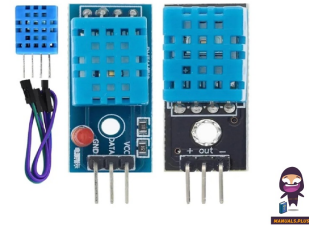


ARDUINO DHT11 Starter Kit



ARDUINO DHT11 Starter Kit User Guide

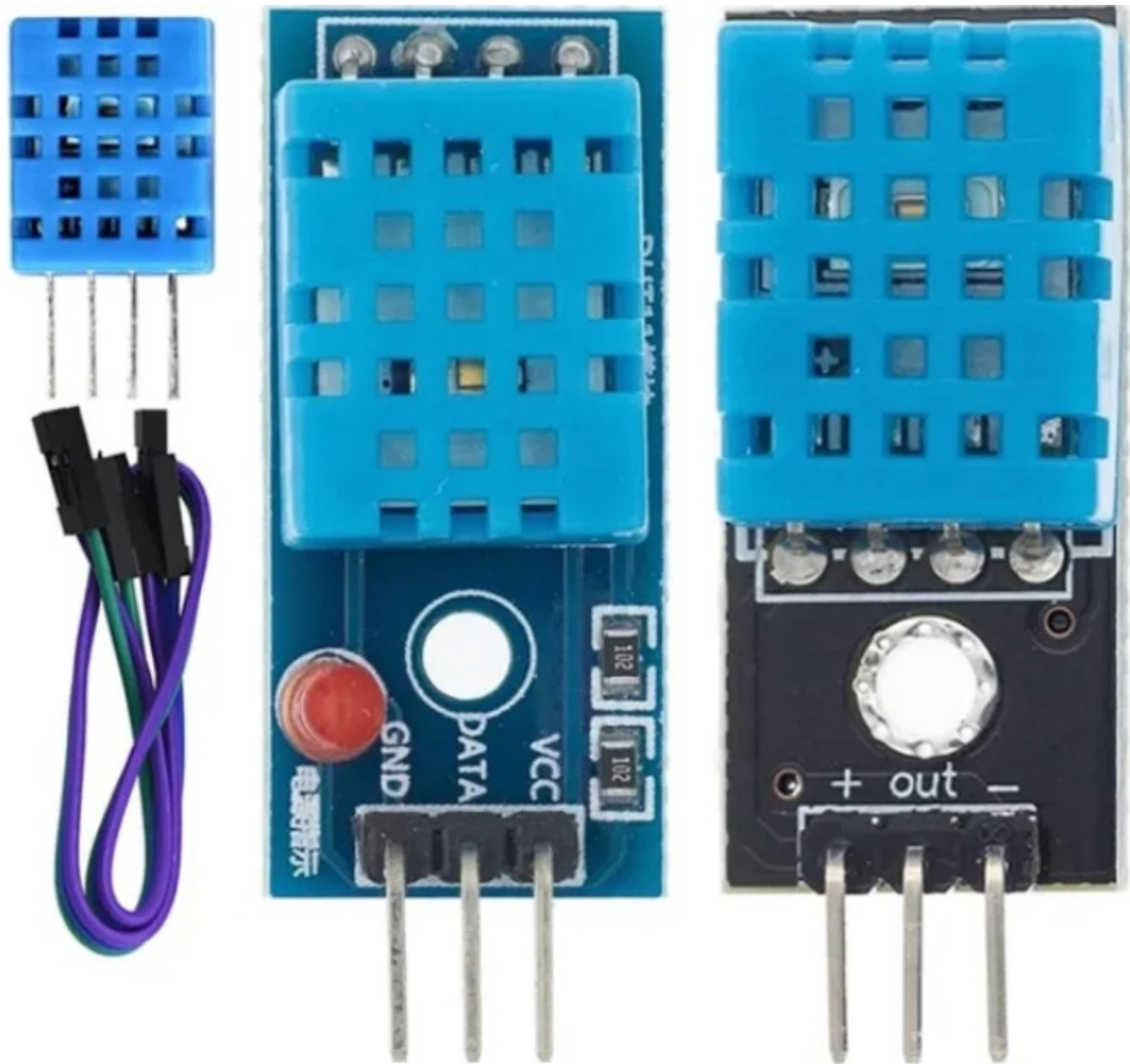
[Home](#) » [ARDUINO](#) » ARDUINO DHT11 Starter Kit User Guide 

