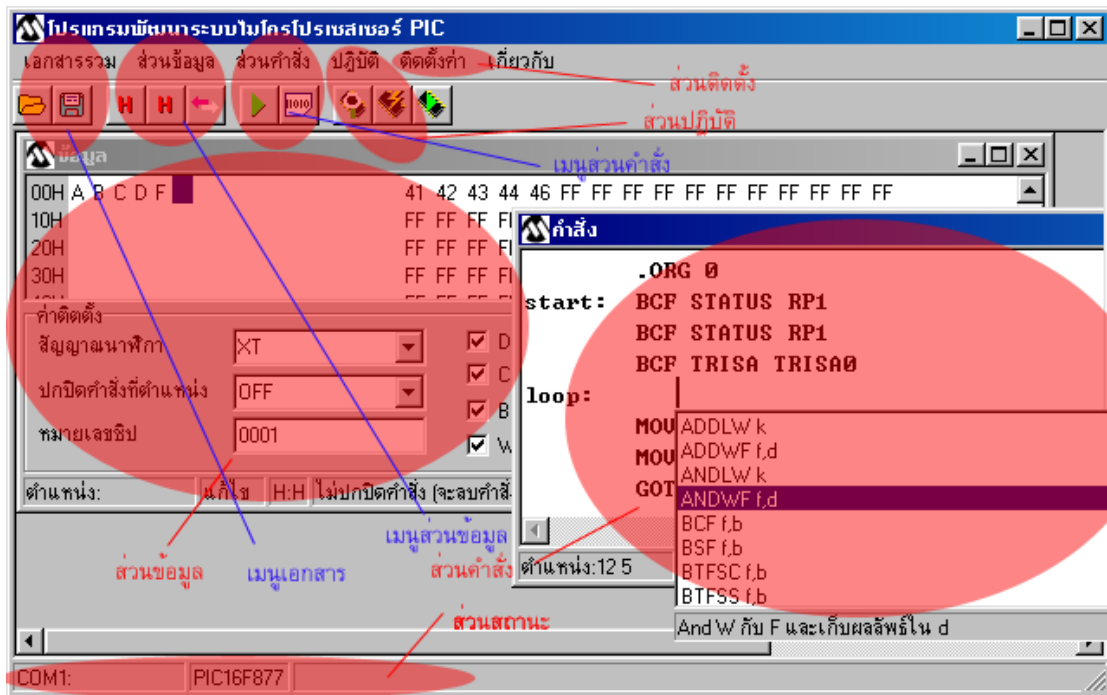


การใช้งานโปรแกรมพัฒนาระบบไมโครโปรเซสเซอร์ PIC

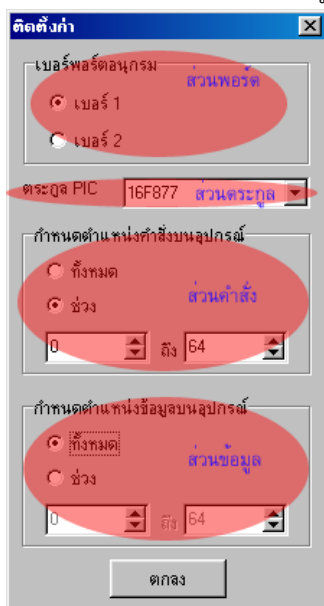


รูปที่ 1 หน้าต่างโปรแกรม

หน้าที่ภายในโปรแกรมจะสามารถแบ่งได้เป็น 6 ส่วนหลักๆ ดังนี้

1. ส่วนติดตั้ง

ส่วนนี้จะมีหน้าที่ติดตั้งข้อกำหนดให้กับโปรแกรม ซึ่งข้อกำหนดเหล่านี้เมื่อกำหนดแล้วจะมีผลต่อการเปิดโปรแกรมในครั้งต่อไปให้อัตโนมัติ ผู้ใช้สามารถกำหนดค่าต่างๆ ได้โดยกดเลือกที่เมนู *ติดตั้งค่า* ตามรูปที่ 1 และหน้าต่าง *ติดตั้งค่า* จะปรากฏดังรูปที่ 2 ซึ่งจะแยกเป็น 4 ส่วนย่อยๆ ได้ดังนี้



