Minor in AI

TinyML

Hardware and Wokwi

May 23, 2025

1 Introduction

TinyML involves deploying machine learning models on small, power-efficient devices. This is made possible by combining the right hardware — microcontrollers, sensors, and memory — with simulation tools like Wokwi. This note introduces you to the physical building blocks of a TinyML system and how to practically prototype with them.

2 Microcontrollers and Processors

Microcontrollers (MCUs) are the backbone of any embedded TinyML system. They are low-power, small-sized computers with essential components built in.

1.1 MCUs (Microcontrollers)

MCUs integrate a CPU, memory, and peripherals into one chip. They are cost-effective and ideal for running lightweight ML models on edge devices. Examples include:

- **Arduino Nano 33 BLE Sense**
- **ESP32**
- **STM32**

1.2 CPUs (Central Processing Units)

CPUs handle the main logic of the program, executing instructions sequentially. In MCUs, they are usually simple and energy-efficient.

1.3 NPUs (Neural Processing Units)

NPUs are specialized for running ML workloads like matrix multiplications efficiently. New MCUs may include NPUs to accelerate inference, reducing computation time and energy.

3 Memory Components

The performance of TinyML models depends heavily on how efficiently they use memory.

2.1 RAM (Random Access Memory)

RAM is volatile memory used during model inference — for temporary variables, sensor data, and intermediate calculations. Most MCUs used in TinyML have 32KB to 512KB RAM.

2.2 Flash Memory

Flash memory stores firmware and pre-trained model weights. It's non-volatile and usually ranges from 256KB to a few MBs on MCUs.

2.3 Clock Speed

The clock speed (e.g., 16 MHz, 80 MHz) determines how fast instructions are executed. While higher speeds offer faster inference, they also consume more power — a trade-off in TinyML design.

4 Sensors: Capturing the Real World

Sensors are the eyes and ears of a TinyML system. They collect data from the environment that can be analyzed or fed into ML models.

3.1 Types of Sensors

- **Temperature**: LM35, DHT11 – measure ambient temperature.
- **Pressure**: BMP180 – used in weather forecasting.
- **Light**: LDR – used in smart lighting systems.
- **Inertial**: IMU (Accelerometer + Gyroscope) – detect motion and orientation.
- **Gas**: MQ135 – detect air quality or smoke.
- **Image**: Camera modules (e.g., OV7670) – used for vision-based applications.

3.2 Active vs Passive Sensors

Active sensors need external power to operate and generate signals actively (e.g., ultrasonic distance sensor). **Passive sensors** detect and respond to environmental input without external energy (e.g., thermocouple, LDR).

3.3 Analog vs Digital Sensors

- **Analog sensors** output continuous values — e.g., voltage proportional to temperature.
- **Digital sensors** give structured binary output — e.g., a DHT11 sends temperature data over a digital pin.

5 Sensor Data Visualization (Conceptual)

To understand how sensors behave, we can visualize their output over time.

Example: LDR Sensor Output Graph

If we plot light intensity (y-axis) vs. time (x-axis), we see how an LDR's voltage changes with ambient light. This helps understand patterns in sensor behavior.

6 Wokwi Simulator

Wokwi is an online simulator that allows you to build, test, and run embedded system projects without physical hardware. It supports a wide range of microcontrollers and sensors.

Why Wokwi?

- Eliminates hardware dependency during early development.
- Quick debugging and visualization of sensor data.
- Easy to share projects with peers.
- Built-in support for Arduino, ESP32, Raspberry Pi Pico.

Features

- Drag-and-drop hardware wiring.
- Sensor simulation (light, temperature, motion, buttons).
- Arduino code integration and real-time serial monitor.

7 Hands-on Simulation: Using Wokwi

You can build your own simulated TinyML pipeline using Wokwi and some basic code. Here's how:

Step-by-step Simulation

1. Visit: `https://wokwi.com`
2. Create a new Arduino project.
3. Add a sensor (like a potentiometer or LDR) and connect it to analog pin A0.
4. Paste the code below:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(500);
}
```

Observe: Adjust the sensor dial and monitor the value changes in the Serial Monitor. This demonstrates real-time analog data collection.

Key Takeaways

1. TinyML hardware involves choosing the right processor (MCU), managing limited memory, and integrating sensors for data input.

2. Understanding sensor types (active/passive, analog/digital) is essential for designing effective TinyML applications.

3. Wokwi is a great simulation tool for prototyping embedded ML solutions without real hardware.

4. Practical simulations using code and virtual sensors help bridge the gap between theory and real-world application.