บทที่ 2
แนะนำซอฟต์แวร์

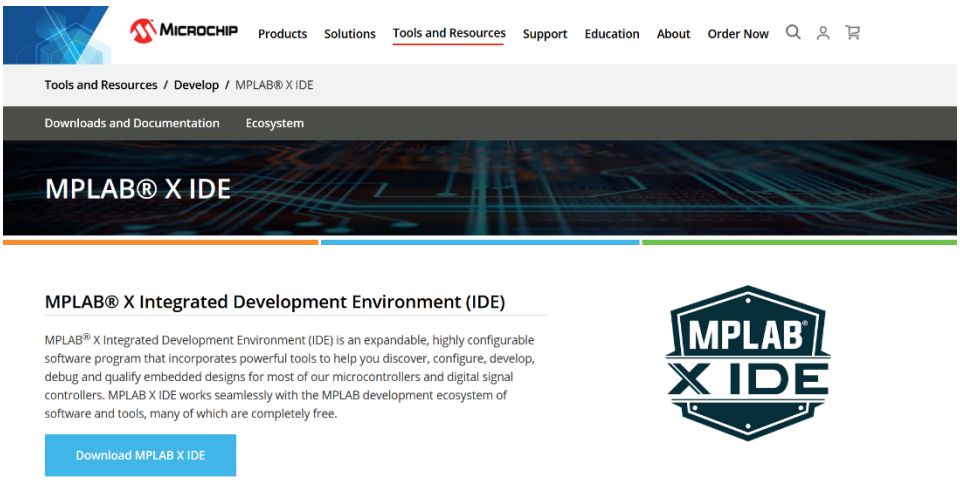
1. บทนำ

ในบทนี้กล่าวถึงซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรมภาษาซี อันประกอบด้วยชุดพัฒนาโปรแกรม MPLAB X IDE และชุดเครื่องมือแปลภาษาซีสำหรับไมโครคอนโทรลเลอร์ 8 บิต MPLAB XC8 Compiler และซอฟต์แวร์สำหรับโปรแกรมชิพซึ่งใช้กับบอร์ดโปรแกรม PICKit2 สำหรับเขียนโปรแกรมที่พัฒนาเสร็จแล้วลงในรอมของไมโครคอนโทรลเลอร์ โดยกล่าวถึงวิธีการติดตั้งและการใช้งานโปรแกรมในระดับเบื้องต้นที่จำเป็นต่อการใช้งาน

2. ติดตั้ง MPLAB X IDE

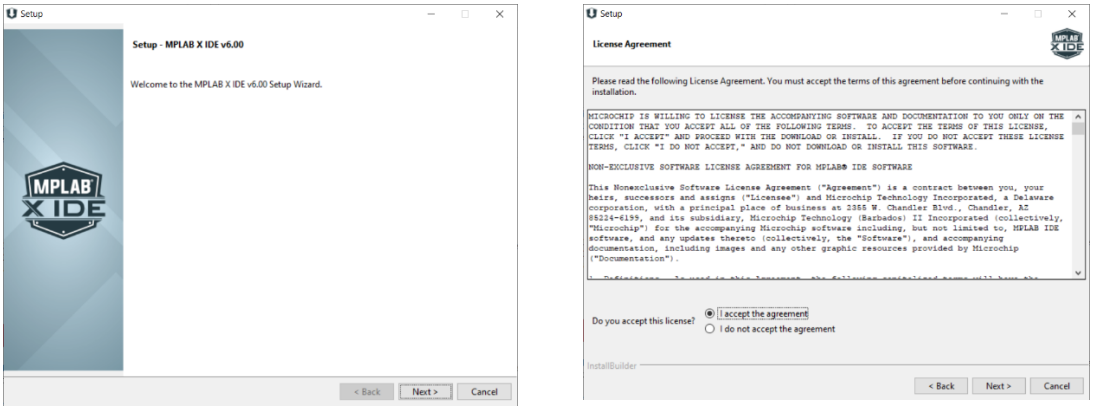
โปรแกรม MPLAB X IDE สามารถดาวน์โหลดได้จาก URL ของเว็บบริษัท Microchip (รูปที่ 2.1) ดังนี้

https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide

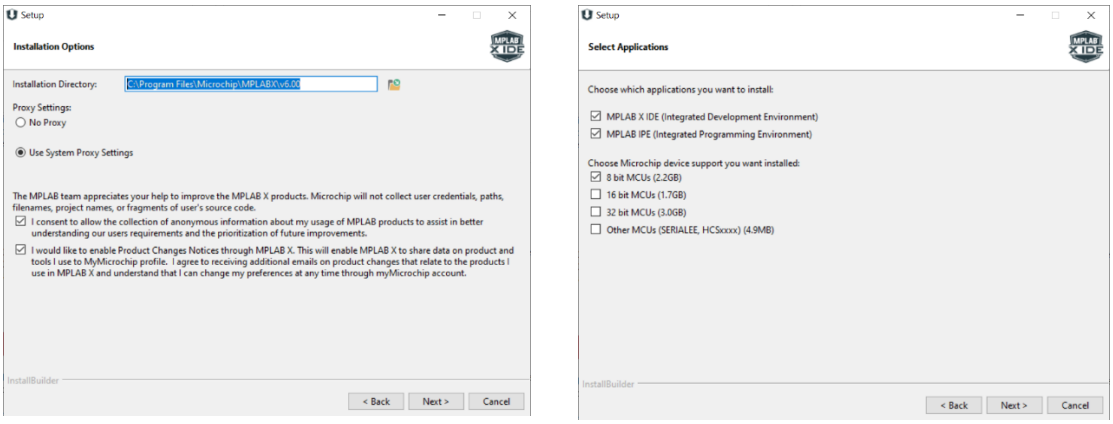


รูปที่ 2.1 หน้าเว็บของ MPLAB X IDE

เมื่อดาวน์โหลดจะได้ไฟล์สำหรับติดตั้งซึ่งขั้นตอนการติดตั้งเป็นดังรูปที่ 2.2 ถึง 2.5

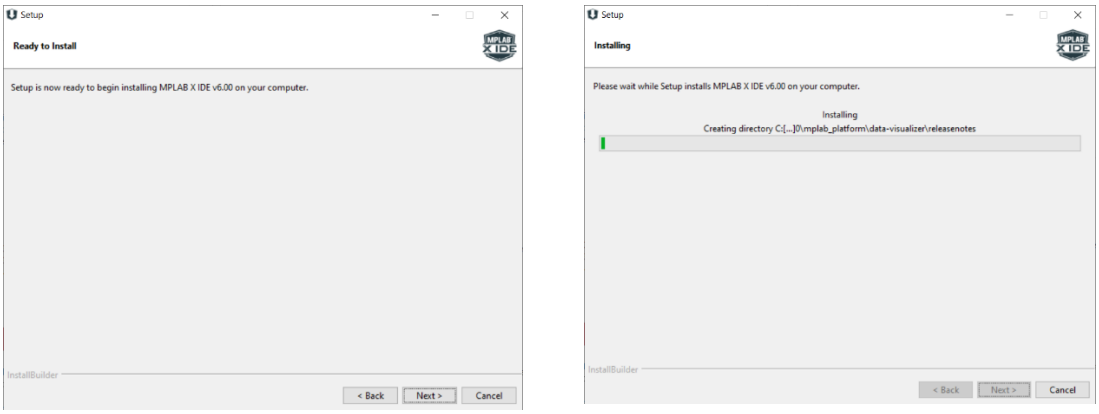


รูปที่ 2.2 (ซ้าย) หน้าแรกของโปรแกรมติดตั้ง MPLAB X IDE
(ขวา) หน้าจอยืนยันการยอมรับข้อตกลงในการใช้งาน

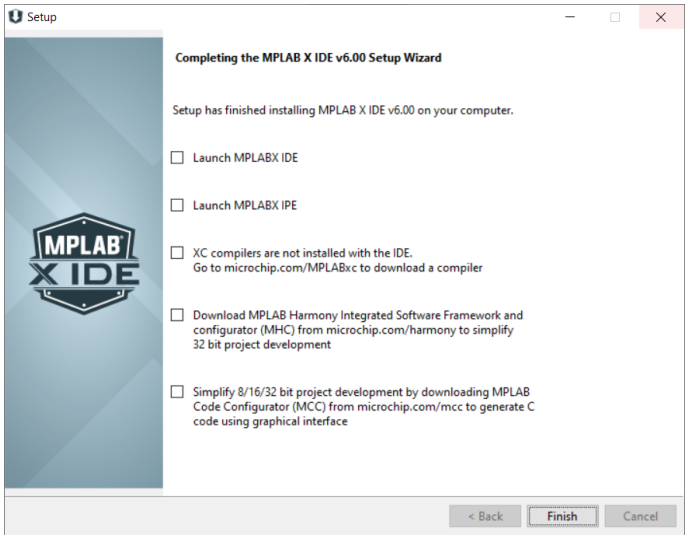


รูปที่ 2.3(ซ้าย) หน้ากำหนดไดเร็กทอรีติดตั้ง MPLAB X IDE
(ขวา) หน้ากำหนดชุดตัวแปลภาษาสำหรับ MPLAB X IDE

จากรูปที่ 2.3 ด้านซ้ายจะพบว่าผู้อ่านสามารถระบุไดเร็กทอรีสำหรับติดตั้งได้ด้วยตนเอง และหลังจากกำหนดเสร็จในขั้นตอนถัดไปเป็นการเลือกติดตั้งเครื่องมือแปลภาษา โดยผู้เขียนเลือกใช้เฉพาะ 8 bit MCU หลังจากนั้นให้ดำเนินการเลือกเข้าสู่ขั้นตอนการติดตั้งถัดไปตามลำดับ และรอจนกว่าการติดตั้งจะเสร็จสิ้นดังรูปที่ 2.5



รูปที่ 2.4 (ซ้าย) ตัวติดตั้ง MPLAB X IDE แจ้งพร้อมติดตั้ง (ขวา) MPLAB X IDE ดำเนินการติดตั้ง

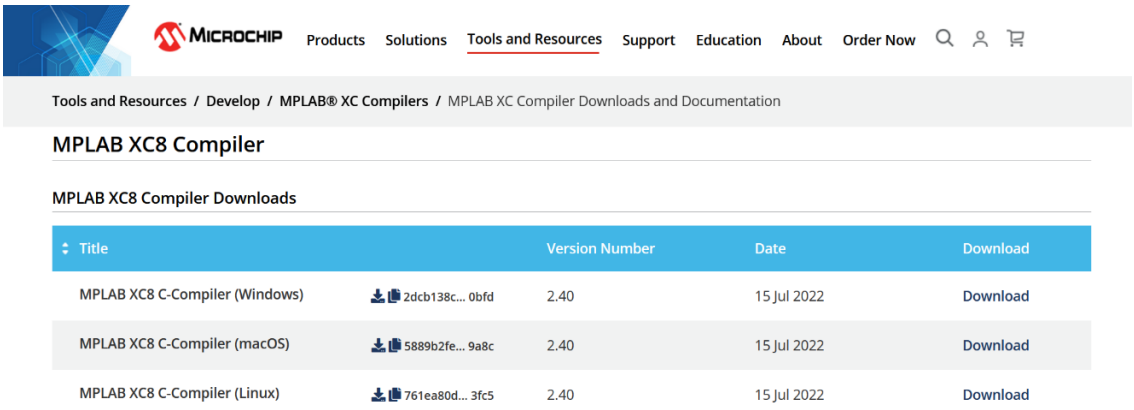


รูปที่ 2.5 หน้าจอรายงานเสร็จสิ้นการติดตั้ง MPLAB X IDE

3. ติดตั้ง MPLAB XC8 C-Compiler

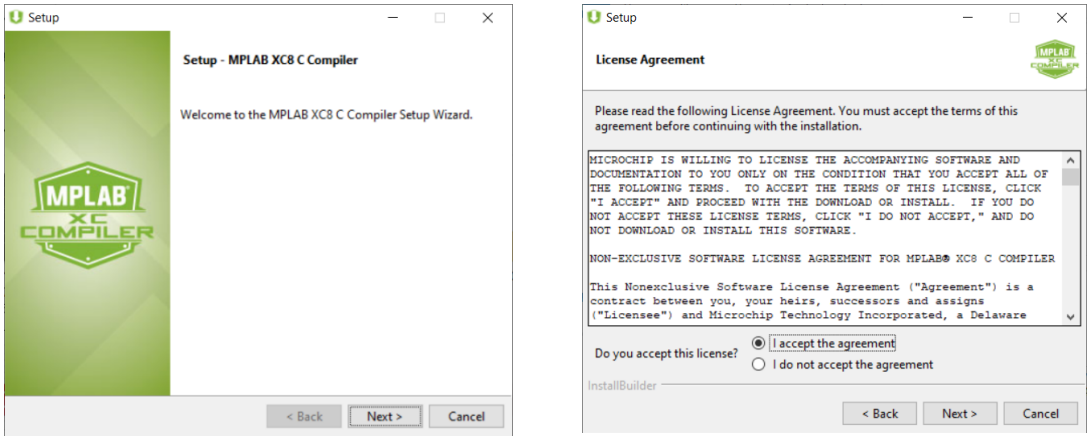
โปรแกรม MPLAB XC8 C-Compiler เป็นชุดตัวแปลภาษาซีสำหรับไมโครคอนโทรลเลอร์สถาปัตยกรรม 8 บิตของบริษัทไมโครชิพ โดยเครื่องมือนี้มีให้เลือกติดตั้งทั้งระบบปฏิบัติการ Windows, Linux และ macOS จาก URL ของเว็บบริษัท Microchip (รูปที่ 2.6) ดังนี้

