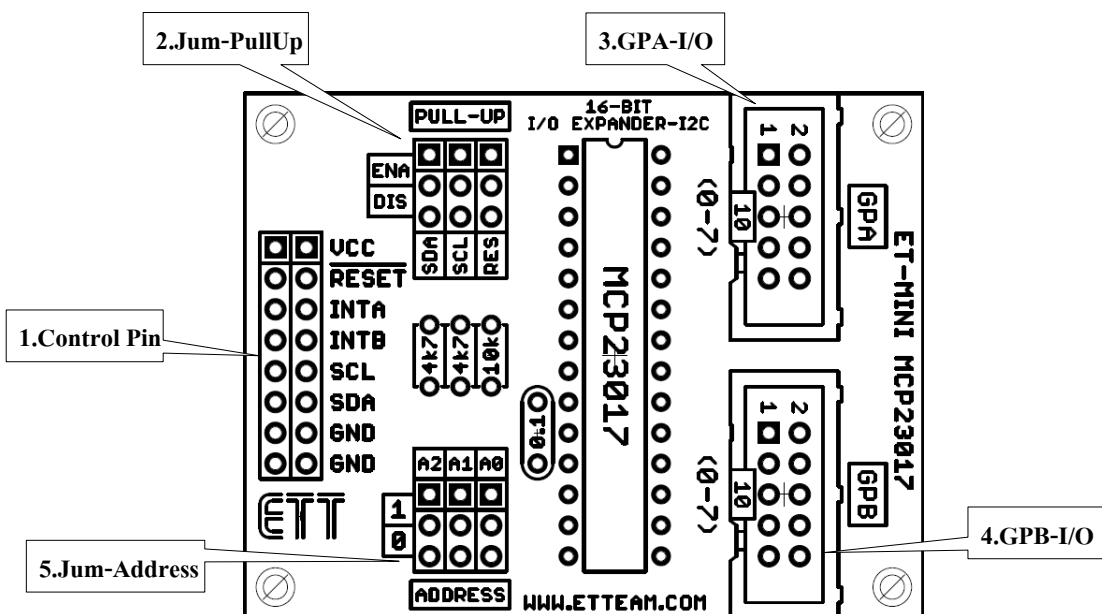


**ET-MINI MCP 23017**

Module ET-MINI MCP 23017 เป็นชุดขยายจำนวน Port I/O ให้กับ MCU ขนาด 16 Bit ซึ่งจะแบ่งออกเป็น 8 Bit 2 Port (GPA,GPB) โดยใช้การ Interface แบบ I2C Bus ซึ่งคุณสมบัติของ Module มีดังนี้

- 16-Bit Bidirectional I/O Port โดย I/O Pin Default จะถูกกำหนดให้เป็น Input
- ความถี่ Clock ในการ Interface I2C อยู่ที่ 100KHz , 400KHz และ 1.7MHz ซึ่งขึ้นอยู่กับระดับแรงดันที่เลี้ยงตัว MCP23017
- Module ทำงานได้ที่ระดับแรงดัน 1.8V -5.5V
- จะมี PIN A0,A1,A2 สำหรับกำหนด Address ให้กับตัว MCP23017 ได้จากภายนอก นั่นคือ เราสามารถต่อ MCP23017 ได้ทั้งหมด 8 ตัว ใน Bus เดียวกัน
- สามารถกำหนด Interrupt Output Pin ให้ Active-High ,Active-Low หรือ Open-Drain
- Pin INTA , INTB สามารถกำหนดให้ทำงานอิสระต่อกันตาม Port GPA,GPB หรือทำงานร่วมกันคือให้สัญญาณ INT ออกมาเหมือนกัน
- สามารถกำหนดการเกิด Interrupt ได้ 2 แหล่ง คือ กำหนดได้จาก ค่าการเปลี่ยนแปลงของ Input Pin เทียบกับค่า Configure ใน Register หรือ ค่าการเปลี่ยนแปลงของ Input Pin ค่าเก่า เทียบกับค่าการเปลี่ยนแปลงของ Input Pin ปัจจุบัน
- สามารถกำหนดสัญญาณ Input Port ให้อ่านค่า Inversion จากค่า Input ที่รับเข้ามาจริงได้
- กินกระแสขณะ Standby สูงสุด 1 uA