รูปที่ 2 หน้าต่างติดตั้งค่า

- ส่วนพอร์ต ใช้เลือกเบอร์พอร์ตอนุกรมที่จะติดต่อกับเครื่องบันทึก PIC
- ส่วนตระกูล ใช้เลือกเบอร์ของ PIC ที่จะทำการบันทึกค่า (burn) ซึ่งจะมีผลกับ ขนาดของข้อมูล (EEPROM) และคำสั่ง (Flash memory) รวมทั้งค่าติดตั้ง (configuration) ของ PIC เป้าหมาย
- ส่วนคำสั่ง เพื่อความเร็วในการบันทึกและอ่านค่าจาก PIC ผู้ใช้สามารถกำหนดตำแหน่งเป็นช่วงที่สนใจได้
- ส่วนข้อมูล เช่นเดียวกับส่วนคำสั่ง โดยจะยินยอมให้เพิ่มหรือลดค่าของตำแหน่งได้ที่ละ 64

2. ส่วนข้อมูล

ส่วนนี้จะเป็นส่วนในการแก้ไข พิจารณาค่าของข้อมูล (EEPROM) รวมทั้งค่าติดตั้ง (configuration) ที่จะบันทึก (burn) ลงบน PIC ในส่วนนี้จะมีเมนู 3 เมนูที่สำคัญดังนี้

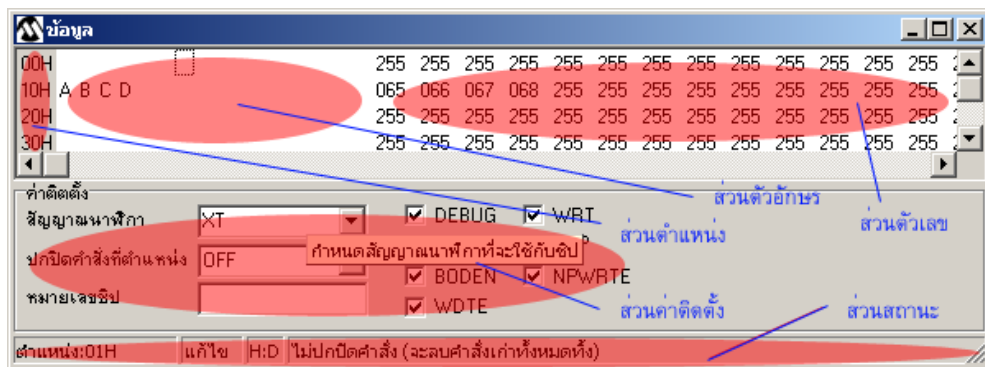
H, **D**, **O**, **B** - ส่วนเลขฐานตำแหน่ง จะทำการเปลี่ยนการแสดงผลค่าตำแหน่งบนหน้าต่าง ข้อมูล ดังรูปที่ 3 ซึ่งจากตัวอย่างจะเป็นการแสดงผลแบบฐาน 16 โดยตัวเลือกจะเป็นได้แบบฐาน 16 (H) , ฐาน 10 (D) , ฐาน 8 (O) , ฐาน 2 (B)

- ส่วนเลขฐานข้อมูล จะทำการเปลี่ยนการแสดงผลค่าข้อมูลบนหน้าต่าง ข้อมูล ดังรูปที่ 3 ซึ่งจากตัวอย่างจะเป็นการแสดงผลแบบฐาน 10 โดยตัวเลือกจะเป็นได้แบบฐาน 16 (H) , ฐาน 10 (D) , ฐาน 8 (O) , ฐาน 2 (B)



- ส่วนสลับหน้าที่ จะทำหน้าที่สลับตัวชี้ (cursor) ระหว่างการป้อนข้อมูลแบบตัวอักษร และ การป้อนข้อมูลแบบตัวเลข

สำหรับส่วนการแก้ไขค่าของข้อมูล จะสามารถแก้ไขได้ตามรูปหน้าต่างที่อยู่ด้านล่าง ซึ่งจะแบ่งเป็นส่วนๆ ได้ดังนี้