Contents

- [1 ARDUINO DHT11 Starter Kit](#)
- [2 Specifications](#)
- [3 Storage LED And Screen Program](#)
- [4 Passive Buzzer Program](#)
- [5 Temperature and Humidity Sensor Program](#)
- [6 Infrared Remote Reception Program](#)
- [7 Photoresistor And Button Program](#)
- [8 FAQ](#)
- [9 Documents / Resources](#)
 - [9.1 References](#)

ARDUINO

ARDUINO DHT11 Starter Kit



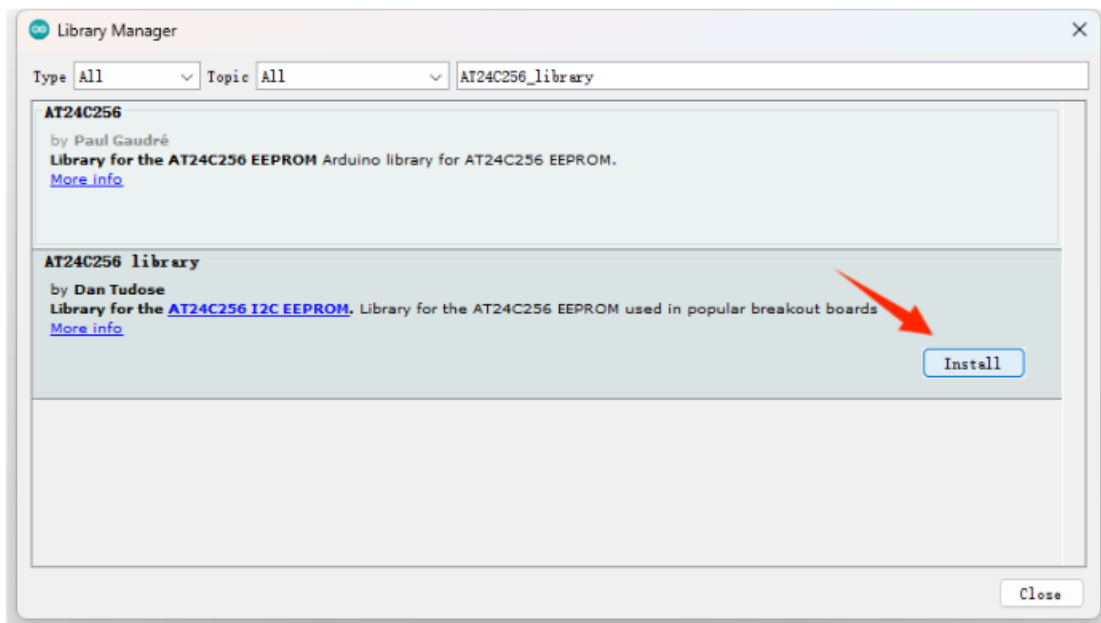
Specifications

- **Lesson 1:** EEPROM Storage Program
- **Lesson 2:** 0.96in LED Screen Program
- **Lesson 3:** MPU6050 Six-Axis Gyroscope Program
- **Lesson 4:** Passive Buzzer Program
- **Lesson 5:** DH11 Temperature and Humidity Sensor Program
- **Lesson 6:** Infrared Remote Reception Program
- **Lesson 7:** Photoresistor Program

Storage LED And Screen Program

Lesson 1 EEPROM Storage Program:

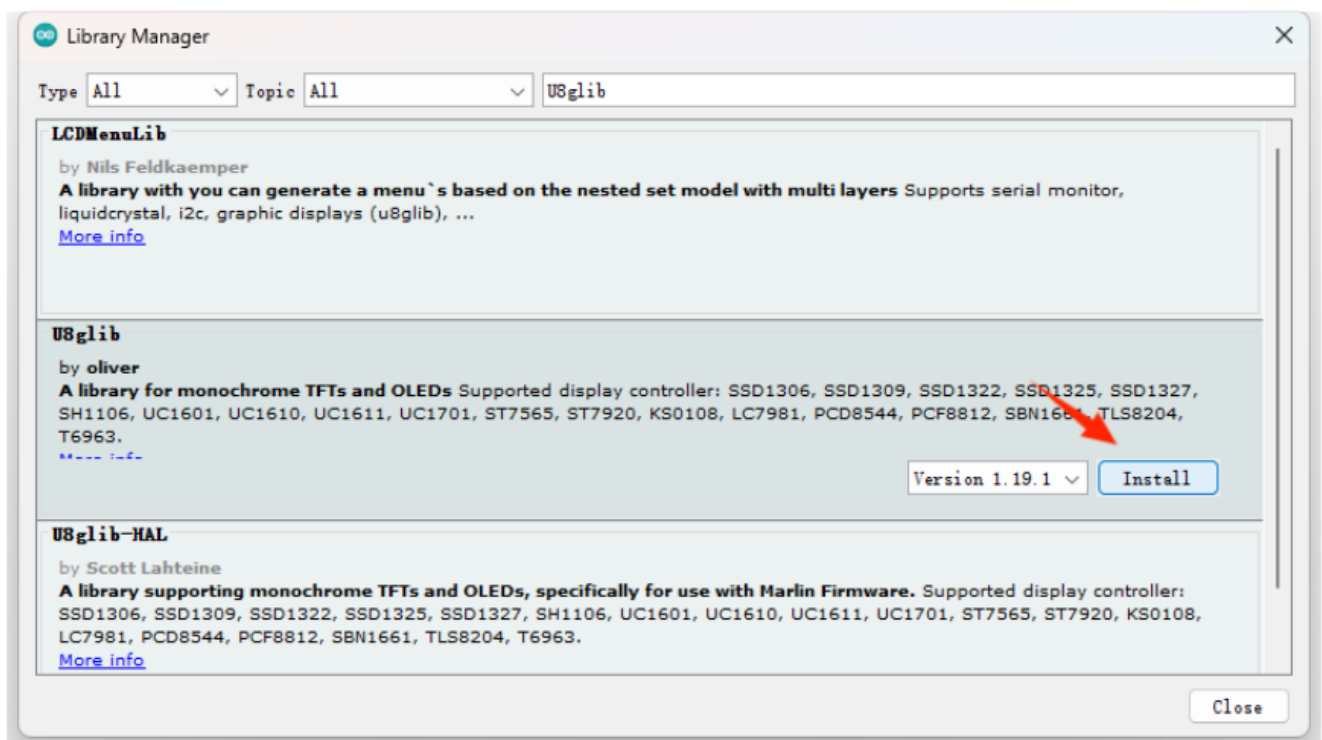
- Click Sketch in the Arduino IDE, select Manage Library in Include Library, search AT24C256_library, and click Install.



- Click File in the Arduino IDE, and select read_wirte in AT24C256_library from Examples.
- Click Upload, and click Serial Monitor in the upper right corner of the IDE.

Lesson 2: 0.96in LED Screen Program:

1. Click Sketch in the Arduino IDE, select Manage Library in Include Library, search U8glib, select U8glib and click Install



2. Click File in the Arduino IDE and select FPS from U8glib in Examples.
 - Find `// U8GLIB_SSD1306_128X64 u8g (U8G_I2C_OPT_NONE | U8G_I2C_OPT_DEV_0); // I2C/TWI` code, delete `///
//` uncomment, click Upload in the upper left corner.

