IR Thermometry Module

BMH06203
Arduino Library V1.0.1 Description

Revision: V1.01 Date: September 07, 2023

www.bestmodulescorp.com

Contents

Introduction

The Best Modules BMH06203 is an IR thermometry module, which uses the I²C communication method. This document provides the description of the BMH06203 Arduino Lib functions and how to install the Arduino Lib. The example demonstrates the function of obtaining the object surface temperature value.

Arduino Lib Functions

Arduino Lib Name: BMH06203		Lib Version: V1.0.1
Constructors & Initialisation		
1	BMH06203(TwoWire *theWire=&Wire)	
	Description	Constructor
	Parameter	*theWire: wire parameter
	Return Value	—
	Note	—
2	void begin(uint8_t i2c_addr=BMH06203_ADDR)	
	Description	Module initialisation
	Parameter	i2c_addr: I²C communcation address, 0x28
	Return Value	void
	Note	—
Performance Functions		
3	float readTemperature(uint8_t TYPE)	
	Description	Obtain the temperature data
	Parameter	TYPE: data type 0x08 (AMB_TEMP): ambient temperature 0x09 (OBJ_TEMP): object surface temperature 0x0A (BODY_TEMP): body temperature
	Return Value	Temperature value, unit: °C
	Note	—
4	void sleep()	
	Description	Set the module to enter the sleep mode
	Parameter	—
	Return Value	void
	Note	—
5	void writeEEPROM(uint8_t addr, uint16_t data)	
	Description	Write the EEPROM
	Parameter	addr: EEPROM address data: data to be written
	Return Value	void
	Note	This function can set the operating mode, transmission rate, mode parameters, etc. For details, refer to the BMH06203 IR thermometry module datasheet
6	uint16_t readEEPROM(uint8_t addr)	
	Description	Read the EEPROM
	Parameter	addr: EEPROM address
	Return Value	Data read from the EEPROM
	Note	This function can read the operating mode, transmission rate, mode parameters, etc. For details, refer to the BMH06203 IR thermometry module datasheet

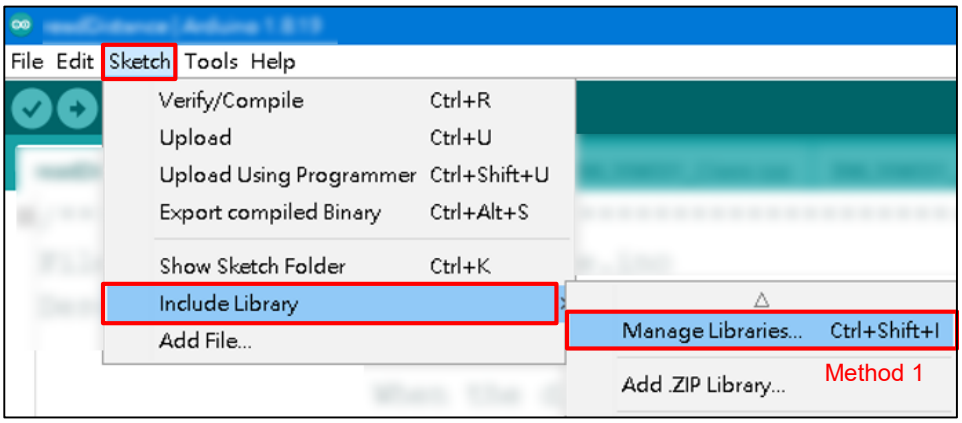
7	void setMode(uint8_t Mode)	
	Description	Set the output mode
	Parameter	Mode: output mode 0x00 (IIC_MODE): I²C mode (default) 0x01 (PWM_MODE): PWM mode 0x02 (IO_MODE1): I/O mode 1 0x06 (IO_MODE2): I/O mode 2
	Return Value	void
	Note	After the output mode is changed, it cannot be used until power on again
Parameter Configuration		
8	void setPWMParam(uint8_t min, uint8_t max)	
	Description	Set the minimum and maximum temperature threshold values for PWM mode waveform output
	Parameter	min: minimum temperature threshold, unit: °C max: maximum temperature threshold, unit: °C
	Return Value	void
	Note	PWM output=(measured temperature-min)/(max-min)
9	void setIOParam(uint8_t threshold)	
	Description	Set the temperature threshold for I/O modes
	Parameter	threshold: temperature threshold, unit: °C
	Return Value	void
	Note	In the IO_MODE1 mode, when measured temperature ≥ temperature threshold, output low, otherwise output high In the IO_MODE2 mode, when measured temperature ≥ temperature threshold, output high, otherwise output low

