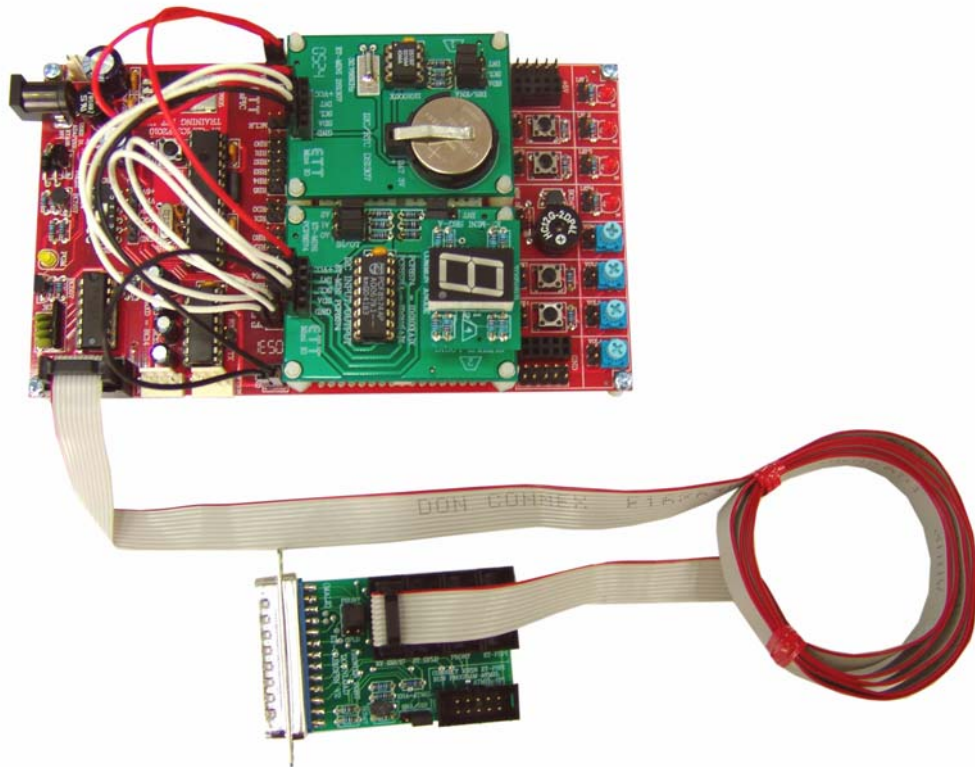


ET-dsPIC30F2010 TRAINING KIT V1.0 / EXP



ET-dsPIC30F2010 TRAINING KIT V1.0 / EXP เป็นชุดทดลอง “dsPIC” ขนาดเล็ก โดยตัวบอร์ดได้รับการออกแบบให้สามารถรองรับการศึกษาเรียนรู้เกี่ยวกับสถาปัตยกรรมของไมโครคอนโทรลเลอร์ตระกูล dsPIC จากค่าย Microchips ในขั้นพื้นฐานได้อย่างครบถ้วน ภายใต้งบประมาณที่เรียกว่าประหยัดและคุ้มค่าที่สุด โดยตัวบอร์ดเลือกใช้ MCU เบอร์ dsPIC30F2010 เป็น CPU ประจำบอร์ด พร้อมทั้งจัดวงจรพื้นฐานอื่นๆที่จำเป็นรวบรวมไว้ภายในบอร์ด ไม่ว่าจะเป็นแหล่งจ่ายไฟ วงจรแสดงผลการทำงานของ Output แบบ LED ,วงจรสร้างสัญญาณ Input Logic แบบ Push-Button Switch ,วงจรสร้างสัญญาณ Input Analog แบบใช้ตัวต้านทาน ปรับค่าแรงดัน 0-5V ,วงจรกำเนิดเสียงด้วยลำโพงขนาดเล็ก ,วงจร Line Driver สำหรับพอร์ตสื่อสารอนุกรม RS232 และวงจร Download แบบ ICSP เป็นต้น

นอกจากนี้แล้วตัวบอร์ดยังได้ออกแบบให้มีพื้นที่สำหรับใช้ต่อทดลองวงจรขนาดเล็ก โดยมีทั้งส่วนที่เป็นลาย PCB แบบจุดไข่ปลา หรือ อาจติดตั้งแผง Photo Board ขนาด 360 จุดเพื่อใช้เป็นพื้นที่ต่อวงจรต่างๆเพิ่มเติมได้เองตามต้องการโดยใช้ชุด “ET-Hardware KIT” นอกจากนี้แล้วยังสามารถนำบอร์ด I/O ขนาดเล็ก ซึ่งทางทีมงาน อีทีที จัดเตรียมไว้เป็นทางเลือกและสนับสนุนการทดลอง ซึ่งเรียกว่าชุดโมดูลการทดลอง “Mini I/O Module” แบบต่างๆ ไม่ว่าจะเป็น ชุดทดลอง DC-Motor ,ชุดทดลอง Stepping Motor ,ชุดทดลอง I2C ต่างๆ ซึ่งมีทั้งชุดทดลอง I2C I/O PCF8574 ,ชุดทดลอง I2C EEPROM 24xx ,ชุดทดลอง I2C RTC เบอร์ DS1307 ,ชุดทดลอง I2C RTC เบอร์ PCF8583 ,ชุดทดลอง SPI โดยใช้ SPI Output เบอร์ 74HC595 ,ชุดทดลอง Matrix Keyboard ขนาด 4x4 ,ชุดทดลอง 7-Segment ฯลฯ

คุณสมบัติของบอร์ด ET-dsPIC30F2010 TRAINING KIT V1.0 EXP

บอร์ด ET-dsPIC30F2010 TRAINING KIT V1.0 EXP จะประกอบไปด้วย วงจรพื้นฐานที่จำเป็นสำหรับการศึกษาเรียนรู้และทดลองใช้งานทรัพยากรต่างๆของ MCU ตระกูล dsPIC โดยภายในบอร์ดได้จัดเตรียมวงจรใช้งานที่จำเป็นไว้ให้ใช้งานอย่างครบถ้วนได้แก่

- ใช้ dsPIC30F2010 เป็น MCU ประจำบอร์ด โดยขาสัญญาณต่างๆจาก MCU จะถูกจัดเรียงไว้อย่างเป็นระเบียบ ผ่าน “Pin Header” พร้อมลายพิมพ์ขาว แสดงชื่อขาสัญญาณอย่างชัดเจน เพื่อให้ง่ายและสะดวกต่อการต่อทดลองใช้งานขาสัญญาณต่างๆ โดยไม่ถูกจำกัดหน้าที่การใช้งานด้วยวงจรแบบใดแบบหนึ่ง ทำให้ผู้ใช้สามารถทดลองใช้งานฟังก์ชันภายในผ่านขาสัญญาณต่างๆ ของ MCU ได้โดยอิสระและครบถ้วนตามต้องการ
- วงจรแหล่งจ่ายไฟ แบบ Bridge Rectifier ขนาด 1A พร้อมวงจร Filter สามารถใช้กับแหล่งจ่ายไฟได้ทั้ง AC และ DC ขนาด 14-18V
- วงจร Regulate ขนาด +5V / 1A สำหรับใช้งานเป็นแหล่งจ่ายไฟเลี้ยงวงจรให้กับ MCU และอุปกรณ์การทดลองต่างๆ พร้อม LED แสดงสถานะสีแดง และจุด Connector เชื่อมต่อใช้งาน ทั้งตัวผู้และตัวเมีย
- วงจร Regulate ขนาด +13V / 100mA สำหรับใช้เป็นแรงดันโปรแกรม ในการ Download ข้อมูลแบบ HEX File ให้กับหน่วยความจำโปรแกรม (Flash) ของ MCU ภายในบอร์ด
- วงจร Download แบบ ICSP โดยใช้งานร่วมกับ “ET-CAP10P V2.0” สำหรับใช้พัฒนาโปรแกรมในบอร์ด พร้อม LED แสดงสถานะสีเขียว สามารถ Download HEX File จากคอมพิวเตอร์ PC ให้กับ MCU ในบอร์ดได้ทันที โดยไม่ต้องใช้เครื่องมือโปรแกรมอื่นๆอีก
- วงจร Line Driver สำหรับพอร์ตสื่อสารอนุกรม RS232 จำนวน 2 ช่อง โดยเชื่อมต่อกับสัญญาณ RC13(TXD) และ RC14(RXD) จำนวน 1 ช่อง ส่วนที่เหลืออีก 1 ช่อง จะปล่อยว่างไว้เป็นอิสระ เพื่อให้ผู้ใช้สามารถต่อทดลองการติดต่อสื่อสาร RS232 โดยใช้สัญญาณจาก Pin I/O อื่นๆของ dsPIC เช่น ต่อกับขาสัญญาณ RF2(RXD) และ RF3(TXD) ได้
- วงจร LED แสดงผลแบบ Sink Current ใช้ไฟเลี้ยง +5V โดยใช้ LED สีแดงขนาด 3mm. จำนวน 4 ชุด สำหรับใช้ในการทดสอบการทำงานของ Output ต่างๆ
- วงจรปรับแรงดัน 0-5V โดยใช้ตัวต้านทานปรับค่าได้แบบเก็อกม้าแบบมีแกนปรับ จำนวน 4 ชุด สำหรับใช้ในการทดสอบการทำงานของ A/D
- วงจร Push-Button Switch จำนวน 4 ชุด สำหรับใช้ทดสอบการทำงานของ Input ต่างๆ
- วงจร Mini Speaker สำหรับใช้ทดสอบการกำเนิดเสียง Beep หรือเสียงอื่นๆ
- พื้นที่สำหรับบัดกรีวงจรเพิ่มเติมขนาด 8cm x 4.5cm หรือใช้เป็นพื้นที่ติดตั้ง Photo Board รุ่น AD100 ขนาด 360 จุด สำหรับต่อทดลองวงจรต่างๆ

คุณสมบัติของ dsPIC30F2010

dsPIC30F2010 เป็น MCU ซึ่งใช้การประมวลผลข้อมูลแบบ 16 บิต จากค่าย Microchips ซึ่งมีจุดเด่นในด้านของความสามารถในการประมวลผลข้อมูลสัญญาณแบบดิจิทัล สำหรับนำไปประยุกต์ใช้งานควบคุมต่างๆ โดยโครงสร้างภายในจะเป็นการผสมผสานระหว่างไมโครคอนโทรลเลอร์ (MCU) และวงจร DSP (Digital Signal Processing) รวมเข้าไว้ด้วยกัน หรืออาจเรียก MCU ตระกูล dsPIC ว่าเป็น DSC หรือ Digital Signal Controller ก็ได้

ซึ่งในปัจจุบัน MCU ในตระกูล dsPIC ของ Microchips นั้นจะมีการผลิตออกมาจำหน่ายให้ผู้ใช้งานได้เลือกใช้งานกันอยู่มากมายหลายเบอร์ตามความเหมาะสมของงาน โดยปัจจุบัน (ตุลาคม 2548) dsPIC จะแบ่งออกเป็น 5 กลุ่ม ใหญ่ๆด้วยกัน ได้แก่ dsPIC30F20xx ,dsPIC30F30xx ,dsPIC30F40xx ,dsPIC30F50xx และ dsPIC30F60xx ซึ่งทุกเบอร์จะใช้โครงสร้างและสถาปัตยกรรมการประมวลผลแบบเดียวกันทั้งหมด แต่จะมีความแตกต่างกันในเรื่องของทรัพยากรภายใน เช่น ขนาดของหน่วยความจำใช้งาน จำนวนของ Peripheral I/O แบบต่างๆ ซึ่งอาจมีการบรรจุไว้ในแต่ละเบอร์ด้วยจำนวนที่ไม่เท่ากัน ซึ่งในที่นี้จะขอยกกล่าวถึงเพียงเฉพาะข้อมูลในส่วนที่เป็นของ dsPIC30F2010 เท่านั้น

คุณสมบัติด้านการประมวลผล

- ใช้สถาปัตยกรรมแบบ RISC โดยมี 84 คำสั่งมาตรฐาน รองรับการทำงานตำแหน่งแอดเดรสแบบต่างๆได้โดยอิสระ โดยรูปแบบโครงสร้างการจัดผังหน่วยความจำจะดัดแปลงมาจากสถาปัตยกรรมของ “Harvard Architecture”
- ชุดคำสั่ง ใช้การอ้างแอดเดรสแบบ 24บิต และการอ้างถึงข้อมูลขนาด 16บิต
- มีหน่วยความจำโปรแกรมแบบ Flash ขนาด 12KByte (4KWord) สามารถทำการลบและโปรแกรมซ้ำใหม่ได้กว่า 100,000 ครั้ง พร้อมระบบป้องกันการอ่าน
- มีหน่วยความจำ RAM ขนาด 512 Byte
- มีหน่วยความจำข้อมูลถาวรแบบ EEPROM ขนาด 1KByte สามารถลบและเขียนซ้ำได้กว่า 1,000,000 ครั้ง และสามารถเก็บรักษาข้อมูลไว้ได้แม้ไม่ได้จ่ายไฟเลี้ยงให้ MCU
- มีรีจิสเตอร์ขนาด 16 บิต ให้ใช้งานจำนวน 16 ชุด
- สามารถประมวลผลด้วยความเร็วสูงสุดที่ 30 MIPS(30 ล้านคำสั่งต่อวินาที)
- รองรับสัญญาณนาฬิกาจากแหล่งกำเนิดภายนอก 0-40 MHz
- รองรับการใช้งานกับแหล่งกำเนิดความถี่แบบ XTAL ค่า 4-10 MHz
- มีวงจรควบคุมความถี่ภายในแบบ Phase-Lock-Loop โดยสามารถกำหนดค่าอัตราการคูณความถี่ได้ 3 ระดับ คือ 4 เท่า ,8 เท่า และ 16 เท่า
- รองรับการทำงาน Interrupt ได้ถึง 27 แหล่ง พร้อมสัญญาณ Interrupt จากภายนอก 3 แหล่ง และสามารถจัดระดับความสำคัญของการ Interrupt ได้ 8 ระดับ

คุณสมบัติของ Peripheral I/O

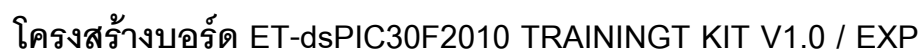
- ขาสัญญาณ I/O สามารถจ่ายกระแส(Source) และ รับกระแส (Sink) ได้มากถึง 25mA
- มี Timer ขนาด 16 บิต จำนวน 3 ชุด และสามารถโปรแกรมใช้งานเป็น Timer แบบ 32 บิต ได้โดยใช้ Timer 16 บิต 2 ช่องรวมกัน
- มี Input Capture ขนาด 16 บิต จำนวน 4 ช่อง
- มี Output Compare/PWM ขนาด 16บิต จำนวน 2 ช่อง
- มีวงจรสื่อสารอนุกรมแบบ SPI จำนวน 1 ช่อง
- มีวงจรสื่อสารอนุกรมแบบ I2C จำนวน 1 ช่อง
- มีวงจรสื่อสารอนุกรมแบบ UART จำนวน 1 ช่อง
- มีวงจร DCPWM สำหรับใช้ควบคุมมอเตอร์ 3 ช่อง
- มีวงจรถอดรหัสแบบ QEIM ขนาด 16บิต จำนวน 1 ช่อง
- มีวงจร A/D ขนาด 10บิต จำนวน 6 ช่อง

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™	I ² C™	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

ตาราง แสดงความแตกต่างของ dsPIC เบอร์ต่างๆ

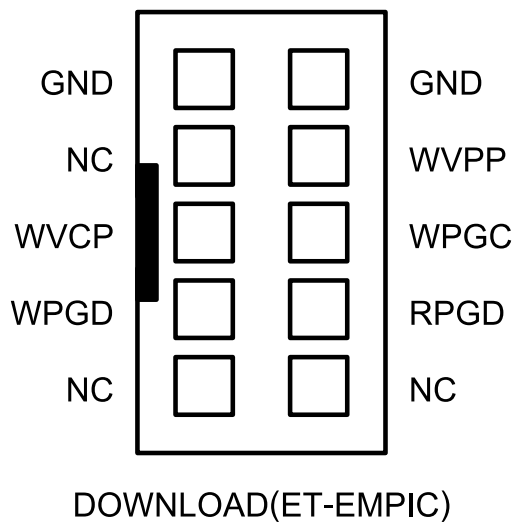
MCLR	1	28	AVDD
EMUD3/AN0/VREF+/CN2/RB0	2	27	AVss
EMUC3/AN1/VREF-/CN3/RB1	3	26	PWM1L/RE0
AN2/SS1/CN4/RB2	4	25	PWM1H/RE1
AN3/INDX/CN5/RB3	5	24	PWM2L/RE2
AN4/QEA/IC7/CN6/RB4	6	23	PWM2H/RE3
AN5/QEB/IC8/CN7/RB5	7	22	PWM3L/RE4
Vss	8	21	PWM3H/RE5
OSC1/CLKI	9	20	VDD
OSC2/CLKO/RC15	10	19	Vss
EMUD1/SOSCI/T2CK/U1ATX/CN1//RC13	11	18	PGC/EMUC/U1RX/SDI1/SDA/RF2
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	12	17	PGD/EMUD/U1TX/SDO1/SCL/RF3
VDD	13	16	FLTA/INT0/SCK1/OCFA/RE8
EMUD2/OC2/IC2/INT2/RD1	14	15	EMUC2/OC1/IC1/INT1/RD0

รูปแสดง การจัดเรียงขาสัญญาณของ dsPIC30F2010

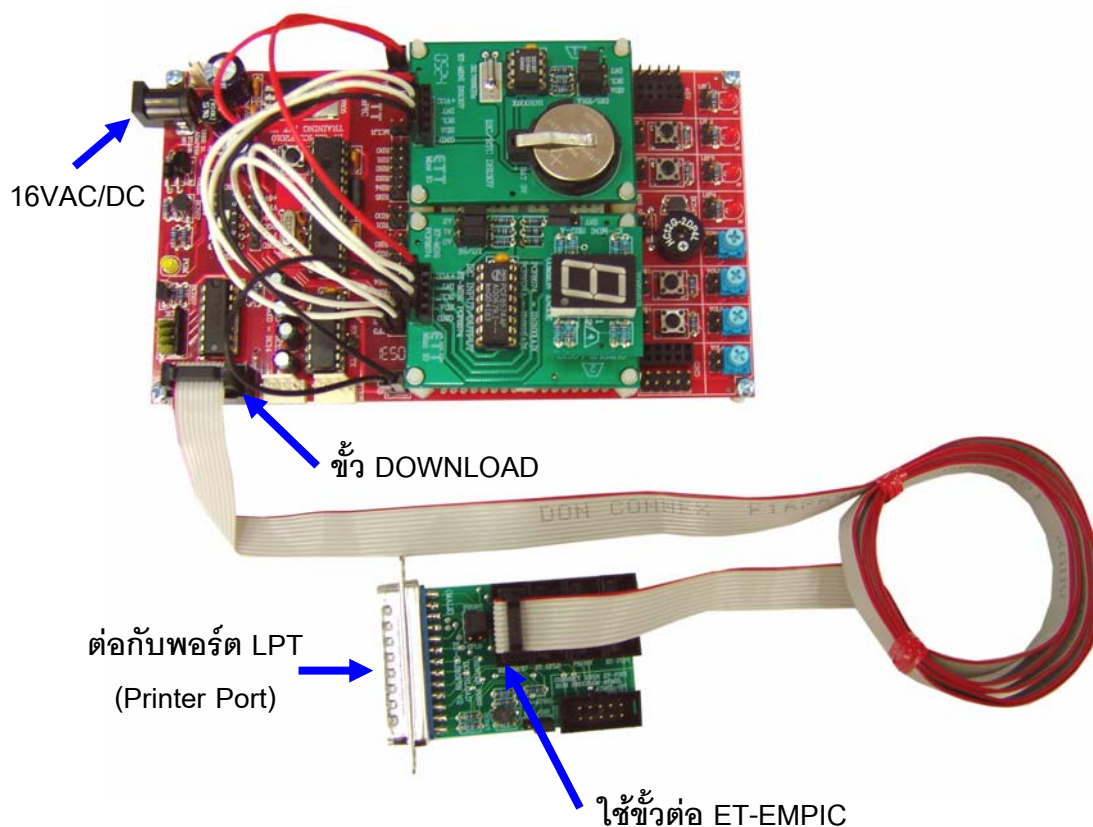


WWW.ETT.CO.TH

ขั้วต่อ DOWNLOAD (ET-EMPIC)



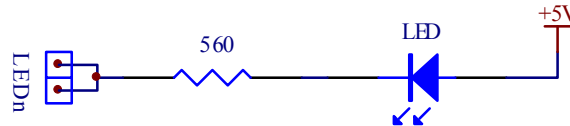
ขั้วต่อ “DOWNLOAD” นี้ จะใช้สำหรับ เชื่อมต่อกับชุด “ET-CAP10PIN V2.0” ผ่านทาง ขั้วต่อของชุดที่ชื่อว่า “ET-EMPIC” โดยใช้สาย แพร่ขนาด 10 Pin เป็นสายเชื่อมต่อสัญญาณ ระหว่างบอร์ดทั้งสองเพื่อใช้สำหรับสั่ง Download HEX File ให้กับหน่วยความจำโปรแกรม (Flash) ของ MCU ในบอร์ด “ET-dsPIC30F2010” ในขณะที่เลือกสวิตช์ “PROG/RUN” ไว้ทางด้าน “PROG” โดยสามารถใช้โปรแกรม “WinPic800” หรือ โปรแกรม “Melabs Programmer” เป็นเป็น โปรแกรมสำหรับสั่งงานก็ได้ตามต้องการ



รูปแสดง การเตรียมการ Download ของบอร์ด ET-dsPIC30F2010

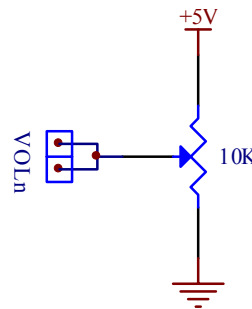
การใช้งาน LED แสดงผล

LED แสดงผลของบอร์ด จะต้องวงจรแบบรับกระแส (Sink Current) โดยใช้กับแหล่งจ่าย +5V ทำงานด้วยลอจิก “0” (0V) และหยุดทำงานด้วยลอจิก “1” (+5V) โดยมีทั้งหมด 4 ชุด โดยวงจรในส่วนนี้จะใช้สำหรับทดสอบการทำงานของ Port I/O ต่างๆที่ทำงานให้ผลเป็น Output แบบลอจิก



