Project Report: Gesture-Based Device Controller

Name: Cirus Chakma

SID: 200495194

Course: ENEL 351

Date: 11 April, 2025

1. Introduction

This report outlines the the gesture-based device controller meant for home appliances and other usages. The main goal was to allow users to control their devices without pressing buttons, but simply by using natural hand gestures. It also includes manual controls for situations when you want to override gesture mode with failsafe buttons. The system is designed to keep the command active until it recognizes a new gesture, which is a great feature for both increased usability and reliability.

2. System Overview

The created system has several critical components. The Grove Gesture sensor PAJ7620U2 is for gesture detection due to its usable I²C connection functionality. It is connected via pins PB8 (SCL) and PB9 (SDA). To make it easier for users to select between active and inactive modes for gesture-processing, we've included a Sparkfun Breakout Capacitive Touch sensor AT42QT101, which sends its output signal to pin PA0 and starts the process of detection. In addition, to assure the system's reliability, we've installed backup push buttons on pins PA1 and PA4. This enables for immediate manual overriding of gesture commands if needed.

Control of the output is handled via GPIO pins PC10 and PC11, which correspond to two relay signals that control devices such as fans and lights through a 5V dual-channel relay. An LED indicator (LD2) provides a clear visual of the operational status, as it lights up to indicate the gesture detection process is active. Managing voltage was essential, as the logic circuits operated safely at 3.3V from the STM32F103RB Nucleo-64 board, while the relay modules and the connected appliances required a reliable 5V supply, which is also provided by the board itself.

3. Components Used

- Microcontroller: STM32F103RB Nucleo-64 board
- Gesture Sensor: Grove Gesture Sensor PAJ7620U2 (Purchased from DigiKey for $16.51)
- Touch Sensor: Sparkfun Breakout Capacitive Touch AT42QT101 (Purchased from DigiKey for $9.88)
- Relay Module: 5V Dual-Channel Relay (Purchased from Amazon for $6)

4. Design Strategy

The initial concept was all about practically applying an IR sensor for real-life day-to-day usages and is cheaper and more convenient. As we went deeper into our lab work and lessons, the design began to change, incorporating both the touch and IR sensors to work together, utilizing two different input sensors. Our key objective was to use these sensors to control two output devices, making it convenient to turn them on and off whenever needed.

Two manual push buttons were then added for improved reliability by overriding sensor functionality when gesture detection failed. These two buttons operate separately, turning on and off the output devices regardless of the gesture sensor status. As a result, the idea evolved into a comprehensive gesture-based system in which the user may activate and deactivate two devices using a capacitive touch sensor while detecting and analyzing movements using an infrared sensor.

5. Design Process

During the initial design phase, the focus was clearly on outlining the system requirements. Emphasis was put on that gesture commands would remain active until another gesture is detected to change them. Plus, straightforward user interaction was prioritized, utilizing the touch sensor and backup buttons. As for the schematic creation, ORCAD(PSPICE) was used and verified the GPIO pins and I²C communication channels to avoid any conflicts or signal interference.

