

ตัวอย่างโปรแกรมและการทดสอบ

สำหรับตัวอย่างโปรแกรมทั้งหมดของ dsPIC30F2010 ที่ทาง อีทีที จัดทำขึ้น เพื่อเป็นแนวทางการใช้งาน ทรัพยากรต่างๆของบอร์ด “ET-dsPIC30F2010” เท่านั้น โดยโปรแกรมได้รับการทดสอบและใช้งานภายใต้เงื่อนไข ที่ระบุไว้ในส่วนคำอธิบายของตัวอย่างนั้นๆ ข้อจำกัดในการใช้งานที่อาจมีอยู่ในตัวอย่างโปรแกรมเป็นหน้าที่ของผู้ใช้ที่ จะนำไปปรับปรุง พัฒนา หรือ ดัดแปลงแก้ไขเพิ่มเติมเอง โดยตัวอย่างโปรแกรมต่างๆใช้เงื่อนไขและข้อกำหนดดังนี้

- ตัวอย่างทั้งหมดได้รับการทดสอบและใช้งานกับ MPLAB C30 Compiler V1.33 เป็นหลัก ถ้าผู้ใช้ เลือกใช้ Compiler ในรุ่นที่แตกต่างจากนี้ไปขอให้ศึกษาข้อแตกต่างของ Compiler และปรับแก้ โปรแกรมเอง ในกรณีที่ไม่สามารถใช้งานร่วมกันกับ Compiler ที่ใช้งานอยู่ได้
- ในการติดตั้งโปรแกรม MPLAB C30 นั้นควรติดตั้งไว้ที่ “c:\pic30_tools\...” เพื่อความสะดวกใน การเรียกใช้งาน และการอ้างอิงการใช้งานจากคู่มือต่างๆของ อีทีที ถ้าผู้ใช้ทำการติดตั้งโปรแกรม แตกต่างไปจากนี้ อาจไม่สามารถทำการทดลองตามการอ้างอิง จากคู่มือการใช้งานที่ทาง อีทีที จัดทำไว้ให้ได้ทั้งหมด ซึ่งเป็นหน้าที่ของผู้ใช้ที่จะต้องทำความเข้าใจและแก้ไขปัญหานั้นเอง
- ต้องกำหนดให้ MCU ใช้สัญญาณนาฬิกาจาก XTAL ค่า 7.3728 MHz ร่วมกับวงจรคูณความถี่ โดยกำหนดอัตราการคูณเป็น 16 เท่า (เลือก Configuration ของ FOSC เป็น XT w/PLL 16X) ซึ่ง จะทำให้ ได้ค่าความถี่ในการทำงานเป็น 117.9648 MHz โดยระบบบัสจะมีความถี่เป็น 1/4 ของ ความถี่นี้ ซึ่งก็คือ 29.4912 MHz โดยตัวอย่างโปรแกรมทั้งหมดจะใช้ค่าความถี่ 29.4912 MHz นี้ เป็นฐานในการคำนวณค่าตัวแปรอื่นๆที่เกี่ยวข้องไม่ว่าจะเป็นการกำหนดค่าการนับของระบบ วงจรที่เกี่ยวข้องกับ Timer / Counter / PWM หรือ Baudrate ของพอร์ตสื่อสารอนุกรม
- ต้องปิดการทำงานของวงจร Watchdog ไว้ เนื่องจากโปรแกรมตัวอย่างทั้งหมดจะไม่ได้จัดการเรื่อง การรีเซ็ต Watchdog ให้อัตโนมัติ
- การทำงานของโปรแกรมตัวอย่างบางส่วน จำเป็นต้องเชื่อมต่อวงจรกับอุปกรณ์ภายนอกซึ่งไม่มีอยู่ ภายในบอร์ด โดยทางอีทีที จะกำหนดรายละเอียดการเชื่อมต่อกำกับไว้ด้วยในส่วนคำอธิบาย ในโปรแกรมตัวอย่างนั้นๆ โดยตัวอย่างโปรแกรมทั้งหมดจะได้รับการทดสอบการทำงานว่าใช้งานได้ ร่วมกับอุปกรณ์ที่กำหนดไว้เท่านั้น ถ้าผู้ใช้ นำตัวอย่างไปเชื่อมต่อกับอุปกรณ์อื่นๆที่ไม่ตรงกับที่ระบุ ไว้ แล้วโปรแกรมไม่สามารถทำงานได้ เป็นหน้าที่ของผู้ใช้ที่จะทำการแก้ไขดัดแปลงเอง
- ตัวอย่างทั้งหมดจะมี Source Code ให้อีกด้วย ผู้ใช้สามารถศึกษาทำความเข้าใจได้จากคำอธิบายใน โปรแกรม ส่วนรายละเอียดการใช้งานอื่นๆสามารถศึกษาเพิ่มเติมได้จาก Data Sheet และคู่มือ ต่างๆที่เกี่ยวข้อง โดยทางอีทีที ได้รวบรวม HEX File ซึ่งทำการสังเคราะห์แล้วจากตัวอย่างโปรแกรม ทั้งหมดโดยรวบรวมไว้ใน Folder ชื่อ “Final HEX Test” เพื่อให้ผู้ใช้สามารถทดลอง Download เพื่อทดสอบดูผลการทำงานได้ทันที โดยไม่ต้องสังเคราะห์ใหม่

ตัวอย่างโปรแกรม PORT_LED_BLINK (LED_BLINK.HEX)

เป็นตัวอย่างโปรแกรม ไฟกระพริบด้วย LED 1 ดวง โดยควบคุมจากพอร์ต RB0 โดยในการทดลองให้ทำการต่อสัญญาณจากพอร์ต RB0 เข้ากับ LED1 Output แล้วดูการทำงานของ LED1 ซึ่งจะกระพริบติดและดับสลับกันไปไม่รู้จบ

- ต่อ RB0 ของบอร์ด ET-dsPIC30F2010 เข้ากับ LED1

ตัวอย่างโปรแกรม PORT_LED_MOVING (LED_MOVING.HEX)

เป็นตัวอย่างโปรแกรม ไฟวิ่งด้วย LED 4 ดวง โดยใช้พอร์ต RB0-RB3 ในการควบคุมการติดและดับของ LED โดยในการทดลองให้ทำการต่อสัญญาณจากพอร์ต RB0-RB3 เข้ากับ LED1-LED4 ตามลำดับ แล้วดูการทำงานของ LED1-LED4 ซึ่งจะติดทีละดวง วิ่งสลับ จากซ้ายไปขวา และ จากขวาไปซ้าย สลับกันไปไม่รู้จบ

