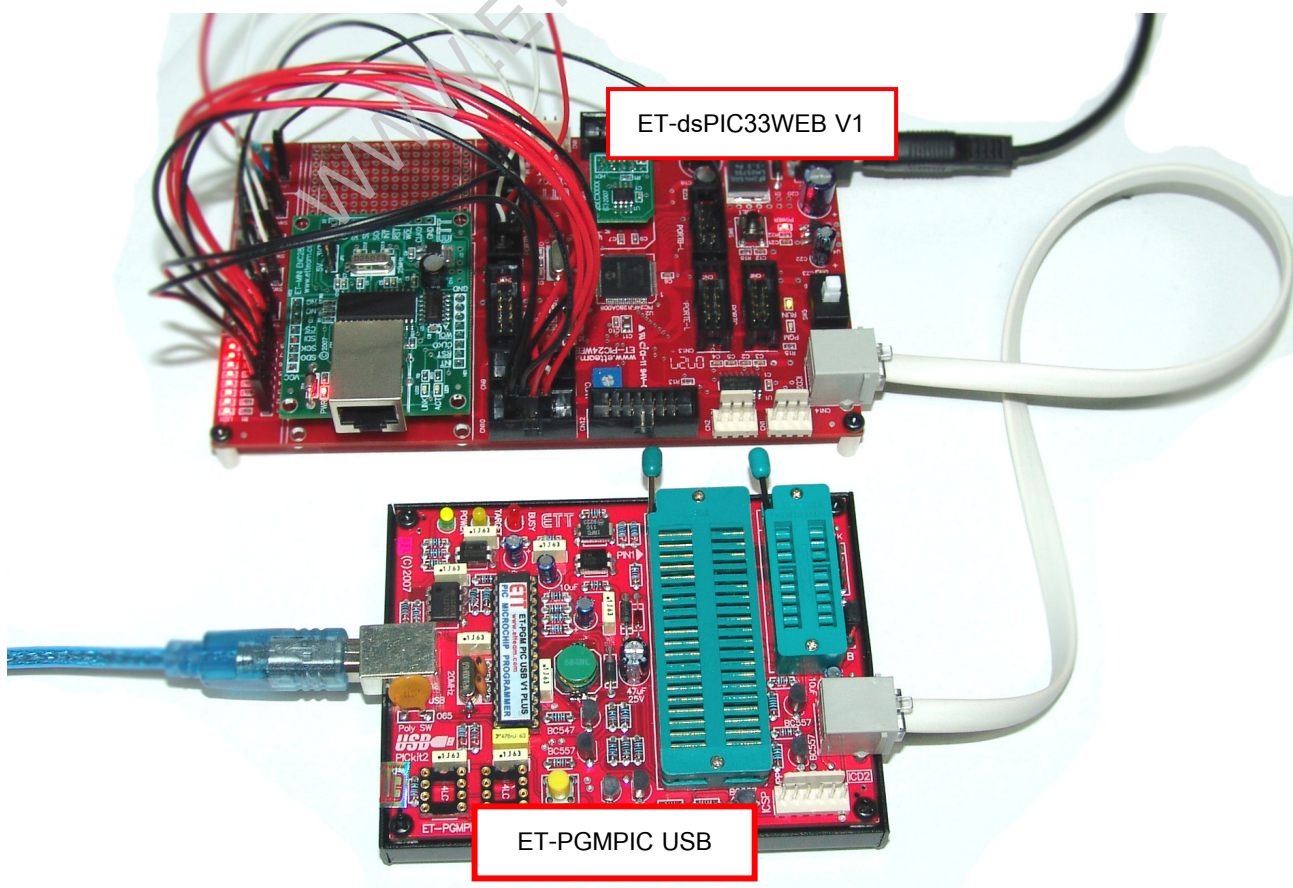


การพัฒนาโปรแกรมของบอร์ด

สำหรับการพัฒนาโปรแกรมของบอร์ด “ET-dsPIC33WEB V1.0” นั้น จะแบ่งออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของการพัฒนาโปรแกรม และการ Download โปรแกรม โดยในส่วนของการพัฒนาโปรแกรมของบอร์ดนั้น จะเป็นการเขียนโปรแกรมเพื่อควบคุมและสั่งงานให้ MCU สามารถทำงานตามจุดประสงค์ที่ได้ออกแบบไว้ ซึ่งในส่วนนี้เป็นหน้าที่ของผู้พัฒนาโปรแกรม ที่จะเลือกใช้ภาษาใดในการพัฒนาโปรแกรม รวมไปถึงการเลือกใช้โปรแกรมสำหรับทำหน้าที่แปลคำสั่งของภาษานั้นๆ ให้เป็นรหัสคำสั่งในรูปแบบของ HEX File สำหรับใช้ Download ให้กับหน่วยความจำโปรแกรม (Flash Memory) ของ MCU เพื่อสั่งงานให้ MCU ปฏิบัติตามคำสั่งในโปรแกรมที่ผู้พัฒนาโปรแกรมเขียนขึ้นมา

สำหรับส่วนของการ Download โปรแกรม หรือการ Download HEX File ที่ได้จากการพัฒนาโปรแกรมให้กับหน่วยความจำโปรแกรม (Flash Memory) ของ MCU นั้น จะต้องอาศัยเครื่องมือทางฮาร์ดแวร์เป็นส่วนประกอบในการพัฒนาโปรแกรมด้วย ซึ่งในกรณีของบอร์ด “ET-dsPIC33WEB V1.0” นั้น จะต้องใช้ เครื่องมือสำหรับโปรแกรมข้อมูลให้กับ MCU ภายในบอร์ดด้วยวิธีการแบบ In-Circuit โดยต้องใช้เครื่อง Programmer หรือใช้ชุดพัฒนาโปรแกรมจำพวก In-Circuit Debugger ที่สนับสนุนการใช้งานร่วมกับ MCU เบอร์ dsPIC33FJ128GP708 ซึ่งใช้ข้อต่อสัญญาณตรงตามมาตรฐาน “ICD2” ของ Microchips เช่น ICD2, Pickit2 ซึ่งทางบริษัท อีทีที ก็มีสนับสนุน คือ “ET-PGMPIC USB” ดังตัวอย่าง



การเขียนโปรแกรมใช้งานกับบอร์ดโดยใช้ MPLAB C30

MPLAB C30 หรือ C30 Tools เป็นโปรแกรมภาษาซี สำหรับใช้แปลคำสั่งของ MCU ตระกูล dsPIC ซึ่งได้รับการพัฒนาขึ้นโดย Microchips เอง โดยข้อกำหนดและรายละเอียดของการเขียนโปรแกรมภาษาซี นั้น จะไม่กล่าวถึงในที่นี้ด้วย โดยถ้าผู้ใช้งานต้องการพัฒนาโปรแกรมให้กับ dsPIC ด้วยภาษาซี แต่ยังไม่มีความรู้เรื่องการเขียนโปรแกรมภาษาซีเลยนั้นขอแนะนำให้ หาหนังสือที่อธิบายเกี่ยวกับการเขียนโปรแกรมภาษาซีในส่วนที่เป็นมาตรฐานตามข้อกำหนดของ “ANSI C” มาศึกษาให้เข้าใจเสียก่อน และสำหรับส่วนของข้อกำหนดปลีกย่อยอื่น ๆ ที่เป็นของ MPLAB C30 เองก็สามารถอ่านเพิ่มเติมได้จากเอกสารและคู่มือการใช้งานของ MPLAB C30 ที่ทาง Microchips จัดทำไว้ได้ โดยสามารถ Download จาก Website ของ Microchips หรือจาก Folder ของ “C:\Program Files\Microchip\MPLAB C30\docs” ที่ทำการติดตั้งโปรแกรม MPLAB C30 ไว้ก็ได้ โดยในที่นี้ จะขอกล่าวแนะนำถึงเฉพาะส่วนของการกำหนดค่าตัวเลือกในโปรแกรมเพื่อใช้งานร่วมกับบอร์ด “ET-dsPIC33WEB V1.0” เท่านั้น โดยในการที่จะใช้งานโปรแกรม MPLAB C30 ในการเขียนโปรแกรมนั้น ผู้ใช้จำเป็นต้องทำการติดตั้งโปรแกรมของ Microchips จำนวน 2 โปรแกรมดังนี้คือ

- MPLAB IDE ซึ่งเป็นโปรแกรม Text Editor ของ Microchips ซึ่งในปัจจุบัน (ตุลาคม 2548) จะเป็นรุ่น 7.21 แล้วสามารถ Download มาใช้งานได้ฟรีจาก Web ของ Microchips
- MPLAB C30 ซึ่งเป็นตัวแปลภาษาซี (C Compiler) ให้เป็นรหัสคำสั่งของ dsPIC ซึ่งตามปกติแล้วโปรแกรมชุดนี้จะต้องซื้อมาใช้งานเอง แต่อย่างไรก็ตามทาง Microchips เองมีรุ่นทดลองใช้งานให้ผู้ใช้งานสามารถ Download มาใช้งานได้เช่นเดียวกันกับ MPLAB IDE

