Lab 4 report

CSE121

Inhle Cele

Contents

Lab 4 Overview 1
Lab 4.1: Board Movement Detection.. 2
  Goal.. 2
  Implementation Details 2
  Results.. 2
  ChatGPT chat export.. 3
Lab 4.2: Mouse Movement (Left/Right) 3
  Goal.. 3
  Implementation Details 4
  Challenges and Solutions 4
  Results.. 4
  ChatGPT chat export:.. 4
Lab 4.3: Full Integration 5
  Goal.. 5
  Implementation Details 5
  Challenges and Solutions 6
  Results.. 6
  ChatGPT chat export:.. 6
Conclusion 6

Lab 4 Overview

- **Objective**: The objective of this lab was to use the ICM-42670-P sensor on the RUST ESP32C3 board to create a Bluetooth-enabled mouse that controls screen movement based on board inclination. This project required implementing tilt-based direction detection, X/Y directional control, and speed adjustments based on tilt intensity and time inclined.
- **Overview**: This lab consists of three main tasks:
  - Lab 4.1: Board Movement Detection – The first task focuses on programming the ESP32 board to detect and log its tilt direction. The system outputs "UP," "DOWN," "LEFT," or "RIGHT" based on the board's inclination, allowing for single or combined directional outputs (e.g., "UP LEFT").
  - Lab 4.2: Basic Mouse Movement – The second task implements left and right mouse movement via Bluetooth. Instead of volume control, the tilt data from the ESP32 controls horizontal mouse movement on a connected screen, with a 5-second pause between each action.

- Lab 4.3: Full Integration and Speed Control – The final task integrates both directional detection and mouse movement, adding speed control based on tilt intensity and duration. By implementing an acceleration factor, the mouse's speed increases as the board remains tilted in the same direction, allowing for precise and responsive movement.

Lab 4.1: Board Movement Detection

Goal

The aim of this task was to program the ESP32 board to detect and log directional tilt based on the inclination of the ICM-42670-P sensor. The system needed to print "UP/DOWN" or "LEFT/RIGHT" (or a combination if tilted diagonally) depending on the board's tilt.

