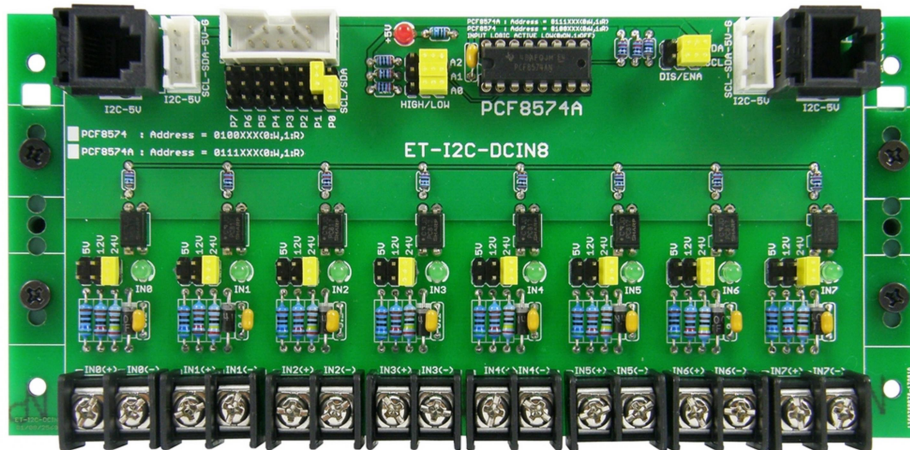


ET-I2C DCIN8

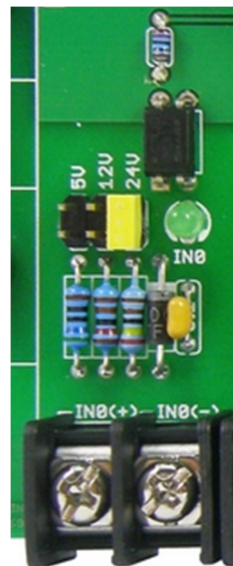
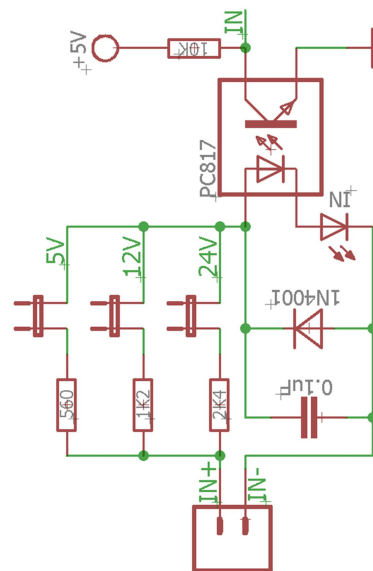
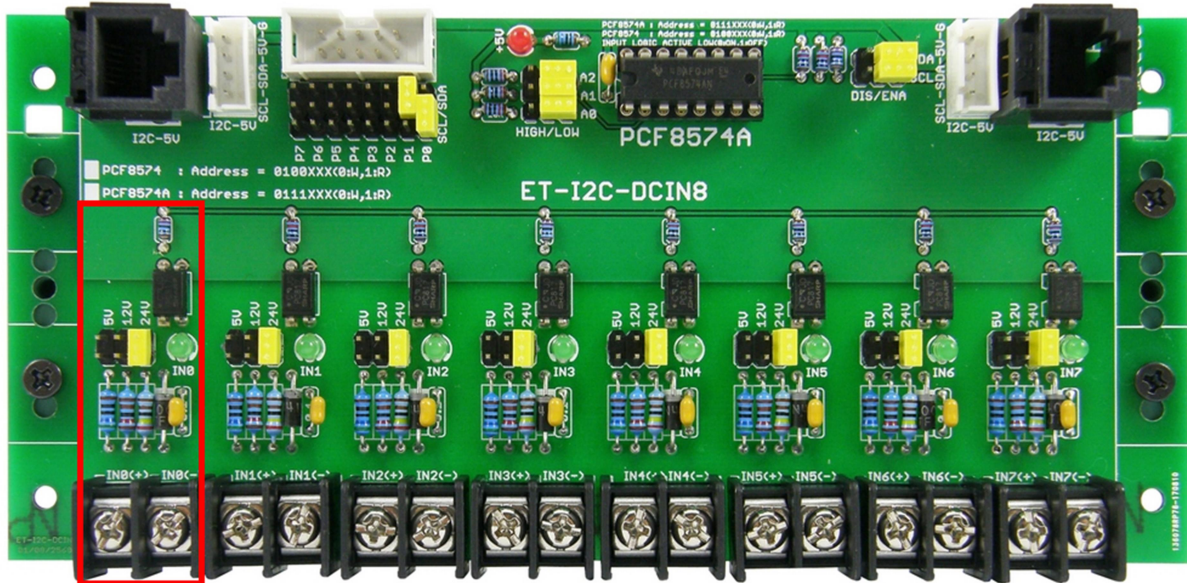


