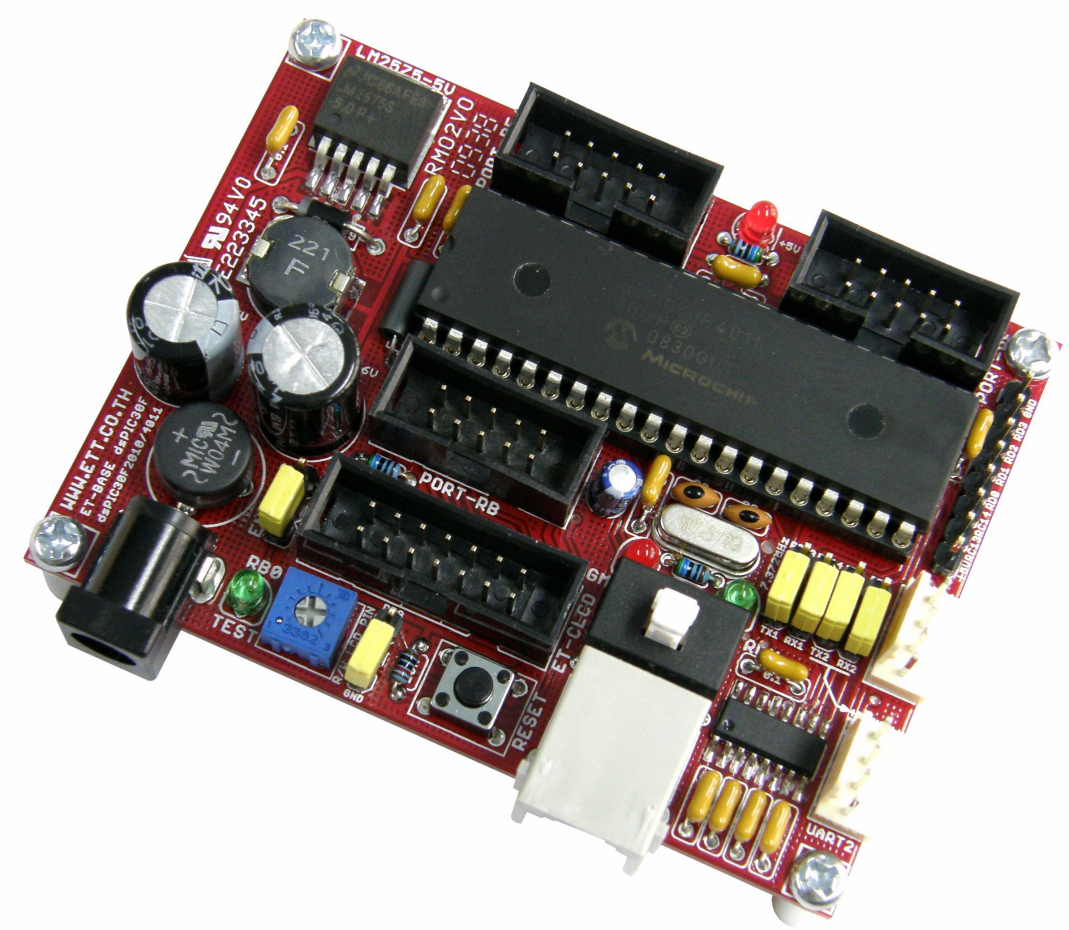
ET-BASE dsPIC30F2010/4011

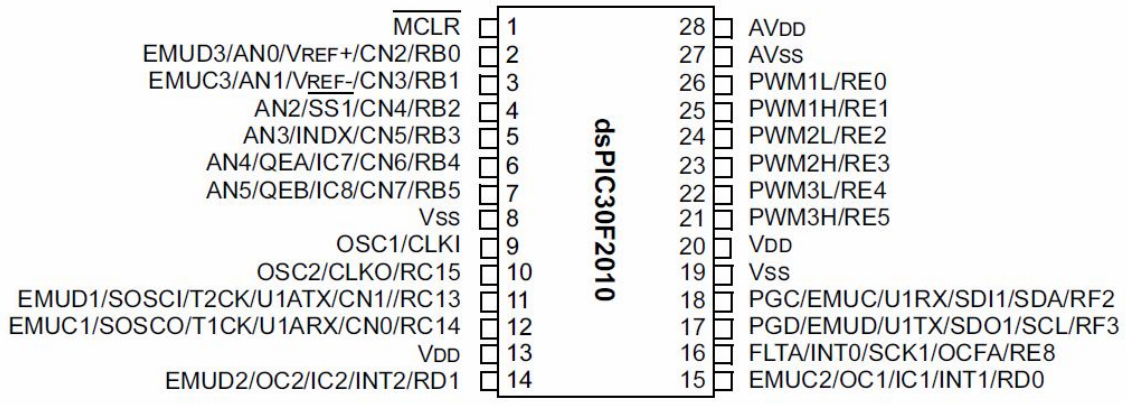


ET-BASE dsPIC30F2010/4011 เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล dsPIC30F ซึ่งเลือกใช้ไมโครคอนโทรเลอร์รุ่น 28 Pin เบอร์ dsPIC30F2010 หรือ รุ่น 40 Pin เบอร์ dsPIC30F4011 ของ Microchips เป็น MCU ประจำบอร์ด โดย dsPIC30F2010/4011 เป็น MCU ซึ่งใช้การประมวลผลข้อมูลแบบ 16 บิต จากค่าย Microchips ซึ่งมีจุดเด่นในด้านของความสามารถในการประมวลผลข้อมูลสัญญาณแบบดิจิตอลเหมาะอย่างยิ่งสำหรับนำไปประยุกต์ใช้ในงานควบคุมต่างๆ โดยโครงสร้างภายในจะเป็นการผสมผสานระหว่างไมโครคอนโทรลเลอร์ (MCU) และวงจร DSP (Digital Signal Processing) รวมเข้าไว้ด้วยกัน หรืออาจเรียก MCU ตระกูล dsPIC30F ว่าเป็น DSC หรือ Digital Signal Controller ก็ได้

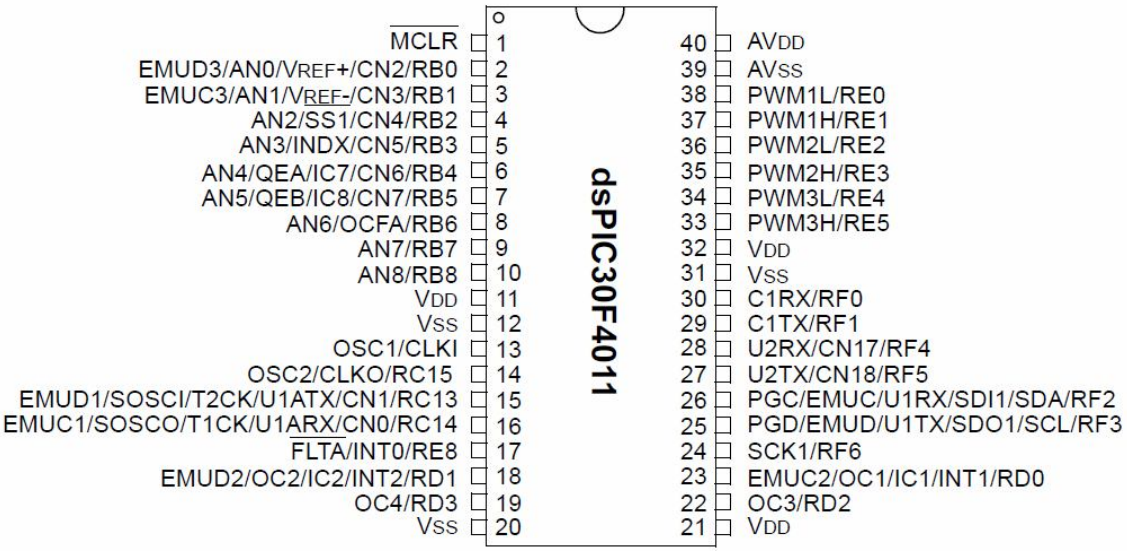
โดยโครงสร้างของบอร์ด ET-BASE dsPIC30F2010/4011 ได้รับการออกแบบให้บอร์ดมีขนาดเล็กเหมาะต่อการนำไปประยุกต์ใช้งานเป็นหลัก โดยภายในบอร์ดได้บรรจุเอาวงจรที่จำเป็นต่อการใช้งาน และสะดวกต่อการพัฒนาโปรแกรม มีความยืดหยุ่น สามารถปรับเปลี่ยนสัญญาณ I/O เพื่อนำไปประยุกต์ใช้งานในลักษณะต่างๆให้สอดคล้องและเหมาะสมกับความต้องการใช้งานได้ในหลายๆลักษณะตามต้องการ

คุณสมบัติของบอร์ด

- เลือกใช้ MCU ตระกูล dsPIC30F2010 หรือ dsPIC30F4011 ของ Microchips เป็น MCU ประจำบอร์ด โดยคุณสมบัติเด่นๆของ MCU ได้แก่
  - มีหน่วยความจำ Flash 12KByte(dsPIC20F2010) หรือ 48KByte(dsPIC30F4011)
  - มีหน่วยความจำ RAM ขนาด 512Byte(dsPIC30F2010) หรือ 2KByte(dsPIC30F4011)
  - มีหน่วยความจำ EEPROM ขนาด 1KByte สำหรับเก็บข้อมูลใช้งาน
  - มีพอร์ต I/O ขนาด 19 Bit(dsPIC30F2010) หรือ 29 Bit(dsPIC30F4011)
  - มี 16Bit Timer/Counter จำนวน 3 ชุด(dsPIC30F2010) หรือ 5 ชุด(dsPIC30F4011)
  - มี Input Capture จำนวน 4 ช่อง
  - มี Output Compare จำนวน 2 ช่อง(dsPIC30F2010) หรือ 4 ช่อง(dsPIC30F4011)
  - มี ADC 10Bit/500Ksps จำนวน 6 ช่อง(dsPIC30F2010) หรือ 9 ช่อง(dsPIC30F4011)
  - มี PWM Motor Control จำนวน 6 ช่อง พร้อม Quadrature Encode Interface(QEI)
  - มี UART จำนวน 1 ช่อง(dsPIC30F2010) หรือ 2 ช่อง (dsPIC30F4011)
  - มี SPI จำนวน 1 ช่อง และ มี I2C จำนวน 1 ช่อง
  - มีวงจร Watchdog, Power-ON Reset, PWM
- ใช้ Crystal ความถี่ 7.3728MHz สามารถใช้ PLL คูณความถี่เพื่อ Run ความถี่ 29.4912MHz ได้
- มีพอร์ตสื่อสารอนุกรม UART แบบ RS232 จำนวน 1 ช่อง สำหรับ dsPIC30F2010 และ 2 ช่อง สำหรับ dsPIC30F4011 พร้อม Jumper สำหรับเลือกใช้งาน UART หรือ GPIO ได้ตามต้องการ โดยใช้ขั้วต่อ UART แบบ CPA-4 Pin มาตรฐาน อีทีที
- มีขั้ว ICSP มาตรฐาน ICD2 แบบ RJ11 สำหรับใช้ร่วมกับชุดพัฒนาโปรแกรมและ Debugger ที่รองรับการทำงานตามมาตรฐาน ICD2 ของ Microchips เช่น ICD2 หรือ Pickit2 ได้
- มี Switch สำหรับสลับสัญญาณระหว่าง Program/Debug(PGM) และ ใช้งานปรกติ(RUN) พร้อม LED แสดงโหมดการทำงานของบอร์ด
- มีขั้วต่อสัญญาณ I/O แบบ Header ขนาด 2x5 จำนวน 3 ชุด และ Header 1x8 Pin อีก 1 ชุด
- Header 14Pin สำหรับ Character LCD พร้อม VR ปรับความสว่าง
- มี Switch Reset สำหรับสั่ง Reset การทำงานของ MCU ภายในบอร์ด
- มี LED สำหรับทดสอบการทำงาน โดยใช้ RB0 ในการควบคุม พร้อม Jumper ตัดต่อสัญญาณ
- Power AC/DC Input พร้อม Regulate แบบ Switching เบอร์ LM2575 ขนาด 5V/1A ลดปัญหาความร้อนจากวงจร Regulate และ LED แสดงสถานะแหล่งจ่าย Power
- ขนาด PCB Size เล็กเพียง 8 x 6 cm.



รูปแสดงการจัดขาสัญญาณของ **dsPIC30F2010**

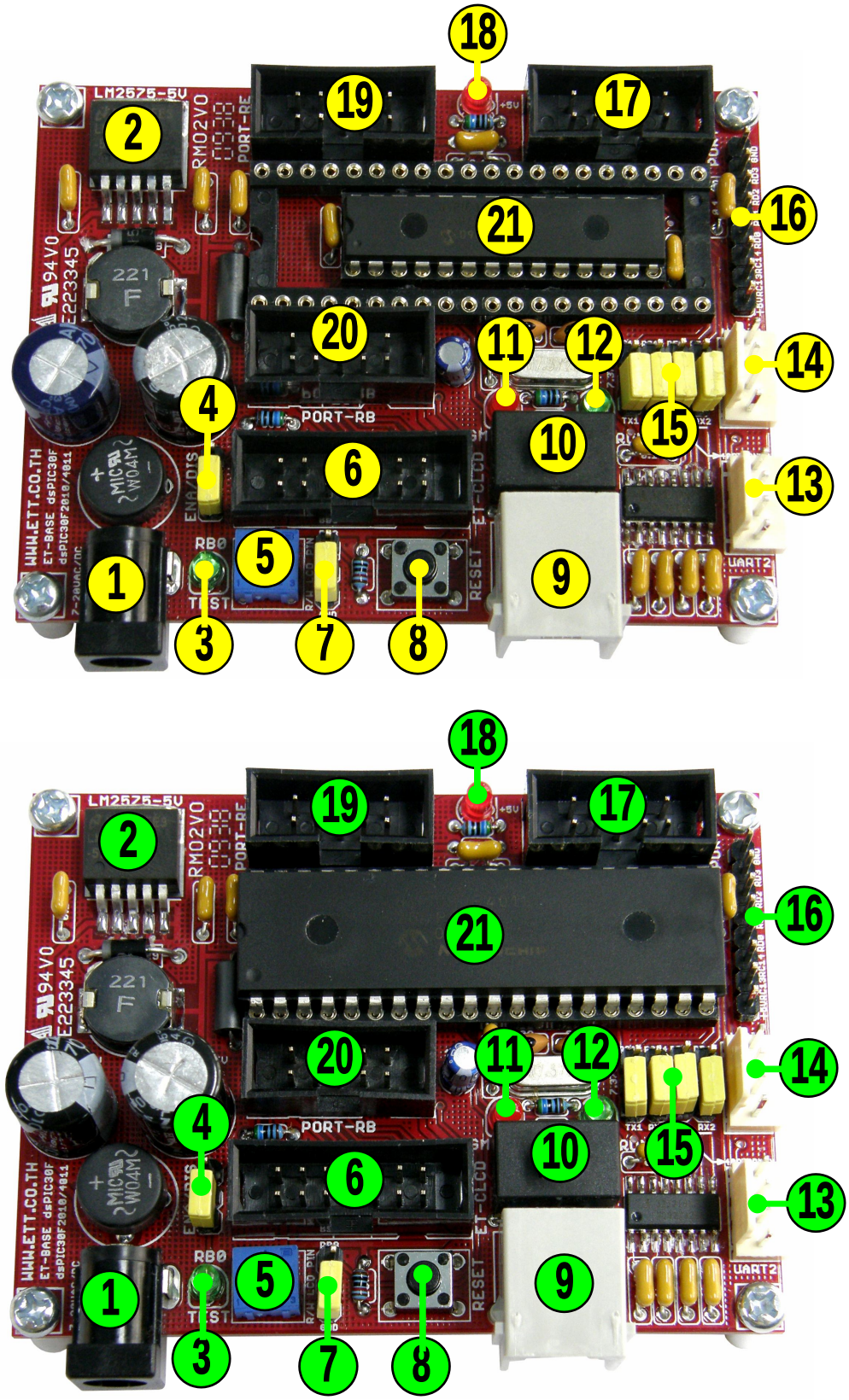


รูปแสดงการจัดขาสัญญาณของ **dsPIC30F4011**

Device	Pins	Program Mem. Bytes/ Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	A/D 10-bit 500 Ksps	Quad Enc	UART	SPI™	I²C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	–
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	–
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	–
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

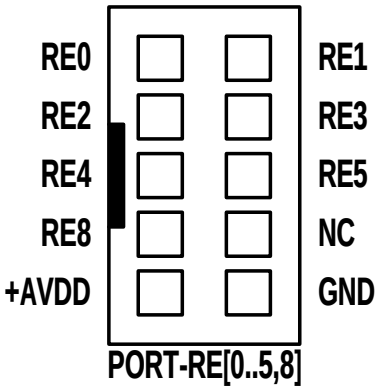
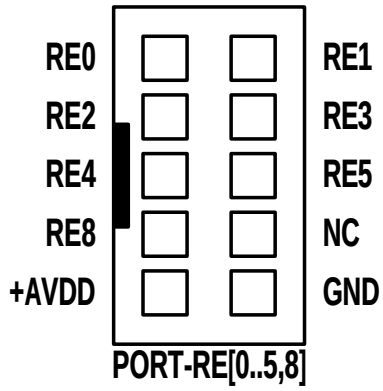
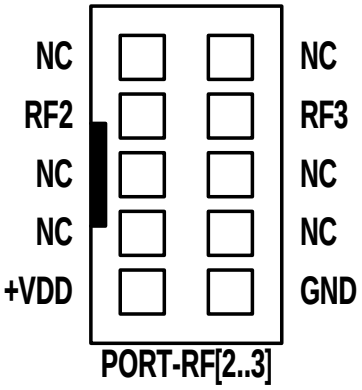
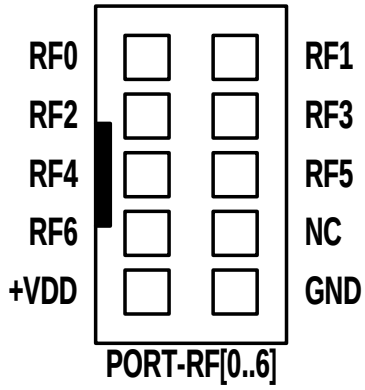
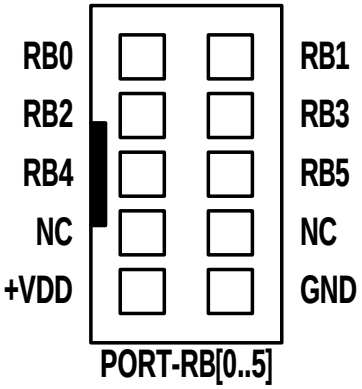
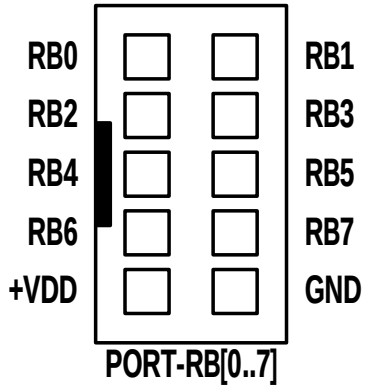
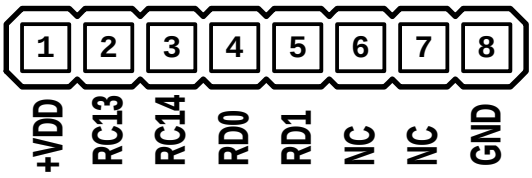
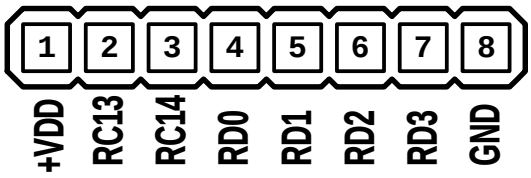
ตารางแสดง คุณสมบัติความแตกต่างของ **dsPIC30F** เบอร์ต่าง ๆ