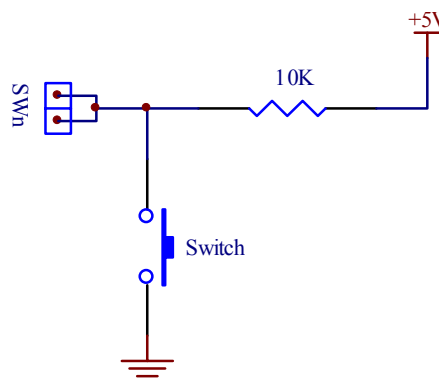
การใช้งานวงจรปรับแรงดัน (0V-5V)

วงจรปรับแรงดันของบอร์ดจะใช้ตัวต้านทานปรับค่าได้แบบเกือกม้า ชนิดมีแกนหมุนสำหรับปรับค่า โดยวงจรนี้ใช้กับแหล่งจ่าย +5V โดยจะให้ Output เป็นแรงดันซึ่งมีค่าระหว่าง 0V ถึง +5V ตามการปรับค่าของตัวต้านทาน ซึ่งมีทั้งหมด 4 ชุด ใช้สำหรับสร้างแรงดัน Input เพื่อทดสอบการทำงานของวงจร A/D



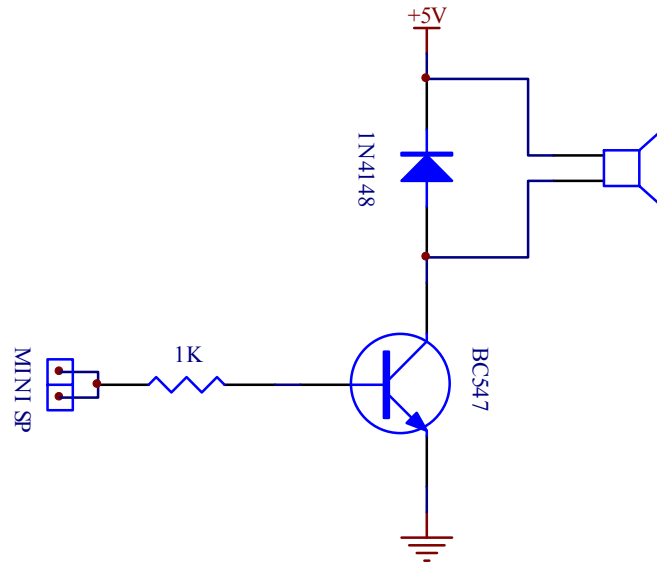
การใช้งานวงจร Push Button Switch

วงจร Push Button Switch จะใช้วงจร Switch แบบ กดติด-ปล่อยดับ (Push Button) พร้อมวงจร Pull-Up ใช้กับแหล่งจ่าย +5V โดยในขณะที่สวิตช์ยังไม่ถูกกดจะให้ค่าสถานะเป็นลอจิก “1” แต่เมื่อสวิตช์ถูกกดอยู่จะให้สถานะเป็นลอจิก “0” โดยวงจรส่วนนี้จะมียู닛ด้วยกัน 4 ชุด ใช้สำหรับทดสอบการทำงานของ I/O ต่างๆที่ต้องควบคุมการทำงานของวงจรด้วย Input แบบลอจิก



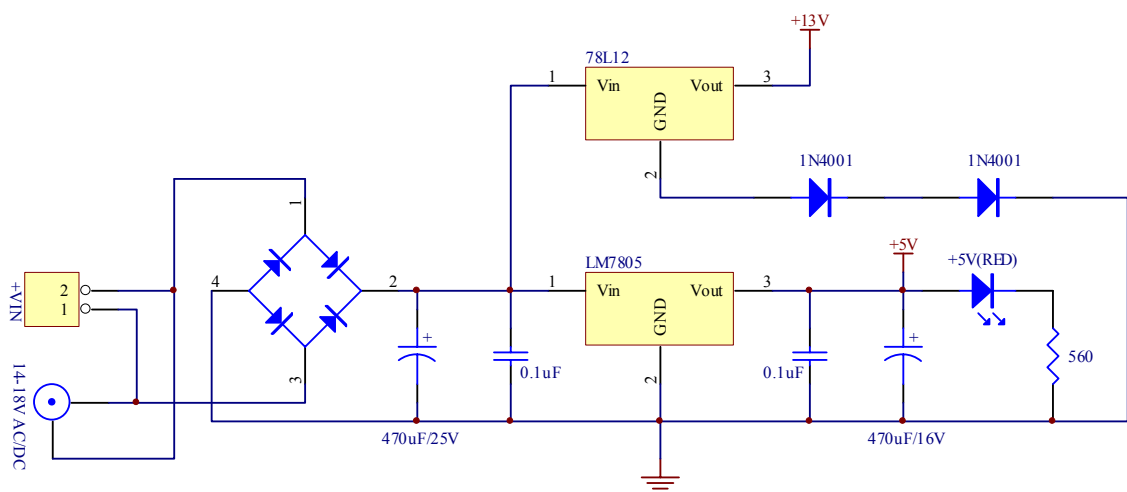
การใช้งาน วงจรกำเนิดเสียง

วงจรกำเนิดเสียง จะใช้ลำโพงขนาดเล็ก (Mini Speaker) พร้อมด้วยวงจรทรานซิสเตอร์แบบ NPN สำหรับขับกระแสให้กับลำโพง ใช้กับแหล่งจ่ายขนาด +5V ทำงานด้วยลอจิก “1” และหยุดทำงานด้วยลอจิก “0” โดยในการทำงานนั้นต้องส่งสัญญาณลอจิกที่เป็นความถี่ต่างๆให้กับลำโพงเพื่อสร้างเป็นความถี่เสียงย่านต่างๆ ตามต้องการ



วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟสามารถใช้งานได้กับไฟ AC และ DC ขนาด 14-18V ได้ทันที โดยวงจรภาคแหล่งจ่ายไฟในส่วนที่เป็นวงจร Regulate นั้นจะมีทั้งส่วนที่เป็น +5V / 1A และ +13V/100mA โดยส่วนที่เป็น +5V จะถูกใช้สำหรับเป็นแหล่งจ่ายไฟเพื่อจ่ายให้กับ MCU และวงจร I/O ต่างๆภายในบอร์ด ส่วนแรงดัน +13V จะใช้สำหรับเป็นแรงดันสำหรับการ Download โปรแกรม แบบ ICSP ของบอร์ด



การพัฒนาโปรแกรมของบอร์ด

สำหรับการพัฒนาโปรแกรมของบอร์ด “ET-dsPIC30F2010 TRAINING KIT V1.0 / EXP” 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-dsPIC30F2010” นั้นสามารถเลือกได้หลายแนวทาง ไม่ว่าจะเป็นการใช้เครื่องมือสำหรับโปรแกรมข้อมูลให้กับ MCU โดยใช้เครื่อง Programmer ที่สนับสนุนการโปรแกรม MCU เบอร์ dsPIC30F2010 หรือใช้ชุดพัฒนาโปรแกรมจำพวก In-Circuit Debugger ที่สนับสนุนการใช้งานร่วมกับ MCU เบอร์ dsPIC30F2010 ซึ่งทั้ง 2 แนวทางข้างต้นที่กล่าวมาแล้วนั้น ทางผู้ใช้งานจะต้องเสียค่าใช้จ่ายในการจัดหาเครื่องมือมาใช้งานเอง แต่อย่างไรก็ตามทางทีมงาน อีทีที ได้ออกแบบวงจรของบอร์ด “ET-dsPIC30F2010” ให้สามารถทำการ Download Hex File จากคอมพิวเตอร์ PC โดยใช้สัญญาณควบคุมจากพอร์ตขนาน หรือ พอร์ต Printer (LPT) โดยใช้ชุดสายต่อสัญญาณรุ่น “ET-CAB10PIN V2.0” เป็นตัวกลางในการเชื่อมต่อได้ทันที โดยที่ผู้ใช้ไม่จำเป็นต้องจัดหาเครื่องมืออื่นใดมาช่วยให้สิ้นเปลืองอีก แต่อย่างไรก็ตามการพัฒนาโปรแกรมของบอร์ดด้วยวิธีการนี้ จะมีข้อจำกัดอยู่ 2 ประการ เมื่อเทียบกับการใช้ชุด “In-Circuit Debugger” ในการพัฒนาโปรแกรม ก็คือ

- ในปัจจุบัน โปรแกรม MPLAB ยังไม่ยอมให้มีการเชื่อมต่อเครื่องมือโปรแกรมอื่นๆที่ไม่ใช่เครื่องมือของ Microchips ผ่านทางเมนูคำสั่งของ MPLAB เอง ดังนั้นผู้ใช้งานจะต้องแยกการทำงานระหว่างการพัฒนาโปรแกรม (เขียน Source โปรแกรม และ แปลเป็น Hex File) และการ Download โปรแกรม (โปรแกรม Hex File ให้หน่วยความจำของ MCU)
- ไม่สามารถตรวจสอบข้อผิดพลาดของโปรแกรมในระหว่างการสั่งให้ MCU ทำงานอยู่ได้ แต่ทำได้เพียงการสั่งโปรแกรม (Download Hex File ให้ MCU) และ ดูผลการทำงานของ MCU ในขณะที่ MCU ทำงานจริงๆ ซึ่งถ้าโปรแกรมที่เขียนขึ้นมาไม่ทำงานหรือทำงานแล้วเกิดความผิดพลาดขึ้น ผู้ใช้จะต้องใช้ความสามารถ ในการ คาดเดา และตรวจสอบหาข้อผิดพลาดเหล่านั้นจาก Source Code ของโปรแกรมเอง ว่าน่าจะมีข้อผิดพลาดจากส่วนใดของโปรแกรมบ้าง แล้วจึงสั่งแก้ไขข้อผิดพลาดนั้นๆ พร้อมกับสั่งแปลโปรแกรมซ้ำใหม่ แล้วจึงสั่ง Download Hex File ให้ MCU ใหม่เพื่อดูผลการทำงาน

การเขียนโปรแกรมใช้งานกับบอร์ดโดยใช้ MPLAB C30

MPLAB C30 หรือ C30 Tools เป็นโปรแกรมภาษาซี สำหรับใช้แปลคำสั่งของ MCU ตระกูล dsPIC ซึ่งได้รับการพัฒนาขึ้นโดย Microchips เอง โดยข้อกำหนดและรายละเอียดของการเขียนโปรแกรมภาษาซี นั้น จะไม่กล่าวถึงในที่นี้ด้วย โดยถ้าผู้ใช้ต้องการพัฒนาโปรแกรมให้กับ dsPIC ด้วยภาษาซี แต่ยังไม่มีความรู้เรื่องการเขียนโปรแกรมภาษาซีเลยนั้นขอแนะนำให้ หาหนังสือที่อธิบายเกี่ยวกับการเขียนโปรแกรมภาษาซีในส่วนที่เป็นมาตรฐานตามข้อกำหนดของ “ANSI C” มาศึกษาให้เข้าใจเสียก่อน และสำหรับส่วนของข้อกำหนดปลีกย่อยอื่นที่เป็นของ MPLAB C30 เองก็สามารถอ่านเพิ่มเติมได้จากเอกสารและคู่มือการใช้งานของ MPLAB C30 ที่ทาง Microchips จัดทำไว้ได้ โดยสามารถ Download จาก Website ของ Microchips หรือจาก Folder ของ “..\pic30_tools\docs” ที่ทำการติดตั้งโปรแกรม MPLAB C30 ไว้ก็ได้ โดยในที่นี้ จะขอกล่าวแนะนำถึงเฉพาะส่วนของการกำหนดค่าตัวเลือกในโปรแกรมเพื่อใช้งานร่วมกับบอร์ด “ET-dsPIC30F2010” 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 โดยในปัจจุบัน(ตุลาคม 2548) จะเป็น Version 1.33 ซึ่งตามปกติแล้วโปรแกรมชุดนี้จะต้องซื้อ มาใช้งานเอง แต่อย่างไรก็ตามทาง Microchips เองมีรุ่นทดลองใช้งานให้ผู้ใช้งานสามารถ Download มาใช้งานได้เช่นเดียวกันกับ MPLAB IDE

โดยโปรแกรมทั้ง 2 ชุดนี้ ทางอีทีที ได้ทำการ Download มาจัดเตรียมไว้ให้ในแผ่น CD-ROM ที่แถมไปกับบอร์ดของ “ET-dsPIC30F2010” 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 ของการติดตั้งโปรแกรม คือ “C:\PIC30_TOOLS\” จะเกิดความสะดวกต่อการใช้งานมากที่สุด โดยเฉพาะในขั้นตอนของการกำหนดการเชื่อมโยงการทำงานระหว่าง MPLAB IDE และ MPLAB C30 โดยในที่นี้จะขออธิบายโดยอ้างถึงตำแหน่งการติดตั้งโปรแกรมหดตั้งที่กล่าวไว้แล้วข้างต้นเท่านั้น ซึ่งถ้าผู้ใช้ทำการสั่งติดตั้งโปรแกรมไว้ยังตำแหน่ง Folder ที่แตกต่างไปจากนี้แล้วขอให้ทำความเข้าใจและดัดแปลงวิธีการกำหนดค่าเองตามที่ติดตั้งโปรแกรมไว้จริงๆด้วย

การกำหนดการเชื่อมโยงการทำงานของ MPLAB IDE และ MPLAB C30

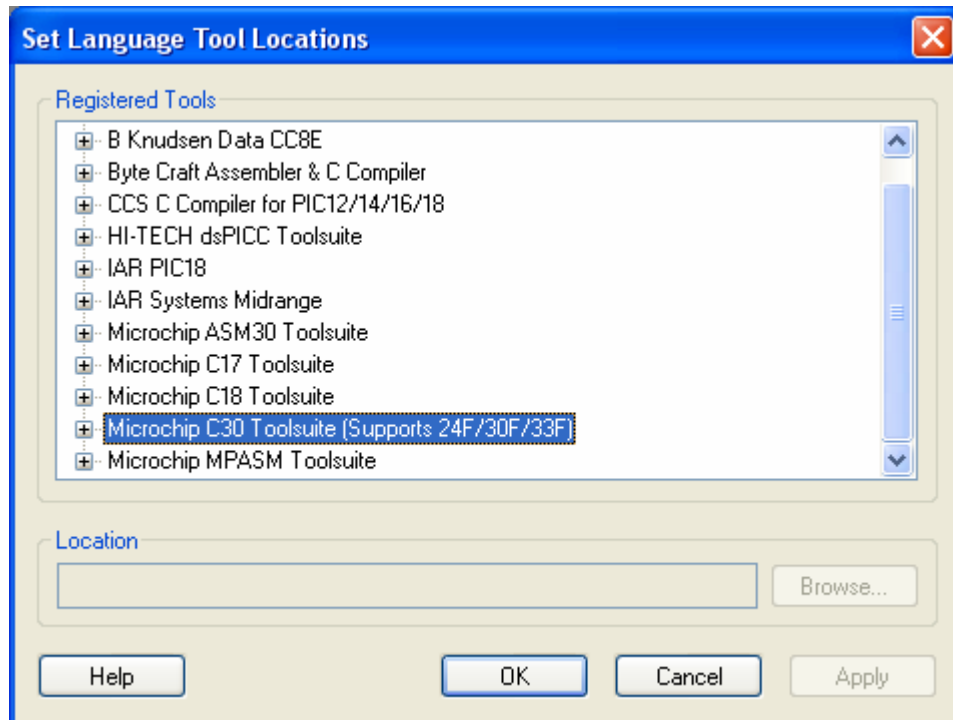
หลังจากทำการติดตั้งโปรแกรม MPLAB IDE และ MPLAB C30 เสร็จเรียบร้อยแล้ว ก่อนที่จะเริ่มต้นใช้งานโปรแกรม MPLAB C30 ได้นั้น ในครั้งแรกจะต้องทำการตั้งค่าการเชื่อมโยงการทำงานระหว่าง MPLAB IDE และ MPLAB C30 ให้ถูกต้องเรียบร้อยแล้ว จึงจะสามารถเรียกใช้งานโปรแกรม MPLAB C30 ผ่านทางโปรแกรม MPLAB IDE ได้อย่างถูกต้อง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

โดยในชุดโปรแกรมของ MPLAB C30 หลังจากติดตั้งโปรแกรมไปแล้ว โปรแกรม ใช้งานต่างๆจะถูกเก็บไว้ใน Folder ชื่อ “C:\PIC30_TOOSL\BIN\” โดยจะมีโปรแกรมหลักๆที่ต้องกำหนดการเชื่อมโยงการทำงานกับ MPLAB IDE อยู่ด้วยกัน 4 โปรแกรมด้วยกันคือ

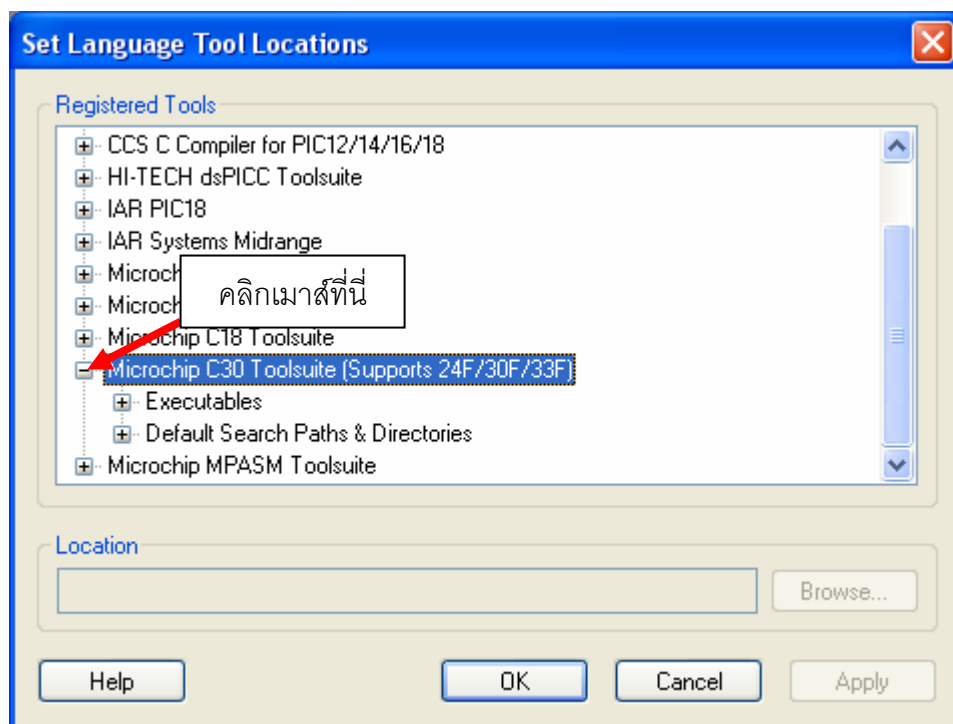
- ไฟล์ “pic30-as.exe” ซึ่งเป็นตัวโปรแกรมหลักสำหรับการใช้งานในการแปลคำสั่งภาษาแอสเซมบลีของ dsPIC (Assembler)
- ไฟล์ “pic30-gcc.exe” ซึ่งเป็นโปรแกรมหลักสำหรับการใช้งานในการแปลคำสั่งภาษาซีของ dsPIC (C Compiler)
- ไฟล์ “pic30-ld.exe” ซึ่งเป็นโปรแกรมหลักสำหรับการรวมไฟล์ต่างๆเข้าด้วยกันเพื่อสร้างเป็น Hex File ของ dsPIC (Linker)
- ไฟล์ “pic30-ar.exe” ซึ่งเป็นโปรแกรมหลักในการจัดการกับ Library

ซึ่งในอันดับแรกก่อนที่จะเริ่มต้นเข้าสู่ขั้นตอนของการใช้งานนั้น จะต้องทำการกำหนดการเชื่อมโยงคำสั่ง ระหว่างโปรแกรม MPLAB IDE และ MPLAB C30 ให้เรียบร้อยแล้ว เพื่อให้โปรแกรม MPLAB IDE จะได้ทราบว่าต้องไปเรียกใช้ไฟล์ต่างๆของ 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

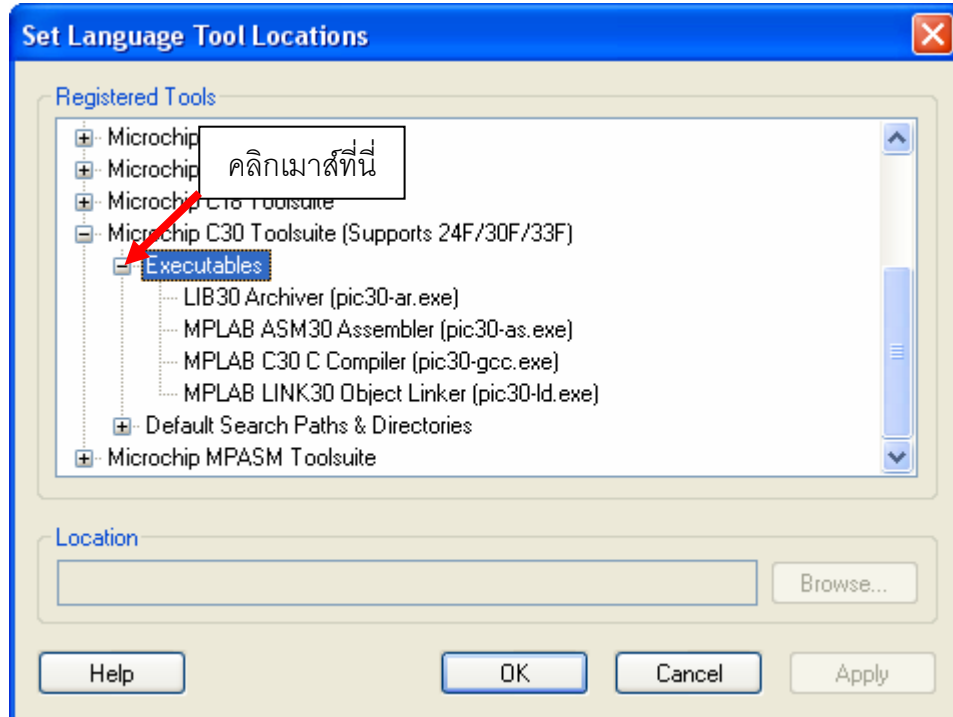


เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่บริเวณตำแหน่งเครื่องหมายบวก(+) ที่หน้าคำสั่งของ Microchip C30 Toolsuite (Supports 24F/30F/33F) ซึ่งจะได้ผลดังรูป



