

คู่มือ Module MH-Z14A CO2 SENSOR

MH-Z14A CO2 SENSOR

เป็น Module Sensor ตรวจวัดก๊าซคาร์บอนไดออกไซด์ในอากาศ (หน่วย ppm) โดยใช้หลักการไม่กระจายตัวของ Infrared (NDIR) ในการตรวจจับก๊าซคาร์บอนไดออกไซด์(CO2) ที่มีอยู่ในอากาศ ตัว Module จะให้ความแม่นยำในการตรวจวัดก๊าซ CO2 สูง , มีการชดเชยอุณหภูมิภายในตัว ทำให้วัดค่าได้ทุกย่านอุณหภูมิการทำงาน เหมาะในการนำไปใช้งานเช่น ทำเป็นอุปกรณ์ตรวจวัดมลพิษทางอากาศ , อุปกรณ์เครื่องปรับอากาศ เป็นต้น

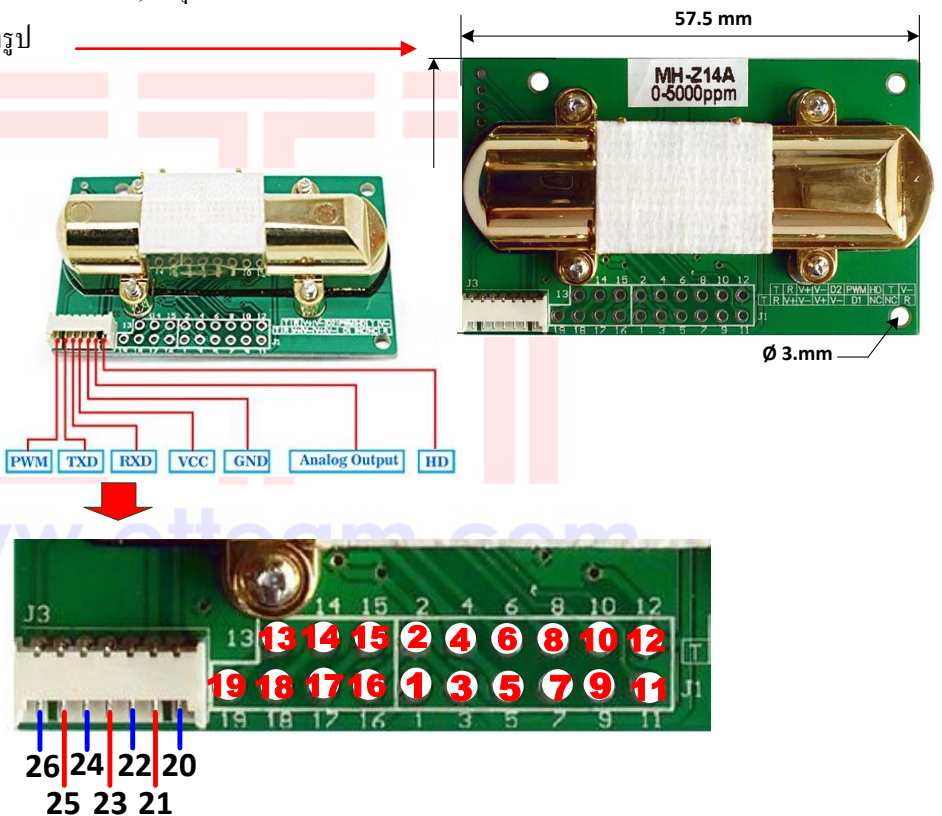
1. คุณสมบัติ Module MH-Z14A

- ไฟเลี้ยง Module DC 4.5V-5.5V , กินกระแสไฟเฉลี่ยน้อยกว่า 60 mA และสูงสุด 150 mA ที่ไฟเลี้ยง 5V
- ย่านการวัด CO2 อยู่ที่ 0-5000 ppm , มีความแม่นยำ ±50 ppm
- ให้สัญญาณ Output 3 รูปแบบ สำหรับเชื่อมต่ออ่านค่าไปใช้งาน
 - 1) ให้ Output แบบ Analog Voltage-(ADC) ย่าน 0.4V-2V
 - 2) ให้ Output แบบ PWM
 - 3) ให้ Output แบบ Digital เชื่อมต่อแบบ Serial (9600,N,8,1) ได้ค่า ppm ออกมาเลย ซึ่งรองรับการเชื่อมต่อที่ระดับแรงดัน 3.3V และ 5V
- ตัว Module ใช้เวลาในการอุ่นเครื่อง 3 นาที , ใช้เวลาในการตอบสนอง Output น้อยกว่า 30 วินาที
- อุณหภูมิการทำงาน 0-50 °C , ความชื้นการทำงาน 0-90% RH , อายุการใช้งาน Module มากกว่า 5 ปี
- ขนาดของ Module พร้อมรูเจาะยึดนี้ोट แสดงดังรูป

