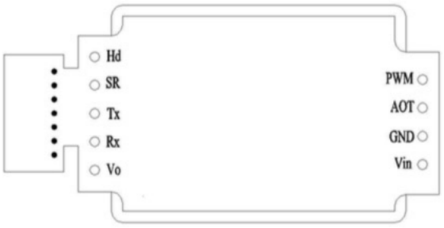
LAB NOTES – ESP8266 (D1 Mini) and MH Z-19

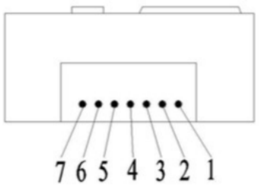
MHZ 19

The MH Z19 is a CO2 sensor that uses NDIR technology to produce pleasingly accurate readings.

All the MH Z19's I've used have pins, and most also have a connector with a ribbon Cable. In most cases I use the ribbon cable as it allows me to play with the positioning of the sensor.

The pinouts are:





Pin	Desc	Colour (Does vary)
1	HD Zero Calibration	Brown
2	Vout (3.3v max 10mA	White
3	GND	Black
4	VIN (5V)	Red
5	RX	Blue
6	TX	Green
7	PWM Out	Yellow

D1 Mini

5V connects to VIN on the MHZ (Pin 4)

GND connects to GND on the MHZ (Pin 3)

D4 is configured as RX (Arduino Pin 2) and connects to TX on the MHZ (Pin 6)

D3 is configured as TX (Arduino Pin 0) and connects to RX on the MHZ (Pin 5)

NOTE – The MH Z19 does occasionally freeze – If you cycle its power it will pretty much always recover.

In our sensors we have code to check for errors – if we get four in a row we de-power it for 4 seconds.

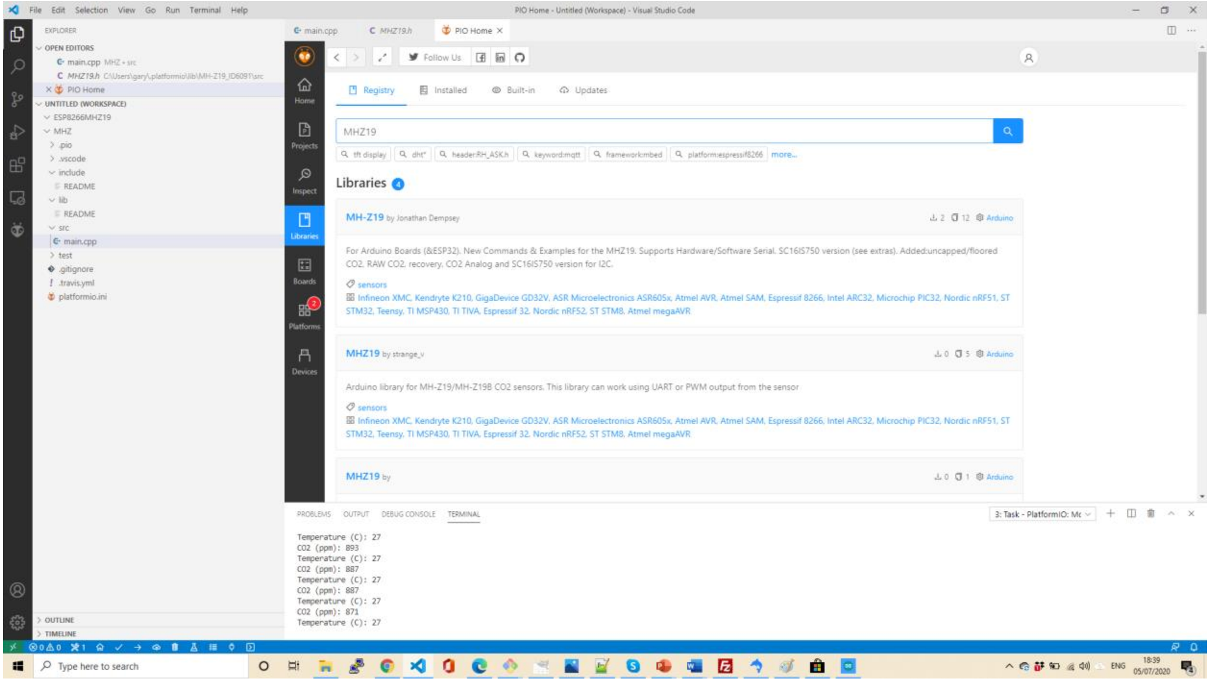
MHZ Library

See John Dempsey's github page - https://github.com/WifWaf/MH-Z19?utm_source=platformio&utm_medium=piohome

Some good info in the README file here.

