

# คู่มือการใช้งาน

**ET IN-SYSTEM DOWNLOAD**

**CP-SPI/RD2 V1**

**CP-SPI/RD2 V2**

**CP-SPI/RD2 V3**

**CPU P89C51RD2**

**64KBYTE FLASH**



**CP-SPI/S8252 V1**

**CP-SPI/S8252 V2**

**CP-SPI/S8252 V3**

**CPU AT89S8252**

**8KBYTE FLASH**

**ETT**  
[www.ett.co.th](http://www.ett.co.th)

**บริษัท อีทีที จำกัด**

1112/96-98 ถนนสุขุมวิท แขวงพระโขนง เขตคลองเตย กรุงเทพฯ 10110 <http://www.etteam.com>

1112/96-98 Sukhumvit Rd., Phrakonong Klongtoey BANGKOK 10110 <http://www.ett.co.th>

TEL 02-712 1120 FAX 02-391 7216

e-mail:sale@etteam.com

## คำนำ

ด้วยความโดดเด่นในหลายๆ ด้านของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทำให้มีผู้นำไปใช้งานอย่างแพร่หลาย ซึ่งมีผลทำให้ไมโครคอนโทรลเลอร์ในตระกูลนี้ได้รับความนิยมอย่างต่อเนื่องตลอดมาจึงทำให้มีผู้ผลิตหลายบริษัทที่ผลิตไมโครคอนโทรลเลอร์ในตระกูลนี้ออกมาสู่ตลาดเพื่อตอบสนองความต้องการของผู้ใช้ โดยผู้ผลิตในแต่ละรายจะเพิ่มความสามารถด้านต่างๆ ลงไปในไมโครคอนโทรลเลอร์ตระกูลนี้ อาทิเช่น การบรรจุหน่วยความจำแบบ FLASH ลงในตำแหน่ง ROM , การเพิ่มหน่วยความจำแบบ EEPROM เพื่อใช้ในการเก็บข้อมูล , เพิ่มความเร็วของการทำงานใน 1 รอบคำสั่ง และ อื่นๆอีกมากมาย

ด้วยเหตุนี้เองที่เป็นแรงผลักดันให้ไมโครคอนโทรลเลอร์ตระกูลนี้ถูกนำไปใช้งานกว้างขวางมากขึ้น ดังนั้นทางทีมงาน ETT จึงได้พัฒนาบอร์ด CP-SPI/S8252 & CP-SPI/RD2 ขึ้นมาซึ่งมีทั้งหมด 3 รุ่น CPU 2 แบบ ด้วยกัน คือ V1.0 , V2.0 และ V3.0 โดยรุ่นที่เป็น CP-SPI/S8252 จะใช้ CPU AT89S8252 เป็น CPU ประจำบอร์ด และรุ่น CP-SPI/RD2 จะใช้ CPU P89C51RD2 เป็น CPU ประจำบอร์ด

ในส่วนของการพัฒนาโปรแกรมก็มีความสะดวกมากกว่าบอร์ดรุ่นเก่าๆ โดยสามารถ Download โปรแกรมจาก PC ลงในบอร์ดโดยผ่านทาง RS232 โดยไม่ต้องมีอุปกรณ์อื่น ๆ มาเสริมในรุ่น CP-SPI/RD2 และผ่านทาง Printer Port ในรุ่น CP-SPI/S8252 ซึ่งทำให้ประหยัดค่าใช้จ่ายด้วยเหตุนี้ทาง ETT จึงได้พัฒนาบอร์ด CP-SPI/S8252 & CP-SPI/RD2 ทั้ง 3 รุ่น ขึ้นมาเพื่อตอบสนองความต้องการกับผู้ใช้และผู้ผลิต CPU

และสำหรับหนังสือเล่มนี้ เป็นหนังสือคู่มือการใช้งานของบอร์ด CP-SPI/S8252 & CP-SPI/RD2 ทั้ง 3 รุ่น คือ V1.0 , V2.0 และ V3.0 ซึ่งจะกล่าวถึงรายละเอียดของบอร์ดแต่ละรุ่น แยกไว้เป็นส่วนสำคัญ โดยเนื้อหาภายในจะกล่าวถึงรายละเอียดของอุปกรณ์สำคัญต่างๆภายในบอร์ดทั้ง 3 รุ่น ตลอดจนขั้นตอนของการ DOWNLOAD PROGRAM และวิธีการใช้งานโปรแกรมที่ใช้สำหรับ DOWNLOAD ทางผู้เขียนและทีมงานอีทีที หวังเป็นอย่างยิ่งว่าคู่มือการใช้งานเล่มนี้ จะสามารถนำไปใช้ควบคู่กับบอร์ด CP-SPI/S8252 & CP-SPI/RD2 ทั้ง 3 รุ่น และใช้สำหรับเป็นเนื้อหาอ้างอิงได้เป็นอย่างดี

เมษายน 2545

ทีมงาน ETT

## สารบัญ

FEARTURE OF CPU	1
SPECFICATION	3
CP-SPI/S8252 V1.0&CP-SPI/RD2 V1.0	
POWER SUPPLY	5
RS232	6
<b>วงจร</b> CP-SPI/S8252 V1.0&CP-SPI/RD2 V1.0	8
CP-SPI/S8252 V2.0&CP-SPI/RD2 V2.0	
RS422 / RS485	9
DS1307	12
EEPROM 2416	17
ADS7841	22
LTC1661	26
LCD แบบ CHARACTER	30
EXP2	31
ET-SDP8	33
<b>วงจร</b> CP-SPI/S8252 V2.0&CP-SPI/RD2 V2.0	35
CP-SPI/S8252 V3.0&CP-SPI/RD2 V3.0	
PCF8574	36
34 PIN I/O BUS	39
<b>วงจร</b> CP-SPI/S8252 V2.0&CP-SPI/RD2 V2.0	42
DOWNLOAD	43
<b>ภาคผนวก</b>	
การใช้งานโปรแกรม WS8252	ผ.1
การใช้งานโปรแกรม P89C51RD2	ผ.4
ตำแหน่งการวางอุปกรณ์	ผ.8

---

## FEARTURE OF CPU (AT89S8252)

- Compatible with MCS-51 TM Products
- 8 K of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance : 1,000 Write/Erase Cycles
- 2 K Bytes EEPROM
  - Endurance : 100,000 Write/Erase Cycles
- 4.0V to 6Voperating Range
- Fully Static Operation : 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 X 8 Bit Internal RAM
- 32 Programmable I/O Line
- Three 16 Bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

---

## FEARTURE OF CPU (P89C51RD2)

- 8051 Central Processing Unit
- On-Chip Flash Program Memory with In-System Programing (ISP) and In-Application Programming (IAP) capability
  - Boot ROM contains low level Flash programing routines for downloading via UART
  - Can be programed by the end-user application (IAP)
  - 6 clock per machine cycle operation (standard)
  - 12 clock per machin cycle operation (optional)
  - Speed up to 20 MHz with 6 clock cycles per machine cycle (40 MHz equivalent performance); up to 33 MHz with 12 clocks per machine cycle
- Fully static operation
- RAM expandable externally to 64 kB
- 4 level priority interrupt
- 7 interrupt source
- Four 8 bit i/o Port
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power Control Mode
  - Clock and be stopped and resumed
  - Idle mode
  - Power down mode
- Programable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Programable Counter Array
- Programmable Counter Array (PCA)
  - PWM
  - Capture/compare

SPECIFICATION OF CP-SPI/S8252 V1.0 &CP-SPI/RD2 V1.0

CPU	AT89S8252 (CP-SPI/S8252 V1.0)
	P89C51RD2 x2 Clock (CP-SPI/RD2 V1.0)
XTAL	18.432 MHz
ROM	8 Kbyte Internal (AT89S8252) 2 Kbyte EEPROM 64 Kbyte Internal (P89C51RD2)
EEPROM	2 Kbyte EEPROM (AT89S8252)
RS232	YES (MAX232)
POWER SUPPLY	YES
ISP DOWNLOAD	YES

SPECIFICATION OF CP-SPI/S8252 V2.0 &CP-SPI/RD2 V2.0

CPU	AT89S8252 (CP-SPI/S8252 V2.0)
	P89C51RD2 x2 Clock (CP-SPI/RD2 V2.0)
XTAL	18.432 MHz
ROM	8 Kbyte Internal (AT89S8252) 2 Kbyte EEPROM 64 Kbyte Internal (P89C51RD2)
EEPROM	2 Kbyte EEPROM (AT89S8252)
LCD	CHARACTER
RS232	YES (MAX232)
RS422	YES 75176 (Option)
RS485 (FULL)	YES 75176 (Option)
RS485 (HALF)	YES 75176 (Option)
ADC	12 BIT 4 CHANNEL (ADS7841)
DAC	10 BIT 2 CHANNEL (LTC1661)
EEPROM	24XX (Option)
RTC	DS1307 (Option)
OPEN COLLECTOR	400 mA 8 Bit (NC6B595)
I <sup>2</sup> C BUS	YES

SPECIFICATION OF CP-SPI/S8252 V2.0 &CP-SPI/RD2 V2.0 (ต่อ)

ET-SDP8	YES
POWER SUPPLY	YES
ISP DOWNLOAD	YES

CP-SPI/S8252 V3.0 &CP-SPI/RD2 V3.0

CPU	AT89S8252 (CP-SPI/S8252 V3.0)
	P89C51RD2 x2 Clock (CP-SPI/RD2 V3.0)
XTAL	18.432 MHz
ROM	8 Kbyte Internal (AT89S8252) 2 Kbyte EEPROM 64 Kbyte Internal (P89C51RD2)
EEPROM	2 Kbyte EEPROM (AT89S8252)
LCD	CHARACTER
RS232	YES (MAX232)
RS422	YES 75176 (Option)
RS485 (FULL)	YES 75176 (Option)
RS485 (HALF)	YES 75176 (Option)
ADC	12 BIT 24 CHANNEL (ADS7841)
DAC	10 BIT 2 CHANNEL (LTC1661)
EEPROM	24XX (Option)
RTC	DS1307 (Option)
OPEN COLLECTOR	400 mA 8 Bit (NC6B595)
I <sup>2</sup> C BUS	YES
I <sup>2</sup> C PORT	PCF8574
ET-SDP8	YES
34 PIN I/O BUS	YES
POWER SUPPLY	YES
ISP DOWNLOAD	YES

**CP-SPI/S8252 V1.0&CP-SPI/RD2 V1.0**

บอร์ด CP-SPI/S8252 V1.0 จะใช้ CPU AT89S8252 เป็น CPU ประจำบอร์ด และบอร์ด CP-SPI/RD2 V1.0 จะใช้ CPU P89C51RD2 เป็น CPU ประจำบอร์ด โดยมีส่วนอื่นๆ ของวงจรที่เหมือนกัน ส่วนการ DOWNLOAD โปรแกรมเข้าสู่ตัวบอร์ดนั้น บอร์ด CP-SPI/S8252 V1.0 จะใช้โปรแกรม WS8252 และบอร์ด CP-SPI/RD2 V1.0 จะใช้โปรแกรม P89C51RD2

บนบอร์ด CP-SPI/S8252 V1.0 & CP-SPI/RD2 V1.0 นอกเหนือจาก CPU แล้ว จะประกอบด้วย อุปกรณ์สำคัญต่าง ๆ ดังนี้

1. Power Supply
  - ขั้ว DC Jack
  - ขั้ว Terminal ชั้นนี้สุด 2 Pin
2. RS232

โดยรายละเอียดของอุปกรณ์บนบอร์ด CP-SPI/S8252 V1.0 & CP-SPI/RD2 V1.0 จะขอกล่าวถึง เฉพาะอุปกรณ์ที่สำคัญบางตัวเท่านั้น

#### POWER SUPPLY

บอร์ด CP-SPI/S8252 V1.0 & CP-SPI/RD2 V1.0 มีชุด POWER SUPPLY (7805) อยู่บนบอร์ด อยู่แล้ว โดยเพียงแต่ป้อนไฟเข้าที่ขั้วอินพุตก็สามารถใช้งานได้ โดยบนบอร์ดออกแบบให้มีขั้วป้อนไฟ 2 ขั้ว โดยขั้วแรกเป็น JACK DC ที่เหมาะกับ ADAPTER ส่วนอีกขั้วหนึ่งเป็นลักษณะของ TERMINAL ชั้นนี้สุด 2 PIN ที่เหมาะกับการต่อกับสายไฟ ซึ่งทั้ง 2 ขั้วนี้จะขนานกันอยู่ สามารถที่จะเลือกใช้ขั้วใดขั้วหนึ่งก็ได้ตามความเหมาะสม

โดยชุด POWER SUPPLY จะมีชุด Regulator สำหรับปรับระดับแรงดันไฟให้อยู่ที่ 5 Vdc ลงที่เพื่อใช้งานบนบอร์ด โดยในส่วนของ Rectifier จะใช้วงจรบริดจ์ ซึ่งทำให้บอร์ด CP-SPI/S8252V1.0 & CP-SPI/RD2 V1.0 สามารถรับไฟได้ทั้งไฟกระแสตรง และไฟกระแสสลับ ซึ่งถ้าเป็นการใช้ไฟกระแสตรงก็ไม่ต้องระวังเรื่องขั้วต่อ เนื่องจากส่วนของ Rectifier เป็นวงจรแบบบริดจ์

ระดับแรงดันไฟที่ต้องป้อนให้กับบอร์ด CP-SPI/S8252V1.0 & CP-SPI/RD2 V1.0 แสดงในตารางด้านล่าง



ชนิดของไฟฟ้า	แรงดันไฟฟ้า
ไฟฟ้ากระแสตรง	9 - 12 V
ไฟฟ้ากระแสสลับ	6 - 9 V

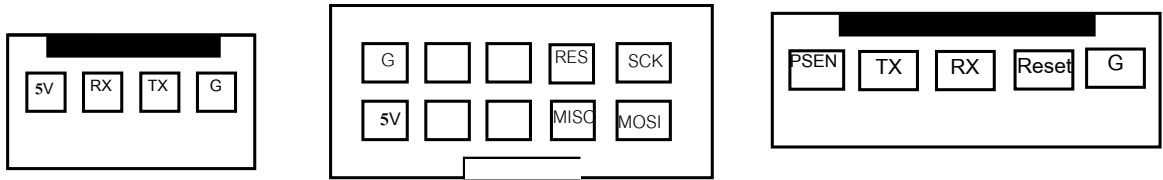
ตารางแสดงค่าระดับแรงดันไฟที่ต้องป้อนให้กับบอร์ด CP-SPI/S8252V1.0 & CP-SPI/RD2 V1.0

## RS232

โดยทั่วไปแล้วระบบของการสื่อสารข้อมูลแบบอนุกรม Asynchronous นั้น นับว่าเป็นระบบของการสื่อสารข้อมูลที่มีประสิทธิภาพดีอีกแบบหนึ่ง ที่มีการพัฒนาขีดความสามารถในการสื่อสารกันเรื่อยมาเป็นลำดับ ซึ่งในปัจจุบันก็ยังเป็นที่นิยมใช้กันอย่างแพร่หลาย โดยเฉพาะอย่างยิ่งการสื่อสารกับเครื่องคอมพิวเตอร์ PC ที่ใช้งานกันอยู่ในปัจจุบันนี้เองก็ยังมีการบรรจุเอาวงจรการสื่อสารอนุกรม Asynchronous รวมเอาไว้ในระบบพื้นฐานของเครื่องด้วยทุกเครื่องเสมอ หรืออาจเรียกได้ว่า มันเป็นอุปกรณ์มาตรฐานของเครื่องคอมพิวเตอร์ที่ขาดไม่ได้เลยก็ว่าได้ ทั้งนี้ก็เนื่องมาจากการสื่อสารแบบอนุกรม นี้มีขีดความสามารถในการรับส่งข้อมูลที่ได้ผลดีและเสียค่าใช้จ่ายน้อย จึงส่งผลให้การสื่อสารแบบนี้ได้รับการพัฒนาและปรับปรุงคุณภาพเรื่อยมาจนกลายเป็นมาตรฐานไปในที่สุด โดยระบบของการสื่อสารแบบนี้ เราสามารถพบเห็นกันได้โดยทั่วไป แต่อาจมีชื่อเรียกแตกต่างกันออกไป ทั้งนี้ขึ้นอยู่กับวงจรภาคที่ใช้เปลี่ยนแปลงระดับของสัญญาณ Logic TTL จากภาคส่ง (Transmitter) ก่อนที่จะส่งสัญญาณนั้นเข้าไปยังสายส่งสัญญาณ และวงจรที่ใช้แปลงระดับของสัญญาณที่รับมาได้จากสายส่ง ก่อนจะส่งให้กับวงจรของภาครับอีกครั้งหนึ่ง ซึ่งนิยมเรียกวางจรส่วนนี้ว่า “Line Driver”

การนำสัญญาณที่ใช้ในการรับส่งข้อมูลแบบอนุกรม (Serial Port) นี้ไปผ่านวงจร เพื่อเปลี่ยนระดับของสัญญาณลอจิก TTL ของภาคส่งให้มีขนาดสูงขึ้นเป็น 12 V เพื่อให้สามารถส่งสัญญาณไปในสายส่งสัญญาณให้ไ้ระยะทางที่ไกลมากขึ้น และในส่วนของภาครับนี้เองก็ต้องทำการเปลี่ยนระดับของสัญญาณที่รับได้จากสายส่งสัญญาณที่เป็นขนาด 12V ให้กลับมาเป็นระดับลอจิก TTL มาตรฐาน เพื่อส่งสัญญาณให้กับวงจรภาครับอีกครั้งหนึ่ง โดยวงจร Line Driver แบบนี้จะเรียกกันโดยทั่วไปว่า “RS232” โดยคุณสมบัติของ RS232 นี้สามารถรับส่งข้อมูลได้ผลดีที่ระยะทางประมาณ 50 ฟุต ทั้งนี้ก็เนื่องมาจาก หากสายส่งสัญญาณมีความยาวมาก ๆ จะทำให้เกิดการสูญเสียของระดับแรงดันในสายส่งจนวงจรที่ภาครับไม่สามารถตรวจสอบระดับของสัญญาณที่ต่ำเกินไปได้ จึงทำให้การรับส่งข้อมูลในระยะทางที่ไกล ๆ ไม่ได้ผลดีเท่าที่ควรและทำให้เกิดความผิดพลาดมากขึ้นการรับส่งที่ใช้มาตรฐานแบบ RS232 ที่พบเห็นกันได้บ่อย ๆ ทั่วไป ได้แก่ Serial Port ของเครื่องคอมพิวเตอร์ส่วนบุคคลที่นิยมเรียกกันว่า Com Port หรือ Port Mouse ซึ่งในเครื่องอาจจะมีหลายชุด บางคนจึงนิยมเรียกกันว่า Com 1 หรือ Com 2 นั่นเอง

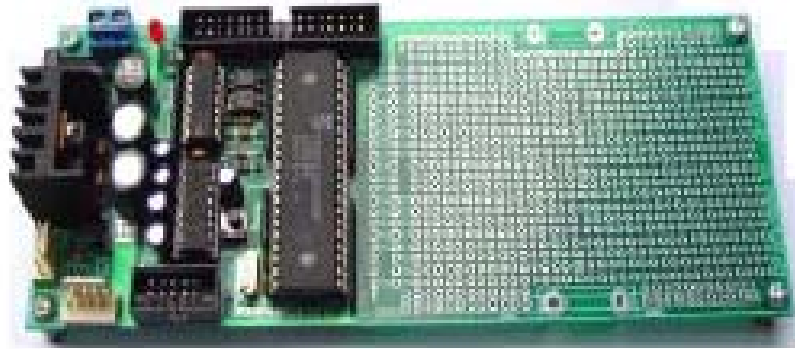
จากรูปที่ 1 แสดงให้เห็นถึงขั้วต่อใช้งานของ RS232 บนบอร์ด CP-SPI/S8252 V1.0 และ CP-SPI/RD2 V1.0 และรูปที่ 2.1 และ 2.2 เป็นขั้ว ซึ่งใช้ในการ LOAD SPI



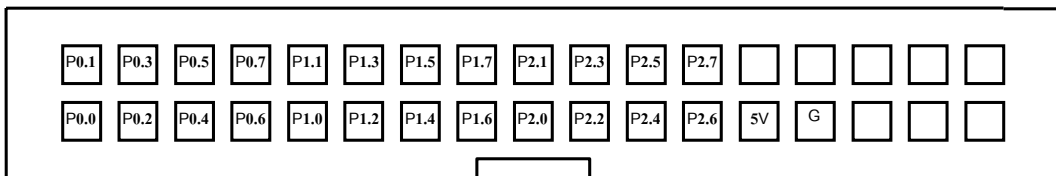
รูปที่ 1 แสดงขั้วต่อ RS232

รูปที่ 2.1 แสดงขั้วต่อ LOAD(S8252) รูปที่ 2.2แสดงขั้วต่อ

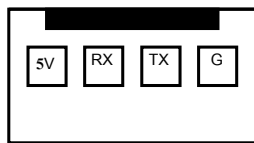
LOAD(RD2)



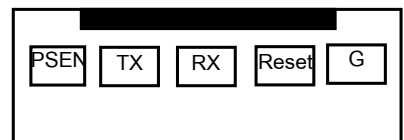
รูปที่ 3 แสดงบอร์ด CP-SPI/S8252 V1.0&CP-SPI/RD2 V1.0



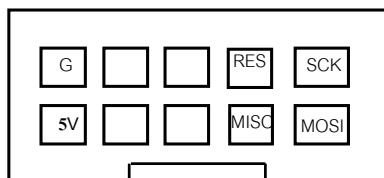
a) ขั้วต่อ 72IOZ80 ของ CPU



b) ขั้วต่อ RS232

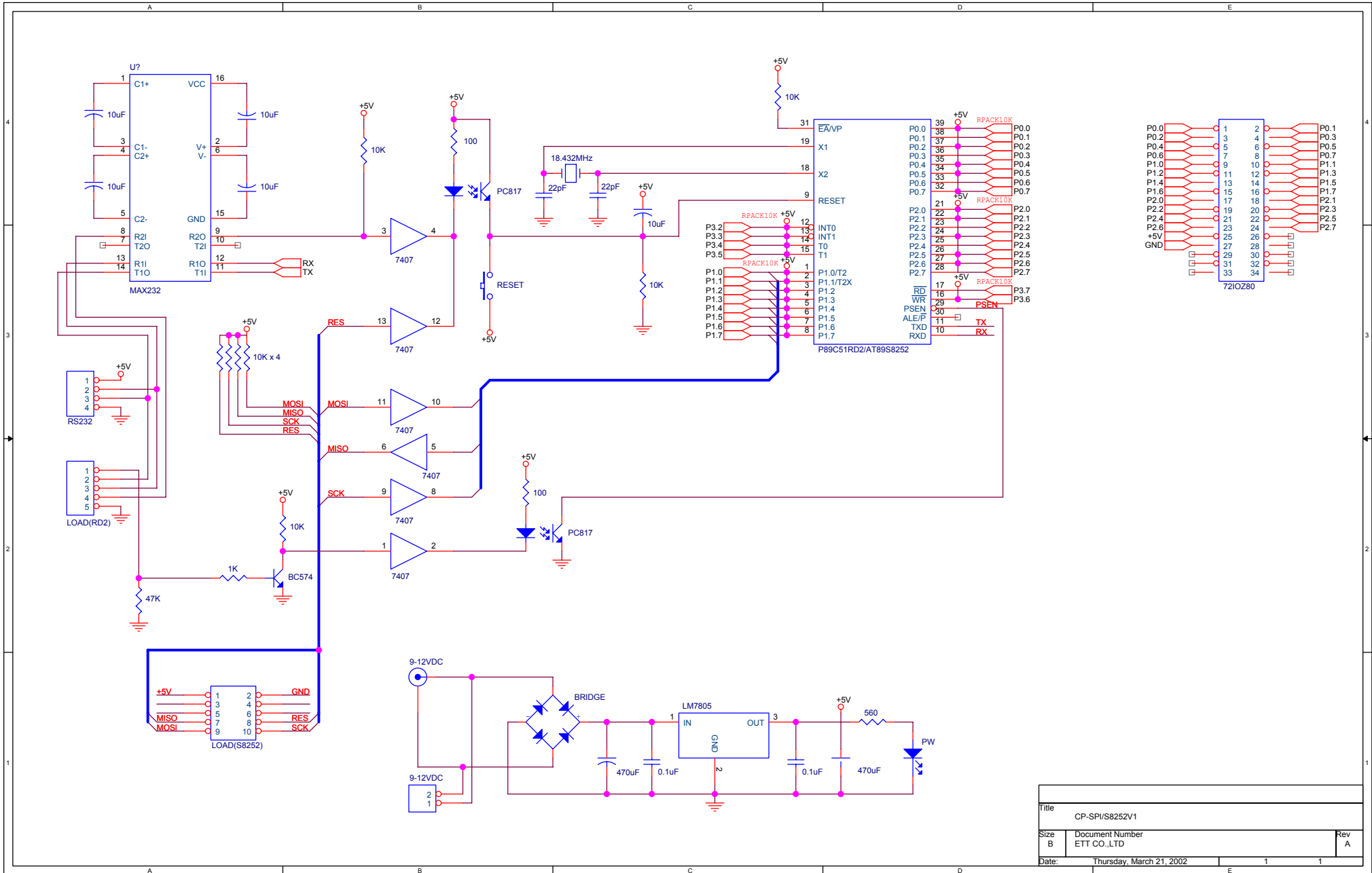


c) ขั้วต่อ Load (RD2)



d) ขั้วต่อ Load(S8252)