โครงสร้างบอร์ด ET-BASE dsPIC30F2010/4011



- **หมายเลข 1** คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟ 7-20VAC/DC
- **หมายเลข 2** คือ IC Regulate แบบ Switching ขนาด 5V/1A
- **หมายเลข 3** คือ LED TEST สำหรับทดสอบการทำงานของบอร์ด โดยควบคุมจาก RB0
- **หมายเลข 4** คือ Jumper สำหรับ ตัด ต่อ สัญญาณ RB0 กับ LED TEST
- **หมายเลข 5** คือ VR ปรับค่า สำหรับใช้ปรับความสว่างของหน้าจอแสดงผล LCD
- **หมายเลข 6** คือ ขั้วต่อ 14PIN IDE สำหรับเชื่อมต่อกับ LCD แบบ Character
- **หมายเลข 7** คือ Jumper สำหรับเลือกรูปแบบการควบคุมขา R/W ของ LCD โดยถ้าใช้ MCU รุ่น 28Pin ต้องเลือกไว้ด้าน GND เสมอและไม่สามารถสั่งอ่านข้อมูลจาก LCD ได้
- **หมายเลข 8** คือ สวิตช์ Reset สำหรับ Reset การทำงานของ MCU เมื่ออยู่ในโหมด Run
- **หมายเลข 9** คือ ขั้วต่อ ICD2 สำหรับใช้เชื่อมต่อกับเครื่องโปรแกรมและดีบักตามมาตรฐาน ICD2
- **หมายเลข 10** คือ สวิตช์ สำหรับเลือกโหมดการทำงานระหว่าง Run(RUN) และ Program(PGM)
- **หมายเลข 11** คือ LED สีแดง แสดงสถานะ PGM เมื่อบอร์ดทำงานใน Program Mode
- **หมายเลข 12** คือ LED สีเขียว แสดงสถานะ RUN เมื่อบอร์ดทำงานใน Run Mode
- **หมายเลข 13** คือ ขั้วต่อ UART2 ซึ่งมีเฉพาะใน MCU รุ่น 40 Pin (dsPIC30F4011) เท่านั้น โดยเป็นสัญญาณแบบ RS232 โดยใช้ Pin ของ RF4(RX2) และ RF5(TX2) เป็นสัญญาณเชื่อมต่อ
- **หมายเลข 14** คือ คือ ขั้วต่อ UART1 โดยเป็นสัญญาณแบบ RS232 มีอยู่ใน MCU ทั้งรุ่น 28 Pin และรุ่น 40 Pin ซึ่งใช้ Pin ของ RC13(TX1),RC14(RX1) เป็นสัญญาณเชื่อมต่อ
- **หมายเลข 15** คือ Jumper สำหรับเลือกการเชื่อมต่อสัญญาณ RC13,RC14,RF4,RF5 ว่าจะใช้ขาสัญญาณดังกล่าวทำหน้าที่เป็นขาสัญญาณรับส่งของ RS232 หรือ GPIO สำหรับใช้งานทั่วไป
- **หมายเลข 16** คือ ขั้วต่อสัญญาณ RC13,RC14,RD0,RD1,RD2 และ RD3 สำหรับใช้งาน โดยถ้าเป็น MCU รุ่น 28Pin จะไม่มีสัญญาณ RD2 และ RD3 ขาสัญญาณดังกล่าวจะปล่อยว่างไว้
- **หมายเลข 17** คือ ขั้วต่อสัญญาณ Port-RF ซึ่งถ้าเป็น MCU รุ่น 40 Pin จะมี 7 บิต คือ RF[0..6] แต่ถ้าเป็น MCU รุ่น 28 Pin จะมีเพียง 2 บิต คือ RF[2] และ RF[3] เท่านั้น
- **หมายเลข 18** คือ LED สำหรับแสดงสถานะ ของแหล่งจ่ายไฟ +5V ของบอร์ด
- **หมายเลข 19** คือ ขั้วต่อสัญญาณ Port-RE ซึ่งจะมี 7 บิต คือ RE[0..6 และ 8]
- **หมายเลข 20** คือ ขั้วต่อสัญญาณ Port-RB ซึ่งถ้าเป็น MCU40Pin จะมี 8 บิต คือ RB[0..7] แต่ถ้าเป็น MCU 28Pin จะมีเพียง 6 บิต คือ RB[0..5] เท่านั้น
- **หมายเลข 21** คือ MCU ประจำบอร์ด โดยถ้าเป็น รุ่น 28Pin จะใช้เบอร์ dsPIC30F2010 แต่ถ้าเป็นรุ่น 40Pin จะใช้เบอร์ dsPIC30F4011

ขั้วต่อสัญญาณต่าง ๆ

ET-BASE dsPIC30F2010	ET-BASE dsPIC30F4011
	
	
	
	

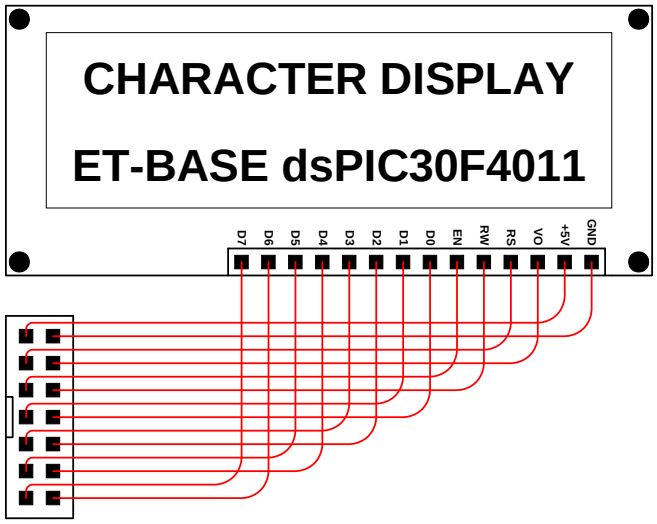
รูปแสดง การจัดเรียงสัญญาณของพอร์ต **I/O** ต่าง ๆของบอร์ด **ET-BASE dsPIC30F2010/4011**

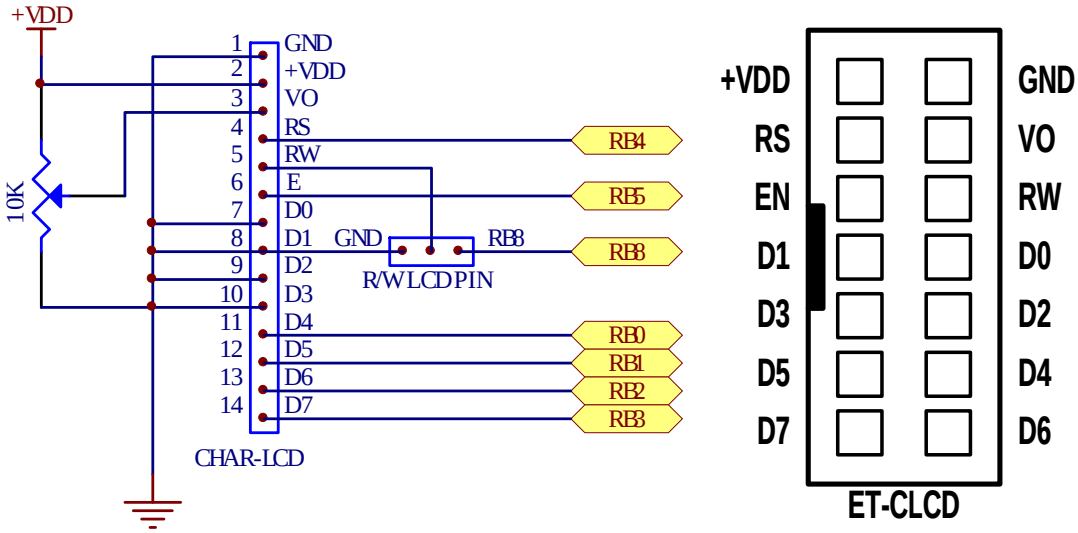
การใช้งาน LCD แสดงผล

สำหรับการเชื่อมต่อ LCD นั้นจะสามารถใช้ได้กับ LCD แบบ Character Dot-Matrix เท่านั้น โดยเชื่อมต่อแบบ 4 บิต Data โดยสัญญาณที่ใช้เชื่อมต่อกับ LCD จะใช้สัญญาณจาก RB[0..5] จำนวน 6บิต ซึ่งถ้าใช้ dsPIC30F4011 สามารถเลือกกำหนดให้ใช้สัญญาณ RB8 เพื่อกำหนดการอ่านข้อมูลกลับจาก LCD ได้ด้วย โดยการเลือก Jumper R/W LCD PIN ไว้ทางด้าน RB8 หรือเลือกไว้ทางด้าน GND เมื่อไม่ต้องการอ่านข้อมูลกลับจาก LCD ได้ตามต้องการ แต่สำหรับกรณีที่ใช้ dsPIC30F2010 จะไม่มีขาสัญญาณ RB8 จึงไม่สามารถสั่งอ่านข้อมูลกลับจาก LCD ได้ และต้องเลือก Jumper ไว้ทางด้าน GND เสมอ โดยในการเชื่อมต่อสายสัญญาณจากขั้วต่อของ พอร์ต LCD ไปยังจอแสดงผล LCD นั้น ให้ยึดชื่อสัญญาณเป็นจุดอ้างอิง โดยให้ต่อสัญญาณที่มีชื่อตรงกันเข้าด้วยกันให้ครบทั้ง 14 เส้น ดังรูป

สัญญาณการเชื่อมต่อกับ LCD

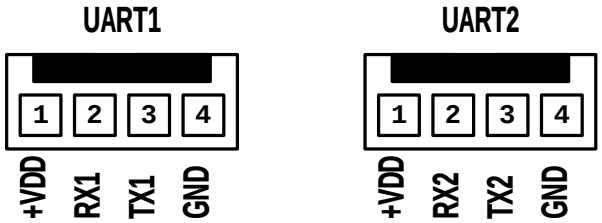
- RS = RB4
- EN = RB5
- RW = GND/RB8
- DB4 = RB0
- DB5 = RB1
- DB6 = RB2
- DB7 = RB3



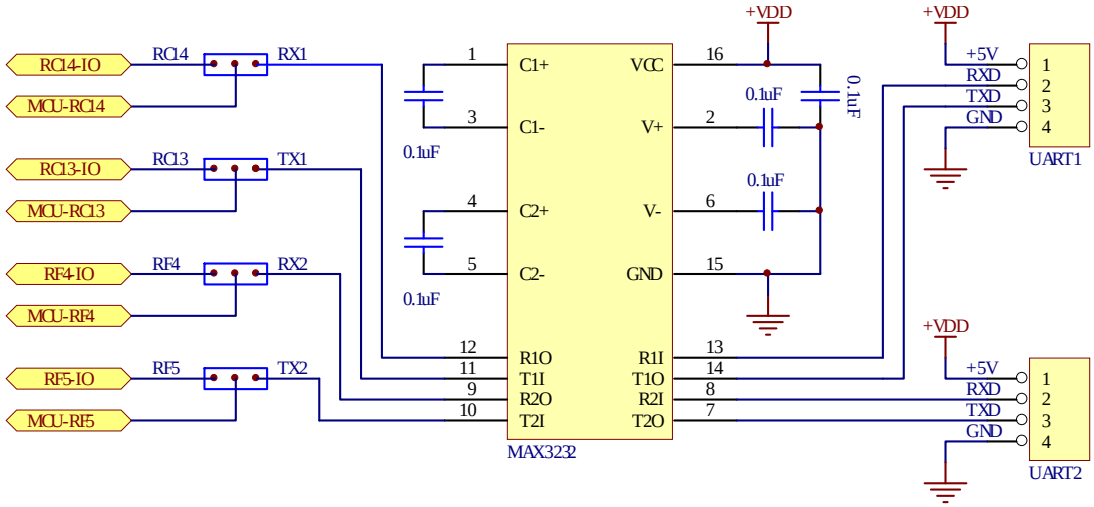


การใช้งาน RS232

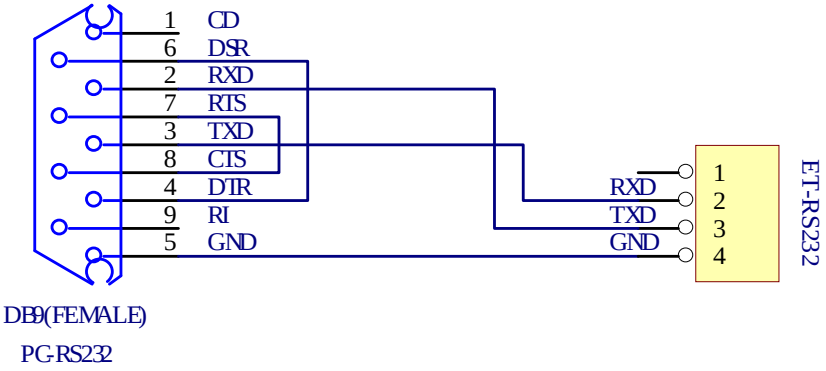
พอร์ต RS232 เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณจาก MAX3232 เรียบร้อยแล้ว โดยถ้าใช้ dsPIC30F2010 จะมี UART จำนวน 1 ช่อง แต่ถ้าใช้ dsPIC30F4011 จะมีวงจร UART ใช้งาน จำนวน 2 ช่อง โดยสัญญาณของ RS232 แต่ละช่อง จะจัดขั้วเป็นแบบ CPA-4PIN (RS232) ดังรูป



โดยวงจรการทำงานของ UART(RS232) ทั้ง 2 ช่อง สามารถเลือกใช้งาน หรือ ไม่ใช้งาน จาก Jumper ได้ เพื่อใช้เลือกว่าจะให้สัญญาณของ MCU ทำหน้าที่เป็น I/O หรือ UART ดังวงจร



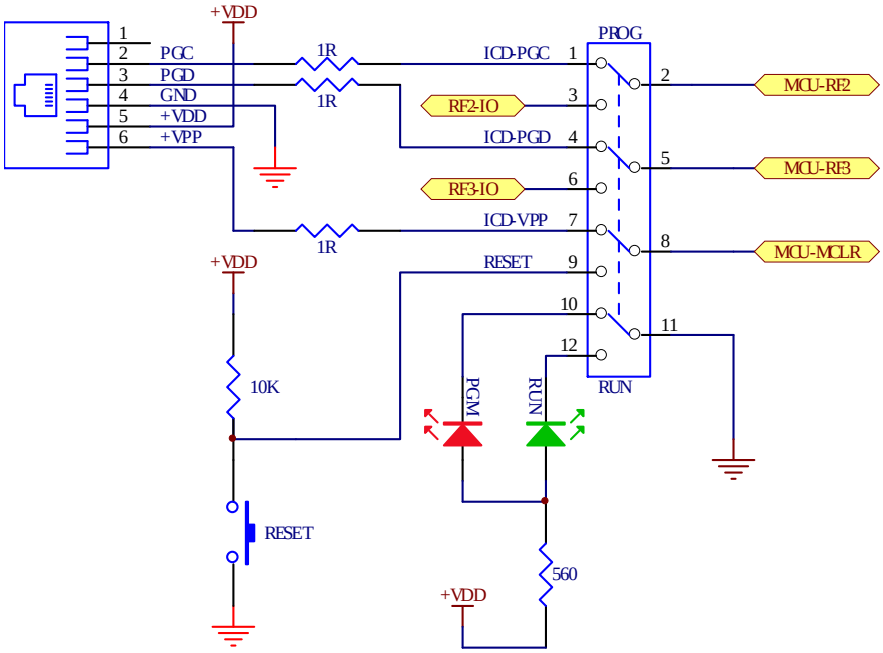
สำหรับ Cable ที่จะใช้ในการเชื่อมต่อ RS232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้วต่อ RS232 ของบอร์ด ET-BASE dsPIC30F2010/4011 นั้น เป็นดังนี้

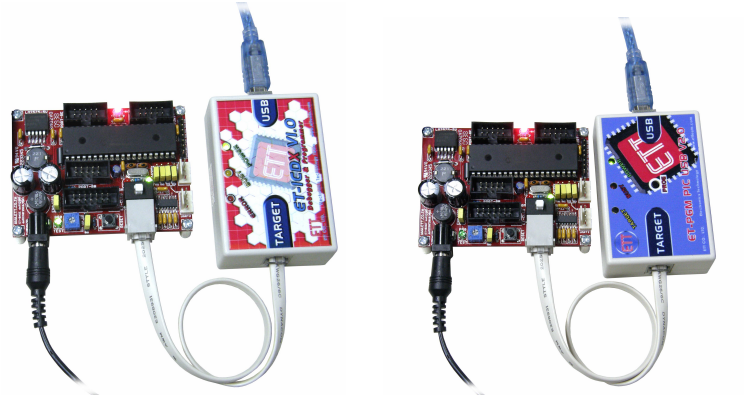


รูป แสดงวงจรสาย Cable สำหรับ RS232

การใช้งาน ICD2

ICD2 จะเป็น Connector แบบ RJ11 สำหรับ Interface กับเครื่องมือพัฒนาโปรแกรมตระกูล PIC ที่มีการจัดขั้วตามมาตรฐาน ICD2 ของ Microchips เช่น ICD2,ICD3,Pickit2 หรือ Pickit3 ซึ่งสามารถใช้งานได้กับเครื่องมือพัฒนาของ Microchips หรือ เทียบเท่า เช่น ET-PGMPIC USB(เทียบเท่า Pickit2) หรือ ET-ICDX(เทียบเท่า ICD2) โดยจะมีสวิตช์สำหรับเลือกตัดต่อสัญญาณของ RF2, RF3 และ MCLR สำหรับใช้ทำหน้าที่เชื่อมต่อกับ Programmer/Debugger หรือ ใช้งานตามปรกติได้ พร้อม LED แสดงสถานะ ว่าการทำงานของสวิตช์อยู่ในตำแหน่งใด โดยถ้าเลือกสวิตช์ไว้ทางด้าน Programmer/Debugger จะเห็น LED สีแดงของ PGM ติดสว่างให้เห็น แต่ถ้าตำแหน่งของสวิตช์อยู่ด้าน Run จะเห็น LED สีเขียว(RUN) ติดสว่างให้เห็น โดยมีการจัดวงจรและสัญญาณตามมาตรฐานของ ICD2 ดังนี้