ซึ่งจะเห็นได้ว่าในหัวข้อ Microchip C30 Toolsuite (Supports 24F/30F/33F) นี้จะประกอบไปด้วย หัวข้อย่อยอีก 2 หัวข้อ คือ Executables และ Default Search Paths & Directories

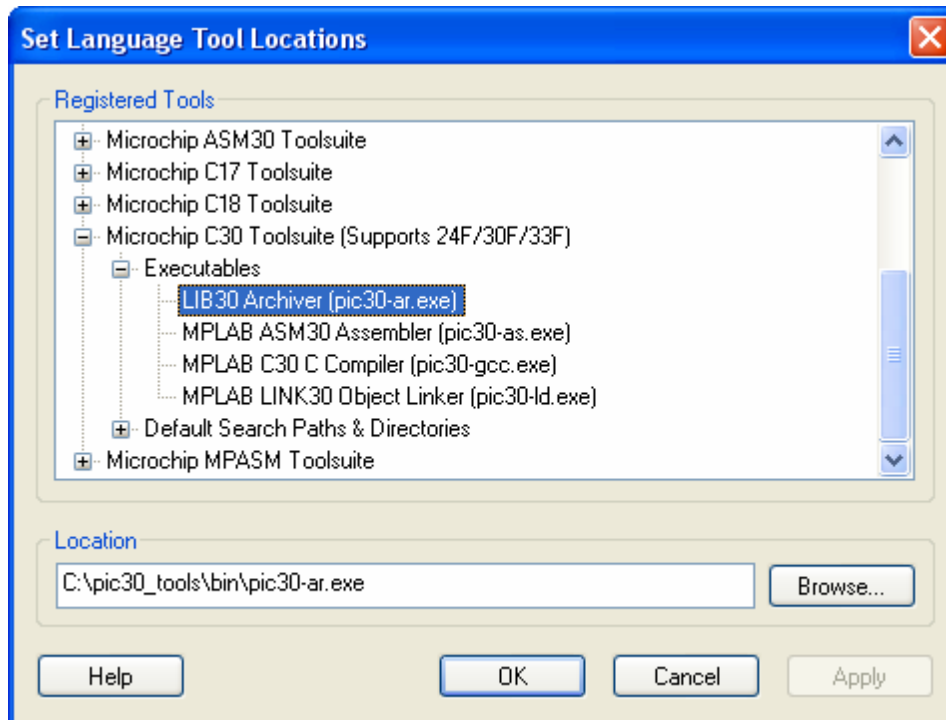
ให้ทำการกำหนดตำแหน่งของไฟล์ของ MPLAB C30 ที่ต้องการให้ MPLAB IDE เรียกใช้ ซึ่งจะมีอยู่ด้วยกันทั้งหมด 4 ไฟล์ โดยให้คลิกเมาส์ ที่บริเวณตำแหน่งเครื่องหมายบวก (+) ที่หน้าหัวข้อ Executables ซึ่งจะได้ผลดังรูป



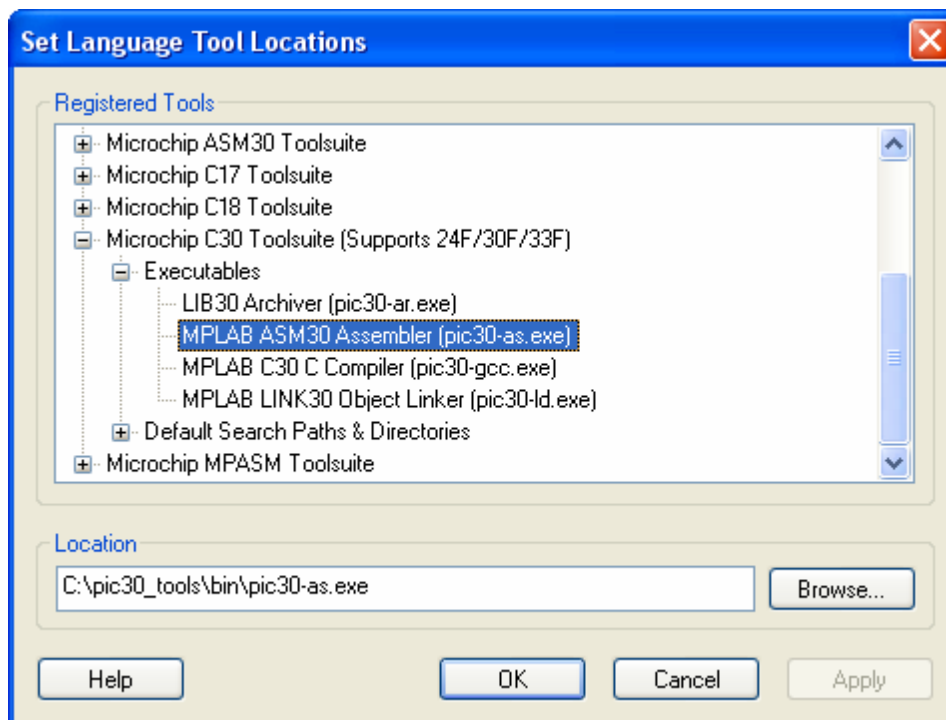
ในขั้นตอนนี้ให้ทำการกำหนดชื่อ และ ตำแหน่ง Folder ที่อยู่ของไฟล์ทั้ง 4 ซึ่งได้แก่ pic30-ar.exe, pic30-as.exe, pic30-gcc.exe และ pic30-ld.exe โดยให้ทำการคลิกเมาส์ที่รายการย่อยของแต่ละหัวข้อ จนปรากฏแถบสีน้ำเงินที่หัวข้อนั้นๆ จากนั้นก็ให้กำหนดตำแหน่ง Folder และชื่อของไฟล์ ให้กับแต่ละหัวข้อ จนครบทั้ง 4 หัวข้อ โดยชื่อไฟล์นั้นต้องกำหนดตามชื่อที่อยู่ในวงเล็บท้ายหัวข้อ ส่วนตำแหน่ง Folder นั้น ตามปกติแล้วจะอยู่ที่ “..\pic30_tools\bin\” เช่น ถ้าติดตั้งโปรแกรม MPLAB C30 ไว้ตามที่แนะนำไว้ในตัวอย่างคือ “c:\pic30_tools\” ไฟล์ที่ใช้สั่ง Run (Execute) ทั้งหมดจะอยู่ที่ “c:\pic30_tools\bin\..”

โดยวิธีการกำหนดชื่อและตำแหน่ง Folder ที่อยู่ของไฟล์ นั้นสามารถทำได้ 2 แบบ คือ การสังคลิกเมาส์ที่ “Browse...” เพื่อกำหนดตำแหน่งที่อยู่ของ Folder ซึ่งเก็บ Execute File ไว้ โดยให้ชี้ไปที่ “..\pic30_tools\bin\..” ตัวอย่างเช่น ถ้าต้องการกำหนดชื่อ Execute File และตำแหน่ง Folder ของไฟล์สำหรับ LIB30 Archiver (pic30-ar.exe) ต้องกำหนดชื่อไฟล์เป็น “pic30-ar.exe” โดยให้คลิกเมาส์ที่ “Browse...” แล้วชี้ไปที่ “c:\pic30_tools\bin\pic30-ar.exe” หรืออาจใช้วิธีการพิมพ์ ชื่อ Folder ในช่อง Location เองเป็น “c:\pic30_tools\bin\pic30-ar.exe” ก็ได้ (ถ้าติดตั้งโปรแกรมไว้ต่างจากนี้ต้องกำหนดให้ตรงกับที่ติดตั้งไว้จริงด้วย)

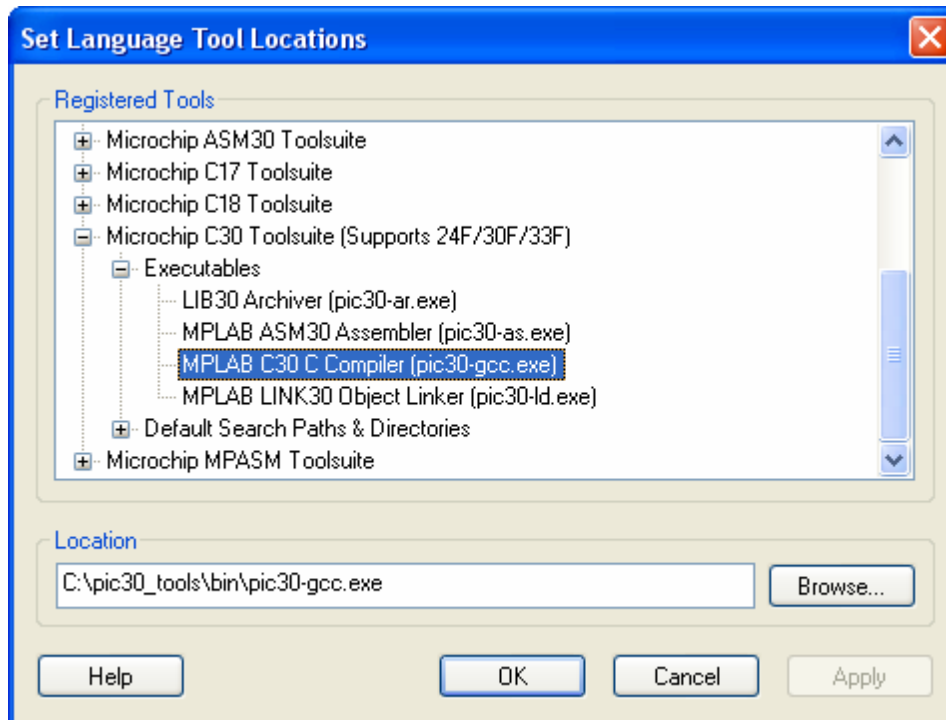
โดยในขั้นตอนนี้ ให้กำหนดชื่อและตำแหน่ง Folder ของ Execute File ให้ครบทั้ง 4 หัวข้อ ด้วย ดังรูปในตัวอย่างต่อไปนี้



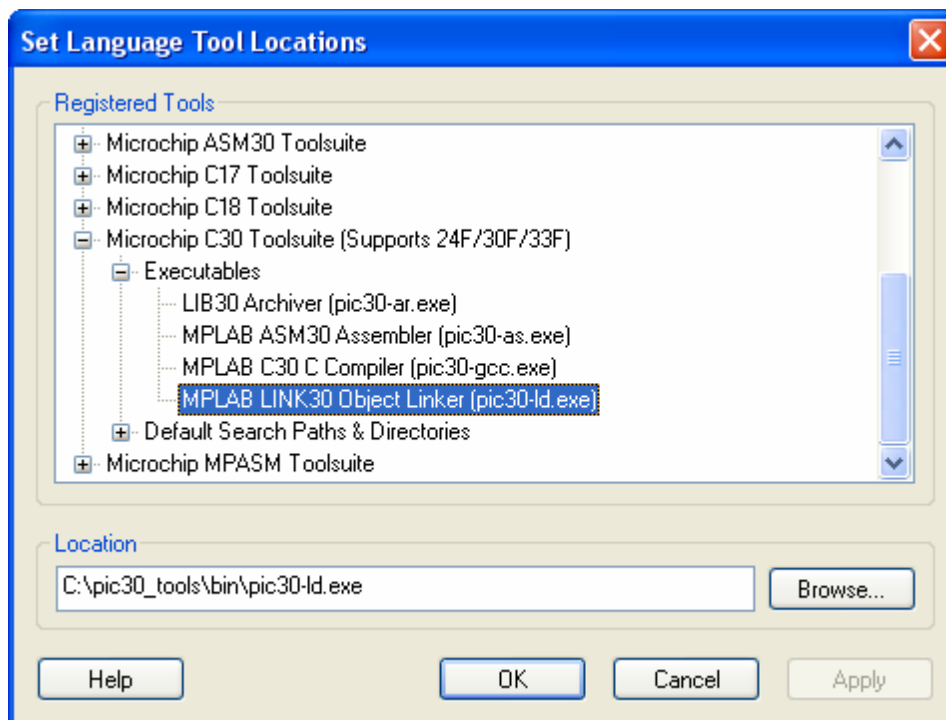
รูปแสดง ลักษณะการกำหนด Execute File ของ LIB30 Archiver



รูปแสดง ลักษณะการกำหนด Execute File ของ MPLAB ASM30 Assembler

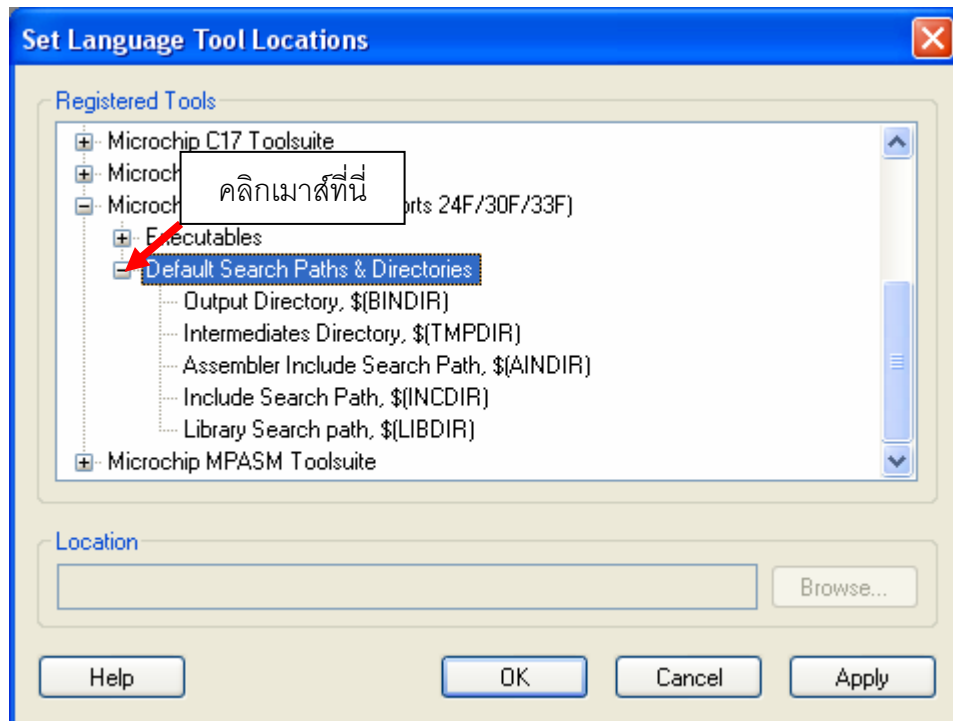


รูปแสดง ลักษณะการกำหนด Execute File ของ MPLAB C30 C Compiler

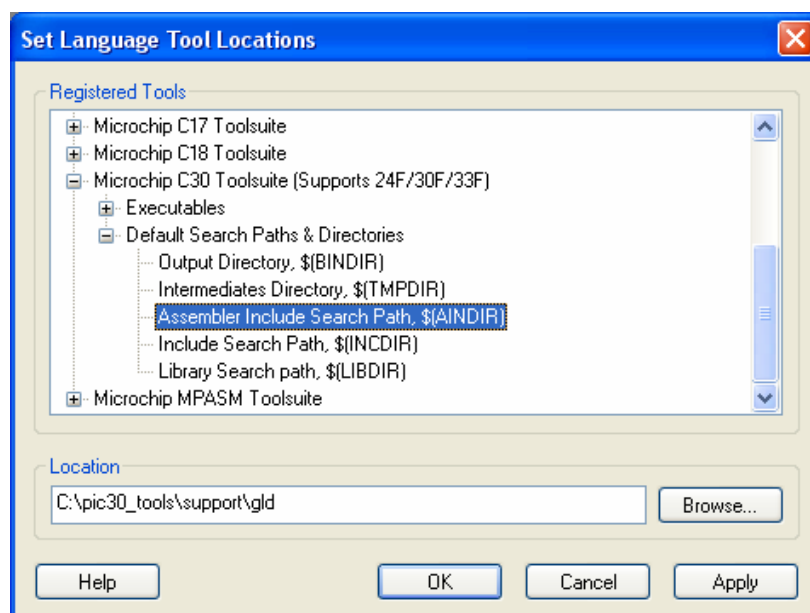


รูปแสดง ลักษณะการกำหนด Execute File ของ MPLAB LINK30 Object Linker

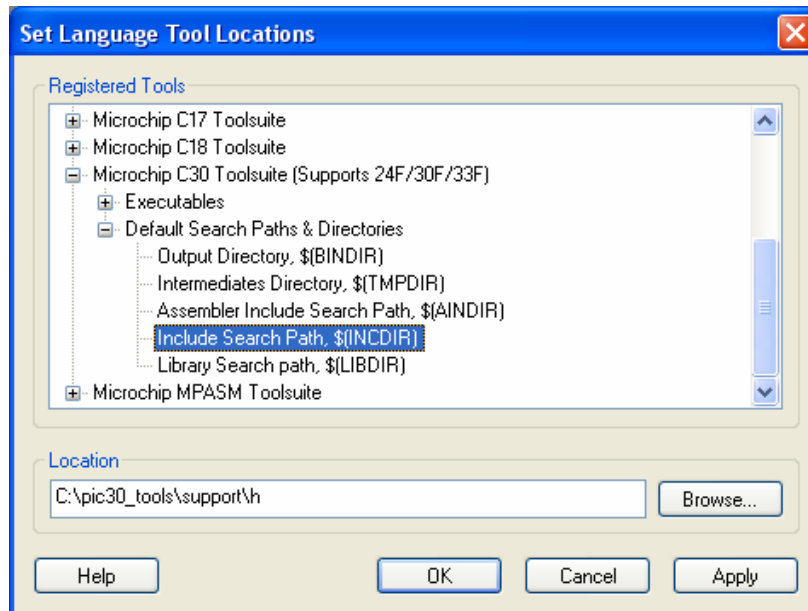
เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่บริเวณตำแหน่งเครื่องหมายบวก(+) ที่หน้าคำสั่งของ Default Search Paths & Directories ซึ่งจะได้ผลดังรูป



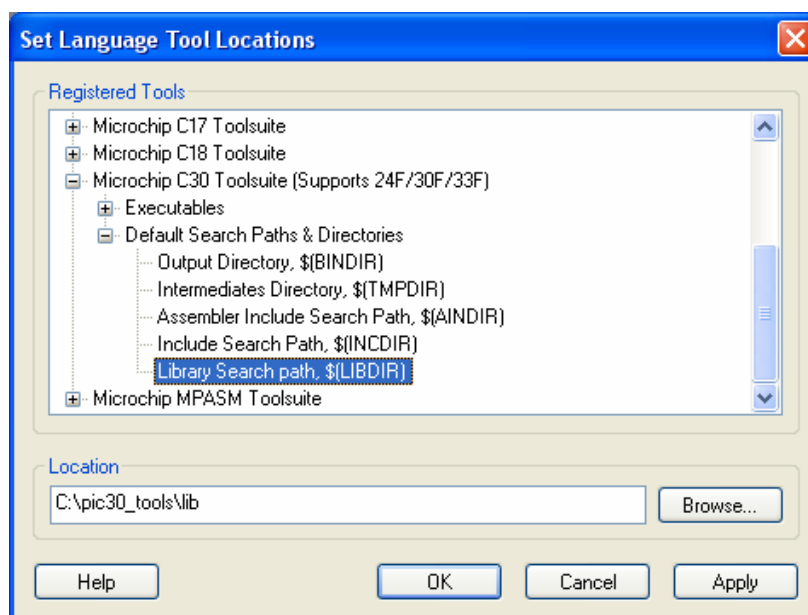
เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่ “Assembler Include Search Path, \$(AINDIR)” จนปรากฏแถบสีน้ำเงินดังรูป แล้วจึงคลิกเมาส์ที่ “Browse...” เพื่อกำหนดตำแหน่งที่อยู่ของ Folder ซึ่งเก็บ Source Code ภาษา Assembly ไว้ โดยให้ชี้ไปที่ “..\pic30_tools\support\gld” หรืออาจใช้วิธีการพิมพ์ ชื่อ Folder ในช่อง Location เองก็ได้ดังรูป (ถ้าติดตั้งโปรแกรมไว้ต่างจากนี้ต้องกำหนดให้ตรงด้วย)