โดยโปรแกรมทั้ง 2 ชุดนี้ ทางอีทีที ได้ทำการ Download มาจัดเตรียมไว้ให้ในแผ่น CD-ROM ที่แถมไปกับบอร์ดของ “ET-dsPIC33WEB V1.0” ด้วยอยู่แล้ว โดยในการติดตั้งโปรแกรมนั้นขอแนะนำให้ผู้ใช้ทำการติดตั้งโปรแกรมในชุดของ MPLAB IDE ก่อนเป็นอันดับแรก โดยขอแนะนำให้ติดตั้งโปรแกรมของ MPLAB IDE ไว้ตามค่า Default ของโปรแกรมติดตั้งเลย คือ “C:\Program Files\Microchip\MPLAB IDE\” จะสะดวกต่อการใช้งานมากกว่า ซึ่งหลังจากทำการติดตั้งโปรแกรม MPLAB IDE เสร็จเรียบร้อยแล้วในครั้งแรกก่อนการใช้นั้นต้องสั่ง Restart เครื่องคอมพิวเตอร์ก่อน หลังจากนั้นแล้ว MPLAB IDE จึงจะสามารถทำงานได้โดยไม่เกิดปัญหา จากนั้นจึงทำการติดตั้งโปรแกรม MPLAB C30 เป็นลำดับถัดไป โดยขอแนะนำให้ทำการติดตั้งโปรแกรมชุดนี้ไว้ตามค่า Default ของการติดตั้งโปรแกรม จะเกิดความสะดวกต่อการใช้งานมากที่สุด โดยเฉพาะในขั้นตอนของการกำหนดการเชื่อมโยงการทำงานระหว่าง MPLAB IDE และ MPLAB C30 โดยในที่นี้จะขออธิบายโดยอ้างถึงตำแหน่งการติดตั้งโปรแกรมหดที่กล่าวไว้แล้วข้างต้นเท่านั้น ซึ่งถ้าผู้ใช้ทำการสั่งติดตั้งโปรแกรมไว้ยังตำแหน่ง Folder ที่แตกต่างไปจากนี้แล้วขอให้ทำความเข้าใจและดัดแปลงวิธีการกำหนดค่าเองตามที่ติดตั้งโปรแกรมไว้จริงๆด้วย

ตัวอย่างการสร้างโปรแกรมภาษาซีของ MPLAB C30

เมื่อทำการติดตั้งโปรแกรมกำหนดการเชื่อมโยงคำสั่งระหว่าง MPLAB IDE และ MPLAB C30 เป็นที่เรียบร้อยแล้ว ต่อจากนี้ไป ผู้ใช้ก็สามารถทำการเรียกใช้งานโปรแกรม MPLAB C30 ผ่านทางโปรแกรม MPLAB IDE ได้แล้ว โดยค่าตัวเลือกต่างๆที่ได้กำหนดไว้แล้วนั้นจะถูกเก็บไว้ใน Configuration ของโปรแกรมตลอดไป จนกว่าจะมีการเปลี่ยนแปลงใหม่ ซึ่งในที่นี้จะขอยกตัวอย่างการเขียนโปรแกรมภาษาซี ของ MPLAB C30 ดัง 1 ตัวอย่างพอเป็นแนวทางให้ผู้ใช้งานทำความเข้าใจ โดยมีลำดับขั้นตอนดังนี้

1. สั่ง Run Program ของ MPLAB IDE ขึ้นมา จากนั้นก็สร้างไฟล์ใหม่ขึ้นมา 1 ไฟล์ สำหรับเขียน Source Code ภาษาซี โดยใช้คำสั่ง “File → New” จากนั้นให้ทำการพิมพ์ Source Code ภาษาซี ใน Work Sheet ของโปรแกรม MPLAB IDE ดังตัวอย่าง

```

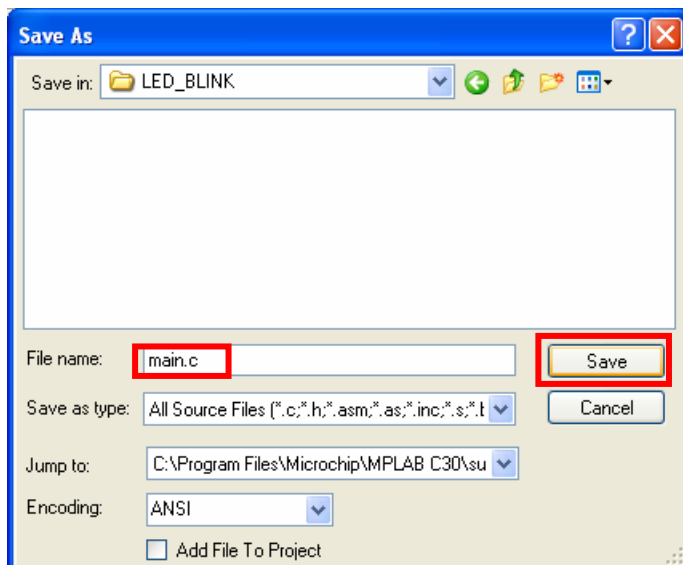
*****
/* Examples Program For "ET-dsPIC33WEB-V1" Board */
/* Hardware      : ET-dsPIC33WEB-V1.0 */
/* Target MCU    : dsPIC33FJ128GP708 */
/*               : X-TAL : 8.00 MHz */
/* Config Fosc   : XTAL = 8MHz (XT+PLL) */
/*               : PLLPRE[4..0] = 00000 (N1=2) */
/*               : VCO Input = 8MHz / 2 = 4MHz */
/*               : PLLDIV[8..0] = 0x26 (M=40) */
/*               : VCO Output = 4 x 40MHz = 160MHz */
/*               : PLLPOST[1:0] = 0:0 (M2=2) */
/*               : Fosc = 160MHz / 2 = 80MHz */
/*               : Fcy = Fosc/2 = 80MHz / 2 = 40MHz */
/* Compiler      : MPLAB + C30 V3.01 */
/* Write By      : Eakachai Makarn(ETT CO.,LTD.) */
/* Last Update   : 16/August/2007 */
/* Function      : Example LED Blink on Port-RE8 */
*****
// ET-dsPIC33WEB-V1 Hardware Board
// Fcy = 40MHz

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

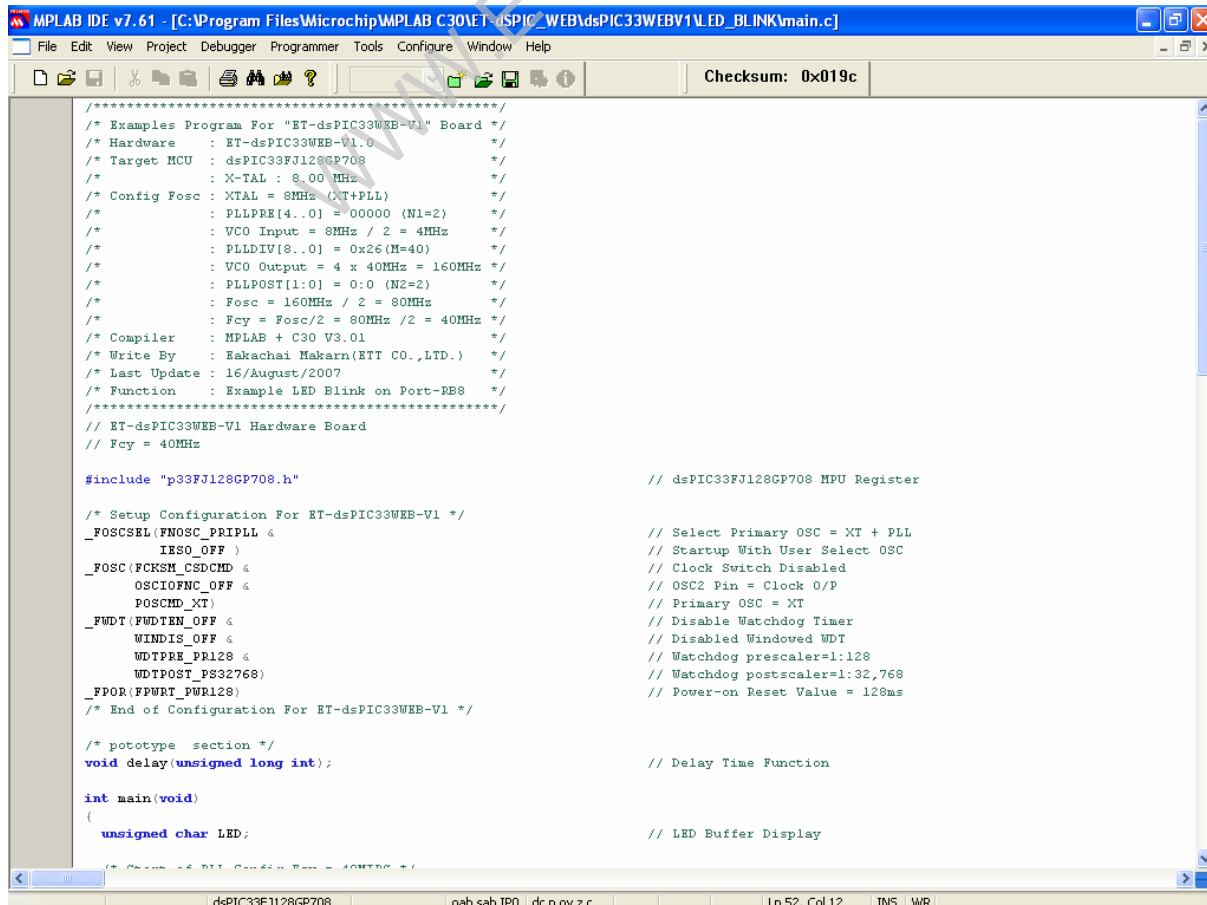
/* Setup Configuration For ET-dsPIC33WEB-V1 */
_FOSCSEL(FNOSC_PRIPLL &                // Select Primary OSC = XT + PLL
        IESO_OFF )                    // Startup With User Select OSC
_FOSC(FCKSM_CSDCMD &                  // Clock Switch Disabled
      OSCIOFNC_OFF &                  // OSC2 Pin = 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
      WDTPRST_PS32768)                // Watchdog postscaler=1:32,768
_FPOR(FPWRT_FWR128)                   // Power-on Reset Value = 128ms
/* End of Configuration For ET-dsPIC33WEB-V1 */

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

2. ทำการสั่งบันทึกไฟล์ที่เขียนขึ้นให้เป็น Text File ภาษาซี โดยให้ทดลองกำหนดชื่อเป็น “main.c” แล้วสั่งบันทึกไว้ใน Folder ชื่อ “...\\Microchip\\MPLAB C30\\examples\\led_blink\\main.c” โดยในที่นี้ให้ใช้คำสั่ง “File → Save As...” แล้วสร้าง Folder ชื่อ led_blink ไว้ภายใต้ Folder ของ examples อีกชั้นหนึ่ง แล้วกำหนดชื่อเป็น “main.c” แล้วเลือก “Save” ดังรูป



ซึ่งจะเห็นว่าเมื่อทำการบันทึกไฟล์เป็น “main.c” ไปแล้ว กลุ่มของตัวอักษรต่างๆที่ได้พิมพ์ไว้ จะถูกจัดแบ่งกลุ่ม โดยใช้สีในการแสดงผลที่แตกต่างกันไปตามหน้าที่ของกลุ่มตัวอักษร เช่น กลุ่มตัวอักษรที่ใช้เป็นคำอธิบาย (Comment) กลุ่มตัวอักษรที่เป็นคำสั่ง และกลุ่มตัวอักษรที่เป็นตัวแปรต่างๆ ซึ่งจุดนี้เป็นข้อดีของ MPLAB IDE ที่สามารถแยกการแสดงผลกลุ่มตัวอักษรตามหน้าที่การใช้งานได้ ทำให้เราสามารถอ่านโปรแกรมได้ง่ายและสะดวกมากยิ่งขึ้นดังรูป