รูปที่ 4 แสดงขั้วต่อใช้งานต่าง ๆ บนบอร์ด CP-SPI/S8252 V1.0&CP-SPI/RD2 V1.0



Title		
CP-SPI/S8252V1		
Size	Document Number	Rev
B	ETT CO.,LTD	A
Date:	Thursday, March 21, 2002	1 1

**CP-SPI/S8252 V2.0&CP-SPI/RD2 V2.0**

บอร์ด CP-SPI/S8252 V2.0 จะใช้ CPU AT89S8252 เป็น CPU ประจำบอร์ด และบอร์ด CP-SPI/RD2 V2.0 จะใช้ CPU P89C51RD2 เป็น CPU ประจำบอร์ด โดยมีส่วนอื่นๆ ของวงจรที่เหมือนกัน ส่วนการ DOWNLOAD โปรแกรมเข้าสู่ตัวบอร์ดนั้น บอร์ด CP-SPI/S8252 จะใช้โปรแกรม WS8252 และ บอร์ด CP-SPI/RD2 จะใช้โปรแกรม P89C51RD2

บอร์ด CP-SPI/S8252 V2.0 & CP-SPI/RD2 V2.0 จะมีอุปกรณ์ต่าง ๆ เหมือนกับ CP-SPI/S8252 V1.0 & CP-SPI/RD2 V1.0 ทุกประการแต่เพิ่มเติมอุปกรณ์บางตัวเข้าไป ดังนี้

1. RS422 / 485
2. RTC (DS1307)
3. I<sup>2</sup>C (EEPROM 24XX)
4. ADC (ADS7841)
5. DAC (LTC1661)
6. LCD แบบ Character
7. EXP2 (NC6B595)
8. ET-SDP8

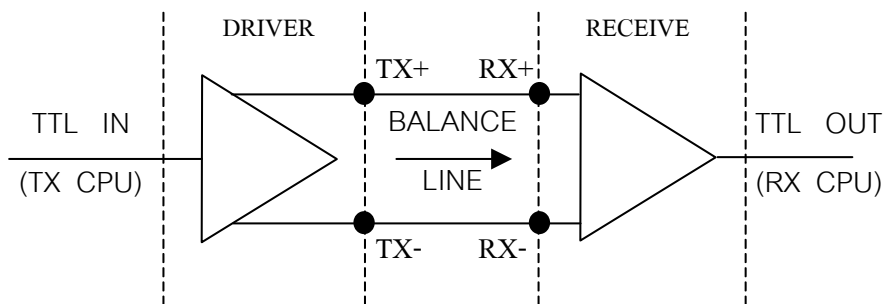
### RS422 และ RS485

เนื่องจากการรับส่งข้อมูลแบบอนุกรมโดยใช้วงจร Line Driver แบบ RS232 นั้นยังมีข้อจำกัดในเรื่องระยะทางซึ่งไม่สามารถปรับปรุงให้ส่งได้ไกลขึ้นกว่าที่เป็นอยู่ได้และยังไม่สามารถเชื่อมต่อกัน ครั้งละหลาย ๆ อุปกรณ์ในเวลาเดียวกันได้ ดังนั้นจึงมีการสร้างวงจร Line Driver แบบใหม่ขึ้นมาทดแทนเพื่อให้สามารถรับส่งสัญญาณได้ในระยะทางที่ไกลขึ้นกว่าเดิมและวิธีการอย่างหนึ่งที่สามารถสร้างได้ง่ายและมีราคาถูกโดยการใช่วงจร Line Driver ที่ใช้สำหรับเปลี่ยนระดับสัญญาณเปลี่ยนระดับของสัญญาณลอจิก TTL จากวงจรภาคส่งไปเป็นสัญญาณแบบ Balance Line ซึ่งอาศัยหลักของการขยายความแตกต่างของสัญญาณในสายรับส่ง ซึ่งจากคุณสมบัติอันนี้ทำให้วงจรในส่วนรับของ Balance Line นั้นสามารถตรวจจับสัญญาณในสายส่งซึ่งมีขนาดตั้งแต่ 200 mV ขึ้นไปได้ ดังนั้นวงจรแปลงระดับสัญญาณ Line Driver แบบนี้จึงสามารถรับส่งข้อมูลในระยะทางไกล ๆ กันได้ โดยถ้าหากความเร็วในการรับส่งน้อยกว่า 10Mbps แล้วจะสามารถรับส่งกันได้ผลดีในระยะทางไม่เกิน 4,000 ฟุต หรือ 1,200 เมตร โดยประมาณ แต่ถ้าความเร็วใน

การรับส่งมากขึ้นระยะทางก็อาจจะลดต่ำลงตามสัดส่วน โดยวงจร Line Driver แบบนี้ถูกกำหนดและยึดถือเป็นมาตรฐานอีกแบบหนึ่งโดย เรียกกันว่า “RS422” ซึ่งนิยมนำมาใช้ทดแทนวิธีการรับส่งข้อมูลแบบ RS232 เพื่อเพิ่มระยะทางการรับส่งให้ได้ไกลมากขึ้น และในปัจจุบันได้มีการพัฒนาวงจร Line Driver แบบนี้ให้สามารถควบคุมสัญญาณเอาต์พุตให้เป็นแบบ Tri-State ได้ ดังนั้นจึงทำให้สามารถทำการรับส่งข้อมูลแบบ 2 ทิศทางในสายส่งเพียงคู่เดียวกันได้ จากคุณสมบัติอันนี้ จึงทำให้เราสามารถเชื่อมต่ออุปกรณ์ได้หลายๆ ชุด ร่วมกันในสายส่งเพียงคู่เดียวเพื่อทำเป็นระบบ Network แบบ Multidrop ได้ และเรียกการจัดวงจรการสื่อสารแบบนี้ว่า “RS485” ซึ่งมีคุณสมบัติของการรับส่งข้อมูลจะเหมือนกับแบบ RS422 เพียงแต่ต้องมีการเขียนโปรแกรมมาควบคุม และจัดลำดับการรับส่งข้อมูลของอุปกรณ์ที่ต่อใช้งานร่วมกันในสายส่งไม่ให้ส่งสัญญาณออกมาในสายส่งพร้อมกันเพราะจะทำให้เกิดการชนกันของข้อมูลทำให้ฝ่ายรับข้อมูลผิดพลาด

#### การเลือก Line Driver แบบ RS422

ในงานด้านการสื่อสารแบบอนุกรมนั้น บางครั้งจำเป็นต้องสื่อสารกันในระยะทางที่ห่างไกลกันมาก ซึ่งการรับส่งแบบ RS232 ไม่สามารถกระทำได้ ทางเลือกหนึ่งซึ่งสามารถนำมาทดแทนเพื่อแก้ปัญหาเรื่องระยะทางโดยไม่ต้องตัดแปลงแก้ไขโปรแกรมเดิมที่เขียนไว้แล้วก็คือการเปลี่ยนวงจรภาค Line Driver จากระบบ RS232 มาเป็นระบบ RS422 แทน โดยวงจรแบบ RS422 จะใช้เทคนิคการรับส่งแบบ Balance Line ซึ่งในวงจร Driver แบบนี้ จะสามารถรับส่งสัญญาณที่มีค่าระหว่าง 2 V ถึง 6 V ได้ดี และในส่วนของวงจรภาครับเอง ก็ยังสามารถตรวจจับสัญญาณที่มีขนาดต่ำถึง 200 mV ได้ด้วย ซึ่งหากการรับส่งโดยใช้อุปกรณ์ที่ตรงตามข้อกำหนดแบบนี้จะสามารถรับส่งกันได้ไกลถึง 1,000 ฟุต แต่ถ้าความเร็วที่ใช้ในการรับส่ง มีค่าต่ำกว่า 10Mbps แล้วยังสามารถเพิ่มระยะทางในการรับส่งได้ถึง 4,000 ฟุต อีกด้วย แต่อย่างไรก็ตามระยะทางในการรับส่งที่ได้ผลดีนั้นยังขึ้นอยู่กับองค์ประกอบอื่นๆ ด้วย เช่นคุณภาพของสายสัญญาณและขั้วต่อของสัญญาณรวมไปถึงระดับของสัญญาณรบกวนที่สายเดินผ่านไป เป็นต้น

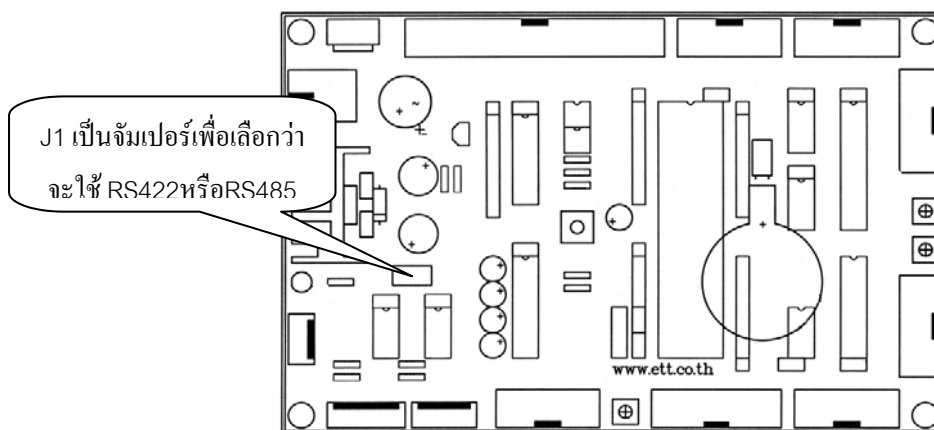


รูปที่ 5 แสดงวิธีการรับส่งข้อมูลแบบ RS422 ในอุดมคติ

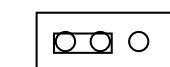
### การเลือกใช้ Line Driver แบบ RS485

การรับส่งข้อมูลแบบนี้ จะมีลักษณะคล้ายกับแบบ RS422 ทั้งหมด แต่การรับส่งแบบนี้จะมีข้อดีพิเศษกว่าก็คือใช้สายสัญญาณในการรับส่งเพียงคู่เดียวเท่านั้น ซึ่งการรับส่งแบบนี้จะใช้การรับส่งแบบ Half Duplex กล่าวคือ สามารถทำหน้าที่ส่งข้อมูลและรับข้อมูลได้โดยใช้สายสัญญาณเดิมเพียงคู่เดียวแต่ในการรับหรือส่งข้อมูลนั้นจะต้องทำแบบผลัดกันรับผลัดกันส่งไม่สามารถทำหน้าที่พร้อมกันได้ทั้ง 2 อย่างในเวลาเดียวกันได้

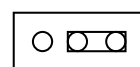
เนื่องจากการสื่อสารแบบนี้ จะใช้สายสัญญาณเพียง 1 คู่ (2 เส้น ) ในการทำหน้าที่ทั้งรับและส่งและยังสามารถต่ออุปกรณ์ร่วมกันได้มากถึง 32 ตัว ในกรณีที่ใช้ IC 75176 และได้ถึง 256 ตัว ในกรณีที่ใช้ IC MAX3088 พร้อมกันในเวลาเดียว ดังนั้นจึงต้องมีการจัดลำดับการรับส่งข้อมูลที่ดีจึงจะสามารถรับส่งกันได้ อย่างมีประสิทธิภาพตามต้องการได้จึงส่งผลทำให้การรับส่งแบบนี้มีความสลับซับซ้อนมากขึ้น เพราะต้องมีการเขียนโปรแกรมเพื่อกำหนดทิศทางรวมทั้งจัดแบ่ง ควบคุมลำดับการสื่อสารภายในสายส่งให้กับอุปกรณ์แต่ละตัวที่อยู่จะทำให้ให้อุปกรณ์ตัวใดทำหน้าที่เป็นตัวส่ง และให้อุปกรณ์ตัวใดทำหน้าที่เป็นตัวรับ ซึ่งจะต้องเขียนโปรแกรมเพื่อควบคุมให้มีการรับส่งข้อมูลออกมาได้เพียงครั้งละ 1 ตัว เท่านั้น เพราะถ้าหากมีการส่งข้อมูลออกมาในสายสัญญาณพร้อมกันในเวลาเดียวกันมากกว่า 1 ตัว แล้วก็จะทำให้ฝ่ายรับไม่สามารถรับข้อมูลที่ถูกต้องทำให้เกิดความผิดพลาดขึ้นได้ ดังนั้นการใช้ระบบ RS485 นี้ จะต้องมีการออกแบบและวางแผนไว้ล่วงหน้าทั้งทางด้านของอุปกรณ์ฮาร์ดแวร์และโปรแกรมที่จะใช้ในการควบคุมการทำงานของระบบ สำหรับวิธีการเชื่อมต่อระบบ RS485 นั้นทำได้หลายแบบ เช่น ต่อกันแบบขนานทั้งระบบ และ ต่อกันแบบ Point – to – Point



รูปที่ 6 แสดงตำแหน่งของจัมเปอร์ J1 สำหรับเลือกใช้งาน RS422/485

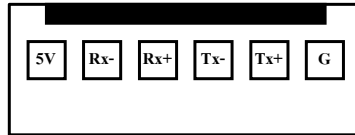


เลือกใช้งาน RS422



เลือกใช้งาน RS485

รูปที่ 7 แสดงการเลือกจัมเปอร์ J1 เพื่อใช้งาน RS422/RS485



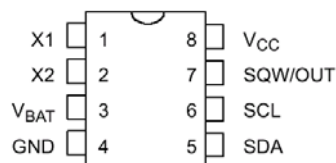
รูปที่ 8 แสดงขั้วต่อใช้งานของ RS422/485

## RTC (DS1307)

### คุณสมบัติ

- เป็นนาฬิกาที่สามารถให้ข้อมูลออกมาเป็น วินาที นาที ชั่วโมง วันที่ของเดือน เดือน วันของสัปดาห์ ปี ซึ่งขีดเขยค่าแล้ว สามารถใช้งานได้ถึงปี 2100
- มี RAM แบบ Nonvolatile จำนวน 56 ไบต์
- สามารถตั้งโปรแกรมให้ส่งสัญญาณรูปสี่เหลี่ยมออกที่เอาต์พุตได้
- สามารถตรวจสอบระบบไฟและสลับไปใช้แบตเตอรี่ได้โดยอัตโนมัติ
- เมื่อใช้พลังงานจากแบตเตอรี่จะใช้ไฟน้อยกว่า 500 nA ที่ 25 C
- สามารถเลือกใช้รุ่นที่ใช้งานในอุตสาหกรรมได้โดยสามารถใช้อุณหภูมิได้ในช่วง -40 ถึง +85

### รายละเอียดของขาต่อใช้งาน



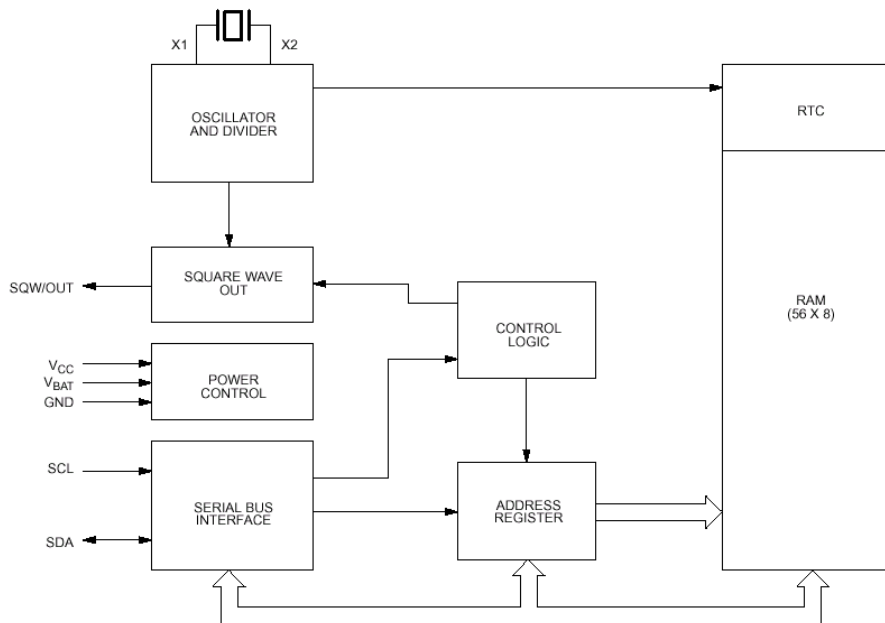
รูปที่ 9 แสดงขาต่อใช้งานของ DS1307

### รายละเอียดของขาสัญญาณ

ตำแหน่ง	สัญลักษณ์	รายละเอียด
1	X1	ขาต่อ Xtal 32.768 KHz
2	X2	ขาต่อ Xtal 32.768 KHz
3	VBAT	ขาต่อแบตเตอรี่ 3 V
4	GND	ขาต่อกราวด์
5	SDA	สัญญาณ Data แบบอนุกรม
6	SCL	สัญลักษณ์ Clock แบบอนุกรม
7	SQW/OUT	สัญญาณเอาต์พุตเป็นแบบสี่เหลี่ยมหรือเป็นลอจิก
8	Vcc	ต่อเข้ากับแหล่งจ่ายไฟ

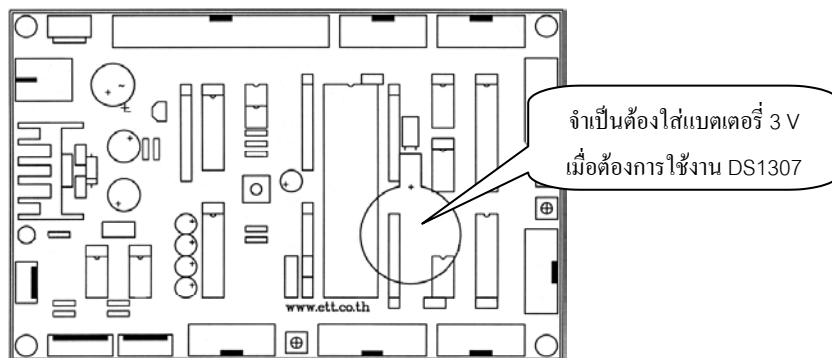
**การทำงาน**

DS1307 เป็นอุปกรณ์ที่ต่ออยู่ในระบบบัสแบบ I<sup>2</sup>C โดยทำตัวเป็น Slave การเข้าถึงข้อมูลภายในสามารถทำได้โดยการส่งเงื่อนไข Start เลขประจำตัวและตำแหน่งของรีจิสเตอร์ตามลำดับ ลงบนบัสแบบ I<sup>2</sup>C ที่มี DS1307 ต่อร่วมอยู่ โดยรีจิสเตอร์จะถูกเข้าถึงต่อไปเรื่อยๆ จนกระทั่งเกิดเงื่อนไข Stop ในระบบบัส เมื่อแรงดันที่ขา Vcc ตกลงต่ำกว่าแรงดันที่ขา Vbat จะสลับตัวเองเข้าสู่โหมดประหยัดพลังงานและใช้พลังงานจากแบตเตอรี่ แต่ในทางกลับกัน DS1307 จะกลับไปทำงานในโหมดปกติเมื่อแรงดันที่ขา Vcc สูงกว่าแรงดันที่ขา Vbat/+0.2 V โดยไดอะแกรมในรูปที่ 10 แสดงถึงส่วนประกอบหลักของ DS1307



รูปที่ 10 แสดงถึงส่วนประกอบหลักของ DS1307

บอร์ด CP-SPI/S8252 V2.0 & CP-SPI/RD2 V2.0 จะใช้ DS1307 ของ DALLAS เป็น RTC ซึ่งสามารถแสดงค่าเวลาได้เป็น ปี เดือน วันที่ของเดือน วันที่ของสัปดาห์ ชั่วโมง นาที และ วินาที โดยภายในยังมี RAM อีก 56 ไบต์ เพื่อใช้เก็บข้อมูล โดยทุกครั้งที่มีการใช้งาน DS1307 จะต้องใส่ BAT 3 V เข้าไปด้วย



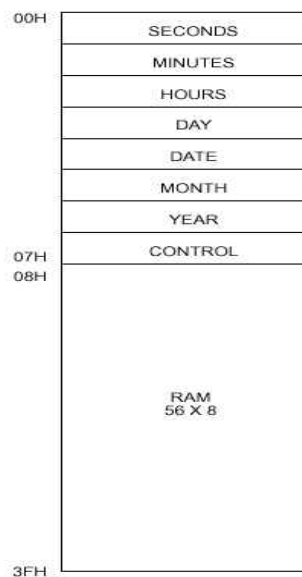
รูปที่ 11 แสดงตำแหน่งของการใส่แบตเตอรี่ เมื่อต้องการใช้งาน DS1307



DS1307 มีการสื่อสารข้อมูลกับไมโครคอนโทรลเลอร์โดยผ่านทาง I<sup>2</sup>C ซึ่งใช้สายสัญญาณเพื่อสื่อสาร 2 เส้นคือสายสัญญาณ DATA (SDA) และสายสัญญาณ CLOCK (SCL)

#### RTC And RAM Address Map

ตำแหน่งของนาฬิกาและ RAM ของ DS1307 แสดงในรูปที่ 12 โดยตำแหน่งของนาฬิกาอยู่ที่ 00H ถึง 07H โดยตำแหน่งของ RAM อยู่ที่ตำแหน่ง 08H ถึง 3FH เมื่อมีการเข้าถึงหน่วยความจำแบบ Multibyte ถ้าการเข้าถึงหน่วยความจำถึง 3FH และจะกลับไปอยู่ที่ 00 H ซึ่งเป็นตำแหน่งเริ่มต้น



รูปที่ 12 แสดงตำแหน่งของนาฬิกาและ RAM ของ DS1307

#### DS1307 Timekeeping Register

##### Control Register

รีจิสเตอร์ควบคุมเป็นส่วนใหญ่ที่ใช้ควบคุมขาสัญญาณ SQW/OUT ถ้าบิต 4 หรือบิต SQWE มีลอจิก 1 จะทำให้ขา SQW/OUT สร้างสัญญาณพัลส์ออกมาตามความถี่ที่กำหนดในบิต RS0 และ RS1 แต่ถ้าบิต 4 มีลอจิก 0 จะทำให้ขา มีลอจิกตามบิต 7 หรือบิต OUT

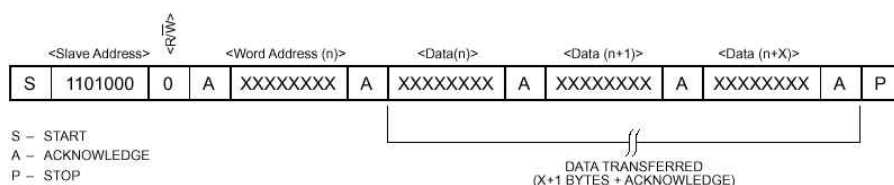
##### Squarewave Output Frequency

การกำหนดค่าความถี่ที่ต้องการให้ออกที่ขา SQWE มีลอจิก 1 สามารถกำหนดได้จากบิต RS0 และ RS1 ตามตารางข้างบน

### การเขียนข้อมูลลง DS1307 (Data Write)

การเขียนข้อมูลลงใน DS1307 จะถูกนำมาใช้เมื่อต้องการตั้งเวลา การกำหนดให้สัญญาณ Pulse ออกที่ขา SQW/OUT หรือแม้กระทั่งการเขียนเข้าไปเก็บไว้ในหน่วยความจำส่วน RAM ที่อยู่ภายใน DS1307

แนวทางการเขียนข้อมูลเข้าใน DS1307 นั้นจะใช้หลักการของการสื่อสารข้อมูลแบบ I<sup>2</sup>C คือการเขียนจะต้องเริ่มต้นจากที่ไมโครคอนโทรลเลอร์ส่งเงื่อนไข Start ไปให้กับ DS1307 แล้วจึงส่งข้อมูลต่อไปอีก 1 ไบต์ โดยข้อมูลไบต์นี้ใช้เก็บ Address และบิต R/W



รูปที่ 13 แสดง Diagram การเขียนข้อมูลลง DS1307

DS1307 ถูกกำหนดให้มีตำแหน่ง Address อยู่ที่ 110100B (ขนาด 7 บิต) ซึ่งในการเขียนข้อมูลจะต้องกำหนดให้บิต R/W เป็น 0 เพราะฉะนั้นเมื่อรวม Address ขนาด 7 บิต เข้ากับบิต R/W จะได้ข้อมูลไบต์ที่จะต้องส่งเป็น 11010000B