- ต่อ RB0 ของบอร์ด ET-dsPIC30F2010 เข้ากับ LED1
- ต่อ RB1 ของบอร์ด ET-dsPIC30F2010 เข้ากับ LED2
- ต่อ RB2 ของบอร์ด ET-dsPIC30F2010 เข้ากับ LED3
- ต่อ RB3 ของบอร์ด ET-dsPIC30F2010 เข้ากับ LED4

ตัวอย่างโปรแกรม TIMER_SQUARE_WAVE_1KHZ (TIMER_BEEP.HEX)

สำหรับตัวอย่างนี้เป็นตัวอย่างการใช้ Timer1 ควบคุม Output ของ RB0 เพื่อสร้างเป็นสัญญาณ Pulse แบบ Square Wave ความถี่ 1 KHz โดยการทำงานของโปรแกรมจะเป็นการเปิดการทำงานของ Timer1 ให้ทำการสร้างสัญญาณ Pulse 1 KHz ออกที่ขา RB0 อยู่ระยะเวลาหนึ่ง จากนั้นก็จะสั่งหยุดการทำงานของ Timer1 พร้อมกับกำหนดให้ RB0 มีสถานะโลจิกเป็น “0” อยู่ระยะเวลาหนึ่ง แล้ววนกลับไปเปิดการทำงาน ของ Timer1 อีกวนรอบอยู่เช่นนี้ไม่รู้จบ ในการทดสอบการทำงานจะใช้ลำโพงเล็กในการแสดงผล โดยให้ต่อ RB0 เข้ากับจุดต่อ MINI SP. ของวงจรขับลำโพง ซึ่งจะได้ผลเป็นการส่งเสียง Beep เป็นจังหวะสลับกันไปไม่รู้จบ หรือถ้าใช้ Oscilloscope วัดสัญญาณที่ขา RB0 ก็จะได้สัญญาณ Pulse ที่มีค่าความถี่ 1KHz ตามจังหวะที่ Timer1 ถูกเปิดการทำงานอยู่

- ต่อ RB0 ของบอร์ด ET-dsPIC30F2010 เข้ากับ MINI SP.

ตัวอย่างโปรแกรม COMPARE_GEN_PWM_SERVO (PWM_SERVO.HEX)

ตัวอย่างโปรแกรมนี้นี้เป็นการแสดงให้เห็นตัวอย่างการใช้ Timer2 ควบคุมการสร้างสัญญาณ PWM ของวงจร Output Compare OC1 เพื่อสร้างสัญญาณ PWM ออกที่ขา RD0 ซึ่งมีค่าความถี่ประมาณ 56.2 Hz โดยมีคาบเวลา Period ของสัญญาณ PWM ประมาณ 17.79mS โดยในการทดสอบการทำงานของโปรแกรมนี้อาจใช้ SERVO MOTOR เป็นตัวทดสอบการทำงาน หรือใช้ Oscilloscope สำหรับวัดรูปคลื่นของสัญญาณแทนก็ได้

โดยการทำงานของโปรแกรมนี้อาจเริ่มจากการกำหนดให้ Timer2 ควบคุมการทำงานของ OC1 ในการสร้างสัญญาณ PWM โดยจะให้ Output PWM ออกที่ขา RD0 โดยการทำงานของโปรแกรมนี้อาจจะวนรอบสั่งให้ OC1 สร้างสัญญาณ PWM ซึ่งมีคาบเวลาคงที่ประมาณ 17.79mS (56.2Hz) แต่จะมีการสั่งปรับค่า Duty Cycle ของสัญญาณ Pulse ในซีก็บวก็ให้ เปลี่ยนแปลง ค่าระหว่าง 0.75mS → 1.50ms → 2.25mS → 1.50mS → 0.75mS วนรอบอยู่เช่นนี้ตลอดเวลา โดยมีการหน่วงเวลาให้สร้างสัญญาณแต่ค่าของ Duty Cycle อยู่ระยะหนึ่ง ซึ่งถ้านำ Servo Motor มาต่อเข้ากับบอร์ด โดยใช้ RD0 เป็นสัญญาณในการควบคุมการทำงานของ Servo Motor ก็จะทำให้เห็นว่า Servo Motor มีการหมุนไปมาระหว่าง “ซ้าย↔กึ่งกลาง↔ขวา” วนรอบอยู่เช่นนี้ไม่รู้จบ

- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของ Servo Motor
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของ Servo Motor
- ต่อ RD0 ของบอร์ด ET-dsPIC30F2010 เข้ากับ สัญญาณควบคุมของ Servo Motor

ตัวอย่างโปรแกรม MCPWM_DC_MOTOR (DC_MOTOR.HEX)

ตัวอย่างนี้เป็นตัวอย่างการใช้งานวงจรควบคุม Motor ของ dsPIC30F2010 โดยเฉพาะซึ่งเรียกว่า MCPWM ซึ่งวงจรส่วนนี้จะมีประสิทธิภาพกว่า PWM ที่สร้างการวงจรของ Output Compare หลายอย่าง โดยเฉพาะมีวงจร Fault Input สำหรับหยุดการทำงานของ PWM เพื่อหยุดการทำงานของ Motor ในทันที ซึ่งความสามารถส่วนนี้จะมีประโยชน์มากในกรณีที่นำ PWM ไปควบคุมการทำงานของ Motor ซึ่งอาจเป็นส่วนประกอบอยู่ในเครื่องจักรกลต่างๆ แล้วเกิดข้อผิดพลาดใดๆขึ้น เราสามารถนำสัญญาณที่ได้จาก Sensor ที่ใช้ตรวจจับความผิดพลาดมาป้อนให้กับ Fault Input เพื่อสั่งหยุดการทำงานของ Motor ในทันทีโดยอัตโนมัติ โดยในตัวอย่างการทดลองนี้จะใช้โมดูล “ET-MINI DC-MOTOR” เป็นอุปกรณ์ร่วมในการทดลองด้วย โดยให้ต่อวงจรประกอบทดลองดังนี้

- ต่อ RB0 (ADC0) ของบอร์ด ET-dsPIC30F2010 เข้ากับ VOL1 (ตัวต้านทานปรับค่าแรงดัน 0-5V)
- ต่อ RE8 (FLTA) ของบอร์ด ET-dsPIC30F2010 เข้ากับ Push-Button SW1 (Error Fault Input)
- ต่อ RE1 (PWM1H) ของบอร์ด ET-dsPIC30F2010 เข้ากับ EN ของบอร์ด ET-MINI DC-MOTOR
- ต่อ RB1 (DIR1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ IN1 ของบอร์ด ET-MINI DC-MOTOR
- ต่อ RB2 (DIR2) ของบอร์ด ET-dsPIC30F2010 เข้ากับ IN2 ของบอร์ด ET-MINI DC-MOTOR
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI DC-MOTOR
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI DC-MOTOR

