บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

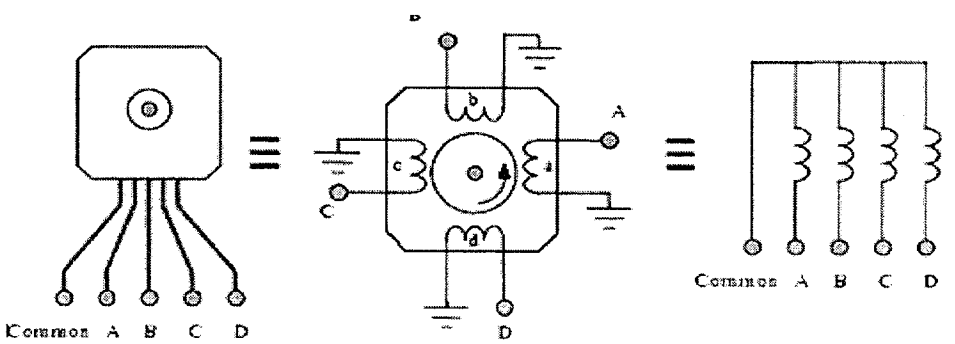
การสร้างระบบอัตโนมัติสามารถทำได้ใน 2 ลักษณะขึ้นกับอุปกรณ์ที่ใช้ทำหน้าที่ประมวลผลได้แก่การสร้างระบบอัตโนมัติโดยอาศัยส่วนประมวลผลกลาง (Central Processing Unit: CPU) หรืออุปกรณ์ที่ใกล้เคียงเช่น ไมโครคอนโทรลเลอร์ ตัวอย่างเช่น ณัฐวุฒิ และคณะ [1] สร้างระบบตรวจจับและยิงเป้าหมายอัตโนมัติ โดยใช้ การประมวลผลภาพจากกล้องวิดีโอเพื่อใช้ในการหาตำแหน่งของวัตถุและใช้ CPU เพื่อคำนวณหาระยะ, ความสูงและความเร็วของเป้าหมาย เพื่อประมาณจังหวะที่จะยิงเป้าหมายอย่างอัตโนมัติ เป็นต้น และการสร้างระบบอัตโนมัติโดยอาศัยชิปวงจรเฉพาะ (System on Chip: SoC) ตัวอย่างเช่น วิทยา และคณะ [2] ออกแบบตัวควบคุมกระบวนการ (Process control) แบบ PID บนชิป์ FPGA เป็นต้น

ระบบความคุมที่อาศัย CPU มีความซับซ้อนในการใช้งานที่น้อยกว่าผู้ใช้จะเน้นที่ซอร์ฟแวร์มากกว่าอาร์ดแวร์ กล่าวคือเราสามารถสร้างระบบขึ้นมาโดยพิจารณาที่ ขั้นตอนการทำงานของโปรแกรมเป็นหลัก เหมาะกับการนำไปใช้ในการสร้างนวัตกรรมใหม่ที่มีการทำงานไม่ซับซ้อนมาก ถ้าการทำงานซับซ้อนมากอาจต้องอาศัยวิธีการสร้างระบบอัตโนมัติแบบชิปวงจรเฉพาะ เนื่องจาก วิธีนี้เราจะต้องคำนึงถึงทั้ง ฮาร์ดแวร์และซอร์ฟแวร์ กล่าวคือ ในการสร้างระบบเราจะเป็นจะต้องพิจารณาถึงขั้นตอนการทำงานของโปรแกรมจากนั้นนำขั้นตอนดังกล่าวมาสังเคราะห์ (สร้าง) เป็นวงจรดิจิทัล วิธีนี้มีความซับซ้อนในการสร้างสูงแต่มีข้อดีที่ มีการประมวลผลภายในที่เร็ว ถ้าต้องนำไปเชื่อมต่อกับอุปกรณ์ภายนอกที่มีความเร็วใกล้เคียงกันก็จะทำให้การทำงานของระบบโดยรวมมีความเร็วมากขึ้นด้วยในงานวิจัยนี้ เราเลือกใช้การสร้างระบบอัตโนมัติโดยอาศัยส่วนประมวลผลกลาง เนื่องจากมีความประสิทธิภาพดีและมีความยืดหยุ่นในการใช้งานมากกว่า

ระบบควบคุมและโปรแกรมรูปแบบแสงบนเวทีแบบไร้สาย เป็นนวัตกรรมหนึ่งที่ประยุกต์ใช้องค์ความรู้ด้านระบบควบคุม, เทคโนโลยีการสื่อสารไร้สายและเทคโนโลยีไมโครคอนโทรลเลอร์ ซึ่งใช้ภาษาซี ในการควบคุม นอกจากนี้ยังมีอุปกรณ์อื่นๆ และความรู้ที่เกี่ยวข้องอีกได้แก่ การใช้งานสเต็ปปิ้งมอเตอร์ (Stepping Motor), การสื่อสารไร้สายโดยอุปกรณ์การรับ-ส่งข้อมูลด้วยคลื่นวิทยุ (RF Module), การพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์และการออกแบบและสร้างวงจรขับ Light emitting diode (LED) ลำดับต่อไปผู้วิจัยจะอธิบายทฤษฎีที่เกี่ยวข้องของระบบ

1. การใช้งานสเต็ปปิ้งมอเตอร์ (Stepping Motor)

เป็นมอเตอร์ไฟฟ้าชนิดหนึ่ง ซึ่งมีอินพุต เป็นกลุ่ม ของ Binary Voltage และ Output การเคลื่อนที่ในเชิงมุม (หมุน) แกนหมุน (Shaft) เป็น Step โดย Resolution ของสเต็ปปิ้งมอเตอร์อยู่ในช่วงตั้งแต่ 0.1- 30 องศาซึ่งขึ้นอยู่กับโครงสร้างของสเต็ปปิ้งมอเตอร์หรือบอกเป็นจำนวน Step ต่อ 1 รอบ สามารถควบคุมตำแหน่งการหมุนได้ ซึ่งจะมีความละเอียดของมุมในการหมุนและ Step ที่แตกต่างกันออกไป ทั้งนี้ก็ขึ้นอยู่กับ ชนิดของมอเตอร์และลักษณะการส่งสัญญาณไปควบคุมมอเตอร์ ทำให้สเต็ปปิ้งมอเตอร์มีความยืดหยุ่นในการนำมาใช้งาน ทำให้สามารถนำมาประยุกต์ใช้งานได้หลายรูปแบบ ไม่ว่าจะเป็นหุ่นยนต์หรือแขนกลต่างๆ เนื่องจากมีความแม่นยำในการควบคุมตำแหน่งสูง โครงสร้างของสเต็ปปิ้งมอเตอร์มีลักษณะดังรูป ซึ่งประกอบด้วย ขดลวด stator 4 ขด ล้อมรอบแกนหมุน

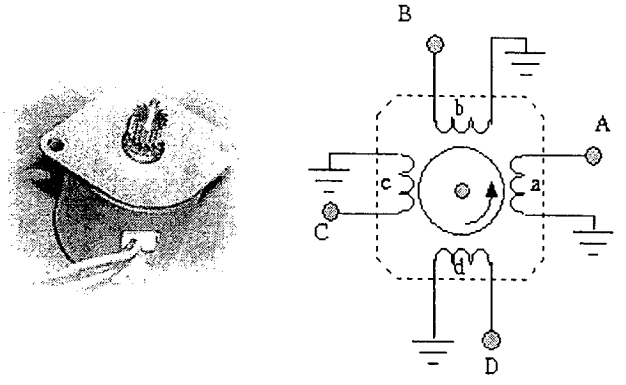


ภาพที่ 1 โครงสร้างของสเต็ปปิ้งมอเตอร์ (Stepping Motor) แบบ 4 phase Stator Coil

1.1 หลักการทำงาน

เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวด Stator Coil a, b, c, d ไม่พร้อมกันนั่นคือ ถ้าเราจ่ายกระแสให้ a ก่อน โดยไม่จ่ายให้ขดอื่น แล้วตามด้วย b, c และ d เรียงตามลำดับ จะทำให้เกิดสนามแม่เหล็ก หมุนวนในลักษณะทวนเข็มนาฬิกา ซึ่งส่วนของ Rotor ที่เป็นแม่เหล็กถาวรก็จะหมุนตามสนามแม่เหล็กไปด้วย คือ ทวนเข็มนาฬิกา

ในทำนองเดียวกันถ้าเราจ่ายกระแสให้ขด a, d, c, b, a ..... ก็จะให้ สนามแม่เหล็กหมุนในทิศทางตามเข็มนาฬิกา ซึ่งส่งผลให้ Rotor หมุนตามเข็มนาฬิกา ด้วย การกำหนดความเร็วของสเต็ปปิ้งมอเตอร์ทำได้โดยการเปลี่ยนแปลง ความเร็วของ การเปลี่ยนการจ่ายกระแสจากขดลวดขดหนึ่งไปยังอีกขดหนึ่งให้เร็วขึ้นการกำหนดความเร็วของสเต็ปปิ้งมอเตอร์ทำได้โดยการเปลี่ยนแปลงความเร็วของการเปลี่ยนการจ่ายกระแสจากขดลวดขดหนึ่งไปยังอีกขดหนึ่งให้เร็วขึ้น



ภาพที่ 2 ภาพสเต็ปมอเตอร์และวงจรสมมูล (Equivalent circuit)

1.2 ชนิดของสเต็ปมอเตอร์

1.2.1 แบบแม่เหล็กถาวร (PERMANENT MAGNET-PM)

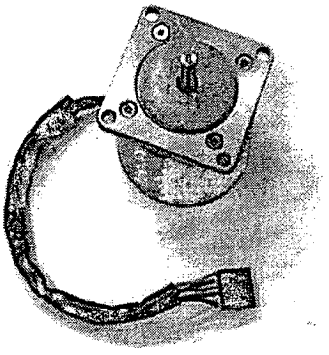
สเต็ปมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล โดยมีโรเตอร์ (ROTOR) เป็นรูปทรง กระบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้อนไฟกระแสตรง ให้กับขดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักต่อโรเตอร์ ทำให้มอเตอร์หมุน มอเตอร์แบบ PM จะเกิดแรงฉุดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

