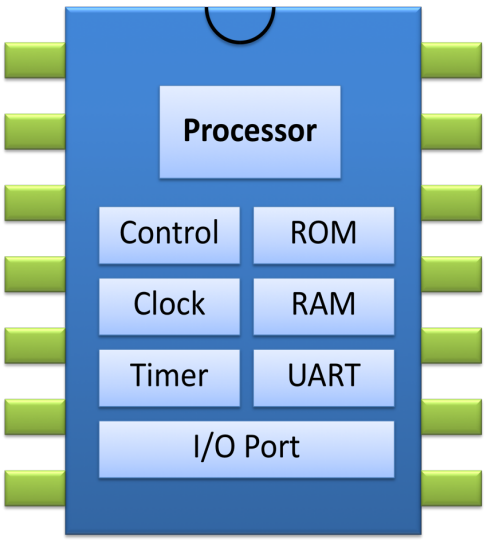
บทที่ 1
แนะนำไมโครคอนโทรลเลอร์

© 2022 โดย จารุต บุศราทิจ และกอบกิจ เติมผาติ

1. บทนำ

ไมโครคอนโทรลเลอร์ที่เขียนย่อเป็นภาษาอังกฤษว่า MCU (Microcontroller Unit) หรือ uC คือคอมพิวเตอร์ขนาดเล็กที่มีหน่วยทำงานทั้งที่เป็นส่วนประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) นาฬิกาหรือคล็อก (Clock) ตัวตั้งเวลาหรือไทเมอร์ (Timer) ส่วนนำเข้า/ส่งออกหรือไอ/โอพอร์ต (Port I/O) และหน่วยความจำ (Memory) อยู่ในชิพเพียงตัวเดียว ดังรูปที่ 1.1 จึงนำไปประยุกต์ใช้กับงานควบคุมอุปกรณ์ต่างๆ โดยบิร์ดควบคุมมีขนาดเล็กและมีราคาที่ถูกกว่าการใช้ระบบที่เป็นคอมพิวเตอร์ที่ใช้งานทั่วไป เพราะคอมพิวเตอร์ที่ใช้งานทั่วไปนั้นจะแยกส่วนการทำงานต่างๆ ไม่ว่าจะเป็นซีพียู หน่วยความจำ หรือไอ/โอพอร์ต ออกจากกัน เพื่อความยืดหยุ่นในการปรับเปลี่ยนให้เหมาะสมกับความต้องการ เช่น ถ้าผู้ใช้ต้องการใช้คอมพิวเตอร์ที่มีศักยภาพในการประมวลผลภาพจะต้องซื้อเฉพาะการ์ดแสดงผลที่มีความสามารถสูงมาใช้ หรือถ้าต้องการหน่วยใช้หน่วยความจำปริมาณมากสำหรับประมวลผลข้อมูลขนาดใหญ่ก็สามารถเพิ่มจำนวนหน่วยความจำชนิดแรม (RAM: Random Access Memory) ให้มากขึ้นตามความต้องการ เป็นต้น



รูปที่ 1.1 โครงสร้างหลักของไมโครคอนโทรลเลอร์

ด้วยการที่ระบบคอมพิวเตอร์โดยทั่วไปนั้นมีความยืดหยุ่นต่อการปรับเปลี่ยนให้เหมาะสมกับความต้องการของผู้ใช้งานจึงทำให้ต้นทุนในการผลิตนั้นมีราคาสูงตามขึ้นไป เนื่องจากจะต้องมีการออกแบบส่วนต่าง ๆ ให้สามารถนำมาประกอบเข้าชุดการทำงานด้วยกันได้ และต้องผลิตอุปกรณ์ให้มีความหลากหลายตรงกับความต้องการของผู้ใช้ที่แตกต่างกัน แต่ในขณะที่การใช้กับงานควบคุมนั้นต้องการระบบคอมพิวเตอร์ที่แตกต่างกัน เช่น ไม่ต้องมีฮาร์ดดิสก์ ไม่ต้องใช้การ์ดจอ ไม่ต้องการจอแสดงผล ไม่ต้องมีเมาส์ ไม่ต้องมีแป้นพิมพ์หรือคีย์บอร์ด ไม่ต้องการหน่วยความจำที่มีความจุเป็นระดับจิกะไบต์ เป็นต้น

จากเหตุผลข้างต้นทำให้ผู้ผลิตชิพหลายบริษัทไม่ว่าจะเป็นอินเทล (Intel) ไมโครชิพ (Microchip) ไซล็อก (Zilog) แอตเมล (Atmel) หรือโมโตโรลา (Motorola) เป็นต้น ได้ผลิตไมโครคอนโทรลเลอร์ขึ้นมา เพื่อใช้กับงานควบคุม โดยผนวกคุณสมบัติพื้นฐานเอาไว้ในตัวชิพเพียงตัวเดียว ทำให้ระบบมีขนาดเล็ก ง่ายต่อการนำไปประยุกต์ใช้งานโดยการต่อไอ/โอพอร์ตของชิพไมโครคอนโทรลเลอร์เข้ากับวงจรทำงานภายนอก และทำให้ต้นทุนในการผลิตถูกลง

จากตัวอย่างการประยุกต์ระบบไมโครคอนโทรลเลอร์ไปใช้งานกับระบบสมองกลฝังตัว (Embedded System) ได้แก่ การตรวจจับอุณหภูมิเพื่อแสดงค่าอุณหภูมิ (รูปที่ 1.2) การตรวจจับอุณหภูมิเพื่อเปิดหรือปิดพัดลม (รูปที่ 1.3) หรือเครื่องคำนวณ (ดังรูปที่ 1.4) จะพบว่าผู้พัฒนาใช้ระบบไมโครคอนโทรลเลอร์ที่เหมือนกัน แต่แตกต่างกันเพียงอุปกรณ์รอบข้าง ด้วยเหตุนี้ ระบบไมโครคอนโทรลเลอร์จึงเหมาะสมต่อการประยุกต์ใช้งานควบคุมที่มีขนาดเล็กที่หลากหลาย เนื่องจากต้นทุนในการผลิตถูกกว่าการใช้คอมพิวเตอร์ทั่วไป ประหยัดพลังงานมากกว่า สามารถทำงานได้ทันทีเมื่อจ่ายแรงดันเข้าสู่ระบบ และสามารถเขียนโปรแกรมควบคุมการทำงานของหน่วยประมวลผลได้โดยตรงไม่จำเป็นต้องผ่านระบบปฏิบัติการที่ซับซ้อน

รูปที่ 1.2 ผังระบบแสดงผลการตรวจจับอุณหภูมิ

รูปที่ 1.3 ผังระบบควบคุมการเปิด/ปิดพัดลมด้วยการตรวจจับอุณหภูมิ

รูปที่ 1.4 ผังระบบเครื่องคำนวณ

2. ไมโครคอนโทรลเลอร์ PIC16F877

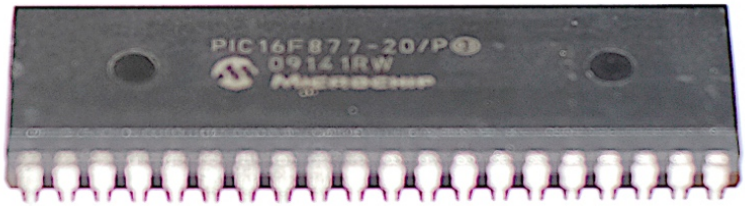
ไมโครคอนโทรลเลอร์ PIC16F877 (รูปที่ 1.5) เป็นไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัทไมโครชิพ เป็นไมโครคอนโทรลเลอร์ที่ถูกออกแบบภายใต้แนวคิดของสถาปัตยกรรมที่เรียกว่าริสก์ (RISC: Reduce Instruction Set Cycle) ซึ่งทำให้ประมวลผลคำสั่งแต่ละคำสั่งใช้สัญญาณนาฬิกาเพียง 1 ถึง 2 วงรอบการทำงานหรือต่อ 1 ถึง 2 ไซเคิล (Cycle) เท่านั้น คุณสมบัติของ PIC16F877 มีดังนี้

1. มีคำสั่งให้ใช้งาน 35 คำสั่ง โดยชุดคำสั่งมีขนาด 16 บิต และอ้างอิงตำแหน่งข้อมูลได้ 8 บิต
2. คำสั่งหนึ่งๆใช้เวลาทำงาน 1 ถึง 2 ไซเคิล
3. ทำงานได้สูงสุดที่ 20 เมกะเฮิรตซ์ (MHz)
4. รองรับการใช้สัญญาณนาฬิกาแหล่งที่ 2 ที่เป็นความถี่ 32 kHz
5. ทำงานแบบไปป์ไลน์ (Pipe-line) ทำให้ ณ เวลาหนึ่งทำงาน 2 อย่างพร้อมกันได้