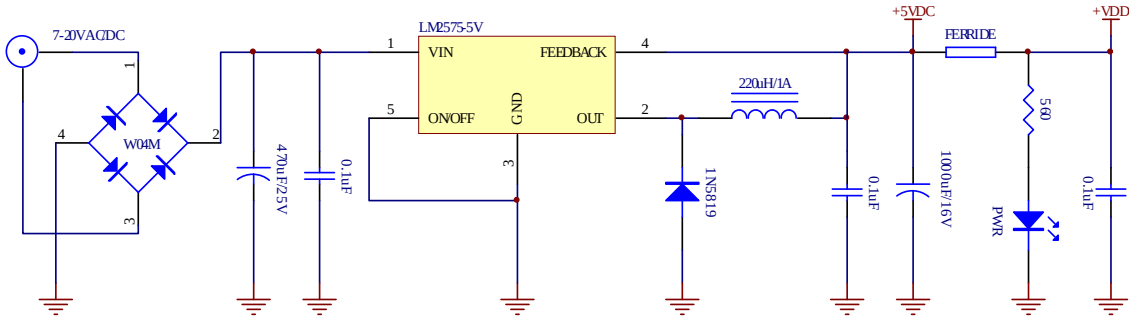




รูปแสดง การต่อบอร์ดกับเครื่องโปรแกรม ET-ICDX(ซ้าย) และ ET-PGM PIC USB(ขวา)

วงจรภาคจ่ายไฟ

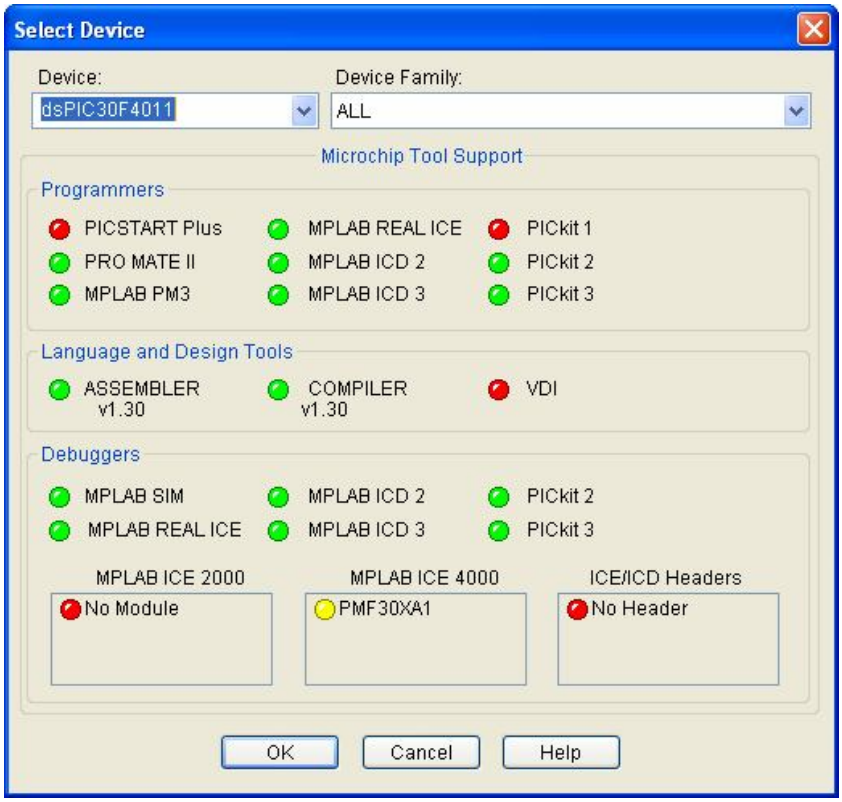
สำหรับวงจรภาคจ่ายไฟของบอร์ด จะใช้วงจร Bridge Rectifier ร่วมกับ Switching Regulate ขนาด 5V/1A สามารถใช้งานได้ทั้งกับไฟ AC และ DC ขนาดตั้งแต่ 7V-20V ได้



การพัฒนาโปรแกรมของบอร์ด

สำหรับการพัฒนาโปรแกรมของบอร์ด “ET-BASE dsPIC30F” นั้น จะแบ่งออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของการพัฒนาโปรแกรม และการ Download โปรแกรม โดยในส่วนของการพัฒนาโปรแกรมของบอร์ดนั้น จะเป็นการเขียนโปรแกรมเพื่อควบคุมและสั่งงานให้ MCU สามารถทำงานตามจุดประสงค์ที่ได้ออกแบบไว้ ซึ่งในส่วนนี้เป็นหน้าที่ของผู้พัฒนาโปรแกรม ว่าจะเลือกใช้ภาษาใดในการพัฒนาโปรแกรม 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-BASE dsPIC30F2010” และ “ET-BASE dsPIC30F4011” นั้น ถูกออกแบบให้ใช้กับเครื่องมือพัฒนาโปรแกรมที่มีคุณสมบัติตรงตามมาตรฐาน ICD2 ของ Microchips ซึ่งจะมีความสะดวกมากในการพัฒนาโปรแกรมร่วมกับชุดโปรแกรม MPLAB และ C30 ของ Microchips เพราะสามารถเขียนโปรแกรม และ Download Code เพื่อทดสอบการทำงาน รวมทั้งการตรวจสอบหาข้อผิดพลาดต่างๆได้โดยง่ายและสะดวก ซึ่งสามารถใช้งานได้กับเครื่องมือของ Microchips หรือเทียบเท่าได้หลายรุ่นดังรูป (รุ่นเครื่องมือที่รองรับคือรุ่นที่มีสัญลักษณ์สีเขียวหน้าชื่อ)



การเขียนโปรแกรมใช้งานกับบอร์ดโดยใช้ MPLAB C30

MPLAB C30 หรือ C30 Tools เป็นโปรแกรมภาษาซี สำหรับใช้แปลคำสั่งของ MCU ตระกูล dsPIC ซึ่งได้รับการพัฒนาขึ้นโดย Microchips เอง โดยข้อกำหนดและรายละเอียดของการเขียนโปรแกรมภาษาซีนั้น จะไม่กล่าวถึงในที่นี้ด้วย โดยถ้าผู้ใช้ต้องการพัฒนาโปรแกรมให้กับ dsPIC ด้วยภาษาซี แต่ยังไม่มีความรู้เรื่องการเขียนโปรแกรมภาษาซีเลยนั้นขอแนะนำให้ หาหนังสือที่อธิบายเกี่ยวกับการเขียนโปรแกรมภาษาซีในส่วนที่เป็นมาตรฐานตามข้อกำหนดของ “ANSI C” มาศึกษาให้เข้าใจเสียก่อน และสำหรับส่วนของข้อกำหนดปลีกย่อยอื่นๆที่เป็นของ MPLAB C30 เองก็สามารถอ่านเพิ่มเติมได้จากเอกสารและคู่มือการใช้งานของ MPLAB C30 ที่ทาง Microchips จัดทำไว้ได้ โดยสามารถ Download จาก Website ของ Microchips หรือจาก Folder ของ “..\MPLAB C30\docs” ที่ทำการติดตั้งโปรแกรม MPLAB C30 ไว้ก็ได้ โดยในที่นี้ จะขอกล่าวแนะนำถึงเฉพาะส่วนของการกำหนดค่าตัวเลือกในโปรแกรมเพื่อใช้งานร่วมกับบอร์ด “ET-BASE dsPIC30F2010/4011” เท่านั้น โดยในการที่จะใช้งานโปรแกรม MPLAB C30 ในการเขียนโปรแกรมนั้น ผู้ใช้จำเป็นต้องทำการติดตั้งโปรแกรมของ Microchips จำนวน 2 โปรแกรมดังนี้คือ

- MPLAB IDE ซึ่งเป็นโปรแกรม Text Editor ของ Microchips ซึ่งในปัจจุบัน (ตุลาคม 2552) จะเป็นรุ่น 8.40 แล้วสามารถ Download มาใช้งานได้ฟรีจาก Web ของ Microchips
- MPLAB C30 ซึ่งเป็นตัวแปลภาษาซี (C Complier) ให้เป็นรหัสคำสั่งของ dsPIC โดยในปัจจุบัน(ตุลาคม 2552) จะเป็น Version 3.20B ซึ่งตามปรกติแล้วโปรแกรมชุดนี้จะต้องซื้อมาใช้งานเอง แต่อย่างไรก็ตามทาง Microchips เองมีรุ่นทดลองใช้งานให้ผู้ใช้สามารถ Download มาใช้งานได้เช่นเดียวกันกับ MPLAB IDE

โดยโปรแกรมทั้ง 2 ชุดนี้ ทางอีทีที ได้ทำการ Download มาจัดเตรียมไว้ให้ในแผ่น CD-ROM ที่แถมไปกับบอร์ดของ “ET-BASE dsPIC30F2010/4011” ด้วยอยู่แล้ว โดยในการติดตั้งโปรแกรมนั้นขอแนะนำให้ผู้ใช้ทำการติดตั้งโปรแกรมในชุดของ MPLAB IDE ก่อนเป็นอันดับแรก โดยขอแนะนำให้ติดตั้งโปรแกรมของ MPLAB IDE ไว้ตามค่า Default ของโปรแกรมติดตั้งเลย คือ “C:\Program Files\Microchip\MPLAB IDE\" จะสะดวกต่อการใช้งานมากกว่า ซึ่งหลังจากทำการติดตั้งโปรแกรม MPLAB IDE เสร็จเรียบร้อยแล้วในครั้งแรกก่อนการใช้งานนั้นต้องสั่ง Restart เครื่องคอมพิวเตอร์ก่อน หลังจากนั้นแล้ว MPLAB IDE จึงจะสามารถทำงานได้โดยไม่เกิดปัญหา จากนั้นจึงทำการติดตั้งโปรแกรม MPLAB C30 เป็นลำดับถัดไป โดยขอแนะนำให้ทำการติดตั้งโปรแกรมชุดนี้ไว้ตามค่า Default คือ C:\Program Files\Microchip\MPLAB C30\" จะเกิดความสะดวกต่อการใช้งานมากที่สุด โดยเฉพาะในขั้นตอนของการกำหนดการเชื่อมโยงการทำงานระหว่าง MPLAB IDE และ MPLAB C30 โดยในที่นี้จะขออธิบายโดยอ้างถึงตำแหน่งการติดตั้งโปรแกรมดังที่กล่าวไว้แล้วข้างต้นเท่านั้น ซึ่งถ้าผู้ใช้ทำการสั่งติดตั้งโปรแกรมไว้ยังตำแหน่ง Folder ที่แตกต่างไปจากนี้แล้วข้อให้ทำความเข้าใจและดัดแปลงวิธีการกำหนดค่าเองตามที่ติดตั้งโปรแกรมไว้จริงๆด้วย

การกำหนดการเชื่อมโยงการทำงานของ MPLAB IDE และ MPLAB C30

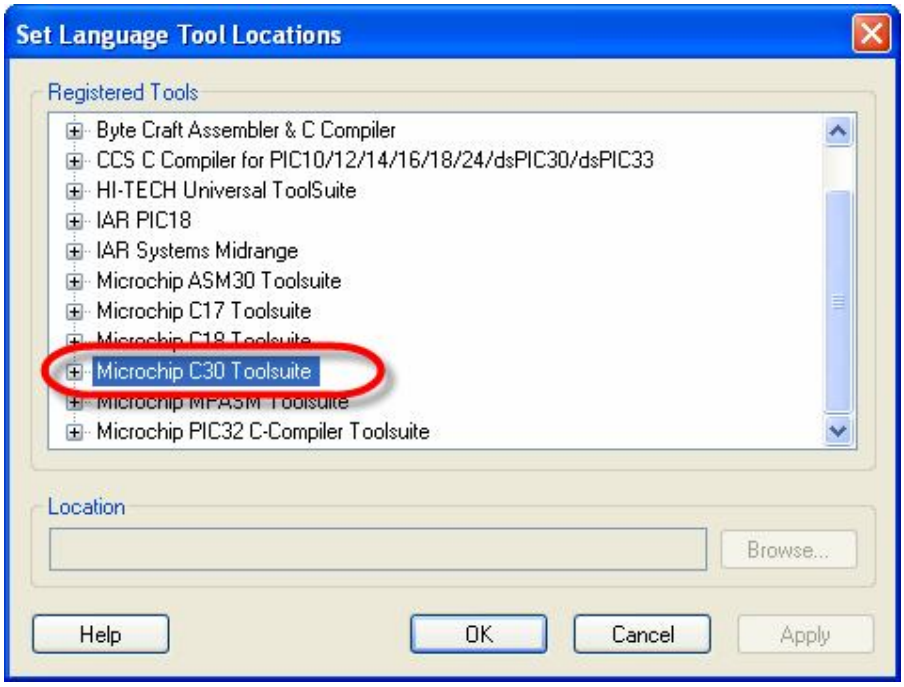
หลังจากทำการติดตั้งโปรแกรม MPLAB IDE และ MPLAB C30 เสร็จเรียบร้อยแล้ว ก่อนที่จะเริ่มต้นใช้งานโปรแกรม MPLAB C30 ได้นั้น ในครั้งแรกจะต้องทำการสั่งกำหนดการเชื่อมโยงการทำงานระหว่าง MPLAB IDE และ MPLAB C30 ให้ถูกต้องเรียบร้อยเสียก่อน จึงจะสามารถเรียกใช้งานโปรแกรม MPLAB C30 ผ่านทางโปรแกรม MPLAB IDE ได้อย่างถูกต้อง ทั้งนี้ก็เนื่องมาจากว่า โปรแกรม MPLAB C30 นั้นจะเป็นเพียงตัวแปลคำสั่ง Text File ที่เป็นภาษาซี (รวมทั้งภาษา Assembly) ให้เป็นรหัสคำสั่งของ dsPIC ในรูปแบบของ Hex File เพียงอย่างเดียวเท่านั้น ส่วนการเขียนโปรแกรม Source Code นั้นจะอาศัยโปรแกรม MPLAB IDE เป็นหลัก ซึ่งการสั่งแปลคำสั่งก็จะต้องกระทำผ่านเมนูคำสั่งของ MPLAB IDE ด้วยเช่นเดียวกัน ซึ่งตามปรกติแล้ว MPLAB IDE สามารถเชื่อมโยงการทำงานร่วมกับโปรแกรมอื่นๆได้อีกหลายโปรแกรม ไม่ได้ใช้งานเฉพาะกับ MPLAB C30 เพียงอย่างเดียวเท่านั้น

โดยในชุดโปรแกรมของ MPLAB C30 หลังจากติดตั้งโปรแกรมไปแล้ว โปรแกรม ใช้งานต่างๆจะถูกเก็บไว้ใน Folder ชื่อ “C:\Program Files\Microchip\MPLAB C30\bin\” โดยจะมีโปรแกรมหลักๆที่ต้องกำหนดการเชื่อมโยงการทำงานกับ MPLAB IDE อยู่ด้วยกัน 4 โปรแกรมด้วยกันคือ

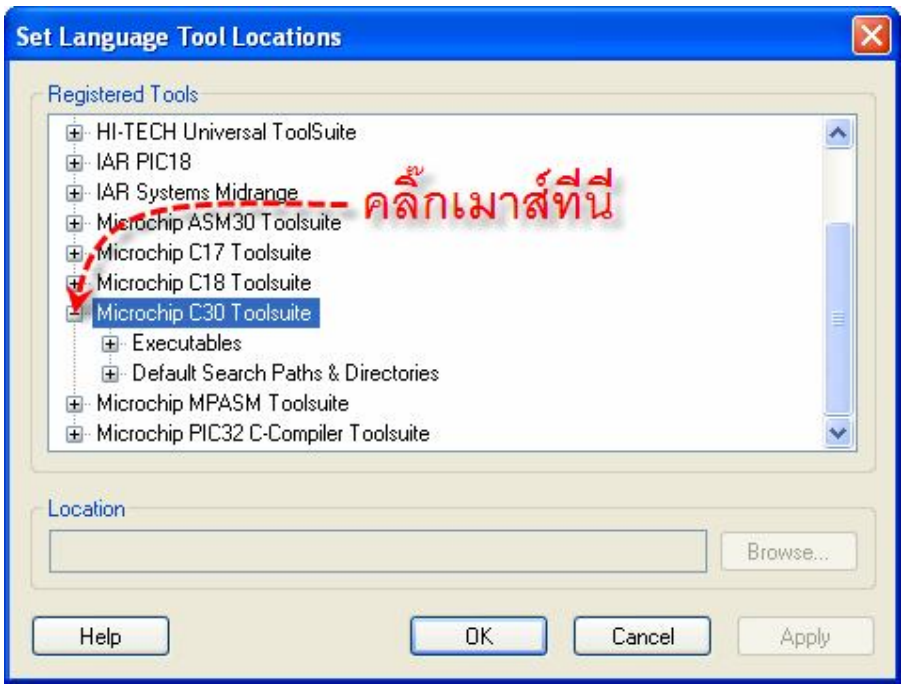
- ไฟล์ “pic30-as.exe” ซึ่งเป็นตัวโปรแกรมหลักสำหรับใช้ในการสั่งงานในการแปลคำสั่งภาษาแอสแซมบลี้ของ dsPIC (Assembler)
- ไฟล์ “pic30-gcc.exe” ซึ่งเป็นโปรแกรมหลักสำหรับใช้ในการสั่งงานในการแปลคำสั่งภาษาซีของ dsPIC (C Complier)
- ไฟล์ “pic30-ld.exe” ซึ่งเป็นโปรแกรมหลักสำหรับใช้ในการรวมไฟล์ต่างๆเข้าด้วยกันเพื่อสร้างเป็น Hex File ของ dsPIC (Linker)
- ไฟล์ “pic30-ar.exe” ซึ่งเป็นโปรแกรมหลักในการจัดการกับ Library

ซึ่งในอันดับแรกก่อนที่จะเริ่มต้นเข้าสู่ขั้นตอนของการใช้งานนั้น จะต้องทำการกำหนดการเชื่อมโยงคำสั่ง ระหว่างโปรแกรม MPLAB IDE และ MPLAB C30 ให้เรียบร้อยเสียก่อน เพื่อที่โปรแกรมMPLAB IDE จะได้ทราบว่าจะต้องไปเรียกใช้ไฟล์ต่างๆของ MPLAB C30 จากที่ใด โดยในการกำหนดการเชื่อมโยงโปรแกรมทั้ง 4 ให้สามารถใช้งานกับ MPLAB IDE นั้นสามารถทำได้ตามลำดับขั้นตอนต่อไปนี้

