



PIR Digital Sensor

BM22S402x-1

Arduino Library V1.0.1 Description

Revision: V1.01 Date: September 11, 2023

www.bestmodulescorp.com

Contents

Introduction	3
Arduino Lib Functions	3
Arduino Lib Download and Installation	6
Arduino Example	7
Example: LEDIndicatesTriggerState	7

Introduction

The BM22S402x-1 (x=1, 2, 3, 4) are four PIR digital sensors from Best Modules, which use the UART communication method. This document provides the description of the BM22S402x-1 Arduino Lib functions and how to install the Arduino Lib. The example uses the BMA26M221 to demonstrate the functions of module parameter configuration and PIR motion detection.

Applicable types:

Part No.	Description
BM22S4021-1 BM22S4022-1 BM22S4023-1 BM22S4024-1	PIR Digital Sensor
BMA26M221	On-board BM22S4021-1 Sensor

Arduino Lib Functions

Arduino Lib Name: BM22S402x-1		Lib Version: V1.0.1
Constructors & Initialisation		
1	BM22S402x_1(HardwareSerial *theSerial=&Serial)	
	Description	Constructor, uses the Hardware Serial interface
	Parameter	*theSerial: Select Hardware Serial interface (default Serial interface)
	Return Value	---
2	BM22S402x_1(uint8_t rxPin, uint8_t txPin)	
	Description	Constructor, uses the Software Serial interface
	Parameter	rxPin: RX pin, connects to the TX pin of the BM22S402x-1 or the BMA26M221 txPin: TX pin, connects to the RX pin of the BM22S402x-1 or the BMA26M221
	Return Value	---
3	void begin()	
	Description	Module initialisation
	Parameter	---
	Return Value	void
Performance Functions		
4	uint8_t getDevID(uint8_t devID[])	
	Description	Get the module ID
	Parameter	devID[]: Store the module ID (ASCII code), length: 10 bytes
	Return Value	Execution result: 0: Execution succeeded 1: Response verification error 2: Communication timeout
Note	The module ID is fixed to BM22S402x1 (converted to characters based on ASCII code table)	

5	uint16_t readCommand(uint8_t cmd)	
	Description	Read the module parameters
	Parameter	cmd: "Read" type command code
	Return Value	Module parameter
	Note	This function can read the STATUS register value, PIR register value, real-time PIR value, etc. Refer to the BM22S402x-1 datasheet for details.
6	bool isStable()	
	Description	Query if the module is stable
	Parameter	—
	Return Value	Module stable status: true: Stable, warm-up is completed false: Unstable, warm-up is not completed
	Note	A warm-up time of 30s is required after the module is powered on
7	bool isTrigger()	
	Description	Query if the module has detected the object
	Parameter	—
	Return Value	Object detection status: true: Object detected false: No object detected
	Note	—
8	bool isInfoAvailable()	
	Description	Query if the module automatic output information has been received
	Parameter	—
	Return Value	Information receiving status: true: Received false: Not received
	Note	—
9	void readInfopacket(uint8_t dataBuf[])	
	Description	Read the module automatic output information (7 bytes)
	Parameter	dataBuf[0], dataBuf[1]: Original PIR value low byte, original PIR value high byte dataBuf[2], dataBuf[3]: Filtered PIR value low byte, filtered PIR value high byte dataBuf[4]: STATUS register value dataBuf[5], dataBuf[6]: Temperature value low byte, temperature value high byte (unit: 0.1°C)
	Return Value	void
	Note	This function should be used after "isInfoAvailable()==true"
10	uint16_t readPIR()	
	Description	Read the filtered PIR value
	Parameter	—
	Return Value	12-bit A/D value
	Note	—
11	uint16_t readRawPIR()	
	Description	Read the original PIR value
	Parameter	—
	Return Value	12-bit A/D value
	Note	—
12	float readTemperature(bool isFahrenheit=false)	
	Description	Read the temperature value
	Parameter	isFahrenheit: Unit selection true: Fahrenheit degree false: Centigrade
	Return Value	Temperature value
	Note	—

