

คู่มือการใช้งาน

User's Manual

CP-PIC V3/458/EXP (ICD2)

CP-PIC V3/877/EXP (ICD2)

CP-PIC V4/458 (ICD2)

CP-PIC V4/877 (ICD2)

PIC



**ETT**  
www.etteam.com

บริษัท อีทีที จำกัด ETT CO., LTD.

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

1112/96-98 Sukhumvit Rd., Phrahanong Klongtoey Bangkok 10110

<http://www.ett.co.th>

Tel : 02-7121120 Fax : 02-3917216

email : sale@etteam.com

## ลักษณะโดยทั่วไป

บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 & V4.0 ที่ได้ออกแบบนี้ เป็นบอร์ดที่ออกแบบไว้ใช้งานกับไมโครคอนโทรลเลอร์ในตระกูล PIC โดยจะสามารถใช้ได้ กับเบอร์ 16F877-20P, 18F442 และ 18F458 หรือเบอร์อื่นๆ ที่มีโครงสร้างและตำแหน่งขาสัญญาณเหมือนกันโดย CPU แต่ละเบอร์ก็จะมีคุณสมบัติที่แตกต่างกันซึ่งสามารถสรุปคุณสมบัติของ CPU แต่ละเบอร์อย่างคร่าวๆ ดังตารางด้านล่าง

DEVICE	Program Memory	Data Memory		CAN Module	I/O (Bit)	OSC max (MHz)	Timers	PLL
	Flash	RAM (Bytes)	EEPROM (Byte)					
PIC 16F877	8K (14-Bit Words)	368	256	NO	33	20MHz	3	NO
PIC 18F442	16 Kbyte	768	256	NO	34	40MHz	4	YES
PIC 18F458	32 Kbyte	1536	256	YES	34	40MHz	4	YES

CPU ดังกล่าวจะบรรจุอยู่ในตัวถังแบบ DIP ขนาด 40 ขา และมีทรัพยากรต่างๆบรรจุไว้ในตัว CPU อย่างครบถ้วน ไม่ว่าจะเป็น ADC/TIMER/COUNTER/PWM หรือ PORT I/O ต่างๆ ซึ่งมีความเหมาะสมในการนำไปประยุกต์ใช้งานในลักษณะต่างๆได้เป็นอย่างดี

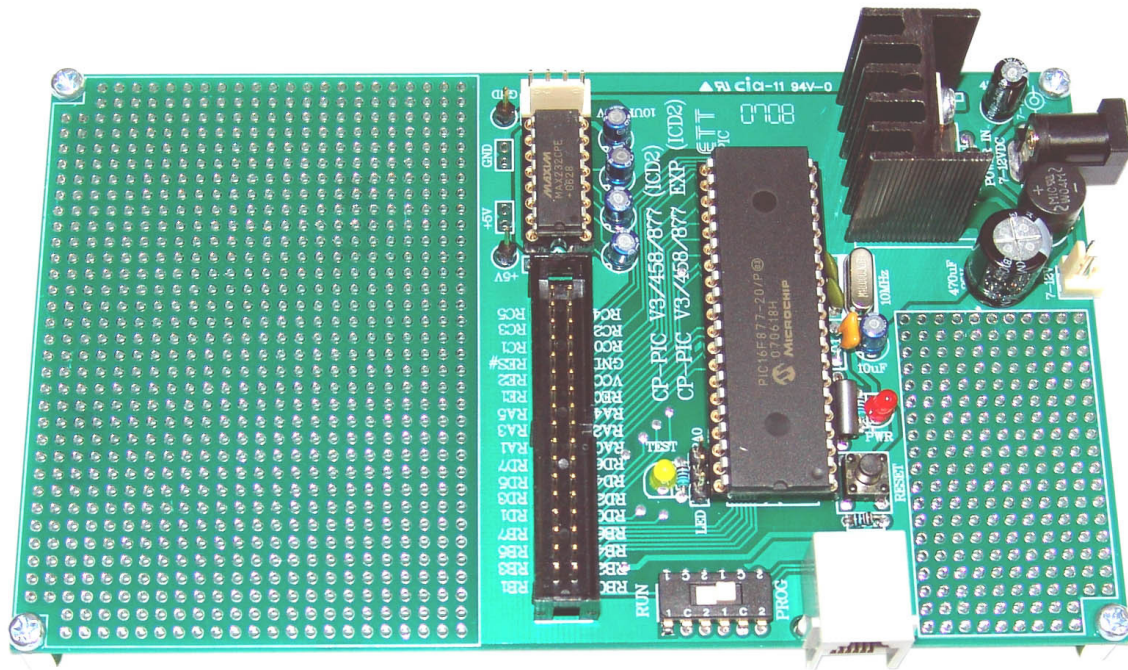
สำหรับอุปกรณ์ I/O ต่างๆ ซึ่งไม่ได้มีบรรจุไว้ในตัว CPU ด้วยทางทีมงาน ETT ก็ได้จัดหาและทำการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ต่างๆที่มีความจำเป็นไว้ให้ด้วยแล้ว ไม่ว่าจะเป็นจอแสดงผลแบบ LCD ระบบฐานเวลา RTC วงจร Line Driver สำหรับการสื่อสารข้อมูลอนุกรมแบบ RS232 และ RS422/485 และยังสามารถให้ผู้ใช้งานเพิ่มเติมอุปกรณ์ I/O อื่นๆเข้าไปได้อีกตามความจำเป็นในการใช้งาน

โดยลักษณะของบอร์ดไมโครคอนโทรลเลอร์ที่ได้ออกแบบนี้ จะแบ่งออกเป็น 3 รุ่น แต่ละรุ่นก็จะมีทรัพยากรสนับสนุนแตกต่างกันไป เพื่อให้สามารถเลือกใช้งานกันตามความเหมาะสมกับงานดังนี้คือ

- **CP-PIC V3.0(ICD2)** เป็นบอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกแบบวงจรเฉพาะส่วนพื้นฐานที่จำเป็น เช่น แหล่งจ่ายไฟ วงจรรีเซ็ต วงจรกำเนิดความถี่สัญญาณนาฬิกา พอร์ตสำหรับ Download โปรแกรม และวงจรสื่อสารอนุกรม ส่วนวงจร I/O ภายนอกนั้น จะไม่ได้จัดเตรียมไว้ให้ด้วย แต่จะทำการต่อสัญญาณ I/O ต่างๆจาก CPU มาไว้ยังหัวต่อ Connector สำหรับให้ผู้ใช้งานไปเชื่อมต่อกับอุปกรณ์ I/O ภายนอกได้โดยง่าย และยังมีพื้นที่เอนกประสงค์สำหรับผู้ใช้งานออกแบบวงจร I/O และต่อวงจร I/O เพิ่มเติมได้เอง เหมาะสำหรับผู้ที่ต้องการนำบอร์ดไปใช้พัฒนางานต้นแบบ โดยการสร้าง I/O ต่างๆขึ้นมาใช้งานเอง ซึ่งในบอร์ดจะประกอบด้วย

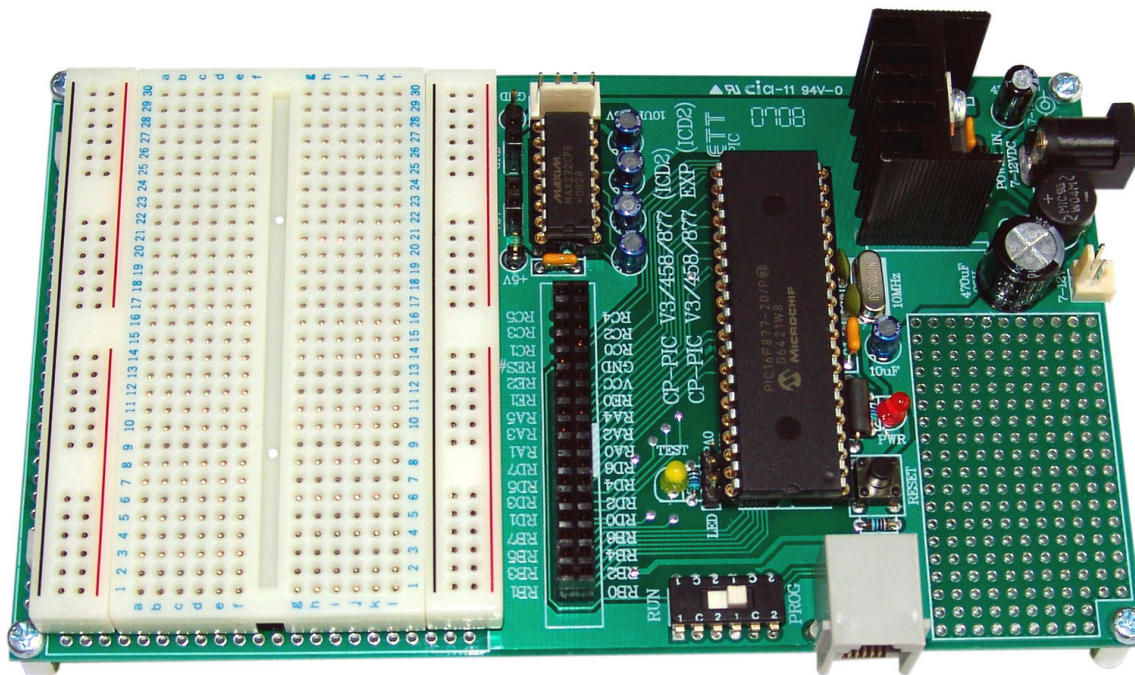
- RS – 232 1 แชนเนล
- ETT CON 34PIN (ET BUS I/O 34PIN)
- 5 Volt Regulator On Board





รูปแสดงบอร์ด CP-PIC V3/458/877 (ICD2)

- CP-PIC V3.0 EXPANSION(ICD2) จะมีลักษณะเดียวกันกับบอร์ด CP-PIC V3.0(ICD2) แต่จะมีแผง Photo Board สำหรับให้ผู้ใช้ต่อทดลองวงจร I/O อย่างง่ายได้เอง เหมาะสำหรับผู้ใช้ที่ต้องการศึกษาเรียนรู้และต้องการทดลองวงจร I/O ต่างๆ ร่วมกับ CPU อย่างง่าย



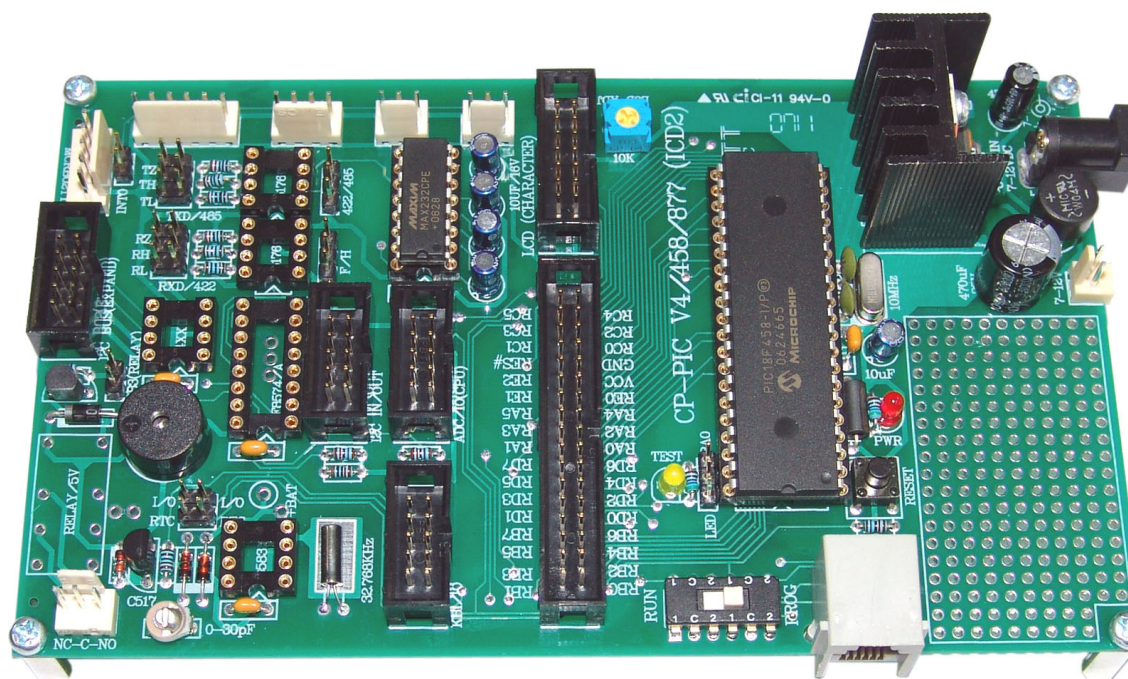
รูปแสดงบอร์ด CP-PIC V3/458/877 EXP (ICD2)

## คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ รุ่น “CP-PIC V3.0 & V4.0(ICD2)”

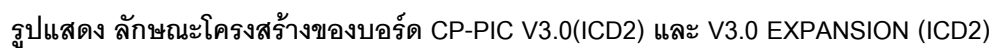
● CP-PIC V4.0 (ICD2) เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ I/O ภายนอกอื่นๆ ที่มีความจำเป็นไว้รองรับการใช้งานในลักษณะต่างๆ เพื่อให้ผู้ใช้งานสามารถนำบอร์ดไปใช้งานในลักษณะงานที่แตกต่างกันได้ โดยไม่ต้องดัดแปลงวงจร หรือ อาจดัดแปลงวงจรเพียงเล็กน้อยสำหรับงานบางอย่าง โดยประกอบด้วยส่วนต่างๆ ดังนี้

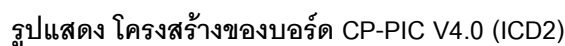
- RS – 232 1 แชนเนล
- RS-422/458 1 แชนเนล (IC 75176 เป็น Option)
- ETT CON 34PIN (ET BUS I/O 34PIN)
- 5 Volt Regulator On Board
- พอร์ตต่อสัญญาณ ICD2 สำหรับเชื่อมต่อสัญญาณโปรแกรมจากภายนอก
- ADC/IO(CPU) พอร์ตสำหรับต่อ อินพุตอนาลอก 8 Channel
- CLCD 14PIN พอร์ตสำหรับต่อ LCD (4 Bit Data)
- RTC #PCF8583P (Option)
- EEPROM ตั้งค่าไบต์ #2432 ขึ้นไป (Option)
- I<sup>2</sup>C IN/OUT เป็น IC ขยายพอร์ต I/O #PCF8574AP (Option)
- KBI/IO 10 Pin สำหรับต่อกับ Keyboard หรือ ใช้เป็น Input /Output Port
- Relay Onboard 5V 1ตัว (Option)
- MCRB02TTL ขั้วต่อ Macnetic Card Reader
- Mini Speaker/Buzzer
- I<sup>2</sup>C BUS(EXPAND)
- PWM1 และ PWM2 ขั้วต่อสำหรับใช้งาน Capture/Compare/PWM

#หมายเหตุ Option คือ ส่วนที่ออกแบบไว้ให้เป็น Socket เปล่าๆหากต้องการใช้งานต้องหาซื้อเพิ่มเอง









สำหรับแหล่งจ่ายไฟของบอร์ด CP-PIC ทั้ง V3.0 และ V4.0 นั้น จะสามารถต่อใช้งานได้ทั้งกับไฟกระแสตรง และกระแสสลับ เนื่องจากในบอร์ดได้จัดเตรียมวงจร RECTIFIER แบบ BRIDGE พร้อมวงจร FILTER และ REGULATOR ขนาด +5V ไว้ให้อย่างครบถ้วนอยู่แล้ว โดยผู้ใช้สามารถป้อนแรงดันไฟตรงหรือไฟสลับที่มีระดับแรงดัน 9 - 12 V โดยสามารถเลือกต่อกับขั้ว CONNECTOR แบบ CPA ขนาด 2 ขา หรือจะต่อผ่านขั้ว CONNECTOR สำหรับ ADAPTER จ่ายไฟก็ได้เช่นกัน โดยการทำงานของแหล่งจ่ายไฟจะมีหลอดแสดงผล LED “VCC” สำหรับแสดงผลการทำงานให้ทราบด้วย

**โหมดการทำงานของบอร์ด**

การทำงานของบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น สามารถกำหนดโหมดการทำงานของบอร์ดได้ 2 โหมดการทำงานด้วยกัน คือ โหมดการโปรแกรม (PROG) และโหมดการทำงานปกติ (RUN)

**การทำงานในโหมดการโปรแกรม (PROG)**

ในโหมดนี้จะใช้สำหรับในกรณีที่ต้องการโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ ในบอร์ด CP-PIC V3.0&V4.0 นี้ ได้ออกแบบ พอร์ต ICD2 สำหรับเชื่อมต่อสัญญาณโปรแกรม (เชื่อมต่อสัญญาณกับเครื่องโปรแกรมภายนอก ET-PGM PIC USB) ในการออกแบบได้ใช้ SLIDE SWITCH เพื่อตัดต่อสัญญาณที่ใช้ในการโปรแกรมหากต้องการทำงานใน โหมดปกติสัญญาณต่างๆ ก็จะถูกแยกออกจากวงจรส่วนของการโปรแกรม ดังนั้นจึงสามารถใช้งานสัญญาณต่างๆ ได้ครบทั้งหมด

