



### Learning Outcomes

- Learn the different power cycles of ESP32 MCU
- Implement a basic power measurement circuit and power calculation

### Instructions:

- Make a comprehensive laboratory report, document your code, and present the results and conclusions. For this lab, you should work in **pairs**.
- The laboratory reports must be sent using the Blackboard upload link. The deadline for Laboratory Assignment 4 is February 23<sup>rd</sup>.

## Experiment 1: Power measurement circuit

In this experiment, we will learn how to put the ESP32 in different power cycles, and measure its power consumption.

Step -1: For the first task, program the ESP32 with a DeepSleep code. You can create your own, or implement the example code in File -> Examples -> ESP32 -> DeepSleep -> TimerWakeUp.

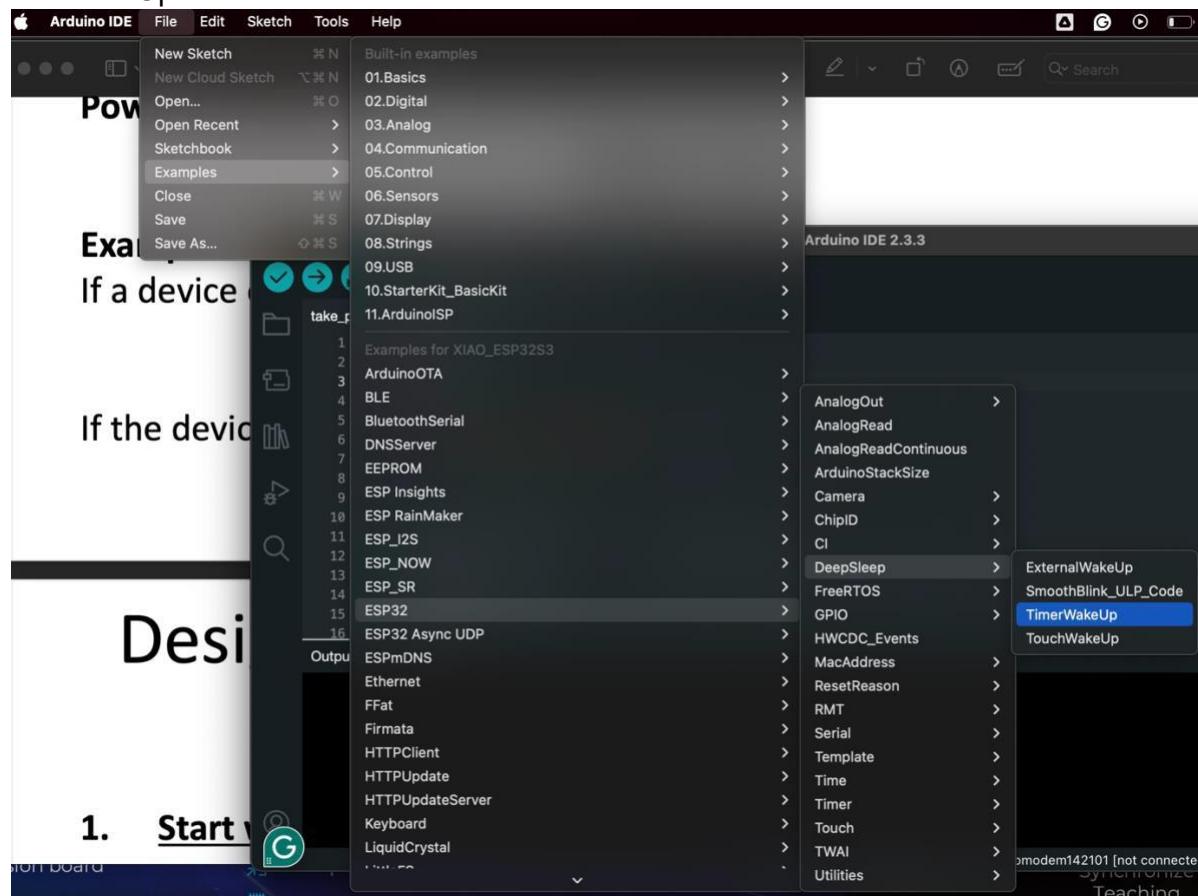


Fig 1 – Example code.