1. สั่ง Run โปรแกรม MPLAB IDE โดยอาจเรียกจาก “ICON” ของโปรแกรมหรือเรียกผ่าน Windows จาก “Start ⟶ Program ⟶ PIC development Tools ⟶ Microchip MPLAB⟶ MPLAB”
2. คลิกเมาส์ที่คำสั่ง “Project ⟶ Set Language Tools Locations..” แล้วเลือกกำหนดการใช้งานโปรแกรม MPLAB IDE ร่วมกับโปรแกรม MPLAB C30 แล้วเลือก “OK” ดังรูป

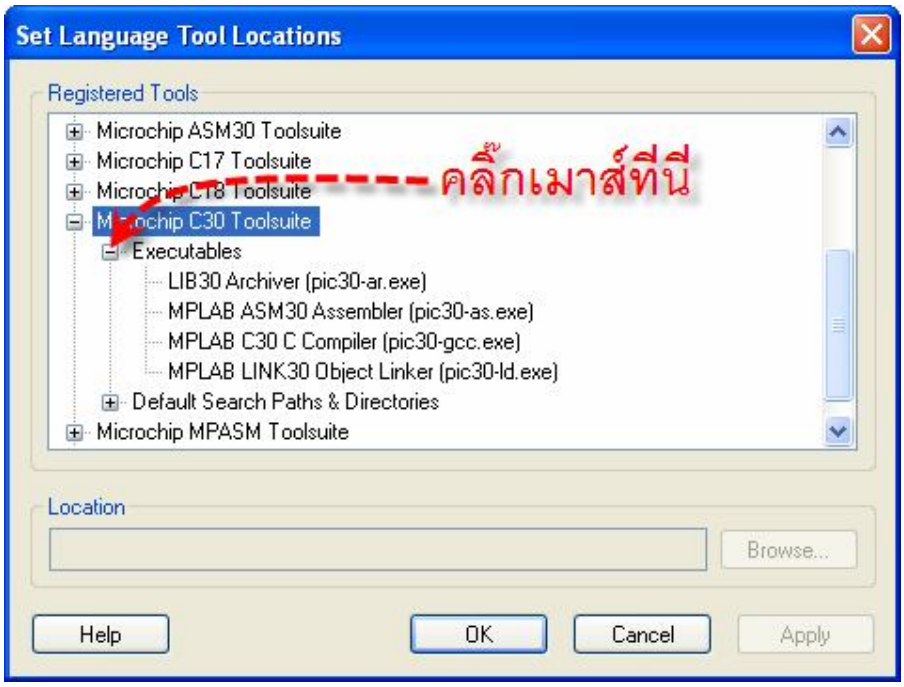


เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่บริเวณตำแหน่งเครื่องหมายบวก(+) ทีหน้าคำสั่งของ Microchip C30 Toolsuite ซึ่งจะได้ผลดังรูป



ซึ่งจะเห็นได้ว่าในหัวข้อ Microchip C30 Toolsuite นี้จะประกอบไปด้วยหัวข้อย่อยอีก 2 หัวข้อ คือ Executables และ Default Search Paths & Directories

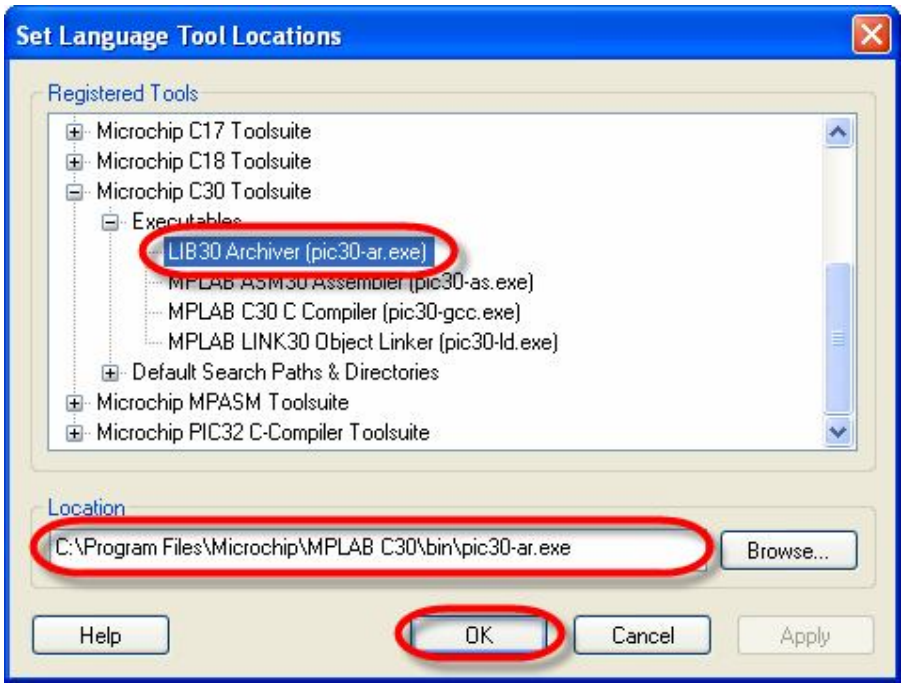
ให้ทำการกำหนดตำแหน่งของไฟล์ของ MPLAB C30 ที่ต้องการให้ MPLAB IDE เรียกใช้ ซึ่งจะมีอยู่ด้วยกันทั้งหมด 4 ไฟล์ โดยให้คลิกเมาส์ ที่บริเวณตำแหน่งเครื่องหมายบวก (+) ที่หน้าหัวข้อ Executables ซึ่งจะได้ผลดังรูป



ในขั้นตอนนี้ให้ทำการกำหนดชื่อ และ ตำแหน่ง Folder ที่อยู่ของไฟล์ทั้ง 4 ซึ่งได้แก่ pic30-ar.exe, pic30-as.exe, pic30-gcc.exe และ pic30-ld.exe โดยให้ทำการคลิกเมาส์ที่รายการย่อยของแต่ละหัวข้อจนปรากฏแถบสีน้ำเงินที่หัวข้อนั้นๆจากนั้นก็ให้กำหนดตำแหน่ง Folder และชื่อของไฟล์ ให้กับแต่ละหัวข้อจนครบทั้ง 4 หัวข้อ โดยชื่อไฟล์นั้นต้องกำหนดตามชื่อที่อยู่ในวงเล็บท้ายหัวข้อ ส่วนตำแหน่ง Folder นั้นตามปรกติแล้วจะอยู่ที่ "..\MPLAB C30\bin\" เช่น ถ้าติดตั้งโปรแกรม MPLAB C30 ไว้ตามที่แนะนำไว้ในตัวอย่างคือ "C:\Program Files\Microchip\MPLAB C30\" ไฟล์ที่ใช้สั่ง Run (Execute) ทั้งหมดจะอยู่ที่ "C:\Program Files\Microchip\MPLAB C30\bin\"

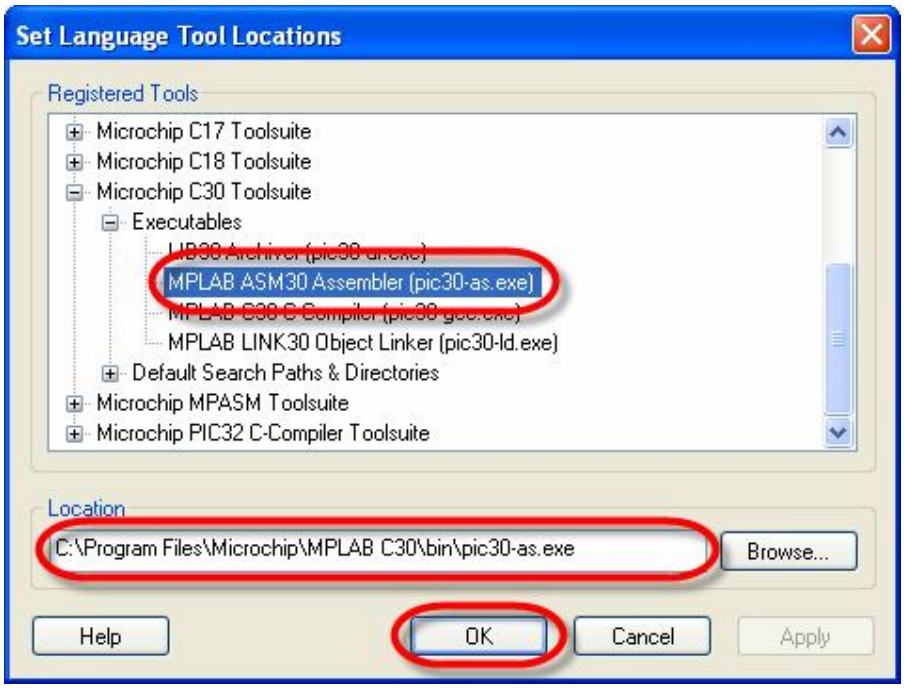
โดยวิธีการกำหนดชื่อและตำแหน่ง Folder ที่อยู่ของไฟล์ นั้นสามารถทำได้ 2 แบบ คือ การสั่งคลิกเมาส์ที่ "Browse..." เพื่อกำหนดตำแหน่งที่อยู่ของ Folder ซึ่งเก็บ Execute File ไว้ โดยให้ชี้ไปที่ "..\MPLAB C30\bin\.." ตัวอย่างเช่น ถ้าต้องการกำหนดชื่อ Execute File และตำแหน่ง Folder ของไฟล์สำหรับ LIB30 Archiver (pic30-ar.exe) ต้องกำหนดชื่อไฟล์เป็น "pic30-ar.exe" โดยให้คลิกเมาส์ที่ "Browse..." แล้วชี้ไปที่ "C:\Program Files\Microchip\MPLAB C30\bin\pic30-ar.exe" หรืออาจใช้วิธีการพิมพ์ ชื่อ Folder ในช่อง Location เองเป็น "C:\Program Files\Microchip\MPLAB C30\bin\pic30-ar.exe" ก็ได้ (ถ้าติดตั้งโปรแกรมไว้ต่างจากนี้ต้องกำหนดให้ตรงกับที่ติดตั้งไว้จริงด้วย)

โดยในขั้นตอนนี้ ให้กำหนดชื่อและตำแหน่ง Folder ของ Execute File ให้ครบทั้ง 4 หัวข้อ ด้วย ดังรูปในตัวอย่างต่อไปนี้

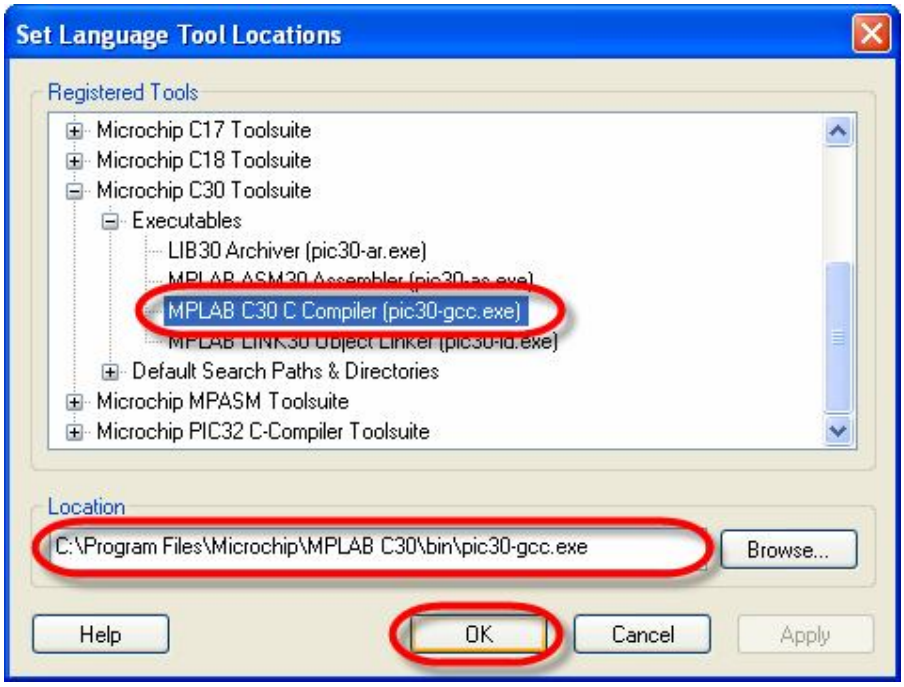


รูปแสดง ลักษณะการกำหนด **Execute File** ของ **LIB30 Archiver**

↓

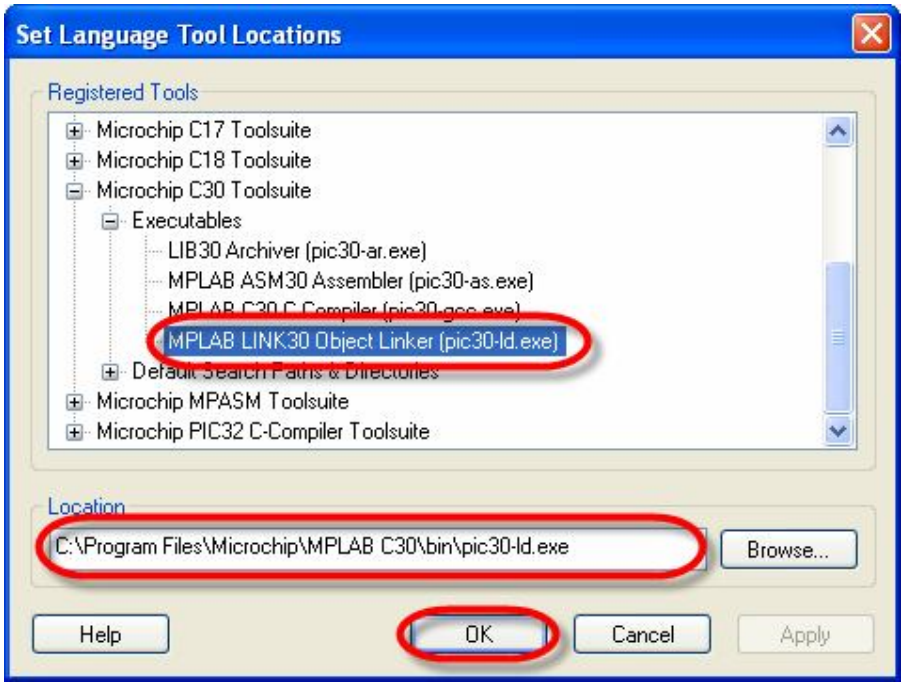


รูปแสดง ลักษณะการกำหนด **Execute File** ของ **MPLAB ASM30 Assembler**



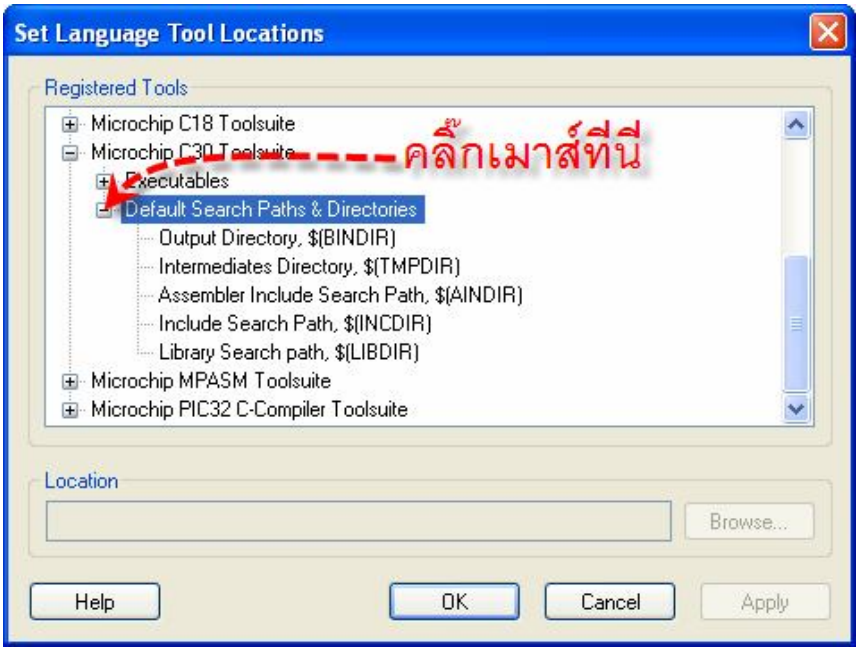
รูปแสดง ลักษณะการกำหนด **Execute File** ของ **MPLAB C30 C Complier**

↓

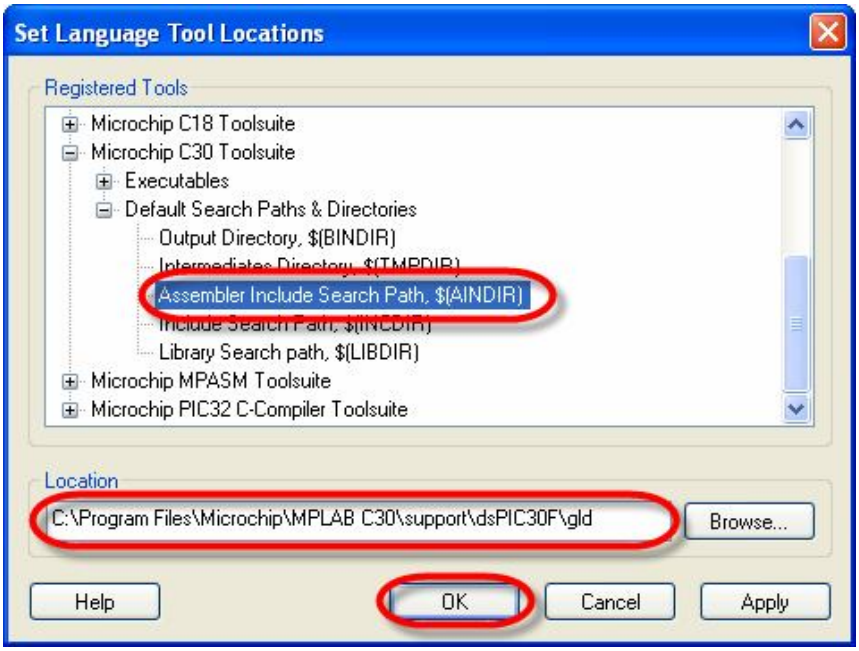


รูปแสดง ลักษณะการกำหนด **Execute File** ของ **MPLAB LINK30 Object Linker**

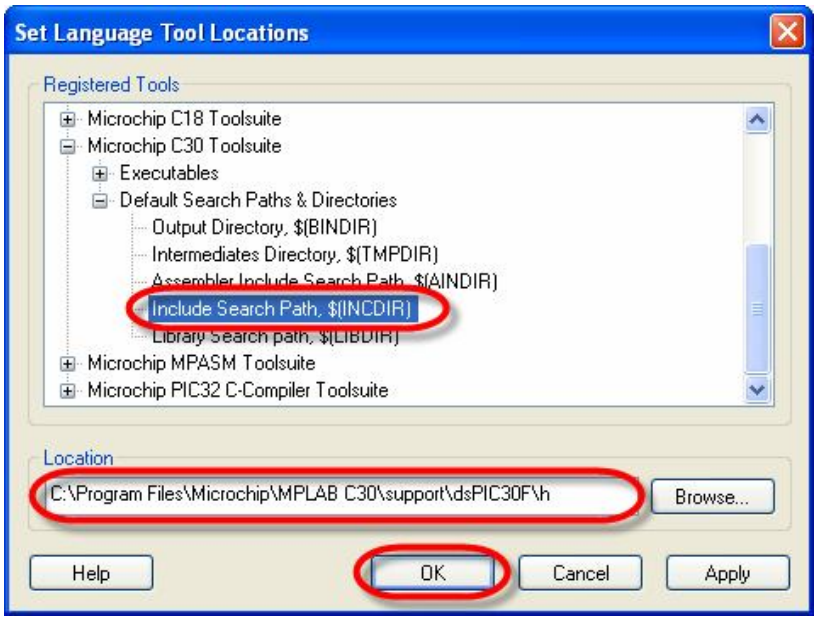
เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่บริเวณตำแหน่งเครื่องหมายบวก(+) ที่หน้าคำสั่งของ Default Search Patchs & Directories ซึ่งจะได้ผลดังรูป