การเข้าสู่โหมดของการโปรแกรมทำได้โดยการเลือกตำแหน่ง SLIDE SWITCH (PROG/RUN) มาที่ตำแหน่ง PROG จากนั้นทำการเชื่อมต่อสัญญาณดาวน์โหลด (ICD2 port ) เข้ากับเครื่องโปรแกรม และ ทำการโปรแกรมได้เลย โดยรายละเอียดของการโปรแกรมศึกษาได้จากคู่มือการใช้งานของเครื่องโปรแกรม

RUN



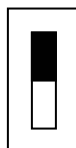
PROG

สวิตช์เลือกโหมด RUN/PROG

**การทำงานใน USER MODE หรือ RUN MODE**

การทำงานในโหมดนี้ คือ การทำให้ CPU กระทำตามคำสั่งต่างๆ ตามโปรแกรมที่เราได้ออกแบบไว้ซึ่งการเข้าสู่โหมดนี้ ทำได้โดยการเลือกตำแหน่ง SLIDE SWITCH (PROG/RUN) มาที่ตำแหน่ง RUN สวิตช์ Slide ก็จะทำให้การแยกสัญญาณต่างๆ ออกจากวงจรในส่วนของการโปรแกรม ฉะนั้นในการใช้งาน I/O Port จึงสามารถนำมาใช้งานได้ครบทั้งหมด

RUN



PROG

## การจัดสรร I/O ของบอร์ด CP-PIC V3.0 &amp; V4.0 (ICD2)

บอร์ด CP-PIC V3.0&V4.0(ICD2) จะใช้ได้กับ CPU เบอร์ 16F877 และ 18F458 ทั้งนี้ขึ้นอยู่กับทางเลือกใช้งาน โดยตัว CPU เหล่านี้จะมีขาสัญญาณที่สามารถนำมาใช้งานเป็น I/O Port ซึ่งในเบอร์ 18F458 จะมี I/O รวมทั้งสิ้น 34 เส้น ส่วน 16F877 มี 33 เส้นประกอบด้วย

- RA0-RA6 จำนวน 7 เส้นสัญญาณ (ส่วน 16F877 จะมีเพียง RA0-RA5 คือ 6 เส้นเท่านั้น)
- RB0-RB7 จำนวน 8 เส้นสัญญาณ
- RC0-RC7 จำนวน 8 เส้นสัญญาณ
- RD0-RD7 จำนวน 8 เส้นสัญญาณ
- RE0-RE2 จำนวน 3 เส้นสัญญาณ

โดยการออกแบบวงจรของบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น ได้พยายามออกแบบวงจรโดยวางโครงสร้างของบอร์ด ให้มีความอ่อนตัวในการใช้งานมากที่สุด เพื่อให้ผู้ใช้งานสามารถนำบอร์ดไปประยุกต์ใช้งานในหลายๆ ลักษณะได้โดยไม่ต้องดัดแปลงโครงสร้างวงจรของบอร์ดไปจากเดิมมากนัก ดังนั้นจึงได้มีการจัดสรรขาสัญญาณ Port I/O ของ CPU ให้สามารถทำงานได้หลายหน้าที่ โดยให้ผู้ใช้สามารถเลือกได้ตามต้องการ ซึ่งหน้าที่การใช้งาน Port I/O ของ CPU ในบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น สามารถสรุปได้ดังต่อไปนี้

**RA0-RA3 และ RA5** ขาสัญญาณเหล่านี้นอกจากจะใช้งานเป็น I/O ปกติได้แล้วยังทำหน้าที่เป็นขาอินพุตของสัญญาณอนาล็อก (AN0-AN4) อีกด้วยดังนั้นเราจึงต่อสายสัญญาณเหล่านี้เข้ากับขั้วต่อ ADC/IO(CPU) เพื่อให้สะดวกต่อการนำไปใช้งาน

**RA4** จะใช้งานในส่วนของแอลซีดี ซึ่งจะต่อเข้ากับขา 6 ของคอนเนคเตอร์ CLCD โดยทำหน้าที่เป็นขา Enable ให้กับแอลซีดี

**RA6/OSC2/CLKO** เป็นขาสัญญาณที่ทำหน้าที่ในหลายส่วน คือ เป็นขา OSC2 และ CLKO จะนำมาใช้เป็นขาสัญญาณ I/O ได้ก็ต่อเมื่อเราใช้คริสตัลลออสซิลเลเตอร์แบบที่เป็นโมดูลสำเร็จสามารถต่อเข้ากับขา OSC1/CLKIN ได้เลยโดยไม่ต้องต่อกับขา RA6/OSC2 ทำให้ ขา RA6 ว่างและนำไปใช้เป็น I/O ได้ แต่ในบอร์ดที่เราออกแบบจะใช้งานขา RA6/OSC2 ร่วมกับ OSC1 ในการรับสัญญาณนาฬิกาจากภายนอกดังนั้น ขา RA6 นี้จึงไม่สามารถต่อออกไปใช้งานได้

**RB0-RB7** สำหรับขาสัญญาณเหล่านี้จะสามารถใช้งานเป็น I/O ได้ปกติ แต่จะมีคุณสมบัติพิเศษคือจะมีวงจรพูลอัพ (Pull-Up) ภายในและยังเป็นแหล่งกำเนิดสัญญาณอินเทอร์รัพต์ต่างๆ ดังนี้

- RB0/INT0 เป็นขาสัญญาณอินเทอร์รัพต์ภายนอก 0
- RB1/INT1 เป็นขาสัญญาณอินเทอร์รัพต์ภายนอก 1
- RB2/INT2 เป็นขาสัญญาณอินเทอร์รัพต์ภายนอก 2
- RB3/INT3 เป็นขาสัญญาณอินเทอร์รัพต์ภายนอก 3 (เฉพาะเบอร์ 18F442)
- RB4-RB7 เป็นขาที่สามารถกำเนิดสัญญาณอินเทอร์รัพต์ได้หากมีการเปลี่ยนแปลงในขาสัญญาณดังกล่าวและมีการ Enable อินเทอร์รัพต์ประเภทนี้ไว้ จึงเหมาะกับการนำไปใช้งานในส่วนของ สวิตช์ คีย์บอร์ด เนื่องจากมีทั้ง อินเทอร์รัพต์และวงจรพูลอัพในตัว



## คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ รุ่น “CP-PIC V3.0 & V4.0(ICD2)”

จากคุณสมบัติดังกล่าวเราจึงจัดสรรการใช้งานดังนี้

RB0/INT0 จะต่อกับขาสัญญาณอินเทอร์รัพท์ของ RTC เบอร์ PCF8583 โดยจะต่อผ่านจัมเปอร์จึงสามารถเลือกที่จะต่อหรือไม่ก็ได้ สามารถเลือกได้โดยการ Shot หรือ Open จัมเปอร์ INT0

RB1/INT1 จะต่อเข้ากับขาสัญญาณอินเทอร์รัพท์ของ PCF8574A และขาสัญญาณอินเทอร์รัพท์ของ Macnetic Card Reader (MCRB02TTL) ทั้งสองส่วนนี้จะต่อผ่านจัมเปอร์ (INT1) ทำให้สามารถเลือกที่จะต่อหรือไม่ต่อก็ได้ การใช้งานจะต้องเลือกใช้งานอย่างใดอย่างหนึ่งเท่านั้น ไม่สามารถใช้งานทั้งสองตัวพร้อมกันได้

RB2 เป็นขาสัญญาณที่ต่อเข้ากับ SPK/BUZZER เพื่อควบคุมการทำงานของ Speaker หรือ Buzzer

RB3 เป็นขาสัญญาณที่ต่อกับวงจรที่ควบคุมการทำงานของรีเลย์(Relay) โดยจะต่อผ่านจัมเปอร์ดังนั้นจึงสามารถเลือกใช้งานหรือไม่ก็ได้โดยการ Shot หรือ Open จัมเปอร์ RB3(RELAY)

RB4-RB7 จะต่อเข้ากับขั้วต่อ KBI/IO สามารถนำไปต่อกับ คีย์บอร์ดประเภท Matrix แบบ 4x4,4x3 หรือ จะใช้เป็น I/O ธรรมดาก็ได้ ในขา RB6 และ RB7 นั้นนอกจากจะต่อกับขั้วต่อ KBI/IO แล้วยังต่อกับสวิตช์ PROG/RUN เพื่อใช้เป็นสัญญาณในการโปรแกรมเมื่ออยู่ในโหมดของการโปรแกรม แต่เมื่ออยู่ในโหมด RUN สามารถนำมาใช้งานเป็น I/O ได้ปกติ

RC0 ขาสัญญาณนี้จะต่อเข้ากับขั้วต่อ แอลซีดี (CLCD) โดยจะต่อเข้ากับขา 4 ของคอนเนคเตอร์ เพื่อทำหน้าที่เป็นขาสัญญาณ RS เพื่อควบคุมการทำงานของ LCD

RC1 เป็นขาสัญญาณที่ต่อเข้ากับขั้วต่อ OC1B เพื่อใช้งานในส่วนของ ขาสัญญาณอินพุตของ Timer 1 หรือ ใช้เป็นขาสัญญาณในส่วนของ Capture2 input /Compare2 Output/PWM2

RC2 เป็นขาสัญญาณที่ต่อเข้ากับขั้วต่อ OC1A เพื่อใช้เป็นขาสัญญาณในส่วนของ Capture1 input /Compare1 Output/PWM1

RC3 สำหรับขาสัญญาณ RC3 จะใช้ทำหน้าที่เป็นขาสัญญาณ SCL ในการติดต่อกับอุปกรณ์ I<sup>2</sup>C Bus และจะต่อเข้ากับ ขั้วต่อ I<sup>2</sup>C EXPAND เพื่อขยายพอร์ต I<sup>2</sup>C BUS

RC4 สำหรับขาสัญญาณ RC4 จะใช้ทำหน้าที่เป็นขาสัญญาณ SDA ในการติดต่อกับอุปกรณ์ I<sup>2</sup>C Bus และจะต่อเข้ากับ ขั้วต่อ I<sup>2</sup>C EXPAND เพื่อขยายพอร์ต I<sup>2</sup>C BUS

RC5 จะใช้เป็นสัญญาณควบคุมการรับส่งข้อมูลในการใช้งาน RS485 โดยจะควบคุมการทำงานของอุปกรณ์ที่เป็น Line Driver ก็คือ IC 75176

RC6 เป็นขาสัญญาณที่ทำหน้าที่ในการส่งข้อมูล (Tx) ในโหมดการสื่อสารอนุกรม RS232,RS422 และ RS485 โดยจะต่อเข้ากับ IC ที่เป็น Line Driver คือ Max 232 และ 75176

RC7 เป็นขาสัญญาณที่ทำหน้าที่ในการรับข้อมูล (Rx) ในโหมดการสื่อสารอนุกรม RS232,RS422 และ RS485 โดยจะต่อเข้ากับ IC ที่เป็น Line Driver คือ Max 232 และ 75176

RD0-RD3 สำหรับขาสัญญาณเหล่านี้จะต่อเข้ากับขั้วต่อ KBI/IO เพื่อใช้งานสำหรับการต่อ คีย์สวิตช์ 4x4 หรือ 4x3 ซึ่งเมื่อใช้งานเป็นคีย์บอร์ดดังกล่าวจะทำงานร่วมกับ พอร์ต RB4-RB7 หรือจะใช้งานเป็น I/O ก็ได้

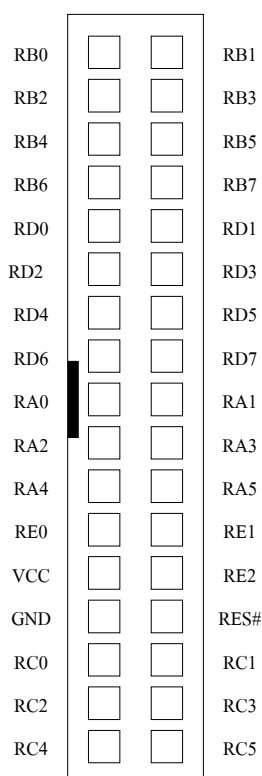
RD4-RD7 ขาสัญญาณเหล่านี้จะทำหน้าที่เป็นขาสัญญาณ Data ที่ใช้ติดต่อกับ LCD โดยจะถูกต่อไปที่คอนเน็คเตอร์ CLCD ซึ่งขั้วต่อ LCD ที่ได้ออกแบบนี้จะเป็นแบบ 4 Bit Data ฉะนั้นในการรับส่งข้อมูลจะทำผ่านสายสัญญาณทั้ง 4 เส้น คือ RD4-RD7

## คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ รุ่น “CP-PIC V3.0 & V4.0(ICD2)”

RE0-RE2 ขาสัญญาณเหล่านี้สามารถใช้งานเป็น I/O ได้ตามปกติ แต่จะมีคุณสมบัติพิเศษคือขาสัญญาณดังกล่าวจะทำหน้าที่เป็นขาอินพุตอนาล็อก (AN5-AN7) เมื่ออยู่ในโหมดของ Analog to Digital โดยเราจะนำไปต่อกับขั้วต่อ ADC/IO(CPU) ทำให้สามารถต่อออกไปใช้งานได้สะดวก

### การใช้งานขั้วต่อ 34PIN (72IOZ80)

สำหรับขั้วต่อ Connector ขนาด 34 PIN ของบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น จะมีอยู่ด้วยกัน 2 แบบ คือ ถ้าเป็นบอร์ด CP-PIC V3.0(ICD2) และ CP-PIC V4.0(ICD2) นั้น จะเป็นขั้วแบบ IDE ขนาด 34 PIN ตัวผู้ ซึ่งขั้วต่อนี้ออกแบบไว้สำหรับให้ผู้ใช้เชื่อมต่อสัญญาณต่างๆของ CPU ออกไปใช้งานกับบอร์ดอื่นๆ ซึ่งอาจเป็นบอร์ดที่ผู้ใช้ออกแบบและสร้างขึ้นเอง หรืออาจใช้บอร์ด I/O ต่างๆที่ ทาง บริษัท อีทีที จำกัด สร้างขึ้นไว้สนับสนุนการใช้งานก็ได้ โดยวิธีการเชื่อมต่อนั้นขอแนะนำให้ใช้สายแพรขนาด 34 PIN จะสะดวกที่สุดเพราะสามารถทำการเชื่อมต่อหรือแยกบอร์ดออกจากกันได้ง่าย ส่วนในกรณีที่ใช้บอร์ดรุ่น CP-PIC V3.0 EXPANSION(ICD2) นั้น ขั้วต่อ 34 PIN จะเป็นแบบ IDE ตัวเมีย เพื่อให้ผู้ใช้สามารถใช้สาย Jumper ต่อสัญญาณต่างๆจากขั้วต่อนี้ไปยังแผงทดลอง Photo Board เพื่อต่อร่วมกับวงจรต่างๆได้โดยง่าย โดยลักษณะการจัดเรียงสัญญาณเป็นดังนี้

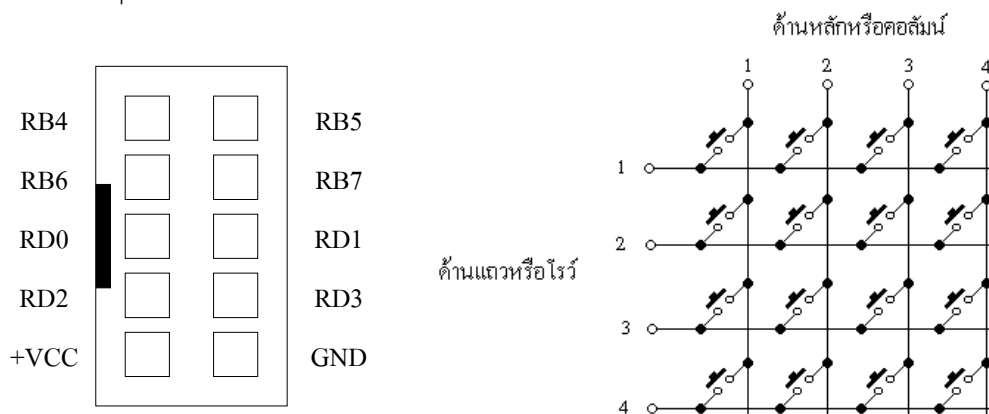


รูปแสดงลักษณะของการจัดเรียงสัญญาณของขั้ว 34PIN

## การใช้งานขั้วต่อ KBI/IO

พอร์ต KBI/IO ถูกจัดไว้ที่ขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมีอยู่เฉพาะในบอร์ด รุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก PORTB(RB4-RB7) และ PORTD(RD0-RD3) ของ CPU ทั้งหมด 8 เส้น ซึ่งเหมาะกับการนำไปประยุกต์ใช้งานในส่วนของการเชื่อมต่อกับวงจรคีย์บอร์ดแบบ Matrix ซึ่งสามารถจะใช้ได้กับคีย์บอร์ดแบบ Matrix ขนาด 4x3 หรือ 4x4 ก็ได้