https://www.microchip.com/en-us/tools-resources/develop/mplab-xc-compilers

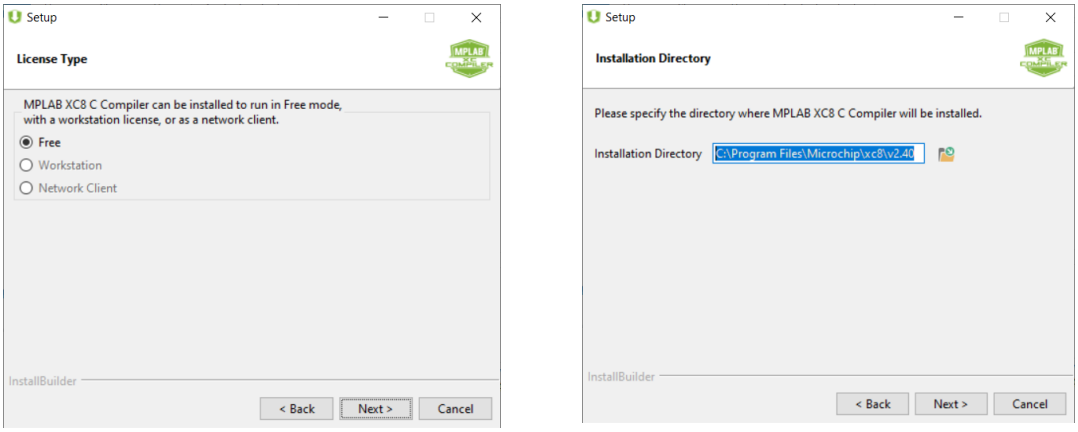


รูปที่ 2.6 หน้าเว็บดาวน์โหลด MPLAB XC8 C-Compiler

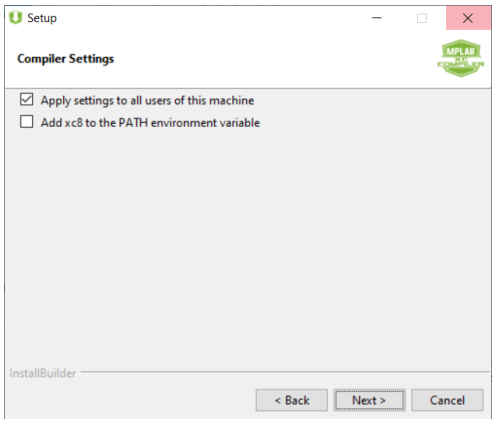
หลังจากได้ไฟล์ติดตั้งให้ดำเนินการติดตั้ง MPLAB XC8 C-Compiler ตามตัวอย่างหน้าจอการติดตั้งในรูปที่ 2.7 ถึง 2.10

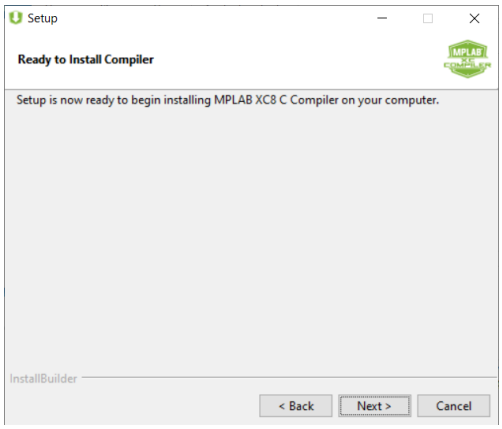


รูปที่ 2.7 (ซ้าย) หน้าจอเริ่มแรกของตัวติดตั้ง MPLAB XC8 C-Compiler
(ขวา) หน้าจอเงื่อนไขการใช้ MPLAB XC8 C-Compiler



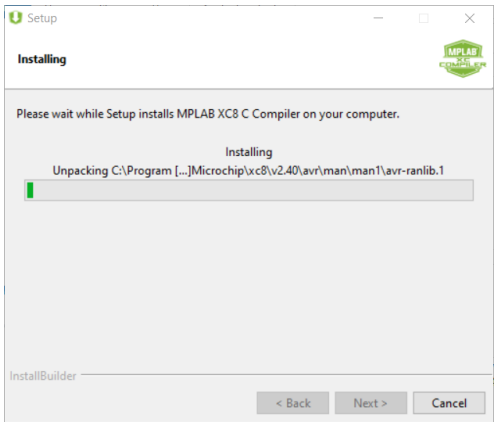
รูปที่ 2.7 (ซ้าย) หน้าจอเลือกประเภทของการรัน XC8 (ขวา) หน้าจอระบุไดเร็กทอรีของ XC8

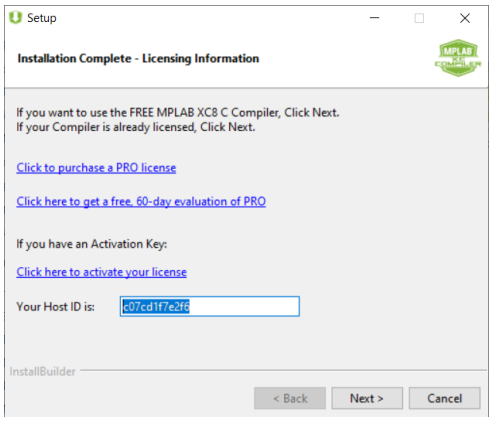




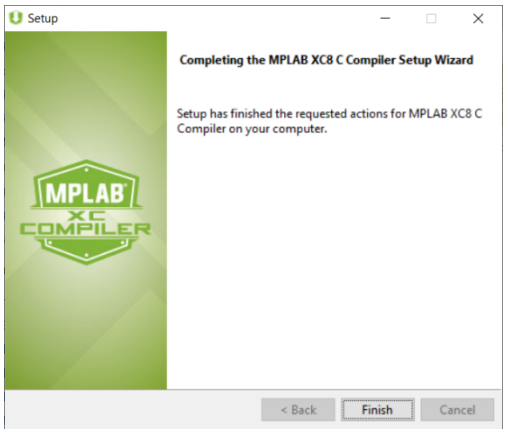
รูปที่ 2.8 (ซ้าย) หน้าจอตั้งค่าการติดตั้ง XC8 (ขวา) หน้าจอรายงานพร้อมติดตั้ง XC8

จากรูปที่ 2.7 ด้านขวาจะพบว่าผู้อ่านสามารถเลือกไดเร็กทอรีสำหรับติดตั้งตัว MPLAB XC8 C-Compiler ได้ และสามารถเลือกการตั้งค่าเพื่อให้ระบบมองเห็นชุดโปรแกรมได้ด้วยการเพิ่ม PATH ดังรูปที่ 2.8 ด้านซ้าย หลังจากนั้นเมื่อคลิก Next ของการเริ่มติดตั้งระบบจะติดตั้งดังรูปที่ 2.9 ด้านซ้าย ให้รอจนกว่าการติดตั้งเสร็จสิ้นดังรูป 2.9 ด้านขวาอันเป็นการใส่รหัสของเครื่องซึ่งโปรแกรมได้สร้างให้ล่วงหน้าแล้ว หลังจากนั้นคลิก Next เพื่อดำเนินงานต่อจะพบหน้าต่างตามรูป 2.10 เป็นการสิ้นสุดการติดตั้ง MPLAB XC8 C-Compiler





รูปที่ 2.9 (ซ้าย) หน้าจอติดตั้งไฟล์ XC8 ลงเครื่อง (ขวา) หน้าจอกำหนดรหัสโฮสต์จากโปรแกรมติดตั้ง XC8

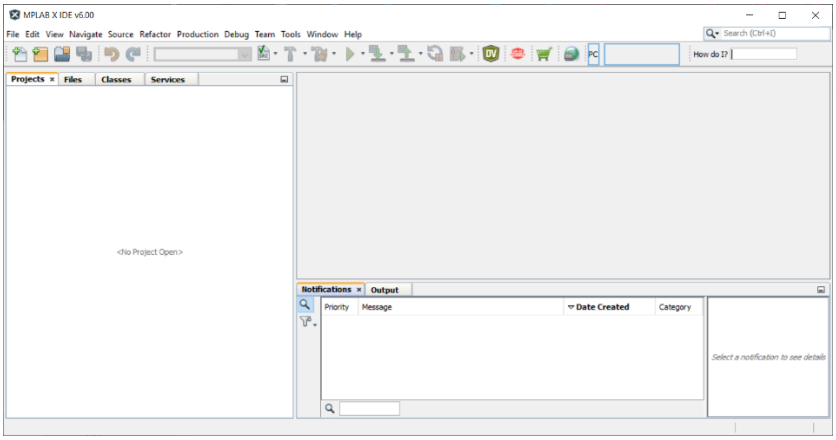


รูปที่ 2.10 หน้าจอเสร็จสิ้นการติดตั้ง MPLAB XC8 C-Compiler

4. ใช้งาน MPLAB X IDE

ลำดับขั้นตอนของการใช้โปรแกรม MPLAB X IDE (รูปที่ 2.11) เพื่อเขียนโปรแกรมภาษาซีประกอบด้วย 5 ขั้นตอน คือ

1. สร้างไฟล์โครงงาน
2. สร้างไฟล์ภาษาซี
3. สร้างตัวแปรค่าเกี่ยวกับเรจิสเตอร์ของไมโครคอนโทรลเลอร์ที่เลือกใช้
4. เขียนโค้ดและคอมไพล์
5. นำไฟล์ผลลัพธ์ไปเขียนลงชิพ