หลังจากที่ส่งไบต์ Address ไปให้กับ DS1307 แล้วถ้า DS1307 รับข้อมูลได้ถูกต้อง ตัว DS1307 จะส่งบิต ACK ออกมาให้กับไมโครคอนโทรลเลอร์ ซึ่งบิต ACK ที่ DS1307 ส่งออกมาให้กับไมโครคอนโทรลเลอร์จะเป็น 0 แต่ถ้าไมโครคอนโทรลเลอร์ตรวจสอบบิตนี้แล้วเป็น 1 แสดงว่า DS1307 ไม่ได้รับตำแหน่ง Address ที่ถูกต้อง ต้องกลับไปเริ่มขบวนการ Start ใหม่

หลังจากที่ไมโครคอนโทรลเลอร์ได้รับบิต ACK ตอบกลับออกมาจาก DS1307 แล้ว ตัวไมโครคอนโทรลเลอร์จะต้องส่งข้อมูลไปอีก 1 ไบต์ โดยข้อมูลไบต์นี้จะเป็นตำแหน่งหน่วยความจำภายใน DS1307 ซึ่งอยู่ที่ตำแหน่ง 0-3F การส่งข้อมูลไบต์นี้จะเหมือนกับการส่งข้อมูลในไบต์แรก คือเมื่อ DS1307 ได้รับข้อมูลถูกต้องแล้ว ตัว DS1307 จะส่งบิต ACK กลับออกมา

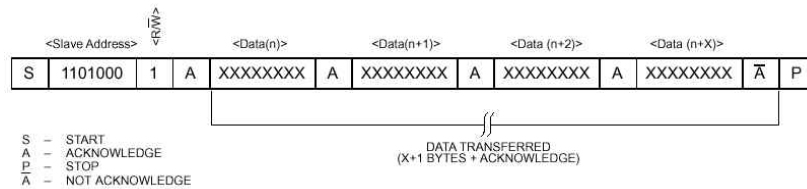
หลังจากที่ DS1307 ได้รับตำแหน่งของหน่วยความจำที่ต้องการเขียนเรียบร้อยแล้ว ไมโครคอนโทรลเลอร์สามารถส่งข้อมูลที่ต้องการเขียนไปยังตำแหน่งดังกล่าวได้เลย ซึ่งเมื่อส่งข้อมูลได้ออกไป 1 ไบต์ ตัว DS1307 จะเพิ่มค่าตำแหน่ง 1 ค่า และส่งบิต ACK กลับออกมา

ถ้าต้องการเขียนข้อมูลในตำแหน่งที่เรียงกันไป สามารถทำได้โดยส่งข้อมูลต่อไปเรื่อยๆ จนครบแล้วจึงส่งเงื่อนไข STOP เพื่อหยุดการทำงาน

แต่ถ้าหากต้องการเปลี่ยนตำแหน่งที่จะติดต่อหรือไม่ทราบว่าจะใช้ตำแหน่งของ DS1307 ซึ่งอยู่ที่ตำแหน่งใด ก็สามารถทำได้โดยการส่งตำแหน่งใหม่ให้กับ DS1307

### การอ่านข้อมูลจาก DS1307 (Data Read)

การอ่านข้อมูลจาก DS1307 จะมีลักษณะคล้ายกับการเขียนข้อมูลให้กับ DS1307 โดยจะต้องให้บิต



R/W เป็น 1 เพราะฉะนั้นข้อมูลไบต์ Address จะมีค่าเป็น 11010001B

รูปที่ 14 แสดง Diagram การอ่านข้อมูลออกจาก DS1307

### I<sup>2</sup>C

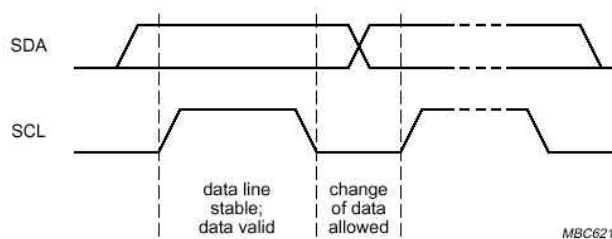
การสื่อสารข้อมูลด้วยระบบ I<sup>2</sup>C บัสนี้เป็นอีกมิติหนึ่งของการสื่อสารระหว่างไมโครคอนโทรลเลอร์และอุปกรณ์ประกอบรวม ระบบ I<sup>2</sup>C เป็นระบบที่ถูกพัฒนาจากบริษัท PHILIPS ดังนั้นอุปกรณ์หลายๆตัวที่มีการสื่อสารแบบ I<sup>2</sup>C จึงถูกผลิตออกมาจากบริษัท PHILIPS

ระบบ I<sup>2</sup>C บัสนี้เป็นการสื่อสารแบบ 2 ทาง โดยใช้สายสัญญาณในการสื่อสารเพียง 2 เส้น โดยสายที่ใช้สื่อสารนี้ คือ SDA ซึ่งเป็นสายสัญญาณข้อมูล และ SCL ซึ่งเป็นสายสัญญาณ Clock โดยสัญญาณทั้ง 2 เส้นนี้สามารถต่อเข้ากับอุปกรณ์ได้มากกว่า 1 ตัว ซึ่งทำให้การใช้งานไมโครคอนโทรลเลอร์มีประสิทธิภาพมากในแง่ของความสัมพันธ์พอร์ตของไมโครคอนโทรลเลอร์

ทางบริษัทผู้พัฒนาระบบ I<sup>2</sup>C ได้ให้คำจำกัดความของการสื่อสารแบบ I<sup>2</sup>C ไว้เพื่อให้นักศึกษาเกิดความเข้าใจเดียวกันโดยมีรายละเอียดต่างๆดังนี้

#### Bit Transfer

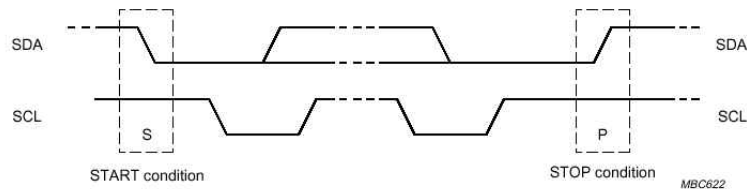
ข้อมูล 1 บิตจะถูกส่งออกไปด้วยช่วงเวลา 1 CLOCK โดยข้อมูลที่สาย SDA จะต้องคงที่ในขณะที่ CLOCK เป็นลอจิก 1



รูปที่ 15 แสดง Timing Diagram ของ Bit Transfer

### Start and Stop Conditions

ทั้งสายสัญญาณ SDA และสายสัญญาณ SCL ถ้าอยู่ในสถานะไม่ BUSY จะเป็นลอจิก 1 การเปลี่ยนแปลงจากลอจิก 1 เป็นลอจิก 0 ของสายสัญญาณ SDA ขณะที่สายสัญญาณ SCL เป็น 1 เรียกว่าการกำหนดเงื่อนไข START แต่การเปลี่ยนแปลงจาก 0 เป็น 1 ของสายสัญญาณ SDA ในขณะที่สายสัญญาณ SCL เป็น 1 เรียกว่าการกำหนดเงื่อนไข STOP



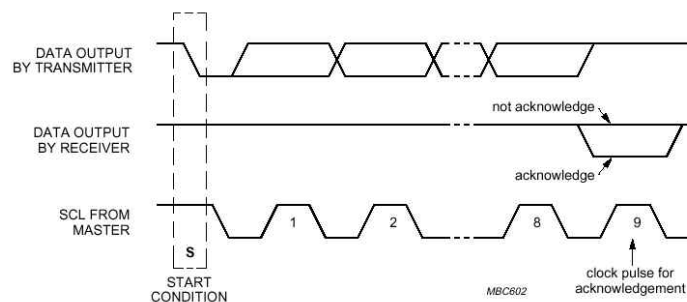
รูปที่ 16 แสดง Timing Diagram ของ Start and Stop Conditions

### System configuration

อุปกรณ์ที่ส่งข้อมูลเรียกว่า TRANSMITTER ส่วนอุปกรณ์ที่รับข้อมูลเรียกว่า RECEIVER และอุปกรณ์ที่ใช้ควบคุมทิศทางการสื่อสารข้อมูลเรียกว่า MASTER และอุปกรณ์ที่ถูกควบคุมจาก MASTER เรียกว่า SLAVE

### Acknowledge

จำนวนไบต์ของข้อมูลที่ถูกส่งระหว่างตัวรับและตัวส่งมิได้ไม่จำกัด ซึ่งเมื่อส่งข้อมูลครบ 1 ไบต์จะต้องส่งบิต ACK ตามออกไป 1 บิต

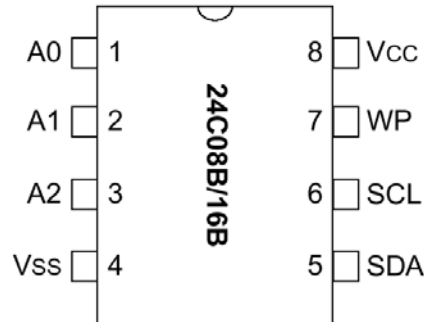


รูปที่ 17 แสดง Timing Diagram ของ Acknowledge

### EEPROM 24XX Family

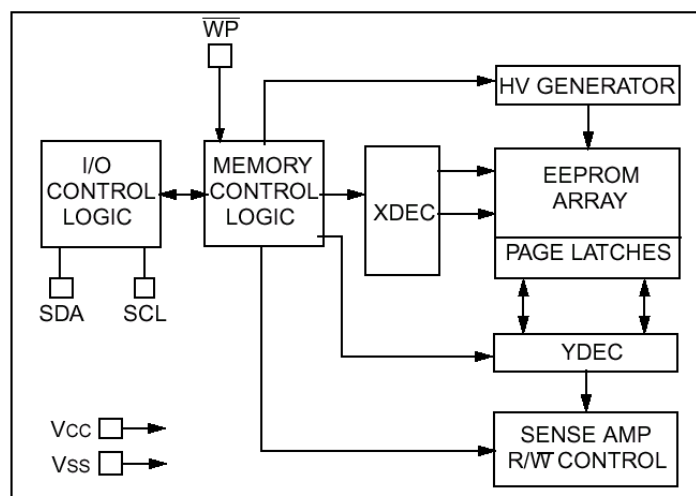
บอร์ด CP-SPI/S8252 V2.0 & CP-SPI/RD2 V2.0 ออกแบบให้ใช้ IC EEPROM ในตระกูล 24XX ได้ เช่น 24C16 , 24C128 , 24C256 โดยจะยกตัวอย่างอธิบายการใช้งานเฉพาะ IC 24C16 เท่านั้น IC 24C16 เป็น EEPROM ที่มีขนาด 2 Kbyte ที่มีการสื่อสารแบบ I<sup>2</sup>C สามารถต่อร่วมกับอุปกรณ์แบบ I<sup>2</sup>C

ตัวอื่น ๆ ได้ในสายสัญญาณคู่เดียวกัน แต่ไม่สามารถต่อ 24C16 ในสายสัญญาณคู่เดียวกัน เพราะ 24C16 มีตำแหน่งที่คงที่เพียงตำแหน่งเดียว คือ ตำแหน่ง 00H โครงสร้างภายนอกและขาสัญญาณแสดงดังรูปด้านล่าง



รูปที่ 18 แสดงโครงสร้างขาใช้งานของ EEPROM 24C16

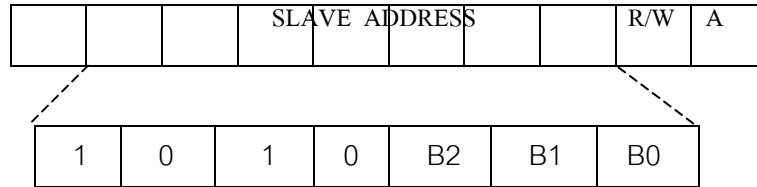
ตำแหน่ง	สัญลักษณ์	รายละเอียด
1	A0	ขากำหนดตำแหน่ง A0
2	A1	ขากำหนดตำแหน่ง A1
3	A2	ขากำหนดตำแหน่ง A2
4	Vss	กราวด์
5	SDA	สัญญาณข้อมูล I <sup>2</sup> C
6	SCL	สัญญาณนาฬิกา
7	WP	ขาป้องกันการเขียน
8	Vcc	แหล่งจ่ายไฟ 5 โวลต์



รูปที่ 19 แสดงโครงสร้างภายในของ EEPROM 24C16

**Device Addressing**

Control byte จะเป็น Byte แรกต่อบิต START Control Byte จะประกอบด้วย Control Code 4 bit จะอยู่ในบิตที่ 7-4 ใน 24C16 มีค่าเป็น 1010 และต่อมาอีก 3 Bit จะเป็น (B2,B1,B0) เป็นบิตที่ใช้เลือก Bank ของหน่วยความจำ บิตที่ 0 เป็นบิต R/W มีหน้าที่กำหนดเงื่อนไขว่าจะเขียนหรืออ่านข้อมูลจาก 24C16



รูปที่ 20 แสดงลักษณะการ Device Address

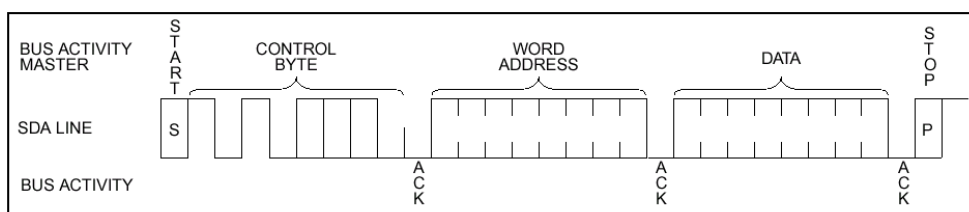
**การเขียนข้อมูล**

การเขียนข้อมูลลงในหน่วยความจำของ 24C16 สามารถทำได้โดยเริ่มจากส่งเงื่อนไข Start และตามด้วย Control Byte บิต R/W ใน Control Byte จะต้องกำหนดค่าให้เป็น 0 ซึ่งการเขียนข้อมูลลงในหน่วยความจำของ 24C16 สามารถแบ่งออกได้เป็นการเขียนข้อมูลครั้งละ 1 ไบต์ และการเขียนข้อมูลครั้งละ 1 Page

**การเขียนข้อมูลแบบ Byte**

การเขียนข้อมูลแบบไบต์ หรือการเขียนข้อมูลครั้งละ 1 ไบต์ลงในหน่วยความจำของ 24C16 สามารถทำได้โดยเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ส่งเงื่อนไข Start ให้กับ 24C16 และตามด้วย Control Byte โดยกำหนดค่า B2,B1,B0 ใน Control Byte ตามค่าตำแหน่งหน่วยความจำที่ต้องการเขียนข้อมูล ส่วนบิต R/W จะต้องถูกกำหนดให้เป็น 0 หลังจากส่ง Control Byte จะต้องรอรับค่าบิต ACK จาก 24C16 ตามกลับออกมา ซึ่งบิต ACK ที่ตอบกลับออกมาจะมีค่าเป็น 0 หลังจากได้รับบิต ACK ที่ตอบกลับออกมาแล้ว ให้ส่งไบต์ตำแหน่งของหน่วยความจำที่ต้องการเขียนข้อมูลออกไปอีก 1 ไบต์ หลังจากนั้นให้รอรับบิต ACK ที่จะตอบกลับออกมาเป็นลอจิก 0 จาก 24C16 เมื่อได้รับบิต ACK ตอบกลับออกมาแล้ว ให้ส่งค่าข้อมูลที่ต้องการเขียนลงในตำแหน่งหน่วยความจำออกไป และรอรับค่าบิต ACK จาก 24C16 หลังจากได้รับบิต ACK ตอบกลับออกมาแล้วให้ให้ส่งเงื่อนไขการ Stop เพื่อยกเลิกการติดต่อ

ถ้าขั้นตอนใดขั้นตอนหนึ่งไม่มีบิต ACK ตอบกลับออกมาให้เริ่มส่งเงื่อนไข Start ใหม่อีกครั้งและ



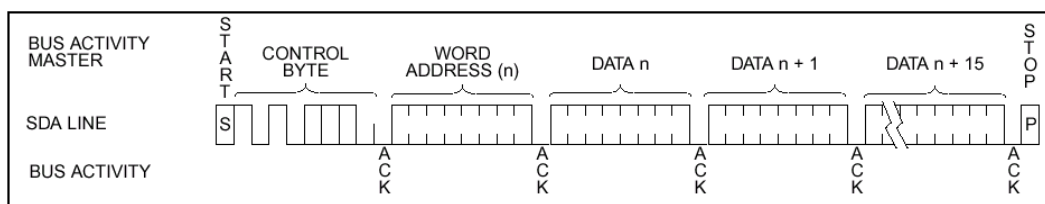
รูปที่ 21 แสดง Timing Diagram ของการเขียนข้อมูลแบบ Byte

เริ่มทำขบวนการตั้งแต่นั้นใหม่

### การเขียนข้อมูลแบบ Page

การเขียนข้อมูลแบบ Page หรือการเขียนข้อมูลครั้งละ 1 Page ภายใน 24C16 แบ่งการเก็บข้อมูลไว้เป็น Page โดยแต่ละ Page สามารถเก็บข้อมูลได้ 16 ไบต์ ดังนั้นการเขียนข้อมูลครั้งละ Page ก็คือการเขียนข้อมูลครั้งละ 16 ไบต์นั่นเอง

การเขียนข้อมูลครั้งละ Page จะเหมือนกับการเขียนข้อมูลครั้งละ 1 ไบต์ จะต้องส่งเงื่อนไข Start ออกไปก่อน แล้วตามด้วย Control Byte หลังจากนั้นรอรับ ACK จาก 24C16 หลังจากได้ค่า ACK แล้วให้ส่งตำแหน่งหน่วยความจำออกไปและรอรับค่าบิต ACK ที่ตอบกลับออกมาแล้วให้ส่งค่าข้อมูลที่ต้องการเขียนลงในตำแหน่งหน่วยความจำออกไป และรอรับค่าบิต ACK จาก 24C16 จึงจะส่งข้อมูลไบต์ต่อไปจนครบ 16 ไบต์ โดยข้อมูลทั้ง 16 ไบต์ จะถูกเก็บเรียงกันไปหน่วยความจำเริ่มตั้งแต่ที่ตำแหน่งที่ถูกส่งออกไปใน Byte Word Address หลังจากที่ได้รับบิต ACK หลังจากส่งข้อมูลออกไปครบทั้ง 16 ไบต์ ให้ส่งบิต Stop เพื่อยกเลิกการเขียนข้อมูล



รูปที่ 22 แสดง Timing Diagram ของการอ่านข้อมูลแบบ Page

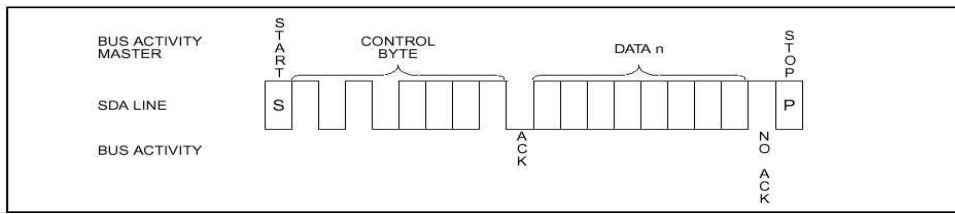
### การอ่านข้อมูล

ขั้นตอนการอ่านข้อมูลจะเหมือนกับขั้นตอนการเขียนข้อมูลจะแตกต่างกันที่บิต RW จะต้องถูกเซ็ทเป็น 1 ซึ่งสามารถแบ่งลักษณะการอ่านข้อมูลได้ 3 แบบ

#### การอ่านข้อมูลในตำแหน่งปัจจุบัน (Current Address Read)

ภายใน 24C16 จะมีตัวชี้ตำแหน่งของหน่วยความจำ ทุกครั้งที่เราส่งตำแหน่งให้กับ 24C16 นั้นค่าของตัวชี้จะถูกเปลี่ยนตามค่าของตำแหน่งที่ถูกส่งเข้าไป และทุกครั้งที่มีการเขียนเข้าไปใน 24C16 หรืออ่านข้อมูลออกจาก 24C16 ในแต่ละไบต์ค่าของตัวชี้จะเพิ่มขึ้นเป็น 1 ค่าโดยอัตโนมัติ

ดังนั้นการอ่านข้อมูลในตำแหน่งปัจจุบันคือการอ่านข้อมูลจาก 24C16 ณ ตำแหน่งที่ตัวชี้อยู่ ซึ่งสามารถทำได้โดยการส่งเงื่อนไข Start และตามด้วย Control Byte โดยกำหนดให้บิต RW มีค่าเป็น 1 หลังจากนั้นให้รอรับบิต ACK หลังจากที่ได้บิต ACK แล้วให้อ่านข้อมูลกลับออกมาจาก 24C16 จำนวน 1 ไบต์ หลังจากนั้นให้ส่งบิต NO ACK ซึ่งก็คือส่งลอจิก 1 และส่งเงื่อนไข Stop ให้กับ 24C16

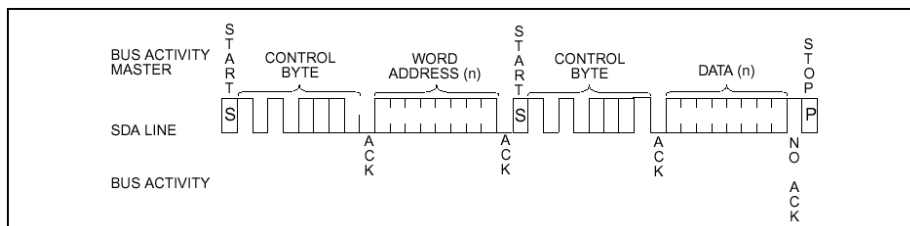


รูปที่ 23 การอ่านข้อมูลในตำแหน่งปัจจุบัน

**การอ่านข้อมูลแบบสุ่ม (Random Read)**

การอ่านข้อมูลแบบสุ่มจะเป็นการอ่านค่าตาม Address ที่ระบุที่ตำแหน่งที่ต้องการอ่าน โดยมีขั้นตอนการอ่านค่าข้อมูลในหน่วยความจำดังนี้

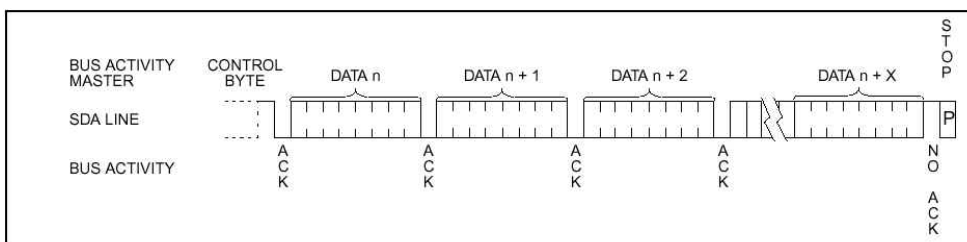
ส่งเงื่อนไข Start และตามด้วย Control Byte โดยให้บิต RW เป็น 0 และ Word Address ซึ่งเป็นตำแหน่งของหน่วยความจำที่ต้องการอ่านข้อมูล หลังจากนั้นให้รอรับบิต ACK เมื่อได้รับบิต ACK แล้วจึงอ่านข้อมูลออกมา 1 ไบต์ ซึ่งเป็นข้อมูลตำแหน่งเดียวกับที่ตัวชี้ชี้อยู่ หลังจากได้ข้อมูลครบทั้ง 8 บิตแล้ว ให้ส่งบิต NO ACK และบิต Stop ให้กับ 24C16 เพื่อยกเลิกการติดต่อ



รูปที่ 24 แสดง Timing Diagram ของการอ่านข้อมูลแบบสุ่ม

**การอ่านข้อมูลเป็นลำดับ (Sequential Read)**

การอ่านข้อมูลในหน่วยความจำแบบลำดับ มีขั้นตอนการอ่านข้อมูลในลำดับเดียวกันกับการอ่านข้อมูลแบบสุ่ม จะแตกต่างที่การอ่านข้อมูลแบบสุ่มจะอ่านข้อมูลออกมาเพียง ไบต์เดียว แต่การอ่านข้อมูลแบบลำดับจะอ่านข้อมูลออกมา X จนกระทั่งส่งบิต NO ACK และบิต Stop ออกไป ซึ่งทุกครั้งที่มีการอ่านข้อมูลออกจาก 24C16 นั้น ค่าของตัวชี้ที่อยู่ใน 24C16 จะเพิ่มค่าขึ้น 1 ค่า เพราะฉะนั้นข้อมูลทีอ่านค่าออกมาได้จึงเป็นข้อมูลที่อยู่ต่อกันเป็นลำดับต่อ ๆ กันออกไป