รูปที่ 3 หน้าต่างข้อมูล

- ส่วนตำแหน่ง ใช้แสดงตำแหน่งของข้อมูล โดยในแต่ละบรรทัดจะประกอบด้วยข้อมูล 16 ตัว
- ส่วนตัวอักษร ใช้แสดงและแก้ไขข้อมูล โดยจะแสดงเป็นตัวอักษรรหัส ascii ซึ่งเมื่อแก้ไขในส่วนตัวอักษรแล้ว โปรแกรมจะไปแก้ไขข้อมูลในส่วนตัวเลขที่ตำแหน่งเดียวกันให้อัตโนมัติ
- ส่วนตัวเลข ใช้แสดงและแก้ไขข้อมูล โดยจะแสดงเป็นตัวเลขที่มีขนาด 1 byte (8 bits) คือมีค่าอยู่ระหว่าง 0 ถึง 255 ซึ่งเมื่อแก้ไขในส่วนตัวเลขแล้ว โปรแกรมจะไปแก้ไขข้อมูลในส่วนตัวอักษรที่ตำแหน่งเดียวกันให้อัตโนมัติ
- ส่วนค่าติดตั้ง ใช้แสดงและแก้ไขค่าติดตั้ง (configuration) ของ PIC ซึ่งค่าติดตั้งจะขึ้นอยู่กับตระกูลของ PIC โดยโปรแกรมจะจัดการแสดงเฉพาะที่ตระกูลนั้นเลือกได้ให้อัตโนมัติ และจะมีคำอธิบายสำหรับแต่ละค่าเมื่อวาง mouse ค้างไว้ ดังในรูปตัวอย่าง
- ส่วนสถานะ ใช้แสดงสถานะของส่วนข้อมูล ซึ่งประกอบด้วยส่วนย่อยๆ เรียงกันดังรูปที่ 3 ดังนี้

ตำแหน่ง:01H

- ส่วนตำแหน่ง ใช้แสดงตำแหน่งปัจจุบันของตัวชี้ (cursor)

แก้ไข

- ส่วนแจ้งการแก้ไข ถ้าผู้ใช้แก้ไขส่วนใดส่วนหนึ่ง โปรแกรม

จะแจ้งผู้ใช้ว่ามีกรแก้ไขเกิดขึ้นที่ส่วนนี้

H:D

- ส่วนเลขฐาน ใช้แสดง เลขฐานของตำแหน่ง และ เลขฐานของข้อมูล ในปัจจุบันที่ผู้ใช้เลือก ดังในตัวอย่าง ตำแหน่งเป็นฐาน 16 และข้อมูลเป็นฐาน 10 ตามลำดับ

ไม่ปิดคำสั่ง (จะลบคำสั่งทั้งหมดทิ้ง)

- ส่วนคำอธิบาย ใช้แสดงคำอธิบายสำหรับคำติดตั้ง ส่วนสัญญาณนาฬิกา และส่วนปิดคำสั่ง ซึ่งจะอธิบายความหมายของคำที่ใช้เลือก

3. ส่วนคำสั่ง

ส่วนนี้จะเป็นส่วนในการแก้ไข พิจารณาคำสั่ง (Flash memory) ที่จะบันทึก (burn) ลงบน PIC ในส่วนนี้จะมีเมนู 2 เมนูที่สำคัญดังนี้



- ส่วนเข้ารหัสคำสั่ง จะทำหน้าที่เข้ารหัสคำสั่งให้เป็นตัวเลขที่พร้อมจะบันทึก (burn) ลงบน PIC โดยถ้าคำสั่งที่ผู้ใช้กรอกมีข้อผิดพลาด โปรแกรมจะแจ้งข้อผิดพลาดในคำสั่ง ตามรูปที่ 4