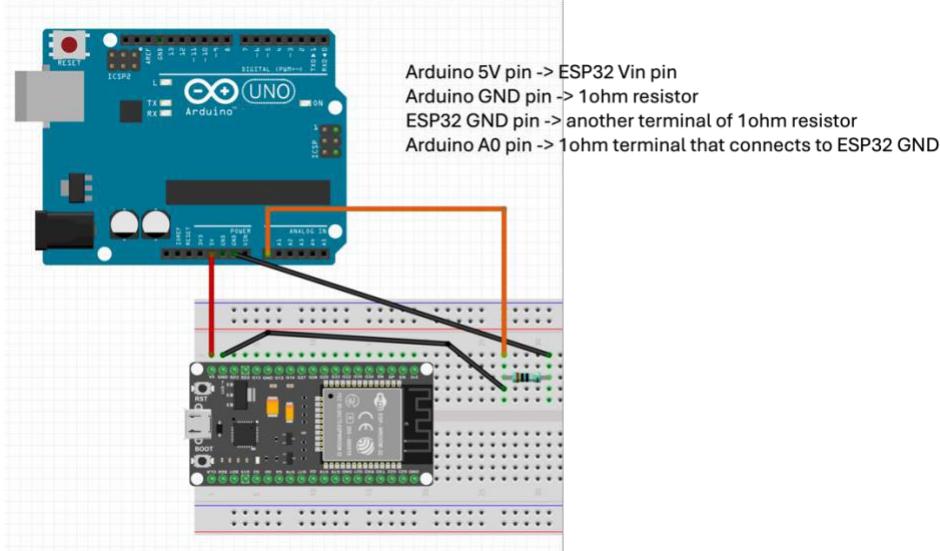
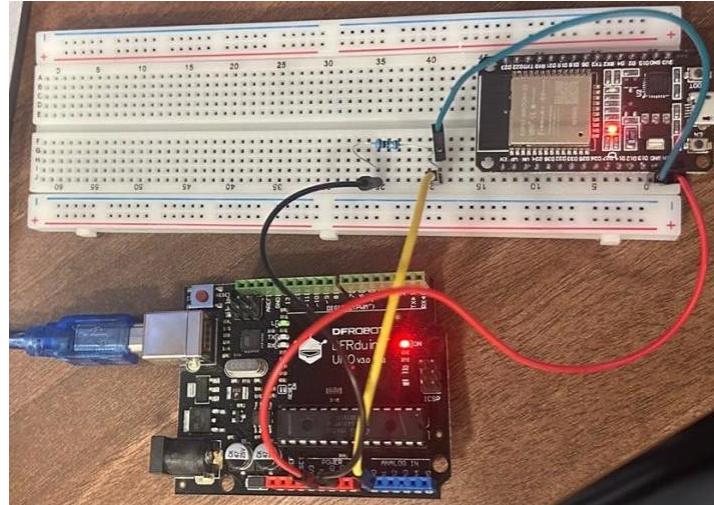
## TIE 204 MVP Studio Lab 4

Step 2 – After programming the ESP32 with the example code, you should observe the following serial output:

The screenshot shows the Arduino Serial Monitor window. The title bar says "93". The tabs are "Output" and "Serial Monitor" (which is selected). The message input field contains "Message (Enter to send message to 'ESP32 Dev Module' on '/dev/cu.usbserial-0001')". The baud rate is set to "115200 baud". The serial output window displays the following text:  
Message (Enter to send message to 'ESP32 Dev Module' on '/dev/cu.usbserial-0001')  
configsip: 0, SPIWP:0xee  
clk\_drv:0x00,q\_drv:0x00,d\_drv:0x00,cs0\_drv:0x00,hd\_drv:0x00,wp\_drv:0x00  
mode:DIO, clock div:1  
load:0x3fff0030,len:4916  
load:0x40078000,len:16436  
load:0x40080400,len:4  
ho 8 tail 4 room 4  
load:0x40080404,len:3524  
entry 0x400805b8  
Boot number: 6  
Wakeup caused by timer  
Setup ESP32 to sleep for every 5 Seconds  
Going to sleep now

In the serial output, you should see how many times it went to sleep and booted itself, “Boot number,” and the reason it woke up, “Wakeup caused by...”