VS Code – Platform IO (Arduino Framework)

1) Set up the project for your board, choosing the Arduino Framework
2) Install the library -



MHZ19

Libraries 4

MH-Z19 by Jonathan Dempsey — 2 — 12 — Arduino

For Arduino Boards (&ESP32). New Commands & Examples for the MHZ19. Supports Hardware/Software Serial. SC16IS750 version (see extras). Added:uncapped/floored CO2, RAW CO2, recovery, CO2 Analog and SC16IS750 version for I2C.

sensors

Infineon XMC, Kendryte K210, GigaDevice GD32V, ASR Microelectronics ASR605x, Atmel AVR, Atmel SAM, Espressif 8266, Intel ARC32, Microchip PIC32, Nordic nRF51, ST STM32, Teensy, TI MSP430, TI TIVA, Espressif 32, Nordic nRF52, ST STM8, Atmel megaAVR

MHZ19 by strange_v — 0 — 5 — Arduino

Arduino library for MH-Z19/MH-Z19B CO2 sensors. This library can work using UART or PWM output from the sensor

sensors

Infineon XMC, Kendryte K210, GigaDevice GD32V, ASR Microelectronics ASR605x, Atmel AVR, Atmel SAM, Espressif 8266, Intel ARC32, Microchip PIC32, Nordic nRF51, ST STM32, Teensy, TI MSP430, TI TIVA, Espressif 32, Nordic nRF52, ST STM8, Atmel megaAVR

MHZ19 by — 0 — 1 — Arduino

```
Temperature (C): 27
CO2 (ppm): 893
Temperature (C): 27
CO2 (ppm): 887
Temperature (C): 27
CO2 (ppm): 887
Temperature (C): 27
CO2 (ppm): 871
Temperature (C): 27
```

Sample Code:

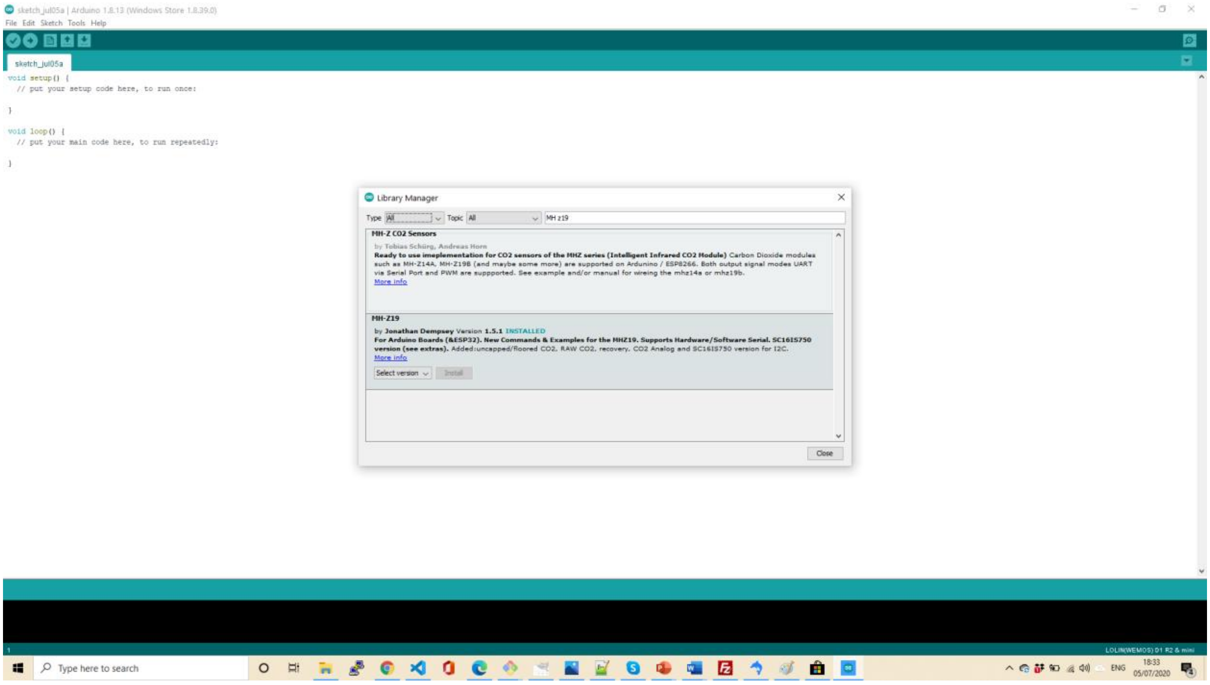
```
#include <Arduino.h>

#include "MHZ19.h"

#include <SoftwareSerial.h>                                // Remove
if using HardwareSerial or Arduino package without SoftwareSerial
support


#define RX_PIN 2                                          // Rx pin
which the MHZ19 Tx pin is attached to

#define TX_PIN 0                                          // Tx pin
which the MHZ19 Rx pin is attached to

#define BAUDRATE 9600                                      // Device
to MH-Z19 Serial baudrate (should not be changed)


MHZ19 myMHZ19;                                             //
Constructor for library


SoftwareSerial mySerial(RX_PIN, TX_PIN);                   // (Uno
example) create device to MH-Z19 serial

//HardwareSerial mySerial(1);                              // (ESP32
Example) create device to MH-Z19 serial
```

```
unsigned long getDataTimer = 0;


void setup()

{

    Serial.begin(9600);                                     //
Device to serial monitor feedback


    mySerial.begin(BAUDRATE);                               // (Uno
example) device to MH-Z19 serial start

    //mySerial.begin(BAUDRATE, SERIAL_8N1, RX_PIN, TX_PIN); //
(ESP32 Example) device to MH-Z19 serial start

    myMHZ19.begin(mySerial);                                //
*Serial(Stream) refence must be passed to library begin().


    myMHZ19.autoCalibration();                              // Turn
auto calibration ON (OFF autoCalibration(false))

}


void loop()

{

    if (millis() - getDataTimer >= 2000)

    {

        int CO2;


        /* note: getCO2() default is command "CO2 Unlimited". This
returns the correct CO2 reading even

        if below background CO2 levels or above range (useful to
validate sensor). You can use the

        usual documented command with getCO2(false) */


        CO2 = myMHZ19.getCO2();                             //
Request CO2 (as ppm)


        Serial.print("CO2 (ppm): ");
        Serial.println(CO2);


        int8_t Temp;
        Temp = myMHZ19.getTemperature();                     //
Request Temperature (as Celsius)
        Serial.print("Temperature (C): ");
        Serial.println(Temp);


        getDataTimer = millis();
    }
}
```