ET-I2C DCIN8 เป็นบอร์ดขยาย Input แบบ OPTO Isolate DC Input ขนาด 8ช่อง ผ่านทาง I2C Bus โดยบอร์ดสามารถใช้ชิพ PCF8574/A เป็นตัวกลางในการเชื่อมต่อระหว่าง MCU และ Input DC จึงทำให้สามารถเลือกกำหนดตำแหน่งแอดเดรสการทำงานของบอร์ดจาก Jumper ให้มีความแตกต่างกันได้ 8 ตำแหน่ง ซึ่งจะทำให้สามารถเลือกใช้บอร์ด ET-I2C DCIN8 ที่ติดตั้งชิพ PCF8574 ร่วมกันในบัสเดียวกันได้มากถึง 8บอร์ด และยังสามารถนำบอร์ด ET-I2C DCIN8 ที่ทำการติดตั้งชิพ PCF8574A มาต่อร่วมในบัสเดียวกันเพิ่มได้อีก 8บอร์ด รวมเป็น 16บอร์ดได้อีกด้วย สำหรับบอร์ดมาตรฐานจาก อีทีที จะติดตั้งชิพเบอร์ PCF8574A มาให้เป็นอุปกรณ์มาตรฐานประจำบอร์ด

Specification

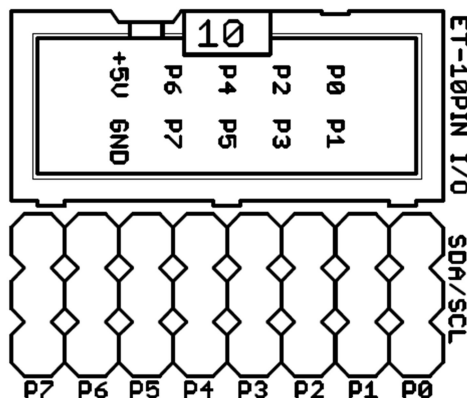
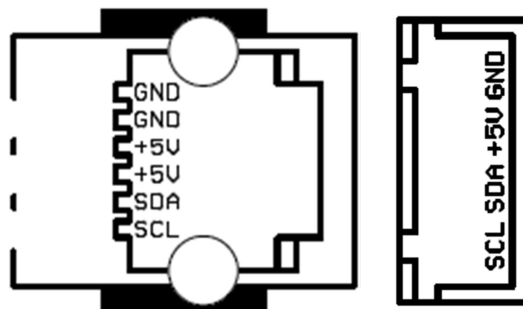
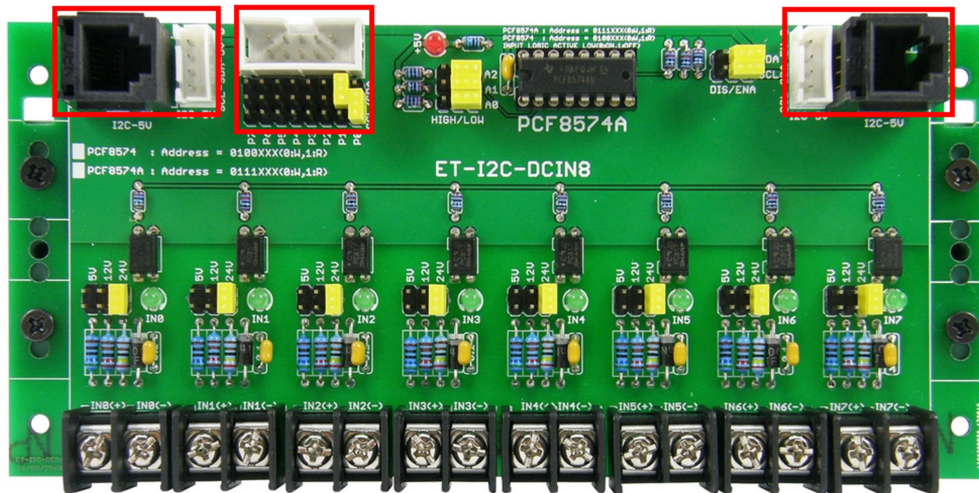
- Interface I2C Bus 5V
 - 1 x 10PIN IDE Connector
 - 2 x RJ11 6PIN Connector
 - 2 x CPA 4PIN Male Block Connector
- 8 Channel Input Opto-Isolate Active LOW Logic
 - 8 x TERMINAL 7.62mm 2PIN For Channel Input
 - +5V / +12V / +24V Input Select with Green LED Indicator
- 1 x Internal Red LED +5V Power Indicator
- 3 x Jumper For 8 Position I2C Address Setting
- 2 x Jumper For Enable/Disable Pull-Up I2C Signal
- Dimension Size 7.5cm x 16.5cm (3 Inch x 6.5 Inch)
- Din Rail 35mm (Option)