Step 3- Now, disconnect the **USB cable of the ESP32** and make the circuit following the schematic:



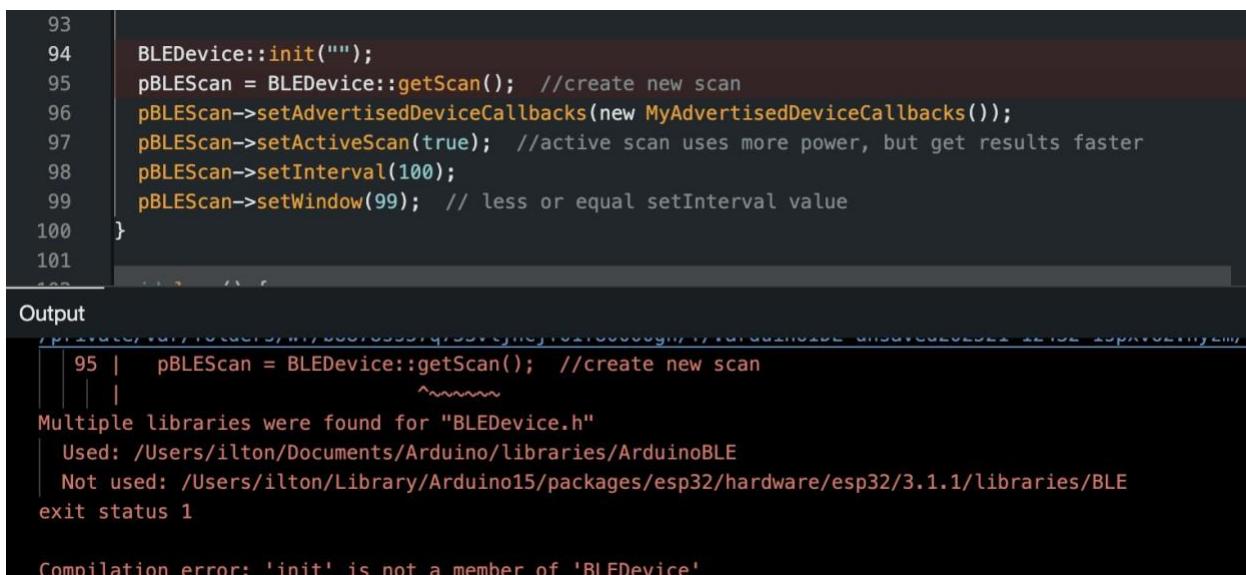


Step 4 – Now, programing only the Arduino UNO and using what was taught to you on the lecture, create a code to read the power of the ESP32 and write down in your report what was the power the ESP32 used when sleeping and the power the ESP32 used when woke up.

## Experiment 2: BLE Power measurement

Step 1 – Now, reconnect the ESP32. Using the example code Beacon\_Scanner (File -> Examples -> BLE -> Beacon Scanner).

Can you program the ESP32 without a program? If you face the problem below, it means your Arduino has installed new libraries and you can not use the ESP32 BLE library.



The screenshot shows the Arduino IDE interface. The code editor displays the following snippet:

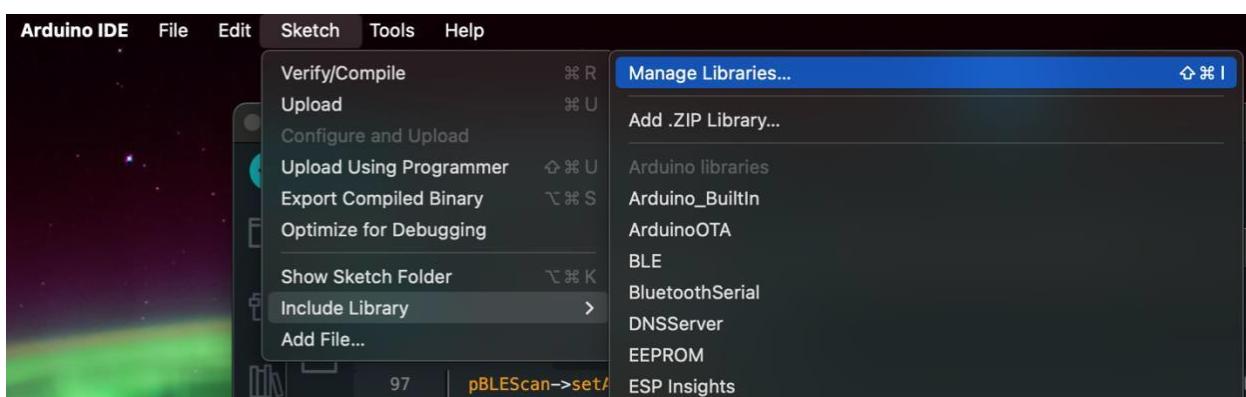
```
93
94     BLEDevice::init("");
95     pBLEScan = BLEDevice::getScan(); //create new scan
96     pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
97     pBLEScan->setActiveScan(true); //active scan uses more power, but get results faster
98     pBLEScan->setInterval(100);
99     pBLEScan->setWindow(99); // less or equal setInterval value
100 }
```