**โครงสร้าง Module และตำแหน่ง PIN ใช้งาน**

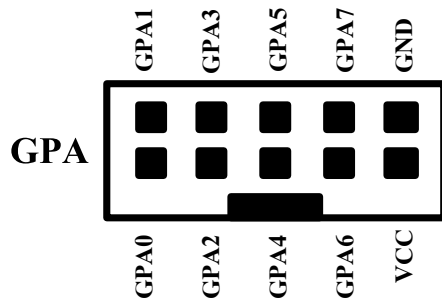


**1. Control Pin :** เป็นขั้วต่อสำหรับให้ MCU ส่งคำสั่งมา Control การทำงานของ Module แบบ I2C โดยมีรายละเอียดดังนี้

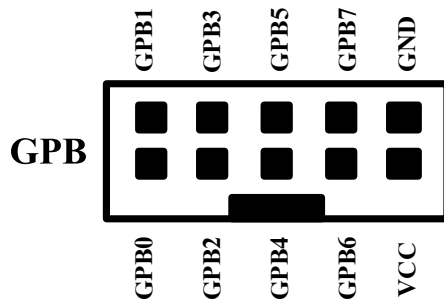
- VCC ขั้วต่อไฟเลี้ยง Module 1.8V-5.5V
- RESET เป็นขั้วต่อ Input เพื่อรับ สัญญาณ Reset (Active = 0)จาก MCU เข้ามาทำการ Reset ตัว Module
- INTA เป็นขั้วต่อ Interrupt Output สำหรับ Interrupt ที่เกิดจาก Port GPA (เมื่อ GPAทำงานเป็น Input)
- INTB เป็นขั้วต่อ Interrupt Output สำหรับ Interrupt ที่เกิดจาก Port GPB (เมื่อ GPB ทำงานเป็น Input)
- SCL เป็นขั้วต่อ Input สำหรับรับสัญญาณ Serial Clock ของ I2C จาก MCU
- SDA เป็นขั้วต่อ Input/Output สำหรับ รับส่ง Serial Data ของ I2C ระหว่าง MCU และตัว Module
- GND เป็นขั้วต่อกราวด์ในส่วนของไฟเลี้ยงของตัว Module

**2. Jum-PullUp :** เป็น Jumper สำหรับ ต่อ R-PullUp ให้ขา SDA,SCL และ Reset ของตัว Module โดยถ้าขา I2C Bus SDA และ SCL ไม่ได้ต่อ R-PullUp จากภายนอกผู้ใช้จะต้อง Set Jumper ไปทางด้าน ENA เพื่อทำการต่อ R-PullUp แต่ถ้าภายนอก Module มีการต่อ R-PullUp ที่ขาทั้งสองนี้ไว้แล้วก็ให้ set Jumper มาทางด้าน DIS ส่วน Jumper RES ควร Set ไปทางด้าน ENAเสมอ เพื่อต่อ R-PullUp ให้ขา Reset เป็น '1' ไว้เสมอ ถึงจะใช้การ Control ผ่าน Pin Reset ก็ตาม เพื่อป้องกัน Module Reset ในสถานะที่เราไม่ต้องการ

**3. GPA-I/O :** เป็นขั้วต่อ I/O สำหรับ Port GPA ขนาด 8-Bit ไปใช้งาน โดยสามารถ Set ให้เป็นได้ ทั้ง Input หรือ Output ในระดับบิตได้ ซึ่งขึ้นอยู่กับคำสั่งส่งไป Set Up Module ของผู้ใช้ โดยมีการจัดเรียง Pin ดังรูปด้านล่าง

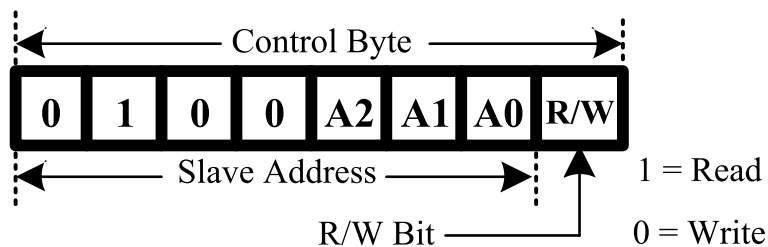


**4. GPB-I/O :** เป็นขั้วต่อ I/O สำหรับ Port GPB ขนาด 8-Bit ไปใช้งาน โดยสามารถ Set ให้เป็นได้ ทั้ง Input หรือ Output ในระดับบิตได้ ซึ่งขึ้นอยู่กับคำสั่งส่งไป Set Up Module ของผู้ใช้ โดยมีการจัดเรียง Pin ดังรูปด้านล่าง