- ส่วนสลับหน้าที่ ในหน้าต่าง คำสั่ง จะมีการทำงาน 2 รูปแบบคือ

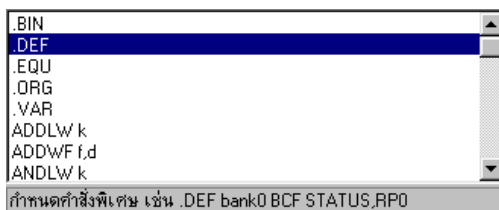
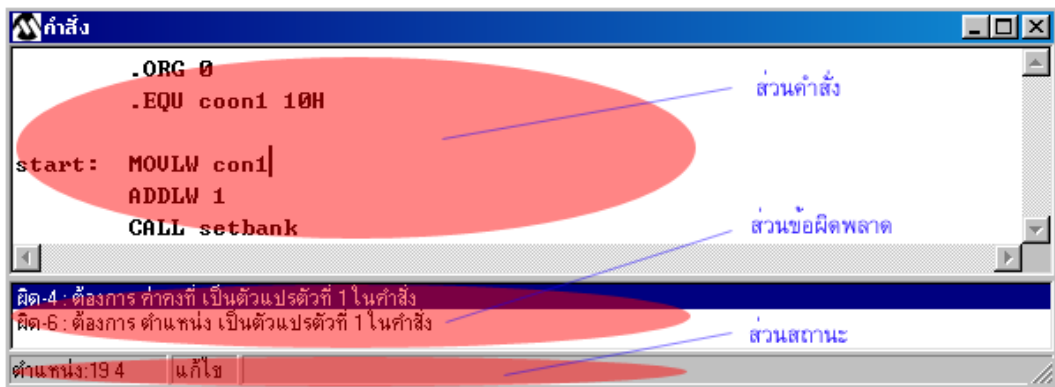
- แบบป้อนคำสั่ง ดังรูปที่ 4

- แบบดูรหัสคำสั่ง ดังรูปที่ 5

ซึ่งเมนูนี้จะทำหน้าที่สลับระหว่างแต่ละแบบ

สำหรับส่วนหน้าต่างคำสั่ง จะอธิบายการทำงานในแต่ละแบบ ดังนี้

- แบบป้อนคำสั่ง ดังรูปที่ 4 ในแบบนี้ จะให้ผู้ใช้ป้อนคำสั่งลงไปและทำการเข้ารหัสคำสั่ง โดยในการป้อนคำสั่ง โปรแกรมนี้ได้อำนวยความสะดวกโดยเพิ่มระบบการเติมเต็มคำสั่ง (code completion) ไว้ เพียงผู้ใช้เคาะแป้นเว้น (space bar) หน้าต่าง เติมเต็ม จะปรากฏขึ้นข้างๆตัวชี้ (cursor) เพื่อให้ผู้ใช้สามารถกดเป็นลูกศรเลือกได้ และเคาะแป้นตกลง (enter) เพื่อให้โปรแกรมกรอกคำสั่งให้อัตโนมัติ



รูปที่ 4 หน้าต่างคำสั่ง แบบป้อนคำสั่ง, หน้าต่างเติมเต็ม

สำหรับส่วนต่างๆ ภายในหน้าต่าง คำสั่ง แบบป้อนคำสั่ง อธิบายได้ดังนี้

- ส่วนคำสั่ง เป็นส่วนที่ให้ผู้ป้อนคำสั่ง ลงไปได้
- ส่วนข้อผิดพลาด เป็นส่วนที่ปรากฏขึ้นเมื่อผู้ใช้ทำการเข้ารหัส (อยู่ในส่วนปฏิบัติ) และเกิดข้อผิดพลาดในคำสั่งที่ผู้ใช้ป้อน ซึ่งผู้ใช้สามารถ double-click ที่ข้อผิดพลาดในส่วนนี้ และโปรแกรมจะเลื่อนส่วนคำสั่งไปที่บรรทัดที่เกิดข้อผิดพลาดนั้น

- ส่วนสถานะ จะแบ่งเป็นส่วนต่างๆ ดังนี้

ตำแหน่ง:1 18

- ส่วนตำแหน่ง ใช้แสดงตำแหน่งปัจจุบันของตัวชี้ (cursor) ซึ่งจะแสดงในรูป ตัวเลขที่บอกถึง เลขบรรทัด และเลขคอลัมน์ ตามลำดับ