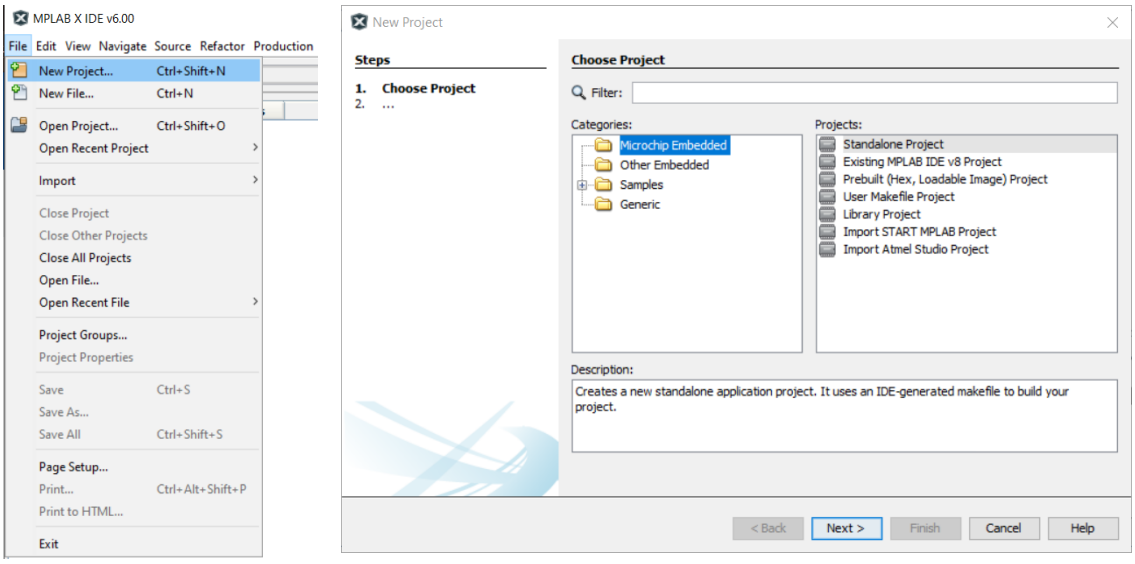


รูปที่ 2.11 หน้าจอโปรแกรม MPLAB X IDE

4.1 สร้างโครงงาน

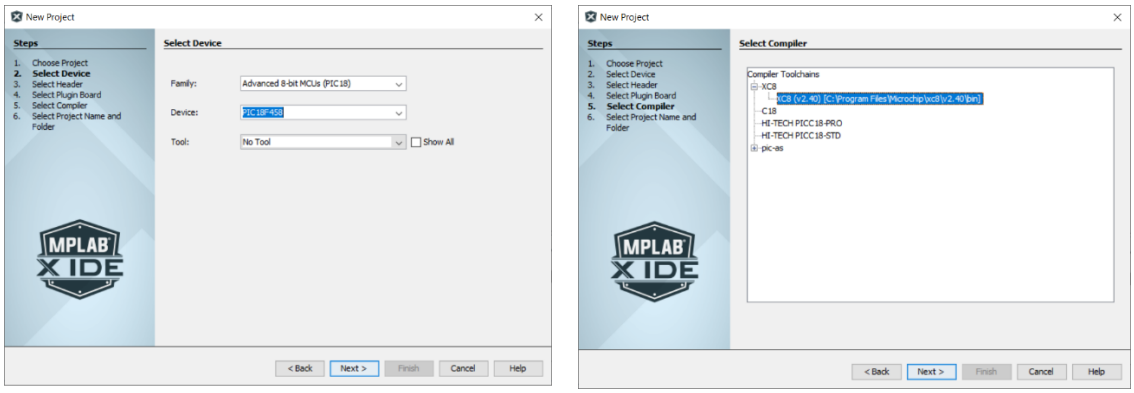
จากหน้าจอในรูปที่ 2.11 ให้คลิกเลือกเมนู File เพื่อเลือกสร้างโครงงานจากรายการ New Project... ดังรูปที่ 2.12 (ซ้าย) หลังจากนั้นจะมีหน้าต่างให้เลือกประเภทของโครงงานที่

ต้องการสร้างดังรูปที่ 2.12 (ขวา) ให้เลือกเป็นประเภท (Categories) แบบ Microchip Embedded และโครงงานเป็น Standalone Project แล้วคลิก Next เพื่อดำเนินการต่อไป

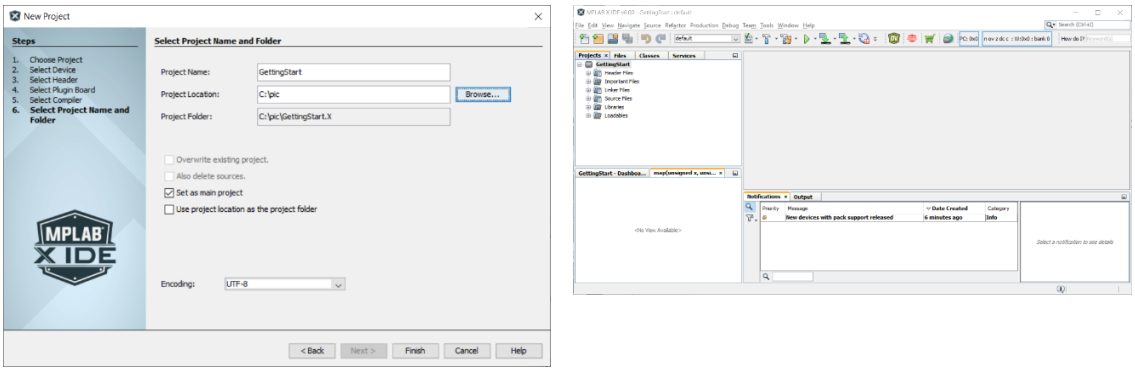


รูปที่ 2.12 (ซ้าย) เมนูสร้างโครงงาน (ขวา) หน้าจอเลือกประเภทโครงงาน

จากรูปที่ 2.13 (ซ้าย) เป็นขั้นตอนการเลือกประเภทของชิไมโครคอนโทรลเลอร์ที่เลือกใช้ โดยในตัวอย่างเลือก PIC18F458 อันเป็นชิพในตระกูล Advanced 8-bit MCUs (PIC18) แล้วคลิก Next เพื่อเลือกชุดเครื่องมือแปลภาษาโปรแกรมภาษาซีดังรูปที่ 2.13 (ขวา) โดยเลือกใช้เป็น XC8 แล้วดำเนินการขั้นตอนกำหนดชื่อโครงงานและไดเร็กทอรีเก็บไฟล์ดังรูปที่ 2.14 (ซ้าย) และคลิกสู่ขั้นตอนสุดท้ายของการสร้างโครงงานคือรอให้ MPLAB X IDE สร้างโครงงานต่าง ๆ ให้เสร็จสิ้นและจะได้ผลลัพธ์ดังรูป 2.14 (ขวา) เป็นการสิ้นสุดขั้นตอนของการสร้างโครงงาน



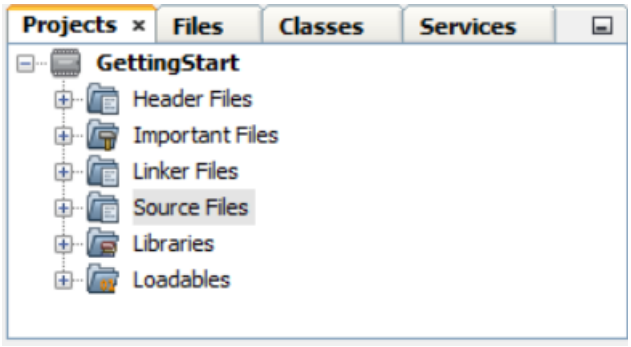
รูปที่ 2.13 (ซ้าย) กำหนดชิพที่ใช้ในโครงงาน (ขวา) เลือกเครื่องมือแปลภาษา



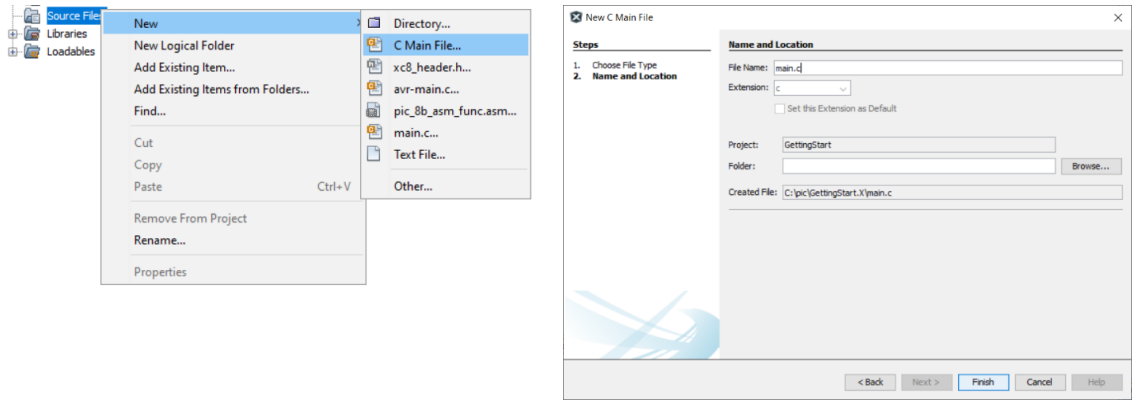
รูปที่ 2.14 (ซ้าย) ตั้งชื่อโครงงานและที่เก็บโครงงาน (ขวา) หน้าจอเมื่อสร้างโครงงานเสร็จแล้ว

4.2 เพิ่มไฟล์ภาษาซี

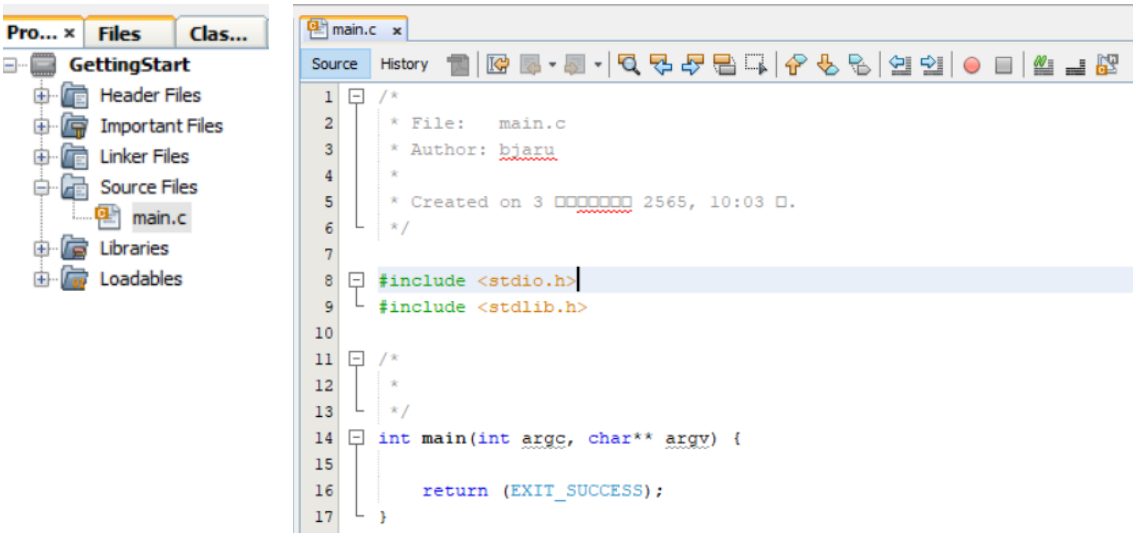
จากหน้าจอในรูปที่ 2.15 จะสังเกตเห็นโฟลเดอร์ชื่อ Source Files จากแท็บ Projects ในรายการโครงงานตามชื่อที่ตั้งไว้ (ในตัวอย่างผู้เขียนตั้งชื่อเป็น GettingSart) ซึ่งจะพบว่าในโฟลเดอร์นี้ยังไม่มีไฟล์ใด ๆ อยู่ภายใน ดังนั้น จึงต้องทำการเพิ่มไฟล์ภาษาซีเข้าไปด้วยการคลิกขวาที่ Source Files จะแสดงรายการเมนูดังภาพที่ 2.16 (ซ้าย) ให้เลือกเป็น C Main File... หลังจากนั้นจะแสดงหน้าต่างดังรูป 2.16 (ขวา) เพื่อกำหนดชื่อของไฟล์ที่สร้าง โดยในตัวอย่างนี้สร้างไฟล์ชื่อ main.c และจะพบว่าในโฟลเดอร์ Source Files จะมีไฟล์ชื่อ main.c ปรากฏขึ้นดังรูปที่ 2.17 (ซ้าย) และข้อมูลภายในไฟล์เป็นดังรูป 2.17 (ขวา)



รูปที่ 2.15 โฟลเดอร์ Source Files สำหรับเพิ่มไฟล์ภาษาซี



รูปที่ 2.16 (ซ้าย) เมนูสำหรับสร้าง C Main File (ขวา) หน้าต่างระบุชื่อไฟล์ภาษาซีที่ต้องการสร้าง



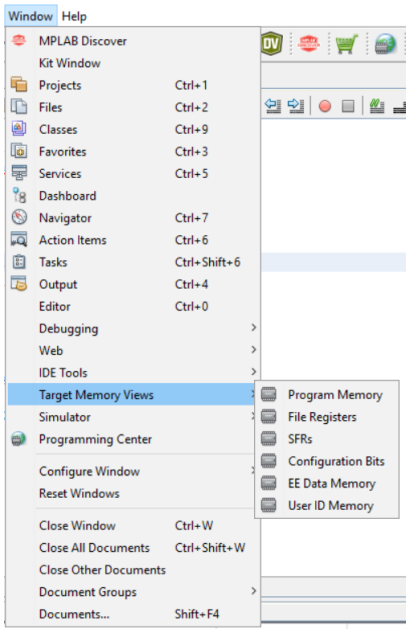
รูปที่ 2.17 (ซ้าย) ชื่อไฟล์ภาษาซีที่สร้างขึ้น (ขวา) โค้ดในไฟล์ภาษาซีที่ MPLAB X IDE สร้างให้

