### Non-Contact Infrared Temperature Sensor GY-906

**ARD2-2190**

- Detect temperatures between 70°C–380°C without contact using infrared light
- Perfect for use with Arduino, Raspberry Pi and other microcontrollers
- Internal 17-bit ADC
- PWM and TWI outputs
- Resolution down to 0.02°C (TWI output)

**Description**

A non-contact IR (infra-red) temperature sensor, capable of sensing between 70°C–380°C down to a resolution of 0.02°C. This module is based on the MELEXIS MLX90614ESF-BAA-000-TU-ND temperature sensor.

Possible applications include: temperature sensing for residential, commercial and industrial air conditioning; windshield defogging; home appliances with temperature control; livestock monitoring; and automotive blind angle detection.

Includes unsoldered 4-pin header.

**Specifications**

<table>
<thead>
<tr>
<th>Power Supply</th>
<th>5V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>GY-906</td>
</tr>
<tr>
<td>Sensor</td>
<td>MLX90614</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>40°C–125°C</td>
</tr>
<tr>
<td>Temperature Sensing Range</td>
<td>70°C–380°C</td>
</tr>
<tr>
<td>Measurement Resolution (PWM Output)</td>
<td>0.14°C</td>
</tr>
<tr>
<td>Measurement Resolution (TWI Output)</td>
<td>0.02°C</td>
</tr>
<tr>
<td>Colour</td>
<td>Blue</td>
</tr>
<tr>
<td>Material</td>
<td>Immersion Gold PCB</td>
</tr>
<tr>
<td>Weight</td>
<td>3g</td>
</tr>
</tbody>
</table>
**Test Code for Arduino**

```c
#include <i2cmaster.h>

void setup(){
    Serial.begin(9600);
    Serial.println("Setup...");
    i2c_init(); //Initialise the i2c bus
    PORTC = (1 << PORTC4) | (1 << PORTC5);//enable pullups
}

void loop(){
    int dev = 0x5A<<1;
    int data_low = 0;
    int data_high = 0;
    int pec = 0;
    i2c_start_wait(dev+I2C_WRITE);
    i2c_write(0x07);
    // read
    i2c_rep_start(dev+I2C_READ);
    data_low = i2c_readAck(); //Read 1 byte and then send ack
    data_high = i2c_readAck(); //Read 1 byte and then send ack
    pec = i2c_readNak();
    i2c_stop();
    //This converts high and low bytes together and processes
    temperature, MSB is an error bit and is ignored for temps
    double tempFactor = 0.02; // 0.02 degrees per LSB
    (measurement resolution of the MLX90614)
    double tempData = 0x0000; // zero out the data
    int frac; // data past the decimal point
    // This masks off the error bit of the high byte, then
    moves it left 8 bits and adds the low byte.
    tempData = (double)(((data_high & 0x007F) << 8) + data_
    low);
    tempData = (tempData * tempFactor)-0.01;
    float celcius = tempData - 273.15;
    float fahrenheit = (celcius*1.8) + 32;
    Serial.print("Celcius: ");
    Serial.println(celcius);
    Serial.print("Fahrenheit: ");
    Serial.println(fahrenheit);
    delay(1000); // wait a second before printing again
}
```

**Source:** http://bildr.org/2011/02/mlx90614-arduino/

**NOTE:** The I2Cmaster library required to run this code can be downloaded for free here.

To make this code work, before you load the code, or even open the Arduino program, the “I2Cmaster” folder must be placed in your Arduino Library.

**Default Library Folder Locations**

- **Mac:** In (home directory)/Documents/Arduino/libraries
- **PC:** My Documents -> Arduino -> libraries
- **Linux:** (home directory)/sketchbook/libraries

---

**Pinout (except Arduino MEGA)**

<table>
<thead>
<tr>
<th>Module</th>
<th>Arduino</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIN</td>
<td>5V</td>
<td>Power Supply</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>Ground Connection</td>
</tr>
<tr>
<td>SCL</td>
<td>A5</td>
<td>Clock</td>
</tr>
<tr>
<td>SDA</td>
<td>A4</td>
<td>Data</td>
</tr>
</tbody>
</table>

**Pinout (Arduino MEGA)**

<table>
<thead>
<tr>
<th>Module</th>
<th>Arduino</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIN</td>
<td>5V</td>
<td>Power Supply</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>Ground Connection</td>
</tr>
<tr>
<td>SCL</td>
<td>D21</td>
<td>Clock</td>
</tr>
<tr>
<td>SDA</td>
<td>D20</td>
<td>Data</td>
</tr>
</tbody>
</table>