6. หน่วยความจำโปรแกรมเป็นแบบแฟลช (Flash) มีขนาด 8 กิโลเวิร์ด (KWord) โดยแต่ละเวิร์ดมีขนาด 14 บิต
7. มีแรม ขนาด 368 ไบต์สำหรับใช้งาน
8. มี EEPROM ขนาด 256 ไบต์สำหรับเก็บข้อมูลแบบถาวร
9. ตอบสนองกับอินเตอร์รัพต์ (Interrupt) ได้ทั้งหมด 14 แหล่ง
10. มีหน่วยความจำแบบสแตก (Stack) ให้ใช้ได้สูงสุด 8 ระดับ
11. รองรับการรีเซ็ตตัวเองเมื่อมีแรงดันไฟฟ้ากระแสตรงจ่ายเข้าชิพ หรือ POR (Power-on-Reset)
12. รองรับการตั้งเวลาทำงานหลังเกิด POR หรือ BOR ที่เรียกว่าการทำ Power-up Timer (PWRT)
13. รองรับการจับเวลาเริ่มต้นของตัวให้สัญญาณนาฬิกาเพื่อทำการรีเซ็ตตัวเอง จนกว่าสัญญาณณาฬิกาจะทำงานเสถียรซึ่งเรียกว่าการทำ Oscillator Start-up Timer (OST)
14. มีตัวตั้งเวลาแบบวอทช์ด็อก (Watchdog Timer) ทำให้สามารถส่งสัญญาณรีเซ็ต (Reset) ไปยังไมโครคอนโทรลเลอร์เมื่อได้มีการประมวลผลนานเกินกว่าเวลาที่กำหนด เพื่อป้องกันการเกิดปัญหาเกี่ยวกับโปรแกรมทำงานผิดพลาดหรือแฮงค์ (Hang)
15. มีระบบป้องกันโค้ด (Code Protection) เพื่อป้องการถูกคัดลอก
16. มีโหมดประหยัดพลังงาน
17. สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้คริสตัล (XTAL) หรือวงจรอาร์ซี (RC)
18. โปรแกรมด้วยไฟแรงดันกระแสตรงขนาด 5 โวลต์
19. โปรแกรมตัวชิพด้วยวิธีอินเซอร์กิตซีเรียล (In-Circuit Serial Programming)
20. ทำงานที่ไฟเลี้ยงแรงดันกระแสตรงตั้งแต่ 2 โวลต์ ถึง 5.5 โวลต์
21. รับกระแสหรือทำเคอเร็นต์ซิงค์ (Current Sink) และผู้ให้กระแสหรือทำเคอเร็นต์ซอร์ส (Current Source) อยู่ที่ 25 มิลลิแอมป์ (mA)
22. มีตัวตั้งเวลา/ตัวนับ (Timer/Counter) 3 ชุด
23. มีโมดูลซีซีพี (CCP) หรือ แคปเจอร์/คอมแพร์และพีดับบลิวเอ็ม (Capture/ Compare/ PWM) สำหรับจับสัญญาณ เปรียบเทียบสัญญาณ และส่งพัลซ์ที่กำหนดความกว้างได้ จำนวน 2 ชุด
    - การตรวจจับสัญญาณนำเข้าทำงานแบบ 16 บิต โดยใช้เวลา 6.25ns ในการทำงาน
    - การเปรียบเทียบสามารถเปรียบเทียบสัญญาณแบบ 16 บิตด้วยความเร็ว 100ns

- การนำออกสัญญาณ PWM กำหนดให้มีค่าของความละเอียดตั้งแต่ 1 ถึง 10 บิต โดยที่ 8 บิตทำงานได้ได้ความถี่ 156kHz และ 10 บิตที่ความถี่ 39kHz โดยประมาณ
    - รองรับการควบคุมมอเตอร์ด้วยการนำออกสัญญาณ PWM ในแบบ 1, 2 หรือ 4 ช่องสัญญาณในเวลาเดียวกัน
    - กำหนดขั้ว (polarity) ของ PWM ได้
    - ตั้งเวลาการสิ้นสุดการทำงานของ PWM ได้

24. มีวงจรแปลงสัญญาแอนะล็อกมาเป็นดิจิทัล (A-TO-D Converter หรือ ADC)
    - มีวงจรแปลงสัญญาณแอนาล็อกเป็นดิจิทัลหรือ ADC (Analog-to-Digital) ที่มีความละเอียดของการแปลงค่าเป็น 10 บิต จำนวน 8 ช่องสัญญาณ พร้อมทั้งรองรับการแปลงค่าทั้งที่อยู่ในโหมดประหยัดพลังงาน และใช้พร้อมกันได้ทั้ง 8 ขา
    - มีโมดูลเปรียบเทียบค่าแอนาล็อกที่โปรแกรมการทำงานแบบมัลติเพล็ก
    - มีโมดูลกำหนดแรงดันอ้างอิง
    - มีโมดูลตรวจสอบแรงดันต่ำ (LVD หรือ Low-Voltage Detection) ในแบบการขัดจังหวะ
    - สามารถสั่งให้ทำการรีเซ็ตตัวเองเมื่อเกิด LVD หรือเรียกว่าการทำ Brown-out Reset (BOR)
25. มีระบบยูเอสอาร์ต (USART) สำหรับต่อกับการสื่อสารแบบอาร์เอส 232 (RS232) พร้อมทั้งรองรับการทำงานแบบมีการขัดจังหวะ และมีระบบตรวจระดับไปเลี้ยง (Brown-out Reset)
26. มีพอร์ตไอ/โอทั้งหมด 5 พอร์ต
27. รองรับ SPI แบบ 3 สายสัญญาณ และ I^2C ในโหมด Master และ Slave
28. สามารถโปรแกรมได้ในตัวเองผ่านทาง ICSP (In-Circuit Serial Programming) ด้วยขา 2 ขา

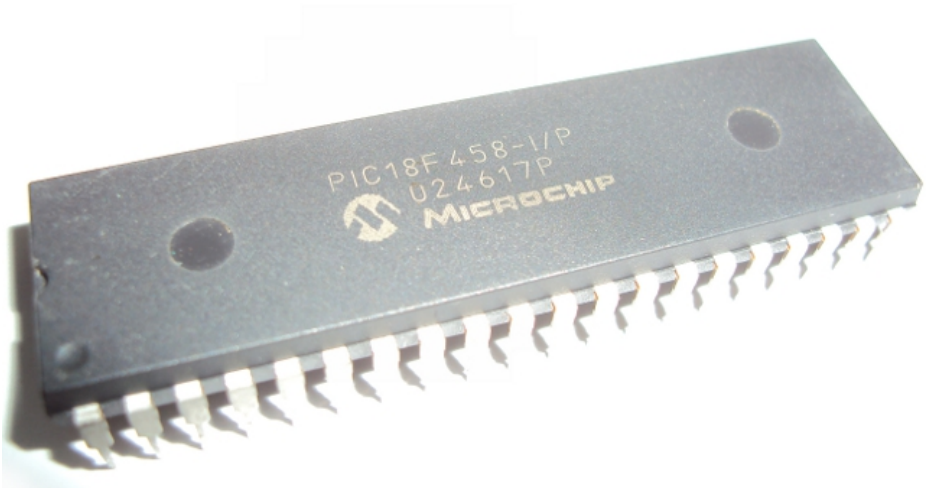


รูปที่ 1.5 ชิพ PIC16F877

3. ไมโครคอนโทรลเลอร์ PIC18F458

ไมโครคอนโทรลเลอร์ PIC18F458 (รูปที่ 1.6) เป็นไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัทไมโครชิพเหมือนกับ PIC16F877 แต่มีการเพิ่มคุณสมบัติพิเศษเข้าไป คือ

1. มีหน่วยความจำโปรแกรมแบบแฟลชขนาด 32 กิโลไบต์
2. ทำงานได้สูงสุดที่ 40 เมกะเฮิรตซ์ โดยใช้ตัวให้สัญญาณ 4MHz ถึง 10MHz เป็นสัญญาณนำเข้าผ่านตัววงจร PLL (Pulse Lock Loop)
3. มีหน่วยความจำแรมขนาด 1,536 ไบต์
4. มีตัวตั้งเวลาและตัวนับ 4 ชุด
   - โมดูล Timer0 ทำงานเป็นตัวนับหรือตัวตั้งเวลาแบบ 8/16บิต พร้อมทั้งสามารถตั้งค่า prescaler ได้ที่ความละเอียด 8 บิต
   - โมดูล Timer1 และ Timer3 ทำงานเป็นตัวนับหรือตัวตั้งเวลาแบบ 16 บิต
   - โมดูล Timer2 ทำงานเป็นตัวตั้งเวลาหรือตัวนับแบบ 8 บิตด้วยเรจิสเตอร์ขนาด 8 บิต และใช้เป็นตัวหลักของการทำ PWM (Pulse Width Modulate)
   - รองรับการใช้ตัวให้สัญญาณนาฬิกาที่ 2 สำหรับ Timer1 และ Timer3 แยกจากตัวสร้างสัญญาณยาฬิกาหลักที่ส่งให้กับไมโครคอนโทรลเลอร์
5. รองรับการขัดจังหวะ (Interrupt) ได้หลายระดับ แบบสามารถกำหนดลำดับความสำคัญได้
6. มีวงจรการคูณระหว่างข้อมูล 8 บิต 2 ชุด ใน 1 วงรอบการทำงาน
7. รองรับการสื่อสารระบบบัสแบบ CAN
   - ทำงานกับมาตรฐาน ISO CAN
   - อัตราการส่งข้อมูลอยู่ที่ 1Mbps
   - เข้ากันได้กับ CAN รุ่น 2.0B

