บทที่2
ไมโครคอนโทรลเลอร์ตระกูล 8051

2.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์มีหลักการในการทำงานคล้ายกับไมโครโปรเซสเซอร์ เพียงแต่ไมโครคอนโทรลเลอร์มักจะมีองค์ประกอบจำพวก หน่วยความจำ และพอร์ตควบคุมอยู่ในไอซีเพียงตัวเดียวเท่านั้น บางครั้งเราอาจจะเรียกไมโครคอนโทรลเลอร์ว่าเป็น ไมโครคอมพิวเตอร์ชิพเดี่ยว (1 chip microcomputer) แต่เมื่อเทียบกับไมโครโปรเซสเซอร์ หน่วยการทำงานภายในไมโครคอนโทรลเลอร์นั้นสามารถใช้งานได้อย่างค่อนข้างจำกัดมากกว่า มีหน่วยความจำชนิด รอม หรืออีพรอม(ROM or EPROM) ภายในขนาดไม่เกิน 4 กิโลไบต์ หน่วยความจำชนิดแรม (RAM) ขนาด 256 ไบต์เท่านั้น และมีพอร์ตขนาดประมาณ 3 ถึง 4 พอร์ต เป็นต้น หากว่ามีความต้องการที่จะเพิ่มเติมจำนวนของหน่วยการทำงานเหล่านี้ ก็จำเป็นต้องนำขาสัญญาณพอร์ตไปใช้เป็นขาสัญญาณบัสแอดเดรส และบัสข้อมูล เพื่อใช้งานกับอุปกรณ์มาตรฐานภายนอกได้

ลักษณะงานที่เหมาะกับการนำไมโครคอนโทรลเลอร์ไปใช้งาน มักเป็นงานประยุกต์ที่เกี่ยวกับการควบคุม หรือการจัดการสัญญาณอินพุต/เอาท์พุต ของวงจรอิเล็กทรอนิกส์ และวงจรดิจิตอลแบบต่างๆ เช่น ระบบแสดงผล หรือระบบเตือนภัย ระบบภายในเครื่องใช้ไฟฟ้า เป็นต้น ซึ่งงานควบคุมเหล่านี้ไม่มีการคิดคำนวณที่ซับซ้อนมากนัก และต้องการพื้นที่ของแผงวงจรควบคุมที่จำกัด

2.1.1 ไมโครคอนโทรลเลอร์ตระกูล 8051

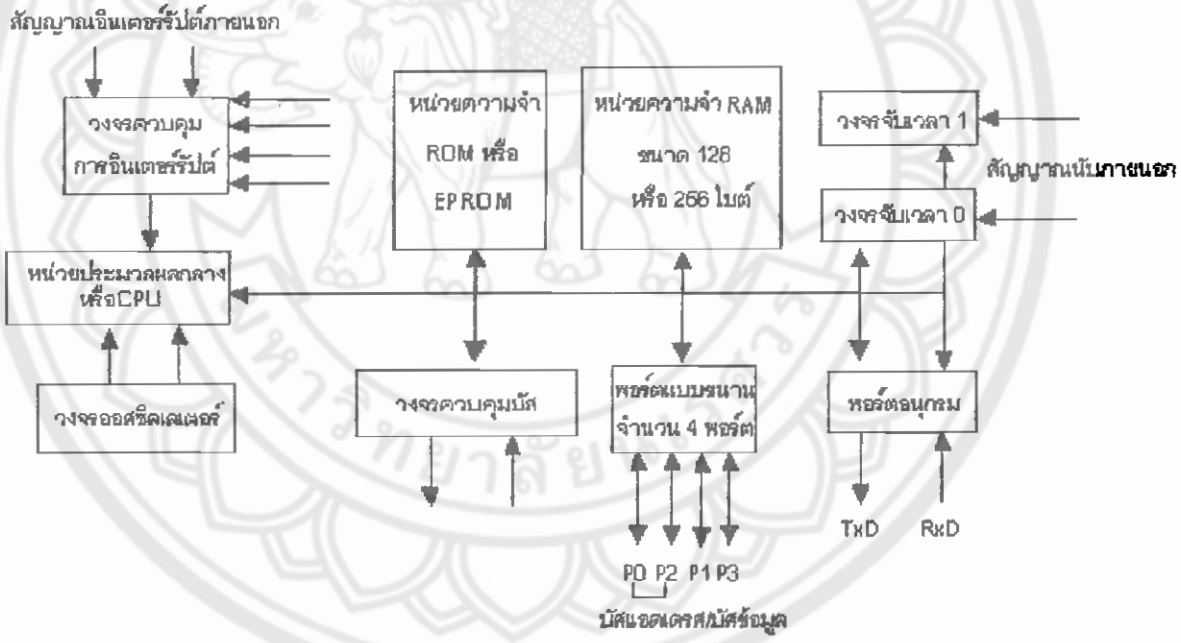
บรรดาไมโครคอนโทรลเลอร์ที่มีการผลิตจากบริษัทต่างๆ จำนวนมากนั้น ไมโครคอนโทรลเลอร์จากบริษัทอินเทล(Intel Cooperation) ในตระกูลเอ็มซีเอส-51 (MCS-51) ได้มีการนำไปใช้งานกันอย่างแพร่หลายมากนับตั้งแต่ปี ค.ศ.1980 เป็นต้นมา

คุณสมบัติพื้นฐานของไมโครคอนโทรลเลอร์ 8051

จากรูป 2.1 แสดงให้เห็นหน่วยการทำงานพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ต่างๆที่อยู่ในตระกูลเอ็มซีเอส – 51 ซึ่งประกอบด้วย

- **หน่วยประมวลผลกลาง**
- **หน่วยประมวลผลสำหรับข้อมูลแบบบิต(Boolean Processor)**
- **ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์**

- **หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบอีพรอม(เบอร์ 8751)หรือแบบรอม(เบอร์8051)**
- **หน่วยความจำแบบแรม ภายใน 28 ไบต์**
- **พอร์ตอินพุต/เอาท์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ**
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวน 2 วงจร
- วงจรสื่อสารแบบอนุกรมแบบ ฟูลดูเพล็กซ์ (Full Dulplex)
- วงจรควบคุมอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภทเพราะมีการกำหนดลำดับความสำคัญได้สองระดับ
- วงจรออสซิลเลเตอร์ภายใน