โดยการทำงานของโปรแกรมจะกำหนดการทำงานของ DCPWM ให้สร้างสัญญาณ PWM ค่าความถี่ประมาณ 57.6 KHz โดยกำหนดให้สามารถปรับค่า Duty Cycle ของ PWM ได้ 1024 ระดับ พร้อมทั้งเปิดการทำงานของวงจรตรวจจับความผิดพลาด Fault ชุด A สำหรับใช้หยุดการทำงานของ PWM (DC Motor) ด้วย จากนั้นก็กำหนดให้ RB0 ทำหน้าที่เป็น ADC Input สำหรับรับค่าแรงดัน 0-5V ที่ได้จากการปรับค่าความต้านทานของตัวต้านทานปรับค่า VOL1 โดย ADC จะแปลงค่าแรงดัน 0-5V เป็นข้อมูล 0-1023 เพื่อนำไปใช้ปรับเปลี่ยนค่า Duty Cycle ของ PWM ที่ใช้ควบคุมความเร็วของมอเตอร์ หลังจากนั้นก็กำหนดให้ RB1 และ RB2 ทำหน้าที่เป็น Output โดยกำหนดให้มีค่าตรงกันข้าม โดย RB1 และ RB2 ใช้เป็นตัวควบคุมทิศทางการหมุนของ DC Motor ซึ่งถ้ามีค่าเหมือนกันจะหยุดการทำงานของมอเตอร์ แต่ถ้ามีค่าต่างกันจะทำให้มอเตอร์หมุนไปตามทิศทางที่กำหนดโดย RB1 และ RB2 ซึ่งตัวอย่างนี้จะกำหนดทิศทางไว้คงที่ ซึ่งถ้าต้องการเปลี่ยนทิศทางก็สามารถกำหนดได้จากค่าที่กำหนดให้กับ RB1 และ RB2 จากนั้นโปรแกรมจะเข้าทำงานในวงรอบ โดยจะวนอ่านค่า ADC แล้วนำค่าที่ได้จากการแปลงค่าของ ADC ไปปรับค่า Duty Cycle ของ PWM ดังนั้นเมื่อปรับค่า VOL1 ก็จะเป็นการปรับค่าความเร็วของ DC Motor นั่นเอง ส่วน SW1 จะใช้จำลองการทำงานแทนสัญญาณความผิดพลาด ซึ่งเมื่อกดสวิตช์จะทำให้ มอเตอร์หยุดหมุนทันที ถึงแม้ว่าสัญญาณความผิดพลาดจะหายไปแล้วก็ตาม โดยในตัวอย่างนี้ถ้าต้องการให้ Motor เริ่มต้นหมุนใหม่หลังเกิดความผิดพลาดแล้ว จะต้องสั่ง Reset ให้ MCU เริ่มต้นทำงานใหม่เสียก่อน

ตัวอย่างโปรแกรม PORT_PIN_INTERRUPT (PORT_INTERRUPT.HEX)

ตัวอย่างนี้เป็นการแสดงตัวอย่างการใช้ PORT ทำหน้าที่เป็นขารับสัญญาณ Interrupt จากภายนอกหรือ External Interrupt โดยจะใช้ Push Button Switch เป็นตัวสร้างสัญญาณ Interrupt และใช้ LED Output เป็นตัวแสดงผลการทำงานของการทำงาน Interrupt โดยในการทดลองนี้ ให้ต่อวงจรดังนี้

- ต่อ RE8 (INT0) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SW1 (Input Interrupt ของ INT0)
- ต่อ RD0 (INT1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SW2 (Input Interrupt ของ INT1)
- ต่อ RB0 ของบอร์ด ET-dsPIC30F2010 เข้ากับ LED1 (ใช้แสดงการทำงานของ INT0)
- ต่อ RB1 ของบอร์ด ET-dsPIC30F2010 เข้ากับ LED2 (ใช้แสดงการทำงานของ INT1)

โดยการทำงานของโปรแกรมจะเป็นการกำหนดให้ RB0 และ RB1 ทำหน้าที่เป็น Output สำหรับขับ LED จากนั้นจะทำการ กำหนดการทำงานของ RE8 และ RD0 ให้ทำหน้าที่เป็น Input รับสัญญาณการ Interrupt จากสัญญาณภายนอก โดย RE8 จะเป็น Input ของ INT0 และ RD0 เป็น Input ของ INT1 โดยในตัวอย่างจะกำหนดให้เกิดการ Interrupt จากสัญญาณที่เป็นขอบขาลง (Falling Edge) ของสัญญาณ Pulse ซึ่งเกิดจากการกดสวิตช์ จากนั้นโปรแกรมก็จะวนรอบอยู่กับที่โดยไม่ทำอะไร เพื่อรอการ Interrupt

ซึ่งผลการทำงานของการทำงาน Interrupt จะอยู่ที่ส่วนของโปรแกรมบริการ Interrupt โดยจะเป็นการสลับสถานะลอจิกของ Output ให้เป็นตรงข้ามเมื่อเกิดการ Interrupt ขึ้นในแต่ละครั้ง โดย INT0 จะควบคุมสถานะทาง Output ของ RB0 ส่วน INT1 ก็จะมาควบคุมสถานะ Output ของ RB1 ดังนั้นเมื่อทดลองทำการกด Push-Button SW แต่ละครั้งก็จะเห็น LED ติดและดับสลับกันไปตลอดเวลา

ตัวอย่างโปรแกรม LCD_4BIT (LCD_16X2.HEX)

ตัวอย่างนี้เป็นการใช้พอร์ต RB0-RB5 ควบคุมการแสดงผลของ LCD แบบ Character โดยในตัวอย่างนี้เลือกใช้ LCD ขนาด 16 ตัวอักษร 2 บรรทัด เป็นอุปกรณ์การทดลอง โดยการทำงานของโปรแกรมจะเป็นการสั่งแสดงข้อความที่หน้าจอ LCD สลับกันไปมาไม่รู้จบ โดยในการเชื่อมต่อสัญญาณระหว่าง dsPIC30F2010 และจอแสดงผล LCD ขนาด 16 ตัวอักษร 2 บรรทัด จะต้องวงจรแบบใช้การควบคุม 4 บิต โดยรายละเอียดดังนี้

- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของ LCD (ขา 1)
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของ LCD (ขา 2)
- ต่อ Output ของตัวต้านทานปรับค่า VOL1 ของบอร์ด ET-dsPIC30F2010 เข้ากับขา VO ของ LCD (ขา 3) เพื่อใช้เป็นตัวปรับความสว่างในการแสดงผลของ LCD
- ต่อ RB4 ของบอร์ด ET-dsPIC30F2010 เข้ากับ RS ของ LCD (ขา 4)
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ R/W ของ LCD (ขา 5)
- ต่อ RB5 ของบอร์ด ET-dsPIC30F2010 เข้ากับ E ของ LCD (ขา 6)
- ต่อ RB0 ของบอร์ด ET-dsPIC30F2010 เข้ากับ D4 ของ LCD (ขา 11)
- ต่อ RB1 ของบอร์ด ET-dsPIC30F2010 เข้ากับ D5 ของ LCD (ขา 12)
- ต่อ RB2 ของบอร์ด ET-dsPIC30F2010 เข้ากับ D6 ของ LCD (ขา 13)
- ต่อ RB3 ของบอร์ด ET-dsPIC30F2010 เข้ากับ D7 ของ LCD (ขา 14)

ตัวอย่างโปรแกรม SPI_74HC595_OUT_LED (74HC595_LED.HEX)

เป็นตัวอย่างโปรแกรม การใช้งาน SPI Master โดยต้องต่อใช้งานร่วมกับ โมดูล ET-MINI I/O BOARD รุ่น “ET-MINI 74HC595” และ “ET-TEST 10P/OUT” โดยการทำงานของโปรแกรมจะเป็นการใช้งานวงจร SPI ของ dsPIC30F2010 ในการส่งสัญญาณ Output ไปยังไอซี 74HC595 ซึ่งเป็นไอซี Output แบบ Serial-In-Parallel-Out โดยจะได้สัญญาณ Output เป็นลอจิกจำนวน 8 Output ซึ่งจากตัวอย่างจะใช้ Output ของ 74HC595 ทั้ง 8 บิต ไปขับ LED ซึ่งอาจใช้บอร์ด “ET-TEST 10P/OUT” ซึ่งเป็นบอร์ด LED Output ขนาด 8 บิตในการแสดงผล หรือจะต่อกับ LED Output ของ บอร์ด ET-dsPIC30F2010 เองก็ได้แต่จะมี LED ให้ทดสอบเพียง 4 Output เท่านั้น โดยผลการทำงานของโปรแกรมจะเห็น LED ทั้ง 8 ดวง ที่ต่อไว้กับ Output ของ 74HC595 ติด และดับ ในลักษณะของไฟวิงทีละดวง จากซ้ายไปขวา และขวาไปซ้าย สลับกันไปไม่รู้จบ โดยวิธีการเชื่อมต่อสัญญาณเพื่อใช้ในการทดลองระหว่างบอร์ด “ET-dsPIC30F2010” และ “ET-MINI 74HC595” มีดังนี้

- ต่อ RF3 (SDO1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ DIN ของบอร์ด ET-MINI 74HC595
- ต่อ RE8 (SCK1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ CLOCK ของบอร์ด ET-MINI 74HC595
- ต่อ RB2 (SS#1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ LATCH ของบอร์ด ET-MINI 74HC595
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI 74HC595
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI 74HC595

ส่วนการต่อ OUTPUT LOGIC ของ “ET-MINI 74HC595” กับ “ET-TEST 10P/OUT” นั้นจะต่อผ่านสายแพรขนาด 10 PIN ซึ่งเป็นขั้วต่อมาตรฐานอยู่แล้ว แต่ถ้าต้องการต่อ OUTPUT LOGIC ของ “ET-MINI 74HC595” กับ LED Output ของบอร์ด ET-dsPIC30F2010 เองนั้น ให้ต่อเฉพาะส่วน OUTPUT LOGIC ของบิต 0-3 ของ 74HC595 เข้ากับ LED1-4 ของ ET-dsPIC30F2010 เท่านั้น

ตัวอย่างโปรแกรม SPI_74HC595_OUT_7SEG (74HC595_7SEG.HEX)

เป็นตัวอย่างโปรแกรม การใช้งาน SPI Master โดยต้องต่อใช้งานร่วมกับ โมดูล “ET-MINI 74HC595” และ “ET-MINI SEG-A” โดยการทำงานของโปรแกรมจะเป็นการใช้งานวงจร SPI ของ dsPIC30F2010 ในการส่งสัญญาณ Output ไปยังไอซี 74HC595 ซึ่งเป็นไอซี Output แบบ Serial-In-Parallel-Out โดยจะได้สัญญาณ Output เป็นลอจิกจำนวน 8 Output ซึ่งจากตัวอย่างจะใช้ Output ของ 74HC595 ทั้ง 8 บิต ไปขับ 7-SEGMENT เพื่อแสดงผล โดยผลการทำงานของโปรแกรมจะเป็นการแสดงค่าตัวเลข “0-9” ต่อเนื่องกันไป ไม่รู้จบ โดยในการเชื่อมต่อสัญญาณระหว่างบอร์ด “ET-dsPIC30F2010” และ “ET-MINI 74HC595” มีดังนี้

- ต่อ RF3 (SDO1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ DIN ของบอร์ด ET-MINI 74HC595
- ต่อ RE8 (SCK1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ CLOCK ของบอร์ด ET-MINI 74HC595
- ต่อ RB2 (SS#1) ของบอร์ด ET-dsPIC30F2010 เข้ากับ LATCH ของบอร์ด ET-MINI 74HC595
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI 74HC595
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI 74HC595

ส่วนการต่อ OUTPUT LOGIC จาก “ET-MINI 74HC595” กับ “ET-MINI 7SEG-A” นั้นสามารถเสียบขั้วต่อสัญญาณของทั้ง 2 บอร์ดเข้าด้วยกันได้ทันที

ตัวอย่างโปรแกรม I2C_PCF8574_OUT_LED (PCF8574A_LED.HEX)

