

# Week 1: Introduction to IoT and ESP32-S3

## ESP-IDF Development Environment

Assist. Prof. Yıldırım YILMAZ

Recep Tayyip Erdoğan University

March 4, 2025

# Outline

- 1 1.1 What is IoT?
- 2 1.2 Introduction to ESP32-S3
- 3 1.3 Introduction to ESP-IDF
- 4 1.4 Setting Up the Development Environment
- 5 1.5 Creating Your First Project
- 6 1.6 Introduction to Serial Communication
- 7 1.7 Homework and Practice
- 8 Key Takeaways

# 1.1 What is IoT?

- **Definition:** The Internet of Things (IoT) refers to a network of physical devices (things) that are embedded with sensors, software, and connectivity to exchange data over the internet.
- **Applications:**
  - Smart homes (lighting, security, thermostats).
  - Wearable devices (fitness trackers, smartwatches).
  - Industrial automation (predictive maintenance, asset tracking).
  - Agriculture (smart irrigation, soil monitoring).
- **Key Components of IoT:**
  - Sensors/Actuators: Collect data or perform actions.
  - Connectivity: Wi-Fi, Bluetooth, LoRa, etc.
  - Data Processing: Edge computing or cloud-based processing.
  - User Interface: Mobile apps, web dashboards.

## 1.2 Introduction to ESP32-S3

- **Overview:**

- ESP32-S3 is a powerful, low-cost microcontroller with Wi-Fi and Bluetooth connectivity.
- Designed for IoT applications, it combines processing power, wireless connectivity, and low power consumption.

- **Key Features:**

- Dual-core Xtensa LX7 processor (up to 240 MHz).
- Wi-Fi 802.11 b/g/n and Bluetooth 5.0 (Classic and BLE).
- 512 KB SRAM, 384 KB ROM, and support for external flash.
- Multiple GPIOs, ADC, DAC, I2C, SPI, UART, and more.
- Built-in security features (secure boot, flash encryption).

- **Typical Use Cases:**

- Home automation, wearable devices, industrial control, and more.

# 1.3 Introduction to ESP-IDF

- **What is ESP-IDF?**

- ESP-IDF (Espressif IoT Development Framework) is the official development framework for ESP32 series chips.
- It provides libraries, tools, and APIs to develop applications for ESP32.

- **Key Features:**

- FreeRTOS-based real-time operating system.
- Comprehensive documentation and examples.
- Support for Wi-Fi, Bluetooth, and other peripherals.

- **Why Use ESP-IDF?**

- Optimized for ESP32 chips.
- Provides low-level control and flexibility.
- Active community and regular updates.

# 1.4 Setting Up the Development Environment

- **Required Tools:**

- **Hardware:**

- ESP32-S3 development board.
    - USB cable for power and programming.

- **Software:**

- VS Code or PlatformIO IDE.
    - ESP-IDF (can be installed via VS Code extension or manually).
    - Python 3.x (required for ESP-IDF tools).

- **Installation Steps:**

- Install ESP-IDF: Download and install ESP-IDF using the official guide.
  - Install VS Code: Download and install VS Code.
  - Set Up the Toolchain: Follow the ESP-IDF setup wizard.
  - Verify Installation: Run `idf.py --version` to check installation.

# 1.5 Creating Your First Project

## • Steps to Create a New Project:

- Open VS Code and use the ESP-IDF extension to create a new project.
- Choose a project template (e.g., "Hello World").
- Build the project using `idf.py build`.
- Connect your ESP32-S3 board via USB.
- Flash the program using `idf.py -p PORT flash`.
- Monitor the output using `idf.py -p PORT monitor`.

## • Understanding the Code:

- The "Hello World" example prints a message to the serial monitor.
- Key components:
  - `app_main()`: Entry point of the program.
  - `printf()`: Used to send data to the serial monitor.

# 1.6 Introduction to Serial Communication

- **What is Serial Communication?**

- A method of transmitting data between devices one bit at a time.
- Commonly used for debugging and logging in embedded systems.

- **Using the Serial Monitor:**

- The ESP-IDF serial monitor (`idf.py monitor`) displays output from the ESP32-S3.
- Use `printf()` or `ESP_LOGI()` to send data to the serial monitor.

- **Example Code:**

# Code Example

```
1 #include <stdio.h>
2 int main() {
3     printf("Hello, world!\n");
4     return 0;
5 }
```

# 1.7 Homework and Practice

- **Tasks:**

- Set up your development environment and verify it works by flashing the "Hello World" example.
- Modify the "Hello World" example to print your name and student ID.
- Explore the ESP-IDF documentation and find at least three example projects related to GPIO or Wi-Fi.

- **Reading:**

- ESP-IDF Getting Started Guide.
- ESP32-S3 Datasheet.

# Key Takeaways from Week 1

- Understand the basics of IoT and the role of ESP32-S3 in IoT applications.
- Set up the ESP-IDF development environment and create your first project.
- Learn how to use the serial monitor for debugging and logging.