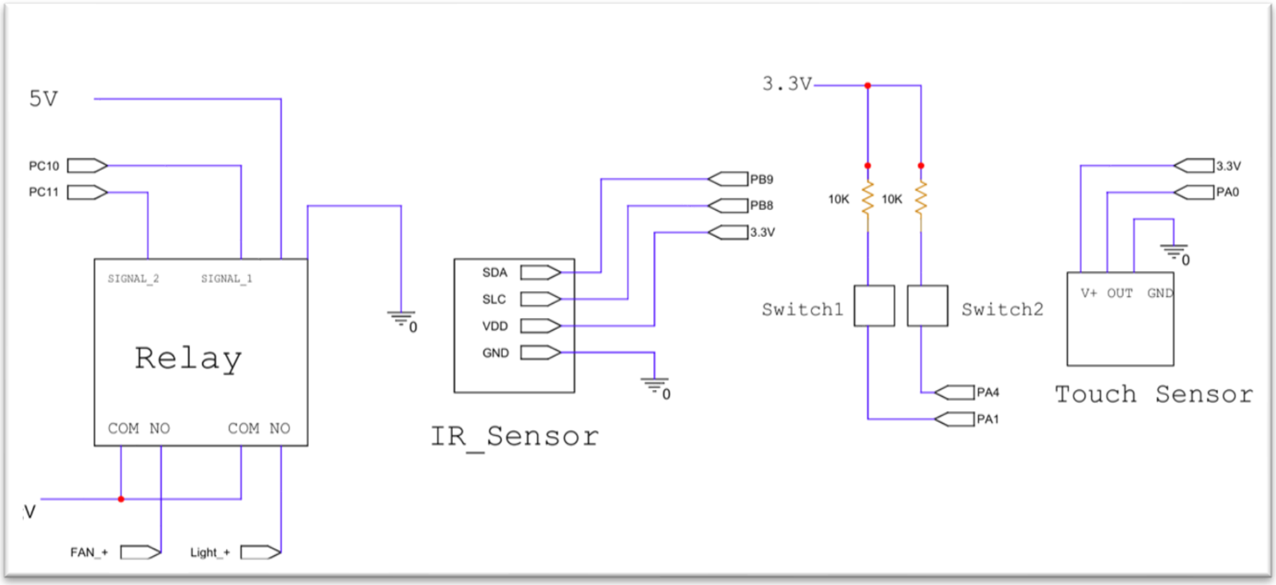

Figure 1: schematic

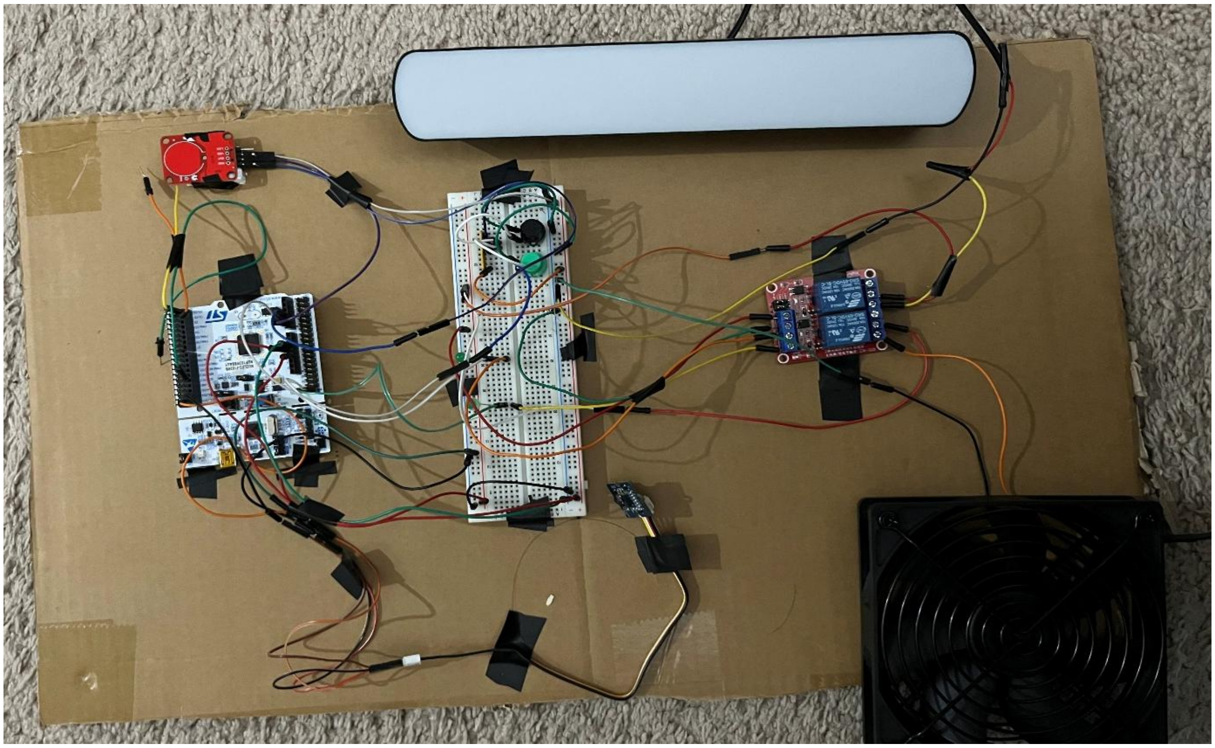

Figure 2 Top View of Project

For developing the codes, precise coding of gesture detection using the Grove Gesture PAJ7620U2 sensor was done. The code initializations were mainly taken from Waveshare and modified afterward to better fit the project. Out of all the gesture and object detection feature codes, only up-down and left-right initializations were used. Capacitive touch inputs were used from the Sparkfun Breakout Capacitive Touch sensor AT42QT101. The software logic was designed to store the most recent valid gesture input and update relay output statuses accordingly. Backup push button functionality was also programmed to temporarily force relay outputs to High and Low regardless of the current gesture state.