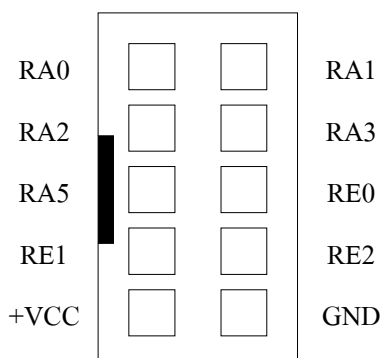
สำหรับในกรณีที่ไม่มีควมจำเป็นต้องใช้งานคีย์บอร์ดแล้ว ขั้วต่อ KBI/IO นี้ก็ยังสามารถนำไปต่อใช้งานเป็น Input หรือ Output ทั่วๆไปได้อีกด้วย



รูปแสดงลักษณะของการจัดเรียงสัญญาณของขั้ว KBI/IO

## การใช้งานขั้วต่อ ADC/IO(CPU)

พอร์ต ADC/IO(CPU) นี้จะถูกเชื่อมต่อออกมาที่ขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมีอยู่เฉพาะบอร์ดในรุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อนี้สามารถนำไปต่อใช้งานเป็นอินพุตของสัญญาณอนาล็อกทั้ง 8 เชนแนล (AN0-AN7) ซึ่งหากไม่ใช้งานในส่วนนี้สามารถใช้งานเป็นอินพุตเอาต์พุตพอร์ตได้ตามปกติ ซึ่งขาสัญญาณต่างๆ ที่นำมาต่อจะเป็นดังรูปด้านล่าง

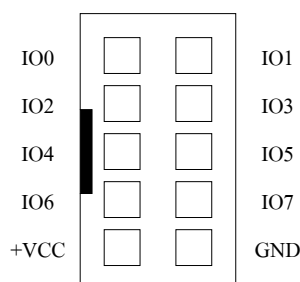


รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้ว ADC/IO(CPU)



### การใช้งานขั้วต่อ I<sup>2</sup>C IN/OUT

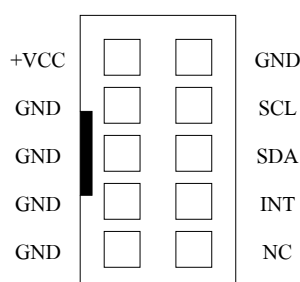
พอร์ต I<sup>2</sup>C IN/OUT นี้จะถูกเชื่อมต่อออกมาที่ขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมียูเฉพาะบอร์ดในรุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อ I<sup>2</sup>C IN/OUT นี้ จะเชื่อมต่อมาจากขาสัญญาณ I/O Port ของ PCF8574/A ซึ่งสามารถโปรแกรมหน้าที่การใช้งาน ให้เป็น Input หรือ Output ก็ได้ตามต้องการจากโปรแกรม แต่ต้องกำหนดหน้าที่ให้เป็น Input หรือ Output อย่างใดอย่างหนึ่งเท่านั้น ไม่สามารถใช้งานทั้งสองหน้าที่พร้อมกันได้ และในการกำหนดหน้าที่ให้เป็น Input หรือ Output ก็ต้องกำหนดให้เหมือนกันทั้ง 8 บิต ด้วย โดยลักษณะของขาสัญญาณที่จัดเรียงไว้ที่ขั้ว I<sup>2</sup>C IN/OUT จะเป็นดังรูป



รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้ว I<sup>2</sup>C IN/OUT

### การใช้งานขั้วต่อ I<sup>2</sup>C BUS EXPAND

พอร์ต I<sup>2</sup>C BUS EXPAND นี้จะถูกเชื่อมต่อออกมาที่ขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมียูเฉพาะบอร์ดในรุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อ I<sup>2</sup>C BUS EXPAND นี้ จะใช้สำหรับการขยายหรือเพิ่มเติมจำนวนอุปกรณ์ที่ใช้การติดต่อสื่อสารแบบ I<sup>2</sup>C ให้กับบอร์ด



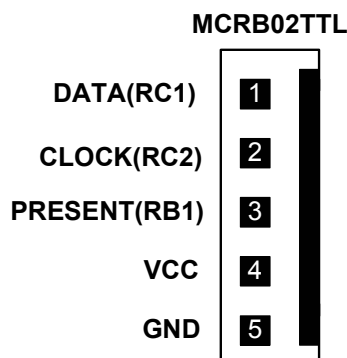
รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้ว I<sup>2</sup>C BUS EXPAND

### การใช้งานเครื่องอ่านบัตรแถบแม่เหล็ก (MAGNETIC-CARD READER)

สำหรับบอร์ด CP-PIC V4.0(ICD2) นั้น จะออกแบบให้สามารถเชื่อมต่อกับเครื่องอ่านบัตรแถบแม่เหล็ก รุ่น “MCR-B02TTL” ได้โดยตรง โดยไม่ต้องดัดแปลงวงจรใดๆทั้งสิ้น โดยในบอร์ดจะจัดเตรียมขั้วแบบ CPA ขนาด 5 PIN ไว้รองรับอยู่แล้ว ผู้ใช้สามารถนำขั้วต่อของเครื่องอ่านบัตรแถบแม่เหล็ก รุ่น “MCR-B02TTL” ของบริษัท อีทีที ต่อเข้าไปได้ทันที สำหรับการเขียนโปรแกรมเพื่อเชื่อมต่อระหว่างบอร์ด CP-PIC V4.0(ICD2) กับเครื่องอ่านบัตรแถบแม่เหล็กนั้น จะสามารถทำได้ 2 แบบ คือ ใช้วิธีการวนรอบตรวจสอบสัญญาณจากเครื่องอ่านบัตรแถบแม่เหล็กเอง ซึ่งวิธีการนี้จะใช้

## คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ รุ่น “CP-PIC V3.0 &amp; V4.0(ICD2)”

สัญญาณจาก CPU เพียง 2 เส้นสัญญาณคือ RC1 ทำหน้าที่เป็น DATA และ RC2 ทำหน้าที่เป็น CLOCK โดยต้องกำหนดคุณสมบัติของสัญญาณทั้ง 2 เส้นให้มีทิศทางเป็น Input ส่วนอีกวิธีหนึ่งคือการใช้วิธีการ Interrupt ซึ่งสามารถทำได้โดยการ SHORT JUMPER “INT1” ที่อยู่ใกล้ๆกับขั้วต่อเครื่องอ่านบัตรแถบแม่เหล็ก ซึ่งการ SHORT JUMPER จะเป็นการเชื่อมต่อสัญญาณอินเทอร์รัพท์ของ CPU เข้ากับ สัญญาณ PRESENT ของเครื่องอ่านบัตรแถบแม่เหล็ก ดังนั้นเมื่อมีการนำบัตรแถบแม่เหล็กไปรูดผ่านเครื่องอ่านบัตรแถบแม่เหล็กก็จะมี การส่งสัญญาณ Interrupt ออกมายัง CPU ซึ่งผู้ใช้ก็เพียงแค่เขียนโปรแกรมสำหรับบริการการ Interrupt ของ INT1 ไว้ก็สามารถใช้งานได้แล้ว



รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้วต่อเครื่องอ่านบัตรแถบแม่เหล็ก MCR-B02TTL

**\*\*\*หมายเหตุ\*\*\*** เนื่องจากการเลือกใช้ Interrupt จากสัญญาณ INT1 ของ CPU นั้น ภายในบอร์ด รุ่น CP-PIC V4.0(ICD2) สัญญาณ INT1 จะถูกออกแบบให้สามารถเลือกกำหนดใช้งานร่วมกับอุปกรณ์หลายๆตัว เช่น การ Interrupt จาก PORT I/O เบอร์ PCF8574/A รวมทั้งการ Interrupt จาก เครื่องอ่านบัตรแถบแม่เหล็ก MCR-B02TTL นี้ ด้วย ดังนั้น ในกรณีที่ต้องการเลือกใช้วิธีการ Interrupt จากเครื่องอ่านบัตรแม่เหล็กนั้น จะต้องทำการ OPEN JUMPER สำหรับเลือกการ Interrupt จากอุปกรณ์อื่นๆที่ไม่เกี่ยวข้องออกเสียก่อน ให้เหลือการเชื่อมต่อสัญญาณ INT1 จาก CPU กับอุปกรณ์เพียงตัวใดตัวหนึ่งเท่านั้น เพื่อให้แน่ใจว่าสัญญาณ Interrupt ที่เกิดขึ้นจะถูกส่งมาออกมาจากเครื่องอ่านบัตรแถบแม่เหล็กเท่านั้น ไม่เช่นนั้นแล้วอาจเกิดความผิดพลาดขึ้นได้

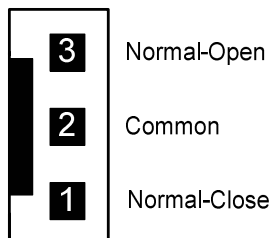
### การใช้งาน OUTPUT RELAY

ภายในบอร์ด CP-PIC V4.0(ICD2) นั้น จะออกแบบวงจรควบคุม RELAY ไว้ให้ผู้ใช้งานสามารถนำไปประยุกต์ใช้งานทั่วไปได้ด้วยจำนวน 1 ชุดโดยวงจร RELAY ดังกล่าวสามารถใช้งานได้ ทั้งหน้าสัมผัสแบบปกติเปิด (Normal Open : NO) และแบบหน้าสัมผัสปกติปิด (Normal Close : NC)

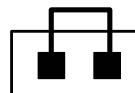
สำหรับสัญญาณ Output ในการควบคุมการทำงานของ RELAY นั้น จะมาจากขา RB3 ของ CPU ซึ่งเมื่อต้องการใช้งาน RELAY จะต้องทำการ SHORT JUMPER RB3(RELAY) ไว้ด้วยเพื่อเชื่อมต่อสัญญาณ RB3 มาทำการควบคุมการทำงานของ RELAY และต้องแน่ใจว่าไม่ได้ต่อสัญญาณ RB3 จากขั้วต่ออื่นๆออกไปใช้งานกับอุปกรณ์ใดๆ นอกเหนือจาก RELAY เนื่องจากสัญญาณ RB3 นั้น นอกจากจะต่อมาไว้ควบคุมการทำงานของ RELAY แล้วยังต่อยังขั้วต่อ 34PIN อีกด้วย โดยการทำงานของ RELAY จะถูกควบคุมการทำงานจากขาสัญญาณ RB3 โดยผู้ใช้งานต้องกำหนดคุณสมบัติของสัญญาณ RB3 ให้ทำหน้าที่เป็น OUTPUT ไว้ด้วย ซึ่งเมื่อขาสัญญาณ RB3 มีสถานะเป็น

OUTPUT และให้สถานะเป็น “1” จะทำให้ RELAY ทำงาน แต่ถ้าสถานะของสัญญาณ RB3 มีค่าเป็น “0” จะทำให้ RELAY หยุดทำงาน

### RELAY



### RB3(RELAY)



รูปแสดง ลักษณะข้อต่อสัญญาณจากหน้าสัมผัสของ RELAY

**\*\*\*หมายเหตุ\*\*\*** เนื่องจากสัญญาณ RB3 ที่นำมาใช้ควบคุมการทำงานของ RELAY จะเป็นสัญญาณเส้นเดียวกับ RB3 ที่ต่อไว้ยังหัว 34PIN ดังนั้น เมื่อต้องการใช้งาน RELAY โดยการควบคุมจาก RB3 แล้ว ต้องแน่ใจว่าไม่ได้ต่อใช้งานสัญญาณ RB3 ในจุดอื่นๆอีก แต่ถ้าหากมีความจำเป็นต้องใช้งาน RB3 พร้อมกับการใช้งาน RELAY ด้วยในเวลาเดียวกัน ก็อาจตัดแปลงวงจรได้โดยการ OPEN JUMPER RB3(RELAY) ออก แล้วใช้วิธีการเชื่อมต่อสายสัญญาณเส้นอื่นๆจาก PORT I/Oของ CPU ที่ไม่ได้ใช้งานมาเข้ากับวงจรควบคุม RELAY แทนก็ได้เช่นเดียวกัน โดยให้เชื่อมต่อสายสัญญาณที่ต้องการไปยัง Jumper RB3(RELAY) ด้านที่ต่อกับตัวต้านทานค่า 1KOhm แต่การตัดแปลงวิธีนี้ควรถอดตัว JUMPER RB3(RELAY) ออกจากบอร์ดเสียก่อน เพื่อจะได้ไม่หลงลืมทำการ SHORT JUMPER นี้ซ้ำอีกในภายหลัง เนื่องจากจะเป็นการ SHORT สัญญาณ RB3 เข้ากับสัญญาณเส้นใหม่ที่บัดกรีมายังวงจร RELAY นี้

### การใช้งานลำโพงขนาดเล็ก หรือ BUZZER

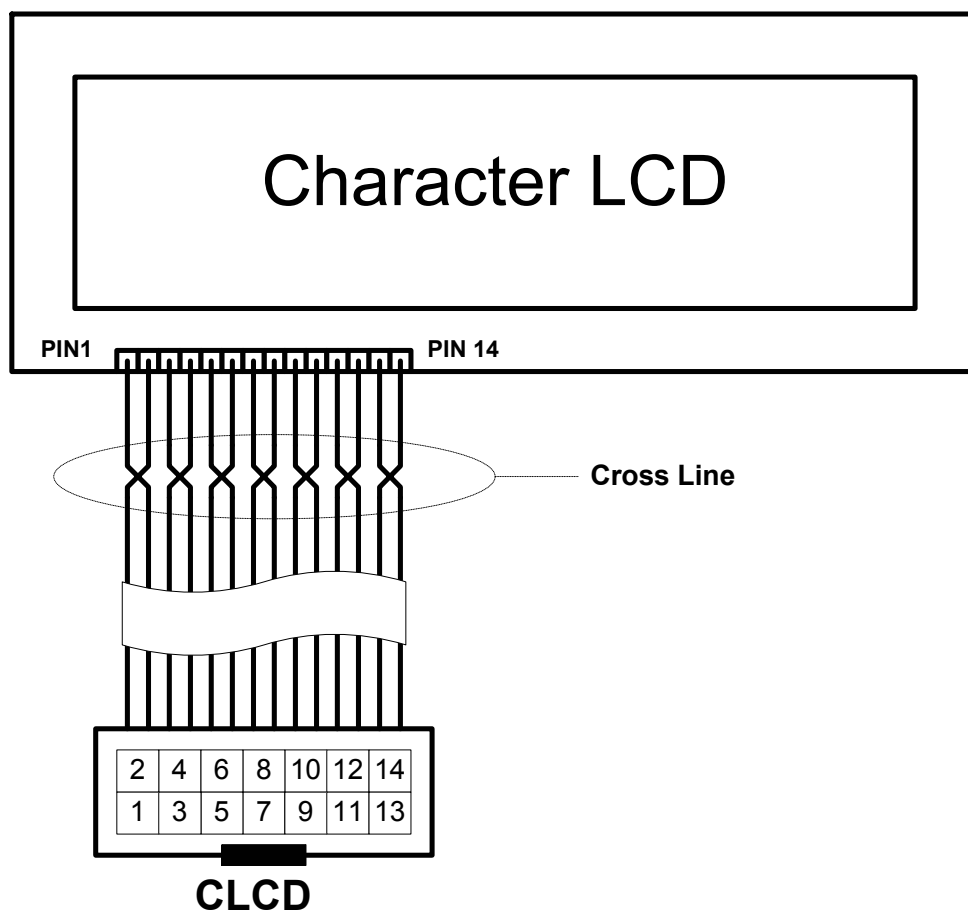
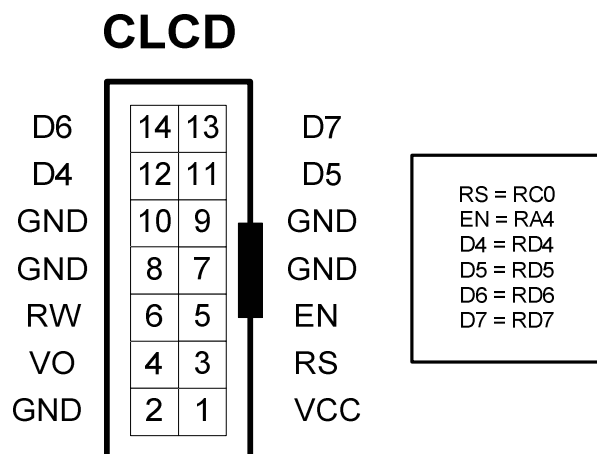
ภายในบอร์ด CP-PIC V4.0(ICD2) จะมีวงจรกำเนิดเสียงรวมอยู่ด้วย 1 จุด ซึ่งในตำแหน่งนี้สามารถเลือกใส่อุปกรณ์กำเนิดเสียงแบบลำโพงขนาดเล็ก หรือ จะเลือกใส่ BUZZER แทนก็ได้เช่นเดียวกัน โดยในกรณีที่เลือกใช้ลำโพงจะมีข้อดีคือ สามารถสร้างความถี่เสียงได้หลากหลายความถี่ตามต้องการแต่การเขียนโปรแกรมจะยุ่งยากกว่า BUZZER เนื่องจากต้องสร้างเป็นสัญญาณความถี่จึงจะสามารถทำให้ลำโพงกำเนิดเสียงให้ได้ ส่วนในกรณีที่เลือกใช้ BUZZER นั้น จะมีข้อดีคือ เขียนโปรแกรมควบคุมการกำเนิดเสียงได้ง่ายกว่าลำโพง เนื่องจากใช้วิธีการ ON หรือ OFF เท่านั้น โดยการ ON บิตควบคุม BUZZER ให้มีสถานะเป็น “1” เท่านั้น BUZZER ก็จะกำเนิดเสียงให้แล้ว แต่ความถี่เสียงของ BUZZER จะไม่สามารถเลือกได้ เหมือนกับลำโพง

