

# Arduino with BH1750 Ambient Light Sensor

The BH1750 is a 16-bit ambient light sensor. In this guide, you'll learn how to use the BH1750 ambient light sensor with the Arduino board. The sensor communicates with a microcontroller using I2C communication protocol.



You'll learn how to wire the sensor to the Arduino board, install the required libraries and use a simple sketch to display the sensor readings in the Serial Monitor.

This tutorial covers the following topics:

- [Introducing the BH1750 Ambient Light Sensor](#)
- [BH1750 Pinout](#)
- [BH1750 I2C Interface](#)
- [Example: BH1750: Read Ambient Light with Arduino](#)



# Introducing BH1750 Ambient Light Sensor

The BH1750 is a 16-bit ambient light sensor that communicates via I2C protocol. It outputs luminosity measurements in lux (SI-derived unit of illuminance). It can measure a minimum of 1 lux and a maximum of 65535 lux.

The sensor may come in different breakout board formats. See pictures below. Both images represent a BH1750 sensor.



## BH1750 Features

Here's a list of the BH1750 sensor features. For more information consult the [BH1750 sensor datasheet](#).

- I2C bus Interface
- Spectral responsibility is approximately human eye response
- Illuminance to digital converter
- Range: 1 – 65535 lux
- Low current by power down function
- 50Hz / 60Hz Light noise reject-function
- It is possible to select 2 different I2 C slave-addresses
- Small measurement variation (+/- 20%)
- The influence of infrared is very small
- Supports continuous measurement mode



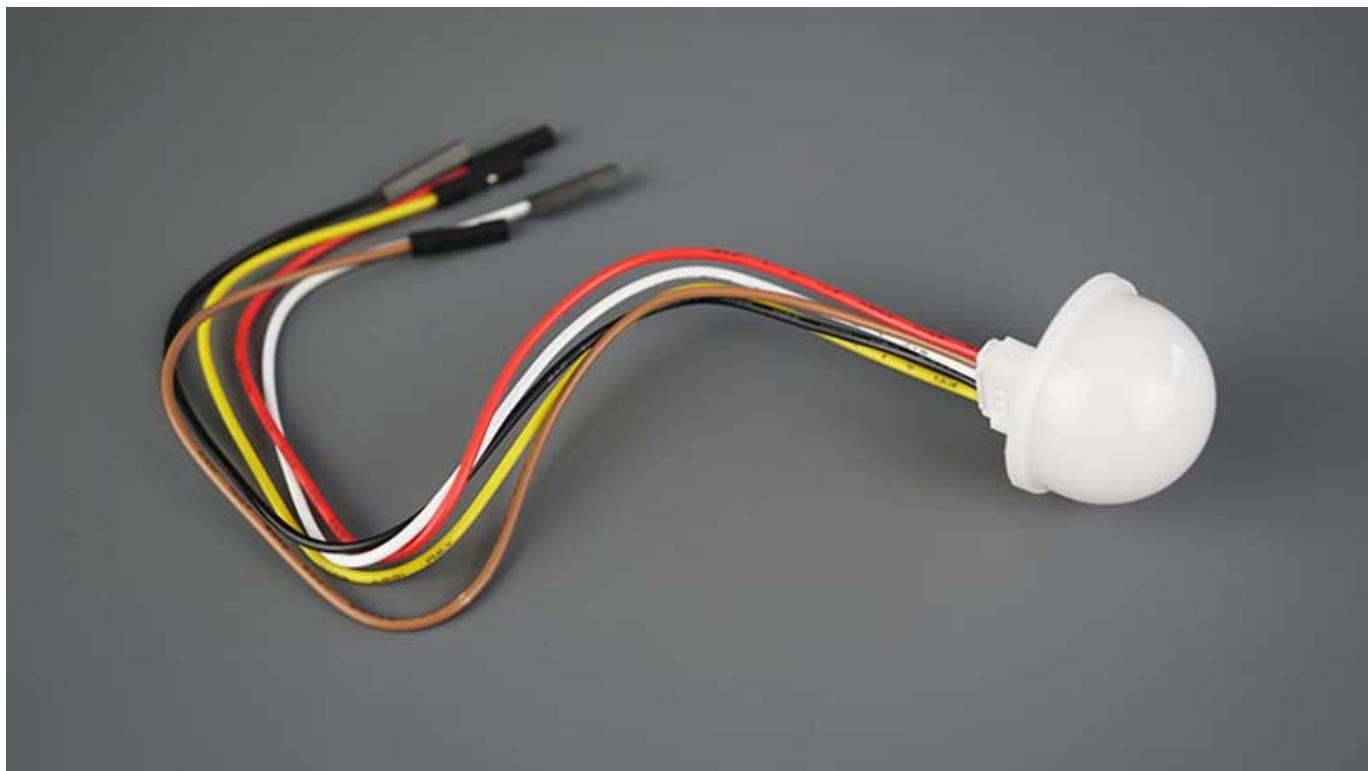
## Measurement Modes

The sensor supports two different measurement modes: *continuous measurement mode*, and *one-time measurement mode*. Each mode supports three different resolution modes.