ภาค DC Input แต่ละช่องสามารถเลือกกำหนดให้รับแรงดันอินพุตได้อิสระต่อกัน 3 ย่าน คือ +5V +12V และ +24V โดย DC Input ของบอร์ด ET-I2C DCIN8 ถูกออกแบบให้ทำงานแล้วได้ค่า Input เป็น Logic LOW และหยุดทำงานแล้วได้ค่า Input เป็น Logic HIGH



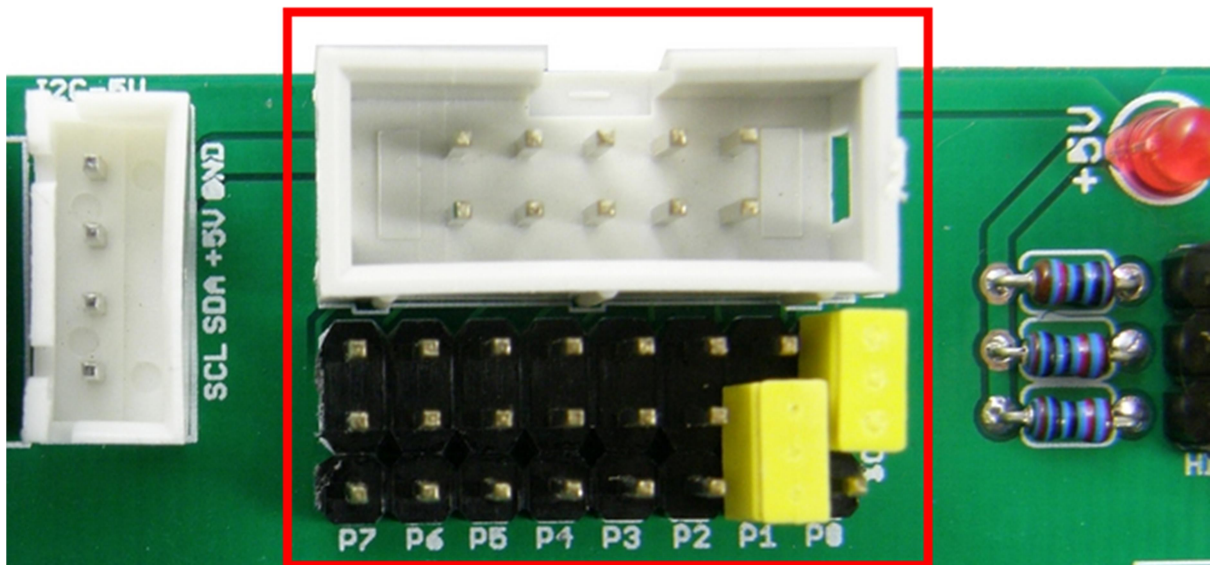
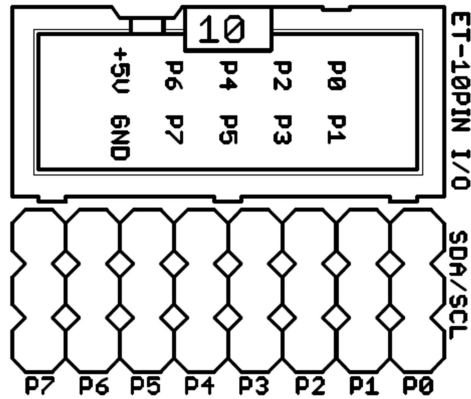
วงจรภาค Opto Isolate DC Input