**5. Jum-Address :** เป็น Jumper สำหรับกำหนดตำแหน่ง Address Slave A0,A1,A2 ให้กับ ตัว MCP23017 เพื่อใช้กำหนดเป็น Control Byte ให้กับตัวอุปกรณ์ดังแสดงในตารางด้านล่าง โดยสามารถกำหนดได้ 8 ค่า นั่นก็คือ สามารถนำ MCP23017 มาต่อขนานกันใน Bus I2C ได้ทั้งหมด 8 ตัวนั่นเอง

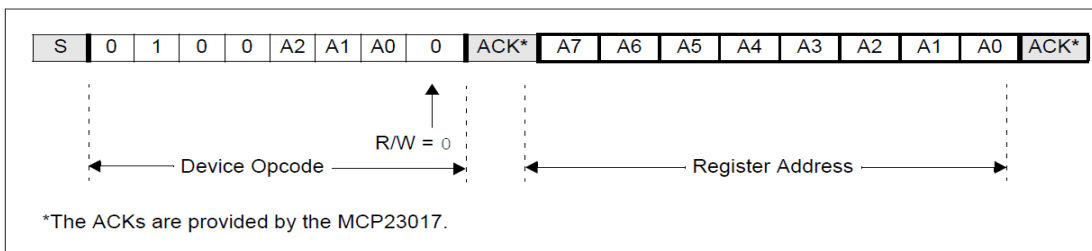
### Format Control Byte MCP23017



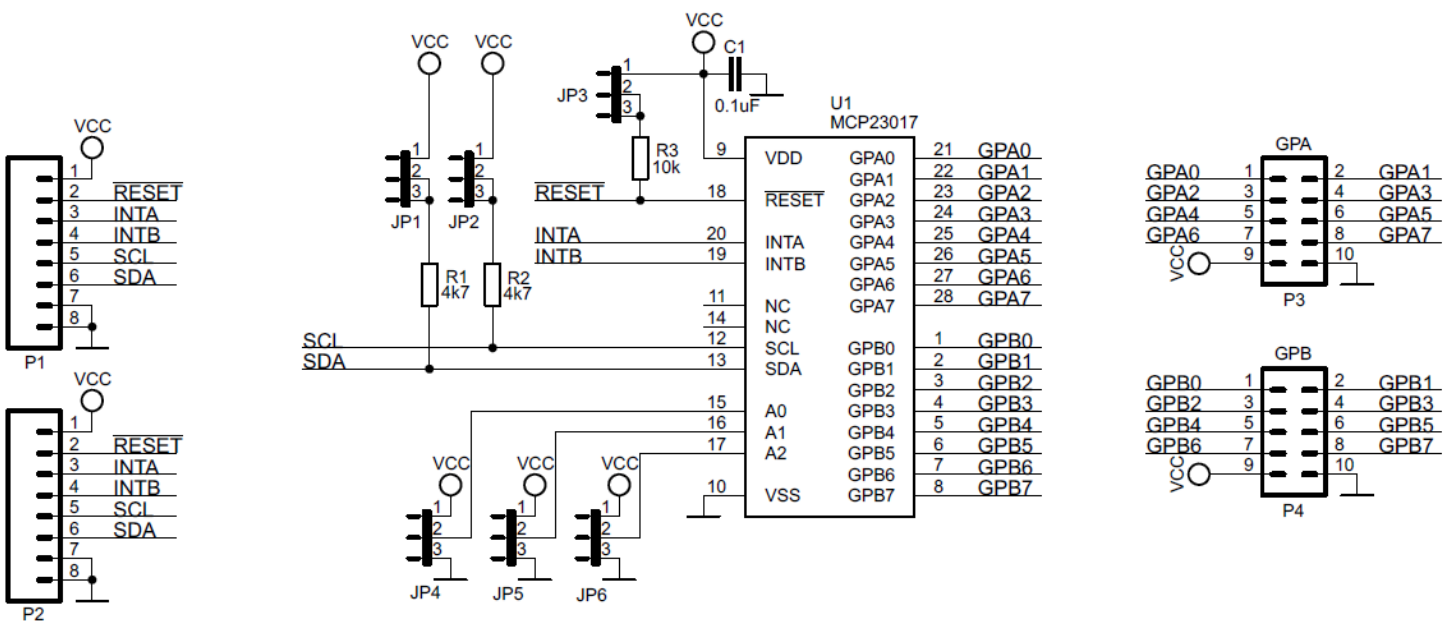
จากตัวอย่างที่ให้มาใน CD เราจะ Set A0 , A1 , A2 ไปทางด้าน '0' ดังนั้น Slave Address A0 , A1 , A2 จะมีค่าเป็น '0' เมื่อแทนลงในตารางด้านบน เราก็จะได้ Control Byte สำหรับใช้เขียน คือ 0x40 และ สำหรับใช้อ่าน คือ 0x41 เป็นต้น

