บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 บทนำ

ชุดทดลองหรือชุดฝึกปฏิบัติการ เป็นเครื่องมือที่สำคัญในการเพิ่มพูนทักษะของผู้เรียน การที่ผู้เรียนได้เรียนรู้เทคโนโลยีใหม่ๆ ย่อมส่งผลให้มีความรู้และทักษะมากเพียงพอที่จะนำไปประกอบวิชาชีพ หรือพัฒนาตนเอง ซึ่งเทคโนโลยี RFID ที่ผู้วิจัยให้ความสนใจนั้นเป็นเทคโนโลยีสำคัญที่เป็นยุทธศาสตร์การวิจัยระดับชาติ ทั้งนี้เนื่องจากความสะดวกในการใช้งาน เพราะเป็นเทคโนโลยีไร้สาย ปลอดภัย เพราะมีการกำหนดรหัสเฉพาะตัว โดยเฉพาะในแวดวงอุตสาหกรรม การจัดเก็บสินค้าคงคลัง ระบบขนส่งสินค้า ระบบการขายสินค้า ระบบรักษาความปลอดภัย

2.2 เทคโนโลยีอาร์เอฟไอดี (RFID)

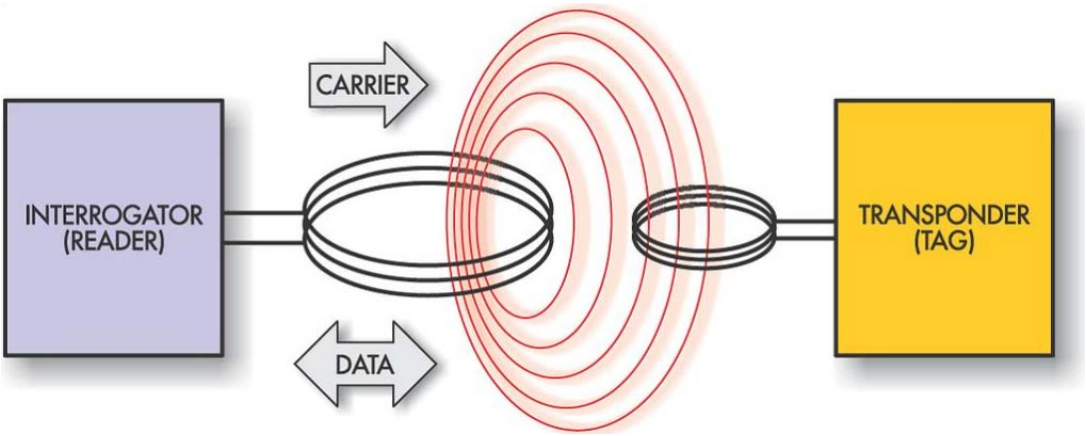
RFID ย่อมาจาก Radio Frequency Identification เป็นระบบฉลากที่ได้ถูกพัฒนามาตั้งแต่ปี ค.ศ. 1980 โดยที่อุปกรณ์ RFID ที่มีการประดิษฐ์ขึ้นใช้งานเป็นครั้งแรกนั้น เป็นผลงานของ Leon Theremin ซึ่งสร้างให้กับรัฐบาลของประเทศรัสเซียในปี ค.ศ. 1945 ซึ่งอุปกรณ์ที่สร้างขึ้นมาในเวลานั้นทำหน้าที่เป็นเครื่องมือดักจับสัญญาณ ไม่ได้ทำหน้าที่เป็นตัวระบุเอกลักษณ์อย่างที่ใช้งานกันอยู่ในปัจจุบัน RFID ในปัจจุบันมีลักษณะเป็นป้ายอิเล็กทรอนิกส์ (RFID Tag) ที่สามารถอ่านค่าได้โดยผ่านคลื่นวิทยุจากระยะห่าง เพื่อตรวจ ติดตามและบันทึกข้อมูลที่ติดอยู่กับป้าย ซึ่งนำไปฝังไว้ในหรือติดอยู่กับวัตถุต่างๆ เช่น ผลิตภัณฑ์ กล่อง หรือสิ่งของใดๆ สามารถติดตามข้อมูลของวัตถุ 1 ชิ้นว่า คืออะไร ผลิตที่ไหน ใครเป็นผู้ผลิต ผลิตอย่างไร ผลิตวันไหน และเมื่อไหร่ ประกอบไปด้วยชิ้นส่วนกี่ชิ้น และแต่ละชิ้นมาจากที่ไหน รวมทั้งตำแหน่งที่ตั้งของวัตถุนั้นๆ ในปัจจุบันว่าอยู่ส่วนใดในโลก โดยไม่จำเป็นต้องอาศัยการสัมผัส (Contact-Less) หรือต้องเห็นวัตถุนั้นๆ ก่อน ทำงานโดยใช้เครื่องอ่านที่สื่อสารกับป้ายด้วยคลื่นวิทยุในการอ่านและเขียนข้อมูล RFID มีข้อได้เปรียบเหนือกว่าระบบบาร์โค้ด คือ

- มีความละเอียด และสามารถบรรจุข้อมูลได้มากกว่า ซึ่งทำให้สามารถแยกความแตกต่างของสินค้าแต่ละชิ้นแม้จะเป็นเดียวกันก็ตาม
- ความเร็วในการอ่านข้อมูลจากแถบ RFID เร็วกว่าการอ่านข้อมูลจากแถบบาร์โค้ดหลายสิบเท่า
- สามารถอ่านข้อมูลได้พร้อมกันหลาย ๆ แถบ RFID
- สามารถส่งข้อมูลไปยังเครื่องรับได้โดยไม่จำเป็นต้องนำไปจ่อในมุมที่เหมาะสมเหมือนการใช้เครื่องอ่านบาร์โค้ด (Non-Line of Sight)

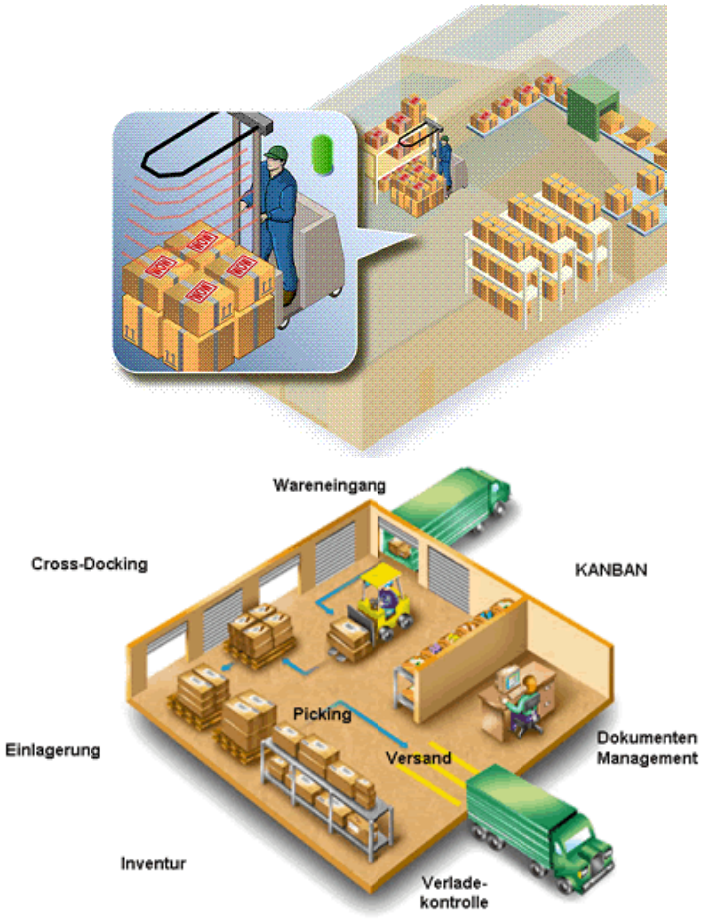
- ค่า เฉลี่ยของความถูกต้องของการอ่านข้อมูลด้วยเทคโนโลยี RFID นั้นจะอยู่ที่ประมาณ 99.5 เปอร์เซ็นต์ ขณะที่ความถูกต้องของการอ่านข้อมูลด้วยระบบบาร์โค้ดอยู่ที่ 80 เปอร์เซ็นต์
- สามารถ เขียนทับข้อมูลได้ จึงทำให้สามารถนำกลับมาใช้ใหม่ได้ซึ่งจะลดต้นทุนของการผลิตป้ายสินค้า ซึ่งคิดเป็นประมาณ 5% ของรายรับของบริษัท
- สามารถขจัดปัญหาที่เกิดขึ้นจากการอ่านข้อมูลซ้ำที่อาจเกิดขึ้นจากระบบบาร์โค้ด
- ความเสียหายของป้ายชื่อ (Tag) น้อยกว่าเนื่องจากไม่จำเป็นต้องติดไว้ภายนอกบรรจุภัณฑ์
- ระบบความปลอดภัยสูงกว่า ยากต่อการปลอมแปลงและลอกเลียนแบบ
- ทนทานต่อความเปียกชื้น แรงสั่นสะเทือน การกระทบกระแทก

2.2.1 ลักษณะการทำงานของระบบ RFID

หลักการเบื้องต้นของ RFID เป็นวิธีการตรวจสอบข้อมูลหรือระบุข้อมูลแบบอัตโนมัติ โดยทำงานผ่านการรับสัญญาณจากป้ายอาร์เอฟไอดี (RFID Tag) เข้าสู่ตัวส่งสัญญาณ ผ่านทางคลื่นวิทยุ Tag ของอาร์เอฟไอดีโดยปกติจะมีขนาดเล็กซึ่งสามารถติดตั้งเข้ากับผลิตภัณฑ์สินค้า สัตว์ บุคคลได้ ซึ่งเมื่อตัวส่งสัญญาณส่งคลื่นวิทยุไป เมื่อTagได้รับสัญญาณ ก็จะส่งสัญญาณกลับพร้อมกับข้อมูลที่เก็บไว้ใน Tag ดังรูปที่ 2.1



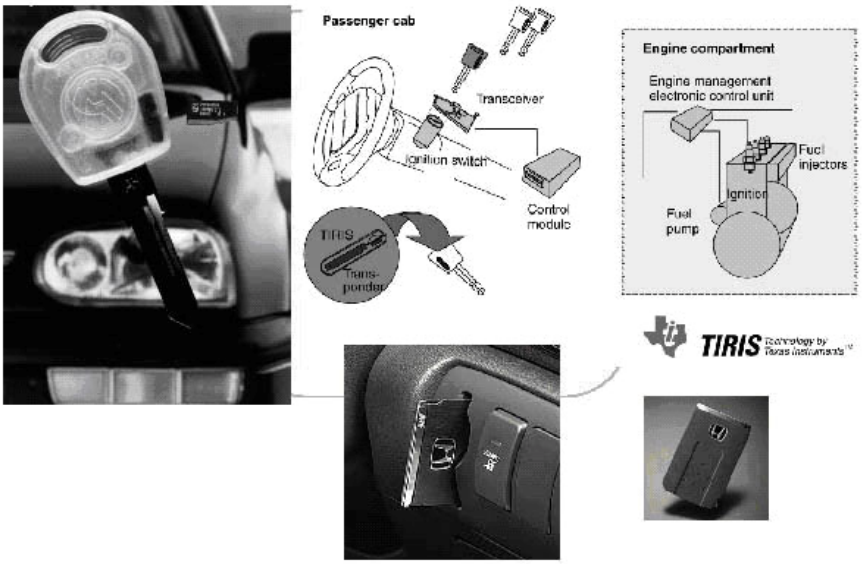
รูปที่ 2.1 ระบบการทำงานของ RFID [1]



รูปที่ 2.2 การใช้ RFID ในระบบจัดเก็บและขนส่งสินค้า [2]

ระบบดังกล่าวสามารถนำมาประยุกต์ใช้ได้หลากหลาย อาทิ การค้าปลีก, การค้าส่ง, การผลิต, การรักษาความปลอดภัย, การทดแทนระบบบาร์โค้ด การเก็บประวัติ และติดตามสัตว์ เป็นต้น โดยตัวอย่างที่เห็นได้ชัดเจนในปัจจุบัน ได้แก่ บัตร Proximity ที่ใช้เป็นบัตรพนักงาน หรือเข้าออกสถานที่ และบัตรโดยสารรถไฟฟ้าใต้ดิน เป็นต้น

รูปที่ 2.3 การใช้ RFID ในระบบห้องสมุด [3] และระบบตั๋วโดยสารรถไฟฟ้า

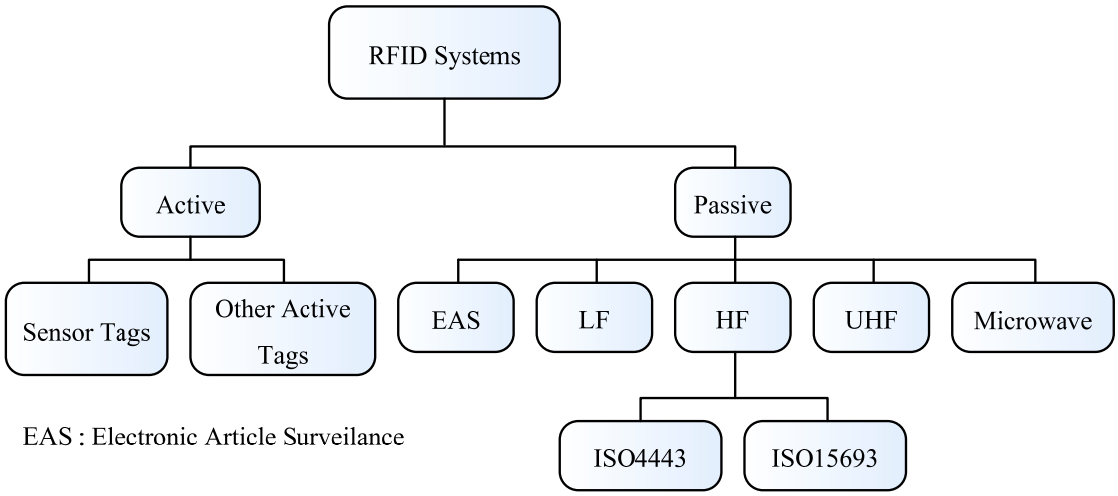


รูปที่ 2.4 การใช้ RFID ในระบบรักษาความปลอดภัยรถยนต์

องค์ประกอบในระบบ RFID มี 2 ส่วนหลักคือ ป้ายอิเล็กทรอนิกส์หรือ Tag มีชื่อเรียกเป็นทางการว่า Transponder, Transmitter & Responder เป็นฉลากที่ผนึกติดกับวัตถุ ใช้บันทึกข้อมูลเกี่ยวกับวัตถุชิ้นนั้น โครงสร้างภายใน Tag ประกอบด้วย ชิปและขดลวด ซึ่งทำหน้าที่เหมือนเสาอากาศที่คอยรับ-ส่งสัญญาณ Tag มี 2 ชนิดใหญ่ๆ คือ Passive RFID Tag ซึ่งไม่จำเป็นต้องรับแหล่งจ่ายไฟใดๆ เพราะมีวงจรกำเนิดไฟฟ้าเหนี่ยวนำขนาดเล็กเป็นแหล่งจ่ายไฟในตัวอยู่แล้ว ระยะการสื่อสารข้อมูลที่ทำได้สูงสุด 1.5 เมตร มีหน่วยความจำขนาดเล็ก (ทั่วไปประมาณ 32–128 บิต) มีขนาดเล็กและน้ำหนักเบา ราคาต่อหน่วยต่ำ และ Active RFID Tag ชนิดนี้จะใช้แหล่งจ่ายไฟจากแบตเตอรี่ขนาดเล็ก มีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกะไบต์ มีระยะการสื่อสารข้อมูลที่ทำได้สูงสุดถึง 6 เมตร แม้ว่า Tag ชนิดนี้จะมีข้อดีอยู่หลายข้อแต่ก็มีข้อเสียอยู่ด้วยเหมือนกัน เช่น มีราคาต่อหน่วยแพง มีขนาดค่อนข้างใหญ่ และมีระยะเวลาในการทำงานที่จำกัด

เครื่องอ่าน (Reader) มีชื่อเรียกเป็นทางการว่า Transceiver, Transmitter & Receiver หน้าที่ของเครื่องอ่านคือ การเชื่อมต่อเพื่ออ่านข้อมูลจาก Tag

การ พัฒนาระบบ RFID มิได้มีจุดประสงค์เพื่อมาแทนที่ระบบอื่นที่มีการพัฒนามาก่อนหน้า เช่น ระบบบาร์โค้ด แต่เป็นการเสริมจุดอ่อนต่างๆ ของระบบอื่นในประเทศไทยมีแนวโน้มการใช้เทคโนโลยี RFID ในหลากหลายด้านทั้งใช้ในด้านการขนส่ง (บัตรทางด่วน บัตรโดยสารรถไฟฟ้า) ด้านการปศุสัตว์ (การให้อาหาร การติดตามโรค) ใช้กับเอกสารราชการ (บัตรพนักงาน บัตรจอดรถ) และการใช้ RFID เพื่อเพิ่มขีดความสามารถในด้าน Logistics โดยใช้ผนึกอิเล็กทรอนิกส์ติด RFID ปิดล็อคตู้คอนเทนเนอร์เพื่อสะดวกในการติดตาม บริหารจัดการขนส่ง ด้านการแพทย์ (บันทึกประวัติการรักษาผู้ป่วย) หรือแม้แต่ในงานของห้องสมุด



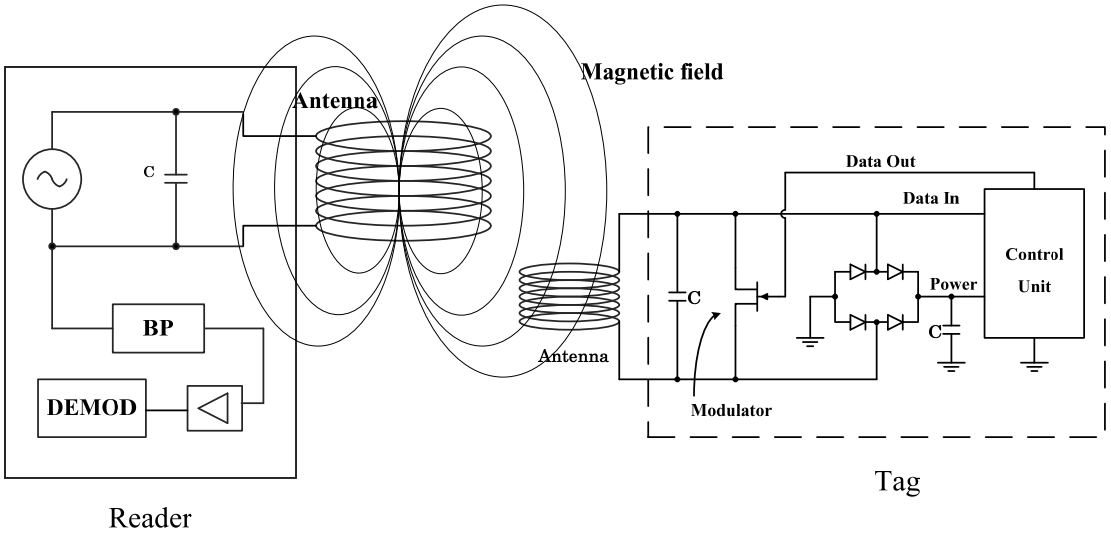
รูปที่ 2.5 แผนผังจำแนกชนิดของระบบ RFID [4]

ระบบ RFID สามารถจำแนกมาตรฐานออกเป็นดังรูปที่ 2.5 โดยจำแนกเป็น 2 กลุ่มคือ Active และ Passive ประโยชน์การนำไปใช้งานก็แตกต่างกันออกไป ตามรูปแบบงาน ความถี่ ความปลอดภัยของข้อมูล การจำแนกชนิดของ Tag ก็อาจจำแนกได้หลายแบบขึ้นอยู่กับมุมมอง ดังนี้

1) จำแนกตามชนิด แบ่งได้เป็น
   - แบบ Passive (ไม่ใช้แบตเตอรี่ ขนาดเล็ก เก็บข้อมูลได้น้อย ระยะอ่านต่ำ)
   - แบบ Active (ใช้แบตเตอรี่ ขนาดใหญ่กว่า เก็บข้อมูลได้มากกว่า ระยะอ่านไกล)
2) จำแนกตามการเก็บข้อมูล
   - แบบอ่านอย่างเดียว
   - แบบเขียนครั้งเดียว
   - แบบเขียน/อ่าน ได้หลายครั้ง
3) จำแนกตามความถี่ใช้งาน
   - Low Frequency (LF) 125-135 KHz
   - Very High Frequency (VHF) 13.56 MHz
   - Ultra High Frequency (UHF) 860 MHz
   - Microwave 2.4 GHz

