



Medium Power Atomiser Module

BMN31K522

Arduino Library Description

Revision: V1.00 Date: May 31, 2023

www.bestmodulescorp.com

Contents

Introduction	3
Arduino Lib Functions	3
Arduino Lib Download and Installation	4
Arduino Example	5
Example: controlAtomization.....	5

Introduction

The BMN31K522 is a medium power atomiser module from Best Modules, which uses the UART communication method. This document provides the description of the BMN31K522 Arduino Lib functions and how to install the Arduino Lib. The example demonstrates the function of atomisation and atomisation rank switching.

Arduino Lib Functions

Arduino Lib Name: BMN31K522		Lib Version: V1.0.1
Constructors & Initialisation		
1	BMN31K522(HardwareSerial *theSerial=&Serial)	
	Description	Constructor, uses hardware Serial interface
	Parameter	*theSerial: Select hardware Serial interface (defaults to Serial interface)
	Return Value	—
	Note	—
2	BMN31K522(uint8_t rxPin, uint8_t txPin)	
	Description	Constructor, uses software Serial interface
	Parameter	rxPin: Pin connects to TX txPin: Pin connects to RX
	Return Value	—
	Note	—
3	void begin()	
	Description	Module initialisation
	Parameter	—
	Return Value	void
	Note	Baud rate defaults to 9600
Performance Functions		
4	uint8_t setAtomizationRank(uint8_t rank)	
	Description	Set atomisation rank
	Parameter	rank: Atomisation rank, ranging from 0~255 (PWM pulse width)
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	—
5	uint8_t setAtomizationTime(uint8_t times)	
	Description	Set atomisation timing time
	Parameter	times: Timing time, ranging from 0~255, unit: min
	Return Value	Execution result: 0: Succeeded 1: Failed
	Note	—
6	uint8_t getAtomizationRank()	
	Description	Get atomisation rank value
	Parameter	—
	Return Value	Atomisation rank: 0~255
	Note	—

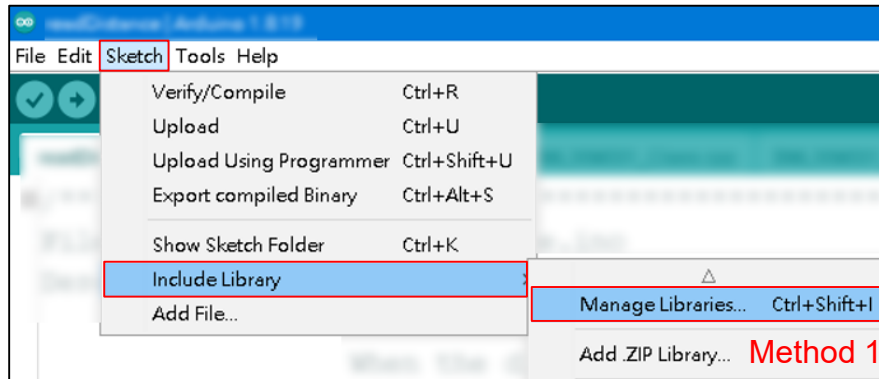
7	uint8_t getAtomizationTime()	
	Description	Get atomisation timing remaining time
	Parameter	—
	Return Value	Timing remaining time(Unit: min)
	Note	—
8	uint8_t getWaterStatus()	
	Description	Get water level status
	Parameter	—
	Return Value	Water level status: 0: Water shortage 1: Water existing
	Note	—

Arduino Lib Download and Installation

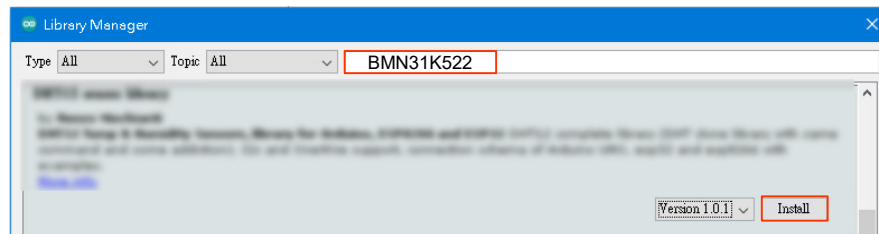
BMN31K522 Library: Refer to the following two methods to install the BMN31K522 Arduino Library

Method 1: Search for installation

Search for installation: Arduino IDE → Sketch → Include Library → Manage Libraries... → Search BMN31K522 → Install