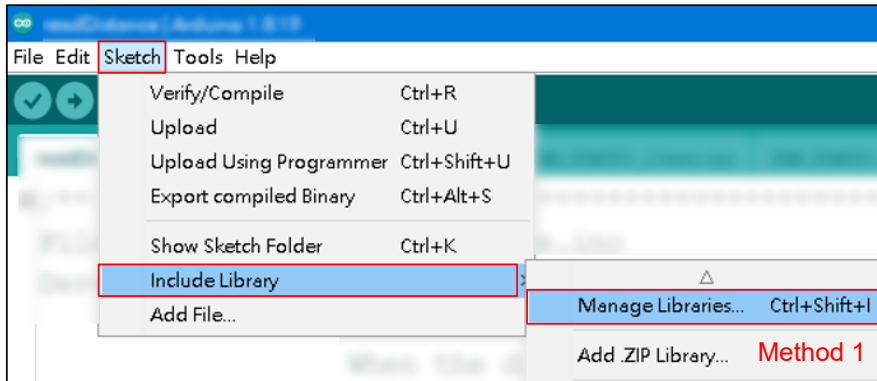
Module Configuration	
13	uint8_t writeCommand(uint8_t cmd, uint16_t param)
	Description Configure the module according to command
	Parameter cmd: "Write" type command code param: Parameters to be written
	Return Value Execution result: 0: Execution succeeded 1: Response verification error 2: Communication timeout 3: Command code error 4: Write failed
Note This function allows users to write parameters to the PIR configuration register as well as set trigger sensitivity and other parameters. Refer to the BM22S402x-1 datasheet for details	
14	uint8_t enablePIR(bool isEnabled=true)
	Description Enable or disable PIR detection
	Parameter isEnabled: Enabling status true: Enable PIR detection false: Disable PIR detection
	Return Value Execution result: 0: Execution succeeded 1: Response verification error 2: Communication timeout 3: Command code error 4: Write failed
Note —	
15	uint8_t reset()
	Description Software reset
	Parameter —
	Return Value Execution result: 0: Execution succeeded 1: Response verification error 2: Communication timeout 3: Command code error
Note Refer to the BM22S402x-1 datasheet for the changes in some register values after reset	
16	uint8_t restoreDefault()
	Description Restore the PIR register and sensitivity register to the default values in the datasheet
	Parameter —
	Return Value Execution result: 0: Execution succeeded 1: Response verification error 2: Communication timeout 3: Command code error 4: Write failed
Note —	
17	uint8_t sleep()
	Description Enter the sleep mode
	Parameter —
	Return Value Execution result: 0: Execution succeeded 1: Response verification error 2: Communication timeout 3: Command code error
Note Sending any command will wake up the module in the sleep mode	

Arduino Lib Download and Installation

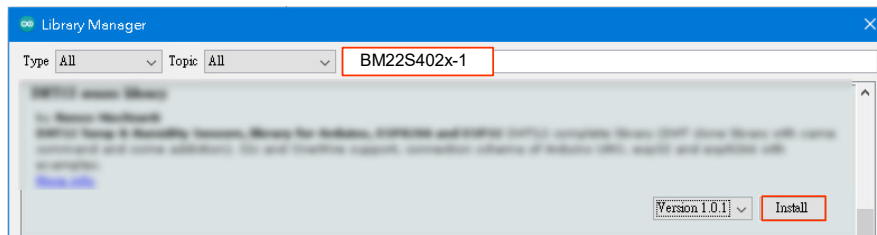
BM22S402x-1 Library: Refer to the following two methods to install the BM22S402x-1 Arduino Library.

Method 1: Search for installation

Search for installation: Arduino IDE → Sketch → Include Library → Manage Libraries... → Search BM22S402x-1 → Install