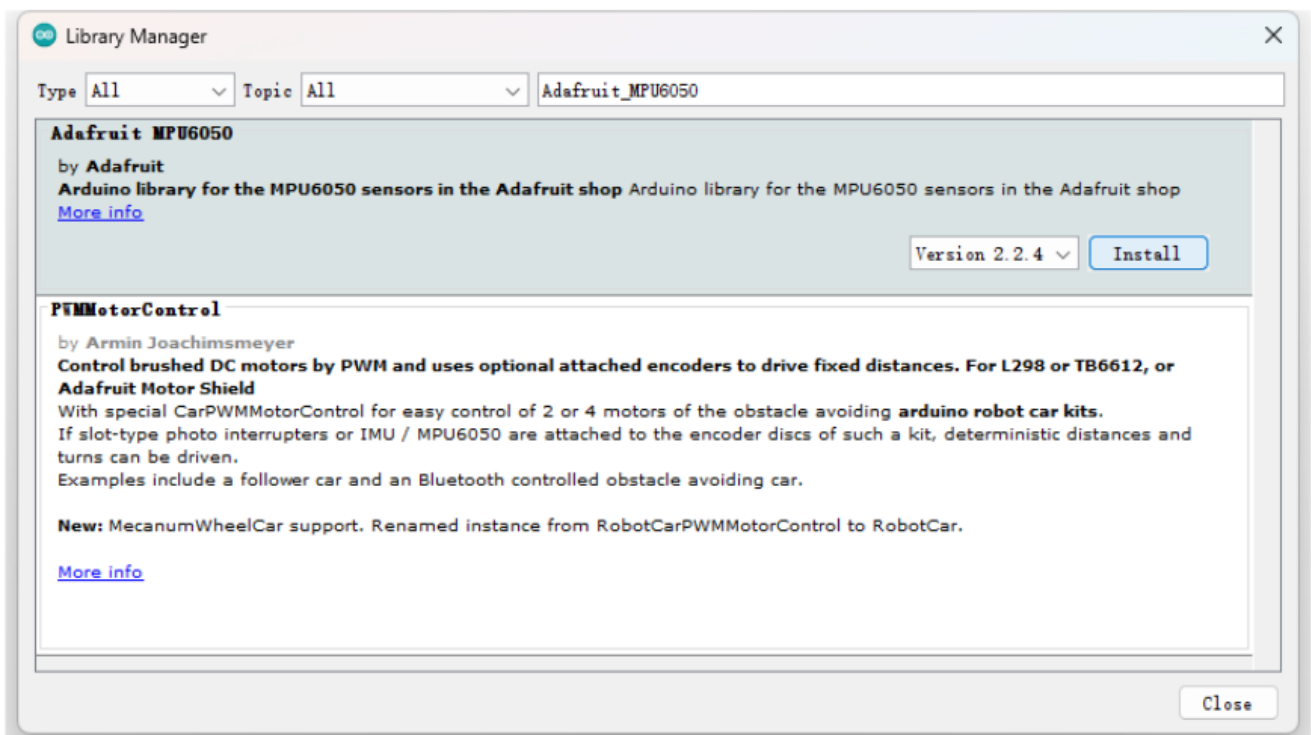
```

163 //U8GLIB_LC7981_240X64 u8g(8, 9, 10, 11, 4, 5, 6, 7, 18, 14, 1
164 //U8GLIB_LC7981_240X128 u8g(8, 9, 10, 11, 4, 5, 6, 7, 18, 14,
165 //U8GLIB_ILI9325D_320x240 u8g(18,17,19,U8G_PIN_NONE,16 );
166 //U8GLIB_SBN1661_122X32 u8g(8,9,10,11,4,5,6,7,14,15, 17, U8G_P
167 //U8GLIB_SSD1306_128X64 u8g(13, 11, 10, 9); // SW SPI Com: SCK
168 //U8GLIB_SSD1306_128X64 u8g(4, 5, 6, 7); // SW SPI Com: SCK =