<b>Low Resolution Mode</b>	4 lux precision	16 ms measurement time
<b>High Resolution Mode</b>	1 lux precision	120 ms measurement time
<b>High Resolution Mode 2</b>	0.5 lux precision	120 ms measurement time

In continuous measurement mode, the sensor continuously measures ambient light values. In one-time measurement mode, the sensor measures the ambient light value once, and then it goes to power down mode.

## Applications



The BH1750 is an ambient light sensor so it can be used in a wide variety of projects.



- to detect if it is day or night;
- to adjust or turn on/off LED's brightness accordingly to ambient light;
- to adjust LCDs and screen's brightness;
- to detect if an LED is lit;
- ...

## BH1750 Pinout



Here's the BH1750 Pinout:

<b>VCC</b>	Powers the sensor (3.3V or 5V)
<b>GND</b>	Common GND
<b>SCL</b>	SCL pin for I2C communication
<b>SDA (Data)</b>	SDA pin for I2C communication
<b>ADD*</b>	Selects address

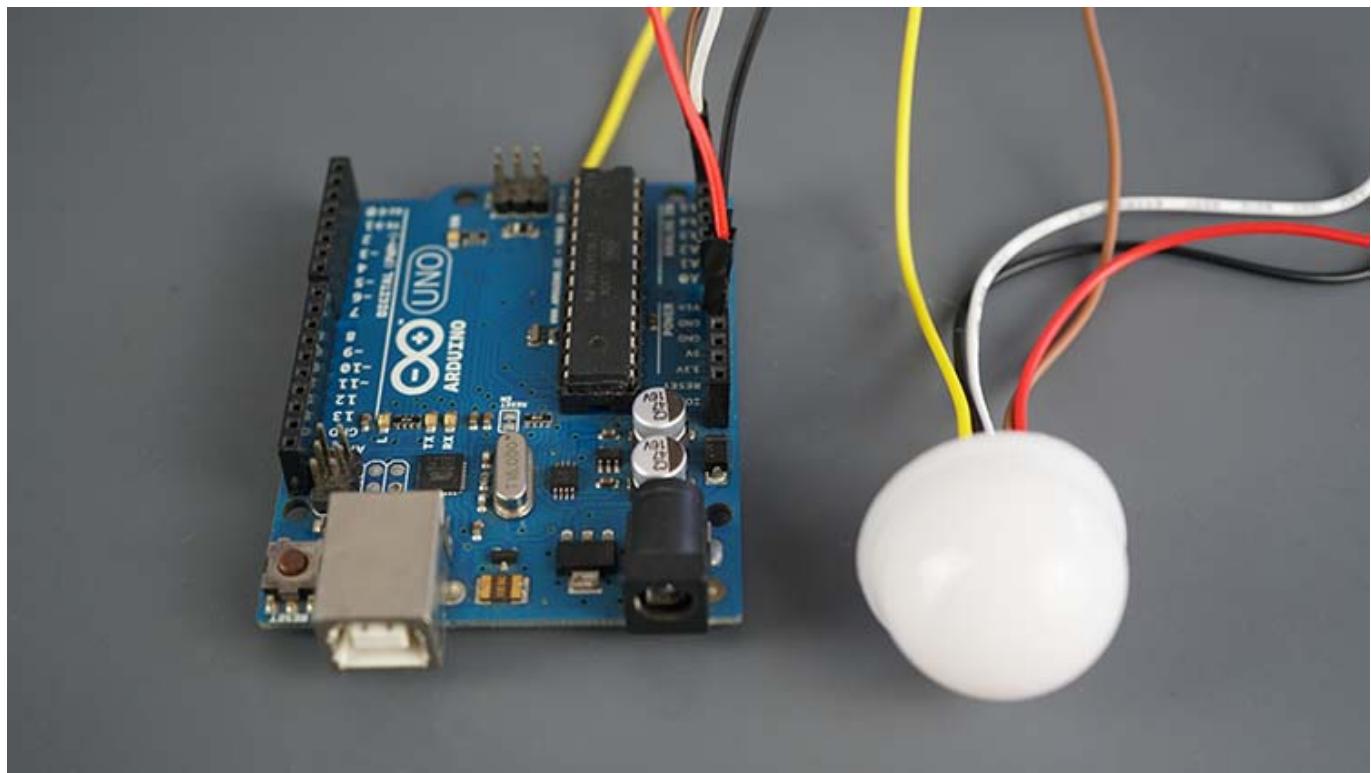


The ADD pin is used to set the I2C sensor address. If the voltage on that pin is less than 0.7VCC (pin is left floating or connected to GND), the I2C address is 0x23 . But, if the voltage is higher than 0.7xVCC (pin is connected to VCC), the address is 0x5C . In summary:

- ADD pin floating or connected to GND → address: 0x23
- ADD pin connected to VCC → address: 0x5C

## BH1750 I2C Interface

The BH1750 ambient light sensor supports I2C interface.



You can connect the BH1750 sensor to the Arduino using the default's I2C pins (these are the pins for the Arduino UNO, if you're using another model, check its I2C pins):

BH1750	Arduino
SCL	A5



# BH1750: Read Ambient Light with Arduino

Now that you are more familiar with the BH1750 sensor, let's test it. In this section, we'll build a simple project that reads the ambient light and displays it in the Arduino IDE Serial Monitor.

## Parts Required

To complete this tutorial you need the following parts:

- [BH1750 ambient light sensor](#)
- [Arduino \(read Best Arduino starter kits\)](#)
- [Breadboard \(optional\)](#)
- [Jumper wires \(optional\)](#)

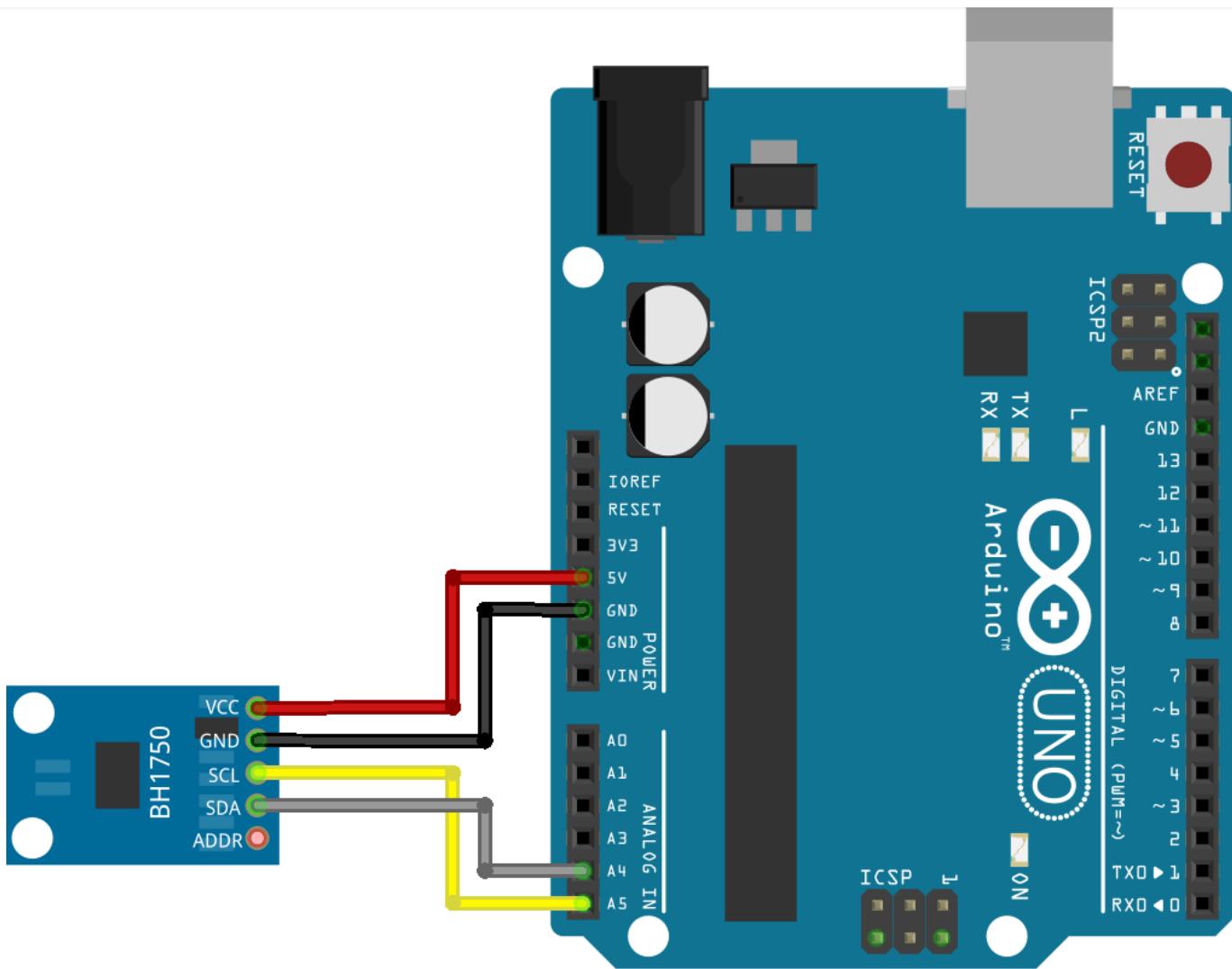
You can use the preceding links or go directly to [MakerAdvisor.com/tools](#) to find all the parts for your projects at the best price!



