



**1D Infrared Gesture Control Digital Module**

**BM32S3021-1**  
**Arduino Library V1.0.4 Description**

Revision: V1.30 Date: May 28, 2025

[www.bestmodulescorp.com](http://www.bestmodulescorp.com)

## Contents

<b>Introduction</b> .....	<b>3</b>
<b>Arduino Lib Functions</b> .....	<b>3</b>
<b>Arduino Lib Download and Installation</b> .....	<b>6</b>
<b>Arduino Example</b> .....	<b>7</b>
Example: getGesture.....	7

## Introduction

The Best Modules BM32S3021-1 is a 1D infrared gesture control digital module, which uses the UART communication method. This document describes the Arduino Lib function of the BM32S3021-1 and how to install the Arduino Lib. The example demonstrates the function to recognize whether a gesture is swiping left or right with the BMS31M002/BMS31M002A module.

Applicable types:

Part No.	Description
BM32S3021-1	1D infrared gesture control digital module
BMS31M002	Integrated BM32S3021-1 module
BMS31M002A	Integrated BM32S3021-1 module

## Arduino Lib Functions

Arduino Lib Name: BM32S3021-1		Lib Version: V1.0.4
<b>Constructors &amp; Initialisation</b>		
1	BM32S3021_1(uint8_t intPin, HardwareSerial *theSerial=&Serial)	
	Description	Constructor, uses hardware serial interface
	Parameter	intPin: INT connected pin *theSerial: selects hardware serial interface (default serial interface)
	Return Value	—
	Note	—
2	BM32S3021_1(uint8_t intPin, uint8_t rxPin, uint8_t txPin)	
	Description	Constructor, uses software serial interface
	Parameter	intPin: INT pin, connect the BM32S3021-1 INT pin or the BMS31M002 INT pin rxPin: RX pin, connect the BM32S3021-1 D pin or the BMS31M002 TX pin txPin: TX pin, connect the BM32S3021-1 C pin or the BMS31M002 RX pin
	Return Value	—
	Note	—
3	void begin(uint16_t baud=9600)	
	Description	Module initialisation
	Parameter	baud: baud rate, fixed at 9600bps
	Return Value	void
	Note	—
<b>Performance Functions</b>		
4	uint8_t getINT()	
	Description	Obtain the INT pin level
	Parameter	—
	Return Value	INT pin level: 0x00: Low level, gesture detected 0x01: High level, gesture not detected
	Note	—

5	uint8_t getIRStatus()	
	Description	Obtain the gesture sensing status
	Parameter	—
	Return Value	The gesture sensing status: Bit 3: Calibration state 0: General mode 1: Calibration Bit 2: Left swipe state 0: Left swipe end 1: Left swipe ture Bit 1: Right swipe state 0: Right swipe end 1: Right swipe ture Bit 0: Proximity sensing state 0: No proximity 1: Proximity
	Note	—
6	uint8_t distanceLearning()	
	Description	Distance learning
	Parameter	—
	Return Value	Distance learning situation 0x00: Learning succeeded 0x01: Learning failed
	Note	—
7	uint8_t getIRGestureNum()	
	Description	Obtain the swipe times
	Parameter	—
	Return Value	Swipe times (-127~128) -n=Right swipe times n=Left swipe times
	Note	—
8	uint16_t getFWVer()	
	Description	Obtain the version number
	Parameter	—
	Return Value	Version number
	Note	For example, if the return value is 0x0100, the version number is V1.00
9	uint8_t reset()	
	Description	Module reset
	Parameter	—
	Return Value	Execution result: 0x00: Succeeded 0x01: Failed
	Note	—
<b>Parameter Configuration &amp; Acquisition</b>		
10	uint8_t getIRDebounce()	
	Description	Obtain the number of the IR trigger debounce times
	Parameter	—
	Return Value	Number of the IR trigger debounce times
	Note	—
11	uint8_t getIRThreshold()	
	Description	Obtain the proximity sensing trigger threshold
	Parameter	—
	Return Value	Proximity sensing trigger threshold
	Note	—