เมื่อมาถึงขั้นตอนนี้ให้ทำการคลิกเมาส์ที่ “Include Search Path, \$(INCDIR)” จนปรากฏแถบสีน้ำเงินดังรูป แล้วจึงคลิกเมาส์ที่ “Browse...” เพื่อกำหนดตำแหน่งที่อยู่ของ Folder ซึ่งเก็บ Header File ไว้ โดยในการกำหนดตำแหน่ง Folder นั้นให้ชี้ไปที่ “..\pic30_tools\support\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 “..\pic30_tools\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 จนกว่าจะมีการเปลี่ยนแปลงใหม่ ซึ่งในที่นี้จะขอยกตัวอย่างการเขียนโปรแกรมภาษาซี ของ MPLAB C30 สัก 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

```

MPLAB IDE v7.21 - [Untitled*]
File Edit View Project Debugger Programmer Tools Configure Window Help

Checksum: 0xd406

/* **** Examples Program For "ET-dsPIC30F2010" Board **** */
/* Hardware      : ET-dsPIC30F2010 TRAINING KIT V1 */
/* Target MCU    : dsPIC30F2010 */
/* X-TAL        : 7.3728 MHz */
/* Run          : 117.9648MHz */
/* Select OSC Mode = XT w/PLL 16x */
/* Compiler      : MPLAB + C30 V1.33 */
/* Last Update   : 1/October/2005 */
/* Function      : Example Use RB0 on Output Mode */
/*               : LED Blink on RB0 */
/* **** */

#include <p30f2010.h>                                // dsPIC30F2010 MPU Register

/* Setup Configuration For ET-dsPIC30F2010 */
_FOSC(CSW_FSCM_ON & XT_PLL16);                      // Enable Clock Switching, Enable Fail-Safe Clock
// Clock Source = Primary XT + (PLL x 16)
_FWDT(WDT_OFF);                                     // Disable Watchdog
_FBORPOR(PBOR_ON & BORV_45 & PWRT_64 & MCLR_EN);    // Enable Brown-Out = 4.5V, Power ON = 64ms, Enable MCLR
_FCS(CODE_PROT_OFF);                                // Code Protect OFF
/* End Configuration For ET-dsPIC30F2010 */

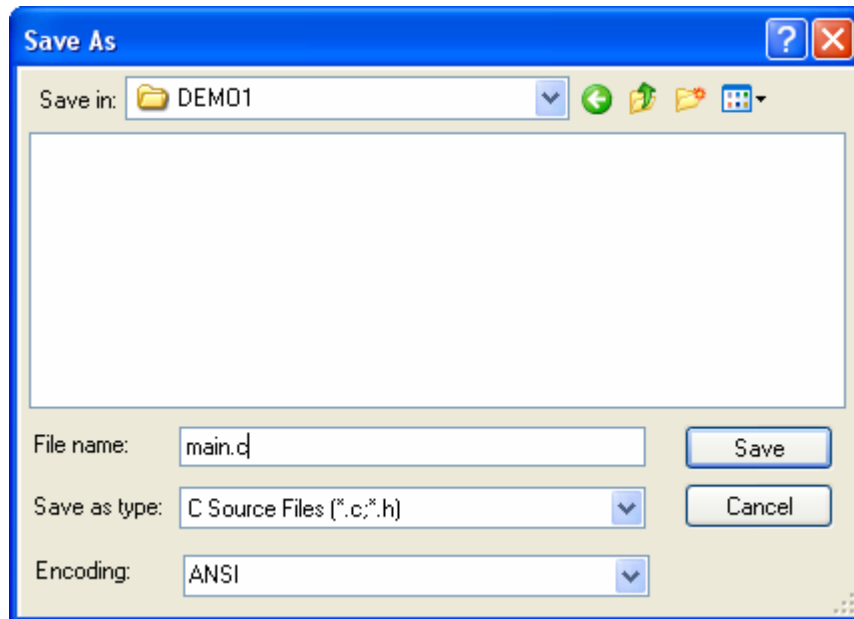
/* prototype section */
void delay_led(unsigned long int);                    // Delay Time Function

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

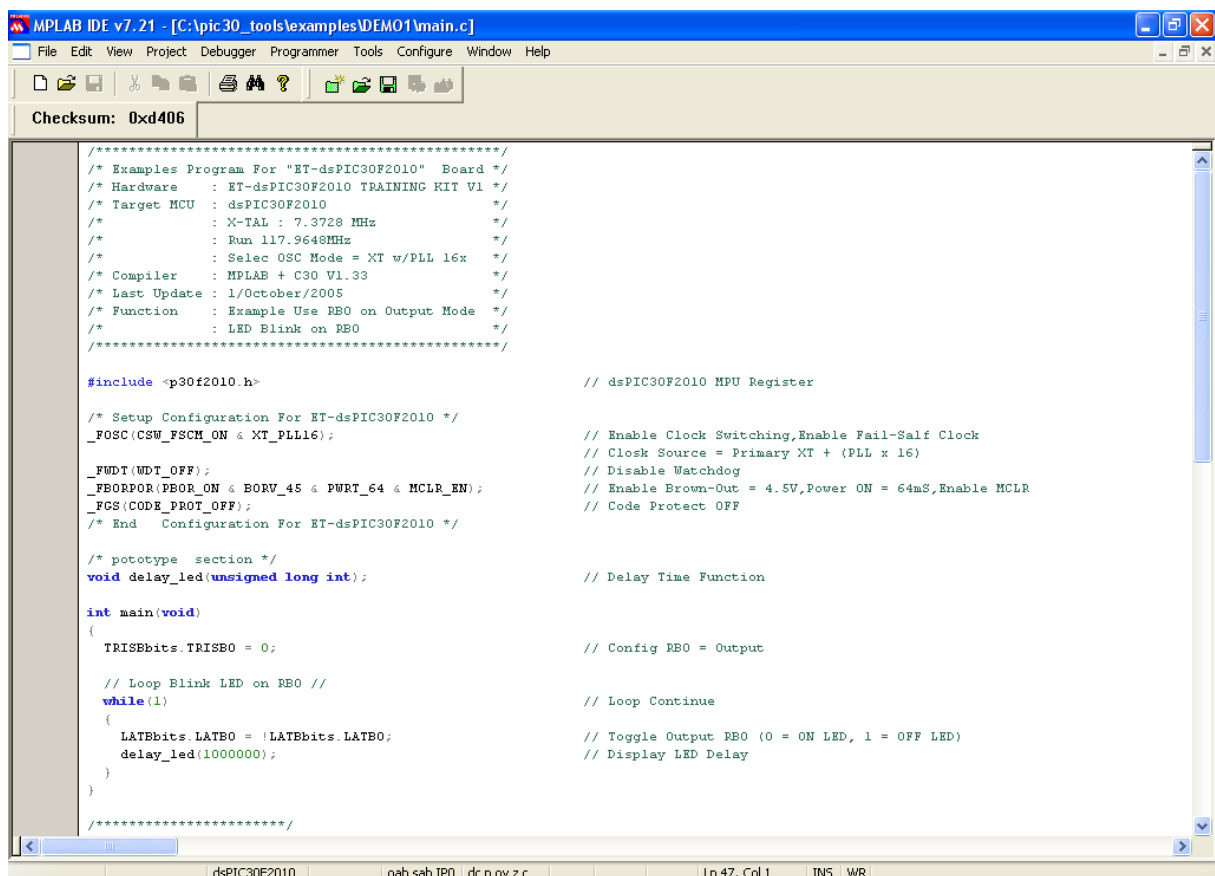
    // Loop Blink LED on RB0 //
    while(1)                                          // Loop Continue
    {
        LATBbits.LATB0 = !LATBbits.LATB0;           // Toggle Output RB0 (0 = ON LED, 1 = OFF LED)
        delay_led(1000000);                          // Display LED Delay
    }
}

/* **** */
  
```

2. ทำการสั่งบันทึกไฟล์ที่เขียนขึ้นให้เป็น Text File ภาษาซี โดยให้ทดลองกำหนดชื่อเป็น “main.c” แล้วสั่งบันทึกไว้ใน Folder ชื่อ “c:\pic30_tools\examples\demo1\main.c” 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 <p30f2010.h> // dsPIC30F2010 MPU Register

_FOSC(CSW_FSCM_ON & XT_PLL16); // Enable Clock Switching
// Enable Fail-Self Clock
// Clock Source = Primary XT
// Enable PLL x 16

_FWDT(WDT_OFF); // Disable Watchdog
_FBORPOR(PBOR_ON & BORV_45 & // Enable Brown-Out = 4.5V
          PWRT_64 & // Power ON = 64mS
          MCLR_EN); // Enable MCLR
_FGS(CODE_PROT_OFF); // Code Protect OFF

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

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

    while(1) // Loop Continue
    {
        LATBbits.LATB0=!LATBbits.LATB0; // Toggle Output RB0
        delay_led(1000000); // Display LED Delay
    }
}