1.2.2 แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE-VR)

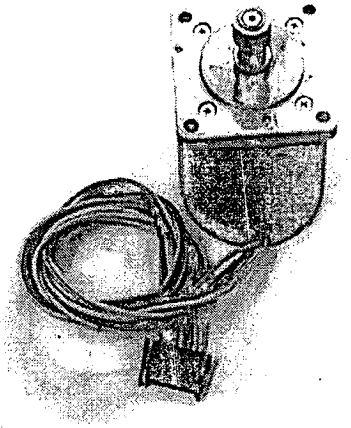
สเต็ปมอเตอร์แบบVR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ทำจากสารเฟอร์โรแมกเนติก กำลังอ่อน มีลักษณะเป็นฟันเลื่อย รูปทรงกระบอกโดยจะมีความสัมพันธ์ โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนซ์ต่ำที่สุด ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างขดกันไป ก็ทำให้มอเตอร์ หมุนไปตำแหน่งต่างๆ กันโรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

1.2.3 แบบผสม(HYBRID-H)

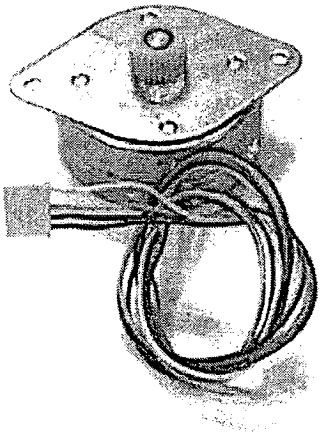
สเต็ปมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดีทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงฉุดยึดโรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ สเต็ปมอเตอร์มีลักษณะดังรูป



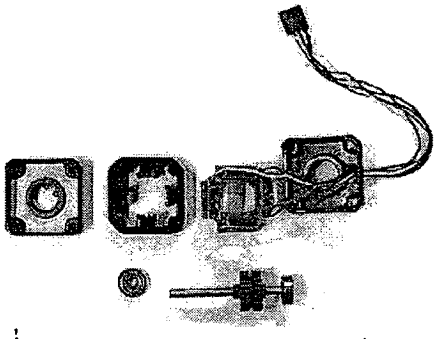
ภาพที่ 3 สเต็ปมอเตอร์แบบมีสาย 5 เส้น



ภาพที่ 4 มอเตอร์แบบมีสาย 6 เส้น



ภาพที่ 5 สเต็ปมอเตอร์หลายแบบไบโพลาร์



ภาพที่ 6 แสดงภาพถ่ายโครงสร้างสเต็ปมอเตอร์

1.3การควบคุมสเต็ปปิ้งมอเตอร์ (Stepping Motor)

ในการควบคุมการทำงานของสเต็ปปิ้งมอเตอร์ สามารถแบ่งได้ออกเป็น 3 รูปแบบ คือ

- ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave)
- ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส
- ควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป

1.3.1 ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave)

ในการควบคุมการหมุนของสเต็ปปิ้งมอเตอร์แบบ 4 เฟสนั้น เราจะต้องกระตุ้นให้มอเตอร์หมุนไปแต่ละ Step โดยจ่ายกระแสไฟฟ้าให้กับสเต็ปปิ้งมอเตอร์ทีละเฟสตามลำดับ หลักการคือเริ่มจากจ่ายกระแสให้กับขดลวด Stator เฟสที่ 1 จากนั้นกระตุ้นเฟสที่ 2 และ เฟสที่ 3 ไปเรื่อยๆ ตามลำดับ จากนั้นก็วนกลับมาที่ขดลวด Stator เฟสที่ 1 อีกครั้งและวน Loop ไปเรื่อยๆ ก็จะทำให้ สเต็ปปิ้งมอเตอร์หมุนและในทางกลับกันถ้าต้องการให้ สเต็ปปิ้งมอเตอร์หมุนกลับทางก็ต้องกระตุ้นขดลวด Stator เฟส 4 เฟส 3 เฟส 2 และ เฟส 1 ตามลำดับ สามารถเขียนขั้นตอนการทำงานเป็นตารางออกมาได้ดังนี้

ตารางที่ 1 ตารางควบคุมแบบ Full Step 1 เฟสหรือแบบเวฟ (wave)

Step	เฟส 4	เฟส 3	เฟส 2	เฟส1
Step 1	ON	OFF	OFF	OFF
Step 2	OFF	ON	OFF	OFF
Step 3	OFF	OFF	ON	OFF
Step 4	OFF	OFF	OFF	ON
Step 5	ย้อนกลับ Step ที่1			

1.3.2 ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส

ในการความคุมสเต็ปปิ้งมอเตอร์แบบ 2 เฟสนั้น เราจะต้องจ่ายกระแสไฟฟ้าเพื่อกระตุ้นขดลวดของมอเตอร์ทีละ 2 เฟส ในเวลาเดี่ยวกันและเรียงกันไปตามลำดับซึ่งได้แสดงดังตารางด้านล่าง โดยสเต็ปปิ้งมอเตอร์จะหมุนเหมือนกับการควบคุมแบบเวฟ แต่การควบคุมแบบ 2 เฟสจะให้แรงบิดที่สูงกว่าแบบเวฟ

ตารางที่ 2 ตารางควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส

Step	เฟส 4	เฟส 3	เฟส 2	เฟส1
Step 1	ON	ON	OFF	OFF
Step 2	OFF	ON	ON	OFF
Step 3	OFF	OFF	ON	ON
Step 4	ON	OFF	OFF	ON
Step 5	ย้อนกลับ Step ที่1			

1.3.3 ควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป

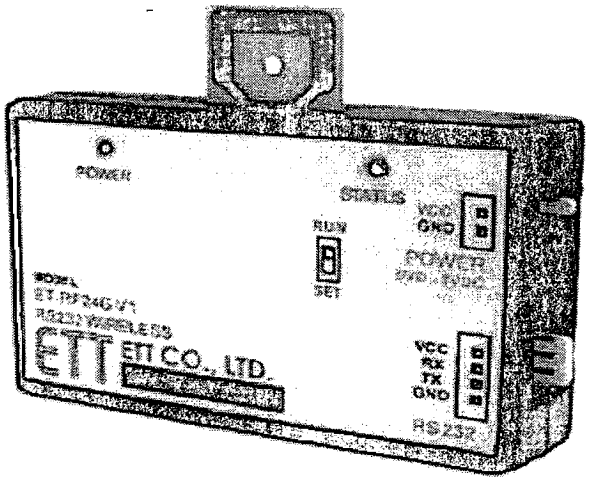
การควบคุมสเต็ปปิ้งมอเตอร์แบบครึ่งสเต็ปจะทำให้เราสามารถเพิ่มความละเอียดในการควบคุมการหมุนของสเต็ปปิ้งมอเตอร์ได้แม่นยำมากขึ้นซึ่งเป็นการผสมผสานระหว่างการควบคุมแบบเวฟและแบบ Full Step 2 เฟสเข้าด้วยกัน ลักษณะการจ่ายกระแสไฟ เพื่อกระตุ้นขดลวดจะแสดงดังตารางด้านล่าง

1.3.4 ข้อดีของสเต็ปปิ้งมอเตอร์เมื่อเปรียบกับมอเตอร์กระแสตรง (DC MOTOR)

- การควบคุมไม่ต้องอาศัยตัวตรวจจับการหมุน
- ไม่ต้องใช้แปรงถ่าน ดังนั้นจึงทำให้ไม่มีส่วนที่จะต้องสึกหรอ และปัญหาของการสปาร์ค (ที่เกิดจากหน้าสัมผัสของแปรงถ่านแหวนตัวนำในโรเตอร์) ที่ทำให้เกิดสัญญาณรบกวน
- การควบคุมโดยทางวงจรดิจิตอลหรือไมโครโพรเซสเซอร์ ทำได้ง่าย และสะดวก

2. การสื่อสารไร้สายโดยอุปกรณ์การรับ-ส่งข้อมูลด้วยคลื่นวิทยุ (RF Module)

เป็นชุดสำเร็จ Signal Converter เปลี่ยนแปลงสัญญาณในระบบ RS-232 ที่เป็นสายให้สามารถส่งข้อมูลออกไปได้ในแบบไร้สาย เป็นสัญญาณ RF ความถี่ 2.4 GHz สามารถรับส่งได้ในระยะไกล 50 m ถึง 280 m มีรูปแบบและความเร็วสูงในการรับ-ส่งข้อมูลสูง โดยส่งแบบ GFSK (Gaussian Frequncy Shift Keying) ความเร็วรับ-ส่งข้อมูลสูง1 Mbps,250 Kbps มีทั้งตัวรับ-ตัวส่งในชุดเดียวกัน โดยเป็นแบบสลับกันทำงานรับส่ง



ภาพที่ 7 อุปกรณ์การรับ-ส่งข้อมูล (RF Module).