**การต่อใช้งาน**

1. ที่ Jumper PULL-UP ให้ Set Jumper RES ไปทางด้าน ENA ส่วน Jumper SDA และ SCL ให้ Set ไปทางด้าน ENA ถ้า PIN ทั้งสองนี้ ไม่มีการต่อ R-PullUp จากภายนอก Module แต่ถ้ามีการต่อ R-PullUp จากภายนอก Module แล้ว ให้ Set Jumper ทั้งสองนี้ไปทางด้าน DIS
2. ที่ Jumper Address ให้ Set Jumper A0,A1,A2 ไปทางด้าน 0 หรือ 1 ก็ได้ แล้วแต่ Address ที่ผู้ใช้ต้องการ จากตัวอย่างโปรแกรมจะ Set ไปที่ตำแหน่ง '0' ทั้งหมด จากนั้นนำค่า A0,A1,A2 ไปแทนค่าลงในตารางด้านบน แล้วอ่านค่า Control Byte สำหรับใช้ เขียน/อ่าน อุปกรณ์ I2C ออกมาเก็บไว้ใช้ในการส่ง Command จากตัวอย่าง จะได้ค่าเป็น 0x40 (W) และ 0x41(R)
3. ต่อสาย SDA และ SCL ของ MCUเข้ากับ PIN SDA และ SCL ของ Module ตามลำดับ ส่วน PIN RESET , INTA, INTB ของ Module ถ้าไม่ได้ใช้งานก็ปล่อยลอยไว้ได้ สุดท้ายจ่ายไฟเลี้ยง DC 5 V ให้กับ Module ที่ PIN VCC และ GND
4. ต่อสายทางด้าน Port GPA หรือ GPB ไปยังอุปกรณ์ที่ต้องการจะ Control เช่น LED หรือ SW. หรือบอร์ด Interface I/O ต่างๆ
5. ทำการเขียนโปรแกรม Interface I2C โดยมีรูปแบบการส่ง Address Register ดังรูปด้านล่าง (ดูตัวอย่างใน CD ประกอบ)



สำหรับการเขียนโปรแกรมควบคุมการใช้งาน Module ให้ดูได้จาก Data Sheet และ ตัวอย่างโปรแกรมที่ให้ไว้ใน CD โดยตัวอย่างที่ให้มาใน CD นั้นจะมีการทำงานดังนี้ คือ ให้พิจารณาการ Control Module จาก ฟังก์ชัน init\_MCP23017() โดยเราจะอ้างอิง Address Register ใน Bank 0 ซึ่งจะถูกกำหนดมาให้ เป็นค่า Address Default อยู่แล้ว จากนั้นก็ทำการ Set Port GPA [0..7] เป็น Output ซึ่งจะต่อ LED Common Anode ไว้ (Active '0') ส่วน Port GPB[0..7] จะกำหนดให้เป็น Input โดยจะต่อ SW. ไว้ พร้อมกำหนดให้ Pull-Up ไว้ด้วย ในส่วนของการ Setup Interrupt เพื่อใช้งาน PIN INTA และ INTB ของ Module ถ้าผู้ใช้ไม่ต้องการใช้งาน ก็ไม่ต้องทำการ Setup ใดๆ จากนั้นก็ให้มาพิจารณาในส่วนของ Main โปรแกรม โดยเราจะอ่านค่าการกด SW จาก Port GPB ถ้า บิตใดถูกกด (กด SW.= '0') ก็จะไปสั่งให้ LED ของ บิตนั้น ใน Port GPA ติด จากนั้นก็จะทำการอ่าน Port GPB อีกครั้งเพื่อตรวจสอบการปล่อย SW.



รูปวงจร ET-MINI MCP 23017