เป็นตัวอย่างโปรแกรม การใช้งาน I2C Master โดยต้องต่อใช้งานร่วมกับ โมดูล “ET-MINI PCF8574A” และ “ET-TEST 10P/OUT” โดยการทำงานของโปรแกรมจะเป็นการใช้งานวงจร I2C ของ dsPIC30F2010 ในการส่งสัญญาณ Output ไปยังไอซี PCF8574A ซึ่งเป็นไอซี I/O แบบ I2C โดยจะได้สัญญาณ Output เป็นลอจิกจำนวน 8 Output ซึ่งจากตัวอย่างจะใช้ Output ของ PCF8574A ทั้ง 8 บิต ไปขับ LED ซึ่งในการทดสอบการทำงานอาจใช้บอร์ด “ET-TEST 10P/OUT” ซึ่งเป็นบอร์ด LED Output ขนาด 8 บิต ในการแสดงผล หรือจะต่อกับ LED Output ของ บอร์ด ET-dsPIC30F2010 เองก็ได้แต่จะมี LED ให้ทดสอบเพียง 4 Output เท่านั้น โดยผลการทำงานของโปรแกรมจะเห็น LED ทั้ง 8 ดวง ที่ต่อไว้กับ Output ของ PCF8574A ติด และดับ ในลักษณะของไฟวิ่งทีละดวง จากซ้ายไปขวา และขวาไปซ้าย สลับกันไปไม่รู้จบ โดยการเชื่อมต่อสัญญาณระหว่างบอร์ด “ET-dsPIC30F2010” และ “ET-MINI PCF8574A” มีดังนี้

- ต่อ RF2(SDA) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SDA ของบอร์ด ET-MINI PCF8574A
- ต่อ RF3(SCL) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SCL ของบอร์ด ET-MINI PCF8574A
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI PCF8574A
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI PCF8574A
- กำหนดตำแหน่งแอดเดรส ของบอร์ด ET-MINI PCF8574A ให้มีค่าเป็น “000” โดยการกำหนดให้ A0-A2 เป็น LO ทั้งหมด
- เลือก Enable การ Pull-Up ของ SCL และ SDA ของ ET-MINI PCF8574A

ส่วนการต่อ OUTPUT LOGIC ของ “ET-MINI PCF8574A” กับ “ET-TEST 10P/OUT” นั้นจะต่อผ่านสายแพรขนาด 10 PIN ซึ่งเป็นขั้วต่อมาตรฐานอยู่แล้ว แต่ถ้าต้องการต่อ OUTPUT LOGIC ของ “ET-MINI PCF8574A” กับ LED Output ของบอร์ด ET-dsPIC30F2010 เองนั้น ให้ต่อเฉพาะส่วน OUTPUT LOGIC ของบิต 0-3 ของ PCF8574A เข้ากับ LED1-4 ของ ET-dsPIC30F2010 เท่านั้น

ตัวอย่างโปรแกรม I2C_PCF8574_OUT_7SEG (PCF8574A_7SEG.HEX)

เป็นตัวอย่างโปรแกรม การใช้งาน I2C Master โดยต้องต่อใช้งานร่วมกับ โมดูล “ET-MINI PCF8574A” และ “ET-MINI SEG-A” โดยการทำงานของโปรแกรมจะเป็นการใช้งานวงจร I2C ของ dsPIC30F2010 ในการส่งสัญญาณ Output ไปยังไอซี PCF8574A ซึ่งเป็นไอซี I/O แบบ I2C โดยจะได้สัญญาณ Output เป็นลอจิกจำนวน 8 Output ซึ่งจากตัวอย่างจะใช้ Output ของ PCF8574A ทั้ง 8 บิต ไปขับ 7-SEGMENT เพื่อแสดงผล โดยผลการทำงานของโปรแกรมจะเป็นการแสดงค่าตัวเลข “0-9” ต่อเนื่องกันไป ไม่รู้จบ โดยในการเชื่อมต่อสัญญาณระหว่างบอร์ด “ET-dsPIC30F2010” และ “ET-MINI PCF8574A” มีดังนี้

- ต่อ RF2 (SDA) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SDA ของบอร์ด ET-MINI PCF8574A
- ต่อ RF3 (SCL) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SCL ของบอร์ด ET-MINI PCF8574A
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI PCF8574A
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI PCF8574A
- กำหนดตำแหน่งแอดเดรส ของบอร์ด ET-MINI PCF8574A ให้มีค่าเป็น “000” โดยการกำหนดให้ A0-A2 เป็น LO ทั้งหมด
- เลือก Enable การ Pull-Up ของ SCL และ SDA ของ ET-MINI PCF8574A

ส่วนการต่อ OUTPUT LOGIC จาก “ET-MINI PCF8574A” กับ “ET-MINI 7SEG-A” นั้นสามารถเสียบหัวต่อสัญญาณของทั้ง 2 บอร์ดเข้าด้วยกันได้ทันที

ตัวอย่างโปรแกรม UART_RX_TX (UART.HEX)

เป็นตัวอย่างการสื่อสารข้อมูลผ่านพอร์ตสื่อสารอนุกรม RS232 ของ dsPIC30F2010 โดยแสดงการรับและส่งข้อมูลระหว่าง dsPIC30F2010 กับ คอมพิวเตอร์ PC ผ่าน Comport โดยตัวอย่างนี้จะใช้พอร์ตสื่อสารอนุกรมของ dsPIC30F2010 ซึ่งใช้สัญญาณ RC13 (TXD) เป็นขาส่งข้อมูล และใช้ RC14 (RXD) เป็นขารับข้อมูล โดยวงจรของบอร์ด ET-dsPIC30F2010 ได้จัดวงจร Line Driver ของ RS232 ไว้ให้เรียบร้อยแล้ว ผู้ใช้สามารถต่อสายสัญญาณ RS232 จาก Comport ของ PC เข้ากับหัว RS232 ของบอร์ดได้ทันที

โดยการทำงานของโปรแกรมจะเป็นการแสดงข้อความต่างๆผ่านทางพอร์ตสื่อสารอนุกรม จากนั้นก็จะรอรับข้อมูลจากพอร์ตสื่อสารอนุกรมพร้อมกับส่งค่าที่รับได้กลับออกมาด้วยทุกครั้ง

- ต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport RS232 ของคอมพิวเตอร์ PC

โดยในการทดสอบการทำงานของโปรแกรมให้ใช้โปรแกรมจำพวก Terminal ต่างๆ เช่น Hyper Terminal หรือ Procomm เป็นตัวกลางในการสื่อสาร โดยให้กำหนดรูปแบบการสื่อสารเป็นดังนี้

- Comport กำหนดเป็น Direct Com(n) โดย (n) หมายถึงหมายเลข Comport ที่ต่อใช้งานอยู่ เช่น ถ้าต่อกับ Com1 ก็กำหนดเป็น Direct Com1
- กำหนดค่า Baud rate = 9600 BPS
- กำหนด Data = 8 Bit
- กำหนด Parity = None
- กำหนด Stop Bit = 1 Bit
- กำหนด Flow Control = None