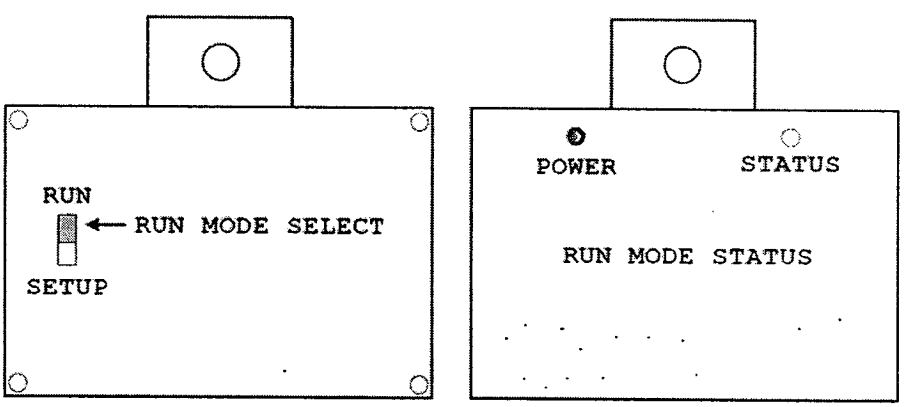
2.1 โหมดการทำงาน

สำหรับโหมดการทำงานของ ET-RF24G V1.0 นั้น จะแบ่งออกเป็น 2 โหมดด้วยกัน โดยการกำหนดโหมดการทำงานของ ET-RF24G V1.0 นั้นจะกระทำผ่าน Switch เลือกโหมด ซึ่งอยู่ด้านใต้กล่อง โดยการเลือกโหมดการทำงานนั้นจะต้องกระทำให้เสร็จเรียบร้อยก่อนการจ่ายไฟให้กับ ET-RF24G V1.0 ด้วยเสมอ เนื่องจากการทำงานของเครื่อง ET-RF24G V1.0 นั้นทำการตรวจสอบโหมดการทำงานของเครื่องจาก Switch เลือกโหมด เฉพาะในช่วงของการจ่ายไฟเลี้ยงให้เครื่องเริ่มต้นทำงานครั้งแรก (Power-ON) เท่านั้น ซึ่งการเปลี่ยนแปลงตำแหน่งการทำงานของ Switch เลือกโหมด หลังจากทำการจ่ายไฟให้กับ ET-RF24G V1.0 ไปแล้ว จะไม่มีผลต่อการทำงานของเครื่องแต่อย่างใด โดยการทำงานของเครื่อง ET-RF24G V1.0 นั้น จะมี LED แสดงสถานะการทำงานของเครื่องจำนวน 2 หลอด คือ LED Power ซึ่งเป็น LED สีแดงโดยที่ LED Power นี้จะติดสว่างให้เห็นตลอดเวลาที่มีการจ่ายไฟเลี้ยงให้เครื่องทำงานอยู่ส่วน LED อีกดวงหนึ่งนั้นจะเป็น LED สีเขียว ใช้แสดงสถานการณ์ทำงานของเครื่อง ซึ่งเรียกว่า LED Status โดย LED Status นี้จะ

เกิดการกระพริบตามจังหวะของการรับส่งข้อมูลกันในแต่ละครั้ง โดยในสภาวะปรกตินั้น ถ้าเครื่องทำงานอยู่ใน RUN Mode หลอด LED Status จะดับอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล แต่ถ้าตัวเครื่องทำงานอยู่ใน Setup Mode หลอด LED Status จะติดอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล โดยโหมดการทำงานของ ET-RF24G V1.0 จะมีอยู่ด้วยกัน 2 โหมด คือ

2.1.1 การใช้งานเครื่อง ET-RF24G V1.0 ใน Run Mode

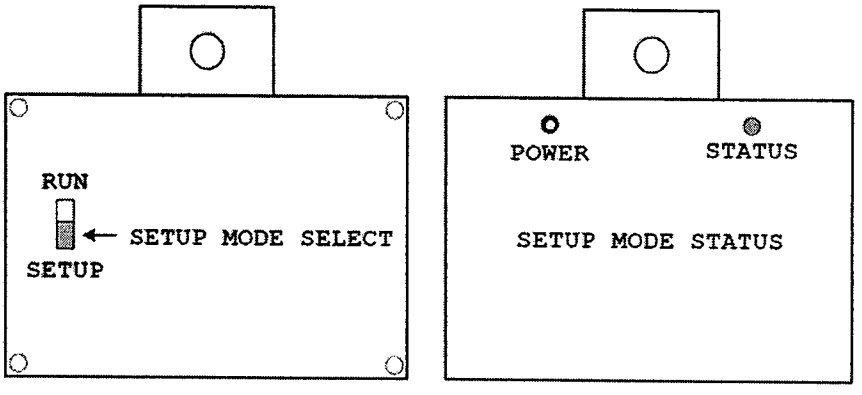
การใช้งานใน Run Mode ซึ่งเป็นโหมดของการใช้งานตามปรกติของเครื่อง โดยเมื่อเครื่อง ET-RF24G V1.0 เข้าทำงานในโหมดนี้แล้ว จะสังเกตเห็นหลอดไฟแสดงสถานะการทำงาน หรือ LED Status ดับอยู่ แต่เมื่อมีการรับ หรือ ส่งข้อมูลเกิดขึ้น สถานะการทำงานของ LED Status จึงจะกระพริบตามจังหวะของการรับส่งข้อมูลนั้นๆ แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED Status จะดับอยู่ตลอดเวลา



ภาพที่ 8 แสดงการเลือกโหมดการทำงานสำหรับใช้งานปรกติ (Run Mode)

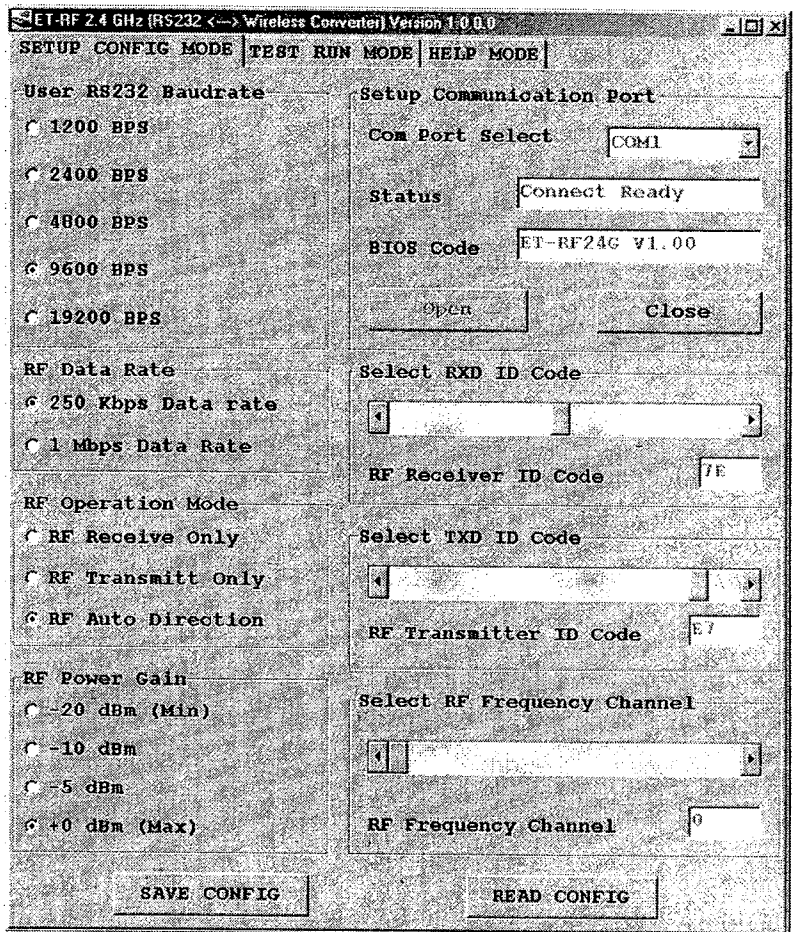
2.1.2 การใช้งานเครื่อง ET-RF24G V1.0 ใน Setup Mode

การใช้งานเครื่อง ET-RF24G V1.0 ใน Setup Mode ซึ่งเป็นโหมดสำหรับใช้กำหนดค่า Configuration ต่างๆ สำหรับควบคุมการทำงานของเครื่อง ET-RF24G V1.0 ที่จะใช้ในขณะที่เครื่องทำงานอยู่ใน Run Mode โดยในการ Setup ค่า Configuration ต่างๆ นั้นจะกระทำร่วมกับโปรแกรม “ET_RF24G_V1.EXE” ของ อีทีที ซึ่งเมื่อเครื่อง ET-RF24G V1.0 เข้าทำงานในโหมด Setup แล้วจะสังเกตเห็นหลอดไฟแสดงสถานะการทำงาน หรือ LED Status ติดสว่างค้างอยู่ตลอดเวลา แต่เมื่อมีการสั่งอ่านหรือเขียนข้อมูลกับบอร์ดสถานะการทำงานของ LED Status จึงจะกระพริบตามจังหวะของการรับส่งข้อมูล แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED Status จะติดค้างอยู่ตลอดเวลา



ภาพที่ 9 แสดงการเลือกโหมดการทำงานสำหรับกำหนดค่า Configuration (Setup Mode)