Search for Installation Step 1

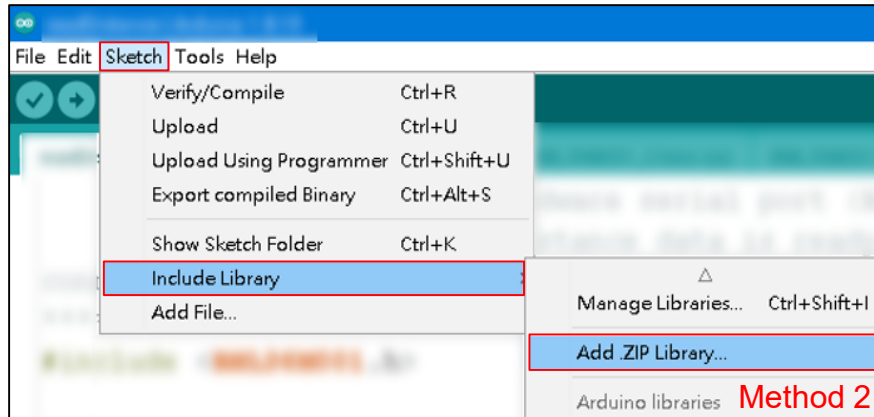


Search for Installation Step 2

Method 2: Download before adding a ZIP library

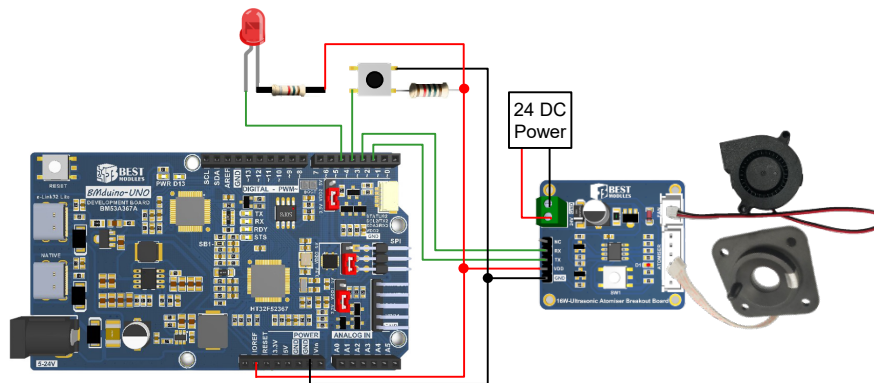
Download method: Open the Best Modules official website (<https://www.bestmodulescorp.com/bmn31k522.html#tab-product2>) and download the BMN31K522 Library from “Arduino example program” under the “DOCUMENTS” menu.

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



Arduino Example

Example: controlAtomization



Physical Connection Diagram

Example function: When the water existing condition is detected, use a button as a switch to control the atomiser module timing operation, the atomisation rank and timing time is fixed. LED ON represents atomisation start, LED OFF represents atomisation stop.

1. Open the example: Arduino IDE → File → Examples → Select Lib (BMN31K522) → Select example (controlAtomization)
2. Example description:
 - a. Create object & initialise object

```
#include <BMN31K522.h>
#define SERIAL_DEBUG (1) // 0-Disable serial debug output
                          // 1-Enable serial debug output

#define ATOMIZER_RANK (255)
#define ATOMIZER_TIME (1)
BMN31K522 myAtomizer(2, 3); // Creat objects
byte atomizerLED = 5; // Atomisation operation LED
byte atomizerSwitch = 4; // Atomisation switch PIN
unsigned long TimeCount_1s = 0; // Count 1s
```

```
byte waterStatus = 0;           // Water shortage status:
                                // 0-Water shortage; 1-Water existing
unsigned long switchOnTime = 0; // Count switching time
bool atomizerOnFlag = false;    // Atomisation operating flag
void setup()
{
  if (SERIAL_DEBUG){
    Serial.begin(9600);
  } myAtomizer.begin();        // Module initialisation
  pinMode(atomizerLED,OUTPUT);
  pinMode(atomizerSwitch,INPUT);
  digitalWrite(atomizerLED,HIGH);
  TimeCount_1s = millis();     // Start timing
}
```

- b. Get the water data every 1s, and display the water data in the serial monitor. Scan to check if the button is pressed to determine whether atomisation is start/stop. When the atomisation starts, get atomisation timing remaining time every 1s; when the time is 0, timing ends and the atomisation stops. LED indicates atomisation operating status, atomisation start-ON, atomisation stop-OFF.

```
void loop() {
  byte switchStatus;
  uint8_t getTime;
  /***** Get water status & remaining time every 1s *****/
  if (millis() - TimeCount_1s >= 1000)
  {
    TimeCount_1s = millis();
    /***** Get water status *****/
    waterStatus=myAtomizer.getWaterStatus();
    if (SERIAL_DEBUG)
    {
      if (waterStatus)
      {
        Serial.println("Atomizer's water status is: Have water");
        // Display the water detection status in the serial monitor
      }
      else
      {
        Serial.println("Atomizer's water status is: No water");
        // Display the water detection status in the serial monitor
      }
    }
  }
  /***** Get remaining time *****/
  if (atomizerOnFlag)
  {
    getTime=myAtomizer.getAtomizationTime();
    if (getTime == 0)
    {
      atomizerOnFlag = false;
      digitalWrite(atomizerLED,HIGH); // LED indicates the
                                      // atomisation opetating status
    }
  }
}
```

```
    if (SERIAL_DEBUG)
    {
        Serial.print("Atomizer work remaining time is:");
        // Display the atomisation timing remaining time
        // in the serial monitor
        Serial.println(getTime);
    }
}
}
/***** Key scanning*****/
if (waterStatus == 1)
{
    switchStatus = digitalRead(atomizerSwitch);
    if (!switchStatus)
    {
        if ((millis() - switchOnTime) > 150) // Switch debounce
        {
            switchOnTime = millis();
            if (!atomizerOnFlag)
            {
                atomizerOnFlag = true;
                digitalWrite(atomizerLED,LOW);
                myAtomizer.setAtomizationRank(ATOMIZER_RANK);
                // Set atomiser rank
                myAtomizer.setAtomizationTime(ATOMIZER_TIME);
                // Set atomiser timing
            }
        }
        else
        {
            atomizerOnFlag = false;
            digitalWrite(atomizerLED,HIGH); // LED indicates the
            // atomisation opetating status
            myAtomizer.setAtomizationRank(0); // Close atomiser work
            myAtomizer.setAtomizationTime(0);
        }
    }
}
else
{
    switchOnTime = millis();
}
/***** Processing for water shortage *****/
else
{
    if (atomizerOnFlag)
    {
        atomizerOnFlag = false;
        digitalWrite(atomizerLED,HIGH);
    }
}
}
```

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 authorise 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.