4.3 สร้างข้อมูลการตั้งค่า

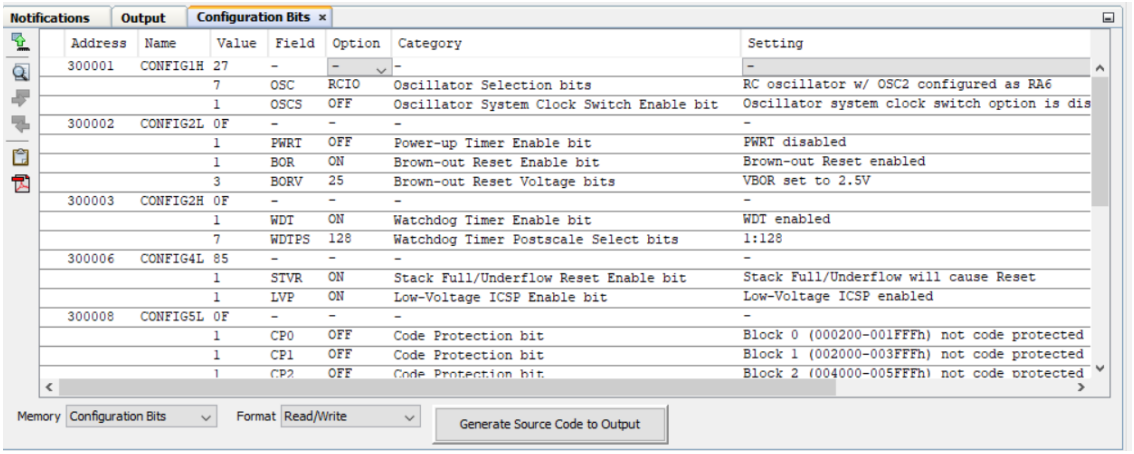
เนื่องด้วยชิพไมโครคอนโทรลเลอร์ที่เลือกใช้นั้นเป็นมีหน่วยทำงานที่มากกว่าหน่วยประมวลผลดังที่ได้กล่าวไปในบทที่ 1 ดังนั้น ผู้เขียนโปรแกรมจึงต้องตั้งค่าการทำงานให้กับเรจิสเตอร์ต่าง ๆ ที่ทำหน้าที่ควบคุมหรือเก็บผลลัพธ์การทำงานของไมโครคอนโทรลเลอร์

โดยเครื่องมือของ MPLAB X IDE รองรับการสร้างข้อมูลของเรจิสเตอร์และค่าเริ่มต้นของเรจิสเตอร์ของชิพไมโครคอนโทรลเลอร์ที่เลือกใช้งานให้อัตโนมัติ ดังนั้น ในหัวข้อนี้จึงกล่าวถึงขั้นตอนของการแสดงรายการเรติสเตอร์และนำออกเรจิสเตอร์พร้อมค่าเริ่มต้นออกมาเป็นข้อความและนำข้อความเหล่านี้คัดลอกลงไฟล์ภาษาซีที่สร้างขึ้นจากขั้นตอนในหัวข้อ 4.2

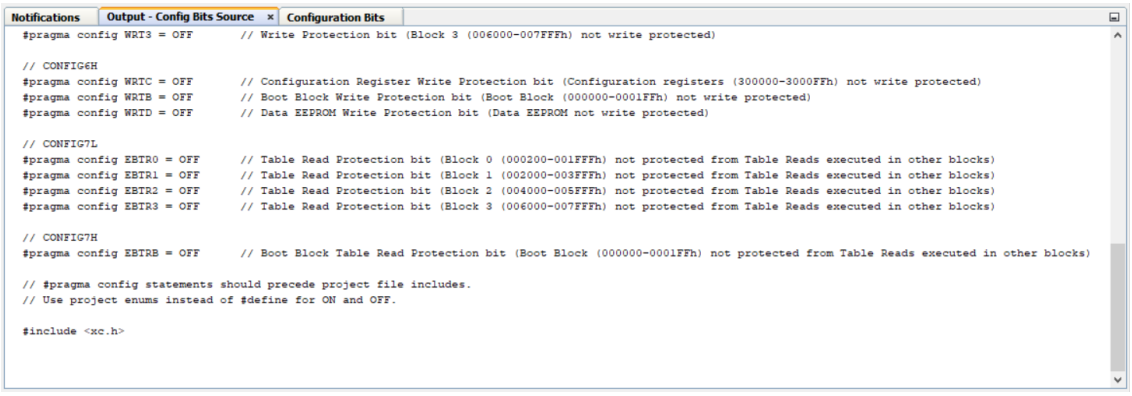
ขั้นตอนแรกให้เลือกรายการเมนู **Configuration Bits** จากเมนู Target Memory Views ของ Window ดังรูปที่ 2.18 จะมีรายการแท็บชื่อ Configuration Bits เพิ่มเข้ามาดังรูปที่ 2.19 ให้คลิกที่ปุ่ม Generate Source Code to Output ที่อยู่ด้านล่างของแท็บ จะได้ผลลัพธ์ของโค้ดที่สร้างขึ้นปรากฏในแท็บ Output ดังรูปที่ 2.20



รูปที่ 2.18 เมนูแสดง Configuration Bits

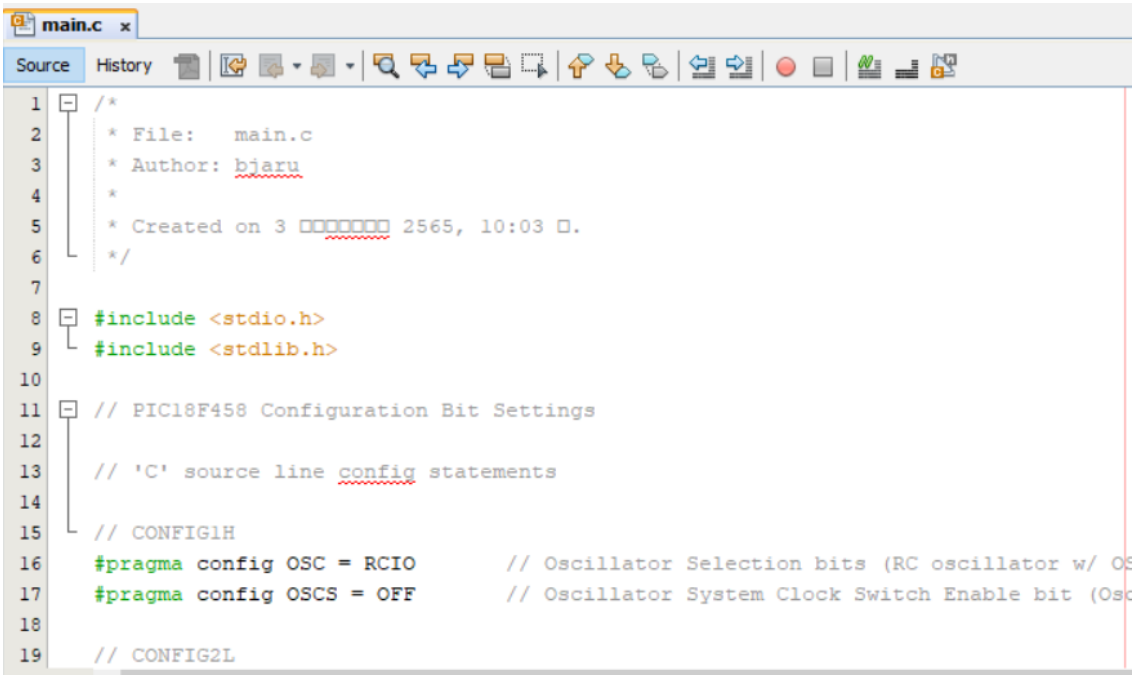


รูปที่ 2.19 แท็บ Configuration Bits



รูปที่ 2.20 โค้ดที่สร้างจาก Configuration Bits

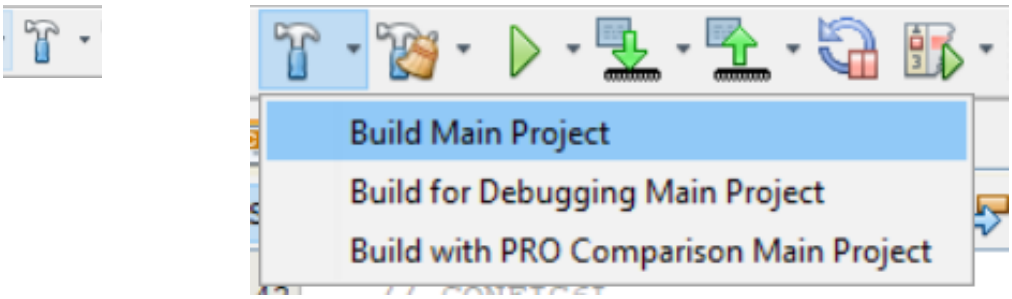
จากรูปที่ 2.20 ให้เลือกคัดลอกโค้ดทั้งหมดไปวางใน main.c ดังผลลัพธ์ในรูปที่ 2.21 หลังจากนั้นให้ทำการบันทึกไฟล์ main.c



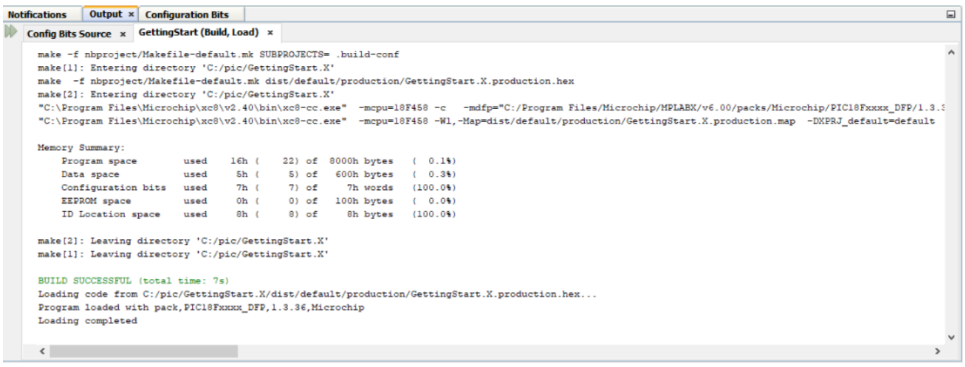
รูปที่ 2.21 ตัวอย่าง main.c ที่เพิ่ม Configuration Bits เข้าไป

4.4 คอมไพล์

หลังจากได้ไฟล์โปรแกรมหลักของภาษาซีและการตั้งค่าเริ่มต้นสำหรับชิพไมโครคอนโทรลเลอร์ PIC เป็นที่เรียบร้อยจากขั้นตอน 4.3 ให้ทดลองทำการคอมไพล์โปรแกรมด้วยการคลิกที่ปุ่มสำหรับคอมไพล์ (Build) ดังรูปที่ 2.22 (ซ้าย) หรือคลิกที่ลูกศรด้านขวาของไอคอนเพื่อให้แสดงรายการวิธีการคอมไพล์จะเป็นดังรูป 2.22 (ขวา) ให้เลือกเป็น Build Main Project ตัว MPLAB XC8 C-Compiler จะดำเนินการคอมไพล์โค้ดในไฟล์ภาษาซีที่อยู่ในโครงงานที่สร้างขึ้น และรายงานผลของการคอมไพล์ในแท็บ Output ดังตัวอย่างรูป 2.23 ถ้ามีการรายงาน BUILD SUCCESSFUL แสดงว่าโค้ดคอมไพล์ผ่านจะเข้าสู่ขั้นตอนนำไฟล์ผลลัพธ์ไปโปรแกรมลงชิพต่อไป แต่ถ้ามีข้อผิดพลาดจะต้องทำการแก้ไขโค้ดจนกว่าจะคอมไพล์ผ่าน

