Energy saving house with ESP32

In this tutorial, you will learn how to set up your ESP32 so that it will automatically turn on its internal LED when it’s dark, and vice versa.

Difficulté **Moyen**

Durée **90 minute(s)**

Catégories **Électronique, Maison, Machines & Outils, Robotique**

Coût **25 EUR (€)**

Sommaire

Introduction

In this tutorial, you will learn how to set up your ESP32 so that it will automatically turn on its internal LED when it’s dark, and vice versa.

Matériaux

- ESP32 board
- micro USB cable to power the board
- laser cut house (downloadable here)
- Breadboard
- 3 jumper wires (one M/F)
- light sensor
- optional (Led)
- 1kohm resistor

Outils

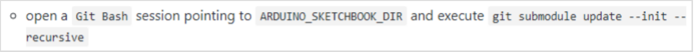
- computer with Arduino IDE installed

Étape 1 - Install Arduino IDE

You will need to use Arduino IDE to code and upload the firmware onto your ESP32 board.
Download the software by visiting Arduino IDE > Scroll down until you see the “Download the Arduino IDE” section and choose the version based upon your operating system (e.g. If you have Windows 7, choose “Windows Installer” / if you have Windows 10, choose “Windows app” > On the next page choose “Just download” and run the installation files.

Étape 2 - Prepare the ESP32 board on your computer

Follow the instructions provided on GitHub for your Operating System. For example, if you have Windows 7 or 10, choose "Instructions for Windows" / if you have a MacBook, choose "Instructions for Mac"
For the "Instructions for Windows" section, you can ignore the following step:



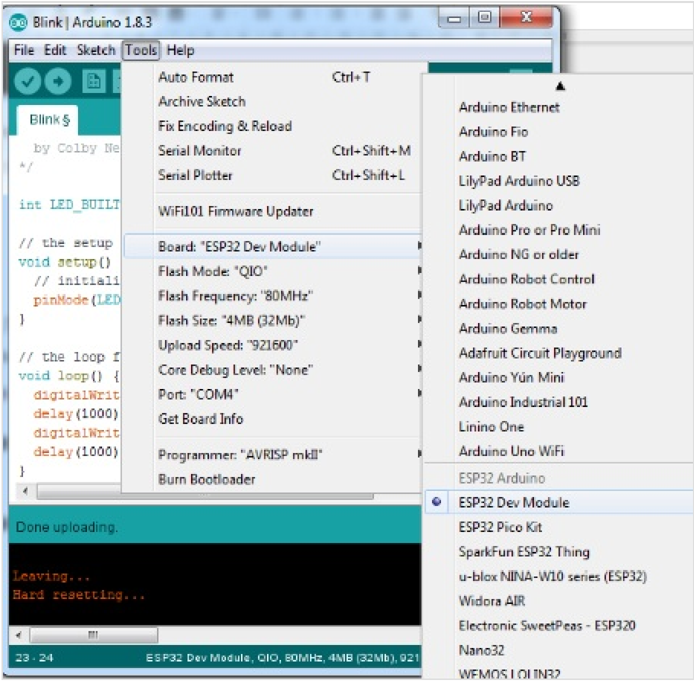
- open a `Git Bash` session pointing to `ARDUINO_SKETCHBOOK_DIR` and execute `git submodule update --init --recursive`

Étape 3 - Check that the board is correctly configured

Launch Arduino IDE and select "ESP32 Dev Module" from the Tools menu > Board.
Fetch the Blink example from File > Examples > 01.Basics > Blink.
write int LED_BUILTIN = 2; at the beginning of the code

```
/*
 ESP 32 Blink
 Turns on an LED on for one second, then off for one second, repeatedly.
 The ESP32 has an internal blue LED at D2 (GPIO 02)
*/
int LED_BUILTIN = 2;
void setup()
{
 pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
 digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
 delay(1000);                       // wait for a second
 digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
 delay(1000);                       // wait for a second
```

Finally, upload the code by using the right arrow (→) button at the top right corner of the window, by choosing Sketch > Upload or by pressing Ctrl+U on the keyboard.

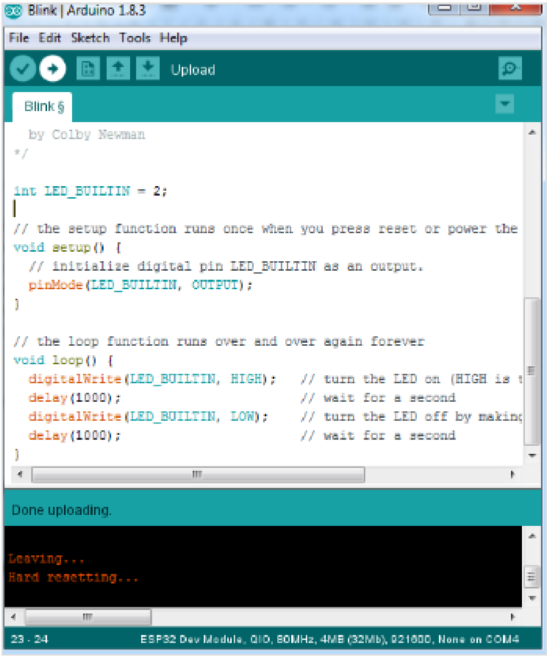


Board: "ESP32 Dev Module"
Flash Mode: "QIO"
Flash Frequency: "80MHz"
Flash Size: "4MB (32Mb)"
Upload Speed: "921600"
Core Debug Level: "None"
Port: "COM4"
Programmer: "AVRISP mkII"

