บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานที่เกี่ยวข้อง เพื่อให้มีความเข้าใจในการออกแบบและสร้างเครื่องลงทะเบียนโดยใช้บัตรประจำตัวแบบพกพาสำหรับบันทึกการเข้าร่วมกิจกรรมการประกันคุณภาพการศึกษาของมหาวิทยาลัย ประกอบด้วยไมโครคอนโทรลเลอร์ ไอซีสร้างฐานเวลา เอสแรม การใช้งานพอร์ตอนุกรม จอแสดงผลแบบผลึกของเหลว เครื่องอ่านแถบแม่เหล็กและแถบแม่เหล็ก และสุดท้ายเป็นโปรแกรมวิชวลเบสิก

2.1 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ได้จัดให้มีส่วนประกอบภายใน เพื่ออำนวยความสะดวกแก่ผู้ใช้ เช่น ไทเมอร์/เคาน์เตอร์ พอร์ตอนุกรม (Serial Port) และไมโครคอนโทรลเลอร์บางตัว อาจมีส่วนอื่นเพิ่มเติมเข้ามาอีก เช่น เบอร์ AT80C515, AT80C535 มีวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (Analog to Digital Converter)

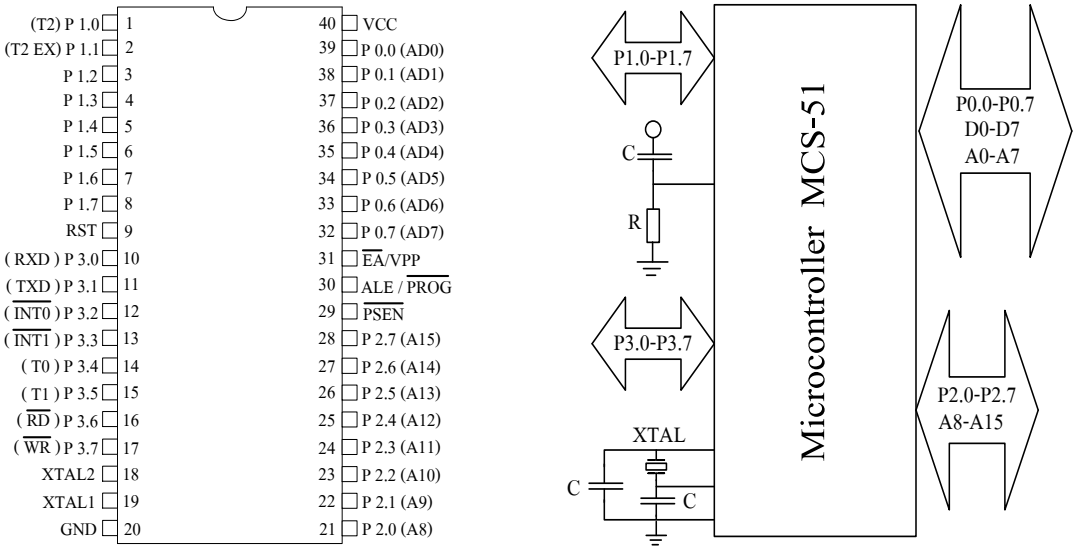
2.1.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์แต่ละเบอร์นั้นจะมีความแตกต่างกันในรายละเอียด ในที่นี้จะขออ้างถึงเบอร์ AT89S52 ของบริษัท Atmel ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต มีคุณสมบัติดังนี้

- มีหน่วยความจำแบบแฟลช (Flash Memory) ขนาด 8 กิโลไบท์
- โปรแกรมผ่านพอร์ตอนุกรมมาตรฐาน SPI (SPI Serial Interface)
- สามารถโปรแกรมและลบซ้ำได้นับ 1000 ครั้ง
- มีหน่วยความจำแบบ EEPROM ขนาด 2 กิโลไบต์
- สามารถโปรแกรมและลบซ้ำได้นับ 100,000 ครั้ง
- ใช้แหล่งจ่ายไฟกระแสตรงขนาด 5 โวลต์ (ทำงานในช่วง 4-6 โวลต์)
- ทำงานด้วยสัญญาณนาฬิกาตั้งแต่ 0-24 MHz
- สามารถป้องกันหน่วยความจำได้ 3 ระดับ
- มีหน่วยความจำข้อมูล (RAM) ขนาด 256 ไบต์
- มี 32 บิตอิสระสามารถเข้าถึงระดับบิตได้
- มีไทเมอร์/ เคาน์เตอร์ขนาด 16 บิต ทั้งหมด 2 ตัวและรองรับอินเตอร์รัปต์ได้ 8 แหล่ง
- สามารถสื่อสารข้อมูลแบบอนุกรมได้ด้วย UART Port
- มีโหมดการทำงานแบบ Low Power Idle และ Power Down สำหรับการประหยัดพลังงาน
- มี Watchdog Timer เพื่อให้การทำงานของระบบสามารถ Reset ได้อัตโนมัติ

2.1.2 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 (AT89S52)

การจัดเรียงขาของ MCS-51 (AT89S52) เป็นไปตามรูปที่ 2.1



รูปที่ 2.1 การจัดขาของ MCS-51(AT89S52)

- **VCC** ต่อไฟเลี้ยง (Supply Voltage 5 V)
- **GND** ต่อกราวดน์ (Ground)
- **Port0 (P0.0-P0.7)** เป็นพอร์ต 2 ทิศทางขนาด 8 บิตสามารถทำงานได้ 2 หน้าที่คือเป็นพอร์ตอินพุตเอาต์พุตทั่วไป และใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอกคือรับส่งข้อมูลและกำหนดแอดเดรสไบต์ต่ำ (A0-A7)
- **Port1 (P1.0-P1.7)** เป็นพอร์ต 2 ทิศทางขนาด 8 บิต มีการต่อความต้านทาน (Pull-Up Resistant) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตเอาต์พุตทั่วไป นอกจากนี้ยังใช้งานเป็นขาอินพุตเอาต์พุตของไทเมอร์ 2
- **Port2 (P2.0-P2.7)** เป็นพอร์ต 2 ทิศทางขนาด 8 บิต มีการต่อความต้านทาน (Pull-Up Resistance) ไว้ภายใน สามารถทำงานได้ 2 หน้าที่คือเป็นพอร์ตอินพุตเอาต์พุตทั่วไป และใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอกคือรับส่งข้อมูล และกำหนดแอดเดรสไบต์สูง (A8-A15)
- **Port3 (P3.0-P3.7)** เป็นพอร์ต 2 ทิศทางขนาด 8 บิตและ มีการต่อความต้านทาน (Pull-Up Resistance) ไว้ภายใน ทำหน้าที่คือเป็นพอร์ตอินพุตเอาต์พุตทั่วไป ยังใช้งานเป็นสัญญาณควบคุมการติดต่อกับหน่วยความจำ การอินเตอร์รัปต์และอื่นๆ

- **RST** เป็นขาอินพุตที่ใช้รับสัญญาณสำหรับรีเซ็ตซีพียู ซีพียูจะถูกรีเซ็ตเมื่อขานี้เป็นลอจิก “1” นาน 2 แมชินไซเคิล หรือ 24 ไซเคิลของสัญญาณนาฬิกา
- **ALE/PROG** ทำหน้าที่เป็นขาเอาต์พุตเมื่อซีพียูต้องการติดต่อกับหน่วยความจำภายนอกจะทำการส่งสัญญาณพัลส์ออกมาที่ขานี้เพื่อทำการแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก และขานี้จะเป็นอินพุตเมื่ออยู่ในระหว่างโปรแกรมแฟลช
- **PSEN** เป็นขาเอาต์พุตใช้ในการติดต่อกับหน่วยความจำโปรแกรมภายนอกคือเมื่อซีพียูทำการประมวลผลกับหน่วยความจำภายนอกขานี้จะแอคทีฟสองครั้ง
- **EA/VPP** เป็นขาเอาต์พุตและต้องการลอจิก “0” เพื่อยอมให้ซีพียูสามารถเข้าถึงหน่วยความจำภายนอกได้ซึ่งอยู่ที่ตำแหน่ง 0000H ถึง FFFFH นอกจากนี้แล้วขานี้ยังใช้รับไฟ 12 โวลต์เพื่อใช้ในระหว่างที่ทำการโปรแกรมแฟลช
- **XTAL1** เป็นขาอินพุตของวงจรออสซิลเลเตอร์แอมปลิไฟเออร์ และยังเป็นอินพุตของวงจรกำเนิดสัญญาณนาฬิกาภายใน
- **XTAL2** เป็นขาเอาต์พุตของวงจรออสซิลเลเตอร์แอมปลิไฟเออร์

2.1.3 สัญลักษณ์และคำสั่งของ MCS-51 เป็นไปตามตารางที่ 2.1

ตารางที่ 2.1 สัญลักษณ์และคำสั่งของ MCS-51

สัญลักษณ์	ความหมาย
Rn	หมายถึง รีจิสเตอร์ใช้งานทั่วไป R0-R7 ที่ถูกเลือกใช้ในขณะนั้น
@Ri	หมายถึง รีจิสเตอร์ใช้งานทั่วไป R0 หรือ R1 ที่เข้าถึงข้อมูลได้โดยอ้อม
direct	หมายถึง ข้อมูลขนาด 8 บิต ที่ใช้กำหนดค่าตำแหน่งหน่วยความจำสำหรับเก็บข้อมูลภายใน ประกอบด้วยหน่วยความจำเก็บข้อมูลทั่วไป ตำแหน่ง 0 - 127 และหน่วยความจำ เก็บข้อมูลที่ใช้เป็นรีจิสเตอร์ใช้งานเฉพาะ ตำแหน่ง 128 - 255
bit	หมายถึง ค่าตำแหน่งของบิตข้อมูลในหน่วยความจำที่เข้าถึงในระดับบิต
#data	หมายถึง ข้อมูลขนาด 8 บิตที่กำหนดในคำสั่ง
#data16	หมายถึง ข้อมูลขนาด 16 บิตที่กำหนดในคำสั่ง
rel	หมายถึง ค่าตำแหน่งหน่วยความจำขนาด 8 บิต ที่คิดแบบมีเครื่องหมาย ในคำสั่ง SJMP และการกระโดดแบบมีเงื่อนไขทุกคำสั่ง
addr11	หมายถึง ค่าตำแหน่งหน่วยความจำขนาด 11 บิต ใช้เป็นค่าตำแหน่งหน่วยความจำปลายทาง (ภายในขอบเขต 2048 ตำแหน่ง) ในคำสั่ง AJMP และ ACALL

ตารางที่ 2.1 สัญลักษณ์และคำสั่งของ MCS-51 (ต่อ)