169 //U8GLIB_SSD1306_128X64 u8g(10, 9); // HW SPI Com: CS = 10, 2
170 //U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE|U8G_I2C_OPT_DEV_0);
171 //U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_DEV_0|U8G_I2C_OPT_NO_A
172 //U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NO_ACK); // Display wh
173 //U8GLIB_SSD1306_ADAFRUIT_128X64 u8g(13, 11, 10, 9); // SW SP
174 //U8GLIB_SSD1306_ADAFRUIT_128X64 u8g(10, 9); // HW SPI Com:

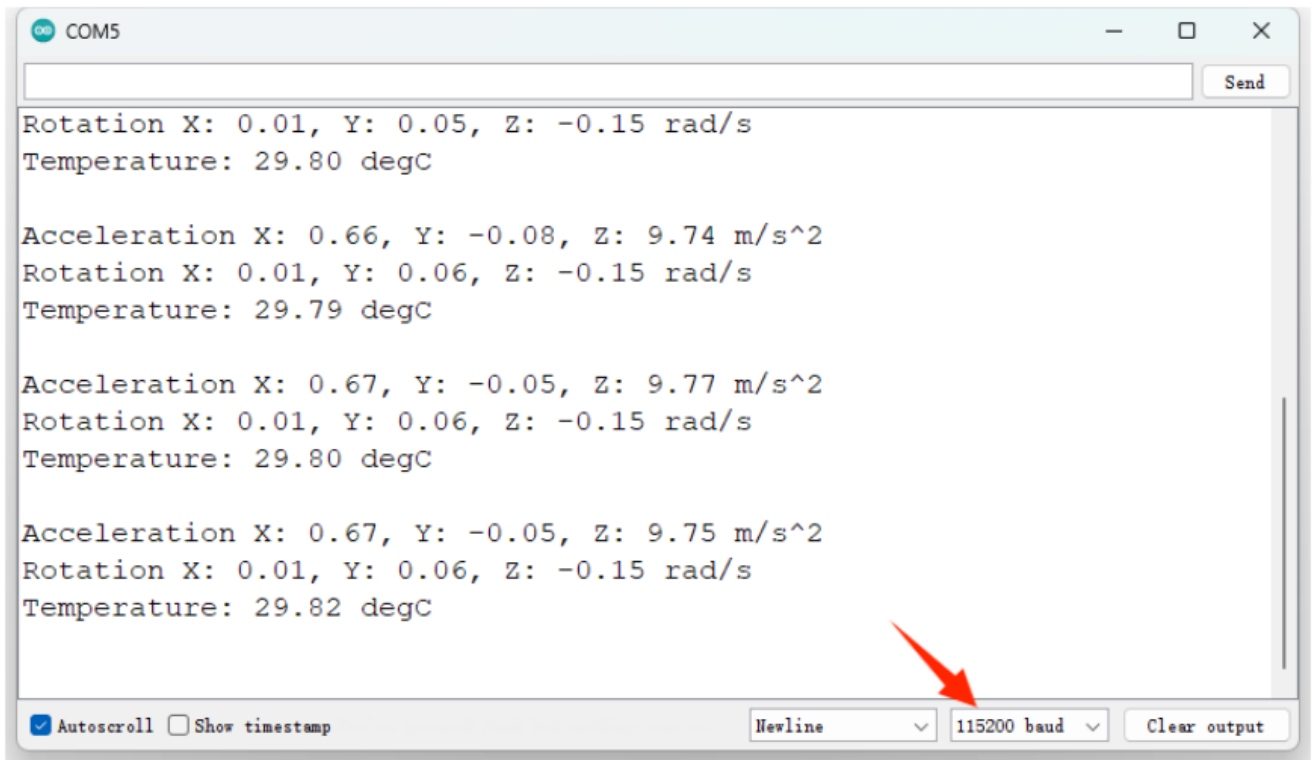
```