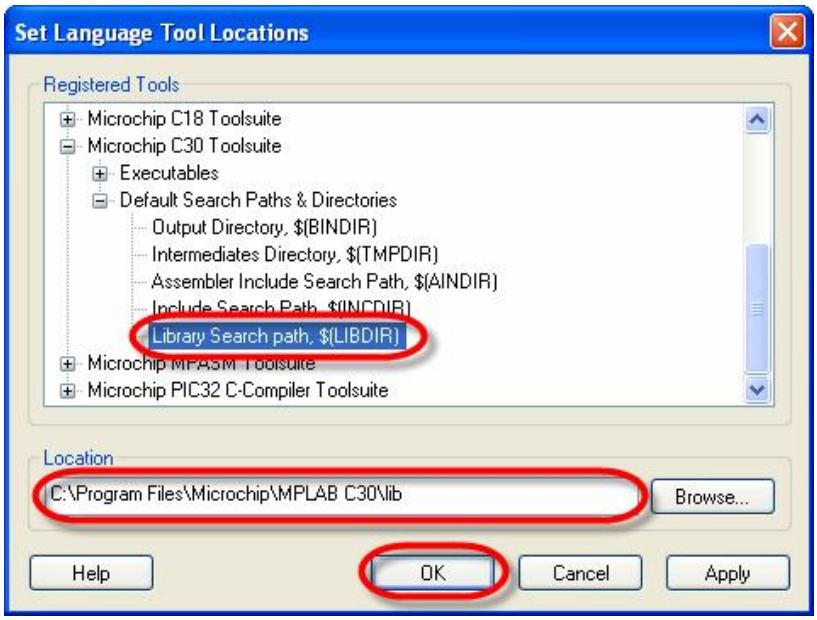
เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่ "Assembler Include Search Path, $(AINDIR)" จนปรากฏแถบสีน้ำเงินดังรูป แล้วจึงคลิกเมาส์ที่ "Browse..." เพื่อกำหนดตำแหน่งที่อยู่ของ Folder ซึ่งเก็บ Source Code ภาษา Assembly ไว้ โดยให้ชี้ไปที่ "..\MPLAB C30\support\dsPIC30F\gld" หรืออาจใช้วิธีการพิมพ์ ชื่อ Folder ในช่อง Location เองก็ได้ (ถ้าติดตั้งโปรแกรมไว้ต่างจากนี้ต้องกำหนดให้ตรงด้วย)



เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่ “Include Search Path, $(INCDIR)” จนปรากฏแถบสีน้ำเงินดังรูป แล้วจึงคลิกเมาส์ที่ “Browse...” เพื่อกำหนดตำแหน่งที่อยู่ของ Folder ซึ่งเก็บ Header File ไว้ โดยในการกำหนดตำแหน่ง Folder นั้นให้ชี้ไปที่ “..\MPLAB C30\support\dsPIC30F \h” หรืออาจใช้วิธีการพิมพ์ ชื่อ Folder ในช่อง Location เองก็ได้ดังรูป (ถ้าติดตั้งโปรแกรมไว้ต่างจากนี้ต้องกำหนดให้ตรงด้วย)



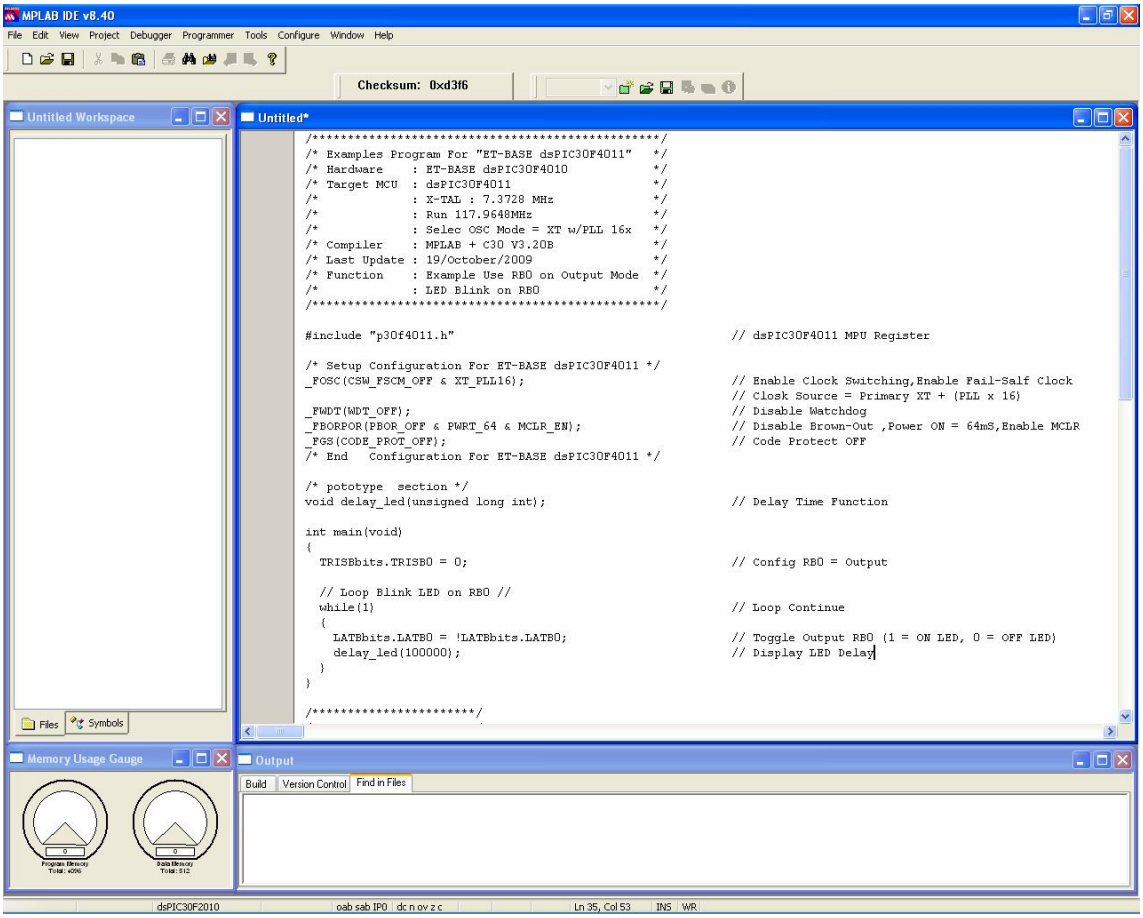
เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่ “Library Search Path, $(LIBDIR)” จนปรากฏแถบสีน้ำเงินดังรูป แล้วจึงคลิกเมาส์ที่ “Browse...” เพื่อกำหนดตำแหน่งที่อยู่ของ Folder ซึ่งเก็บ Library File ไว้ โดยในการกำหนดตำแหน่ง Folder นั้นให้ชี้ไปที่ “..\MPLAB C30\support\dsPIC30F \lib” หรืออาจใช้วิธีการพิมพ์ ชื่อ Folder ในช่อง Location เองก็ได้ดังรูป (ถ้าติดตั้งโปรแกรมไว้ต่างจากนี้ต้องกำหนดให้ตรงด้วย)



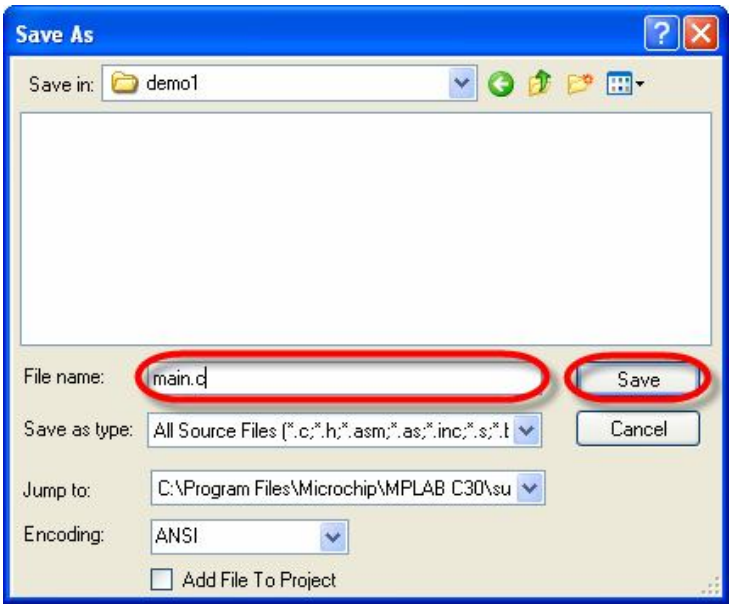
ตัวอย่างการสร้างโปรแกรมภาษาซีของ MPLAB C30

เมื่อทำการกำหนดการเชื่อมโยงคำสั่งระหว่าง MPLAB IDE และ MPLAB C30 เป็นที่เรียบร้อยแล้ว ต่อจากนี้ไป ผู้ใช้ก็สามารถทำการเรียกใช้งานโปรแกรม MPLAB C30 ผ่านทางโปรแกรม MPLAB IDE ได้แล้ว โดยค่าตัวเลือกต่างๆที่ได้กำหนดไว้แล้วนั้นจะถูกเก็บไว้ใน Configuration ของโปรแกรมตลอดไป จนกว่าจะมีการสั่งเปลี่ยนแปลงใหม่ ซึ่งในที่นี้จะขอยกตัวอย่างการเขียนโปรแกรมภาษาซี ของ MPLAB C30 สัก 1 ตัวอย่างพอเป็นแนวทางให้ผู้ใช้ทำความเข้าใจ โดยมีลำดับขั้นตอนดังนี้

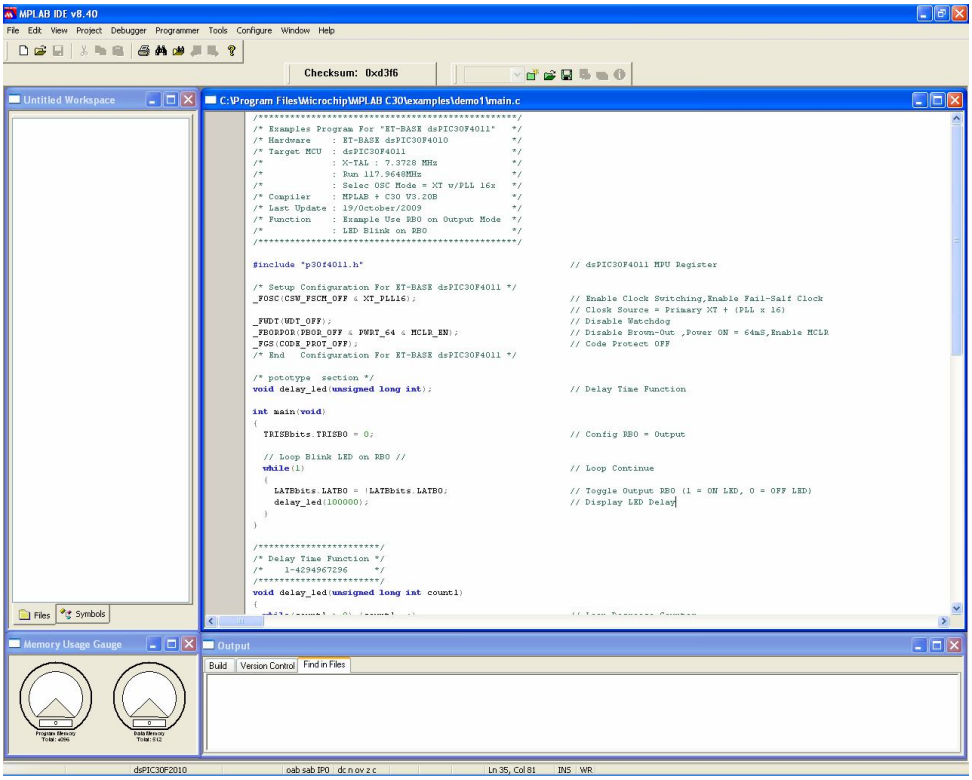
1. สั่ง Run Program ของ MPLAB IDE ขึ้นมา จากนั้นก็สั่งสร้างไฟล์ใหม่ขึ้นมา 1 ไฟล์ สำหรับเขียน Source Code ภาษาซี โดยใช้คำสั่ง “File ⟶ New” จากนั้นให้ทำการพิมพ์ Source Code ภาษาซี ใน Work Sheet ของโปรแกรม MPLAB IDE ดังตัวอย่าง



2. ทำการสั่งบันทึกไฟล์ที่เขียนขึ้นให้เป็น Text File ภาษาซี โดยให้ทดลองกำหนดชื่อเป็น “main.c” แล้วสั่งบันทึกไว้ใน Folder ชื่อ C:\Program Files\Microchip\MPLAB C30\examples\demo1\” โดยในที่นี้ให้ใช้คำสั่ง “File ⟶ Save As...” แล้วสร้าง Folder ชื่อ demo1 ไว้ภายใต้ Folder ของ examples อีกชั้นหนึ่ง แล้ว กำหนดชื่อเป็น “main.c” แล้วเลือก “Save” ดังรูป



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



```
#include "p30f4011.h"                // dsPIC30F4011 MPU Register

/* Setup Configuration For ET-BASE dsPIC30F4011 */
_FOSC(CSW_FSCM_OFF & XT_PLL16);      // Disable Clock Switching
                                     // Enable Fail-Salf Clock
                                     // Clock Source=Primary XT + (PLLx16)
_FWDT(WDT_OFF);                      // Disable Watchdog
_FBORPOR(PBOR_OFF & MCLR_EN);        // Disable Brown-Out ,Enable MCLR
_FGS(CODE_PROT_OFF);                 // Code Protect OFF
/* End   Configuration For ET-BASE dsPIC30F4011 */

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

int main(void)
{
  TRISBbits.TRISB0 = 0;              // Config RB0 = Output

  // Loop Blink LED on RB0 //
  while(1)                           // Loop Continue
  {
    LATBbits.LATB0 = !LATBbits.LATB0; // Toggle RB0 (1=ON LED, 0=OFF LED)
    delay_led(100000);               // Display LED Delay
  }
}

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

แสดง ตัวอย่าง Source Code สำหรับใช้ทดลองการทำงาน

สำหรับตัวอย่างนี้จะเป็นการสั่งให้ใช้พอร์ต RB0 ทำหน้าที่เป็น Output ขับ LED ให้ติดและดับสลับกันไปไม่รู้จบในลักษณะของไฟกระพริบ ซึ่งวิธีการทดสอบการทำงานของโปรแกรมนี้ โดยใช้กับบอร์ด ET-BASE dsPIC30F4011 นั้นทำได้โดยต่อสัญญาณจาก RB0 เข้ากับ LED Output ของบอร์ด โดยจะเห็นผลการทำงานของโปรแกรมแกรม คือ LED จะกระพริบ ติดและดับอย่างต่อเนื่องตลอดเวลา

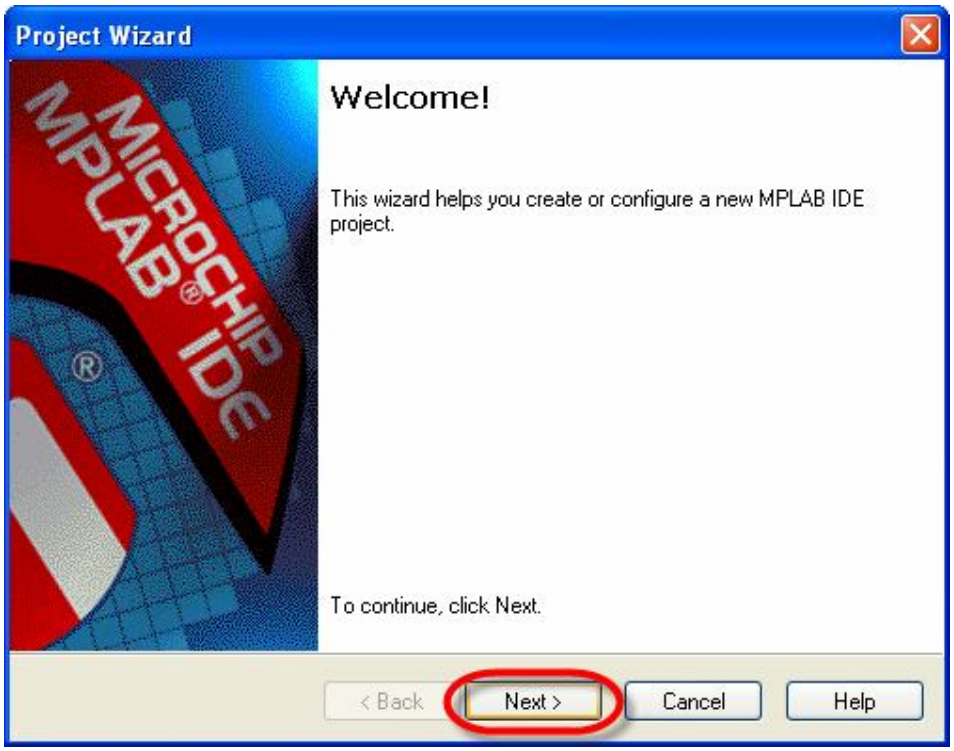
ส่วนที่น่าสนในของโปรแกรมคือบรรทัดคำสั่ง