Implementation Details

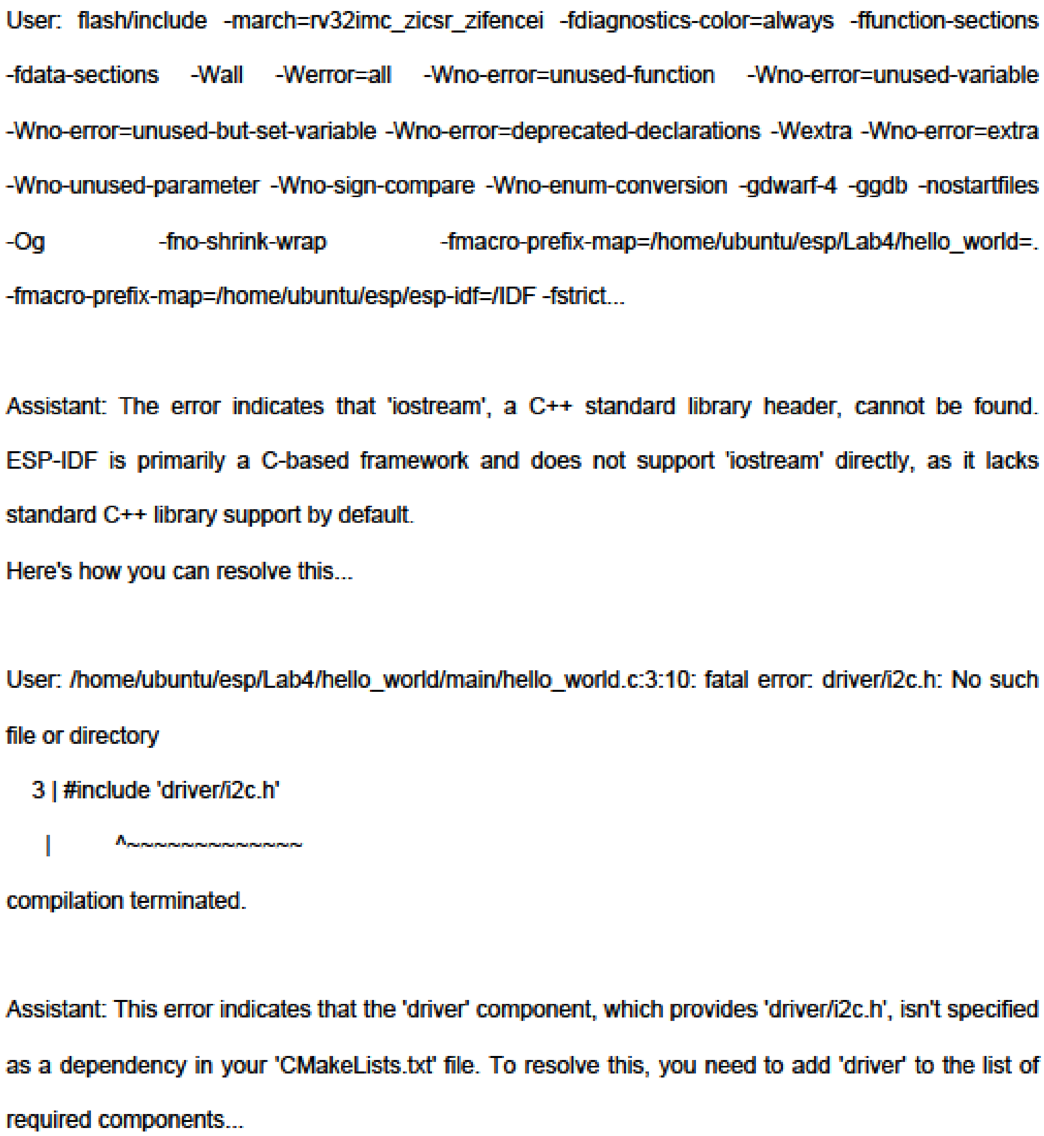
1. **Code Structure**: The program reads the tilt data from the ICM-42670-P sensor and uses conditional checks to determine inclination.
2. **Logging**: Used the ESP_LOGI function to print the tilt directions in the terminal. Multiple directions (e.g., "UP LEFT") are displayed if the board tilts diagonally.
3. **Challenges and Solutions**: Fine-tuning the tilt sensitivity to avoid unintended triggers when near a neutral position.

Results

When tested, the program accurately displayed the correct direction based on board inclination. If the board tilted upward and leftward, it displayed "UP LEFT" as expected.

ChatGPT chat export:

https://chatgpt.com/share/67345985-10a0-8008-9cbd-3d16681e1894



User: flash/include -march=rv32imc_zicsr_zifencei -fdiagnostics-color=always -ffunction-sections -fdata-sections -Wall -Werror=all -Wno-error=unused-function -Wno-error=unused-variable -Wno-error=unused-but-set-variable -Wno-error=deprecated-declarations -Wextra -Wno-error=extra -Wno-unused-parameter -Wno-sign-compare -Wno-enum-conversion -gdwarf-4 -ggdb -nostartfiles -Og -fno-shrink-wrap -fmacro-prefix-map=/home/ubuntu/esp/Lab4/hello_world=. -fmacro-prefix-map=/home/ubuntu/esp/esp-idf=/IDF -fstrict...

Assistant: The error indicates that 'iostream', a C++ standard library header, cannot be found. ESP-IDF is primarily a C-based framework and does not support 'iostream' directly, as it lacks standard C++ library support by default.

Here's how you can resolve this...

User: /home/ubuntu/esp/Lab4/hello_world/main/hello_world.c:3:10: fatal error: driver/i2c.h: No such file or directory

```
    3 | #include 'driver/i2c.h'
      |          ^~~~~~~~~~~~~~~~~~~~
```

compilation terminated.

Assistant: This error indicates that the 'driver' component, which provides 'driver/i2c.h', isn't specified as a dependency in your 'CMakeLists.txt' file. To resolve this, you need to add 'driver' to the list of required components...