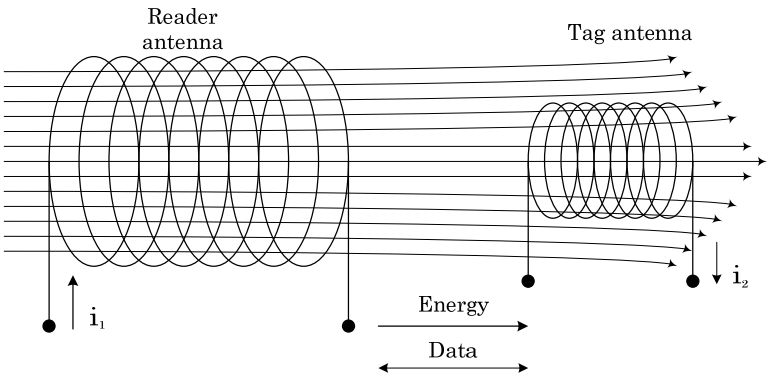
2.2.2 หลักการทำงานของระบบ RFID

หลักการสื่อสารของระบบ RFID อย่างที่ได้กล่าวมาข้างต้นว่าเป็นระบบที่ใช้คลื่นความถี่วิทยุในการสื่อสารระหว่าง Tag กับเครื่องอ่าน โดยการนำสัญญาณข้อมูลมามอดูเลต (Modulation) กับคลื่นพาห์ (Carrier) ที่เป็นคลื่นความถี่วิทยุโดยมีสายอากาศ (Antenna) อยู่ภายในเครื่องอ่านเป็นตัวรับ



รูปที่ 2.6 การสื่อสารข้อมูลระหว่าง Tag กับเครื่องอ่าน

1) หลักการทำงานของ Tag แบบพาสซีฟ ชนิดนี้ทำงานได้ไม่ต้องอาศัยแหล่งจ่ายไฟภายนอก โดยทั่วไปการทำงานของ Tag แบบพาสซีฟ ในย่านความถี่ต่ำ(LF) และย่านความถี่สูง (HF) จะใช้หลักการคู่ควบแบบเหนี่ยวนำ(Inductive Coupling) ซึ่งเกิดจากการอยู่ใกล้กันของขดลวดระหว่างTagกับเครื่องอ่านที่กำลังทำงาน ทำให้เกิดการถ่ายเทพลังงานจากเครื่องอ่านไปยังไมโครชิปผ่านสนามแม่เหล็กที่เกิดขึ้น เมื่อไมโครชิปได้รับพลังงานจะส่งข้อมูลในหน่วยความจำที่ผ่านการมอดูเลตกับคลื่นพาห์แล้วออกมาทางสายอากาศ เครื่องอ่านจะตรวจจับความเปลี่ยนแปลงของคลื่นพาห์แปลงออกมาเป็นข้อมูลแล้วทำการถอดรหัสเพื่อนำข้อมูลไปใช้งานต่อไป ลักษณะเงื่อนไขในการทำงานแบบชักพาทำให้การอ่านข้อมูลทำได้ไม่ไกลมากนัก แสดงดังรูปที่ 2.7

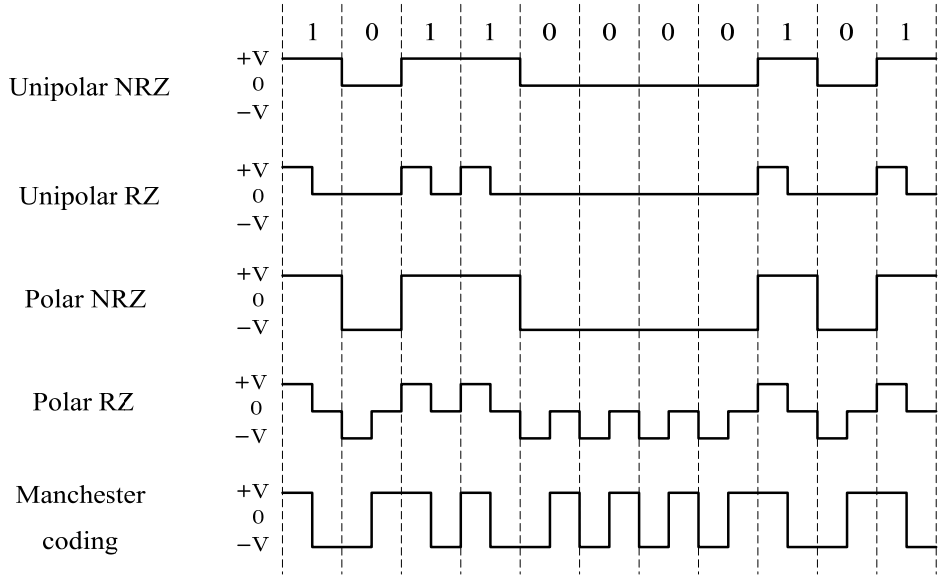


รูปที่ 2.7 สนามแม่เหล็กจากกระบวนการควบคู่แบบเหนี่ยวนำ

ส่วนในระบบความถี่ UHF จะใช้การใช้สนามแม่เหล็กไฟฟ้าด้วยการควบคู่แบบแพร่กระจาย (Propagation Coupling) โดยที่สายอากาศของเครื่องอ่านจะทำหน้าที่ส่งพลังงานแม่เหล็กไฟฟ้าในรูปคลื่นออกมา ซึ่งเมื่อTagได้รับสัญญาณ Tagก็จะทำงานโดยการส่งสัญญาณรหัสภายในTagไปยังเครื่องอ่าน ทั้งนี้การทำงานที่ความถี่ต่างกัน จะทำให้คุณสมบัติการรับส่งสัญญาณต่างกันรวมทั้งประสิทธิภาพโดยรวมก็ขึ้นอยู่กับเงื่อนไขอื่นๆ ด้วยเช่น ขนาดของสายอากาศหรือสัญญาณรบกวน

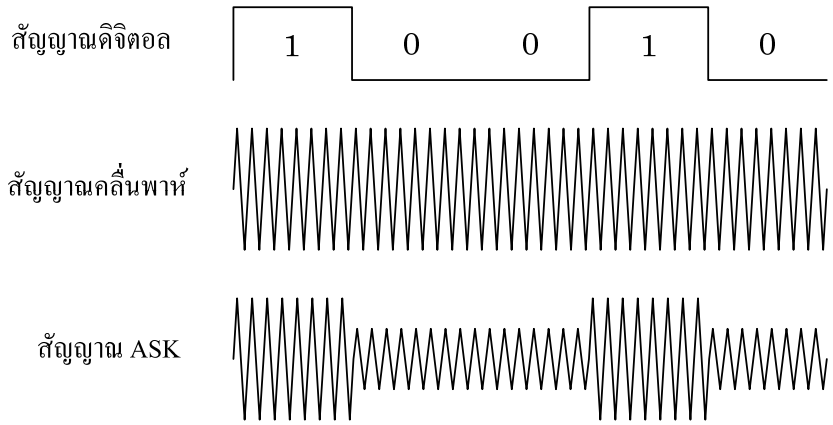
2) หลักการทำงานของ Tag แบบแอ็กทีฟ ชนิดนี้ต้องอาศัยแหล่งจ่ายไฟจากแบตเตอรี่ภายนอก เพื่อจ่ายพลังงานให้กับวงจรภายในโดย Tag แบบแอ็กทีฟจะทำการส่งข้อมูลออกมาก็ต่อเมื่อได้รับสัญญาณจากเครื่องอ่านและเครื่องบอกตำแหน่ง ซึ่งสัญญาณจะถูกปล่อยออกมาเป็นระยะๆ ตลอดเวลาการใช้งาน Tag ชนิดนี้ อาจพบได้ในระบบต่างๆ เช่น ระบบจ่ายเงินทางด่วน หรือด่านตรวจ

3) หลักการเบื้องต้นในการรับและส่งข้อมูลระหว่างTagและเครื่องอ่าน โดยทั่วไปเป็นไปตามกระบวนการทางด้านการสื่อสารระบบดิจิตอล นั่นคือ การเตรียมข้อมูลดิจิตอลที่จะส่งผ่านโดยการทำการเข้ารหัสให้อยู่ในเหมาะสมสำหรับการส่งผ่านช่องสัญญาณ(Channel) คำว่าเหมาะสมหมายถึงว่าสัญญาณมีโอกาสจะถูกส่งผ่านช่องสัญญาณที่มีสัญญาณรบกวน(Noise) โดยมีค่าผิดพลาดน้อยที่สุดเท่าที่เป็นไปได้ ซึ่งวิธีการเข้ารหัสนั้นมีได้หลายแบบแสดงดังรูปที่ 2.8 ซึ่งการเลือกใช้นั้นขึ้นอยู่กับช่องสัญญาณที่จะส่งผ่าน ตัวอย่างเทคนิคการเข้ารหัส เช่น การเข้ารหัสสัญญาณแบบ NRZ, RZ หรือการเข้ารหัสแบบ Manchester เป็นต้น



รูปที่ 2.8 การเข้ารหัสแบบต่างๆ

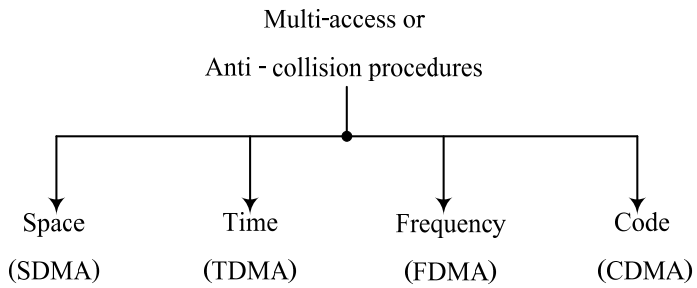
หลังจากการเข้ารหัสสัญญาณแล้ว สัญญาณจะถูกทำการมอดูเลตกับคลื่นพาห์ย่านที่สูงกว่าเพื่อทำการส่งรับข้อมูลในย่านนั้นๆ ตัวอย่างเช่นในการมอดูเลตแบบ ASK (Amplitude Shift Keying) ค่าแอมปลิจูดของคลื่นพาหะจะถูกเปลี่ยนเป็นรหัสเลขฐานสอง แสดงดังรูปที่ 2.9



รูปที่ 2.9 การมอดูเลตแบบ ASK

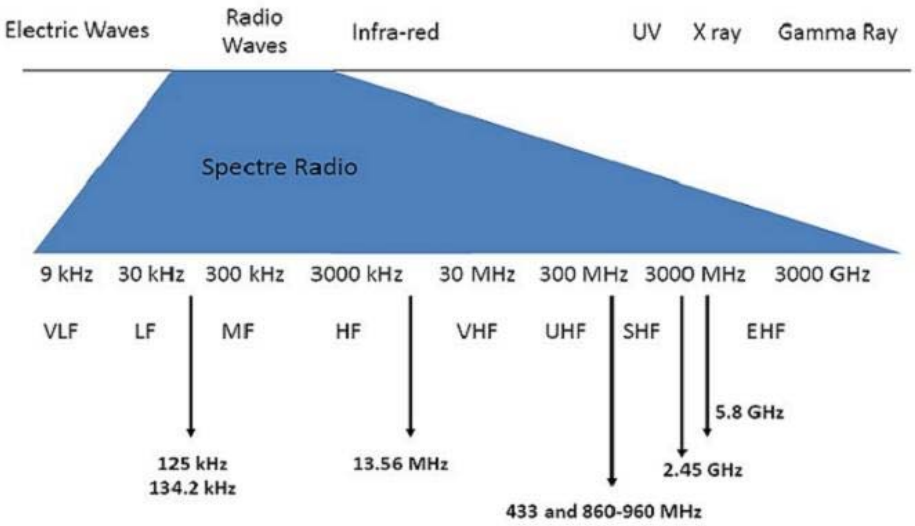
4) การป้องกันการชนกันของสัญญาณข้อมูล (Anti-collision)

เมื่อมี Tag หลายๆ อันเข้ามาอยู่ใกล้เครื่องอ่าน Tag แต่ละตัวจะมีพลังงานเพียงพอที่จะส่งข้อมูลของตัวเองมาที่เครื่องอ่านพร้อมๆ กันทำให้เครื่องอ่านไม่สามารถแยกแยะข้อมูลที่ส่งมาได้ ซึ่งเราเรียกปรากฏการณ์นี้ว่า การชนกันของข้อมูล (Collision) วิธีการแก้ไขโดยการทำ การเพิ่มฟังก์ชั่นป้องกันการชนกันบนTagและเครื่องอ่าน (Anti-Collision) ซึ่งจะมีหลายเทคนิค เช่น จัดคิวการอ่าน Tag โดยทำเป็นช่วงเวลาสั้นๆ เมื่อ Tag ถูกอ่านแล้วจะไม่มีการอ่านซ้ำอีก เช่น เทคนิค SDMA, TDMA, FDMA, CDMA หรือเทคนิคขั้นสูงจะใช้ FTDMA และการกระโดดความถี่ (Frequency Hopping) เข้ามาช่วย เทคนิคต่างๆ ที่ใช้ป้องกันการชนกัน (Collision) ของสัญญาณข้อมูล



รูปที่ 2.10 เทคนิคการป้องกันการชนกันของข้อมูล

1) คลื่นพาหะและมาตรฐานของระบบ RFID



รูปที่ 2.11 แถบคลื่นความถี่ใช้งานของระบบ RFID [5]

ในปัจจุบันคลื่นพาหะที่ใช้งานกันในระบบ RFID จะอยู่ในย่านความถี่พลเรือน (Industrial Scientific Medical: ISM) ซึ่งเป็นย่านความถี่ที่กำหนดในการใช้งานในเชิงการแพทย์ วิทยาศาสตร์ และอุตสาหกรรมสามารถใช้งานได้โดยไม่ตรงกับย่านความถี่ที่ใช้งานในการ สื่อสาร โดยทั่วไปมี 4 ย่านความถี่ใช้งาน คือสำหรับคลื่นพาหะที่ใช้กันในระบบ RFID อาจแบ่งออกได้เป็น 4 ย่านใหญ่ๆ แสดงดังรูปที่ 2.11 ได้แก่

ก) ย่านความถี่ต่ำ (Low Frequency : LF)

ข) ย่านความถี่สูง (High Frequency : HF)

ค) ย่านความถี่สูงยิ่ง (Ultra High Frequency : UHF)

ง) ย่านความถี่ไมโครเวฟ (Microwave frequency)

ตารางที่ 2.1 ระยะการสื่อสารข้อมูลแต่ละความถี่ของระบบ RFID

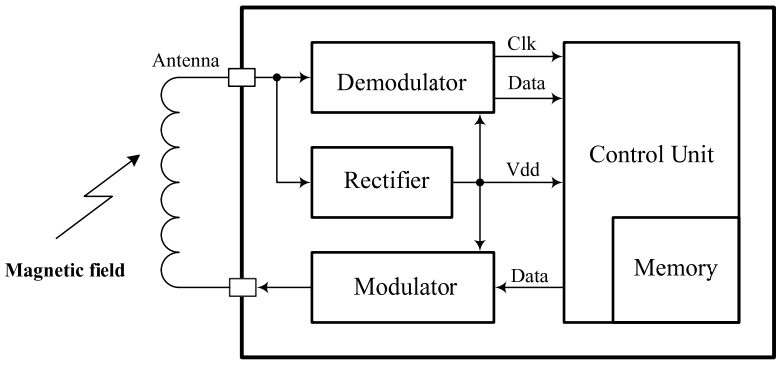
ความถี่	ระยะที่อ่านได้
125 kHz ถึง 134 kHz	น้อยกว่า 1 เมตร (10 เซนติเมตร)
13.56 MHz	น้อยกว่า 1.5 เมตร (ประมาณ 1 เมตร)
860 MHz ถึง 960 MHz	2 เมตร ถึง 5 เมตร 1 เมตร ถึง 100 เมตร (Tag แบบแอ็กทีฟ)
2.45 GHz	น้อยกว่า 1 เมตร (Tag แบบพาสทีฟ) 1 เมตร ถึง 15 เมตร (Tag แบบแอ็กทีฟ)

สองย่านความถี่แรกจะเหมาะสำหรับงานที่มีระยะการสื่อสารข้อมูลในระยะใกล้ โดยย่านความถี่ต่ำ (LF) 125kHz และ 134kHz ซึ่งนิยมใช้สำหรับการควบคุมการเข้าออกพื้นที่และการลงทะเบียนสัตว์ ย่านความถี่สูง (HF) 13.56MHz นิยมใช้กับบัตรอเนกประสงค์แบบไร้สัมผัสและหนังสือเดินทางอิเล็กทรอนิกส์ ส่วนย่านความถี่สูงยิ่ง(UHF) 860MHz และ 960MHz ถูกนำมาใช้กับงานที่มีระยะการสื่อสารข้อมูลในระยะไกล เช่น ระบบเก็บค่าทางด่วน ระบบขนส่งสินค้า เป็นต้น

2.2.3 RFID Tag

1) Passive Tag

สามารถทำงานได้โดยไม่ใช้แหล่งจ่ายพลังงานไฟฟ้าภายนอก เพราะภายใน Tag มีวงจรกำเนิดไฟฟ้าขนาดเล็ก เป็นแหล่งจ่ายพลังงานไฟฟ้าในตัว การผลิตพลังงานไฟฟ้าจะอาศัยสนามแม่เหล็กไฟฟ้าจากเครื่องอ่าน ทำให้ Tag ชนิดนี้มีระยะการอ่านข้อมูลได้ไม่ไกลมากนัก ระยะการอ่านสูงสุดอยู่ที่ประมาณ 1 เมตร ทั้งนี้ขึ้นอยู่กับกำลังของเครื่องส่งและคลื่นความถี่วิทยุที่ใช้ Tag ชนิดนี้มีหน่วยความจำน้อย โดยทั่วไปอยู่ที่ 16 ถึง 1,024 ไบต์ มีขนาดเล็กและน้ำหนักเบา ราคาค่อนข้างถูก โครงสร้างภายใน Tag ประกอบ 3 ภาคส่วนหลักๆ ส่วนแรก คือภาคการควบคุมการรับส่งสัญญาณวิทยุที่ประกอบด้วยสัญญาณข้อมูลส่วนนี้ประกอบด้วย ภาคมอดูเลต (Modulation) และภาคดีมอดูเลต (Demodulation) เป็นภาคทำหน้าที่ผสมหรือแยกสัญญาณข้อมูลกับคลื่นพาห์(Carrier) ส่วนที่สองคือหน่วยควบคุม (Control Unit) หน่วยนี้รับหน้าที่จัดจัดการเกี่ยวกับกระบวนการทางดิจิตอลทั้งหมด รวมทั้งจัดการเกี่ยวกับหน่วยความจำ(Memory) หน่วยความจำอาจเป็นแบบแรม (RAM) รอม (ROM) หรืออีอีพรอม(EEPROM) ขึ้นอยู่กับการเลือกใช้งานส่วนสุดท้ายคือวงจรที่ทำหน้าที่จ่ายพลังงานให้กับTag แบบพาสซีฟแสดงดังรูปที่ 2.12 อย่างไรก็ตามโครงสร้างภายในของ Tag ต่างรุ่นกัน บางครั้งก็อาจมีไม่ครบทุกส่วน ซึ่งรายละเอียดโครงสร้างตลอดจนรายละเอียดในการทำงานก็สามารถสืบค้นจากข้อมูลของผู้ผลิต

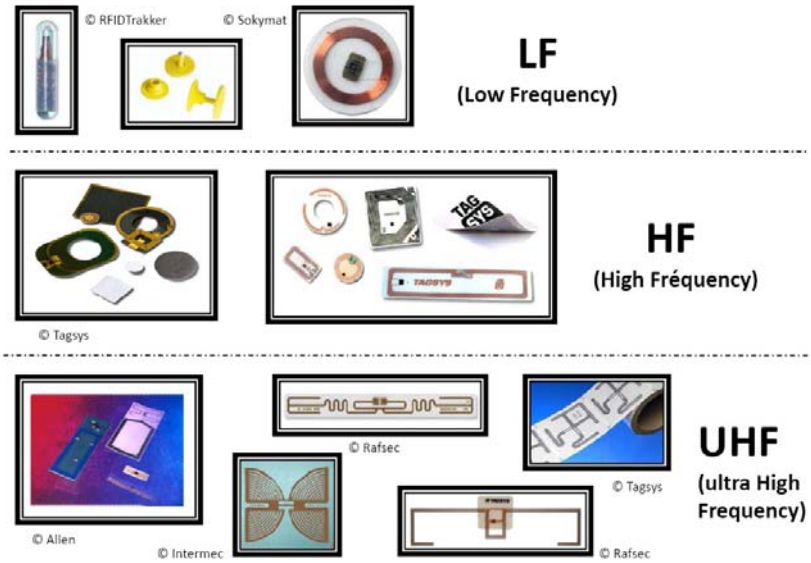


รูปที่ 2.12 โครงสร้างภายในของ Tag แบบพาสซีฟ

2) Active Tag

Tag แบบแอคทีฟ ต่างจากแบบพาสซีฟ คือมีแบตเตอรี่อยู่ภายในตัวเป็นแหล่งจ่ายไฟขนาดเล็ก เพื่อป้อนพลังงานไฟฟ้าให้ Tag โดยปกติการที่ต้องใช้แบตเตอรี่จึงทำให้ Tag ชนิดแอคทีฟมีอายุการใช้งานจำกัดตามอายุของแบตเตอรี่ เมื่อแบตเตอรี่หมด ไม่สามารถนำกลับมาใช้ใหม่ได้ เนื่องจากจะมีการซีลที่ตัว Tag จึงไม่สามารถเปลี่ยนแบตเตอรี่ได้ ถ้าสามารถออกแบบวงจรของ Tag ให้กินกระแสไฟน้อยๆ ก็อาจจะมีอายุการใช้งานนานนับสิบปี Tag ชนิดนี้จะมีหน่วยความจำภายในขนาดใหญ่ มีกำลังส่งสูงและระยะการรับส่งข้อมูลไกล ทำงานในบริเวณที่มีสัญญาณรบกวนได้ดี