สัญลักษณ์	ความหมาย
addr16	หมายถึง ค่าตำแหน่งหน่วยความจำขนาด 16 บิต ใช้เป็นค่าตำแหน่งหน่วยความจำปลายทาง (ภายในขอบเขต 65536 ตำแหน่ง) ในคำสั่ง LJMP และ LCALL
คำสั่ง	ความหมาย
MOV A,Rn	ย้ายข้อมูลจาก Rn ไป A
MOV A,direct	ย้ายข้อมูลจากหน่วยความจำ Direct ไป A
MOV A,@Ri	ย้ายข้อมูลจากหน่วยความจำที่เก็บอยู่ในตำแหน่ง Ri ไป A
MOV A,#data	ย้ายค่าคงที่ 8 บิตไปเก็บที่ A
MOV Rn,A	ย้ายข้อมูลจาก A ไป Rn
MOV Rn,direct	ย้ายข้อมูลจากหน่วยความจำ direct ไป Rn
MOV direct,A	ย้ายข้อมูลจาก A ไปยังหน่วยความจำ Direct
MOV direct,Rn	ย้ายข้อมูลจาก Rn ไปยังหน่วยความจำ Direct
MOV direct,direct	ย้ายข้อมูลระหว่างหน่วยความจำภายใน
MOV direct,@Ri	ย้ายข้อมูลจากหน่วยความจำที่เก็บอยู่ในตำแหน่ง Ri ไปยังหน่วยความจำ Direct
MOV direct,#data	ย้ายค่าคงที่ 8 บิต ไปยังหน่วยความจำ Direct
MOV @Ri,A	ย้ายข้อมูลใน A ไปยังหน่วยความจำ Ri
MOV @Ri,direct	ย้ายข้อมูลจากหน่วยความจำ direct ไปยังหน่วยความจำ Ri
MOV @Ri,#data	ย้ายค่าคงที่ 8 บิต ไปยังหน่วยความจำ Ri
MOV DPTR,#data16	ย้ายค่าคงที่ 16 บิต ไปยัง DPTR
MOVC A,@A+DPTR	ย้ายข้อมูลจากหน่วยความจำข้อมูลที่สัมพัทธ์กับ DPTR ไปยัง A
MOVC A,@A+PC	ย้ายข้อมูลจากหน่วยความจำข้อมูลที่สัมพัทธ์กับ PC ไปยัง A
MOVX A,@Ri	ย้ายข้อมูลจากหน่วยอินพุตที่เก็บอยู่ในตำแหน่ง Ri ไปยัง A
MOVX A,@DPTR	ย้ายข้อมูลจากหน่วยความจำที่เก็บอยู่ในตำแหน่ง DPTR ไปยัง A
MOVX @Ri,A	ย้ายข้อมูลที่เก็บอยู่ใน A ไปยังหน่วยเอาต์พุตตำแหน่ง Ri
MOVX @DPTR,A	ย้ายข้อมูลที่เก็บอยู่ใน A ไปยังหน่วยความจำตำแหน่ง DPTR
MOV C,bit	ย้ายค่า bit ไปยัง Carry Flag

ตารางที่ 2.1 สัญลักษณ์และคำสั่งของ MCS-51 (ต่อ)

คำสั่ง	ความหมาย
MOV bit,C	ย้ายค่าแฟล็ก Carry ไปยัง Bit
JNC rel	กระโดด ถ้าค่าแฟล็ก Carry เป็น 0
JC rel	กระโดด ถ้าค่าแฟล็ก Carry เป็น 1
JB bit,rel	กระโดด ถ้าค่า Bit เป็น 1
JNB bit,rel	กระโดด ถ้าค่า Bit เป็น 0
JBC bit,rel	กระโดด ถ้าค่า Bit เป็น 1 และเปลี่ยนค่า Bit เป็น 0
JZ rel	กระโดดไปยังตำแหน่งที่สัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าหากค่า A เป็นค่า 00H
JNZ rel	กระโดดไปยังตำแหน่งที่สัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าหากค่า A ไม่เป็นค่า 00H
CJNE A,direct,rel	เปรียบเทียบค่า A กับหน่วยความจำ Direct และกระโดดไปยังตำแหน่งที่สัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าค่าไม่เท่ากัน
CJNE A,#data,rel	เปรียบเทียบค่า A กับค่าคงที่ และกระโดดไปยังตำแหน่งที่สัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าค่าไม่เท่ากัน
CJNE Rn,#data,rel	เปรียบเทียบค่า Rn กับค่าคงที่ และกระโดดไปยังตำแหน่งที่สัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าค่าไม่เท่ากัน
CJNE @Ri,#data,rel	เปรียบเทียบค่าใน Ri กับค่าคงที่ และกระโดดไปยังตำแหน่งที่สัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าค่าไม่เท่ากัน
DJNZ Rn,rel	ลดค่าใน Rn และกระโดดไปยังตำแหน่งที่สัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าค่าไม่เป็น 0
DJNZ direct,rel	ลดค่าในหน่วยความจำ Direct และกระโดดไปยังตำแหน่งสัมพัทธ์กับตำแหน่งปัจจุบัน ถ้าค่าไม่เป็น 0
SJMP rel	กระโดดไปยังตำแหน่งสัมพัทธ์กับตำแหน่งปัจจุบัน
AJMP addr11	กระโดดไปยังตำแหน่งจากค่าแอดเดรส 11 บิต
LJMP addr16	กระโดดไปยังตำแหน่งจากค่าแอดเดรส 16 บิต

ตารางที่ 2.1 สัญลักษณ์และคำสั่งของ MCS-51 (ต่อ)

คำสั่ง	ความหมาย
JMP @A+DPTR	กระโดดไปยังตำแหน่งที่สัมพัทธ์กับ A+DPTR
ACALL addr11	ไปทำโปรแกรมย่อยจากค่าแอดเดรส 11 บิต
LCALL addr16	ไปทำโปรแกรมย่อยจากค่าแอดเดรส 16 บิต
RET	สิ้นสุดการทำโปรแกรมย่อย แล้วกลับไปยังตำแหน่งถัดไปที่กระโดดมา
RETI	สิ้นสุดการทำโปรแกรมย่อย อินเตอร์รัปท์
INC A	เพิ่มค่าใน A
INC Rn	เพิ่มค่าใน Rn
INC direct	เพิ่มค่าในหน่วยความจำ Direct
INC @Ri	เพิ่มค่าในหน่วยความจำที่ตำแหน่ง Ri
INC DPTR	เพิ่มค่าใน DPTR
DEC A	ลดค่าใน A
DEC Rn	ลดค่าใน Rn
DEC direct	ลดค่าในหน่วยความจำ Direct
DEC @Ri	ลดค่าในหน่วยความจำที่เก็บอยู่ใน Ri
DEC DPTR	ลดค่าใน DPTR
MUL AB	คูณ A กับ B แล้วเก็บค่าใน BA
DIV AB	หาร A กับ B แล้วเก็บค่าใน A เก็บเศษใน B
DA A	ทำ Decimal Adjust ค่าใน A
ADD A,Rn	บวกค่า Rn กับ A
ADD A,direct	บวกค่าในหน่วยความจำDirect กับ A เก็บผลลัพธ์ใน A
ADD A,@Ri	บวกค่าในหน่วยความจำตำแหน่ง Ri กับ A เก็บผลลัพธ์ใน A
ADD A,#data	บวกค่าคงที่ 8 บิต กับ A เก็บผลลัพธ์ใน A
ADDC A,Rn	บวกค่า Rn กับ A พร้อมแฟล็ก Carry เก็บผลลัพธ์ใน A
ADDC A,direct	บวกค่าในหน่วยความจำ Direct กับ A พร้อมแฟล็ก Carry เก็บผลลัพธ์ใน A
ADDC A,@Ri	บวกค่าในหน่วยความจำตำแหน่ง Ri กับ A พร้อมแฟล็ก Carry เก็บผลลัพธ์ใน A

ตารางที่ 2.1 สัญลักษณ์และคำสั่งของ MCS-51 (ต่อ)

คำสั่ง	ความหมาย
ADDC A,#data	บวกค่าคงที่ 8 บิต กับ A พร้อมแฟล็ก Carry เก็บผลลัพธ์ใน A
SUBB A,Rn	ลบค่า Rn กับ A พร้อมแฟล็ก Borrow เก็บผลลัพธ์ใน A
SUBB A,direct	ลบค่าในหน่วยความจำ Direct กับ A พร้อมแฟล็ก Borrow เก็บผลลัพธ์ใน A
SUBB A,@Ri	ลบค่าในหน่วยความจำที่เก็บอยู่ใน Ri กับ A พร้อมแฟล็ก Borrow เก็บผลลัพธ์ใน A
SUBB A,#data	ลบค่าคงที่ 8 บิต กับ A พร้อมแฟล็ก Borrow เก็บผลลัพธ์ใน A
CLR A	ทำค่าใน A ให้เป็นศูนย์
CPL A	กลับค่าบิตใน A เป็นตรงข้ามทุกบิต
RL A	หมุนบิตใน A ไปทางซ้าย 1 บิตและบิต 0 เป็นค่าจากบิต 7
RLC A	หมุนบิตใน A ไปทางซ้าย 1 บิตและค่าจากบิต 7 นำไปเก็บในแฟล็ก Carry และบิตที่อยู่ในแฟล็ก Carry ไปเป็นบิตที่ 0
RR A	หมุนบิตใน A ไปทางขวา 1 บิตและบิต 7 เป็นค่าจากบิต 0
RRC A	หมุนบิตใน A ไปทางขวา 1 บิตและค่าจากบิต 0 นำไปเก็บในแฟล็ก Carry และบิตที่อยู่ในแฟล็ก Carry ไปเป็นบิตที่ 7
SWAP A	สลับค่าสี่บิตซ้ายกับสี่บิตขวาภายใน A
PUSH direct	ย้ายข้อมูลหน่วยความจำ Direct ไปเก็บยัง Stack
POP direct	ย้ายข้อมูลจาก Stack ไปยังหน่วยความจำ Direct
CLR C	ทำค่าแฟล็ก Carry ให้เป็น 0
CLR bit	ทำค่า Bit ให้เป็น 0
SETB C	ทำค่าแฟล็ก Carry ให้เป็น 1
SETB bit	ทำค่า Bit ให้เป็น 1
CPL C	กลับค่าแฟล็ก Carry ให้เป็นตรงข้าม
CPL bit	กลับค่า Bit ให้เป็นตรงข้าม
ORL C,bit	OR ค่า Bit กับแฟล็ก Carry เก็บผลลัพธ์ใน C
ORL C,/bit	OR ค่า Not Bit กับแฟล็ก Carry เก็บผลลัพธ์ใน C
ANL C,bit	AND ค่า Bit กับแฟล็ก Carry เก็บผลลัพธ์ใน C
ANL C,/bit	AND ค่า Not Bit กับแฟล็ก Carry เก็บผลลัพธ์ใน C

2.2 ไอซีสร้างฐานเวลาจริง

การเชื่อมต่อไอซีสร้างฐานเวลาจริง (RTC : Real Time Clock) ซึ่งมีหน้าที่สร้างฐานเวลาจริงให้ไมโครคอนโทรลเลอร์โดยใช้ IC DS1307 ต่อเชื่อมกับไมโครคอนโทรลเลอร์เพียง 2 บิต คือ ขา SDA และ SCL โดย IC DS1307 จะให้ข้อมูลเกี่ยวกับวันเวลาทั้งหมด ได้แก่ วินาที,นาที,ชั่วโมง วันในสัปดาห์,วันที่,เดือนและปีค.ศ. IC DS1307 จำเป็นต้องต่อแบตเตอรี่ไว้ตลอดเวลาไม่ว่าจะใช้งานหรือไม่ เพื่อเป็นไฟเลี้ยงวงจรภายใน IC DS1307 ให้ทำงานต่อไป