รูปที่ 2.1 แผนภาพบล็อกแสดงหน่วยทำงานพื้นฐานของเอ็มซีเอส – 51

(ที่มา : การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 สุนทร วิทูสุรพจน์)

2.1.2 หน่วยความจำของไมโครคอนโทรลเลอร์ 8051

- หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมภายใน : มีหน่วยความจำแบบรอมขนาด 4 กิโลไบต์ประกอบอยู่

ภายในไอซี และเหมาะสมกับการนำไปใช้ในวงจรทางอุตสาหกรรมที่มีจำนวนการผลิตมาก เนื่องจากมีผลทำให้ต้นทุนการผลิต ค่าใช้จ่ายในการผลิตต่อหน่วยลดลงมาก

หน่วยความจำโปรแกรมภายนอก : เป็นการใช้หน่วยความจำอีพรอม (หรือ รอม) เชื่อมต่อเข้ากับระบบของ 8051 เนื่องมาจากราคาถูก เหมาะกับการทำงานต้นแบบเพื่อพัฒนาโปรแกรม

- หน่วยความจำข้อมูล

หน่วยความจำข้อมูลภายใน : 8051 มีหน่วยความจำภายในจำนวนทั้งหมด 256 ไบต์โดยแบ่งออกเป็น 2 ลักษณะ คือ พื้นที่เฉพาะของหน่วยประมวลผลกลางใช้งานเท่านั้น ซึ่งเรามักจะเรียกกันว่า รีจิสเตอร์ และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา

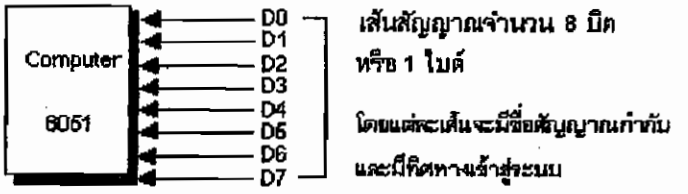
หน่วยความจำข้อมูลภายนอก : การใช้หน่วยความจำภายนอกเป็นวิธีการแก้ปัญหาอย่างหนึ่งในกรณีที่มีความต้องการหน่วยความจำสำหรับการเก็บข้อมูลชั่วคราว หรือตัวแปรเกินกว่าหน่วยความจำของข้อมูลภายใน ซึ่งมีขนาดเพียง 128 ถึง 256 ไบต์เท่านั้น บางครั้งการใช้หน่วยความจำข้อมูลภายนอกยังเหมาะกับงานประยุกต์บางอย่างที่จำเป็นต้องมีการเก็บสำรองข้อมูลบางอย่างเอาไว้ไม่ให้สูญหาย แม้ว่าจะไม่มีการจ่ายไฟให้กับระบบ

2.1.3 พอร์ตอินพุต /เอาท์พุต ของ 8051

พอร์ต หมายถึง แอดเดรสหนึ่งที่ได้รับการกำหนดไว้เพื่อถ่ายโอนข้อมูลระหว่างไมโครคอนโทรลเลอร์ กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อขึ้นอยู่กับทิศทางการไหลของข้อมูล เมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก ดังรูป 2.2 ดังนั้นการนำเข้าข้อมูลจากภายนอกเรียกว่า การอินพุต (input) และในกรณีตรงกันข้ามเพื่อส่งออกข้อมูลก็เรียกว่า การเอาท์พุต (output)

รูปที่ 2.2 ก การส่งผ่านข้อมูลผ่านทางพอร์ตเอาท์พุตซึ่งเป็นเส้นสัญญาณ 8 เส้นไปให้อุปกรณ์ภายนอก

(ที่มา: การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 สุนทร วิทูสุรพจน์)



รูปที่ 2.2 ข การส่งข้อมูลจากอุปกรณ์ภายนอกเข้ามายังพอร์ตอินพุตของระบบ
ซึ่งมีลักษณะเป็นเส้นสัญญาณ 8 เส้น
(ที่มา : การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 สุนทร วิทูสุรพจน์)

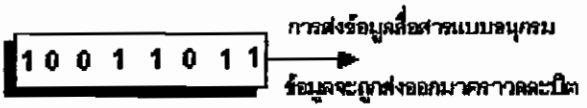
2.1.4 กระบวนการรับ และส่งข้อมูลอนุกรมของ 8051

กระบวนการรับและส่งข้อมูลอนุกรมของ 8051 จะเริ่มต้นขึ้น ภายหลังเมื่อมีการเขียนข้อมูลลงในรีจิสเตอร์ SBUF ข้อมูลนี้จะถูกจัดการด้วยวิธีการทางฮาร์ดแวร์ในการเลื่อนบิตและส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ออกส่งออกครบถ้วนแล้ว จึงจะทำการกำหนดค่าของแฟล็ก TI ให้เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้รีจิสเตอร์ SBUF ว่าง และพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้ว ในกรณีที่ผู้เขียนข้อมูลใหม่ในรีจิสเตอร์ SBUF โดยไม่รอให้แฟล็ก TI มีค่าเป็น 1 ก่อนจะมีผลทำให้ข้อมูลที่ส่งออกไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่าบิต REN (Recierver Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีบิตของข้อมูลถูกส่งเข้ามาจากภายนอกระบบฮาร์ดแวร์ของ 8051 จึงจะทำการเลื่อนบิตเหล่านี้เข้ามาโดยอัตโนมัติ และเมื่อบิตสุดท้ายเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายมาเก็บยังรีจิสเตอร์ SBUF และทำการกำหนดให้แฟล็ก RI ให้มีค่าเป็น 1 ซึ่งมีผลทำให้เกิดการอินเตอร์รัปต์โปรแกรมขึ้น

2.2 การสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรม เป็นการรับหรือการส่งข้อมูลในลักษณะของบิตหรือกลุ่มของบิตครางละหนึ่งบิตเป็นลำดับเรื่อยไปจนสิ้นสุด การสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมาก เนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกันจึงจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียง สองหรือสาม เส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก ลองพิจารณาเปรียบเทียบระหว่างการสื่อสาร 2 ประเภท



