Esp 32

Algorithm for ESP32 turning on and off an internal LED light.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1- Components**

What we need to create this simple project is

- 1 ESP32 device.

- 1 Breadboard.

- USB cable

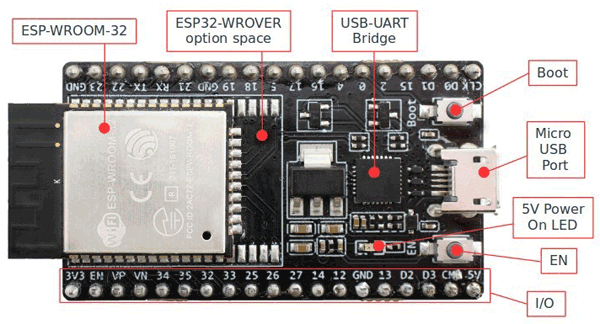
- IDE (we’re using Arduino IDE)

2- **ESP32 Specifications and Pinout**

ESP32 has a dual-core processor which means it has two processors. The ESP32 will run on breakout boards and modules from **160MHz up to 240MHz**.

The ESP32 chip comes with 48 pins with multiple functions. Not all pins are exposed in all ESP32 development boards, and some pins cannot be used.

For more details about pin description, you can refer [ESP32 datasheet](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).



(fig 1 - esp32 device detailed)

3- **Preparing ESP32 in Arduino IDE** on your device(pc/laptop)

Step 1: First you need to download and install **Arduino IDE** software. You can get it from which you can download from <https://www.arduino.cc/en/Main/Software>

Step 2: After installing, open IDE and go to

**Files -> Preferences** and open preference window and in the **“Additional Boards Manager URL’s”** :

This box may be empty or contain some other URL. You just have to paste the below URL into this box :

<https://dl.espressif.com/dl/package_esp32_index.json>

Step 3: After pasting the given URL, Now press OK and the window will disappear

**4- build the hardware:**

Use: Bread board and ESP32

Connect the ESP32 to the breadboard.

**5- Programming ESP32 With Arduino IDE**

Step 1: connect your ESP32 to your computer using a micro-USB cable, make sure that Red LED goes high after connecting it with the PC.

Step 2: Now you have to select your board; so go to**Tools-> Boards** and select **“ESP32 Dev Module”.**

Step3: create a new project, write your code to turn on/off led

**code in C :**

#define LED1 2  
  
void setup() {  
pinMode(LED1,OUTPUT);  
}  
  
void loop() {  
delay(1000);  
digitalWrite(LED1,HIGH);  
delay(1000);  
digitalWrite(LED1,LOW);  
}

Step 4**:** Uploading the code

To upload your code into the device, all you have to do is connect your ESP32 to your computer you’re currently working with and then press upload.

After uploading the code, you will see a window like this and finds **“Done uploading”**.

Now that you are done uploading it into the ESP32, you can see that the light is blinking and that’s how you know your code works and uploaded.

Ref:

<https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>

https://medium.com/@fdika24/esp32-101-turning-on-led-lights-e4e7dc85d77

<https://www.arduino.cc/en/Main/Software>