IC DS1307 สามารถกำเนิดสัญญาณพัลส์ที่ขา 7 (SQW/Out) ส่งออกมาตลอดเวลา ในกรณีที่มีการอีนาเบิลวงจรกำหนดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่สัญญาณนี้สามารถเลือกได้ 4 ค่า คือ 1 Hz,4.096 kHz,8.192 kHz,32 kHz พร้อมกันนั้นจะมีการเก็บค่าของเวลาไว้ในหน่วยความจำแรมภายใน ซึ่งมีขนาดรวม 64 ไบต์ แต่จัดให้เก็บข้อมูลเวลาและรีจีสเตอร์ควบคุมจำนวน 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์ ถ้าใช้ V_{BAT} เท่ากับ 3V แล้วไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} IC DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันทีจะไม่มีการกำหนดสัญญาณพัลส์ออกมาแต่วงจรสร้างฐานเวลายังคงทำงาน เพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง IC DS1307 จึงมีค่าของเวลาที่เป็นจริง วงจรสื่อสารอนุกรมภายใน IC DS1307 เป็นช่องทางการสื่อสารระหว่าง IC DS1307 กับอุปกรณ์มาสเตอร์ในระบบ I^2C Bus ผู้ใช้สามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำที่ใช้งานทั่วไปได้ โดยรายละเอียดการเก็บค่าเวลาและรีจีสเตอร์ควบคุมในหน่วยความจำแรมภายใน IC DS1307 เป็นไปตามตารางที่ 2.2

ตารางที่ 2.2 การเก็บค่าเวลาและรีจีสเตอร์ควบคุมในหน่วยความจำภายใน IC DS1307

<table>
<tr><th>ค่าของข้อมูล</th><th>ADDR</th><th>BIT 7</th><th>BIT 6</th><th>BIT 5</th><th>BIT 4</th><th>BIT 3</th><th>BIT 2</th><th>BIT 1</th><th>BIT 0</th></tr>
<tr><td>00 – 59</td><td>00H</td><td>CH</td><td colspan="3">ข้อมูลวินาที (หลักสิบ)</td><td colspan="4">ข้อมูลวินาที (หลักหน่วย)</td></tr>
<tr><td>00 – 59</td><td>01H</td><td>X</td><td colspan="3">ข้อมูลนาที (หลักสิบ)</td><td colspan="4">ข้อมูลนาที (หลักหน่วย)</td></tr>
<tr><td rowspan="2">01 – 12
00 - 23</td><td rowspan="2">02H</td><td rowspan="2">X</td><td>12 ชม.</td><td>AM / PM</td><td rowspan="2">ข้อมูล
ชม.
(หลักสิบ)</td><td colspan="4" rowspan="2">ข้อมูลชั่วโมง (หลักหน่วย)</td></tr>
<tr><td>24 ชม
(“0”).</td><td>ข้อมูล
ชม.
(หลักสิบ)</td></tr>
<tr><td>1 – 7</td><td>03H</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td colspan="3">ข้อมูลวันในสัปดาห์</td></tr>
</table>

ตารางที่ 2.2 การเก็บค่าเวลาและรีจีสเตอร์ควบคุมในหน่วยความจำภายใน IC DS1307 (ต่อ)

ค่าของข้อมูล	ADDR	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
01 – 31	04H	X	X	ข้อมูลวันที่ (หลักสิบ)		ข้อมูลวันที่ (หลักหน่วย)			
01 – 12	05H	X	X	X	ข้อมูลเดือน (หลักสิบ)	ข้อมูลเดือน (หลักหน่วย)			
00 - 99	06H	ข้อมูลปี (หลักสิบ)				ข้อมูลปี (หลักหน่วย)			
	07H	OUT	X	X	SQWE	X	X	RS1	RS0

Note: ถ้า Bit 4 (SQWE) เป็น “1” ส่งผลให้มีสัญญาณพัลส์ออกที่ขา 7 (SQW/Out) โดยความถี่ขึ้นอยู่กับค่าของ RS1 และ RS0 ซึ่งเป็นไปตามตารางที่ 2.3

ถ้า Bit 4 (SQWE) เป็น “0” ส่งผลให้ที่ขา 7 (SQW/Out) มีลอจิกตาม Bit 7 (Out)

ตารางที่ 2.3 ค่า RS1, RS0 และความถี่

RS1	RS0	f (SQW/Out)
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

2.3 เอสแรม (SRAM)

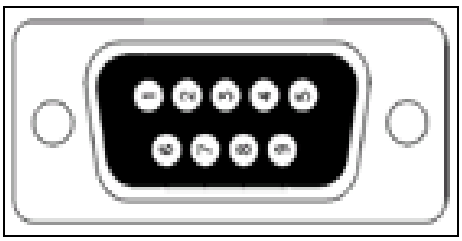
SRAM หรือ Static RAM คือ แรม (RAM) ซึ่งเก็บรักษาข้อมูลบิตไว้ในหน่วยความจำของมันตราบเท่าที่ยังมีกระแสไฟฟ้าหล่อเลี้ยงอยู่ ไม่เหมือนกับดีแรม (DRAM) ที่เก็บข้อมูลไว้ในเซลซึ่งประกอบขึ้นด้วยตัวเก็บประจุ (Capacitor) และทรานซิสเตอร์ (Transistor) SRAM ไม่ต้องการการรีเฟรช ทั้งยังมีความเร็วในการเข้าถึงข้อมูลที่สูงกว่า SRAM มักถูกทำเป็นหน่วยความจำแคชในเครื่องคอมพิวเตอร์ หรือเป็นส่วนหนึ่งของแรมของตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก (Digital to Analog Converter) บนวีดีโอการ์ด รวมไปถึงการใช้เป็นหน่วยความจำในส่วนการขยายความจำของไมโครคอนโทรลเลอร์ ถ้า SRAM มีแบตเตอรี่เลี้ยงวงจรตลอดเวลาเรียกว่า Nonvolatile SRAM

2.4 การใช้งานพอร์ตอนุกรม (RS-232)

การสื่อสารแบบอนุกรม นับว่ามีความสำคัญต่อการใช้งานไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้แป้นพิมพ์ และจอภาพของ PC เป็น อินพุต และ เอาต์พุต ในการติดต่อ หรือควบคุมไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อยเพียง 3 เส้นเท่านั้น คือ สายส่งสัญญาณ TX สายรับสัญญาณ RX และสายกราวด์ GND โดยปกติพอร์ตอนุกรม RS-232 จะสามารถต่อสายได้ยาว 50 ฟุตโดยประมาณ ขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทาง และ ปริมาณสัญญาณรบกวน ลักษณะของคอนเน็คเตอร์ของพอร์ตอนุกรมเป็นไปตามรูปที่ 2.2 และ 2.3 การจัดขาเป็นไปตามรูปที่ 2.4 ส่วนความหมายของขาต่างๆ เป็นไปตามตารางที่ 2.4 และการเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 เป็นไปตามรูปที่ 2.5 และ 2.6

รูปที่ 2.2 พอร์ตอนุกรมของ PC -DB9 ตัวผู้

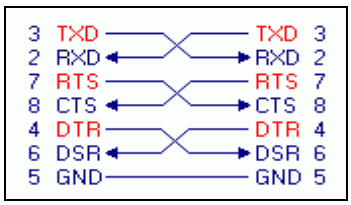
รูปที่ 2.3 พอร์ตอนุกรมอุปกรณ์ภายนอก DB9 ตัวเมีย



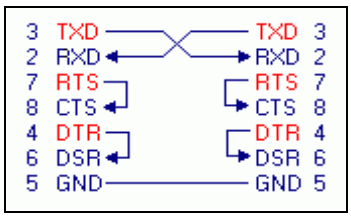
รูปที่ 2.4 การจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9

ตารางที่ 2.4 ความหมายของขาต่างๆ ของ DB9

Pin	Description	Type
1	Data Carrier Detect (DCD)	Input
2	Received Data (RXD)	Input
3	Transmitted Data (TXD)	Output
4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request To Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input



รูปที่ 2.5 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null Modem



รูปที่ 2.6 การต่ออุปกรณ์ภายนอกผ่าน DB9 Null Modem แบบ 3 เส้น

2.4.1 การทำงานของขาสัญญาณ DB9

TXD เป็นขาที่ใช้ส่งข้อมูล

RXD เป็นขาที่ใช้รับข้อมูล

DTR แสดงสภาวะพอร์ตฝ่าย DTE พร้อมทำงาน

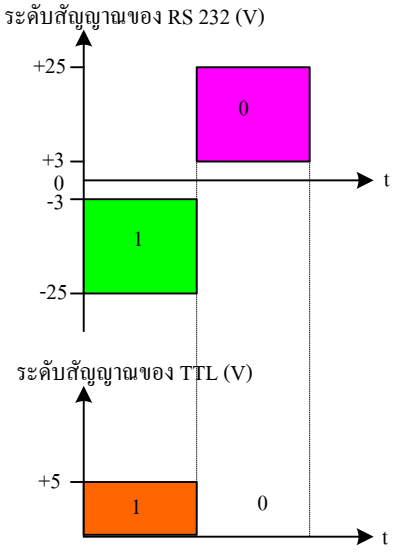
DSR แสดงสภาวะพอร์ตฝ่าย DCE พร้อมทำงาน

เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วยในขณะเดียวกันจะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่

RTS แสดงสภาวะร้องขอการส่งข้อมูลของ DTE

CTS แสดงสภาวะพร้อมรับข้อมลของ DCE และ **GND** ขา Ground

ระดับสัญญาณของ RS232 เป็นไปตามรูปที่ 2.7



รูปที่ 2.7 ระดับสัญญาณของ RS232 และระดับสัญญาณของ TTL

สัญญาณรบกวนที่เกิดขึ้นในสายนำสัญญาณมักจะมีแรงดันเป็นบวก เมื่อเทียบกับกราวดน์ เพื่อป้องกันสัญญาณรบกวนนี้ จึงออกแบบแรงดันของโลจิก "1" เป็นลบ คืออยู่ในช่วง -3V ถึง -25V ส่วนแรงดันของโลจิก "0" อยู่ในช่วง +3V ถึง +25V และเหตุที่ระดับสัญญาณ ของ RS232 อยู่ในช่วง +25V ถึง -25V ก็เพื่อให้ต่อสายสัญญาณไปได้ไกลขึ้น ดังนั้นจึงจำเป็นจะต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL

2.4.2 อัตราการรับส่งข้อมูล (Bit Rate)

- คือความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิตต่อวินาทีเช่น 300, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000 เป็นต้น

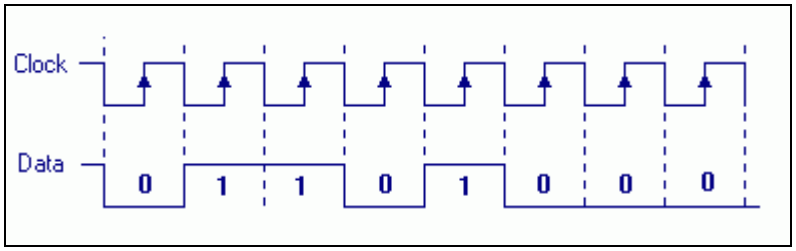
- การเลือกอัตราการรับส่งข้อมูลขึ้นอยู่กับ ชนิดของสายสัญญาณ ระยะทาง และปริมาณสัญญาณรบกวน