สำหรับสัญญาณ Output ในการควบคุมการทำงานของ ลำโพง หรือ BUZZER นั้น จะมาจากขา RB2 ของ CPU ซึ่งการควบคุมการทำงานของลำโพง หรือ BUZZER ผู้ใช้ต้องทำการกำหนดคุณสมบัติของสัญญาณ RB2 ให้ทำหน้าที่เป็น OUTPUT ไว้ด้วย ซึ่งเมื่อขาสัญญาณ RB2 มีสถานะเป็น OUTPUT และให้สถานะเป็น “1” จะทำให้ ลำโพง หรือ BUZZER ทำงาน แต่ถ้าสถานะของสัญญาณ RB2 มีค่าเป็น “0” จะทำให้ ลำโพง หรือ BUZZER หยุดทำงาน



## การใช้งานจอแสดงผลแบบ LCD (Dot-Matrix Character LCD)

บอร์ด CP-PIC V4.0(ICD2) สามารถใช้เชื่อมต่อกับจอแสดงผล LCD แบบ Dot-Matrix โดยเชื่อมต่อผ่านทาง Connector ขนาด 14 PIN และใช้ตัวต้านทานปรับค่าได้แบบเก็อกม้าขนาด 10K สำหรับปรับระดับความสว่างของหน้าจอ LCD โดยวงจรในการเชื่อมต่อ LCD ของบอร์ดนี้จะออกแบบวงจรให้ใช้วิธีการควบคุมการทำงานแบบ "DATA 4-BIT"



รูปแสดง ขาสัญญาณของขั้วต่อ CLCD

## คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ รุ่น “CP-PIC V3.0 & V4.0(ICD2)”

สำหรับวิธีการเชื่อมต่อสัญญาณจากขั้วต่อ LCD ของบอร์ดไปเข้ากับตัว LCD นั้น เพื่อความสะดวกขอแนะนำให้ใช้สายแพร ขนาด 14 เส้น เป็นตัวเชื่อมต่อสัญญาณระหว่างบอร์ดและตัว LCD จะสะดวกมากที่สุด ซึ่งในปัจจุบันพบว่า ลักษณะขั้วสัญญาณที่อยู่ทางด้านของจอแสดงผล LCD เองนั้น ที่พบเห็นได้ทั่วไป จะมีอยู่ด้วยกัน 2 แบบ คือ

- แบบที่เป็นขั้วต่อแบบแถวเดี่ยว ขนาด 14PIN (HEADER 14X1) โดยการต่อสายของ LCD แบบนี้ จะใช้สายแพรขนาด 14 เส้น แบบเข้าหัว CONNECTOR ไว้ด้านเดียว สำหรับเสียบกับขั้วต่อ CLCD ภายในบอร์ด CP-PIC V4.0 ส่วนปลายสายอีกด้านหนึ่งของสายแพรทั้ง 14 เส้น จะต้องนำไปบัดกรีเข้ากับขั้วต่อของตัว LCD ให้ครบทั้ง 14 เส้น โดยในการบัดกรีจะต้องสลับปลายสายเป็นคู่ๆเรียงลำดับกันไป คือ ขา 2 สลับกับ 1, ขา 4 สลับกับ 3... ขา 14 สลับกับ 13 ตามลำดับ กล่าวคือ สายเส้นที่ 1 ต่อกับ PIN2 ของ LCD ส่วนสายเส้นที่ 2 จะต้องต่อกับ PIN1 ของ LCD และในทำนองเดียวกันสายเส้นที่ 3 ก็จะต้องต่อกับ PIN4 ของ LCD อย่างนี้เรื่อยไปจนครบทั้ง 14 เส้น
- แบบที่เป็นขั้วต่อแบบแถวคู่ 14PIN (HEADER ขนาด 7X2) โดยการต่อสายของ LCD แบบนี้ จะใช้สายแพรขนาด 14 เส้น แบบเข้าหัว CONNECTOR ไว้ทั้งสองด้าน โดยในการเชื่อมต่อนั้น ให้ต่อสายแต่ละด้านเข้ากับขั้วต่อ โดยให้ตำแหน่งของ PIN1 ของขั้วต่อแต่ละด้านอยู่ในตำแหน่งที่ตรงกันก็สามารถใช้งานได้แล้ว

**\*\*\*หมายเหตุ\*\*\*** ใน CLCD บางรุ่นนั้น อาจมีขั้วต่อสัญญาณเพิ่มเป็น ขนาด 16 PIN ซึ่งในกรณีนี้ ก็ยังคงใช้วิธีการเชื่อมต่อแบบเดิม คือจะใช้สัญญาณในการเชื่อมต่อระหว่าง CLCD กับบอร์ด เพียงแค่ 14 PIN เท่านั้น ส่วนสัญญาณขา 15 และ 16 ที่เพิ่มเข้ามานั้นจะเป็นขาไฟเลี้ยงของวงจร LED Back-Light (A และ K) ซึ่งถ้า LCD ที่ซื้อมาใช้งานมี LED Back-Light รวมอยู่ด้วย ขอแนะนำให้แยกต่อไฟเลี้ยง LED Back-light เข้ากับแหล่งจ่ายไฟ +5V โดยตรงต่างหากก็ได้ หรือถ้าต้องการให้ LED Back-Light ทำงานตลอดเวลา ก็อาจทำการต่อขาสัญญาณ (A) หรือขา 15 เข้ากับขา 2 ของ LCD ส่วนขา (K) ก็ให้ต่อเข้ากับขา 1 ของ LCD ก็ได้เช่นกัน

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

สำหรับอุปกรณ์แบบ I<sup>2</sup>C Bus ที่ใช้ในบอร์ด CP-PIC V4.0 นั้น จะออกแบบให้สามารถติดตั้งใช้งานอุปกรณ์ I<sup>2</sup>C ได้พร้อมกันในบอร์ดทั้งหมดด้วยกัน 3 ตัว คือ

- I<sup>2</sup>C RTC เบอร์ PCF8583 ของ PHILIPS
- EEPROM ในตระกูล 24XX ซึ่งสามารถเลือกใช้ได้หลายเบอร์หลายผู้ผลิต ขึ้นอยู่กับขนาดความจุของหน่วยความจำที่ต้องการจะใช้ ซึ่งในบอร์ด CP-PIC V4.0(ICD2) นั้น สามารถติดตั้งใช้งานหน่วยความจำ EEPROM แบบ I<sup>2</sup>C นี้ได้ตั้งแต่ เบอร์ 24XX32, 24XX64, 24XX128, 24XX256 หรือ 24XX512 เป็นต้น
- I<sup>2</sup>C I/O เบอร์ PCF8574 หรือ PCF8574A ของ Phillips

โดยอุปกรณ์ I<sup>2</sup>C ทั้ง 3 ตัวนี้จะต่อร่วมกันอยู่ภายในบัสเดียวกัน และใช้สัญญาณ RC3 เป็นขาสัญญาณ SCL และใช้สัญญาณ RC4 เป็นสัญญาณ SDA ในการควบคุมบัส ซึ่ง CPU จะทำหน้าที่เป็นตัวแม่ในการควบคุมบัส นอกจากนี้แล้วยังสามารถขยายอุปกรณ์จำพวก I<sup>2</sup>C นี้ได้อีก แต่ต้องเป็นอุปกรณ์ที่มีรหัสควบคุม Control Word ไม่ซ้ำกันกับอุปกรณ์ที่มีอยู่แล้วภายในบอร์ดด้วยโดยอาจเชื่อมต่อผ่านทางขั้วต่อ “I<sup>2</sup>C EXPANSION” ที่บอร์ดจัดเตรียมไว้ให้แล้วก็ได้

## การใช้งาน Interrupt ของอุปกรณ์ I<sup>2</sup>C

สำหรับสัญญาณ Interrupt จากอุปกรณ์ I<sup>2</sup>C นั้น เนื่องจาก CPU มีสัญญาณ Input ที่ใช้สำหรับรับการ Interrupt จากภายนอก 2 สัญญาณ คือ INT0 และ INT1 ดังนั้น เราจึงได้ออกแบบการต่อใช้งานสัญญาณ Interrupt ดังนี้ คือ

- ต่อสัญญาณการ Interrupt INT1 ให้กับเครื่องอ่านบัตรแม่เหล็ก
- ต่อสัญญาณการ Interrupt INT0 ให้กับ RTC PCF8583
- ต่อสัญญาณการ Interrupt INT1 ให้กับ PCF8574/A

จะเห็นว่าสัญญาณ Interrupt INT1 จะถูกต่อใช้งานถึงสองส่วนคือ เชื่อมต่อกับสัญญาณ Interrupt จากเครื่องอ่านบัตรแม่เหล็ก และ ต่อกับสัญญาณ Interrupt ของ PCF8574/A ดังนั้นในการใช้งานจะต้องเลือกใช้งานอย่างใดอย่างหนึ่งเท่านั้น ซึ่งสามารถทำได้โดยการกำหนดการทำงานให้กับ จัมเปอร์ INT1 เช่น หากใช้งาน Interrupt ของ PCF8574/A ก็ให้ Short Jumper INT1 ของ PCF8574/A และ Open Jumper INT1 ของเครื่องอ่านบัตรแม่เหล็ก MCRB02TTL

## แอดเดรสของอุปกรณ์ I<sup>2</sup>C

เนื่องจากคุณสมบัติของ BUS แบบ I<sup>2</sup>C นั้น สามารถเชื่อมต่ออุปกรณ์ต่างๆที่ใช้วิธีการสื่อสารแบบ I<sup>2</sup>C ได้มากมายหลายตัวภายในบัสเดียวกันได้ เพียงแต่มีข้อแม้ว่า อุปกรณ์ที่จะนำมาต่อร่วมกันภายในบัสเดียวกันนั้น จะต้องมียุทธศาสตร์ในการติดต่อสื่อสาร (Control Byte) ที่ไม่ซ้ำกัน ซึ่งอุปกรณ์บางตัวนั้น ผู้ผลิตได้มีการออกแบบให้สามารถกำหนดค่ารหัสตำแหน่ง Control Byte ได้มากกว่า 1 ค่าเพื่อให้สามารถเชื่อมต่ออุปกรณ์ประเภทเดียวกันร่วมกันภายในบัสเดียวกันได้มากกว่า 1 ตัว โดยใช้วิธีการกำหนดค่าโลจิกให้กับขาสัญญาณสำหรับใช้ระบุตำแหน่ง (Address) ของอุปกรณ์เบอร์นั้นๆได้เอง เช่น I/O Port เบอร์ PCF8574 นั้น สามารถต่อร่วมกันภายในบัสเดียวกันได้มากถึง 8 ตัว และยังสามารถเชื่อมต่ออุปกรณ์ I/O Port ที่มีคุณสมบัติเหมือนกันแต่มีรหัสตำแหน่งที่แตกต่างกันคือ PCF8574A เพิ่มเติมได้อีก 8 ตัว ซึ่งจะเห็นได้ว่าอุปกรณ์ I/O Port นั้นสามารถทำการเพิ่มเติมเข้าไปในระบบบัสเดียวกันได้มากถึง 16 ตัว และในทำนองเดียวกัน หน่วยความจำ EPROM เบอร์ 24LC256 ก็สามารถต่อร่วมกันภายในบัสเดียวกันได้มากถึง 8 ตัว จากตัวอย่างข้างต้นจะเห็นได้ว่าภายในบัสเดียวกันของ I<sup>2</sup>C นั้น อุปกรณ์เพียง 2 ประเภท คือ I/O และ EPROM สามารถต่อร่วมกันภายในบัสเดียวกันได้มากถึง 24 ตัว คือ I/O PCF8574 8 ตัว ,PCF8574A 8ตัว และ 24LC256 อีก 8 ตัว และยังสามารถนำอุปกรณ์ I<sup>2</sup>C อื่นๆที่มีรหัสตำแหน่งของ Control Byte ไม่ซ้ำกันมาต่อเพิ่มเติมได้อีก แต่อย่างไรก็ตามถึงแม้ว่าอุปกรณ์แบบ I<sup>2</sup>C นี้ยอมให้มีการเชื่อมต่อร่วมกันภายในบัสเดียวกันได้หลายตัวภายในระบบบัสเดียวกันก็ตาม แต่ในทางปฏิบัติแล้วอาจเกิดข้อจำกัดในเรื่องของโหลด (FAN-IN/FAN-OUT) เนื่องจากคุณสมบัติของ Port I/O ของ CPU เอง ก็มีข้อจำกัดในการขับเคลื่อนให้กับโหลดได้ประมาณ 25mA เท่านั้น ซึ่งคงไม่สามารถต่ออุปกรณ์ร่วมกันในบัสได้โดยไม่จำกัดจำนวนเหมือนในทฤษฎีบอกไว้ ซึ่งในความเป็นจริงอาจต้องพิจารณาตามความเหมาะสมและความจำเป็นในการใช้งานจริงๆด้วยว่าในระบบบัสหนึ่งของ I<sup>2</sup>C นั้นควรต่ออุปกรณ์ในบัสจำนวนเท่าใด



หน้าที่และเบอร์ ของอุปกรณ์ I <sup>2</sup> C	รหัสตำแหน่งมาตรฐาน ในการอ่าน/เขียน	รหัสตำแหน่งของบอร์ด CP-PIC V4.0	
		รหัสตำแหน่งในการอ่าน	รหัสตำแหน่งในการเขียน
RTC : PCF8583	[1][0][1][0][0][0][X][?]	[1][0][1][0][0][0][1][1]	[1][0][1][0][0][0][1][0]
E <sup>2</sup> PROM:24XX	[1][0][1][0][X][X][X][?]	[1][0][1][0][1][0][0][1]	[1][0][1][0][1][0][0][0]
I/O : PCF8574	[0][1][0][0][X][X][X][?]	[0][1][0][0][0][0][0][1]	[0][1][0][0][0][0][0][0]
I/O : PCF8574A	[0][1][1][1][X][X][X][?]	[0][1][1][1][0][0][0][1]	[0][1][1][1][0][0][0][0]

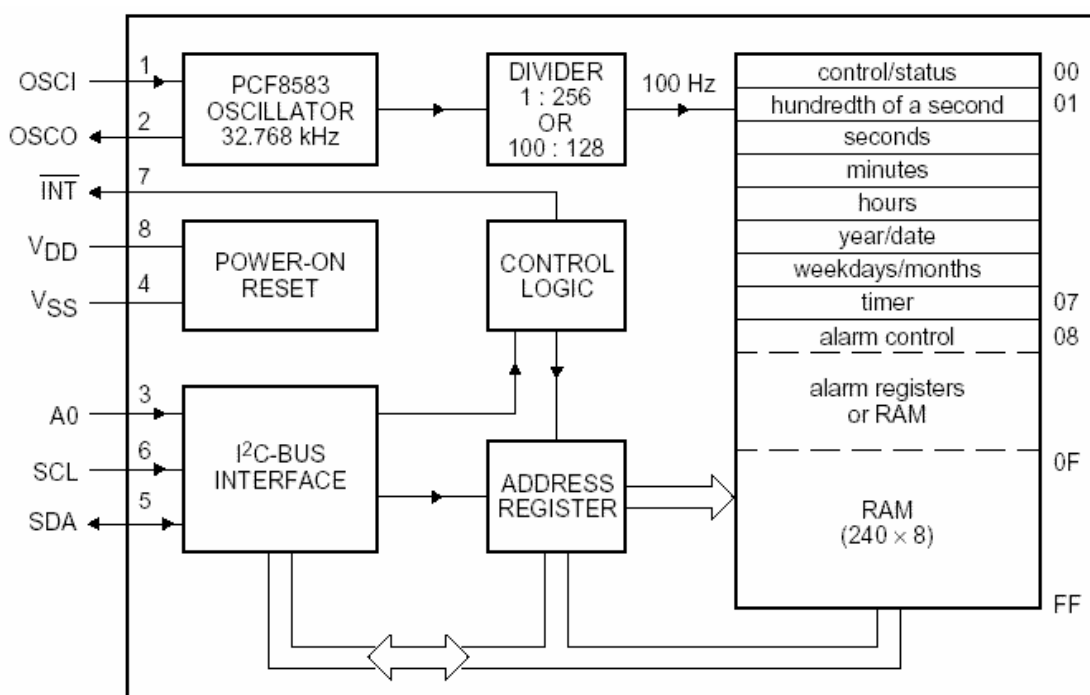
### ตารางแสดง รหัสตำแหน่งของอุปกรณ์ I2C ภายในบอร์ด CP-PIC V4.0