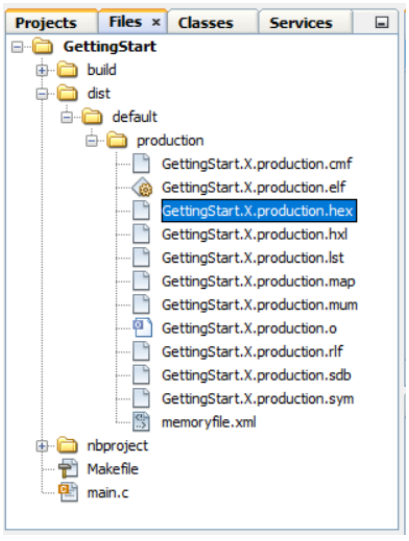


รูปที่ 2.22 (ซ้าย) ไอคอนคอมไพล์ (ขวา) รายการคำสั่งย่อยสำหรับคอมไพล์

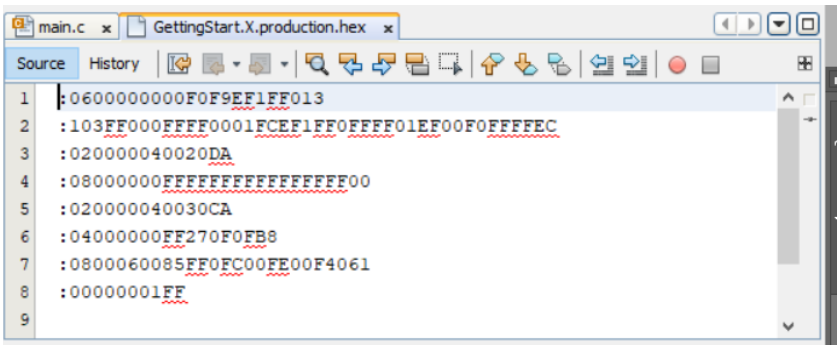


รูปที่ 2.23 ตัวอย่างผลลัพธ์จากการคอมไพล์

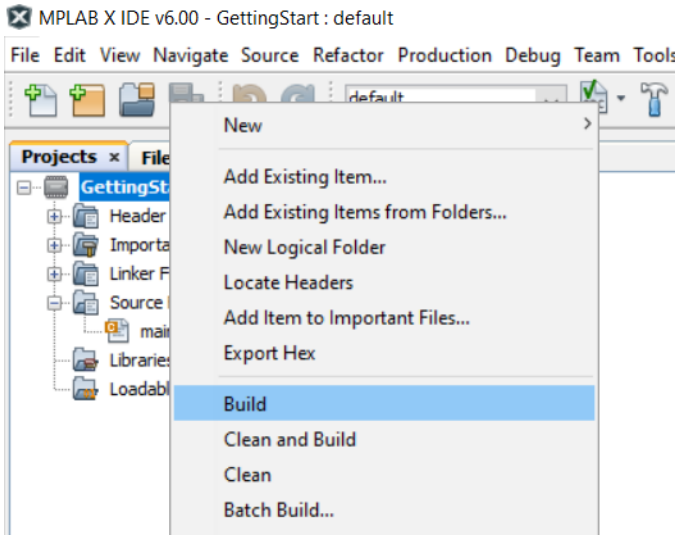
ไฟล์ผลลัพธ์ที่นำไปเขียนลงชิพจะมีนามสกุล .HEX โดยดูได้จากแท็บ Files ในโฟลเดอร์ production ที่อยู่ภายใน ชื่อโครงงาน/dist/default ดังรูปที่ 2.24 และเมื่อดับคลิกที่ไฟล์จะถูกเปิดออกมาดังรูปที่ 2.25 ส่วนถ้าต้องการเข้านำไปโปรแกรมลงชิพด้วยโปรแกรม PICKit2 จะต้องเข้าไปในโฟลเดอร์ production ในไดเร็ก default ที่อยู่ในโฟลเดอร์ dist ภายในไดเร็กทอรีที่สร้างโครงงานเอาไว้ ส่วนกรณีที่มีหลายโครงงานอยู่ในแท็บ Projects ผู้อ่านสามารถเลือกคอมไพล์ด้วยการคลิกขวาที่ชื่อโครงงานที่ต้องการคอมไพล์แล้วเลือกรายการ Build ดังรูป 2.26 ได้เช่นกัน



รูปที่ 2.24 รายการไฟล์ใน production



รูปที่ 2.25 ตัวอย่างข้อมูลใน .HEX



รูปที่ 2.26 เมนู Build เมื่อคลิกขวาเลือกที่โครงงาน

ตัวอย่างไฟล์ main.c สำหรับ PIC18F458 ที่ได้ทำการปรับค่าบิตการทำงานแล้วเป็นดังนี้

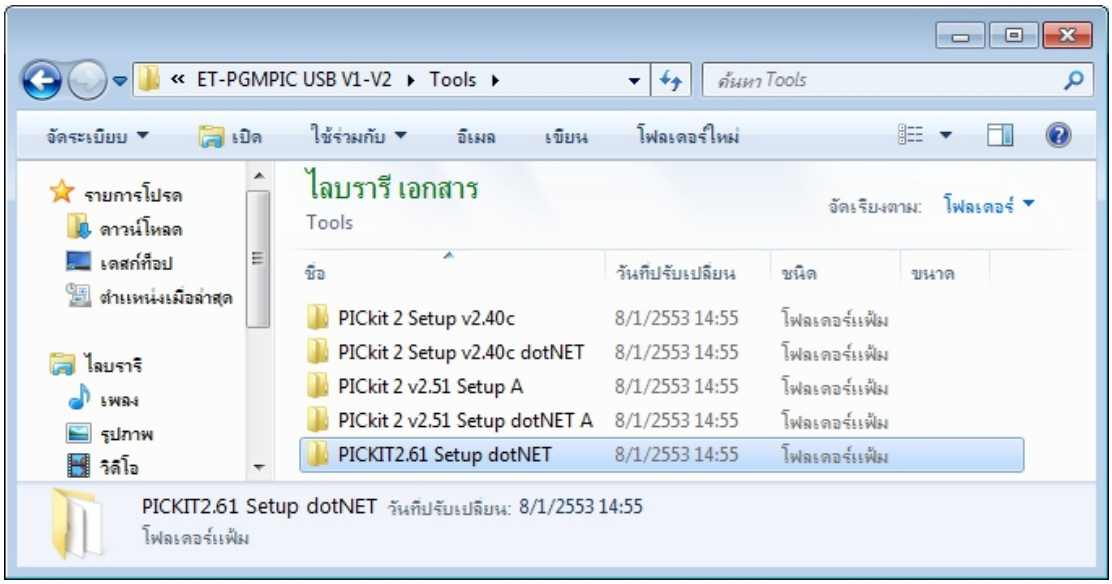
```
#pragma config OSC = HS

#pragma config OSCS = ON
#pragma config PWRT = OFF
#pragma config BOR = ON
#pragma config BORV = 25
#pragma config WDT = OFF
#pragma config WDTPS = 128
#pragma config STVR = ON
#pragma config LVP = ON
#pragma config CP0 = OFF
#pragma config CP1 = OFF
#pragma config CP2 = OFF
#pragma config CP3 = OFF
#pragma config CPB = OFF
#pragma config CPD = OFF
#pragma config WRT0 = OFF
#pragma config WRT1 = OFF
#pragma config WRT2 = OFF
#pragma config WRT3 = OFF
#pragma config WRTC = OFF
#pragma config WRTB = OFF
#pragma config WRTD = OFF
#pragma config EBTR0 = OFF
#pragma config EBTR1 = OFF
#pragma config EBTR2 = OFF
#pragma config EBTR3 = OFF
#pragma config EBTRB = OFF
#include <xc.h>
#define _XTAL_FREQ 20000000
void main(void) {
    return;
}
```

และตัวอย่าง main.c สำหรับ PIC16F877 เป็นดังนี้

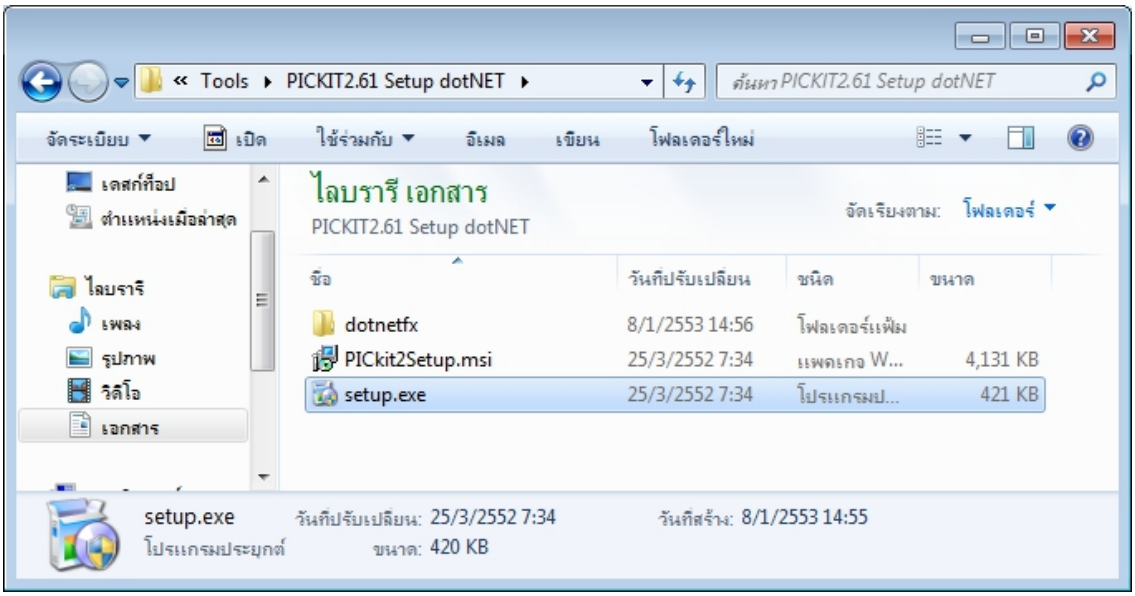
```
#pragma config FOSC =
#pragma config WDTE =
#pragma config PWRTE =
#pragma config CP = OFF
#pragma config BOREN = ON
#pragma config LVP = ON
#pragma config CPD = OFF
#pragma config WRT = ON

#include <xc.h>
#define _XTAL_FREQ 20000000


void main(void) {
    return;
}
```

5. ติดตั้ง PICKit2

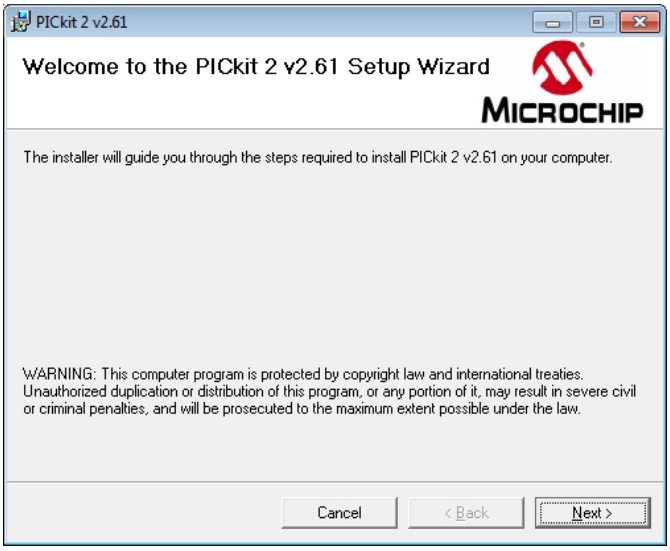
การติดตั้งโปรแกรม PICkit 2 ให้เรียกจากโฟลเดอร์ที่ชื่อว่า PICKIT 2.61 Setup dotNET (รูปที่ 2.27) ให้เข้าไปในโฟลเดอร์ดังกล่าวเพื่อเรียกโปรแกรมติดตั้งที่ชื่อว่า setup.exe (รูปที่ 2.28) แล้วมีหน้าต่างดังรูปที่ 2.29 ปรากฏขึ้นมา