Wiring and Connections

The wiring and connection process played a critical role in ensuring the proper functionality and responsiveness of the gesture-based device controller. Each component was connected with careful attention to voltage compatibility, signal integrity, and microcontroller pin mapping using their own datasheets.

- **STM32F103RB Nucleo-64 Board** plays the role of the central controller, supplying both 3.3V for logic-level components and 5V for the relay module using its onboard voltage output ports.

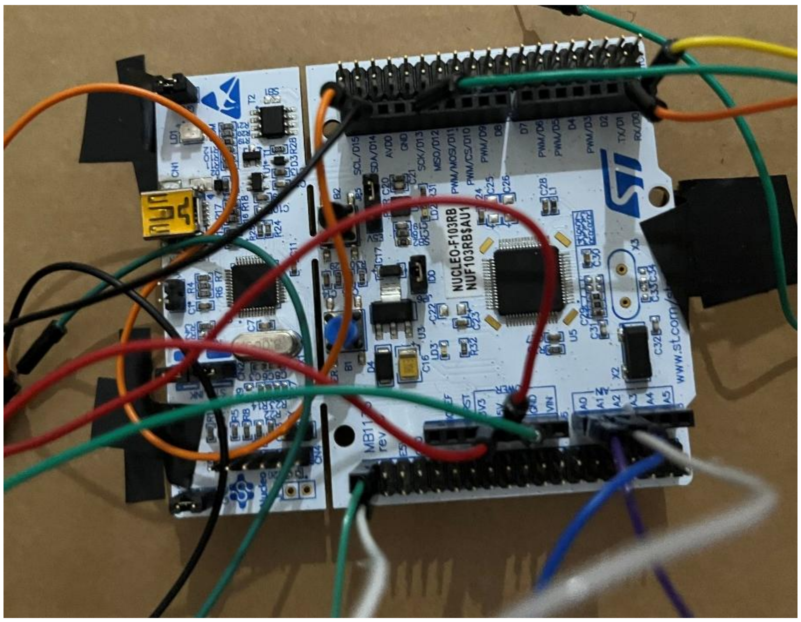

Figure 3:Nucleo Board

- **Grove Gesture Sensor (PAJ7620U2)** was interfaced via I²C2 communication using pins PB8 (SCL) and PB9 (SDA). The sensor was powered with 3.3V and grounded to the board's common GND, enabling real-time gesture recognition features.

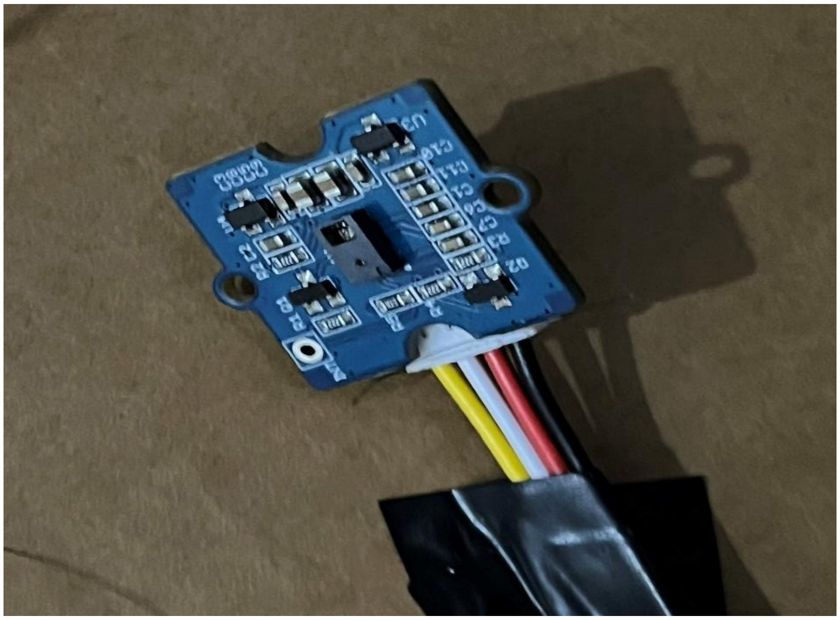

Figure 4: IR sensor

- **SparkFun Capacitive Touch Sensor (AT42QT101)** was connected with its output pin wired to PA0 on the board, while power and ground lines were tied to 3.3V and GND respectively. This sensor toggled the gesture detection mode and required careful soldering to ensure the HIGH signal was consistently received.

