



How do I complete the Lab tasks in Microcontroller Lab? Explain step by step.

Required Equipment

Microcontroller board (Arduino Uno / STM32 Nucleo / ESP32 / PIC board)

USB cable & programmer/debugger (if needed)

Breadboard & jumper wires

LED (1–2 pcs), push button (1 pc)

Resistors: 220 Ω (for LED), 10 k Ω (for button pull-down)

Computer with IDE installed (Arduino IDE / STM32CubeIDE / MPLAB X depending on MCU)

Pre-lab Preparation

Read about the differences between microcontrollers and microprocessors.

Review microcontroller architecture: CPU, Flash memory, RAM, I/O ports.'

Install the required IDE and drivers for your platform.

Lab Activities

Part A: Toolchain Setup (30 min)

Connect the microcontroller board to your computer.

Open the IDE, select the correct board and COM port.

Upload a sample "Hello World" program (LED blink).

Example (Arduino C code):

```
void setup() {  
  pinMode(13, OUTPUT); // Set pin 13 as output  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // LED ON  
  delay(1000); // Wait 1 sec  
  digitalWrite(13, LOW); // LED OFF  
  delay(1000); // Wait 1 sec  
}
```

Part B: Digital Input (45 min)

Wire a push button to pin 2 with a pull-down resistor.

Modify the code to read the button state.

Print button status to the Serial Monitor.

Example snippet:

```
int buttonPin = 2;  
  
void setup() {  
  pinMode(buttonPin, INPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  int state = digitalRead(buttonPin);  
  Serial.println(state);  
  delay(200);  
}
```

Part C: Input → Output Control (1 hr)

Connect LED to pin 8.

Write a program where the LED turns ON only when the button is pressed.

Extend: Press → LED toggles state (like a switch).

Part D: Debugging Practice (30 min)

Use Serial.print() (or debugger breakpoints) to observe variable values.

Introduce deliberate error (wrong pin, missing resistor) → troubleshoot.

Provide the circuit diagram and pin diagram for ESP32-WROOM-DA

Here is the step-by-step guide to completing the Microcontroller Lab tasks, followed by the circuit and pin diagrams for the ESP32-WROOM-DA.

Step-by-Step Lab Tasks

Required Equipment

- Microcontroller board (ESP32-WROOM-DA in this case)
- USB cable & programmer/debugger (if needed)
- Breadboard & jumper wires
- LEDs (1–2 pcs), push button (1 pc)
- Resistors: 220 Ω for LED, 10 k Ω for button pull-down
- Computer with IDE installed (Arduino IDE for ESP32)

Pre-lab Preparation

1. Read about differences between microcontrollers and microprocessors.
2. Review microcontroller architecture: CPU, Flash memory, RAM, I/O ports.
3. Install Arduino IDE and ESP32 board drivers on your computer.

Part A: Toolchain Setup (30 min)

1. Connect the ESP32-WROOM-DA board to your computer via USB.
2. Open Arduino IDE, select "ESP32 Dev Module" and correct COM port.
3. Upload a sample "Hello World" program (LED blink).

Example Arduino code for LED blink on ESP32:

```
void setup() {  
  pinMode(2, OUTPUT);    // GPIO2 usually connected to built-in LED on ESP32  
}
```

```
void loop() {
  digitalWrite(2, HIGH); // LED ON
  delay(1000);           // Wait 1 sec
  digitalWrite(2, LOW);  // LED OFF
  delay(1000);           // Wait 1 sec
}
```

Part B: Digital Input (45 min)

1. Wire a push button to GPIO pin 0, with a 10 kΩ pull-down resistor to GND.
2. Modify code to read the button state and print it to Serial Monitor.

Example code snippet:

```
int buttonPin = 0;
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  int state = digitalRead(buttonPin);
  Serial.println(state);
  delay(200);
}
```

Part C: Input → Output Control (1 hr)

1. Connect an LED to GPIO pin 4 with 220 Ω resistor in series to GND.
2. Write a program where the LED turns ON only when the button is pressed.
3. Extend program so pressing the button toggles LED state like a switch.

Part D: Debugging Practice (30 min)

1. Use `Serial.print()` statements or debugger breakpoints to observe variables.
2. Introduce deliberate faults (e.g., incorrect pin wiring or missing resistor).
3. Troubleshoot and fix the issues.

ESP32-WROOM-DA Pin Diagram and Circuit Information

- The ESP32-WROOM-DA has 41 pins including 24 GPIOs.
- Key pins used for this lab are:
 - GPIO0: Push button input
 - GPIO2: Built-in LED or external LED output
 - GPIO4: External LED output
 - GND: Ground reference
 - 3V3: Power supply (3.3V)

- The module includes two PCB antennas.
- See the attached pin layout diagram from the ESP32-WROOM-DA datasheet below for exact pin positions.

Basic Circuit Diagram for Lab Tasks on ESP32-WROOM-DA

- Connect one leg of push button to GPIO0 and the other leg to GND through 10 kΩ resistor (pull-down).
- Connect LED anode (+) to GPIO4 via 220 Ω resistor, LED cathode to GND.
- USB powers the ESP32 board and allows programming via the Arduino IDE.

This setup will enable performing the lab steps effectively using the ESP32-WROOM-DA board.

[\[1\]](#) [\[2\]](#) [\[3\]](#)

If needed, more detailed hardware schematics or code examples can be provided.



1. <https://techtionions.com/esp32-pinout/>
2. <https://esp32io.com/tutorials/esp32-button-led>
3. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-da_datasheet_en.pdf
4. <https://forum.arduino.cc/t/esp-32-wroom-32-pin-allocation/1228704>
5. https://docs.sunfounder.com/projects/umsk/en/latest/03_esp32/esp32_lesson01_button.html
6. https://www.youtube.com/watch?v=yxU_Kw2de08
7. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
8. <https://soslab.net/articles/esp32-tutorial-how-to-control-an-led-with-a-button-toggle-debounce>
9. <https://oshwlab.com/Zulfahmi27/esp32-devkit-v1>
10. https://cdn-shop.adafruit.com/product-files/5344/5344_esp32-wroom-da_datasheet_en.pdf
11. <https://randomnerdtutorials.com/esp32-pcb-weather-station-interface-circuit-diagram-circuit/>
12. <https://www.youtube.com/watch?v=QUNKY87Da7A>
13. <https://circuitdigest.com/esp32-projects>
14. <https://circuitdigest.com/microcontroller-projects/getting-started-with-esp32-with-arduino-ide>
15. <https://randomnerdtutorials.com/getting-started-with-esp32/>
16. <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>
17. <https://techexplorations.com/guides/esp32/begin/digitalout>
18. <https://www.scribd.com/document/697908789/esp-32-board-schematic>