รูปที่ 2.3 ก ข้อมูลสื่อสารแบบอนุกรม ข้อมูล 1 ไบต์จะถูกส่งออกมาครางละบิต เป็นลำดับไป จนครบทั้ง 8 จำนวน
(ที่มา : การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 สุนทร วิทูสุรพจน์)

รูปที่ 2.3 ข ข้อมูลสื่อสารแบบขนาน ข้อมูลแต่ละบิตภายใน 1 ไบต์จะถูกส่งออกมาพร้อมกันในลักษณะแบบขนาน
(ที่มา : การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 สุนทร วิทูสุรพจน์)

2.2.1 รูปแบบของข้อมูลอนุกรม

วิธีการที่จะทำให้ข้อมูลสื่อสารอนุกรมมีความถูกต้องมากยิ่งขึ้นจะใช้การเพิ่มเติมบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง ได้แก่

2.2.1.1 บิตเริ่มต้น (Start Bit)

บิตเริ่มต้นมีหน้าที่สำหรับการบ่งบอกให้วงจรฮาร์ดแวร์ด้านรับทราบถึง ตำแหน่งจุดเริ่มต้นของบิตข้อมูลกลุ่มใหม่ เพื่อที่จะทำการปรับจังหวะของสัญญาณการรับข้อมูลให้ตรงกัน ดังนั้นบิตเริ่มต้นจึงถูกเพิ่มเข้าไปก่อนมีการส่งข้อมูลจริง ตามปกติแล้วค่าของบิตเริ่มต้นมักจะเป็นระดับลอจิกที่ตรงข้ามกับระดับลอจิกของสภาวะของสายสื่อสารขณะเมื่อมีการส่งข้อมูล (Idle State) ตัวอย่าง เช่น หากสภาวะของสายเมื่อไม่มีข้อมูลอยู่เป็นลอจิกสูง บิตเริ่มต้นก็จะเป็นระดับลอจิกต่ำ เป็นต้น

2.2.1.2 บิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่ (Parity Bit)

บิตนี้มีหน้าที่เพื่อการตรวจสอบความถูกต้องของข้อมูล โดยทั่วไปมักจะเรียกว่า บิตพาริตี้

และจะนำไปแทรกต่อท้ายข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของบิตข้อมูลที่เป็น 1 ซึ่งจะเป็นได้ 2 ลักษณะคือ พาริตี้คู่ (Even Parity) หรือพาริตี้คี่ (Odd Parity) ตัวอย่างเช่น ระบบที่ติดต่อกันโดยระบุว่าจะใช้พาริตี้คู่ (Even Parity) ทางด้านส่งจะนำข้อมูลที่ส่งมาพิจารณา หากจำนวนบิตที่มีค่าเป็น 1 เป็นเลขจำนวนคู่อยู่แล้วค่าของบิตพาริตี้จะมีค่าเป็นศูนย์ แต่หากว่าจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคี่ ค่าของบิตพาริตี้จะมีค่าเป็น 1 การพิจารณาทางด้านรับเป็นการตรวจสอบบิตที่มีค่าเป็น 1 ของข้อมูลที่ได้รับมาทั้งหมดรวมทั้งบิตพาริตี้ ถ้ามีค่าเป็นเลขคู่ แสดงว่าข้อมูลที่รับมาถูกต้อง แต่หากไม่เป็นเลขคู่แสดงว่าเกิดความผิดพลาดของข้อมูลขึ้น เป็นต้น

2.2.1.3 บิตสุดท้าย (Stop Bit)

บิตสุดท้ายเป็นบิตที่เพิ่มขึ้นเพื่อระบุขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตสุดท้ายอาจมีจำนวนมากกว่า 1 บิตก็ได้ คือ 1 บิต 1 ½ บิต และ 2 บิต ดังนั้นกรณีของการส่งข้อมูล 8 บิตพร้อมทั้งบิตที่เพิ่มเข้าไปโดยสมบูรณ์คือ บิตเริ่มต้น บิตพาริตี้ และบิตสุดท้าย รวมทั้งสิ้น 12 บิต

2.2.2 ฮาร์ดแวร์ที่เกี่ยวข้องกับการสื่อสารข้อมูล

อุปกรณ์ที่ใช้ในการสื่อสารข้อมูลจะเป็นแบบอะซิงโครนัส เนื่องจากไมโครคอมพิวเตอร์ส่วนมากใช้การสื่อสารแบบนี้

2.2.2.1 พอร์ตอะซิงโครนัสหรือ พอร์ต RS-232C

พอร์ต RS-232C นี้ทำหน้าที่รับส่งข้อมูลในแบบอนุกรมเรียกว่า "Universal Asynchronous Adapter" เหตุที่มีชื่อเรียกว่า RS-232C ก็เนื่องจากสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของอเมริกาเหนือ EIA ได้กำหนดมาตรฐานของอุปกรณ์สื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS-232C

ขั้วต่อมาตรฐานทั่วไปเป็นแบบ 25 ขา หรือแบบ DB25 แต่บางแบบอาจใช้ขั้วต่อแบบ 9 ขา อีก 16 ขาไม่ใช้เพราะการสื่อสารแบบอะซิงโครนัสไม่จำเป็นต้องใช้สัญญาณ 16 ขาที่หายไป ลักษณะการใช้งานเพียง 9 ขาก็มีสัญญาณเพียงพอที่สามารถทำการสื่อสารได้
หน้าที่สำคัญของการสื่อสารแบบอะซิงโครนัสคือ
การรับสัญญาณ

1. เปลี่ยนสัญญาณเข้ามาแบบอนุกรมให้เป็นขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่ได้รับ
3. ตัดบิตสุดท้ายและบิตพาริตี้ออก
4. ส่งสัญญาณให้ซีพียูรู้ว่ารับสัญญาณแล้ว