## Schematic – Arduino with BH1750

Wire the BH1750 sensor to the Arduino I2C pins. You can follow the next schematic diagram.





You can also follow the next table:

BH1750	Arduino
VCC	5V
GND	GND
SCL	A5
SDA (Data)	A4
ADD*	Don't connect



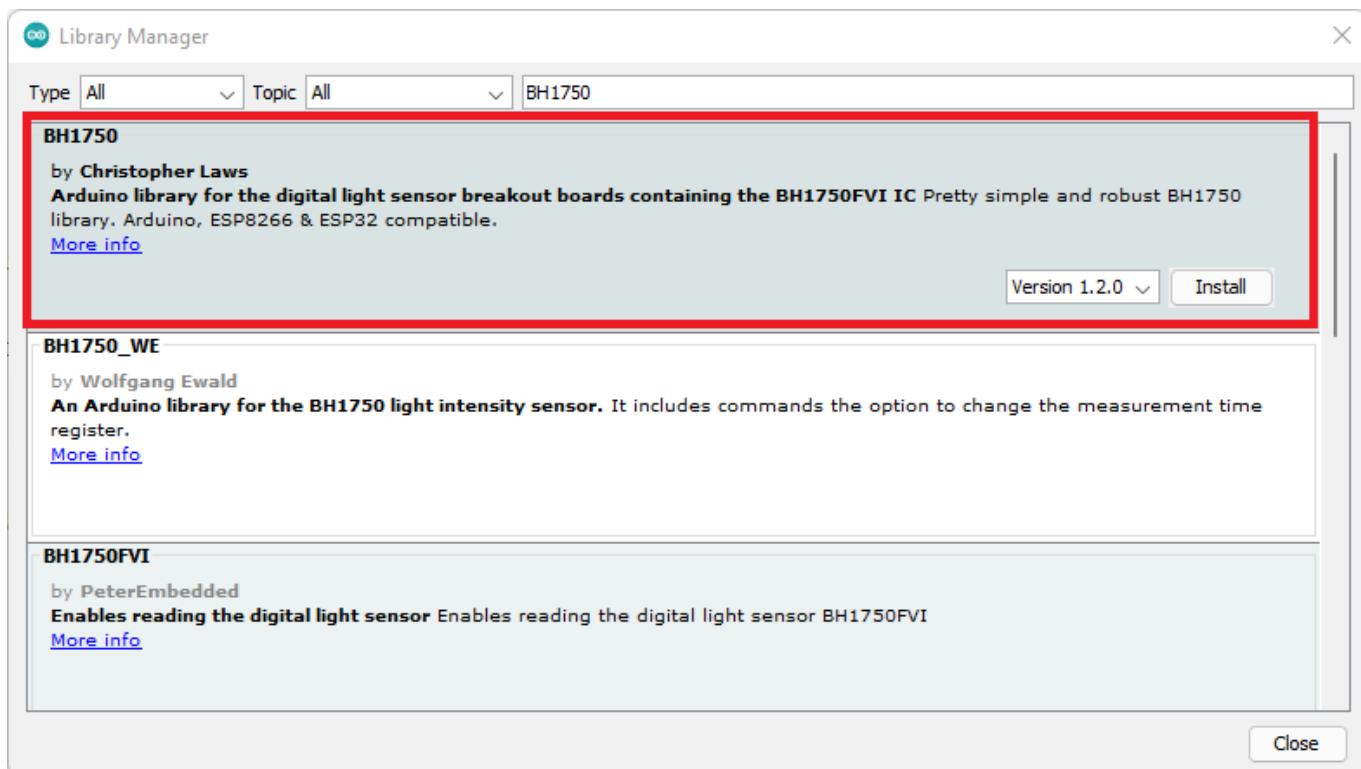
By not connecting the **ADD** pin, we're selecting **0x23** I2C address. Connect it to 3.3V to select **0x5C** address instead.