Arduino Lib Download and Installation

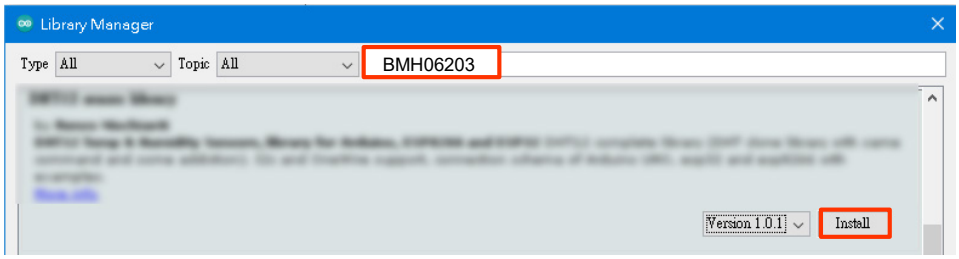
BMH06203Library: Refer to the following two methods to install the BMH06203 Arduino Library.

Method 1: Search for installation

Arduino IDE→Sketch→Include Library→Manage Libraries...→Search BMH06203→Install



Search for Installation Step 1

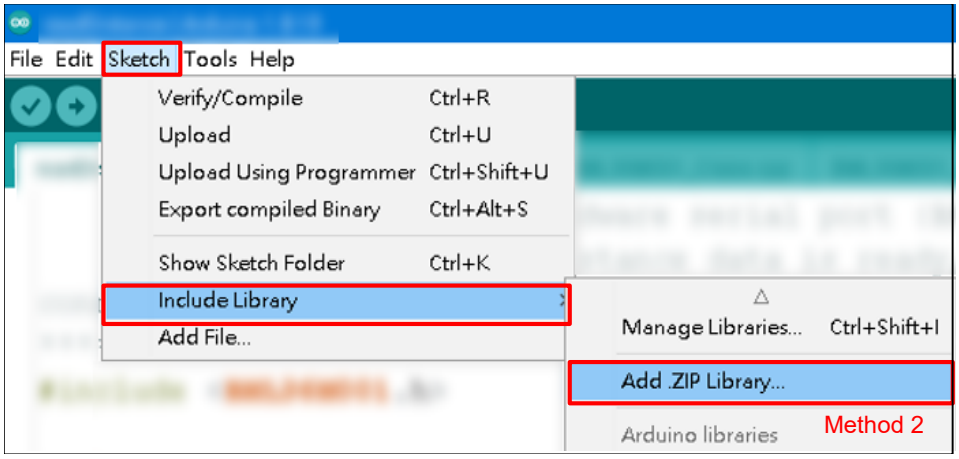


Search for Installation Step 2

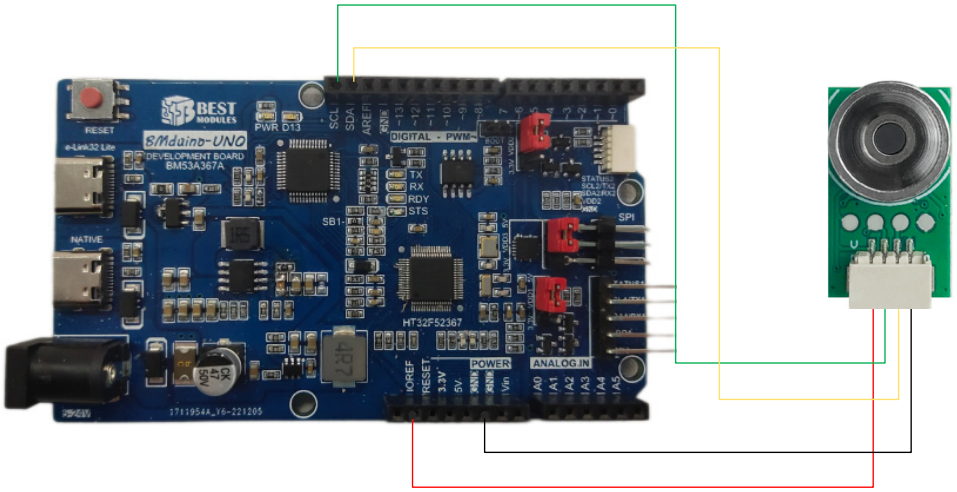
Method 2: Download the .ZIP library before adding it

Download the Arduino example (BMH06203 Library) under the DOCUMENTS menu from the Best Modules website (https://www.bestmodulescorp.com/bmh06203.html).