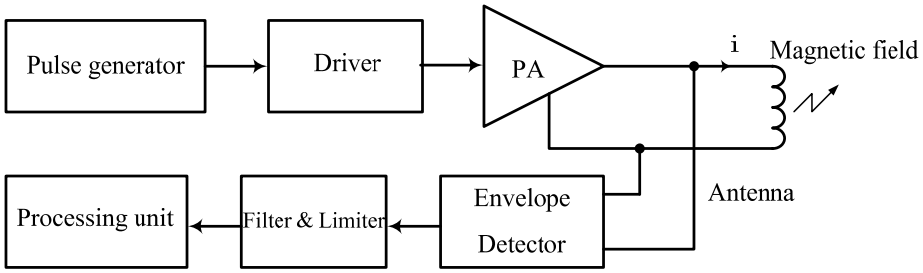
รูปที่ 2.13 Active Tag [6]



รูปที่ 2.14 ลักษณะของ Tag ที่ใช้กับแต่ละความถี่ [5]

3) เครื่องอ่าน (RFID Reader)

หน้าที่หลักของเครื่องอ่านคือ การเชื่อมต่อเพื่ออ่านหรือเขียนข้อมูลลงในTag ด้วยคลื่นความถี่วิทยุ ภายในเครื่องอ่านจะประกอบด้วย เสาอากาศที่ทำจากขดลวดทองแดงหรือลายทองแดง เพื่อใช้รับหรือส่งสัญญาณ ภาครับและภาคส่งสัญญาณวิทยุ วงจรควบคุมการอ่านหรือเขียนข้อมูลซึ่งมักจะเป็นวงจรจำพวกไมโครคอนโทรลเลอร์ และส่วนของการติดต่อกับคอมพิวเตอร์



รูปที่ 2.15 โครงสร้างภายในเครื่องอ่าน

(ก) เครื่องอ่าน RFID แบบพกพา

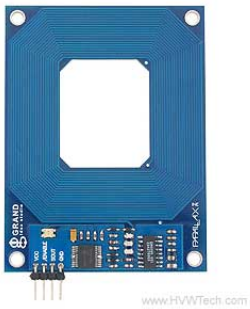
(ข) เครื่องอ่าน RFID แบบติดผนัง

(ค) เครื่องอ่าน RFID แบบประตู

รูปที่ 2.16 เครื่องอ่าน RFID แบบต่างๆ

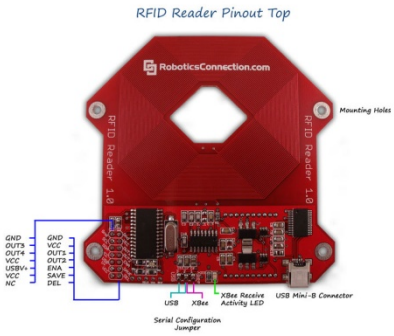
โครงสร้างภายในเครื่องอ่าน RFID เครื่องอ่าน RFID มีองค์ประกอบหลักเริ่มจากส่วนกำเนิดสัญญาณรูปสี่เหลี่ยม(Pulse Generator) ความถี่พาห์เพื่อส่งสัญญาณไปภาคขับ(Driver) เพื่อเพิ่มสมรรถนะในการขับภาคขยายกำลัง (Power Amplifier: PA) ซึ่งทำหน้าที่ในขับกระแสต่อไปสัญญาณต่อไปยังขดลวด เพื่อเหนี่ยวนำทำให้เกิดสนามแม่เหล็กเชื่อมโยงไปยัง Tag ขณะเดียวกันขดลวดดังกล่าวก็จะทำหน้าที่เป็นเสมือนสายอากาศรับสัญญาณสนามแม่เหล็กความถี่พาห์ที่ถูกมอดูเลตกับข้อมูลจำเพาะของ Tag จากนั้นส่วนตรวจจับขอบ (Envelope Detector) จะทำการแยกข้อมูลออกจากคลื่นความถี่พาห์และขยายจนกระทั่งได้รับศักดาของข้อมูลตามมาตรฐานลอจิก เพื่อส่งต่อเข้าส่วนประมวลผลข้อมูล (Processing Unit) ต่อไป โดยทั่วไปหน่วยประมวลผลข้อมูลที่อยู่ภายในเครื่องอ่าน (RFID Reader) มักใช้เป็นไมโครคอนโทรลเลอร์ ซึ่งอัลกอริธึมที่อยู่ภายในโปรแกรมจะทำหน้าที่ถอดรหัสข้อมูล (Decoder) ที่ได้รับและทำหน้าที่ติดต่อกับคอมพิวเตอร์ รูปร่าง ลักษณะทั่วไปของเครื่องอ่านจะแตกต่างกันไปตามประเภทการใช้งาน เช่น แบบที่ใช้มือถือขนาดเล็ก แบบติดผนัง หรือขนาดใหญ่เท่าประตู (Gate Size) แสดงดังรูปที่ 2.16

2.2.4 RFID Reader

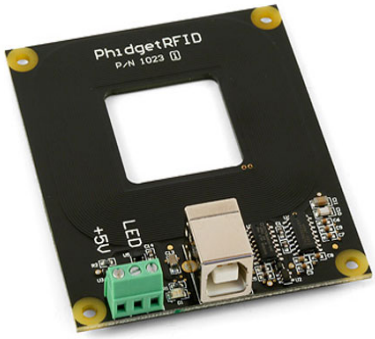


(ก) RFID Reader ของ Parallax [7]

(ข) RFID Reader ID-20 ของ ID Innovations [8]



(ค) RFID Reader ของ Trossen Robotics [9]

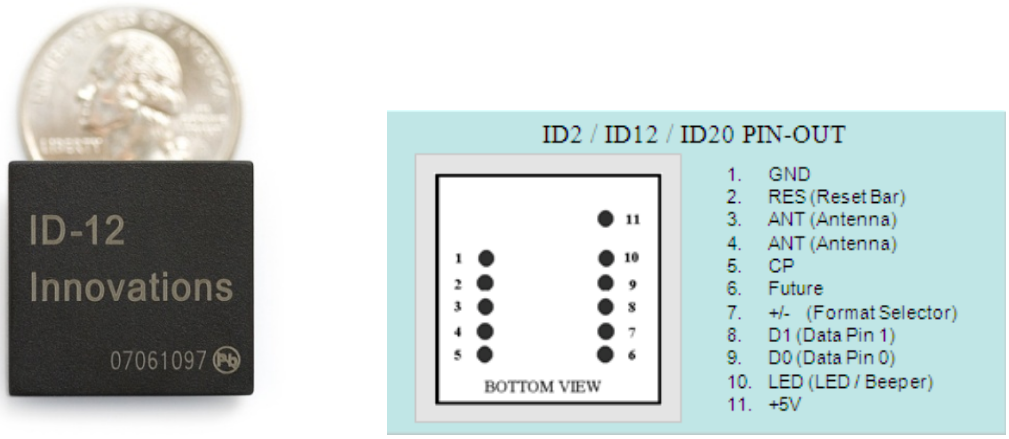


ง) RFID Reader ของ Phidgets [10]

รูปที่ 2.17 RFID Reader ของบริษัทต่างๆ

เครื่องอ่านอาร์เอฟไอดี (RFID Reader) ที่มีจำหน่ายทั้งในตลาดเมืองไทยและต่างประเทศ ก็มีอยู่หลายชนิด ซึ่งในการพัฒนาเครื่องต้นแบบ สิ่งประดิษฐ์ สำหรับนักศึกษานั้น ต้องมองไปที่ RFID Reader แบบ Passive เนื่องจากหาซื้อง่ายและราคาไม่แพง เหมาะกับการศึกษา โดยมีทั้งย่านความถี่ 13.56MHz และ 125KHz หรือชนิดอ่าน/เขียน หรืออ่านอย่างเดียว ให้เลือกใช้

ในงานวิจัยชิ้นนี้ ผู้วิจัยได้เลือก RFID Reader รุ่น ID-12 ของ ID Innovation [8] มาใช้ในการทดลอง ID-12 เป็น RFID Reader ชนิดอ่านอย่างเดียว ทำงานในย่านความถี่ 125kHz มีสายอากาศในตัว ซึ่งจะสะดวกต่อผู้ใช้กรณีไม่ต้องการความยุ่งยากในการออกแบบหรือหาสายอากาศที่เหมาะสมมาใช้ แต่หากต้องการเชื่อมต่อกับสายอากาศภายนอกเพื่อให้การรับสัญญาณดีขึ้นหรือยืดหยุ่นมากขึ้นก็มีขาเอาท์พุตรองรับเช่นกัน



รูปที่ 2.18 รูปร่างและตำแหน่งขาของ RFID Reader รุ่น ID-12 [8]

ตำแหน่งขาใช้งานของ ID-12 จำนวน 11 ขา แต่ถูกใช้งานจริงประมาณ 6-9 ขา โดยสามารถกำหนดรูปแบบของข้อมูลเป็น ASCII, Wiegand26 หรือ Magnet Emulation ได้โดยกำหนดที่ตำแหน่งของการเชื่อมต่อของขา 7 เมื่ออ่านข้อมูลจาก Tag แล้ว ID-12 จะส่งข้อมูลออกทางขา D0 (ขา 9) และ D1 (ขา 8) ส่วนคุณสมบัติทางกายภาพของ ID-12 แสดงในตารางที่ 2.2 โดยมีการเปรียบเทียบคุณสมบัติกับรุ่นอื่นๆ ของผู้ผลิตเดียวกัน ID-12 มีระยะการอ่านประมาณ 12 เซนติเมตร (จากการทดลองจริงได้ต่ำกว่าเล็กน้อย) ย่านความถี่ใช้งาน125KHz รองรับมาตรฐานของ Tag แบบ EM4001 หรือเทียบเท่า ข้อมูลที่รับจาก Tag ใช้การถอดรหัสแบบ Manchester Coding 64 bit หรือ modulus 64 ใช้กับไฟฟ้ากระแสตรง 5VDC ที่กระแส 13mA

ตารางที่ 2.2 คุณสมบัติของ RFID Reader ของ ID Innovations

Parameters	ID-2	ID-12	ID-20
Read Range	N/A (no internal antenna)	12+ cm using 50mm ISO card	16+ cm using 50mm ISO card
Dimensions	21 mm x 19 mm x 6 mm	26 mm x 25 mm x 7 mm	40 mm x 40 mm x 9 mm
Frequency	125 kHz	125 kHz	125 kHz
Card Format	EM 4001 or compatible	EM 4001 or compatible	EM 4001 or compatible
Encoding	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64
Power Requirement	5 VDC @ 13mA nominal	5 VDC @ 30mA nominal	5 VDC @ 65mA nominal
I/O Output Current	+/-200mA PK	-	-
Voltage Supply Range	+4.6V through +5.4V	+4.6V through +5.4V	+4.6V through +5.4V

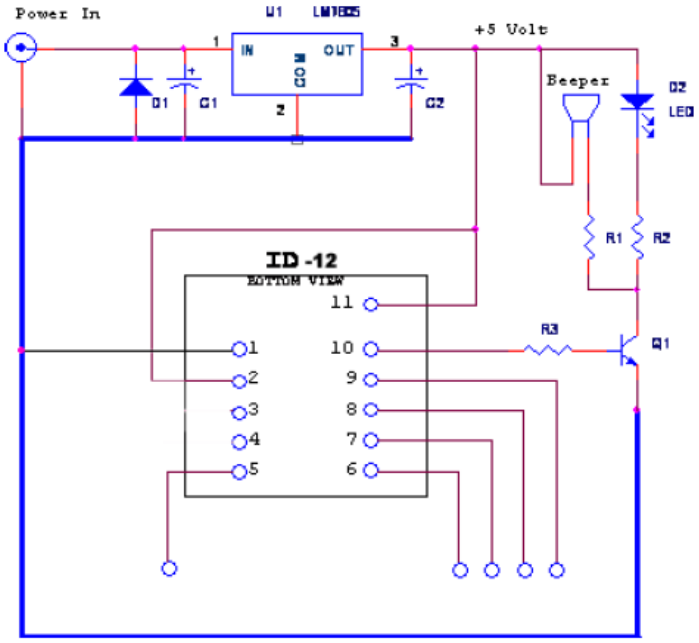
ตารางที่ 2.3 อธิบายตำแหน่งขาและหน้าที่ของ ID-12

Pin No.	Description	ASCII	Magnet Emulation	Wiegand26
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V	GND 0V
Pin 2	Strap to +5V	Reset Bar	Reset Bar	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna	Antenna
Pin 5	Card Present	No function	Card Present *	No function
Pin 6	Future	Future	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND	Strap to Pin 10	Strap to +5V
Pin 8	Data 1	CMOS	Clock *	One Output *
Pin 9	Data 0	TTL Data (inverted)	Data *	Zero Output *
Pin 10	3.1 kHz Logic	Beeper / LED	Beeper / LED	Beeper / LED
Pin 11	DC Voltage Supply	+5V	+5V	+5V

* Requires 4K7 Pull-up resistor to +5V

ตารางที่ 2.3 แสดงตำแหน่งและหน้าที่ของขาต่างๆ กรณีที่ต้องการสื่อสารข้อมูลแบบ ASCII ทำได้โดยเชื่อมต่อขา 7 ลงกราวด์ หากต้องการข้อมูลแบบ Wiegand26 ทำได้โดยต่อขา 7 กับแหล่งจ่าย +5VDC หรือข้อมูลแบบ Magnet Emulation ให้ต่อขา 7 กับขา 10 ซึ่งข้อมูลทั้ง 3 รูปแบบนี้จะมีรหัสรูปแบบที่แตกต่างกัน ถ้าต้องการระยะการอ่านที่มากขึ้นสามารถทำการติดตั้งสายอากาศภายนอกโดยต่อสายอากาศเข้าที่ขา 3 และขา 4

การต่อ ID-12 ไปใช้งานสามารถทำได้ดังรูปที่ 2.19 โดยต่อกับแหล่งจ่ายไฟฟ้ากระแสตรงขนาด 5 โวลต์ ซึ่งเป็นระดับแรงดัน TTL ทำให้สามารถเชื่อมต่อไปใช้งานกับไมโครคอนโทรลเลอร์ได้โดยตรง การใช้งานสามารถต่อใช้งานจากขา D0 (ขา 9) หรือขา D1 (ขา 8) ส่วนสถานะของการอ่านข้อมูลออกที่ขา 10 ผู้ใช้จะต้องต่อ Transistor เพื่อสวิตช์ให้กระแสไหลผ่าน Transistor ซึ่งจะนำไปขับ Buzzer และ LED ให้มีการกระพริบกรณีที่มีการอ่านข้อมูล



COMPONENT LIST
R1 = 100R
R2 = 1K
R3 = 1K
C1 = 100uF 16V
C2 = 100uF 10V
Beeper = 2.7-3.5KHz 100R
D1 = 1N4001
D2 = GREEN LED
U1 = LM7805
Q1 = UTC8050 (NPN)
ID2 = ID Innovations ID2

* Please Note the ID2 has an internal tuning capacitor of 1.5nF and this makes the total tuning capacity = 2.5nF

The 3.1Khz Beeper Logic is centered for most Beepers in range 2.7-3.5Khz

รูปที่ 2.19 ผังวงจรใช้งานของ ID-12

Output Data Structure – ASCII

STX (02h)	DATA (10 ASCII)	CHECK SUM (2 ASCII)	CR	LF	ETX (03h)

[The 1byte (2 ASCII characters) Check sum is the "Exclusive OR" of the 5 hex bytes (10 ASCII) Data characters.]

Output Data Structure – Wiegand26

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P	E	E	E	E	E	E	E	E	E	E	E	E	O	O	O	O	O	O	O	O	O	O	O	O	P
Even parity (E)													Odd parity (O)												

P = Parity start bit and stop bit

Output Data Magnetic ABA Track2

10 Leading Zeros	SS	Data	ES	LCR	10 Ending Zeros

[SS is the Start Character of 11010, ES is the end character of 11111, LRC is the Longitudinal Redundancy Check.]

รูปที่ 2.20 รูปแบบข้อมูลของ ID-12

สำหรับ ID-12 จะมีจุดเด่นตรงที่สามารถเลือกรูปแบบข้อมูล (Data Format) ได้ถึง 3 รูปแบบขึ้นอยู่กับความต้องการโดยใช้เงื่อนไขของการต่อของขา 7 (Format Selector +/-) ดังรูปที่ 2.20

ASCII Format กรณีเลือกส่งข้อมูลอนุกรมแบบ ASCII สำหรับ ID-12 จะต้องตั้งค่าพารามิเตอร์ของพอร์ตอนุกรมเป็น 9600,N,8,1 (Flow control None) ซึ่ง Tag ส่งข้อมูลมายัง ID-12 จะส่งมาขนาด 16 Bytes ประกอบด้วย Data และ Checksum เป็นรหัส ASCII ฐานสิบหก (2 ASCII ต่ออักษร 1 Byte)

Output Format – Serial ASCII 9600, N,8,1

02 (1byte)	10 ASCII Hex Data Characters (10bytes)	2 ASCII char's Checksum (2byte)	CR (1byte)	LF (1byte)	03 (1byte)

รูปที่ 2.21 Output Format ของ ASCII

ตัวอย่างเช่น ใช้ Hyper-Terminal 9600,N,8,1 (Flow control None) เมื่ออ่านค่าได้ดังรูป

☻041A21EE34E5
♥

หมายความว่า ID-12 จะอ่านค่าจาก Tag ได้ดังนี้

Start Byte ☻ = 02 ASCII

End Byte ♥ = 03 ASCII

Hex Data 5 Bytes = 041A21EE3A ASCII

และ Checksum = E5

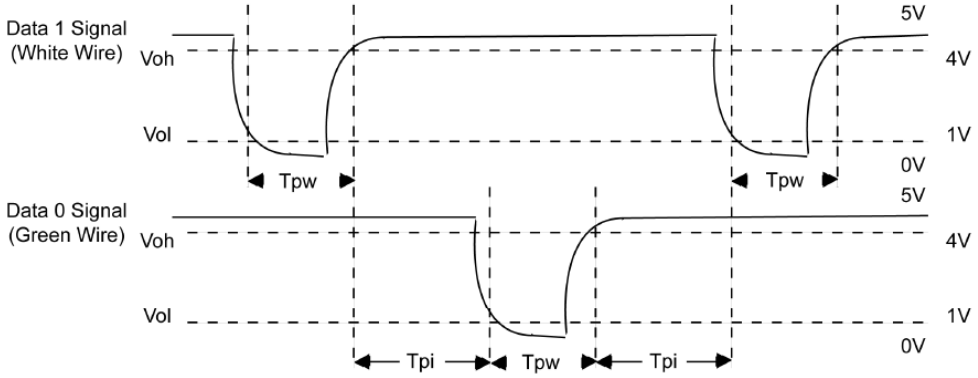
โดย Checksum เกิดจากการนำ Data ทั้ง 5 Bytes มาแปลงเป็นเลขฐานสองแล้ว Exclusive OR กันและแปลงกลับมาเป็นเลขฐานสิบหกอีกครั้ง

$$04_{hex} \oplus 1A_{hex} \oplus 21_{hex} \oplus EE_{hex} \oplus 34_{hex} = E5_{hex}$$

Wiegand26 Format เป็นรูปแบบของข้อมูลใน Tag แบบหนึ่งซึ่งถูกจัดเป็นมาตรฐานอุตสาหกรรม มีขนาด 26 บิต ใน 26 บิต ประกอบด้วย Facility code ขนาด 8 บิต และ ID number ขนาด 16 บิต สำหรับ Facility code ขนาด 8 บิต สามารถกำหนดเป็นรหัสได้ 256 รหัส (0-255) ส่วน ID number สามารถกำหนดเป็นค่า ID ได้ 65,536 ID (0-65,535) เนื่องจากข้อจำกัดของจำนวนรหัส รหัสจึงอาจมีการทำสำเนาได้ รูปแบบเต็มของ 26-bit Wiegand แสดงดังตารางที่ 2.4