Done uploading.

```
Leaving...
Hard resetting...
```

23 - 24 ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 921



```
  by Colby Newman
*/

int LED_BUILTIN = 2;

// the setup function runs once when you press reset or power the
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is t
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by makin
  delay(1000);                       // wait for a second
}
```

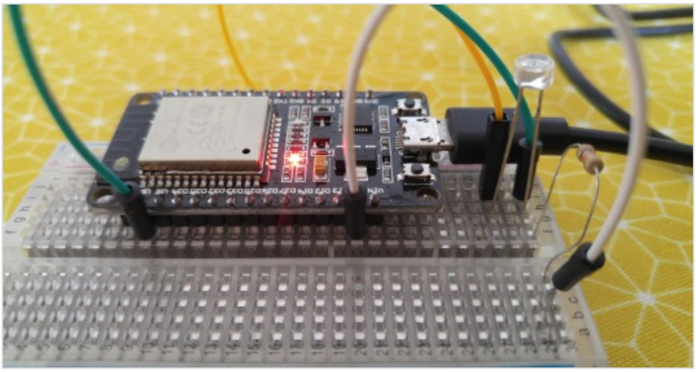
Done uploading.

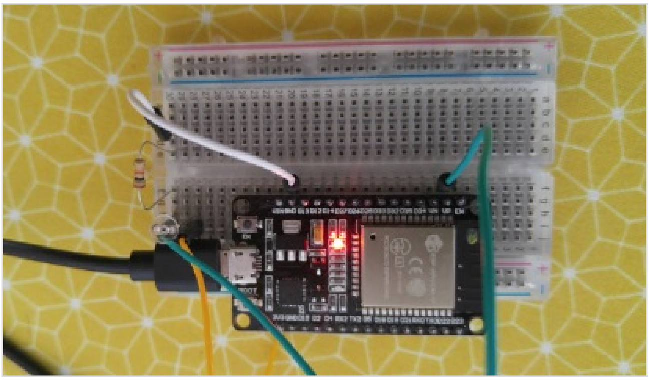
```
Leaving...
Hard resetting...
```

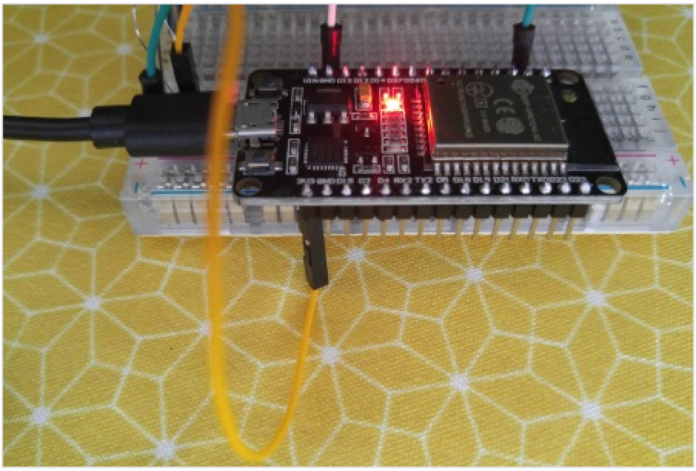
23 - 24 ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 921600, None on COM4

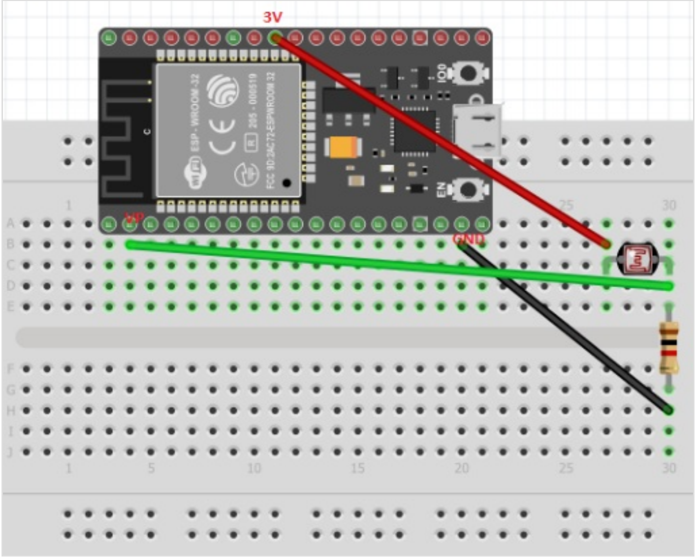
Étape 4 - Wiring the photoresistor to the ESP32

The shorter leg of the photoresistor is connected to 3V on the ESP32. The other leg is connected to pin VP (or 36) and at the same time to a 1kohm resistor, which in turn is connected to GND on the ESP32.
Note: if it's the first time you're using a breadboard, check out this video to understand how breadboards work.