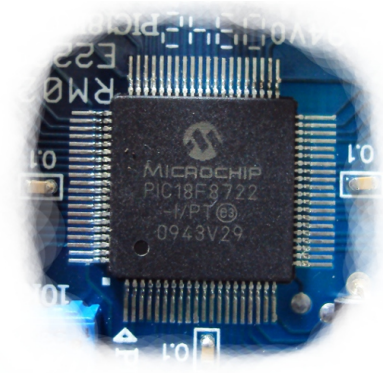


รูปที่ 1.6 ชิพ PIC18F458

4. ไมโครคอนโทรลเลอร์ PIC18F8722

ไมโครคอนโทรลเลอร์ PIC18F8722 เป็นไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัทไมโครชิพ มีตัวถังแบบ TQFP ชนิด 80 ขา (รูปที่ 1.7) ซึ่งมีคุณสมบัติดังต่อไปนี้

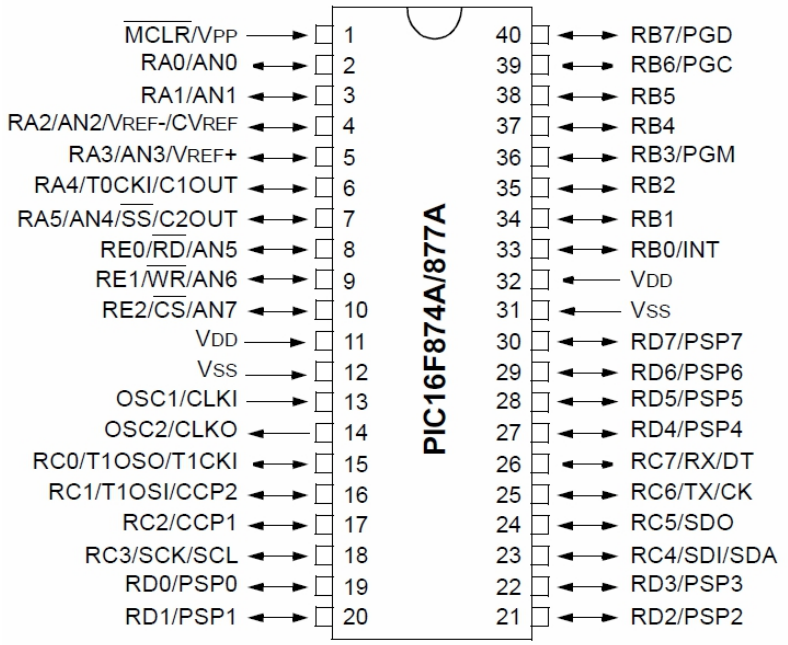
1. มีคำสั่งให้ใช้งาน 75 คำสั่ง และคำสั่งเพิ่มเติมสำหรับเปิดการทำงานอีก 83 คำสั่ง
2. มีหน่วยความจำโปรแกรมแบบแฟลชขนาด 128 กิโลไบต์
3. ทำงานได้สูงสุดที่ 40 เมกะเฮิรตซ์
4. มีหน่วยความจำแรมขนาด 3,936 ไบต์
5. มี EEPROM ขนาด 1,024 ไบต์สำหรับเก็บข้อมูลแบบถาวร
6. ตอบสนองกับอินเตอร์รัพต์ได้ทั้งหมด 29 แหล่ง
7. มีพอร์ตไอ/โอให้ใช้งาน 9 พอร์ต ได้แก่
   - พอร์ต A
   - พอร์ต B
   - พอร์ต C
   - พอร์ต D
   - พอร์ต E
   - พอร์ต F
   - พอร์ต G
   - พอร์ต H
   - พอร์ต J
8. มีไทเมอร์ 5 ตัว
9. มีโมดูลด้านแคปเจอร์/คอมแพร์และพีดับบลิวเอ็ม 2 ชุด
10. มีโมดูลแคปเจอร์/คอมแพร์และพีดับบลิวเอ็มเพิ่มเติม 3 ชุด
11. มีโมดูลสื่อสารอนุกรม 2 ชุด
12. มีวงจรแปลงสัญญาแอนาล็อกมาเป็นดิจิทัลที่มีความละเอียด 10 บิต จำนวน 16 ช่อง



รูปที่ 1.7 ชิพ PIC18F8722

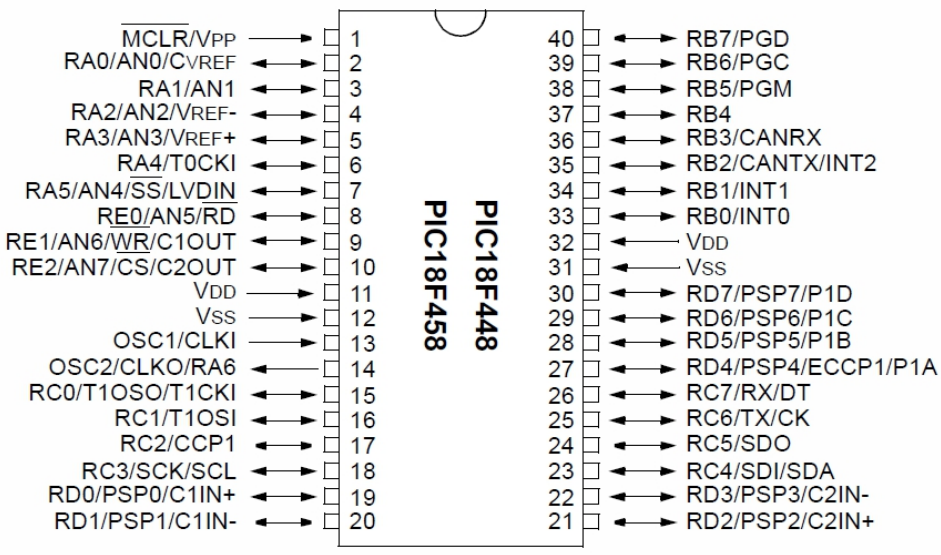
5. การจัดวางขาของชิพและวงจรพื้นฐาน

ชิพ PIC16F877 และ PIC18F458 มีตัวถังแบบดิพ (DIP) จำนวน 40 ขา และมีการจัดวางขาดังรูปที่ 1.8 และ 1.9