## Installing the BH1750 Library

There are several libraries to read from the BH1750 sensor. We'll use the [BH1750 library by Christopher Laws](#). It is compatible with the ESP32, ESP8266, and Arduino.

Open your Arduino IDE and go to **Sketch > Include Library > Manage Libraries**. The Library Manager should open.

Search for “**BH1750**” on the search box and install the *BH1750 library by Christopher Laws*.



## Code – Reading BH1750 Ambient Light Sensor

Copy the following code to your Arduino IDE. This code simply reads ambient light in lux and displays the values on the Serial Monitor. It is the example code from the library called **BH1750test** (you can access it in **File > Examples > BH1750 > BH1750test**)



```
/*
 Example of BH1750 library usage. This example initialises the BH1750
 */

#include <Wire.h>
#include <BH1750.h>

BH1750 lightMeter;

void setup(){
    Serial.begin(9600);

    // Initialize the I2C bus (BH1750 library doesn't do this automatically)
    Wire.begin();
    // On esp8266 you can select SCL and SDA pins using Wire.begin(D2, D1)
    // For Wemos / Lolin D1 Mini Pro and the Ambient Light shield use
    // SCL = D2, SDA = D1

    lightMeter.begin();

    Serial.println(F("BH1750 Test begin"));
}

void loop() {
    float lux = lightMeter.readLightLevel();
    Serial.print("Light: ");
    Serial.print(lux);
    Serial.println(" lx");
}
```

[View raw code](#)

The library also provides other examples worth exploring.

## How the Code Works



We start by including the required libraries. The `Wire.h` library to use I2C communication protocol and the `BH1750.h` library to read from the sensor.

```
#include <Wire.h>
#include <BH1750.h>
```

Then, we create a `BH1750` object called `lightMeter`.

```
BH1750 lightMeter;
```

In the `setup()`, initialize the Serial Monitor at a baud rate of 9600.

```
Serial.begin(9600);
```

Initialize I2C communication protocol. It will start an I2C communication on the microcontroller's default I2C pins. If you want to use different I2C pins, pass them to the `begin()` method like this `Wire.begin(SDA, SCL)`.

```
Wire.begin();
```

Initialize the sensor using the `begin()` method on the `BH1750` object (`lightMeter`).

```
lightMeter.begin();
```

In the `loop()`, we create a variable called `lux`, that saves the luminance values. To get the value, you simply call the `readLightLevel()` function on the `BH1750` object (`lightMeter`).



```
float lux = lightMeter.readLightLevel();
```

Finally, display the measurement on the Serial Monitor.

```
Serial.print("Light: ");
Serial.print(lux);
Serial.println(" lx");
```

You get and print a new reading every second.

```
delay(1000);
```

## Demonstration

Now, you can upload the code to your board. First, connect your board to your computer. Then, go to **Tools > Board** and select the Arduino board you're using. Go to **Tools > Port** and select the COM port your board is connected to. Finally, click on the upload button.



After successfully uploading the code, open the Serial Monitor at a baud rate of 9600.

New luminance readings should be printed in the Serial Monitor.



```
Light: 1417.50 lx
Light: 219.17 lx
Light: 99.17 lx
Light: 102.50 lx
Light: 106.67 lx
Light: 109.17 lx
Light: 109.17 lx
Light: 110.00 lx
Light: 366.67 lx
Light: 72.50 lx
Light: 2.50 lx
Light: 2.50 lx
Light: 2.50 lx
```

Autoscroll  Show timestamp    Newline    9600 baud    Clear output

## Other Useful Functions

The library we're using with the BH1750 sensor provides other examples that illustrate other useful functions and features. You can [check all BH1750 library examples here](#).

## Setting Measurement Mode

By default, the library uses the continuous high resolution measurement mode, but you can change it by passing the desired measurement mode to the `begin()` method when initializing the sensor. For example:

```
lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE)
```

Here's a list of all available modes:



- BH1750\_CONTINUOUS\_HIGH\_RES\_MODE (default)
- BH1750\_CONTINUOUS\_HIGH\_RES\_MODE\_2
- BH1750\_ONE\_TIME\_LOW\_RES\_MODE
- BH1750\_ONE\_TIME\_HIGH\_RES\_MODE
- BH1750\_ONE\_TIME\_HIGH\_RES\_MODE\_2

See the properties of each mode in [this previous section](#).