2. ข้อต่อใช้งาน Module

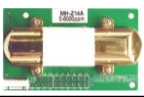
CO ₂ [ppm]	Air Quality
2100	BAD Heavily contaminated indoor air Ventilation required
2000	
1900	
1800	
1700	MEDIocre Contaminated indoor air Ventilation recommended
1600	
1500	
1400	
1300	FAIR
1200	
1100	GOOD
1000	
900	EXCELLENT
800	
700	
600	
500	
400	

Table 2 - Classification of Indoor Air Quality



- Pin 1, 15, 17, 23 = เป็นข้อต่อไฟเลี้ยง Module Sensor ขั้วบวก 4.5V-5.5V DC
- Pin 2, 3, 12, 16, 22 = เป็นข้อต่อไฟเลี้ยง Module Sensor ขั้ว GND
- Pin 4, 21 = เป็นขั้ว Analog Output Voltage โดยให้แรงดัน output ในระดับ 0.4V-2V
- Pin 6, 26 = เป็นขั้ว PWM Output
- Pin 8, 20 = เป็นขั้ว HD ใช้สำหรับ Set Zero Calibrate ให้กับ Module โดย jump Pin นี้ ไปยัง GND เป็นเวลา มากกว่า 7 วินาที ขณะที่ Module ทำงานอยู่
- Pin 7, 9 = ไม่ใช้งาน
- Pin 11, 14, 18, 24 = เป็นขั้ว Uart (Rx-TTL Level) สำหรับใช้รับคำสั่งจากภายนอกเพื่อสื่อสารอ่านค่าแบบ Digital
- Pin 10, 13, 19, 25 = เป็นขั้ว Uart (Tx-TTL Level) สำหรับใช้ส่งค่า Data ของ Sensor แบบดิจิทัลออกไปภายนอก

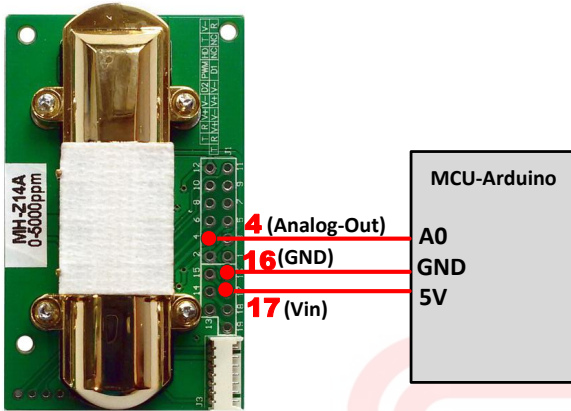
****ในการต่อใช้งานนั้นผู้ใช้สามารถเลือกต่อจุดที่เป็นขั้วสัญญาณเดียวกันจากจุดใดมาใช้งานก็ได้ตามที่ผู้ใช้สะดวก****



3. การต่อใช้งาน

สำหรับการต่อใช้งานเพื่ออ่านค่า CO2 จาก Module Sensor นี้ผู้ใช้สามารถเลือกต่อวงจรเพื่ออ่านค่ามาใช้งานได้ 3 รูปแบบ ตามที่ผู้ใช้งานต้องการ ดังนี้

3.1) **ใช้การต่ออ่านค่าแบบ Analog Voltage (ADC)** : โดยจะต่อที่ Pin4 หรือ Pin21 ของ Module Sensor ไปยังขา ADC ของ MCU ดังรูปตัวอย่างด้านล่าง โดยใช้ MCU-Arduino ซึ่งตัว Sensor จะให้ค่าแรงดันออกมาอยู่ในช่วง 0.4V-2V วิธีนี้ผู้ใช้จะต้องเขียนโปรแกรมอ่านค่า CO2 ในเทอมของค่า ADC ก่อนแล้วนำค่าไปคำนวณเพื่อปรับค่า CO2 ให้อยู่ในหน่วย ppm



```

ตัวอย่าง Function โปรแกรม(Arduino)

const int analogPin = A0;

int readPPM_ADC() //Function Read CO2(Analog)
{
  Float v = analogRead(analogPin) * 5.0 / 1023.0 ; //แปลงค่า Digital เป็น V
  int ppm = int((v - 0.4) * 3125.0) ; //เปลี่ยนค่า V เป็น ppm
  return ppm ; //ค่า CO2 หน่วย ppm ที่อ่านได้
}