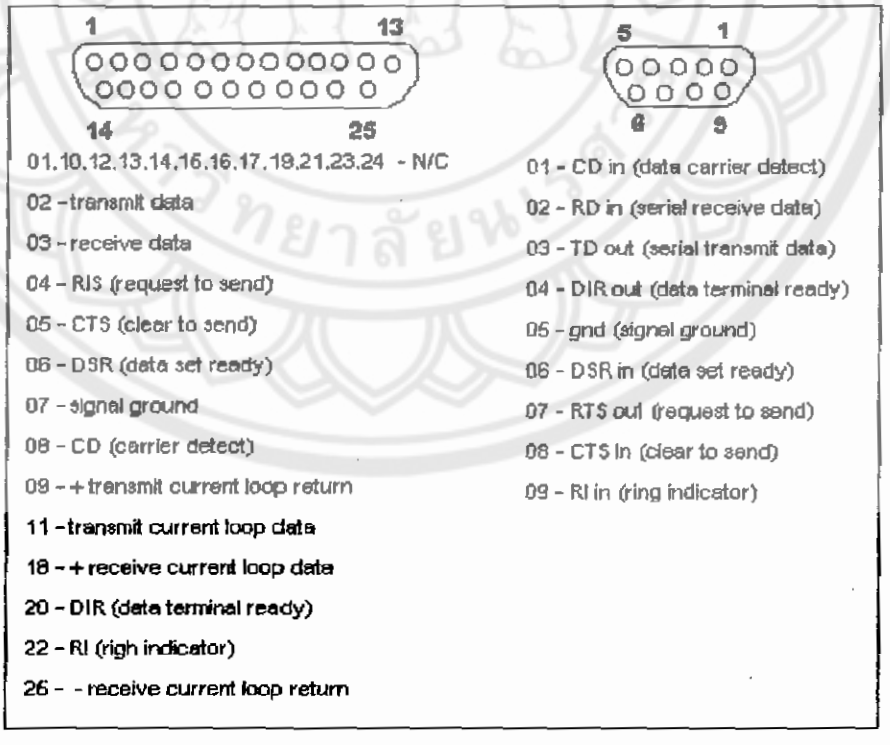
การส่งสัญญาณ

1. **เปลี่ยนสัญญาณขนานจากซีพียูแล้วค่อยทยอยส่งออกเป็นแบบอนุกรม**
2. **เพิ่มบิตสุดท้ายและบิตพาริตี้**
3. **เพิ่มสัญญาณควบคุมโมเด็มที่เชื่อมต่อ (ถ้ามี)**

2.2.2.2 สายสัญญาณ

สายสัญญาณเป็นปัจจัยที่สำคัญอย่างหนึ่งในการสื่อสารผ่านพอร์ตอนุกรม RS-232 เนื่องจากเป็นสิ่งเชื่อมโยงให้เครื่องต้นทาง และปลายทางสามารถติดต่อถึงกันได้ การส่งข้อมูลระหว่างคอมพิวเตอร์ให้ได้ผลลัพธ์ที่สมบูรณ์ ควรใช้สาย Null Modem ที่มีฉนวนหุ้มสายที่อยู่ด้านในเพื่อป้องกันสัญญาณรบกวน แต่ในกรณีที่ไม่สามารถหาสายดังกล่าวได้เราสามารถสร้างสายสัญญาณขึ้นได้เอง แต่สายแบบนี้จะมีสัญญาณรบกวนค่อนข้างสูงเพราะไม่มีฉนวนหุ้ม การต่อสายสัญญาณสามารถทำได้โดยใช้สายโทรศัพท์แบบ 3 เส้นสัญญาณ มีลักษณะเป็นสายกลม

การต่อสายสัญญาณ สิ่งที่ต้องทำความเข้าใจก่อน คือ "ขาสัญญาณ" ของหัวต่อทั้งสองชนิด สำหรับขาสัญญาณที่นำไปใช้งานมีรายละเอียดดังนี้



รูปที่ 2.4 ขาสัญญาณของ RS-232 แบบ 25 ขา และ 9 ขา

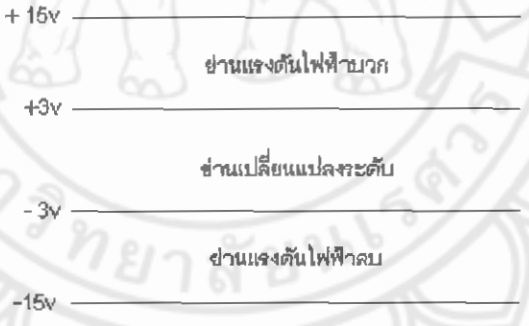
(ที่มา : Delphi Episode II เทคนิคและการพัฒนาโปรแกรมด้วยเดลไฟ ประพนธ์ อัศวภานุวัฒน์)

2.2.3 ลักษณะของสัญญาณ RS-232C

เพื่อเป็นหลักประกันว่าข้อมูลถูกส่งออกไปอย่างถูกต้องและอุปกรณ์ถูกควบคุมอย่างถูกต้อง จำเป็นต้องมีข้อตกลงในเรื่องของสัญญาณที่ใช้ มาตรฐาน RS-232C กำหนดย่านของแรงดันไฟฟ้าในสัญญาณเพื่อสนองจุดประสงค์ข้างต้นดังแสดงในตาราง 2.1 และรูปที่ 2.5

ตารางที่ 2.1 มาตรฐานของการใช้แรงดันไฟฟ้า
(ที่มา: การสื่อสารข้อมูลและไมโครคอมพิวเตอร์เน็ตเวอร์ค ไพศาล สงวนหมู่ ,ยืน ภู่วรวรรณ)

มาตรฐานของการใช้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	สถานภาพลิจิก	สถานภาพของสัญญาณ	ฟังก์ชันในการควบคุม
บวก	0	สเปซ(space)	OFF
ลบ	2	มาร์ค(Mark)	OFF