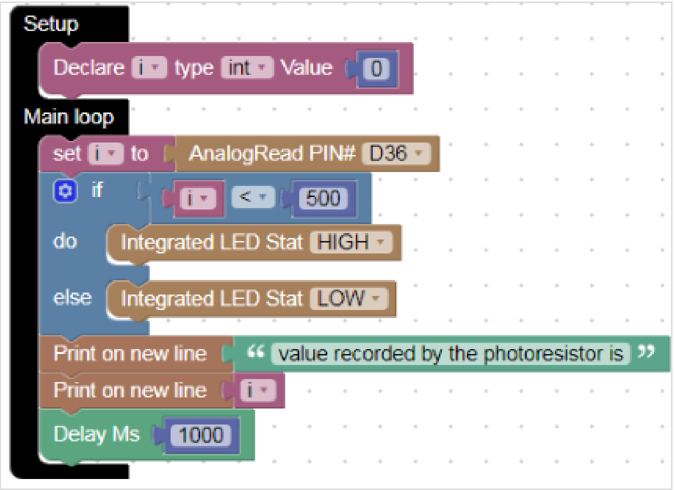




Étape 5 - Programming on tuniot

Let's create a program that turns the internal LED on and off depending on the amount of light recorded by the photoresistor. More precisely, the LED will be on when it's dark, and off when it's light.
For that we need to reach: http://easycoding.tn/esp32/demos/code/
Choose the appropriate blocks to create the code displayed below (see image).

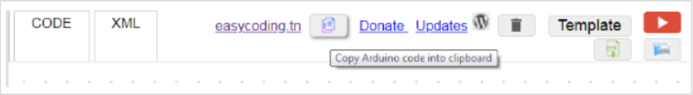
1. "Variables" section - Declare 'i' type 'int' Value + "Math" section - the actual value "0"
2. "Variables" section - set 'i' to + "IN/OUT" section > "Analog" subsection - AnalogREAD PIN#
3. "Logic" section - if... do & 'i' < 500 ('i' taken from "Variables" section & '500' taken from "Math" section)
4. "IN/OUT" section > "Digital" subsection - Integrated LED Stat
5. 'else' appears if you click on the "Settings" wheel / icon next to 'if' and choose 'else' from the list
6. "Serial" section - Print on new line
7. "Various" section - Delay Ms 1000

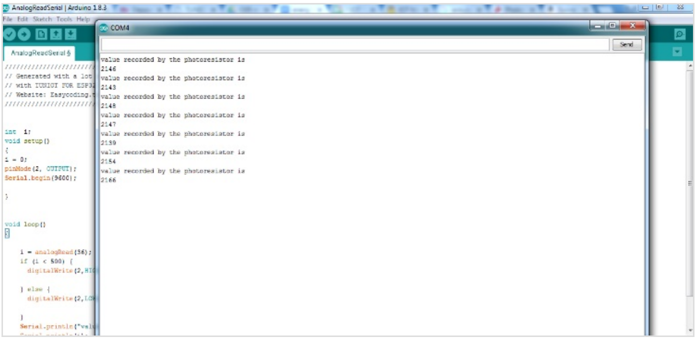


```
Setup
  Declare i type int Value 0
Main loop
  set i to AnalogRead PIN# D36
  if i < 500
  do Integrated LED Stat HIGH
  else Integrated LED Stat LOW
  Print on new line " value recorded by the photoresistor is "
  Print on new line i
  Delay Ms 1000
```

Étape 6 - Programming on Arduino IDE

To upload the code on Arduino IDE, click on the “Copy Arduino code into clipboard” button.
then paste the code onto Arduino IDE, and upload it to the ESP32.
If you click on Serial Monitor (top right of the Arduino IDE window, below the “X” button), you should see the value that the photoristor is recording at any given time.





```
//////////////////////
// Generated with a lot
// with TUNIOT FOR ESP32
// Website: Easycoding.t
//////////////////////


int  i;
void setup()
{
i = 0;
pinMode(2, OUTPUT);
Serial.begin(9600);

}


void loop()
{

    i = analogRead(36);
    if (i < 500) {
      digitalWrite(2,HIG

    } else {
      digitalWrite(2,LOW

    }
    Serial.println("valu
```

```
value recorded by the photoresistor is
2146
value recorded by the photoresistor is
2143
value recorded by the photoresistor is
2148
value recorded by the photoresistor is
2147
value recorded by the photoresistor is
2139
value recorded by the photoresistor is
2154
value recorded by the photoresistor is
2166
```

Notes et références

This tutorial has been developed as part of the iTech project, co-financed by the Erasmus + programme of the European Union.
For more details contact info@digijeunes.com