void delay_led(unsigned long int count1)
{
    while(count1 > 0){count1--;} // Loop Decrease Counter
}

```

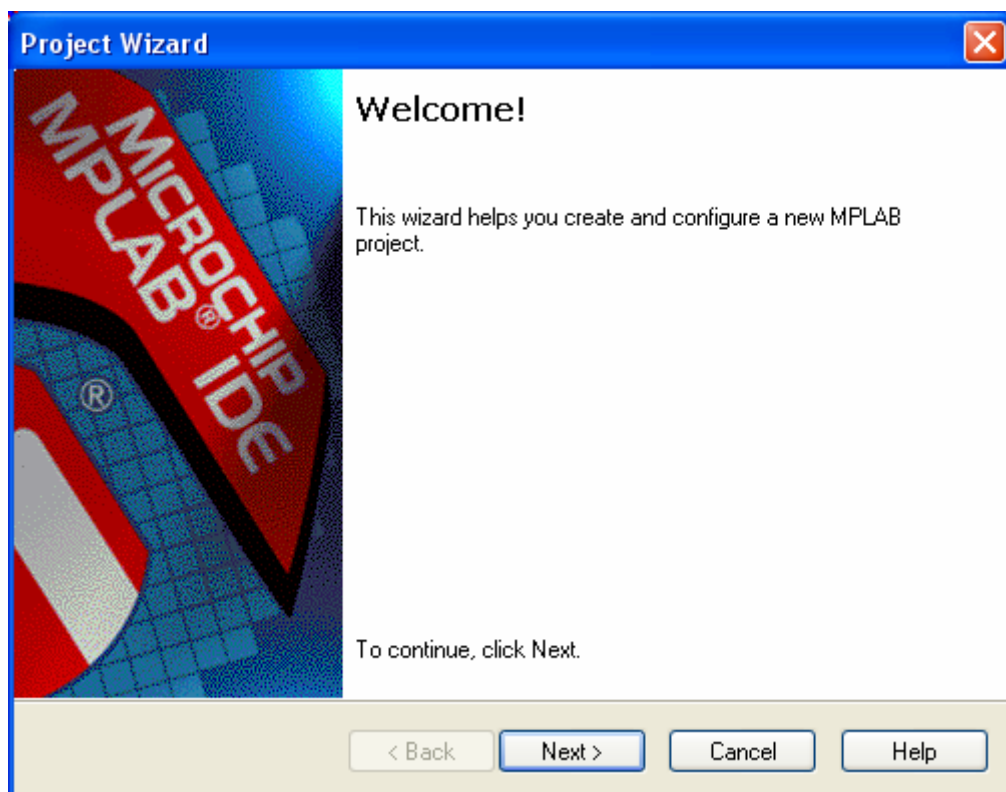
แสดง ตัวอย่าง Source Code สำหรับใช้ทดลองการทำงาน

สำหรับตัวอย่างนี้จะเป็นการสั่งให้ใช้พอร์ต RB0 ทำหน้าที่เป็น Output ขับ LED ให้ติดและดับสลับกันไปไม่รู้จักจบในลักษณะของไฟกระพริบ ซึ่งวิธีการทดสอบการทำงานของโปรแกรมนี้ โดยใช้กับบอร์ด ET-dsPIC30F2010 นั้นทำได้โดยต่อสัญญาณจาก RB0 เข้ากับ LED Output ของบอร์ด โดยจะเห็นผลการทำงานของโปรแกรมคือ LED จะกระพริบ ติดและดับอย่างต่อเนื่องตลอดเวลา

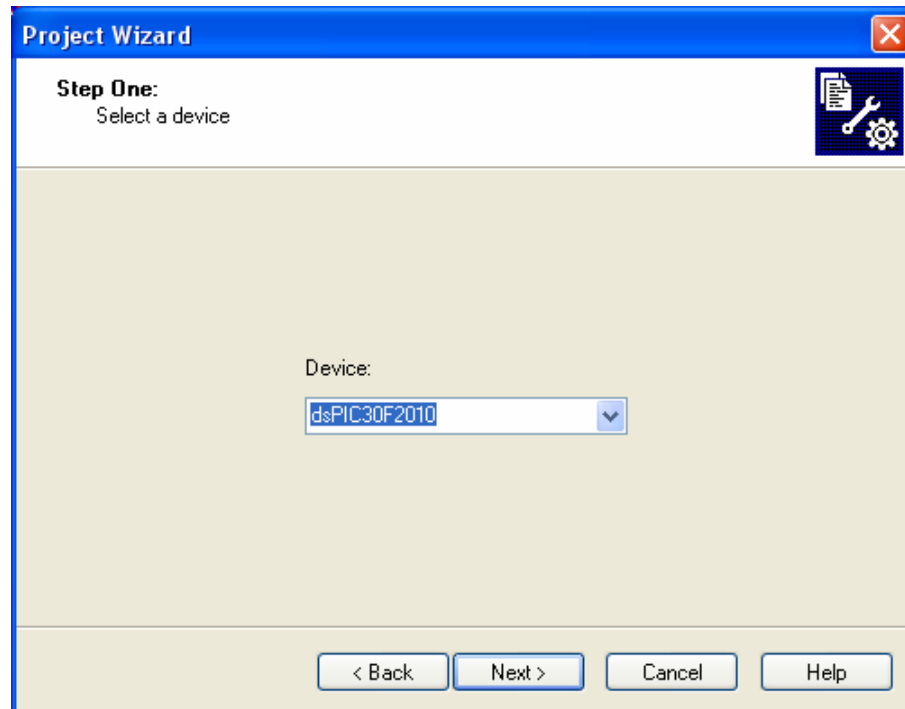
ส่วนที่น่าสนใจของโปรแกรมคือบรรทัดคำสั่ง “_FOSC(CSW_FSCM_ON & XT_PLL16);” ของโปรแกรมไปจนถึงบรรทัดคำสั่ง “_FGS(CODE_PROT_OFF);” โดยในส่วนนี้เป็นส่วนของการกำหนดค่า Configuration ของ MCU เบอร์ “dsPIC30F2010” เพื่อใช้กับบอร์ด “ET-dsPIC30F2010”

3. ทำการสร้าง Project File เพื่อใช้สั่งผนวกไฟล์ต่างๆ ที่เกี่ยวข้องเข้าด้วยกัน ทั้งนี้เนื่องมาจากว่า ภาษาซีของ MPLAB C30 นั้น ถูกออกแบบให้มีความอ่อนตัวในการทำงาน ดังนั้นจึงมีการจัดสร้าง และแบ่งแยกไฟล์ออกเป็นหลายๆไฟล์ตามหน้าที่การใช้งาน เพื่อให้ผู้ใช้สามารถเรียกไฟล์ต่างๆ เหล่านั้นเข้ามาใช้งานร่วมกับ Source Code ที่เขียนขึ้นมาได้ได้ง่าย โดยไม่ต้องเสียเวลาเขียน Source Code เองทั้งหมด ซึ่งจะทำให้ผู้ใช้สามารถลดเวลาในการเขียนโปรแกรมไปได้เป็นอย่างมากเนื่องจากเพียงแค่ทำการสั่งผนวกไฟล์ที่ทาง MPLAB C30 สร้างเตรียมไว้ให้ เข้ากับ Source Code ที่ผู้ใช้เขียนขึ้นใหม่แล้วสั่งแปลโปรแกรมก็จะได้ไฟล์ที่มีความสมบูรณ์ต่อการใช้งานแล้ว โดย จากตัวอย่าง Source Code ที่ได้ทดลองเขียนไปแล้วในข้างต้น ก็เช่นเดียวกัน จะเห็นได้ว่าการสั่ง ผนวกไฟล์ชื่อ “p30f2010.h” เข้ามาใช้งานด้วย ซึ่งทำให้ไม่ต้องเสียเวลาไปสั่งประกาศชื่อและ ตำแหน่งรีจิสเตอร์ต่างๆของ dsPIC30F2010 ให้เสียเวลา แต่สามารถอ้างถึงชื่อของรีจิสเตอร์ต่างๆ ในโปรแกรมได้ทันที โดยวิธีการกำหนดคุณสมบัติของ Project File มีดังนี้

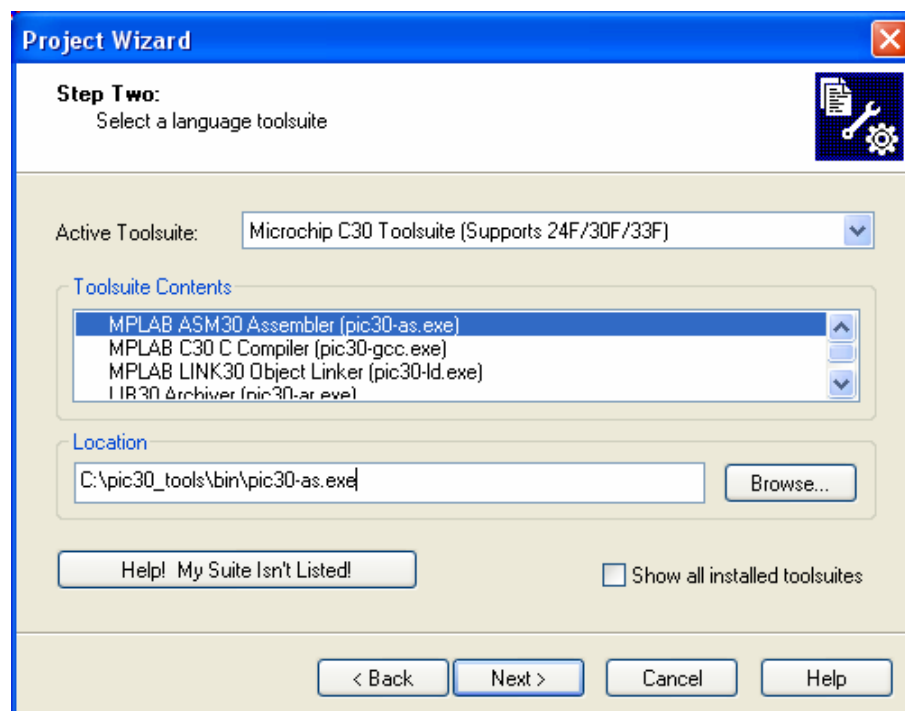
- สั่งกำหนดคุณสมบัติของ project File โดยใช้คำสั่ง “Project → Project Wizard...” ซึ่ง จะได้ผลดังรูป จากนั้นให้เลือก “Next >” เพื่อไปยังขั้นตอนต่อไป



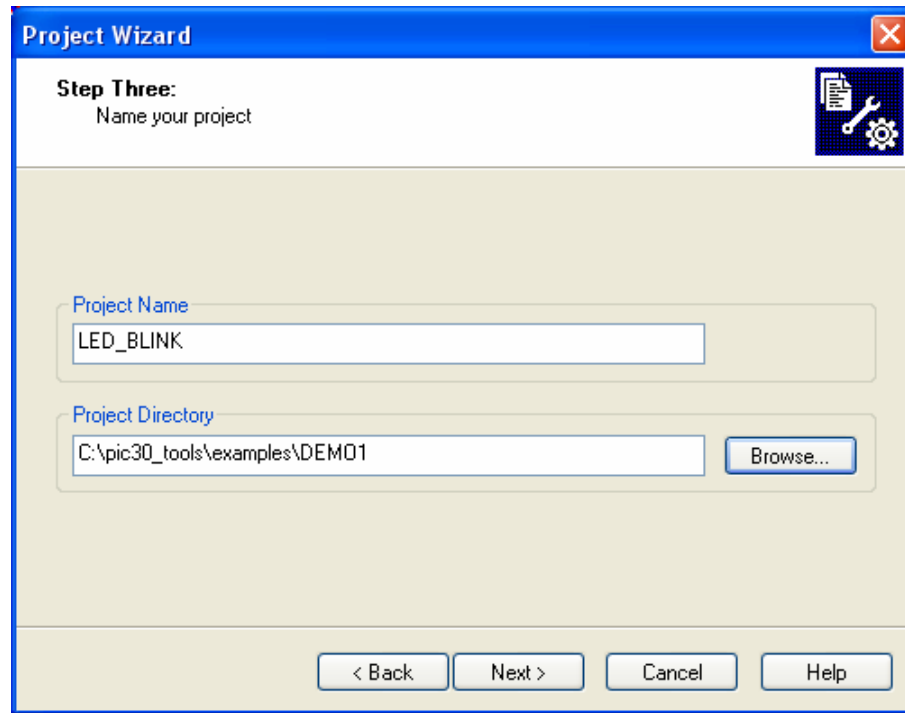
หลังจากเลือก “Next >” แล้ว โปรแกรมจะขอให้กำหนดเบอร์ของ MCU ที่จะใช้งานร่วมกับ โปรแกรมที่เขียนขึ้น ซึ่งให้เลือกกำหนดเป็น “dsPIC30F2010” จากนั้นเลือก “Next >” เพื่อข้ามไปทำงานยัง ขั้นตอนต่อไปดังรูป



ในขั้นตอนนี้จะเป็นการเลือกที่จะใช้โปรแกรมชุดใดในการแปลคำสั่ง เนื่องจาก MPLAB IDE สามารถใช้งานร่วมกับชุดโปรแกรมต่างๆมากมายหลายโปรแกรม ซึ่งในที่นี้ให้เลือกกำหนดใช้โปรแกรม ของ MPLAB C30 โดยการเลือกกำหนดตัวเลือกของ “Active Toolsuite” ให้เป็นของ MPLAB C30 โดยกำหนดตัวเลือกเป็น “Microchip C30 Toolsuite (Supports 24F/30F/33F)” ดังรูป แล้วเลือก “Next >”



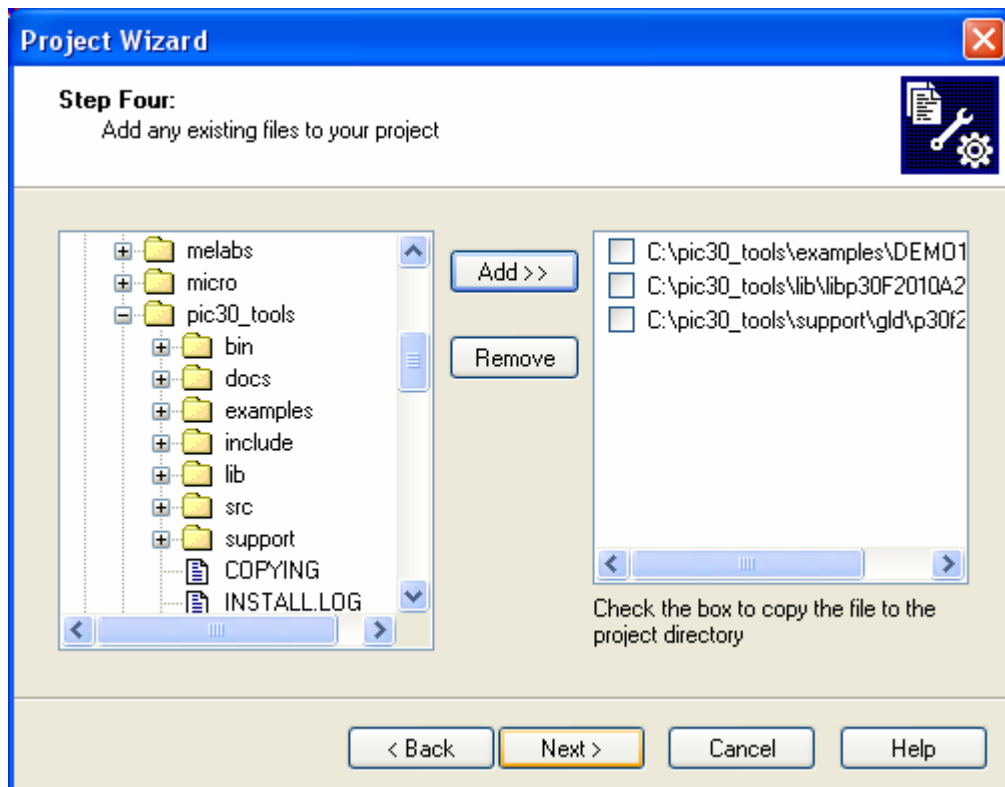
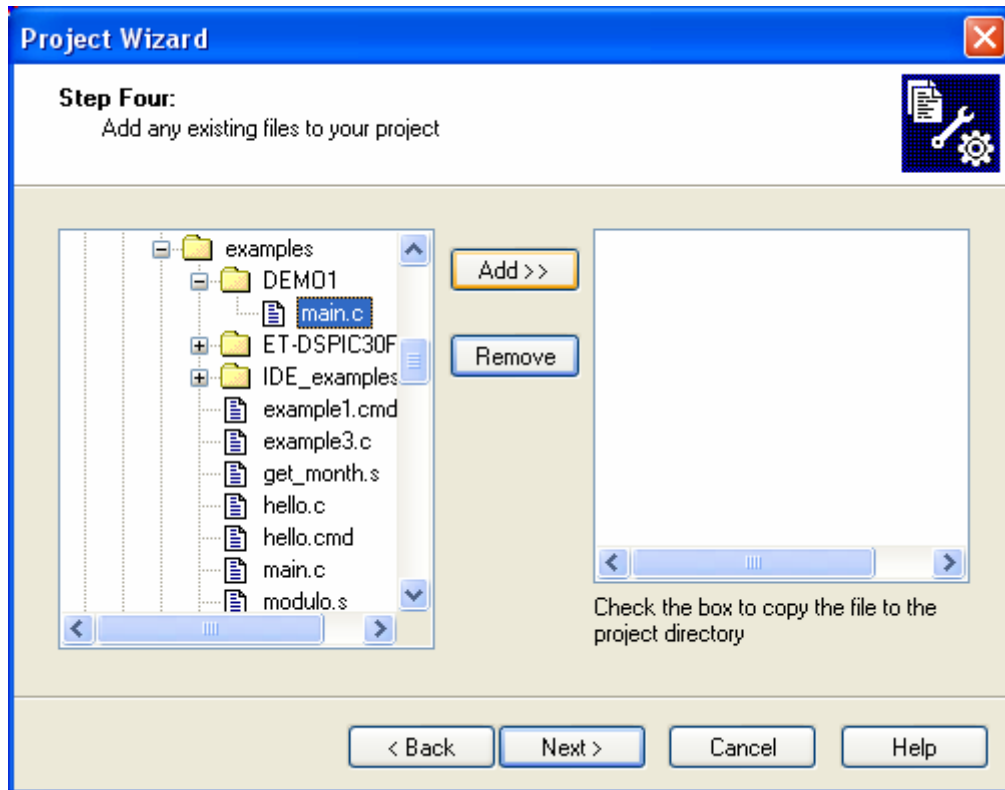
ในขั้นตอนนี้จะเป็นการ กำหนดชื่อ Project และตำแหน่ง Folder ที่จะใช้เก็บไฟล์ต่างๆที่ได้จากการทำงานของ Project โดยให้กำหนดชื่อเป็น “LED_BLINK” แล้วกำหนดตำแหน่ง Folder เป็น DEMO1 โดยกำหนดไว้ภายใต้ Folder ชื่อ examples ของ MPLAB C30 ดังรูป แล้วเลือก “Next >”



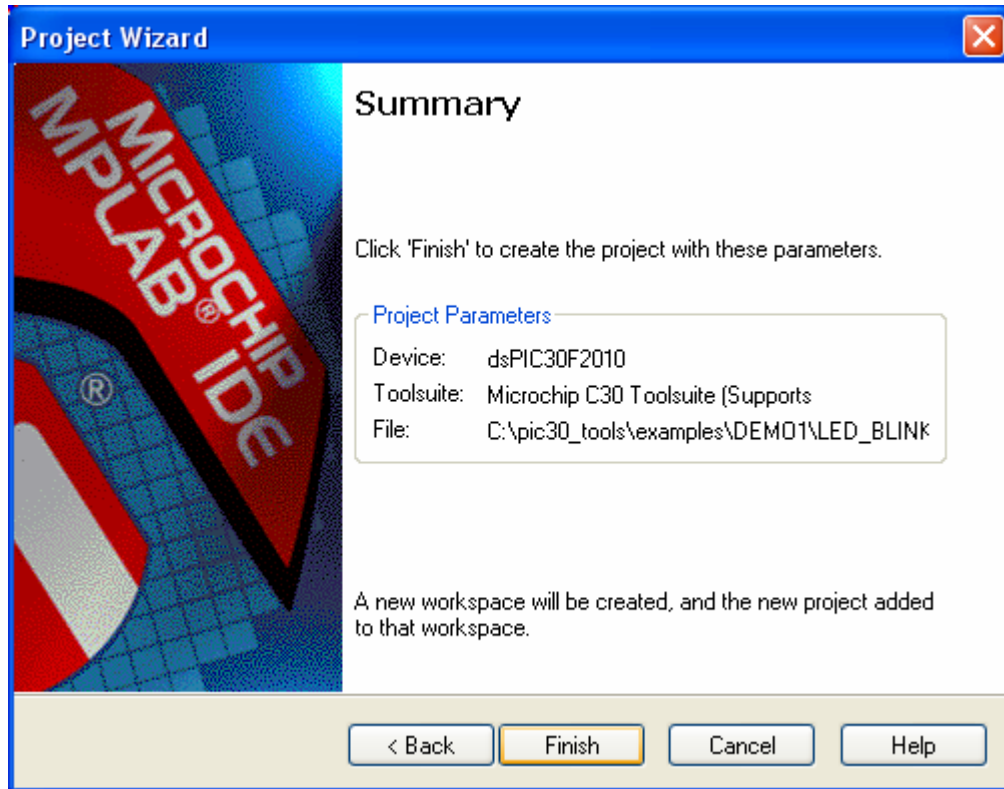
ในขั้นตอนนี้จะเป็นการสั่งผนวกไฟล์ต่างๆเข้าไว้ด้วยกันภายใต้ชื่อ Project ของ LED_BLINK โดยให้ทำการสั่งผนวกไฟล์ทั้งหมด 3 ไฟล์เข้าไว้ใน Project ดังนี้

- สั่งผนวกไฟล์ชื่อ “main.c” ซึ่งเป็น Source Code ที่เราได้เขียนและสั่งบันทึกไว้แล้วในขั้นตอนที่ผ่านมาโดยเก็บอยู่ใน c:\pic0_tools\examples\demo1\main.c”
- สั่งผนวกไฟล์ชื่อ “libp30f2010a2.a” ซึ่งเป็น Library ที่ใช้กับ MCU เบอร์ dsPIC30F2010 ซึ่งทาง MPLAB C30 ได้จัดสร้างเตรียมไว้ให้ใช้งาน โดยถ้าติดตั้งโปรแกรมตามตัวอย่างไฟล์ดังกล่าวจะเก็บอยู่ใน “c:\pic30_tools\lib\libp30f2010a2.a”
- สั่งผนวกไฟล์ชื่อ “p30f2010a2.gld” ซึ่งเป็น Script File ของ dsPIC30F2010 ที่ทาง MPLAB C30 สร้างเตรียมไว้ให้ โดยถ้าติดตั้งโปรแกรมตามตัวอย่างไฟล์ดังกล่าวจะเก็บอยู่ใน “c:\pic30_tools\support\gld\p30f2010a2.gld”

โดยในการสั่งผนวกไฟล์ทั้ง 3 ดังกล่าวให้ทำการคลิกเมาส์ไปยัง “ICON” ของไฟล์จากตำแหน่ง Folder ที่กล่าวไว้ในข้างต้นที่ผ่านมาแล้วเลือก “Add >>” จนชื่อไฟล์ดังกล่าวปรากฏอยู่ที่กรอบหน้าต่างด้านขวาของโปรแกรม ซึ่งให้เลือกทำตามวิธีการนี้จนสามารถสั่ง “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” ซึ่งจะทำให้ได้ไฟล์ Output มีชื่อเดียวกับ Project File ที่สร้างไว้แต่มีนามสกุลเป็น HEX ซึ่งจากตัวอย่าง Project นี้เมื่อสั่งแปลโปรแกรมแล้วถ้าไม่เกิดข้อผิดพลาดใดๆจะได้ Output ไฟล์ชื่อ “LED_BLINK.HEX” โดยไฟล์ดังกล่าวจะถูกสร้างและเก็บไว้ในตำแหน่ง Folder ของ Project คือ “c:\pic30_tols\examples\demo1\led_blink.hex” โดยผู้ใช้สามารถสั่ง Download Hex File ชื่อ “LED_BLINK.HEX” นี้ให้กับบอร์ด “ET-dsPIC30F2010” เพื่อทดสอบการทำงานได้ทันที

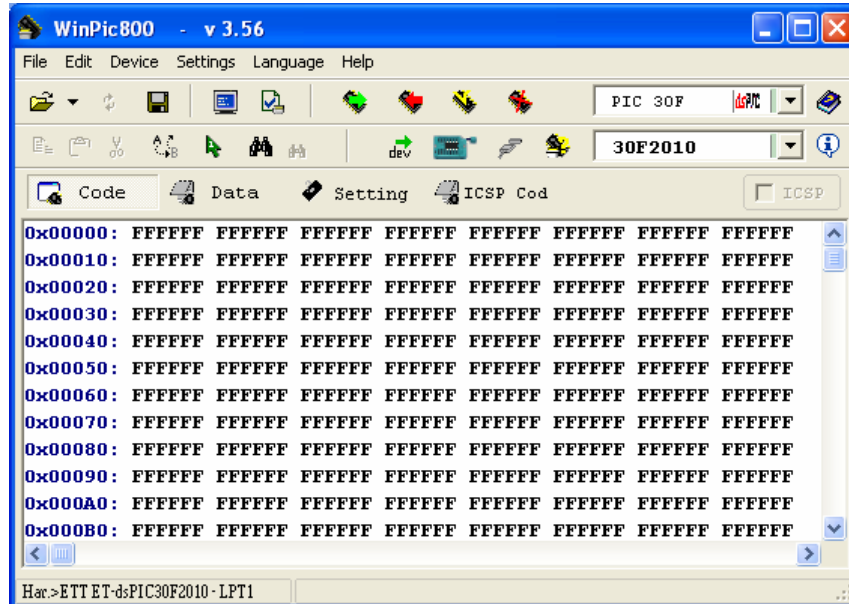
การ Download HEX File ให้บอร์ดด้วยโปรแกรม “WinPic800”

การ Download Hex File ให้กับบอร์ดนั้น ในปัจจุบัน (พฤษภาคม 2548) สามารถเลือกใช้โปรแกรม “WinPic800” หรือ “melabs Programmer” ก็ได้ตามต้องการ โดยไม่ว่าจะเลือกใช้โปรแกรมชุดใดเป็นโปรแกรมสั่ง Download ก็สามารถใช้งานได้เหมือนกันทุกประการ เพียงแต่วิธีการสั่งงานโปรแกรมอาจมีรายละเอียดและหน้าตาของโปรแกรมที่แตกต่างกันไปบ้าง แต่ไม่ว่าจะเลือกใช้โปรแกรมชุดใดในการสั่งงานก็ตามจำเป็นจะต้องใช้ชุด “ET-CAP10PIN V2.0” เป็นตัวกลางในการเชื่อมต่อสัญญาณจากพอร์ตขนาน (LPT) ของคอมพิวเตอร์ PC กับบอร์ด โดยในการ Download นั้นจะต้องใช้ Adapter จ่ายไฟให้กับบอร์ดด้วยค่าแรงดันไม่ต่ำกว่า 14V ไม่เช่นนั้นแล้วจะไม่สามารถสั่ง Download HEX File ให้กับ MCU ได้

สำหรับโปรแกรม “WinPic800” นั้น เป็นโปรแกรมประเภท “Free Ware” ใช้สำหรับ Download Hex File ให้กับ MCU ตระกูล PIC และ dsPIC พัฒนาขึ้นโดย “Sisco Benach Font” ซึ่งทางอีทีที ได้ร่วมสนับสนุนค่าใช้จ่ายให้กับทางผู้พัฒนาโปรแกรม เพื่อขอสิทธิ์ในการนำโปรแกรมดังกล่าวมาแจกจ่ายเพื่อใช้งานกับสินค้าของทางอีทีทีด้วยแล้ว ซึ่งผู้ใช้งานสามารถทำการ Download มาใช้งานได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆ โดยโปรแกรมตัวนี้ในปัจจุบันได้รับการปรับปรุงมาอย่างต่อเนื่องเป็นลำดับจนถึงรุ่นล่าสุดคือ “3.56” แล้ว ซึ่งในรุ่นนี้สามารถใช้งานได้กับ MCU ตระกูล PIC และ dsPIC ของ Microchips ได้มากมายหลายเบอร์รวมทั้งเบอร์ “dsPIC30F2010” ด้วย โดยขั้นตอนของการ Download HEX File ด้วยโปรแกรม “WinPic800” มีลำดับขั้นตอนเป็นดังนี้

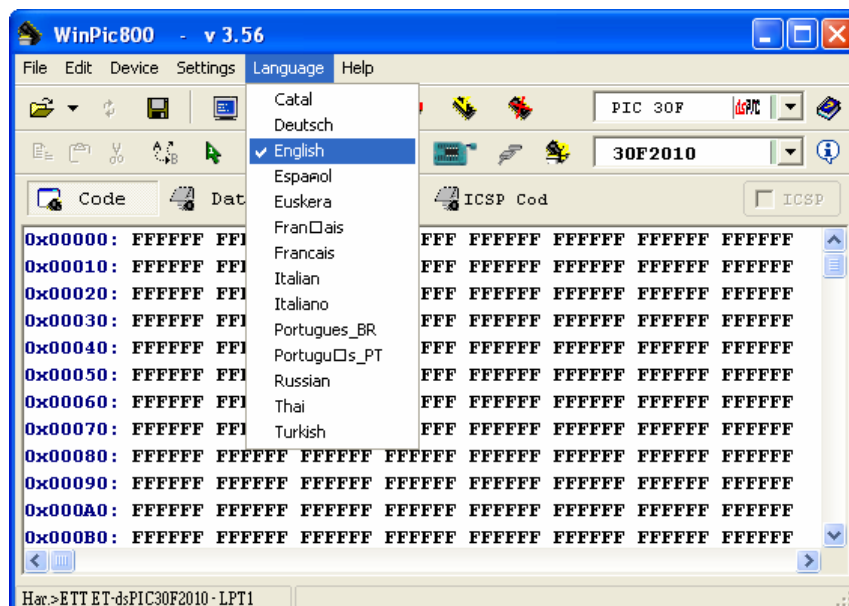
1. ติดตั้งโปรแกรม “WinPic800” ในเครื่องคอมพิวเตอร์ให้เรียบร้อยพร้อมใช้งาน
2. เขียนโปรแกรมและแปลคำสั่งของโปรแกรมที่เขียนให้เป็น “Hex File” ซึ่งวิธีการเขียนโปรแกรมและการสั่งแปลโปรแกรมจะไม่ขอล่าถึงในที่นี้ด้วย ทั้งนี้ขึ้นอยู่กับผู้ใช้งานว่าจะเลือกใช้ภาษาอะไรในการพัฒนาโปรแกรม และจะเลือกใช้โปรแกรม Assembler หรือ Compiler ตัวใดเป็นตัวแปลคำสั่งให้ แต่ท้ายที่สุดแล้วไม่ว่าจะเขียนโปรแกรมด้วยภาษาอะไร หรือ ใช้โปรแกรมใด เป็นตัวแปลคำสั่งก็จะได้ไฟล์ Output ที่จะนำมาใช้ในการสั่งงาน MCU เหมือนๆกันในรูปแบบของ “Hex File” ซึ่งในที่นี้จะขอล่าถึงเพียงวิธีการสั่ง Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ภายในบอร์ด “ET-dsPIC30F2010” เท่านั้น ส่วนวิธีการซึ่งจะทำให้ได้ Hex File มานั้นจะไม่ขอล่าถึง
3. ติดตั้งชุด “ET-CAP10PIN V2.0” เข้ากับพอร์ตขนาน (LPT Port หรือ Printer Port) ของเครื่องคอมพิวเตอร์ PC เพื่อใช้ในการสั่ง Download Hex File ด้วยโปรแกรม “WinPic800”
4. ต่อสายแพร์ขนาด 10 PIN จากขั้วต่อ “ET-EMPIC” ของบอร์ด “ET-CAP10PIN V2.0” เข้ากับขั้วต่อสำหรับ Download ของบอร์ด “ET-dsPIC30F2010” ที่ตำแหน่ง “DOWNLOAD(ET-EMPIC)”
5. เลือกสวิตช์สำหรับเลือกโหมดการทำงานของบอร์ด (RUN/PROG) ให้อยู่ในโหมด Program โดยให้เลือกสวิตช์ไว้ทางด้าน “PROG”

6. ทำการจ่ายไฟให้กับบอร์ด “ET-dsPIC30F2010” ซึ่งจะต้องใช้แหล่งจ่ายไฟซึ่งมีค่าแรงดันไม่น้อยกว่า 14V ด้วยถ้าแหล่งจ่ายไฟมีค่าแรงดันต่ำกว่านี้จะไม่สามารถสั่งโปรแกรมได้สำเร็จ
7. สั่ง Run Program ของ “WinPIC800” เพื่อเริ่มต้นขั้นตอนของการ Download HEX File

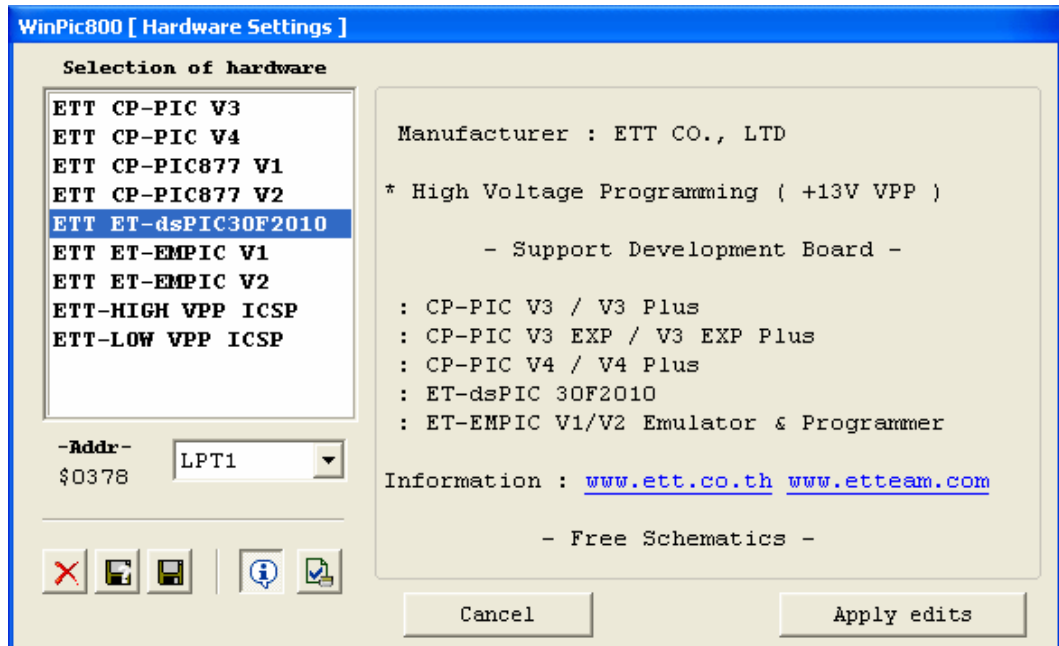


รูปแสดง ลักษณะของโปรแกรม “WinPic800”

8. ถ้ายังไม่ได้กำหนดค่าตัวเลือกให้กับโปรแกรม “WinPic800” เพื่อใช้สั่ง Download Hex File ให้กับ MCU เบอร์ “dsPIC30F2010” ร่วมกับระบบฮาร์ดแวร์ของ อีทีที มาก่อน ให้ทำการกำหนดค่าตัวเลือกของโปรแกรมสำหรับใช้งานร่วมกับ MCU ตระกูล dsPIC ให้เรียบร้อยดังนี้
 - กำหนดภาษาที่จะใช้ในการสั่งงานโปรแกรม โดยขอแนะนำให้เลือกเป็นภาษาอังกฤษ โดยให้คลิกเมาส์ที่เมนูคำสั่ง Language แล้วเลือก English ดังรูป