รูปที่ 1.8 การวางขาของ PIC16F877

ที่มา : p.1 ของ PIC16F87X Datasheet



รูปที่ 1.9 การวางขาของ PIC18F458

ที่มา : p.4 ของ PIC18F458 Datasheet

หน้าที่ของขาของชิพไมโครคอนโทรลเลอร์เบอร์ PIC16F877 และ PIC18F458 เป็นดังนี้

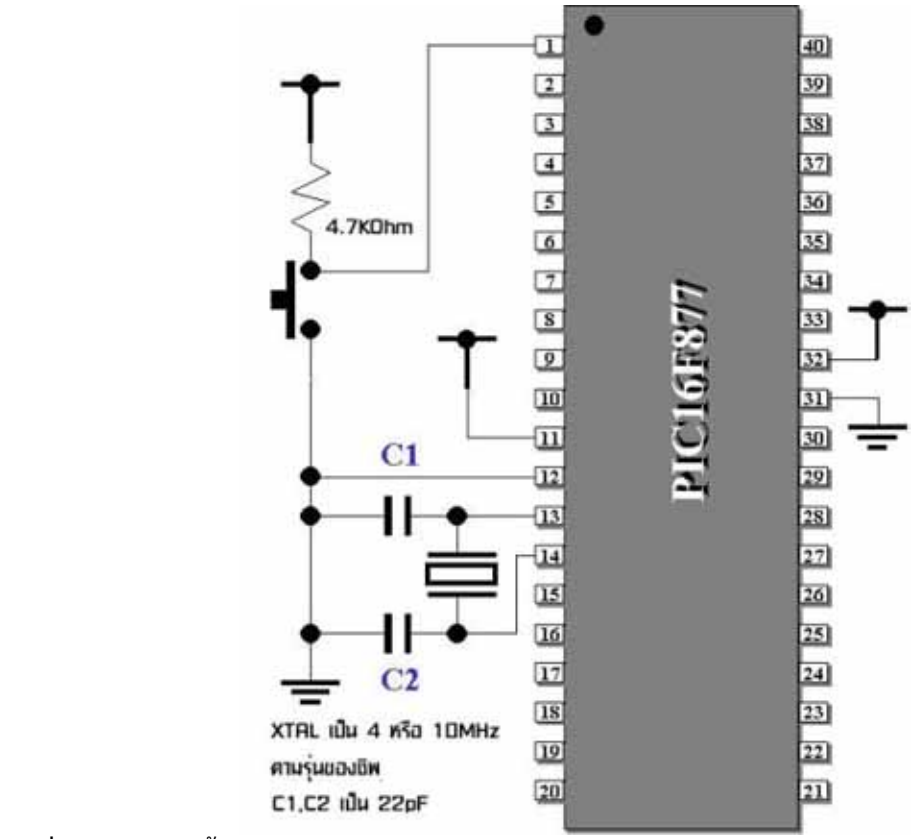
- **RA0/AN0** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 0 ของพอร์ตเอ (RA0) หรือใช้เป็นขาสำหรับรับสัญญาณแบบแอนาล็อกช่องที่ 0 (AN0)
- **RA1/AN1** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 1 ของพอร์ตเอ (RA1) หรือใช้เป็นขาสำหรับรับสัญญาณแบบแอนาล็อกช่องที่ 1 (AN1)
- **RA2/AN2/VREF-** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 2 ของพอร์ตเอ (RA2) ใช้เป็นขาสำหรับรับสัญญาณแบบแอนาล็อกช่องที่ 2 (AN2) หรือเป็นขาสำหรับอ้างอิงแรงดันที่เป็นค่าลบ (VREF-)
- **RA3/AN3/VREF+** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 3 ของพอร์ตเอ (RA3) ใช้เป็นขาสำหรับรับสัญญาณแบบแอนาล็อกช่องที่ 3 (AN3) หรือเป็นขาสำหรับอ้างอิงแรงดันที่เป็นค่าบวก (VREF+)
- **RA4/T0CKI** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 4 ของพอร์ตเอ (RA4) หรือใช้เป็นขาสำหรับเป็นตัวตั้งเวลาช่องที่ 0 (T0) หรือเป็นขารับสัญญาณนาฬิกา (CKI)
- **RA5/SS'/AN4** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 5 ของพอร์ตเอ (RA5) หรือใช้เป็นขาสำหรับรับสัญญาณแบบแอนาล็อกช่องที่ 4 (AN4) หรือเป็นขาสัญญาณกำหนดให้เป็นสลาฟ (SS': Slave Select) เพื่อใช้กับการสื่อสารพอร์ตอนุกรมแบบซิงโครนัสซีเรียลพอร์ต (Synchronous Serial Port)
- **RB0/INT** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 0 ของพอร์ตบี (RB0) หรือใช้เป็นขาสำหรับรับสัญญาณขัดจังหวะ หรืออินเทอร์รัพต์จากภายนอก (INT) RB1 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 1 ของพอร์ตบี (RB1)
- **RB2** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 2 ของพอร์ตบี (RB2)
- **RB3/PGM** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 3 ของพอร์ตบี (RB3) หรือขาสัญญาณสำหรับการโปรแกรมแบบแรงดันต่ำ
- **RB4** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 4 ของพอร์ตบี (RB4)
- **RB5** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 5 ของพอร์ตบี (RB5)
- **RB6/PGC** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 6 ของพอร์ตบี (RB6) หรือขาสัญญาณนาฬิกาสำหรับโหมดการโปรแกรม
- **RB7/PGD** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 7 ของพอร์ตบี (RB7) หรือขาสัญญาณข้อมูลสำหรับโหมดการโปรแกรม

- **RC0/T1OSO/T1CKI** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 0 ของพอร์ตซี (RC0) หรือขาสัญญาณออสซิลเลตอร์สำหรับส่งออกของตัวตั้งเวลาช่องที่ 1 (T1OSO) หรือขานำเข้า (Input Pin) สัญญาณของตัวตั้งเวลาช่องที่ 1 (T1CKI)
- **RC1/T1OSI/CCP2** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 1 ของพอร์ตซี (RC1) หรือขาสัญญาณออสซิลเลตอร์สำหรับนำเข้าของตัวตั้งเวลาช่องที่ 1 (T1OSI) หรือเป็นขาสัญญาณสำหรับตัวจับสัญญาณนำเข้าช่องที่ 2 (Capture 2 input) หรือสัญญาณส่งออกของการเปรียบเทียบของช่องที่ 2 (Compare 2 Output) หรือขาส่งสัญญาณออกพีดับเบิลยูเอ็ม2 (PWM 2 Output)
- **RC2/CCP1** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 2 ของพอร์ตซี (RC2) หรือเป็นขาสัญญาณสำหรับตัวจับสัญญาณนำเข้าช่องที่ 1 (Capture1 input) หรือสัญญาณส่งออกของการเปรียบเทียบของช่องที่ 1 (Compare 1 output) หรือขาส่งสัญญาณออกพีดับเบิลยูเอ็ม1 (PWM 1 output)
- **RC3/SCK/SCL** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 3 ของพอร์ตซี (RC3) หรือขาสัญญาณนาฬิกาสำหรับทำการสื่อสารแบบซิงโครนัส (SCK) หรือเป็นขาสัญญาณนาฬิกาสำหรับการทำงานในโหมด I^2C และ SPI (SCL)
- **RC4/SDI/SDA** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 4 ของพอร์ตซี (RC4) หรือขาสัญญาณนำเข้าข้อมูลในการสื่อสารแบบซิงโครนัส (SDI) หรือทำหน้าที่เป็นขานำเข้า/ส่งออกสัญญาณข้อมูลในโหมด I^2C (SDA)
- **RC5/SDO** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 5 ของพอร์ตซี (RC5) หรือขาสัญญาณส่งออกข้อมูลในการสื่อสารแบบซิงโครนัส (SDO)
- **RC6/TX/CK** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 6 ของพอร์ตซี (RC6) หรือขาสัญญาณนำข้อมูลออกในการสื่อสารอนุกรม (TX) หรือเป็นขาสัญญาณนาฬิกาในโหมดการสื่อสารแบบซิงโครนัส (CK)
- **RC7/RX/DT** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 7 ของพอร์ตซี (RC7) หรือขาสัญญาณรับข้อมูลนำเข้าจากการสื่อสารอนุกรม (RX) หรือเป็นขาสัญญาณข้อมูลในโหมดการสื่อสารแบบซิงโครนัส (DT)
- **RD0/PSP0** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 0 ของพอร์ตดี (RD0) หรือใช้เป็นสลาฟพอร์ตที่ 0 (PSP0) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์

- **RD1/PSP1** เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 1 ของพอร์ตดี (RD1) หรือใช้เป็นสลาฟพอร์ตที่ 1 (PSP1) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์
- RD2/PSP2 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 2 ของพอร์ตดี (RD2) หรือใช้เป็นสลาฟพอร์ตที่ 2 (PSP2) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์
- RD3/PSP3 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 3 ของพอร์ตดี (RD3) หรือใช้เป็นสลาฟพอร์ตที่ 3 (PSP3) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์
- RD4/PSP4 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 4 ของพอร์ตดี (RD4) หรือใช้เป็นสลาฟพอร์ตที่ 4 (PSP4) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์
- RD5/PSP5 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 5 ของพอร์ตดี (RD5) หรือใช้เป็นสลาฟพอร์ตที่ 5 (PSP5) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์
- RD6/PSP6 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 6 ของพอร์ตดี (RD6) หรือใช้เป็นสลาฟพอร์ตที่ 6 (PSP6) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์
- RD7/PSP7 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 7 ของพอร์ตดี (RD7) หรือใช้เป็นสลาฟพอร์ตที่ 7 (PSP7) ในกรณีที่ต่อเข้ากับระบบบัสของไมโครโพรเซสเซอร์
- RE0/RD/AN5 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 0 ของพอร์ตอี (RE0) หรือเป็นขาสัญญาณควบคุมการอ่านในโหมดพารัลเรล (Parallel Mode) สลาฟพอร์ต (RD) หรือใช้เป็นขานำเข้าสัญญาณแอนาล็อกช่องที่ 5 (AN5)
- RE1/WR/AN6 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 1 ของพอร์ตอี (RE1) หรือเป็นขาสัญญาณควบคุมการเขียนในโหมดพารัลเรลสลาฟพอร์ต (WR) หรือใช้เป็นขานำเข้าสัญญาณแอนาล็อกช่องที่ 6 (AN6)
- RE2/CS/AN7 เป็นขาสัญญาณสำหรับนำเข้าและส่งออกขาที่ 2 ของพอร์ตอี (RE2) หรือเป็นขาสัญญาณควบคุมการเลือก (Control Select) ในโหมดพารัลเรลสลาฟพอร์ต (CS) หรือใช้เป็นขานำเข้าสัญญาณแอนาล็อกช่องที่ 7 (AN7)