แก้ไข

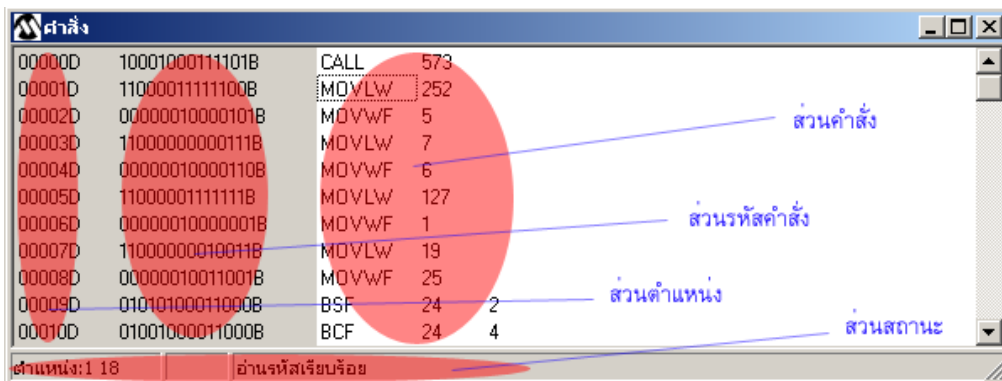
- ส่วนแจ้งการแก้ไข ถ้าผู้ใช้แก้ไขส่วนใดส่วนหนึ่ง โปรแกรมจะแจ้งผู้ใช้งานว่าการแก้ไขเกิดขึ้นที่ส่วนนี้

อ่านรหัสเรียบร้อย

- ส่วนการเข้ารหัสคำสั่ง ใช้บอกผู้ใช้ถึงผลของการเข้ารหัส

- ส่วนเติมเต็ม จะเป็นหน้าต่างที่ปรากฏขึ้นเมื่อ ผู้ใช้เคาะแป้นเว้น (space bar) ในส่วนคำสั่ง และโปรแกรมตี ความในบรรทัดที่เคาะแป้นนั้นว่า มีสิ่งที่สามารถป้อนต่อได้ เช่น คำสั่ง (opcode) , ตัวตั้ง (operand) โดยในหน้าต่าง เติมเต็ม จะมีคำอธิบายของคำสั่งที่เลือก อยู่ด้านล่างของหน้าต่าง

- แบบดูรหัสคำสั่ง ในแบบนี้จะใช้ในการดูรหัสคำสั่งที่ได้หลังจากทำการเข้ารหัสคำสั่ง



รูปที่ 5 หน้าต่าง คำสั่ง แบบดูรหัสคำสั่ง

ส่วนประกอบทั้งหมดในหน้าต่าง คำสั่ง แบบดูรหัสคำสั่ง มีดังนี้

- ส่วนตำแหน่ง จะเป็นตัวเลขที่แสดงถึงตำแหน่งของรหัสคำสั่ง โดยจะแสดงเป็นฐาน 10
- ส่วนรหัสคำสั่ง จะเป็นตัวเลขที่แสดงถึงค่าของรหัสคำสั่งนั้นๆ โดยจะแสดงเป็นฐาน 2
- ส่วนคำสั่ง เพื่อความสะดวกกับผู้ใช้ในการไล่อรหัสคำสั่ง ก็จะมีการถอดรหัสคำสั่ง (disassembly) เตรียมไว้ ซึ่งตัวเลขในส่วนนี้จะเป็นฐาน 10 ทั้งหมด