รูปที่ 2.27 โฟลเดอร์ PICKIT 2.61 ในแผ่นซีดีของ ET-PGM PIC USB V1-V2

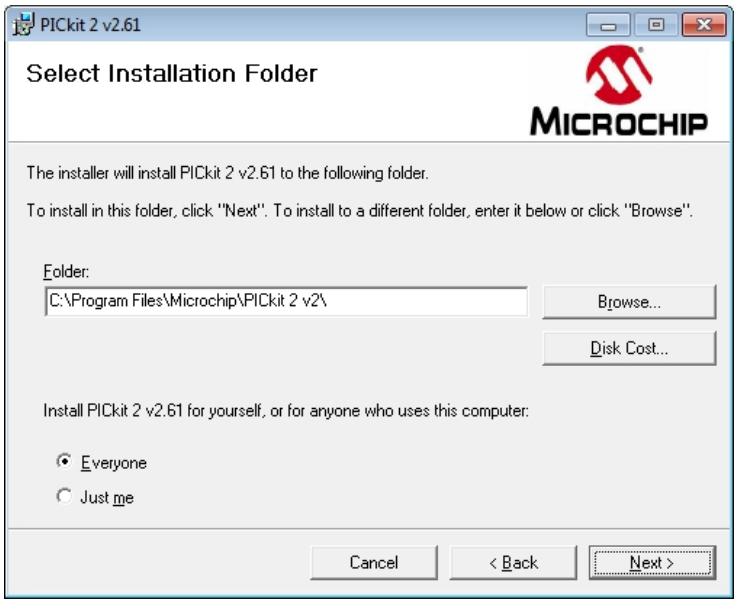


รูปที่ 2.28 ไอคอนโปรแกรมติดตั้ง PICkit 2

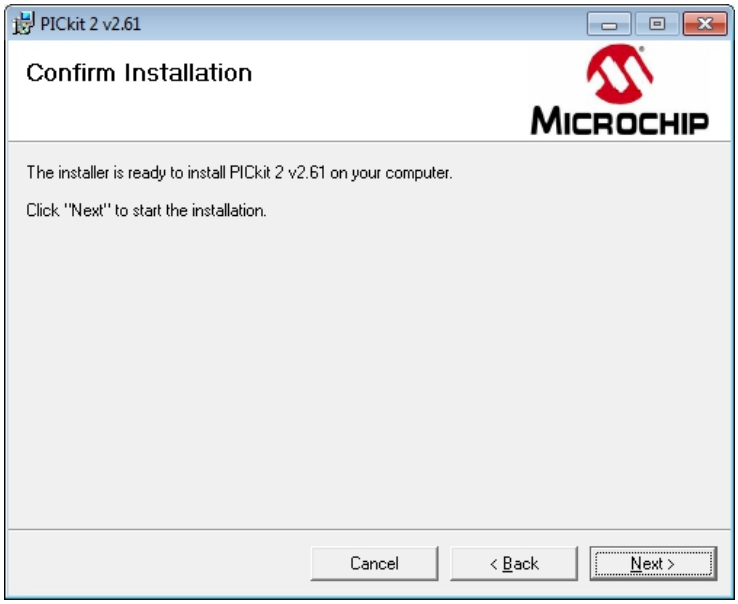


รูปที่ 2.29 หน้าจอแรกของโปรแกรมติดตั้ง PICkit 2

จากหน้าจอแรกของโปรแกรมติดตั้ง PICkit 2 (รูปที่ 2.29) ให้คลิกที่ปุ่ม Next เพื่อเข้าสู่หน้าจอกำหนดโฟลเดอร์เก็บโปรแกรม (รูปที่ 2.30) ซึ่งปกติแล้วโปรแกรม PICkit 2 จะติดตั้งอยู่ที่ C:\Program Files\Microchip\PICkit 2 v2\ แต่สามารถระบุให้ติดตั้งในโฟลเดอร์อื่นได้ด้วยการพิมพ์ชื่อโฟลเดอร์ใหม่ที่ช่องกำหนดโฟลเดอร์ (ทับชื่อโฟลเดอร์ที่โปรแกรมกำหนด) หรือคลิกที่ปุ่ม Browse... เพื่อเปิดหน้าต่างสำหรับเลือกโฟลเดอร์ปลายทาง และเมื่อกำหนดเสร็จแล้วให้คลิกที่ปุ่ม Next > เพื่อเข้าสู่หน้าจอเตรียมการ ซึ่งเมื่อโปรแกรมติดตั้งได้ทำการเตรียมการสำหรับติดตั้งเสร็จจะรายงานผลการทำงานดังรูปที่ 2.31 หลังจากนั้นให้คลิกที่ปุ่ม Next > อีกครั้งหนึ่งเพื่อดำเนินการขั้นต่อไป

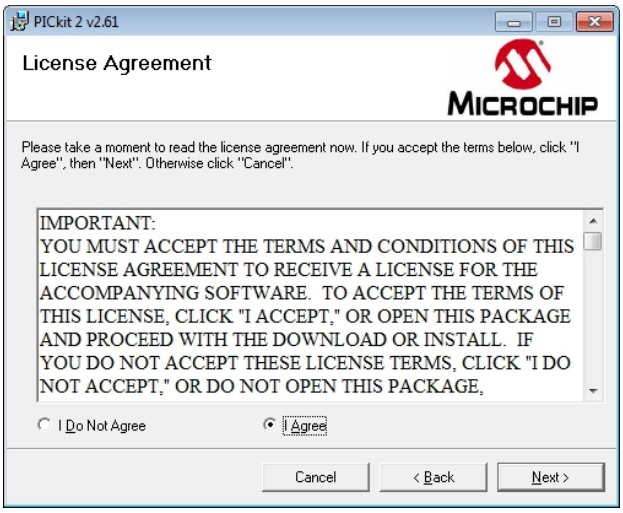


รูปที่ 2.30 หน้าต่างกำหนดโฟลเดอร์ติดตั้งโปรแกรม PICkit 2

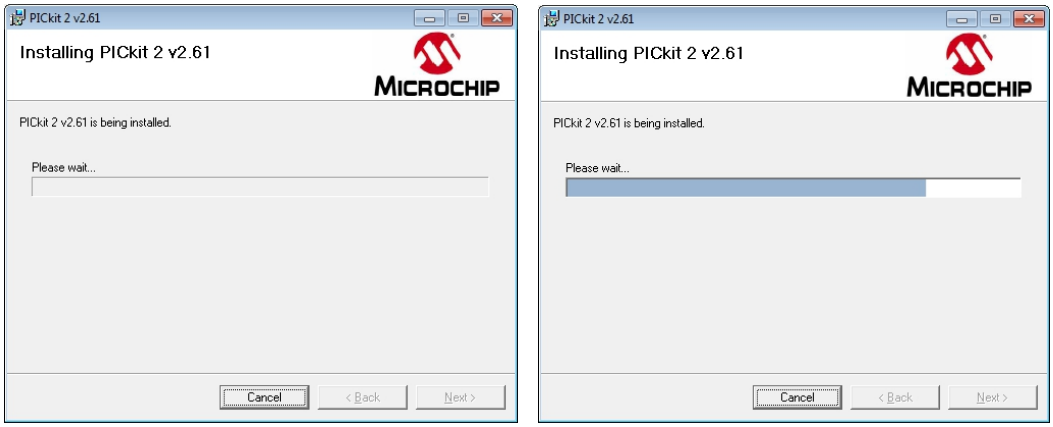


รูปที่ 2.31 หน้าจอรายการผลการเตรียมการติดตั้งของ PICkit 2

เมื่อผ่านจากหน้าจอรายงานผลการเตรียมการจะเป็นหน้าจอของคำบรรยายสิทธิ์การใช้โปรแกรม (รูปที่ 2.32) เมื่ออ่านจบ จะต้องยอมรับข้อกำหนดดังกล่าวด้วยการเลือก I Agree แล้วคลิกที่ปุ่ม Next > เพื่อให้โปรแกรมติดตั้งทำการติดตั้งแฟ้มต่างๆ ลงเครื่องคอมพิวเตอร์ได้ (รูปที่ 2.33)

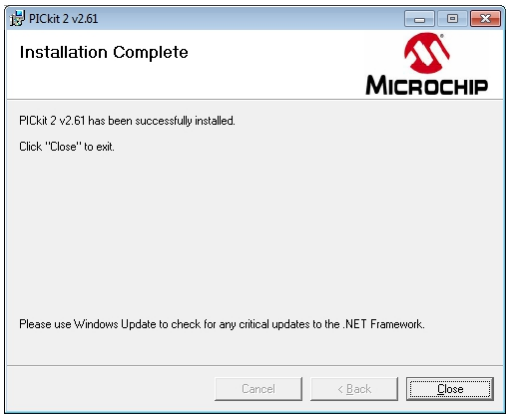


รูปที่ 2.32 หน้าจอบรรยายสิทธิ์ของการใช้โปรแกรม PICkit 2



รูปที่ 2.33 หน้าจอรายการความคืบหน้าในการติดตั้งโปรแกรม PICkit 2

เมื่อโปรแกรมแสดงหน้าจอโปรแกรมดังรูปที่ 2.34 แสดงว่าการติดตั้งโปรแกรม PICkit 2 นั้นได้ดำเนินจนเสร็จสิ้นแล้ว ในขั้นตอนนี้ให้คลิกที่ปุ่ม Close เพื่อปิดการทำงานของโปรแกรมติดตั้ง และจะเห็นว่ามีไอคอน (รูปที่ 2.35) และรายการโปรแกรมของโปรแกรม PICkit 2 ติดตั้งอยู่ในระบบเป็นที่เรียบร้อย



รูปที่ 2.33 หน้าจอรายงานผลความสำเร็จในการติดตั้งโปรแกรม PICkit 2

6. ใช้งาน PICKit2

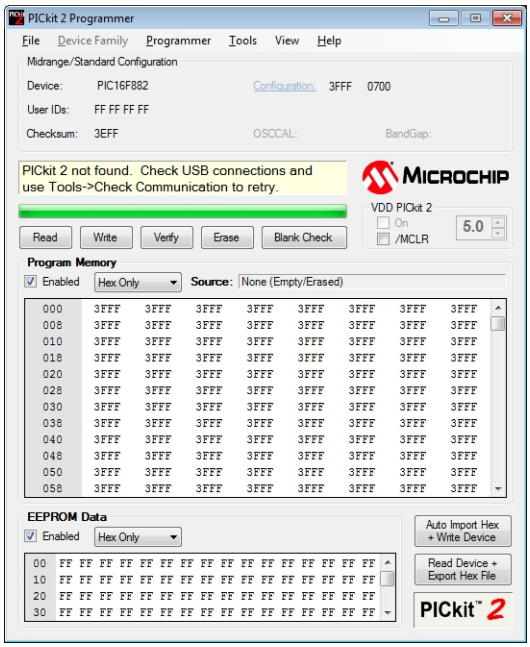
การตั้งค่าการทำงานของโปรแกรม PICkit 2 นั้นจะถูกกำหนดโดยอัตโนมัติ เนื่องจากทุกครั้งที่โปรแกรม PICkit 2 เริ่มต้นทำงานจะดำเนินการตรวจสอบสถานะของบอร์ดโปรแกรมชิพที่เชื่อมต่อกับพอร์ตยูเอสบีว่าเป็นเครื่องโปรแกรมที่ PICkit 2 รู้จักหรือไม่ ถ้าใช่โปรแกรมจะตรวจสอบสถานะของแรงดันและประเภทของชิพไมโครคอนโทรลเลอร์ที่ติดตั้งอยู่บนเครื่องโปรแกรม

การเรียกโปรแกรม PICkit 2 ทำได้ 2 วิธี คือ เรียกจากไอคอนดังรูปที่ 2.34 หรือเรียกผ่านทางรายการเมนูโปรแกรมจากเมนูเริ่มต้นของวินโดวส์ หลังจากนั้นจะเข้าสู่หน้าจอโปรแกรมหลัก (รูปที่ 2.36) แต่ถ้าการเรียกโปรแกรมนั้นเป็นการเรียกใช้ครั้งแรก จะมีหน้าจอ (รูปที่ 2.35) แสดงขึ้นมาเพื่อบอกให้ทราบว่าโปรแกรม PICkit 2 ต้องการที่จะปรับปรุงระบบปฏิบัติการก่อนที่จะเรียกให้ตัวโปรแกรมทำงาน ซึ่งต้องคลิกที่ปุ่ม ตกลง เพื่อให้การดำเนินการดังกล่าวนี้ดำเนินต่อไป