- ทำการกำหนดรูปแบบการเชื่อมต่อทางฮาร์ดแวร์ของบอร์ดให้ถูกต้อง โดยการคลิกเมาส์ที่เมนูคำสั่ง “Settings → Hardware” โดยในกรณีที่ใช้งานโปรแกรม “WinPIC800” ร่วมกับบอร์ด “ET-dsPIC30F2010” โดยใช้สาย Download ของ อีทีที รุ่น “ET-CAP10PIN V2.0” ให้กำหนดรูปแบบการเชื่อมต่อทางฮาร์ดแวร์เป็น “ETT ET-dsPIC30F2010” หรืออาจเลือกเป็น “ETT-HIGH VPP ICSP” ก็ได้ตามต้องการ ส่วน LPT นั้นให้กำหนดตามความเป็นจริงซึ่งปกติจะเป็น “LPT1” แล้วเลือก “Apply edits” ดังรูป



- ทำการกำหนดเงื่อนไขการทำงานของโปรแกรมว่าต้องการให้โปรแกรมทำงานตามลำดับขั้นตอนอย่างไรบ้าง โดยการคลิกเมาส์ที่เมนูคำสั่ง “Settings → Software” ดังรูป

ที่ Tab ของ Program ให้เลือก

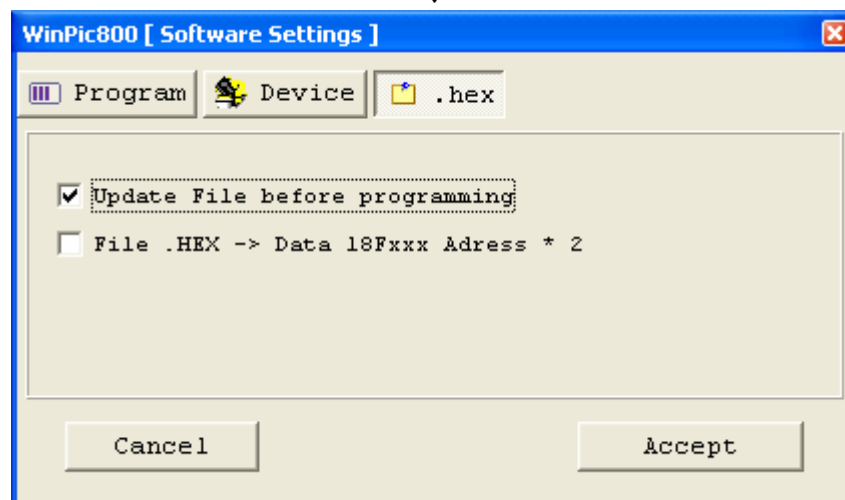
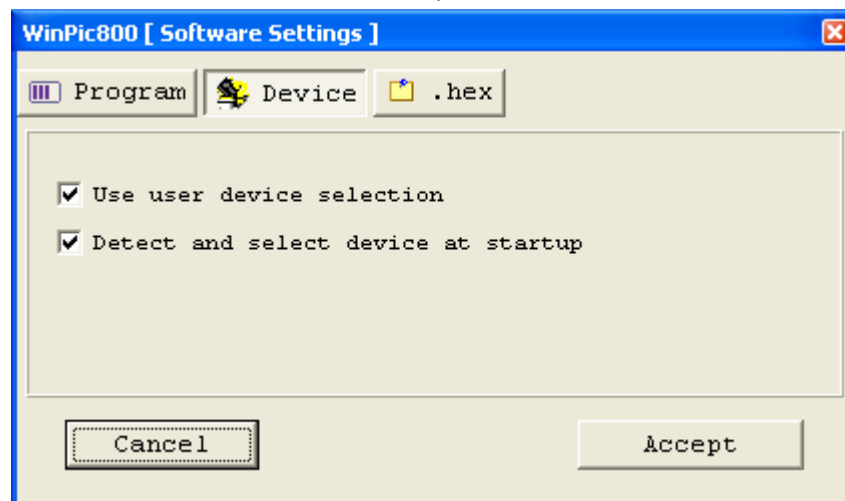
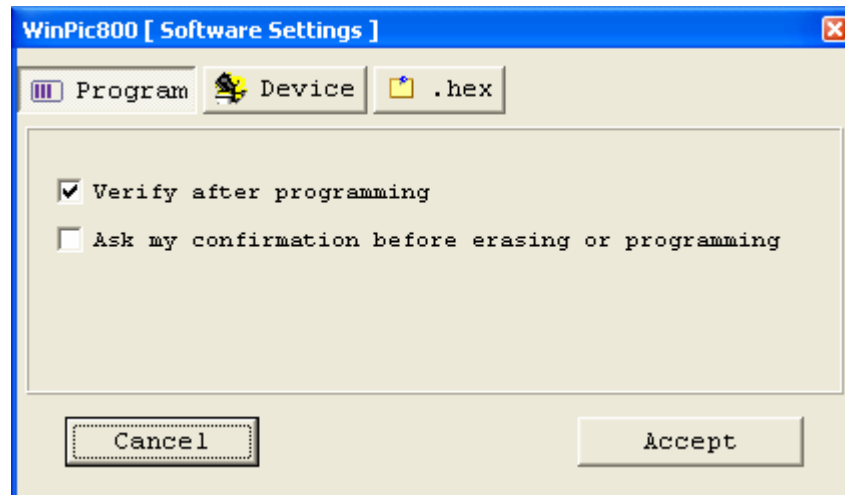
- ✓ Verify after programming

ที่ Tab ของ Device ให้เลือก

- ✓ Use user device selection
- ✓ Detect and select device at startup

ที่ Tab ของ Hex ให้เลือก

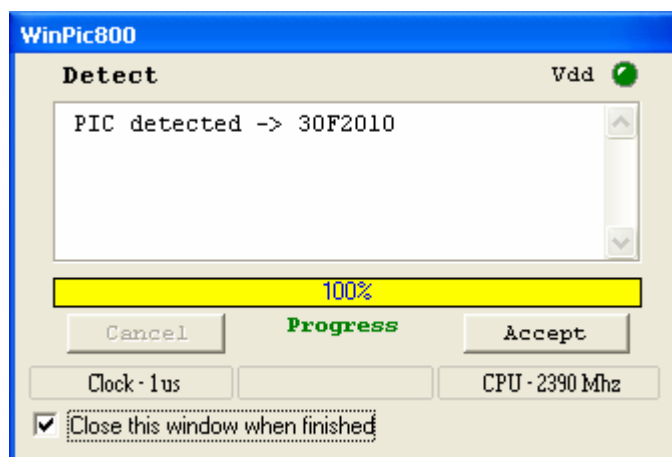
- ✓ Update File before programming



ซึ่งในขั้นตอนนี้อาจเลือกทุกขั้นตอนเลย แต่ถ้าเห็นว่าเลือกทุกขั้นตอนแล้วทำให้เสียเวลาในการ Download นานเกินไปก็อาจลดหัวข้อที่ไม่จำเป็น เช่น Verify after programming ออกก็ได้ จากนั้นให้เลือกที่ "Accept" เพื่อบันทึกค่าตัวเลือกละไว้ใช้งานกับโปรแกรมต่อไป

9. หลังจากทำการกำหนดค่าตัวเลือกต่างๆให้กับโปรแกรมเป็นที่เรียบร้อยแล้ว หลังจากนั้นเป็นต้นไปก็สามารถสั่งงานโปรแกรม “WinPic800” ได้ตามต้องการ โดยค่าตัวเลือกที่กำหนดไว้จะถูกบันทึกไว้เพื่อใช้เป็นเงื่อนไขในการทำงานของโปรแกรมตลอดไปจนกว่าจะมีการเปลี่ยนแปลงใหม่ ซึ่งนั่นก็หมายความว่าในครั้งต่อไปถ้าต้องการใช้งานโปรแกรมร่วมกับบอร์ด “ET-dsPIC30F2010” อีกก็ไม่ต้องจำเป็นต้องเข้าไปกำหนดค่าตัวเลือกในโปรแกรมใหม่ให้เสียเวลาอีก สามารถข้ามขั้นตอนของการกำหนดค่าตัวเลือก (ข้อ 8) ไปสั่งงานโปรแกรมได้ทันทีเลย โดยวิธีการสำหรับสั่งงานโปรแกรม “WinPic800” นั้นสามารถทำได้ 2 วิธี คือการสั่งงานผ่านทาง เมนูคำสั่ง หรือสั่งงานผ่าน “Button” คำสั่งต่างๆของโปรแกรมได้ตามต้องการ ดังตัวอย่าง

- สั่ง Open Hex File เพื่อใช้ในการ Download ให้กับ MCU โดยสามารถสั่งผ่านเมนูคำสั่งของ “File → Open” หรือคลิกเมาส์ที่ Button คำสั่ง  แล้วเลือกกำหนดชื่อและที่อยู่ของ Hex File ที่ต้องการจะใช้ Download ตามต้องการ
- สั่งตรวจสอบการเชื่อมต่อทางฮาร์ดแวร์กับ MCU โดยสามารถสั่งงานผ่านเมนูคำสั่งของ “Device Detect PIC” หรือคลิกเมาส์ที่ Button คำสั่ง  ซึ่งโปรแกรมจะทำการอ่านรหัส ID Code ของ MCU พร้อมกับแสดงชื่อเบอร์ของ MCU ที่ตรวจพบให้ทราบ โดยถ้าการทำงานถูกต้องในกรณีที่ใช้กับบอร์ด “ET-dsPIC30F2010” จะต้องได้ผลดังรูป

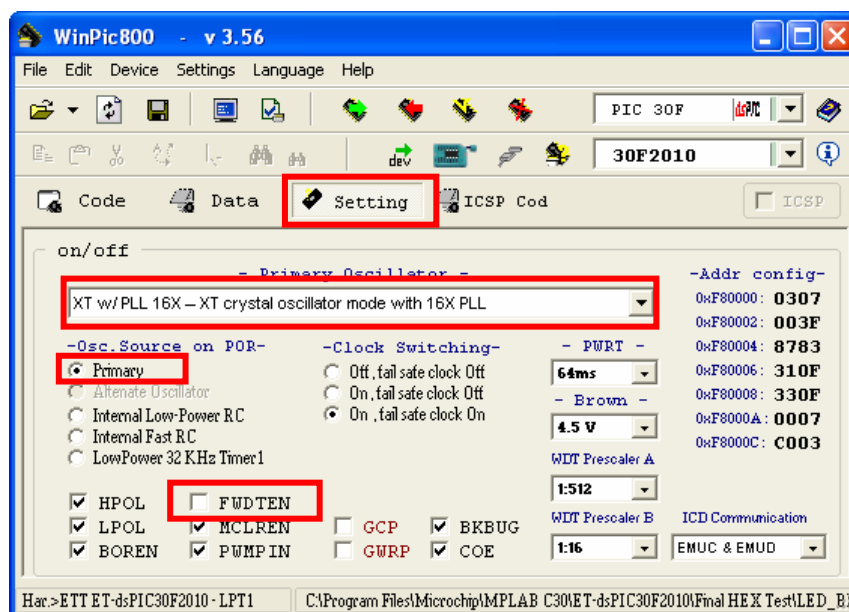


ถ้าการตรวจสอบเบอร์ MCU เกิดความผิดพลาด โดยถ้าโปรแกรมแสดงเบอร์ไม่ถูกต้อง ให้ลองตรวจสอบตำแหน่งสวิตช์เลือกการทำงาน “PROG/RUN” ว่าถูกเลือกไว้ทางด้าน “PROG” ถูกต้องแล้วหรือยัง สายสัญญาณต่างๆต่อ แน่น สนิท ดีหรือยัง ค่าแรงดันของแหล่งจ่ายไฟที่ใช้มีค่าถูกต้องหรือไม่ (14V-18V) รวมทั้งการกำหนดรูปแบบการเชื่อมต่อทางฮาร์ดแวร์ของโปรแกรมกำหนดไว้เป็น “ETT-HIGH VPP ICSP” หรือ “ETT ET-dsPIC30F2010” ถูกต้องหรือไม่ ถ้าทุกอย่างเรียบร้อยแล้วลองทำซ้ำดูอีกครั้งหนึ่งจนได้ค่าเบอร์ MCU ที่ถูกต้อง


- ตรวจสอบค่า Configuration ของ “dsPIC30F2010” ว่ากำหนดไว้ถูกต้องหรือไม่ โดยในกรณีที่ใช้งานกับบอร์ด “ET-dsPIC30F2010” นั้นต้องกำหนดค่า Configuration ของ MCU ในส่วนของการเลือกกำหนดค่าความถี่ของสัญญาณนาฬิกา และการทำงานของวงจร Watchdog ด้วยค่าตัวเลือกดังนี้

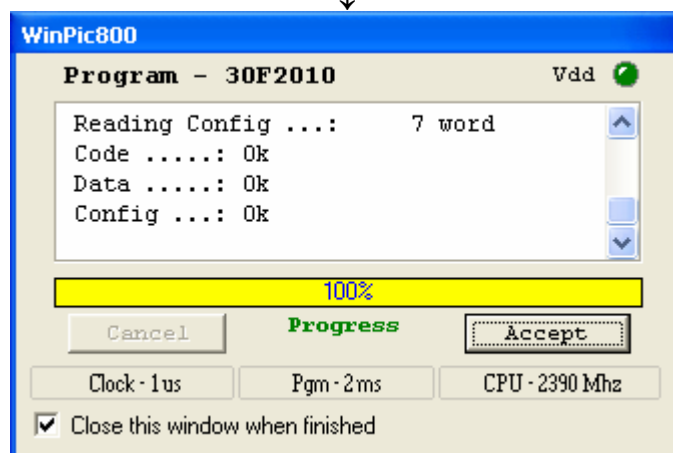
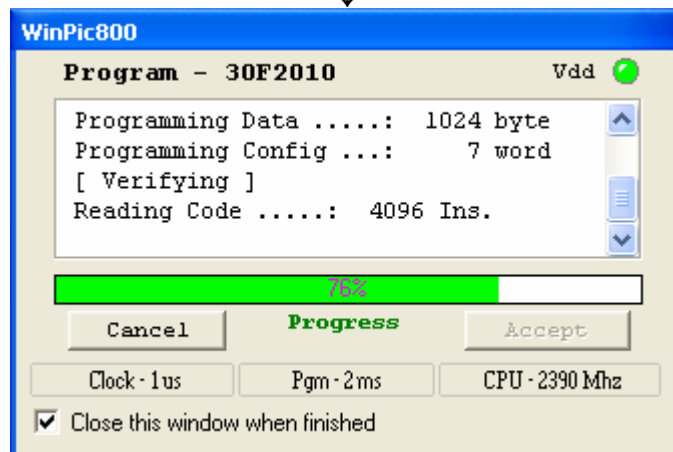
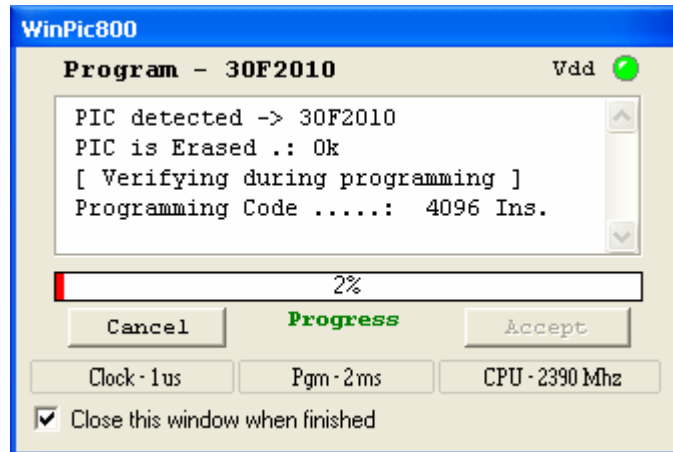
- ✓ OSC Source on POR ให้เลือกกำหนดเป็น Primary ซึ่งเป็นการเลือกกำหนดให้ MCU เลือกใช้สัญญาณนาฬิกาจากวงจร Primary Clock Source
- ✓ Primary Oscillator ให้เลือกเป็น XT w/PLL 16X - XT crystal oscillator mode with 16X PLL ซึ่งเป็นการเลือกสัญญาณนาฬิกาของ Primary Clock Source จากตัวกำเนิดความถี่ XTAL ค่า 4-10MHz ภายนอกซึ่งต่อไว้ที่ขา OSC1,OSC2 พร้อมกับเปิดการทำงานของวงจรคูณความถี่ไว้ที่ 16 เท่าของความถี่ XTAL ด้วย
- ✓ FWDTEN ต้องไม่เลือกเพื่อเป็นการสั่งปิดการทำงานของ Watchdog

ส่วนค่า Configuration อื่นๆ สามารถเลือกกำหนดได้เองตามต้องการ ซึ่งตามปกติแล้วค่าของ Configuration นี้จะถูกกำหนดค่าไว้ในขั้นตอนของการพัฒนาโปรแกรมอยู่แล้ว โดยเมื่อสั่งแปลโปรแกรมเป็น HEX นั้นค่า Configuration จะฝังอยู่ใน HEX File ด้วยเสมอ แต่อย่างไรก็ตามถ้าผู้ใช้ไม่ได้กำหนดค่าที่ถูกต้องให้กับ Configuration ไว้ด้วย ในขั้นตอนของการพัฒนาโปรแกรมก็อาจทำให้ค่า Configuration มีค่าไม่ถูกต้องตามต้องการ ดังนั้นจะต้องแก้ไขค่าตัวเลือกของ Configuration ให้ถูกต้องก่อนสั่ง Program โดยผู้ใช้งานสามารถตรวจสอบค่าของ Configuration ในปัจจุบันได้จากหน้าจอโปรแกรม โดยการคลิกเมาส์เลือกไปที่แท็บของ “Setting” ดังรูป



รูปแสดงการกำหนดค่า Configuration สำหรับบอร์ด “ET-dsPIC30F2010”

- สั่ง Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ด โดยสามารถสั่งผ่านเมนูคำสั่งของ “Device → Program All” หรือคลิกเมาส์ที่ Button คำสั่ง  ซึ่งหลังจากขั้นตอนนี้เสร็จสิ้นลง ถ้าไม่เกิดความผิดพลาดใดๆ เมื่อเลือกสวิตช์กำหนดโหมดการทำงานของบอร์ดไปยังด้าน “RUN” จะเห็นว่า MCU จะเริ่มต้นทำงานทันที



รูปแสดง ลำดับขั้นตอนการทำงานของ การสั่งงานด้วยคำสั่ง “Program All”

การ Download HEX File ให้บอร์ดด้วยโปรแกรม “melabs Programmer”

สำหรับโปรแกรม “melabs Programmer” ก็เป็นอีกโปรแกรมหนึ่งซึ่งสามารถนำมาใช้งานในการสั่ง Download Hex File ให้กับ MCU ของบอร์ด “ET-dsPIC30F2010” ได้ดีเช่นเดียวกับ “WinPic800” โดยโปรแกรม “melbas Programmer” เป็นของ “micro Engineering Labs, Inc.” ซึ่งเป็นผู้พัฒนาโปรแกรม PIC Basic Pro Compiler สำหรับ MCU ตระกูล PIC ที่หลายคนรู้จักกันเป็นอย่างดีนี้มาแล้ว โดยโปรแกรม “melbas Programmer” รุ่น Beta สามารถ Download มาใช้ได้ฟรีโดยไม่เสียค่าใช้จ่ายเช่นเดียวกัน สำหรับวิธีการใช้งานโปรแกรม “melbas Programmer” ร่วมกับบอร์ด “ET-dsPIC30F2010” มีดังต่อไปนี้

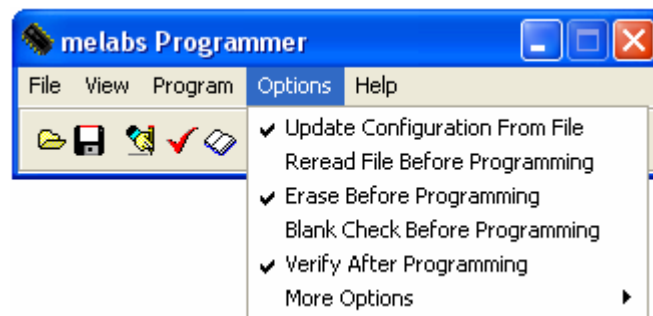
1. ติดตั้งโปรแกรม “melbas Programmer” ในเครื่องคอมพิวเตอร์ให้เรียบร้อยพร้อมใช้งาน
2. เขียนโปรแกรมและแปลคำสั่งของโปรแกรมที่เขียนให้เป็น “Hex File” ซึ่งวิธีการเขียนโปรแกรมและการสั่งแปลโปรแกรมจะไม่ขอกล่าวถึงในที่นี้ด้วย ทั้งนี้ขึ้นอยู่กับผู้ว่าจะเลือกใช้ภาษาอะไรในการพัฒนาโปรแกรม และจะเลือกใช้โปรแกรม Assembler หรือ Compiler ตัวใดเป็นตัวแปลคำสั่งให้ได้ไฟล์ Output ที่จะนำมาใช้ในการสั่งงาน MCU เหมือนๆกันในรูปแบบของ “Hex File” ซึ่งในที่นี้จะขอกล่าวถึงเพียงวิธีการสั่ง Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ภายในบอร์ด “ET-dsPIC30F2010” เท่านั้น ส่วนวิธีการซึ่งจะทำให้ได้ Hex File มานั้นจะไม่ขอกล่าวถึง
3. ติดตั้งชุด “ET-CAP10PIN V2.0” เข้ากับพอร์ตขนาน (LPT Port หรือ Printer Port) ของเครื่องคอมพิวเตอร์ PC เพื่อใช้ในการสั่ง Download Hex File ด้วยโปรแกรม “melbas Programmer”
4. ต่อสายแพรขนาด 10 PIN จากขั้วต่อ “ET-EMPIC” ของบอร์ด “ET-CAP10PIN V2.0” เข้ากับขั้วต่อสำหรับ Download ของบอร์ด “ET-dsPIC30F2010” ที่ตำแหน่ง “DOWNLOAD(ET-EMPIC)”
5. เลือกสวิตช์สำหรับเลือกโหมดการทำงานของบอร์ด (RUN/PROG) ให้อยู่ในโหมด Program โดยให้เลือกสวิตช์ไว้ทางด้าน “PROG”
6. ทำการจ่ายไฟให้กับบอร์ด “ET-dsPIC30F2010” ซึ่งจะต้องใช้แหล่งจ่ายไฟซึ่งมีค่าแรงดันไม่น้อยกว่า 14V ด้วยถ้าแหล่งจ่ายไฟมีค่าแรงดันต่ำกว่านี้จะไม่สามารถสั่งโปรแกรมได้สำเร็จ
7. สั่ง Run Program ของ “melabs Programmer” เพื่อเริ่มต้นขั้นตอนของการ Download HEX File ซึ่งจะได้ผลดังรูป



รูปแสดง ลักษณะของโปรแกรม “melabs Programmer”

8. ถ้ายังไม่ได้กำหนดค่าตัวเลือกให้กับโปรแกรม “melabs Programmer” เพื่อใช้สั่ง Download Hex File ให้กับ MCU เบอร์ “dsPIC30F2010” ร่วมกับระบบฮาร์ดแวร์ของ อีทีที มาก่อน ให้ทำการกำหนดค่าตัวเลือกของโปรแกรมสำหรับใช้งานร่วมกับ MCU ตระกูล dsPIC ให้เรียบร้อยดังนี้

- กำหนดการทำงานของคำสั่ง โดยให้เลือกเมาส์ที่เมนูคำสั่ง “Options” แล้วเลือกคลิกเมาส์หน้าตัวเลือกคำสั่งต่างๆจนปรากฏเครื่องหมาย (✓) หน้าคำสั่งดังนี้
 - ✓ Update Configuration From File เป็นการกำหนดให้โปรแกรมทำการเปลี่ยนแปลงค่าตัวเลือกของ Configuration ตามค่าที่กำหนดได้จาก Hex File ที่สั่ง Open มาใช้งานในโปรแกรม
 - ✓ Erase Before Programming เป็นการสั่งให้โปรแกรมทำการลบข้อมูลจากหน่วยความจำของ MCU ก่อนสั่งโปรแกรมโดยอัตโนมัติ ซึ่งจะทำให้เราไม่ต้องเสียเวลาไปลบข้อมูลด้วยคำสั่ง Erase เองให้เสียเวลา
 - ✓ Verify After Programming เป็นการสั่งให้โปรแกรมทำการสั่ง Verify หรือตรวจสอบความถูกต้องของข้อมูลด้วยหลังจากโปรแกรมเสร็จแล้ว



- ทำการสั่งกำหนดค่าตัวเลือกพิเศษเพิ่มเติมให้กับโปรแกรมอีก โดยให้ทำการคลิกเมาส์ที่คำสั่ง “Options → More Options” แล้วเลือกกำหนดค่าต่างๆดังนี้
 - ✓ ให้คลิกเมาส์หน้าตัวเลือกของคำสั่ง “Verify Device Target ID” จนปรากฏเครื่องหมาย (✓) หน้าคำสั่ง เพื่อเป็นการสั่งให้โปรแกรมตรวจสอบความถูกต้องของเบอร์ MCU ว่าตรงตามที่เราเลือกไว้หรือไม่