2.4.3 รูปแบบการสื่อสารแบบอนุกรม

มีด้วยกันอยู่ 2 แบบ คือแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous)

2.4.3.1 การสื่อสารแบบซิงโครนัส (Synchronous)

-การรับส่งข้อมูลจะมีสัญญาณนาฬิกาซึ่งเป็นตัวกำหนดจังหวะเวลาการส่งข้อมูลรวมอยู่ด้วยอีกเส้นหนึ่ง ใช้คู่กับสัญญาณข้อมูล ตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ดเป็นไปตามรูปที่ 2.8

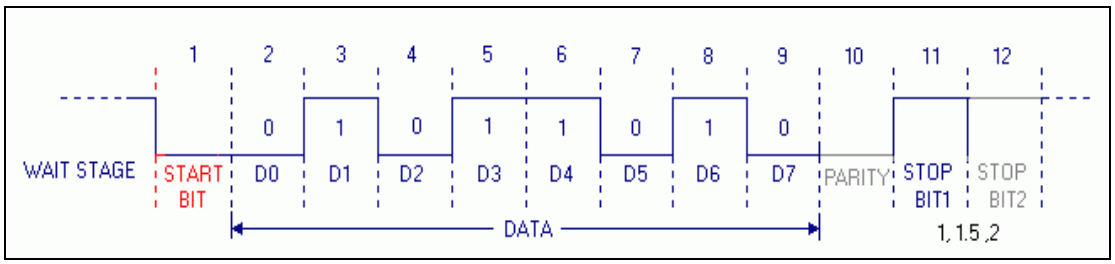


รูปที่ 2.8 การส่งสัญญาณจากคีย์บอร์ด

2.4.3.2 การสื่อสารแบบอะซิงโครนัส (Asynchronous)

- การรับส่งข้อมูลโดยที่ไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วย แต่จะใช้ให้ตัวส่งและตัวรับ มีอัตราการรับส่งข้อมูลที่เท่ากัน รูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start Bit) มีขนาด1 บิต
2. บิตข้อมูล (Data) มีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) มีขนาด 1 บิตหรือไม่มี
4. บิตหยุด (Stop Bit) มีขนาด 1, 1.5, 2 บิต



รูปที่ 2.9 การรับส่งข้อมูลแบบไม่มีสัญญาณนาฬิกา

- เมื่อไม่มีการส่งข้อมูล ขา Data จะมีสถานะเป็นโลจิก “1” หรือ สถานะหยุดรอ (Waiting Stage)
- เมื่อเริ่มต้นส่งข้อมูลจะให้ขา Data เป็นโลจิก “0” เป็นจำนวน 1 บิต เรียกว่าบิตเริ่มต้น (Start bit)
- จากนั้นจะเริ่มต้นส่งข้อมูล โดยส่งบิตต่ำไปก่อน (LSB)
- แล้วตามด้วยพาริตี้บิต (จะมีหรือไม่ก็ได้ ขึ้นอยู่กับการติดตั้งค่าของทั้งสองฝ่าย)
- สุดท้ายตามด้วยโลจิก “1” อย่างน้อย 1 บิต (มีขนาด 1, 1.5 หรือ 2 บิต) เพื่อแสดงว่าสิ้นสุดข้อมูล

การรับและส่งข้อมูลแบบอนุกรมยังแบ่งออกเป็นลักษณะการใช้งานได้ 3 แบบคือ

1. แบบซิมเพลกซ์ (Simplex) เป็นการรับหรือส่งข้อมูลแบบทิศทางเดียวเท่านั้น
2. แบบฮาล์ฟดูเพลกซ์ (Half Duplex) เป็นการรับหรือส่งข้อมูลแบบสลับกัน

คือเมื่อด้านหนึ่งส่งอีกด้านหนึ่งเป็นฝ่ายรับสลับกัน ไม่สามารถรับ-ส่งข้อมูลในเวลาเดียวกันได้

3. แบบฟูลดูเพลกซ์ (Full Duplex) สามารถรับ-ส่งข้อมูลในเวลาเดียวกันได้

2.5 จอแสดงผลแบบผลึกของเหลว

จอแสดงผลแบบผลึกของเหลว (Liquid Crystal Display: LCD) เป็นอุปกรณ์แสดงผลที่นิยมใช้ในปัจจุบันเนื่องจากมีความเหมาะสมหลายๆ ด้าน เช่น การใช้กระแสไฟฟ้าต่ำ และสามารถแสดงผลเป็นตัวอักษรและตัวเลขหรือแสดงเป็น กราฟฟิก (เฉพาะรุ่น) ได้ จอแสดงผลแบบผลึกของเหลว ซึ่งต่อไปนี้จะขอเรียกว่า “จอแสดงผล LCD” ซึ่งมีขาสัญญาณ 14 ขาหรือ 16 ขา (เพิ่ม Back Light) มีลักษณะเป็นไปตามรูปที่ 2.10 โดยแต่ละขาจะมีหน้าที่การทำงานดังนี้

รูปที่ 2.10 จอแสดงผล LCD

1. ขา 1 (Vss) เป็นขากราวด์ของจอแสดงผล LCD

2. ขา 2 (Vcc) เป็นขาแรงดันไฟฟ้าขนาด +5 โวลต์

3. ขา 3 (Vee) เป็นขารับแรงดันไฟฟ้าเพื่อปรับความเข้มตัวอักษร โดยต่อกับกราวดน์และมีความเข้มสูงสุด และถ้าต่อกับ Vcc จะมีความเข้มต่ำสุด

4. ขา 4 (RS) เป็นขาเลือกที่มีรีจิสเตอร์ภายในซึ่งมีอยู่ 2 ตัวคือ รีจิสเตอร์คำสั่ง (Instruction Register: IR) และรีจิสเตอร์ข้อมูล (Data Register: DR) ขณะเดียวกันถ้าเป็นลอจิก “1” จะเป็นการเลือกรีจีสเตอร์ข้อมูล DR และถ้าเป็นลอจิก “0” จะเป็นการเลือกรีจีสเตอร์ IR

5. ขา 5 (R/W) เป็นตัวเลือกว่าจะเขียนหรือจะอ่านข้อมูล ถ้าขานี้ เป็นลอจิก “1” จะเป็นการอ่านข้อมูลจากจอแสดงผล LCD แต่ถ้าเป็นลอจิก “0” จะเขียนข้อมูลไปที่จอแสดงผล LCD

6. ขา 6 (E) เป็นขากำหนดสภาพการรับเขียนอ่านข้อมูล (Enable) ซึ่งเป็นสัญญาณ Clock

7. ขา 7 – ขา 14 (DB0-DB7) เป็นขาที่ใช้ในการรับส่งรหัสข้อมูลหรือรหัสคำสั่ง

8. ขา 15 และขา 16 เป็นขาไฟเลี้ยงวงจร Back Light โดยขา 15 เป็นแรงดัน +5 Vและขา 16 เป็น Ground

2.5.1 พื้นฐานในการใช้งานจอแสดงผล LCD

การเขียนข้อมูลให้กับจอแสดงผล LCD แบ่งเป็น 2 ลักษณะคือการเขียนคำสั่งและการเขียนข้อมูลโดยกำหนดด้วยขา RS คือถ้าขา RS เป็นลอจิก “0” หมายถึงการเขียนคำสั่งควบคุมการทำงานของจอแสดงผล LCD และถ้าขา RS เป็นลอจิก “1” หมายถึงการเขียนข้อมูลไปที่จอแสดงผล LCD

หลักการในการเขียนข้อมูลให้กับจอแสดงผล LCD นี้คือเมื่อมีการเขียนข้อมูลส่งไปยังจอแสดงผล LCD จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่งตามค่าเวลาปฏิบัติการ (Execute Time) โดยระบบของไมโครคอนโทรลเลอร์จะสามารถตรวจสอบได้จากแฟลก BF และถ้าเรียบร้อยแล้วจึงจะสามารถเขียนข้อมูลชุดต่อไปได้ กรณีเป็นการต่อแบบพอร์ตอินพุตเอาต์พุต คือ ไม่สามารถอ่านข้อมูลกลับได้ โดยระบบไมโครคอนโทรลเลอร์ก็จะต้องใช้วิธีการหน่วงเวลาแทน ในการเขียนข้อมูลให้กับจอแสดงผล LCD นั้น สามารถที่จะทำได้ทั้ง 8 บิตและ 4 บิต โดยในกรณี 4 บิตใช้สายสัญญาณข้อมูลเพียง 4 เส้นคือขา DB4-DB7 ใช้สำหรับระบบไมโครคอนโทรลเลอร์แบบ 4 บิต หรือเพื่อการประหยัดสาย การเขียนข้อมูลจะเหมือนกับ 8 บิต แต่ให้เขียน 2 ครั้ง คือขา DB4-DB7 ก่อนแล้วตามด้วยขา DB0-DB3 ต้องกำหนดตามค่า DL ในคำสั่ง ฟังก์ชั่นเซ็ต

หน่วยความจำภายในจอแสดงผล LCD เรียกว่า DDRAM (Display Data RAM) คือหน่วยความจำภายในของจอแสดงผล LCD ที่เป็นบัฟเฟอร์ของข้อมูลโดยถ้าเขียนรหัสแอสกี (ASCII) ใดๆ ลงไปในหน่วยความจำนี้ โดยจะปรากฏเป็นตัวอักษรที่หน้าจอทันที ส่วน CGRAM (Character Generator RAM) คือ หน่วยความจำแรมภายในของจอแสดงผล LCD สำหรับเก็บภาพ

ตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง 8 ตัว โดยสามารถอ้างตำแหน่งได้ทั้งหมด 64 ไบต์ คือ 8 ไบต์ต่อหนึ่งตัวอักษร

2.5.2 การกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของจอแสดงผล LCD เป็นไปตามตารางที่ 2.5 และรายละเอียดของแต่ละคำสั่งเป็นไปตามตารางที่ 2.6-2.16

ตารางที่ 2.5 การกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของจอแสดงผล LCD

Instruction	RS	R/W	Data Bit								EXE.Time(μs)
			7	6	5	4	3	2	1	0	
Clear Display	0	0	0	0	0	0	0	0	0	1	1640
Cursor at Home	0	0	0	0	0	0	0	0	1	X	1640
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	40
Display On/Off	0	0	0	0	0	0	1	D	C	B	40
Display Shift	0	0	0	0	0	1	S/C	R/L	X	X	40
Function Set	0	0	0	0	1	DL	N	F	X	X	40
Set CGRAM Address	0	0	0	1	CGRAM Address						40
Set DDRAM Address	0	0	1	DDRAM Address							40
Busy Address Read	0	1	BF	Address							0
CGRAM, DDRAM WR	1	0	Write Data								40
CGRAM, DDRAM RD	1	1	Read Data								40