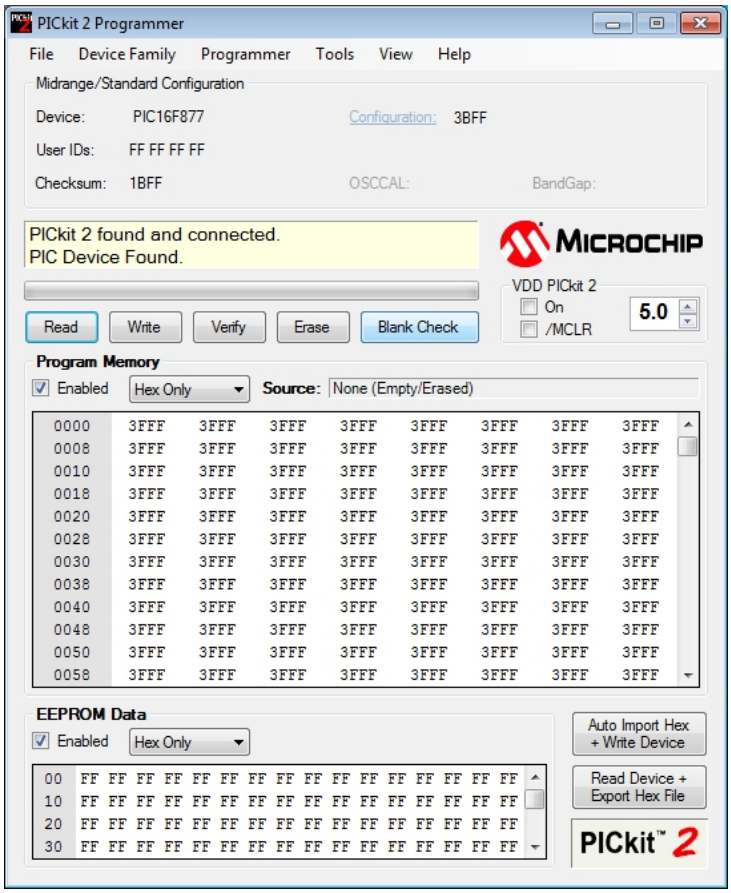
รูปที่ 2.34 ไอคอนโปรแกรม PICkit 2

รูปที่ 2.35 หน้าต่างเริ่มต้นการทำงานครั้งแรกของ PICkit 2

ถ้าโปรแกรม PICkit 2 ไม่พบเครื่องโปรแกรมชิพจะการรายงานว่า PICkit 2 not found. Check USB connections and use Tools->Check Communication to retry. (รูปที่ 2.36) ซึ่งหมายความว่าโปรแกรมไม่พบการเชื่อมต่อระหว่างบอร์ดโปรแกรมชิพกับเครื่องพีซี ให้ดำเนินการเชื่อมต่อบอร์ดโปรแกรมชิพเข้ากับยูเอสบีพอร์ตให้เรียบร้อย หลังงจากนั้นเลือกรายการ Check จากเมนู Tools เพื่อดำเนินการตรวจสอบการเชื่อมต่ออีกครั้ง แต่ถ้าการเชื่อมต่อระหว่างโปรแกรมกับบอร์ดโปรแกรมชิพมีการเชื่อมต่อกันสำเร็จจะรายงานผลดังรูปที่ 2.37 พร้อมทั้งมีข้อความรายงานผลว่า PICkit 2 found and connected. PIC Device Found. พร้อมทั้งระบุด้วยว่าชิพที่ติดตั้ง (Device) บนบอร์ดโปรแกรมชิพคือ PIC16F877



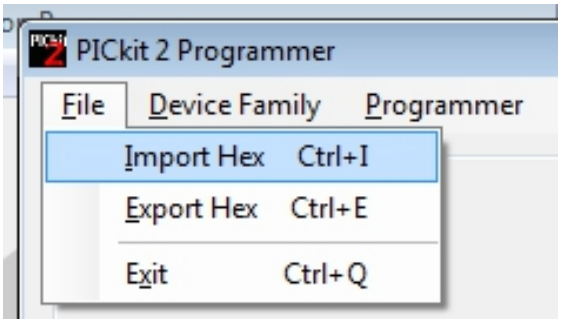
รูปที่ 2.36 หน้าต่างหลักของโปรแกรม PICkit 2 และรายงานให้ทราบว่าไม่พบเครื่องโปรแกรมชิพ



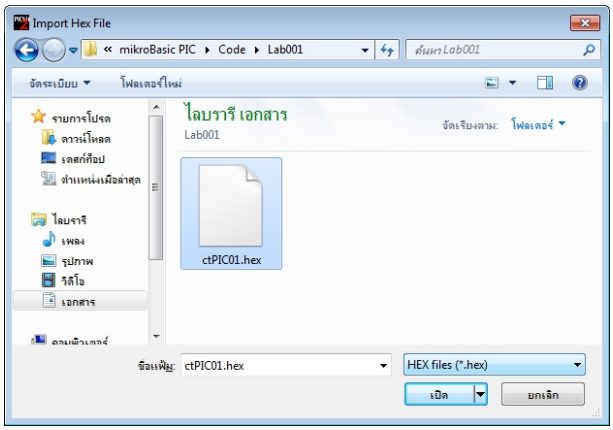
รูปที่ 2.37 หน้าต่างหลักของโปรแกรม PICkit 2 และรายงานให้ทราบว่าพบเครื่องโปรแกรมชิพ

การเปิดแฟ้ม คือ การนำเข้าแฟ้มฐานสิบหก เพื่อใช้สำหรับนำข้อมูลจากแฟ้มที่นำเข้านี้ไปใช้สำหรับการโปรแกรมลงชิพไมโครคอนโทรลเลอร์ต่อไป

การเปิดแฟ้มทำได้ด้วยการเลือกที่เมนู File ของโปรแกรม PICkit 2 และเลือกรายการ Import Hex (รูปที่ 2.38) หรือใช้แป้นลัด คือ กดแป้น Ctrl ค้างแล้วกดแป้นอักษร I หลังจากนั้นหน้าต่าง Import Hex File ปรากฏขึ้นมา (รูปที่ 2.39) สำหรับให้งานเลือกแฟ้มที่ต้องการนำเข้า เมื่อเลือกเสร็จให้คลิกที่ปุ่ม เปิด ถ้าโปรแกรมสามารถนำเข้าแฟ้มฐานสิบหกได้สำเร็จจะรายงานผลการทำงานดังรูปที่ 2.40



รูปที่ 2.38 เมนู Import Hex ของโปรแกรม PICkit 2

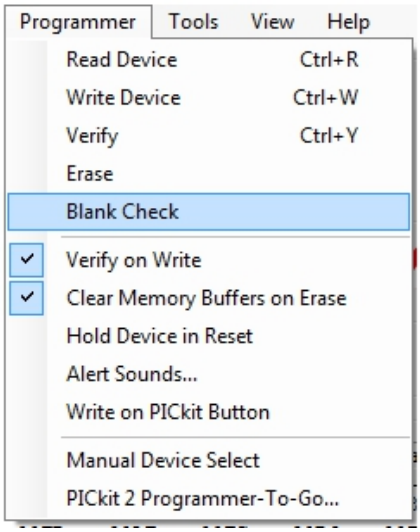


รูปที่ 2.39 หน้าจอ Import Hex File ของโปรแกรม PICkit 2

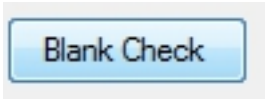
รูปที่ 2.40 รายงานผลการนำเข้าแฟ้มฐานสิบหกของโปรแกรม PICkit 2

การตรวจสอบความพร้อมโปรแกรมของชิพหรือการทำแบล็งเช็ค (Blank Check) คือ การตรวจสอบว่าชิพไมโครคอนโทรลเลอร์นั้นว่างหรือไม่ ซึ่งมีความหมายอีกนัยหนึ่งว่าชิพไมโครคอนโทรลเลอร์ได้ผ่านการโปรแกรมข้อมูลมาก่อนหรือไม่

การสั่งการเพื่อตรวจสอบความพร้อมโปรแกรมของชิพทำได้ด้วยการเลือกรายการเมนูที่เรียกว่า Blank Check (รูปที่ 2.41) ซึ่งเป็นรายการย่อยจากเมนู Programmer หรือคลิกจากปุ่ม Blank (รูปที่ 2.42) ได้โดยตรง



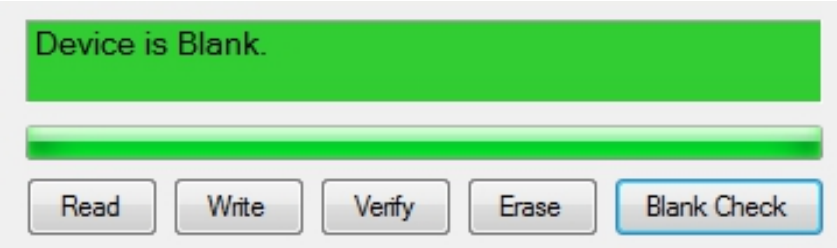
รูปที่ 2.41 เมนู Blank Check ของโปรแกรม PICkit 2



รูปที่ 2.42 ปุ่มคำสั่ง Blank Check ของโปรแกรม PICkit 2

เมื่อสั่งการตรวจสอบความพร้อมโปรแกรมของชิพ ตัวโปรแกรม PICkit 2 จะดำเนินการตรวจสอบว่าอุปกรณ์นั้นถูกโปรแกรมไปแล้วหรือไม่ (รูปที่ 2.43) และถ้าพบว่าชิพว่างเปล่า ซึ่งหมายถึง ชิพไม่ได้ถูกโปรแกรมมาก่อนหน้าที่จะดำเนินการตรวจสอบความพร้อมโปรแกรมของชิพ จะรายงานผลดังรูปที่ 2.44 แต่ถ้าชิพถูกโปรแกรมมาก่อน ต้องดำเนินการสั่งลบข้อมูลในชิพก่อนโปรแกรมเข้าไปใหม่เสมอ

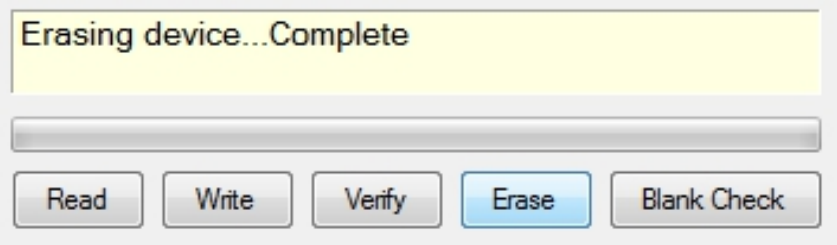
รูปที่ 2.43 หน้าจอรายงานสถานะการทำ Blank Check ของโปรแกรม PICkit 2



รูปที่ 2.44 หน้าจอรายงานผลว่าชิพพร้อมสำหรับถูกโปรแกรมของโปรแกรม PICkit 2

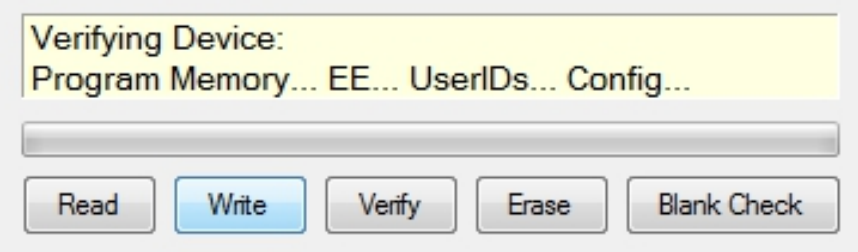
การลบ คือ การสั่งให้โปรแกรม PICkit 2 ดำเนินการลบข้อมูลที่เก็บอยู่ในชิพออกไป จึงทำให้ข้อมูลที่อยู่ก่อนหน้านี้นั้นหายไปจากชิพ และเมื่อการลบข้อมูลในชิพเสร็จสิ้นจะทำให้ชิพมีความพร้อมสำหรับการโปรแกรมชิพครั้งต่อไป นอกเสียจากว่า ชิพที่ลบนั้นหมดอายุการใช้งานจึงไม่สามารถดำเนินการลบ หรือโปรแกรมต่อไปได้อีก

ในการสั่งลบข้อมูลในชิพทำได้ด้วยการคลิกที่ปุ่ม Erase (รูปที่ 2.45) หลังจากนั้นโปรแกรม PICkit 2 จะดำเนินการสั่งลบข้อมูลในชิพ และถ้าดำเนินการสำเร็จจะรายงานผลว่า Erasing device…Complete

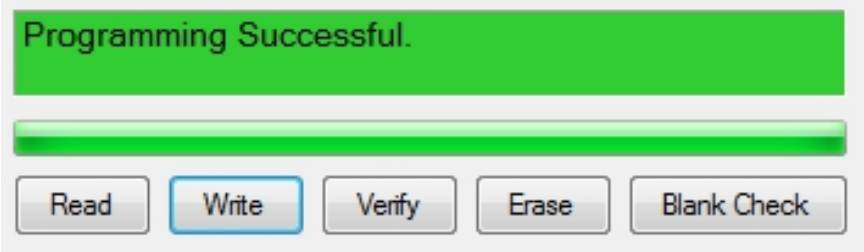


รูปที่ 2.45 หน้าจอรายงานผลการสั่งลบข้อมูลในชิพของโปรแกรม PICkit 2

การโปรแกรม คือ การสั่งเพื่อนำข้อมูลที่นำเข้าจากแฟ้มฐานสิบหกเขียนลงในตัวชิพ เพื่อนำชิพที่ถูกโปรแกรมแล้วนี้ไปใช้งานหรือทดสอบการทำงานกับบอร์ดควบคุมที่ได้ออกแบบเอาไว้ การสั่งโปรแกรมสามารถทำได้โดยการคลิกที่ปุ่ม Write หลังจากนั้นโปรแกรม PICkit 2 จะตรวจสอบสถานะของอุปกรณ์ (รูปที่ 2.46) และดำเนินการเขียนข้อมูลลงในตัวชิพ สุดท้ายเมื่อดำเนินการเขียนจนสำเร็จจะรายงานความสำเร็จในการโปรแกรมชิพให้ทราบด้วยข้อความว่า Programming Successful. (รูปที่ 2.47)