Arduino IDE

1. Select your board
2. Install the library

Install MH Z19 Library – Pick Jonathan Dempsey's Library



sketch_jul05a | Arduino 1.8.13 (Windows Store 1.8.39.0)

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Library Manager

Type: All Topic: All MH z19

MH-Z CO2 Sensors
by Tobias Schürg, Andreas Horn
Ready to use imeplementation for CO2 sensors of the MHZ series (Intelligent Infrared CO2 Module) Carbon Dioxide modules such as MH-Z14A, MH-Z19B (and maybe some more) are supported on Ardunino / ESP8266. Both output signal modes UART via Serial Port and PWM are suppported. See example and/or manual for wireing the mhz14a or mhz19b.
More info

MH-Z19
by **Jonathan Dempsey** Version **1.5.1** INSTALLED
For Arduino Boards (&ESP32). New Commands & Examples for the MHZ19. Supports Hardware/Software Serial. SC16IS750 version (see extras). Added:uncapped/floored CO2, RAW CO2, recovery, CO2 Analog and SC16IS750 version for I2C.
More info

Sample Code:

```
#include "MHZ19.h"

#include <SoftwareSerial.h>                              // Remove
if using HardwareSerial or Arduino package without SoftwareSerial
support


#define RX_PIN 2                                        // Rx pin
which the MHZ19 Tx pin is attached to

#define TX_PIN 0                                        // Tx pin
which the MHZ19 Rx pin is attached to

#define BAUDRATE 9600                                    // Device
to MH-Z19 Serial baudrate (should not be changed)


MHZ19 myMHZ19;                                           //
Constructor for library
```

```
SoftwareSerial mySerial(RX_PIN, TX_PIN);                   // (Uno
example) create device to MH-Z19 serial
//HardwareSerial mySerial(1);                              // (ESP32
Example) create device to MH-Z19 serial

unsigned long getDataTimer = 0;

void setup()
{
    Serial.begin(9600);                                     //
Device to serial monitor feedback

    mySerial.begin(BAUDRATE);                               // (Uno
example) device to MH-Z19 serial start
    //mySerial.begin(BAUDRATE, SERIAL_8N1, RX_PIN, TX_PIN); //
(ESP32 Example) device to MH-Z19 serial start
    myMHZ19.begin(mySerial);                                //
*Serial(Stream) refence must be passed to library begin().

    myMHZ19.autoCalibration();                              // Turn
auto calibration ON (OFF autoCalibration(false))
}

void loop()
{
    if (millis() - getDataTimer >= 2000)
    {
        int CO2;

        /* note: getCO2() default is command "CO2 Unlimited". This
returns the correct CO2 reading even
        if below background CO2 levels or above range (useful to
validate sensor). You can use the
        usual documented command with getCO2(false) */
        CO2 = myMHZ19.getCO2();                             //
Request CO2 (as ppm)


        Serial.print("CO2 (ppm): ");
        Serial.println(CO2);


        int8_t Temp;
        Temp = myMHZ19.getTemperature();                     //
Request Temperature (as Celsius)
        Serial.print("Temperature (C): ");
        Serial.println(Temp);


        getDataTimer = millis();
    }
}
```