## Wrapping Up

In this tutorial, you've learned how to use the BH1750 ambient light sensor with the Arduino Uno. The sensor is very easy to use. It uses I2C communication protocol, which makes wiring simple, and the library provides methods to easily get the readings.

We hope you found this tutorial useful. Tell us in the comments below in which project would you use the BH1750 sensor.

We have tutorials for other sensors with the Arduino board that you may like:

- [Arduino with BMP388: Altimeter Sensor](#)
- [Arduino with DS18B20: Temperature Sensor](#)
- [Arduino with BME680: Gas, Pressure, Humidity, and Temperature Sensor](#)
- [Arduino with BME280: Temperature, Humidity, and Pressure Sensor](#)
- [Arduino DHT11/DHT22: Temperature, and Humidity Sensor](#)
- [Arduino HC-SR04: Ultrasonic Distance Sensor](#)
- [Arduino PIR: Motion Sensor](#)
- [Arduino BMP180: Pressure Sensor](#)
- [Arduino LM35: Temperature Sensor](#)

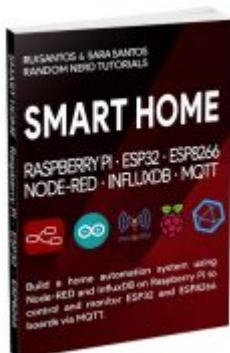
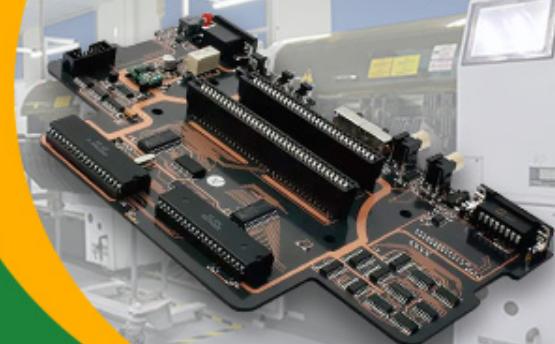
Thanks for reading.



# ONLY \$5 for 10 PCBs

✓ 24-hour Build Time ✓ Quality Guaranteed

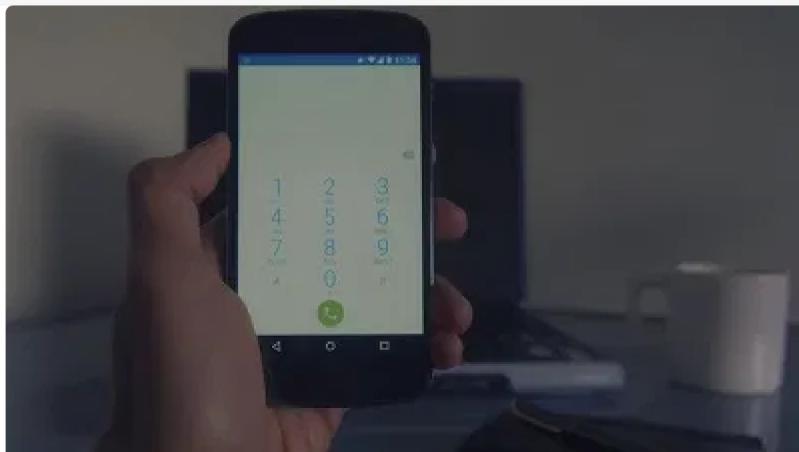
✓ Most Soldermask Colors:

[Order now](#)[www.pcbway.com](http://wwwpcbway.com)