Search for Installation Step 1

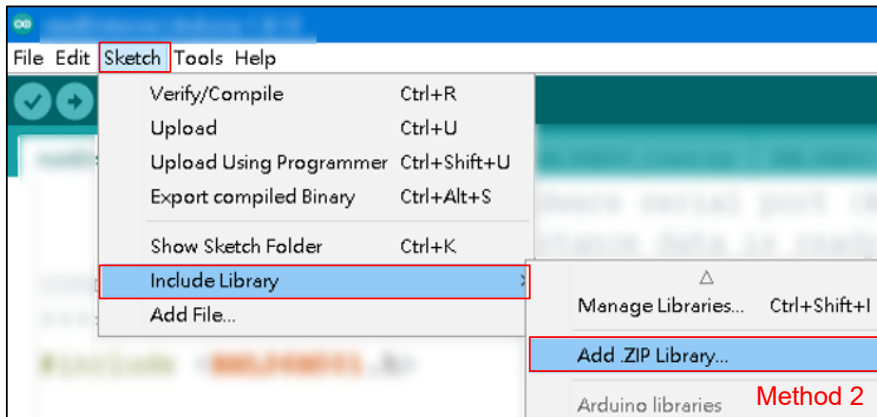


Search for Installation Step 2

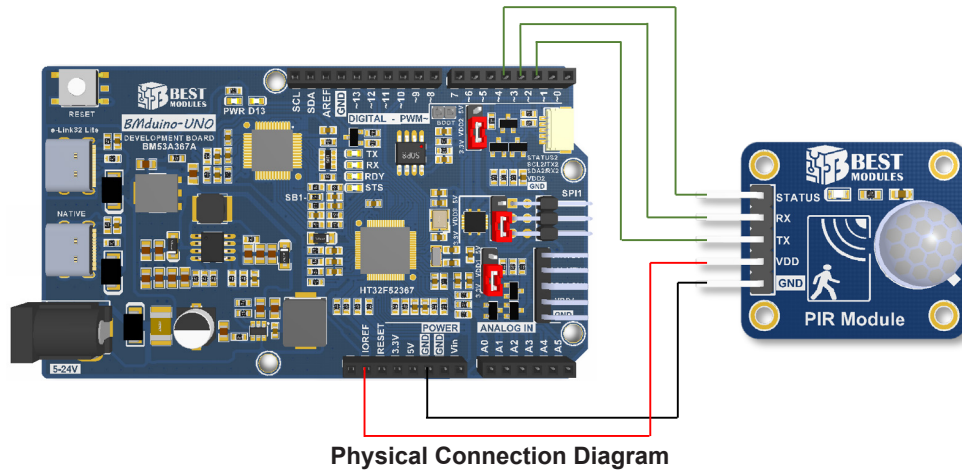
Method 2: Download before adding a ZIP library

Download method: Open the Best Modules official website (<https://www.bestmodulescorp.com/bm22s402x-1.html>) and download the BM22S402x-1 Library from “Arduino example program” under the “DOCUMENTS” menu.

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



Arduino Example



Example: LEDIndicatesTriggerState

Example function: Use LED to indicate the triggering status of the module. When the module detects an object, the module LED on, otherwise off.

1. Open the example: Arduino IDE→File→Examples→Select Lib (BM22S402x-1)→Select example (LEDIndicatesTriggerState)
2. Example description:
 - a. Create object & initialise object

```
#include <BM22S402x-1.h> // Call the BM22S402x-1 library
#define RX_PIN 2        // PIR_TX
#define TX_PIN 3        // PIR_RX
#define STATUS 13       // PIR_STATUS
/* Create object & Set Software serial pin */
BM22S402x_1 pir(RX_PIN, TX_PIN);
uint8_t error = 0;
void setup()
{
    /* Module initialisation */
    pir.begin();// Software serial initialisation(baud rate: 38400bps),
                // set Pin 4 to input mode
    pinMode(STATUS, OUTPUT);
    digitalWrite(STATUS, LOW);
    /* Configure serial monitor */
    Serial.begin(9600); // Serial initialisation, baud rate: 9600bps
    /* Configure the module parameter */
    error += pir.writeCommand(0x07, L1); // Sensitivity: Level 1 (most
                                        // sensitive)
    error += pir.writeCommand(0x05, 0x6B);// PIR register configuration
    if (error == 0)
    {
        Serial.println("Setting succeeded!");
    }
}
```

```

else
{
  Serial.println("Setting failed.");
}
while (pir.isStable() == false);
Serial.println("Module stabilized.");
}