ตารางที่ 2.6 การลบหน้าจอ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับการลบหน้าจอนั้น ทำการเขียนตัวอักษรว่างลงไปในหน่วยความจำ ภายในจอแสดงผล LCD ทั้งหมดและกำหนดค่าที่อยู่ของหน่วยความจำภายในจอแสดงผล LCD เป็น 0 พร้อมทั้งเคอร์เซอร์จะกลับไปที่ตำแหน่งซ้ายบนสุดของจอภาพ

ตารางที่ 2.7 Cursor at Home

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	X

สำหรับกำหนดค่าที่อยู่ (Address) ของหน่วยความจำภายในจอแสดงผล LCD เป็น 0 พร้อมทั้งตัวเคอร์เซอร์จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลในหน่วยความจำภายในจอแสดงผล LCD ไม่มีการเปลี่ยนแปลง

ตารางที่ 2.8 Entry Mode Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

I/D = 0 กำหนดทิศทางของเคอร์เซอร์ของหน่วยความจำภายในจอแสดงผล LCD ให้เป็นแบบลดตำแหน่งลง

I/D = 1 กำหนดทิศทางของเคอร์เซอร์ของหน่วยความจำภายในจอแสดงผล LCD ให้เป็นแบบเพิ่มตำแหน่งขึ้น

S = 0 เมื่อเขียนข้อมูลแล้วตัวเคอร์เซอร์ จะถูกเลื่อนไปตามทิศทางของค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้วตัวเคอร์เซอร์ จะอยู่กับที่และตัวอักษรจะถูกดันตามทิศทางของทิศทางของเคอร์เซอร์ ในการกำหนดลักษณะของการแสดงผลนี้ให้กำหนดก่อนการเขียนข้อมูลลงในหน่วยความจำภายในจอแสดงผล LCD และเมื่อกำหนดแล้วจะต้องไม่ใช้คำสั่งลบหน้าจออีก

ตารางที่ 2.9 Display On/Off

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D = 0 กำหนดให้ปิดหน้าจอ

D = 1 กำหนดให้เปิดหน้าจอ

C = 0 กำหนดให้ปิดเคอร์เซอร์

C = 1 กำหนดให้เปิดเคอร์เซอร์ โดยเคอร์เซอร์จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่งเคอร์เซอร์

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่งเคอร์เซอร์ กระพริบเป็นรูปสี่เหลี่ยม

ตารางที่ 2.10 Display Shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	X	X

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

S/C = 0 กำหนดให้เลื่อนเคอร์เซอร์ตามทิศทางซ้ายหรือขวาไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความซึ่งจะแสดงตามทิศทางซ้ายหรือขวาไป 1 หลักทุกบรรทัด

ตารางที่ 2.11 Function Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	X	X

DL = 0 กำหนดให้การติดต่อกับจอแสดงผล LCD เป็นแบบ 4 บิต ซึ่งการกำหนดค่าทิศทางของเคอร์เซอร์สามารถกกระทำได้ที่ขา DB4-DB7

DL = 1 กำหนดให้การติดต่อกับจอแสดงผล LCD เป็นแบบ 8 บิต โดยการกำหนดค่าทิศทางของเคอร์เซอร์สามารถกระทำได้ที่ขา DB4-DB7 และถ้ามีการกำหนดให้เป็นแบบ 4 บิต ตั้งแต่ครั้งแรกหลังจากการจ่ายกระแสไฟฟ้าจะทำให้จอแสดงผล LCD มีการรับข้อมูลแบบ 4 บิต ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1 บรรทัด

N = 1 กำหนดจำนวนบรรทัดแบบ 2 บรรทัดขึ้นไป

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5*7 จุดหรือ 5*8 จุด

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5*10 จุด

ตารางที่ 2.12 Set CGRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM Address					

สำหรับการกำหนดตำแหน่งของแรมเก็บตัวอักษร (CGRAM) เพื่อใช้ในการกำหนดรูปแบบการแสดงผลที่กำหนดขึ้นเองไม่เกิน 8 รูปแบบ

ตารางที่ 2.13 Set DDRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM Address						

สำหรับการกำหนดตำแหน่งของหน่วยความจำภายในจอแสดงผล LCD ในการอ่าน และเขียนข้อมูลที่ต่อจากนี้จะเป็นไปตามตำแหน่งที่กำหนดทันที ซึ่งตำแหน่งของหน่วยความจำภายในจอแสดงผล LCD ในแต่ละรุ่น จะมีความแตกต่างกันเนื่องจากจำนวนตัวอักษรต่อบรรทัดไม่เท่ากัน

ตารางที่ 2.14 Busy Flag and Address Read

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	Address						

สำหรับการอ่านค่าแฟลก BF (Busy Flag) ซึ่งจะบอกถึงความพร้อมของจอแสดงผล LCD ในการรับข้อมูล ถ้าแฟลก BF เป็นลอจิก “0” หมายความว่า พร้อมที่จะรับข้อมูลชุดต่อไปได้แต่ถ้าแฟลก BF เป็นลอจิก “1” หมายความว่ายังไม่พร้อม นอกจากนี้ยังเป็นการอ่านค่าตำแหน่งหน่วยความจำภายในจอแสดงผล LCD ด้วย

ตารางที่ 2.15 Write Data to DDRAM or CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	Data							

สำหรับการเขียนข้อมูลลงหน่วยความจำภายในจอแสดงผล LCD หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ตำแหน่งจะถูกเพิ่มหรือลดลงโดยอัตโนมัติตามที่กำหนดจากทิศทางของเคอร์เซอร์ ในคำสั่ง Entry Mode Set และการเขียนข้อมูลลงหน่วยความจำภายในจอแสดงผล LCD หรือหน่วยความจำแรม ขึ้นกับว่าก่อนหน้าคำสั่งนี้กำหนดตำแหน่งที่ใด

ตารางที่ 2.16 Read Data from DDRAM LCD or CGRAM

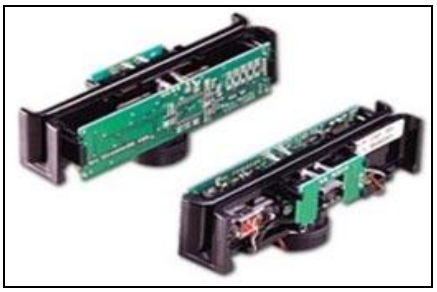
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	Data							

สำหรับการอ่านข้อมูลจากหน่วยความจำภายในของจอแสดงผล LCD หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ตำแหน่งจะถูกเพิ่มหรือลดลงโดยอัตโนมัติตามที่กำหนดจากค่าทิศทางของเคอร์เซอร์ ในคำสั่ง Entry Mode Set และการอ่านหน่วยความจำภายในจอแสดงผล LCD หรือ CGRAM ขึ้นกับว่าก่อนหน้าคำสั่งนี้มีการกำหนดตำแหน่งที่ใด

2.6 เครื่องอ่านบัตรแม่เหล็ก

2.6.1 พื้นฐานของเครื่องอ่านบัตรแม่เหล็ก

เครื่องอ่านบัตรแม่เหล็ก (Magnetic Card Reader) มีลักษณะการทำงานคล้ายการทำงานของ Audio Cassettes คือ ในการทำงานของ Audio Cassettes Tape นั้น เมื่อนำม้วนเทปใส่ในเครื่องเล่นเทปทั่วไป เริ่มทำการอ่านเทป เมื่อแถบแม่เหล็กของม้วนเทปเริ่มเคลื่อนที่ผ่านหัวอ่านเทป ซึ่งทำมาจากโลหะชนิดพิเศษที่พันด้วยขดลวดเล็กๆ ภายใน กระแสไฟฟ้าจำนวนน้อย จะไหลผ่านเส้นลวดนี้ไปยังเทป กระแสไฟฟ้านี้จะไหลไปยัง Tape Bias สัญญาณการเหนี่ยวนำนี้จะทำให้เกิดสนามแม่เหล็กเล็กๆ ในช่องว่างหรือ ช่องเปิดในหัวอ่านเทป ซึ่ง Audio Tape เหล่านี้ จะให้สัญญาณแม่เหล็กเล็กๆ เป็นล้านๆ สัญญาณออกมา ซึ่งส่วนประกอบของหัวอ่านแถบแม่เหล็กและลักษณะของเครื่อง Magnetic Card Reader เป็นไปตามรูปที่ 2.11 และ 2.12 ตามลำดับ



รูปที่ 2.11 ส่วนประกอบของหัวอ่านแถบแม่เหล็ก

รูปที่ 2.12 เครื่อง Magnetic Card Reader

โดยเครื่องอ่านแถบแม่เหล็กจะอ่านได้เฉพาะข้อมูลที่บันทึกด้วยความละเอียด 75 BPI คือ เฉพาะในส่วนของแถบแม่เหล็กแบบที่ 2 (Track 2) เท่านั้น โดยมีสัญญาณที่ใช้เชื่อมต่อเพื่ออ่านข้อมูลจากเครื่องอ่านแถบแม่เหล็กทั้งหมด 3 สัญญาณ โดยเป็นสัญญาณเอาต์พุตส่งออกมาจากเครื่องอ่านแถบแม่เหล็กทั้งหมด ได้แก่สัญญาณ Data, Clock และ Present โดยความสัมพันธ์ของสัญญาณทั้ง 3 เป็นไปตามรูปที่ 2.13 และมีขาไฟเลี้ยงวงจรที่ต้องต่อให้เครื่องอ่านอีก ดังนี้คือ

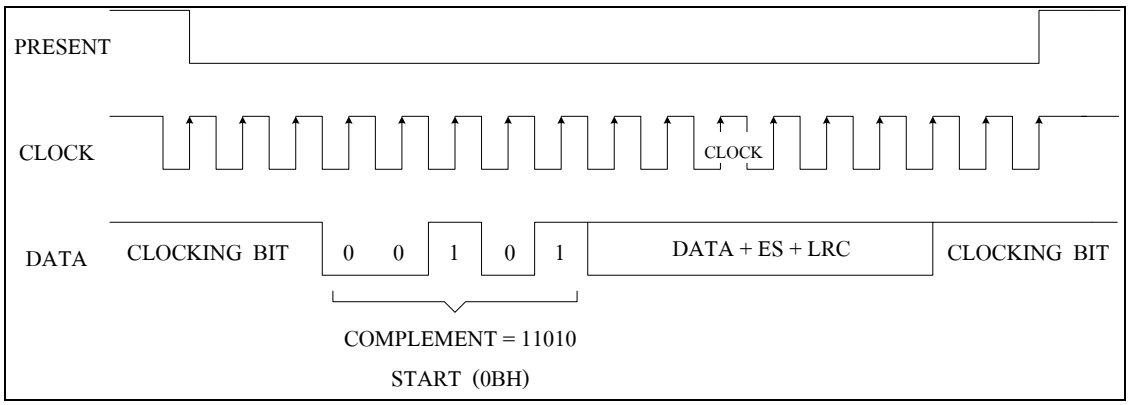
- ไฟเลี้ยงวงจร (+Vcc) มีค่าเป็น +5 VDC ซึ่งต่อจากภายนอกให้เครื่องอ่านแถบแม่เหล็ก