 - ✓ ให้เลือกการทำงานของคำสั่ง Program โดยให้เลือกกำหนดขั้นตอนการทำงานของคำสั่งดังนี้ (Options → More Options → Program)
 - Code
 - Data
 - Configuration



- ✓ ให้เลือกการทำงานของคำสั่ง Verify โดยให้เลือกกำหนดขั้นตอนการทำงานของคำสั่งดังนี้ (Options → More Options → Verify)
 - Code
 - Data
 - Configuration

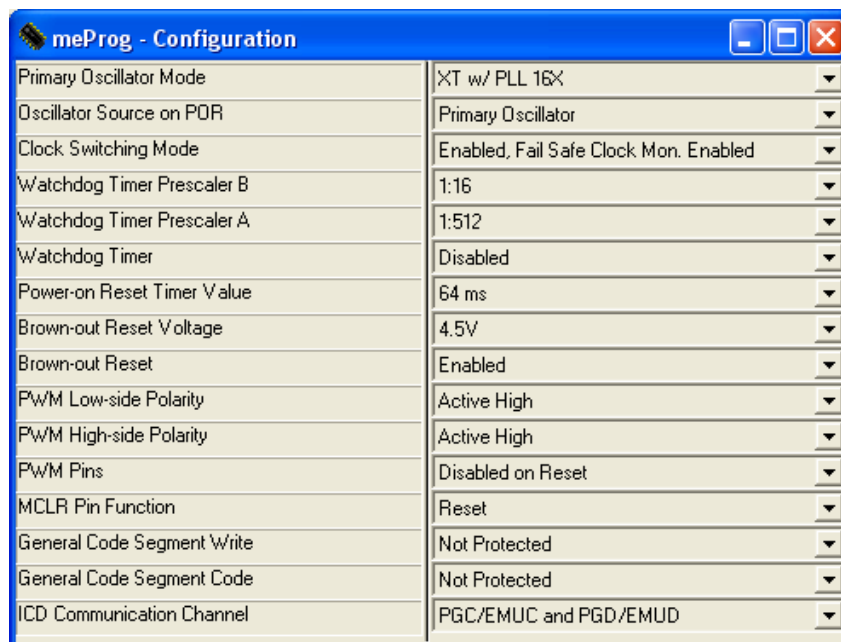
- ✓ ให้เลือกการทำงานของคำสั่ง Read โดยให้เลือกกำหนดขั้นตอนการทำงานของคำสั่งดังนี้ (Options → More Options → Read)
 - Code
 - Calibration
 - Data
 - Used ID
 - Configuration
 - Device ID

- ✓ ให้เลือกการทำงานของคำสั่ง Blank Check โดยให้เลือกกำหนดขั้นตอนการทำงานของคำสั่งดังนี้ (Options → More Options → Blank Check)
 - Code
 - Data
 - Configuration

- ✓ ให้เลือกการทำงานของคำสั่ง Erase โดยให้เลือกกำหนดขั้นตอนการทำงานของคำสั่งดังนี้ (Options → More Options → Erase)
 - Code
 - Data
 - Configuration / User ID


ซึ่งหลังจากกำหนดค่าตัวเลือกต่างๆ เหล่านี้เป็นที่เรียบร้อยแล้ว ค่าที่กำหนดไว้แล้วนี้ จะคงอยู่ตลอดไปจนกว่า จะมีการสั่งแก้ไขค่าตัวเลือกเหล่านี้ใหม่ ดังนั้นในครั้งต่อไปก็สามารถเรียกใช้งานโปรแกรมผ่านคำสั่งต่างๆ ที่ต้องการได้ทันที โดยไม่จำเป็นต้องมาทำการกำหนดค่าเหล่านี้อีกต่อไป

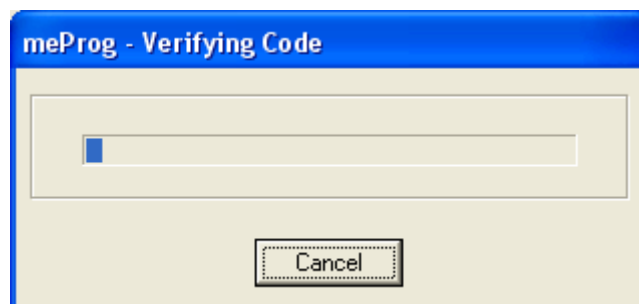
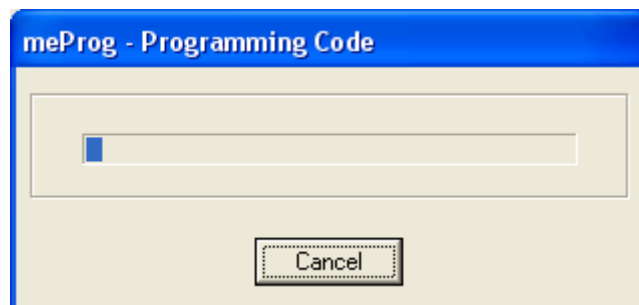
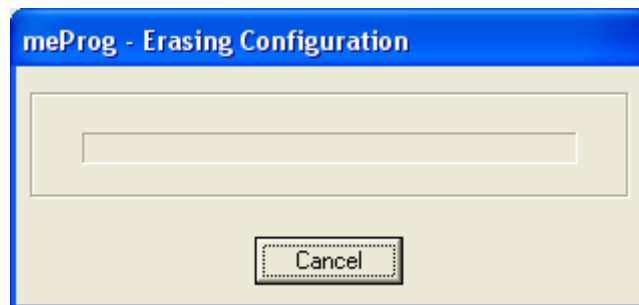
9. ในการสั่งงานโปรแกรมนั้นสามารถทำได้ 3 แบบ คือการสั่งงานผ่านเมนูคำสั่ง หรือ การสั่งงานผ่าน Button ของคำสั่ง หรือการสั่งงานผ่านคีย์ลัด (Hot Key) ของแต่ละคำสั่ง โดยไม่ว่าจะเลือกใช้วิธีการสั่งงานโปรแกรมด้วยวิธีการใดก็จะให้ผลการทำงานที่เหมือนกัน โดยในการสั่งงานโปรแกรมนี้นี้
- สั่งเปิด Hex File ที่จะต้องจะสั่งโปรแกรมไปยังหน่วยความจำของ MCU โดยใช้เมนูคำสั่ง “File → Open...” หรือกดคีย์ “CTRL+O” หรือคลิกเมาส์ที่ Button คำสั่ง  แล้วเลือกกำหนดชื่อและที่อยู่ของ Hex File ตามต้องการ
 - สั่งตรวจสอบค่า Configuration ของ MCU โดยใช้คำสั่ง “View Configuration” หรือกดคีย์ “CTRL+F” หรือคลิกเมาส์ที่ Button คำสั่ง  ซึ่งจะปรากฏหน้าต่างโปรแกรมสำหรับแสดงค่า Configuration ของ MCU ที่ได้จาก Hex File ที่สั่ง Open ไว้แล้วดังตัวอย่าง



รูปแสดงค่า Configuration ของ dsPIC30F2010 ที่จะใช้กับบอร์ด “ET-dsPIC30F2010”

ซึ่งตามปกติแล้วค่าของ Configuration เหล่านี้จะถูกกำหนดไว้ในขั้นตอนของการพัฒนาโปรแกรมอยู่แล้ว โดยผู้ใช้งานต้องรู้และเข้าใจเองว่าโปรแกรมที่พัฒนาขึ้นมามีการทำงานอย่างไร จำเป็นต้องใช้ทรัพยากรใดบ้าง และการทำงานของโปรแกรมเกี่ยวข้องกับค่าของ Configuration ใดบ้าง เพื่อจะได้กำหนดค่าการทำงานของ Configuration ให้กับโปรแกรมได้อย่างถูกต้องและสอดคล้องกับโปรแกรม ซึ่งถ้าใช้ตัวแปลคำสั่งของ Microchips นั้นค่าของ Configuration จะถูกรวมไว้ใน Hex File ในขั้นตอนของการสั่งแปลโปรแกรมด้วยอยู่แล้ว โดยค่าของ Configuration นี้จะเปลี่ยนแปลงไปตามข้อมูลที่ได้จากการสั่ง Open Hex File ขึ้นมาในโปรแกรมโดยอัตโนมัติ แต่อย่างไรก็ตามถ้าหากว่าในขั้นตอนของการพัฒนาโปรแกรมนั้น ผู้ใช้ไม่ได้กำหนดค่าของ Configuration ที่ถูกต้องไว้ด้วย ก็อาจส่งผลทำให้ MCU ไม่สามารถทำงานได้อย่างถูกต้องตามที่ต้องการ โดยในกรณีของบอร์ด “ET-dsPIC30F2010” นั้นต้องกำหนดค่าตัวเลือกดังรูป

- ส่งโปรแกรมข้อมูลจาก Buffer (จาก Hex File) ให้กับหน่วยความจำของ MCU โดยส่งผ่านเมนูคำสั่ง “Program → Program” หรือกดคีย์ “CTRL+P” หรือคลิกเมาส์ที่คำสั่ง  โดยการทำงานของคำสั่งนี้จะเป็นการส่ง Download ข้อมูลจาก Hex File ที่สั่งเปิดไว้แล้วไปยังหน่วยความจำของ MCU ซึ่งหลังจากเสร็จเรียบร้อยแล้วเมื่อเลือกสวิตช์กำหนดโหมดการทำงานของบอร์ดกลับไปยังตำแหน่ง “RUN” จะเห็นว่า MCU เริ่มต้นทำงานตามคำสั่งในโปรแกรมที่สั่ง Download ให้ในทันที โดยการทำงานของคำสั่งนี้จะมีลำดับขั้นตอนและกระบวนการทำงานของคำสั่งตามที่กำหนดไว้แล้วในข้างต้นอันได้แก่
 - ✓ Erase (Code ,Data ,Configuration/Used ID)
 - ✓ Program(Code, Data ,Configuration)
 - ✓ Verify(Code ,Data ,Configuration)



รูปแสดง ลำดับขั้นตอนการทำงานของการสั่งงานด้วยคำสั่ง “Program”

การตรวจสอบปัญหา MCU ไม่ทำงานหลังการ Download แล้ว

ตามปกติแล้วหลังจากที่ทำการ Download Hex File ให้กับ MCU เป็นที่เรียบร้อยแล้ว หลังจากที่เราเปลี่ยนโหมดการทำงานของบอร์ดจาก “PROG” มาเป็น “RUN” เป็นที่เรียบร้อยแล้ว MCU จะเริ่มต้นทำงานตามคำสั่งในโปรแกรมที่ทำการ Download ให้ทันที ซึ่งถ้าเกิดปัญหาหรือข้อผิดพลาดขึ้นในระหว่างขั้นตอนของการ Download โปรแกรมนั้น โปรแกรมก็จะแจ้งข้อผิดพลาดนั้นๆให้ทราบอยู่แล้ว แต่สำหรับในกรณีที่ขั้นตอนของการ Download ไม่เกิดอาการผิดพลาดใดๆเลย แต่ MCU ไม่ทำงานตามที่ต้องการ หลังจากทำการ Download เสร็จสิ้นไปเรียบร้อยแล้ว ปัญหาที่มักพบอยู่เสมอๆได้แก่

- การกำหนดค่า Configuration ของ MCU ก่อนการสั่ง Download ไม่ถูกต้องและสัมพันธ์กับฮาร์ดแวร์ที่ใช้งานอยู่ ซึ่งปัญหานี้เป็นปัญหาที่พบบ่อยที่สุด ทั้งนี้เนื่องมาจากว่าการทำงานของ MCU ตระกูล dsPIC นั้นจะใช้ Configuration เป็นตัวกำหนดเงื่อนไขการทำงานของวงจรภายในต่างๆ ไม่ว่าจะเป็นการเลือกการทำงานของสัญญาณนาฬิกา การกำหนดการทำงานของวงจร Watchdog และวงจร Brown-Out Detect ต่างๆ ซึ่งรายละเอียดของค่า Configuration ต่างๆโดยละเอียดนั้น ผู้ใช้สามารถศึกษารายละเอียดได้จากหัวข้อการกำหนดค่า Configuration ของ dsPIC30F2010 แต่ในที่นี้จะขอลำถึงเฉพาะในส่วนที่ต้องกำหนดให้สอดคล้องกับระบบฮาร์ดแวร์ของ “ET-dsPIC30F2010” เท่านั้น ซึ่งได้แก่

- ✓ สัญญาณนาฬิกาของบอร์ด “ET-dsPIC30F2010” จะใช้วงจรกำเนิดความถี่แบบ XTAL ค่า 7.3728 MHz โดยต่อกับขาสัญญาณ OSC1 และ OSC2 โดยที่ MCU เบอร์ dsPIC30F2010 สามารถรับความถี่จากตัวกำเนิดความถี่แบบ XTAL ได้สูงสุด 10MHz และทำงานได้ที่ความถี่สูงสุด 120 MHz ดังนั้นเพื่อให้สามารถใช้งาน MCU ได้อย่างเต็มประสิทธิภาพ จำเป็นต้องกำหนดให้วงจรคูณความถี่ (PLL) ทำงานด้วย โดยค่าอัตราการคูณความถี่ ที่เหมาะสมกับบอร์ด “ET-dsPIC30F2010” มากที่สุดคือ 16 เท่า ซึ่งจะทำให้ MCU ทำงานที่ความถี่ 117.9648 MHz โดยโปรแกรมตัวอย่างต่างๆที่จัดทำขึ้นโดย อีทีที นั้นจะอ้างอิงค่าความถี่ของสัญญาณนาฬิกาว่า 117.9648 MHz เป็นหลักด้วยเช่นเดียวกัน โดยผู้ใช้ต้องกำหนดค่า Configuration สำหรับเลือกสัญญาณนาฬิกาให้กับ MCU ด้วยค่าตัวเลือกดังนี้

- กำหนด Oscillator Source เป็น Primary Oscillator
- กำหนด Primary Oscillator Mode เป็น “XT w/PLL1 6X”

ซึ่งถ้าผู้ใช้เลือกกำหนดค่า Configuration เป็นอย่างอื่น จะทำให้ MCU ไม่ทำงานหรือทำงานได้ไม่ถูกต้องตามที่กำหนดไว้

- ✓ การทำงานของ Watchdog ควรกำหนดเป็น “Disable” ไว้ก่อน เพื่อเป็นการสั่งปิดการทำงานของ Watchdog ไม่ให้ทำงาน เนื่องจากถ้ากำหนดให้วงจร Watchdog ทำงานอยู่แต่โปรแกรมที่ผู้ใช้เขียนขึ้นไม่รองรับการใช้งานร่วมกับ Watchdog ด้วย อาจเกิดความผิดพลาดขึ้น โดย MCU จะถูก Watchdog สั่ง Reset อยู่เรื่อยๆทำให้การทำงานของโปรแกรมไม่ต่อเนื่อง ซึ่งดูเหมือน MCU ทำงานผิดพลาด โดยโปรแกรมตัวอย่างของทาง อีทีที ก็จะได้จัดการเรื่อง Watchdog ด้วยเช่นกัน ดังนั้นถ้าต้องการใช้งานตัวอย่างโปรแกรมของ อีทีที ด้วยจะต้องสั่งปิดการทำงานของ Watchdog ด้วยเสมอ

แต่อย่างไรก็ตาม ตัวอย่างโปรแกรมทั้งหมดที่เขียนขึ้นโดย อีทีที นั้นได้ทำการกำหนดค่าการทำงานของ Configuration ต่างๆที่เหมาะสมไว้ใน Source Code ของโปรแกรมด้วยอยู่แล้ว ผู้ใช้สามารถสั่ง Download Hex File ของตัวอย่างโปรแกรมจาก อีทีที ตามค่า Configuration ที่ได้จาก Hex File ได้ทันทีโดยไม่ต้องดัดแปลงหรือแก้ไขค่า Configuration ใดๆให้เสียเวลาอีก

- สวิตช์เลือกโหมดถูกเลื่อนกลับมายังตำแหน่ง “RUN” ไม่หมดทำให้ขาสัญญาณ MCLR ไม่เชื่อมต่อกับวงจร RESET อย่างถูกต้องจึงทำให้ MCU ไม่ทำงาน
- โปรแกรมที่ผู้ใช้เขียนขึ้นนั้นไม่สามารถทำงานได้อย่างถูกต้องเองอยู่แล้ว ซึ่งปัญหานี้อาจทดสอบได้ โดยการทดลองสั่ง Download โปรแกรมตัวอย่างที่ส่งมาเป็น Hex File ไว้เรียบร้อยแล้ว ซึ่งโปรแกรม ตัวอย่าง ต่างๆเหล่านี้ ได้รับการทดสอบ แล้วว่ามีความถูกต้องและทำงานได้จริงตามเงื่อนไขที่ทาง อีทีที กำหนดไว้
- ในกรณีที่การสั่ง Run โปรแกรม “melabs Programmer” แล้วเกิดการผิดพลาดเรื่องการโหลด Driver แสดงว่าในขั้นตอนของการติดตั้งโปรแกรมนั้นผู้ใช้อย่างยังไม่ได้สั่งติดตั้ง Driver ไว้ด้วย ให้ทำการสั่งติดตั้ง Driver ให้เรียบร้อยโดยสั่งงานผ่านเมนูคำสั่งของ Windows โดยใช้ “Start → Program melabs → Programmer Beta → NT,2000,XP Driver Installer” จากนั้นก็จะสามารถเรียกใช้งานโปรแกรม “melabs Programmer” ได้ตามปกติ

การกำหนดค่า Configuration ให้กับ dsPIC30F2010

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

โดยสำหรับบอร์ด Et-dsPIC30F2010 START KIT V1 / EXP นั้น ในกรณีที่ทำการพัฒนาโปรแกรมร่วมกับชุดพัฒนา “ET-CAP10 V2.0” ร่วมกับโปรแกรม “WinPic800” หรือ “melabs Programmer” นั้น จะสามารถกำหนดค่าตัวเลือกต่างๆให้กับ Configuration Register ผ่านทางหน้าจอของโปรแกรมโดยตรง หรือกำหนดค่าจากค่า Configuration ที่ได้จาก HEX File ของ MPLAB C30 ก็ได้ ดังตัวอย่าง