```

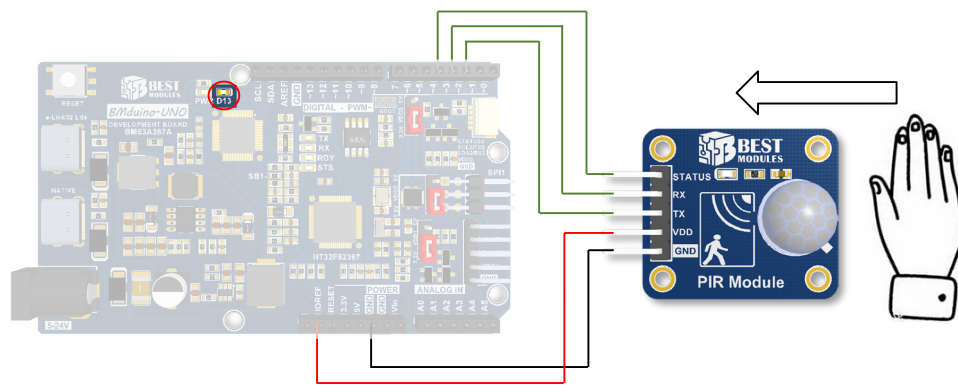
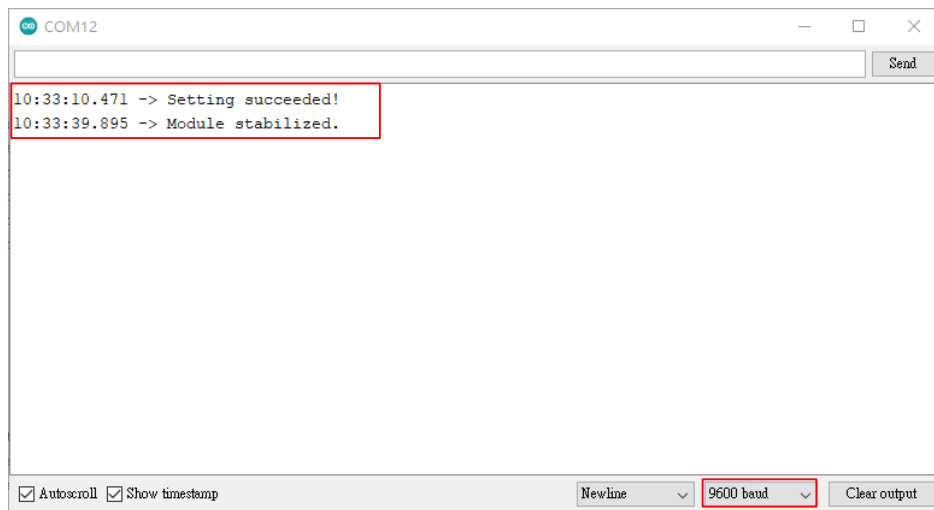
- b. When the module is triggered, the development board LED (D13) will be turned on. After a set delay time, the module will be released from the triggered status, the LED will be turned off.

```

void loop()
{
  if (pir.isTrigger())
  {
    digitalWrite(STATUS, HIGH); // Module is triggered
  }
  else
  {
    digitalWrite(STATUS, LOW); // Module is not triggered
  }
}

```

3. Open the serial monitor, after “Module stabilized” is displayed, observe the development board LED (D13). When a hand swings over the module, the LED will be turned on, after a delay time, the LED will be turned off.



Copyright© 2023 by BEST MODULES CORP. All Rights Reserved.

The information provided in this document has been produced with reasonable care and attention before publication, however, BEST MODULES does not guarantee that the information is completely accurate. The information contained in this publication is provided for reference only and may be superseded by updates. BEST MODULES disclaims any expressed, implied or statutory warranties, including but not limited to suitability for commercialization, satisfactory quality, specifications, characteristics, functions, fitness for a particular purpose, and non-infringement of any third-party's rights. BEST MODULES disclaims all liability arising from the information and its application. In addition, BEST MODULES does not recommend the use of BEST MODULES' products where there is a risk of personal hazard due to malfunction or other reasons. BEST MODULES hereby declares that it does not authorize the use of these products in life-saving, life-sustaining or safety critical components. Any use of BEST MODULES' products in life-saving/sustaining or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold BEST MODULES harmless from any damages, claims, suits, or expenses resulting from such use. The information provided in this document, including but not limited to the content, data, examples, materials, graphs, and trademarks, is the intellectual property of BEST MODULES (and its licensors, where applicable) and is protected by copyright law and other intellectual property laws. No license, express or implied, to any intellectual property right, is granted by BEST MODULES herein. BEST MODULES reserves the right to revise the information described in the document at any time without prior notice. For the latest information, please contact us.