Lesson 3: MPU6050 Six-Axis Gyroscope Program:

1. Click Sketch in the Arduino IDE, select Manage Library in Include Library, search for Adafruit_MPU6050, and click Install.



2. Click File in the Arduino IDE and select basic_readings in Adafruit_MPU6050 in Examples.
3. Click Upload, click Serial Monitor in the upper right corner of IDE, and switch from 9600baud to 115200baud.



4. Because the initial values of all axes of MPU-6050 cannot be consistent, when Acceleration's X and Y axes are not equal to 0 m/s² and Z axes are not equal to 9.8 m/s², and the X, Y and Z of Rotation are not equal to 0 rad/s, you can increase or reduce the error values through the program. Make the initial value of the output relatively correct.

Passive Buzzer Program

Lesson 4: Passive Buzzer Program:

```

#define Pot A3
#define Buzzer 8
int PotBuffer = 0;
void setup()
{
  pinMode(Buzzer,OUTPUT); // The buzzer pin is set to output
}
void loop()
{
  PotBuffer = analogRead(Pot);    // Reading the AD value
  for(int i = 0 ; i < 50 ; i++)    // Cycle 50 times
  {
    digitalWrite(Buzzer,HIGH);    // Set the output high level
    delayMicroseconds(PotBuffer); // The delay PotBuffer value is us
    digitalWrite(Buzzer,LOW);     // Set the output low level
    delayMicroseconds(100);       // Delay 100us
  }
  delay(1000);                    // Delay 1000ms
}

```

Temperature and Humidity Sensor Program

Lesson 5: DHT11 Temperature and Humidity Sensor Program:

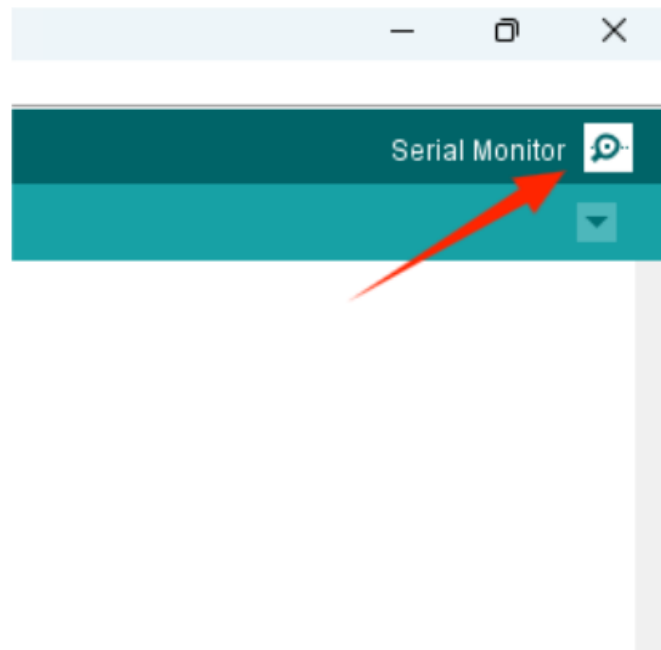
1. Click Sketch in the Arduino IDE, select Manage Library in Include Library, search for DHT11, select DFRobot_DHT11, and click Install.



2. Click File in the Arduino IDE, and select readDHT11 in DFRobot_DHT11 in Examples.
3. Change #define DHT11_PIN 10 to #define DHT11_PIN 3 and click IDE home page Upload.

```
1  /*!  
2  * @file readDHT11.ino  
3  * @brief DHT11 is used to read the temperature and humidity.  
4  *  
5  * @copyright Copyright (c) 2010 DF Robot  
6  * @license The MIT License (MIT)  
7  * @author [Wuxiao] (xiao.wu@dfrobot.com)  
8  * @version V1.0  
9  * @date 2018-09-14  
10 * @url https://github.com/DFRobot/DFRobot-DHT11  
11 */  
12  
13 #include <DFRobot_DHT11.h>  
14 DFRobot_DHT11 DHT;  
15 #define DHT11_PIN 3
```