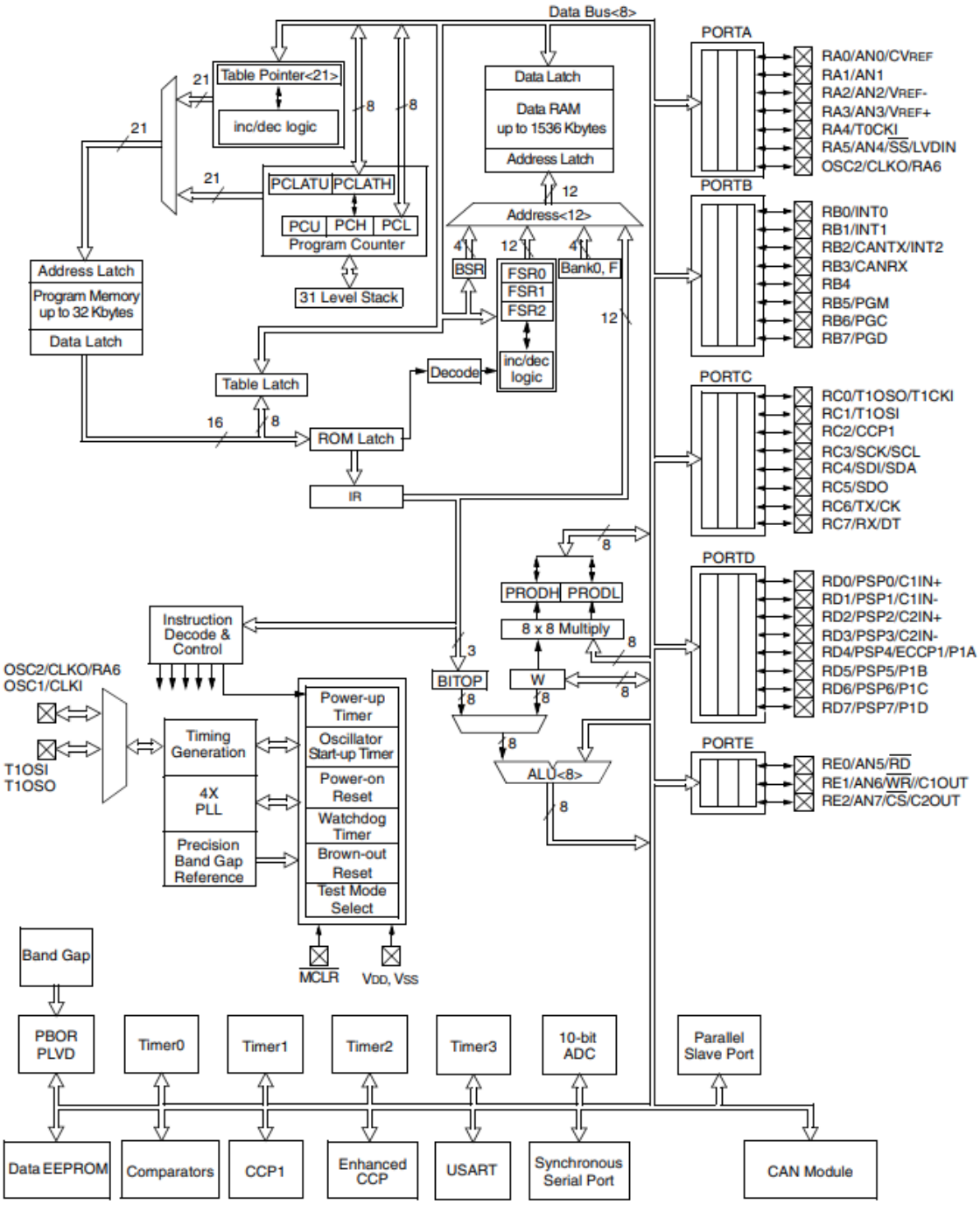
สำหรับวงจรพื้นฐานของไมโครคอนโทรลเลอร์ PIC ที่ใช้สำหรับให้ระบบทำงานได้ (ไม่รวมวงจรสำหรับโปรแกรมชิพ) และเป็นวงจรสำหรับใช้เชื่อมต่อกับอุปกรณ์ต่อพ่วงที่กล่าวถึงในบทถัดไปของหนังสือเล่มนี้เป็นดังรูปที่ 1.10 ซึ่งรายการอุปกรณ์ที่ต้องใช้ประกอบด้วย

1. ตัวต้านทาน R1 ความต้านทาน 4.7 กิโลโอห์ม
2. สวิตช์ S1 เป็นสวิตช์แบบกดติดปล่อยดับ
3. คริสตัล Q1 ความถี่ 4 ถึง 20 เมกะเฮิรตซ์ (หนังสือเล่มนี้เลือกใช้ 10 เมกะเฮิรตซ์)
4. ตัวเก็บประจุ C1 และ C2 มีความจุ 22 พิโกฟารัด
5. ไอซีไมดครคอนโทรลเลอร์ IC1 เบอร์ PIC16F877 หรือ PIC18F458 ที่รองรับความถี่สัญญาณนาฬิกาที่ 10 เมกะเฮิรตซ์



รูปที่ 1.10 วงจรพื้นฐานสำหรับไมโครคอนโทรลเลอร์ PIC

ภายในไมโครคอนโทรลเลอร์มักเป็นการนำการทำงานหลากหลายอย่างมารวมกันภายใต้ชิพเพียงตัวเดียว เช่นเดียวกันกับตัวไมโครคอนโทรลเลอร์ PIC18F458 เองมีส่วนประกอบภายในตาม**รูปที่** 1.11 พบว่าการทำงานภายในของชิพประกอบไปด้วยส่วนของวงจรต่าง ๆ ตามคุณสมบัติที่ได้กบ่าวถึงในช่วงต้นบทความ และเมื่อพิจารณาในรายละเอียดจะเห็นถึงความสัมพันธ์ของแต่ละส่วนว่าเชื่อมต่อกันอย่างไร โดยหัวลูกศรของเส้นต่าง ๆ เป็นตัวระบุทิศทางการรับ/ส่งข้อมูลของแต่ละส่วน จึงขอกล่าวถึงแบบสรุปดังนี้



รูปที่ 1.11 Block Diagram ของ PIC18F458
ที่มา: p.11, Figure 1-2: PIC18F448/458 Block Diagram

1. Data Bus หรือบัสข้อมูลมีขนาด 8 บิตตามสถาปัยกรรมของชิพเชื่อโยงเข้ากันกับทุกโมดูลภายในชิพที่มีการรับ/ส่งข้อมูลระหว่างกัน
2. ขนาดของ Address Bus หรือบัสอ้างอิงตำแหน่งจะมีขนาด 21 บิต
3. ข้อมูลที่มีการรับส่งมีทั้งแบบ 8 บิตและ 16 บิต (ดูจาก Data Latch)

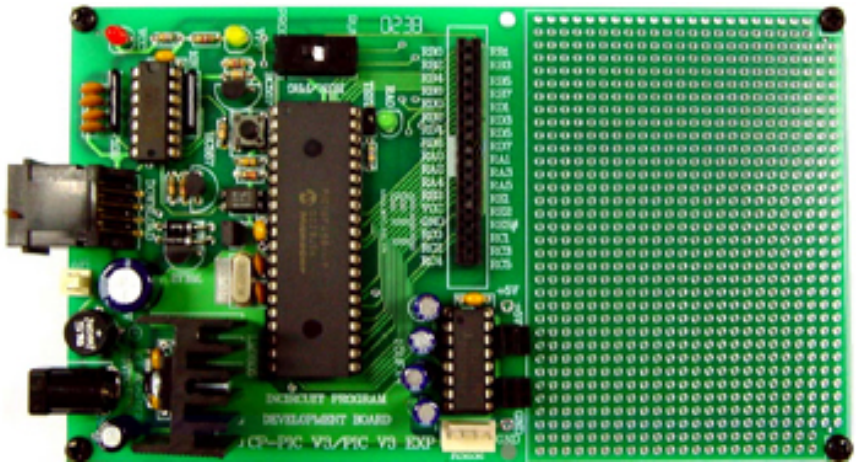
4. พอร์ตใช้งานมีดังนี้
   - PORTA มีขา RA0, RA1, RA2, RA3, RA4, RA5, OSC2
   - PORTB มีขา RB0, RB1, RB2, RB3, RB4, RB5, RB6, RB7
   - PORTC มีขา RC0, RC1, RC2, RC3, RC4, RC5, RC6, RC7
   - PORTD มีขา RD0, RD1, RD2, RD3, RD4, RD5, RD6, RD7
   - PORTE มีขา RE0, RE1, RE2
5. การประมวลผลของหน่วย ALU กระทำแบบ 8 บิต และมีภาคคำนวณข้อมูล 8×8 หรือ 8 บิต คูณ กับ 8 บิตในการทำงานครั้งเดียว