The output window shows the following error message:

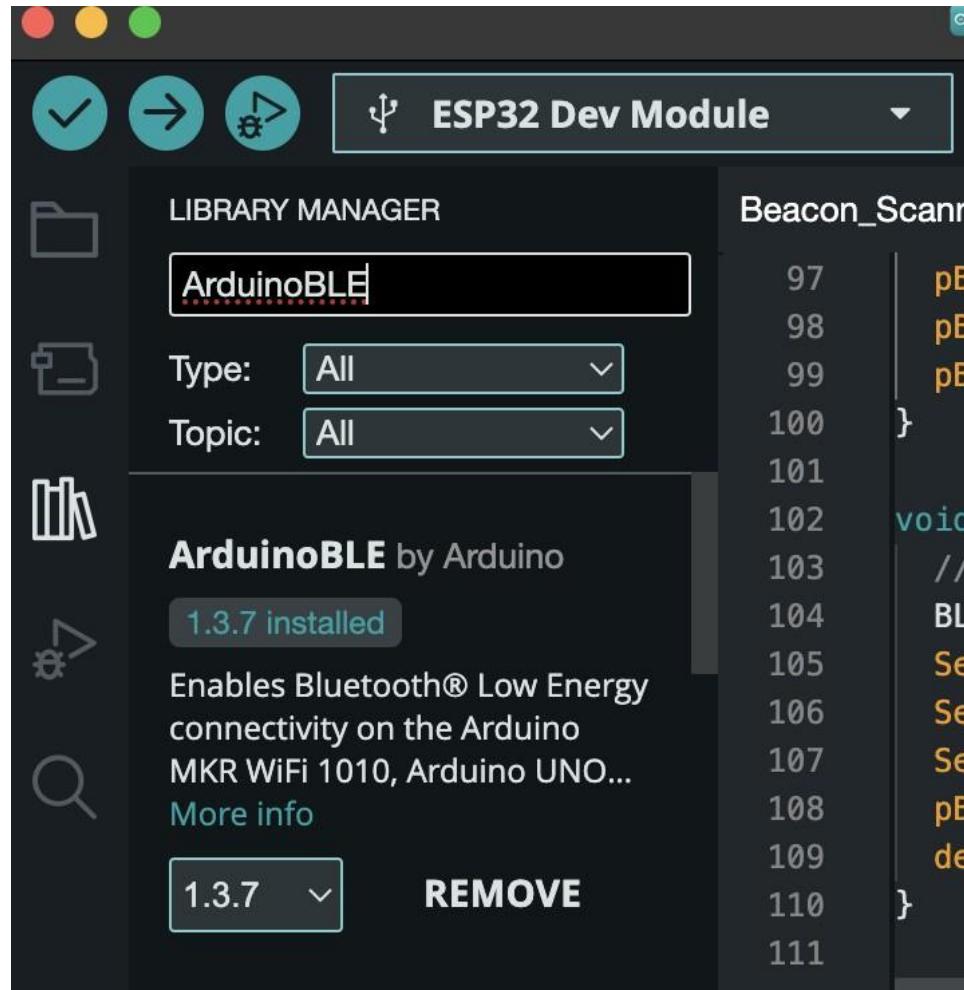
```
95 |     pBLEScan = BLEDevice::getScan(); //create new scan
|           ^
Multiple libraries were found for "BLEDevice.h"
Used: /Users/ilton/Documents/Arduino/libraries/ArduinoBLE
Not used: /Users/ilton/Library/Arduino15/packages/esp32/hardware/esp32/3.1.1/libraries/BLE
exit status 1

Compilation error: 'init' is not a member of 'BLEDevice'
```

To solve this, go to Sketch -> Include Library -> Manage Libraries



Look for ArduinoBLE, and remove it.



After removing ArduinoBLE, try to program the code once again. You should now be able to program the ESP32 with a BLE code.

Step 2 – Now, observe in the serial port that the ESP32 is scanning for BLE devices and then stays idle for 2000ms after the scan is done. Repeat the step 4 of Experiment 1, and write down in your