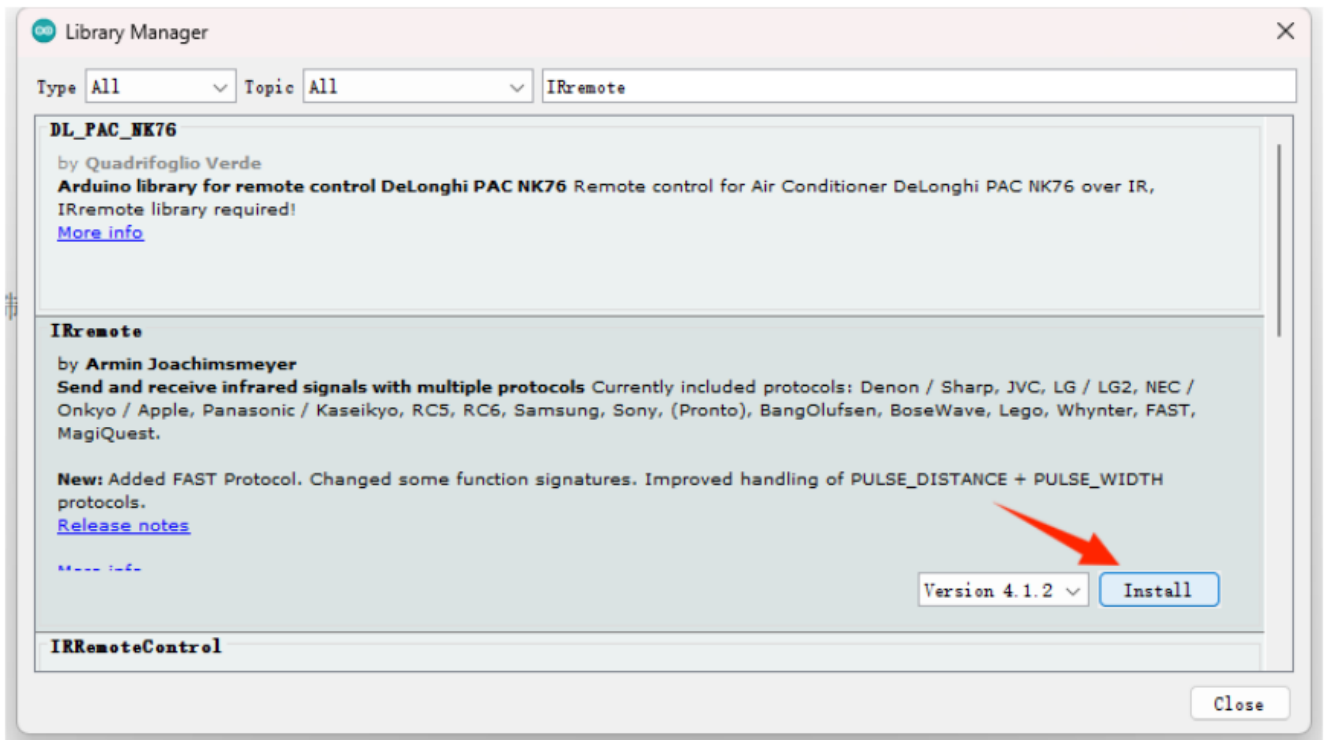
4. Click Serial Monitor in the upper right corner of the IDE and switch 9600baud to 115200baud. Wait about 1S to get the current temperature&humidity.



