PIR Detector Digital Module

BM22S4221-1
Arduino Library Description

Revision: V1.01 Date: September 07, 2023

www.bestmodulescorp.com

Contents

Introduction

The Best Modules BM22S4221-1 is a PIR detector digital module, which uses the UART communication method. This document provides the description of the BM22S4221-1 Arduino Lib functions and how to install the Arduino Lib. The example uses the BMA46M422 module to demonstrate the functions of the alarm detection and others.

Applicable Part No.:

Part No.	Description
BM22S4221-1	PIR Detector Digital Module
BMA46M422	On-board BM22S4221-1 Module

Arduino Lib Functions

Arduino Lib Name: BM22S4221-1		Lib Version: V1.0.1
Constructors & Initialisation		
1	BM22S4221_1(uint8_t statusPin, HardwareSerial* theSerial)	
	Description	Constructor, uses hardware UART
	Parameter	statusPin: STATUS pin, which is connected to the BM22S4221-1 STATUS pin or the BMA46M422 STA pin * theSerial: Hardware UART communication interface selection
	Return Value	—
	Note	—
2	BM22S4221_1(uint8_t statusPin, uint8_t rxPin, uint8_t txPin)	
	Description	Constructor, uses software UART
	Parameter	statusPin: STATUS, which is connected to the BM22S4221-1 STATUS pin or the BMA46M422 STA pin rxPin: RX pin, which is connected to the BM22S4221-1 or the BMA46M422 TX pin txPin: TX pin, which is connected to the BM22S4221-1 or the BMA46M422 RX pin
	Return Value	—
	Note	—
3	void begin()	
	Description	Module initialisation
	Parameter	—
	Return Value	—
	Note	The baud rate is fixed at 9600
Performance Functions		
4	uint8_t getSTATUS()	
	Description	Get the STATUS pin level
	Parameter	—
	Return Value	STATUS pin level: 0: Low level 1: High level
	Note	—

5	uint8_t requestInfoPackage(uint8_t buff[])	
	Description	Get all current device data
	Parameter	buff[]: Store the read data (25-byte)
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	For the data content, refer to the BM22S4221-1 datasheet
6	uint8_t getFWVer()	
	Description	Get the module software version
	Parameter	—
	Return Value	Module software version FWVer
	Note	—
7	uint8_t getProDate(uint8_t buff[])	
	Description	Get the module production date
	Parameter	buff[]: It is used to store production date, buff[0]/buff[1]/buff[2] : Year/Month/Day
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	Example: 0x21, 0x06, 0x2, represents 2021/6/2
8	bool isAotuTx()	
	Description	Get whether TX auto output is enabled or not
	Parameter	—
	Return Value	Judgment result: true: Yes false: Not
	Note	—
9	uint8_t getStatusPinActiveMode()	
	Description	Get the alarm output level
	Parameter	—
	Return Value	Alarm level: 0x80: High level 0x00: Low level
	Note	—
10	uint8_t getVBG()	
	Description	Gets the current internal V_{BG} voltage A/D value
	Parameter	—
	Return Value	V_{BG} voltage A/D value
	Note	—
11	bool isInfoAvailable()	
	Description	Determine whether the data is a 25-byte packet automatically output by the module
	Parameter	—
	Return Value	Judgment result: true: Yes false: Not
	Note	—
12	uint8_t readInfoPackage(uint8_t array[])	
	Description	Read a 25-byte data which is automatically output by the TX pin
	Parameter	array[]: Store the read data
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	The data is read passively. For the data content, refer to the BM22S4221-1 datasheet