ซึ่งการกำหนดค่า Configuration ให้กับ ET-RF24G V1.0 นั้น จะต้องกระทำในขณะที่ตัวเครื่อง ทำงานอยู่ใน Setup Mode เท่านั้น (เลือก Switch กำหนดโหมดไว้ทางด้าน Setup แล้วจ่ายไฟให้เครื่องเริ่มต้นทำงาน) โดยค่าของ Configuration ต่างๆ นั้นจะถูกใช้สำหรับเป็นเงื่อนไขในการทำงานของ ET-RF24G V1.0 ในขณะที่อยู่ใน Run Mode ดังนั้น ก่อนการเริ่มต้นใช้งานเครื่องในครั้งแรกนั้น จึงจำเป็นอย่างยิ่งที่จะต้องทำการกำหนดค่าของ Configuration ต่างๆ ให้ถูกต้องและตรงกับความต้องการที่จะใช้งานเสียก่อน โดยเมื่อทำการกำหนดค่าตัวเลือกต่างๆของ Configuration เรียบร้อยแล้วก็สามารถเปลี่ยนโหมดการทำงานของตัวเครื่องกลับเป็น Run Mode พร้อมกับการปิดไฟที่จ่ายให้กับตัวเครื่อง (Power-Off) ชั่วขณะหนึ่ง จากนั้นจึงเริ่มต้นจ่ายไฟให้กับตัวเครื่องใหม่ (Power-On) ก็สามารถใช้งาน ET-RF24G V1.0 ตามค่าของ Configuration ที่กำหนดไว้แล้วได้ทันที โดยค่าตัวเลือกต่าง ๆ ของ Configuration ที่ได้กำหนดไว้แล้วจะถูกเก็บไว้ภายในตัวเครื่องอย่างถาวร ถึงแม้ว่าจะไม่ได้ทำการจ่ายไฟให้กับตัวเครื่องแล้วก็ตาม ดังนั้นเมื่อทำการกำหนดค่า Configuration ต่างๆ เรียบร้อยแล้ว ถ้าไม่มีการเปลี่ยนแปลงเงื่อนไขการทำงานของตัวเครื่องต่างไปจากเงื่อนไขเดิมที่ได้กำหนดไว้แล้ว ก็ไม่จำเป็นต้องทำการกำหนดค่า Configuration ใหม่อีกแต่อย่างใด โดยทุกๆ ครั้งที่เริ่มต้นจ่ายไฟเข้าเครื่องในครั้งแรกนั้น การทำงานของ ET-RF24G V1.0 จะเป็นไปตามเงื่อนไขที่กำหนดไว้ใน Configuration เสมอทุกๆ ครั้ง โดยคุณสมบัติของ Configuration ต่างๆ นั้นมีดังนี้



ภาพที่ 10 แสดงรูปโปรแกรมที่ใช้สำหรับกำหนดค่า Configuration ของ ET-RF24G V1.0

• User RS232 Baud rate ใช้สำหรับกำหนดค่าความเร็วในการรับส่งข้อมูลทางด้านRS232 ของเครื่อง ในขณะที่ทำงานอยู่ใน Run Mode ซึ่งสามารถกำหนดได้ 5 ค่าคือ

- 1200 BPS
- 2400 BPS
- 4800 BPS
- 9600 BPS
- 19200 BPS

• RF DATA Rate

ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของ ET-RF24G V1.0 ซึ่งจะต้องกำหนดให้เครื่อง ET-RF24G V1.0 ทุก ๆ ตัวที่จะนำมาใช้ติดต่อสื่อสารกัน มีค่าอัตราความเร็วในการรับส่งข้อมูลด้าน RF หรือ RF Data Rate นี้มีค่าเท่ากันทั้งหมด ซึ่งถ้ากำหนด ค่า

ความเร็วต่างกันจะไม่สามารถรับส่งข้อมูลกันได้ ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้ จะมีผลต่อระยะทางการรับส่งข้อมูลด้วย ซึ่งถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Kbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้น โดยค่า RF Data Rate สามารถกำหนดได้ 2 ค่า คือ

- 250 Kbps
- 1 Mbps

• **RF Operation Mode**

ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V1.0 ซึ่งสามารถกำหนดหน้าที่ การทำงานได้ 3 แบบ ด้วยกันคือ

- RF Receive Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS-232 ตลอดเวลา

- RF Transmit Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RS232 จากขา RX เพื่อเปลี่ยนเป็นข้อมูลแบบ GFSK (Guassian Frequency Shift Keying) และส่งออกไปทางด้าน RF ตลอดเวลา

- RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการทำงานนี้เครื่อง ET-RF24G V1.0 จะรอตรวจสอบข้อมูลทั้งจากด้าน RS-232 และด้าน RF อยู่ตลอดเวลาโดยถ้าได้รับข้อมูลจากด้าน RS-232 ก็จะทำการแปลงแล้วส่งออกทางด้าน RF จากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรอรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะแปลงเป็นข้อมูลแล้วส่งออกไปทางด้าน RS-232 โดยอัตโนมัติ

• **RF Power Gain**

เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0 dBm เป็นค่ากำลังส่งสูงสุด ส่วน –20 dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับคือ

- -20 dBm (กำลังส่งต่ำสุด)
- -10 dBm
- -5 dBm
- +0 dBm (กำลังส่งสูงสุด)

• **RXD ID Code**

เป็นรหัส ID Code ของเครื่อง ET-RF24G V1.0 ในโหมดของการรับข้อมูลจาก RF โดยเมื่อเครื่อง ET-RF24G V1.0 ด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข ID Code ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ โดยเมื่อเครื่อง ET-RF24G V1.0 ที่อยู่ทางด้านรับทำการรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอาเฉพาะส่วนของข้อมูลที่รับเข้ามาได้ เพื่อเปลี่ยนเป็นข้อมูลแบบ RS-232 แล้วส่งออกไปทางด้าน TX ของ RS232 แต่ถ้ารหัส ID Code ที่รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้ เครื่อง ET-RF24G V1.0 จะทิ้งข้อมูลชุดนั้นไปทันที โดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบขอเลขฐานสิบหก (00H-FFH)

• **TXD ID Code**

เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหา โดยที่เครื่อง ET-RF24G V1.0 ที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไป ทางด้าน RF โดยรหัสของ TXD ID Code นี้หมายถึง รหัส RXD ID Code ของฝ่ายรับที่ต้องการ ส่งข้อมูลไปหานั่นเอง โดยค่า TXD ID Code นั้นสามารถกำหนดได้ 256 ค่า ในรูปแบบของเลขฐานสิบหก (00H-FFH)

• **RF Frequency Channel**

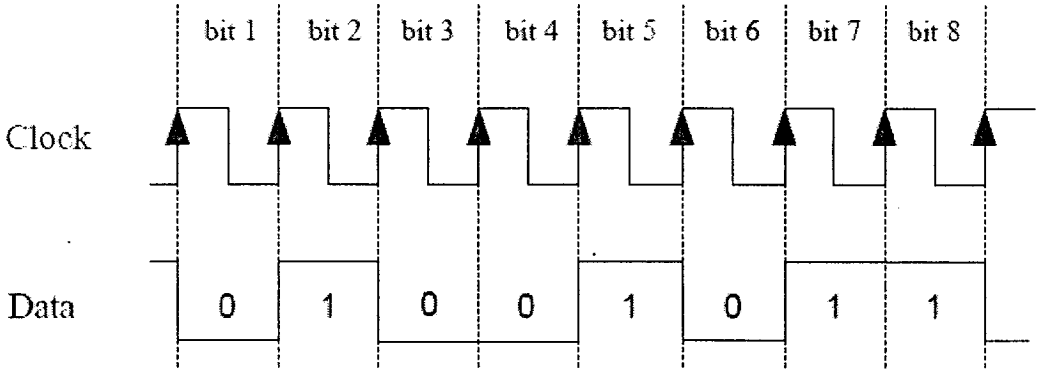
เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124) โดยการที่เครื่อง ET-RF24G V1.0 จะทำการรับส่งข้อมูลกันได้นั้นจะต้องกำหนดช่องความถี่ที่ตรงกัน และใช้อัตราความเร็ว RF Data Rate ที่เท่ากันด้วย ซึ่งที่สามารถเลือกกำหนดช่องความถี่ RF Frequency Channel ได้นั้น จะมีประโยชน์เป็นอย่างมาก ในกรณีที่มีการใช้งานเครื่อง ET-RF24G V1.0 จำนวนหลาย ๆ กลุ่ม ในบริเวณพื้นที่ใกล้เคียงกัน โดยให้กำหนดช่องความถี่ของ ET-RF24G V1.0 กลุ่มที่จะสื่อสารข้อมูลร่วมกันไว้ ที่ช่องความถี่เดียวกัน ส่วนกลุ่มอื่นๆ ก็ให้เลือกกำหนดช่องความถี่ที่แตกต่างกันออกไป เพื่อลดปัญหาการรบกวนกัน

2.2 พอร์ตอนุกรม (Serial Port)

ในการสื่อสารข้อมูลนั้นพอร์ตอนุกรมจะมีความเร็วในการสื่อสารที่ช้ากว่าแบบขนาน เพราะการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตขนานสามารถส่งข้อมูลทีละหลายๆ บิตพร้อมๆกันได้ แต่ข้อดีของการสื่อสารข้อมูลแบบอนุกรมคือ สามารถส่งข้อมูลได้ในระยะทางที่ไกลกว่าการสื่อสารข้อมูลแบบขนาน และใช้สายสัญญาณที่น้อยกว่าการสื่อสารข้อมูลแบบขนานประเภทของการสื่อสารแบบอนุกรมแบ่งตามลักษณะสัญญาณในการส่งได้ 2 แบบ คือแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous)

2.2.1 การสื่อสารแบบซิงโครนัส (Synchronous)

สำหรับการสื่อสารแบบซิงโครนัสนี้จะใช้สัญญาณนาฬิกาควบคุมการรับส่งสัญญาณ เช่น สายคีย์บอร์ดคอมพิวเตอร์ โดยจะมีสายสัญญาณเส้นหนึ่งเป็นสายสัญญาณนาฬิกา ส่วนอีกเส้นหนึ่งเป็นสายของข้อมูล (และมักมีสายกราวด์ด้วย) ซึ่งลักษณะสัญญาณของการสื่อสารแบบซิงโครนัส ได้แสดงดังภาพที่ 11



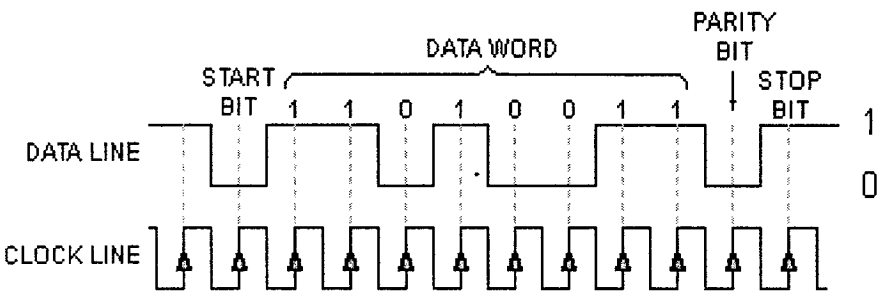
ภาพที่ 11 ลักษณะการสื่อสารแบบซิงโครนัส

สำหรับการสื่อสารแบบซิงโครนัสนี้เหมาะสำหรับการทำงานในระยะใกล้ ข้อมูลที่จะส่งมีไม่มากนัก เพราะถ้าระยะทางไกลขึ้นจะทำให้สัญญาณนาฬิกามีปัญหา อีกทั้งต้องมีสายหลายเส้นทำให้สิ้นเปลืองมากขึ้น