```

#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
_CONFIG3(JTAGEN_OFF);   // Disable JTAG Interface
/* 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 &= 0xFFFE0; // 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
}

```

แสดง ตัวอย่าง Source Code สำหรับใช้ทดลองการทำงาน

สำหรับตัวอย่างนี้จะเป็นการสั่งให้ใช้พอร์ต RB8 ทำหน้าที่เป็น Output ขับ LED ให้ติดและดับสลับกันไปไม่รู้จักในลักษณะของไฟกระพริบ ซึ่งวิธีการทดสอบการทำงานของโปรแกรมนี้นี้ โดยใช้กับบอร์ด

ET-dsPIC33WEB V1.0 นั้นทำได้โดยต่อสัญญาณจาก RB8(PORTB-H) เข้ากับ LED1 Output ของบอร์ด โดยจะเห็นผลการทำงานของโปรแกรมแกรม คือ LED1 จะกะพริบ ติดและดับอย่างต่อเนื่องตลอดเวลา

โดยส่วนที่สำคัญและมีความจำเป็นมากที่สุดของโปรแกรมก็คือส่วนของคำสั่งที่ใช้ในการกำหนดค่าของ Configuration Bit ให้กับ MCU โดยค่านี้จะถูกแปลและใช้กำหนดค่าการทำงานของ Fuse Bit ในตัว MCU ในขั้นตอนของการ Download Code ให้กับ MCU โดยในตัวอย่างเป็นการกำหนดให้ MCU ทำงานด้วยสัญญาณนาฬิกา 8MHz จากภายนอก ร่วมกับวงจรคูณความถี่ (Phase Lock Loop :PLL) ซึ่งตัวอย่างโปรแกรมต่างๆที่ทางอีทีที สร้างขึ้น จะอ้างอิงจาก Configuration ดังต่อไปนี้

- กำหนดให้ MCU ทำงานจากสัญญาณนาฬิกา : XTAL = 8MHz (XT+PLL)
- กำหนดค่าการทำงานของวงจรคูณความถี่ Phase Lock Loop ดังนี้
 - 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

```

/* Setup Configuration For ET-dsPIC33WEB-V1 */
_FOSCSEL(FNOSC_PRIPLL & IESO_OFF ) // Select Primary OSC = XT + PLL
_FOSC(FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMD_XT) // Startup With User Select OSC
// Clock Switch Disabled
// OSC2 Pin = Clock O/P
// Primary OSC = XT
_FWDT(FWDTEN_OFF & WINDIS_OFF & WDTPRE_PR128 & WDTPOST_PS32768) // Disable Watchdog Timer
// Disabled Windowed WDT
// Watchdog prescaler=1:128
// Watchdog postscaler=1:32,768
_FPOR(FPWRT_PWR128) // Power-on Reset Value = 128ms
_CONFIG3(JTAGEN_OFF); // Disable JTAG Interface
/* End of Configuration For ET-dsPIC33WEB-V1 */

```

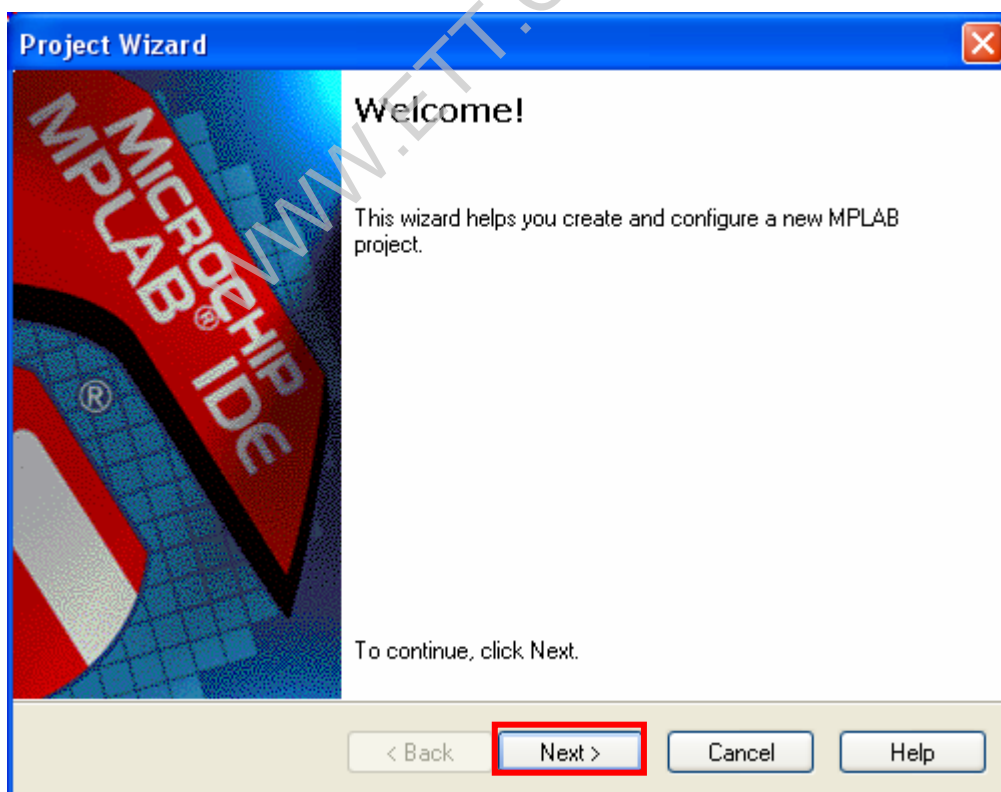
ส่วนของคำสั่งในการกำหนดค่าการทำงานของ Phase Lock Loop เพื่อทำการคูณความถี่จากสัญญาณนาฬิกา 8.00MHz ที่ต่อไว้ภายนอก เพื่อให้ได้ความถี่ Fosc เป็น 80 MHz เพื่อให้ CPU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 40MIPS (Fcy = 40MHz) ดังตัวอย่าง