13	uint8_t resetModule()	
	Description	Reset the module
	Parameter	—
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	—
14	uint8_t restoreDefault()	
	Description	Restore the module to factory settings
	Parameter	—
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	—
Configuration Parameter		
15	uint8_t setAutoTx(uint8_t state)	
	Description	Set whether the automatic output function is enabled or not
	Parameter	state: Whether the auto output function is enabled or not 0x08 (AUTO): Enabled 0x00 (PASSIVE): Disabled
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	—
16	uint8_t setStatusPinActiveMode(uint8_t state)	
	Description	Set the alarm output level
	Parameter	state: Alarm output level 0x08 (HIGH_LEVEL): High level 0x00 (LOW_LEVEL): Low level
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	—
17	uint8_t setOpaGain(uint8_t value)	
	Description	Set the internal OPA gain
	Parameter	value: The OPA gain parameter, the range is 0~31
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	OPA gain=128+value×8
18	uint8_t setAlarmThreshold(uint8_t Threshold)	
	Description	Set the PIR alarm threshold
	Parameter	Threshold: PIR alarm threshold, the range is 15~120
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	Threshold=Alarm Trigger Value - |PIR center voltage value|

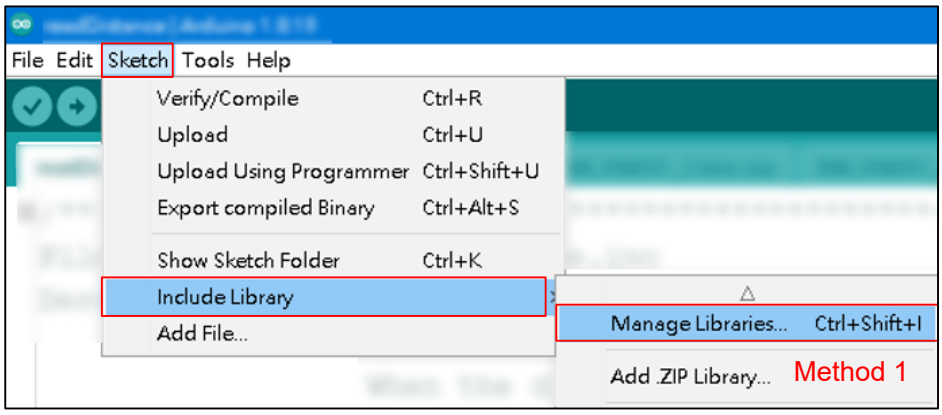
19	uint8_t setAlarmDetectDelay(uint8_t time)	
	Description	Set the delay time for detection again after the alarm is triggered
	Parameter	time: Delay time, in s
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	The module factory default setting is 3s
20	uint8_t setAlarmOutputTime(uint8_t time)	
	Description	Set the alarm signal STATUS pin duration output time
	Parameter	time: Duration output time, in s
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	The module factory default setting is 3s
21	uint8_t setPreheaTime(uint8_t time)	
	Description	Set the warm-up time
	Parameter	time: Warm-up time, the range is 30~127s
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	The module factory default setting is 30s

Arduino Lib Download and Installation

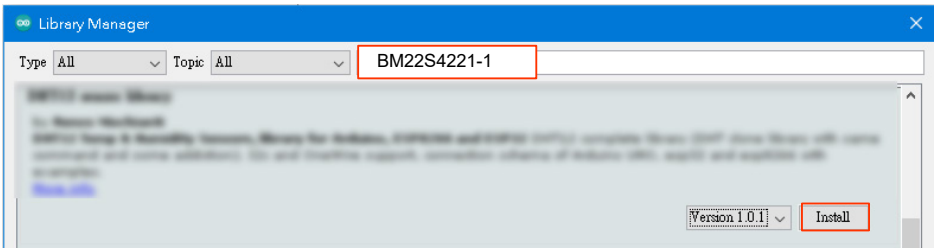
BM22S4221-1 Library: Refer to the following two methods to install the BM22S4221-1 Arduino library