```

ตัวอย่างการต่ออ่านค่า CO2 แบบ Analog Voltage(ADC)

สมการคำนวณหาค่า CO2 (ppm) = $(V_{out}-0.4) * (5000/1.6)$ *เนื่องจาก Sensor ให้แรงดัน Output อยู่ในช่วง 0.4-2V และมีย่านการวัดค่า ppm อยู่ในช่วง 0-5000 จึงได้สมการตามข้างต้น ถ้าค่าช่วงการวัดแรงดันและย่านการอ่านค่าเป็นค่าอื่น จะต้องแก้ไขตัวเลขในสมการด้วย

3.2) **ใช้การต่ออ่านค่าแบบ PWM** : โดยจะต่อที่ Pin6 หรือ Pin26 ของ Module Sensor ไปยังขา PWM ของ MCU ดังรูปตัวอย่างด้านล่าง โดยใช้ MCU-Arduino ในการเขียนโปรแกรมอ่านค่า ผู้ใช้จะต้องอ่านค่าเวลาของสัญญาณ pulse 1 Cycle ในช่วง Level High และ Level Low ออกมาให้ได้แล้วนำค่าเวลาที่อ่านได้มาคำนวณหาค่า CO2 ในหน่วย ppm ตามสมการ

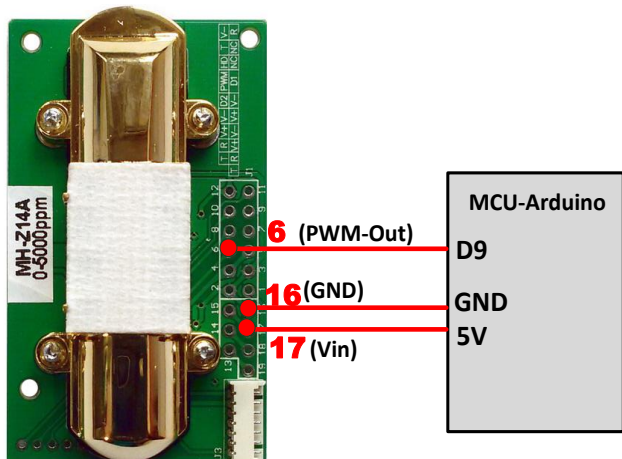
$$CO2(ppm) = 5000x(TH-2ms) / (TH+TL-4ms)$$

เมื่อ 5000 = ย่านการวัดค่า ppm ของ sensor

(ให้เปลี่ยนตามย่าน Sensor ที่นำมาใช้จริง)

TH = ค่าเวลาของ Pulse ช่วง Level High

TL = ค่าเวลาของ Pulse ช่วง Level Low



```

ตัวอย่าง Function โปรแกรม(Arduino)

int readPPM_PWM() { //Function Read CO2(PWM)

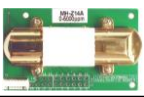
  int ppm ;

  while (digitalRead(pwmPin) == LOW) {} ; //wait for pulse to go high
  long t0 = millis() ; //Keep Time begin pulse High
  while (digitalRead(pwmPin) == HIGH) {} ; //wait for pulse to go low
  long t1 = millis() ; //Keep Time end pulse High
  while (digitalRead(pwmPin) == LOW) {} ; //wait for pulse to go high again
  long t2 = millis() ; //Keep Time begin pulse High cycle2
  long th = t1-t0 ; //Calculate Time for pluse high level
  long tl = t2-t1 ; //Calculate Time for pluse high level
  ppm = 5000L * (th - 2) / (th + t1 - 4) ; //Calculate Value CO2 ppm unit
  while (digitalRead(pwmPin) == HIGH) {} ; //wait for pulse to go low
  delay(10); //allow output to settle.

  return ppm ; }

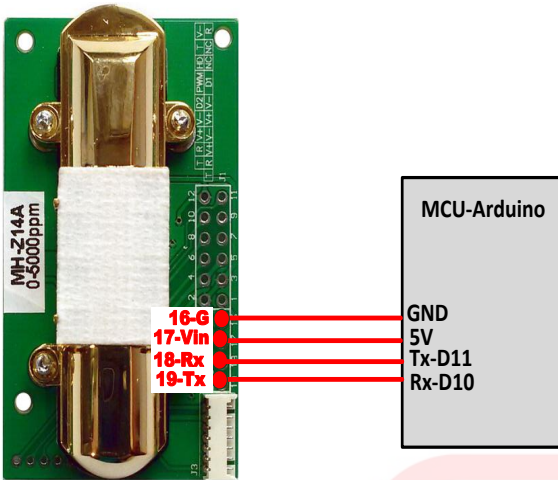
```