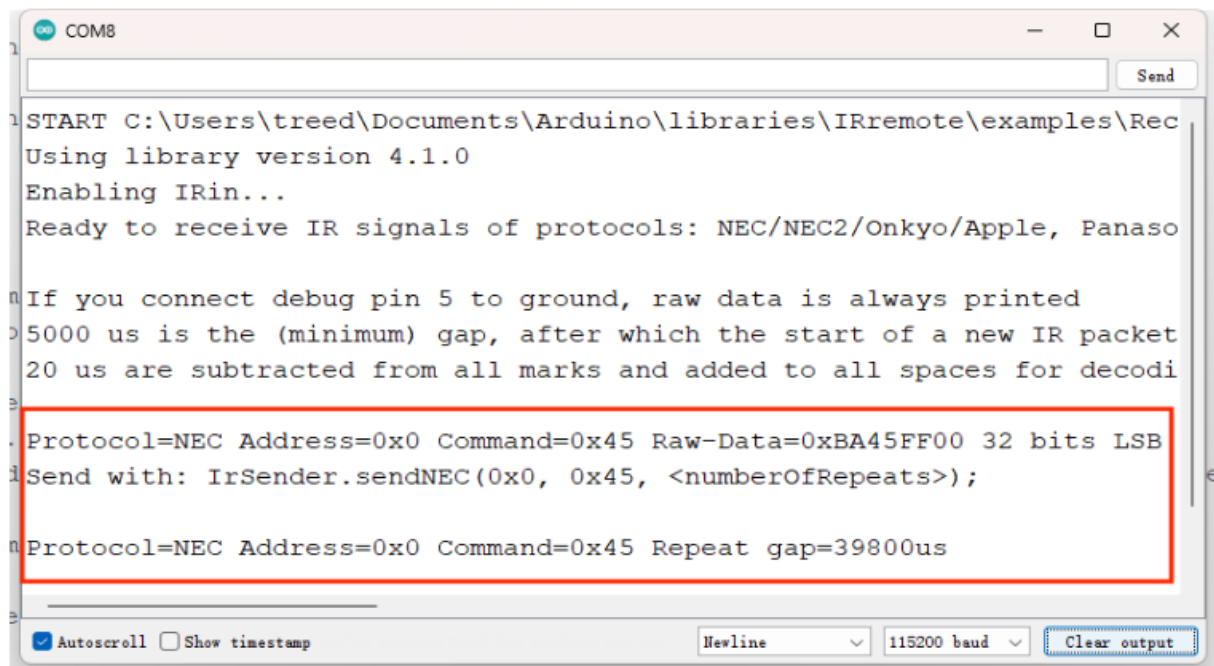
Infrared Remote Reception Program

Lesson6: Infrared Remote Reception Program

1. Click Sketch in the Arduino IDE, select Manage Library in Include Library, search for IRremote, and click Install



2. Click File in the Arduino IDE and select ReceiveDemo from IRremote in Examples.
3. Click Upload, click Serial Monitor in the upper right corner of IDE, and switch from 9600baud to 115200baud. Use the matching remote control to align the infrared receiving module and press any key. When corresponding data appears, the module will run normally.



Photoresistor And Button Program

Lesson7: Photoresistor Program:


```

#define ADpin A3
#define LED 13
int ADBuffer = 0;
void setup()
{
  pinMode(LED,OUTPUT);
  Serial.begin(9600);          // The baud rate is 9600
}
void loop()
{
  ADBuffer = analogRead(ADpin);  // Reading the AD value
  Serial.print("AD = ");
  Serial.println(ADBuffer);
  if(ADBuffer > 800)              // If the ADBuffer value is larger than the set
value, the illumination intensity is smaller than the set value
  {
digitalWrite(LED,HIGH);  // Light up LED
  }
  else
  {
    digitalWrite(LED,LOW);  // Turn off LED
  }
  delay(500);                // Delay 500ms
}

```

Lesson8: Button Program:

```

#define KEY0 digitalRead(4)
#define KEY1 digitalRead(5)
#define KEY0_PRES 1
#define KEY1_PRES 2

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  int key;
  key=KEY_Scan(0);// Scan key
  if (key==1)
  {  digitalWrite(LED_BUILTIN, HIGH); }
  if (key==2)
  {  digitalWrite(LED_BUILTIN, LOW);  }
}

u8 KEY_Scan(u8 mode)
{
  pinMode(4,INPUT_PULLUP);
  pinMode(5,INPUT_PULLUP);
  static u8 key_up=1;// Press the button to release the sign
  if(mode)key_up=1;  // Support link
  if(key_up&&(KEY0==0||KEY1==0))
  {
    delay(10);// Jitter elimination
    key_up=0;
    if(KEY0==0)return KEY0_PRES;
    else if(KEY1==0)return KEY1_PRES;
  }else if(KEY0==1&&KEY1==1)key_up=1;
  return 0;// No key to press }

```

FAQ

Frequently Asked Questions

- **Q: How do I troubleshoot if my program is not working?**
 - A: Check the connections to ensure they are properly set up. Verify that the libraries are correctly installed in the Arduino IDE. Make sure the code is error-free and matches the instructions provided in the manual.

Documents / Resources

