

# Lab1

## 1. Build log of a simple project

The screenshot shows the PlatformIO IDE interface. The Explorer sidebar on the left displays the project structure for 'HELLO\_XIAO'. The main editor window shows the 'main.cpp' file with the following code:

```
src > main.cpp > loop()
1  /* hello_xiao   A first sketch for the Seeed Studio XIAO ESP32C3 in PlatformIO*
2
3  *This example code is in the public domain.*
4
5  *Michel Deslierres June 15, 2020 */
6
7 #include <Arduino.h> // needed in PlatformIO*
8
9 void setup() {
10 | Serial.begin(115200);
11 }
12
13 void loop() {
14 | Serial.println("Hello XIAO!");
15 | delay(2000);
16 }
```

The Terminal tab at the bottom shows the build process:

```
CONFIGURATION: https://docs.platformio.org/page/boards/espressif32/seeed_xiao_esp32c3.html
PLATFORM: Espressif 32 (6.12.0) > Seeed Studio XIAO ESP32C3
HARDWARE: ESP32C3 160MHz, 320KB RAM, 4MB Flash
DEBUG: Current (cmsis-dap) External (cmsis-dap, esp-bridge, esp-built-in, esp-prog, iot-bus-jtag, jlink, minimodule, olimex-arm-usb-ocd, olimex-arm-usb-ocd-h, olimex-arm-usb-tiny-h, olimex-jtag-tin-y, tumpa)
PACKAGES:
- framework-arduinoespressif32 @ 3.20017_241212+sha.dcc1105b
- tool-esptoolpy @ 2.40900_250804 (4.9.0)
- toolchain-riscv32-esp @ 8.4.0+2021r2-patch5
LDF: Library Dependency Finder => https://bit.ly/configure-pio-ldf
LDF Modes: Finder ~ chain, Compatibility ~ soft
Found 34 compatible libraries
Scanning dependencies...
Dependency Graph
|-- Bounce2 @ 2.72.0
Building in release mode
Compiling .pio/build/seeded_xiao_esp32c3/src/main.cpp.o
Linking .pio/build/seeded_xiao_esp32c3/firmware.elf
Retrieving maximum program size .pio/build/seeded_xiao_esp32c3/firmware.elf
Checking size .pio/build/seeded_xiao_esp32c3/firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [ ] 4.2% (used 13636 bytes from 327680 bytes)
Flash: [ == ] 17.9% (used 234146 bytes from 1310720 bytes)
Building .pio/build/seeded_xiao_esp32c3/firmware.bin
esptool.py v4.9.0
Creating esp32c3 image...
Merged 2 ELF sections
Successfully created esp32c3 image.
===== [SUCCESS] Took 1.23 seconds =====
* Terminal will be reused by tasks, press any key to close it.
```

## 2. Serial monitor output

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** pane on the left, showing project files: HELLO\_XIAO, .pio/build, seed\_xiao\_esp32c3, project.checksum, .vscode, c\_cpp\_properties.json, extensions.json, launch.json, include, README, lib, README, src, CMakeLists.txt, main.cpp (selected), test, .gitignore, CMakeLists.txt, platformio.ini, and sdkconfig.seed\_xiao\_esp32c3.
- EDITOR** pane at the top right, displaying the main.cpp file content:

```
#include <Arduino.h> // needed in PlatformIO
void setup() {
    Serial.begin(115200);
}
void loop() {
    Serial.println("Hello XIAO!");
    delay(2000);
}
```

- TERMINAL** pane at the bottom right, showing the output of the device monitor task:

```
* Executing task: platformio device monitor
--- Terminal on /dev/cu.usbmodem101 | 115200 8-N-1
--- Available filters and text transformations: colorize, debug, default, direct, esp32_exception_decoder, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+H
Hello XIAO!
Hello XIAO!
Hello XIAO!
Hello XIAO!
Hello XIAO!
Hello XIAO!
```

### 3. Output on LED pressed

```

src > C: main.cpp > loop()
14 void setup() {
24     pinMode(LED_PIN, OUTPUT);
25     digitalWrite(LED_PIN, LOW); // Start with LED off
26
27     // Configure debounced button
28     button.attach(SWITCH_PIN, INPUT_PULLUP); // Use internal pull-up
29     button.interval(50); // Debounce interval in ms
30     button.setPressedState(LOW); // Button is pressed when LOW
31
32     Serial.println("Setup complete. Waiting for button presses...\n");
33 }
34
35 void loop() {
36     // Update button state
37     button.update();
38
39     // Check if button was pressed
40     if (button.pressed()) {
41         // Toggle LED state
42         ledState = !ledState;
43         digitalWrite(LED_PIN, ledState ? HIGH : LOW);
44
45         // Log the state change
46         Serial.print("Button pressed - LED is now: ");
47         Serial.println(ledState ? "ON" : "OFF");
48     }
49 }

```

PROBLEMS    OUTPUT    TERMINAL    PORTS

> < TERMINAL

- \* Executing task: platformio device monitor

```

--- Terminal on /dev/cu.usbmodem101 | 115200 8-N-1
--- Available filters and text transformations: colorize, debug, default, direct, esp32_exception_decoder, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H

--- LED Switch Control Initialized ---
XIAO ESP32C3 - Tactile Switch with LED Control
=====

Setup complete. Waiting for button presses...

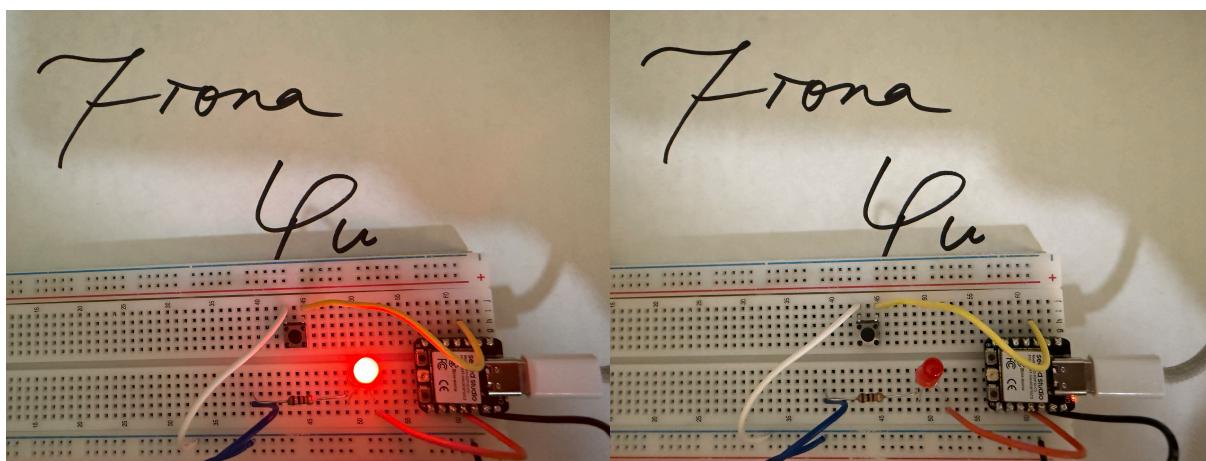
Button pressed - LED is now: ON
Button pressed - LED is now: OFF

```

... | ... X

Build T... ✓  Upload... ✓  Monitor...

#### 4. Photo of LED pressed



#### 5. Github Repo

Name	Last commit message	Last commit date
 ..		
 hello_xiao	Add files via upload	1 minute ago
 Lab1_report_with_photos.pdf	Add files via upload	1 minute ago