2.2.2 การสื่อสารแบบอะซิงโครนัส

สำหรับการสื่อสารแบบอะซิงโครนัสนั้น จะใช้สายข้อมูลเพียงตัวเดียว แต่จะใช้รูปแบบการส่งข้อมูล หรือ Bit Pattern เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นส่วนตรวจสอบความถูกต้องของข้อมูล และส่วนไหนเป็นส่วนปิดท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาคส่งและภาครับ ซึ่งจะมีอุปกรณ์

พิเศษที่ชื่อว่า UART หรือ Universal Asynchronous Receiver / Transmitter คอยควบคุมการรับและส่งข้อมูล ซึ่งลักษณะสัญญาณของการสื่อสารแบบอะซิงโครนัส ได้แสดงดังภาพที่ 12



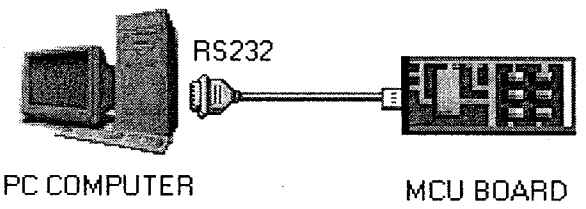
ภาพที่ 12 ลักษณะสัญญาณของการสื่อสารแบบอะซิงโครนัส

2.3 การสื่อสารข้อมูลตามมาตรฐาน RS-232

มาตรฐาน RS-232 ได้จัดพิมพ์ขึ้นเมื่อ ปี ค.ศ. 1969 โดยสถาบันแห่งหนึ่งในสหรัฐอเมริกา ซึ่งเป็นที่ยอมรับกันมีชื่อว่า Electronic Industrials Association หรือเรียกสั้น ๆ ว่า EIA โดยได้กำหนดเป็นมาตรฐาน สำหรับการติดต่อสื่อสารข้อมูลระหว่าง Data Terminal Equipment (DTE) เป็นอุปกรณ์สำหรับส่งข้อมูล (output) และ Data Communication Equipment (DCE) เป็นอุปกรณ์สำหรับรับข้อมูล(Input)

DCE: อุปกรณ์ที่มีฟังก์ชั่นการทำงานต่าง ๆ ที่ให้เกิดการเชื่อมต่อ ทำให้การเชื่อมต่อยังดำรงต่อไปและยุติการเชื่อมต่อ นอกจากนี้ยังใช้เปลี่ยนลักษณะของสัญญาณและสร้างรหัสสัญญาณต่าง ๆ ที่จำเป็นต้องใช้ในการสื่อสารข้อมูลระหว่าง DTA (Data Terminal Equipment) และ Data Circuit โดย DCE อาจเป็นส่วนใดส่วนหนึ่งของคอมพิวเตอร์หรือไม่ก็ได้

DTE: เป็นอุปกรณ์ที่ประกอบไปด้วยตัวส่งข้อมูล (Data Source) หรือตัวรับข้อมูล (Data Link) หรือเป็นทั้งตัวส่งและตัวรับข้อมูลก็ได้ เป็นอุปกรณ์ที่ประกอบด้วย Function Unit ต่อไปนี้ Control Logic, Buffer Store และอุปกรณ์อินพุตหรือเอาท์พุตจำนวนหนึ่งตัวหรือมากกว่าก็ได้ หรือรวมเครื่องคอมพิวเตอร์เข้าไปด้วยกันก็ได้ DTE อาจจะรวมส่วน Error Control, Synchronization และความสามารถในการบ่งหรือระบุว่าต้องการเกี่ยวข้องกับอุปกรณ์ตัวใด (Station Identification Capability) เข้าไปด้วยกันก็ได้



ภาพที่ 13 การใช้งานพอร์ตอนุกรม RS232

การสื่อสารแบบอนุกรม นับว่ามีความสำคัญ ต่อการใช้งานไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้แป้นพิมพ์และจอภาพของ PC เป็นอินพุตและเอาต์พุตในการติดต่อหรือควบคุมไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อย เพียง 3 เส้นเท่านั้น คือ

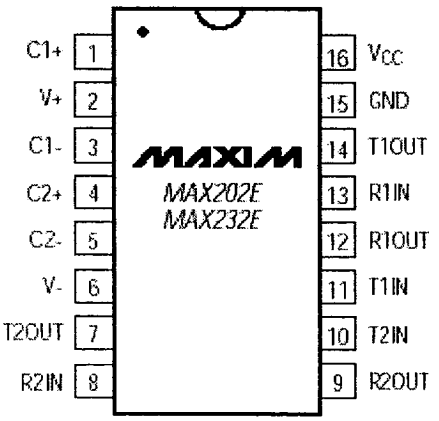
- สายส่งสัญญาณ TX
- สายรับสัญญาณ RX
- สายกราวด์ GND

โดยปกติพอร์ตอนุกรม RS-232 จะสามารถต่อสายได้ยาวกว่า 50 ฟุตโดยประมาณ ขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทางและปริมาณสัญญาณรบกวน

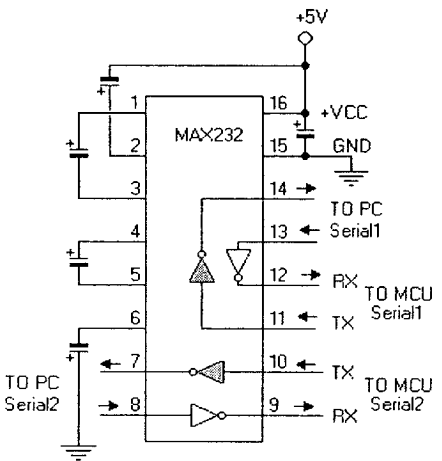
2.4 การเชื่อมต่อพอร์ตอนุกรม (Serial Port) กับไมโครคอนโทรลเลอร์ (แบบ Hardware)

การใช้งานพอร์ตอนุกรมกับไมโครคอนโทรลเลอร์ตระกูล PIC-18F458 จะมีวงจรสื่อสารผ่านพอร์ตอนุกรมแบบยูอาร์ท (UART: Universal Asynchronous Receiver Transmitter) หรือ แบบFull Duplex คือ สามารถรับ-ส่งข้อมูล ได้พร้อมกันอยู่ 1 ชุด สำหรับไมโครคอนโทรลเลอร์ตระกูล PIC-18F458 จะเป็นขา RC7 (PIN-26) เป็นขารับข้อมูล (RX) และ RC6 (PIN-25) เป็นขาส่งข้อมูล (TX) สำหรับการเชื่อมต่อกับคอมพิวเตอร์นั้นจะใช้ไอซีเบอร์ MAX232 เป็นตัวปรับระดับสัญญาณจาก MCU เป็นระดับสัญญาณมาตรฐานของ RS-232

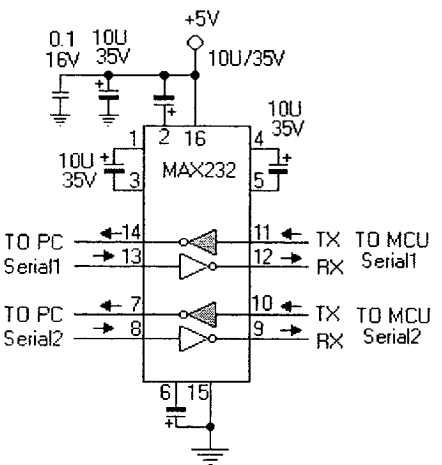
MAX232 เป็นไอซีที่แปลงระดับสัญญาณของ RS-232 มาเป็นระดับ TTL และในทำนองเดียวกันก็แปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณ RS-232



ภาพที่ 14 แสดงตำแหน่งขาของไอซี MAX232



ภาพที่ 15 ภาพวงจรภายใน



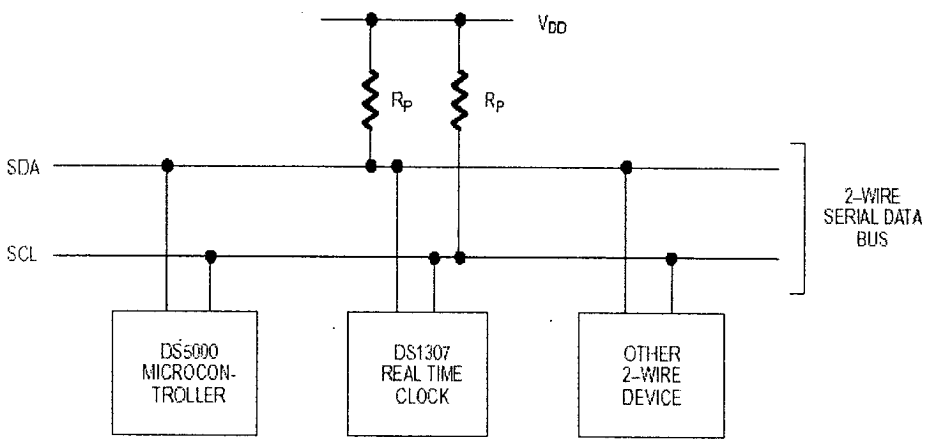
ภาพที่ 16 การต่อใช้งาน MAX232 กับ ไมโครคอนโทรลเลอร์