12	uint8_t getIRQTrigerTime()	
	Description	Obtain the IRQ trigger time when the gesture has setup
	Parameter	—
	Return Value	Return value: The IRQ trigger time parameter when the gesture has setup IRQ trigger time=(return value×4)ms
	Note	—
13	uint8_t getIRContinutyGestureTime()	
	Description	Obtain the cumulative continuous swipe time
	Parameter	—
	Return Value	Return value: Cumulative continuous swiping duration parameter Cumulative continuous swiping time=(return value×64)ms
	Note	—
14	uint8_t getIRFastestGestureTime()	
	Description	Obtain the fastest gesture detection time
	Parameter	—
	Return Value	Return value: The fastest gesture detection time parameter The fastest gesture detection time=(20+(return value×4))ms
	Note	—
15	uint8_t getIRSlowestGestureTime()	
	Description	Obtain the slowest gesture detection time
	Parameter	—
	Return Value	Return value: The slowest gesture detection time parameter The slowest gesture detection time=(return value×16)ms
	Note	—
16	uint8_t setIRDebounce(uint8_t debounce=7)	
	Description	Set the IR trigger debounce time
	Parameter	debounce: IR trigger debounce time, range: 0~255, default: 7
	Return Value	Execution result: 0x00: Succeeded 0x01: Failed
	Note	—
17	uint8_t setIRThreshold(uint8_t threshold=16)	
	Description	Set the gesture trigger threshold
	Parameter	threshold: gesture trigger threshold, range: 10~200, default: 16
	Return Value	Execution result: 0x00: Succeeded 0x01: Failed
	Note	—
18	uint8_t setIRQTrigerTime(uint8_t irqTime=50)	
	Description	Set the IRQ trigger time when the gesture has setup
	Parameter	irqTime: IRQ trigger time parameter, range: 0~255, default: 50 IRQ trigger time=(irqTime×4)ms
	Return Value	Execution result: 0x00: Succeeded 0x01: Failed
	Note	—
19	uint8_t setIRContinutyGestureTime(uint8_t irTime=30)	
	Description	Set the cumulative continuous swipe time
	Parameter	irqTime: cumulative continuous swipe time parameter, range: 0~255, default: 30 Cumulative continuous swipe time=(irqTime×64)ms
	Return Value	Execution result: 0x00: Succeeded 0x01: Failed
	Note	—

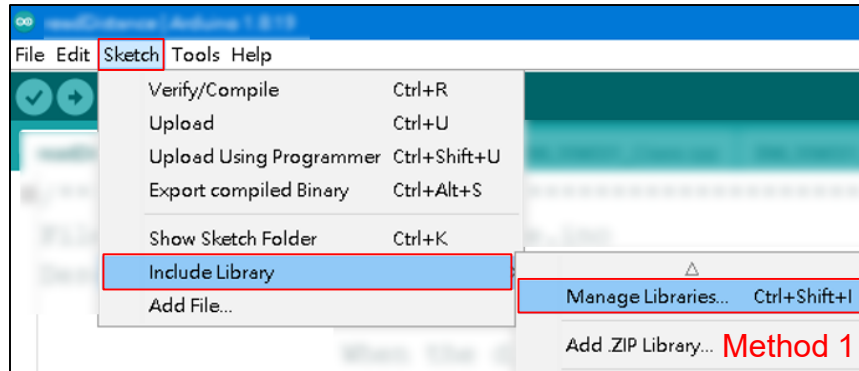
20	uint8_t setIRFastestGestureTime(uint8_t irTime=0)	
	Description	Set the fastest gesture detection time
	Parameter	irqTime: The fastest gesture detection time parameter, range: 0~200 default: 0 The fastest gesture detection time=(20+(irqTime×4))ms
	Return Value	Execution result: 0x00: Succeeded 0x01: Failed
	Note	—
21	uint8_t setIRSlowestGestureTime(uint8_t irTime=80)	
	Description	Set the slowest gesture detection time
	Parameter	irqTime: The slowest gesture detection time parameter, range: 0~200 default: 80 The slowest gesture detection time=(irqTime×16)ms
	Return Value	Execution result: 0x00: Succeeded 0x01: Failed
	Note	—

## Arduino Lib Download and Installation

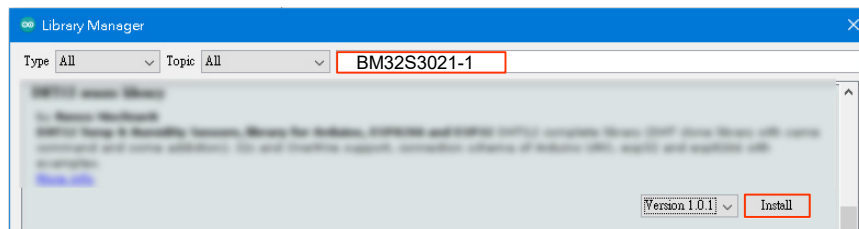
BM32S3021-1Library: Refer to the following two methods to install the BM32S3021-1 Arduino Library.

### Method 1: Search for installation

Arduino IDE → Sketch → Include Library → Manage Libraries... → Search BM32S3021-1 → Install