ตารางที่ 2.4 26-bit Wiegand Format

บิต	ความหมาย
Bit 1	Even parity ระหว่างบิต 2 ถึง 13
Bit 2 ถึง 9	Facility code (0 ถึง 255); บิต 2 เป็น MSB
Bit 10 ถึง 25	ID number (0 ถึง 65,535); บิต 10 เป็น MSB
Bit 26	Odd parity ระหว่างบิต 14 ถึง 25



รูปที่ 2.22 Data Bit Timing ของ Wiegand26

ลักษะการเปลี่ยนแปลงของสัญญาณข้อมูลแสดงดังรูปที่ 2.22 ทั้ง Data0 และ Data1 จะมีสถานะลอจิกเป็น High (เหนือระดับ Voh) จนกระทั่งตัวอ่านพร้อมที่จะส่งชุดข้อมูล ตัวอ่านจะใส่ข้อมูลทำให้เกิดเป็นสัญญาณพัลส์ลบ (ต่ำกว่า Vol) โดยสัญญาณพัลส์ของทั้ง Data1 และ Data0 จะไม่ทับซ้อนกัน โดยขนาดของความกว้างของพัลส์ต่ำสุดและสูงสุด จะแสดงดังตารางที่ 2.5

ตารางที่ 2.5 ช่วงเวลาของสัญญาณข้อมูล

สัญลักษณ์	คำอธิบาย	ช่วงเวลา
Tpw	Pulse Width Time	100 us
Tpi	Pulse Interval Time	1 ms

2.3 ไมโครคอนโทรลเลอร์

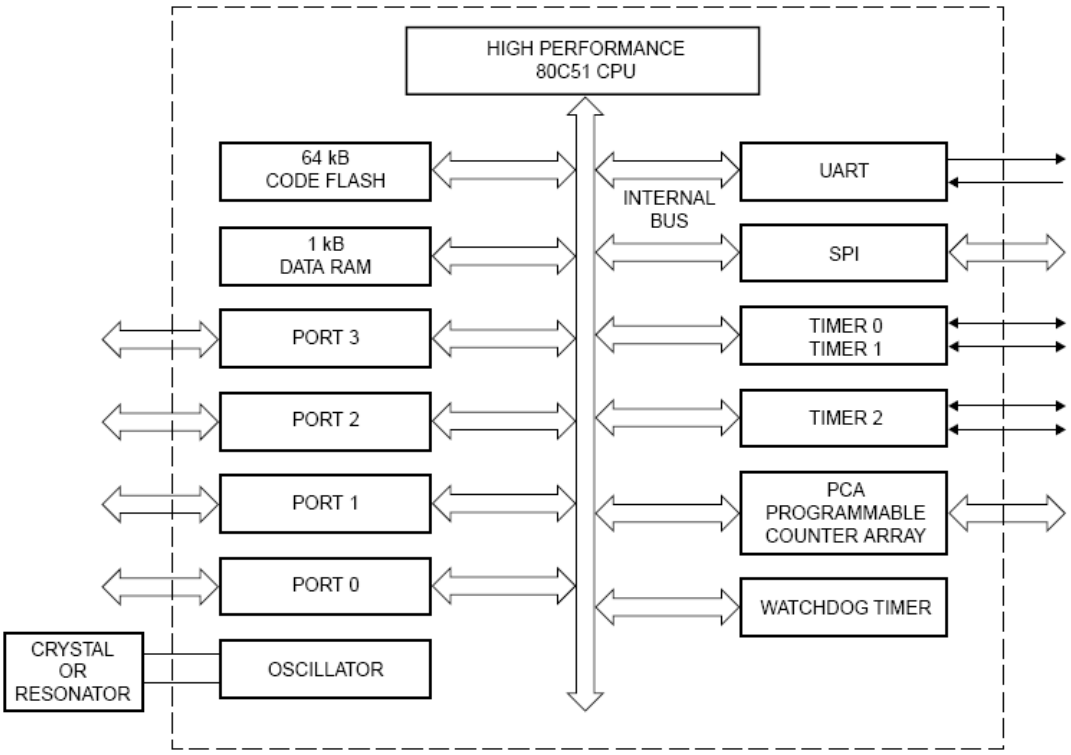
ไมโครคอนโทรลเลอร์ทุกตระกูลสามารถงานร่วมกับ RFID ได้ ในงานวิจัยนี้เลือกใช้อยู่ 2 ตระกูลคือ MCS-51 และ PIC ซึ่งทั้งคู่เป็นไมโครคอนโทรลเลอร์ที่ได้รับความนิยมสูง

2.3.1 ไมโครคอนโทรลเลอร์ MCS-51 [11]

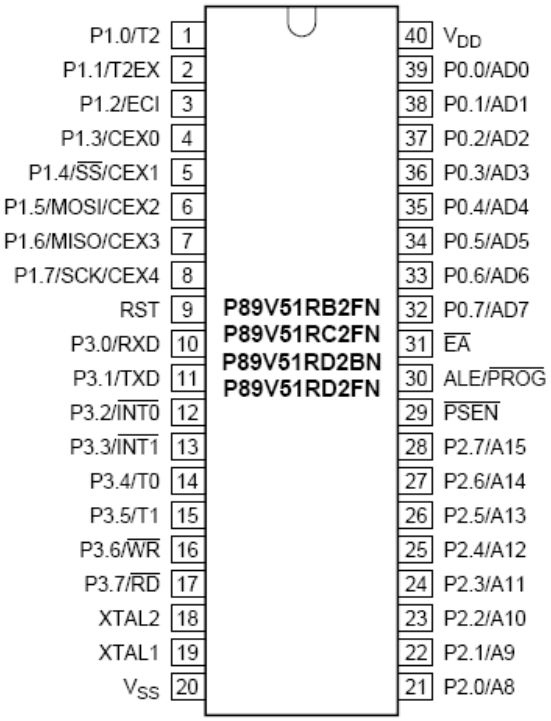
MCS-51 ที่เลือกใช้เป็นเบอร์ P89V51RD2 เป็นไมโครคอนโทรลเลอร์แบบ 40 ขา ขนาด 8 บิต มีหน่วยความจำแบบแฟลช 64 KB หน่วยความจำข้อมูล 1 KB จำนวนพอร์ตอินพุต-เอาท์พุต ให้ใช้งาน 4 พอร์ต มีโครงสร้างภายในดังรูปที่ 2.23 และมีคุณสมบัติอื่นๆ ดังนี้

1) ทำงานที่แรงดัน 5 โวลต์ และที่สัญญาณนาฬิกา 0-40 MHz
2) หน่วยความจำแบบแฟลช 64 KB
3) รองรับอัตราสัญญาณนาฬิกาในแบบ 12-clock และ X2 mode (6-clock)
4) อินพุท-เอาท์พุท 8 บิต จำนวน 4 พอร์ต
5) มีไทเมอร์/เคาน์เตอร์ 16 บิต 3 ตัว
6) รองรับการอินเตอร์รัพได้จาก 4 แหล่ง

7) รองรับระดับลอจิก TTL และ CMOS



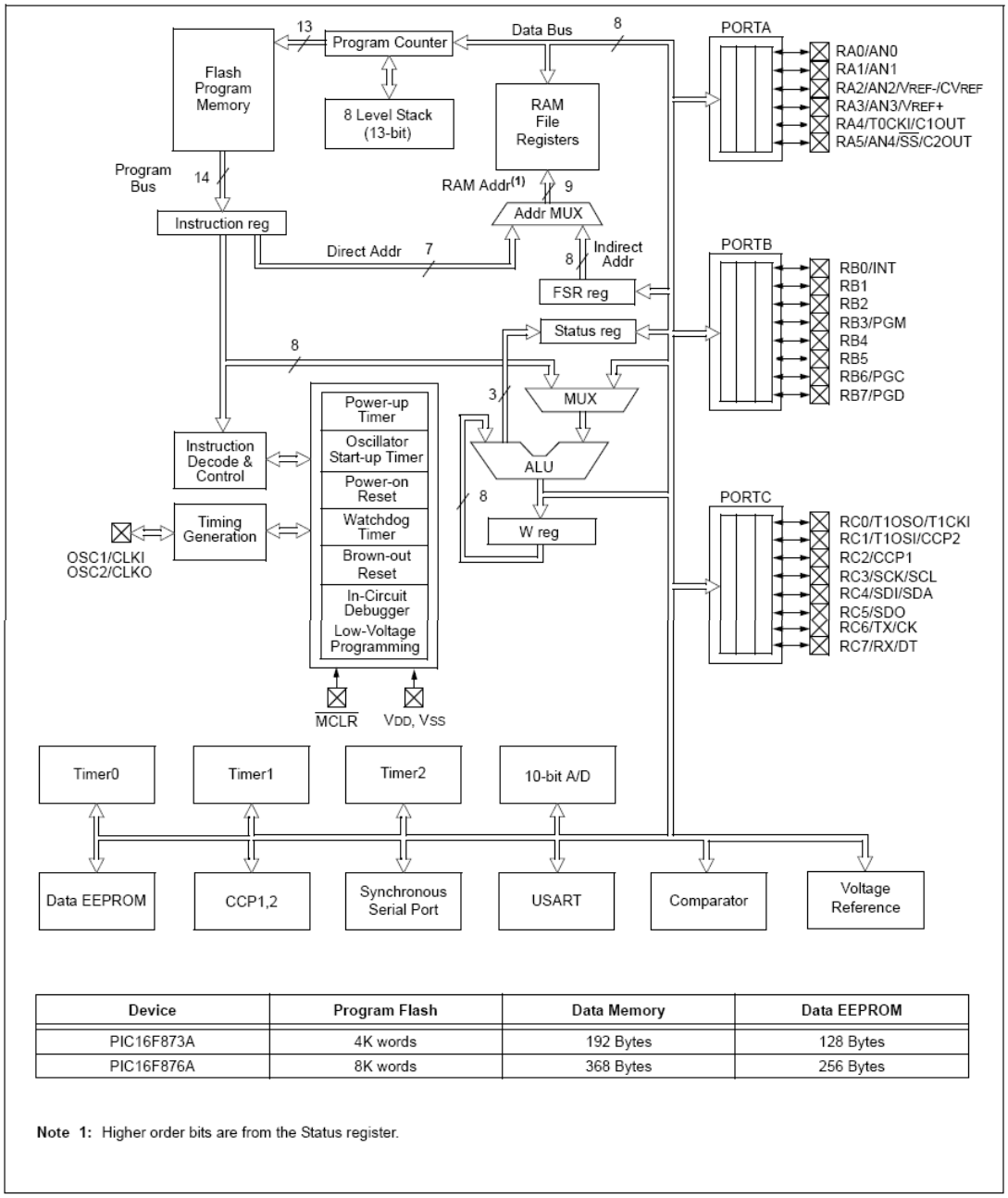
รูปที่ 2.23 โครงสร้างภายในของ P89V51RD2



รูปที่ 2.24 การจัดตำแหน่งขาของ P89V51RD2

2.3.2 ไมโครคอนโทรลเลอร์ PIC [12]

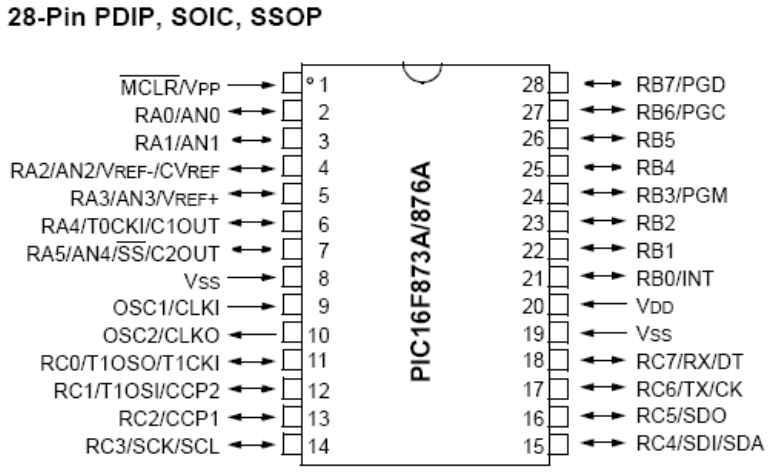
ไมโครคอนโทรลเลอร์ ตระกูล PIC ซึ่งในงานวิจัยนี้เลือกใช้ เบอร์ PIC16F876A ซึ่งผลิตโดยบริษัท Microchip เป็นไมโครคอนโทรลเลอร์ขนาดเล็กแต่มีประสิทธิภาพสูง และมีฟังก์ชั่นการทำงานมาก เป็นที่นิยมใช้มากขึ้น มีโครงสร้างภายในดังรูปที่ 2.25 และรูปแบบการจัดตำแหน่งขาดังรูปที่ 2.26



Device	Program Flash	Data Memory	Data EEPROM
PIC16F873A	4K words	192 Bytes	128 Bytes
PIC16F876A	8K words	368 Bytes	256 Bytes

Note 1: Higher order bits are from the Status register.

รูปที่ 2.25 โครงสร้างภายในของ PIC16F876A



รูปที่ 2.26 การจัดตำแหน่งขาของ PIC16F876A

โครงสร้างภายในประกอบด้วย ซีพียู หน่วยความจำโปรแกรม หน่วยความจำข้อมูล ส่วนติดต่อพอร์ต ส่วนจัดการสัญญาณนาฬิกาส่วนควบคุมการอินเตอร์รัพท์ และฟังก์ชั่นอื่นๆ อีกมาก โดยมีคุณสมบัติที่สำคัญดังนี้

1) เป็น RISC CPU มีคำสั่งใช้งานเพียง 35 คำสั่ง
2) กระทำคำสั่งโดยใช้สัญญาณนาฬิกาเพียงไซเคิลเดียว ยกเว้นสาขาของโปรแกรมใช้สองไซเคิล
3) ทำงานที่ความถี่สัญญาณนาฬิกา 0-20MHz
4) หน่วยความจำโปรแกรมแบบแฟลช 8K x 14 Word
5) หน่วยความจำข้อมูล 368 bytes
6) ตอบสนองแหล่งกำเนิดอินเตอร์รัพท์ 14 แหล่ง
7) มีพอร์ตอินพุตและเอาท์พุต 3 พอร์ต (Port A, B, C)

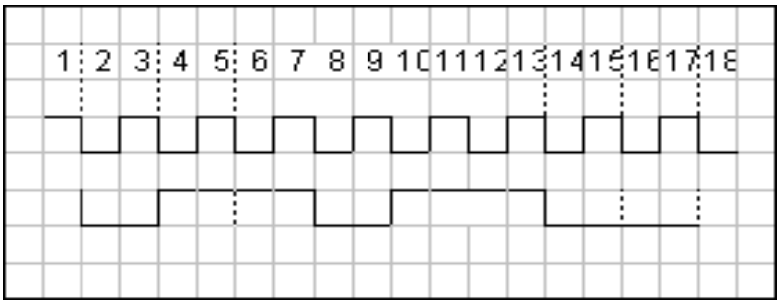
ฟังก์ชั่นอื่นผู้ศึกษาสามารถหาข้อมูลเพิ่มเติมได้จากหลายแหล่ง เมื่อเปรียบเทียบกับไมโครคอนโทรลเลอร์รุ่นพี่ที่ได้รับความนิยมสูงมากเช่นกัน คือ PIC6F877 ก็แทบจะเหมือนกันทุกประการ ยกเว้นจำนวนขาที่ PIC16F877 มีมากกว่าคือมี 40 ขา เพราะมี I/O Port มากกว่า 1 Port นั่นเอง

2.4 การสื่อสารข้อมูล

2.4.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบ คือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส

การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อสื่อสารกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมอย่างน้อยที่สุด 3 เส้นคือสัญญาณนาฬิกาข้อมูล และกราวด์ รูปที่ 2.27 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2.27 ลักษณะข้อมูลอนุกรมแบบซิงโครนัส

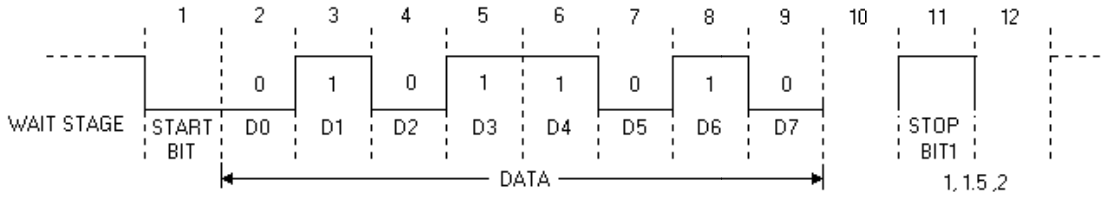
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูลหรือบอดเรต (Baud Rate) มีหน่วยเป็น บิตต่อวินาที (Bit per Second: BPS) รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

1) บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต

2) บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต

3) บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี

4) บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5, หรือ 2 บิต



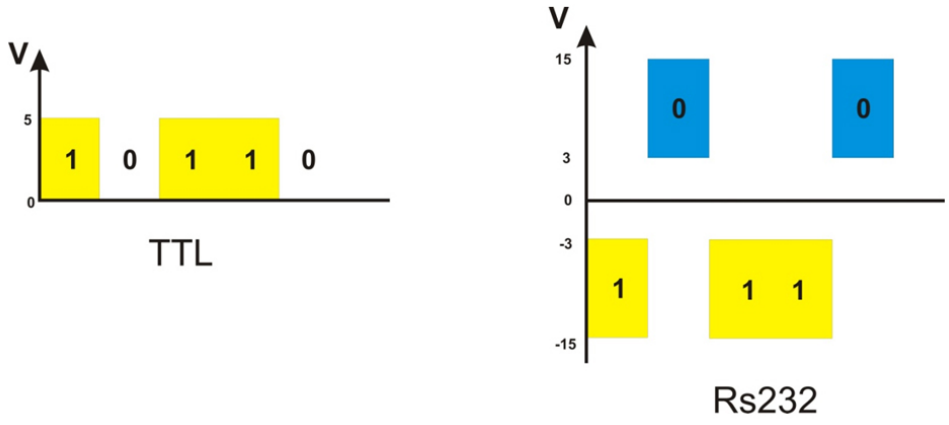
รูปที่ 2.28 รูปข้อมูลอนุกรมแบบอะซิงโครนัส

จากรูปที่ 2.28 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิก “1” เรียกสถานะนี้ว่าสถานะหยุดรอ (Waiting Stage การเริ่มต้นส่งข้อมูลจะเริ่มให้จากการให้ขา DATA มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัส เรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือค่าบอเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้ส่งพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115,200 บิต ต่อวินาที เนื่องจากบอดเรต คือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่างข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้อัตราการถ่ายทอดข้อมูลในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาทีก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (Odd), แบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้

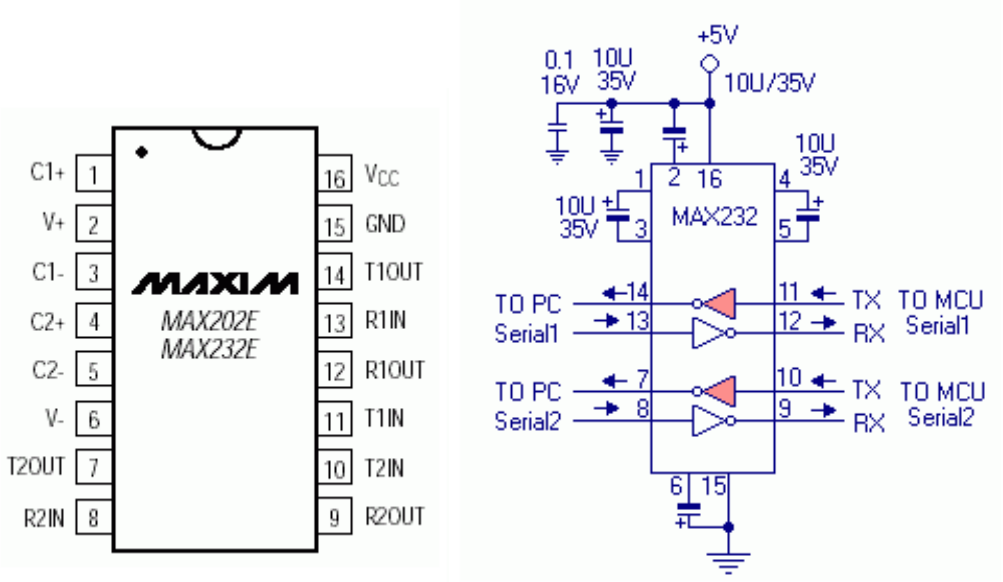
2.4.2 มาตรฐานการสื่อสารข้อมูลแบบอนุกรม RS232 [13]