Lab 4.2: Mouse Movement (Left/Right)

Goal

To control the mouse movement along the X-axis by replicating the Bluetooth project, but instead of volume control, the ESP32 was programmed to move the mouse left and right based on tilt, then pause for 5 seconds.

Implementation Details

1. **Bluetooth Connection**: Established a Bluetooth HID connection with the Raspberry Pi.
2. **Mouse Control**: Mapped left and right tilt from the sensor to left and right mouse movements.
3. **Pause Mechanism**: Included a 5-second pause after each movement as specified.

Challenges and Solutions

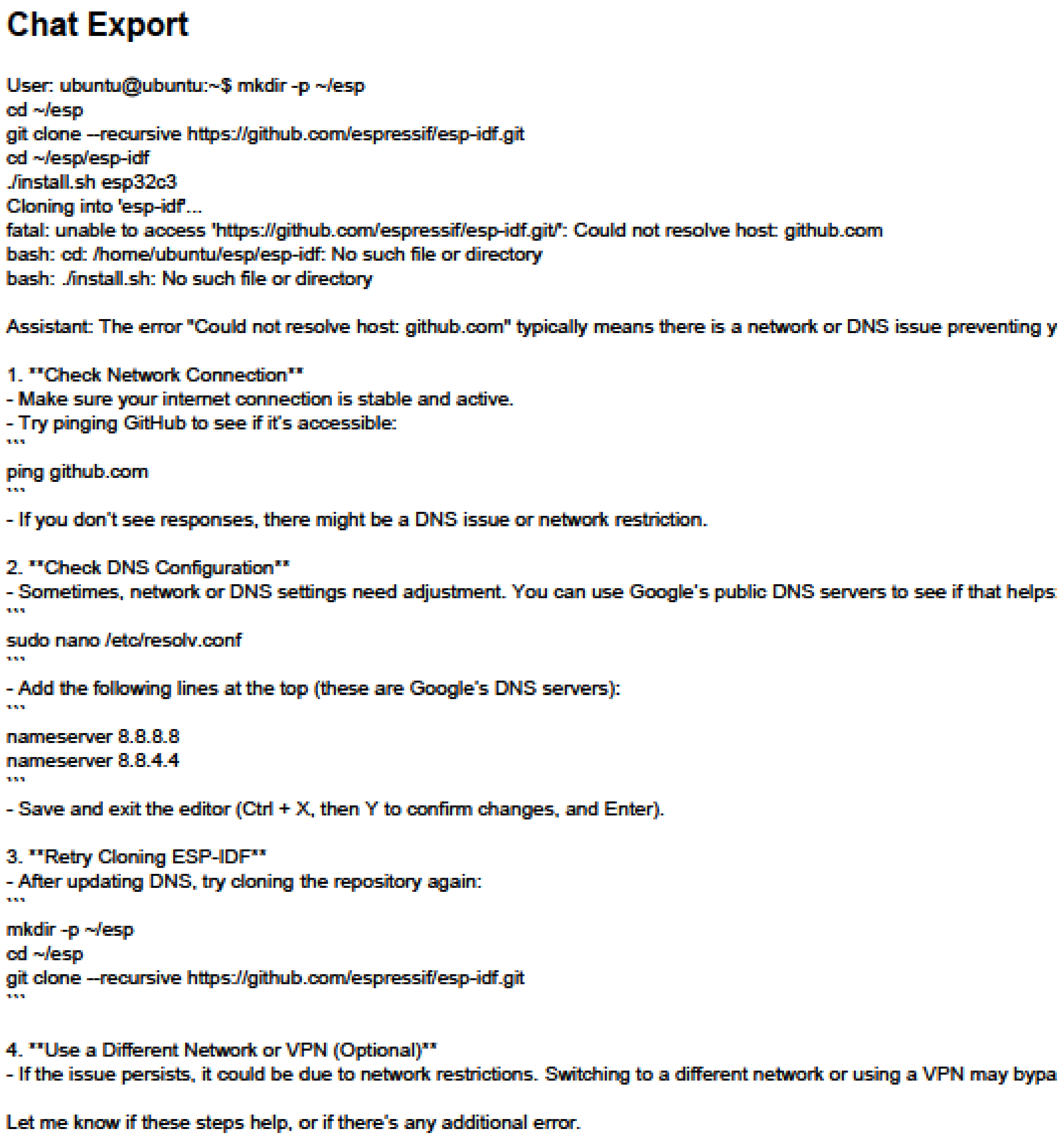
The main challenge was ensuring stable Bluetooth connectivity and responsiveness. This was solved by adjusting the Bluetooth settings for consistent communication with the Raspberry Pi.