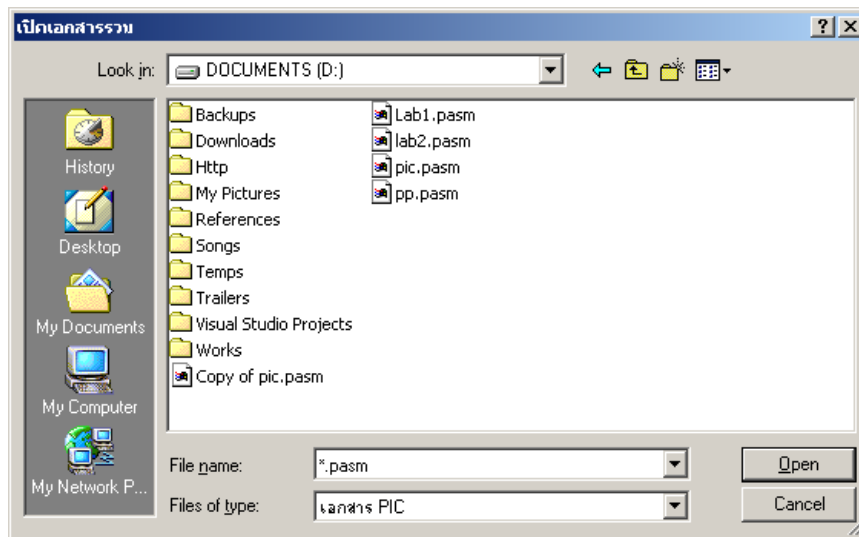
4. ส่วนเอกสาร

ส่วนนี้จะมีหน้าที่ ในการเปิด และบันทึกค่าของข้อมูล (EEPROM) คำสั่ง (Flash memory) และค่าติดตั้ง (configuration) ลงเอกสาร (file) PASM (PIC Assembly) โดยส่วนนี้จะมีเมนู 2 เมนูที่สำคัญดังนี้

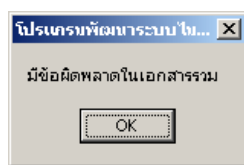


- ส่วนเปิดเอกสารรวม จะเป็นการเปิดเอกสารเก่า โดยจะมีหน้าต่าง เปิดเอกสารรวม ปรากฏขึ้นเพื่อให้ผู้ใช้เลือกเอกสารที่ต้องการเปิด ซึ่งภายในเอกสารรวมจะบันทึกตระกูลของ PIC ที่ใช้กับเอกสารนั้นไว้ ถ้าเอกสารรวมมีข้อผิดพลาด ก็จะมีหน้าต่าง แจ้งข้อผิดพลาดในเอกสารรวม ปรากฏขึ้น นอก

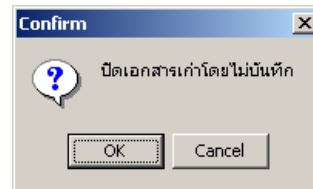
จากนี้ ถ้าผู้ใช้แก้ไขค่าใดๆ และยังไม่ได้บันทึกเอกสาร จะมีหน้าต่าง **แจ้งปิดเอกสารเก่า** เพื่อเตือนผู้ใช้ที่ยังไม่ได้บันทึกการแก้ไข



