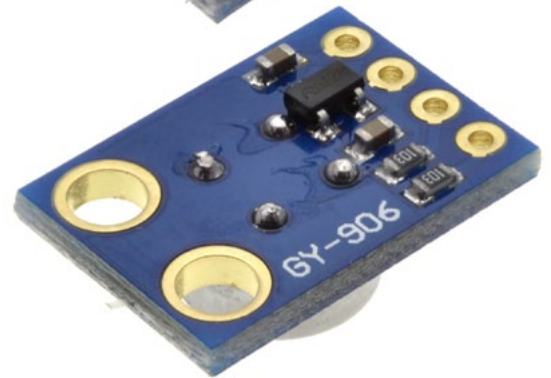
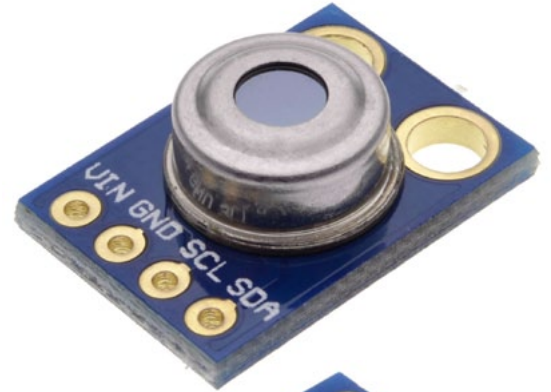


ARD 2 **Arduino Compatibles** Controllers, Shields, Modules & Sensors

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

Power Supply	5V
Model	GY-906
Sensor	MLX90614
Operating Temperature	40°C–125°C
Temperature Sensing Range	70°C–380°C
Measurement Resolution (PWM Output)	0.14°C
Measurement Resolution (TWI Output)	0.02°C
Colour	Blue
Material	Immersion Gold PCB
Weight	3g

Test Code for Arduino

```
#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 a 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)

Module	Arduino	Function
VIN	5V	Power Supply
GND	GND	Ground Connection
SCL	A5	Clock
SDA	A4	Data

Pinout (Arduino MEGA)

Module	Arduino	Function
VIN	5V	Power Supply
GND	GND	Ground Connection
SCL	D21	Clock
SDA	D20	Data