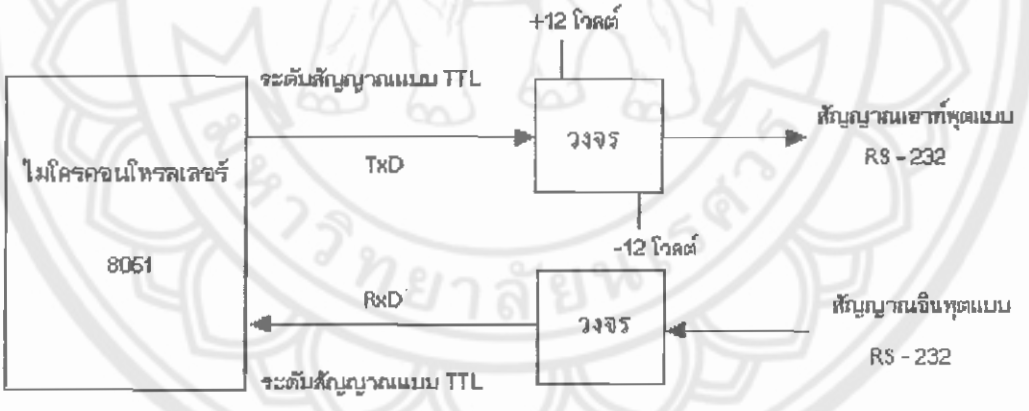


รูปที่ 2.5 ย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS-232C
(ที่มา: การสื่อสารข้อมูลและไมโครคอมพิวเตอร์เน็ตเวอร์ค ไพศาล สงวนหมู่ ,ยืน ภู่วรวรรณ)

บนสายข้อมูล(เช่น สาย 2 และ สาย 3)แรงดันไฟฟ้าบวกแสดงถึงค่าลอจิก 0 และแรงดันไฟฟ้าลบแสดงถึงค่าลอจิก 1 บนสาย Handshaking (เช่น DIR และ DSR) แรงดันไฟฟ้าบวกแสดงว่าส่งข้อมูลได้ ส่วนแรงดันไฟฟ้าลบหมายถึงหยุดส่งข้อมูล

2.2.4 การเชื่อมต่อแบบมาตรฐาน RS-232C

ในการเชื่อมต่อแบบอนุกรมเข้ากับอุปกรณ์คอมพิวเตอร์ต่างๆ เช่น คอมพิวเตอร์ เทเล็กซ์ หรือโทรพิมพ์ เป็นต้น มักจะกำหนดใช้การเชื่อมต่อตามมาตรฐาน RS-232C ทั้งนี้เพื่อให้มีการใช้งานเส้นสัญญาณหรือรูปแบบของตัวเชื่อมต่อที่สอดคล้องกัน จะได้ลดปัญหาการเข้ากันไม่ได้ ระหว่างสัญญาณของอุปกรณ์ที่มาเชื่อมต่อกันทั้งสองด้านให้น้อยลง เนื่องจากระดับโวลเตจที่ใช้และการแทนความหมายของระดับลอจิกตามมาตรฐานนี้แตกต่างไปจากที่ใช้งานกันในระบบดิจิตอลทั่วไป โดยระดับสัญญาณของ RS –232C เป็นแบบไบโพลาร์(Bipolar) ระดับโวลเตจทางด้านลบช่วง –3V ถึง –15V แทนค่าลอจิก 1 และโวลเตจทางด้าน +3V ถึง +15V แทนค่าลอจิก 0 ดังนั้นจะเห็นได้ว่ามีความจำเป็นต้องเพิ่มเติมอุปกรณ์หรือวงจรพิเศษเข้าไป เพื่อเปลี่ยนระดับโวลเตจจากระดับ 0V ถึง +5V จากขาสัญญาณของ 8051เป็นระดับโวลเตจที่สูงกว่าค่า +3V หรือต่ำกว่า–3 V ดังรูป 2.6 ซึ่งแสดงให้เห็นระดับสัญญาณแบบ TTL จากขาสัญญาณ TxD และ RxD ของ 8051 จะต้องถูกปรับเปลี่ยนไปเป็นระดับสัญญาณ RS-232 ก่อนที่จะทำการส่งออกไปในสายนำสัญญาณต่อไป

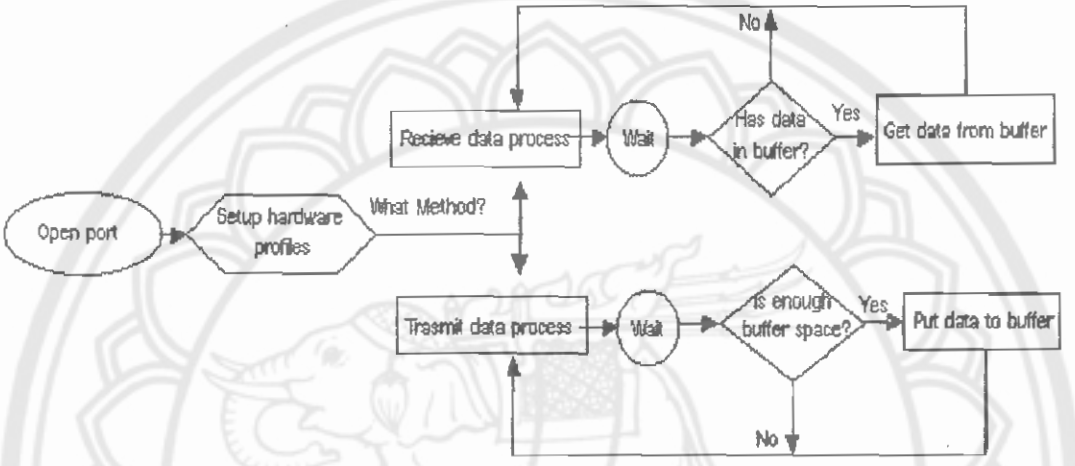


รูปที่ 2.6 แนวการเปลี่ยนแปลงระดับสัญญาณแบบ TTL จากไมโครคอนโทรลเลอร์ 8051 ไปเป็นระดับสัญญาณ RS-232 และมีการเปลี่ยนแปลงระดับสัญญาณอินพุตแบบ RS –232 ไปเป็น**ระดับสัญญาณแบบ TTL ก่อนที่จะได้เชื่อมต่อกับขาสัญญาณของไมโครคอนโทรลเลอร์ 8051**

(ที่มา : การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 สุนทร วิทูสุรพจน์)