\*\*\*หมายเหตุ\*\*\*

- ค่า X หมายถึงค่าลอจิกของขาสัญญาณ Address ของอุปกรณ์ ที่กำหนดในวงจร
- ค่า ? หมายถึงบิตสำหรับกำหนดว่าต้องการเขียน หรือ อ่าน ข้อมูลกับอุปกรณ์
- เนื่องจาก RTC เบอร์ PCF8583 และ E<sup>2</sup>PROM เบอร์ในกลุ่ม 24XX นั้นมีรหัสตำแหน่ง 4บิตแรก ซ้ำกัน หรือ อยู่ในกลุ่มเดียวกัน ซึ่งในบอร์ด CP-PIC V4.0 นั้นออกแบบให้ RTC เบอร์ PCF8583 มีรหัสตำแหน่งของ Control Byte คงที่เป็น 1010001X ไว้ ส่วน E<sup>2</sup>PROM ก็กำหนดรหัสตำแหน่ง Control Byte ไว้ที่ 1010100X ดังนั้นถ้า ต้องการเพิ่มเติมอุปกรณ์ใดๆเข้าไปอีกต้องกำหนดให้ค่า Control Byte ของอุปกรณ์ที่จะต่อเพิ่มเข้าไปไม่ซ้ำกับค่า Control Byte ทั้งสองดังกล่าวนี้ด้วย

### การใช้งาน I<sup>2</sup>C RTC เบอร์ PCF8583

สำหรับวงจรฐานเวลา RTC นั้น ในบอร์ด CP-PIC V4.0 นั้น จะเลือกใช้ Chips Support ของ PHILIPS เบอร์ PCF8583 ซึ่งเป็นชิพฐานเวลาแบบ I<sup>2</sup>C-Bus และมีฐานเวลาให้ใช้งานอย่างครบถ้วน ตั้งแต่ วินาที/นาฬิกา/ชั่วโมง/วันที่/เดือน/วันในสัปดาห์ และปีคศ. นอกจากนี้ยังมีความอ่อนตัวในการใช้งานค่อนข้างดีเกี่ยวกับระบบเวลา เช่น ค่าของ ชั่วโมงสามารถกำหนดได้จากโปรแกรมว่าจะให้เป็นระบบ 12 ชั่วโมง หรือ 24 ชั่วโมง และในส่วนของวันที่และวันในสัปดาห์ก็สามารถปรับเปลี่ยนได้เองว่า เดือนใดมี 28/29/30 หรือ 31 วันอย่างอัตโนมัติ ซึ่งนอกจากจะใช้งานเป็นฐานเวลา RTC แล้ว PCF8583 นี้ยังมีฟังก์ชันพิเศษในการตั้งเวลาสำหรับเปิดปิดการทำงานของอุปกรณ์ต่างๆ (ALARM FUNCTION) ได้อีกด้วย นอกจากนี้แล้วในตัวของ RTC เองยังมีหน่วยความจำ RAM ขนาด 8บิต จำนวน 240ไบต์ สำหรับให้ผู้ใช้งานไปใช้งานเก็บข้อมูลได้อย่างอิสระ เช่น อาจนำไปใช้ในการเก็บค่าการตั้ง เวลา เพื่อใช้ ตั้งเวลาเปิด-ปิด อุปกรณ์ไฟฟ้า เป็นต้น



รูปแสดงโครงสร้างภายในของ RTC เบอร์ PCF8583

จะเห็นได้ว่า PCF8583 ประกอบขึ้นจากวงจรหลายส่วน เช่น วงจร Power-on Reset วงจรเชื่อมต่อแบบ I<sup>2</sup>C วงจรถอดรหัสตำแหน่งแอดเดรส วงจรหารความถี่ และวงจรกำเนิดความถี่ขนาด 32.768KHz โดยต้องต่อคริสตัลจากภายนอกให้กับขา OSC1 และ OSCO ด้วย สำหรับหน่วยความจำนั้น PCF8583 จะมีโครงสร้างของหน่วยความจำขนาด 8บิต จำนวน 256 ไบท์ โดยจัดสรรสำหรับแบ่งออกเป็นรีจิสเตอร์ของส่วนที่เป็นฐานเวลาจำนวน 16ไบท์(00H-0FH) และใช้งานเป็น หน่วยความจำ RAM ทั่วไปได้อีก 240ไบท์(10H-FFH) ซึ่งในการประยุกต์ใช้งานนั้น ตามปรกติแล้วจะสามารถใช้งานในหน้าที่ของ RTC(Clock Mode) หรือใช้งานเป็นตัวนับ Counter (Event Counter) สำหรับนับความถี่จากขาสัญญาณ OSC1 ก็ได้ แต่สำหรับวงจรของ PCF8583 ภายในบอร์ด CP-PIC V4.0 นั้นจะออกแบบให้ใช้งาน PCF8583 ในโหมด RTC หรือ Clock Mode เท่านั้น

### การติดต่อสื่อสารกับ RTC เบอร์ PCF8583

ในการเขียนโปรแกรมติดต่อกับ RTC นั้น จะใช้วิธีการเชื่อมต่อแบบมาตรฐาน I<sup>2</sup>C Bus โดยใน RTC เบอร์ PCF8583 นี้จะมีตำแหน่งแอดเดรสในการติดต่อภายในบัส หรือ Control Byte เป็น “1010001X” ดังนั้นในการติดต่อกับ RTC ไม่ว่าจะเป็นการเขียนข้อมูลหรืออ่านข้อมูลจากตัว RTC ก็ตามที่ หลังจากสร้างสภาวะเริ่มต้น (Start Condition) แล้วจะต้องส่งค่า Control Byte ของตัว RTC ในบัส ด้วยค่า “1010001X” เพื่อบอกให้ RTC รับรู้ว่า CPU ต้องการจะอ่านหรือเขียนข้อมูลให้กับ RTC จากนั้นจึงส่งรหัส ไบท์แอดเดรส เพื่อระบุตำแหน่งแอดเดรสเริ่มต้นภายในตัว RTC ที่ต้องการจะอ่านหรือเขียนข้อมูลให้กับ RTC เป็นลำดับต่อไป โดยถ้าเป็นตำแหน่งแอดเดรสของฐานเวลาภายในตัว RTC จะมีค่าตำแหน่งแอดเดรสอยู่ระหว่าง 00H-0FH แต่ ถ้าเป็นตำแหน่งแอดเดรสของ RAM ภายในตัว RTC จะมีค่าอยู่ระหว่าง 10H-FFH ตามลำดับ โดยรหัส Control Byte ของ RTC นั้นมีลักษณะโครงสร้างดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
1	0	1	0	0	0	A0	R/W

รูปแสดง โครงสร้างของ Control Byte ของ PCF8583

\*\*\*หมายเหตุ\*\*\* บิต0 (R/W) นั้นจะใช้สำหรับกำหนดว่าจะอ่านหรือเขียนข้อมูลจากอุปกรณ์

ซึ่งจะเห็นว่าตามสภาวะปกติแล้ว Control Byte ของ PCF8583 นั้น สามารถเลือกได้ 2 ค่า โดยการกำหนดลอจิกให้กับขาสัญญาณ A0 ของ PCF8583 เอง ดังนั้นในระบบบัสเดียวกัน จึงสามารถทำการติดตั้ง RTC เบอร์ PCF8583 นี้ได้ 2 ตัว โดยต้องกำหนดให้ขาสัญญาณ A0 ของแต่ละตัวมีสภาวะเป็น “0” และ “1” ซึ่ง ตัวที่ขาสัญญาณ A0 มีสภาวะเป็น “0” ก็จะมีรหัส Control Byte เป็น “1010000X” ส่วนตัวที่ขาสัญญาณ A0 ได้รับสภาวะลอจิกเป็น “1” ก็จะมีรหัส Control Byte เป็น “1010001X” แทน

แต่สำหรับบอร์ด CP-PIC V4.0 นั้น จะกำหนดให้ขาสัญญาณ A0 ของ PCF8583 มีสภาวะทางลอจิกเป็น “1” คงที่ไว้เลย ดังนั้น RTC เบอร์ PCF8583 ในบอร์ด CP-PIC V4.0 นั้นจึงมีรหัส Control Byte คงที่เป็น “1010001X” เสมอ

\*\*\*หมายเหตุ\*\*\* ค่า X หรือ บิต0 ใน Control Byte เป็นบิตสำหรับกำหนดคุณสมบัติในการอ่านหรือเขียนข้อมูลกับอุปกรณ์ I<sup>2</sup>C โดยถ้าหากว่าบิต0 มีค่าเป็น “0” จะหมายถึง CPU ต้องการเขียนค่าไปยังอุปกรณ์ แต่ถ้าค่าในบิต0 มีค่าเป็น “1” จะหมายถึง CPU ต้องการอ่านค่าจากอุปกรณ์ เช่นรหัส Control Byte ของ RTC เบอร์ PCF8583 มีค่า “1010001X” ถ้าต้องการเขียนค่าไปยัง RTC จะต้องส่งรหัส Control Byte เป็น “10100010” แต่ถ้าต้องการอ่านค่าจาก RTC ก็จะต้องส่งรหัส Control Byte ด้วยค่า “10100011” แทน เป็นต้น

## การใช้งานหน่วยความจำ E<sup>2</sup>PROM (24XX)

หน่วยความจำ Serial EEPROM ที่ใช้ในบอร์ดจะใช้การเชื่อมต่อแบบ I<sup>2</sup>C-Bus ในตระกูล 24XX ซึ่งหน่วยความจำแบบนี้มีคุณสมบัติที่น่าสนใจหลายประการคือ มีตัวถังขนาดเล็ก ใช้สัญญาณในการเชื่อมต่อเพียงเส้น และสามารถเก็บรักษาข้อมูลไว้ได้นานกว่า 200 ปี นอกจากนี้ยังสามารถลบและเขียนซ้ำได้ถึง 1 ล้านครั้ง (อ้างอิงจาก Microchip) จึงสามารถนำไปประยุกต์ใช้งาน ในด้านที่เกี่ยวข้องกับการเก็บรักษาข้อมูลสำหรับงานต่างๆได้ดี

โดยผู้ใช้งานสามารถเลือกติดตั้งหน่วยความจำเพื่อใช้งาน กับบอร์ดได้มากมายหลายเบอร์ ขึ้นอยู่กับจุดประสงค์และขนาดของหน่วยความจำที่ต้องการ โดยให้เลือกใช้ E<sup>2</sup>PROM ในตระกูล 24XX (I<sup>2</sup>C Bus) ในกลุ่มที่สามารถกำหนดตำแหน่งรหัส Control Byte ของหน่วยความจำจากฮาร์ดแวร์ (ขาสัญญาณ A2,A1 และ A0) ได้ เช่น เบอร์ 24XX32,64,128 และ 24XX256 ของ Microchip เป็นต้น

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
1	0	1	0	A2	A1	A0	R/W

#### รูปแสดงรหัส Control Byte ของ 24XX32/64/128/256 ของ Microchip

สำหรับหน่วยความจำเบอร์ 24XX32, 24XX64, 24XX128 และ 24XX256 ของ Microchip นั้น จะเห็นได้ว่ารหัส Control Byte ในตำแหน่ง 4 บิตบน (บิต7, 6, 5 และ 4) จะมีค่าเป็น “1010” ส่วน บิต3 บิต2 และ บิต1 นั้นจะขึ้นอยู่กับสถานะทางลอจิกของขาสัญญาณ A2, A1 และ A0 ในวงจร ซึ่งจากคุณสมบัติดังกล่าวจะทำให้สามารถทำการต่อหน่วยความจำดังกล่าวได้มากถึง 8 ตัวภายในระบบบัสเดียวกัน โดยกำหนดสถานะของขา สัญญาณ ลอจิก แอ็คเตวส์ที่แตกต่างกันออกไป โดยสามารถสรุปให้เห็นได้ดังตารางต่อไปนี้

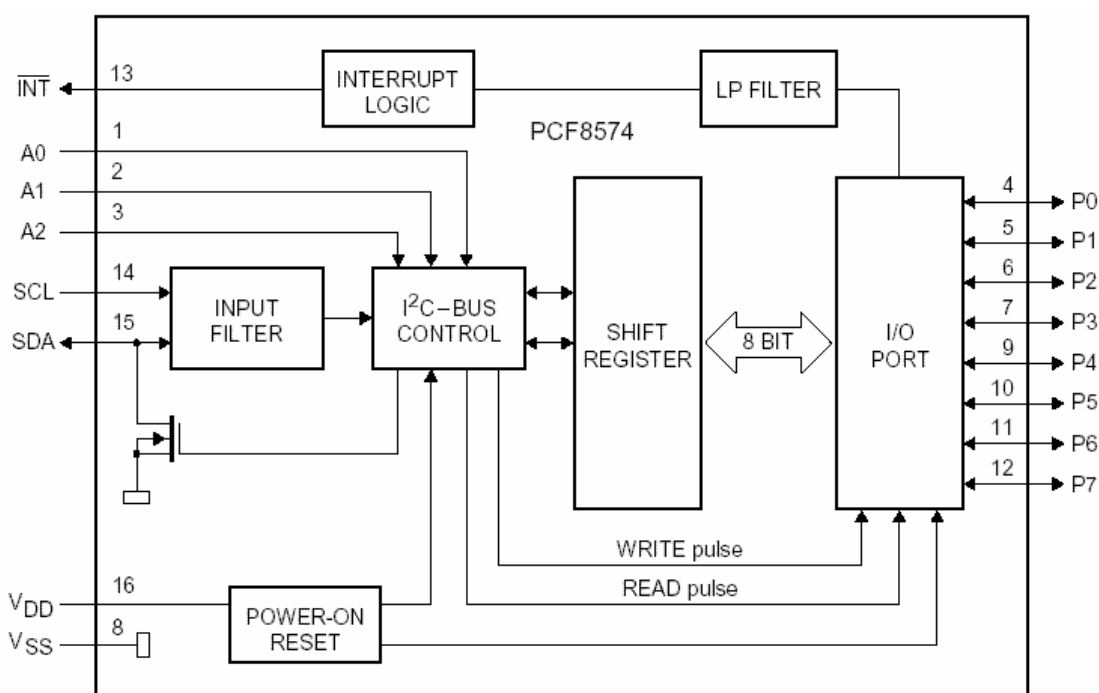
เบอร์(ความจุ)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
24XX32 (4Kx8)	1	0	1	0	A2	A1	A0	R/W
24XX64 (8Kx8)	1	0	1	0	A2	A1	A0	R/W
24XX128 (16Kx8)	1	0	1	0	A2	A1	A0	R/W
24XX256 (32Kx8)	1	0	1	0	A2	A1	A0	R/W

#### ตารางแสดงรหัส Control Byte ของหน่วยความจำแบบ I<sup>2</sup>C Bus ของ Microchip

จากตารางจะเห็นได้ว่า หน่วยความจำ E<sup>2</sup>PROM แบบ I<sup>2</sup>C-BUS นั้น 24XX32/64/128/256 ของ Microchip นั้นจะมีรหัส Control Code ที่เหมือนกันทุกเบอร์ แต่จะมีความแตกต่างกันที่ A0-A2 ดังนั้นเมื่อทำการติดตั้งใช้งานหน่วยความจำเบอร์เหล่านี้กับบอร์ด CP-PIC V4.0 แล้วจะมีรหัส Control Byte เป็น “1010100X” คงที่ตลอด แต่ถ้ามีการต่อหน่วยความจำเหล่านี้เพิ่มเติมจากภายนอกบอร์ดแล้วรหัส Control Byte ก็ขึ้นอยู่กับข้อกำหนดสถานะทางลอจิกให้กับขาสัญญาณ A2, A1 และ A0 ของหน่วยความจำที่ต่อไว้

#### การใช้งาน I/O Port แบบ I<sup>2</sup>C (PCF8574/A)

ตามปกติแล้ว CPU เบอร์ PIC 16F877, 18F442 และ 18F458 นั้นจะมีพอร์ต I/O สำหรับให้ผู้ใช้สามารถนำไปใช้งานได้มากถึง 5 พอร์ต อยู่แล้ว ซึ่งในส่วนของบอร์ด CP-PIC V3.0 นั้น พอร์ต I/O ทั้งหมดของ CPU จะปล่อยให้ผู้ใช้เลือกใช้งานอย่างอิสระตามต้องการ แต่สำหรับบอร์ดรุ่น CP-PIC V4.0 นั้น พอร์ต I/O ต่างๆของ CPU จะถูกจัดสรรออกไปใช้งานในวงจรต่างๆ ดังได้กล่าวอธิบายมาแล้วในข้างต้น แต่ถ้าหากว่าผู้ใช้งานมีความจำเป็นต้องใช้งานพอร์ต I/O จำนวนมาก และจำนวนพอร์ต I/O ของ CPU ที่มีอยู่ไม่เพียงพอต่อการใช้งานแล้ว ผู้ใช้ก็สามารถทำการเพิ่มเติมพอร์ต I/O ได้อีก โดยในบอร์ด CP-PIC V4.0 นั้นจะออกแบบให้ผู้ใช้สามารถทำการเพิ่มเติม พอร์ต I/O แบบ I<sup>2</sup>C ซึ่งมีขนาด I/O จำนวน 8 บิต I/O โดยใช้ไอซี สำหรับทำหน้าที่เป็นพอร์ต I/O ของ Phillips เบอร์ PCF8574 หรือ PCF8574A โดย PCF8574/A มีโครงสร้างดังรูป