รูปที่ 25 แสดง Timing Diagram ของการอ่านข้อมูลแบบลำดับ

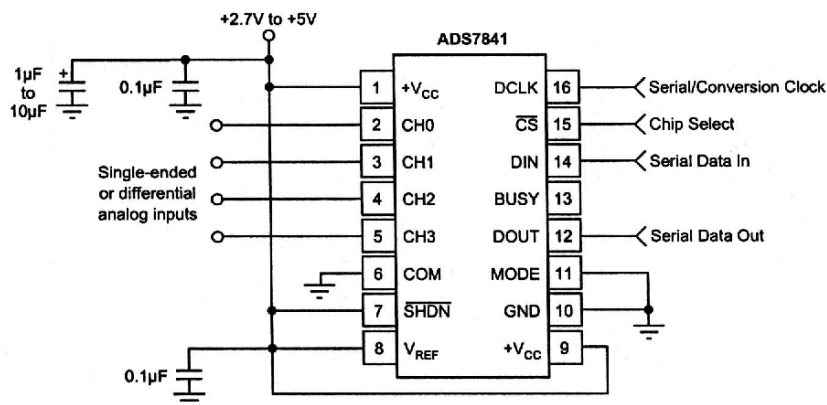


## ADS7841

### ทฤษฎีของการทำงาน

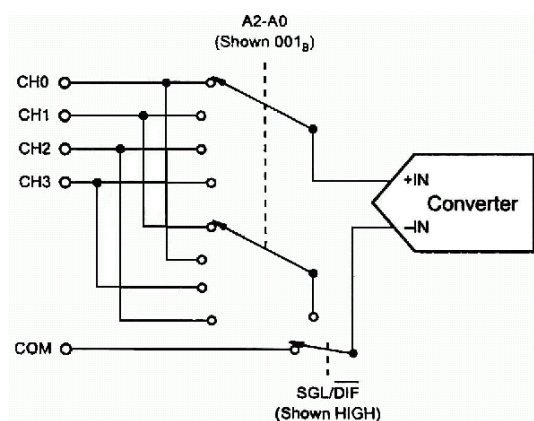
ADS7841 เป็นตัวแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC) โดยใช้วิธีการ Successive Approximation Resister (SAR) โดยโครงสร้างอยู่บนพื้นฐานของ Capacitive Redistribution ซึ่งได้รวมวงจรสุ่มและคงค่าสัญญาณ (Sampling/Hold) ไว้ภายใน

โครงสร้างการทำงานแสดงดังรูปที่ 26 โดย ADS7841 ต้องการแรงดันอ้างอิงและสัญญาณนาฬิกาจากภายนอก สามารถทำงานโดยใช้แหล่งจ่ายไฟเพียงแหล่งเดียวในช่วง 2.7V – 5.25V โดยแรงดันอ้างอิงที่จะต่อภายนอกจะอยู่ในช่วง 100mV ถึง +Vcc โดยค่าของแรงดันอ้างอิงจะเป็นตัวกำหนดช่วงของแรงดันอินพุตของ ADS7841 โดยตรง ค่ากระแสเฉลี่ยของแรงดันอ้างอิงขึ้นอยู่กับอัตราการแปลงสัญญาณของ ADS7841



รูปที่ 26 แสดงโครงสร้างการทำงานของ ADS7841

### แรงดันอินพุต (ANALOG INPUT)



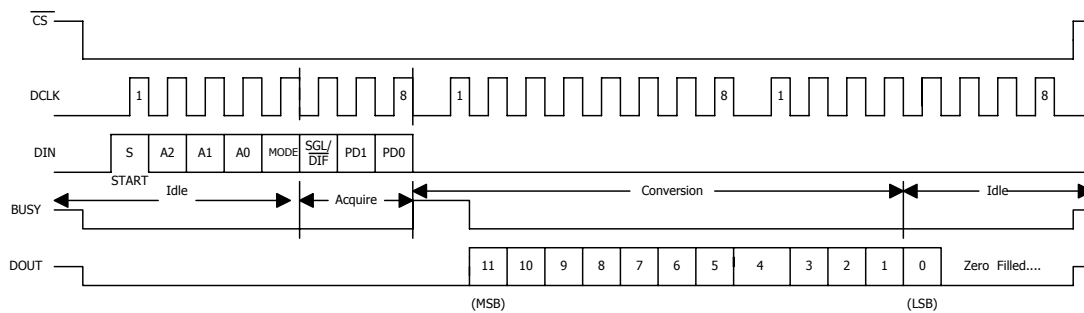
รูปที่ 27 แสดงไดอะแกรมของวงจรสลับสัญญาณอินพุตของ ADS7841

**แรงดันอ้างอิง (REFERENCE INPUT)**

แรงดันอ้างอิงที่ป้อนให้กับ ADS7841 จะเป็นตัวกำหนดระดับแรงดันอินพุตสูงสุดทั้ง 4 ช่องสัญญาณ โดยระดับแรงดันอินพุตสามารถกำหนดได้อยู่ในช่วง 100mV ถึง +Vcc สัญญาณอินพุตคือระดับความแตกต่างของแรงดันที่ขา + IN และ -IN ดังแสดงในรูปที่ 27 ตัวอย่างเช่น กำหนดให้ ADC ทำงานในโหมด Single-Ended โดยกำหนดให้แรงดันอ้างอิงอยู่ที่ 1.25V และขาสัญญาณ COM ของ ADC ถูกต่อลงกราวด์ ดังนั้นสัญญาณอินพุต (CH0-CH3) จะสามารถถูกแปลงเป็นดิจิตอลได้เฉพาะแรงดันในช่วง 0V ถึง 1.25V แต่ถ้าสัญญาณ COM ถูกป้อนด้วยแรงดัน 0.5V จะทำให้แรงดันอินพุตของแต่ละช่องสัญญาณอยู่ในช่วง 0.5V ถึง 1.75V

**การเชื่อมต่อสัญญาณดิจิตอล (DIGITAL INTERFACE)**

รูปที่ 28 แสดงการเชื่อมต่อสัญญาณดิจิตอลของ ADS7841 โดยไดอะแกรมนี้สมมุติให้ ADC เชื่อมต่อกับไมโครคอนโทรลเลอร์หรือ DSP โดยเชื่อมต่อแบบอนุกรม แต่การสื่อสารระหว่างไมโครคอนโทรลเลอร์กับ ADC จะต้องใช้สัญญาณนาฬิกา 8 คาบ โดยที่การแปลงสัญญาณ 1 ครั้งจะต้องใช้ 3 การสื่อสาร ดังนั้นหมายความว่าต้องใช้สัญญาณนาฬิกา 24 คาบ ให้กับขาสัญญาณ DCLK



รูปที่ 28 ไดอะแกรมแสดง 24 Clock ต่อการแปลงกลับ

**ไบนารีควบคุม (Control Byte)**

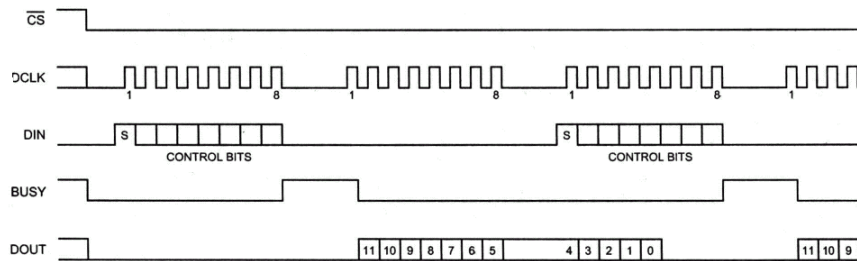
ตารางด้านล่าง แสดงการควบคุมบิต โดยบิตแรกคือบิต 'S' จะเป็นลอจิก 1 เพื่อแสดงถึงจุดเริ่มต้นของไบนารีควบคุม ซึ่ง ADS7841 จะไม่สนใจสัญญาณที่ขา DIN จนกระทั่งมันสามารถตรวจสอบบิตเริ่มต้นได้ และสามบิตต่อมา (A2,A1,A0) เป็นบิตที่ใช้เลือกช่องสัญญาณ

Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
S	A2	A1	A0	MODE	SGL/DIF	PD1	PD0

ตาราง แสดงการควบคุมบิต

### 16-Clock per Conversion

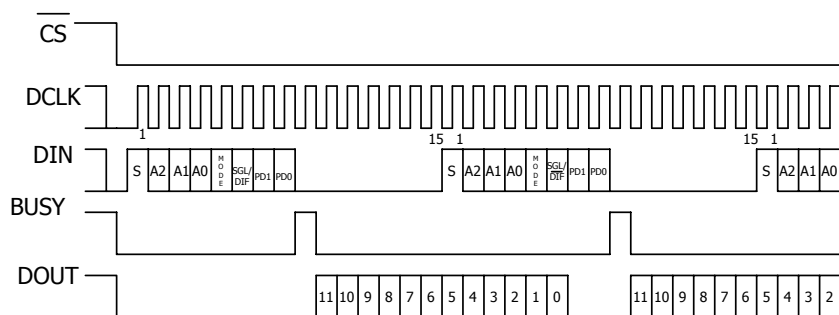
บิตควบคุมของการแปลงครั้งที่  $n+1$  จะซ้อนทับการแปลงครั้งที่  $n$  เพื่อให้เกิดการแปลงสัญญาณทุกๆ 16 คาบ ดังที่แสดงในรูปที่ 29 ซึ่งรูปนี้แสดงถึงความเป็นไปได้ของการสื่อสารข้อมูลแบบอนุกรมระหว่างตัวประมวลผลกับ ADC ซึ่งวิธีการนี้สามารถแปลงสัญญาณได้ภายใน 1.6 ms



รูปที่ 29 Timing Diagram ของ 16 Clock per Conversion

### 15-Clocks per Conversion

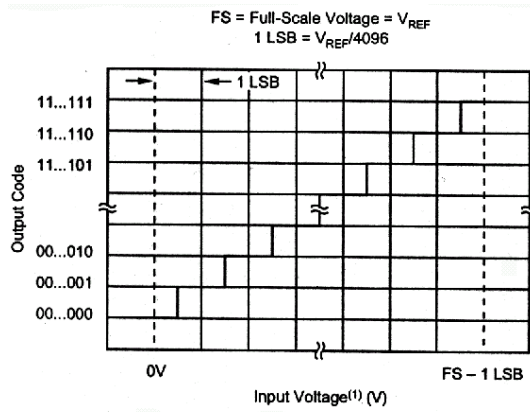
รูปที่ 30 แสดงวิธีการที่เร็วที่สุดในการกำหนดสัญญาณนาฬิกาให้กับ ADS7841 ซึ่งวิธีการนี้จะไม่สามารถทำงานได้กับไมโครคอนโทรลเลอร์หรือตัวประมวลผลสัญญาณดิจิทัล ซึ่งโดยทั่วไปไม่สามารถสื่อสารข้อมูลแบบอนุกรมด้วย 15-Clock ได้ อย่างไรก็ตามวิธีการนี้ถูกออกแบบไว้ให้ใช้งานกับ Gate Array (FPGA) หรือหรือวงจรรวมที่ถูกสร้างขึ้นมาเฉพาะงาน (ASICs) ซึ่งวิธีการนี้จะเพิ่มอัตราการแปลงสัญญาณได้สูงสุด



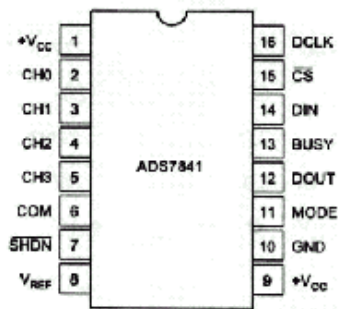
รูปที่ 30 อัตราการแปลงกลับสูงสุดในการกำหนดสัญญาณนาฬิกาให้กับ ADS7841

**แบบข้อมูล(Data Format)**

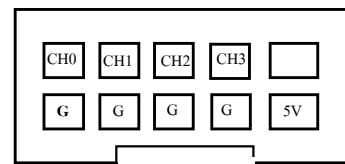
สัญญาณเอาต์พุตของ ADS7841 แบบไบนารีแสดงในรูปที่ 31 ซึ่งแสดงถึงค่าของเอาต์พุตที่ได้เมื่อค่าของแรงดันที่ระดับต่างๆ โดยไม่รวมถึงผลที่เกิดจาก offset และ ความผิดพลาดจากอัตราขยาย



รูปที่ 31 สัญญาณเอาต์พุตของ ADS7841

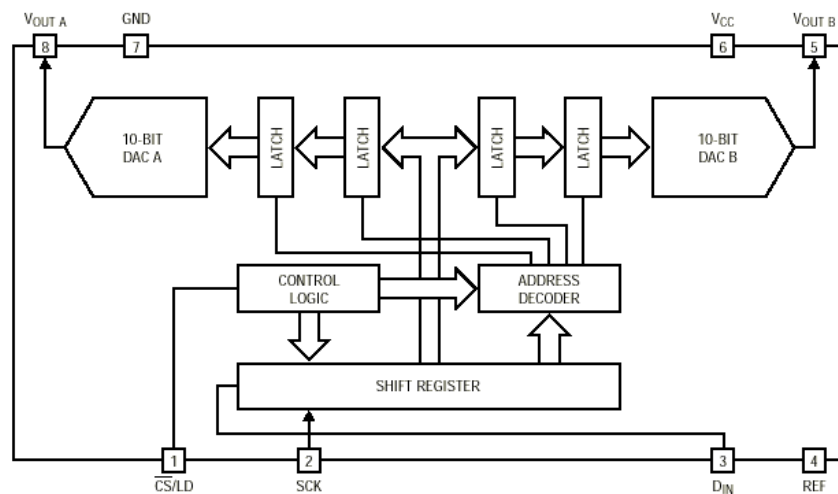


รูปที่ 32(ก) แสดงโครงสร้างของ ADS7841



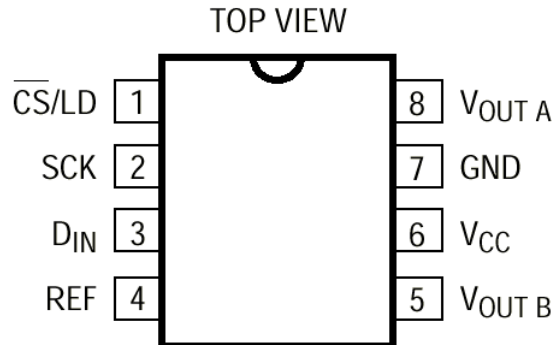
รูปที่ 32 (ข) แสดงขั้วต่อของ ADS 7841

**ไอซี DAC แบบอนุกรม (LTC1661)**



รูปที่ 33 แสดงโครงสร้างภายในของ LTC1661

ไอซี DAC เบอร์ LTC1661 จัดเป็น ไอซีจำพวก Chips Support ของบริษัท “Linear Technology” สามารถใช้สร้างสัญญาณ Analog ซึ่งมีความละเอียดสูงถึง 10บิต (1024ระดับ) ได้พร้อมกัน 2 ช่องสัญญาณในเวลาเดียวกัน



รูปที่ 34 แสดงตำแหน่งขาสัญญาณของ LTC1661

■  $\overline{\text{CS/LD}}$  เป็นสัญญาณ Chips Select ทำงานที่สภาวะลอจิกศูนย์ ใช้สำหรับเขียนค่า “Control Word” ให้กับ DAC โดยเมื่อสัญญาณ CS นี้มีสภาวะเป็นศูนย์ DAC จะพร้อมรับข้อมูลจากขา DIN ตามการควบคุมของสัญญาณ SCK แต่เมื่อสัญญาณ CS นี้มีสภาวะเป็นลอจิกหนึ่ง DAC จะไม่สนใจสภาวะของสัญญาณ SCK และ DIN เลย

■ SCK เป็นสัญญาณนาฬิกา (Clock Input) ใช้สำหรับทำหน้าที่ควบคุมการเลื่อนบิตข้อมูล(DIN) ของ “Control Word” ให้กับรีจิสเตอร์ของ DAC โดยทำงานที่ช่วงขอบขาขึ้น

■ DIN เป็นสัญญาณข้อมูล (Data Input) ใช้สำหรับบอกให้ DAC รับรู้คำสั่ง โดยปรกติแล้วข้อมูลที่ส่งให้กับ DAC จะมีขนาด 16บิต (1-Word) โดยสัญญาณข้อมูล(DIN) นี้จะถูกเลื่อนให้กับรีจิสเตอร์ของ DAC ครั้งละ 1 บิต ตามการควบคุมของสัญญาณนาฬิกา(SCK) โดยค่าของข้อมูลจะถูกเลื่อนเข้าไปยังรีจิสเตอร์ของ DAC ในช่วงขอบขาขึ้นของสัญญาณนาฬิกา ดังนั้นจะต้องทำการจัดเตรียมสถานะของสัญญาณข้อมูล(DIN) นี้เตรียมไว้ในช่วงที่สัญญาณนาฬิกามีสภาวะเป็น “0” เท่านั้น เมื่อสัญญาณนาฬิกาเปลี่ยนสถานะจาก “0” เป็น “1” (ขอบขาขึ้น) ค่าของบิตข้อมูล(DIN) ก็จะถูกเลื่อนไปยังรีจิสเตอร์ของ DAC ตามสถานะของสัญญาณข้อมูลที่เป็นอยู่ในขณะที่สัญญาณนาฬิกาเปลี่ยนจาก “0” เป็น “1” พอดี

■ REF เป็นขาสัญญาณแรงดันอ้างอิง ใช้สำหรับเปรียบเทียบการสร้างสัญญาณ Analog Output โดยต้องกำหนดค่าของแรงดันให้ขาสัญญาณ REF นี้มีค่าอยู่ระหว่าง 0V ถึง +VCC

■  $V_{\text{OUTA}}, V_{\text{OUTB}}$  เป็นขาสัญญาณ Analog Output ของไอซี DAC โดยขนาดของสัญญาณ Output จะแปรผันกับค่าของข้อมูลดิจิทัลและแรงดันอ้างอิงที่ป้อนให้กับขา REF ดังนี้

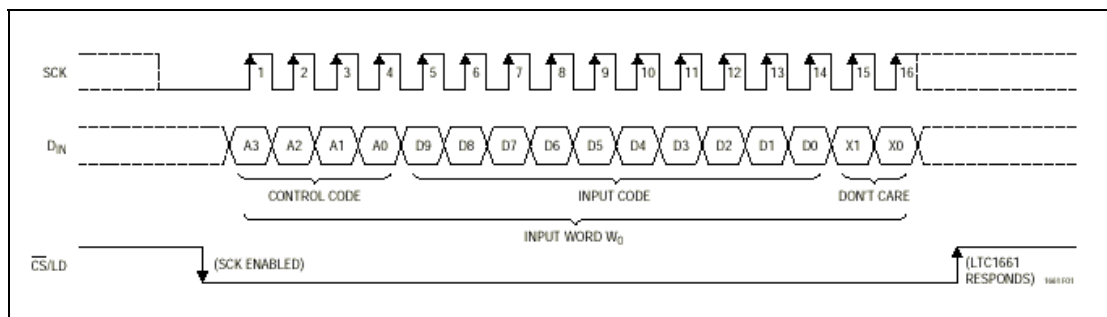
$$0 \leq V_{\text{OUTA}}, V_{\text{OUTB}} \leq V_{\text{REF}} (1023/1024)$$

- VCC เป็นขาแหล่งจ่ายไฟเลี้ยงให้กับตัวไอซี มีค่าระหว่าง  $2.7V \leq VCC \leq 5.5V$
- GND เป็นขา GND ของระบบ

สำหรับรหัส “Control Word” ของ DAC นั้นจะมีขนาด 16 บิต ประกอบด้วยค่า “Control Code” (A3..A0) จำนวน 4 บิต ตามด้วยค่าข้อมูล(D9..D0) จำนวน 10 บิต ส่วนที่เหลืออีก 2 บิต สุดท้ายนั้นไม่มีความหมายต่อการทำงานของ DAC อาจมีค่าเป็น “1” หรือ “0” ก็ได้ แต่จำเป็นต้องมี 2บิต นี้ไว้ เพื่อให้จำนวนบิตข้อมูลครบ 16 บิตพอดี โดยโครงสร้างของ Control Word มีดังนี้

ตำแหน่ง	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ชื่อบิต	A3	A2	A1	A0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X1	X0
หน้าที่	Control Code				Input Code (Data)										ไม่สนใจ	

- Control Code มีขนาด 4 บิต (A3..A0) ใช้สำหรับกำหนดหน้าที่การทำงานของ DAC
- Input Code มีขนาด 10 บิต (D9..D0) ใช้สำหรับกำหนดระดับของสัญญาณ Analog Output ของ DAC มีค่าระหว่าง 000H-3FFH(0-1023 ระดับ) โดยที่ระดับของสัญญาณ Analog Output นั้นจะมีค่าสัมพันธ์กับ Input Code นี้คือ  $V_{OUT} = (Input\ Code \div 1024) \times V_{REF}$
- X1 และ X0 เป็น บิตข้อมูล จำนวน 2บิต ซึ่งไม่มีความหมายใดๆต่อการทำงานของ DAC แต่ต้องกำหนดไว้เพื่อให้จำนวนบิตข้อมูลของ Control Word ครบสมบูรณ์



รูปที่ 35 แสดง Timing Diagram ของความสัมพันธ์ระหว่างสัญญาณ CS,SCK และ DIN

โดยในการส่ง “Control Word” ให้กับ DAC นั้นจะต้องเริ่มต้นด้วยการจัดเตรียมสัญญาณ CS ให้มีค่าเป็น “1” ส่วน SCK ให้มีค่าเป็น “0” รอไว้ก่อน จากนั้นจึงกำหนดให้ CS มีค่าเป็น “0” เพื่อเริ่มต้นการส่งข้อมูลให้กับ DAC แล้วจึงกำหนดค่าสถานะของ DIN ด้วยค่า Control Word บิตนัยสำคัญสูงสุด(A3)ก่อนเป็นอันดับแรก จากนั้นจึงทำการเปลี่ยนสถานะของ SCK จาก “0” เป็น “1” เพื่อเลื่อนบิตข้อมูลจาก DIN(A3) ให้

กับ DAC แล้วจึงทำการเปลี่ยนสถานะของ SCK จาก “1” เป็น “0” เพื่อทำการจัดเตรียมบิตข้อมูลถัดไป(A2)ให้กับขา DIN โดยจะวนทำอย่างนี้ไปเรื่อยๆจนครบ 16 บิต แล้วจึงทำให้ CS กลับมามีค่าเป็น “1” เพื่อเป็นการสิ้นสุดการส่ง Control Word ให้กับ DAC ซึ่งจะเห็นได้ว่าเราต้องทำการเปลี่ยนแปลงสถานะของบิตข้อมูลให้กับ DIN ในช่วงที่สัญญาณ SCK มีค่าเป็น “0” และ ทำการเลื่อนบิตข้อมูลให้กับ DAC ในช่วงขอบขาขึ้นของสัญญาณ SCK (“0” เป็น “1”)

Control Code	Input Register Status	DAC Register Status	Power Down Status	รายละเอียดการทำงานพอสั่งเขป (Comment)
0-0-0-0	ไม่เปลี่ยน	ไม่เปลี่ยน	ไม่เปลี่ยน	ไม่เปลี่ยนแปลงการทำงานใดๆของ DAC
0-0-0-1	Load DAC-A	ไม่เปลี่ยน	ไม่เปลี่ยน	โหลดค่า Data (D0..D9) ให้ รีจิสเตอร์ DAC-A
0-0-1-0	Load DAC-B	ไม่เปลี่ยน	ไม่เปลี่ยน	โหลดค่า Data (D0..D9) ให้รีจิสเตอร์ DAC-B
0-0-1-1		สงวนไว้		-
0-1-0-0		สงวนไว้		-
0-1-0-1		สงวนไว้		-
0-1-1-0		สงวนไว้		-
0-1-1-1		สงวนไว้		-
1-0-0-0	ไม่เปลี่ยน	สร้างสัญญาณ (Update Output)	เริ่มทำงาน (Wake)	DAC ออกจาก Sleep Mode และ สร้าง $V_{OUT-A,B}$ ด้วยค่าเดิมในรีจิสเตอร์ DAC-A,B
1-0-0-1	Load DAC-A	สร้างสัญญาณ Output Analog	เริ่มทำงาน (Wake)	DAC ออกจาก Sleep โหลดค่า Data ให้ DAC-A สร้าง $V_{OUT-A,B}$ ด้วยค่าในรีจิสเตอร์ DAC-A,B
1-0-1-0	Load DAC-B	สร้างสัญญาณ Output Analog	เริ่มทำงาน (Wake)	DAC ออกจาก Sleep โหลดค่า Data ให้ DAC-B สร้าง $V_{OUT-A,B}$ ด้วยค่าในรีจิสเตอร์ DAC-A,B
1-0-1-1		สงวนไว้		-