2.2.5 หลักการทำงาน

ในการใช้งานพอร์ตสื่อสาร ได้ลำดับขั้นตอนการดำเนินการออกเป็นส่วนย่อยๆ เพื่อใช้บริการพอร์ตสื่อสารกับระบบวินโดวส์ มีลำดับการทำงานดังต่อไปนี้



รูปที่ 2.7 RS – 232 Serial communication scheme

(ที่มา : Delphi Episode II เทคนิคและการพัฒนาโปรแกรมด้วยเดลไฟ ประพนธ์ อัศวภานุวัฒน์)

ตารางที่ 2.2 หลักการทำงานของพอร์ตสื่อสาร

(ที่มา : Delphi Episode II เทคนิคและการพัฒนาโปรแกรมด้วยเดลไฟ ประพนธ์ อัศวภานุวัฒน์)

เครื่องต้นทางทำหน้าที่ส่งข้อมูล	เครื่องปลายทางทำหน้าที่รับข้อมูล
1. เปิดพอร์ตสื่อสาร 2. กำหนดค่าเบื้องต้น 3. ตรวจสอบอินพุตบัฟเฟอร์ (Input Buffer(Rx)) ว่ามีข้อมูลอยู่หรือไม่? 4. ถ้ามี ทำการเคลื่อนย้ายข้อมูลออกจากบัฟเฟอร์ ดำเนินการประมวลผล และรอรับข้อมูลชุดต่อไป	1. เปิดพอร์ตสื่อสาร 2. กำหนดค่าเบื้องต้น 3. ตรวจสอบว่ามีเอาท์พุตบัฟเฟอร์ (Output Buffer (Tx)) เพียงพอที่จะส่งข้อมูลหรือไม่ ? 4. ถ้ามี ดำเนินการส่งข้อมูล

2.3 สเต็ปเปอร์มอเตอร์ (Stepper Moter)

สเต็ปเปอร์มอเตอร์มีความแตกต่างจากมอเตอร์ทั่วไปโดย เมื่อป้อนกำลังไฟฟ้าให้กับมันๆ จะหมุนตามเส้นรอบวงและหยุด ซึ่งต่างจากมอเตอร์ทั่วๆไปซึ่งจะหมุนทันทีและตลอดเวลา สเต็ปเปอร์มอเตอร์สามารถกำหนดตำแหน่งของการหมุนด้วยตัวเลขได้โดยละเอียดโดยการใช้คอมพิวเตอร์เป็นตัวกำหนด

สเต็ปเปอร์มอเตอร์สามารถใช้งานได้ในระบบเปิด (Open Loop System) นั่นคือสามารถทำงานได้โดยละเอียดโดยไม่ต้องการสัญญาณป้อนกลับ (Feedback) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งได้อย่างถูกต้อง จำเป็นต้องมีการป้อนกลับไปให้ระบบรับรู้ และจะต้องบอกได้ว่าตำแหน่งถูกต้องแล้วหรือผิดพลาด (error)

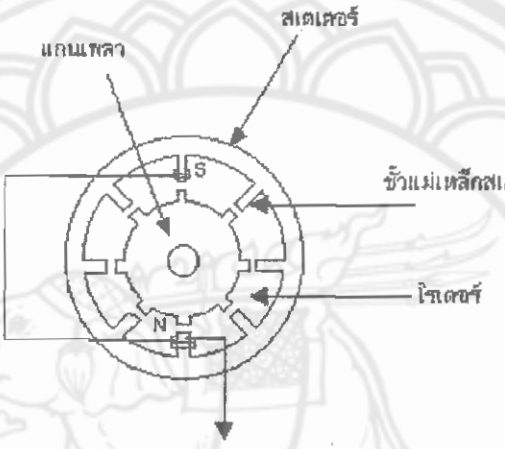
วิธีหนึ่งที่ใช้กันโดยทั่วไปกับสเต็ปเปอร์มอเตอร์ก็คือ การใช้สวิตซ์ติดตั้งไว้ที่ตำแหน่งที่ต้องการตรวจจับ (Limit Switch) เมื่อสเต็ปเปอร์มอเตอร์เริ่มหมุนและหมุนจนกระทั่งถึงตำแหน่งของสวิตซ์ตรวจจับสัญญาณก็จะป้อนกลับเข้าสู่ระบบและทราบการทำงานของสเต็ปเปอร์มอเตอร์ได้ตลอดเวลา ซึ่งโดยปกติในวงจรคอนโทรลเลอร์จะมีการกำหนดจุดอ้างอิง (Referance Point) ไว้ด้วย เพื่อให้เริ่มต้นทำงานและอ้างอิงตำแหน่งได้อย่างถูกต้อง

โครงสร้างของขั้วแม่เหล็กโดยสเตเตอร์ประกอบขึ้นจากแผ่นเหล็กวงแหวนที่มีซี่ยื่นออกมา แต่ละซี่เหล่านั้นจะมีคอยล์สวมพันอยู่ ดังนั้นเมื่อป้อนกระแสไฟฟ้าผ่านคอยล์ทำให้เกิดสนามแม่เหล็กไฟฟ้า (electromagnetic) ขึ้น ด้านตรงข้ามของแต่ละขั้วแม่เหล็กได้รับกระแสไฟฟ้าในขณะเดียวกัน แต่จะไหลวนในทิศทางตรงกันข้ามทำให้เกิดสนามแม่เหล็กในทิศตรงกันข้ามขึ้น ดังแสดงในรูปที่ 2.8 ดังนั้นการเพิ่มจำนวนของขั้วแม่เหล็กมากขึ้นจะเพิ่มจำนวนสเต็ปต่อวงรอบมากขึ้นตามไปด้วย