การเชื่อมต่อกับบอร์ด จะสามารถเลือกต่อสัญญาณเพื่อส่งอ่านค่า Input ของบอร์ดผ่านทางขั้ว I2C ซึ่งมีให้เลือกต่อได้ถึง 3 แบบ คือ ขั้วต่อแบบ 10PIN IDE มาตรฐาน ET-10PIN BUS สำหรับใช้เชื่อมต่อกับบอร์ดของ อีทีที ที่มีขั้วสัญญาณเชื่อมต่อเป็นแบบ 10PIN IDE โดยวงจรในส่วนนี้จะมี Jumper สำหรับเลือกกำหนดได้ว่าจะใช้สัญญาณบิตจากขั้ว 10PIN IDE ทำหน้าที่เป็น SCL และ SDA ได้ตามต้องการอีกด้วย นอกจากนี้แล้วยังมีขั้วต่อแบบ RJ11 แบบ 6Pin และขั้วต่อแบบ CPA-4Pin Block ซึ่งมีให้อย่างละ 2 ชุด บนบอร์ด สำหรับรับสัญญาณเข้ามาควบคุมและต่อพ่วงสัญญาณออกไปควบคุมบอร์ดถัดไปในกรณีที่ต้องการจำนวน Input/Output แบบ I2C จำนวนหลายๆบอร์ดร่วมกันได้อีกด้วย



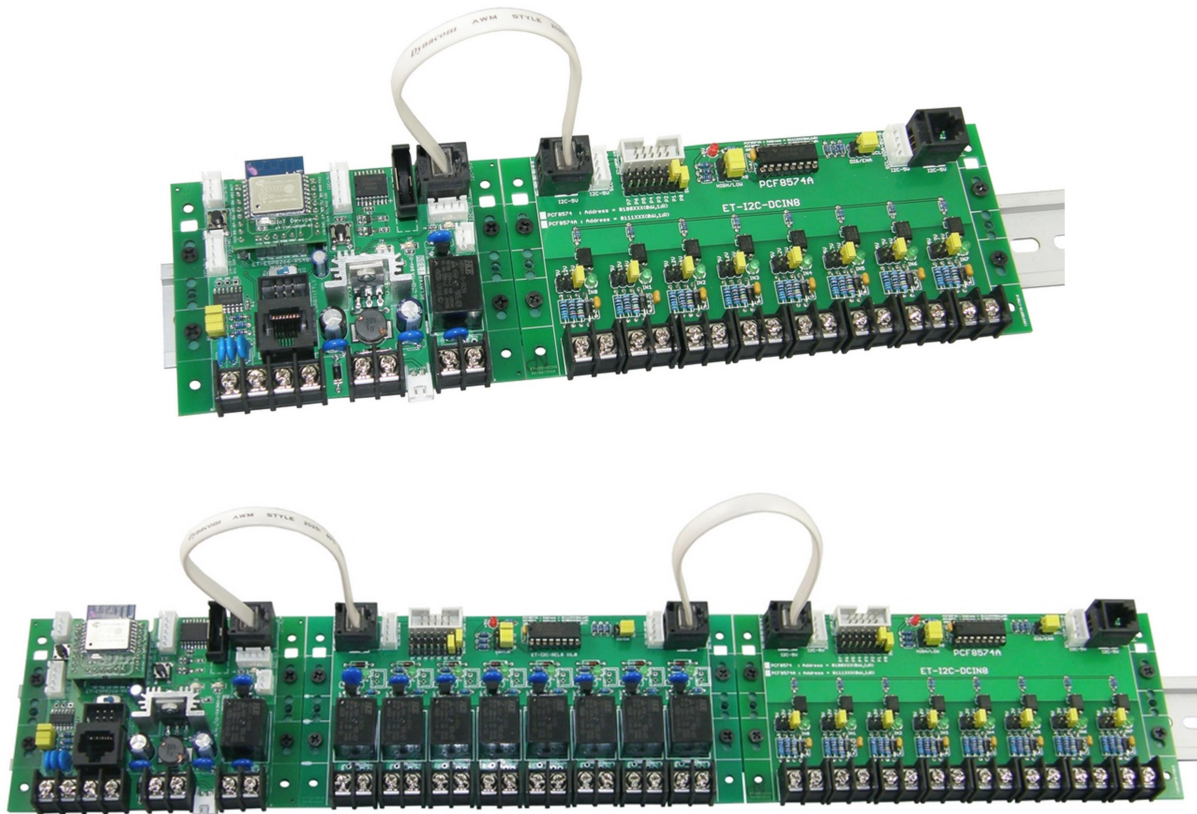
รูปแสดง ลักษณะและตำแหน่งขั้วสัญญาณที่ใช้ในการเชื่อมต่อ I2C Bus