- Ground มีค่าเป็น 0 VDC ซึ่งต่อจากภายนอกให้เครื่องอ่านแถบแม่เหล็ก

- สัญญาณข้อมูล (Data) เป็นสัญญาณข้อมูลเอาต์พุตส่งออกมาจากเครื่องอ่านแถบแม่เหล็ก ซึ่งเป็นสัญญาณของข้อมูลที่อ่านได้จากบัตรแถบแม่เหล็ก โดยในการอ่านสัญญาณข้อมูลนี้ต้องอ่านแบบอนุกรมทีละบิต โดยต้องพิจารณาให้สัมพันธ์สอดคล้องกับสัญญาณนาฬิกา (Clock) ของบัตรด้วย ซึ่งสัญญาณข้อมูลที่อ่านได้นี้จะกลับสภาวะกับสัญญาณจริง คือ มีสภาวะเป็นตรงข้ามกันอยู่ ดังนั้น เมื่ออ่านสัญญาณจากเครื่องอ่านแถบแม่เหล็กได้แล้วจะต้องกลับสภาวะของสัญญาณ (Complement) เสียก่อนจึงจะได้ข้อมูลที่ถูกต้อง

- สัญญาณนาฬิกา (Clock) เป็นสัญญาณเอาต์พุตส่งออกมาจากเครื่องอ่านแถบแม่เหล็ก ใช้เป็นสัญญาณอ้างอิงในการอ่านสัญญาณข้อมูลจากบัตร โดยการอ่านสัญญาณข้อมูลแต่ละบิตนั้น ต้องอ่านในขณะที่สัญญาณนาฬิกานี้เป็นขอบขาขึ้น (Rising Edge) เสมอ ซึ่งสัญญาณข้อมูลที่อ่านได้นั้นจะเริ่มต้นจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อนเป็นอันดับแรก

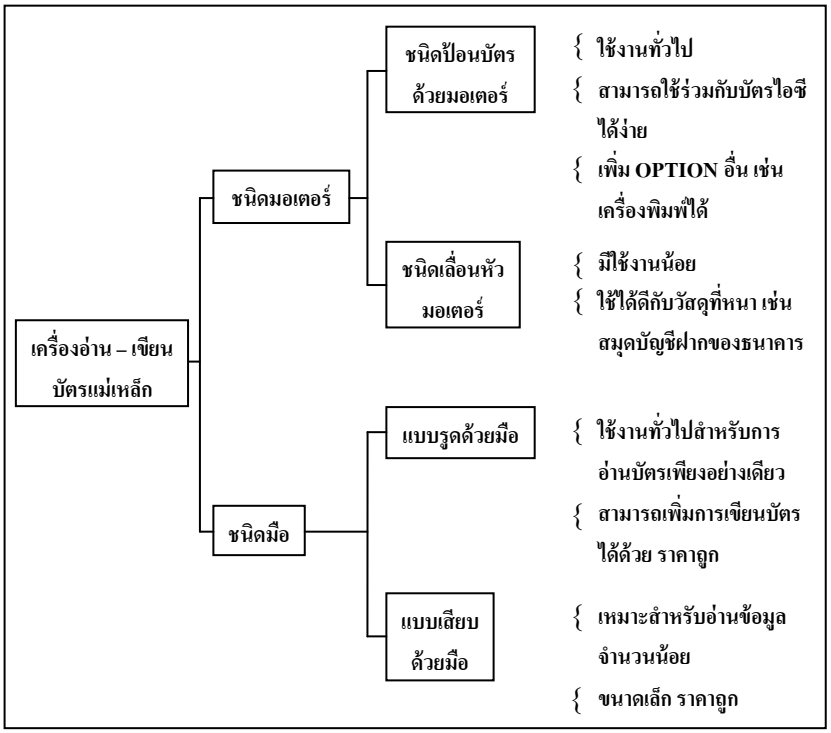
- สัญญาณ Present เป็นสัญญาณสถานะเอาต์พุตส่งออกมาจากเครื่องอ่านแถบแม่เหล็ก เมื่อสัญญาณนี้เป็นลอจิก “0” แสดงว่าเครื่องอ่านแถบแม่เหล็กเริ่มต้นส่งข้อมูลออกมา หรือมีการนำบัตรแถบแม่เหล็กไปรูดผ่านหัวอ่านสัญญาณของเครื่อง เมื่อสัญญาณนี้หมดลงจะกลับเป็นโลจิก “1” อีกครั้งหนึ่ง แสดงว่าข้อมูลจากการอ่านในครั้งนั้นถูกส่งออกไปหมดสิ้นเรียบร้อยแล้ว ซึ่งเราอาจใช้ประโยชน์จากสัญญาณนี้ ส่งเป็นสัญญาณ Interrupt ให้ CPU เพื่อทำการอ่านข้อมูล



รูปที่ 2.13 ความสัมพันธ์ของสัญญาณ Data, Clock และ Present ของเครื่องอ่านแถบแม่เหล็ก

2.6.2 ชนิดของเครื่องอ่านบัตรแม่เหล็ก

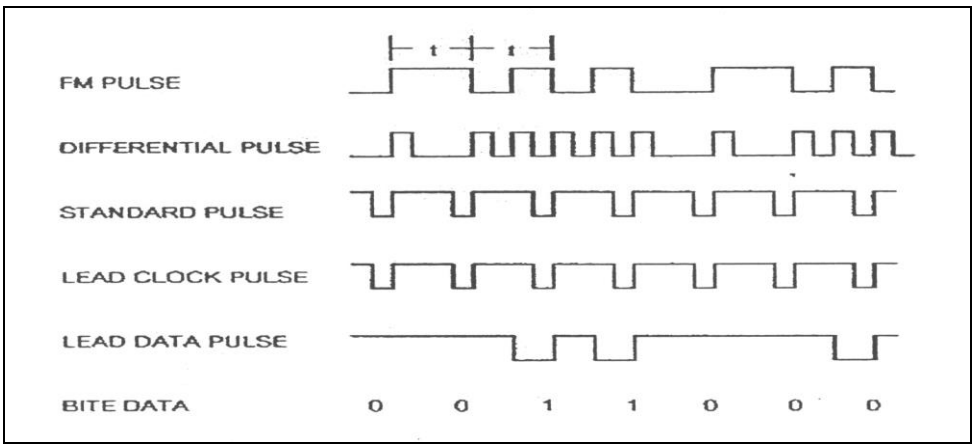
แผนผังดังในรูปที่ 2.14 แสดงชนิดของเครื่องอ่าน–เขียนบัตรแม่เหล็กโดยแบ่งตามวิธีการป้อนบัตรเข้าสู่ตัวเครื่อง นอกจากนี้การอินเตอร์เฟสเข้ากับอุปกรณ์อื่น (TTL, RS232 หรืออื่น ๆ) และลักษณะการใช้งาน ประกอบเข้ากับเครื่องอ่าน หรือวางตั้งโต๊ะตัวเดียว จะทำให้โครงสร้างหรือวงจรของเครื่องอ่านเขียนบัตรแม่เหล็กแตกต่างกันบ้าง เครื่อง ATM ทั่วไปจะใช้เครื่องอ่านและเขียนบัตรแม่เหล็กแบบใช้มอเตอร์เป็นตัวป้อนบัตร ส่วนเครื่องอ่านบัตรประจำตัว (ID) ของผู้ใช้เครื่องคอมพิวเตอร์และอื่นๆ จะใช้แบบรูดด้วยมือ ส่วนเครื่องอ่านบัตรแบบเสียบบัตรจะพบในเครื่องถ่ายเอกสารแบบหยอดเหรียญ หรือบัตรเสียบที่ใช้เป็นกุญแจห้องของโรงแรม เป็นต้น



รูปที่ 2.14 ชนิดของเครื่องอ่านและเขียนบัตรแม่เหล็ก

2.6.3 การถอดรหัสสัญญาณข้อมูลออกมาจากเครื่องอ่านบัตรแม่เหล็ก

การแยกสัญญาณข้อมูลออกจากสัญญาณ Clock จะใช้วงจร Decoder โดยมี Time Chart ของ FM Demodulation ดังรูปที่ 2.15 การถอดเฉพาะสัญญาณข้อมูลออกมาจากสัญญาณพัลส์ที่ได้นี้จะต้องตรวจดูว่ามีการกลับขั้วของพัลส์ FM ในช่วง Standard Time ซึ่งมีค่าประมาณ 75% ของความยาว 1 Bit (75% ของเวลา) หรือไม่ ถ้าไม่มีค่าของข้อมูลจะเป็น “0” และถ้ามีค่าของข้อมูลจะเป็น “1”



รูปที่ 2.15 Time chart ของ FM demodulation

การสร้าง Standard Time เพื่อแยกข้อมูลออกมานั้นมี 2 วิธี คือความยาวคงที่และแบบความยาวเปลี่ยนแปลงได้ แบบความยาวคงที่จะกำหนด Standard Time ให้มีค่าคงที่และเปรียบเทียบกับทุกบิต ส่วนแบบที่ความยาวเปลี่ยนแปลงได้จะวัดความยาวของบิตก่อนหน้า และสร้าง Standard Time จากจุดนั้น แล้วนำไปเปรียบเทียบกับบิตถัดไป สำหรับการสร้าง Standard Time แบบเปลี่ยนแปลงความยาวได้นั้นขณะที่กำลังอ่านบัตรอยู่ ถ้าความเร็วในการป้อนบัตรเปลี่ยนแปลง Standard Time จะเปลี่ยนแปลงตามไปด้วยจึงใช้วิธีการนี้กับเครื่องอ่านบัตรที่ป้อนบัตรด้วยมือ เนื่องจากความเร็วในการรูดด้วยมือนั้นค่าไม่แน่นอน การแยกข้อมูลโดยการตรวจหา Peak ตามแบบ FM นั้นยอมให้ระดับเอาต์พุตที่อ่านได้เปลี่ยนแปลงมาก (เรียกว่า Jitter) จะเป็นสาเหตุให้การอ่านข้อมูลผิดพลาดได้ Jitter ทั้งการอ่านและบันทึกข้อมูลมีสาเหตุหลายประการ ในการออกแบบใช้งานจะต้องให้ความสำคัญกับปัญหานี้อย่างมาก

2.6.4 การอ่านข้อมูลจากบัตรแถบแม่เหล็ก

การใช้เครื่องอ่านแถบแม่เหล็กเพื่อให้อ่านข้อมูลจากบัตรแถบแม่เหล็กนั้น ต้องใส่บัตรเข้าไปในช่องรับบัตรจากด้านหน้าตามทิศทางและตำแหน่งที่กำหนดไว้เท่านั้น โดยให้สังเกตตำแหน่งของหัวอ่านกับส่วนที่เป็นแถบแม่เหล็กของบัตรต้องอยู่ในแนวเดียวกัน สำหรับการอ่านข้อมูลจากบัตรนั้นขอแนะนำให้อ่านในขณะที่สอดบัตรเข้าไปในช่องรับบัตรเท่านั้น เพราะถ้าไปอ่านในขณะที่ดึงบัตรกลับออกมาจะทำให้ได้ข้อมูลที่ย้อนถอยหลังกลับ ซึ่งจะทำให้ยากแก่การถอดรหัสที่ได้จากการอ่าน และเนื่องจากเครื่องอ่านแถบแม่เหล็กสามารถอ่านข้อมูลจากบัตรแถบแม่เหล็กได้เฉพาะข้อมูลแบบที่ 2 (Track 2) เท่านั้น ดังนั้นจึงขอกล่าวถึงเฉพาะวิธีการอ่านข้อมูลแบบที่ 2 เพียงอย่างเดียว โดย Format ของข้อมูลในแถบแม่เหล็กแบบที่ 2 (Track 2) ซึ่งใช้บันทึกสัญญาณด้วยความ