ตัวอย่างการต่ออ่านค่า CO2 แบบ PWM



คู่มือ Module MH-Z14A CO2 SENSOR

3.3) **ใช้การต่ออ่านค่าแบบ Digital (Serial-Uart):** โดยตามตัวอย่างจะต่อที่ Pin 19(Tx) และ Pin 18(Rx) หรือ Pin อื่นที่เป็น Pin Rx และ Tx ของ Module Sensor ไปยังขา Rx และ Tx ของ MCU (ต่อแบบไขว้สาย Rx,Tx) ดังรูปตัวอย่างด้านล่าง โดยใช้ MCU-Arduino ซึ่ง Pin Rx,Tx ของ MUC จะต้องเป็นระดับไฟ TTL คือ 5V หรือ 3.3 V ในการเขียนโปรแกรมอ่านค่า แบบ Serial นี้ จะใช้ Baud Rate ในการสื่อสารที่ 9600 ,N, data bit =8, stop bit =1, P เริ่มต้น โปรแกรมก็จะ



ตัวอย่างการต่ออ่านค่า CO2 แบบ Digital

ต้องส่งคำสั่งสำหรับอ่านค่า CO2 (0x86) จำนวน 9 byte ไปให้ Sensor จากนั้นก็ให้โปรแกรมวนรอรับ Data ที่ตัว Sensor ส่งกลับมา 9 byte แล้วนำ data ที่รับมาคำนวณหา ค่า CRC แล้วนำไปเปรียบเทียบกับค่าที่อ่านได้ใน byte 8 ว่าตรงกันหรือไม่ ถ้าไม่ตรงให้ออกจาก function เพื่อกลับไปอ่านค่าใหม่ แต่ถ้าตรงกันแสดงว่าข้อมูลที่รับมาถูกต้อง ให้นำ data ที่รับมาใน Byte2(High) และ Byte3(Low) ซึ่งเป็น Byte ที่เก็บค่า CO2 มารวมเป็นค่าเดียวกันในรูปแบบฐาน 10 ก็จะได้ค่า PPM ส่งออก Function ไปใช้งาน

สำหรับคำสั่งในการอ่านค่า CO2 , การ Return ค่าของ Sensor , การคำนวณค่า CRC มีรูปแบบตามตารางด้านล่าง

การคำนวณค่า CRC

$$CRC = [0xFF - (byte1+byte2+byte3+byte4+byte5+byte6+byte7)] + 0x01$$

การต่ออ่านค่าทั้งแบบข้างต้น แบบ Serial หรือ Uart (Digital Format) จะใช้งานได้ดีสุด ตามด้วยแบบ PWM และ Analog ตามลำดับ โดยการต่อแบบ Serial นั้นจะรองรับระดับสัญญาณ TTL ทั้ง 3.3V และ 5 V ในการอ่านค่ามาแสดงผลถ้าในโปรแกรมไม่มีการหน่วงเวลาเพื่อรอเวลา Sensor preheat ตอนเริ่มต้นจ่ายไฟเลี้ยงให้ sensor ค่าที่อ่านได้อาจจะยังไม่ถูกต้องจริงๆ จนกว่าจะผ่านไป 3 นาทีที่ค่าที่อ่านถึงจะเป็นค่านำไปใช้งานได้ (เมื่อผ่านการ Calibrate Zero แล้ว)

ตัวอย่าง Function โปรแกรม(Arduino)

```
#include <SoftwareSerial.h>

SoftwareSerial sensor(10, 11); // RX, TX

const byte requestReading[] = {0xFF, 0x01, 0x86, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x79}; //Command Read Value Co2

byte result[9] ; //keep Data from CO2 after Sent data to Sensor

void setup() { sensor.begin(9600) ; }

int readPPMSerial() { //Function Read CO2(Digital)

  for (int i = 0; i < 9; i++) {

    sensor.write(requestReading[i]); //Sent Command Read CO2

  }

  while (sensor.available() < 9) {} ; // wait for response

  for(int i = 0; i < 9; i++)

  {

    result[i] = sensor.read(); //Read data from Sensor

  }

  //----- Calculate Check-Sum Data (CRC) -----

  byte crc = 0; int byt ;

  for (byt = 1 ; byt < 8; byt++) {

    crc += result[byt] ; //Add data Respond Byte1 to byte7

  }

  crc = 255 - crc ; //Invert crc

  crc++ ; //crc+1

  if (result [8] != crc) //Check crc From Value cal with Value Response

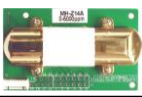
  { return 0 ; } // data error exit function if data true next line

  int high = result[2];

  int low = result[3];

  return high * 256 + low ; //Total Data Byte High-Low then return Value

} //end function
```