การเชื่อมต่อสัญญาณผ่านขั้ว 10PIN IDE จะกระทำผ่านสายแพรขนาด 10PIN ซึ่งในขั้ว 10PIN จะมีสัญญาณ 8เส้น ซึ่งสามารถเลือกกำหนดได้ว่าการใช้สัญญาณเส้นใด เป็น SCL และ SDA ก็ให้ทำการเลือกกำหนด Jumper ของ Pin P0-P7 ที่ต้องการให้ทำหน้าที่เป็น SCL หรือ SDA ของ I2C Bus ได้ตามต้องการ



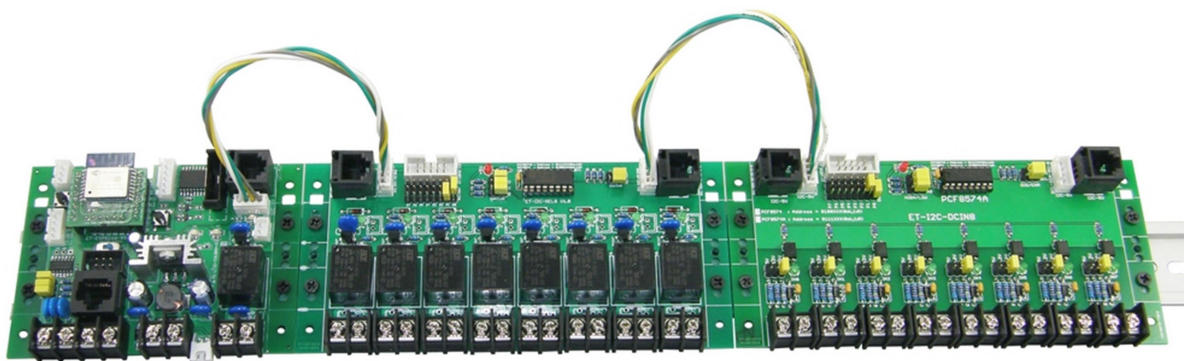
ตัวอย่างการเชื่อมต่อผ่าน 10PIN IDE และกำหนด Jumper ให้ P0 = SDA , P1 = SCL

การเชื่อมต่อ I2C โดยใช้หัว RJ11 6Pin



ตัวอย่างการต่อ I2C กับบอร์ด MCU ผ่านทางสาย RJ11 6PIN

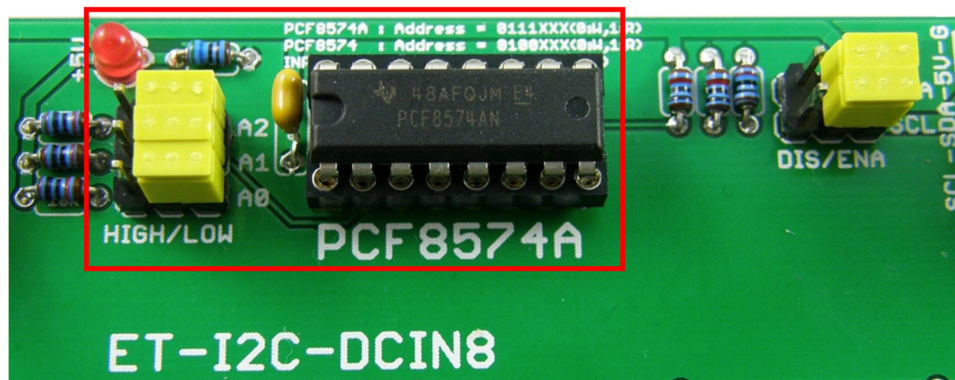
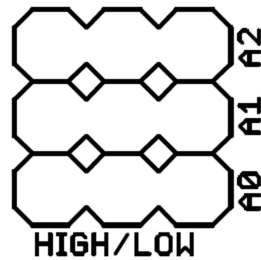
การเชื่อมต่อ I2C ผ่านหัว CPA 4Pin Block