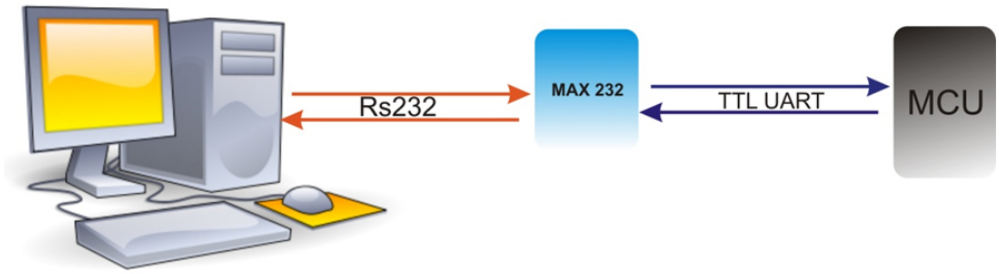
รูปที่ 2.29 ระดับแรงดันของ RS232 และ TTL

มาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรม RS232 จะมีค่าระดับแรงดันในช่วง -15 โวลต์ ถึง +15 โวลต์ ส่วนในระบบของไมโครคอนโทรลเลอร์นั้นจะให้สัญญาณออกมาในลักษณะสัญญาณ TTL ซึ่งมีระดับแรงดันอยู่ในช่วง 0 ถึง 5 โวลต์ โดยระดับลอจิกของ RS232 Logic “0” จะอยู่ในช่วง 3 ถึง 15 โวลต์ Logic “1” จะอยู่ในช่วง -3 ถึง -15 โวลต์ ส่วน TTL Logic “0” จะเท่ากับ 0 โวลต์ Logic “1” จะเท่ากับ 5V หรือสำหรับอุปกรณ์รุ่นใหม่บางตัวจะอยู่ในช่วง 3.3 โวลต์

ดังนั้น การทำให้ไมโครคอนโทรลเลอร์สามารถสื่อสารผ่านพอร์ตอนุกรมตามมาตรฐาน RS232 ได้นั้นจะต้องปรับระดับสัญญาณ TTL ของไมโครคอนโทรลเลอร์ให้มีระดับเดียวกันกับระดับแรงดันตามมาตรฐาน RS232 โดยสามารถใช้ไอซี MAX232 เป็นตัวปรับระดับสัญญาณดังกล่าว

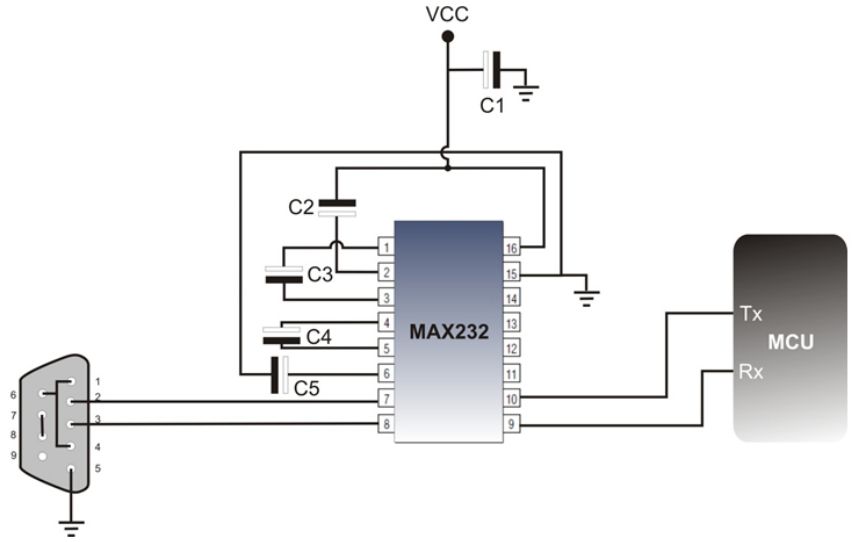


รูปที่ 2.30 ตำแหน่งขาและโครงสร้างภายใน MAX232



รูปที่ 2.31 รูปแบบการแปลงสัญญาณระหว่าง TTL และ RS232

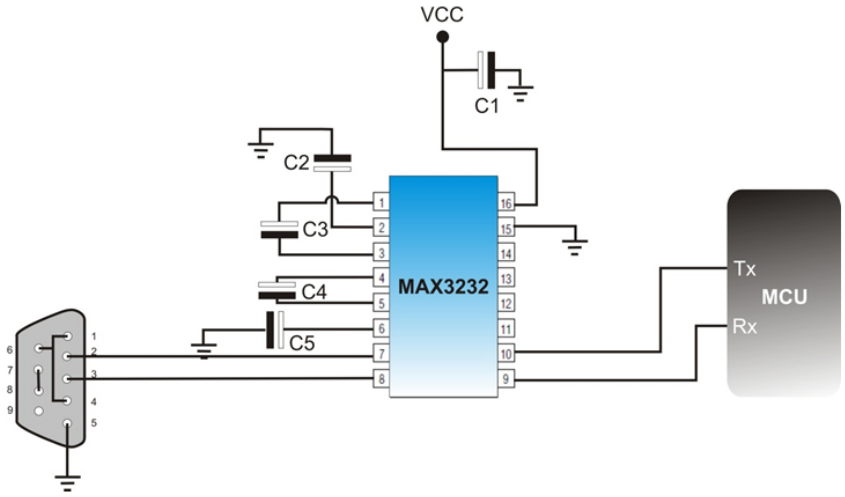
โดยรูปแบบการต่อใช้งาน MAX232 จะเปลี่ยนสัญญาณ TTL เป็น RS232 ในฝั่งส่ง และเปลี่ยน RS232 เป็น TTL ในฝั่งรับ ดังรูปที่ 2.31 ส่วนวิธีการแปลงสัญญาณ RS232 เป็น TTL ระดับ 0



Vcc = 5V

C1 – C5 = 10 uF

รูปที่ 2.32 รูปแบบการแปลงสัญญาณระหว่าง TTL 5VDC และ RS232

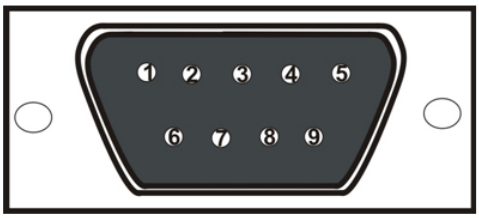


Vcc = 3.3V

C1-C5 = 0.1uF

รูปที่ 2.33 รูปแบบการแปลงสัญญาณระหว่าง TTL 3.3VDC และ RS232

พอร์ตอนุกรม RS232 ของเครื่องคอมพิวเตอร์จะสามารถเชื่อมต่อสายสัญญาณได้ราว 50 ฟุต ขึ้นอยู่กับชนิดของสายสัญญาณ, ระยะทาง, และปริมาณสัญญาณรบกวน โดยพอร์ตอนุกรมของ PC จะเป็นคอนเน็คเตอร์แบบ DB9 ตัวผู้ พอร์ตอนุกรมของอุปกรณ์ภายนอก จะเป็นคอนเน็คเตอร์แบบ DB9 ตัวเมีย



DB-9M	Function	Abbreviation
Pin #1	Data Carrier Detect	CD
Pin #2	Receive Data	RD or RX or RXD
Pin #3	Transmitted Data	TD or TX or TXD
Pin #4	Data Terminal Ready	DTR
Pin #5	Signal Ground	GND
Pin #6	Data Set Ready	DSR
Pin #7	Requests To Send	RTS
Pin #8	Clear To Send	CTS
Pin #9	Ring Indicator	RI

รูปที่ 2.34 ตำแหน่งขาของ DB9 Connector และหน้าที่การใช้งาน

2.4.3 UART

UART ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารแบบอนุกรม หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัสแล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าคอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังทำการแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์ทราบด้วย เช่น อัตราความเร็วในการรับส่งข้อมูล (Buad Rate) รูปแบบการส่งข้อมูล ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล ภายใน UART จะมีส่วนของวงจรสร้างอัตราการถ่ายทอดข้อมูลแบบโปรแกรมได้ (Programmable Quadrate Generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSR & DCD) ของพอร์ตอนุกรม RS232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและภาครับจะไม่ถูกกลับสถานะ UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS232 ก่อนส่งออกไปจากคอมพิวเตอร์สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

1) COM 1 : 3F8H
2) COM 2 : 2F8H
3) COM 3 : 3E8H
4) COM 4 : 2E8H

2.5 โปรแกรมภาษาคอมพิวเตอร์ที่ใช้ในการควบคุมไมโครคอนโทรลเลอร์ [14]

การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ เพื่อทำงานร่วมกับ RFID จำเป็นอย่างยิ่งที่จะต้องมีความรู้ในภาษาคอมพิวเตอร์ อาทิ ภาษาซี (C Language) ภาษาแอสเซมบลี (Assembly) และหากมีการควบคุมหรือสั่งงานไมโครคอนโทรลเลอร์ผ่านจอมอนิเตอร์ในลักษณะ GUI บน Windows ก็ควรต้องมีความรู้เกี่ยวกับ Visual Basic หรือ Visual Studio และหากต้องการเชื่อมต่อระบบกับฐานข้อมูล เช่น ฐานข้อมูลการจัดเก็บสินค้า ระบบบริหารทรัพยากรบุคคล ก็ต้องมีความรู้ด้านโปรแกรมฐานข้อมูลด้วย เช่น Microsoft Access ซึ่งการที่นักศึกษาจะมีความรู้ครบถ้วนทุกอย่างในระยะเวลาการศึกษาที่จำกัด ก็เป็นเรื่องยาก ดังนั้นงานวิจัยชิ้นนี้จึงให้ความสำคัญในประเด็นการควบคุมด้วยไมโครคอนโทรลเลอร์เป็นหลัก และหากนักศึกษาสนใจประเด็นใดเป็นพิเศษ ก็สามารถศึกษาเพิ่มเติมด้วยตนเองได้ โดยเฉพาะในยุคปัจจุบันสื่อต่างๆ ก็สามารถสืบค้นได้ไม่ยากนัก

2.5.1 ภาษาซี (C Language)

โครงสร้างของภาษาซี ประกอบด้วยไฟล์ส่วนหัวของโปรแกรม (Header Files) และส่วนของตัวโปรแกรม ไฟล์ส่วนหัวเป็นไฟล์ที่มีส่วนขยายเป็น .h ใช้ร่วมในการคอมไพล์โปรแกรม ส่วนตัวโปรแกรมจะเริ่มต้นด้วยฟังก์ชั่น main() ซึ่งเป็นฟังก์ชั่นหลักของโปรแกรมมีเครื่องหมายปีกกาเปิด ({) เป็นการเริ่มต้นการเขียนโปรแกรม และเครื่องหมายปีกกาปิด (}) เป็นเครื่องหมายจบโปรแกรม ภายในฟังก์ชั่น main() จะประกอบด้วยชุดคำสั่งและฟังก์ชั่นต่างๆ ซึ่งเกือบทั้งหมดจะปิด

ตารางที่ 2.6 ชนิดของตัวแปรในภาษาซี

ชนิดตัวแปร	ขนาด	ช่วงของข้อมูล	การกำหนดตัวแปร
1. Character	8 บิต	-128 ถึง +127	char
2. Unsigned Character	8 บิต	0 ถึง 255	unsigned char
3. Integer	16 บิต	-32,768 ถึง +32,767	int
4. Unsigned Integer	16 บิต	0 ถึง 65,535	unsigned int
5. Long Integer	32 บิต	-2,147,483,648 ถึง +2,147,483,647	long int
6. Floating Point	32 บิต	3.4x10-38 ถึง -3.4x10+38	float

การกำหนดตัวแปร จะต้องกำหนดให้เหมาะสมกับการใช้งาน เช่น การคำนวณจุดทศนิยมควรกำหนดแบบ float หรือใช้ในการวนรอบไม่เกิน 127 รอบควรเป็นแบบ char การกระทำทางคณิตศาสตร์หรือการคำนวณใช้เครื่องหมายให้ถูกต้อง ส่วนการเปรียบเทียบข้อมูลจะใช้ในรูปแบบของตัวกระทำเปรียบเทียบ ซึ่งมักจะใช้ร่วมกับคำสั่ง if เพื่อตรวจสอบเงื่อนไขการทำงาน

ตารางที่ 2.7 ตัวกระทำทางคณิตศาสตร์

เครื่องหมาย	ความหมาย
+	การบวก
-	การลบ
*	การคูณ
/	การหาร
%	การหารคิดเฉพาะเศษ
++	การเพิ่มค่าขึ้น 1
--	การลดค่าลง 1

ตารางที่ 2.8 ตัวกระทำเปรียบเทียบ

เครื่องหมาย	ความหมาย
>	มากกว่า
>=	มากกว่าหรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่าหรือเท่ากับ
==	เท่ากับ
!=	ไม่เท่ากับ

ตารางที่ 2.9 ตัวกระทำลอจิก

เครื่องหมาย	ความหมาย
&&	AND
||	OR
!	NOT

ตัวกระทำลอจิกส่วนใหญ่จะใช้ร่วมกับคำสั่ง if เพื่อตรวจสอบเงื่อนไขการทำงาน ส่วนตัวกระทำระดับบิต จะสามารถเข้าถึงข้อมูลในระดับบิตได้ ช่วยให้สะดวกในการเขียนโปรแกรมตรวจสอบบิตข้อมูล

ตารางที่ 2.10 ตัวกระทำระดับบิต

เครื่องหมาย	ความหมาย
~	NOT
&	AND
|	OR
^	Exclusive OR
<<	Shift Left Bits
>>	Shift Right Bits

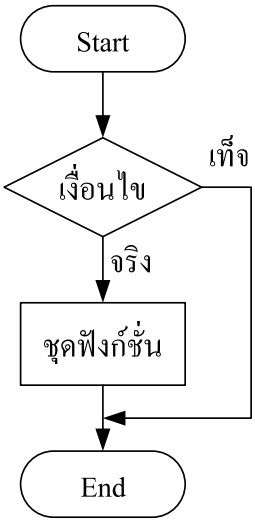
ฟังก์ชั่นการสั่งงานของภาษาซี ซึ่งเป็นองค์ประกอบหลักที่สำคัญมีดังนี้

ฟังก์ชั่น if เป็นฟังก์ชั่นที่ใช้ในการเปรียบเทียบข้อมูลเพื่อใช้ในการตัดสินใจว่าจะทำงานในเงื่อนไขหรือทางเลือกอื่นใด มีรายละเอียดดังนี้

ฟังก์ชั่น if ทางเลือกเดียว ถ้าเงื่อนไขในการเปรียบเทียบเป็นจริงจะทำคำสั่งในชุดฟังก์ชั่น ถ้าไม่จริงจะจบการทำงาน

<u>รูปแบบ</u>

```
if (ตัวแปร ตัวกระทำเปรียบเทียบ ค่าคงที่หรือตัวแปร)
        {
        ชุดฟังก์ชั่น
        }
```

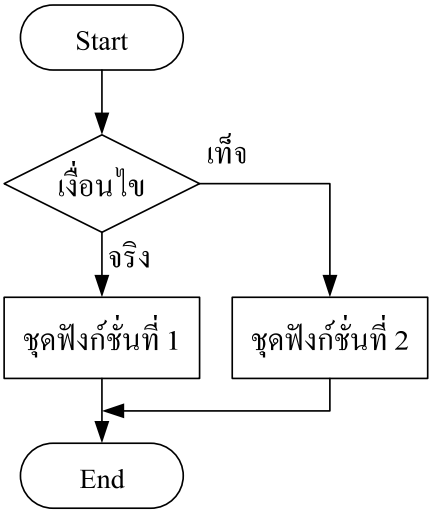


รูปที่ 2.35 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น if ทางเลือกเดียว

ฟังก์ชั่น if สองทางเลือก จะใช้ร่วมกับ else ถ้าเงื่อนไขในการเปรียบเทียบเป็นจริงจะทำคำสั่งในชุดฟังก์ชั่นที่ 1 ถ้าไม่จริงจะทำคำสั่งในชุดฟังก์ชั่นที่ 2

<u>รูปแบบ</u>

```
if (ตัวแปร ตัวกระทำเปรียบเทียบ ค่าคงที่หรือตัวแปร)
        {  ชุดฟังก์ชั่นที่ 1
        }
else
        {  ชุดฟังก์ชั่นที่ 2
        }
```

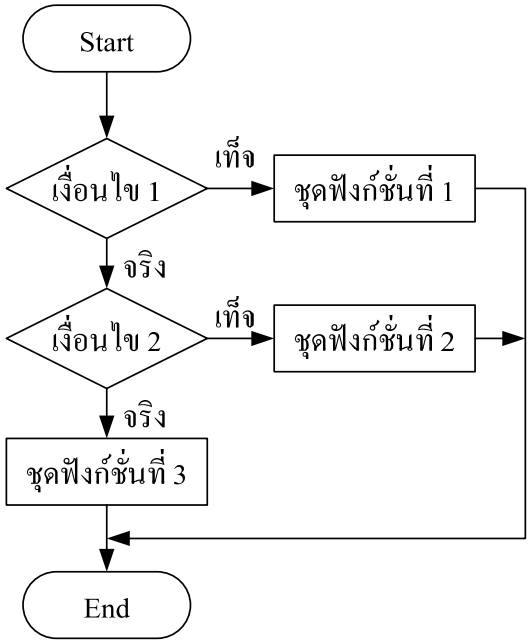


รูปที่ 2.36 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น if สองทางเลือก

ฟังก์ชั่น if หลายทางเลือก จะใช้ร่วมกับ else ถ้าเงื่อนไขในการเปรียบเทียบเป็นจริงจะทำคำสั่งในชุดฟังก์ชั่นที่ 1 ถ้าเป็นเท็จจะเปรียบเทียบกับเงื่อนไขที่ 2 ถ้าเป็นจริงจะทำคำสั่งในชุดฟังก์ชั่นที่ 2 ถ้าเป็นเท็จจะทำคำสั่งในชุดฟังก์ชั่นที่ 3

รูปแบบ

```
if (ตัวแปร ตัวกระทำเปรียบเทียบ ค่าคงที่หรือตัวแปร)
        {  ชุดฟังก์ชั่นที่ 1
        }
else if (ตัวแปร ตัวกระทำเปรียบเทียบ ค่าคงที่หรือตัวแปร)
        {  ชุดฟังก์ชั่นที่ 2
        }
else
        {  ชุดฟังก์ชั่นที่ 3
        }
```



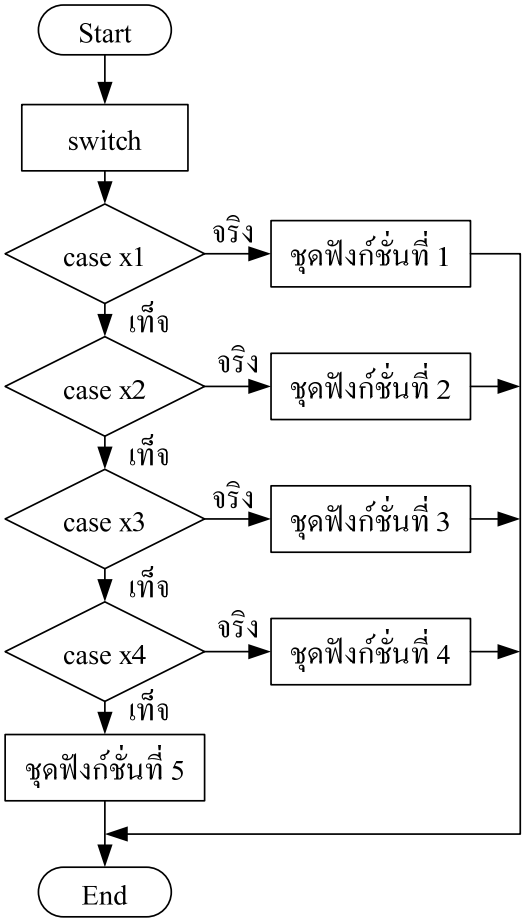
รูปที่ 2.37 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น if หลายทางเลือก

ฟังก์ชั่น switch เป็นฟังก์ชั่นที่ใช้ในการเปรียบเทียบข้อมูลหลายๆ ทางเลือก แต่จะไม่สามารถเปรียบเทียบแบบมากกว่าหรือน้อยกว่าได้ แต่จะเปรียบเทียบข้อมูลกับค่าคงที่

รูปแบบ

```
switch (v)
        {
        case x1 : ชุดฟังก์ชั่นที่ 1 break;
        case x2 : ชุดฟังก์ชั่นที่ 2 break;
        case x3 : ชุดฟังก์ชั่นที่ 3 break;
        case x4 : ชุดฟังก์ชั่นที่ 4 break;
        default : ชุดฟังก์ชั่นที่ 5
        }
```