```

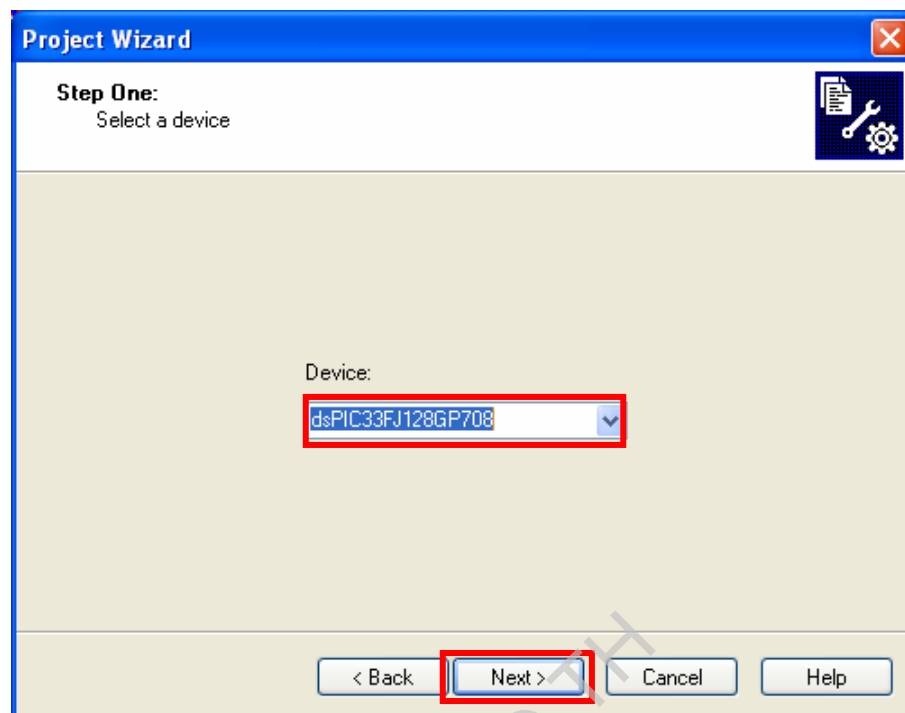
/* Start of PLL Config Fcy = 40MIPS */
CLKDIV &= 0xFFE0; // PLLPRE[4..0] = 00000 (N1=2)
CLKDIV &= 0xFF3F; // PLLPOST[1:0] = 00 (N2=2)
PLLFBD = 0x26; // PLLDIV[8..0] = 000100110 (M=40)
while(!OSCCONbits.LOCK); // Wait PLL to Lock
OSCCONbits.CLKLOCK = 1; // Disable PLL Modify
/* End of PLL Config Fly =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

```

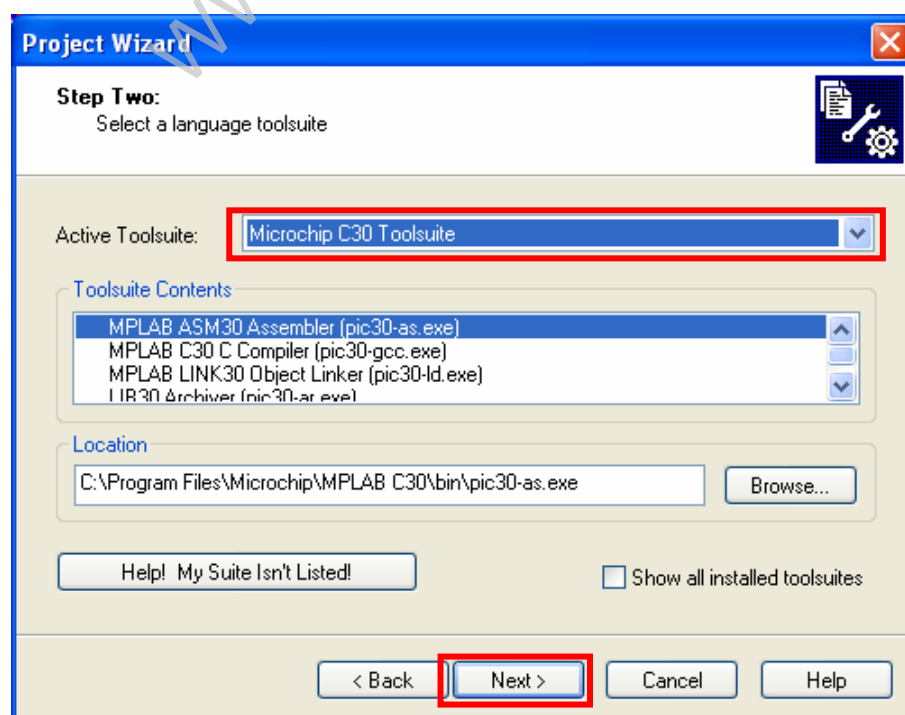
3. ทำการสร้าง Project File เพื่อใช้ส่งผนวกไฟล์ต่างๆ ที่เกี่ยวข้องเข้าด้วยกัน ทั้งนี้เนื่องมาจากว่า ภาษาซีของ MPLAB C30 นั้น ถูกออกแบบให้มีความอ่อนตัวในการทำงาน ดังนั้นจึงมีการจัดสร้าง และแบ่งแยกไฟล์ออกเป็นหลายๆไฟล์ตามหน้าที่การใช้งาน เพื่อให้ผู้ใช้สามารถเรียกไฟล์ต่างๆ เหล่านั้นเข้ามาใช้งานร่วมกับ Source Code ที่เขียนขึ้นมาได้ได้ง่าย โดยไม่ต้องเสียเวลาเขียน Source Code เองทั้งหมด ซึ่งจะทำให้ผู้ใช้สามารถลดเวลาในการเขียนโปรแกรมไปได้เป็นอย่างมากเนื่องจากเพียงแค่ทำการส่งผนวกไฟล์ที่ทาง MPLAB C30 สร้างเตรียมไว้ให้ เข้ากับ Source Code ที่ผู้ใช้เขียนขึ้นใหม่แล้วส่งแปลโปรแกรมก็จะได้ไฟล์ที่มีความสมบูรณ์ต่อการใช้งานแล้ว โดย จากตัวอย่าง Source Code ที่ได้ทดลองเขียนไปแล้วในข้างต้น ก็เช่นเดียวกัน จะเห็นได้ว่าการส่ง ผนวกไฟล์ชื่อ “p33FJ128GP708.h” เข้ามาใช้งานด้วย ซึ่งทำให้ไม่ต้องเสียเวลาไปส่งประกาศชื่อ และตำแหน่งรีจิสเตอร์ต่างๆของ “dsPIC33FJ128GP708” ให้เสียเวลา แต่สามารถอ้างถึงชื่อของ รีจิสเตอร์ต่างๆในโปรแกรมได้ทันที โดยวิธีการกำหนดคุณสมบัติของ Project File มีดังนี้
- สั่งกำหนดคุณสมบัติของ project File โดยใช้คำสั่ง “Project → Project Wizard...” ซึ่ง จะได้ผลดังรูป จากนั้นให้เลือก “Next >” เพื่อไปยังขั้นตอนต่อไป