ตาราง แสดงหน้าที่ของ Control Code สำหรับสั่งงาน DAC

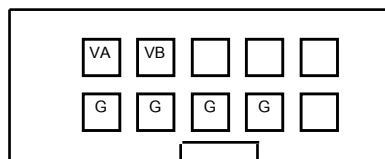
Control Code	Input Register Status	DAC Register Status	Power Down Status	รายละเอียดการทำงานพอสังเขป (Comment)
1-1-0-0		สงวนไว้		-
1-1-0-1	ไม่เปลี่ยน	ไม่เปลี่ยน	เริ่มทำงาน (Wake)	DAC ออกจาก Sleep Mode และ สร้าง $V_{OUT-A,B}$ ด้วยค่าเดิมในรีจิสเตอร์ DAC-A,B
1-1-1-0	ไม่เปลี่ยน	ไม่เปลี่ยน	หยุดทำงาน (Sleep)	DAC หยุดทำงาน (Sleep) ค่า DAC-A และ B ไม่เปลี่ยน ส่วน $V_{OUT}$ เป็น High Impedance
1-1-1-1	Load DAC-A และ DAC-B	สร้างสัญญาณ (Update Output)	เริ่มทำงาน (Wake)	Update DAC-A,B ด้วยค่า Data (D0..D9) และ สร้าง $V_{OUT-A,B}$ ด้วยค่าในรีจิสเตอร์ DAC ใหม่

#### ตาราง(ต่อ) แสดงหน้าที่ของ Control Code สำหรับสั่งงาน DAC

จะเห็นได้ว่า DAC นั้นจะมีโหมดการทำงานอยู่ 2 โหมดด้วยกันคือ Wake Mode และ Sleep Mode ซึ่งตามปรกติส่วนมากแล้วเราจะใช้งาน DAC กันใน Wake Mode

■ Wake Mode เป็นการทำงานตามปรกติ ส่วนของสัญญาณ Analog Output ยังคงมีค่าตามที่กำหนดไว้

■ Sleep Mode เป็นโหมดประหยัดพลังงาน ซึ่งการทำงานในโหมดนี้ DAC ต้องการพลังงานน้อยมาก คือประมาณ  $1\mu A$  ในการเก็บรักษาค่าของข้อมูลใน DAC Register ไว้ ส่วนสัญญาณ Analog Output ทั้ง 2 ช่อง จะมีสถานะเป็น “High Impedance” ซึ่งเราสามารถสั่งให้ DAC เข้าสู่การทำงานในโหมดนี้ได้ โดยการส่ง Control Word ที่มีรหัส Control Code เป็น “1-1-1-0” แต่เมื่อต้องการให้ DAC กลับมาทำงานตามปรกติใหม่ อีกครั้งหนึ่งสามารถทำได้โดยการส่ง Control Word ที่มีรหัส Control Code เป็น “1-0-0-0” ซึ่งสัญญาณ Analog Output ทั้ง 2 ช่องของ DAC จะกลับมามีค่าเหมือนกับก่อนเข้าทำงานใน Sleep Mode ทุกประการ



รูปที่ 36 แสดงให้เห็นถึงขั้วต่อใช้งานของ LTC1661



## LCD ( Character Mode)

หน่วยแสดงผลแบบ LCD ถูกออกแบบให้ใช้งานได้กับ LCD ชนิด Character ควบคุมการทำงานของจอ LCD แบบ 4 บิต เพื่อประหยัดพอร์ต ซึ่งตำแหน่งของพอร์ต LCD ขนาด 14 Pin ในบอร์ด จะแสดงได้ดังรูป โดยการต่อสายจะต้องพิจารณาเรื่องของตำแหน่งขาของพอร์ต LCD บนบอร์ด (ขาสัญญานบนบอร์ดจะสลับขาสัญญานที่อยู่บน LCD)

จากรูปจะพบว่าบนบอร์ด มีVR ที่ใช้ปรับความสว่าง (Contrast) ของ LCD ซึ่งในหัวข้อถัดไปจะอธิบายถึงการต่อสาย LCD เข้ากับบอร์ด แต่ก่อนที่จะอธิบายถึงสาย LCD จะขอกล่าวถึงการใช้งานของ LCD อย่างคร่าว ๆ ก่อน

โดยส่วนใหญ่แล้ว LCD ที่มีจำหน่ายในเมืองไทยจะมีลักษณะคล้ายกันไม่ว่าจะเป็นชุดคำสั่งในการควบคุมการทำงานของ LCD หรือคาบเวลาในการควบคุมการทำงานของ LCD ซึ่งโดยส่วนใหญ่แล้ว LCD ชนิด Character ไม่ว่าจะมีความละเอียดจะมีการควบคุมการทำงานเหมือนกันหมด LCD จะมีขาที่ต้องใช้ควบคุม 14 สัญญาน ยกเว้น LCD ที่มีไฟส่องหลัง (Back Light) จะมีสัญญาณเพิ่มขึ้นไปอีก 2 ขา คือขา VCC และGND ของไฟส่องหลัง

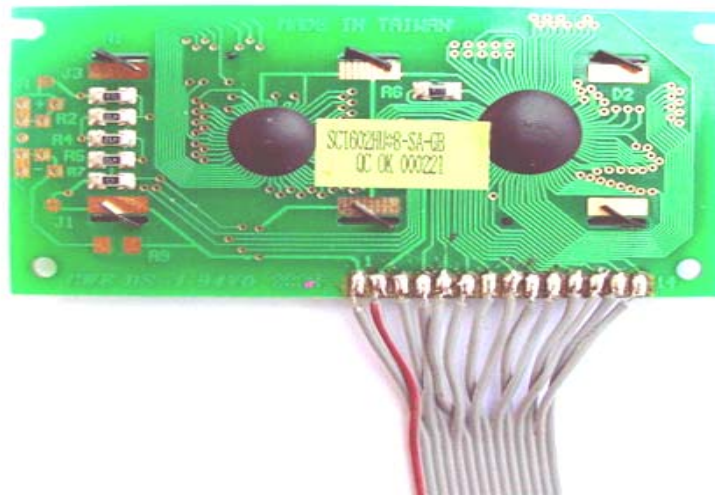
การใช้งาน LCD สามารถแบ่งออกเป็น 2 ลักษณะ คือการควบคุมการทำงานของ LCD แบบ 8 บิต และการควบคุมการทำงานแบบ 4 บิต การควบคุมการใช้งาน LCD แบบ 8 บิต โดยส่วนใหญ่จะถูกใช้งานเมื่อต่อ LCD อยู่กับ BUS ซึ่งจะทำให้การใช้งาน LCD มีความง่ายขึ้น (การใช้งานแบบ 8 บิต หมายถึงการใช้สายสัญญาณ DATA ทั้ง 8 บิต ในการควบคุม LCD)

การใช้งาน LCD แบบ 4 บิต โดยส่วนใหญ่จะถูกนำไปใช้งานได้ง่าย เมื่อต่อ LCD อยู่กับพอร์ต ซึ่งทำให้ประหยัดพอร์ตของไมโครคอนโทรลเลอร์ (การใช้งานแบบ 4 บิต หมายถึงการต่อใช้งานขา DATA เพียง 4 ขาบน D7-D4 เท่านั้น โดย 4 ขาล่าง D3-D0 จะถูกต่อลงกราวด์)

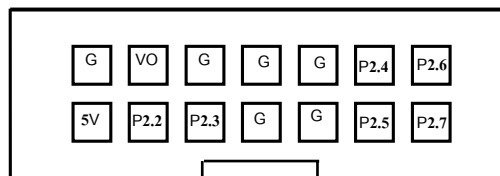
### **การเชื่อมต่อ LCD เข้ากับบอร์ด**

จากรูปที่ 37 แสดงการต่อสาย LCD เพื่อที่จะนำไปต่อกับบอร์ด โดยการต่อสาย LCD จะต้องระวังเรื่องของสายเป็นสิ่งสำคัญ เนื่องจากถ้าต่อสายผิดจะทำให้ LCD เกิดความเสียหายได้ ซึ่งการต่อ LCD จะต้องพิจารณาจากคู่มือของ LCD ในเรื่องของขาเป็นสำคัญซึ่งจะต้องต่อให้ตรงกับ PIN ขาที่กำหนดบนบอร์ด

ตัวอย่างการต่อ LCD แบบ 16 × 1 ในการต่อสายจะต้องสลับสายระหว่าง 1-2 , 3-4 , ....13-14 อันเนื่องมาจากบอร์ด จะมีการสลับสายของขาสัญญานในลักษณะเป็นคู่



รูปที่ 37 แสดงการสลับสายของสายสัญญาณในลักษณะเป็นคู่

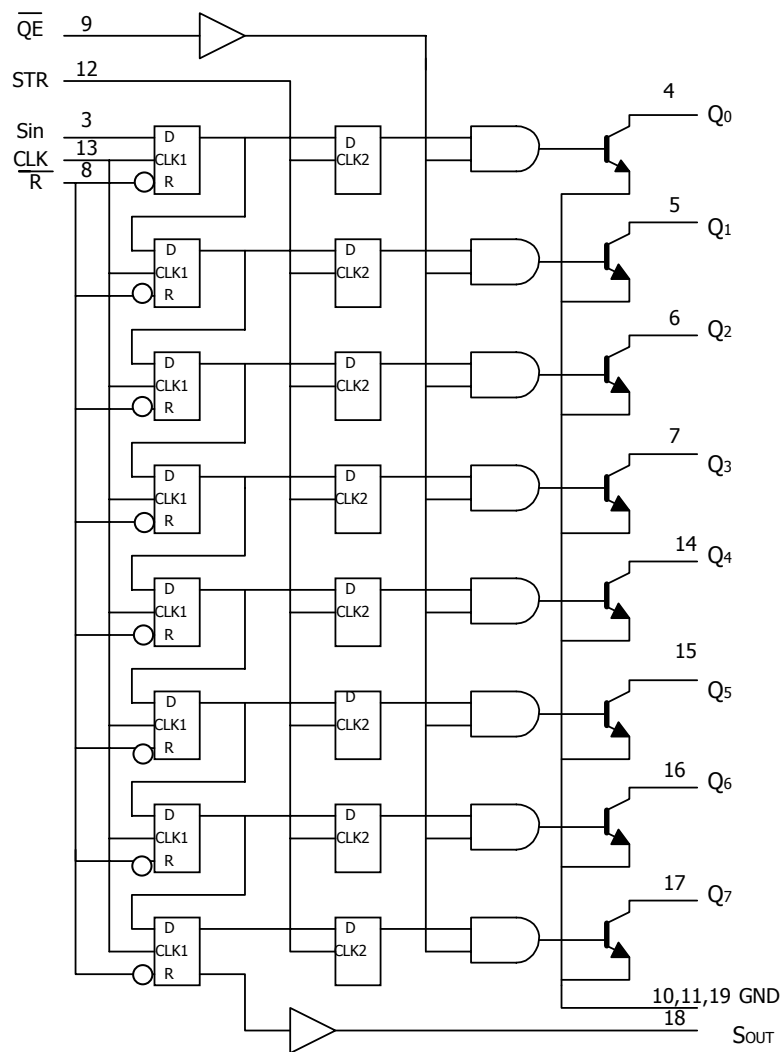


รูปที่ 38 แสดงขั้วต่อใช้งานของ LCD แบบ Character

**หมายเหตุ** เพื่อให้แน่ใจว่าการต่อสาย LCD ไม่ผิดพลาด ควรใช้ Meter ในการตรวจสอบสาย หลังจากแน่ใจแล้วว่า LCD ต่อสายถูกต้องแล้ว ให้ต่อไฟเข้ากับ บอร์ด แล้ว ปรับความสว่างของจอ LCD จะเห็นแถบ LCD เป็นแถบดำครึ่งจอ (ในกรณีที่ใช้ LCD แบบ 16 ตัวอักษร 1 บรรทัด)

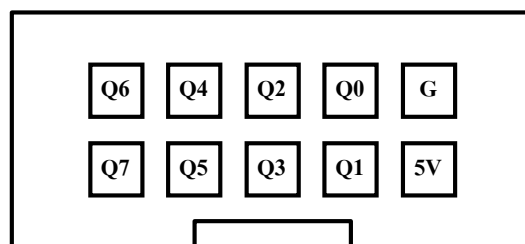
## EXP2 (NC6B595)

NC6B595 เป็นอุปกรณ์ DRIVE ที่ให้เอาต์พุตเป็นแบบ OPEN COLLECTOR ซึ่งสามารถขับโหลดได้โดยตรง ในขณะที่การควบคุมจะเหมือนกับการควบคุม 74HC595 ซึ่งสามารถใช้ Diagram และ Timing Diagram ของ 74HC595 มาเขียนโปรแกรมควบคุมได้เลย โดย Diagram ของ NC6B595 แสดงดังรูป



รูปที่ 39 แสดงไดอะแกรมของ NC6B595

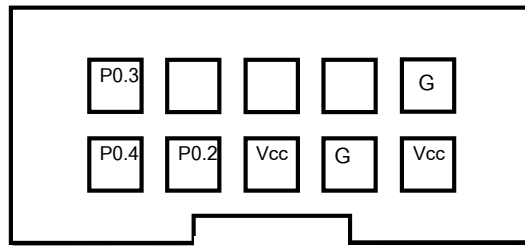
จากที่ได้กล่าวมาว่าเอาต์พุตของ NC6B595 เป็นแบบ OPEN COLLECTOR จึงสามารถนำไปขับโหลดได้โดยตรง แต่ถ้านำไปขับโหลดประเภท INDUCTOR เช่น RELAY จะต้องมี DIODE ต่อกับโหลดอยู่ด้วย โดยสัญญาณ Q0 – Q7 จะต่อเข้ากับขั้วต่อแบบ IDC ขนาด 10 PIN ดังนี้



รูปที่ 40 แสดงขั้วต่อ EXP2

ขั้ว ET-SDP8

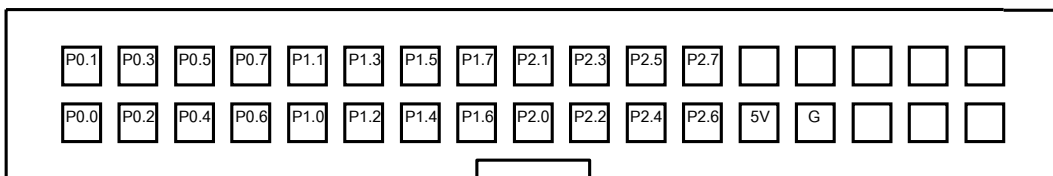
จากรูปที่ 41 แสดงให้เห็นขั้ว ET-SDP8 ที่อยู่บนบอร์ด CP-SPI/S8252 V2.0& CP-SPI/RD2 V2.0



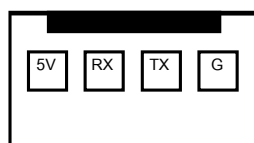
รูปที่ 41 แสดงขั้วต่อใช้งานของ ET-SDP8



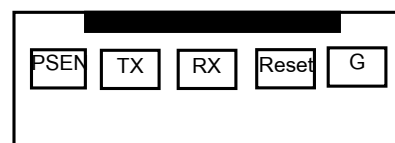
รูปที่ 42 แสดงบอร์ด CP-SPI/S8252 V2.0&CP-SPI/RD2 V2.0



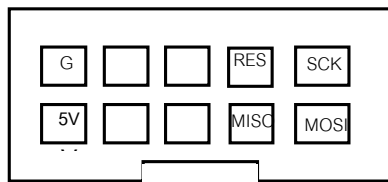
a) ขั้วต่อ 72IOZ80



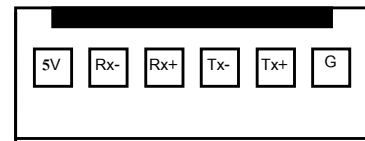
b) ขั้วต่อ RS232



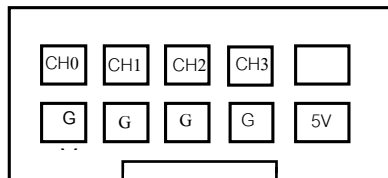
c) ขั้วต่อ LOAD(RD2)



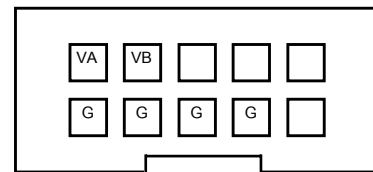
d) ขั้วต่อ LOAD(S8252)



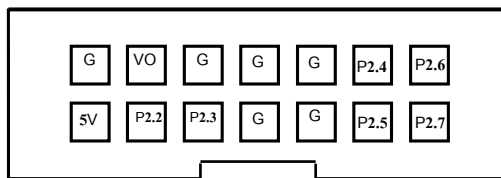
e) ขั้วต่อ RS422/485



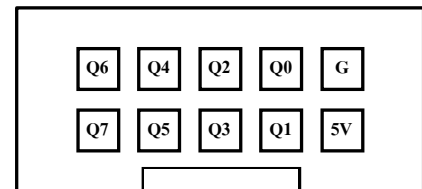
f) ขั้วต่อ ADS7841



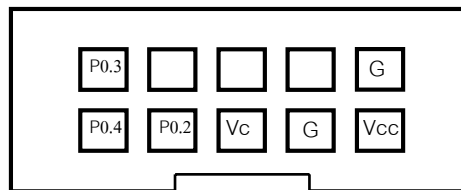
g) ขั้วต่อ LTC1661



h) ขั้วต่อ LCD Character

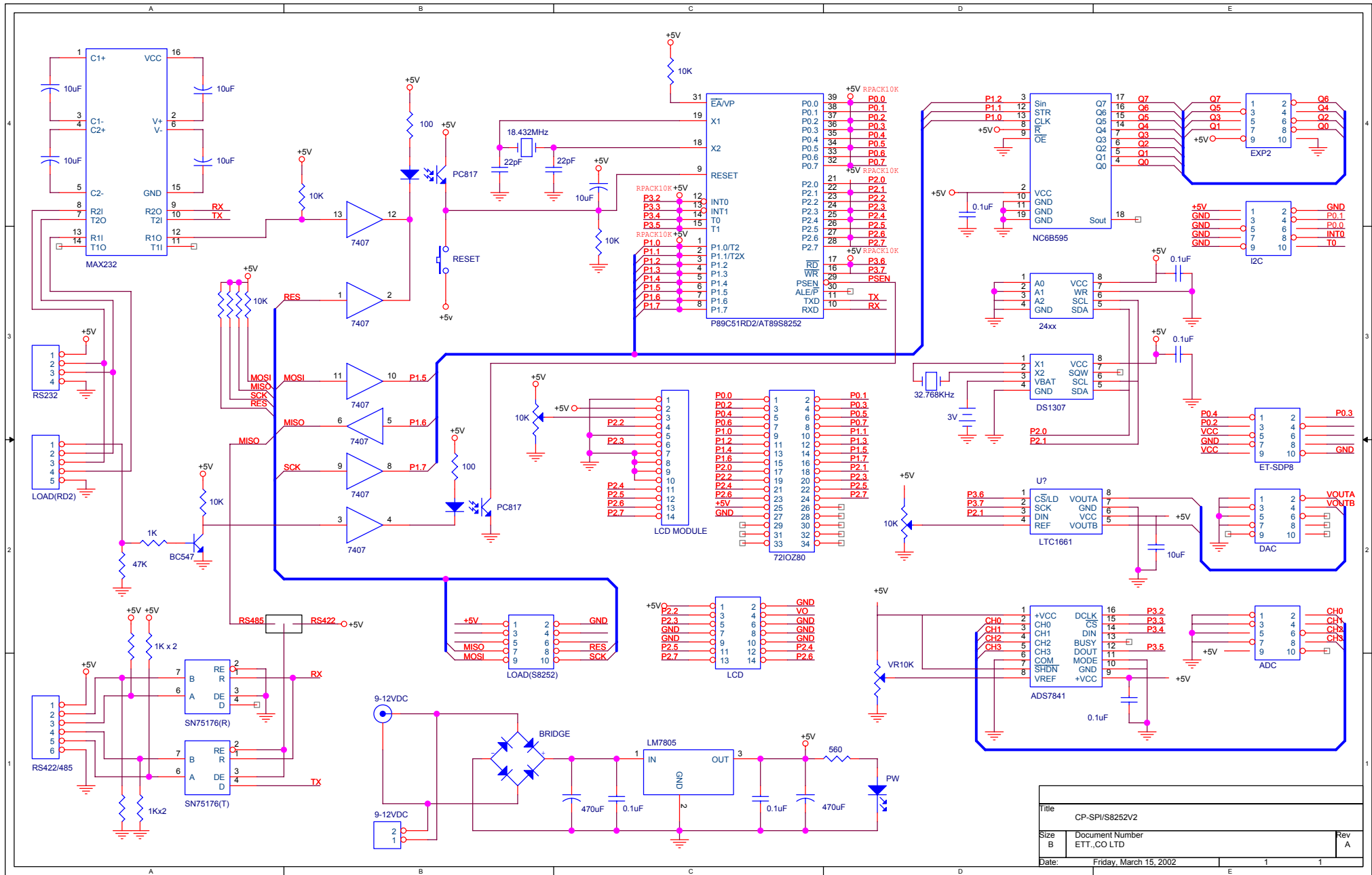


i) ขั้วต่อ EXP2



j) ขั้วต่อ ET-SDP8

รูปที่ 43 แสดงขั้วต่อใช้งานของอุปกรณ์ต่าง ๆ บนบอร์ด CP-SPI/S8252 V2.0&CP-SPI/RD2 V2.0



Title			
CP-SPI/S8252V2			
Size	Document Number		Rev
B	ETT...CO LTD		A
Date:	Friday, March 15, 2002	1	1

### CP-SPI/S8252 V3.0&CP-SPI/RD2 V3.0

บอร์ด CP-SPI/S8252 V3.0 จะใช้ CPU AT89S8252 เป็น CPU ประจำบอร์ด และบอร์ด CP-SPI/RD2 V3.0 จะใช้ CPU P89C51RD2 เป็น CPU ประจำบอร์ด โดยมีส่วนอื่นๆ ของวงจรที่เหมือนกัน ส่วนการ DOWNLOAD โปรแกรมเข้าสู่ตัวบอร์ดนั้น บอร์ด CP-SPI/S8252 V3.0 จะใช้โปรแกรม WS8252 และบอร์ด CP-SPI/RD2 V3.0 จะใช้โปรแกรม P89C51RD2

ในบอร์ด CP-SPI/S8252 V3.0 & CP-SPI/RD2 V3.0 จะเหมือนกับ CP-SPI/S8252 V2.0 & CP-SPI/RD2 V2.0 ทุกประการเพียงแต่มีการเพิ่มอุปกรณ์ที่สำคัญเข้าไปอีก 2 ตัว ดังนี้

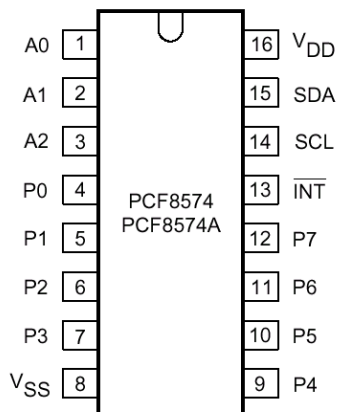
1. I<sup>2</sup>C PORT (PCF8574)
2. 34 PIN I/O BUS

#### PCF8574

##### รายละเอียดเบื้องต้น

PCF8574 เป็นอุปกรณ์ที่ใช้ขยายเอาต์พุตและอินพุตของไมโครคอนโทรลเลอร์โดยผ่านทางระบบบัสแบบ I<sup>2</sup>C โดยในแต่ละขาสามารถเป็นได้ทั้งอินพุตและเอาต์พุต โดย PCF8574 จำนวน 1 ตัวสามารถนำมาทำเป็นอินพุตหรือเอาต์พุตได้ 8 I/O แต่ถ้าต้องการ I/O ที่มากกว่า 8 จะต้องต่อ PCF8574 เพิ่มขึ้นอีก ซึ่งในตัวเดียวกันสามารถต่อ PCF8574 ได้ 8 ตัว โดยกำหนดตำแหน่งของแต่ละตัวให้แตกต่างจากขา A0, A1, A2 อีกทั้งยังสามารถสร้างสัญญาณอินเตอร์รัพท์ให้กับไมโครคอนโทรลเลอร์เมื่อมีสัญญาณเข้ามาทางอินพุตได้อีกด้วย

