Ambient Light Detection Module

BME82M131
Arduino Library V1.0.2 Description

Revision: V1.10 Date: July 16, 2024

www.bestmodulescorp.com

Contents

Introduction

The Best Modules BME82M131 is an ambient light detection module, which uses the I²C communication method. This document describes the Arduino Lib function of the BME82M131 and how to install the Arduino Lib. The example demonstrates how to obtain the ambient light intensity.

Arduino Lib Functions

Arduino Lib Name: BME82M131		Lib Version: V1.0.2
Constructors & Initialisation		
1	BME82M131(uint8_t intPin=2, TwoWire *theWire=&Wire)	
	Description	Constructor
	Parameter	intPin: INT pin, connect to the module INT pin *theWire: Select the I²C interface
	Return Value	—
	Note	—
2	void begin(uint8_t i2c_addr=0x48)	
	Description	Module initialisation
	Parameter	i2c_addr: I²C address
	Return Value	void
	Note	—
Performance Functions		
3	uint8_t getINT()	
	Description	Obtain INT pin level
	Parameter	—
	Return Value	INT pin level 1: High level, the ALS is between the high threshold and low threshold 0: Low level, the ALS is above the high threshold or below the low threshold
	Note	The ALS interrupt status can be selected. When the ALS interrupt is disabled, the INT pin is always high level
4	uint8_t getNumber()	
	Description	Obtain the cascade module number
	Parameter	—
	Return Value	Cascade module number
	Note	—
5	uint16_t readALS(uint8_t sensornumber)	
	Description	Obtain the specified module ALS value
	Parameter	sensornumber: Module ID
	Return Value	ALS value
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
6	uint16_t readWhite(uint8_t sensornumber)	
	Description	Obtain the specified module white-balance value
	Parameter	sensornumber: Module ID
	Return Value	White-balance value
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide

7	float readLux(uint8_t sensorNmuber)	
	Description	Obtain the specified module ambient light intensity
	Parameter	sensorNmuber: Module ID
	Return Value	Ambient light intensity, unit: lx
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide Ambient light intensity=ALS output data/ALS gain×Responsivity, unit: lx Refer to the Communication Protocol chapter of the user guide
8	uint8_t getInterruptFlag(uint8_t sensorNmuber)	
	Description	Obtain the specified module interrupt flag
	Parameter	sensorNmuber: Module ID
	Return Value	Interrupt status 0: INT_STATUS_NONE (Low threshold interrupt flag=0, high threshold interrupt flag=0) 1: INT_STATUS_HIGH (high threshold interrupt flag=1) 2: INT_STATUS_LOW (Low threshold interrupt flag=1) 3: INT_STATUS_BOTH (Low threshold interrupt flag=1, high threshold interrupt flag=1)
	Note	The interrupt flag will be automatically cleared after using this function ID: Refer to the Multi-board Cascade chapter of the user guide
Performance Functions – Parameter Set/Obtain		
9	uint8_t getPowerSavingMode(uint8_t sensorNmuber)	
	Description	Obtain the specified module power-saving mode
	Parameter	sensorNmuber: Module ID
	Return Value	Power-saving mode 0: POWER_SAVING_MODE1 1: POWER_SAVING_MODE2 2: POWER_SAVING_MODE3 3: POWER_SAVING_MODE4
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
10	uint8_t getPowerSavingModeStatus(uint8_t sensorNmuber)	
	Description	Obtain the specified module power-saving mode status
	Parameter	sensorNmuber: Module ID
	Return Value	Status 0: Disable 1: Enable
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
11	uint8_t getMoudleStatus(uint8_t sensorNmuber)	
	Description	Obtain the specified module ALS switch status
	Parameter	sensorNmuber: Module ID
	Return Value	Status 0: Open 1: Close
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
12	uint8_t isInterruptEnabled(uint8_t sensorNmuber)	
	Description	The specified module ALS interrupt status
	Parameter	sensorNmuber: Module ID
	Return Value	Status 0: Interrupt disabled 1: Interrupt enabled
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
13	uint8_t getPersistence(uint8_t sensorNmuber)	
	Description	Obtain the number of ALS persistent protections for the specified module
	Parameter	sensorNmuber: Module ID
	Return Value	The number of persistent protections 0: 1 1: 2 2: 4 3: 8
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide

14	uint8_t getIntegrationTime(uint8_t sensorNmuber)	
	Description	Obtain the specified module ALS integration time
	Parameter	sensorNmuber: Module ID
	Return Value	Integration time 0: 25ms 1: 50ms 2: 100ms 3: 200ms 4: 400ms 5: 800ms
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
15	uint8_t getGain(uint8_t sensorNmuber)	
	Description	Obtain the specified module ALS gain
	Parameter	sensorNmuber: Module ID
	Return Value	Gain 0: 1 1: 2 2: 1/8 3: 1/4
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
16	uint16_t getHighThreshold(uint8_t sensorNmuber)	
	Description	Obtain the specified module ALS high threshold
	Parameter	sensorNmuber: Module ID
	Return Value	The high threshold of the sensor
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
17	uint16_t getLowThreshold(uint8_t sensorNmuber)	
	Description	Obtain the specified module ALS low threshold
	Parameter	sensorNmuber: Module ID
	Return Value	Low threshold data
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
18	bool setPowerSavingMode(uint8_t sensorNmuber, PSM_MODE_t mode)	
	Description	Set the specified module power-saving mode
	Parameter	sensorNmuber: Module ID mode: Power-saving mode selection 0x00 (POWER_SAVING_MODE1): MODE1 0x01 (POWER_SAVING_MODE2): MODE2 0x02 (POWER_SAVING_MODE3): MODE3 0x03 (POWER_SAVING_MODE4): MODE4
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide MODE4 is the strongest; MODE1 is the weakest
19	bool setPowerSavingModeStatus(uint8_t sensorNmuber, PSMEN_t psmen)	
	Description	Set the specified module switch status of the power-saving mode
	Parameter	sensorNmuber: Module ID psmen: Status selection 0x00 (POWER_SAVING_MODE_DISABLE): Disable 0x01 (POWER_SAVING_MODE_ENABLE): Enable
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide

20	bool ALSOn(uint8_t sensorNmuber)	
	Description	Open the specified module ALS
	Parameter	sensorNmuber: Module ID
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
21	bool ALSDown(uint8_t sensorNmuber)	
	Description	Close the specified module ALS
	Parameter	sensorNmuber: Module ID
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
22	bool setInterruptStatus(uint8_t sensorNmuber, INTEN_t ie)	
	Description	Set the specified module ALS interrupt switch status
	Parameter	sensorNmuber: Module ID ie: Status selection 0x00 (INT_DISABLE): Disable 0x01 (INT_ENABLE): Enable
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
23	bool setPersistence(uint8_t sensorNmuber, PERS_t pers)	
	Description	Set the number of ALS persistent protections for the specified module
	Parameter	sensorNmuber: Module ID pers: The number of persistent protections 0x00 (PERS_1): 1 0x01 (PERS_2): 2 0x02 (PERS_4): 4 0x03 (PERS_8): 8
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
24	bool setIntegrationTime(uint8_t sensorNmuber, IT_TIME_t it)	
	Description	Set the specified module ALS integration time
	Parameter	sensorNmuber: Module ID it: Integration time selection 0x00 (IT_25ms): 25ms 0x01 (IT_50ms): 50ms 0x02 (IT_100ms): 100ms 0x03 (IT_200ms): 200ms 0x04 (IT_400ms): 400ms 0x05 (IT_800ms): 800ms
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide The longer the ALS integration time, the lower the resolution The resolution is as low as 0.0036lx/ct

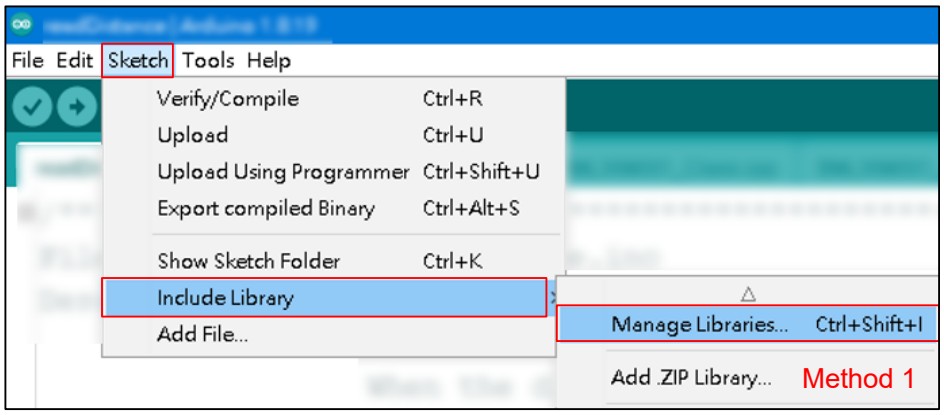
25	bool setGain(uint8_t sensorNmuber, GAIN_t gain)	
	Description	Set the specified module ALS gain
	Parameter	sensorNmuber: Module ID gain: Gain selection 0x00 (GAIN_1): 1 0x01 (GAIN_2): 2 0x02 (GAIN_1_8): 1/8 0x03 (GAIN_1_4): 1/4
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
26	bool setHighThreshold(uint8_t sensorNmuber, uint16_t threshold)	
	Description	Set the specified module ALS high threshold
	Parameter	sensorNmuber: Module ID threshold: 0x0000~0xffff
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide
27	bool setLowThreshold(uint8_t sensorNmuber, uint16_t threshold)	
	Description	Set the specified module ALS low threshold
	Parameter	sensorNmuber: Module ID threshold: 0x0000~0xffff
	Return Value	Perform result true: Succeeded false: Failed
	Note	ID: Refer to the Multi-board Cascade chapter of the user guide

Arduino Lib Download and Installation

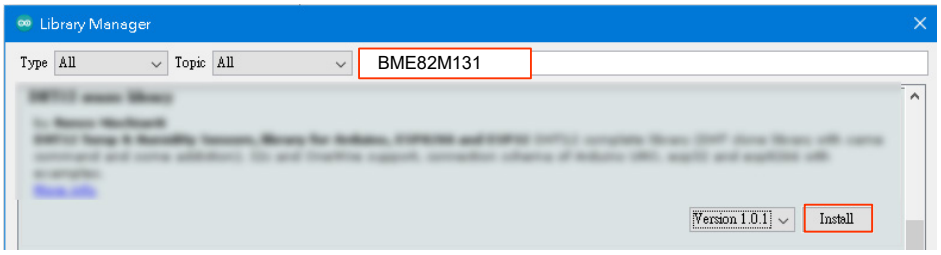
BME82M131 Library: Refer to the following two methods to install the BME82M131 Arduino Library.