**Search for Installation Step 1**

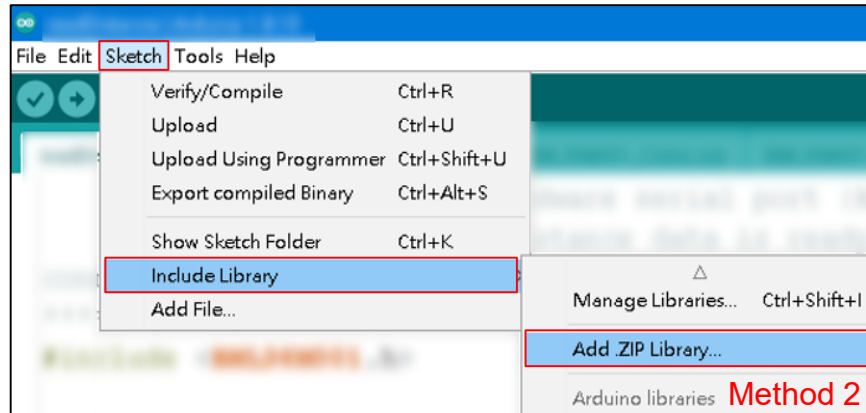


**Search for Installation Step 2**

## Method 2: Download the .ZIP library before adding it

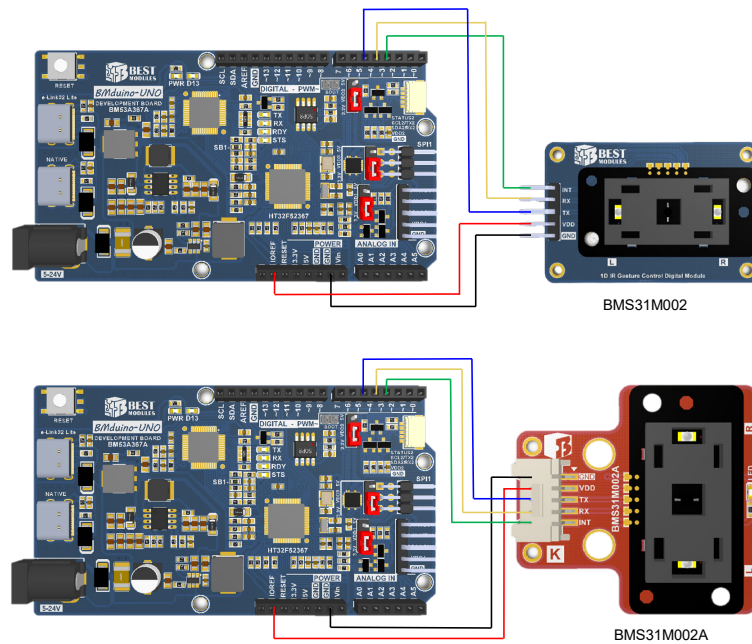
Download the Arduino example (BM32S3021-1 Library) under the DOCUMENTS menu from the Best Modules website (<https://www.bestmodulescorp.com/bm32s3021-1.html>).

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



## Arduino Example

### Example: getGesture



**Physical Connection Diagram**

Function: When the module has detected an object, the development board will obtain the gesture sensing state of the module to determine whether the object is swiping left or right, and display it in the serial monitor.

1. Open the example: File → Examples → Select Lib (BM32S3021-1) → Select example (getGesture)

2. Example Description:

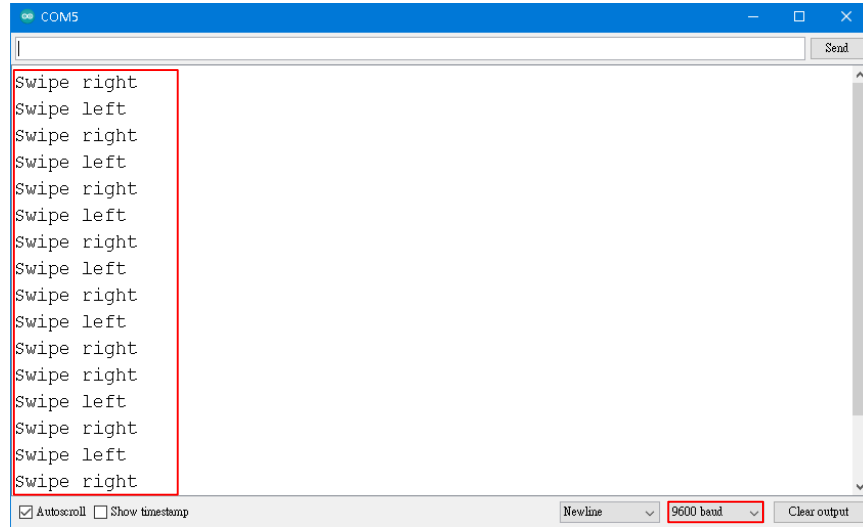
a. Create object & module initialization and configuration

```
#include "BM32S3021-1.h"
BM32S3021_1 myGesture(3,5,4); // Create object, the INT pin connects to
                               // the development board D3, the TX pin connects to
                               // the development board D5, the RX pin connects to
                               // the development board D4
uint8_t irStatus = 0;
void setup()
{
  myGesture.begin();           // Module initialisation
  Serial.begin(9600);          // Configure the serial monitor
}
```

b. When the module is triggered, the development board will detect the gesture of the module and display it in the serial monitor.

```
void loop()
{
  if(!myGesture.getINT ()) // Obtain the INT line status to determine
                           // whether the module is triggered
  {
    /** Detecte the gesture and display it in the serial monitor**/
    irStatus=myGesture.getIRStatus();// Obtain the gesture sensing
status
    if(!(irStatus&0x08))
    {
      if(irStatus&0x02) // Determine whether to swipe right
      {
        Serial.println("Swipe right");// A "Swipe right" message will
        // be displayed in the serial monitor
      }
      else if(irStatus&0x04) // Determine whether to swipe left
      {
        Serial.println("Swipe left");// A "Swipe left" message will
        // be displayed in //the serial monitor
      }
    }
  }
}
```

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



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.