ตัวอย่างการต่อ I2C กับบอร์ด MCU ผ่านทางสาย CPA 4PIN Block

การกำหนด Address I2C

บอร์ด ET-I2C REL8 สามารถเลือกติดตั้งชิพ PCF8574 หรือ PCF8574A สำหรับควบคุมการทำงานของ Relay ได้ แต่ตามปกติแล้วบอร์ดมาตรฐานจาก อีทีที จะติดตั้งชิพบอร์ด PCF8574 มาให้ ซึ่ง สามารถเลือกกำหนดตำแหน่ง Address การเชื่อมต่อของชิพ ได้จาก Jumper A0,A1 และ A2 ภายในบอร์ดให้มีตำแหน่งการติดต่อสัญญาณที่ไม่ซ้ำกันได้ 8 ตำแหน่ง โดยการเลือกกำหนด Jumper ให้สัญญาณ A0,A1,A2 มีสถานะเป็น LOW หรือ HIGH ที่ไม่ซ้ำกันได้ดังตาราง



การกำหนด Jumper เลือก Address			ตำแหน่งแอดเดรส		
A2	A1	A0	Address	PCF8574	PCF8574A
LOW	LOW	LOW	0	0x41 : 0100 000(1:R)	0x71 : 0111 000(1:R)
LOW	LOW	HIGH	1	0x43 : 0100 001(1:R)	0x73 : 0111 001(1:R)
LOW	HIGH	LOW	2	0x45 : 0100 010(1:R)	0x75 : 0111 010(1:R)
LOW	HIGH	HIGH	3	0x47 : 0100 011(1:R)	0x77 : 0111 011(1:R)
HIGH	LOW	LOW	4	0x49 : 0100 100(1:R)	0x79 : 0111 100(1:R)
HIGH	LOW	HIGH	5	0x4B : 0100 101(1:R)	0x7B : 0111 101(1:R)
HIGH	HIGH	LOW	6	0x4D : 0100 110(1:R)	0x7D : 0111 110(1:R)
HIGH	HIGH	HIGH	7	0x4F : 0100 111(1:R)	0x7F : 0111 111(1:R)

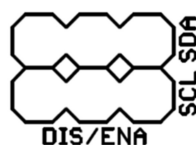
ตารางแสดง ตำแหน่งแอดเดรส I2C Bus ของบอร์ดในกรณีใช้เป็น **Control Word** ของ I2C

การกำหนด Jumper เลือก Address			ตำแหน่งแอดเดรส		
A2	A1	A0	Address	PCF8574	PCF8574A
LOW	LOW	LOW	0	0x20 : 0010 0000(1:R)	0x38 : 0011 1000(1:R)
LOW	LOW	HIGH	1	0x21 : 0010 0001(1:R)	0x39 : 0011 1001(1:R)
LOW	HIGH	LOW	2	0x22 : 0010 0010(1:R)	0x3A : 0011 1010(1:R)
LOW	HIGH	HIGH	3	0x23 : 0010 0011(1:R)	0x3B : 0011 1011(1:R)
HIGH	LOW	LOW	4	0x24 : 0010 0100(1:R)	0x3C : 0011 1100(1:R)
HIGH	LOW	HIGH	5	0x25 : 0010 0101(1:R)	0x3D : 0011 1101(1:R)
HIGH	HIGH	LOW	6	0x26 : 0010 0110(1:R)	0x3E : 0011 1110(1:R)
HIGH	HIGH	HIGH	7	0x27 : 0010 0111(1:R)	0x3F : 0011 1111(1:R)