## SMART HOME with Raspberry Pi, ESP32, ESP8266 [eBook]

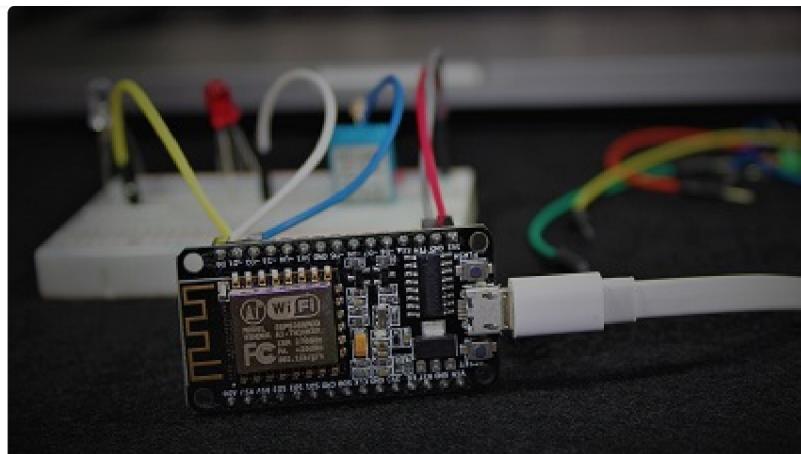
Learn how to build a home automation system and we'll cover the following main subjects:  
Node-RED, Node-RED Dashboard, Raspberry Pi, ESP32, ESP8266, MQTT, and InfluxDB  
[database DOWNLOAD »](#)

## Recommended Resources

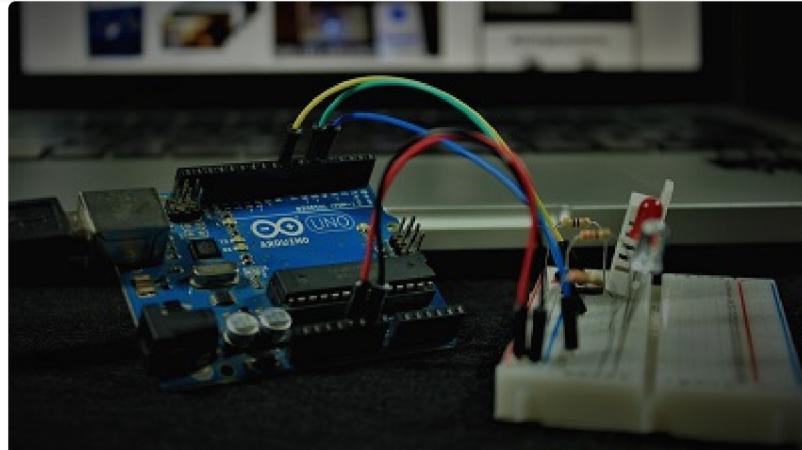


[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino and Node-RFID





[\*\*Home Automation using ESP8266 eBook and video course\*\*](#) » Build IoT and home automation projects.



[\*\*Arduino Step-by-Step Projects\*\*](#) » Build 25 Arduino projects with our course, even with no prior experience!

## What to Read Next...



[ESP32 PWM with Arduino IDE \(Analog Output\)](#)

[ESP32 with PIR Motion Sensor using Interrupts and Timers](#)



[\[SOLVED\] Reconnect ESP32 to Wi-Fi Network After Lost Connection](#)

**Enjoyed this project? Stay updated by subscribing our newsletter!**

Your Email Address

SUBSCRIBE

## 14 thoughts on “Arduino with BH1750 Ambient Light Sensor”



Lahcene Akroud

March 11, 2022 at 4:57 pm

Hi,

I really did not understand if this project is intended to work with Arduino o



code to ESP32.

Please help clarify this.

Lahcene

[Reply](#)



**Sara Santos**

March 11, 2022 at 7:23 pm

Hi.

It is for the Arduino. It is fixed now.

For the ESP32, check this: <https://randomnerdtutorials.com/esp32-bh1750-ambient-light-sensor/>

Regards,

Sara

[Reply](#)



**Richard Mann**

April 2, 2022 at 5:10 am

I am building a model railroad and would like to use this to turn on night lights (street and building) how would I send to an output pin? Thanks I know this should be easy but just learning

[Reply](#)





Aart V.

April 8, 2022 at 7:25 am

Hi,

I got both types of BH1750 sensors (the “bare” one and the one with the dome) and I wonder how I can use them simultaneously on a UNO. One using I2C address 0x23 and the other one 0x5C. I understand the hardware part, but wonder how the software can be set up to get readings from both sensors. I checked the example that comes with the BH1750 library but that uses 2 different I2C busses (hardware and a software implementation). How can it be done with only one I2C bus (using different addresses)?

Thanks in advance!

[Reply](#)



Sara Santos

April 8, 2022 at 9:44 am

Hi.

You would need to create two different instances for the sensor.

BH1750 lightMeter1;

BH1750 lightMeter2;

Then, creat two Wire instances on different pins.

Finally, call the begin method for each sensor by passing as argument, the address and the corresponding wire instance. See line 65:

<https://github.com/claws/BH1750/blob/master/src/BH1750.h>

We have a tutorial about using multiple I2C sensors at the same time on theESP32. The tutorial can easily be adjusted for an Arduino, see here:



I hope this helps.