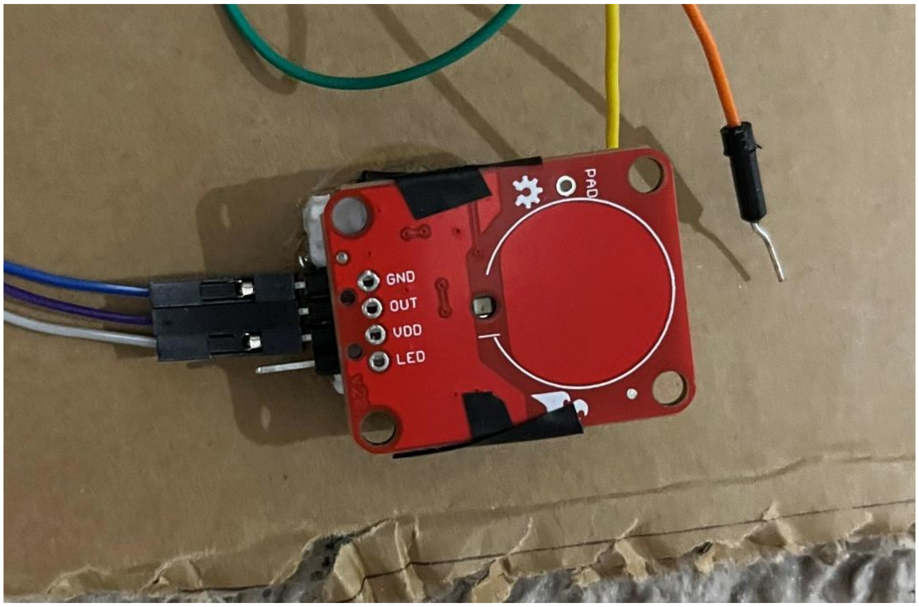

Figure 5: Touch Sensor

- **Manual Push Buttons** were connected to PA1 and PA4. Each button had one leg connected to 3.3V and the other to its respective GPIO pin, configured with internal pull-down resistors in the software. When pressed, they sent a HIGH signal to override the current gesture control.
- **5V Dual-Channel Relay Module** received control signals through PC10 and PC11, corresponding to the IR_UD and IR_LR outputs. It was powered directly from the board’s 5V line, with a shared GND connection. This relay

switched connected loads (fan and light) based on gesture or button inputs.

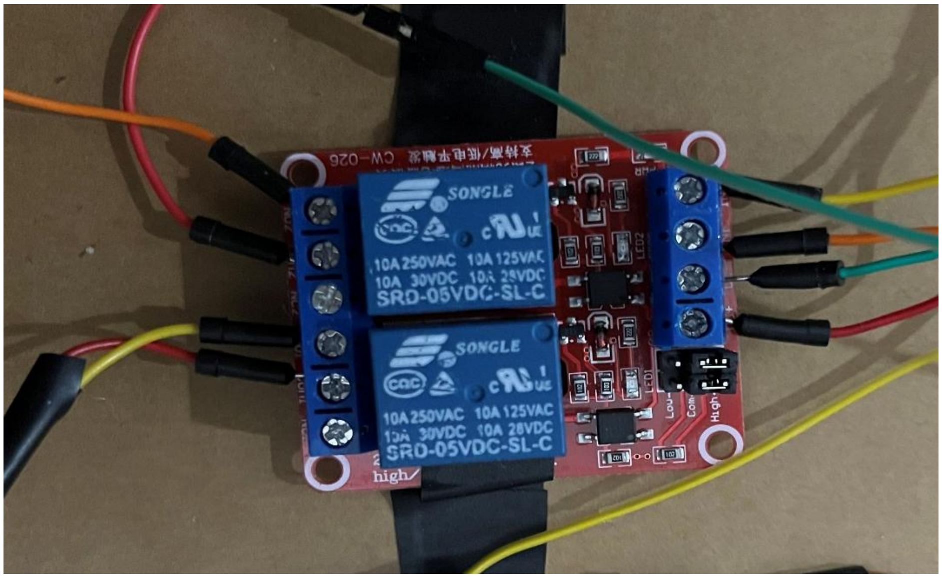

Figure 6: Relay

- **LED Indicator (LD2)** on the Nucleo board was used to indicate the active state of gesture mode and was controlled through output logic in firmware.