##### โครงสร้างภายนอกและขาต่อใช้งาน

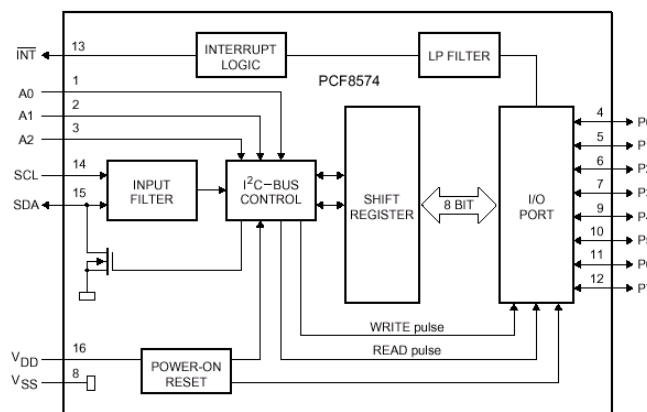


รูปที่ 44 แสดงโครงสร้างภายนอกและขาต่อใช้งานของ PCF8574

ตารางแสดงตำแหน่งขาของ PCF8574

ตำแหน่ง	สัญลักษณ์	รายละเอียด
1	A0	แอดเดรส A0
2	A1	แอดเดรส A1
3	A2	แอดเดรส A2
4	P0	อินพุท/เอาต์พุท P0
5	P1	อินพุท/เอาต์พุท P1
6	P2	อินพุท/เอาต์พุท P2
7	P3	อินพุท/เอาต์พุท P3
8	Vss	กราวด์
9	P4	อินพุท/เอาต์พุท P4
10	P5	อินพุท/เอาต์พุท P5
11	P6	อินพุท/เอาต์พุท P6
12	P7	อินพุท/เอาต์พุท P7
13	INT	อินเตอร์รัพท์
14	SCL	สัญญาณClock ของ I <sup>2</sup> C
15	SDA	สัญญาณข้อมูลของ I <sup>2</sup> C
16	Vdd	ไฟบวก

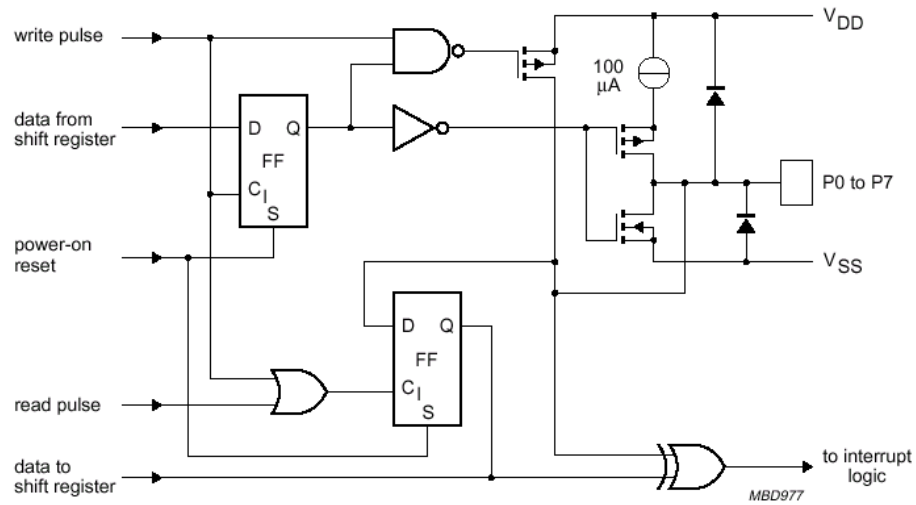
โครงสร้างภายใน



รูปที่ 45 แสดงโครงสร้างภายในของ PCF8574



### โครงสร้างภายใน PORT

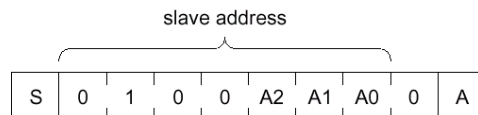


รูปที่ 46 แสดงโครงสร้างภายใน PORT ของ PCF8574

### Quasi – bidirectional I/O

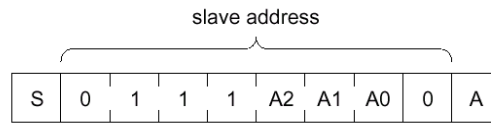
คือวงจรที่สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต โดยไม่ต้องใช้สัญญาณควบคุมทิศทาง เมื่อจ่ายไฟให้กับ PCF8574 ขา I/O ทุกขาจะมีลอจิก 1 โดยจะต้องกำหนดให้ I/O เป็น 1 ก่อนที่จะใช้งาน PCF8574 เป็นอินพุต

### การกำหนดตำแหน่งของ PCF8574



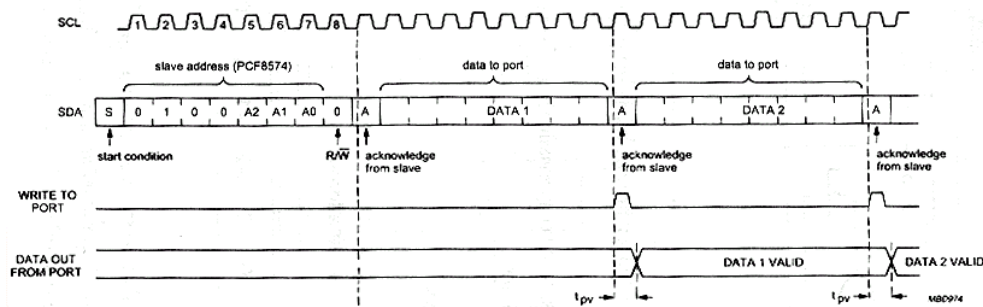
รูปที่ 47 แสดงการกำหนดตำแหน่งของ PCF8574

PCF8574 มีอยู่ 2 รุ่น คือ PCF8574 และ PCF8574A โดยทั้ง 2 รุ่นสามารถใช้งานได้เหมือนกัน จะแตกต่างกันเพียงไบต์ข้อมูลที่บรรจุตำแหน่งเท่านั้น โดยรูปที่ 47 เป็นการกำหนดตำแหน่งของ PCF8574 และรูปที่ 48 เป็นการกำหนดตำแหน่งของ PCF8574A



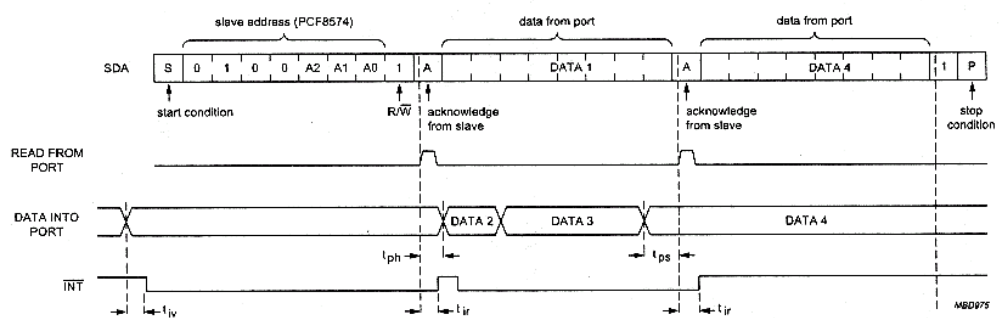
รูปที่ 48 แสดงการกำหนดตำแหน่งของ PCF8574A

**การเขียนข้อมูลออกพอร์ตของ PCF8574**



รูปที่ 49 แสดงไคอะแกรมของการเขียนข้อมูลออกพอร์ตของ PCF8574

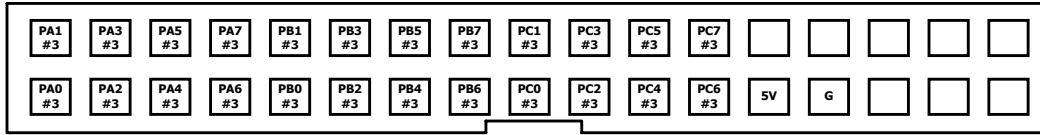
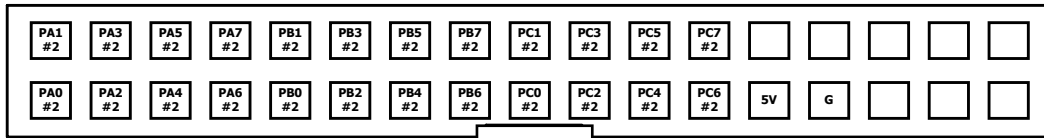
**การอ่านข้อมูลออกพอร์ตของ PCF8574A**



รูปที่ 50 แสดงการอ่านข้อมูลออกพอร์ตของ PCF8574A

**ขั้ว 34 PIN I/O BUS (72IOZ80)**

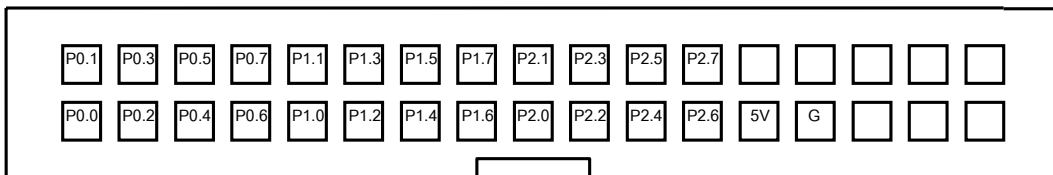
จากรูปที่ 51 แสดงให้เห็นขั้วต่อ 34 PIN I/O BUS (72IOZ80) ของ PCF8574 ที่มีอยู่บนบอร์ด CP-SPI/S8252V3.0 & CP-SPI/RD2 V3.0 ทั้งหมด 2 ชุด



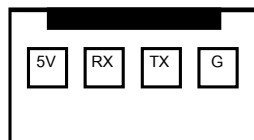
รูปที่ 51 แสดงให้เห็นขั้วต่อ 34 PIN I/O BUS (72IOZ80) ที่ #2 และ #3



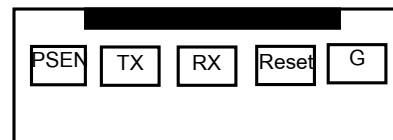
รูปที่ 52 แสดงบอร์ด CP -SPI/S8252 V3.0&CP-SPI /RD2 V3.0



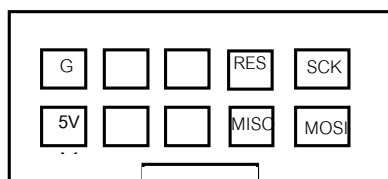
a) ขั้วต่อ 72IOZ80



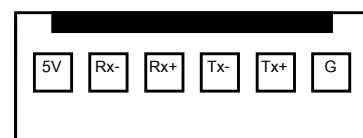
b) ขั้วต่อ RS232



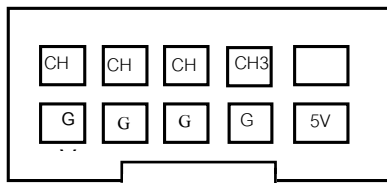
c) ขั้วต่อ LOAD(RD2)



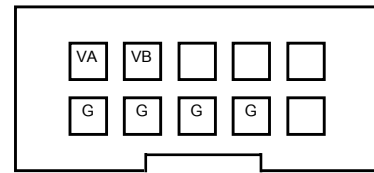
d) ขั้วต่อ LOAD(S8252)



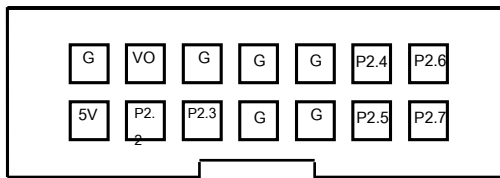
e) ขั้วต่อ RS422/485



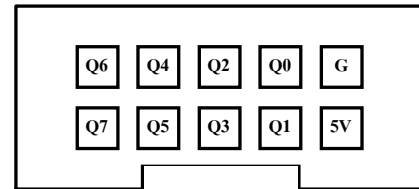
ก) ขั้วต่อ ADS7841



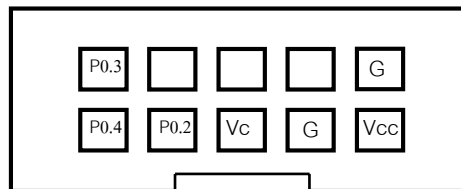
ข) ขั้วต่อ LTC1661



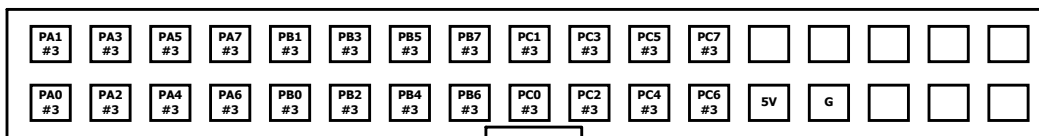
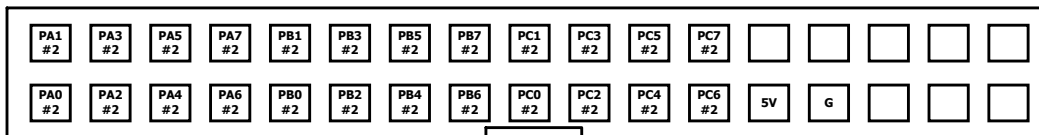
ค) ขั้วต่อLCD Character



ง) ขั้วต่อ EXP2

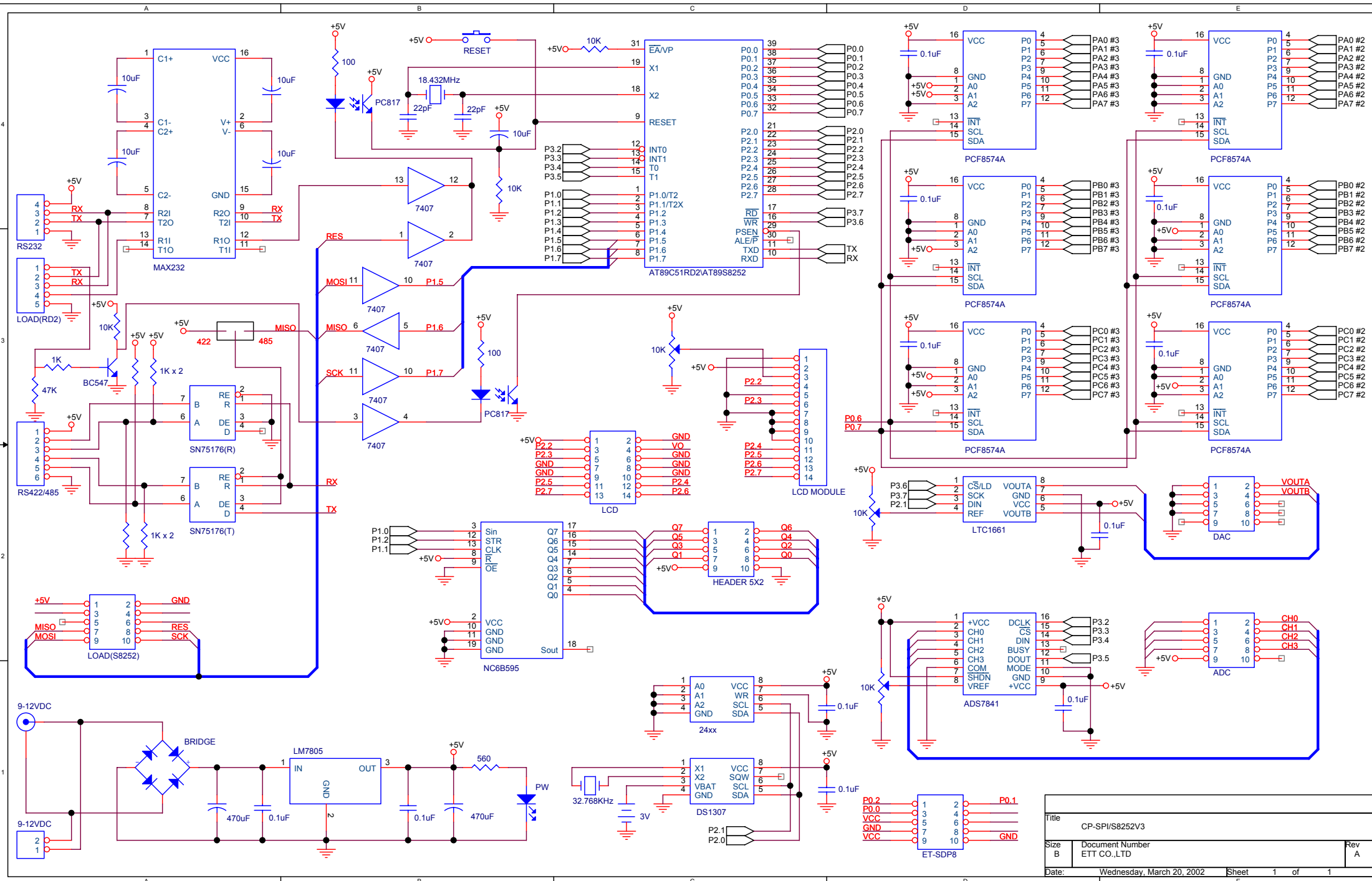


จ) ขั้วต่อ ET-SDP8



ฉ) ขั้วต่อ 72IOZ80 ของ PCF8574 #2 และ #3

รูปที่ 53 แสดงขั้วต่อใช้งานของอุปกรณ์ต่าง ๆ บนบอร์ด CP-SPI/S8252 V3.0&CP-SPI/RD2 V3.0



Title		
CP-SPI/S8252V3		
Size	Document Number	Rev
B	ETT CO.,LTD	A
Date:	Wednesday, March 20, 2002	Sheet 1 of 1

## การใช้งานโปรแกรม DOWNLOAD สำหรับบอร์ด CP-SPI/S8252 V1,V2,V3

การ Download Program สามารถทำได้ 2 วิธี คือ การ Download Program ทีละลำดับขั้นตอน และ การDownload Program แบบ Auto

### การ Download Program ทีละลำดับขั้นตอน

1. ทำการ Copy โปรแกรม WS8252 จากแผ่นโปรแกรม ลงในเครื่องคอมพิวเตอร์
2. ทำการเชื่อมต่อวงจรเพื่อเตรียมความพร้อมก่อนการใช้งาน ด้วยการต่อสาย ET-PSPI โดยขั้ว 25 PIN ต่อเข้ากับ Printer Port และขั้ว 10 PIN ต่อเข้าบอร์ด CP-SPI/S8252 จากนั้นทำการต่อ Power Supply เข้ายังตัวบอร์ด
3. ทำการเรียกใช้งานโปรแกรม WS8252 ซึ่งจะได้นหน้าต่างแสดงดังรูป



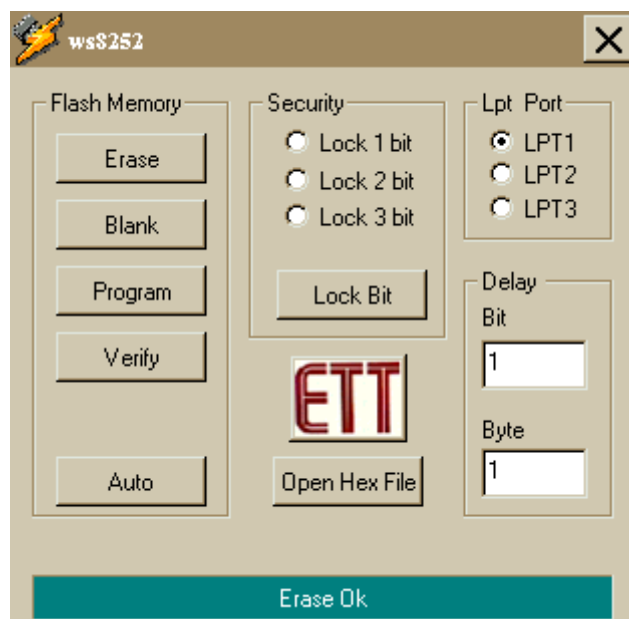
4. เลือก Printer Port ที่เชื่อมต่อจากเครื่องคอมพิวเตอร์ โดยการคลิกเมาส์เลือก ในส่วนของ Ltp Port โดยการเลือกนั้นต้องมีความสัมพันธ์กับ Printer Port ที่ถูกต่อเชื่อมต่อกับคอมพิวเตอร์

5. ทำการป้อนค่า Bit และ Byte ในส่วนของ Delay ให้มีค่า 0-9 (สามารถดูรายละเอียดการป้อนค่าได้จากภาคผนวก)

6. เปิดไฟล์ .HEX ที่ต้องการมาเก็บไว้ใน Data Buffer สามารถทำได้ โดยใช้เมาส์คลิกที่ Open Hex File ทำการเลือกไฟล์ที่ต้องการ ที่หน้าต่างการทำงานทางด้านล่างสุดจะปรากฏข้อความตามรูป ซึ่งแสดงว่าการเรียกโปรแกรมเสร็จสมบูรณ์



7. คลิกเมาส์ที่ปุ่ม Erase เพื่อทำการลบข้อมูลเก่าที่มีอยู่ใน CPU ออก เมื่อเสร็จสิ้นแล้วจะปรากฏข้อความ ตามรูป



8. คลิกเมาส์ที่ปุ่ม Program เป็นการส่งโปรแกรมที่อยู่ใน Data Buffer มาเก็บไว้ใน CPU เมื่อเสร็จสมบูรณ์แล้วจะปรากฏข้อความตามรูป

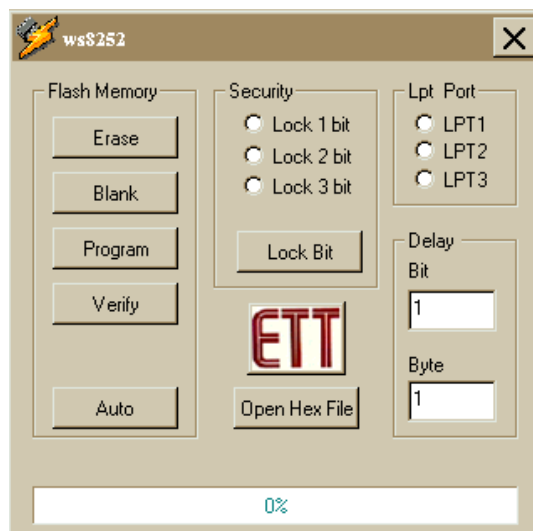


**หมายเหตุ** เมื่อเราทำการโหลดข้อมูลเข้าสู่ CPU เสร็จสิ้นตามข้อ 8 แล้ว โปรแกรมจะทำการ RUN ผลที่เอาที่พูดทันที

**ขั้นตอนการ Download Program โดยใช้วิธี Auto**

1. หลังจากที่เรทำการ Copy โปรแกรม WS8252 จากแผ่นโปรแกรม ลงในเครื่องคอมพิวเตอร์ และทำการเชื่อมต่อวงจรเพื่อเตรียมความพร้อมก่อนการใช้งาน ด้วยการต่อสาย ET-PSPI โดยขั้ว 25 PIN ต่อเข้ากับ Printer Port และขั้ว 10 PIN ต่อเข้ากับบอร์ด CP-SPI/S8252 จากนั้นทำการต่อ Power Supply เข้ายังตัวบอร์ด

3. ทำการเรียกใช้งานโปรแกรม WS8252 ซึ่งจะได้หน้าต่างแสดงดังรูป





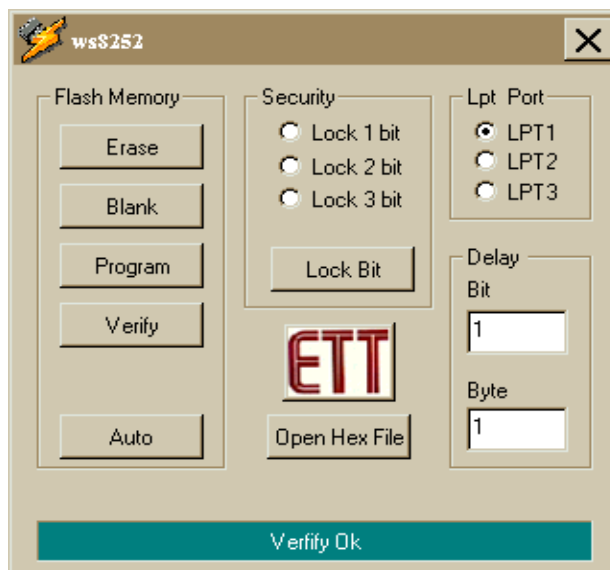
4. เลือก Printer Port ที่เชื่อมต่อจากเครื่องคอมพิวเตอร์ โดยการคลิกเมาส์เลือก ใน ส่วนของ Lpt Port โดยการเลือกนั้นต้องมีความสัมพันธ์กับ Printer Port ที่ถูกต่อเชื่อมต่อกับ คอมพิวเตอร์

5. ทำการป้อนค่า Bit และ Byte ในส่วนของ Delay ให้มีค่า 0-9 (สามารถดูรายละเอียดการ ป้อนค่าได้จากภาคผนวก)