Method 1: Search for installation

Arduino IDE→Sketch→Include Library→Manage Libraries→Search BM22S4221-1→Install



Search for Installation Step 1

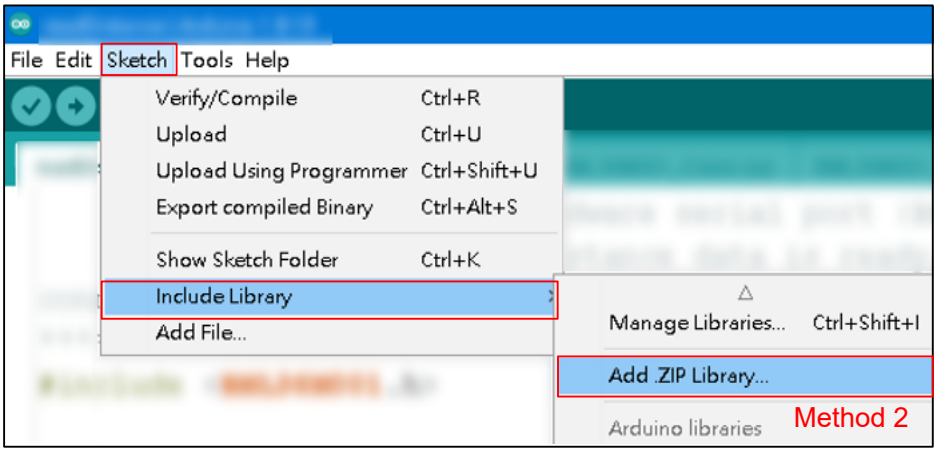


Search for Installation Step 2

Method 2: Download the .ZIP library before adding it

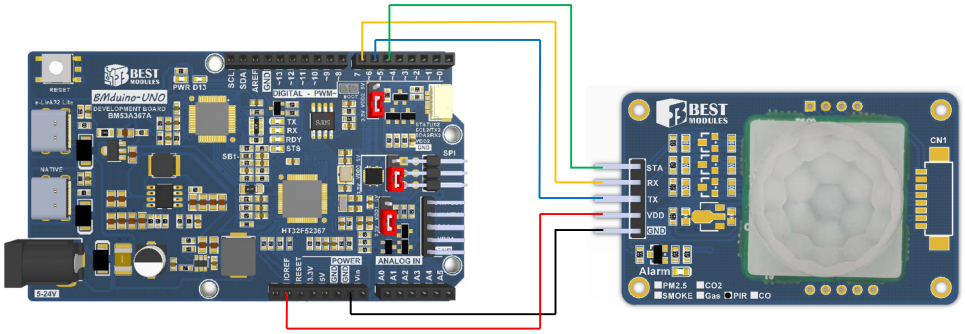
Download the Arduino example (BM22S4221-1 Library) under the DOCUMENTS menu from the Best Modules website (https://www.bestmodulescorp.com/bm22s4221-1.html)

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



Arduino Example

Example1: readStatusPin



Physical Connection Diagram

Example function: The LED13 flashes slowly when the module is in a normal status, the LED13 is always on when the module alarm is detected. The corresponding information of the module status change will be printed on the serial port monitor.

1. Open an example program: Arduino IDE→File→Examples→Select Lib (BM22S4221-1)→Select the program (readStatusPin)
2. Example program description:
   a. Create object.

```
#include "BM22S4221-1.h"
uint8_t STATUS=5;            // Chanhe the STATUS pin
uint8_t flag=3;
BM22S4221_1 PIR(STATUS,6,7); // Software Serial
```

   b. Initialise the module and configure I/O.

```
void setup()
{
  Serial.begin(9600);
  PIR.begin();
  pinMode(STATUS,INPUT);
  pinMode(13,OUTPUT);
}
```

c. The module status is detected cyclically, the LED flashes differently in the normal/alarm state.

```
if(PIR.getSTATUS()==HIGH&&flag!=1)
{
  Serial.println("Alarm! an object passes by");// Trigger alarm status
  flag=1;
}
if(flag!=0&&PIR.getSTATUS()==LOW)
{
  flag=0;
  Serial.println("Module normal;no alarm"); // The module is currently
                                            // in a normal state
}
switch(flag)
{
  case 0:    // Normal
  digitalWrite(13,1);
  delay(100);
  digitalWrite(13,LOW);
  delay(900);
  break;
  case 1:    // Alarm
  digitalWrite(13,1);
  delay(1000);
  break;
}
```

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