ตารางแสดง ตำแหน่งแอดเดรส I2C Bus ของบอร์ดในกรณีใช้กับ Library ของ Arduino

การเลือกกำหนด Pull-Up ให้ I2C Bus

ปรกติแล้วสัญญาณ I2C จะถูกขับผ่าน MOSFET ที่เป็น Open Drain ซึ่งมีความจำเป็นต้องต่อวงจร Pull-Up ให้กับสัญญาณของ I2C Bus ทั้ง 2 เส้น คือ SCL และ SDA ในบอร์ดด้วยเสมอเพื่อให้สัญญาณมีระดับลอจิกเท่ากับระดับลอจิกของขาสัญญาณ MCU ที่นำมาขับ ซึ่งบอร์ด MCU ที่นำมาต่อเพื่อขับนั้น บางวงจรอาจทำการติดตั้งตัวต้านทานสำหรับ Pull-Up ไว้อยู่แล้วภายในบอร์ดของ MCU เอง ซึ่งในกรณีนี้ผู้ใช้ก็ต้องทำการปลดการ Pull-Up ภายในบอร์ดของ ET-I2C DCIN8 ออกโดยการเลือก Jumper ของ SDA และ SCL ไว้ทางด้าน DIS(Disable) แต่ถ้าบอร์ด MCU ที่เป็นตัวขับ ไม่ได้ต่อวงจร Pull-Up ไว้ให้ ผู้ใช้ก็ต้องทำการเชื่อมต่อการ Pull-Up ภายในบอร์ด ET-I2C DCIN8 ที่จัดเตรียมไว้ในบอร์ด โดยเลือก Jumper ของ SDA และ SCL ไว้ทางด้าน ENA(Enable) ด้วย



ตัวอย่างการกำหนด Jumper เพื่อ Enable การ Pull-Up สัญญาณ SDA และ SCL

ตัวอย่างโปรแกรมอ่านค่า Input จาก ET-I2C DCIN8

```

#include <Wire.h>                                // I2C Bus

#include "pcf8574.h"                              // PCF8574/A

//=====

#define SerialDebug Serial                      // USB Serial

//=====

#define SDA_I2C      D2                        // PD1
#define SCL_I2C      D3                        // PD0

//=====

//=====

PCF8574 PCF8574A_INPUT_EXP0(0x38);    // PCF8574A = 0111,000+(0:W,1:R)
//=====
PCF8574 PCF8574A_INPUT_EXP1(0x39);    // PCF8574A = 0111,001+(0:W,1:R)
//=====
PCF8574 PCF8574A_INPUT_EXP2(0x3A);    // PCF8574A = 0111,010+(0:W,1:R)
//=====
PCF8574 PCF8574A_INPUT_EXP3(0x3B);    // PCF8574A = 0111,011+(0:W,1:R)
//=====
PCF8574 PCF8574A_INPUT_EXP4(0x3C);    // PCF8574A = 0111,100+(0:W,1:R)
//=====
PCF8574 PCF8574A_INPUT_EXP5(0x3D);    // PCF8574A = 0111,101+(0:W,1:R)
//=====
PCF8574 PCF8574A_INPUT_EXP6(0x3E);    // PCF8574A = 0111,110+(0:W,1:R)
//=====
PCF8574 PCF8574A_INPUT_EXP7(0x3F);    // PCF8574A = 0111,111+(0:W,1:R)
//=====

//=====

byte this_input;

byte last_input;

//=====

```



```
void setup()
{
    //=====
    Wire.begin();                // Initial I2C Bus
    //=====
    last_input = 0xFF;
    PCF8574A_INPUT_EXP0.write8(0xFF);
    //=====
    SerialDebug.begin(115200);    // Debug USART
    while(!SerialDebug);         // wait USB Serial Complete
    //=====
    SerialDebug.println();
    SerialDebug.println("Demo Test ET-I2C DCIN8");
    //=====
}

void loop()
{
    this_input = PCF8574A_INPUT_EXP0.read8();
    if(this_input != last_input)
    {
        last_input = this_input;
        SerialDebug.print("Input = ");
        if(this_input == B01111111) SerialDebug.print("0");    // 1111111 -> 01111111
        SerialDebug.println(this_input,BIN);
    }
    delay(100);
}
```