รูปแสดง Block Diagram ของ PCF8574/A

นอกจากนี้แล้วผู้ใช้งานยังสามารถทำการขยาย จำนวนพอร์ต I/O ของ PCF8574/A นี้ได้อีกมากถึง 15 ตัว (120 บิต I/O) ทางข้อต่อ “I²C EXPAND” ของบอร์ด เนื่องจาก PCF8574 หรือ PCF8574A นั้น สามารถต่อร่วมกัน ภายในระบบบัสเดียวกันได้มากถึงอย่างละ 8 ตัว กล่าวคือ ในระบบบัสของ I²C นั้น จะสามารถต่อใช้งาน PCF8574 ได้มากถึง 8 ตัว และยังสามารถต่อพอร์ต I/O เบอร์ PCF8574A ได้อีก 8 ตัว รวมเป็น 16 ตัวภายในบัสเดียวกัน โดยการกำหนดตำแหน่งแอดเดรสของอุปกรณ์แต่ละตัวให้มีความแตกต่างกัน ซึ่งตามปกติแล้ว PCF8574 หรือ PCF8574A นั้น จะมีขาสัญญาณแอดเดรสจำนวน 3 เส้น คือ A0,A1 และ A2 โดยการกำหนดสถานะทางโลจิกให้กับขาสัญญาณแอดเดรสทั้ง 3 ให้มีค่าไม่ซ้ำกัน โดย PCF8574 และ PCF8574A นั้น จะมีคุณสมบัติและวิธีการใช้งานที่เหมือนกันทุกประการ แตกต่างกันเพียงรหัส Control Byte เท่านั้น โดยโครงสร้างของรหัส Control Byte ของ PCF8574 และ PCF8574A สามารถแสดงให้เห็นได้ดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	1	0	0	A2	A1	A0	R/W

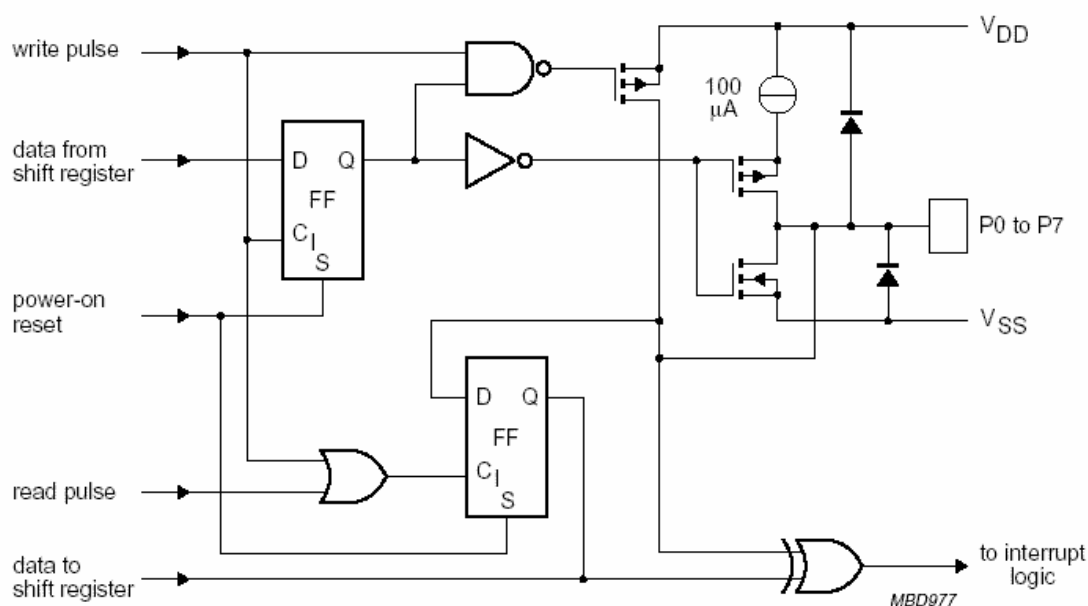
รูปแสดง รหัส Control Byte ของ PCF8574

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	1	1	1	A2	A1	A0	R/W

รูปแสดง รหัส Control Byte ของ PCF8574A

สำหรับรหัส Control Byte ของพอร์ต I/O เบอร์ PCF8574 หรือ PCF8574A ของบอร์ดนั้น จะถูกกำหนดไว้ตายตัว โดยขาสัญญาณแอดเดรส A0,A1 และ A2 จะถูกกำหนดสภาวะลอจิกเป็น “0” ไว้ทั้งหมด ซึ่งในกรณีที่ผู้ใช้ทำการติดตั้งพอร์ต I/O เบอร์ PCF8574 จะมีรหัส Control Byte เป็น “0100000X” แต่สำหรับกรณีที่ผู้ใช้ทำการติดตั้งพอร์ต I/O เบอร์ PCF8574A จะมีรหัส Control Byte เป็น “011000X” แทน

โดยที่ พอร์ต I/O เบอร์ PCF8574/A นั้น ตามปกติแล้วจะสามารถใช้งานเป็น Input หรือ Output ก็ได้ตามต้องการ แต่จำเป็นต้องเลือกกำหนดหน้าที่เพียงหน้าที่เดียวเท่านั้น ไม่สามารถใช้งานเป็นทั้ง Input และ Output ในเวลาเดียวกันได้ โดยลักษณะโครงสร้างภายในของขาสัญญาณ I/O ของ PCF8574 เป็นดังนี้



รูปแสดง ลักษณะโครงสร้างของขาสัญญาณ I/O แต่ละขาของ PCF8574/A



## การใช้งานพอร์ตสื่อสารอนุกรม RS232/RS422/RS485

ภายในตัว CPU เบอร์ PIC 16F877, 18F442 และ 18F458 ที่ใช้กับบอร์ด CP-PIC V3.0&V4.0 นั้น จะมีวงจรสื่อสารแบบอนุกรม (Universal Synchronous Asynchronous Receiver Transmitter : USART) บรรจุรวมไว้ด้วยแล้ว ซึ่งวงจรส่วนนี้ผู้ใช้งานสามารถทำการเขียนโปรแกรมควบคุมการสื่อสารข้อมูลของ CPU กับอุปกรณ์อื่นๆได้ตามต้องการ โดยในส่วนของโปรแกรมนั้น ผู้ใช้สามารถกำหนดรูปแบบของการสื่อสารข้อมูลได้เองจากโปรแกรมที่เขียนขึ้น ไม่ว่าจะเป็นความเร็วในการสื่อสาร (Baudrate) จำนวนบิตข้อมูลในการรับส่ง (Data Bit) การกำหนดบิตตรวจสอบความถูกต้องของข้อมูล (Parity) และคุณสมบัติอื่นๆ ซึ่งในรายละเอียดส่วนนี้จะไม่ขอกล่าวถึงขอให้ผู้ใช้ศึกษาจากคู่มือสถาปัตยกรรมทางฮาร์ดแวร์หรือ Data Sheet ของ CPU เบอร์ ต่างๆ เหล่านี้เอง

ซึ่งปกติแล้วขาสัญญาณสำหรับ รับ-ส่ง ข้อมูลของ CPU นั้น สามารถนำไปเชื่อมต่อกับขาสัญญาณ รับ-ส่ง ของอุปกรณ์อื่นๆได้แล้ว โดยขาส่ง (TX) ของ CPU ต้องนำไปต่อกับขารับ (RX) ของอุปกรณ์ที่จะนำมาสื่อสารกัน ส่วนขา รับข้อมูล (RX) ของ CPU ก็ต้องต่อกับขาส่งข้อมูล (TX) จากอุปกรณ์ที่จะนำมาสื่อสารกัน แต่เนื่องจากขาสัญญาณ RX และ TX ของ CPU นั้น จะสามารถเชื่อมต่อกับสัญญาณที่มีคุณสมบัติเป็นแบบ ระดับลอจิก TTL เท่านั้น ซึ่งถ้าใช้วิธีการ เชื่อมต่อสัญญาณรับส่งของ CPU กับอุปกรณ์โดยตรงนั้น จะสามารถสื่อสารกันได้เพียงระยะทางใกล้ๆหรือภายใน แผงวงจรเดียวกันเท่านั้น ไม่สามารถสื่อสารกันด้วยระยะทางไกลๆได้ ดังนั้นบอร์ด CP-PIC V3.0&V4.0 จึงได้ออกแบบ วงจร Line Driver สำหรับทำหน้าที่เป็น Buffer เพื่อเปลี่ยนแปลงระดับสัญญาณทางไฟฟ้าของขาสัญญาณ รับ-ส่ง ข้อมูลของ CPU ที่เป็น แบบ TTL ให้สามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลมากขึ้น (สามารถอ่านรายละเอียด เพิ่มเติมได้จากหัวข้อ “ความรู้ทั่วไปเกี่ยวกับการสื่อสารอนุกรม” ในภาคผนวกท้ายเล่มของคู่มือนี้) โดยบอร์ด CP-PIC V4.0 นั้น จะสามารถเลือกกำหนดรูปแบบของวงจร Line Driver สำหรับการสื่อสารอนุกรมได้ 3 แบบด้วยกัน คือ

## การสื่อสารอนุกรมแบบ RS232

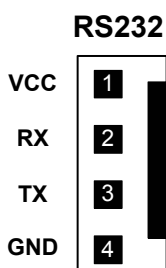
ในกรณีนี้จะต้องทำการติดตั้งไอซี Line Driver เพื่อเปลี่ยนระดับสัญญาณทางไฟฟ้าของขาสัญญาณสำหรับ รับ-ส่ง ข้อมูลแบบ TTL ของ CPU (RX และ TX) ให้เป็นระดับสัญญาณทางไฟฟ้าแบบ RS232 ( $\pm 12V$ ) โดยการติดตั้ง ไอซีเบอร์ MAX232 เพื่อทำหน้าที่เปลี่ยนระดับสัญญาณ TTL จากขาสัญญาณส่งข้อมูล (TX) ของ CPU ให้เป็นระดับ สัญญาณ  $\pm 12V$  สำหรับส่งไปยังขารับสัญญาณ (RX) ของอุปกรณ์ภายนอก และในทางกลับกัน ก็จะทำหน้าที่เปลี่ยน ระดับสัญญาณส่ง (TX) แบบ RS232 ( $\pm 12V$ ) จากอุปกรณ์ภายนอก ให้กลับมาเป็นระดับ TTL เพื่อส่งให้กับขารับข้อมูล (RX) ของ CPU ด้วย โดยเมื่อเปลี่ยนระดับสัญญาณในการรับส่งข้อมูลจาก TTL มาเป็นแบบ RS232 นี้แล้วจะทำให้ สามารถทำการ รับ-ส่ง ข้อมูลกับอุปกรณ์ภายนอกที่ใช้ระดับสัญญาณทางไฟฟ้าในการ รับ-ส่ง แบบเดียวกัน (RS232) ได้ไกลขึ้น ประมาณ 50ฟุต หรือ ประมาณ 15 เมตร โดยสามารถทำการ รับ-ส่ง ข้อมูลกับอุปกรณ์ต่างๆได้ในลักษณะ ของตัวต่อตัว (Point-to-Point) เท่านั้น

สำหรับสายสัญญาณที่จะนำมาใช้สำหรับทำการสื่อสารแบบ RS232 นั้น จะใช้สัญญาณเพียง 2-3 เส้น เท่านั้น ทั้งนี้ขึ้นอยู่กับความต้องการในการสื่อสารว่าต้องการสื่อสารแบบทิศทางเดียวหรือสองทิศทาง

## คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ รุ่น “CP-PIC V3.0 &amp; V4.0(ICD2)”

- การสื่อสาร RS232 แบบสองทิศทาง ซึ่งจะมีทั้งการรับข้อมูลและส่งข้อมูลไปมา ระหว่างด้านรับและด้านส่ง โดยในกรณีนี้จะต้องใช้สายสัญญาณจำนวน 3 เส้น สัญญาณรับข้อมูล (RXD) สัญญาณส่งข้อมูล (TXD) และสัญญาณอ้างอิง (GND) โดยในการเชื่อมต่อสายนั้นจะต้องทำการสลับสัญญาณกับอุปกรณ์ปลายทางด้วย คือ สัญญาณส่ง (TXD) จากบอร์ด CP-PIC V3.0&V4.0 จะต้องต่อเข้ากับสัญญาณรับ (RXD) ของอุปกรณ์ และสัญญาณส่ง (TXD) จากอุปกรณ์ก็ต้องต่อกับสัญญาณรับ (RXD) ของบอร์ด ส่วนสัญญาณอ้างอิง (GND) จะต้องต่อตรงถึงกัน จึงจะสามารถทำการ รับ-ส่ง ข้อมูลกันได้
- การสื่อสาร RS232 แบบทิศทางเดียว ซึ่งอาจเป็นการรับข้อมูลจากด้านส่งเพียงอย่างเดียว หรืออาจเป็นการส่งข้อมูลออกไปยังปลายทางเพียงอย่างเดียว โดยไม่มีการโต้ตอบข้อมูลซึ่งกันและกัน ซึ่งวิธีนี้จะใช้สายสัญญาณเพียง 2 เส้น เท่านั้น โดยถ้าเป็นทางด้านส่ง ก็จะต่อเพียงสัญญาณส่ง (TXD) และสัญญาณอ้างอิง (GND) แต่ถ้าเป็นทางด้านรับ ก็จะต่อเพียงสัญญาณรับ (RXD) และ สัญญาณอ้างอิง (GND) เท่านั้น

โดยหัวต่อของสัญญาณ RS232 ของบอร์ด CP-PIC V3.0&V4.0 นั้น จะเป็นจุดเชื่อมต่อของสัญญาณ รับ-ส่ง ข้อมูล ที่เปลี่ยนระดับสัญญาณเป็นแบบ RS232 แล้ว ซึ่งจะมีลักษณะเป็นแบบหัว CPA ขนาด 4 PIN สำหรับใช้เป็นจุดเชื่อมต่อสัญญาณ รับ-ส่ง ข้อมูลกับอุปกรณ์ภายนอก โดยมีลักษณะการจัดเรียงสัญญาณดังนี้



## แสดงหัวต่อสัญญาณ RS232 ของบอร์ด CP-PIC V3.0 &amp; V4.0

ซึ่งจะเห็นได้ว่าหัวต่อสัญญาณ RS232 ของบอร์ดนั้น จะมีทั้งหมด 4 เส้น แต่ในการ รับ-ส่ง ข้อมูลแบบปรกติ นั้น จะใช้สัญญาณเพียงแค่ 3 เส้น คือ RXD, TXD และ GND เท่านั้น ส่วน +VCC ซึ่งเป็นไฟเลี้ยงวงจร +5V นั้น จะไม่จำเป็นต้องนำมาใช้ในการสื่อสารกันแต่อย่างใด โดย +VCC หรือ +5V นี้ จะออกแบบเพื่อไว้ในกรณีที่อุปกรณ์ปลายทางเป็นวงจรขนาดเล็กและไม่สะดวกที่จะหาแหล่งจ่ายไฟให้กับอุปกรณ์ปลายทางด้วย ก็อาจต่อไฟเลี้ยงวงจร +VCC นี้ ออกไปให้กับอุปกรณ์ปลายทางด้วยก็ได้เช่นกัน

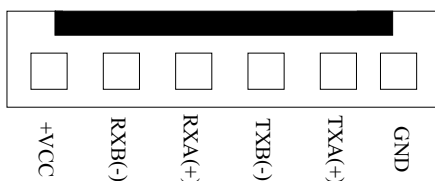
**\*\*\*\*หมายเหตุ\*\*\*\*** สำหรับไอซี Line Drive แบบ RS232 นั้น จะจัดเป็นอุปกรณ์มาตรฐานของบอร์ดในตระกูล CP-PIC V3.0&V4.0 ซึ่งจะมีติดตั้งให้ไปกับบอร์ดอยู่แล้ว ผู้ใช้ไม่ต้องจัดหาเพิ่มเติม แต่พึงระลึกไว้เสมอว่า จะต้องทำการติดตั้งไอซี Line Driver สำหรับเลือกชนิดสัญญาณทางไฟฟ้าของการสื่อสารอนุกรม ได้เพียงอย่างเดียวอย่างหนึ่งเท่านั้น เช่น เมื่อเลือกติดตั้งไอซี Line Driver เป็นแบบ RS232 โดยการติดตั้ง MAX232 ในบอร์ดแล้ว จะต้องไม่ติดตั้งไอซี Line Driver แบบอื่น เช่น RS422 หรือ RS485 เข้าไปด้วย เพราะจะทำให้ไม่สามารถรับส่งข้อมูลกันได้อย่างถูกต้อง ผู้ใช้ต้องเลือกติดตั้งไอซี Line Driver อย่างใดอย่างหนึ่งเท่านั้น