ตัวอย่างโปรแกรม I2C_RTC_DS1307 (DS1307.HEX)

เป็นตัวอย่างโปรแกรม การใช้งาน I2C Master โดยต้องต่อใช้งานร่วมกับ โมดูล “ET-MINI DS1307” โดยการทำงานของโปรแกรมจะเป็นการใช้งานวงจร I2C ของ dsPIC30F2010 ในการติดต่อสื่อสารกับ DS1307 ซึ่งเป็นไอซี นาฬิกา Real Time Clock โดยโปรแกรมจะทำการตั้งเวลาของ ชั่วโมง นาที วินาที ให้มีค่าเป็น 00:00:00 จากนั้นก็จะวนอ่านค่าเวลาของ ชั่วโมง นาที วินาที มาแสดงผลให้ทราบ โดยใช้พอร์ตสื่อสารอนุกรม RS232 ในการแสดงผล ด้วย อัตรา Baudrate = 9600 โดยการดูผลการทดลองนี้อาจใช้โปรแกรมจำพวก Terminal ต่างๆ เช่น โปรแกรม Hyper Terminal หรือ Procomm ในการแสดงผลการทำงานของโปรแกรมได้ตามต้องการ โดยการเชื่อมต่อสัญญาณเพื่อใช้ในการทดลองระหว่างบอร์ด “ET-dsPIC30F2010” และ “ET-MINI DS1307” มีดังนี้

- ต่อ RF2(SDA) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SDA ของบอร์ด ET-MINI DS1307
- ต่อ RF3(SCL) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SCL ของบอร์ด ET-MINI DS1307
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI DS1307
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI DS1307
- เลือก Enable การ Pull-Up ของ SCL และ SDA ของบอร์ด ET-MINI DS1307
- ต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport RS232 ของคอมพิวเตอร์ PC

ในการทดสอบให้ต่อสาย RS232 จากขั้วต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport ของเครื่องคอมพิวเตอร์ PC จากนั้นกำหนดค่าการเชื่อมต่อเป็นดังนี้

- Comport กำหนดเป็น Direct Com(n) โดย (n) หมายถึงหมายเลข Comport ที่ต่อใช้งานอยู่ เช่น ถ้าต่อกับ Com1 ก็กำหนดเป็น Direct Com1
- กำหนดค่า Baud rate = 9600 BPS
- กำหนด Data = 8 Bit
- กำหนด Parity = None
- กำหนด Stop Bit = 1 Bit
- กำหนด Flow Control = None

ตัวอย่างโปรแกรม I2C_RTC_PCF8583 (PCF8583.HEX)

เป็นตัวอย่างโปรแกรม การใช้งาน I2C Master โดยต้องต่อใช้งานร่วมกับ โมดูล “ET-MINI PCF8583” โดยการทำงานของโปรแกรมจะเป็นการใช้งานวงจร I2C ของ dsPIC30F2010 ในการติดต่อสื่อสารกับ PCF8583 ซึ่งเป็นไอซี นาฬิกา Real Time Clock โดยโปรแกรมจะทำการตั้งเวลาของ ชั่วโมง นาที วินาที ให้มีค่าเป็น 00:00:00 จากนั้นก็จะวนอ่านค่าเวลาของ ชั่วโมง นาที วินาที มาแสดงผลให้ทราบ โดยใช้พอร์ตสื่อสารอนุกรม RS232 ในการแสดงผล ด้วย อัตรา Baudrate = 9600 โดยการดูผลการทดลองนี้อาจใช้โปรแกรมจำพวก Terminal ต่างๆ เช่น โปรแกรม Hyper Terminal หรือ Procomm ในการแสดงผลการทำงานของโปรแกรมได้ตามต้องการ โดยการเชื่อมต่อสัญญาณระหว่างบอร์ด “ET-dsPIC30F2010” และ “ET-MINI PCF8583” มีดังนี้

- ต่อ RF2(SDA) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SDA ของบอร์ด ET-MINI PCF8583
- ต่อ RF3(SCL) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SCL ของบอร์ด ET-MINI PCF8583
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI PCF8583
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI PCF8583
- เลือก Enable การ Pull-Up ของ SCL และ SDA ของบอร์ด ET-MINI PCF8583
- ต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport RS232 ของคอมพิวเตอร์ PC

ในการทดสอบให้ต่อสาย RS232 จากขั้วต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport ของเครื่องคอมพิวเตอร์ PC จากนั้นกำหนดค่าการเชื่อมต่อเป็นดังนี้

- Comport กำหนดเป็น Direct Com(n) โดย (n) หมายถึงหมายเลข Comport ที่ต่อใช้งานอยู่ เช่น ถ้าต่อกับ Com1 ก็กำหนดเป็น Direct Com1
- กำหนดค่า Baud rate = 9600 BPS
- กำหนด Data = 8 Bit
- กำหนด Parity = None
- กำหนด Stop Bit = 1 Bit
- กำหนด Flow Control = None

ตัวอย่างโปรแกรม I2C_EEP_24LC32 (EEP_24LC32.HEX)

เป็นตัวอย่างโปรแกรม การใช้งาน I2C Master โดยต้องต่อใช้งานร่วมกับ โมดูล “ET-MINI 24XX” โดยการทำงานของโปรแกรมจะเป็นการใช้งานวงจร I2C ของ dsPIC30F2010 ในการติดต่อสื่อสารกับ 24LC32 ซึ่งเป็นไอซี EEPROM แบบ I2C จำนวน 4 ตัว ซึ่งต่อร่วมกันอยู่ในบอร์ดเดียวกัน โดยโปรแกรมจะทำการเขียนค่าคงที่ 55H และ AAH ไปยังตำแหน่งแอดเดรส 0000H และตำแหน่งแอดเดรส 0FFFH ซึ่งเป็นตำแหน่งแอดเดรสเริ่มต้น และตำแหน่งแอดเดรสสุดท้ายของ EEPROM แต่ละตัว พร้อมกับสั่งอ่านค่าจาก EEPROM กลับออกมาเปรียบเทียบกับค่าที่เขียนเข้าไปว่ามีค่าตรงกันหรือไม่ โดยถ้าค่าที่เขียนและอ่านตรงกันก็จะแสดงผลเป็น PASS แต่ถ้าไม่ตรงก็จะแสดงผลเป็น FAIL โดยใช้พอร์ตสื่อสารอนุกรม RS232 ในการแสดงผล ด้วย อัตรา Baudrate = 9600 โดยการดูผลการทดลองนี้อาจใช้โปรแกรมจำพวก Terminal ต่างๆ เช่นโปรแกรม Hyper Terminal หรือ Procomm ในการแสดงผลการทำงานของโปรแกรมได้ตามต้องการ โดยการเชื่อมต่อสัญญาณระหว่างบอร์ด “ET-dsPIC30F2010” และ “ET-MINI 24XX” สำหรับใช้ในการทดลองนี้มีดังนี้