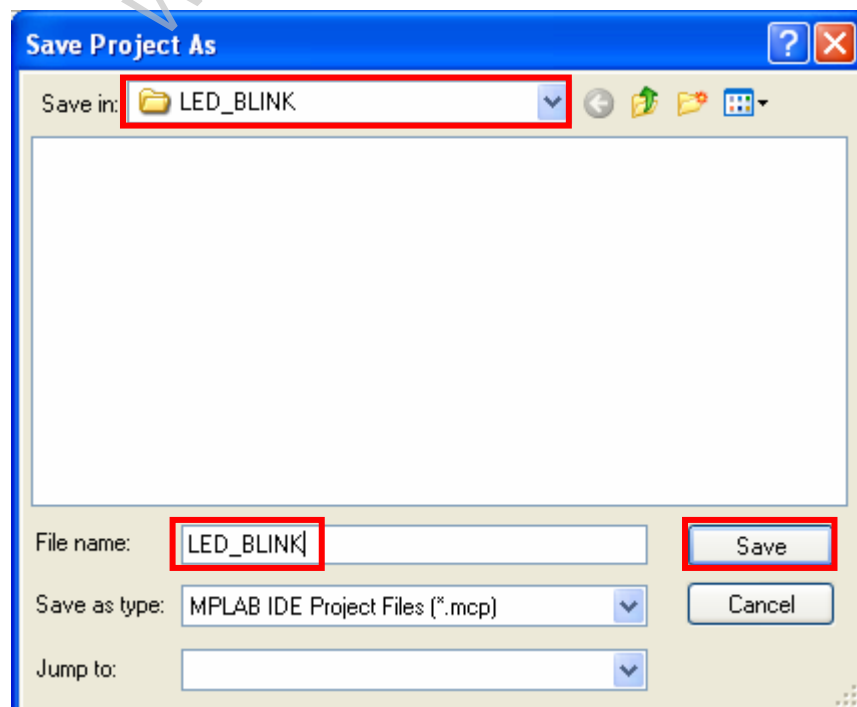
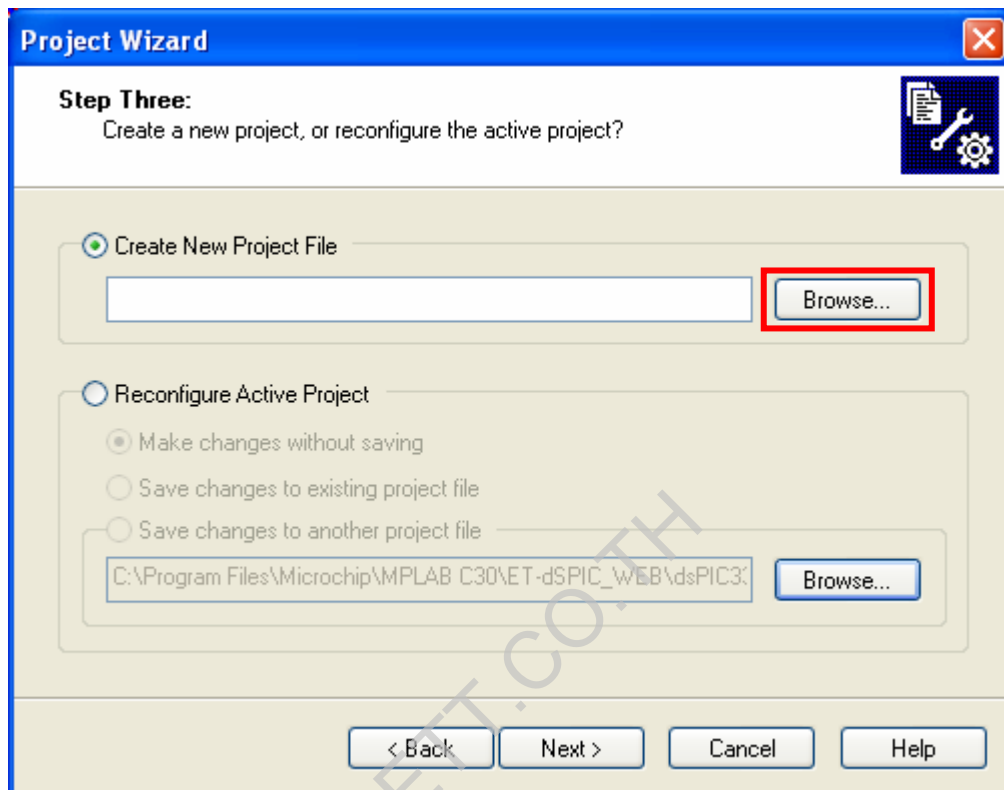
หลังจากเลือก “Next >” แล้ว โปรแกรมจะขอให้กำหนดเบอร์ของ MCU ที่จะใช้งานร่วมกับ โปรแกรมที่เขียนขึ้น ซึ่งให้เลือกกำหนดเป็น “dsPIC33FJ128GP708” จากนั้นเลือก “Next >” เพื่อข้ามไป ทำงานยังขั้นตอนต่อไปดังรูป



ในขั้นตอนนี้จะเป็นการเลือกที่จะใช้โปรแกรมชุดใดในการแปลคำสั่ง เนื่องจาก MPLAB IDE สามารถใช้งานได้กับชุดโปรแกรมต่างๆมากมายหลายโปรแกรม ซึ่งในที่นี้ให้เลือกกำหนดใช้โปรแกรม ของ MPLAB C30 โดยการเลือกกำหนดตัวเลือกของ “Active Toolsuite” ให้เป็นของ MPLAB C30 โดยกำหนด ตัวเลือกเป็น “Microchip C30 Toolsuite” ดังรูป แล้วเลือก “Next >”



ในขั้นตอนนี้จะเป็นการ กำหนดชื่อ Project และตำแหน่ง Folder ที่จะใช้เก็บไฟล์ต่างๆที่ได้จากการทำงานของ Project โดยให้กำหนดชื่อเป็น “LED_BLINK” ดังรูป แล้วเลือก “Next >”

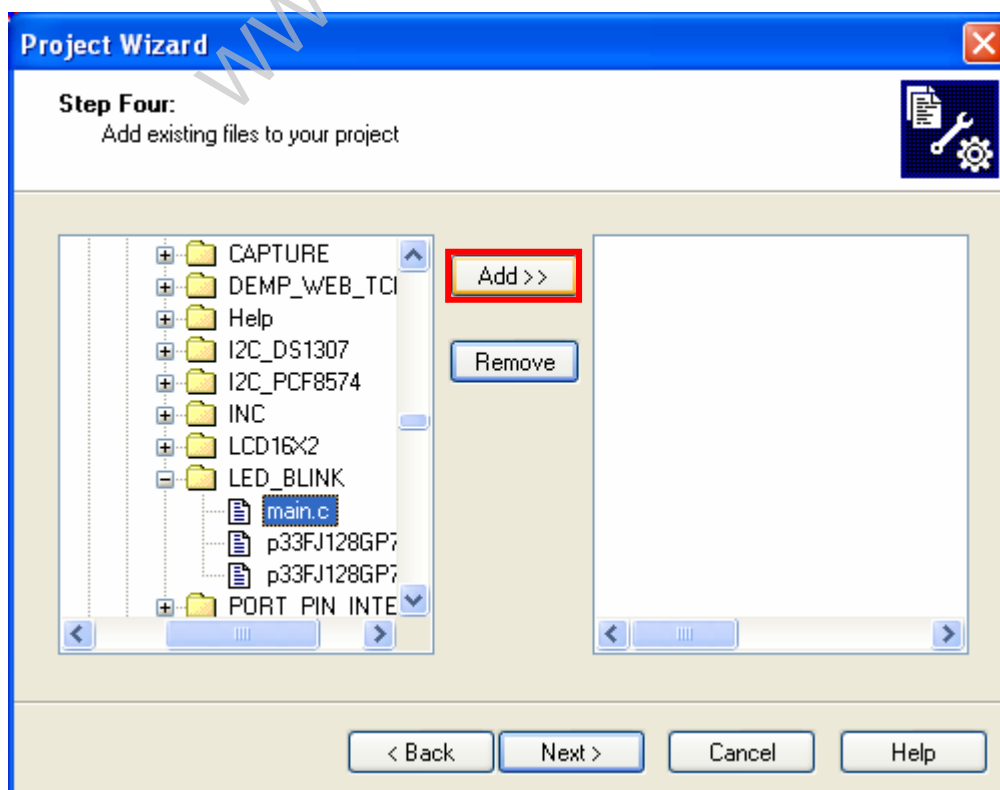


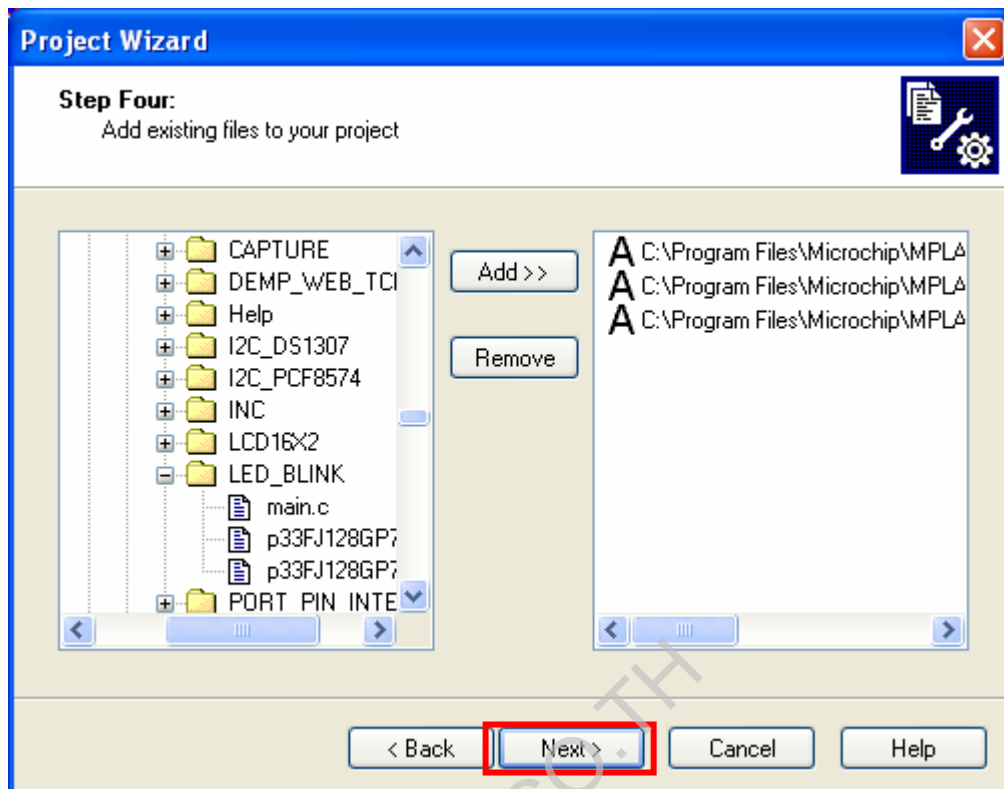
ในขั้นตอนนี้จะเป็นการส่งผนวกไฟล์ต่างๆเข้าไว้ด้วยกันภายใต้ชื่อ Project ของ LED_BLINK โดยให้ทำการส่งผนวกไฟล์ทั้งหมด 3 ไฟล์เข้าไว้ใน Project ดังนี้

- ส่งผนวกไฟล์ชื่อ “main.c” ซึ่งเป็น Source Code ที่เราได้เขียนและสั่งบันทึกไว้แล้วในขั้นตอนที่ผ่านมาโดยเก็บอยู่ใน ...\\LED_BLINK\\main.c”
- ส่งผนวกไฟล์ชื่อ “p33FJ128GP708.h” ซึ่งเป็น Header ของภาษาซี ในการอ้างถึงรีจิสเตอร์ต่างๆของ MCU เบอร์ “dsPIC33FJ128GP708” โดยถ้าติดตั้งโปรแกรม C30 ตามค่า Default ไฟล์ดังกล่าวจะเก็บอยู่ใน “...\\MPLAB C30\\support\\h\\”
- ส่งผนวกไฟล์ชื่อ “p33FJ128GP708.gld” ซึ่งเป็น Script File ของ dsPIC33FJ128GP708 ที่ทาง MPLAB C30 สร้างเตรียมไว้ให้ โดยถ้าติดตั้งโปรแกรมตาม C30 ตามค่า Default ไฟล์ดังกล่าวจะเก็บอยู่ใน “...\\MPLAB C30\\support\\gld\\”