ไมโครคอนโทรลเลอร์จะใช้สัญญาณนาฬิกาเป็นตัวขับเคลื่อนการทำงาน โดยมองเป็นลักษณะของ วงรอบ (Cycle) ซึ่งระบุเอาไว้ว่า 1 คำสั่งนั้นจะประกอบไปด้วย 1-2 วงรอบ โดยในแต่ละวงรอบนั้นจะแบ่งเป็น 4 ส่วน คือ Q1, Q2, Q3 และ Q4 ด้วยเหตุนี้ ความเร็วโดยรวมของ PIC จึงเท่ากับ ค่าความถี่ของสัญญาณนาฬิกา หาร ด้วย 4 แต่ในรุ่น PIC18F458 มีวงจร PLL ทำให้สามารถเร่งความเร็วสัญญาณนาฬิกาสูงสุดที่ 4 เท่า ทำให้ใช้คริสตัลความถี่เดียวกับไมโครคอนโทรลเลอร์ PIC รุ่นที่ต่ำกว่าแต่สามารถเร่งการทำงานได้เร็วกว่าได้สูงสุด 4 เท่า หรือในทางกลับกัน หมายความว่า ชิพในตระกูลนี้ทำงาน 1 วงรอบการทำงานภายใน 1 สัญญาณนาฬิกา ยกเว้นคำสั่งกระโดดและการเปรียบเทียบแล้วกระโดดที่มีขั้นตอนการทำงานมากกว่าการคำนวณหรือโอนข้อมูลจึงต้องใช้จำนวนสัญญาณนาฬิกาสูงกว่าปกติ

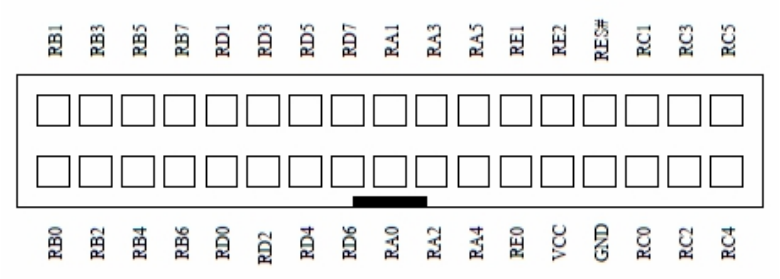
6. แนะนำบอร์ด CP-PIC V3

บอร์ด CP-PIC V3 เป็นบอร์ดไมโครคอนโทรลเลอร์ของบริษัทอีทีที (รูปที่ 1.12 และ 1.13) ที่ถูกออกแบบขึ้นมาเพื่อใช้กับชิพ PIC16F877 และ PIC18F458 โดยบอร์ดคุณสมบัติพื้นฐานดังนี้

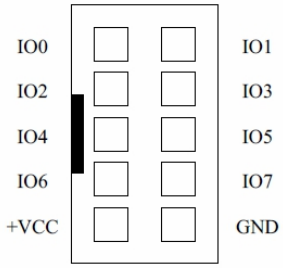
1. มีวงจรภาคจ่ายไฟแบบกระแสตรง
2. มีวงจรรีเซ็ต
3. มีวงจรกำเนิดความถี่สัญญาณนาฬิกาที่ความเร็ว 20 เมกะเฮิรตซ์
4. มีวงจรสำหรับโปรแกรมชิพแบบแรงดันสูง หรือ High Voltage
5. มีวงจรสื่อสารอนุกรมแบบอาร์เอส 232
6. มีคอนเนคเตอร์ 34 ขา จัดเรียงตามมาตรฐานของบริษัทอีทีที (รูปที่ 1.14) มีพื้นที่ว่างสำหรับต่อวงจรเพิ่มเติม



รูปที่ 1.12 บอร์ด CP-PIC V3



รูปที่ 1.13 การจัดเรียงขาของคอนเนคเตอร์ 34 ขา ตามมาตรฐานของอีทีที



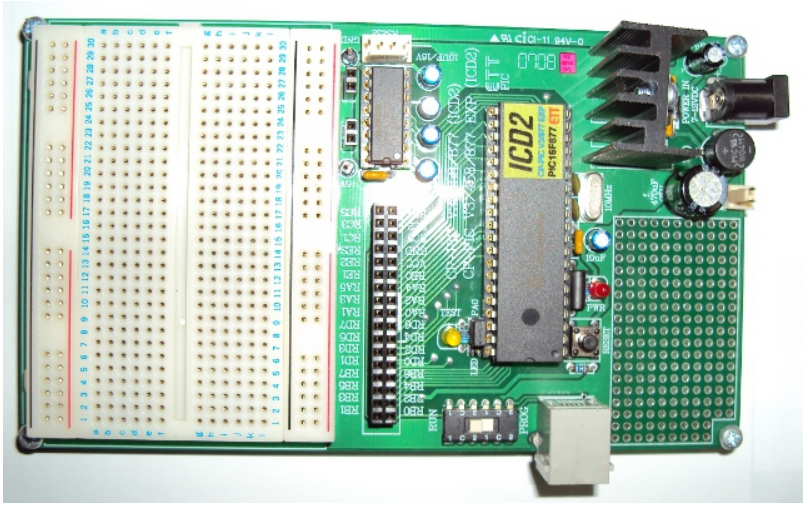
รูปที่ 1.14 การจัดเรียงขาของคอนเนคเตอร์ 10 ขา ตามมาตรฐานของอีทีที

การโปรแกรมชิพผ่านทางบอร์ด CP-PIC V3 ผู้พัฒนาควรถอดการเชื่อมต่อกับวงจรที่ออกแบบก่อน เพราะถ้าวงจรภายนอกมีการใช้ขาสัญญาณเดียวกับการโปรแกรมชิพจะทำให้การโปรแกรมชิพผิดพลาด และเมื่อโปรแกรมชิพเสร็จเรียบร้อยควรถอดสายเชื่อมต่อที่ใช้สำหรับการโปรแกรมออกก่อนทดลองการทำงานเสมอ

7. แนะนำบอร์ด CP-PIC V3/877 (ICD2)

บอร์ด CP-PIC V3/877 (ICD2) เป็นบอร์ดไมโครคอนโทรลเลอร์ของบริษัทอีทีที (รูปที่ 1.15) ที่ถูกออกแบบขึ้นมาเพื่อใช้กับชิพ PIC16F877 โดยไม่มีภาคโปรแกรมชิพเหมือนกับ CP-PIC V3 แต่ผู้พัฒนาสามารถโปรแกรมชิพได้ด้วยเครื่องมือโปรแกรมชิพไมโครคอนโทรลเลอร์รุ่น ET-PGM PIC USB V1 ที่มีความสะดวกในการพัฒนามากกว่า (อ่านได้จากหัวข้อ **แนะนำบอร์ด ET-PGM PIC USB V1**) ตัวบอร์ด CP-PIC V3/877 (ICD2) มีคุณสมบัติพื้นฐานดังนี้

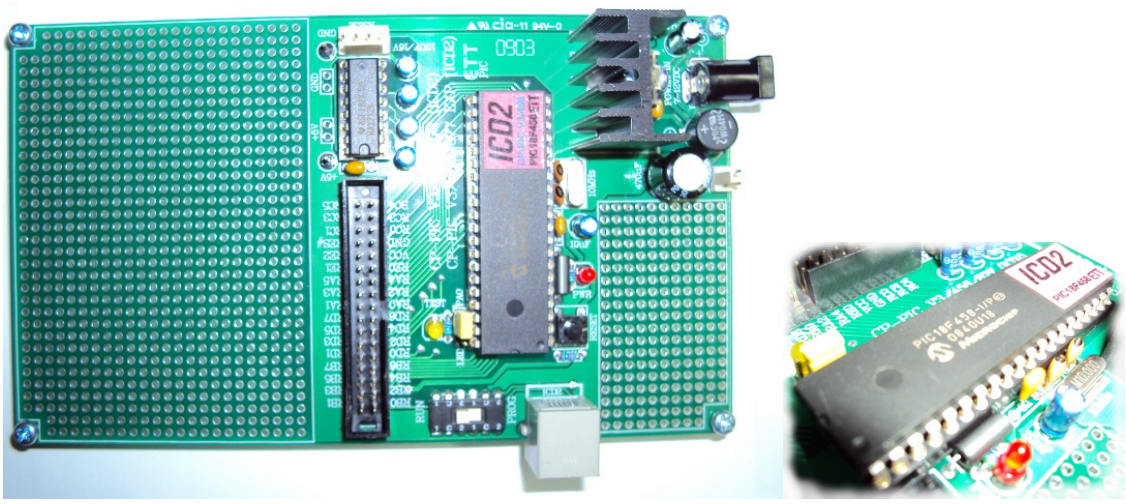
1. มีวงจรภาคจ่ายไฟแบบกระแสตรง
2. มีวงจรรีเซ็ต
3. มีวงจรกำเนิดความถี่สัญญาณนาฬิกาที่ความเร็ว 10 เมกะเฮิรตซ์
4. มีวงจรสื่อสารอนุกรมแบบอาร์เอส 232
5. มีคอนเนคเตอร์ 34 ขา จัดเรียงตามมาตรฐานของบริษัทอีทีที (รูปที่ 1.9)
6. มีขั้ว RJ-11 (ICD2) สำหรับดาวน์โหลดโปรแกรมเข้าตัวโครคอนโทรลเลอร์
7. มีพื้นที่ว่างสำหรับต่อวงจรเพิ่มเติม