ฟังก์ชั่น switch จะนำตัวแปร v มาเปรียบเทียบกับ x1 – x4 ถ้าตรงกับ case ใดก็ทำตามชุดฟังก์ชั่นนั้น ถ้าไม่ตรงกับเงื่อนไขใด ก็ทำตามชุดฟังก์ชั่นหลัง default



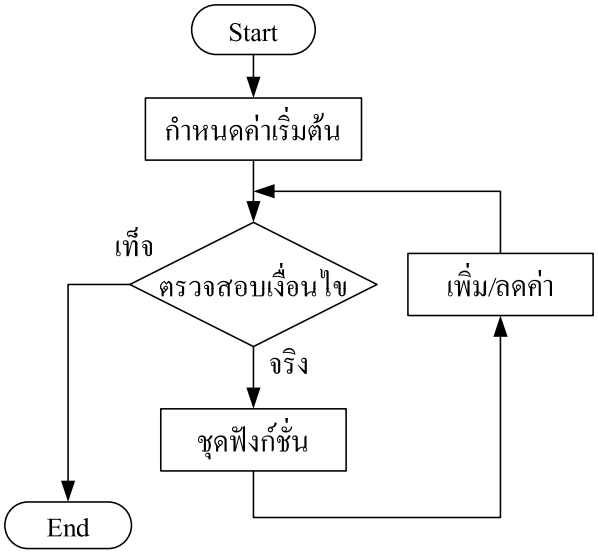
รูปที่ 2.38 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น switch

ฟังก์ชั่น for เป็นฟังก์ชั่นวนลูป ใช้ในลักษณะให้โปรแกรมทำงานวนรอบซ้ำๆ เพื่อลดขนาดของโปรแกรมและเพิ่มความเร็วในการประมวลผล

รูปแบบ

```
for (ค่าเริ่มต้น; เงื่อนไข; เพิ่มหรือลดค่า)
        {
        ชุดฟังก์ชั่น
        }
```

ฟังก์ชั่น for จะเริ่มจากการกำหนดค่าเริ่มต้นให้ตัวแปร แล้วตรวจสอบเงื่อนไข ถ้าเป็นจริง จะเพิ่มหรือลดค่าให้กับตัวแปร ถ้าเป็นเท็จจะออกจากลูป

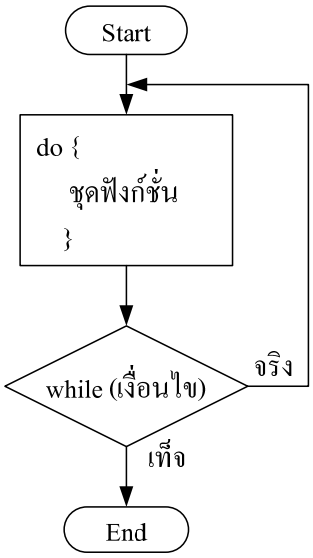


รูปที่ 2.39 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น for

ฟังก์ชั่น do while เป็นฟังก์ชั่นวนลูป ใช้ในลักษณะให้โปรแกรมทำงานวนรอบซ้ำๆ โดยจะทำตามชุดฟังก์ชั่นก่อน จากนั้นจะตรวจสอบเงื่อนไข ถ้าเป็นจริงจะทำชุดฟังก์ชั่นในลูปต่อ ถ้าเป็นเท็จจะออกจากการวนลูป

รูปแบบ

```
do
        {
        ชุดฟังก์ชั่น
        }while (เงื่อนไข);
```

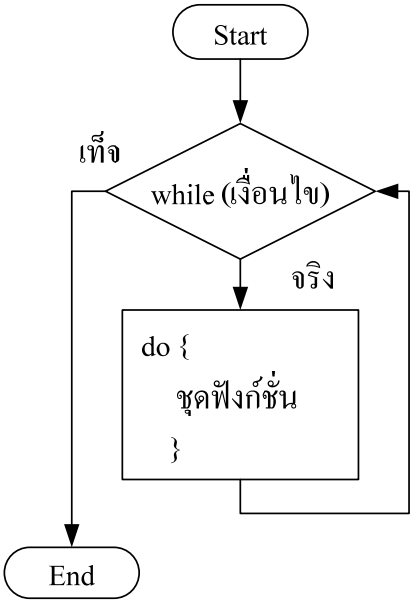


รูปที่ 2.40 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น do while

ฟังก์ชั่น while เป็นฟังก์ชั่นวนลูป ใช้ในลักษณะให้โปรแกรมทำงานวนรอบซ้ำๆ โดยจะตรวจสอบเงื่อนไขก่อน ถ้าเป็นจริงจะทำชุดฟังก์ชั่นในลูปต่อ ถ้าเป็นเท็จจะออกจากการวนลูป

รูปแบบ

```
while (ตรวจสอบเงื่อนไข)
        {
        ชุดฟังก์ชั่น
        }
```



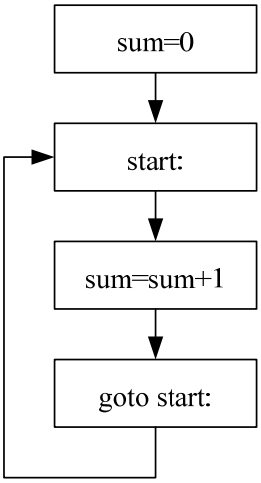
รูปที่ 2.41 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น while

ฟังก์ชั่น goto เป็นฟังก์ชั่นทำหน้าที่กระโดดไปยังตำแหน่งที่กำหนดเพื่อข้ามการทำงานหรือการวนรอบการทำงาน

รูปแบบ

```
        goto Label;
```

Label คือตำแหน่งที่จะกระโดดไป



รูปที่ 2.42 โฟลว์ชาร์ตการทำงานของฟังก์ชั่น goto

2.5.2 ภาษาแอสเซมบลี (Assembly Language) [14]

ภาษาแอสเซมบลี (Assembly Language) เป็นภาษาคอมพิวเตอร์ที่การเขียนโปรแกรมจะใช้คำย่อสั้นๆ เพื่อสื่อความหมายแทนรหัสคำสั่งของภาษาเครื่อง เพื่อให้ง่ายต่อการใช้งาน เช่น คำสั่ง MOV ย่อมากจาก MOVE หมายถึงการโอนย้ายข้อมูล เมื่อเขียนโปรแกรมเสร็จ ต้องใช้ตัวแปลภาษาที่เรียกว่า แอสเซมเบลอร์ (Assembler) ทำการแปลเป็นภาษาเครื่อง (Machine Code) หรือไฟล์ฐานสิบหก (Hexadecimal File) จากนั้นจึงนำไปเขียนลงในหน่วยความจำของไมโครคอนโทรลเลอร์ต่อไป จากรูปที่ 2.43 แสดงถึงโครงสร้างของภาษาแอสเซมบลีซึ่งประกอบด้วย ฟิลด์ต่างๆ 4 ฟิลด์ ดังนี้

Lable	**Mnemonic**	**Operand**	**Comment**
AddR1 :	ORG	0000H	; Start address
	MOV	A,#00H	; A=00H
	MOV	R1,#05H	; R1=05H
	ADD	A,R1	; A=A+R1
	DJNZ	R1,AddR1	; Jump AddR1=0
	END		

รูปที่ 2.43 โครงสร้างโปรแกรมของภาษาแอสเซมบลี

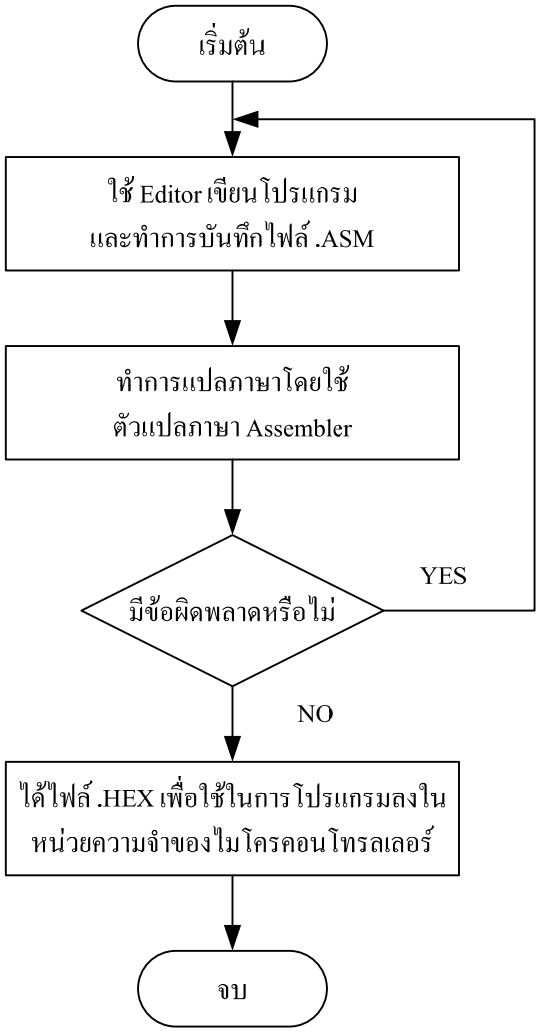
ฟิลด์ที่ 1 Label ใช้กำหนดตำแหน่งในการกระโดด หรือการเรียกโปรแกรมย่อย ขึ้นต้นด้วยตัวอักษร ปิดท้ายด้วยเครื่องหมายคอลลอน (:)

ฟิลด์ที่ 2 Mnemonic คือคำสั่งของภาษาแอสเซมบลี จะเขียนในลักษณะคำย่อ เพื่อสื่อความหมายในโปรแกรม

ฟิลด์ที่ 3 Operand คือตัวดำเนินการหรือตัวกระทำตามคำสั่ง

ฟิลด์ที่ 4 Comment คือส่วนอธิบายโปรแกรม

ขั้นตอนการเขียนโปรแกรมภาษาแอสเซมบลี จะเริ่มจากการใช้โปรแกรมสร้างข้อความ (Text Editor) เช่น Notepad หรือ WordPad ในการเขียนโปรแกรม จากนั้นจึงทำการบันทึกไฟล์ให้มีนามสกุลเป็น .ASM แล้วจึงทำการแปลภาษาหรือแอสเซมเบลอร์ (Assembler) ถ้ามีข้อผิดพลาดก็ต้องกลับไปแก้ไขโปรแกรมให้ถูกต้อง แล้วจะได้ไฟล์ที่มีนามสกุล .HEX ซึ่งเป็นไฟล์ฐานสิบหก ซึ่งเป็นภาษาเครื่อง เพื่อใช้ในการโปรแกรมลงบนหน่วยความจำของไมโครคอนโทรลเลอร์ต่อไป ดังโฟลว์ชาร์ตในรูปที่ 2.44

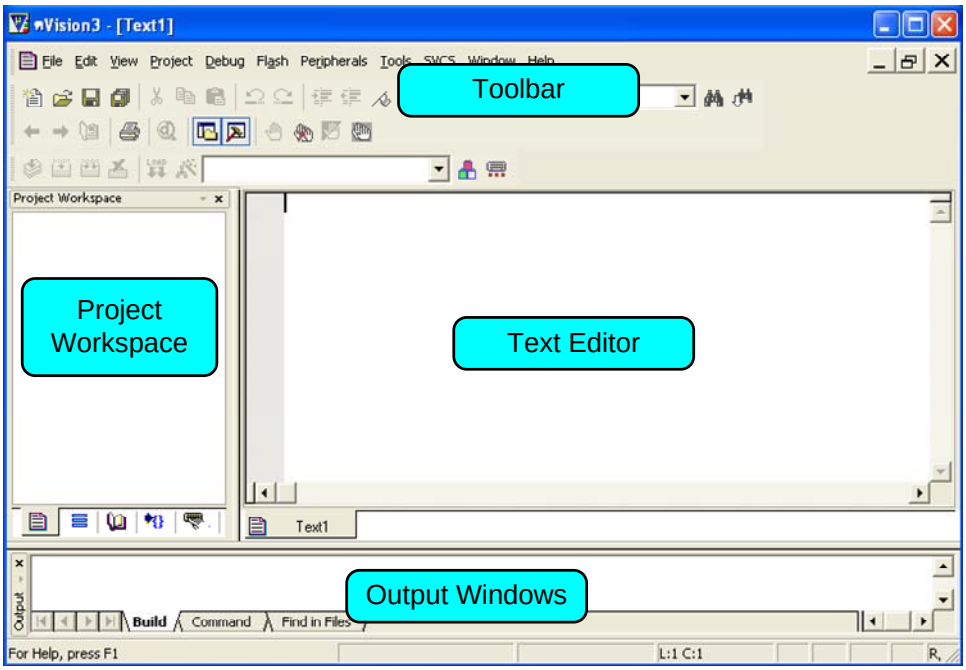


รูปที่ 2.44 โฟลว์ชาร์ตขั้นตอนการเขียนโปรแกรมภาษาแอสเซมบลีเพื่อควบคุมไมโครคอนโทรลเลอร์

2.6 ซอร์ฟแวร์ คอมไพเลอร์ และอุปกรณ์โปรแกรม

2.6.1 โปรแกรม Keil uVision3 [15]

โปรแกรม Keil uVision3 ถูกสร้างโดยบริษัท อาร์ม ลิมิเต็ด เป็นคอมไพเลอร์ภาษา C สำหรับ MCS-51 โดยสามารถเขียนและทำการคอมไพล์ไฟล์ .HEX ได้ โดยเวอร์ชั่นทดลองใช้งาน (ดาวน์โหลดฟรี) สามารถเขียนโปรแกรมรองรับความจุ 2 Kbyte



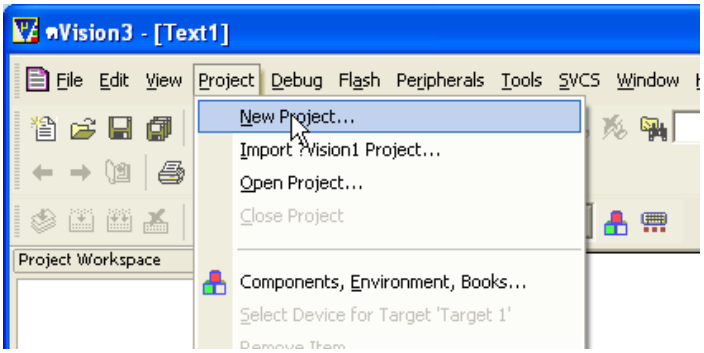
รูปที่ 2.45 ส่วนประกอบของหน้าต่าง Keil uVision3

องค์ประกอบของ Keil uVision3 ประกอบด้วย

Project Workspace	เป็นหน้าต่างแสดงโปรเจคไฟล์
Output Windows	แสดงผลการคอมไพล์ไฟล์ที่ทำการ Build Target
Text Editor	พื้นที่เขียนโปรแกรม
Toolbar	เครื่องมือของ Keil uVision3

การใช้งานก็ไม่ยุ่งยากนัก สามารถทำตามขั้นตอนได้ดังนี้

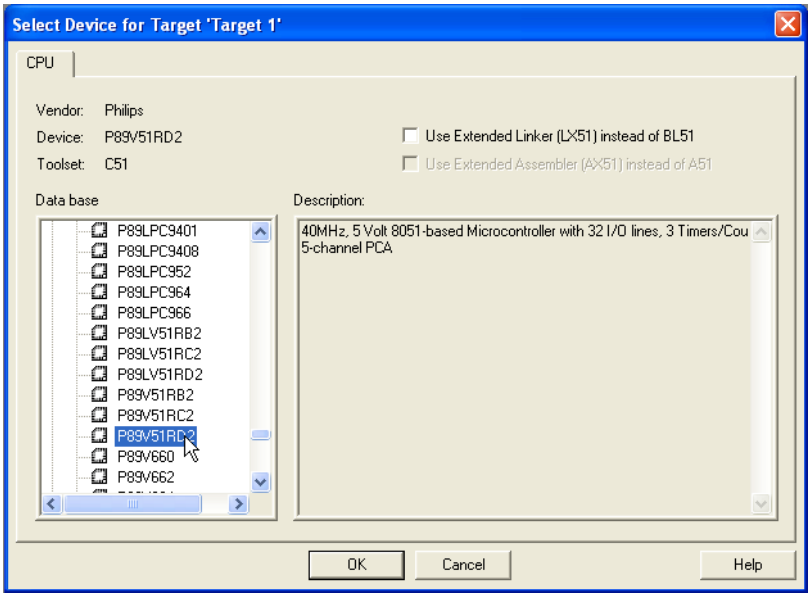
1) เปิดโปรแกรม Keil uVision3
2) สร้างโปรเจคใหม่ Project > New Project



รูปที่ 2.46 สร้าง New Project

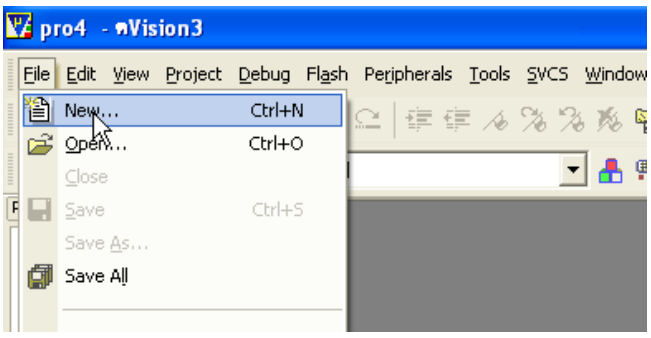
3) ตั้งชื่อโปรเจคและกำหนดตำแหน่งในการบันทึก

4) เลือกไมโครคอนโทรลเลอร์ที่ใช้งาน



รูปที่ 2.47 การเลือกไมโครคอนโทรลเลอร์

5) สร้างไฟล์โปรแกรมภาษาซี File > New



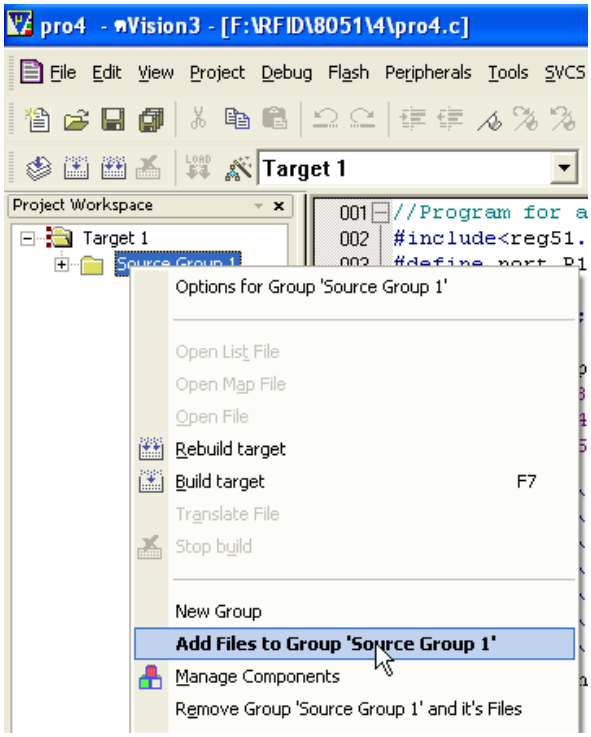
รูปที่ 2.48 การสร้างไฟล์งานภาษาซี

6) ทำการเขียนโปรแกรมภาษาซี และบันทึกเป็นไฟล์ .C

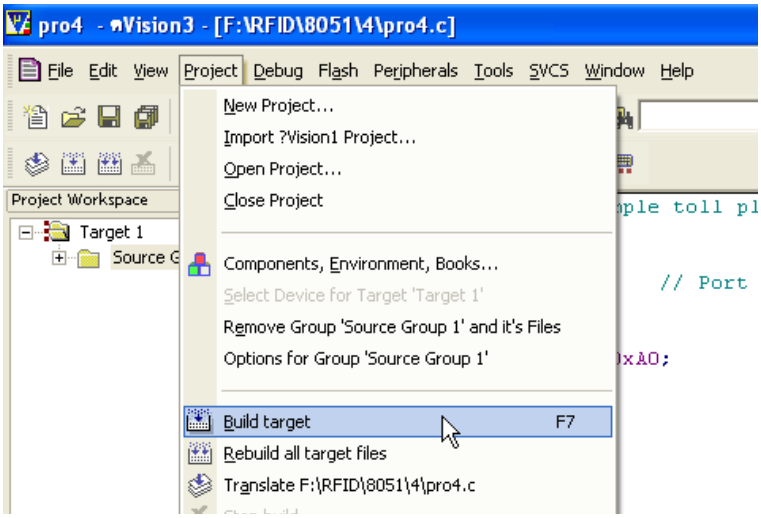
7) ทำการ Add ไฟล์ภาษาซีไว้ในโปรเจค Add File to Group ‘Source Group 1’