6. Testing Strategies and Procedures

The testing phase started with a focus on validating the components one at a time. Individual units, including the Grove Gesture sensor, Sparkfun capacitive touch sensor, relay modules, and push buttons were thoroughly examined, ensuring they met our standards for accuracy, responsiveness, and reliability. Repeated executed gestures—up, down, left, and right swipes were tested—to ensure the sensor's data was accurate via the I²C interface.

Following that, integration testing was performed and ensured that the entire system would work properly. Tests explicitly targeted the persistent command capability, executing a gesture, waiting for confirmation of command retention, and then overriding orders using backup buttons to assure immediate system reaction. Final validation included real-world load circumstances, such as desktop cooler fan and low voltage LED, to ensure correct relay performance and system stability.

Testing Process Challenges:

The testing process encountered a few inconsistencies mainly due to complexities and lack of explanation found in IR sensor datasheets. Initially, understanding how the gesture sensor specifically identified gestures such as up, down, left, and right was challenging, as the sensor supported additional complex gestures and object detection functions that were not clearly detailed explained in provided datasheets. These ambiguities in documentation required time-consuming experimentations. USART debugging techniques were essential, serving as a critical troubleshooting tool to determine if gestures were being correctly recognized by the microcontroller. But, USART was only done for testing purposes and is not included in the main project according to the function specification.

Testing the capacitive touch sensor was initially complicated because of issues with precisely soldering the sensor pins. Improper soldering resulted in inconsistent and unreliable HIGH signal detections, preventing the microcontroller from reliably recognizing rising edges. After addressing soldering concerns with improved procedures, the capacitive sensor produced consistent and dependable signal output, allowing for proper system responses.

Testing the manual push buttons and relay modules were noticeably easier, thanks to the practical experiences picked up from previous lab assignments. Knowing these components well allowed me to swiftly validate and integrate them into the overall system, quickly confirming they were functioning as intended.

7. Deviations from Functional Specifications

Throughout the project, several deviations or adjustments were made from the initially planned functional specifications occurred. The main deviation involved the reassignment of GPIO pin connections. Originally specified I²C communication pins (PB6 and PB7) were changed to PB8 and PB9 due to their clearer labeling and easier access on the STM32F103RB Nucleo board. Similarly, relay input signals were moved to pins PC10 and PC11, again chosen for their ease of identification and accessibility on the microcontroller.

Push button pins that were originally intended for other GPIO pins were reassigned to PA4 and PA1 to simplify wiring and improve circuit readability. The output pin of the capacitive touch sensor, which was absent in function implementation before, was also relocated to PA0, greatly simplifying the overall hardware configuration.

Perhaps the another crucial deviation was deciding to remove the step-up voltage converter we had initially planned. After taking a closer look, we realized that the STM32F103RB Nucleo-64 board offers a solid and consistent 5V output. This discovery allowed us to simplify our design by cutting out unnecessary components in the power management system.

Regardless of the modifications made to the project function specification, the final project design retained the core function of the targeted design.

8. User Operation

Operation guidelines were established for ease of use:

- The system requires connections to 3.3V (logic) and 5V (relay modules).

- Users activate gesture mode via the Sparkfun capacitive touch sensor at PA0, with LED LD2 illuminating to confirm activation.
- Performing swipe gestures in the sensor's field of view alters relay outputs on PC10 (vertical) and PC11 (horizontal).
- Backup override buttons on PA1 and PA4 allow immediate manual control, temporarily setting outputs high irrespective of gesture commands.
- Touching the sensor again exits gesture mode, turning off LD2 and resetting outputs to a default inactive state.

9. Conclusion

The Gesture-Based Device Controller successfully achieved the project's objective, providing reliable and intuitive gesture-based control using day-to-day useful devices. With persistent command states and manual override capabilities, the system ran successfully. The merging of touch toggling and backup overrides ensures enhanced reliance and user-friendly operation. Future improvements could include expanded gesture recognition, voice activation, and wireless control capabilities.