2.5 การเชื่อมต่ออุปกรณ์แบบ I^2C

I^2C BUS ย่อมาจาก Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้นๆว่า I^2C BUS (ไอ-แสคว-ซี-บัส) เป็นการสื่อสารอนุกรมแบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ Serial Data (SDA) และสาย Serial Clock (SCL) ซึ่งสามารถ เชื่อมต่ออุปกรณ์ จำนวนหลาย ๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

2.6 การเชื่อมต่ออุปกรณ์แบบ I^2C BUS

I^2C BUS ใช้สายสัญญาณ 2 เส้น คือ SCL, SDA สำหรับติดกับอุปกรณ์แบบ 2 ทิศทาง โดยที่ขาสัญญาณทั้ง 2 จะต้องต่อกับตัวต้านทานแบบ Pull Up 2-10 K เนื่องจากเอาต์พุตมีลักษณะเป็น แบบ Open Drain หรือเป็นแบบ Open Collector เพื่อให้เอาต์พุตเชื่อมต่อกันได้หลายตัว

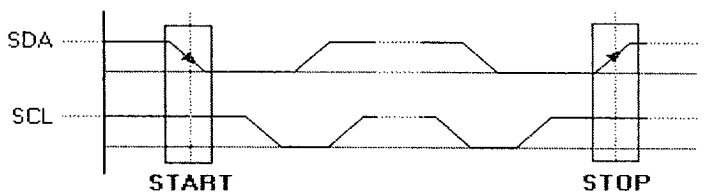


ภาพที่ 17 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I^2C BUS

2.6.1 การเขียน - อ่านข้อมูลกับอุปกรณ์แบบ I^2C BUS

การรับ-ส่งข้อมูลแบบ I^2C BUS MCU จะเริ่มต้นการส่งข้อมูลด้วยการส่งสถานะเริ่มต้น (Start Conditions) เพื่อแสดงการขอใช้บัสแล้วตามด้วย รหัสควบคุม (Control Byte) ซึ่งประกอบ ด้วยรหัส ประจำตัวอุปกรณ์ Device ID, Device Address ,และ Mode ในการเขียนหรืออ่านข้อมูลเมื่ออุปกรณ์ รับทราบว่า MCU ต้องการ จะติดต่อด้วยก็ต้องส่งสถานะรับรู้(Acknowledge) หรือ แจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้องและเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่ง สถานะสิ้นสุด (Stop Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัส สถานะบัสว่าง คือเมื่อบัสไม่ได้ถูกใช้งาน ทั้ง SCL และ SDA จะเป็น 1 ทั้งคู่

2.6.2 การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I^2C Bus (Start and Stop Conditions)



ภาพที่ 18 การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I^2C BUS

- ลักษณะการกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I^2C BUS

เมื่อต้องการส่งข้อมูล MCU จะต้องส่งสถานะเริ่มต้น (Start Conditions) คือให้ SDA เปลี่ยนจาก 1 มาเป็น 0 ในขณะที่ SCL มีค่าเป็น 1 และ เมื่อสิ้นสุดการการใช้บัส MCU จะต้องส่งสถานะสิ้นสุด (Stop Conditions) คือให้ SDA เปลี่ยนจาก0 มาเป็น 1ในขณะที่ SCL มีค่าเป็น 1

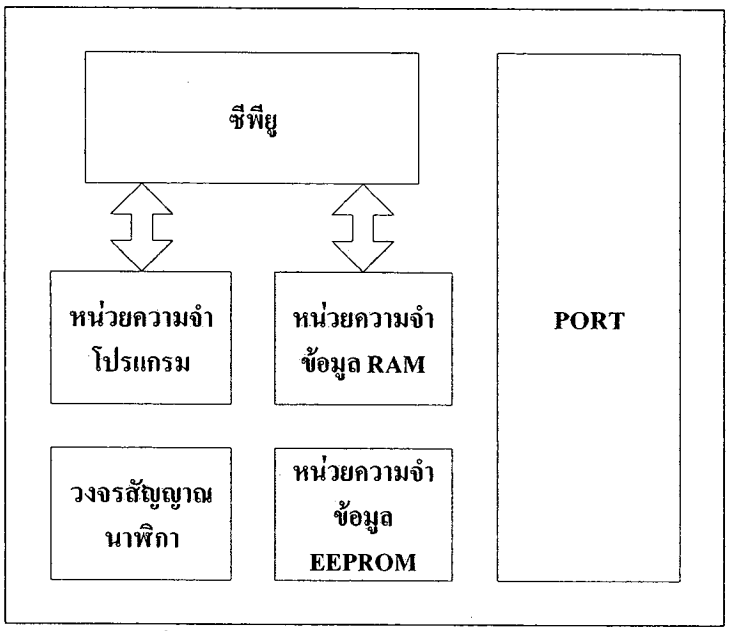
2.6.3 รหัสควบคุมของ I^2C BUS (Control Byte)

รหัสควบคุมของ I^2C BUS ประกอบด้วยรหัสประจำตัวของอุปกรณ์ (Device ID) ประกอบด้วยบิต 1-7 และบิต 0 เป็นบิตควบคุมการเขียนอ่านรหัสประจำตัวของอุปกรณ์ ประกอบด้วยรหัสประจำตัวจากผู้ผลิต Product ID 4 บิต (บิต 4-7) ที่เปลี่ยนแปลงแก้ไขไม่ได้ และ Device Address 3 บิต (บิต 1-3) ซึ่งผู้ใช้สามารถกำหนดเองได้ รวมแล้วเป็นรหัส 7 บิต ใช้ระบุตัวอุปกรณ์ ที่ต่ออยู่บนบัส จะมีค่าซ้ำกันไม่ได้

2.6.4 ช่วงเวลารับส่งบิตข้อมูลของ I^2C BUS

- สภาวะการรับ-ส่งข้อมูล จะกระทำในขณะที่ขา SCL เป็น 1
- สภาวะการเปลี่ยนแปลงข้อมูล จะกระทำในขณะที่ขา SCL เป็น 0

3. โครงสร้างของไมโครคอนโทรลเลอร์



ภาพที่ 19 โครงสร้างของไมโครคอนโทรลเลอร์

3.1 โครงสร้างของไมโครคอนโทรลเลอร์

จากภาพที่ 2 – 10 โครงสร้างของไมโครคอนโทรลเลอร์ ประกอบไปด้วย 5 ส่วน สำคัญคือ

3.1.1 หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) ทำหน้าที่ในการประมวลผลคำสั่งที่ได้รับจากหน่วยความจำ

3.1.2 หน่วยความจำ (Memory) ซึ่งแบ่งออกเป็น 2 ส่วนคือ หน่วยความจำเก็บโปรแกรม (program memory) ทำหน้าที่คล้ายๆกับฮาร์ดดิสก์ในคอมพิวเตอร์ ข้อมูลไม่สูญหายแม้ไม่มีไฟเลี้ยง และหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดาษทดในการคำนวณของซีพียูและเป็นที่พักข้อมูลในการทำงานชั่วคราว ข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยงคล้ายกับหน่วยความจำแรม (Ram) ในคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่หน่วยความจำจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายเมื่อไม่มีไฟเลี้ยง และเป็นแบบอีอีพรอม(EEPROM : Erasable Electrically Programmable Read-Only Memory)ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3.1.3 ส่วนติดต่ออุปกรณ์ภายนอกหรือเรียกว่าพอร์ต (Port) มีด้วยกัน 2 ลักษณะ คือ พอร์ตรับสัญญาณ หรือพอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output

Port) ส่วนนี้มีความสำคัญมาก เนื่องจากใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก และอุปกรณ์ภายนอกเหล่านั้นนั่นเอง ที่เป็นสื่อกลางในการติดต่อกับมนุษย์ ยกตัวอย่างพอร์ตอินพุตใช้ต่อกับสวิตซ์เพื่อรับข้อมูลที่ผู้ใช้งานกดป้อนเข้ามา ซึ่งเหมือนกับการใช้คีย์บอร์ดในการป้อนข้อความ เข้าเครื่องคอมพิวเตอร์ พอร์ตเอาต์พุตใช้ต่อกับลำโพงเพื่อขับเสียง ต่อกับหลอดไฟเพื่อแสดงผลต่อกับมอเตอร์เพื่อควบคุมการหมุน ต่อกับหน่วยความจำเพื่อเพิ่มพื้นที่ในการเก็บข้อมูล หากเปรียบเทียบกับคอมพิวเตอร์ พอร์ตเอาต์พุตก็คือส่วนที่ต่อกับเครื่องพิมพ์สำหรับพิมพ์ข้อมูลออกมาและส่วนที่ต่อกับจอมอนิเตอร์เพื่อแสดงภาพ เป็นต้น

3.1.4 เส้นทางสัญญาณหรือบัส (Bus) การติดต่อแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียู หน่วยความจำและพอร์ต จะกระทำบนสายสัญญาณจำนวนมาก เรียกว่า เส้นทางสัญญาณหรือบัส โดยแบ่งเป็นบัสข้อมูล (Data bus),บัสแอดเดรส (Address bus) และบัสควบคุม (Control bus) บัสข้อมูลเป็นสายสัญญาณที่บรรจุข้อมูล สำหรับการประมวลผลทั้งหมดขนาดของบัสจะขึ้นอยู่กับความสามารถในการประมวลผลของซีพียู และเทคโนโลยีของไมโครคอนโทรลเลอร์ตัวนั้นๆ สำหรับในงานทั่วไป ขนาดของบัสข้อมูลคือ 8 บิต และในปัจจุบันมีการพัฒนาไปถึง 16,32 และ 64 บิตแล้วบัสแอดเดรสเป็นสายสัญญาณที่บรรจุค่าตำแหน่งของหน่วยความจำ โดยการติดต่อกับหน่วยความจำนั้น ซีพียูต้องกำหนดตำแหน่งที่ต้องการอ่านหรือเขียนก่อน ซึ่งก็คือการกำหนดค่าแอดเดรส จำนวนสายสัญญาณของบัสแอดเดรสจึงต้องมีจำนวนมาก และถ้ายิ่งมีมากเท่าใด จะเป็นการแสดงถึงความจุของหน่วยความจำ ที่ไมโครคอนโทรลเลอร์ตัวนั้นสามารถติดต่อได้ สามารถคำนวณได้จากจำนวนแอดเดรสของหน่วยความจำ = 2^n โดยที่ n คือจำนวนสายสัญญาณ