8) ทำการ Build target เพื่อคอมไพล์ไฟล์ที่สร้างขึ้น

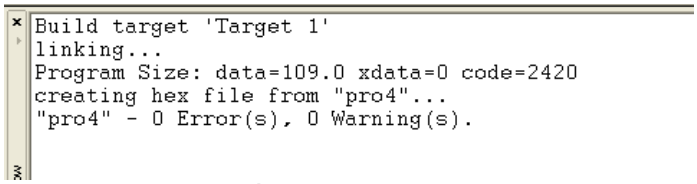
9) ผลการคอมไพล์หากไม่มีข้อผิดพลาด จะแสดงดังรูปที่ 2.51



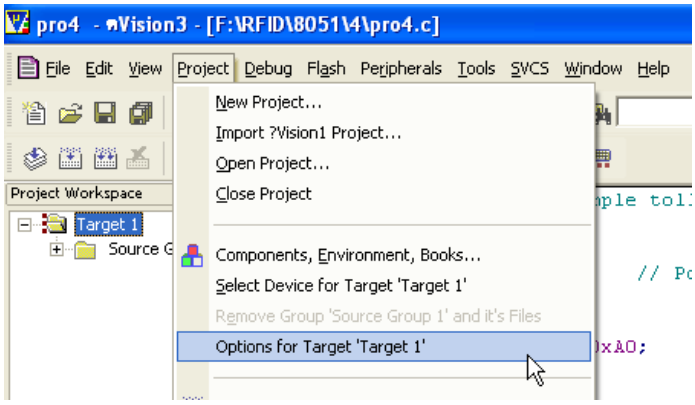
รูปที่ 2.49 การ Add File ไว้ใน Project



รูปที่ 2.50 การ Build target



รูปที่ 2.51 แสดงผลการคอมไพล์

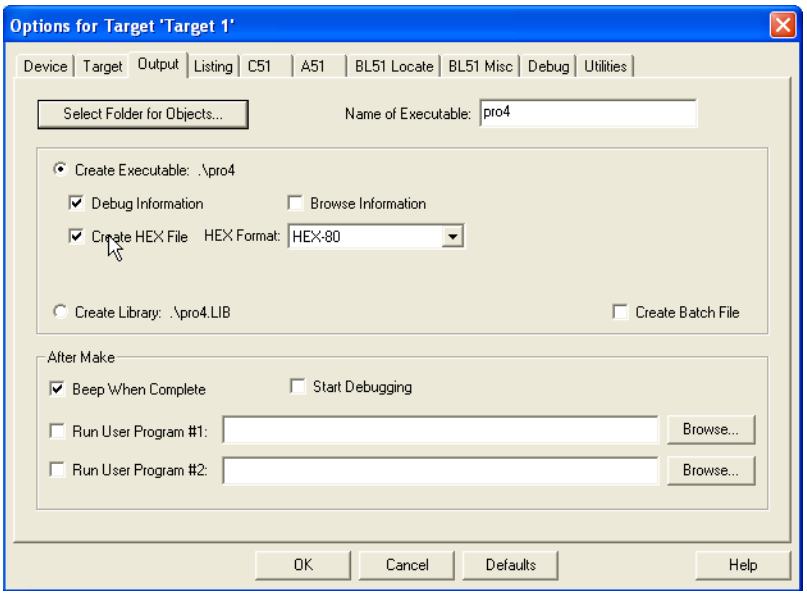


รูปที่ 2.52 แสดงหน้า Form Layout

10) ทำการสร้างไฟล์ .HEX เพื่อใช้สำหรับการโปรแกรมลงไมโครคอนโทรลเลอร์ โดยเลือก Project > Option for Target ‘Target 1’

11) ที่หน้าต่าง Options for Target เลือก Output > Create HEX File

12) ทำการ Build Target อีกครั้ง จะได้ไฟล์ .HEX ที่แปลงมาจาก .C เพื่อไปใช้งานต่อไป



รูปที่ 2.53 แสดงหน้า Form Layout

2.6.2 โปรแกรม Flash Magic [16]

Flash Magic เป็นโปรแกรมสำหรับการดาว์นโหลดไฟล์ .HEX ลงบนไมโครคอนโทรลเลอร์ ซึ่งพัฒนาโดยบริษัท NXP Semiconductor (Philips เดิม) ซึ่งเป็นผู้ผลิตไมโครคอนโทรเลอร์ MCS-51 รายใหญ่ ในกรณีที่ใช้ไมโครคอนโทรลเลอร์เบอร์ P89V51RD2 ของ Philips นั้น วิธีการ Download โปรแกรมให้กับบอร์ดจะทำได้โดยง่ายกว่าไมโครคอนโทรลเลอร์เบอร์อื่นๆ เนื่องจากสามารถใช้วงจรสื่อสารอนุกรม RS232 ในการ Download โปรแกรมได้ทันที โดยไม่ต้องจัดวงจรควบคุมอื่นๆ ให้กับไมโครคอนโทรลเลอร์อีก โดยมีขั้นตอนดังนี้

1) ต่อสายสัญญาณ RS232 ระหว่างคอมพิวเตอร์กับบอร์ดไมโครคอนโทรลเลอร์ที่ตำแหน่งของขั้วต่อสาย RS232 พร้อมกับจ่ายไฟให้กับบอร์ดเพื่อพร้อมรับคำสั่ง

2) สั่ง Run โปรแกรม Flash Magic

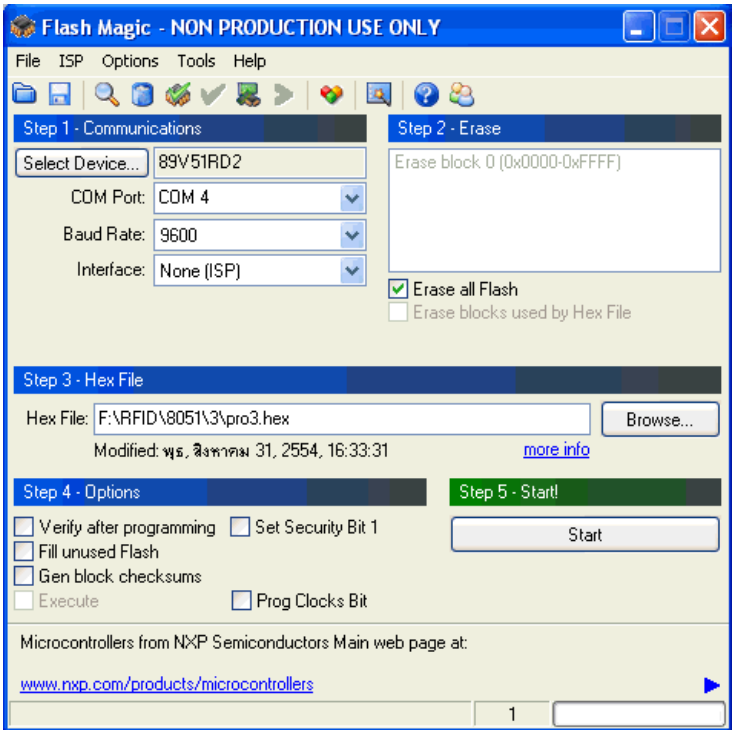
3) เลือกกำหนด Comport ตามที่ต่อสายไว้และกำหนด Baud rate เป็น 9600 หรือ 19200

4) เลือกกำหนด Device ที่ใช้ คือ P89V51RD2

5) เลือกกำหนด รูปแบบการลบข้อมูล ซึ่งถ้าไม่แน่ใจว่าไมโครคอนโทรลเลอร์ถูก Lock ไว้หรือไม่ให้เลือก Erase All Flash

6) เลือกกำหนด Hex File ที่ต้องการ Download ตามต้องการ

7) เลือก Verify After Programming



รูปที่ 2.54 หน้าต่างใช้งานโปรแกรม Flash Magic

8) เลือก Start เพื่อสั่ง Download ข้อมูลให้กับไมโครคอนโทรลเลอร์ซึ่งจะปรากฏหน้าต่างบอกให้ RESET การทำงานของไมโครคอนโทรลเลอร์ให้เริ่มต้นทำงานใน ISP Mode ดังรูป

รูปที่ 2.55 ซอร์ฟแวร์สั่งให้ทำการ Reset ไมโครคอนโทรลเลอร์

9) ให้ทำการกดสวิตช์ RESET ที่บอร์ดไมโครคอนโทรลเลอร์ หน้าต่างจะหายไป และสังเกตเห็นโปรแกรมเริ่มต้นทำงาน ตามขั้นตอนต่างๆที่เลือกไว้ ซึ่งในขั้นตอนนี้ให้รอจนเสร็จ

10) ให้ทำการกดสวิตช์ RESET ที่บอร์ดไมโครคอนโทรลเลอร์อีกครั้งหนึ่งเพื่อให้ไมโครคอนโทรลเลอร์เริ่มต้นทำงาน ซึ่งจะสังเกตเห็นไมโครคอนโทรลเลอร์เริ่มต้นทำงานตามโปรแกรมที่ได้ทำการ Download ไปแล้วทันที

2.6.3 โปรแกรม MPLAB IDE และ HI-TECH C [17]

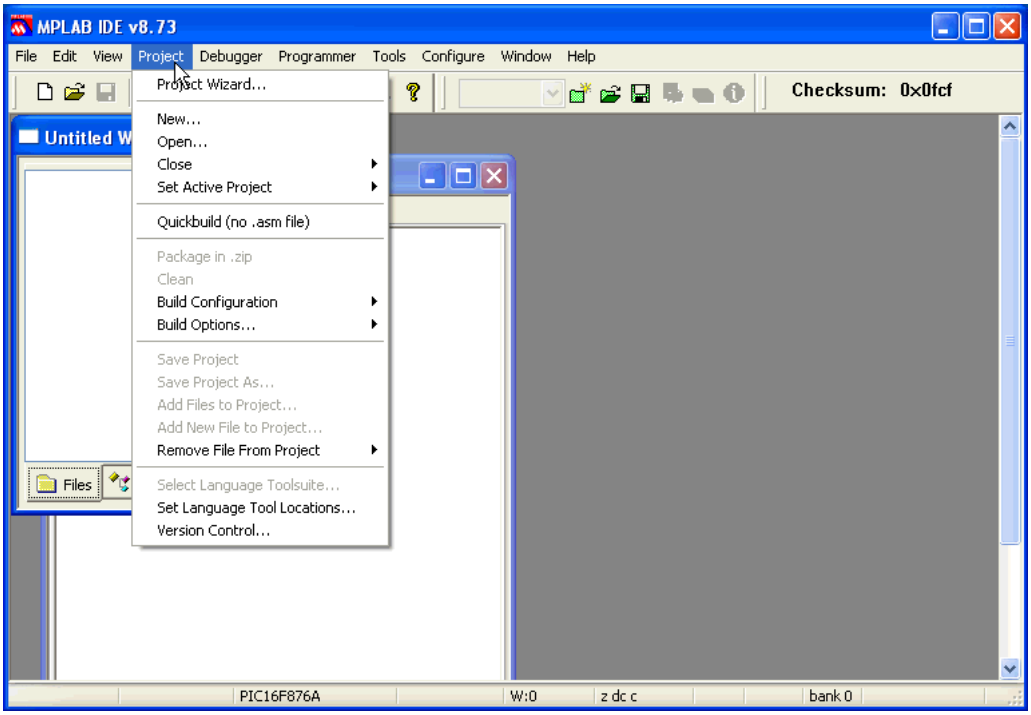
MPLAB IDE เป็นซอร์ฟแวร์ที่บรรจุเครื่องมือในการเขียนโปรแกรมไว้ครบถ้วนทั้ง Text Editor สำหรับเขียนทั้งภาษา C และ Assembly

HI-TECH C Complier เป็นโปรแกรมแปลภาษา C เป็นภาษาเครื่อง ซึ่งจะผนวกมากับ MPLAB IDE

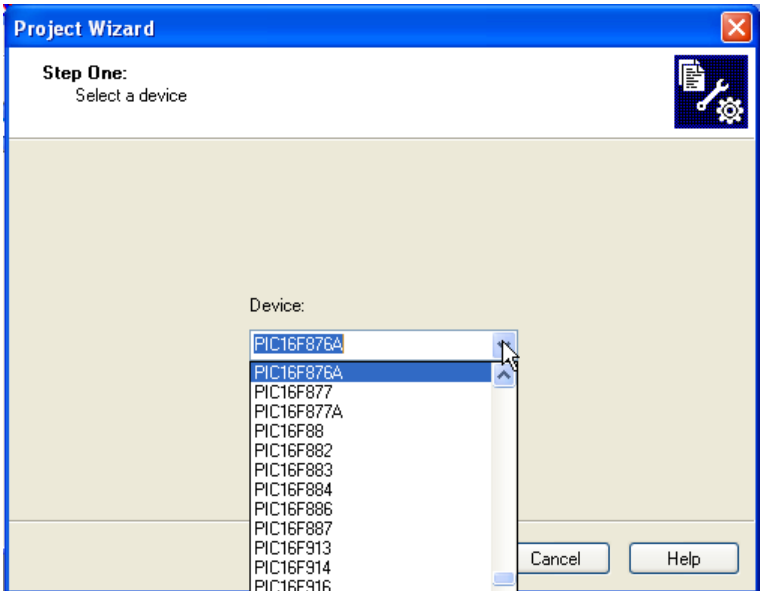
สำหรับการใช้งานสามารถอธิบายอย่างคร่าวๆ พร้อมรูปประกอบดังนี้

1) สร้างโปรเจค Project > Project Wizard
2) จะเข้าสู่หน้าต้อนรับให้เลือก Next
3) ทำการเลือกไมโครคอนโทรลเลอร์ที่ใช้งาน
4) เลือก Complier ซึ่ง MPLAB IDE ได้ตั้งค่าเป็น HI-TECH C เป็น default ไว้แล้ว เลือก Next
5) ทำการสร้าง New project ตั้งชื่อ
6) ไปที่ File > New เพื่อเปิด Text Editor
7) เขียนโปรแกรมภาษา C
8) ทำการ Add โปรแกรมภาษา C ใน Project
9) ทำการคอมไพล์ และสร้างไฟล์ .HEX

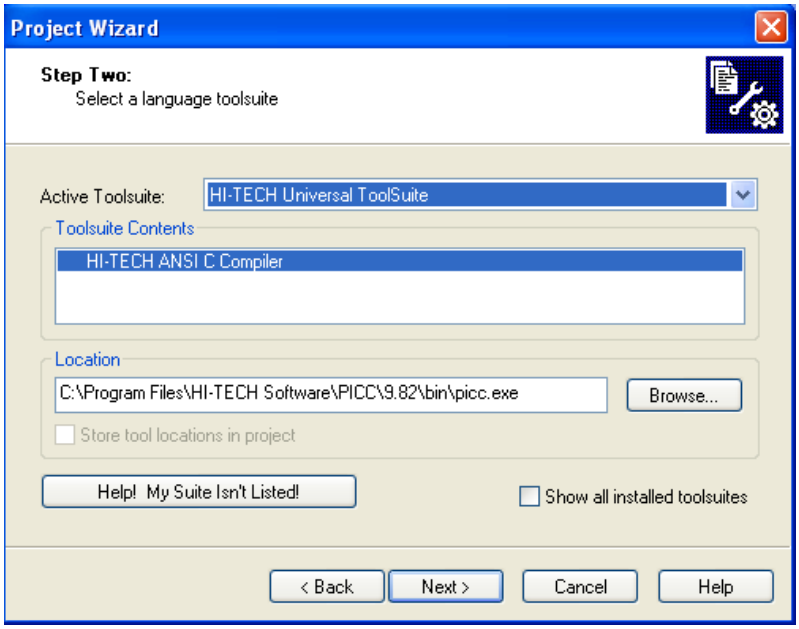
เมื่อทำผ่านทุกขั้นตอนก็จะได้ไฟล์ .HEX มาเพื่อโปรแกรมลงบนไมโครคอนโทรลเลอร์ ซึ่งอาจใช้ซอร์ฟแวร์โปรแกรม PICkit2 ก็ได้



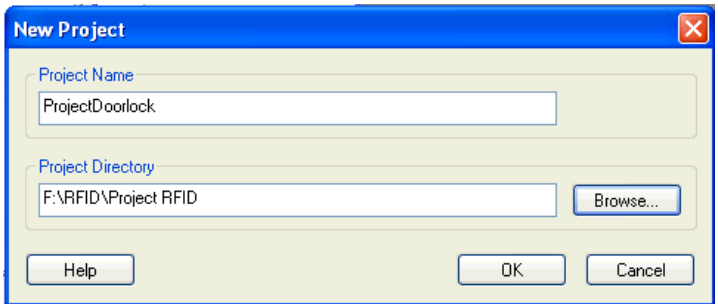
รูปที่ 2.56 การสร้าง Project



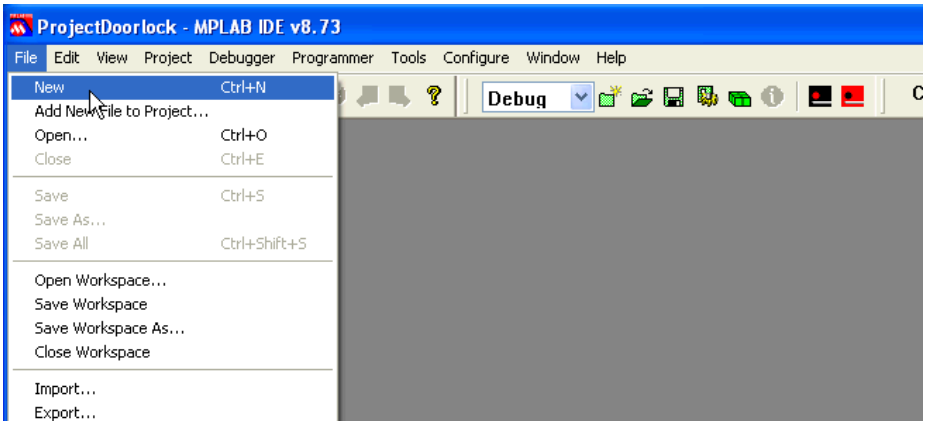
รูปที่ 2.57 การเลือกเบอร์ไมโครคอนโทรลเลอร์



