Northwestern University

McCormick School of Engineering
Department of Electrical Engineering and Computer Science

ECE 347 – Microprocessor System Projects
Winter 2021

Computer Input Team
Design Document

Alex Kolar, alexanderkolar2021@u.northwestern.edu

Eugene Choe, eugenechoe2022@u.northwestern.edu

Jean Selep, JeanSelep2021@u.northwestern.edu

Jiayue Bai, JiayueBai2022@u.northwestern.edu

Client: Professor Henschen
Date: March 20th, 2021
Version: TWO

Table of Contents

Executive Summary

Introduction

Currently, there exist many computer applications where traditional mouse and keyboard input is impractical or a hindrance to performance. These hindrances include computer users with limited or no use of one hand; large presentations; computer gaming; and more. Therefore, it is necessary to provide a means of computer input that is intuitive, mobile, and easily usable with one hand.

Several devices exist to address this problem, including specially designed keyboards that may be used with one hand and special virtual reality gloves that may be worn by the user. However, such custom keyboards are often physically bulky and stationary, while most existing virtual reality gloves are targeted specifically for the gaming market and thus provide limited assistance to users or presenters – including wired connections preventing mobility and a limited range of inputs unsuitable to general computer use.

Our design, a wireless glove capable of providing general computer input and fully customizable by the user, will address these issues.

Overview

As described in the specification document in Appendix B, the glove will be a fully wireless input device attached to the user's hand. It includes a camera that looks at an IR (infrared) LED base station, providing real-time position tracking of the user's hand, and will communicate input information to the user's computer. The LED base station will also be a means of charging the glove when not in use. A software program will be running on the user's computer to interpret the raw input data from the glove based on user preferences and provide an interactive GUI for device management.

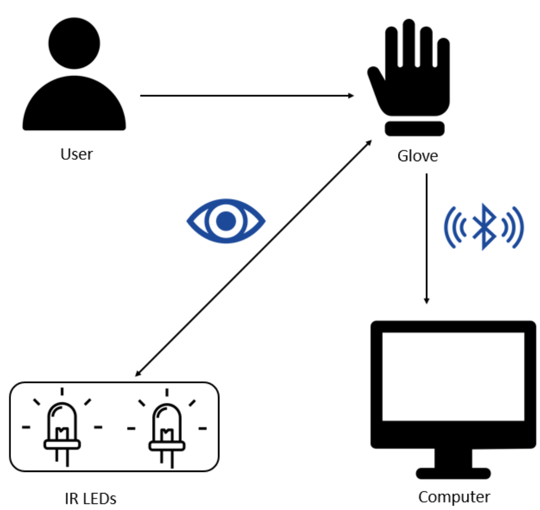


Figure 1: Diagram of the computer input system.

Physical Design

Glove Design

For the physical design of the glove, there were three different designs considered. For every design, however, there were a few constants. One constant was that we would have 2 flex sensors sewn along the index and middle fingers. The second was to have the microprocessor attached to the glove on the back of the hand. To attach the microprocessor, we plan to use adhesive velcro strips; one part adheres to the microprocessor and the other part is sewn into the glove. With this, we will be able to easily disconnect the microprocessor if needed without damaging any of the electrical components of the microprocessor. Another constant in the physical design is sewing in a pouch for the rechargeable battery for the microprocessor. The rechargeable battery must be disconnected from the microprocessor in order to charge, and it would be dangerous for the battery to be swinging around during use. Thus, a small sleeve sewn into the glove is necessary in order to cradle the battery, while also being easy to remove when charging. As a final constant, we will have two buttons sewn to the fingertips of the index and middle fingers.

Figure 2: Placement of components on the glove, top view.

The variation in design is the placement of the IR camera on the glove. The IR camera needs to be able to see two IR LEDs on the base station stand and use the distance between the two to determine the distance of the glove from the IR camera. We came up with two alternatives for the placement of the IR camera.

1. *On top of the wrist*
   With the camera on top of the wrist, the camera will not get in the way if the user wants to rest their hands on the desk when not using the glove. However, while researching actions the user would want to do with the glove, we found that often, the user would have their hand facing the computer's screen. Without some correction to this behavior, it would affect the camera's line of sight to the LEDs and limit accuracy.