รูปที่ 1.15 บอร์ด CP-PIC V3/877

8. แนะนำบอร์ด CP-PIC V3/458 (ICD2)

บอร์ด CP-PIC V3/458 (ICD2) เป็นบอร์ดไมโครคอนโทรลเลอร์ของบริษัทอีทีที (รูปที่ 1.16) ที่ใช้ชิพ PIC18F458 เป็นชิพประจำบอร์ด แต่มีคุณสมบัติเช่นเดียวกับ CP-PIC V3/877 (ICD2)

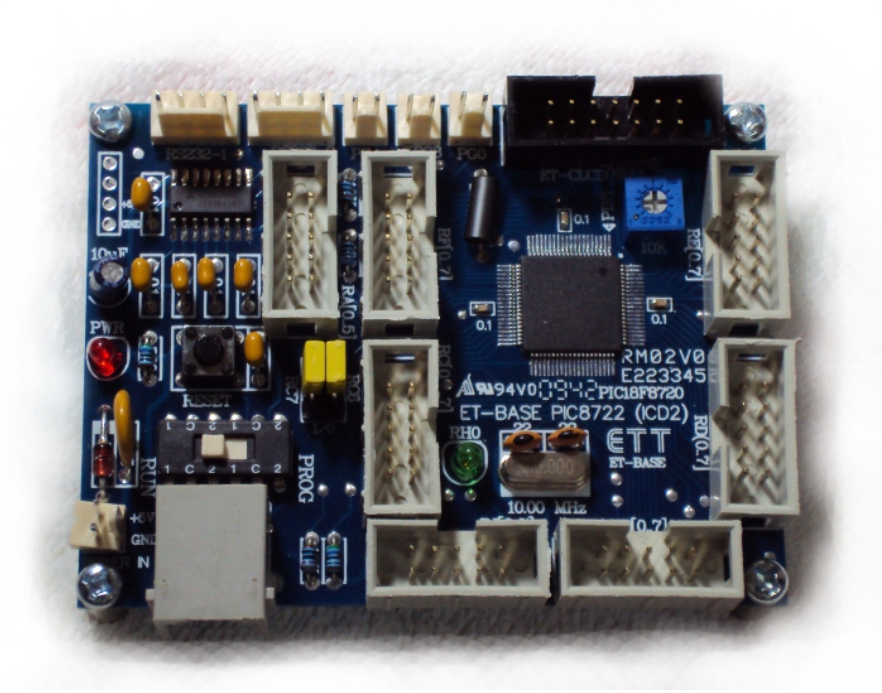


รูปที่ 1.16 บอร์ด CP-PIC V3/458 (ICD2)

9. แนะนำบอร์ด ET-BASE PIC8722 (ICD2)

บอร์ด ET-BASE PIC8722 (ICD2) เป็นบอร์ดไมโครคอนโทรลเลอร์ของบริษัทอีทีที (รูปที่ 1.17) ที่ใช้กับชุด ET-PGM PIC USB V1 ผ่านทางพอร์ตยูเอสบี และเลือกใช้ชิพไมโครคอนโทรลเลอร์รุ่น PIC18F8722 เป็นไมโครคอนโทรลเลอร์ประจำบอร์ด ซึ่งบอร์ด ET-BASE PIC8722 (ICD2) มีคุณสมบัติดังต่อไปนี้

1. ชิพมีหน่วยความจำแบบแฟลชความจุ 128 กิโลไบต์
2. ชิพมีหน่วยความจำแรมขนาด 3,936 ไบต์
3. ชิพมีหน่วยความจำแบบ EEPROM ขนาด 1,024 ไบต์
4. มีวงจรกำเนิดความถี่สัญญาณนาฬิกาที่ความเร็ว 10 เมกะเฮิรตซ์ โดยสามารถตั้งให้ทำงานได้สูงสุดที่ 40 เมกะเฮิรตซ์
5. มีพอร์ตไอ/โอทั้งหมด 70 บิต
6. มีคอนเนคเตอร์ 10 ขา ตามมาตรฐานของบริษัท อีทีที จำนวน 7 ชุด (รูปที่ 1.14)
7. มีคอนเนคเตอร์ 14 ขาสำหรับเชื่อมต่อกับแอลซีดีแบบอักขระ
8. มีขั้ว RJ-11 (ICD2) สำหรับดาวน์โหลดโปรแกรมเข้าตัวโครคอนโทรลเลอร์
9. มีวงจรสื่อสารอนุกรมแบบอาร์เอส 232 จำนวน 2 ชุด
10. ชิพมีวงจรแปลงสัญญาณแอนาล็อกเป็นดิจิทัล (A TO D) ขนาด 10บิต จำนวน 16 ช่องสัญญาณ
11. ชิพมีวงจรตั้งเวลา (Timer) ตัวนับ (Counter) สร้างพีดับเบิลยูเอ็ม (PWM) และวอทช์ด็อก (Watch Dog
12. แผ่นวงจรขนาด 6.2 x 8.1 ซม.



รูปที่ 1.17 บอร์ด ET-BASE PIC8722 (ICD2)

10. แนะนำบอร์ด ET-PGM PIC USB V1

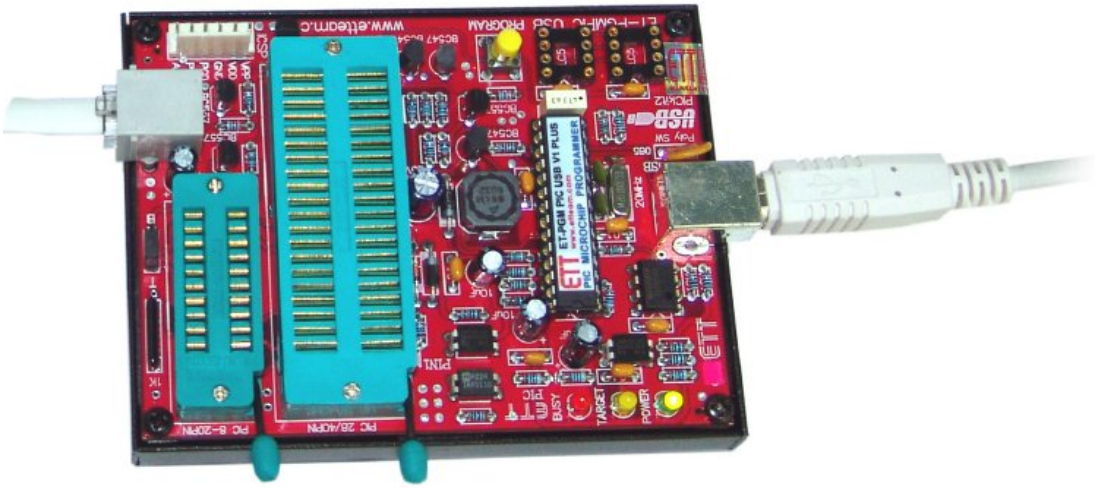
บอร์ด ET-PGM PIC USB V1 และ ET-PGM PIC USB V1 Plus เป็นเครื่องโปรแกรมชิพไมโครคอนโทรลเลอร์ PIC ชนิดหน่วยความจำแบบแฟลชของบริษัท อีทีที จำกัด ซึ่งเครื่องโปรแกรมชุดนี้มีคุณสมบัติเทียบเท่ากับเครื่องโปรแกรมชิพรุ่น PICKIT 2 ของบริษัทไมโครชิพ ที่ผู้ใช้สามารถโปรแกรมชิพ PIC จากเครื่องคอมพิวเตอร์ได้โดยตรงผ่านทางพอร์ตยูเอสบี โดยเครื่องโปรแกรมจะใช้ไฟเลี้ยงจากพอร์ตยูเอสบีทำให้ไม่ต้องพ่วงเครื่องจ่ายไฟที่มีแรงดัน 16 โวลต์เหมือนกับบอร์ด CP-PIC V3 และด้วยการเชื่อมต่อผ่านพอร์ตยูเอสบีในการรับส่งข้อมูล ทำให้เครื่องโปรแกรมชิพรุ่นนี้มีความเร็วในการโปรแกรมชิพที่สูงกว่าวิธีการโปรแกรมผ่านพอร์ตขนาน หรือพอร์ตอนุกรมเป็นอย่างมาก และนอกจากนี้ผู้ใช้สามารถปรับรุ่นของเฟิร์มแวร์ของบอร์ดให้รองรับชิพไมโครคอนโทรลเลอร์รุ่นใหม่ๆ ได้ด้วยการใช้เฟิร์มแวร์ตัวเดียวกับชุด PICKIT 2 ของบริษัทไมโครชิพ

คุณสมบัติของ ET-PGM PIC USB V1 และ ET-PGM PIC USB V1 Plus (รูปที่ 1.18) มีดังนี้

1. สามารถใช้งานกับชิพไมโครคอนโทรลเลอร์ของบริษัทไมโครชิพในรุ่นดังต่อไปนี้
   1.1 รุ่น 12 บิตและ 14 บิต ได้แก่
      1.1.1 PIC10FXXX
      1.1.2 PIC12FXXX
      1.1.3 PIC16FXXX