- ต่อ RF2(SDA) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SDA ของบอร์ด ET-MINI 24xx
- ต่อ RF3(SCL) ของบอร์ด ET-dsPIC30F2010 เข้ากับ SCL ของบอร์ด ET-MINI 24xx
- ต่อ +5V ของบอร์ด ET-dsPIC30F2010 เข้ากับ +VCC ของบอร์ด ET-MINI 24xx
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของบอร์ด ET-MINI 24xx
- เลือก Enable การ Pull-Up ของ SCL และ SDA ของบอร์ด ET-MINI 24xx
- ติดตั้งหน่วยความจำเบอร์ 24LC32 ให้กับบอร์ด ET-MINI 24xx ทั้ง 4 ตัว
- ต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport RS232 ของคอมพิวเตอร์ PC

ในการทดสอบให้ต่อสาย RS232 จากขั้วต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport ของเครื่องคอมพิวเตอร์ PC จากนั้นกำหนดค่าการเชื่อมต่อเป็นดังนี้

- Comport กำหนดเป็น Direct Com(n) โดย (n) หมายถึงหมายเลข Comport ที่ต่อใช้งานอยู่ เช่น ถ้าต่อกับ Com1 ก็กำหนดเป็น Direct Com1
- กำหนดค่า Baud rate = 9600 BPS
- กำหนด Data = 8 Bit
- กำหนด Parity = None
- กำหนด Stop Bit = 1 Bit
- กำหนด Flow Control = None

ตัวอย่างโปรแกรม CAPTURE (CAPTURE_PULSE.HEX)

ตัวอย่างนี้เป็นการแสดงให้เห็นการใช้ Capture1 วัดคาบเวลาของสัญญาณ Pulse โดยใช้ Timer2 เป็นตัวนับค่าเวลาให้กับ Capture1 โดยใช้ RD0 เป็นขารับสัญญาณ Input ของ Capture1 โดยการทำงานของโปรแกรมนี้จะแสดงผลค่าการวัดทางพอร์ตสื่อสารอนุกรม RS232 โดยใช้ Baudrate = 9600 BPS โดยจะแสดงผลค่าความกว้างของสัญญาณ Pulse จากขอบขาของ Pulse แรก ถึง ขอบขาของ Pulse ถัดไป เป็นค่าของจำนวน Cycle ของสัญญาณนาฬิกาที่นับได้ โดยโปรแกรมจะกำหนดให้ใช้ Timer2 ทำหน้าที่เป็นตัวนับให้กับ Capture1 โดย Timer2 จะถูกกำหนดให้นับทุกๆคาบเวลาของสัญญาณนาฬิกาของ Bus ซึ่งมีค่า 29.4912 MHz (33.90842 nS) ซึ่ง Capture1 จะถูกกำหนดให้ทำงานทุกๆช่วงขอบขาของ (Falling Edge) ของสัญญาณ Pulse ที่ตรวจจับจากขา RD0 โดยตัวอย่างนี้สามารถใช้วัด Pulse ที่มีค่าความถี่ระหว่าง 450 Hz ถึง 30 KHz ถ้าสูงหรือต่ำกว่านี้จะได้อ่านค่าไม่ถูกต้อง ตัวอย่างเช่น ถ้าป้อนสัญญาณ Pulse ค่าความถี่ 1KHz ให้กับขา RD0 ก็จะได้ผลประมาณ “29491” Cycle ทั้งนี้ก็เนื่องมาจากว่า 1 Cycle มีค่า $1 / 29.4912\text{MHz}$ หรือ 33.90842 nS ซึ่งถ้าสัญญาณ Pulse มีค่าความถี่ 1KHz ก็จะมีค่าคาบเวลาเป็น $1 / 1\text{KHz} = 1\text{ms}$ ซึ่งจะต้องรอให้ Timer นับเป็นจำนวน 29491 ครั้ง จึงจะได้ค่าเวลา 1ms พอดี

- ต่อ RD0 ของบอร์ด ET-dsPIC30F2010 เข้ากับสัญญาณ Pulse ที่ต้องการวัดค่า
- ต่อ GND ของบอร์ด ET-dsPIC30F2010 เข้ากับ GND ของวงจรสร้าง Pulse
- ต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport RS232 ของคอมพิวเตอร์ PC

ในการทดสอบให้ต่อสาย RS232 จากขั้วต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport ของเครื่องคอมพิวเตอร์ PC จากนั้นกำหนดค่าการเชื่อมต่อเป็นดังนี้

- Comport กำหนดเป็น Direct Com(n) โดย (n) หมายถึงหมายเลข Comport ที่ต่อใช้งานอยู่ เช่น ถ้าต่อกับ Com1 ก็กำหนดเป็น Direct Com1
- กำหนดค่า Baud rate = 9600 BPS
- กำหนด Data = 8 Bit
- กำหนด Parity = None
- กำหนด Stop Bit = 1 Bit
- กำหนด Flow Control = None

ตัวอย่างโปรแกรม ADC_SHOW_UART (ADC.HEX)

ตัวอย่างนี้เป็นการแสดงการใช้ ADC ขนาด 10 บิต จำนวน 4 ช่อง วัดค่าแรงดัน 0-5V แล้วแสดงผลเป็นค่าของข้อมูลที่แปลงค่าได้ระหว่าง 0-1023 ค่า โดยใช้ RB0-RB3 เป็น Input ของ ADC0-ADC3 และแสดงผลเป็นค่าข้อมูลตัวเลขให้เห็นทางพอร์ตสื่อสารอนุกรม RS232 โดยใช้ Baudrate = 9600BPS โดยในการทดลองให้ต่อวงจรสำหรับประกอบทดลองดังนี้