3.1.5 วงจรกำเนิดสัญญาณนาฬิกาเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานทั้งหมดในไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับการกำหนดจังหวะโดยใช้สัญญาณนาฬิกาหากสัญญาณนาฬิกามีความถี่สูง จังหวะในการทำงานของไมโครคอนโทรลเลอร์ก็จะมีมากตาม ส่งผลให้ไมโครคอนโทรลเลอร์นั้นมีความเร็วในการประมวลผลสูงตามไปด้วย

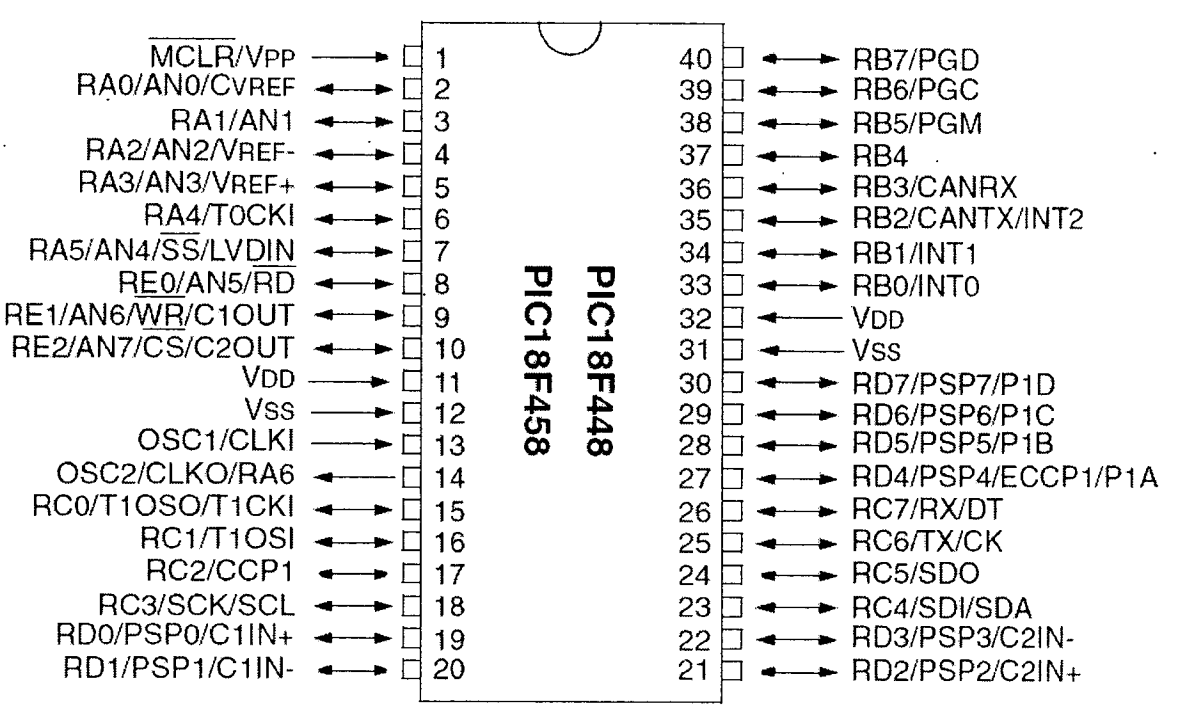
ตารางที่ 3 แสดงภาพรวมของ PIC16Fxxx/18Fxxx

ชิพ	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล		CAN Module	จำนวน I/O (บิต)	OSC (MHz)	Timer	PLL
		RAM	EEPROM					
16F84	1K Word	68	64	ไม่มี	13	4-10	1	ไม่มี
16F877	8K Word	368	256	ไม่มี	33	4-20	3	ไม่มี
18F442	16 KB	768	256	ไม่มี	34	40	4	มี
18F458	32 KB	1536	256	มี	34	40	4	มี

- 16F84/877 นั้น ใช้หน่วยความจำโปรแกรมขนาด 14 บิต ต่อ 1 คำ (Word)

- PLL (Phase Lock Loop) คือ วงจรที่สร้างความถี่ของสัญญาณนาฬิกาเป็น 4 เท่าจาก XTAL ดังนั้น ถ้าเลือกโหมด PLL ก็จะใช้ XTAL ได้ไม่เกิน 10 MHz

3.2 การจัดขาของไอซี



ภาพที่ 20 แสดงขาของไมโครคอนโทรลเลอร์ตระกูล PIC-18F458

จากภาพที่ 20 ขาของ PIC-18F458 กับ 18F448 จะสามารถดูได้จาก data sheet แต่ละขา จะมีหน้าที่แตกต่างกันไปซึ่งแยกออกเป็น พอร์ต A, พอร์ต B, พอร์ต C, พอร์ต D, พอร์ต E โดยพื้นฐานแล้วพอร์ตแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาท์พุตเป็นดิจิตอล ยกเว้นพอร์ต A และพอร์ต E ที่สามารถทำงานเป็นตัวรับสัญญาณอนาล็อกแปลงเป็นค่าดิจิตอล เพื่อนำมาวัดปริมาณทางฟิสิกส์ต่างๆ ที่เห็นได้อย่างชัดเจน คือ นำมาวัดความต่างศักย์

คุณสมบัติของ PIC18F458

- มีคำสั่งให้ใช้งาน 35 คำสั่ง
- คำสั่งหนึ่งๆใช้เวลาทำงาน 1 ถึง 2 Cycle
- ทำงานได้สูงสุดที่ความถี่สัญญาณนาฬิกา 40 MHz
- ทำงานแบบ Pipe-Line (มี 2 ท่อ) ทำให้ ณ เวลาหนึ่งทำงาน 2 อย่างพร้อมกัน
- หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 32 KB
- มีขนาดหน่วยความจำ (RAM) 1536 ไบต์
- มี EEPROM ขนาด 256 ไบต์
- ตอบสนองกับอินเตอร์รัปได้ทั้งหมด 14 แหล่ง
- มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
- มีระบบ Power On Reset, Power Up Timer, Oscillator Start-Up Timer
- มีระบบ Code Protection
- สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้
- สามารถโปรแกรมด้วยไฟ +5 VDC ได้
- ใช้การโปรแกรมแบบ In-Circuit Serial Programming
- ทำงานที่ไฟเลี้ยง 2 VDC ถึง 5.5 VDC
- Current Sink และ Current Source อยู่ที่ 25 mA
- มี Timer/Counter 4 ตัว
- มีโมดูล Capture/Compare/PWM อีก 2 ชุด
- มี A-D Converter แบบ 10 บิต จำนวน 8 ช่องนำเข้าในตัวเอง
- มีระบบ USART สำหรับต่อกับ การสื่อสารแบบ RS232 หรือดีกว่า
- มีระบบตรวจระดับไปเลี้ยง (Brown Out Reset)
- มี I/O พอร์ตทั้งหมด 5 พอร์ต แต่ละพอร์ตมีจำนวนบิตไม่เท่ากัน

3.3 สัญญาณนาฬิกา

ไมโครคอนโทรลเลอร์จะทำงานได้ต้องมีสัญญาณนาฬิกาให้กับตัวซึ่งในหนึ่งไซเคิล (Clock Bus) ของซีพียูจะประกอบไปด้วยสัญญาณนาฬิกาภายนอกจำนวน 4 ไซเคิล คือ Q1, Q2, Q3 และ Q4 ดังนั้นความถี่ที่ซีพียูประมวลผลต่อหนึ่งคำสั่งจะเท่ากับความถี่ ของสัญญาณนาฬิกาภายนอกหารด้วย 4 หรือหากจะพิจารณาความเร็วของไมโครคอนโทรลเลอร์ตระกูล PIC-18F458 สามารถประมวลผลต่อหนึ่งคำสั่งเท่ากับ 1/4 เท่าของความถี่ออสซิลเลเตอร์ภายนอก

3.4 โหมดสัญญาณนาฬิกา

PIC18F458 สามารถเลือกโหมดสัญญาณนาฬิกาเพื่อกำหนดสัญญาณการทำงานได้มากถึง 4 โหมด โดยการกำหนดที่บิด FOSC1 ในรีจิสเตอร์ Configuration Word ในการทำงานจะต้องเลือกโหมดหนึ่ง ดังรายละเอียด

3.4.1 โหมด LP (Low Power Crystal) ใช้กับคริสตอลหรือเซรามิกเรโซเนเตอร์พลังงานต่ำความถี่ 32KHz - 200KHz

3.4.2 โหมด LP (Crystal/Resonator) ใช้กับคริสตอลหรือเซรามิกเรโซเนเตอร์พลังงานต่ำความถี่ 200KHz - 4MHz

3.4.3 โหมด HS (High Speed Crystal/Resonator) ใช้กับคริสตอลหรือเซรามิกเรโซเนเตอร์พลังงานต่ำความถี่ 4MHz - 20MHz

3.4.4 โหมด RC สามารถกำหนดค่าความถี่ได้จากค่าความต้านทานและตัวเก็บประจุที่ต่อภาย นอกเข้ากับขา OSC1/CLKIN

การเลือกใช้ตัวเก็บประจุ (C1, C2) สำหรับวงจรรีโซเนเตอร์แบบเซรามิก จะคำนึงถึงความถี่ต่าง ๆ ที่ใช้ ดังแสดงในตารางที่ 4

ตารางที่ 4 แสดงค่าตัวเก็บประจุที่เหมาะสมกับความถี่ที่ใช้

Osc Type	Crystal Freq	Cap Range C1	Cap Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