## การสื่อสารอนุกรมแบบ RS422

ในกรณีนี้จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176 หรือ MAX3088 จำนวน 1-2 ตัว เพื่อทำหน้าที่เปลี่ยนระดับสัญญาณการไฟฟ้าในการ รับ-ส่ง แบบ TTL จาก CPU ให้เป็นระดับสัญญาณแบบ Balance Line เพื่อ รับ-ส่งสัญญาณกับอุปกรณ์ที่มีระดับสัญญาณทางไฟฟ้าเป็นแบบ Balance Line เหมือนกัน โดยถ้าต้องการใช้การสื่อสารแบบ 2 ทิศทาง ก็จะต้องติดตั้งไอซี Line Driver จำนวน 2 ตัว โดยแบ่งเป็นตัวแปลงสัญญาณทางด้านรับ 1 ตัว และตัวแปลงสัญญาณด้านส่งอีก 1 ตัว แต่ถ้าต้องการสื่อสารแบบทิศทางเดียวก็อาจทำการติดตั้งไอซี Line Driver เพียงตัวเดียว โดยถ้าต้องการให้เป็นฝ่ายรับข้อมูลเพียงอย่างเดียวก็ให้ติดตั้งไอซี Line Driver เฉพาะในตำแหน่งของ “RXD/422” เพียงตัวเดียว แต่ถ้าต้องการให้เป็นฝ่ายส่งข้อมูลเพียงอย่างเดียวก็ให้ทำการติดตั้งไอซี Line Driver เฉพาะในตำแหน่ง “TXD/485” เพียงตัวเดียวเท่านั้น

ซึ่งการสื่อสารแบบ RS422 นี้ สามารถนำไปทดแทนการสื่อสารแบบ RS232 ได้ทันที โดยไม่ต้องดัดแปลงหรือแก้ไขโปรแกรมเลย ซึ่งการสื่อสารโดยใช้ระดับสัญญาณในการ รับ-ส่ง แบบ RS422 นี้จะมีข้อดี คือสามารถทำการสื่อสารกันได้ในระยะทางที่ไกลขึ้นกว่าแบบ RS232 มาก กล่าวคือ สามารถจะทำการ รับ-ส่ง ข้อมูลกันได้ในระยะทางประมาณ 4000 ฟุต หรือ 1200 เมตร หรือ 1.2 กิโลเมตรเลยทีเดียว เพียงแต่ต้องใช้สายสัญญาณที่ออกแบบมาสำหรับรองรับการใช้งานในด้านการสื่อสารแบบนี้โดยเฉพาะ ซึ่งได้แก่ สายสัญญาณแบบ UTP (Un-Shielded Twist Pair) หรือ STP (Shielded Twist Pair) โดยการสื่อสารด้วยระดับสัญญาณทางไฟฟ้าแบบ RS422 นี้ ถ้าเป็นการสื่อสารแบบ 2 ทิศทาง คือ ทั้งรับข้อมูลและส่งข้อมูล จะสามารถทำการรับส่งข้อมูลกับอุปกรณ์ต่างๆได้ในลักษณะของตัวต่อตัว (Point-to-Point) เหมือนกับ RS232 ทุกประการ แต่ในกรณีที่เป็นการสื่อสารแบบทิศทางเดียวนั้น สามารถจะทำการต่อขนานสัญญาณทางด้านรับ จำนวนหลายๆจุด เข้ากับสัญญาณส่งเพียงจุดเดียวได้ โดยถ้าเลือกใช้อิซี Line Driver เบอร์ 75176 จะสามารถต่อขนานจำนวนอุปกรณ์สำหรับด้านรับข้อมูลได้ประมาณ 32จุด แต่ถ้าเลือกใช้อิซี Line Driver เบอร์ MAX3088 นั้น จะสามารถต่อขนานจำนวนอุปกรณ์ทางด้านรับข้อมูลได้มากถึง 256 จุด เลยทีเดียว แต่ถ้าเป็นอุปกรณ์ทางด้านส่งนั้น จะไม่สามารถนำมาต่อขนานสัญญาณส่งข้อมูลเข้าด้วยกันมากกว่า 1 จุด เหมือนทางด้านฝ่ายรับได้ ซึ่งวงจร Line Driver แบบ RS422 นี้จะมีอยู่เฉพาะในบอร์ดรุ่น CP-PIC V4.0 เท่านั้น

สำหรับลักษณะของหัวต่อของสัญญาณ RS422 นั้น จะเป็นแบบ CPA ขนาด 6 PIN ดังรูป โดยในการสื่อสารกันนั้น จะใช้สายสัญญาณในการ รับ-ส่ง ข้อมูลกัน จำนวน 4 เส้น สัญญาณ คือ สัญญาณในการรับข้อมูล จำนวน 2 เส้น คือ RXA (RX+) และ RXB (RX-) และสัญญาณในการส่งข้อมูลอีก 2 เส้น คือ TXA (TX+) และ TXB (TX-) ซึ่งในการต่อสัญญาณนั้น จะต้องทำการต่อสัญญาณในลักษณะของการสลับกัน คือ สัญญาณส่งจะต้องต่อเข้ากับสัญญาณรับ นั่นก็คือ สัญญาณ RXA (RX+) จะต้องต่อกับ TXA (TX+) ส่วน RXB (RX-) ก็จะต้องต่อกับ TXB (TX-) ด้วยเช่นกัน โดยลักษณะของหัวต่อสัญญาณ RS422 เป็นดังรูป



แสดงหัวต่อสัญญาณ RS422/485 ของบอร์ด CP-PIC V4.0 เมื่อเลือกเป็น RS422

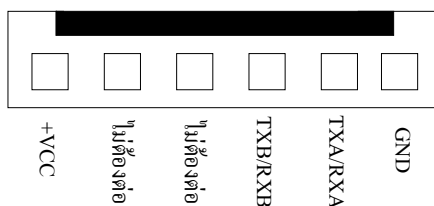
## การสื่อสารอนุกรมแบบ RS485

ในการสื่อสารแบบ RS485 นี้จะมีคุณสมบัติของสัญญาณทางไฟฟ้าเหมือนกับ RS422 ทุกประการ เพียงแต่ว่าในการสื่อสารแบบ RS485 นี้จะใช้สายสัญญาณในการรับส่งข้อมูลกันเพียง 2 เส้น เท่านั้น แต่จะมีความพิเศษกว่าแบบ RS422 ตรงที่ ทิศทางของสัญญาณจะสามารถปรับเปลี่ยนได้จากโปรแกรม กล่าวคือ สัญญาณทั้ง 2 เส้น นี้สามารถจะสลับหน้าที่เป็นด้านส่ง และ เป็นด้านรับได้ ตามต้องการ โดยการควบคุมจาก CPU โดยจากบอร์ด CP-PIC V4.0 นั้น จะกำหนดให้สัญญาณ RC5 ทำหน้าที่สำหรับควบคุมทิศทางของข้อมูลว่าจะให้เป็นรับหรือส่ง โดยถ้าควบคุมให้ RC5 มีสถานะเป็น “1” จะเป็นการกำหนดทิศทางให้เป็นฝ่ายส่งข้อมูล แต่ถ้าสถานะของ RC5 เป็น “0” จะเป็นการกำหนดทิศทางให้เป็นฝ่ายรับข้อมูล ซึ่งจากคุณสมบัติข้อนี้จะทำให้การสื่อสารแบบ RS485 สามารถทำการต่อขนานอุปกรณ์ร่วมกันในสายส่งเดียวกันได้จำนวนหลายๆจุด โดยถ้าใช้ไอซี Line Driver เบอร์ 75176 จะสามารถต่อขนานอุปกรณ์กันได้จำนวน 32 จุด แต่ถ้าเลือกใช้อิซี Line Driver เบอร์ MAX3088 แล้วจะสามารถต่อขนานอุปกรณ์ในสายคู่เดียวกันได้มากถึง 256 จุด เลยทีเดียว แต่มีข้อแม้ว่า เมื่อมีการต่ออุปกรณ์ขนานกันในสายสัญญาณคู่เดียวกันมากกว่า 2 จุดแล้ว จะต้องเขียนโปรแกรมควบคุมให้มีการส่งข้อมูลออกมาในสายครั้งละ 1 จุดเท่านั้น เพราะถ้ามีการกำหนดทิศทางของข้อมูลให้เป็นส่งในเวลาเดียวกันมากกว่า 1 จุดแล้วจะทำให้เกิดการชนกันของข้อมูลและไม่สามารถสื่อสารกันได้อย่างถูกต้อง

โดยเมื่อต้องการใช้วิธีการสื่อสารแบบ RS485 นี้ จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176 หรือ MAX3088 ในตำแหน่งของ “TXD/485” เพียงตัวเดียว พร้อมกับเลือกกำหนดเป็นแบบ RS485 ดังนี้

- ทำการเลือก Jumper สำหรับเลือก “422/485” ไว้ทางด้าน 485 (RS485)
- ทำการเลือก Jumper “F/H” ไว้ทางด้าน H (Half Duplex)
- ทำการ Short Jumper สำหรับต่อตัวต้านทาน Fail Safe Resister คือ “TL” ไว้
- ทำการ Short Jumper สำหรับต่อตัวต้านทาน Fail Safe Resister คือ “TH” ไว้
- สายสัญญาณที่ใช้จะต่อจาก TXB(TX-) และ TXA(TX+) เพียง 2 เส้น ออกไปใช้งาน

ซึ่งในการสื่อสารข้อมูลแบบ RS485 นี้ จะต้องเขียนโปรแกรมขึ้นมารองรับการสื่อสารโดยเฉพาะ เนื่องจากทิศทางของข้อมูลสามารถจะกำหนดจากโปรแกรมได้โดยตรง ซึ่งการสื่อสารวิธีนี้จะมีข้อดีคือ ใช้สายสัญญาณในการรับส่งน้อยเส้น แต่จะเสียเวลาในการสื่อสารมากกว่าวิธีอื่นๆ เนื่องจากว่า การสื่อสารแบบนี้จะไม่สามารถทำการรับและส่งข้อมูลในเวลาเดียวกันได้ แต่จะต้องใช้วิธีการ ผลัดกันรับ ผลัดกันส่ง แทน ซึ่งในความเป็นจริงแล้วในปัจจุบันนี้ ราคาของสายสัญญาณแบบ 2 เส้น และ 4 เส้น แทบจะไม่มี ความแตกต่างกันเลย ดังนั้นเพื่อลดความยุ่งยากในการเขียนโปรแกรมสำหรับควบคุมการรับส่งข้อมูลของ CPU ขอแนะนำให้เลือกใช้วิธีการสื่อสารแบบ RS422 จะง่ายและสะดวกเร็วกว่ากันมาก



แสดงขั้วต่อสัญญาณ RS422/485 ของบอร์ด CP-PIC V4.0 เมื่อเลือกเป็น RS485

### การกำหนด Jumper สำหรับการสื่อสารแบบ RS422/485

เนื่องจากวงจร Line Driver ของพอร์ตสื่อสารอนุกรมของบอร์ดนั้น ออกแบบให้ผู้ใช้สามารถเลือกกำหนดได้หลายแบบ ดังนั้น จึงต้องมีการใช้ Jumper สำหรับเป็นตัวเลือกรูปแบบการสื่อสารร่วมด้วย โดยจะมี Jumper ที่เกี่ยวข้องกับการใช้งานการสื่อสารแบบ RS422 และ RS485 ดังต่อไปนี้ คือ

- Jumper 422/485 เป็น Jumper สำหรับเลือกกำหนดการทำงานของไอซี Line Driver ในตำแหน่ง TXD/485 ให้ทำงานเป็นแบบ RS422 หรือ RS485 โดยถ้าต้องการให้เป็นแบบ RS422 จะต้องกำหนด Jumper ไว้ทางด้าน “422” ซึ่งจะทำให้ไอซี Line Driver ตำแหน่ง “TXD/485” ทำหน้าที่เป็นฝ่ายส่งข้อมูลเพียงอย่างเดียว แต่ถ้าต้องการใช้งานแบบ RS485 จะต้องกำหนด Jumper ไว้ทางด้าน “485” เพื่อกำหนดให้ไอซี Line Driver ในตำแหน่ง “TXD/485” ทำหน้าที่เป็นทั้งฝ่ายรับและฝ่ายส่ง ตามการควบคุมของสัญญาณ RC5
- Jumper F/H (Full/Half) เป็น Jumper ใช้สำหรับเลือกกำหนดรูปแบบการสื่อสารให้เป็นแบบ Full Duplex (F) หรือ Half Duplex (H) โดยถ้าต้องการใช้งานแบบ RS422 จะต้องเลือกกำหนด Jumper นี้ไว้ทางด้าน F(Full Duplex) แต่ถ้าต้องการใช้งานแบบ RS485 จะต้องเลือกกำหนด Jumper นี้ไว้ทางด้าน H(Half Duplex) แทน
- Jumper RL เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ RXB (RX-) หรือ Fail Safe Resister เพื่อให้สัญญาณ RXB (RX-) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกลๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- Jumper RH เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ RXA (RX+) หรือ Fail Safe Resister เพื่อให้สัญญาณ RXA (RX+) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกลๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- Jumper RZ เป็น Jumper สำหรับเลือกกำหนดการต่อตัวต้านทาน RZ เพื่อชดเชย ค่าความต้านทานของสายสัญญาณ (Impedance) ทางด้านรับ ซึ่งถ้าหากว่ามีการต่อสายสัญญาณในการรับส่งเป็นระยะทางไกลๆแล้วก็ควรทำการ Short Jumper นี้ไว้ด้วยเนื่องจากเมื่อสายมีความยาวมากๆจะเกิดค่าความต้านทานในสายขึ้น ดังนั้นจึงต้องทำการต่อค่าความต้านทานจากภายนอกไปชดเชยค่าความต้านทานของสายสัญญาณด้วย โดยเมื่อทำการ Short Jumper ตำแหน่ง RZ นี้ไว้ก็จะเป็นการต่อตัวต้านทาน

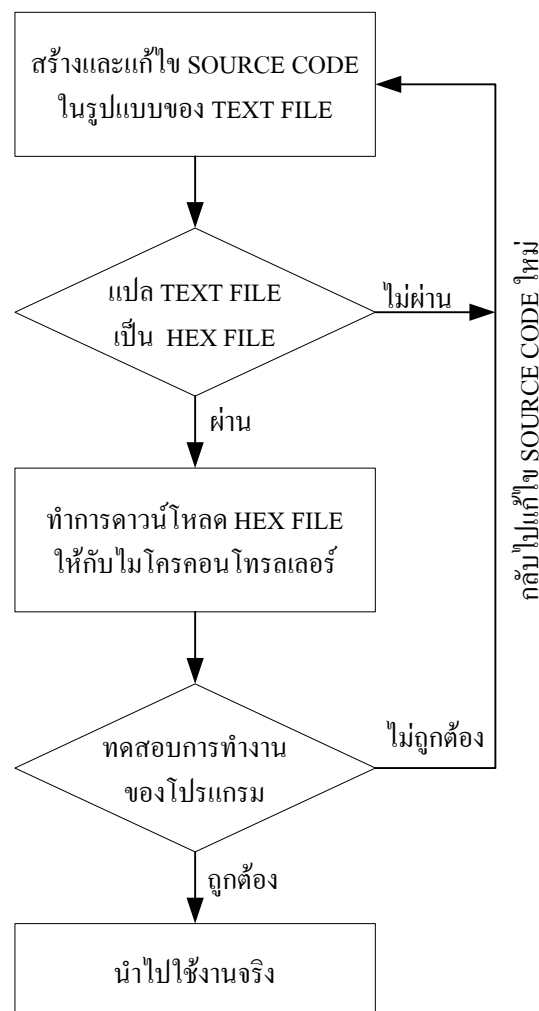
- คร่อมระหว่าง RXA (RX+) และ RXB (RX-) ไว้ แต่ถ้าหากว่าต่อสายสัญญาณในระยะทางที่ไม่ไกลมากนัก ก็ให้ทำการ Open Jumper นี้ออกก็ได้
- **Jumper TL** เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ TXB (TX-) หรือ Fail Safe Resister เพื่อให้สัญญาณ TXB (TX-) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกลๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งเมื่อใช้งานเป็นแบบ RS485 หรือใช้งานเป็นตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อรวมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
  - **Jumper TH** เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ TXA (TX+) หรือ Fail Safe Resister เพื่อให้สัญญาณ TXA (TX+) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกลๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งเมื่อใช้งานเป็นแบบ RS485 หรือใช้งานเป็นตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อรวมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
  - **Jumper TZ** เป็น Jumper สำหรับเลือกกำหนดการต่อตัวต้านทาน TZ เพื่อชดเชย ค่าความต้านทานของสายสัญญาณ (Impedance) ทางด้านส่ง ซึ่งถ้าหากว่ามีการต่อสายสัญญาณในการรับส่งเป็นระยะทางไกลๆแล้วก็ควรทำการ Short Jumper นี้ไว้ด้วยเนื่องจากเมื่อสายมีความยาวมากจะเกิดค่าความต้านทานในสายขึ้น ดังนั้นจึงต้องทำการต่อค่าความต้านทานจากภายนอกไปชดเชยค่าความต้านทานของสายสัญญาณด้วย โดยเมื่อทำการ Short Jumper ตำแหน่ง TZ นี้ไว้ก็จะเป็นการต่อตัวต้านทานคร่อมระหว่าง TXA (TX+) และ TXB (TX-) ไว้ แต่ถ้าหากว่าต่อสายสัญญาณในระยะทางที่ไม่ไกลมากนัก ก็ให้ทำการ Open Jumper นี้ออกก็ได้