```
/* Setup Configuration For ET-BASE dsPIC30F2010/4011 */
_FOSC(CSW_FSCM_OFF & XT_PLL16);      // Disable Clock Switching
                                     // Enable Fail-Salf Clock
                                     // Clock Source=Primary XT + (PLLx16)
_FWDT(WDT_OFF);                      // Disable Watchdog
_FBORPOR(PBOR_OFF & MCLR_EN);        // Disable Brown-Out ,Enable MCLR
_FGS(CODE_PROT_OFF);                 // Code Protect OFF
/* End   Configuration For ET-BASE dsPIC30F4011 */
```

โดยในส่วนนี้เป็นส่วนของการกำหนดค่า Configuration ของ MCU เบอร์ “dsPIC30F2010/4011” เพื่อใช้กับบอร์ด “ET-BASE dsPIC30F2010/4011”

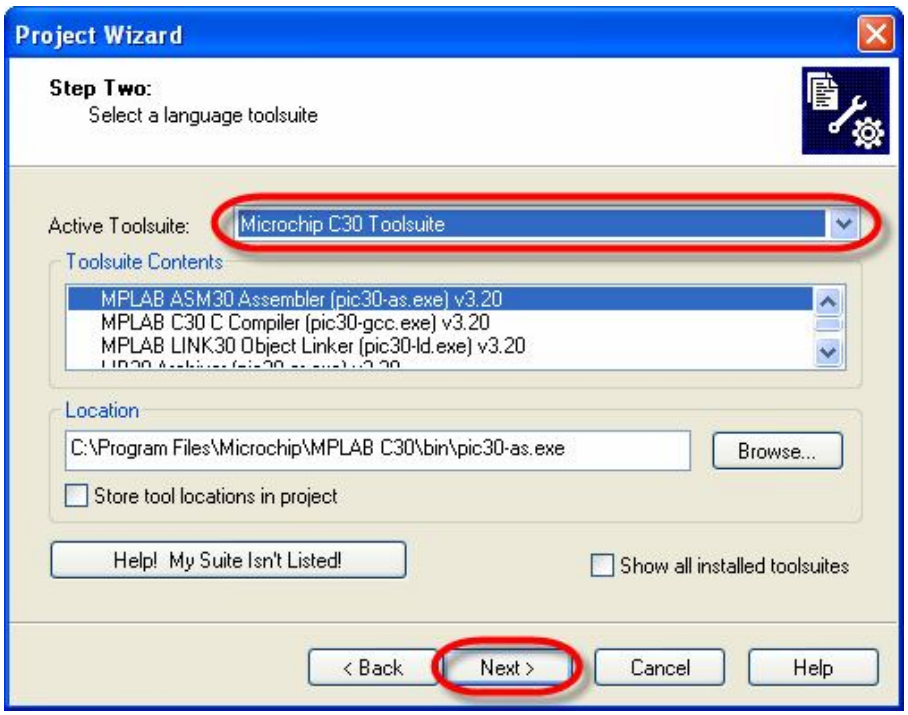
3. ทำการสร้าง Project File เพื่อใช้สั่งผนวกไฟล์ต่างๆ ที่เกี่ยวข้องเข้าด้วยกัน ทั้งนี้เนื่องมาจากว่าภาษาซีของ MPLAB C30 นั้น ถูกออกแบบให้มีความอ่อนตัวในการใช้งาน ดังนั้นจึงมีการจัดสร้างและแบ่งแยกไฟล์ออกเป็นหลายๆไฟล์ตามหน้าที่การใช้งาน เพื่อให้ผู้ใช้สามารถเรียกไฟล์ต่างๆ เหล่านั้นเข้ามาใช้งานร่วมกับ Source Code ที่เขียนขึ้นมาได้ได้ง่าย โดยไม่ต้องเสียเวลาเขียน Source Code เองทั้งหมด ซึ่งจะทำให้ผู้ใช้สามารถลดเวลาในการเขียนโปรแกรมไปได้เป็นอย่างมากเนื่องจากเพียงแต่ทำการสั่งผนวกไฟล์ที่ทาง MPLAB C30 สร้างเตรียมไว้ให้ เข้ากับ Source Code ที่ผู้ใช้เขียนขึ้นใหม่แล้วสั่งแปลโปรแกรมก็จะได้ไฟล์ที่มีความสมบูรณต่อการใช้งานแล้ว โดยจากตัวอย่าง Source Code ที่ได้ทดลองเขียนไปแล้วในข้างต้น ก็เช่นเดียวกัน จะเห็นได้ว่ามีการสั่งผนวกไฟล์ชื่อ “p30f2010.h” หรือ “p30f4011.h” เข้ามาใช้งานด้วย ซึ่งทำให้ไม่ต้องเสียเวลาไปสั่งประกาศชื่อและตำแหน่งรีจิสเตอร์ต่างๆของ dsPIC30F2010/4011 ให้เสียเวลา แต่สามารถอ้างถึงชื่อของรีจิสเตอร์ต่างๆในโปรแกรมได้ทันที โดยวิธีการกำหนดคุณสมบัติของ Project File มีดังนี้

   - สั่งกำหนดคุณสมบัติของ project File โดยใช้คำสั่ง “Project → Project Wizard...” ซึ่งจะได้ผลดังรูป จากนั้นให้เลือก “Next >” เพื่อไปยังขั้นตอนต่อไป

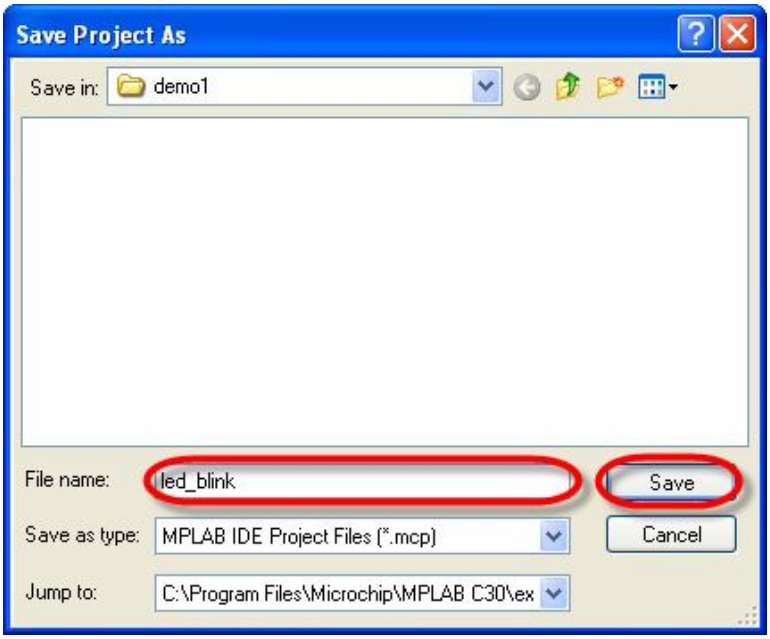


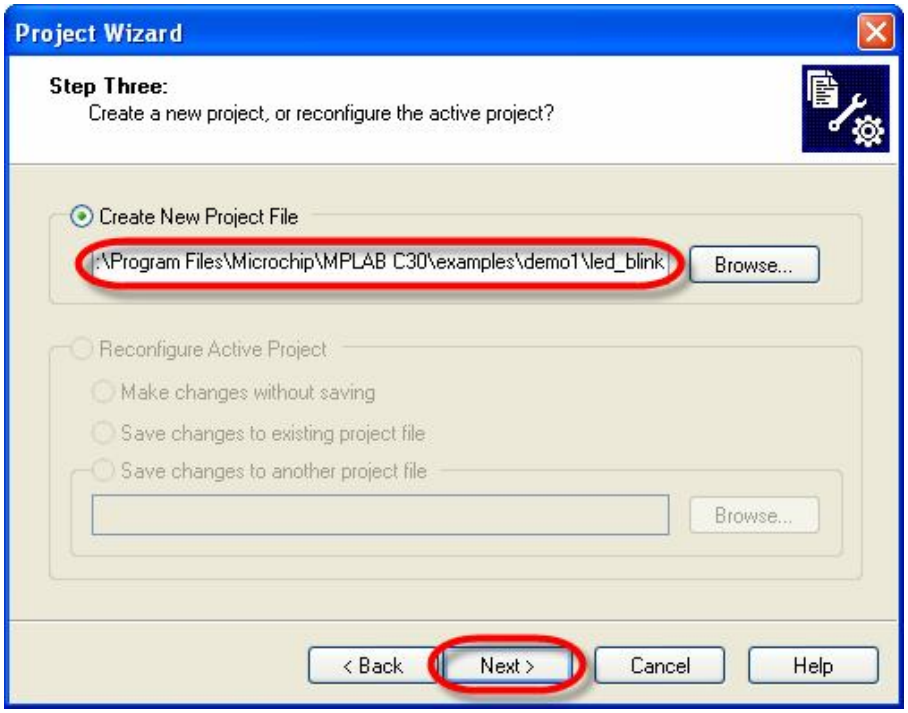
หลังจากเลือก “Next >” แล้ว โปรแกรมจะรอให้กำหนดเบอร์ของ MCU ที่จะใช้งานร่วมกับโปรแกรมที่เขียนขึ้น ซึ่งให้เลือกกำหนดเป็น “dsPIC30F4011” จากนั้นเลือก “Next >” เพื่อข้ามไปทำงานยังขั้นตอนต่อไปดังรูป

ในขั้นตอนนี้จะเป็นการเลือกว่าจะใช้โปรแกรมชุดใดในการแปลคำสั่ง เนื่องจาก MPLAB IDE สามารถใช้งานได้กับชุดโปรแกรมต่างๆมากมายหลายโปรแกรม ซึ่งในที่นี้ให้เลือกกำหนดใช้โปรแกรม ของ MPLAB C30 โดยการเลือกกำหนดตัวเลือกของ “Active Toolsuite” ให้เป็นของ MPLAB C30 โดยกำหนดตัวเลือกเป็น “Microchip C30 Toolsuite” ดังรูป แล้วเลือก “Next >”



ในขั้นตอนนี้จะเป็นการ กำหนดชื่อ Project และตำแหน่ง Folder ที่จะใช้เก็บไฟล์ต่างๆที่ได้จากการทำงานของ Project โดยให้กำหนดชื่อเป็น “led_blink” แล้วกำหนดตำแหน่ง Folder เป็น demo1 โดยกำหนดไว้ภายใต้ Folder ชื่อ examples ของ MPLAB C30 ดังรูป แล้วเลือก “Next >”



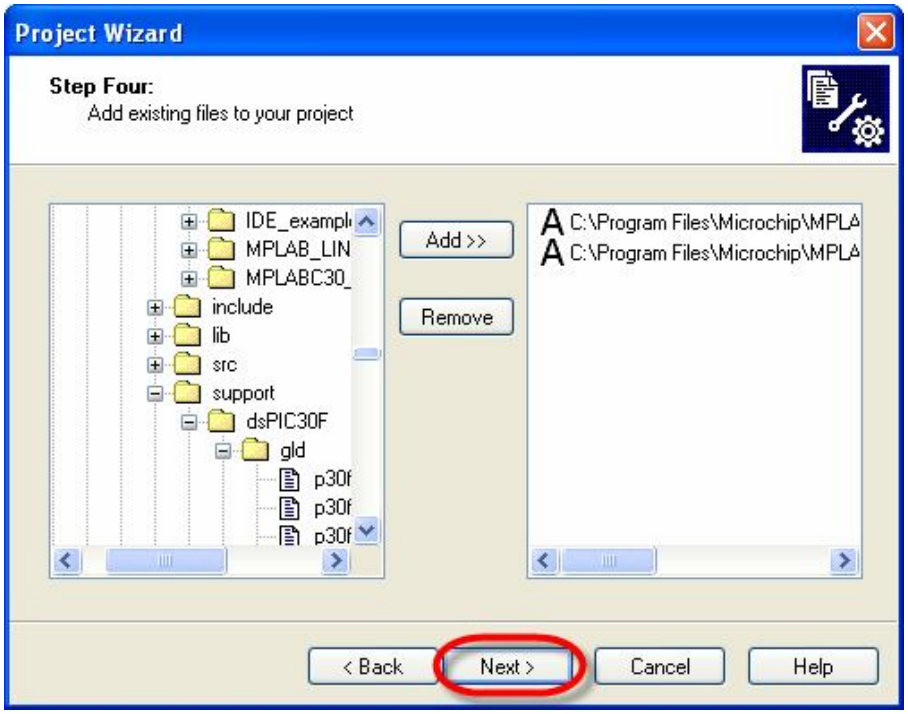


ในขั้นตอนนี้จะเป็นการสั่งผนวกไฟล์ต่างๆเข้าไว้ด้วยกันภายใต้ชื่อ Project ของ led_blink โดยให้ทำการสั่งผนวกไฟล์ทั้งหมด 3 ไฟล์เข้าไว้ใน Project ดังนี้

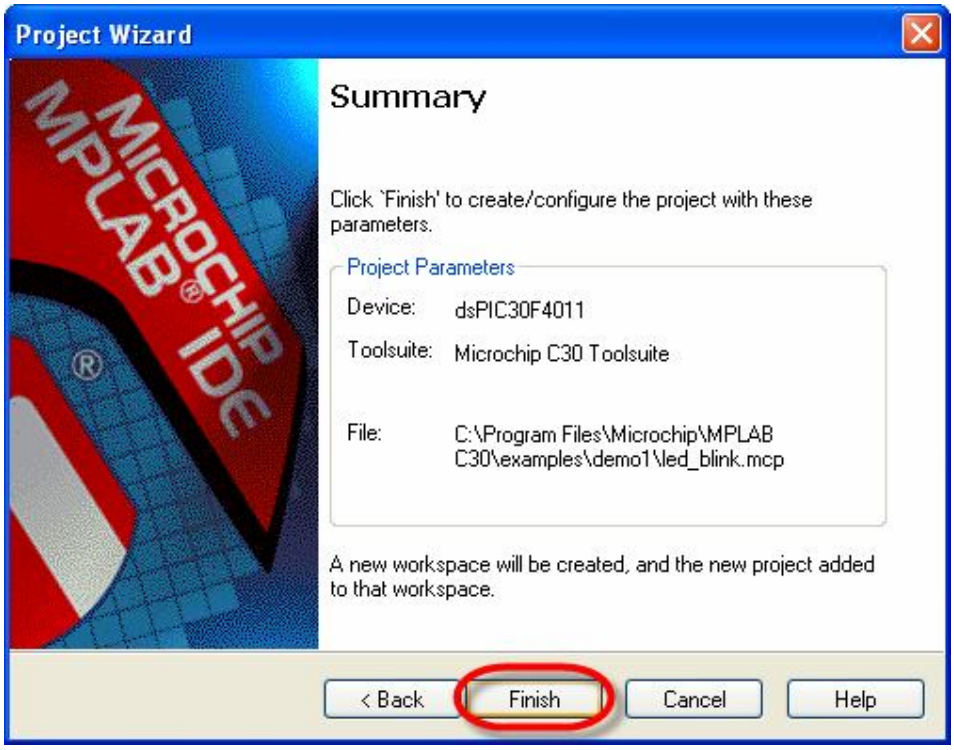
- สั่งผนวกไฟล์ชื่อ “main.c” ซึ่งเป็น Source Code ที่เราได้เขียนและสั่งบันทึกไว้แล้วก่อนหน้านี้ใน C:\Program Files\Microchip\MPLAB C30\examples\demo1\main.c”
- สั่งผนวกไฟล์ชื่อ “p30f4011.gld” ซึ่งเป็น Script File ของ dsPIC30F4011 ที่ทาง MPLAB C30 สร้างเตรียมไว้ให้ ถ้าใช้ dsPIC30F2010 ให้เลือกเป็น “p30f2010.gld” โดยถ้าติดตั้งโปรแกรมตามตัวอย่างไฟล์ดังกล่าวจะเก็บอยู่ใน “C:\Program Files\Microchip\MPLAB C30\support\dsPIC30F\gld\”

โดยในการสั่งผนวกไฟล์ทั้ง 2 ดังกล่าวให้ทำการคลิกเมาส์ไปยัง “ICON” ของไฟล์จากตำแหน่ง Folder ที่กล่าวไว้ในข้างต้นที่ผ่านมาแล้วเลือก “Add >>” จนชื่อไฟล์ดังกล่าวไปปรากฏอยู่ที่กรอบหน้าต่างด้านขวาของโปรแกรม ซึ่งให้เลือกทำตามวิธีการนี้จนสามารถสั่ง “Add” ไฟล์ได้ครบทั้ง 2 ไฟล์ แล้วเลือก “Next >” เพื่อไปยังขั้นตอนต่อไปดังรูป

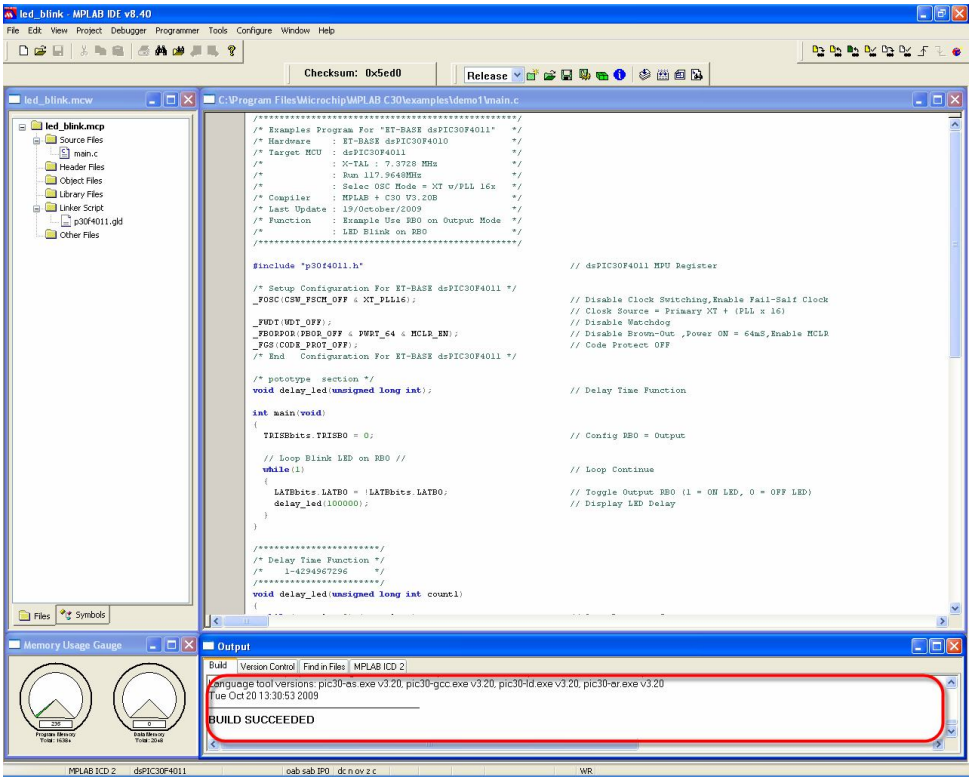
↓



เมื่อสั่งผนวกไฟล์ทั้งหมดเข้ากับ Project ไฟล์ที่สร้างขึ้นเป็นที่เรียบร้อยแล้ว โปรแกรมจะรายงานผลโดยแสดงค่าพารามิเตอร์ต่างๆที่ได้กำหนดไว้แล้วให้ทราบดังรูป ให้เลือก “Finish” เป็นอันเสร็จขั้นตอนของการสร้าง Project File ของ led_blink



ซึ่งหลังจากกำหนดค่าต่างๆให้กับ Project File เป็นที่เรียบร้อยแล้ว ผู้ใช้ก็สามารถทำการสั่งแปลคำสั่งของโปรแกรม “main.c” ที่เขียนขึ้นได้ทันที โดยใช้คำสั่ง “Project → Build All” ซึ่งจะทำให้ได้ไฟล์ Output มีชื่อเดียวกับ Project File ที่สร้างไว้แต่มีนามสกุลเป็น HEX ซึ่งจากตัวอย่าง Project นี้เมื่อสั่งแปลโปรแกรมแล้วถ้าไม่เกิดข้อผิดพลาดใดๆจะได้ Output ไฟล์ชื่อ “led_blink.hex” โดยไฟล์ดังกล่าวจะถูกสร้างและเก็บไว้ใน Folder ของ Project คือ “C:\Program Files\Microchip\MPLAB C30\examples\demo1\” โดยผู้ใช้สามารถสั่ง Download Hex File ชื่อ “led_blink.hex” นี้ให้กับบอร์ดเพื่อทดสอบการทำงานได้ทันที



ซึ่งในกรณีที่มีเครื่องโปรแกรมต่อไว้กับบอร์ดเรียบร้อยแล้ว ผู้ใช้สามารถเลือกสวิตช์เพื่อเปลี่ยนโหมดการทำงานของบอร์ดมาไว้ที่ PGM (LED PGM สีแดงติดสว่าง) จากนั้นเลือกที่เมนูคำสั่งในโปรแกรม MPLAB โดยให้เลือกที่ Programmer-> Select Programmer แล้วเลือกรุ่นของเครื่องมือที่ใช้อยู่จริง

- ในกรณีที่ใช้เครื่องโปรแกรมของ อีทีที รุ่น ET-ICDX ให้เลือกเป็น MPLAB ICD2
- ในกรณีที่ใช้เครื่องโปรแกรมของ อีทีที รุ่น ET-PGM PIC USB ให้เลือกเป็น PICkit 2

เมื่อทำการเลือกเครื่องมือเรียบร้อยแล้วสามารถสั่ง Program ได้ทันที โดยให้เลือกที่เมนู Programmer->Program แล้วรอจนการทำงานเสร็จเรียบร้อย จากนั้นก็เลือกสวิตช์เพื่อเปลี่ยนโหมดการทำงานของบอร์ดกลับไปยัง Run (LED RUN สีเขียวติดสว่าง) แล้วกดสวิตช์รีเซ็ตที่บอร์ดจะเห็นบอร์ดเริ่มต้นทำงานทันที โดยถ้าเลือก Jumper ของ LED Test (ENA/DIS) ไว้ทางด้าน ENA จะเห็น LED กระพริบให้เห็นทันที

การกำหนดค่า Configuration ให้กับ dsPIC30F2010/4011

สำหรับ dsPIC30F2010/4011 นั้นจะมีรีจิสเตอร์พิเศษจำนวน 4 ชุด ซึ่งเรียกว่า Configuration Register สำหรับใช้กำหนดคุณสมบัติและควบคุมการทำงานของวงจรต่างๆใน MCU ซึ่งการกำหนดค่าให้กับรีจิสเตอร์พิเศษทั้ง 4 ชุดนี้ จะต้องกระทำในขั้นตอนของการโปรแกรมด้วยวิธีการแบบ ICSP หรือจากเครื่องโปรแกรมภายนอกเท่านั้น

โดยสำหรับบอร์ด ET-BASE dsPIC30F2010/4011 นั้น ในกรณีที่ใช้การพัฒนาโปรแกรม ร่วมกับ MPLAB และ C30 นั้น จะสามารถกำหนดค่าตัวเลือกต่างๆให้กับ Configuration Register จาก Code คำสั่งที่เขียนขึ้นได้ทันที ดังตัวอย่าง

