

Cozir Sensor Arduino Example

Goal:

This is a guide to getting started using the Cozir sensors with an Arduino/Atmel microcontroller. This example can be ported over to many MCU types, but the provided code and wiring is specifically for use with an Arduino. When the example is finished we will have a device that can get a co2 reading from a Cozir gss 20% and report the value back to the computer over USB.

Previous Knowledge:

While it is possible to follow this guide with no experience, it would be helpful to be familiar with the following:

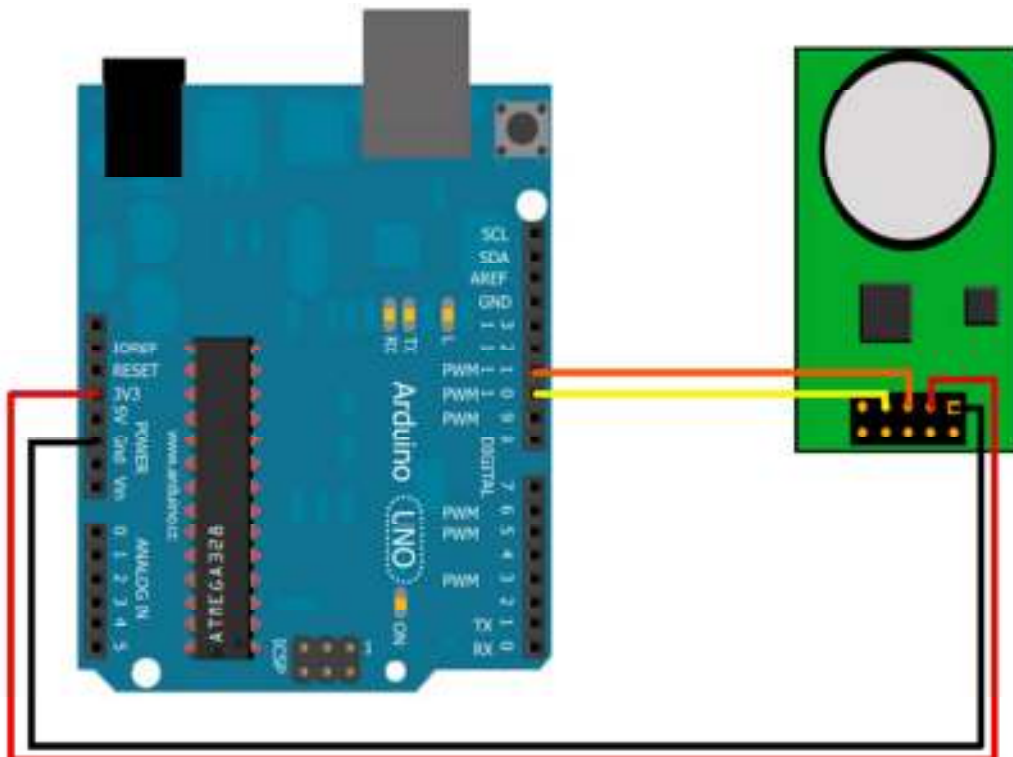
- Basic Arduino use
- The “SoftwareSerial.h” Library for arduino
- Cozir UART communication

Wiring:

The first step is to wire up the sensor to your Arduino. This example shows the Arduino UNO with a Gss 20% sensor, but any of the Arduino series will work. Just make sure to select the proper board under Tools>>Board>> in your Arduino IDE.

1) Make the following Connections:

Wire	Arduino	Sensor
Red	3.3V	3.3v
Black	GND	0v
Yellow	Digital pin #11	Tx
Orange	Digital pin #10	Rx



Writing the Code:

The next step is to write the code to drive the device. This sample code records values from the sensor and reports them back a host computer over usb.

```
/*
```

Cozir Sample code

Written by: Jason Berger (for Co2meter.com)

This sketch connects will connect to a Cozir gss or SprintIR sensor and report readings back to the host computer over usb. The value is stored in a global variable 'co2' and can be used for any number of applications.

pin connections:

Arduino_____Cozir Sensor

GND ----- 1 (gnd)

3.3v----- 3 (Vcc)

10 ----- 5 (Rx)

11 ----- 7 (Tx)

```
*/
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial mySerial(10, 11); // RX, TX
```

```
String val= ""; //holds the string of the value
```

```
double co2 =0; // holds the actual value
```

```
double multiplier = 10; //each range of sensor has a different value.
```

```
    // up to 2% =1
```

```
    // up to 65% = 10
```

```
    //up to 100% = 100;
```

```
uint8_t buffer[25];
```

```
uint8_t ind =0;
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);           //Start Serial connection with host
```

```
    Serial.println("Co2Meter.com Cozir Sample");
```

```
mySerial.begin(9600);           //Start Serial connection with Sensor

}

void loop()
{
  //Cozir sensors ship from the factory in streaming mode
  //So we read incoming bytes into a buffer until we get '0x0A' which is the ASCII value for new-line
  while(buffer[ind-1] != 0x0A)
  {
    if(mySerial.available())
    {
      buffer[ind] = mySerial.read();
      ind++;
    }
  }

  report(); //Once we get the '0x0A' we will report what is in the buffer
}

void report()
{
  //Cycle through the buffer and send out each byte including the final linefeed

  /*
  each packet in the stream looks like "Z 00400 z 00360"
  'Z' lets us know its a co2 reading. the first number is the filtered value
  and the number after the 'z' is the raw value.
  We are really only interested in the filtered value

  */

  for(int i=0; i < ind+1; i++)
  {
    if(buffer[i] == 'z') //once we hit the 'z' we can stop
      break;

    if((buffer[i] != 0x5A)&&(buffer[i] != 0x20)) //ignore 'Z' and white space
    {
      val += buffer[i]-48; //because we break at 'z' the only bytes getting added are the numbers
                          // we subtract 48 to get to the actual numerical value
                          // example the character '9' has an ASCII value of 57. [57-48=9]
    }
  }
}
```

```
}  
}  
  
    co2 = (multiplier * val.toInt()); //now we multiply the value by a factor specific ot the sensor. see the  
Cozir software guide  
  
    Serial.print( "Co2 = ");  
    Serial.print(co2);  
    Serial.println(" ppm");  
    ind=0; //Reset the buffer index to overwrite the previous packet  
    val=""; //Reset the value string  
}
```