0x86 คำสั่งให้อ่านค่า CO2								
Send command								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Starting byte	Sensor No.	command	-	-	-	-	-	Check value
0xFF	0x01	0x86	0x00	0x00	0x00	0x00	0x00	0x79
ค่า Return จาก Sensor								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Starting byte	Sensor No.	High 8 bit concentration	Low 8 bit concentration	-	-	-	-	CRC
0xFF	0x01	HIGH	LOW					CRC
รวมค่า PPM Byte2 กับ byte3(Dec) = (byte2*256)+byte3								

4. การ Calibrate Zero

การนำ Module Sensor ไปใช้งานในครั้งแรกผู้ใช้จะต้องทำการ Calibrate Zero ให้กับ Sensor เสียก่อน โดยค่าเริ่มต้นที่ควรจะอ่านได้ในสภาวะปกติหลัง Calibrate Zero คือ 400 ppm โดย Sensor นี้สามารถ Calibrate Zero ได้ 3 วิธี โดยก่อนทำการ Calibrate Zero ผู้ใช้จะต้องต่อ Sensor ตามวงจรการต่ออ่านค่าดังที่กล่าวไปแล้วข้างต้นแบบใดแบบหนึ่งเพื่อให้ Sensor ทำงาน จากนั้นให้นำ Sensor ไปไว้ในสภาวะแวดล้อมที่ปราศจากก๊าซ CO2 เป็นเวลามากกว่า 20 นาที แล้วจึงทำการ Calibrate Zero ให้ Sensor โดยเลือกใช้วิธีใดวิธีหนึ่งดังนี้

4.1) Calibrate Zero ด้วยมือ : ทำได้โดย ต่อ หรือ Control Pin HD (Pin 8 หรือ 20) ให้เป็น Logic Low หรือต่อไปยัง GND ค้างไว้ประมาณ 7วินาที แล้วปล่อย Pin HD ลอย จากนั้นให้ Power ON Sensor ใหม่ ค่า CO2 ก็ควรจะอ่านได้ประมาณ 400 ppm หลัง Calibrate Zero แล้ว

4.2) Calibrate Zero โดยการส่ง Command : วิธีนี้ผู้ใช้จะต้องต่อ Sensor ให้ทำงาน ตามวงจรการต่ออ่านค่าแบบ Digital แล้วทำการส่ง Command Calibrate Zero (0x87) ไปยัง Sensor โดย Command มีรูปแบบดังนี้

0x87-zero point value calibration command (คำสั่งใช้ Calibrate Zero)								
Send command								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Starting byte	Sensor No.	command	-	-	-	-	-	CRC
0xFF	0x01	0x87	0x00	0x00	0x00	0x00	0x00	0x78
No return value Note: zero point value is 400ppm, before sending the zero point calibration command, pls make sure the sensor running on the environment below 400ppm at least 20 mins								

4.3) Auto Calibrate : วิธีนี้จะถูก Set เป็นค่า default ให้กับ Sensor มาจากผู้ผลิต โดยตัว Sensor จะทำการ Calibrate Zero ด้วยตัวของมันเอง ทุกๆ 24 ชั่วโมงหลังจาก Power-On ซึ่งเหมาะใช้กับ office หรือ ห้องนั่งเล่นเท่านั้น ไม่เหมาะกับ ห้องเรียน กระเจ๊ก , ฟาร์ม หรือ ห้องเย็น ดังนั้นถ้าใช้การ Calibrate Zero ใน 2 วิธีด้านบน จะต้องปิดการ Calibrate ฟังก์ชันนี้ โดยการส่ง Command 0x79 ไปยัง Sensor และกำหนด Byte3 เป็น 0x00 (ในการส่ง command ต้องต่อ Sensor ให้ทำงาน ตามวงจรการต่ออ่านค่าแบบ Digital) โดย Command มีรูปแบบดังนี้

0x79-start/stop auto-calibration function								
Send command (คำสั่งใช้ Start/Stop Auto Calibrate)								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Starting byte	Sensor No.	command			-	-	-	CRC
0xFF	0x01	0x79	0xA0/0x00	0x00	0x00	0x00	0x00	0x86:Stop
No return value Note: Byte3 IS 0xA0 means start open auto-calibration function, byte3 is 0x00 means stop the auto calibration function Before shipment, the default is auto-calibration function is open.								