6. เปิดไฟล์ .HEX ที่ต้องการมาเก็บไว้ใน Data Buffer สามารถทำได้ โดยใช้เมาส์คลิกที่ Open Hex File ทำการเลือกไฟล์ที่ต้องการ ที่หน้าต่างการทำงานทางด้านล่างสุดจะปรากฏข้อความตามรูป ซึ่งแสดงว่าการเรียกโปรแกรมเสร็จสมบูรณ์



7. คลิกเมาส์ที่ปุ่ม Auto โปรแกรมจะทำงานเป็นขั้นตอนและ Run โดยอัตโนมัติ ซึ่งขณะที่ ทำการ Run โปรแกรมนั้นจะปรากฏหน้าต่างซึ่งมีข้อความดังนี้

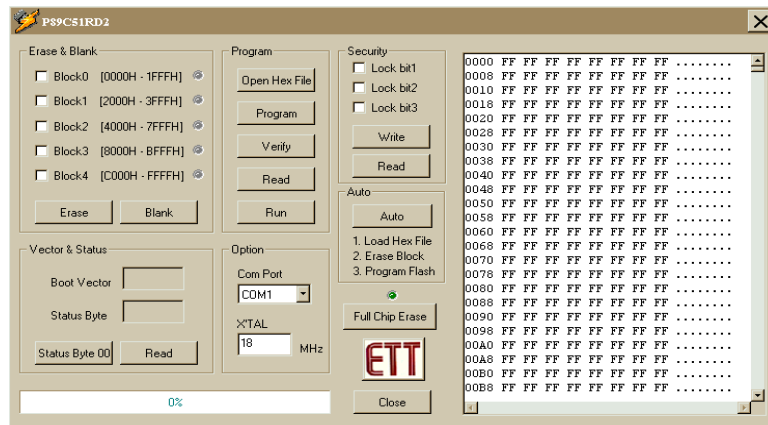


## การใช้งานโปรแกรม DOWNLOAD สำหรับบอร์ด CP-SPI/RD2 V1,V2,V3

การ Download Program สามารถทำได้ 2 วิธี คือ การ Download Program ทีละลำดับขั้นตอน และ การDownload Program แบบ Auto

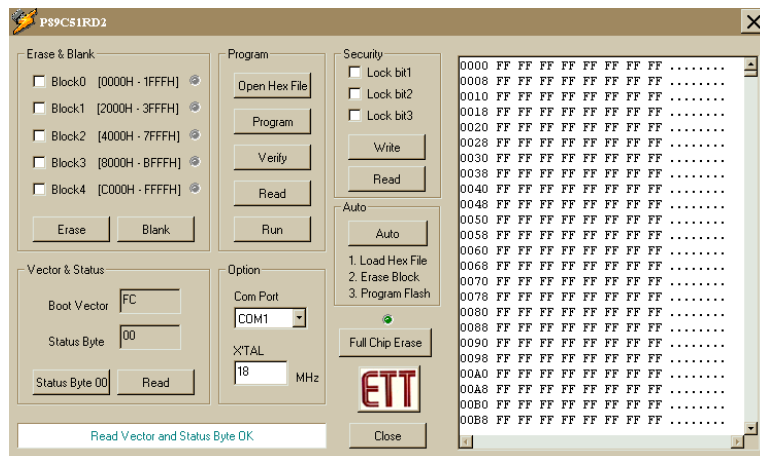
### การ Download Program ทีละลำดับขั้นตอน

1. ทำการ Copy โปรแกรม P89C51RD2 จากแผ่นโปรแกรม ลงในเครื่องคอมพิวเตอร์
2. ทำการเชื่อมต่อวงจรเพื่อเตรียมความพร้อมก่อนการใช้งาน ด้วยการต่อสาย SPI/RD2 (โดยขั้ว 9 PIN ต่อเข้ากับ Com Port ของคอมพิวเตอร์ และขั้ว 5 PIN ต่อเข้ากับบอร์ด CP-SPI/RD2) ดังรูป พร้อมต่อ Power Supply เข้ายังตัวบอร์ด จากนั้น RUN โปรแกรม P89C51RD2 ซึ่งจะได้นหน้าต่างแสดงดังรูป

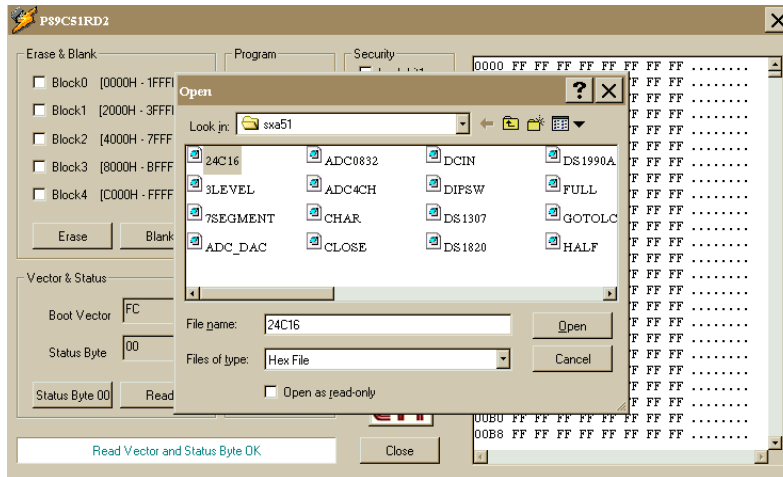


3. เลือก Com Port ในส่วนของ Option ให้ตรงกับ Com Port ที่เชื่อมต่อบนเครื่อง PC ในที่นี้ทำการเชื่อมต่อกับ Com1 ของ PC และป้อนค่า xtal ตามค่าที่ใช้งานจริงดังตัวอย่างคือใช้ค่า Xtal เท่ากับ 18.432 MHz (หรือใช้ 18 ค่าเดิมที่ปรากฏอยู่เลยก็ได้)

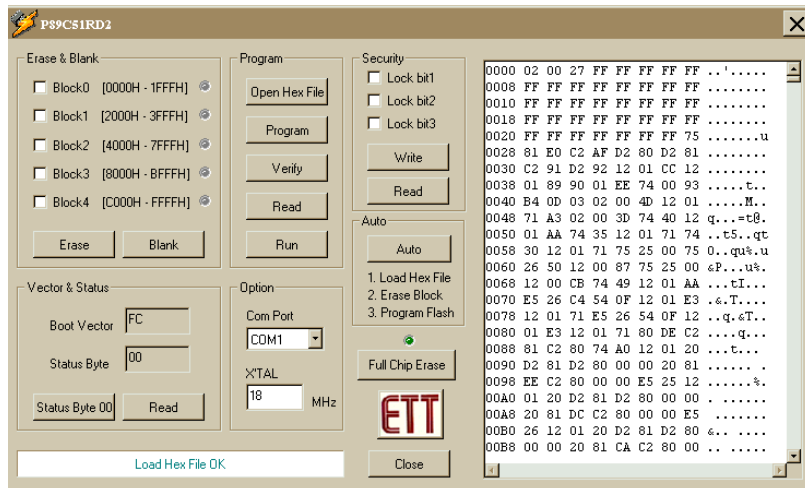
4. ใช้เมาส์คลิกที่ Read ในส่วนของ Vector&Status เพื่อตรวจสอบค่า Status Byte จะต้องเป็นซึ่ง จะปรากฏหน้าต่างการทำงานดังรูป โดย BOOT VECTOR=FC , STATUS BYRE=00



5. การเปิดไฟล์ .HEX ที่ต้องการโหลดลงใน CPU สามารถทำได้ในส่วนของ Program โดยใช้เมาส์คลิกที่ Open Hex File ซึ่งจะได้ดังรูป ในตัวอย่างเป็นการโหลดไฟล์ชื่อ 24C16.HEX เมื่อเลือกไฟล์ที่ต้องการได้แล้ว ให้เมาส์คลิกที่ OK



จากนั้นจะสามารถดู CODE ที่เป็น HEX ได้จากหน้าต่างทางด้านขวาสุด และเมื่อทำการเลือกไฟล์โปรแกรม เสร็จสิ้นแล้วจะปรากฏข้อความในส่วนของ Display ด้านล่าง



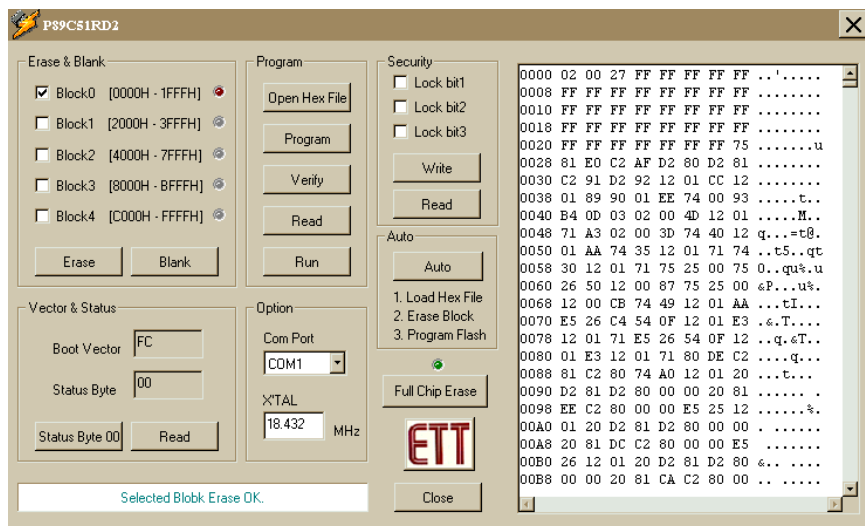
6. โปรแกรมข้อมูลจะปรากฏอยู่บนหน้าต่างของ Buffer ทางขวามือ

```

0000 02 00 27 FF FF FF FF FF ...'.....
0008 FF FF FF FF FF FF FF FF .....
0010 FF FF FF FF FF FF FF FF .....
0018 FF FF FF FF FF FF FF FF .....
0020 FF FF FF FF FF FF FF 75 .....u
0028 81 E0 C2 AF D2 80 D2 81 .....
0030 C2 91 D2 92 12 01 CC 12 .....
0038 01 89 90 01 EE 74 00 93 .....t..
0040 B4 0D 03 02 00 4D 12 01 .....H..
0048 71 A3 02 00 3D 74 40 12 q...=t8.
0050 01 AA 74 35 12 01 71 74 ...t5..qt
0058 30 12 01 71 75 25 00 75 0..qt%.u
0060 26 50 12 00 87 75 25 00 &P...u%.
0068 12 00 CB 74 49 12 01 AA ...tI...
0070 E5 26 C4 54 0F 12 01 E3 .&.T...
0078 12 01 71 E5 26 54 0F 12 ..q.&T..
0080 01 E3 12 01 71 80 DE C2 ...q...
0088 81 C2 80 74 A0 12 01 20 ...t...
0090 D2 81 D2 80 00 00 20 81 .....
0098 EE C2 80 00 00 E5 25 12 .....%.
00A0 01 20 D2 81 D2 80 00 00 .....
00A8 20 81 DC C2 80 00 00 E5 .....
00B0 26 12 01 20 D2 81 D2 80 &...
00B8 00 00 20 81 CA C2 80 00 ..
    
```

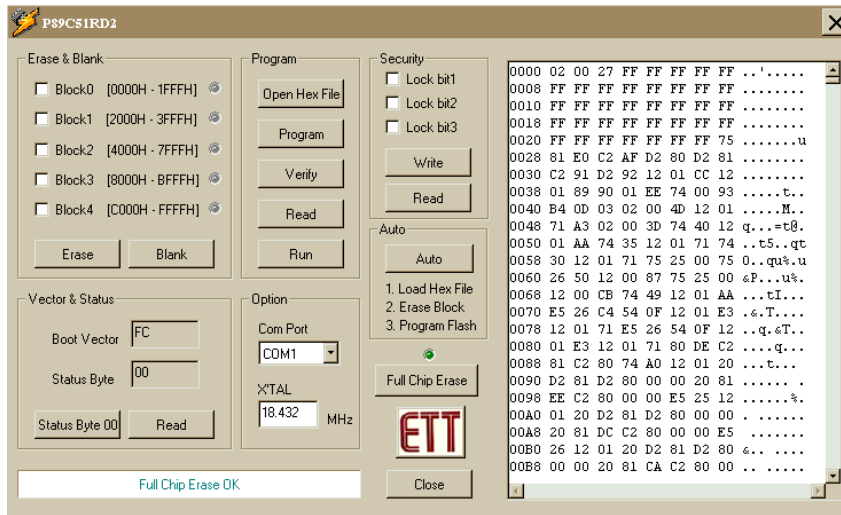
7. ทำการลบข้อมูลเก่าใน CPU ออก ซึ่งสามารถทำได้ในส่วนของ Erase&Blank โดยใช้เมาส์คลิกเลือก Block ที่ต้องการจะลบ (ถ้าโปรแกรมที่ทำการ Download มา มีค่าAddressอยู่ในช่วง 0000H-1FFFFH ให้ทำการเลือกลบข้อมูลเก่าในส่วนของ Block0 ก็เพียงพอแล้ว แต่ถ้าหากโปรแกรมที่ Download มี Address จาก 0000H-3FFFFH ให้ทำการคลิกเมาส์เพื่อเลือกลบที่ Block0 และ Block1 ก็เพียงพอ)

8. จากนั้นคลิกเมาส์ที่ปุ่ม Erase วิธีการ Erase แบบนี้ของ P89C51RD2 เป็นการลบข้อมูลเก่าเป็น Block เมื่อทำการลบเสร็จจะปรากฏข้อความตาม Display ด้านล่าง

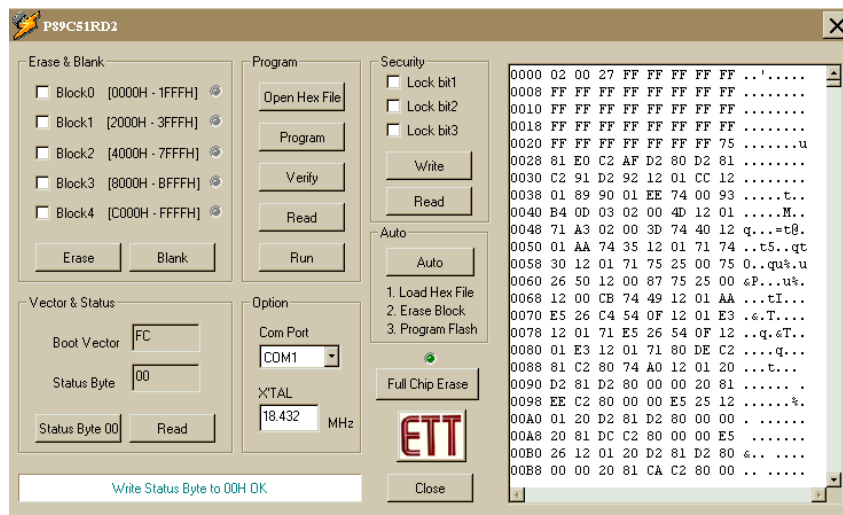


**หมายเหตุ** วิธีการลบข้อมูลภายใน CPU สามารถทำได้ 2 วิธี คือการลบทีละ Block (ตามข้อ 8) และอีกวิธีหนึ่งคือ การลบข้อมูลทั้งหมดใน CPU คือ การ Full Chip Erase แต่วิธีการลบแบบ Full Chip Erase ทำให้ Status Byte ถูกแก้ไขจาก 00 เป็น FF จึงทำให้ต้องคลิกเมาส์ที่ Status Byte 00 เพื่อทำการแก้ไขค่า Status Byte เป็น 00 และผลที่เกิดจากการ Full Chip Erase อีกอย่างหนึ่งก็คือ ถ้าผู้ใช้มีการ Lock Bit ในส่วนของ Security เอาไว้ เมื่อเราทำการ Full Chip Erase ก็จะทำให้การ Lock Bit ถูกยกเลิกออกไปด้วย

การลบข้อมูลเก่าด้วยวิธี Full Chip Erase ทำได้โดยการคลิกเมาส์ที่ ปุ่ม Full Chip Erase ไฟลีย์เขียวจะกระพริบ จนหยุดกระพริบ ซึ่งจะปรากฏตามหน้าต่างการทำงานดังรูป

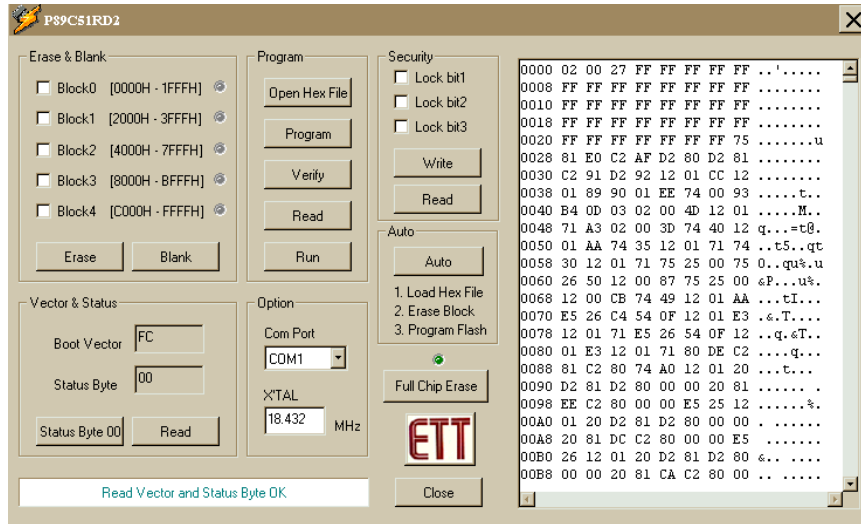


จากนั้น ให้ทำการคลิกเมาส์ที่ Status Byte 00 (เป็นการทำให้ค่า Status Byte เป็น 00) จะปรากฏข้อความที่จอ Display ตามหน้าต่างการทำงาน ดังรูป

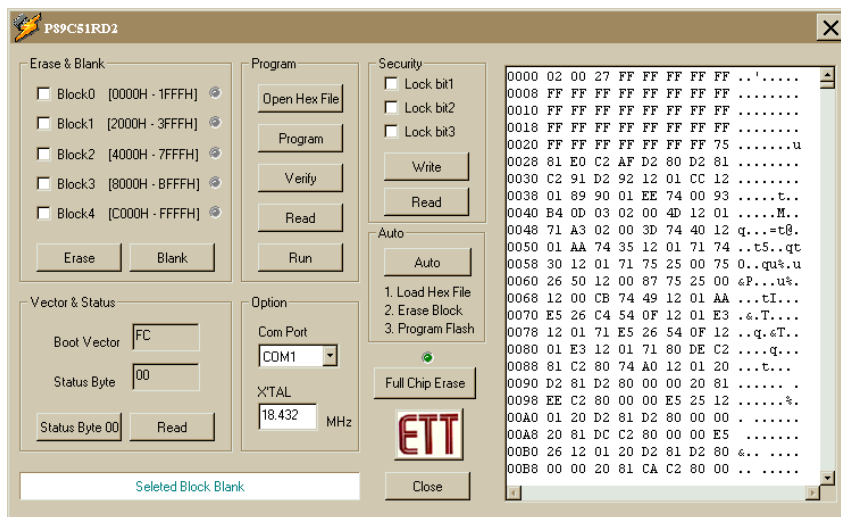


ทำการอ่านค่า Status Byte ใหม่ โดยการคลิกเมาส์ที่ปุ่ม Read ใน ส่วนของ Vector&Status

ค่า Status จะกลับไปเป็นค่า 00 เหมือนเดิม ปรากฏตามหน้าต่างการทำงานดังรูป

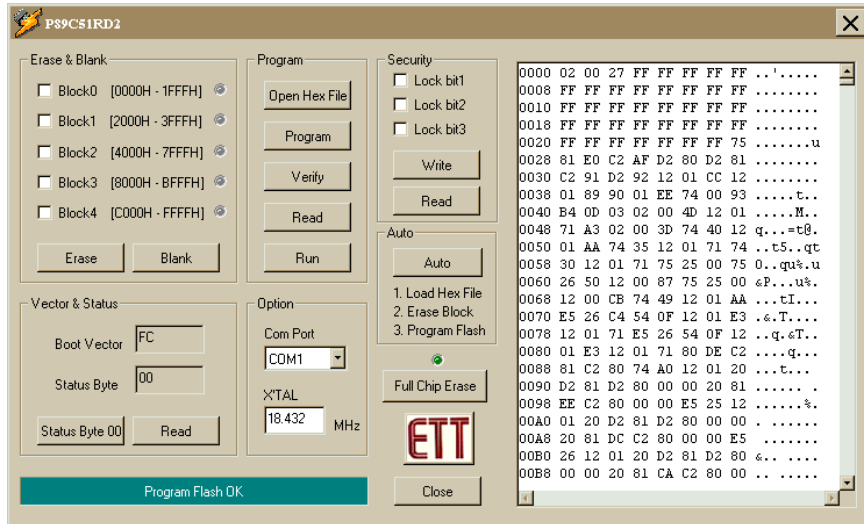


เมื่อเราทำการลบข้อมูลเก่าใน CPU โดยการ ใช้วิธี Erase อย่างใดอย่างหนึ่งแล้ว สามารถทำการตรวจสอบได้ว่าข้อมูลเก่าใน CPU ถูกลบออกไปแล้วจริงๆ โดย การคลิกเมาส์ที่ปุ่ม Blank ในส่วน ของ Erase & Blank ซึ่งถ้าข้อมูลถูกลบออกแล้วจริงจะปรากฏหน้าต่างการทำงานดังรูป

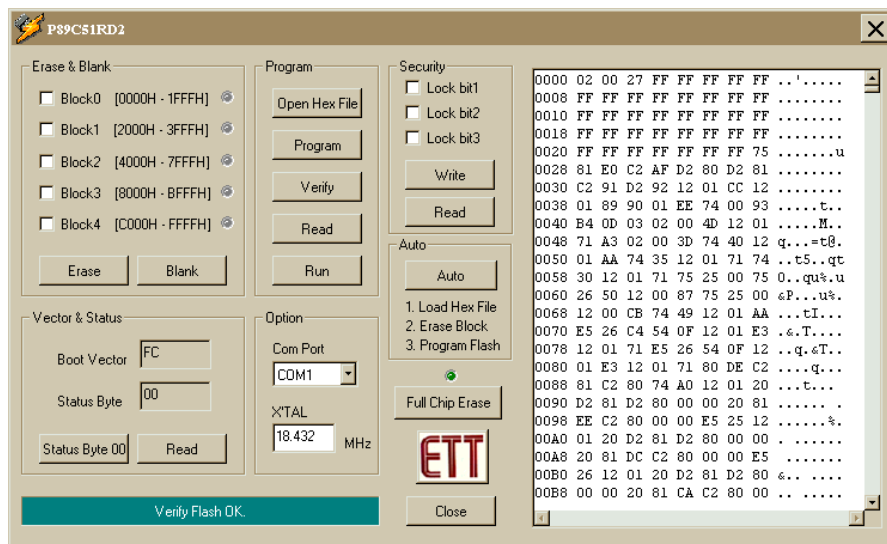


9. ใช้เมาส์คลิกที่ปุ่ม Program เป็นการส่งโปรแกรมข้อมูลจาก Buffer มวลลงใน CPU ซึ่งโปรแกรมจะนำข้อมูลใน Buffer ของหน้าต่าง Buffer ลงในหน่วยความจำของ CPU ตาม

ตำแหน่งที่อยู่ใน Buffer เมื่อส่งโปรแกรมลง CPU เสร็จสมบูรณ์แล้วจะปรากฏข้อความที่ Display ด้านล่าง



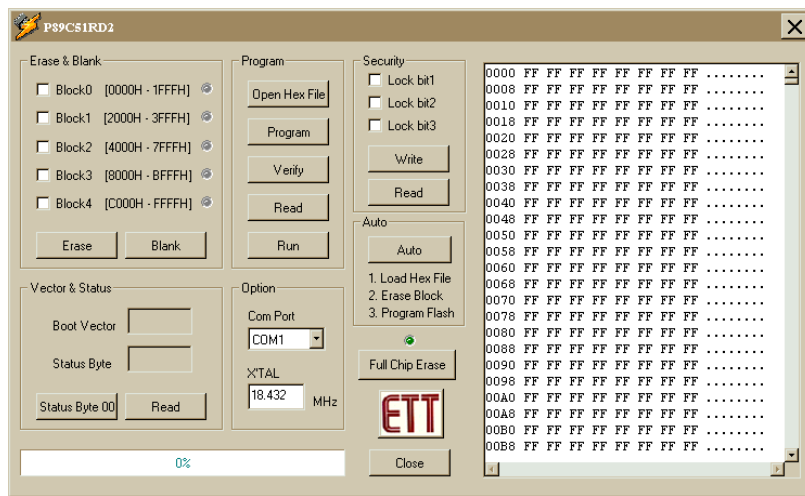
10. หากต้องการตรวจสอบว่า ข้อมูลโปรแกรมที่โหลดจาก Buffer กับ โปรแกรมภายใน CPU ตรงกันหรือไม่ให้ทำการคลิกเมาส์ที่ปุ่ม Verify ถ้าหากโปรแกรมของทั้ง 2 ส่วนตรงกันจะปรากฏข้อความตาม Display ด้านล่าง