หนาแน่นของสนามแม่เหล็ก 75 BPI ซึ่งจะบรรจุจำนวนรหัสของข้อมูลต่างๆใน Track นี้ ได้สูงสุดไม่เกิน 40 ตัวอักษร โดยนับรวมรหัสควบคุมและรหัสตรวจสอบด้วย ซึ่งมีรายละเอียดตามตารางที่ 2.17 และ 2.18

ตารางที่ 2.17 ข้อมูลในแถบแม่เหล็กแบบที่ 2

SS	PAN	FS	Addition Data	ES	LRC

SS = รหัสข้อมูลแสดงการเริ่มต้นของข้อมูลในบัตร (Start Sentinel) มีรหัสข้อมูลเป็น 0BH(;)

PAN = ข้อมูลแสดงหมายเลขของบัตร (มีจำนวนข้อมูลในส่วนนี้สูงสุดไม่เกิน 19 หลัก)

FS = รหัสข้อมูลแสดงการแบ่งแยกข้อมูล (Field Separator) มีรหัสข้อมูลเป็น 0DH (=)

Additional Data = เป็นข้อมูลเพิ่มเติมอื่นๆ ของบัตร เช่น เดือน/ปี ที่ออกบัตรและหมดอายุ

ES = รหัสข้อมูลแสดงการสิ้นสุดของข้อมูลในบัตร (End Sentinel) มีรหัสข้อมูลเป็น 0FH (?)

LRC = ข้อมูลตรวจสอบความผิดพลาด (Longitudinal Redundancy Check)

ตารางที่ 2.18 รายละเอียดเพิ่มเติมของข้อมูลในแถบแม่เหล็กแบบที่ 2

0000000000 SS Data, Data, Data,…, Data Es LRC 0000000000

จากตารางที่ 2.18 ซึ่งใช้แสดงลักษณะการจัดเรียงของข้อมูลที่เก็บไว้ในแถบแม่เหล็กในส่วนของ Track 2 ซึ่งจัดเรียงลำดับความสำคัญจากซ้ายไปขวาแบบอนุกรมทีละบิต จะสังเกตเห็นว่ามีข้อมูลซึ่งเป็นลอจิกศูนย์ หรือที่เรียกว่า “Clocking Bit” นำหน้าและปิดท้ายข้อมูลจริงอยู่ ซึ่งลักษณะการจัดเก็บข้อมูลของ Track 2 จะเก็บข้อมูลด้วยการเข้ารหัสแบบ “Module5” ซึ่งในแต่ละชุดข้อมูลจะประกอบด้วยข้อมูล 5 บิต โดย 4 บิตแรก (D0…D3) เป็นรหัสข้อมูลแบบ BCD ส่วนบิตที่ 5 (D4) เป็นพาริตี้บิตแบบคี่ (Odd) เพื่อใช้ตรวจสอบความถูกต้องของข้อมูลในแต่ละไบท์ (5-Bits) ที่อ่านได้ ซึ่งหากค่าของบิตพาริตี้ผิดพลาด แสดงว่าการอ่านข้อมูลนั้นล้มเหลว โดยเริ่มทำการอ่านข้อมูลเมื่อสัญญาณ Present เริ่มเป็นศูนย์ก่อน และทำการอ่านข้อมูลในทุกๆ ขอบขาขึ้นของสัญญาณนาฬิกาเสมอ ซึ่งข้อมูลในส่วนก่อนเริ่มต้นและหลังจากสิ้นสุดของการอ่านนี้ จะมีค่าเป็นศูนย์ (สัญญาณ Data = Logic “1”) นำหน้าและปิดท้ายข้อมูลจริงอยู่ ซึ่งเรียกว่า “Clocking Bit” ซึ่งข้อมูลในส่วนนี้เราไม่ต้องสนใจ แต่ให้ตรวจสอบและรอจนกว่าจะเริ่มเป็นข้อมูล Start บิต (0BH หรือ

11010 ในที่นี้เรียงลำดับความสำคัญจากซ้ายไปขวา ซึ่งบิตเริ่มต้นของรหัส Start หรือ 0BH ต้องเริ่มด้วย 1 เป็นบิตแรกเสมอ) ดังนั้นในการอ่านเราต้องรอจนกว่าจะพบสัญญาณข้อมูลมีค่าเป็น “1” (สัญญาณ Data = Logic “0” เพราะกลับสภาวะกันอยู่) จึงเริ่มเก็บข้อมูลชุดละ 5 บิตไปเรื่อยๆ จนถึงรหัสจบ (0FH) ซึ่งเมื่อพบรหัสจบแล้วจะมีข้อมูลตามมาอีก 1 ไบท์ ซึ่งเป็นข้อมูลสำหรับตรวจสามารถหาได้จากการนำเอาข้อมูลในแต่ละไบท์ (ไม่คิดพาริตี้บิต) ตั้งแต่เริ่มต้นจนสิ้นสุดมาทำการ XOR กับไบท์ถัดไปเรื่อยๆ ตามลำดับ ซึ่งหากผลลัพธ์สุดท้ายที่ได้ไม่เท่ากับค่าของ “LRC” ที่อ่านมาได้ แสดงว่าการอ่านข้อมูลทั้งหมดล้มเหลว

สำหรับข้อมูลของรหัส BCD ที่ใช้สำหรับเครื่องอ่านแถบแม่เหล็กนี้จะเป็นข้อมูลชุดละ 5 บิต โดยเป็นข้อมูลจริง 4 บิต และเป็นรหัสตรวจสอบพาริตี้อีก 1 บิต ซึ่งมีจำนวนทั้งหมด 16 อักขระ ตามตารางที่ 2.19

ตารางที่ 2.19 ข้อมูลของรหัส BCD สำหรับเครื่องอ่านแถบแม่เหล็ก

Parity	D3	D2	D1	D0	Character	Function
1	0	0	0	0	0(0H)	Data
0	0	0	0	1	1(1H)	Data
0	0	0	1	0	2(2H)	Data
1	0	0	1	1	3(3H)	Data
0	0	1	0	0	4(4H)	Data
1	0	1	0	1	5(5H)	Data
1	0	1	1	0	6(6H)	Data
0	0	1	1	1	7(7H)	Data
0	1	0	0	0	8(8H)	Data
1	1	0	0	1	9(9H)	Data
1	1	0	1	0	:(AH)	Control
0	1	0	1	1	;(BH)	Start Sentinel
1	1	1	0	0	<(CH)	Control
0	1	1	0	1	=(DH)	Field Seprator
0	1	1	1	0	>(EH)	Control
1	1	1	1	1	?(FH)	End Sentinel

2.7 โครงสร้างของบัตรแม่เหล็ก

บัตรแม่เหล็กที่มีใช้กันอยู่ในปัจจุบันมีอยู่ด้วยกันมากมายหลายชนิดดังตารางที่ 2.20 และ 2.21

ตารางที่ 2.20 ชนิดและการใช้งานบัตรแม่เหล็ก

ชนิด	เนื้อวัสดุ	ความหนาของบัตร	การใช้งาน
บัตรพลาสติก (PLASTIS CARD)	-PVC ชนิดแข็ง -PVCA ชนิดแข็ง	0.76 mm	- บัตร ATM - บัตรเครดิต - บัตรประจำตัว (ID)
PET CARD	-POLYETHYLENE THEREPHTHALATE	0.2 – 0.4 mm	- บัตรโทรศัพท์ - บัตรช็อปปิ้งต่างๆ
บัตรกระดาษ (COMPOSITE PAPER CARD)	-กระดาษอย่างดี -กระดาษเคลือบพลาสติก -กระดาษบันทึกข้อมูล	0.2 – 0.76 mm	-ตั๋วรถไฟ -ตั๋วทางด่วน -ตั๋วเครื่องบิน -บัตรโรงแรม

ตารางที่ 2.21 ตัวอย่างการใช้งานบัตรแม่เหล็ก

การใช้งาน	ชนิดของบัตร	หมายเหตุ(อุปกรณ์และระบบ)
การเงิน การธนาคาร	บัตร ATM บัตรเครดิต บัตรหลักทรัพย์ บัตรช้อปปิ้ง	เครื่อง ATM /CD/ระบบธนาคารอัตโนมัติ ระบบ CAT/POS/ระบบ POS ของปั๊มน้ำมัน การจ่ายเงินซื้อหลักทรัพย์ CASHING/CD การซื้อสินค้าในห้างสรรพสินค้า/ร้านค้า
การคมนาคม	ตั๋วรถไฟแม่เหล็ก ตั๋วผ่านทางด่วน ตั๋วเครื่องบิน ตั๋วจอดรถ	เครื่องขายตั๋วรถไฟอัตโนมัติ ระบบเก็บเงินค่าผ่านทางด่วนอัตโนมัติ ระบบขายตั๋วเครื่องอัตโนมัติ ระบบควบคุมที่จอดรถ
การสื่อสาร	บัตรโทรศัพท์	โทรศัพท์สาธารณะแบบใช้บัตร

ตารางที่ 2.21 ตัวอย่างการใช้งานบัตรแม่เหล็ก (ต่อ)

การใช้งาน	ชนิดของบัตร	หมายเหตุ(อุปกรณ์และระบบ)
การสำนักงาน(OA.FA)	บัตรประจำตัวพนักงาน บัตรประจำตัวนักเรียน บัตรเครื่องถ่ายเอกสาร บัตรตรวจโรค บัตรโรงแรม	ระบบเช็คเวลาเข้า-ออกงาน ระบบยืม-คืนหนังสือห้องสมุด ระบบควบคุมเครื่องถ่ายเอสาร ระบบควบคุมโรค ระบบเช็คอิน-เอาต์ของโรงแรม
การรักษาความปลอดภัย	บัตรปิด–เปิดล็อก	ระบบควบคุมการเข้า-ออก