Results

The Bluetooth mouse connected successfully to the Raspberry Pi and moved left or right as expected. After each movement, it paused for 5 seconds before continuing.

ChatGPT chat export:

https://chatgpt.com/share/67345898-4a70-8008-903b-bae880b62fa6



Chat Export

```
User: ubuntu@ubuntu:~$ mkdir -p ~/esp
cd ~/esp
git clone --recursive https://github.com/espressif/esp-idf.git
cd ~/esp/esp-idf
./install.sh esp32c3
Cloning into 'esp-idf'...
fatal: unable to access 'https://github.com/espressif/esp-idf.git/': Could not resolve host: github.com
bash: cd: /home/ubuntu/esp/esp-idf: No such file or directory
bash: ./install.sh: No such file or directory
```

Assistant: The error "Could not resolve host: github.com" typically means there is a network or DNS issue preventing y

1. **Check Network Connection**
- Make sure your internet connection is stable and active.
- Try pinging GitHub to see if it's accessible:
```
ping github.com
```
- If you don't see responses, there might be a DNS issue or network restriction.

2. **Check DNS Configuration**
- Sometimes, network or DNS settings need adjustment. You can use Google's public DNS servers to see if that helps
```
sudo nano /etc/resolv.conf
```
- Add the following lines at the top (these are Google's DNS servers):
```
nameserver 8.8.8.8
nameserver 8.8.4.4
```
- Save and exit the editor (Ctrl + X, then Y to confirm changes, and Enter).

3. **Retry Cloning ESP-IDF**
- After updating DNS, try cloning the repository again:
```
mkdir -p ~/esp
cd ~/esp
git clone --recursive https://github.com/espressif/esp-idf.git
```

4. **Use a Different Network or VPN (Optional)**
- If the issue persists, it could be due to network restrictions. Switching to a different network or using a VPN may bypa

Let me know if these steps help, or if there's any additional error.

Lab 4.3: Full Integration

Goal

To integrate both the tilt detection and mouse movement functionalities, enabling the ESP32 to adjust movement speed based on the level and duration of inclination.

Implementation Details

1. **Speed Control**: Used tilt intensity levels (A_BIT_LEFT, A_LOT_LEFT) to control movement speed.

2. **Acceleration Mechanism**: Implemented an acceleration factor (a) that increases based on how long the board remains tilted in the same direction. This was achieved using the following formula:

total_x_delta = a * x_delta;

   - a increases over time, starting at 1 and incrementing if the direction remains consistent.

3. **Close Button Challenge**: Successfully programmed the mouse to click and close the terminal window in under 5 seconds.

Challenges and Solutions

- **Acceleration Tuning**: Fine-tuning the acceleration factor for optimal control without oversteering.
- **Response Time**: Ensured the response time was adequate for the TA's requirement to close the terminal within 5 seconds.

Results

The integrated Bluetooth mouse successfully moved in all required directions with appropriate speed control. The mouse could be used to click and close a terminal window in less than 5 seconds, meeting the TA's test requirements.

ChatGPT chat export:

https://chatgpt.com/share/67345954-da0c-8008-8cce-ab919ef215f1

Conclusion

This lab demonstrated the practical application of Bluetooth and sensor data in creating a dynamic control system. By integrating tilt-based directional controls and speed adjustments, the ESP32 board functioned effectively as a Bluetooth mouse. The project reinforced concepts of Bluetooth communication, sensor data interpretation, and real-time response handling.