Regards,  
Sara

[Reply](#)



Aart V.

April 8, 2022 at 3:06 pm

Hi Sara,

Thanks for your fast reply! But I still have a problem. In my project I use a lot of pins (display, other sensors) and have no more pins available.

I checked the ESP32 tutorial and used the i2c scanner. I found the sensors can be connected at the same time (the ADD pin for sensor 1 floating and for sensor 2 connected to VCC). The sensors are detected on different addresses (x023 and x05C).

Is it possible to set up the BH1750 library so that lightMeter1 uses x023 and lightMeter2 uses x05C?

Thanks in advance again!

[Reply](#)



Sara Santos

April 8, 2022 at 8:45 pm

...



ADD pin floating or connected to GND → address: 0x23

ADD pin connected to VCC → address: 0x5C

Regards,

Sara

[Reply](#)



**NJI RUTH MBIKANG**

November 27, 2022 at 6:11 pm

hello when i upload the code and check on the serial monitor, i see [BH1750]

Device is not configured!

Light: -2.00 lx

even after applying torchlight on i get the same results. please i need help

[Reply](#)



**Sara Santos**

November 27, 2022 at 11:44 pm

Check you're wiring.

It is probably not wired properly.

[Reply](#)



**NJI RUTH MBIKANG**

December 26, 2022 at 1:44 pm



my wiring is correct im sure of it

[Reply](#)



**Sara Santos**

November 27, 2022 at 11:45 pm

You may also need to check the sensor I2C address:

[https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/LCD\\_I2C/I2C\\_Scanner.ino](https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/LCD_I2C/I2C_Scanner.ino)

[Reply](#)



**NJI RUTH MBIKANG**

December 26, 2022 at 1:43 pm

it says “no I2C devices found”

[Reply](#)



**MHDCHD**

August 12, 2023 at 10:57 am



analog 4&5 and it should go well.

[Reply](#)



**Bob**

November 27, 2023 at 12:27 am

The one thing that would be great to add would be the ability to write this data to a csv file so you can import into a spreadsheet. Just seeing in the serial monitor isn't that useful.

Thanks

[Reply](#)

## Leave a Comment

Name \*



Website

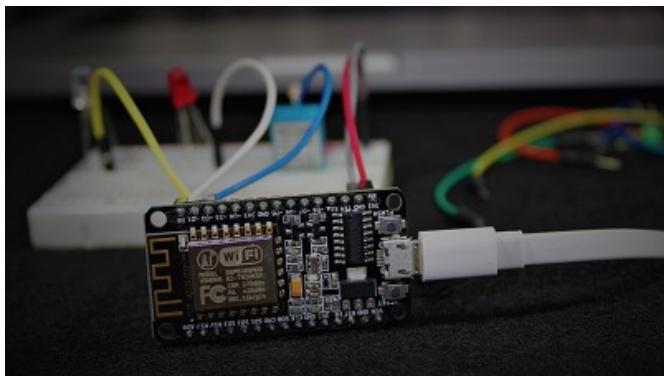
Notify me of follow-up comments by email.

Notify me of new posts by email.

**Post Comment**

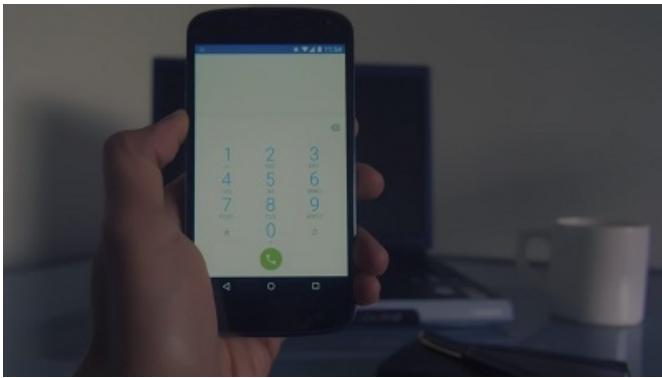


[Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »](#)



[Home Automation using ESP8266 eBook and video course »](#) Build IoT and home automation projects.





**Build Web Servers with ESP32 and  
ESP8266 » boards to control outputs and  
monitor sensors remotely.**

[About](#)   [Support](#)   [Terms and Conditions](#)   [Privacy Policy](#)   [Refunds](#)   [Complaints' Book](#)

[MakerAdvisor.com](#)   [Join the Lab](#)

Copyright © 2013-2023 · RandomNerdTutorials.com · All Rights Reserved