รูปที่ 6 หน้าต่างเปิดเอกสารรวม



รูปที่ 7 หน้าต่างแจ้งข้อผิดพลาดในเอกสารรวม



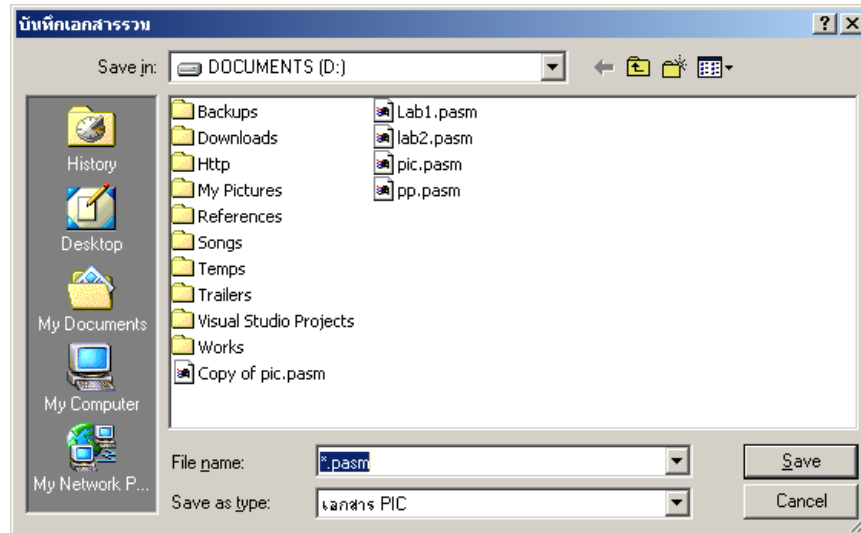
รูปที่ 8 หน้าต่างแจ้งปิดเอกสารเก่า



- ส่วนบันทึกเอกสารรวม จะเป็นการบันทึกค่าลงเอกสาร โดยจะเป็นการบันทึกทับเอกสารเก่าที่ผู้ใช้เปิดขึ้น ถ้าผู้ใช้ต้องการบันทึกเอกสารในชื่อใหม่ ให้เลือก **บันทึกเอกสารรวมใหม่** ที่เมนู **เอกสารรวม** ดังรูปที่ 7 และจะมีหน้าต่าง **บันทึกเอกสารรวม** ปรากฏขึ้นเพื่อให้ผู้ใช้เลือกชื่อที่จะบันทึก

เอกสารรวม	ส่วนข้อมูล	ส่วนคำสั่ง
เปิดเอกสารรวม		Ctrl+O
บันทึกเอกสารรวม		Ctrl+S
บันทึกเอกสารรวมใหม่		

รูปที่ 9 เมนูบันทึกเอกสารรวมใหม่



รูปที่ 10 หน้าต่างบันทึกเอกสารรวม

5. ส่วนปฏิบัติ

ส่วนนี้จะมีหน้าที่ในการติดต่อกับเครื่องบันทึก PIC ซึ่งผู้ใช้สามารถสั่งงานได้โดยกดที่เมนูตามรูปที่ 1 และส่วนนี้ประกอบด้วยหน้าที่ 3 อย่างดังนี้



- ส่วนอ่านข้อมูลและคำสั่งจากอุปกรณ์ จะทำการอ่านข้อมูล (EEPROM) และคำสั่ง (Flash memory) รวมทั้งค่าติดตั้ง (configuration) จาก PIC เป้าหมาย

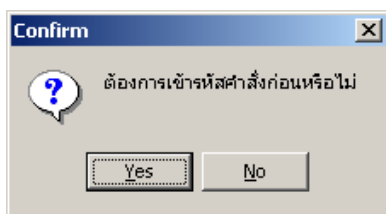


- ส่วนนำข้อมูลและคำสั่งลงอุปกรณ์ จะทำการบันทึก (burn) ข้อมูล (EEPROM) และคำสั่ง (Flash memory) รวมทั้งค่าติดตั้ง (configuration) ลงบน PIC เป้าหมาย โดยก่อนจะทำการบันทึก โปรแกรมจะถามถึงการเข้ารหัสคำสั่งตั้งหน้าต่าง สอบถามการเข้ารหัส ในรูปที่ 11 ถ้าไม่ทำการเข้ารหัส โปรแกรมจะนำคำสั่งคำสั่งเก่ามาบันทึก และขั้นตอนในการบันทึก โปรแกรมจะทำการลบทั้งค่าข้อมูลและค่าคำสั่งเก่าบน PIC เป้าหมายออกก่อนที่จะบันทึกค่าใหม่ลงไป

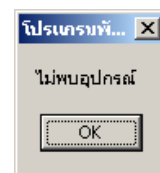


- ส่วนสั่งอุปกรณ์ทำงาน จะสั่งให้ PIC เป้าหมายเริ่มทำงาน เพื่อให้ผู้ใช้ตรวจสอบความถูกต้องในการทำงาน มีผลเช่นเดียวกับปุ่ม Reset บน PIC เป้าหมาย

ถ้าผู้ใช้ไม่ได้ต่อเครื่องบันทึกกับพอร์ตอนุกรมตามเบอร์ที่กำหนดไว้จะขึ้นหน้าต่างตามรูปที่ 12