Add .ZIP library: Arduino IDE→Sketch→Include Library→Add .ZIP Library...



Arduino Examples



Physical Connection Diagram

Example 1: setMode_I2C

Example 1 function: Set the module to be in the I²C mode

1. Open the example: File→Examples→Select Lib (BMH06203)→Select example (setMode_I2C)
2. Example Description:

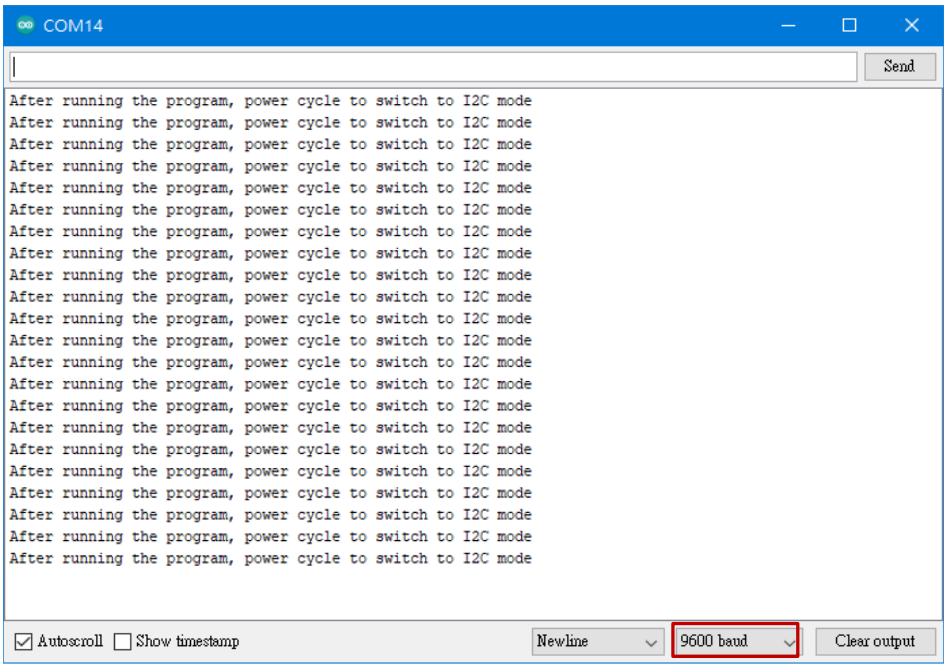
a. Create object & initialise object

```
#include "BMH06203.h"
BMH06203 mytherm(&Wire); // Create object
void setup()
{
  /* Switch other modes to I²C output mode */
  pinMode(19,OUTPUT);
  digitalWrite(19,LOW);  // Pull the module SCL pin low for 50ms
  delay(50);     // Wait 50ms and write the EEPROM to set to I²C mode
  mytherm.setMode(IIC_MODE);
  mytherm.begin();
  Serial.begin(9600);
}
```

b. On the serial monitor, prompt users to power on the module again

```
void loop()
{
  Serial.print("After running the program, power cycle to switch to
  I2C mode");
  // Power on again to switch to I²C mode
}
```

3. Open the serial monitor and set the baud rate to be 9600. The serial monitor will display as follows.



Example 2: readTemperature

Example 2 function: Obtain the object surface temperature and display it on the serial port monitor

In this example, reading the temperature value operates in the I²C mode:

If the module is in the I²C mode, run the example directly.

If the module is not in the I²C mode, run example 1 to switch the mode to I²C mode, then power on it again and run the example.

1. Open the example: File→Examples→Select Lib (BMH06203)→Select example (readTemperature)
2. Example Description:
   a. Create object & initialise object

```
#include "BMH06203.h"
BMH06203 mytherm(&Wire); // Create object
void setup()
{
  mytherm.begin();
  Serial.begin(9600);    // Module initialisation
}
```

b. Obtain the object surface temperature value and display it on the serial monitor

```
void loop()
{
  // Obtain the object surface temperature value
  Serial.print("The OBJTemp is:");
  Serial.print(mytherm. readTemperature(OBJ_TEMP));
  Serial.println("°C");
}
```

3. Open the serial monitor and set the baud rate to be 9600. The serial monitor will display as follows.