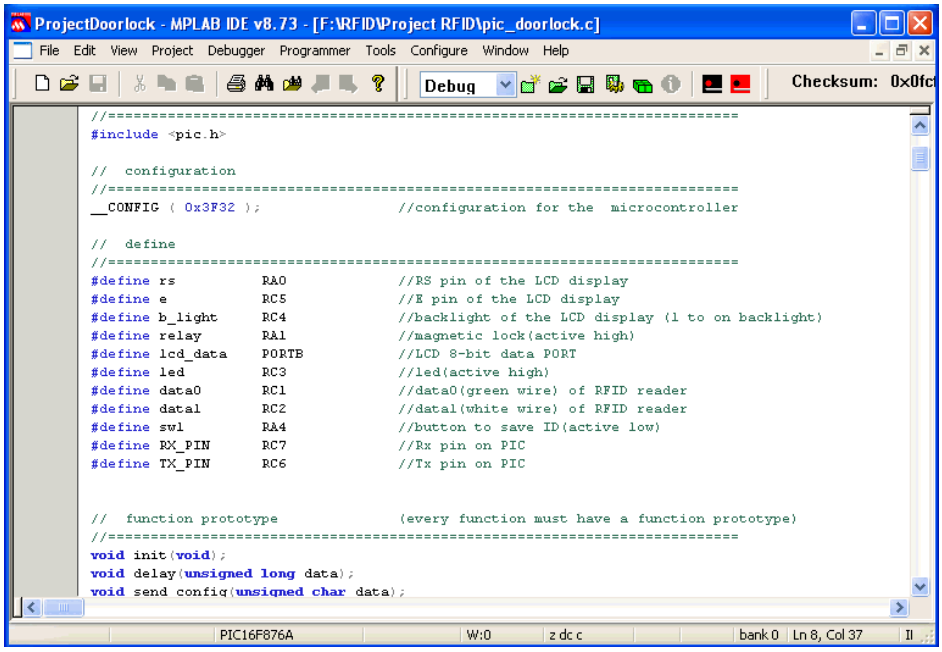
รูปที่ 2.58 การเลือก Complier



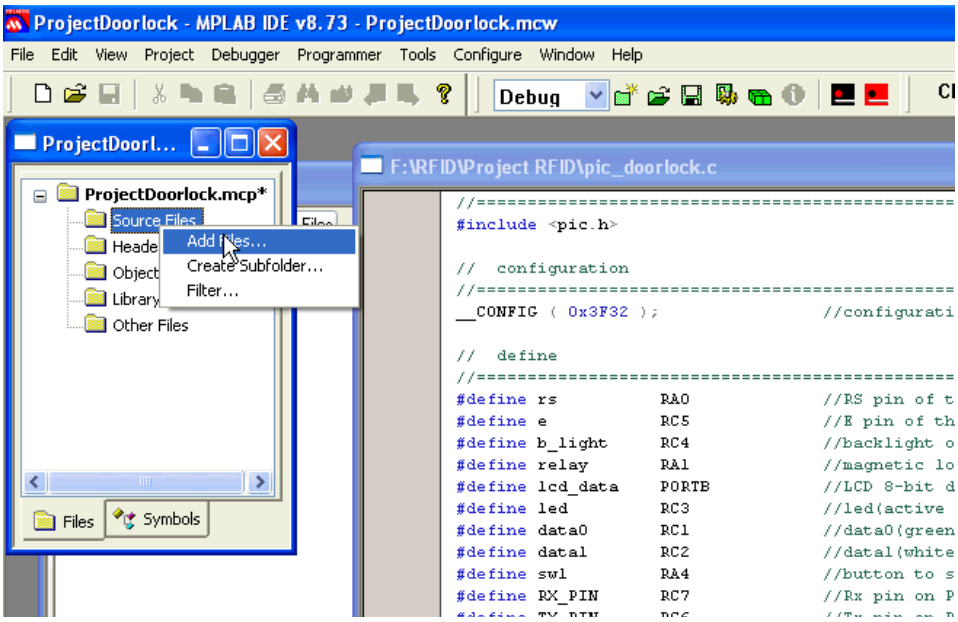
รูปที่ 2.59 สร้าง New Project



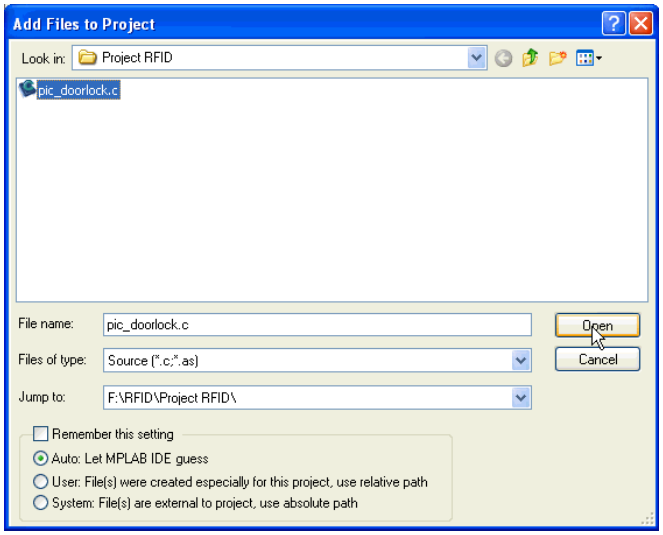
รูปที่ 2.60 เปิด Text Editor



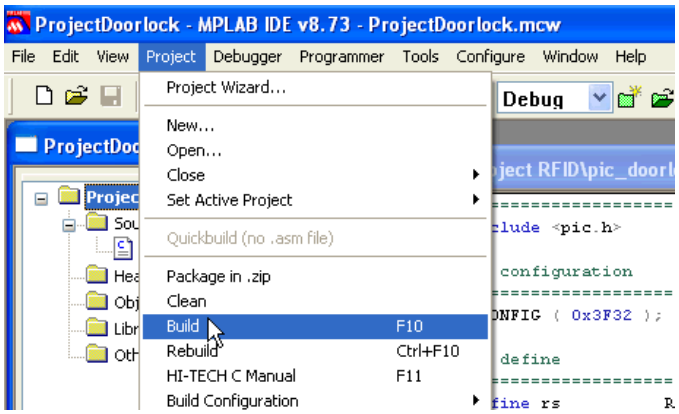
รูปที่ 2.61 เขียนโปรแกรมภาษา C



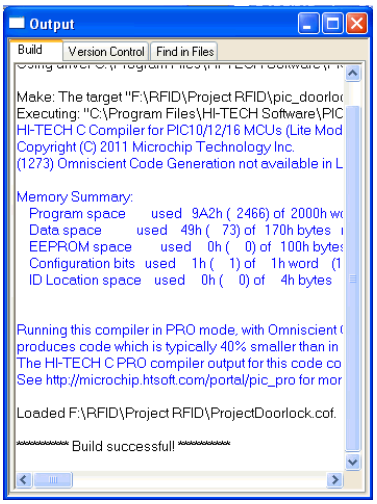
รูปที่ 2.62 การเพิ่มไฟล์ภาษา C ใน Project



รูปที่ 2.63 เลือกไฟล์ที่จะเพิ่มใน Project



รูปที่ 2.64 ทำการ Build ไฟล์ .HEX



รูปที่ 2.65 ผลการคอมไพล์

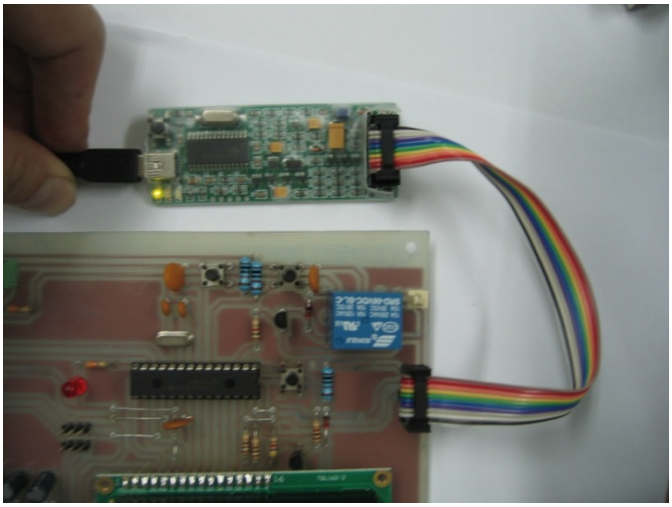
2.6.4 โปรแกรม PICkit2 [18]

PICkit2 เป็นซอร์ฟแวร์สำหรับโหลดโปรแกรมลงในไมโครคอนโทรลเลอร์ PIC พัฒนาโดยบริษัท Microchip ผู้ผลิต PIC และเป็น Freeware สามารถโหลดโปรแกรมได้หลายเบอร์ โดยใช้งานร่วมกับบอร์ดโหลดโปรแกรมผ่านพอร์ต USB การใช้งานโปรแกรม PICkit2 มีขั้นตอนดังนี้

1) ทำการเชื่อมต่ออุปกรณ์ PIC Programmer กับเครื่องคอมพิวเตอร์ด้วยสาย USB LED สีเขียวจะติด

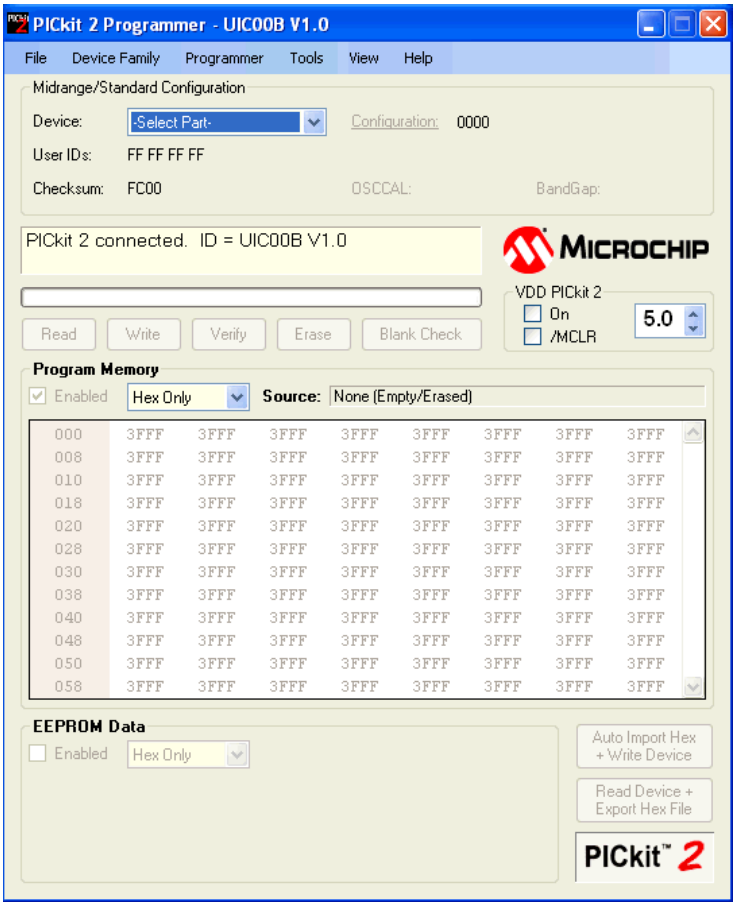
รูปที่ 2.66 การเชื่อมต่อกับสาย USB

2) ต่อสายโปรแกรมเข้ากับชุดทดลอง



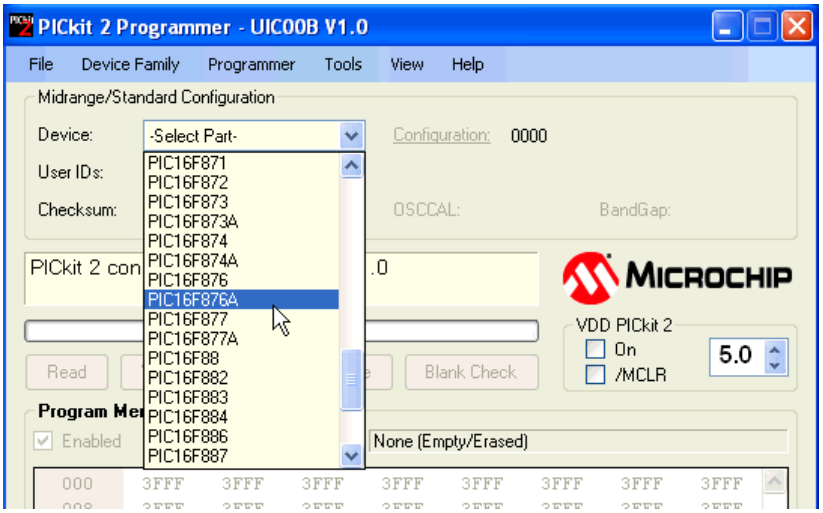
รูปที่ 2.67 การเชื่อมต่อกับบอร์ดชุดทดลอง

3) เปิดโปรแกรม PICkit2 จะได้หน้าจอดังรูปที่ 2.68 โดยโปรแกรมจะตรวจสอบการสื่อสารโดยอัตโนมัติ



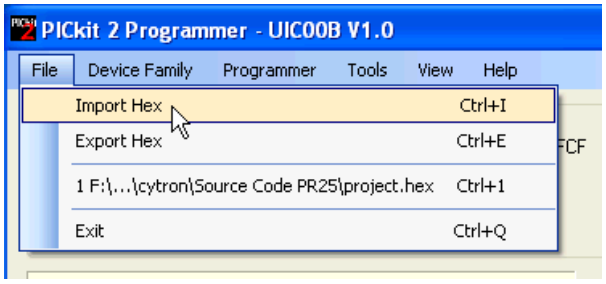
รูปที่ 2.68 หน้าจอ PICkit2

4) ทำการเลือก Device ให้ตรงกับที่ใช้



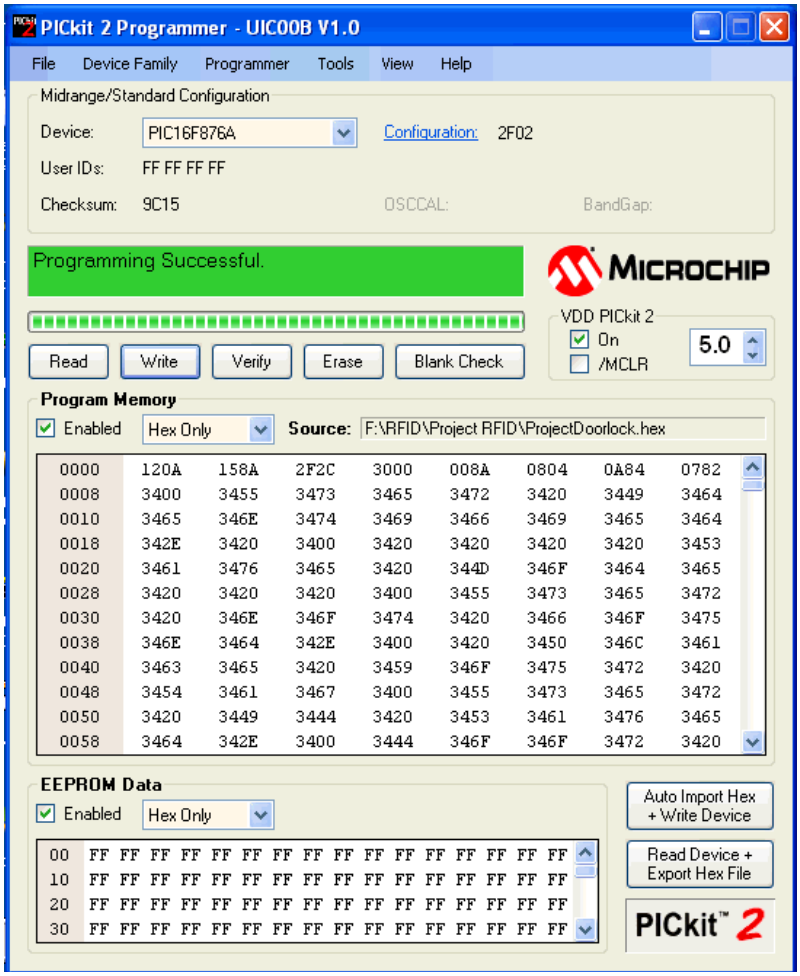
รูปที่ 2.69 เลือกเบอร์ไมโครคอนโทลเลอร์

5) ทำการ Import ไฟล์ .HEX ที่จะโหลดลงบน PIC



รูปที่ 2.70 Import Hex file

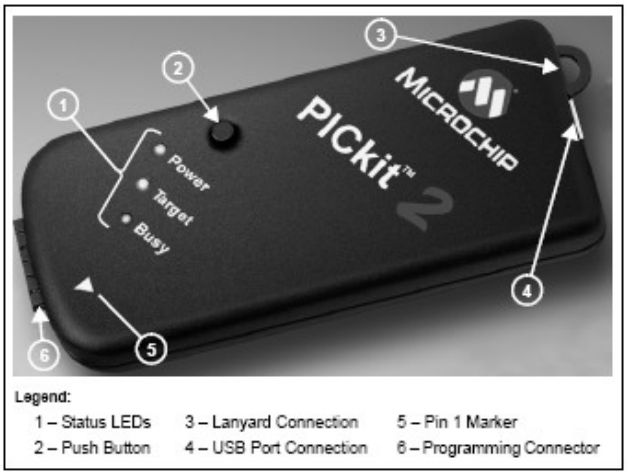
6) กอปุ่ม Write รอจนโปรแกรมเสร็จ จะแสดงข้อความ Programming Successful



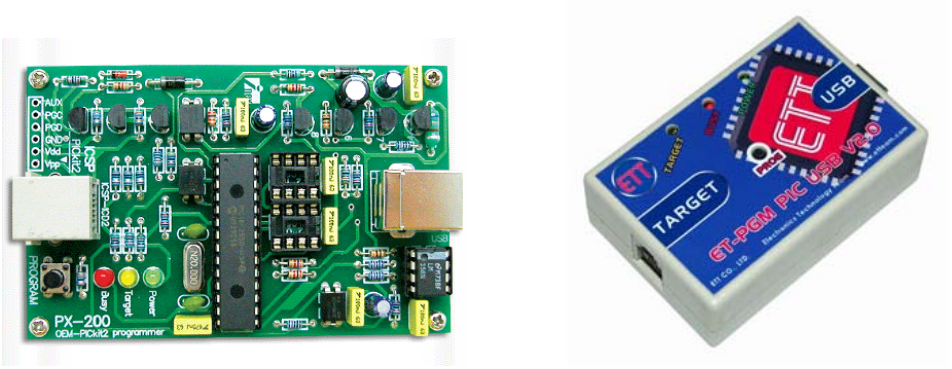
รูปที่ 2.71 การโปรแกรมเสร็จสมบูรณ์

2.6.5 อุปกรณ์โปรแกรม

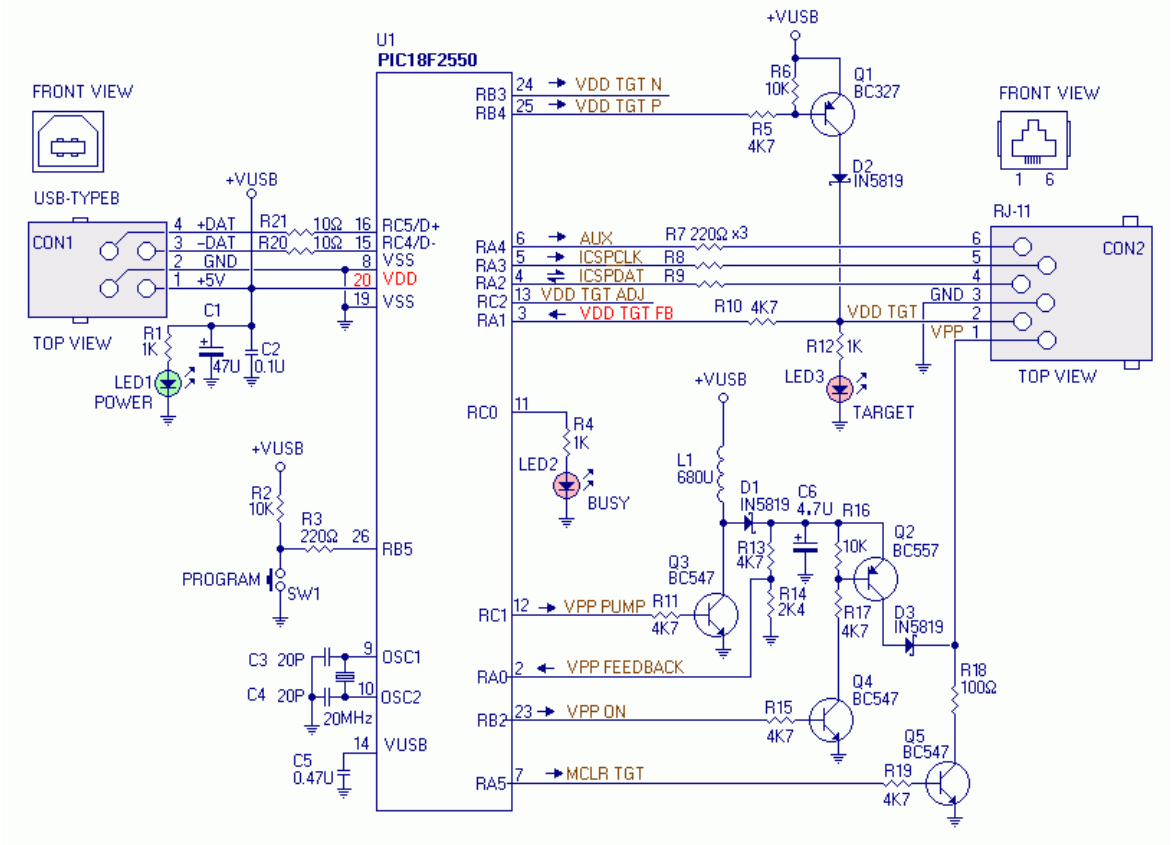
อุปกรณ์ดาวน์โหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์ ดังรูปที่ 2.72 ถึง 2.73 ที่จริงแล้วก็เป็นวงจรที่คล้ายคลึงกัน ต่างๆกัน เพียงผู้ผลิต แต่ละชุดล้วนสามารถใช้งานโหลดโปรแกรม .HEX ลงบนไมโครคอนโทรลเลอร์ PIC ได้โดยสื่อสารกับคอมพิวเตอร์ผ่านพอร์ต USB แต่ทั้งนี้หากผู้ใช้งานไม่มีอุปกรณ์โปรแกรม เหล่านี้ ก็สามารถสร้างขึ้นใช้งานเองได้ โดยใช้วงจรในรูปที่ 2.74 ซึ่งอาจยุ่งยากบ้างเล็กน้อย เนื่องจากวงจรจะมีไมโครคอนโทรลเลอร์เบอร์ PIC18F2550 ที่จะต้องทำการโหลด Firmware จากผู้ผลิต (Microchip) ลงบน PIC18F2550 ก่อนจึงจะใช้งานได้ โดยการโหลด Firmware สามารถกระทำผ่านพอร์ตอนุกรม RS232 (Com Port) ได้



รูปที่ 2.72 อุปกรณ์ PIC Programmer รุ่น PICkit2 ของ Microchip [18]



รูปที่ 2.73 อุปกรณ์ PIC Programmer รุ่น PX200 ของ inex [19] และ รุ่น ET-PGM PIC USB V2 ของ ETT [20]



รูปที่ 2.74 วงจรเครื่องโปรแกรม PIC ผ่าน USB (PICkit2 Compatible) [21]