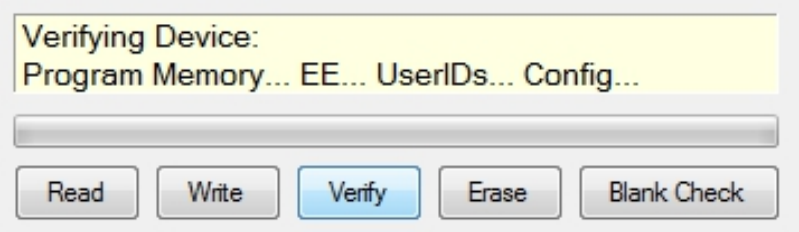
รูปที่ 2.46 หน้าจอแสดงการดำเนินการเขียนข้อมูลลงชิพของโปรแกรม PICkit 2



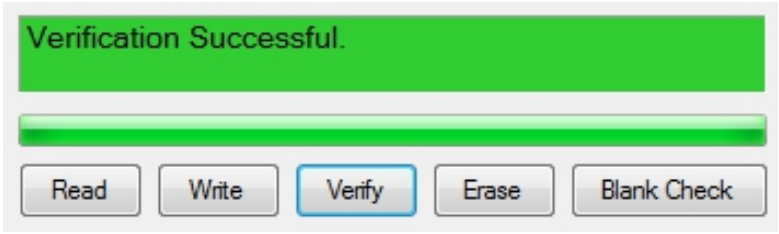
รูปที่ 2.47 หน้าจอรายงานความสำเร็จในการโปรแกรมชิพของโปรแกรม PICkit 2

การตรวจสอบความถูกต้อง คือ การสั่งให้โปรแกรม PICkit 2 ตรวจสอบว่าข้อมูลในชิพไมโครคอนโทรลเลอร์นั้นตรงกันกับข้อมูลที่นำเข้ามาจากแฟ้มฐานสิบหกหรือไม่ ซึ่งการทำการตรวจสอบความถูกต้องนี้มีความสำคัญอย่างมาก เนื่องจาก ถ้าข้อมูลในชิพไมโครคอนโทรเลอร์ไม่ตรงกับข้อมูลที่ต้องการเขียนนั้นหมายความว่าการเขียนมีความผิดพลาดและมีผลต่อการทำงานของชิพเมื่อนำชิพนี้ไปใช้งาน ด้วยเหตุนี้ ก่อนที่จะนำชิพไปใช้จริงจึงจำเป็นที่ต้องตรวจสอบความถูกต้องก่อนนำไปใช้งานเสมอ

การสั่งตรวจสอบความถูกต้องทำได้ด้วยการคลิกที่ปุ่ม Verify หลังจากนั้นโปรแกรม PICkit 2 จะดำเนินการสั่งการตรวจสอบอุปกรณ์ (รูปที่ 2.48) และถ้าพบว่าข้อมูลในชิพไมโครคอนโทรลเลอร์นั้นตรงกันกับข้อมูลที่นำเข้ามาจากแฟ้มฐานสิบหก (ข้อมูลที่สั่งเขียนลงในชิพไมโครคอนโทรลเลอร์) จะรายงานผลว่า Verification Successful. (รูปที่ 2.49) ซึ่งหมายความว่าข้อมูลจากทั้ง 2 แหล่งนั้นมีความตรงกัน



รูปที่ 2.48 หน้าจอแสดงการดำเนินการตรวจสอบความถูกต้องของโปรแกรม PICkit 2

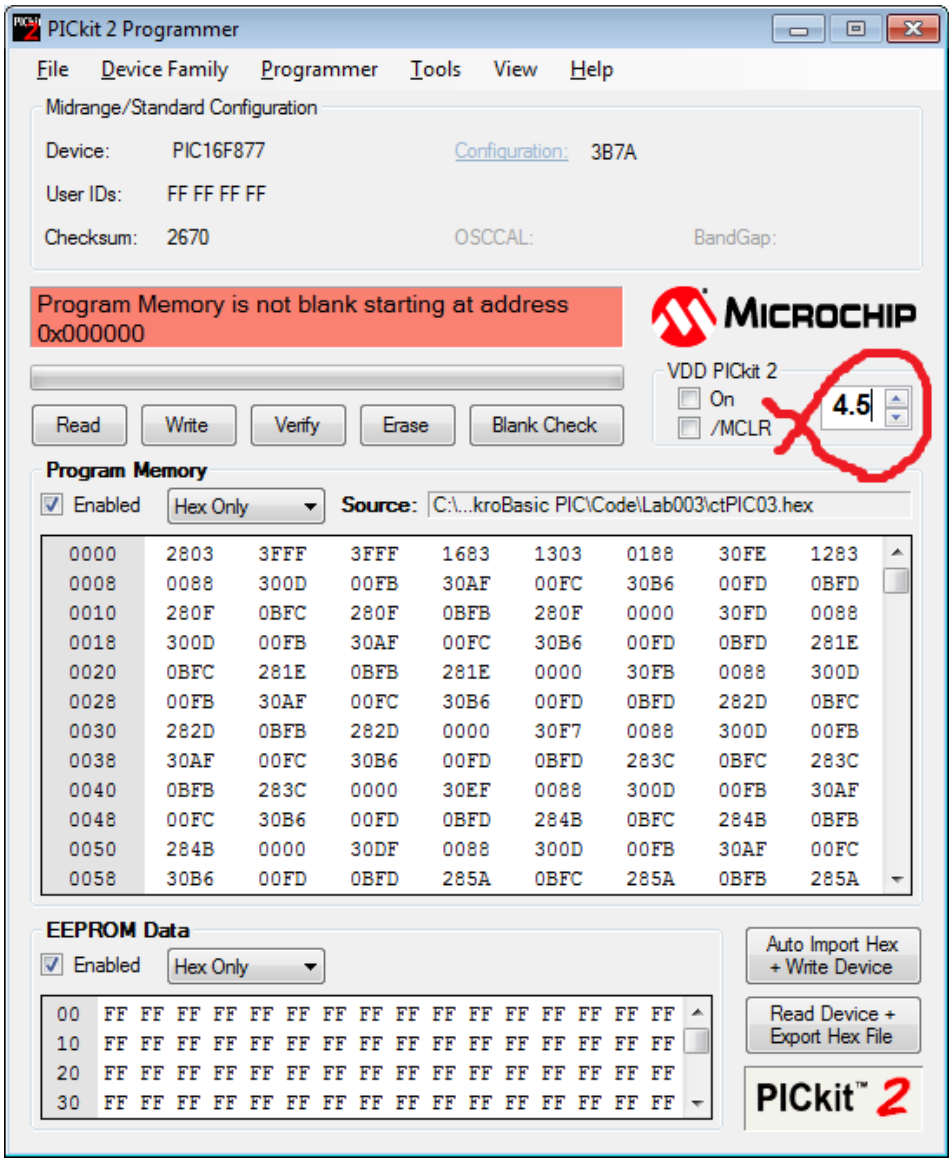


รูปที่ 2.49 หน้าจอรายงานความสำเร็จในการตรวจสอบความถูกต้องของโปรแกรม PICkit 2

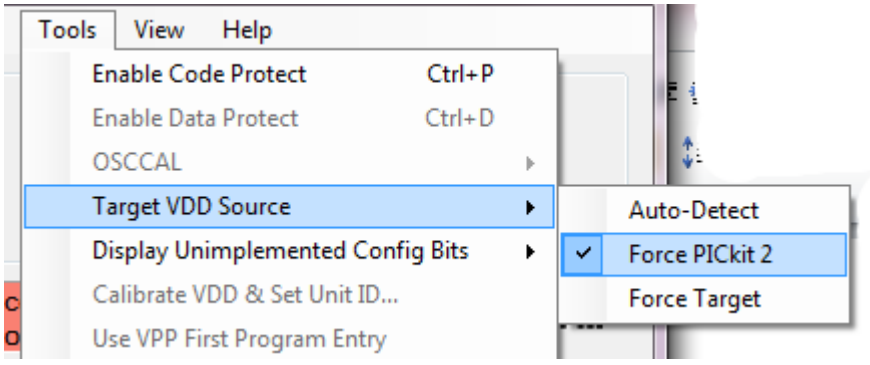
กรณีที่การตรวจสอบให้ผลที่ไม่ตรงกัน ต้องสั่งลบ แล้วโปรแกรมใหม่อีกครั้ง ถ้ายังคงไม่ตรงกันอีกให้ตรวจสอบรุ่นของชิพไมโครคอนโทรลเลอร์ว่าตรงกับโค้ดโปรแกรมที่พัฒนาขึ้นหรือไม่ ตัวอย่างเช่น ในการพัฒนาระบบระบุรุ่นของไมโครคอนโทรลเลอร์เป็น PIC16F877 แต่ใช้กับบอร์ดที่เป็นรุ่น PIC18F45 ให้ทดลองปรับรุ่นให้ตรงกัน หลังจากนั้นสั่งแปลภาษาแล้วเขียนใหม่อีกครั้ง เป็นต้น

ในบางครั้งหลังจากการแก้ไขตามที่ยกตัวอย่างอาจยังให้ผลเหมือนเดิมคือ การตรวจสอบให้ผลที่ไม่ตรงกัน ให้ทดลองเปลี่ยนชิพเป็นชิพตัวใหม่ เนื่องจากหน่วยความจำโปรแกรมของชิพนั้นหมดอายุการทำงาน แต่อย่างไรก็ดี ในบางครั้ง ถึงแม่จะขึ้นเตือนดังกล่าว ชิพก็ยังคงสามารถทำงานได้อย่างปกติ จึงคาดเดาเป็นว่า โปรแกรม PICkit 2 อาจจะมีข้อบกพร่อง หรือเกิดจากระยะเวลาที่ใช้ในการสื่อสารเพื่อตรวจสอบการตรงกันนั้นช้ากว่าที่กำหนดเอาไว้

การโปรแกรมชิพ PIC16F877 ด้วยโปรแกรม PICkit 2 จะต้องทำการกำหนดระดับแรงดัน VDD ให้เป็น 4.5 ขึ้นไปจึงจะทำให้การโปรแกรมชิพดำเนินการได้สำเร็จ ดังรูปที่ 2.50 หรือให้คลิกเลือกที่รายการเมนู Tools และเลือกรายการย่อยที่ชื่อ Target VDD Source และเลือกรายการเป็น Force PICkit 2 ดังรูปที่ 2.51 เพื่อเลือกแหล่งจ่ายไฟสำหรับเขียนชิพจากเครื่องโปรแกรมชิพแทนการใช้แหล่งจ่ายไฟจากบอร์ดทดลอง



รูปที่ 2.50 กำหนดแรงดัน VDD ในการโปรแกรมชิพด้วยโปรแกรม PICkit 2



รูปที่ 2.51 กำหนดแรงดัน VDD ในการโปรแกรมชิพให้มาจากเครื่องโปรแกรม

7. สรุป

จากบทนี้ได้ทราบถึงวิธีการใช้โปรแกรมที่ทำหน้าที่เขียนข้อมูลลงชิพไมโครคอนโทรลเลอร์ PICkit 2 ในระดับตั้งแต่การติดตั้ง การตั้งค่า การเปิดแฟ้ม การตรวจสอบความพร้อมในการโปรแกรมชิพ การลบ การโปรแกรม และการตรวจสอบความถูกต้อง พร้อมทั้งแนะนำแนวทางแก้ปัญหาที่อาจเกิดจากการใช้งานโปรแกรมในระดับเบื้องต้นที่เพียงพอสำหรับการใช้งาน

นอกจากนี้จะพบว่าการใช้เครื่องมือเขียนโปรแกรม MPLAB X IDE จะต้องอาศัยตัวแปลภาษาแยกตามประเภทของชิพไมโครคอนโทรลเลอร์ที่เลือกใช้ โดยในที่นี้เลือกใช้ MPLAB XC8 C-Compiler สำหรับหน่วยประมวลผลที่เป็นสถาปัตยกรรมแบบ 8 บิต โดยในบทนี้ได้อธิบายการติดตั้งเครื่องมือทั้ง 2 พร้อมทั้งแนะนำการสร้างโครงงาน การสร้างไฟล์ภาษาซีสำหรับเขียนโปรแกรม การเพิ่มการตั้งค่าไมโครคอนโทรลเลอร์ และการคอมไพล์เป็นการสิ้นสุดของการเขียนโปรแกรม และนำโปรแกรมไปโปรแกรมชิพด้วย PICKit2 ต่อไป