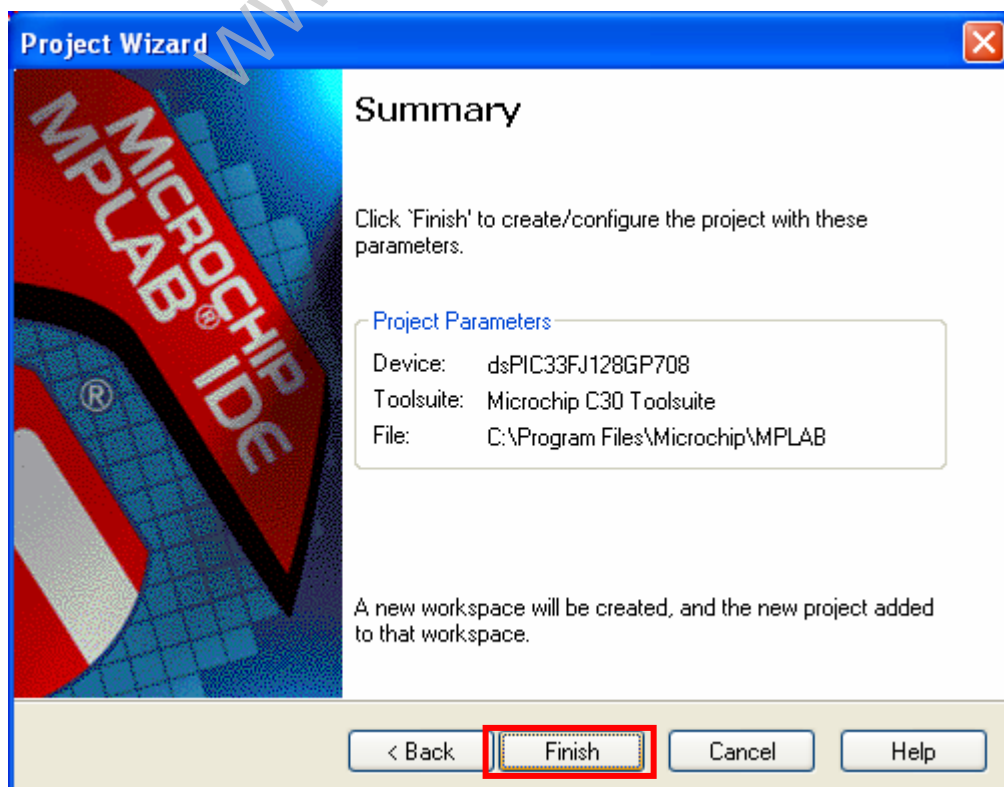
หมายเหตุ ค่า Default มาตรฐานของการติดตั้งโปรแกรมคอมไพเลอร์ C30 โดยปรกติแล้วโปรแกรมจะถูกติดตั้งไว้ที่ “C:\\Program Files\\Microchip\\MPLAB C30”

โดยในการส่งผนวกไฟล์ทั้ง 3 ดังกล่าวให้ทำการคลิกเมาส์ไปยัง “ICON” ของไฟล์จากตำแหน่ง Folder ที่กล่าวไว้ในข้างต้นที่ผ่านมาแล้วเลือก “Add >>” จนชื่อไฟล์ดังกล่าวไปปรากฏอยู่ที่กรอบหน้าต่างด้านขวาของโปรแกรม ซึ่งให้เลือกทำตามวิธีการนี้จนสามารถส่ง “Add” ไฟล์ได้ครบทั้ง 3 ไฟล์ แล้วเลือก “Next >” เพื่อไปยังขั้นตอนต่อไปดังรูป

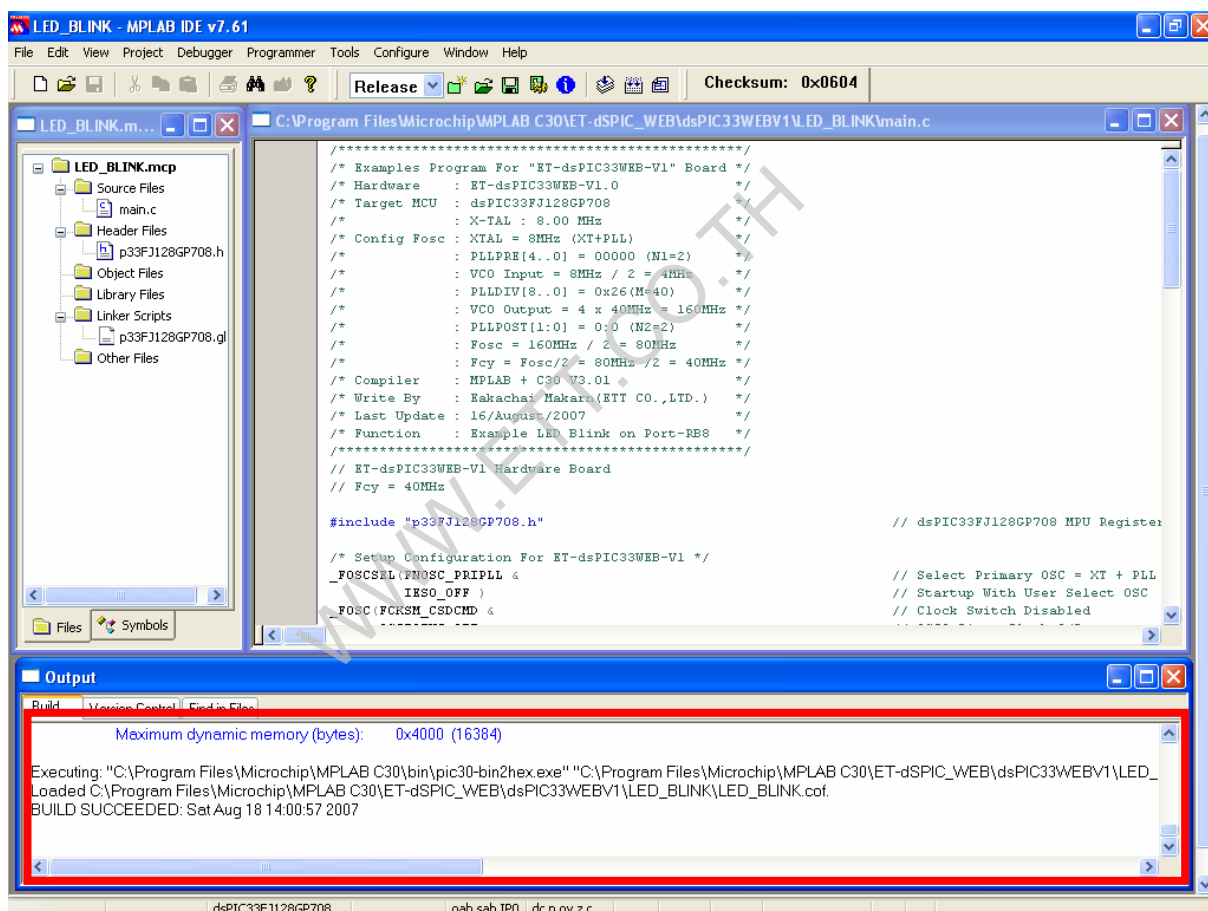




เมื่อสั่งผนวกไฟล์ทั้งหมดเข้ากับ Project ไฟล์ที่สร้างขึ้นเป็นที่เรียบร้อยแล้ว โปรแกรมจะรายงานผล โดยแสดงค่าพารามิเตอร์ต่างๆที่ได้กำหนดไว้แล้วให้ทราบดังรูป ให้เลือก “Finish” เป็นอันเสร็จขั้นตอนของการสร้าง Project File ของ BLINK_LED



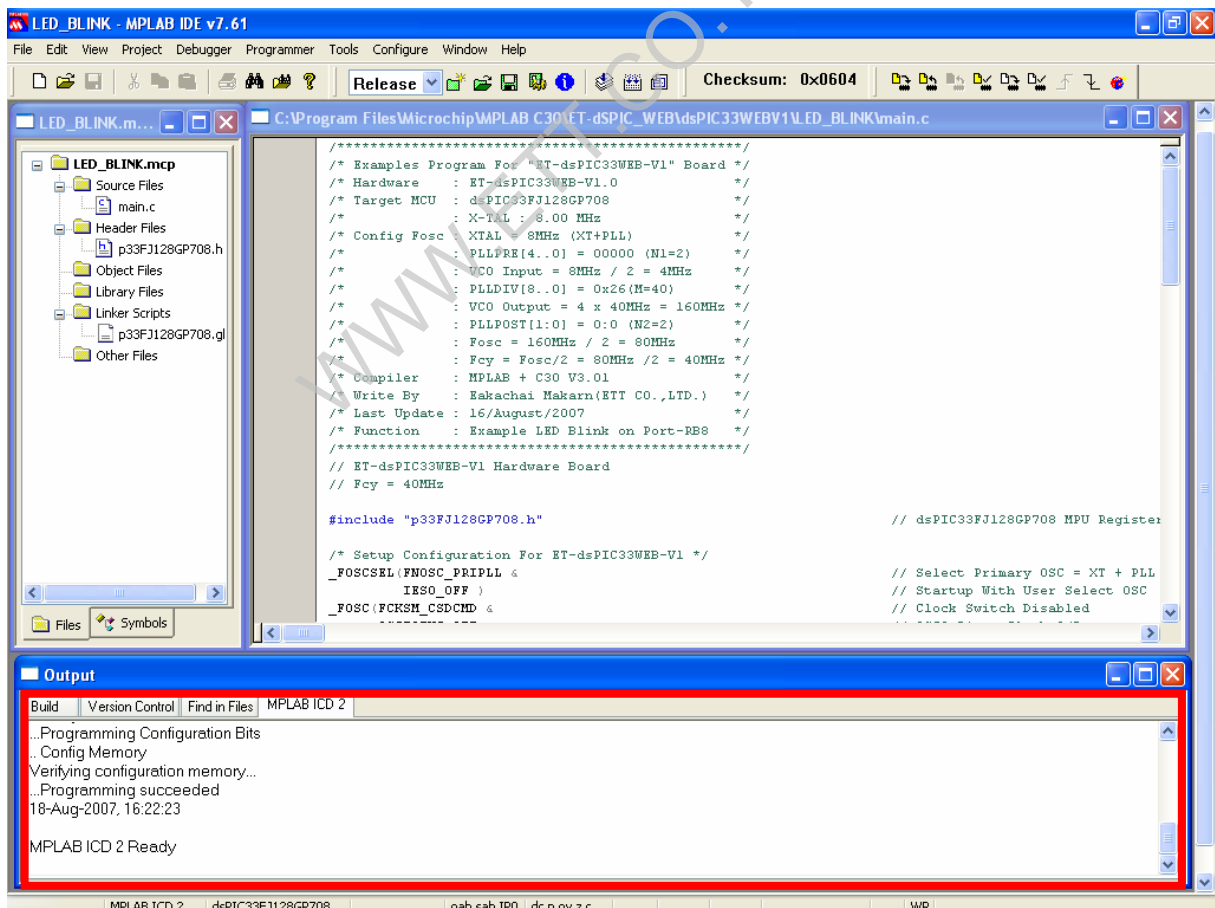
ซึ่งหลังจากกำหนดค่าต่างๆให้กับ Project File เป็นที่เรียบร้อยแล้ว ผู้ใช้ก็สามารถทำการสั่งแปลคำสั่งของโปรแกรม “main.c” ที่เขียนขึ้นได้ทันที โดยใช้คำสั่ง “Project → Build All” ซึ่งจะทำได้ไฟล์ Output มีชื่อเดียวกับ Project File ที่สร้างไว้แต่มีนามสกุลเป็น HEX ซึ่งจากตัวอย่าง Project นี้เมื่อสั่งแปลโปรแกรมแล้วถ้าไม่เกิดข้อผิดพลาดใดๆจะได้ Output ไฟล์ชื่อ “LED_BLINK.HEX” โดยไฟล์ดังกล่าวจะถูกสร้างและเก็บไว้ในตำแหน่ง Folder ของ Project คือ “...\led_blink\led_blink.hex” โดยผู้ใช้สามารถสั่ง Download Hex File ชื่อ “LED_BLINK.HEX” นี้ให้กับบอร์ด “ET-dsPIC33WEB V1.0” เพื่อทดสอบการทำงานได้ทันที



การ Download Code ให้กับบอร์ด