รูปที่ 11 หน้าต่างสอบถามการเข้ารหัส



รูปที่ 12 หน้าต่างแจ้งไม่พบอุปกรณ์

6. ส่วนสถานะ

ส่วนนี้จะมีหน้าที่ในการแสดงสถานะของโปรแกรม ซึ่งจะแยกเป็น 3 สถานะเรียงจากซ้ายไปขวา ตามรูปที่ 1 ดังนี้

COM1: ทำงาน

- ส่วนพอร์ต จะใช้บอกสถานะของเบอ์พอร์ตที่จะติดต่อกับ
เครื่องบันทึกซึ่งผู้ใช้เป็นผู้เลือก และบอกสถานะการติดต่อกับเครื่องบันทึกว่าทำงานอยู่หรือไม่

PIC16F877

- ส่วนตระกูล จะใช้บอกสถานะของตระกูล PIC ที่ผู้ใช้เลือก

อ่านข้อมูล 0 จาก (0 ถึง 640)

- ส่วนคำสั่ง จะใช้บอกสถานะของคำสั่งที่โปรแกรมส่งไปให้
เครื่องบันทึก เช่น สั่งอ่าน , เขียน , สั่ง PIC ทำงาน

การเขียนภาษา Assembly บนโปรแกรมพัฒนาระบบไมโครโปรเซสเซอร์ PIC

ภาษา Assembly ที่ใช้ภายในโปรแกรม จะเป็นภาษาที่ง่าย และไม่มีรูปแบบมากนัก โดยจะสามารถจัดหมวดหมู่
ของคำสั่งได้ดังนี้

1. directive เป็นคำสั่งที่ขึ้นต้นด้วย . เป็นคำสั่งที่ไม่ได้ถูกเข้ารหัส แต่เป็นคำสั่งพิเศษที่ช่วยในการเข้ารหัส
ให้กับโปรแกรม ซึ่งมีอยู่ ดังนี้

.org เป็นคำสั่งที่ใช้กำหนดตำแหน่งที่จะเริ่มเข้ารหัส โดยถ้าไม่กำหนด จะเริ่มที่ตำแหน่งที่ 0 เสมอ

.bin เป็นคำสั่งที่ใช้ในการกำหนดรหัสคำสั่งเอง

.equ เป็นคำสั่งที่ใช้ในการกำหนดชื่อค่าคงที่

2. opcode เป็นคำสั่งปกติที่ใช้ในการเข้ารหัส จำนวน 35 คำสั่งตามมาตรฐานคำสั่งของ PIC ซึ่งผู้อ่าน
สามารถเปิดอ่านในคู่มือของ PIC ได้ ซึ่งมีรูปแบบดังนี้

<opcode> <operand1> <operand2>

หรือ

<opcode> <operand1> , <operand2>

สำหรับค่าคงที่ในตัวตั้ง (operand) สามารถเขียนในรูปแบบฐาน 16,10,8 และ 2 ได้ เพียงใส่ตัวอักษร

H,D,O และ B ตามหลังตัวเลข เพื่อระบุเลขฐาน เช่น MOVLW FCH โดยถ้าผู้ใช้ไม่ระบุเลขฐาน

โปรแกรมจะถือว่าเป็นเลขฐาน 10

3. label เป็นการประกาศชื่อตำแหน่งเพื่อความง่ายในการเขียนโปรแกรม ซึ่งมีรูปแบบดังนี้

<ชื่อ label>:

โดย label จะต้องถูกประกาศไว้ก่อนหน้า opcode หรือไม่มี opcode ตามมาก็ได้เช่น

loop: หรือ loop: MOVLW 10B แต่ label ไม่สามารถประกาศไว้หน้า directive ได้ ต้องขึ้นบรรทัดใหม่

4. remark ใช้ในการเขียนคำอธิบายของคำสั่ง ซึ่งมีรูปแบบดังนี้

;<คำอธิบาย>

ซึ่งการเขียน remark สามารถเขียน ด้านหลังต่อท้าย directive , opcode หรือ label ก็ได้ เช่น

loop: ; beginning of main task หรือ .EQU con1 20H ; counter register