

Lab 4

Part 1:

Part 1 was just moving the esp32c3 and using the gyro sensor on the board, and the terminal detecting what direction it was moving in.

Part 2:

Part 2 was connecting the esp32c3 to my laptop using bluetooth and allowing it to move my mouse cursor left and right. It required me to use the ble_hidd_demo folder from the esp-idf examples and change the hit task function to do what I wanted it to do, which in this case, was moving the mouse value plus or minus 30.

Part 3:

Part 3 combined both parts stated above, and used the gyro sensor on the board to move my mouse in whatever x/y direction the esp was moving in. More or less the same thing as parts 1 and 2, just combining everything.

Resources:

ChatGPT

chat.openai.com

<https://docs.espressif.com/projects/esp-matter/en/latest/esp32/faq.html>

APACHE license:

<https://drive.google.com/drive/u/1/folders/15uqJ8zO42qNE-cszcmcdxg0Aig6x94aC>

Other:

https://docs.espressif.com/projects/esp-idf/en/v5.1/esp32/api-reference/bluetooth/esp_hidd.html

ChatGPT Code:

```
c Copy code

#include <stdio.h>
#include <driver/i2c.h>
#include "esp_log.h"
#include "mpu6050.h" // Include the MPU6050 driver header

#define I2C_MASTER_SCL_IO 22 // GPIO number for I2C master clock
#define I2C_MASTER_SDA_IO 21 // GPIO number for I2C master data
#define I2C_MASTER_NUM I2C_NUM_0 // I2C port number for master
#define I2C_MASTER_FREQ_HZ 100000 // I2C master clock frequency

#define MPU6050_ADDR 0x68 // MPU6050 I2C address

static const char *TAG = "MPU6050";

void app_main(void) {
    // Initialize I2C
    i2c_config_t conf = {
        .mode = I2C_MODE_MASTER,
        .sda_io_num = I2C_MASTER_SDA_IO,
        .scl_io_num = I2C_MASTER_SCL_IO,
        .sda_pullup_en = GPIO_PULLUP_ENABLE,
        .scl_pullup_en = GPIO_PULLUP_ENABLE,
        .master.clk_speed = I2C_MASTER_FREQ_HZ,
    };
    i2c_param_config(I2C_MASTER_NUM, &conf);
    i2c_driver_install(I2C_MASTER_NUM, conf.mode, 0, 0, 0);

    // Initialize MPU6050
    mpu6050_t mpu6050 = {
        .port = I2C_MASTER_NUM,
        .addr = MPU6050_ADDR,
        .sda_io_num = I2C_MASTER_SDA_IO,
        .scl_io_num = I2C_MASTER_SCL_IO,
    };
    mpu6050_init(&mpu6050);

    while (1) {
        mpu6050_acceleration_t accel;
        mpu6050_read_acceleration(&mpu6050, &accel);


        // Determine the direction
        char *direction = "";
        if (accel.x > threshold) direction = "RIGHT";
        else if (accel.x < -threshold) direction = "LEFT";

        if (accel.y > threshold) direction = (strlen(direction) > 0) ? "UP RIGHT" :
        else if (accel.y < -threshold) direction = (strlen(direction) > 0) ? "DOWN L

        ESP_LOGI(TAG, "Direction: %s", direction);

        vTaskDelay(1000 / portTICK_PERIOD_MS); // Delay for a second
    }
}
```


c

 Copy code

```
void read_gyro() {
    int16_t x = (i2c_read(0x0B) << 8) | i2c_read(0x0C);
    int16_t y = (i2c_read(0x0D) << 8) | i2c_read(0x0E);

    if (y > 300 && (x > -300 && x < 300)) {
        ESP_LOGI("", "UP");
    }
    else if (y < -300 && (x > -300 && x < 300)) {
        ESP_LOGI("", "DOWN");
    }
    else if (x > 300 && (y > -300 && y < 300)) {
        ESP_LOGI("", "LEFT");
    }
    else if (x < -300 && (y > -300 && y < 300)) {
        ESP_LOGI("", "RIGHT");
    }
    else if (y > 200 && (x < -300)) {
        ESP_LOGI("", "UP RIGHT");
    }
    else if (y < -200 && (x < -300)) {
        ESP_LOGI("", "DOWN RIGHT");
    }
    else if (y > 200 && (x > 300)) {
        ESP_LOGI("", "UP LEFT");
    }
    else if (y < -200 && (x > 300)) {
        ESP_LOGI("", "DOWN LEFT");
    }
}
```

c

 Copy code

```
void hid_demo_task(void *pvParameters)
{
    vTaskDelay(1000 / portTICK_PERIOD_MS);
    while(1) {
        vTaskDelay(2000 / portTICK_PERIOD_MS);
        if (sec_conn) {
            ESP_LOGI(HID_DEMO_TAG, "Move mouse");
            esp_hidd_send_mouse_value(hid_conn_id, 0, 30, 0);
            vTaskDelay(5000 / portTICK_PERIOD_MS);
            esp_hidd_send_mouse_value(hid_conn_id, 0, -30, 0);
        }
    }
}
```