2.7.1 คุณสมบัติที่ดีของบัตรแม่เหล็ก

2.7.1.1 บัตรแม่เหล็กสามารถแก้ไขข้อมูลได้

2.7.1.2 ความเชื่อถือได้ของข้อมูลที่บันทึกไว้มีค่าสูง

2.7.1.3 เครื่องเขียน–อ่านบัตรแม่เหล็กมีมากกว่าบัตรชนิดอื่น ๆ

2.7.1.4 ข้อมูลที่บันทึกจะต้องมองด้วยตาเปล่าไม่เห็น จึงรักษาความลับได้ดี

2.7.1.5 บัตรแม่เหล็กสามารถพกพาและใช้งานได้สะดวก

2.7.1.6 ราคาบัตรแม่เหล็กไม่แพง

2.8 โปรแกรมวิชวลเบสิก

โปรแกรมวิชวลเบสิก (Visual Basic) เป็นโปรแกรมของบริษัท ไมโครซอฟต์ จำกัด มีความสามารถในการสร้างโปรแกรม .exe โดยสามารถกำหนดรูปแบบของรหัสเทียม (P-Code) หรือรหัสแท้ (Native Code) ได้โดยส่วนใหญ่แล้วโปรแกรมวิชวลเบสิกจะใช้เป็นโปรแกรมที่ผู้ใช้จะนำไปใช้เป็นฐานข้อมูล เพราะมีการเชื่อมต่อที่สวยงามและใช้ง่าย สามารถตอบสนองต่อเหตุการณ์ (Event) โดยสามารถที่จะเลือกเหตุการณ์ใดๆ ให้มีการกระตุ้น (Activate) ตามสถานการณ์ที่เหมาะสมได้เช่นกัน ซึ่งช่วยให้มีความสมบูรณ์และถูกต้องตามแนวคิดของการโปรแกรมเชิงวัตถุหรือ Object Oriented Programming (OOP) มากยิ่งขึ้นและสามารถนำไปใช้กับระบบปฏิบัติการวินโดวส์ 95 หรือวินโดวส์ 98 ได้ โดยการนำความสามารถในด้านการควบคุมโปรแกรมแบบการควบคุมที่สามารถเพิ่มเติมจากโปรแกรมการควบคุมที่มีอยู่เดิม (Active X) และสามารถควบคุมการสื่อสารและความสามารถพื้นฐานต่างๆ ซึ่งจะใช้ความสามารถในจุดนี้ไปประยุกต์ใช้โดยอาศัยการควบคุมแบบการควบคุมพอร์ตอนุกรม (MS Comm) ซึ่งอยู่ในการควบคุมการสื่อสาร เพื่อใช้ส่ง

สัญญาณ ไปยังพอร์ตอนุกรม (Serial Port) เพื่อทำการควบคุมทางด้านฮาร์ดแวร์ที่นำมาต่อที่พอร์ตอนุกรม ส่วนการควบคุมพื้นฐานต่าง ๆ นั้นจะใช้ในการออกแบบลักษณะของโปรแกรมที่ออกแบบโดยใช้การควบคุมต่างๆ ที่อยู่ในตารางเครื่องมือของโปรแกรมวิชวลเบสิก

โปรแกรมวิชวลเบสิกได้รับการพัฒนาให้มีความสามารถยิ่งขึ้นเป็นอย่างมากซึ่งช่วยให้พัฒนาโปรแกรมที่สามารถเพิ่มเติมจากโปรแกรมการควบคุมที่มีอยู่เดิม (Active X) การควบคุม ที่สามารถเพิ่มเติมจากโปรแกรมการควบคุมที่มีอยู่เดิม (Active X) สำหรับการเขียนโปรแกรมบนอินเตอร์เน็ตได้อย่างสมบูรณ์และสามารถสร้างไฟล์ .exe ในรูปแบบของรหัสแท้ (Native Code) ได้อีกด้วย โปรแกรมวิชวลเบสิกมีความสามารถและเทคโนโลยีใหม่ๆ ได้โดยโปรแกรมการควบคุมที่สามารถเพิ่มเติมจากโปรแกรมการควบคุมที่มีอยู่เดิม (Active X) ได้ถูกกำหนดเป็นมาตรฐานด้านเทคโนโลยีในปัจจุบันโดยมีความสามารถดังต่อไปนี้

2.8.1 คอมไพล์โปรแกรมในรูปแบบรหัสแท้

โปรแกรมวิชวลเบสิกรุ่น Professional หรือ Enterprise Edition สามารถสนับสนุนการสร้างโปรแกรมแบบ .exe ทั้งรูปแบบของรหัสเทียม (P-Code) หรือรหัสแท้ (Native Code) โดยที่การแปลโปรแกรมเพื่อสร้างไฟล์ .exe ในรูปแบบรหัสแท้นั้นและผู้อ่านสามารถที่จะทำให้รหัสสมบูรณ์ในรูปแบบของ .exe ทั้งในด้านของความเร็วและขนาดของไฟล์ .exe ได้หลายๆ เงื่อนไขอีกด้วย นอกจากนี้ยังสามารถเลือกสร้างรหัส .exe ที่เหมาะสมกับการทำงานเป็นการเฉพาะได้ ทำให้รหัสที่ได้สามารถโหลดและเข้าถึงข้อมูลได้เร็วที่สุด

รหัสเทียมเป็นรหัสที่โปรเซสเซอร์ไม่สามารถที่จะนำไปปฏิบัติการได้โดยตรงเพราะเป็นรหัสที่ถูกออกแบบเป็นการเฉพาะ เพื่อช่วยลดขนาดของไฟล์ .exe ที่ได้จากการแปลโปรแกรม รหัสแท้เป็นรหัสที่โปรเซสเซอร์สามารถนำไปประมวลผลได้ทันที ดังนั้นจึงทำให้โปรแกรมสามารถทำงานได้รวดเร็วกว่ารหัสเทียมเป็นอย่างมาก

2.8.2 การสร้างการควบคุมที่สามารถเพิ่มเติมจากโปรแกรมการควบคุมที่มีอยู่เดิม (Active)

โปรแกรมวิชวลเบสิกสามารถสร้างการควบคุมแบบ .ocx ด้วยเทคโนโลยีแบบการควบคุมที่สามารถเพิ่มเติมจากโปรแกรมการควบคุมที่มีอยู่เดิม (Active X) โดยตรงซึ่งจะทำให้สามารถสร้างการควบคุมแบบ .ocx ได้หลากหลาย โดยที่สามารถจะรวมการควบคุมมาตรฐานเช่น Text Box หรือ Picture Box เป็นต้น

2.8.3 การทำงานแบบหลายโปรแกรม

เนื่องจากโปรแกรมวิชวลเบสิกสนับสนุนการสร้างการควบคุม .ocx ดังนั้นในขั้นตอนการทดสอบรหัสของการควบคุมแบบ .ocx นั้น ต้องดำเนินการเช่นเดียวกับการใช้ควบคุมในโปรแกรมทั่วๆ ไปโดยการเพิ่มการควบคุมดังกล่าวลงในรูปแบบของโปรแกรมอื่นๆ เพราะ การควบคุม .ocx ไม่สามารถทำงานได้โดยลำพังเพื่อรองรับการทดสอบการควบคุม .ocx ที่สร้างขึ้นมาต้องมีการสนับสนุนการทำงานในลักษณะของหลายๆ โปรแกรม สำหรับความสามารถที่สำคัญของโปรแกรมวิชวลเบสิกคือสนับสนุนการพัฒนาโปรแกรมในรูปแบบลูกข่าย-แม่ข่าย

2.8.4 การสร้างโปรแกรมที่ทำงานภายใต้ระบบเครือข่ายสามชั้น

สามารถสร้างโปรแกรมที่ทำงานภายใต้ระบบเครือข่ายสามชั้น (Three-Tiered) โดยอาศัยสถาปัตยกรรมเทคโนโลยีแบบส่วนประกอบของวัตถุแบบกระจาย (Distributed component object Model: DCOM) สำหรับโปรแกรมด้านลูกข่ายแบบ 32 บิต ซึ่งสามารถที่จะทำงานภายใต้ระบบเครือข่ายหรืออินเตอร์เน็ตได้อย่างมีประสิทธิภาพ

2.8.5 การสร้างโปรแกรมเชื่อมต่อ

สามารถสร้างการเชื่อมต่อแบบการติดต่อสื่อสารทางเดียว (Single-Document Interface: SDI) หรือแบบการติดต่อสื่อสารหลายทาง (Multi-Document Interface: MDI) และนอกจากนี้ยังสามารถสร้างโปรแกรมที่มีลักษณะเหมือนกันกับโปรแกรม Windows Explorer ซึ่งการสร้างโปรแกรมในรูปแบบ Single-Document Interface หรือ Multi-Document Interface ขึ้นกับความเหมาะสมของโปรแกรมแต่ละประเภท

2.8.6 การออกแบบหรือสร้างโครงร่างแบบอัตโนมัติ (Wizard)

การสร้างโปรแกรมหรือการควบคุมด้วยโปรแกรมวิชวลเบสิกมิใช่สิ่งที่ยากอีกต่อไปเพราะโปรแกรมวิชวลเบสิกได้มีการออกแบบหรือสร้างโครงร่างแบบอัตโนมัติ ที่จะช่วยออกแบบ หรือสร้างโครงร่าง (Layout) ของโปรแกรมหรือควบคุมให้โดยอัตโนมัติ

2.8.7 วิธีการเพิ่มความสามารถของโปรแกรมวิชวลเบสิก (Extensibility Model)

สภาพแวดล้อมของโปรแกรมวิชวลเบสิกได้รับการพัฒนาให้มีความสามารถซึ่งมีความยืดหยุ่นมากขึ้นและสามารถเพิ่มโปรแกรมเพื่อขยายขีดความสามารถของโปรแกรมวิชวลเบสิกได้อย่างเต็มที่ สามารถปรับปรุงหรือตบแต่งสภาพแวดล้อมในการพัฒนาโปรแกรมได้

2.8.8 การสร้างเหตุการณ์

สามารถสร้างเหตุการณ์ (Event) ให้เป็นขั้นตอนได้และสามารถเลือกเหตุการณ์ใดๆ ให้มีการกระตุ้น (Activate) ตามสถานการณ์ที่เหมาะสมได้

2.8.9 การจดจำคำสั่งหรือความสามารถต่าง ๆ

ความสามารถของการจดจำคำสั่งหรือความสามารถต่างๆ (Smart Code Editor) จะช่วยให้ไม่ต้องจดจำคำสั่งหรือความสามารถต่างๆ ของลักษณะควบคุมอีกต่อไป สาเหตุที่โปรแกรมวิชวลเบสิกเป็นภาษาที่เหมาะสำหรับการเรียนรู้ในการเขียนโปรแกรมนั้นเนื่องจากโปรแกรมวิชวลเบสิกมีข้อดีหลายประการคือ

2.8.9.1 ง่ายต่อการเรียนรู้สำหรับผู้เริ่มต้น ทั้งเรื่องไวยากรณ์ของภาษาเองและเครื่องมือสำหรับใช้งาน ดังชื่อที่ได้บอกไว้แล้วว่าเป็นพื้นฐานซึ่งเหมาะสำหรับผู้เริ่มต้น

2.8.9.2 ความนิยมของภาษา โดยอาจกล่าวได้ว่าภาษาเบสิกเป็นภาษาที่มีคนเรียนรู้กันมาก และใช้งานมากที่สุดในประวัติศาสตร์ของคอมพิวเตอร์

2.8.9.3 การพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วในด้านการประมวลผลและเรื่องความสามารถใหม่ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต เป็นต้น

2.8.9.4 ผู้พัฒนาสำคัญของโปรแกรมวิชวลเบสิก คือบริษัทไมโครซอฟท์ซึ่งจัดได้ว่าเป็นบริษัทที่ได้การยอมรับในวงการคอมพิวเตอร์ปัจจุบัน เราจึงมั่นใจว่าโปรแกรมวิชวลเบสิกจะมีการพัฒนาและปรับปรุงและคงอยู่ไปอีกนาน