NAME	**K.VISHNU**
DEPT	**ECE**
REG NO	**420121106059**
COLLEGE CODE	**4201**
GROUP	**IBM-GROUP 5**

Module 3: Getting started with ESP32 And Wokwi platform:

Introduction

We will go through the steps necessary to set up an ESP32 / ESP8266 development board with the Arduino Cloud IoT. To test it out, we will send random values from the board to the cloud, and set up a switch that can enable the built-in LED on the board.

ESP32 Simulation

The ESP32 is a popular WiFi and Bluetooth-enabled microcontroller, widely used for IoT Projects. Wokwi simulates the ESP32, ESP32-C3, ESP32-S2, ESP32-S3, ESP32-C6 (beta), and ESP32-H2

Getting Started

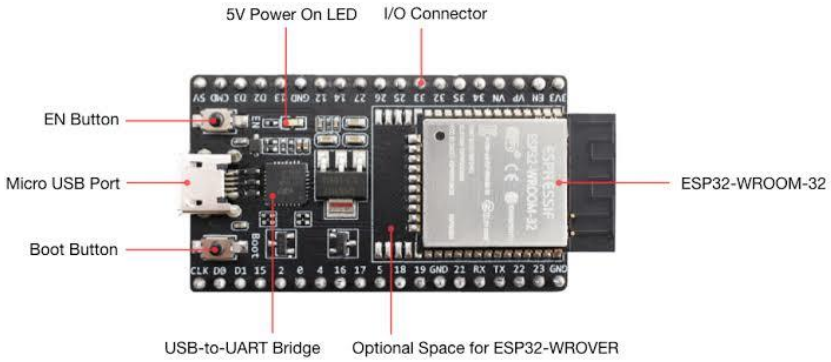
We can use the ESP32 simulator to run different kinds of applications:

ESP32 Arduino Core projects (including ESP-IDF projects)

MicroPython and CircuitPython projects (examples at https://wokwi.com/micropython)

Rust projects (see https://wokwi.com/rust)

Custom application firmware files (e.g. applications built using the ESP-IDF)



MicroPython:

Start from the MicroPython ESP32 Project Template, or from the MicroPython ESP32 Blink Example.

Custom Application Firmware:

Open the ESP32 custom application project template, and press “F1” in the code editor. Then choose “Upload Firmware and Start Simulation…”. Choose any .bin, .elf or .uf2 file from your computer and the simulation will start

Simulator Examples

ESP-IDF Examples

The following examples use the ESP-IDF functions. They are compiled using Arduino ESP32 Core:

- Blink using FreeRTOS API
- Binary LED counter using FreeRTOS tasks
- GPIO button input + interrupts
- WiFi Example

Custom firmware offset:

When loading a custom firmware, you can specify the offset of the firmware in the flash memory. By default, Wokwi will look at the firmware binary and try to figure out the offset automatically, based on the presence of the bootloader and the type of the chip. If Wokwi can't figure out the offset, it will assume that your firmware is an application firmware and load it at offset 0x10000.

Changing the MAC address

You can change the MAC address of the WiFi interface by adding the following attribute to the chip:

Attribute	Description	Default
macAddress	MAC address of the WiFi interface, e.g. "24:0a:c4:12:45:56"	"24:0a:c4:00:01:10"