หลังจากทำการเขียนโปรแกรมและสั่งคอมไพล์จนได้ HEX เรียบร้อยแล้ว ขั้นตอนการพัฒนาโปรแกรมของบอร์ดก็จะเหลือเพียงการ Download Code ให้กับ MCU ซึ่งในขั้นตอนนี้จะต้องใช้เครื่องมือในการ Download Code ร่วมด้วย ซึ่งอาจเป็น ICD2 หรือ Pickit2 ก็ได้

สำหรับในกรณีที่ใช้ ICD2 นั้นจะสามารถสั่ง Download Code ได้จากโปรแกรม MPLAB โดยตรงเลย โดยให้ทำการเชื่อมต่อสายสัญญาณจาก "ICD2" เข้ากับบอร์ด "ET-dsPIC33WEB V1.0" แล้วเลือกสวิตช์ Mode ให้อยู่ในตำแหน่ง PGM (สวิตช์อยู่ในตำแหน่งถูกกด LED PGM สีแดงติดสว่าง) จากนั้นเลือกกำหนดการเชื่อมต่อกับ ICD2 โดยการคลิกเมาส์ที่เมนูคำสั่ง "Programmer → Select Program → MPLAB ICD 2" จากนั้นให้สั่ง Download Code โดยคลิกเมาส์ที่เมนูคำสั่ง "Programmer → Program" ซึ่งเครื่องก็จะทำการ Download Code ให้เองโดยอัตโนมัติ จนเมื่อเสร็จเรียบร้อยแล้ว จะมีข้อความแจ้งให้ทราบคือ "MPLAB ICD2 Ready" ดังรูป

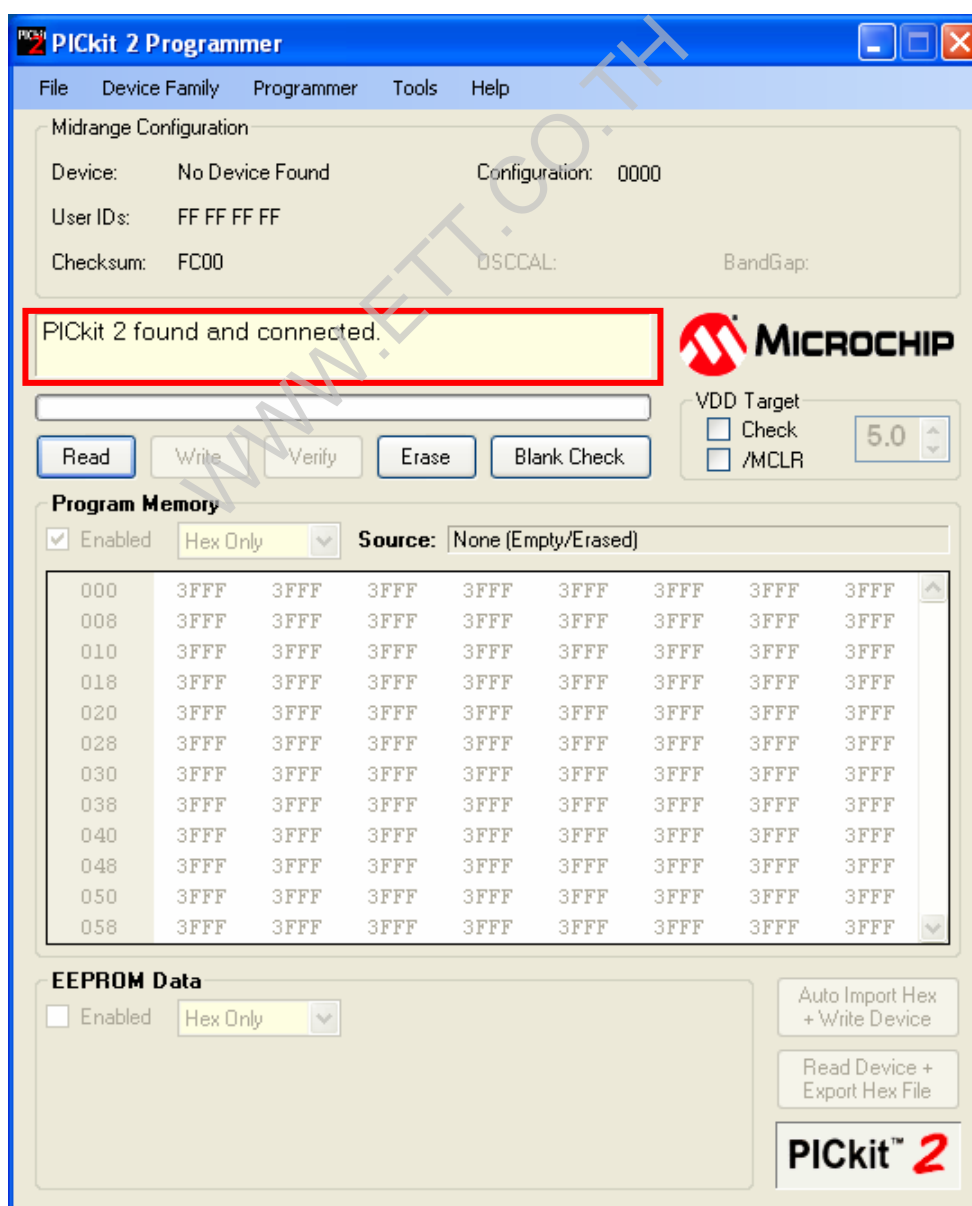


ซึ่งหลังจากทำการ Download Code เรียบร้อยแล้วให้เลือกสวิตช์ Mode ไปไว้ในตำแหน่ง Run โดยให้ตำแหน่งของสวิตช์ Mode อยู่ในตำแหน่งปล่อย และ LED Run สีเขียวติดสว่าง จากนั้นให้กดสวิตช์ Reset 1 ครั้ง บอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่ Download ให้แล้วในทันที

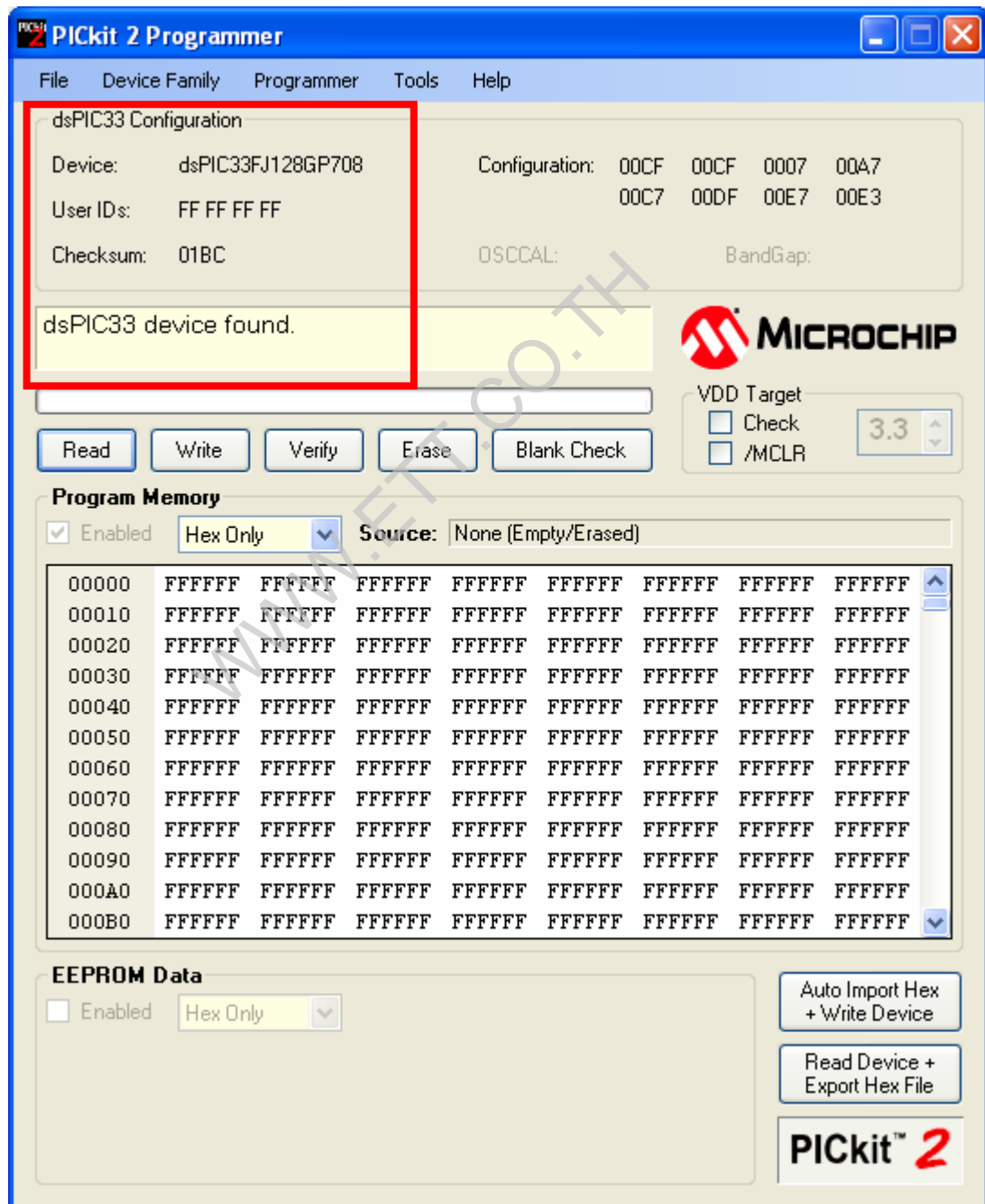
การ Download Code ด้วย “ET-PGMPIC USB”

สำหรับในกรณีที่ผู้ใช้เลือกใช้เครื่องโปรแกรม Pickit2 หรือ ET-PGMPIC USB นั้น ในปัจจุบัน โปรแกรม MPLAB ยังไม่ได้สร้างการเชื่อมต่อคำสั่งกับเครื่องโปรแกรมห้ ดังนั้นผู้ใช้จึงไม่สามารถสั่ง Download Code ผ่านทางโปรแกรม MPLAB ได้เหมือน ICD2 แต่จะต้องสลับไปให้โปรแกรม Pickit2 เพื่อสั่ง Download Code เองดังนี้

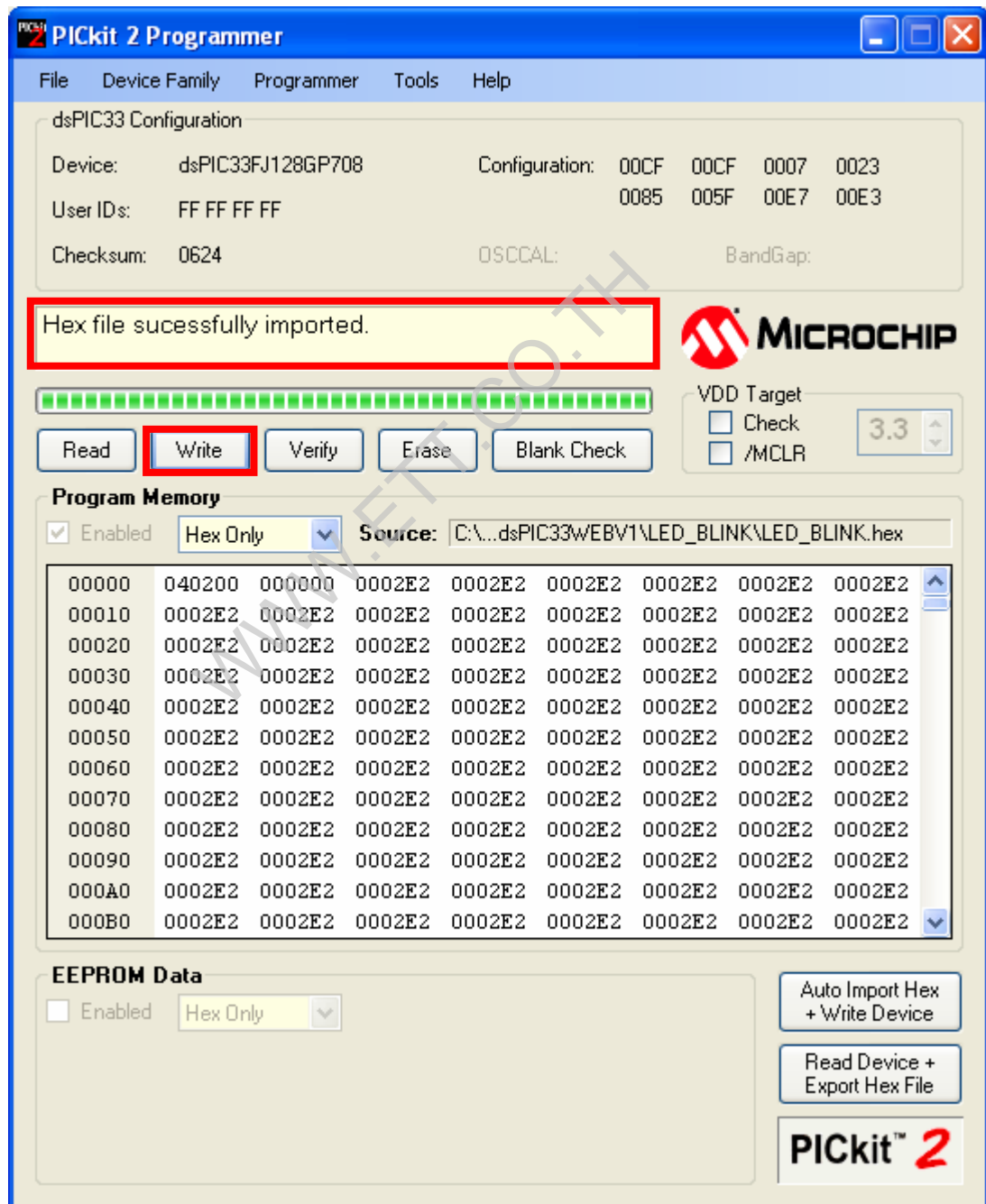