**\*\*\*ข้อสังเกต\*\*\*** จะเห็นได้ว่าวงจร Line Driver ทั้งแบบ RS422 และ RS485 นั้นจะมีความใกล้เคียง กันมาก แต่มีข้อแตกต่างอย่างหนึ่งที่เห็นได้ชัดเจนที่สุด คือ ถ้าเป็นแบบ RS422 จะไม่สามารถสลับเปลี่ยน ทิศทางการรับส่งข้อมูลด้วยโปรแกรมได้ ซึ่งทิศทางการรับส่งจะกำหนดตายตัวจากวงจร แต่ถ้าเป็นแบบ RS485 นั้น จะสามารถสลับควบคุมทิศทางการรับส่งจากโปรแกรมได้ว่าจะให้ทำหน้าที่เป็นฝ่ายรับ หรือฝ่ายส่ง อย่างใดอย่างหนึ่งได้ตามต้องการได้



## การพัฒนาโปรแกรมของบอร์ด CP-PIC V3.0&amp;V4.0(ICD2)

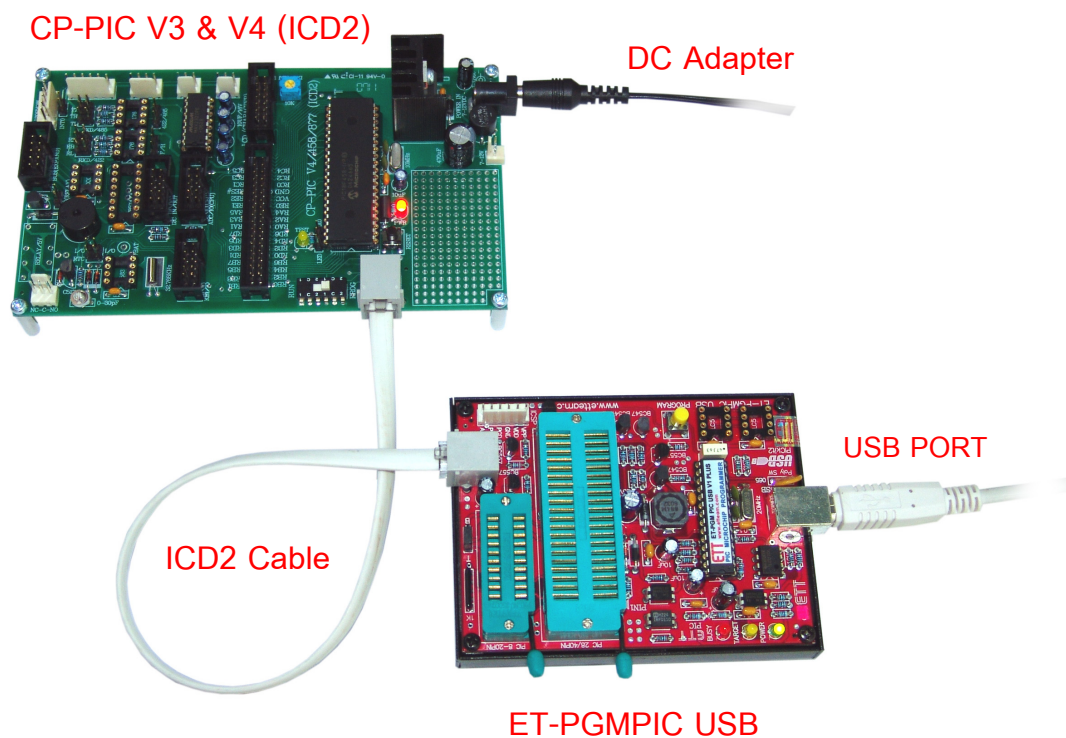
สำหรับการพัฒนาโปรแกรมนั้น ผู้ใช้งานสามารถเลือกที่จะใช้ภาษาใดก็ได้ในการพัฒนา เช่น ภาษาAssembly ,Basic หรือ ภาษา C ทั้งนี้ขึ้นอยู่กับความถนัดของผู้ใช้งาน แต่สุดท้ายแล้วจะต้องได้ไฟล์ที่จะโปรแกรมให้กับ CPU นั้นก็คือ HEX FILE ดังนั้นในการพัฒนาโปรแกรมจึงต้องมี Compiler สำหรับแปลภาษาที่เราเขียน (TEXT FILE) ให้เป็นภาษาที่ไมโครคอนโทรลเลอร์เข้าใจ (HEX FILE) สำหรับในที่นี้จะขอกล่าวถึงเฉพาะวิธีการ Download Hex File ให้กับบอร์ดเท่านั้น ส่วนวิธีการเขียนโปรแกรมและการส่งแปลคำสั่งให้ได้เป็น Hex File นั้น ขอให้ผู้ใช้ศึกษาจากข้อกำหนดของโปรแกรมแปลภาษาที่จะนำมาใช้ในการเขียนโปรแกรมเอง ซึ่งบอร์ด CP-PIC V3.0&V4.0(ICD2) จะออกแบบพอร์ตสัญญาณสำหรับโปรแกรม (ICD2 Port) ให้สามารถทำการโปรแกรมข้อมูลลง CPU ได้โดยการเชื่อมต่อกับเครื่องโปรแกรมจากภายนอก (เช่น เครื่องโปรแกรม ET-PGMPIC USB) โดยในขั้นตอนการพัฒนาโปรแกรมนั้นสามารถสรุปเป็นโฟลว์ชาร์ต ดังนี้



แผนผัง แสดงขั้นตอนในการพัฒนาโปรแกรม

## ขั้นตอนการดาวน์โหลดโปรแกรมด้วยเครื่องโปรแกรม ET-PGM PIC USB

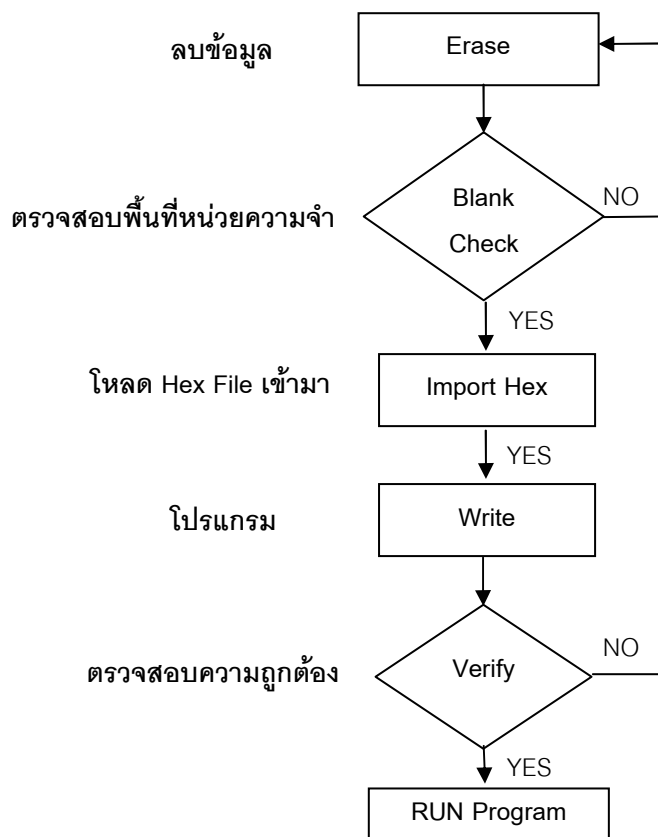
1. เชื่อมต่อ ET-PGM PIC USB เข้ากับพอร์ต USB ของคอมพิวเตอร์
2. เชื่อมต่อสายสัญญาณโปรแกรมจากพอร์ต ICD2 ของเครื่องโปรแกรม ET-PGM PIC USB มาเข้าที่พอร์ต ICD2 ของ บอร์ด CP-PIC V3 & V4(ICD2) และ ต่อแหล่งจ่ายไฟเข้ากับบอร์ดดังรูปต่อไปนี้



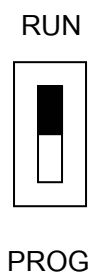
3. เลื่อนตำแหน่งสวิตช์ของบอร์ด CP-PIC V3 & V4 มาที่ตำแหน่ง PROG



4. เปิดโปรแกรม PicKit 2 ซึ่งในขั้นตอนนี้เราสามารถทำการ Erase , Blank Check ,Write , Read และ Verify ได้ตามต้องการ (รายละเอียดให้ศึกษาในคู่มือการใช้งานเครื่องโปรแกรม ET-PGM PIC USB) โดยทั่วไปขั้นตอนการโปรแกรมแบบคร่าวๆ จะเป็นดังโฟลว์ชาร์ตต่อไปนี้



5. เมื่อต้องการกลับสู่โหมดการ RUN โปรแกรมให้เลื่อนสวิตช์ RUN/PROG มาที่ตำแหน่ง RUN



## ปัญหาที่อาจเกิดขึ้น และแนวทางการแก้ไข

CPU ไม่ทำงานหรือทำงานผิดพลาด อาจเกิดขึ้นจากหลายสาเหตุดังนี้

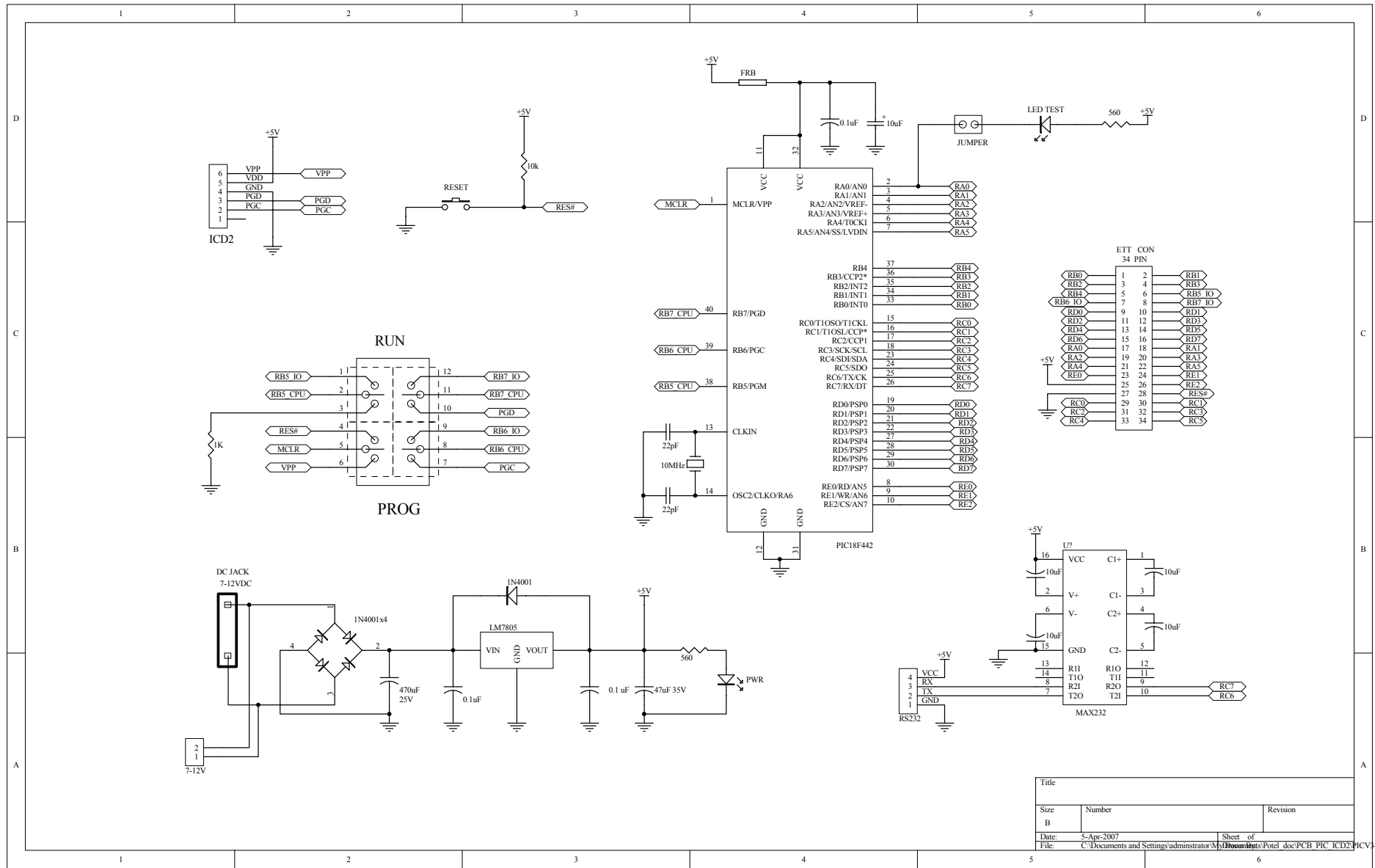
- สวิตช์ PROG/RUN อยู่ในตำแหน่ง PROG ทำให้ CPU อยู่ในโหมดโปรแกรมจึงไม่สามารถทำงานตามโปรแกรมได้ การแก้ไขให้เลือกสวิตช์มาที่โหมด RUN เมื่อต้องการให้ CPU ทำงานในโหมด Run Program
- การตั้งค่า Configuration โดยเฉพาะใน โหมดของ OSC มีผลมาก เพราะ PIC 18F458 สามารถทำงานได้ 2 ความถี่ คือ ความถี่จาก OSC ภายนอก(HS : ความถี่ 10 MHz) และ โหมด PLL (H4 : OSC ภายนอก x 4 ก็คือ 40MHz) ให้ตรวจสอบให้ถูกต้องว่าใช้งานในโหมดไหน
- Watchdog Timer หากเลือกการทำงานนี้ ใน Configuration แล้วไม่เขียนโปรแกรมไป Clear ค่า Watchdog จะทำให้เกิดการรีเซต ตลอดเวลาตามค่า Watchdog Timer ดังนั้นจึงควรตรวจสอบก่อนว่ามีการใช้งาน Watchdog Timer หรือไม่ ถ้าไม่ก็ไม่ควรเลือกการทำงานในส่วนนี้
- การ Set Jumper ต่างๆไม่ถูกต้อง ให้ตรวจสอบให้ถูกต้อง ซึ่งสามารถดูรายละเอียดได้จากหัวข้อ รายละเอียดการใช้งาน Jumper
- ในกรณีที่เขียนด้วยภาษา BASIC ( ใช้ PIC Basic Pro) จะต้องมีการกำหนดค่าความถี่ที่ใช้งานโดยใช้คำสั่ง DEFINE OSC \_ \_ ซึ่งหากไม่มีการ กำหนดความถี่ให้กับ โปรแกรม มันจะถือว่าเป็นความถี่ 4 MHz ซึ่งจะทำให้โปรแกรมทำงานผิดพลาดได้ดังนั้นเราจึงควรกำหนดให้ถูกต้อง

ตัวอย่าง

```
DEFINE   OSC   10   ‘ การกำหนดความถี่ 10 MHz
DEFINE   OSC   40   ‘ การกำหนดความถี่ 40 MHz
```

- การเปลี่ยนโหมดความถี่ เนื่องจาก PIC 18F458 ในบอร์ด CP-PIC V3.0&V4.0 สามารถทำงานได้ทั้ง 10MHz(HS) และ 40 MHz(H4) ฉะนั้นในการเปลี่ยนโหมดการทำงานจาก 10 MHz ไปเป็น 40MHz หรือจาก 40MHz เป็น 10MHz เมื่อทำการโปรแกรมเปลี่ยนโหมดเสร็จ CPU จะยังคงทำงานในโหมดความถี่เดิมอยู่ จนกว่าจะมีการปลดไฟเลี้ยง แล้วจ่ายไฟเข้ามาใหม่ CPU จึงจะเปลี่ยนมาทำงานในโหมดความถี่ใหม่ที่โปรแกรมให้ครั้งสุดท้าย

# CP-PIC V3 & V3 EXP (ICD2) Circuit



Title		
Size	Number	Revision
B		
Date:	5-Apr-2007	Sheet of
File: C:\Documents and Settings\administrator\My Documents\potel doc\PCB PIC\ICD2\PCV3-U.DDB		

## CP-PIC V4 (ICD2) Circuit