```
/* Setup Configuration For ET-BASE dsPIC30F2010/4011 */
_FOSC(CSW_FSCM_OFF & XT_PLL16);       // Disable Clock Switching
                                      // Enable Fail-Salf Clock
                                      // Clock Source=Primary XT + (PLLx16)
_FWDT(WDT_OFF);                       // Disable Watchdog
_FBORPOR(PBOR_OFF & MCLR_EN);         // Disable Brown-Out ,Enable MCLR
_FGS(CODE_PROT_OFF);                  // Code Protect OFF
/* End   Configuration For ET-BASE dsPIC30F4011 */
```

โดยรายรายละเอียดและ Keyword ของ Macro ทั้งหมดที่ใช้ในการกำหนดค่า Configuration ของ dsPIC30F2010/4011 นั้น ผู้ใช้สามารถดูได้จาก “..\MPLAB C30\support\dsPIC30F\h\p30f2010.h” หรือ “..\MPLAB C30\support\dsPIC30F\h\p30f4011.h”

หมายเหตุ สำหรับบอร์ด ET-BASE dsPIC30F2010/4011 ของ อีทีที นั้น ถ้าต้องการใช้งานกับโปรแกรมตัวอย่างต่างๆที่ทางอีทีที จัดทำขึ้นโดยไม่เกิดปัญหา จะต้องเลือก Oscillator เป็นแบบ **“Primary”** โดยใช้ **“XT w/PLL 16X-XT crystal oscillator mode with 16X PLL”** พร้อมกับยกเลิกบิต **“FWDTEN”** เพื่อปิดการทำงานของ Watchdog ด้วยเสมอ ส่วนตัวเลือกอื่นๆสามารถเลือกกำหนดได้ตามต้องการ

FOSC (Oscillator Configuration Register)

เป็น Configurations Register ขนาด 24 บิต โดยมีตำแหน่งแอดเดรสของรีจิสเตอร์อยู่ที่ F80000H ทำหน้าที่สำหรับใช้เลือกกำหนดคุณสมบัติและแหล่งกำเนิดของสัญญาณนาฬิกาที่จะป้อนให้กับ MCU ของ MCU เพื่อใช้เป็นสัญญาณนาฬิกาของระบบ ซึ่งตามปรกติแล้วระบบสัญญาณนาฬิกาที่สามารถกำหนดใช้กับ dsPIC30F2010/4011 จะสามารถเลือกกำหนดได้หลายแหล่งขึ้นอยู่กับการออกแบบวงจรใช้งาน **แต่ในกรณีที่ใช้กับบอร์ด ET-BASE dsPIC30F2010/4011 นั้น ระบบสัญญาณนาฬิกาจะได้รับการออกแบบให้ใช้งานกับตัวกำเนิดความถี่แบบ XTAL ค่า 7.3728MHz และเพื่อให้สามารถใช้งาน MCU ได้อย่างเต็มประสิทธิภาพและสามารถใช้งานกับตัวอย่างโปรแกรมต่างๆ ที่ทาง อีทีที จัดทำขึ้น ได้โดยไม่เกิดปัญหาจะต้องเลือกกำหนดแหล่งกำเนิดของสัญญาณนาฬิกาเป็นแบบ "Primary Oscillator" ร่วมกับ "XT w/PLL 16x" เท่านั้น** ซึ่งการเลือกค่าดังกล่าวเป็นการเลือกกำหนดให้ MCU ใช้สัญญาณนาฬิกาจากวงจรกำเนิดความถี่แบบ XTAL ซึ่ง ต่อเข้ากับขา OSC1 และ OSC2 พร้อมทั้งเปิดการทำงานของวงจรคูณความถี่ Phase-Lock-Loop ด้วยอัตราการคูณ 16 เท่า ด้วย ซึ่งจะส่งผลให้ค่าความถี่ของสัญญาณนาฬิกาของบอร์ด ET-BASE dsPIC30F2010/4011 มีค่า 117.9648 MHz ซึ่งความถี่ของสัญญาณนาฬิกาดังกล่าวจะถูกนำไปหารให้เหลือ ¼ ก่อนที่จะป้อนให้กับระบบของ MCU ดังนั้นค่าความถี่ของสัญญาณนาฬิกาที่ใช้ในบอร์ด ET-BASE dsPIC30F2010/4011 จะมีค่าเท่ากับ 29.4912 MHz สำหรับรายละเอียดการเลือกกำหนดค่าของบิตต่างๆใน PSOC Configurations Register จะมีดังต่อไปนี้

บิตข้อมูล	23	22	21	20	19	18	17	16
ชื่อบิต	-	-	-	-	-	-	-	-

บิตข้อมูล	15	14	13	12	11	10	9	8
ชื่อบิต	FCKSM[1:0]		-	-	-	-	FOS[1:0]	

บิตข้อมูล	7	6	5	4	3	2	1	0
ชื่อบิต	-	-	-	-	FPR[3..0]			

ตาราง แสดงค่าบิตต่างๆของ FOSC Configurations Register

- **FCKSM[1:0]** ใช้สำหรับกำหนดการทำงานของวงจรสลับสัญญาณนาฬิกาและตรวจสอบการทำงานของสัญญาณนาฬิกา
  - 0:0 = เป็นการเลือกเปิดการทำงาน (Enable) ทั้งส่วนของวงจรสลับสัญญาณนาฬิกา และ วงจรตรวจสอบความผิดพลาดของสัญญาณนาฬิกา(Fail-Safe Clock Monitor)
  - 0:1 = เป็นการเลือกเปิด (Enable) การทำงานของวงจรสลับสัญญาณนาฬิกา แต่ในส่วนของวงจรตรวจสอบความผิดพลาดของสัญญาณนาฬิกาจะถูกปิด (Disable) ไว้

- 1:X = เป็นการเลือกปิด (Disable)ทั้งส่วนของวงจรสลับสัญญาณนาฬิกา และ วงจรตรวจสอบความผิดพลาดของสัญญาณนาฬิกา (Fail-Safe Clock Monitor)

- **FOS[1:0]** ใช้สำหรับเลือกแหล่งกำเนิดสัญญาณนาฬิกาของ MCU ซึ่งสามารถเลือกได้ 4 แบบ แต่สำหรับในกรณีของบอร์ด ET-BASE dsPIC30F2010/4011 ควรเลือกเป็น "1:1" เท่านั้น
  - 0:0 = เลือกใช้สัญญาณนาฬิกาจาก XTAL 32KHz ภายใน MCU
  - 0:1 = เลือกใช้สัญญาณนาฬิกาจาก RC ภายในแบบย่านความถี่สูง
  - 1:0 = เลือกใช้สัญญาณนาฬิกาจาก RC ภายในแบบย่านความถี่ต่ำและใช้พลังงานต่ำ
  - 1:1 = เลือกใช้สัญญาณนาฬิกาจาก Primary Oscillator โดยกำหนดคุณสมบัติการทำงานของวงจรร่วมกับบิต FPR[3:0] อีกต่อหนึ่ง

- **FPR[3:0]** ใช้สำหรับเลือกการทำงานของสัญญาณนาฬิกาของ Primary Oscillator ซึ่งค่าของตัวเลือกในส่วนนี้จะมีผลต่อการทำงานของ MCU ก็ต่อเมื่อเลือกกำหนดแหล่งของสัญญาณนาฬิกาเป็น Primary Oscillator แล้ว โดยคุณสมบัติการทำงานของ FPR{3:0] มีดังนี้
  - **000x = "XLT"** เป็นการเลือกใช้ XTAL ค่า 200KHz-4MHz ต่อกับขา OSC1,OSC2
  - **001x = "HS"** เป็นการเลือกใช้ XTAL ค่า 10MHz-25MHz ต่อกับขา OSC1,OSC2
  - **0100 = "XT"** เป็นการเลือกใช้ XTAL ค่า 4MHz-10MHz ต่อกับขา OSC1,OSC2
  - **0101 = "XT PLL 4x"** เป็นการเลือกใช้ XTAL ค่าระหว่าง 4MHz - 10MHz ต่อเข้ากับ OSC1 และ OSC2 โดยเปิดการทำงานของวงจรคูณความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 4 เท่าของความถี่ XTAL ด้วย
  - **0110 = "XT PLL 8x"** เป็นการเลือกใช้ XTAL ค่าระหว่าง 4MHz - 10MHz ต่อเข้ากับ OSC1 และ OSC2 โดยเปิดการทำงานของวงจรคูณความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 8 เท่าของความถี่ XTAL ด้วย
  - **0111 = "XT PLL 16x"** เป็นการเลือกใช้ XTAL ค่าระหว่าง 4MHz - 10MHz ต่อเข้ากับ OSC1 และ OSC2 โดยเปิดการทำงานของวงจรคูณความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 16 เท่าของความถี่ XTAL ด้วย
  - **1000 = "ERCIO"** เป็นการเลือกใช้สัญญาณนาฬิกาจากวงจร RC ภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปรกติ
  - **1001 = "ERC"** เป็นการเลือกใช้สัญญาณนาฬิกาจากวงจร RC ภายนอก โดยต่อสัญญาณนาฬิกาเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น Output ของสัญญาณนาฬิกาที่ได้ โดยมีค่าความถี่เป็น ¼ หรือ Fosc/4

- **1011 = "EC"** เป็นการเลือกใช้สัญญาณนาฬิกาค่า 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น Output ของสัญญาณนาฬิกาที่รับจาก OSC1 โดยมีค่าความถี่เป็น ¼ หรือ Fosc/4
- **1100 = "ECIO"** เป็นการเลือกใช้สัญญาณนาฬิกาค่า 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปรกติ
- **1101 = "EC PLL 4x"** เป็นการเลือกใช้สัญญาณนาฬิกาค่า 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปรกติ โดยเปิดการทำงานของวงจรคูณความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 4 เท่าของความถี่ที่ป้อนให้กับขา OSC1 ด้วย
- **1110 = "EC PLL 8x"** เป็นการเลือกใช้สัญญาณนาฬิกาค่า 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปรกติ โดยเปิดการทำงานของวงจรคูณความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 8 เท่าของความถี่ที่ป้อนให้กับขา OSC1 ด้วย
- **1111 = "EC PLL 16x"** เป็นการเลือกใช้สัญญาณนาฬิกาค่า 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปรกติ โดยเปิดการทำงานของวงจรคูณความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 16 เท่าของความถี่ที่ป้อนให้กับขา OSC1 ด้วย

หมายเหตุ การกำหนดค่า Configuration ของ FOSC นั้น จะมีผลต่อการทำงานของ MCU โดยตรง ซึ่งถ้ากำหนดไม่ถูกต้องจะทำให้ MCU ไม่สามารถทำงานได้ หรือทำงานได้ไม่ถูกต้อง ถึงแม้ว่าผู้ใช้จะเขียนโปรแกรมได้อย่างถูกต้องทุกประการก็ตามที โดยในกรณีของการใช้งานบอร์ด "ET-BASE dsPIC30F2010/4011" นั้น จะเห็นได้ว่าวงจรกำเนิดความถี่จะถูกออกแบบให้ใช้ตัวกำเนิด XTAL ค่า 7.3728 MHz ต่อกับขา OSC1 และ OSC2 ของ MCU โดยตรง ซึ่งนั่นก็หมายความว่าการกำหนดค่า Configuration ของ FOSC ที่จะใช้งานกับบอร์ด "ET-BASE dsPUIC30F2010/4011" นั้นจะต้องกำหนดเป็น Primary แบบ XT เท่านั้น ซึ่งถ้าเลือกเป็นอย่างอื่นจะทำให้ MCU ไม่สามารถทำงานได้ แต่ในกรณีที่ต้องการใช้งานตัวอย่างโปรแกรมที่สร้างโดย อีทีที จะต้องกำหนดการทำงานของ FOSC ให้เปิดการทำงานของวงจรคูณความถี่จากตัวกำเนิดความถี่ค่า 7.3728MHz ด้วยอัตรา 16 เท่าด้วย (XT PLL16X) ถ้าเลือกเป็น XT หรือ XT PLL 4 หรือ XT PLL 8 ถึงแม้ว่า MCU จะสามารถทำงานได้ แต่ก็จะส่งผลทำให้ได้ค่าความถี่ ไม่ตรงกับที่คำนวณไว้ในโปรแกรมจะทำให้การทำงานของโปรแกรมไม่ถูกต้อง เช่น ความเร็วของการทำงานช้าลง ค่าอัตรา Baud rate ของพอร์ตสื่อสารอนุกรมผิดเพี้ยนไปเป็นต้น ดังนั้นต้องกำหนดค่า Configuration ของ FOSC เป็นแบบ "XT PLL16" เสมอ MCU จึงจะทำงานได้อย่างถูกต้อง

FWDT (Watchdog Timer Configuration Register)

เป็น Configurations Register ขนาด 24 บิต โดยมีตำแหน่งแอดเดรสของรีจิสเตอร์อยู่ที่ F80002H ใช้สำหรับกำหนดการทำงานของ Watchdog ซึ่งในกรณีที่ใช้กับบอร์ด ET-BASE dsPIC30F2010/4011 และต้องการใช้งานร่วมกับโปรแกรมตัวอย่างต่างๆที่ทางอีทีทีจัดทำขึ้น ควรกำหนดค่าของบิต “PWTEN” ให้เป็น “0” เพื่อปิดการทำงานของ Watchdog ไว้ก่อน ซึ่งถ้าต้องการใช้งาน Watchdog ผู้ใช้อาจทำการปรับปรุงการทำงานของโปรแกรมที่เขียนขึ้น โดยเพิ่มคำสั่งสำหรับรีเซ็ตค่าการนับของ Watchdog ในส่วนต่างๆของโปรแกรมตามความเหมาะสมแล้วจึงสั่งเปิดการทำงานของ Watchdog ในภายหลัง โดยค่าตัวเลือกต่างๆของ FWDT มีดังนี้

บิตข้อมูล	23	22	21	20	19	18	17	16
ชื่อบิต	-	-	-	-	-	-	-	-

บิตข้อมูล	15	14	13	12	11	10	9	8
ชื่อบิต	-	-	-	-	-	-	-	-

บิตข้อมูล	7	6	5	4	3	2	1	0
ชื่อบิต	-	-	FWPSA[1:0]		FWPSB[3:0]			

ตาราง แสดงค่าบิตต่างๆของ FWDT Configurations Register