```
/* Setup Configuration For ET-dsPIC30F2010 */
_FOSC(CSW_FSCM_ON &                // Enable Clock Switching
      XT_PLL16);                    // Enable Fail-Safe Clock
                                    // Clock Source = Primary XT
                                    // Enable Phase Lock Loop x 16
_FWDT(WDT_OFF);                    // Disable Watchdog
_FBORPOR(PBOR_ON &                  // Enable Brown-Out
          BORV_45 &                  // Brown-Out = 4.5Volt
          PWRT_64 &                  // Power ON = 64 mS
          MCLR_EN);                 // Enable MCLR
_FGS(CODE_PROT_OFF);                // Code Protect OFF
/*End Configuration For ET-dsPIC30F2010 */
```

โดยรายละเอียดและ Keyword ของ Macro ทั้งหมดที่ใช้ในการกำหนดค่า Configuration ของ dsPIC30F2010 นั้น ผู้ใช้สามารถดูได้จาก “..\pic30_tools\support\h\p30f2010a2.h”

*****หมายเหตุ***** สำหรับบอร์ด ET-dsPIC30F2010 Training Kit V1.0 / EXP 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 ซึ่งตามปกติแล้วระบบสัญญาณนาฬิกาของ dsPIC30F2010 จะสามารถเลือกกำหนดได้หลายแหล่งขึ้นอยู่กับการทำงานของวงจรใช้งาน แต่ในกรณีที่ใช้กับบอร์ด ET-dsPIC30F2010 นั้น ระบบสัญญาณนาฬิกาจะได้รับการออกแบบมาให้ใช้งานกับตัวกำเนิดความถี่แบบ 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 พร้อมทั้งเปิดการทำงานของวงจรควบคุมความถี่ Phase-Lock-Loop ด้วยอัตราการคูณ 16 เท่า ด้วย ซึ่งจะส่งผลให้ค่าความถี่ของสัญญาณนาฬิกาของบอร์ด ET-dsPIC30F2010 มีค่า 117.9648 MHz ซึ่งความถี่ของสัญญาณนาฬิกาดังกล่าวจะถูกนำไปหารให้เหลือ 1/4 ก่อนที่จะป้อนให้กับระบบของ MCU ดังนั้นค่าความถี่ของสัญญาณนาฬิกาของระบบ ที่ใช้ในบอร์ด ET-dsPIC30F2010 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 = เป็นการเลือกเปิดการทำงานของ (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-dsPIC30F2010 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 ตามปกติ
 - 1001 = “ERC” เป็นการเลือกใช้สัญญาณนาฬิกาจากวงจร RC ภายนอก โดยต่อสัญญาณนาฬิกาเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น Output ของสัญญาณนาฬิกาที่ได้ โดยมีค่าความถี่เป็น $\frac{1}{4}$ หรือ $F_{osc}/4$

- 1011 = “EC” เป็นการเลือกใช้สัญญาณนาฬิกา 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น Output ของสัญญาณนาฬิกาที่รับจาก OSC1 โดยมีค่าความถี่เป็น $\frac{1}{4}$ หรือ $F_{osc}/4$
- 1100 = “ECIO” เป็นการเลือกใช้สัญญาณนาฬิกา 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปกติ
- 1101 = “EC PLL 4x” เป็นการเลือกใช้สัญญาณนาฬิกา 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปกติ โดยเปิดการทำงานของวงจรรักษาความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 4 เท่าของความถี่ที่ป้อนให้กับขา OSC1 ด้วย
- 1110 = “EC PLL 8x” เป็นการเลือกใช้สัญญาณนาฬิกา 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปกติ โดยเปิดการทำงานของวงจรรักษาความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 8 เท่าของความถี่ที่ป้อนให้กับขา OSC1 ด้วย
- 1111 = “EC PLL 16x” เป็นการเลือกใช้สัญญาณนาฬิกา 0-40MHz จากแหล่งกำเนิดภายนอก โดยต่อเข้ากับขา OSC1 ส่วน OSC2 จะใช้เป็น I/O ตามปกติ โดยเปิดการทำงานของวงจรรักษาความถี่ (Phase-Lock-Loop) ด้วยค่าอัตราการคูณเป็น 16 เท่าของความถี่ที่ป้อนให้กับขา OSC1 ด้วย

หมายเหตุ การกำหนดค่า Configuration ของ FOSC นั้น จะมีผลต่อการทำงานของ MCU โดยตรง ซึ่งถ้ากำหนดไม่ถูกต้องจะทำให้ MCU ไม่สามารถทำงานได้ หรือทำงานได้ไม่ถูกต้อง ถึงแม้ว่าผู้ใช้จะเขียนโปรแกรมได้อย่างถูกต้องทุกประการก็ตามที่ โดยในกรณีของการใช้งานบอร์ด “ET-dsPIC30F2010” นั้น จะเห็นได้ว่าวงจรมีความถี่จะถูกออกแบบให้ใช้ตัวกำเนิด XTAL ค่า 7.3728 MHz ต่อกับขา OSC1 และ OSC2 ของ MCU โดยตรง ซึ่งนั่นก็หมายความว่า การกำหนดค่า Configuration ของ FOSC ที่จะใช้งานกับบอร์ด “ET-dsPIC30F2010” นั้นจะต้องกำหนดเป็น Primary แบบ XT เท่านั้น ซึ่งถ้าเลือกเป็นอย่างอื่นจะทำให้ MCU ไม่สามารถทำงานได้ แต่ในกรณีที่ต้องการใช้งานด้วยโปรแกรมที่สร้างโดย อีทีที จะต้องกำหนดการทำงานของ FOSC ให้เปิดการทำงานของวงจรรักษาความถี่จากตัวกำเนิดความถี่ค่า 7.3728MHz ด้วยอัตรา 16 เท่าด้วย (XT PLL16X) ถ้าเลือกเป็น XT หรือ XT PLL 4 หรือ XT PLL 8 ถึงแม้ว่า MCU จะสามารถทำงานได้ แต่ก็ส่งผลกระทบได้ค่าความถี่ ไม่ตรงกับที่คำนวณไว้ในโปรแกรม จะทำให้การทำงานของโปรแกรมไม่ถูกต้อง เช่น ความเร็วของการทำงานช้าลง ค่าอัตรา Baud rate ของพอร์ตสื่อสารอนุกรมผิดเพี้ยนไป เป็นต้น 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-dsPIC30F2010 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 ซึ่งในกรณีนี้ผู้ใช้ต้องเขียนคำสั่งเพื่อรีเซ็ตค่าการนับของ Watchdog ก่อนเกิดการ Overflow ในส่วนต่างๆของโปรแกรมเองด้วย ไม่เช่นนั้นแล้วเมื่อการนับของ Watchdog เกิดการ Overflow ขึ้น วงจร Watchdog จะสั่งรีเซ็ตการทำงานของ 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 โดยในระบบไมโครคอนโทรลเลอร์นั้น 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 โดยการเขียนคำสั่งสำหรับรีเซ็ตค่าการนับของ Watchdog แทรกไว้ในส่วนต่างๆของโปรแกรม ก่อนที่ค่าการนับของ Watchdog จะเกิดการ Overflow ขึ้น แต่ถ้าหากว่า MCU เกิดการหยุดการทำงานขึ้น ไม่ว่าจะเกิดจากสาเหตุใดก็ตาม MCU ก็จะไม่สามารถส่งสัญญาณไปรีเซ็ตค่าการนับของ Watchdog ได้ ดังนั้น MCU ก็จะถูก Watchdog รีเซ็ตให้กลับมาเริ่มต้นทำงานใหม่ได้อีก

แต่อย่างไรก็ตามในการพัฒนาโปรแกรมใช้งานนั้น ถ้าผู้พัฒนาโปรแกรมยังไม่มี ความชำนาญในการเขียนโปรแกรมเพียงพอ โดยเฉพาะในระหว่างที่เป็นช่วงเริ่มต้นของการเรียนรู้ นั้น ควรสั่งปิดการทำงานของ Watchdog ไว้ก่อน ไม่เช่นนั้นแล้วจะเป็นการเพิ่มความสับสนให้กับผู้ใช้ได้ เนื่องจากถ้ามีการแทรกคำสั่งสำหรับรีเซ็ตค่าการนับของ Watchdog ไม่เหมาะสมก็อาจทำให้ Watchdog ส่งรีเซ็ตการทำงานของ MCU ในขณะที่ MCU กำลังทำงานตามโปรแกรมปกติอยู่ก็เป็นได้ ดังนั้นขอแนะนำให้สั่งปิดการทำงานของ Watchdog ไว้ก่อน จนเมื่อสามารถพัฒนาโปรแกรมร่วมกับวงจรส่วนอื่นๆ จนโปรแกรมสามารถทำงานได้อย่างถูกต้องครบถ้วนแล้ว จึงค่อยเปิดการทำงานและแทรกคำสั่งสำหรับ รีเซ็ตค่าการนับของ Watchdog ในภายหลัง ซึ่งตัวอย่างโปรแกรมต่างๆที่ทาง อีทีที จัดทำขึ้นนั้นก็ได้จัดการเรื่อง 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 ตามปกติ 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 ในระหว่างการเกิดการรีเซ็ตอยู่ 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] ใช้สำหรับกำหนดระดับค่าของแรงดันที่จะใช้เป็นจุดอ้างอิงในการตรวจสอบความผิดปกติของระดับแหล่งจ่ายของวงจร 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

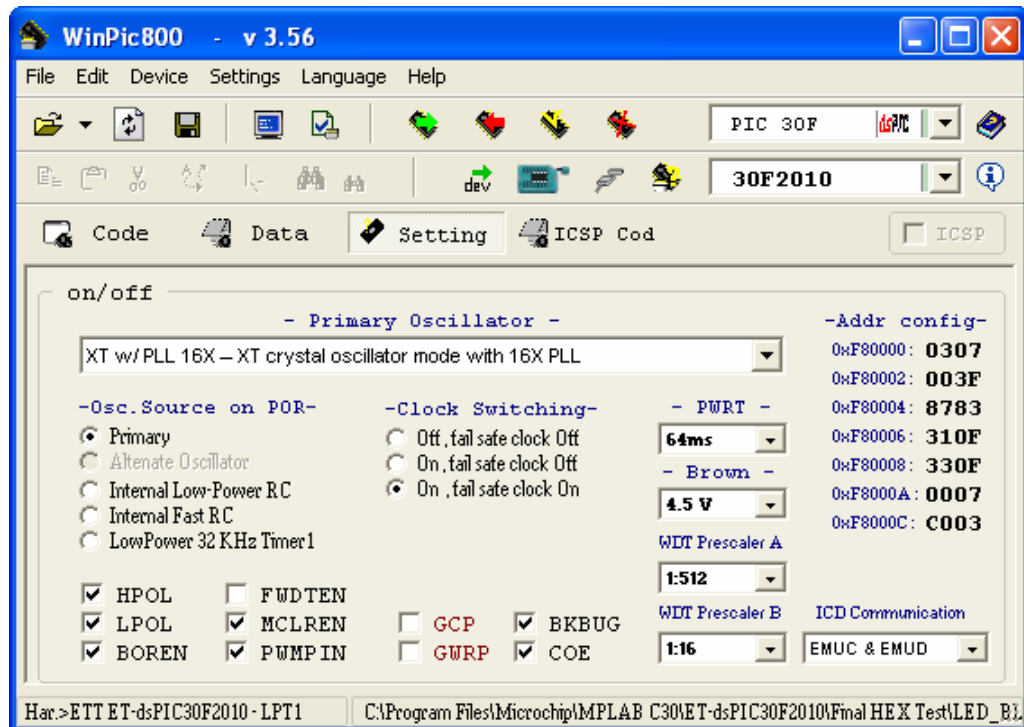
บิตข้อมูล	23	22	21	20	19	18	17	16
ชื่อบิต	-	-	-	-	-	-	-	-

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

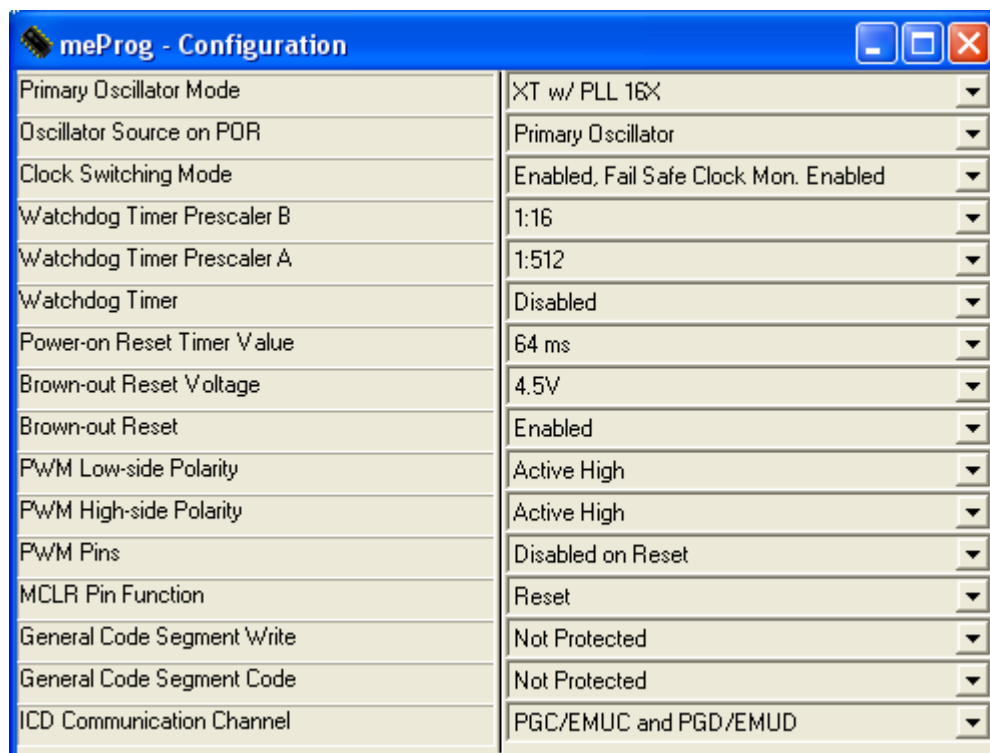
บิตข้อมูล	7	6	5	4	3	2	1	0
ชื่อบิต	-	-	-	-	-	-	GCP	GWRP

ตาราง แสดงค่าบิตต่างๆของ FGS Configurations Register

- **GCP** ใช้สำหรับกำหนดการป้องกันการอ่านข้อมูลของหน่วยความจำโปรแกรม หรือ Flash Memory จากภายนอก โดยถ้ากำหนดให้บิตนี้เป็น “1” จะเป็นการปิด (Disable) ระบบป้องกันการอ่านข้อมูลจากหน่วยความจำ Flash แต่ถ้าเลือกกำหนดให้บิตนี้เป็น “0” จะเป็นการเปิด (Enable) ระบบป้องกันการอ่านข้อมูลจากหน่วยความจำ Flash ซึ่งจะทำให้ไม่สามารถส่งอ่านข้อมูลจากหน่วยความจำ Flash ใน MCU จากภายนอกได้ โดยถ้าเลือกกำหนดให้บิตนี้มีค่าเป็น “0” แล้วจะไม่สามารถส่งเปลี่ยนค่าให้กลับมาเป็น “1” ได้อีก นอกจากจะส่งลบข้อมูลออกจากหน่วยความจำทั้งหมดเสียก่อน ซึ่งหลังจากส่งลบข้อมูลในหน่วยความจำโปรแกรมทั้งหมดแล้วบิตนี้จะมีค่าเป็น “1” โดยอัตโนมัติ
- **GWRP** ใช้สำหรับกำหนดการป้องกันการเขียนข้อมูลของหน่วยความจำโปรแกรม หรือ Flash Memory โดยถ้ากำหนดให้บิตนี้เป็น “1” จะเป็นการปิด (Disable) ระบบป้องกันการเขียนข้อมูลใหม่ให้กับหน่วยความจำ Flash แต่ถ้าเลือกกำหนดให้บิตนี้เป็น “0” จะเป็นการเปิด (Enable) ระบบป้องกันการเขียนข้อมูลให้กับหน่วยความจำ Flash ซึ่งจะทำให้ไม่สามารถส่งเขียนข้อมูลให้กับหน่วยความจำ Flash ใน MCU เพิ่มเติมได้อีกจนกว่าจะส่งลบข้อมูลเดิมทั้งหมดออกเสียก่อน โดยถ้าเลือกกำหนดให้บิตนี้มีค่าเป็น “0” แล้วจะไม่สามารถส่งเปลี่ยนค่าให้กลับมาเป็น “1” ได้อีก นอกจากจะส่งลบข้อมูลออกจากหน่วยความจำทั้งหมดเสียก่อน ซึ่งหลังจากส่งลบข้อมูลในหน่วยความจำโปรแกรมทั้งหมดแล้วบิตนี้จะมีค่าเป็น “1” โดยอัตโนมัติ



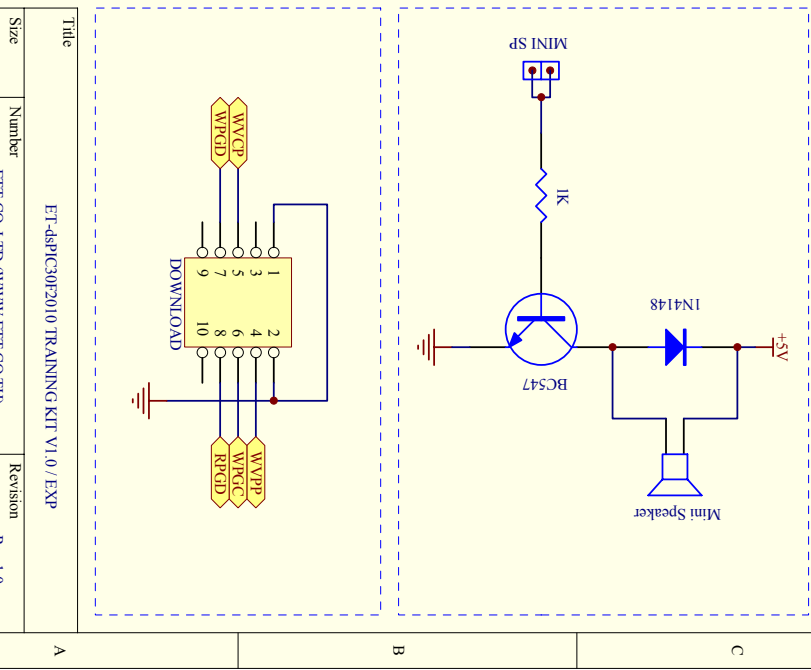
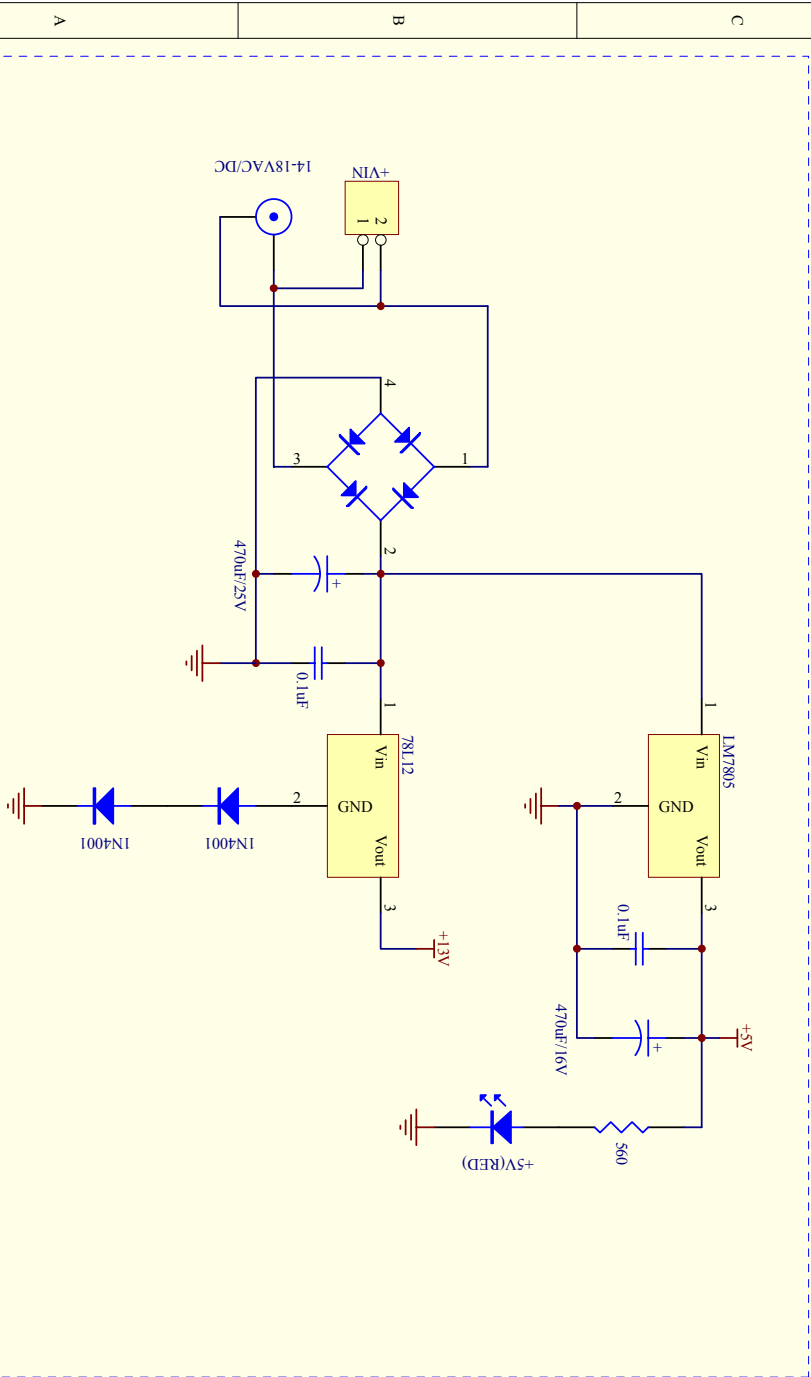
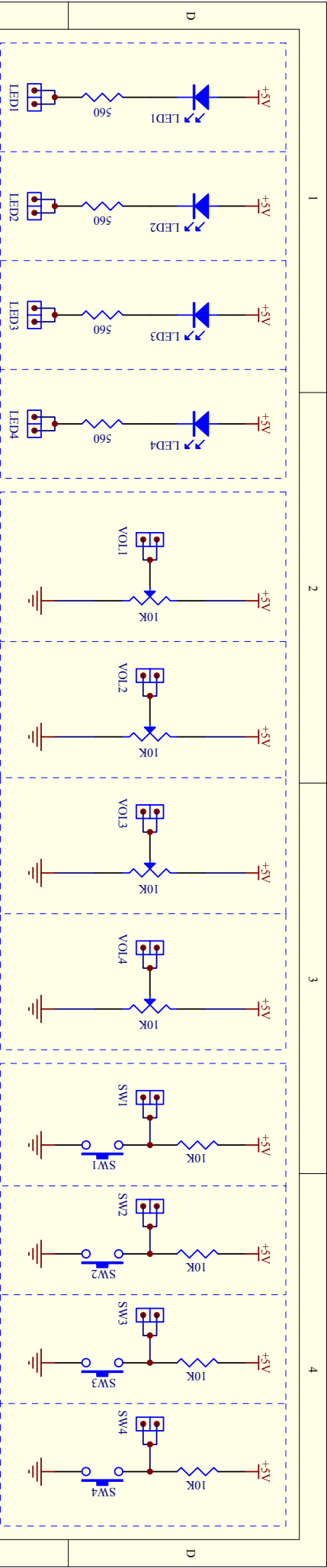
รูปแสดง ค่า Configuration ที่เหมาะสมของ ET-dsPIC30F2010 โดยใช้ “WinPic800”



รูปแสดง ค่า Configuration ที่เหมาะสมของ ET-dsPIC30F2010 โดยใช้ “melabs Programmer”



Title			ET-dsPIC300F2010 TRAINING KIT V1.0 / EXP	
Size	Number	Revision		
A4	ETT CO., LTD (WWW.ETT.CO.TH)	Rev 1.0		
Date:	8-Oct-2005	Sheet 1 of 2		
File:	D:\Scheme\ET-dsPIC300F2010.Dsb	Drawn By:	Eatechapa Matkam	



Title		
ET-dsPIC30F2010 TRAINING KIT V1.0 / EXP		
Size	Number	Revision
A4	ETT CO.,LTD (WWW.ETT.CO.TH)	Rev 1.0
Date:	8-Oct-2005	Sheet 2 of 2
File:	D:\Scheme\ET-dsPIC30F2010\1.Ddb	Drawn by: Eakachai Makam