- ต่อ RB0 ของบอร์ด ET-dsPIC30F2010 เข้ากับ VOL1 (ตัวต้านทานปรับค่าแรงดัน 0-5V)
- ต่อ RB1 ของบอร์ด ET-dsPIC30F2010 เข้ากับ VOL2 (ตัวต้านทานปรับค่าแรงดัน 0-5V)
- ต่อ RB2 ของบอร์ด ET-dsPIC30F2010 เข้ากับ VOL3 (ตัวต้านทานปรับค่าแรงดัน 0-5V)
- ต่อ RB3 ของบอร์ด ET-dsPIC30F2010 เข้ากับ VOL4 (ตัวต้านทานปรับค่าแรงดัน 0-5V)
- ต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport RS232 ของคอมพิวเตอร์ PC

โดยการดูผลการทดลองให้ใช้โปรแกรมจำพวก Terminal เช่น Hyper Terminal หรือ Procomm เป็นตัวกลางในการสื่อสาร โดยให้กำหนดรูปแบบการสื่อสารเป็นดังนี้

- Comport กำหนดเป็น Direct Com(n) โดย (n) หมายถึงหมายเลข Comport ที่ต่อใช้งานอยู่ เช่น ถ้าต่อกับ Com1 ก็กำหนดเป็น Direct Com1
- กำหนดค่า Baud rate = 9600 BPS
- กำหนด Data = 8 Bit
- กำหนด Parity = None
- กำหนด Stop Bit = 1 Bit
- กำหนด Flow Control = None

ในการทดลองนั้นให้ทำการปรับค่าตัวต้านทานปรับค่าแบบเกือกม้าที่ตำแหน่ง VOL1-VOL4 แล้วดูผลการเปลี่ยนแปลงของค่าผลลัพธ์ ADC แต่ละช่องที่หน้าจอโปรแกรม Terminal

ตัวอย่างโปรแกรม EEPROM (EEPROM_INTERNAL.HEX)

ตัวอย่างนี้เป็นตัวอย่างการอ่านเขียนค่าข้อมูลกับหน่วยความจำ EEPROM ในตัว dsPIC30F2010 ซึ่งมีขนาด 16บิต 512 ตำแหน่ง โดยจะแสดงผลการทำงานของโปรแกรมด้วยพอร์ตสื่อสารอนุกรม RS232 โดยใช้ความเร็ว Baudrate = 9600BPS โดยตัวอย่างนี้จะกำหนดตารางแบบ Array ของค่าคงที่ขนาด 16บิต จำนวน 31 ตำแหน่ง โดยจะกำหนดค่าเป็นรหัส ASCII ของ 'A'-'Z' , 'a'-'z' และ '0'-'9' ไว้จากนั้นจะสั่งให้โปรแกรมทำการเขียนค่าข้อมูลให้กับหน่วยความจำ EEPROM จากตำแหน่งเริ่มต้น 7FFC00H เป็นต้นไป โดย EEPROM จะมีตำแหน่งสิ้นสุดอยู่ที่ 7FFFFFFH ซึ่งมีทั้งหมด 1024 ตำแหน่ง แต่เนื่องจากการอ่านเขียนข้อมูลของ EEPROM นั้นจะต้องกระทำครั้งละ 16 บิต หรือ 1 WORD เสมอ ดังนั้นในการอ้างตำแหน่งแอดเดรสเพื่อติดต่อกับหน่วยความจำของ EEPROM จะต้องอ้างเป็นตำแหน่งของเลขจำนวนที่เป็นคู่เสมอ เช่น เมื่อต้องการเขียนค่าข้อมูล 1234H ให้กับหน่วยความจำ EEPROM ในตำแหน่งแรก ซึ่งก็คือ 7FFC00H จะทำให้ข้อมูลค่า 12H ถูกเก็บในตำแหน่ง 7FFC00H และข้อมูลค่า 34H ถูกเก็บไว้ในตำแหน่ง 7FFC01H ซึ่งไม่ว่าจะอ้างตำแหน่งเป็น 7FFC00H หรือ 7FFC01H ก็จะเป็นตำแหน่งของข้อมูลชุดเดียวกัน เนื่องจากการเข้าถึงข้อมูลเป็นแบบ 16บิต ซึ่งถึงแม้ว่า EEPROM ของ dsPIC30F2010 จะมีขนาด 1KByte แต่ก็ต้องใช้การเข้าถึงข้อมูลแบบ 16บิต เสมอ ดังนั้นจึงอาจกล่าวได้ว่าตำแหน่งการเข้าถึง EEPROM ของ dsPIC30F2010 จะมี 512 ตำแหน่ง โดยแต่ละตำแหน่งใช้เก็บข้อมูลได้ 16บิต(1 WORD) โดยมีตำแหน่งการเข้าถึงตำแหน่งแรกคือ 7FFC00H(7FFC01H) และมีตำแหน่งถัดไปเป็น 7FFC02H(7FFC03H) และมีตำแหน่งสุดท้ายเป็น 7FFFFFFE(7FFFFFFH) ดังนั้นเป็นต้น โดยตัวอย่างโปรแกรมนี้อาจจะสั่งเขียนค่ารหัส ASCII ของ 'A' - 'Z', 'a' - 'z' และ '0'-'9' ให้กับ EEPROM โดยเริ่มต้นที่ตำแหน่ง 7FFC00H เป็นตำแหน่งแรก โดยเก็บข้อมูลเป็นคู่ คือ 'A' และ 'B' จะเก็บไว้ตำแหน่งเดียวกัน โดยให้ 'A' เป็นไบต์สูง และ 'B' เป็นไบต์ต่ำ ซึ่งข้อมูลจะมีทั้งหมด 31 ชุด จากนั้นก็จะสั่งอ่านข้อมูลจาก EEPROM ออกมาแสดงผลทางพอร์ตสื่อสารอนุกรม RS232

- ต่อ RS232 ของบอร์ด ET-dsPIC30F2010 เข้ากับ Comport RS232 ของคอมพิวเตอร์ PC

โดยการดูผลการทดลองของตัวอย่างนี้ให้ใช้โปรแกรมจำพวก Terminal เช่น Hyper Terminal หรือ Procomm เป็นตัวกลางในการสื่อสาร โดยให้กำหนดรูปแบบการสื่อสารเป็นดังนี้

- Comport กำหนดเป็น Direct Com(n) โดย (n) หมายถึงหมายเลข Comport ที่ต่อใช้งานอยู่ เช่น ถ้าต่อกับ Com1 ก็กำหนดเป็น Direct Com1
- กำหนดค่า Baud rate = 9600 BPS
- กำหนด Data = 8 Bit
- กำหนด Parity = None
- กำหนด Stop Bit = 1 Bit
- กำหนด Flow Control = None