1. ต่อสายสัญญาณ ICD2 จาก ET-PGMPIC USB เข้ากับบอร์ด ET-dsPIC33WEB V1.0 โดยให้เลือกกำหนด Jumper “B/T” ของ ET-PGMPIC USB ไว้ในตำแหน่ง “B” (Target Board) พร้อมกับกำหนดสวิตช์เลือก Mode ของบอร์ด “ET-dsPIC33WEB V1.0” ไว้ในตำแหน่ง PGM (LED PGM สีแดงติดสว่าง)
2. สั่ง Run โปรแกรม Pickit2 ถ้าทุกอย่างถูกต้องจะได้ผลดังรูป



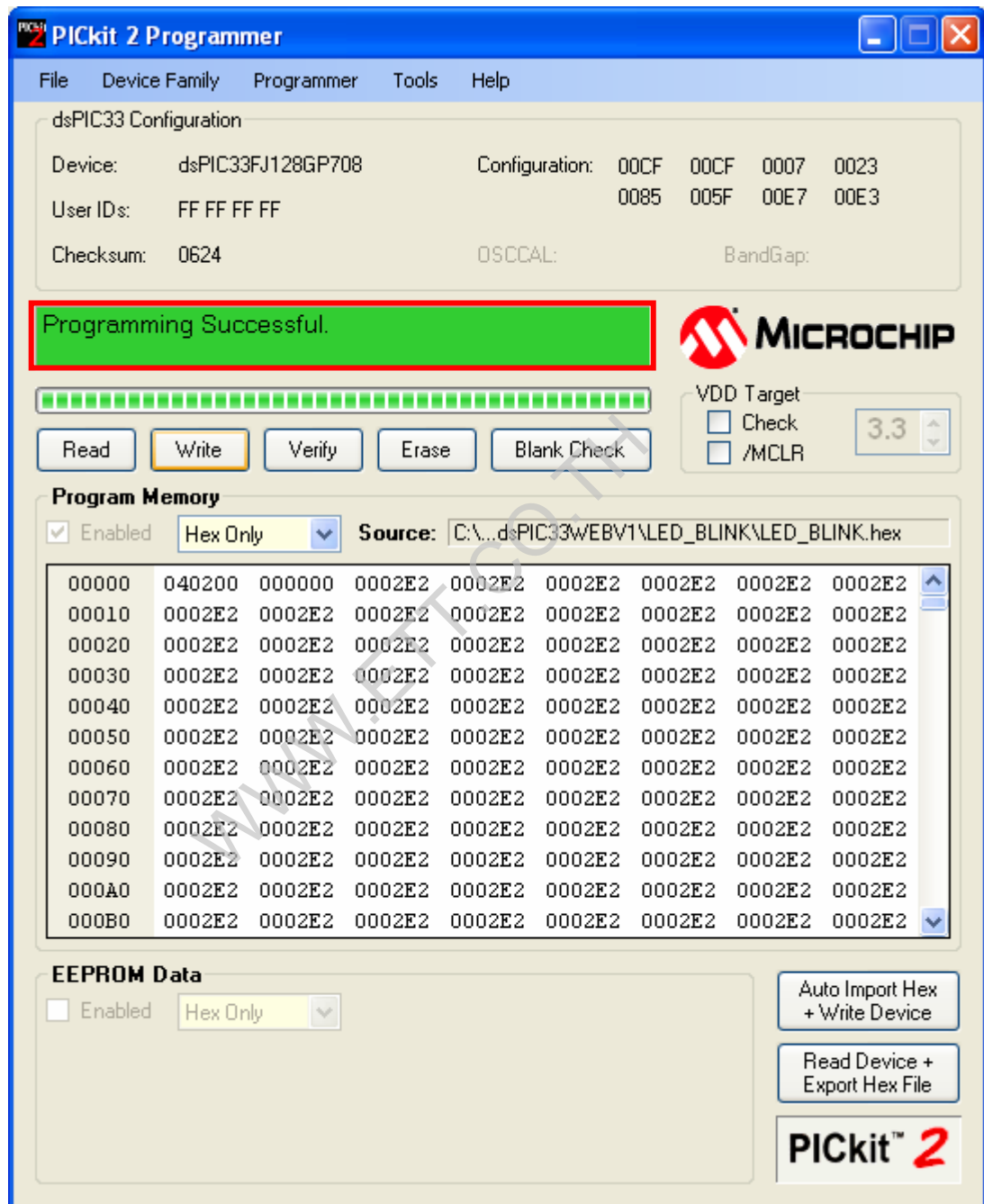
3. ทำการคลิกเมาส์ที่เมนูคำสั่ง “Device Family → dsPIC33” ซึ่งเครื่องโปรแกรมจะทำการเชื่อมต่อกับ MCU พร้อมกับอ่านค่า Configuration ต่างๆของ MCU ขึ้นมาแสดงผลให้เห็นที่หน้าจอของโปรแกรม ให้เห็นในทันที ซึ่งในกรณีของบอร์ด “ET-dsPIC33WEB V1.0” นั้น ถ้าทุกอย่างถูกต้องที่หน้าจอของโปรแกรมจะแสดงข้อความ “dsPIC33 device found” พร้อมกับมีการแสดงเบอร์ของ MCU (Device) เป็น “dsPIC33FJ128GP708” ดังรูป



4. ให้ทำการสั่ง Load HEX File ที่ได้จากการแปลโปรแกรมของ C30 โดยให้ทำการคลิกเมาส์ที่เมนูคำสั่ง “File → Import Hex” แล้วเลือก HEX File ที่ได้จากจากแปลคำสั่งของ C30 ซึ่งถ้าทุกอย่างถูกต้อง จะปรากฏข้อความ “Hex file successfully imported” ซึ่งหมายถึงการสั่ง Load Hex ไฟล์เรียบร้อยแล้ว ซึ่งผู้ใช้สามารถสั่ง Download Code จาก Hex File ให้กับหน่วยความจำของ MCU ได้ทันที โดยการคลิกเมาส์ที่ปุ่มคำสั่ง “Write” ดังรูป



5. เครื่อง “ET-PGM PIC USB” จะเริ่มต้นทำการ Download Code ให้กับ MCU ในทันที เมื่อการ Download Code เสร็จเรียบร้อยแล้ว จะมีข้อความ “Programming Successful” ดังรูป



6. หลังจากทำการ Download Code เรียบร้อยแล้ว ให้ทำการเลือกกำหนดการทำงานของสวิตช์ Mode ของบอร์ด ET-dsPIC33WEB V1.0 ไปไว้ในตำแหน่ง Run โดยให้ตำแหน่งของสวิตช์ Mode อยู่ในตำแหน่งปล่อย และ LED Run สีเขียวติดสว่าง จากนั้นให้กดสวิตช์ Reset 1 ครั้ง บอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่ Download ให้แล้วในทันที