11. ทำการคลิกเมาส์ที่ปุ่ม Run เพื่อดูผลการทำงานของโปรแกรม

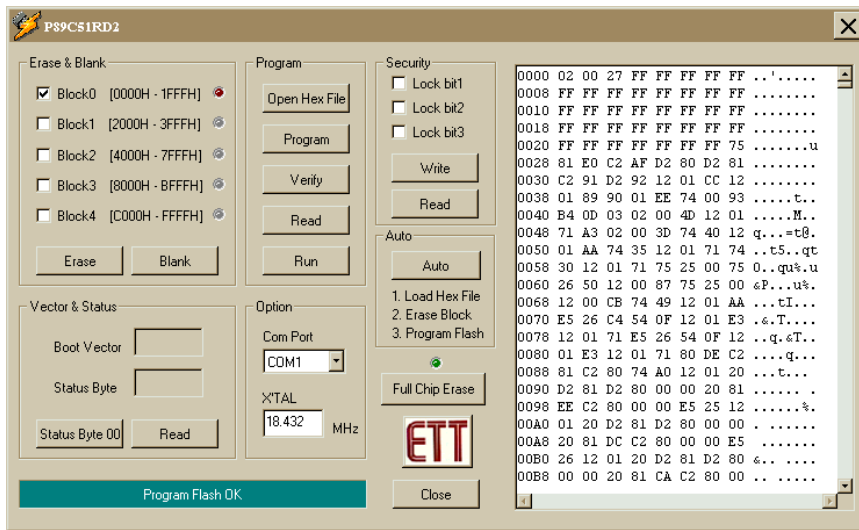
**ขั้นตอนการ Download โดยวิธีการ Auto**

1. ทำการ Copy โปรแกรม P89C51RD2 จากแผ่นโปรแกรม ลงในเครื่องคอมพิวเตอร์
2. ทำการเชื่อมต่อวงจรเพื่อเตรียมความพร้อมก่อนการใช้งาน ด้วยการต่อสาย SPI/RD2 (โดยขั้ว 9 PIN ต่อเข้ากับ Com Port ของคอมพิวเตอร์ และขั้ว 5 PIN ต่อเข้ากับบอร์ด CP-SPI/RD2) ดังรูป พร้อมต่อ Power Supply เข้ายังตัวบอร์ด จากนั้น RUN โปรแกรม P89C51RD2 ซึ่งจะได้นหน้าต่างแสดงดังรูป



3. เลือก Com Port ในส่วนของ Option ให้ตรงกับ Com Port ที่เชื่อมต่อบนเครื่อง PC ในที่นี้ทำการเชื่อมต่อกับ Com1 ของ PC และป้อนค่า xtal ตามค่าที่ใช้งานจริงดังตัวอย่างคือใช้ค่า Xtal เท่ากับ 18.432 MHz
4. ทำการเปิดไฟล์โปรแกรม โดยคลิกเมาส์ที่ Open Hex File จะปรากฏหน้าต่างให้ทำการเลือกไฟล์ (หากเราไม่ได้ทำการ Open Hex File ใหม่ โปรแกรมจะจำไฟล์ล่าสุดที่เรียกเปิดใช้งาน ถึงแม้จะเปิดเครื่องคอมพิวเตอร์ใหม่ก็ตาม)
5. ทำการเลือกไฟล์ที่ต้องการ จากนั้นคลิกเมาส์ที่ OK จะปรากฏข้อความที่หน้าต่างการทำงานดังรูป
6. ทำการคลิกเลือก Block ในส่วนของ Erase&Blank
7. จากนั้นคลิกเมาส์ที่ปุ่ม Auto โปรแกรมจะทำงานเป็นขั้นตอนโดยอัตโนมัติ จนหน้าต่างการทำงานปรากฏข้อความดังรูป จึงจะถือว่าเสร็จสิ้นการ โหลดไฟล์





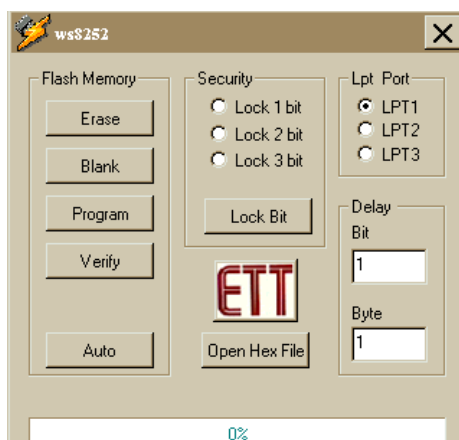
8. คลิกเมาส์ที่ปุ่ม RUN เพื่อทำการดูผลการทำงาน

### ข้อปฏิบัติในการ Download Program

1. ขณะที่มีการใช้งาน RS232 / RS422 / RS485 ใดๆอย่างหนึ่ง ต้องทำการถอดสาย Download RD2 ออกก่อนด้วยทุกครั้ง ในทางกลับกัน หากมีการ Download RD2 ก็ต้องทำการถอดสาย RS232 / RS422 / RS485 ด้วยเช่นกัน เพราะจะทำให้ไม่สามารถทำการ Download ได้

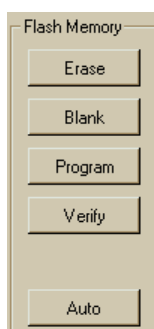
2. ในขณะที่ใช้ IC MAX232 เพื่อเลือกใช้ Line Driver ของ RS232 ต้องไม่เลือก Line Driver RS422/485 คือต้องไม่ใช่ IC SN75176 ทั้ง 2 ตัว ในทำนองเดียวกัน หากทำการเลือก Line Driver RS422/485 จะต้องทำการใส่ IC SN75176 ทั้ง 2 ตัว แต่ต้องไม่ใช่ IC MAX232

## การใช้งานโปรแกรม WS8252



หน้าต่างการทำงานของเบรแกรม WS8252

### Flash Memory



ปุ่ม Erase ใช้สำหรับลบข้อมูลเก่าที่อยู่ใน CPU

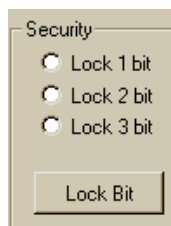
ปุ่ม Blank ใช้สำหรับตรวจสอบการลบข้อมูลเก่าที่อยู่ใน CPU

ปุ่ม Program ใช้สำหรับส่งโปรแกรมที่ทำการโหลดมาจากไฟล์เข้าสู่ CPU

ปุ่ม Verify ใช้สำหรับเปรียบเทียบโปรแกรมที่โหลดมาจากไฟล์กับโปรแกรมที่โหลดเข้าสู่ CPU

ปุ่ม Auto เป็นปุ่มที่สั่งงานให้โปรแกรมทำการตามขั้นตอนโดยอัตโนมัติจนจบกระบวนการ โดยไม่ต้องกดปุ่มเลือกทั้ง 4 ปุ่มข้างต้น เลย

## Security



ในส่วนนี้เป็นส่วนที่สำคัญมากอีกส่วนหนึ่งของการใช้งาน คือเป็นส่วนที่เกี่ยวกับการป้องกันโปรแกรม โดยมีสถานะให้เลือก 3 สถานะดังนี้

Lock bit1 จะไม่สามารถใช้คำสั่ง MOVC กระทำกับหน่วยความจำภายนอกได้ (เมื่อสัญญาณ EA เป็นลอจิก 1) และไม่สามารถ โปรแกรมข้อมูลข้อมูลเข้าไปยังหน่วยความจำแบบ Flash ทั้งใน Parallel Mode และ Serial Mode

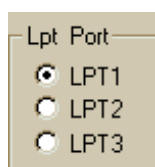
Lock bit2 เหมือนกับการ Lock bit1 แต่สามารถป้องกันการอ่านข้อมูลได้

Lock bit3 เหมือนกับการ Lock bit2 แต่ไม่สามารถปฏิบัติตามคำสั่งจากหน่วยความจำภายนอกได้ไม่ว่า EA จะมีสถานะเป็น 0 หรือ 1

## Open Hex File

เป็นปุ่มที่ใช้สำหรับเลือกไฟล์โปรแกรมมาเก็บไว้ใน Data Buffer

## Lpt Port



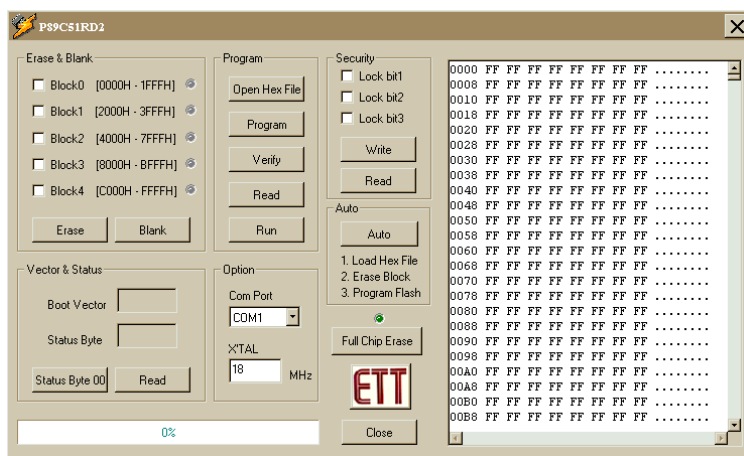
ใช้สำหรับเลือก Printer Port ที่ทำการเชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์

## Delay



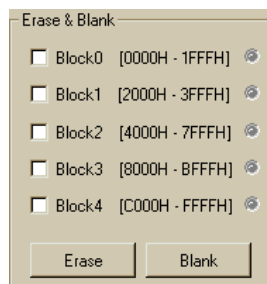
เป็นส่วนที่กำหนดให้ทำการป้อนค่าการหน่วงเวลาของ Bit และ Byte ค่าที่ป้อนมีค่าระหว่าง 1-9 โดยการป้อนค่าทั้ง 2 ค่านี้จะต้องมีความสัมพันธ์กับความเร็วของเครื่องคอมพิวเตอร์ ผู้ใช้สามารถทำการเปลี่ยนแปลงค่าให้สอดคล้องกับความเร็วของเครื่องคอมพิวเตอร์ได้ตามความเหมาะสม

## การใช้งานโปรแกรม P89C51RD2



### หน้าต่างการทำงานของโปรแกรม P89C51RD2

#### Erase&Blank



เป็นส่วนที่ใช้สำหรับการลบข้อมูลเก่าที่มีอยู่ใน CPU ให้ว่างเพื่อเตรียมพร้อมสำหรับการรับข้อมูลใหม่ที่จะ Download เข้าไปภายในตัวมันเองใหม่ โดยมีรายละเอียดดังนี้

ภายใน CPU จะแบ่งพื้นที่ของหน่วยความจำออกเป็น 5 ส่วน คือ Block0 ถึง Block4 โดยสามารถทำการลบข้อมูลเก่าในหน่วยความจำได้โดยการคลิกเลือกที่ช่องว่าง แล้วคลิกที่ Erase เมื่อไฟทางขวามือหยุดกระพริบแสดงว่าทำการลบข้อมูลเก่าใน CPU เสร็จแล้ว ถ้าต้องการเช็คว่าได้ทำการลบข้อมูลเก่าใน CPU หหมดแล้วหรือยัง ให้ทำการคลิกที่ปุ่ม Blank ถ้าทำการลบหมดแล้ว จะปรากฏข้อความว่าที่จอด้านล่างว่า

ในกรณีที่เรากำลังต้องการโหลดโปรแกรมที่มีตำแหน่ง Address อยู่ในช่วง 0000H-1FFFFH ให้ทำการเลือกข้อมูลเก่าใน CPU ในส่วนของ Block0 แต่ถ้ากรณีที่ต้องการโหลดโปรแกรมที่มีตำแหน่ง Address อยู่ในช่วง 0000H-BFFFFH ให้ทำการเลือกข้อมูลเก่าใน CPU ในส่วนของ Block0 ถึง Block3

ปุ่ม Erase ใช้สำหรับลบข้อมูลเก่าที่อยู่ใน CPU

ปุ่ม Blank ใช้สำหรับตรวจสอบการลบข้อมูลเก่าที่อยู่ใน CPU

## Program



Open Hex File ใช้สำหรับเลือกเปิดไฟล์โปรแกรมในรูปแบบของ Hex File ที่ต้องการจะ Download มาใช้งาน

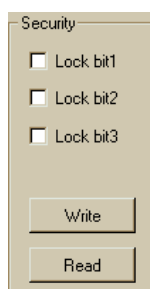
Program ใช้สำหรับนำไฟล์โปรแกรมที่ Download มา ไปเก็บไว้ใน CPU

Verify ใช้สำหรับการเปรียบเทียบโปรแกรมที่แสดงบน Buffer กับโปรแกรมที่อยู่ใน CPU

Read ใช้สำหรับเป็นการอ่านข้อมูลที่อยู่ใน CPU ออกมาไว้ที่ Buffer

Run ใช้สำหรับ Run โปรแกรม

## Security



ในส่วนนี้เป็นส่วนที่สำคัญมากอีกส่วนหนึ่งของการใช้งาน ก็เป็นส่วนที่เกี่ยวกับการป้องกันโปรแกรม โดยมีสถานะให้เลือก 3 สถานะดังนี้

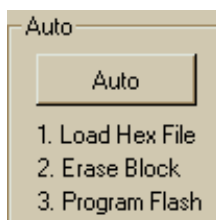
Lock bit1 เป็นการเลือกสถานะของ bit1 คือ ห้ามไม่ให้มีการเขียน โปรแกรมทับ โปรแกรมที่มีอยู่ใน CPU,สามารถทำการอ่านข้อมูลจาก CPU ออกมาได้,สามารถใช้คำสั่ง MovXได้

Lock bit2 เป็นการเลือกสถานะของ bit2 คือ สามารถทำการเขียน โปรแกรมทับ, โปรแกรมที่มีอยู่ใน CPU ได้ , ห้ามไม่ให้ทำการอ่านข้อมูลจาก CPU ออกมาได้ , สามารถใช้คำสั่ง MovXได้

Lock bit3 เป็นการเลือกสถานะของ bit3 คือ สามารถทำการเขียน โปรแกรมทับ โปรแกรมที่มีอยู่ใน CPU ได้ , สามารถทำการอ่านข้อมูลจาก CPU ออกมาได้ , ห้ามใช้คำสั่ง MovX

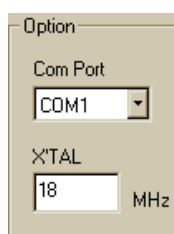
เมื่อทำการเลือก Lock bit ใด bit แล้ว ให้ทำการคลิกที่ปุ่ม Write แต่ถ้าต้องการจะตรวจสอบการทำได้โดยการคลิกที่ Read ซึ่งจะมีการแสดงสถานะว่า ได้ทำการ Lock bit ใดบ้าง

### Auto



ในส่วนของ Auto นี้ จะเป็นส่วนที่ใช้ดำเนินงาน Download Program ได้โดยอัตโนมัติ ตามลำดับขั้นตอนข้อ 1-3 ที่แสดงอยู่ดังกล่าว (วิธีการใช้งานได้ในส่วนของการใช้งานโปรแกรม Download สำหรับบอร์ด CP-SPI/RD2 หน้า 53)

### Option



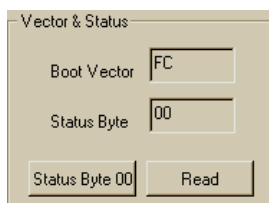
Com P ort ใช้สำหรับเลือก Comport ที่ต่ออยู่กับเครื่อง PC

Xtal ให้ทำการใส่ค่าที่ใช้งานจริงในที่นี้คือ 18.432 MHz (หรือใช้ค่าที่ปรากฏอยู่เดิมขณะเปิดโปรแกรม คือ 18 ก็ได้)

## Full Chip Erase

เป็นปุ่มที่ใช้ลบข้อมูลเก่าทั้งหมดที่อยู่ใน CPU แต่การลบแบบนี้จะทำให้ค่า Status Byte มีค่าเป็น FFH ดังนั้นจึงต้องไปแก้ค่า Status Byte ให้เป็น 00H หลังจากการ Full Chip Erase ทุกครั้ง

## Vector & Status



ส่วนนี้ก็ถือว่าเป็นส่วนที่มีความสำคัญในการใช้งาน โดยมีปุ่มการใช้งานดังนี้

**Boot Vector** โปรแกรมจะกำหนดไว้เป็น FC ซึ่งเป็น โปรแกรมที่จะรอรับคำสั่งที่ถูกส่งออกจากเครื่อง PC (ตำแหน่ง Boot Vector อยู่ที่ FC00H แต่ให้กำหนดเฉพาะค่า FC)

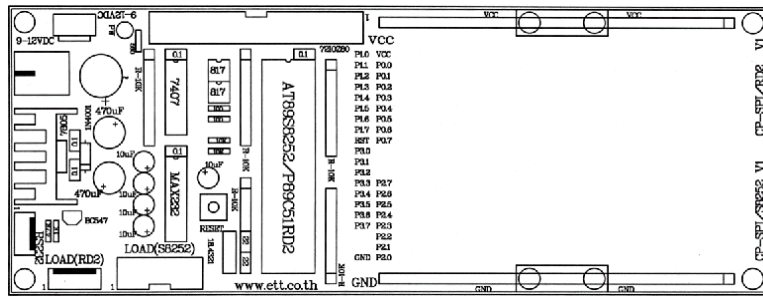
**Status** นั้นเป็นค่าตำแหน่งที่ CPU จะไปทำงานหลังจากถูก Reset ถ้า Status ถูกกำหนดให้เป็น 00H หลังจากที่ CPU ถูก Reset จะทำให้ CPU กระโดดไปทำงานที่ตำแหน่ง 0000H

**ข้อควรระวัง** เมื่อ RUN โปรแกรมแล้วอย่าทำการแก้ไขค่า Boot Vector จากค่า FC ไปเป็นค่าอื่นอย่างเด็ดขาด เพราะถ้าเปลี่ยนไปเป็นค่าอื่นแล้ว จะ RUN โปรแกรมอื่น ๆ ไม่ได้และในกรณีที่เปลี่ยนค่า Boot Vector ไปเป็นค่าอื่นๆ แล้ว สามารถแก้ไขกลับด้วยเครื่อง Copy แบบใช้ไฟสูงเท่านั้น (สามารถนำตัว CPU มาแก้ไข Boot Vector ได้ที่บริษัท อีทีที)

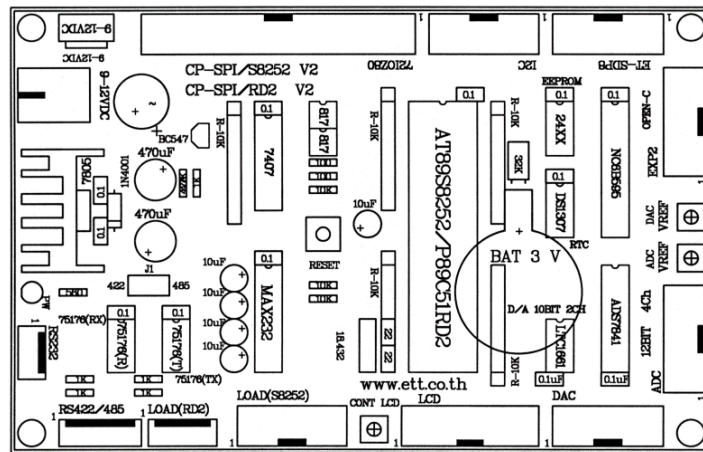
**Close** ใช้สำหรับปิดหน้าต่างการทำงานของโปรแกรม



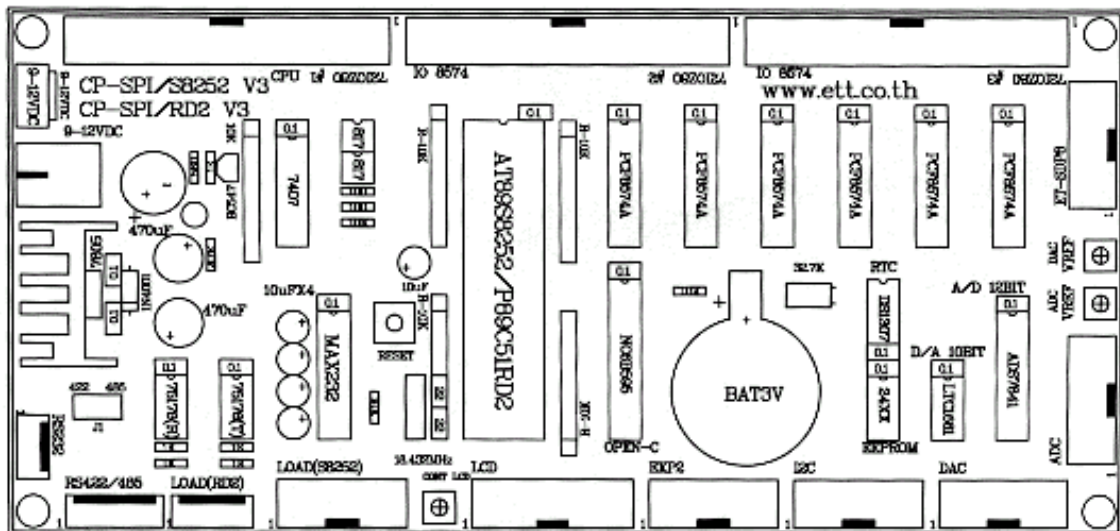
ตำแหน่งการวางอุปกรณ์ของบอร์ด CP-SPI/S8252 V1.0&CP-SPI/RD2 V1.0

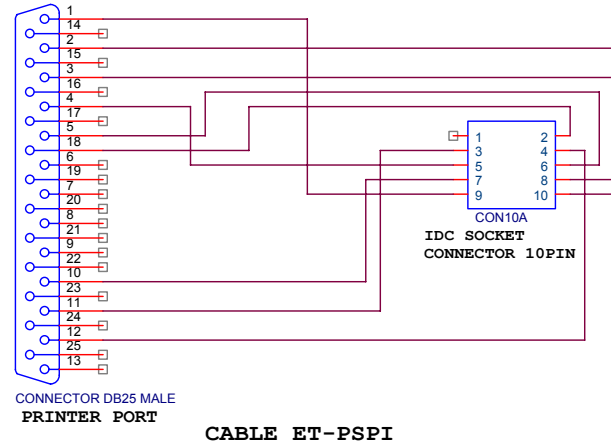
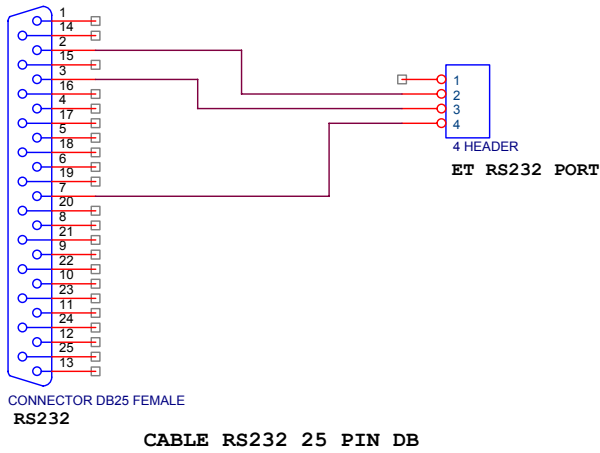
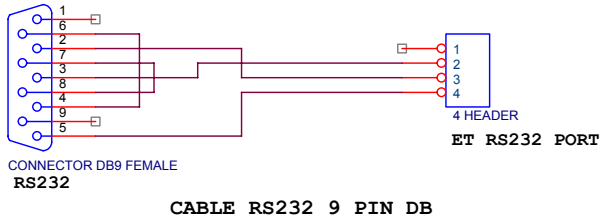
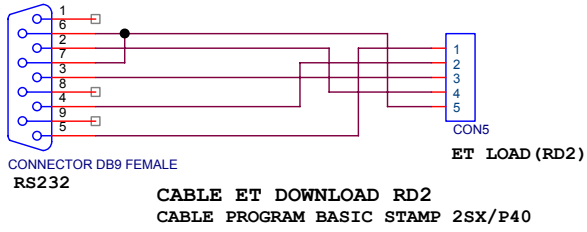


ตำแหน่งการวางอุปกรณ์ของบอร์ด CP-SPI/S8252 V2.0&CP-SPI/RD2 V2.0



ตำแหน่งการวางอุปกรณ์ของบอร์ด CP-SPI/S8252 V3.0&CP-SPI/RD2 V3.0





Title		
ETT CO.,LTD.		
Size B	Document Number CABLE	Rev 1
Date:	Friday, May 17, 2002	Sheet 1 of 1