1.2 รุ่น PIC18FXXX
1.3 รุ่น PIC18FXJXX
1.4 รุ่น PIC18FXKXX
1.5 รุ่น PIC24XX
1.6 รุ่น dsPIC33XX

2. เฉพาะรุ่น ET-PGM PIC USB V1 Plus สามารถโปรแกรมชิพไมโครคอนโทรลเลอร์โดยใช้ เท็กซ์ทูล (Text Tools) แบบ 40 ขา และ 20 ขา เพื่อโปรแกรมชิพไมโครคอนโทรลเลอร์ที่มีขาตั้งแต่ 8 ขา ถึง 40 ขา
3. บอร์ดมีขั้วต่อสำหรับการโปรแกรมชิพตามมาตรฐาน ICD2 ของบริษัทไมโครชิพ 2 แบบ คือ
   3.1 ขั้วต่อชนิด RJ11
   3.2 ขั้วต่อชนิดเวเฟอร์ (Wafer) 2.54 มม. ขนาด 6 ขา
4. สั่งโปรแกรมได้ด้วยการกดสวิทช์โปรแกรม (Program) บนบอร์ด
5. ต่อใช้งานกับตัวแปลง (Adapter) เพื่ออีมูลเลเตอร์ (Emulator) ได้โดยตรงกับบอร์ดไมโครคอนโทรลเลอร์
6. ขนาดของบอร์ด คือ 10 x 7.5 ซม. ซึ่งมาพร้อมกับฐานพลาสติก



รูปที่ 1.18 เครื่องโปรแกรมชิพรุ่น ET-PGM PIC USB V1 Plus

11. ขั้นตอนการพัฒนาระบบ

การพัฒนาระบบจะต้องอาศัยเครื่องคอมพิวเตอร์ส่วนบุคคลสำหรับการพัฒนาซอฟต์แวร์และโปรแกรมชิพ หลังจากนั้นผู้พัฒนาจึงนำบอร์ดไมโครคอนโทรลเลอร์ไปทดสอบทำงานจริง ถ้าพบข้อผิดพลาดใดจะต้องดำเนินการแก้ไขซอฟต์แวร์ หรือฮาร์ดแวร์ที่ออกแบบให้ถูกต้อง ดังนั้น สามารถสรุปเป็นขั้นตอนการพัฒนาระบบสมองกลฝังตัวด้วยไมโครคอนโทรลเลอร์ PIC16/18 ได้ดังนี้

1. วิเคราะห์ปัญหา และออกแบบฮาร์ดแวร์/อุปกรณ์ต่อพ่วงสำหรับประยุกต์ใช้งาน
2. ทำต้นแบบบอร์ดควบคุม และอุปกรณ์ต่อพ่วง
3. ออกแบบซอฟต์แวร์สำหรับควบคุมฮาร์ดแวร์ที่ออกแบบในขั้นตอนที่ 1
4. พัฒนาซอฟต์แวร์ด้วยโปรแกรมภาษาที่เลือก
   4.1. เขียนโค้ดตามที่ออกแบบไว้ในขั้นตอนที่ 3 ด้วยภาษาที่เลือกใช้ ผ่านทางชุดพัฒนาซอฟต์แวร์ โดยในหนังสือเล่มนี้เลือกใช้ชุดพัฒนาซอฟต์แวร์ MPLAB X IDE ของบริษัทไมโครชิพ และตัวแปลภาษาซีในชุดพัฒนา MPLAB XC8 ที่รองรับไมโครคอนโทรลเลอร์ตระกูล 8 บิต และเป็นเครื่องมือที่แจกจ่ายให้ใช้งานฟรี รองรับระบบปฏิบัติการ Windows, Linux และ macOS
   4.2. สั่งแปลภาษาเพื่อให้ได้รหัสเครื่อง ซึ่งจะอยู่ในรูปของแฟ้มเลขฐานสิบหกหรือเฮ็กซ์ไฟล์ (HEX file) ถ้าโค้ดโปรแกรมที่เขียนนั้นไม่สามารถแปลภาษาได้ถูกต้องตามหลักการเขียนโปรแกรมของภาษาที่เลือกใช้จะต้องกลับไปแก้ไขโค้ดที่เขียนในขั้นตอนที่ 4.1 หรือถ้าพบว่าเกิดความผิดพลาดจากการออกแบบในขั้นตอนที่ 3 จะต้องตรวจสอบการออกแบบอีกครั้ง
   4.3. ทดลองดีบัก (Debug) หรือทดสอบการทำงานของโปรแกรม เพื่อหาข้อผิดพลาดในการใช้งาน เช่น ตรวจสอบการกำหนดช่วงค่าของตัวแปรนำเข้าว่าคงทนต่อความผิดพลาดหรือไม่ หรือ ทดสอบผลลัพธ์ของการคำนวณด้วยการจำลองค่านำเข้าเพื่อให้ได้ผลลัพธ์ที่ถูกต้อง เป็นต้น และถ้าไม่พบข้อผิดพลาดของการทำงานให้ไปขั้นตอนที่ 5
5. นำซอฟต์แวร์ที่พัฒนาเสร็จแล้วโปรแกรมลงในชิพ ทางชุดโปรแกรม ET-PGM PIC USB V1 หรืออุปกรณ์ที่เข้ากันได้กับ PICKit2 ด้วยโปรแกรม PICKit2
6. ทดสอบการทำงานของระบบ
   6.1 ถ้าพบข้อผิดพลาดจากการทำงานของฮาร์ดแวร์ให้กลับไปที่ขั้นตอนที่ 1
   6.2 ถ้าพบข้อผิดพลาดจากการทำงานของซอฟต์แวร์ให้กลับไปที่ขั้นตอนที่ 3
7. สร้างบอร์ดควบคุมใช้งานจริง

12. สรุป

จากบทนี้ได้รู้จักไมโครคอนโทรลเลอร์ว่ามีความแตกต่างกับคอมพิวเตอร์ที่ใช้งานทั่วไปอย่างไร และทราบถึงคุณลักษณะและคุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล PIC ได้แก่ PIC16F877 PIC18F458 และ PIC18F8722 รวมถึงการจัดเรียงขาของชิพไมโครคอนโทรลเลอร์ พร้อมทราบถึงหน้าที่ของขาเหล่านั้น ได้ทราบถึงวงจรทำงานพื้นฐานของระบบไมโครคอนโทรลเลอร์ PIC ได้รู้จักบอร์ดสำเร็จรูปทั้ง 3 รุ่นที่ใช้ไมโครคอนโทรลเลอร์ PIC เบอร์ PIC16F877 PIC18F458 และ PIC18F8722 ได้รู้จักเครื่องโปรแกรมชิพไมโครคอนโทรเลอร์ผ่านทางการเชื่อมต่อมาตรฐาน ICD2 ของบริษัทไมโครชิพ และได้ทราบถึงขั้นตอนของการพัฒนาระบบไมโครคอนโทรลเลอร์ทั้ง 7 ขั้นตอน

จุดแข็งของระบบไมโครคอนโทรลเลอร์เมื่อนำมาใช้กับงานสมองกลฝังตัว ได้แก่

1. บันทึกโปรแกรมลงในชิพและสั่งให้ทำตามโปรแกรมที่บันทึกอยู่ในชิพได้
2. เชื่อมประสานกับอุปกรณ์รอบข้างได้หลากหลาย
3. สามารถผนวกส่วนประกอบพิเศษอยู่ในตัวชิพ อันได้แก่ วงจรตั้งเวลา วงจรเปรียบเทียบ วงจรแปลงสัญญาณ เป็นต้น
4. รองรับวงจรการพัฒนา อันได้แก่ โปรแกรมการทำงานได้ ทดสอบการทำงานได้ และหาข้อผิดพลาดของโปรแกรมได้
5. มีขนาดเล็กทำให้ผนวกหรือฝังตัวเข้ากับอุปกรณ์ได้ง่าย

ในบทที่ 2 กล่าวถึงการใช้ซอฟต์แวร์โปรแกรมชิพไมโครคอนโทรลเลอร์ PIC ได้แก่ โปรแกรม PICkit 2 สำหรับการโปรแกรมชิพผ่านทางเครื่องโปรแกรม ET-PGM PIC USB V1 หรือเครื่องโปรแกรมรุ่นอื่น ๆ ที่ออกแบบให้ทำงานได้เหมือนกับเครื่องโปรแกรมชิพรุ่น PICkit2 ของบริษัทไมโครชิพ และอธิบายการใช้งาน MPLAB X IDE ที่ติดตั้ง MPLAB XC8 C-Compiler เป็นการเบื้องต้น เพื่อใช้งานภาษา C สำหรับไมโครคอนโทรลเลอร์ 8 บิต และการใช้งานเครื่องมือสำหรับสร้างโครงงาน ตั้งค่า เขียนโปรแกรม คอมไพล์ และนำไฟล์ผลลัพธ์ไปโปรแกรมลงชิพผ่านทาง PICKit2 ต่อไป