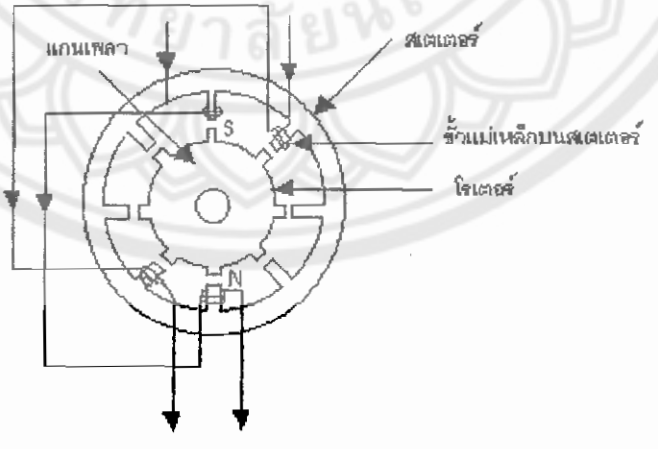
อย่างไรก็ตาม ผู้ใช้สามารถเพิ่มจำนวนสเต็ปต่อรอบได้อีกวิธีหนึ่งโดยไม่ต้องปรับเปลี่ยนโครงสร้างภายใน โดยทำการจ่ายกำลังไฟฟ้าไปยังขั้วแม่เหล็กสองขั้วที่อยู่ใกล้กันในเวลาเดียวกัน ซึ่งจะทำให้โรเตอร์หยุดหมุนอยู่ระหว่างกลางของสองขั้วแม่เหล็กนั้น หรือเคลื่อนที่ไปครึ่งสเต็ปเท่านั้นและวิธีการนี้ยังช่วยให้เกิดแรงบิด (torque) มากขึ้นด้วย ดังแสดงในรูป 2.9

สเต็ปเปอร์มอเตอร์โดยทั่วไปมีจำนวนของขั้วแม่เหล็กหรือจำนวนสเต็ปต่อรอบเป็นจำนวนมาก ปกติอยู่ที่ประมาณ 100-400 สเต็ปต่อรอบ การมีจำนวนสเต็ปมากๆ นี้ ไม่ได้เพิ่มที่จำนวนขั้วแม่เหล็กไฟฟ้าที่สเตเตอร์ แต่ทำโดยการเพิ่มจำนวนซี่ขั้วแม่เหล็กที่โรเตอร์ จำนวนสเต็ปต่อรอบทั้งหมดจะได้จากการคูณขั้วแม่เหล็กบนสเตเตอร์และจำนวนซี่ที่โรเตอร์ เช่น ถ้ามีขั้วแม่เหล็ก 3 ขั้วบนสเตเตอร์และ 8 ซี่ที่ขั้วแม่เหล็กของโรเตอร์ สเต็ปเปอร์มอเตอร์ตัวนี้ทำงานที่ 24 สเต็ปต่อรอบหรือหมุนเป็นมุม 15 องศาต่อรอบ

การใช้วงจรดิจิตอลคอนโทรลเลอร์กำหนดการจ่ายกำลังไฟฟ้าเข้าสู่ขดลวดสเตเตอร์แบบซีเควนเชียลทำให้สามารถควบคุมการเคลื่อนที่ทุกสเต็ปได้เช่นเดียวกับการควบคุมในวงจรดีซีเซอร์โว(DC servo) แต่การควบคุมดิจิตอลไม่จำเป็นต้องมีการป้อนกลับ การเคลื่อนที่ทุกสเต็ปได้จากการคำนวณเรอบหรือมุมนการหมุนที่ต้องการ แล้วจึงส่งข่อมูลที่ได้ไปควบคุมการหมุนของมอเตอร์ พิกัดในการทำงาน อาทิ ความเร็ว มุมในการเคลื่อนที่ ตำแหน่งของเพลาถูกกำหนดจากข้อมูลที่ส่งมาควบคุม



รูปที่ 2.8 สเต็ปเปอร์มอเตอร์ที่มีต่อการต่อวงจรขดลวดภายในเพื่อกระตุ้นการให้เกิดขั้วแม่เหล็กขึ้น 1 ขั้ว ในทิศทางตรงกันข้าม ส่วนขดลวดอื่นๆ จะไม่ถูกกระตุ้นเลย
(ที่มา : รวมโครงงานอิเล็กทรอนิกส์ไมโครคอมพิวเตอร์ ไกรวุฒิ โรจน์ประเสริฐสุด)

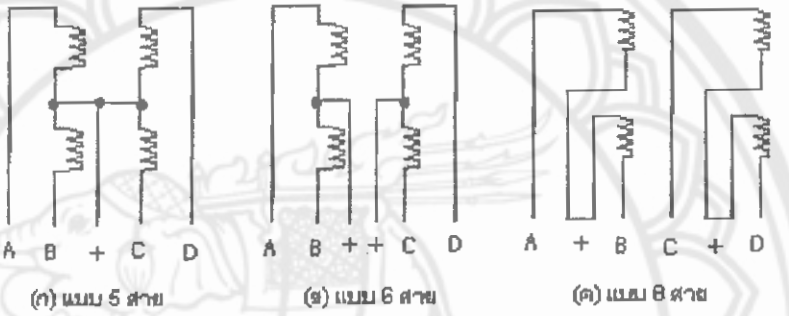


รูปที่ 2.9 การต่อวงจรขดลวดแบบกระตุ้นให้เกิดขั้วแม่เหล็กพร้อมกัน 2 ขั้วที่อยู่ใกล้กันทำให้โรเตอร์เคลื่อนที่มาหยุดอยู่ระหว่างขั้วแม่เหล็กทั้งสอง
(ที่มา : รวมโครงงานอิเล็กทรอนิกส์ไมโครคอมพิวเตอร์ ไกรวุฒิ โรจน์ประเสริฐสุด)

สเต็ปเปอร์มอเตอร์สามารถแบ่งออกได้เป็น 2 ชนิดคือ

1. แบบไบโพลาร์ (Bipolar motors) ภายในมอเตอร์จะมีขดลวดจำนวน 2 ขดเท่านั้น การควบคุมให้มอเตอร์เปลี่ยนทิศทางทำได้โดยการเปลี่ยนทิศทางการไหลของกระแสที่ป้อนให้กับขดลวดตามจังหวะที่เหมาะสม มอเตอร์ประเภทนี้จะมีสายต่อใช้งานเพียง 4 สาย