- **FWTEN** ใช้สำหรับกำหนดการทำงานของ Watchdog โดยถ้ากำหนดให้บิตนี้เป็น “1” จะเป็นการสั่งเปิดการทำงานของ Watchdog ซึ่งในกรณีนี้ผู้ใช้ต้องเขียนคำสั่งเพื่อสั่งรีเซ็ตค่าการนับของ Watchdog ก่อนเกิดการ Overflow ในส่วนต่างๆของโปรแกรมเองด้วย ไม่เช่นนั้นแล้วเมื่อการนับของ Watchdog เกิดการ Overflow ขึ้น วงจร Watchdog จะสั่งรีการทำงานของ MCU ให้เริ่มต้นใหม่ทันที แต่ถ้ากำหนดค่าของบิตนี้ให้เป็น “0” จะเป็นการปิดการทำงานของ Watchdog ซึ่งขอแนะนำให้สั่งปิดการทำงานของ Watchdog ไปก่อน จนผู้ใช้สามารถเขียนโปรแกรมได้อย่างชำนาญแล้ว ถ้าต้องการใช้งานวงจร Watchdog จึงค่อยสั่งเปิดการทำงานของ Watchdog ตามต้องการในภายหลัง
- **FWPSA[1:0]** ใช้สำหรับกำหนดค่า Prescaler ของวงจร Prescale ชุด A เพื่อกำหนดค่าการลดทอนความถี่สัญญาณนาฬิกาที่จะป้อนให้กับวงจร Prescale ชุด B เพื่อใช้เป็นค่าการนับของ Watchdog
  - 0:0 = กำหนดการ Prescale เป็น 1:1
  - 0:1 = กำหนดการ Prescale เป็น 1:8

- 1:0 = กำหนดการ Prescale เป็น 1:64
- 1:1 = กำหนดการ Prescale เป็น 1:512

- **FWPSB[3:0]** ใช้สำหรับกำหนดค่า Prescale ของวงจร Prescale ชุด B เพื่อกำหนดค่าการนับของ Watchdog โดยสามารถกำหนดได้ 16 ระดับคือ
  - 0000 = กำหนดการ Prescale เป็น 1:1
  - .
  - .
  - 1110 = กำหนดการ Prescale เป็น 1:15
  - 1111 = กำหนดการ Prescale เป็น 1:16

หมายเหตุ การทำงานของวงจร Watchdog นั้นจะช่วยเพิ่มประสิทธิภาพการทำงานของ MCU ในการใช้งานให้เกิดประสิทธิภาพสูงสุด โดยในระบบไมโครคอนโทรเลอร์นั้น จะอาศัย Watchdog ช่วยตรวจสอบการทำงานของ MCU ไม่ให้หยุดทำงาน ซึ่งเหมาะกับการใช้งานวงจรที่ต้องทำการอย่างต่อเนื่องตลอดเวลาเป็นเวลานานๆ ซึ่งหลักการทำงานของวงจร Watchdog นั้นจะเป็นวงจรนับอิสระวงจรหนึ่ง ซึ่งสามารถกำหนดค่าการของจำนวนการนับสูงสุด (Time-Out) ที่แน่นอนให้กับวงจรได้ โดยเมื่อวงจรการนับของ Watchdog นับไปจนถึงค่าการนับสูงสุดแล้วค่าการนับจะกลับมาเริ่มต้นใหม่ ซึ่งเรียกว่าการ Overflow โดยในช่วงนี้จะทำให้วงจร Watchdog ส่งสัญญาณไปรีเซ็ตการทำงานของ MCU ให้เริ่มต้นทำงานใหม่

ซึ่งเราจะใช้ประโยชน์จากวงจร Watchdog ในการตรวจสอบการทำงานของ MCU โดยการเขียนคำสั่งสำหรับสั่งรีเซ็ตค่าการนับของ Watchdog แทรกไว้ในส่วนต่างๆของโปรแกรม ก่อนที่ค่าการนับของ Watchdog จะเกิดการ Overflow ขึ้น แต่ถ้าหากว่า MCU เกิดการหยุดการทำงานขึ้น ไม่ว่าจะเกิดจากสาเหตุใดก็ตาม MCU ก็จะไม่สามารส่งสัญญาณไปรีเซ็ตค่าการนับของ Watchdog ได้ดังนั้น MCU ก็จะถูก Watchdog รีเซ็ตให้กลับมาเริ่มต้นทำงานใหม่ได้อีก

แต่อย่างไรก็ตามในการพัฒนาโปรแกรมใช้งานนั้น ถ้าผู้พัฒนาโปรแกรมยังไม่มีความชำนาญในการเขียนโปรแกรมเพียงพอ โดยเฉพาะในระหว่างที่เป็นช่วงเริ่มต้นของการเรียนรู้นั้น ควรสั่งปิดการทำงานของ Watchdog ไว้ก่อน ไม่เช่นนั้นแล้วจะเป็นการเพิ่มความสับสนให้กับผู้ใช้ได้ เนื่องจากถ้ามีการแทรกคำสั่งสำหรับรีเซ็ตค่าการนับของ Watchdog ไม่เหมาะสมก็อาจทำให้ Watchdog สั่งรีเซ็ตการทำงานของ MCU ในขณะที่ MCU กำลังทำงานตามโปรแกรมปรกติอยู่ก็เป็นได้ ดังนั้นขอแนะนำให้สั่งปิดการทำงานของ Watchdog ไว้ก่อน จนเมื่อสามารถพัฒนาโปรแกรมร่วมกับวงจรส่วนอื่นๆ จนโปรแกรมสามารถทำงานได้อย่างถูกต้องครบทั้งหมดแล้ว จึงค่อยเปิดการทำงานและแทรกคำสั่งสำหรับ รีเซ็ตค่าการนับของWatchdog ในภายหลัง ซึ่งตัวอย่างโปรแกรมต่างๆที่ทาง อีทีที จัดทำขึ้นนั้นก็ไม่ได้จัดการเรื่อง Watchdog ไว้ด้วยเช่นเดียวกัน ดังนั้นควรกำหนดค่า Configuration ของ FWDT ให้ปิดการทำงานของ Watchdog ไว้ด้วย

FBORPOR (BOR and POR Configuration Register)

เป็น Configurations Register ขนาด 24 บิต โดยมีตำแหน่งแอดเดรสของรีจิสเตอร์อยู่ที่ F80004H ใช้สำหรับกำหนดการทำงานของวงจร Power-ON Reset และ Brown-Out Reset

บิตข้อมูล	23	22	21	20	19	18	17	16
ชื่อบิต	-	-	-	-	-	-	-	-

บิตข้อมูล	15	14	13	12	11	10	9	8
ชื่อบิต	MCLREN	-	-	-	-	PWMPIN	HPOL	LPOL

บิตข้อมูล	7	6	5	4	3	2	1	0
ชื่อบิต	BOREN	-	BORV[1:0]		-	-	FPWRT[1:0]	

ตาราง แสดงค่าบิตต่างๆของ FBORPOR Configurations Register

- **MCLREN** ใช้สำหรับกำหนดการทำงานของขาสัญญาณ MCLR โดยถ้ากำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการเปิดการทำงาน (Enable) ของขาสัญญาณ MCLR เป็น External Reset ตามปรกติ แต่ถ้ากำหนดให้บิตนี้มีค่าเป็น “0” จะเป็นการปิดการทำงาน (Disable) ของขาสัญญาณ MCLR
- **PWMPIN** ใช้สำหรับกำหนดการทำงานของขาสัญญาณ PWM ในระหว่างการรีเซ็ต ซึ่งใน dsPIC30F2010 จะมีอยู่ 3 ชุด โดยถ้ากำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการกำหนดให้ขา Output ของ PWM ถูกควบคุมโดยวงจร Input/Output และมีสถานะเป็น Tri-State ในระหว่างเกิดการรีเซ็ตอยู่ แต่ถ้ากำหนดให้บิตนี้มีค่าเป็น “0” จะเป็นการกำหนดให้ขา Output ของ PWM ถูกควบคุมจากวงจร PWM โดยในขณะเกิดการรีเซ็ตอยู่จะมีสถานะเป็น Output
- **HPOL** ใช้สำหรับกำหนดลักษณะการทำงานของขา PWMH เมื่อเกิดการทำงาน (Active) โดยถ้ากำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการกำหนดให้ขาสัญญาณ PWMH เป็น High ในขณะ Active แต่ถ้ากำหนดให้บิตนี้เป็น “0” จะเป็นการกำหนดให้ PWMH เป็น Low ในขณะ Active
- **LPOL** ใช้สำหรับกำหนดลักษณะการทำงานของขา PWML เมื่อเกิดการทำงาน (Active) โดยถ้ากำหนดให้บิตนี้มีค่าเป็น “1”จะเป็นการกำหนดให้ขาสัญญาณ PWML เป็น High ในขณะ Active แต่ถ้ากำหนดให้บิตนี้เป็น “0” จะเป็นการกำหนดให้ PWML เป็น Low ในขณะ Active
- **BOREN** ใช้สำหรับกำหนดการทำงานของวงจร Brown-Out Reset โดยถ้ากำหนดเป็น “1” จะเป็นการเปิดการทำงานของ Brown-Out แต่ถ้ากำหนดเป็น “0” จะเป็นการปิดการทำงานของ Brown-Out

- **BORV[1:0]** ใช้สำหรับกำหนดระดับค่าของแรงดันที่จะใช้เป็นจุดอ้างอิงในการตรวจสอบความผิดพลาดของระดับแหล่งจ่ายของวงจร Brown-Out Reset
  - 0:0 = ให้วงจร Brown-Out ทำงานเมื่อแรงดันต่ำกว่า 4.5V
  - 0:1 = ให้วงจร Brown-Out ทำงานเมื่อแรงดันต่ำกว่า 4.2V
  - 1:0 = ให้วงจร Brown-Out ทำงานเมื่อแรงดันต่ำกว่า 2.7V
  - 1:1 = ให้วงจร Brown-Out ทำงานเมื่อแรงดันต่ำกว่า 2.0V

- **FPWRT[1:0]** ใช้สำหรับกำหนดค่าเวลาการทำงานของวงจร Power-ON Reset
  - 0:0 = เป็นการปิดการทำงาน (Disable) ของวงจร Power-ON Reset
  - 0:1 = เป็นการเลือกเปิดการทำงานของวงจร Power-ON Reset โดยมีค่าเวลา 4mS
  - 1:0 = เป็นการเลือกเปิดการทำงานของวงจร Power-ON Reset โดยมีค่าเวลา 16mS
  - 1:1 = เป็นการเลือกเปิดการทำงานของวงจร Power-ON Reset โดยมีค่าเวลา 64mS

FGS (General Code Segment Configuration Register)

เป็น Configurations Register ขนาด 24 บิต โดยมีตำแหน่งแอดเดรสของรีจิสเตอร์อยู่ที่ F8000AH ใช้สำหรับกำหนดระบบการป้องกันข้อมูลของหน่วยความจำใน MCU

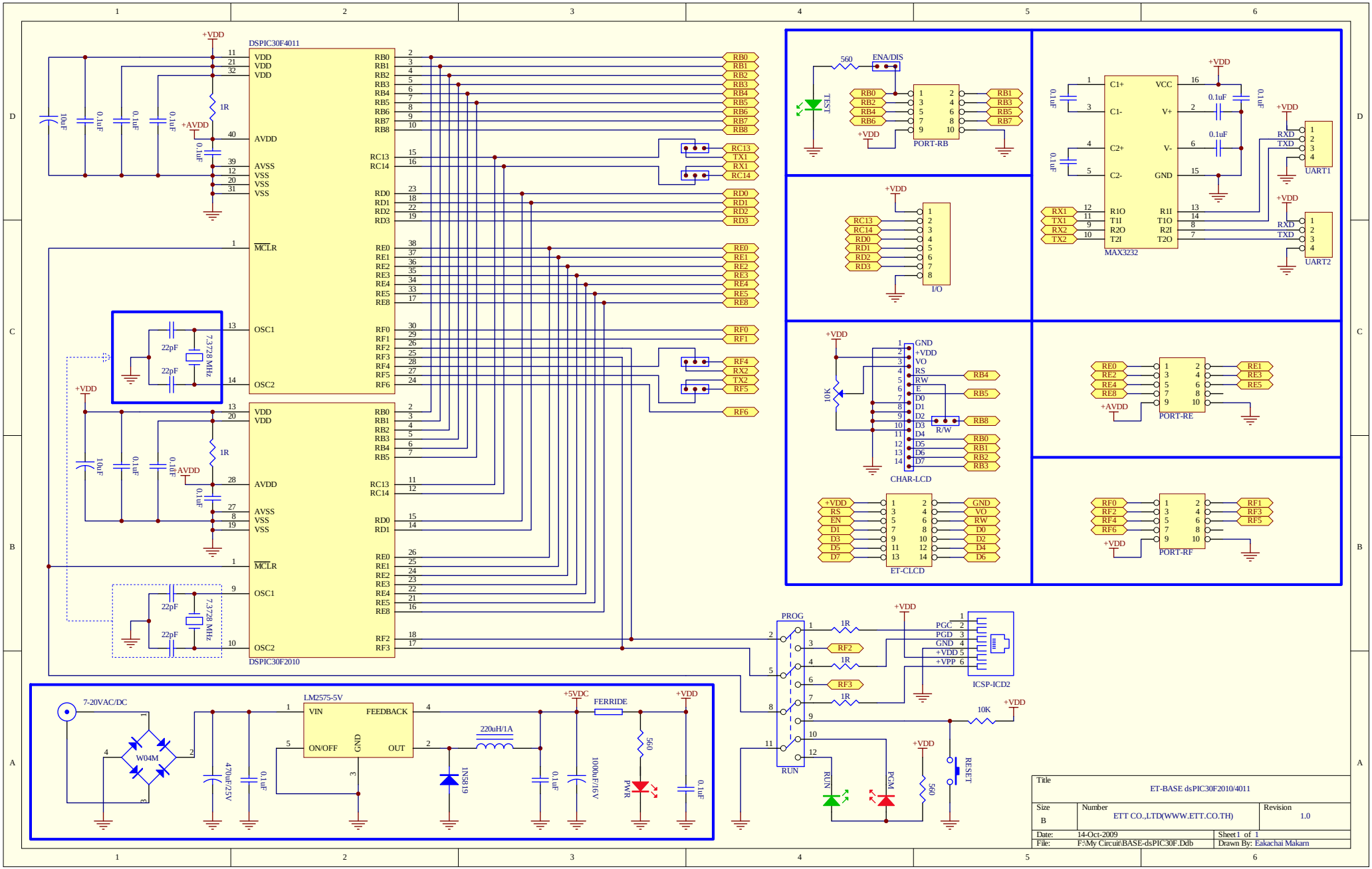
บิตข้อมูล	23	22	21	20	19	18	17	16
ชื่อบิต	-	-	-	-	-	-	-	-

บิตข้อมูล	15	14	13	12	11	10	9	8
ชื่อบิต	-	-	-	-	-	-	-	-

บิตข้อมูล	7	6	5	4	3	2	1	0
ชื่อบิต	-	-	-	-	-	-	GCP	GWRP

ตาราง แสดงค่าบิตต่าง ๆของ FGS Configurations Register

- **GCP** ใช้สำหรับกำหนดการป้องกันการอ่านข้อมูลของหน่วยความจำโปรแกรม หรือ Flash Memory จากภายนอก โดยถ้ากำหนดให้บิตนี้เป็น “1” จะเป็นการปิด (Disable) ระบบป้องกันการอ่านข้อมูลจากหน่วยความจำ Flash แต่ถ้าเลือกกำหนดให้บิตนี้เป็น “0” จะเป็นการเปิด (Enable) ระบบป้องกันการอ่านข้อมูลจากหน่วยความจำ Flash ซึ่งจะทำให้ไม่สามารถสั่งอ่านข้อมูลจากหน่วยความจำ Flash ใน MCU จากภายนอกได้ โดยถ้าเลือกกำหนดให้บิตนี้มีค่าเป็น “0” แล้วจะไม่สามารถสั่งเปลี่ยนค่าให้กลับมาเป็น “1” ได้อีก นอกจากจะสั่งลบข้อมูลออกจากหน่วยความจำทั้งหมดเสียก่อน ซึ่งหลังจากสั่งลบข้อมูลในหน่วยความจำโปรแกรมทั้งหมดแล้วบิตนี้จะมีค่าเป็น “1” โดยอัตโนมัติ
- **GWRP** ใช้สำหรับกำหนดการป้องกันการเขียนข้อมูลของหน่วยความจำโปรแกรม หรือ Flash Memory โดยถ้ากำหนดให้บิตนี้เป็น “1” จะเป็นการปิด (Disable) ระบบป้องกันการเขียนข้อมูลใหม่ให้กับหน่วยความจำ Flash แต่ถ้าเลือกกำหนดให้บิตนี้เป็น “0” จะเป็นการเปิด (Enable) ระบบป้องกันการเขียนข้อมูลให้กับหน่วยความจำ Flash ซึ่งจะทำให้ไม่สามารถสั่งเขียนข้อมูลให้กับหน่วยความจำ Flash ใน MCU เพิ่มเติมได้อีกจนกว่าจะสั่งลบข้อมูลเดิมทั้งหมดออกเสียก่อน โดยถ้าเลือกกำหนดให้บิตนี้มีค่าเป็น “0” แล้วจะไม่สามารถสั่งเปลี่ยนค่าให้กลับมาเป็น “1” ได้อีก นอกจากจะสั่งลบข้อมูลออกจากหน่วยความจำทั้งหมดเสียก่อน ซึ่งหลังจากสั่งลบข้อมูลในหน่วยความจำโปรแกรมทั้งหมดแล้วบิตนี้จะมีค่าเป็น“1” โดยอัตโนมัติ


DSPIC30F4011
DSPIC30F2010
+VDD
+AVDD
1R
10uF
0.1uF
22pF
7.3728 MHz
MCLR
OSC1
OSC2
AVDD
VDD
VSS
RB0
RB1
RB2
RB3
RB4
RB5
RB6
RB7
RB8
RC13
RC14
RD0
RD1
RD2
RD3
RE0
RE1
RE2
RE3
RE4
RE5
RE8
RF0
RF1
RF2
RF3
RF4
RF5
RF6
RX2
TX2
RX1
TX1
7-20VAC/DC
W04M
470uF/25V
LM2575-5V
VIN
FEEDBACK
ON/OFF
GND
OUT
220uH/1A
1N5819
1000uF/16V
+5VDC
FERRIDE
PWR
560
PROG
RUN
PGC
PGD
+VPP
ICSP-ICD2
10K
RESET
PGM
TEST
ENA/DIS
PORT-RB
I/O
MAX3232
C1+
C1-
C2+
C2-
VCC
V+
V-
R1O
R1I
T1I
T1O
R2O
R2I
T2I
T2O
UART1
UART2
RXD
TXD
PORT-RE
PORT-RF
CHAR-LCD
R/W
ET-CLCD
RS
EN
D0
D1
D2
D3
D4
D5
D6
D7
VO
GND
Title
ET-BASE dsPIC30F2010/4011
Size
B
Number
ETT CO.,LTD(WWW.ETT.CO.TH)
Revision
1.0
Date:
14-Oct-2009
Sheet 1 of 1
File:
F:\My Circuit\BASE-dsPIC30F.Ddb
Drawn By: Eakachai Makarn