ชิพแบบ XT เป็นออสซิลเลเตอร์คริสตอลแบบมาตรฐาน ซึ่งอาจต้องการคริสตอล แบบสตริปคัตท์ AT (AT Strip-Cut) เพื่อหลีกเลี่ยงการโอเวอร์ไดรฟ์ (Overdrive)

วิธีการเชื่อมต่ออุปกรณ์ RC เข้ากับไมโครคอนโทรลเลอร์ตระกูล PIC-18F458 สำหรับค่า Rext ที่น้อยกว่า 2.2 กิโลโอห์ม ทำให้สัญญาณออสซิลเลเตอร์ที่ได้อาจจะไม่คงที่หรือหยุดนิ่ง สำหรับค่า Rext ที่มีค่าสูงมากๆ (เช่น 1 เมกะโอห์ม) ออสซิลเลเตอร์จะมีความไวต่อสัญญาณรบกวน ความชื้นและสภาวะแวดล้อมภายนอก ดังนั้นควรจะใช้ค่า Rext ให้มีค่าอยู่ในช่วง 5 กิโลโอห์มถึง 100 กิโลโอห์ม ถึงแม้ว่าออสซิลเลเตอร์จะทำงานได้โดยไม่ต้องต่อตัวเก็บประจุภายนอก (Cext=0 pF) แต่ควรใส่ค่าตัวเก็บประจุที่มากกว่า 20 pF เพื่อลดสัญญาณรบกวนและให้สัญญาณมีความคงที่ ถ้าไม่มีตัวเก็บประจุหรือตัวเก็บประจุภายนอกมีค่าน้อยเกินไป จะทำให้ความถี่ออสซิลเลเตอร์มีการเปลี่ยนแปลงอย่างกะทันหัน เนื่องจากการเปลี่ยนแปลงที่ตัวเก็บประจุภายนอก เช่น ที่แผ่นวงจรพิมพ์บริเวณตัวเก็บประจุหรือตัวนำของตัวเก็บประจุ

3.5 หน้าที่ของพอร์ตที่ใช้งาน

- A0-RA3 และ RA5 จะใช้งานเป็น I/O ปกติ และทำหน้าที่เป็นขาอินพุตของสัญญาณอนาล็อก (AN0-AN4)

- RA4 เป็นขา I/O

- RA6/OSC2/CLKO ทำหน้าที่ในหลายส่วน คือ เป็นขา OSC2 และ CLKO จะนำมาใช้เป็นขาสัญญาณ I/O ได้ก็ต่อเมื่อใช้คริสตอลออสซิลเลเตอร์ แบบที่เดเป็นโมดูลสำเร็จสามารถต่อเข้ากับขา OSC1/CLKIN ได้เลย โดยที่ไม่ต้องต่อกับขา RA6/OSC2 ทำให้ขา RA6 ว่างและนำไปใช้เป็น I/O ได้

- RB0-RB7 สามารถใช้งานเป็น I/P แต่มีคุณสมบัติพิเศษ คือวงจรมีพูลอัพ (Pull-Up) ภายในและเป็นแหล่งกำเนิดสัญญาณอินเตอร์รัปต่างๆ ดังนี้

- RB0/INT0 เป็นขาสัญญาณอินเตอร์รัปภายนอก 0

- RB1/INT1 เป็นขาสัญญาณอินเตอร์รัปภายนอก 1

- RB2/INT2 เป็นขาสัญญาณอินเตอร์รัปภายนอก 2

- RB3/INT3 เป็นขาสัญญาณอินเตอร์รัปภายนอก 3

- RB4-RB7 เป็นขาที่สามารถกำเนิดสัญญาณอินเตอร์รัปได้

4. การพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์

การพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์นั้นจำเป็นที่จะต้องรู้หลักการดังต่อไปนี้

4.1 ภาษาซี (C)

ภาษาซีได้รับการพัฒนามาจากภาษาบีซีพีแอล (BCPL) และ บี (B) ซึ่งต่างก็เป็นภาษาโปรแกรมที่รู้จักข้อมูลเพียงรูปแบบเดียวคือ เป็นข้อมูลเวิร์ด (Word) ยังไม่สามารถจัดการกับข้อมูลแบบตัวอักษรได้ ในขณะเดียวกัน ผู้เขียนโปรแกรมหรือโปรแกรมเมอร์ต่างก็ยังมีความต้องการภาษาโปรแกรมคอมพิวเตอร์ที่โครงสร้างสามารถจัดการกับข้อมูล (Data) และตัวแปร (Variable) ในรูปแบบต่างๆได้เพิ่มมากขึ้น ทั้งข้อมูลที่เป็นเลขจำนวนเต็ม (Integer), ตัวเลขทศนิยม (Floating - point) หรือตัวอักษร (Character) จึงมีการเพิ่มเติมความสามารถเหล่านี้ลงในภาษาบี และกำหนดชื่อใหม่เป็น ภาษาซี

ภาษาซีได้รับการออกแบบมาให้ทำงานกับคอมพิวเตอร์ และเขียนโปรแกรมด้วยคำสั่งที่สามารถทำความเข้าใจได้ไม่ยาก จึงจัดโปรแกรมภาษาซีว่าเป็น ภาษาคอมพิวเตอร์ระดับกลาง เมื่อนำมาทำงานกับไมโครโปรเซสเซอร์ที่ทำงานด้วยภาษาเครื่อง จึงจำเป็นอย่างยิ่งที่จะต้องมี ตัวแปลภาษาหรือคอมไพเลอร์ (Compiler) เพื่อแปลภาษาซีนั้นเป็นภาษาเครื่อง

4.2 คอมไพเลอร์

การเขียนโปรแกรมด้วยภาษาซี เพื่อนำไปใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ นั้นในทางปฏิบัติจริง ๆ ไม่ใช่เนื้อโค้ดของโปรแกรมภาษาซี ที่ถูกนำลงไปบรรจุในหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ ตัวรหัสข้อมูลที่ใช้งานจริงนั้น ได้มาจากการแปลภาษาซี เป็นรหัสภาษาเครื่อง หรือ แมชีนโค้ดด้วยซอฟต์แวร์ที่เรียกกันว่า คอมไพเลอร์

การพัฒนาระบบงานของไมโครคอนโทรเลอร์ด้วยโปรแกรมภาษา C มีขั้นตอนโดยสรุปดังนี้

4.2.1 เขียนโปรแกรมภาษาซี ด้วยเท็กซ์เอดิเตอร์ (Text Editor) หรือพื้นที่สำหรับเขียนโปรแกรมในกรณีที่ซอฟต์แวร์นั้นจัดมาเป็นชุดในแบบไอดีอี (IDE)

4.2.2 คอมไพล์หรือแปลงภาษาซี เป็นภาษาแอสเซมบลี (Assembly) ของไมโครคอนโทรเลอร์ตัวนั้นๆ

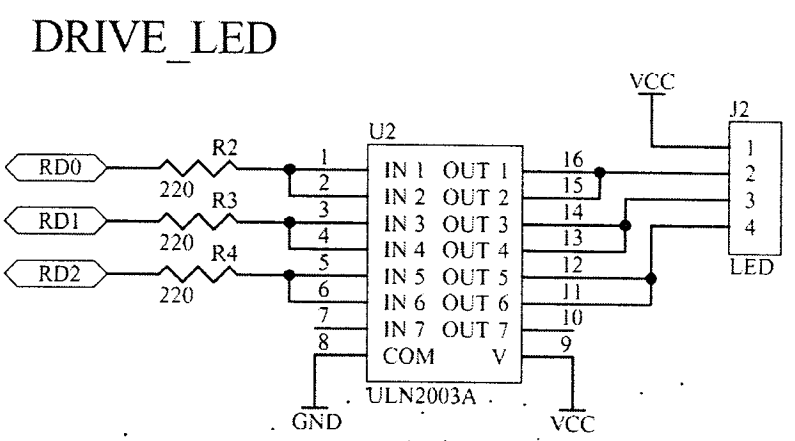
4.2.3 แอสเซมเบลอร์ (Assembler) จากภาษาแอสเซมบลีเป็นภาษาเครื่องหรือแมชีนโค้ดในรูปของเลขฐานสิบหก

4.2.4 ดาวน์โหลดซอร์สโค้ดที่ได้จากการแอสเซมเบลอร์ลงสู่หน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์

2.4.2.5 ทดลองและตรวจสอบการทำงาน หากไม่สมบูรณ์ให้กลับไปแก้ไขตั้งแต่ขั้นตอนที่ 1 อย่างไรก็ตาม ในขั้นตอนที่ 2 และ 3 ผู้ใช้งานจะไม่เห็นถึงกระบวนการทำงาน เนื่องจากซอฟต์แวร์แปลภาษาหรือคอมไพเลอร์ได้รวมการทำงานใน 2 ขั้นตอนนี้ไว้ด้วยกัน

5. การใช้งานวงจรขับ LED

ไอซีที่ใช้ในการขับโหลดกระแสสูงมักจะมีวงจรทางเอาท์พุตเป็นแบบคอลเล็กเตอร์เปิดทำให้สามารถใช้กับแรงดันสูง สำหรับไอซีขับ LED ที่ใช้คือ ไอซี ULN 2003 เป็นไอซีที่ภายในบรรจุอินเวอร์เตอร์เกต 7 ตัวใช้กับแรงดันได้สูงสุด +30 โวลต์กระแสเอาท์พุตสูงสุดในแต่ละขาเท่ากับ 500 มิลลิแอมป์ทั้งนี้ขึ้นอยู่กับความสามารถในการจ่ายกระแสของแหล่งจ่ายไฟด้วย นอกจากนั้นยังมีการต่อไดโอดป้องกันแรงดันไหลย้อนกลับจากอุปกรณ์เอาท์พุต



ภาพที่ 21 รูปวงจรขับ LED