2. แบบยูนิโพลาร์ (Unipolar motors) มอเตอร์ประเภทนี้จะมีการแท็ป(trap) ที่จุดกลางของขดลวดทั้งสองขด ทำให้สามารถแบ่งขดลวดภายในออกได้เป็น 4 ชุด สายเชื่อมต่อภายนอกอาจมีได้ตั้งแต่ 5,6 หรือ 8 สาย ดังรูปที่ 2.10



(ก) แบบ 5 สาย (ข) แบบ 6 สาย (ค) แบบ 8 สาย

รูปที่ 2.10 การเชื่อมต่อวงจรภายในของสเต็ปเปอร์มอเตอร์ชนิดยูนิโพลาร์ในแบบต่างๆ
(ที่มา : เซมิคอนดักเตอร์ ฉบับที่ 183 ไกรวุฒิ โรจประเสริฐสุด)

การควบคุมสเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์ในการเกิดการหมุนจะต้องป้อนกระแสไฟฟ้าเข้าสู่ขดลวดตามลำดับที่เหมาะสม ซึ่งโดยทั่วไปมี 3 วิธี คือ แบบ 2 เฟส แบบเวฟ และแบบครึ่งสเต็ป

- แบบ 2 เฟส (Two phase sequence) การขับสเต็ปเปอร์มอเตอร์แบบนี้จะมีการป้อนกระแสไฟฟ้าเข้าสู่ขดลวดจำนวน 2 ชุด ในขณะเดียวกันทำให้วิธีนี้สามารถให้แรงบิด (torque) ได้สูง

ตารางที่ 2.3 การขับสเต็ปเปอร์มอเตอร์แบบ 2 เฟส
(ที่มา : เซมิคอนดักเตอร์ ฉบับที่ 183 ไกรวุฒิ โรจประเสริฐสุด)

สเต็ปที่	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

- **แบบเวฟ (Wave Sequence) การขับมอเตอร์แบบนี้จะมีการป้อนกระแสไฟเข้าสู่ขดลวดเพียง 1 ชุด ในเวลาใดเวลาหนึ่งเท่านั้น วิธีนี้จะทำให้การหมุนเป็นไปอย่างราบรื่นกว่าแบบ 2 เฟส แต่ให้แรงบิดต่ำกว่า**

ตารางที่ 2.4 การขับสเต็ปเปอร์มอเตอร์แบบเวฟ

(ที่มา : เซมิคอนดักเตอร์ ฉบับที่ 183 ไกรวุฒิ โรจน์ประเสริฐสุด)

สเต็ปที่	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

- แบบครึ่งสเต็ป (Half step Sequence) เป็นวิธีการขับมอเตอร์ที่ผสมระหว่างแบบปกติและแบบเวฟ โดยวิธีนี้จะทำให้มอเตอร์หมุนได้ครั้งละครึ่งสเต็ปจากค่ามุมต่อสเต็ปตามปกติ เช่น จากเดิมหมุนสเต็ปละ 3.6 องศา (100 step /รอบ) จะเหลือเพียงหมุนสเต็ปละ 1.8 องศา(200 step/รอบ) แม้ว่าการขับแบบนี้จะมีข้อข้อดีตรงที่สามารถเพิ่มความละเอียดในการหมุนได้ แต่แรงบิดที่ได้มีค่าไม่สูงมากนัก เนื่องจากในบางช่วงอาจมีการป้อนกระแสไฟเข้าสู่มอเตอร์เพียง 1 ขดเท่านั้น

ตารางที่ 2.5 การขับสเต็ปเปอร์มอเตอร์แบบครึ่งสเต็ป

(ที่มา : เซมิคอนดักเตอร์ ฉบับที่ 183 ไกรวุฒิ โรจน์ประเสริฐสุด)

สเต็ปที่	A	B	C	D
1	1	0	0	1
2	1	0	0	0
3	1	1	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	1	1
8	0	0	0	1

2.4 โปรแกรมโปรเทล (Protel)

โปรเทล (Protel) เป็นโปรแกรมสำหรับออกแบบทาง Schematic และ PCB Design โดยมี Advance PCB Design เป็นโปรแกรมที่ใช้สำหรับออกแบบแผ่นพีซีบี แบบ 32 บิต ด้วยโปรแกรมนี้เราสามารถสร้างไฟล์เอาท์พุตเป็นไฟล์รูเจาะได้ซึ่งเรียกว่า "NC Drill file" โดยสร้างจากเมนู File> Nc Drill และข้อมูลที่ได้มาจะมีลักษณะดังนี้

M 71
M 48
T 01-
T 02-
%
T 01
X 0001Y0005
Y 0009
.
.
.
T 02
.
.
.
M 30

ความหมายของข้อมูล

M71	หมายถึง	เป็นตัวที่ชี้ว่าเป็นโปรแกรม NC Drill ของ Protel
M48	หมายถึง	เริ่มต้นข้อมูล
T01 ถึง T02	หมายถึง	หมายเลขของดอกสว่าน (Tool size)
%	หมายถึง	เริ่มต้นของข้อมูลพิกัด X , Y โดยมี T01 เป็นชื่อของเครื่องมือ (tool)กำกับไว้ตอนต้นของข้อมูล
X 0001Y0005	หมายถึง	พิกัด X , Y
M30	หมายถึง	สิ้นสุดข้อมูล

หมายเหตุ

สำหรับพิกัด X , Y ในบรรทัดใดที่มีแต่ X หรือมีแต่ Y หมายความว่าที่ระบุในบรรทัดนี้เป็นค่าเดียวกับบรรทัดก่อนหน้า

2.5 โปรแกรมไมโครซิม(Microsim)

ไมโครซิม เป็นโปรแกรมที่ทำงานเกี่ยวกับการสร้างไฟล์สำหรับการออกแผ่นพีซีบี เช่นเดียวกับโปรแกรมโปรเทล ข้อมูลที่ได้จากโปรแกรมนี้มีลักษณะเหมือนกับโปรเทลทุกอย่างเพียงแต่จะต่างกันตรงที่ บรรทัดแรกของข้อมูลจะเป็น M72 ซึ่งจะเป็นตัวบ่งบอกว่าข้อมูลชุดนี้เป็นของโปรแกรมไมโครซิม