Method 1: Search for installation

Arduino IDE→Sketch→Include Library→Manage Libraries...→Search BME82M131→Install



Search for Installation Step 1

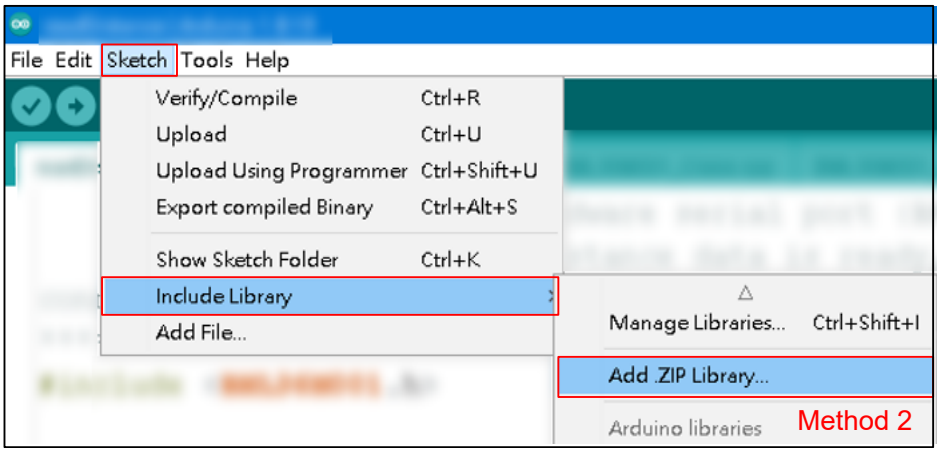


Search for Installation Step 2

Method 2: Download the .ZIP library before adding it

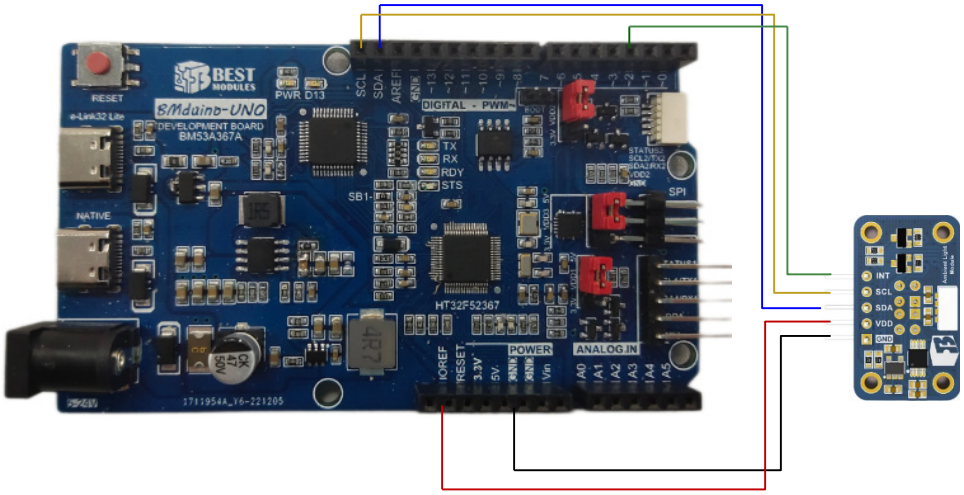
Download the Arduino example (BME82M131 Library) under the DOCUMENTS menu from the Best Modules website (https://www.bestmodulescorp.com/bme82m131.html)

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



Arduino Example

Example: readLux



Physical Connection Diagram

Example implemented function: Obtain the light intensity and display it on the serial port monitor.

1. Open the example: Arduino IDE→File→Examples→Select Lib (BME82M131)→Select example (readLux)
2. Example Description:
   a. Create object & initialise module

```
#include <BME82M131.h>
BME82M131 ALS;
void setup()
{
  Serial.begin(9600);             // Configure the serial monitor
  ALS.begin();                    // Module initialisation
  Serial.print(ALS.getNumber()); // Obtain the number of connected
                                  // module
  Serial.print(" modules are ");
  Serial.println("Connected!");
  Serial.println("modle is Connected!");
}
```

   b. Obtain the light intensity data and display it in the serial monitor

```
void loop()
{
  Serial.print("Lux1: ");
  Serial.print(ALS.readLux(1)); // Obtain the light intensity from the
                                // module with ID=1 and print
  delay(1000);
}
```

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