2. *On bottom of the wrist*
   The camera on the bottom hand would allow the LEDs to be seen by the IR camera while the user has their palm facing the screen. However, this placement would mean discomfort if the user was resting their hand on the desk while not using the glove as a mouse. While it is uncertain how often this would occur, it would be a point of significant discomfort for a user. For this prototype, we decided that it would be preferable to maximize accuracy of the sensor over potential user comfort when not using the glove. However, in future development, we hope to place the camera on the glove in a way that would be comfortable for the user regardless of how they are using the glove.

Figure 3: Placement of components on the glove, bottom view.

Another choice that we had to evaluate was what type of glove to use for the design. The glove had to be comfortable enough for long-term wear while also being substantial enough to sew on the sensors, not have the stitches pulled out, and protect the user if the sensors get too hot. We considered three options:

1. *Knit Glove*
   Knit gloves are common and come in many different colors and sizes, providing a wide selection of options. Knit gloves are also the cheapest option that is substantial enough to hold all of the electrical components. However, knit gloves are made to keep one's hands warm, creating a large possibility of the user's hands overheating while using the glove. Because of this, knit gloves are not going to be used for this project.

2. *Rubber Glove*
   The rubber gloves would provide insulation to the user should an electrical piece heat up while also being a solid base for the sensors. The main drawback of the rubber gloves however is the discomfort from wearing the glove for an extended period. Often wearing rubber gloves is for small amounts of time while washing dishes or other cleaning tasks.

Users may feel sweaty and uncomfortable after 30 minutes of using the gloves especially if any heat is generated by the electrical components such as the rechargeable battery. This makes rubber gloves a less-than-ideal choice for this project.

3. *Thin Synthetic Glove*
   Thin synthetic gloves are often used as liners for heavier gloves or used by artists that use drawing tablets. They come in different thicknesses for different uses allowing for gloves that do not prioritize retaining heat, the biggest issue when selecting gloves for this project. Because of these options, we plan on using a thin synthetic glove for this project that would allow us to attach the various components while keeping the user's hand cool. For this project specifically, we plan on using a compression glove that is made to help relieve pain for people typing. This kind of glove is made with long-term computer use in mind and would be thin enough to keep the user's hand cool while also providing a solid foundation for the electrical components.

Base Station Design

For the base station, there are two main components: the charger for the glove and the stand for the IR LED bar. The charger for the glove will plug into a Micro-USB wall adapter and be able to charge the battery. The LED bar will be a square array of four LEDs that the IR camera on the glove can see and use to compute position information. The IR LED bar will be on a goose-neck with an adjustable height connected to a metal base that has the charger coming out of it. When the charger is not being used to charge the glove, the Micro-USB will be used to power the IR LED bar since, in this prototype, the glove will not be charging while also in use. In a future design, this power situation would be designed such that the glove can be charged and used at the same time.

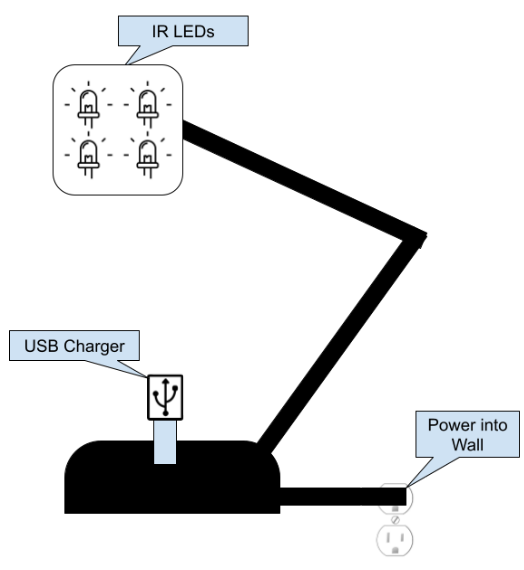


Figure 4: The base station.

Hardware Design

The Glove

Microprocessor

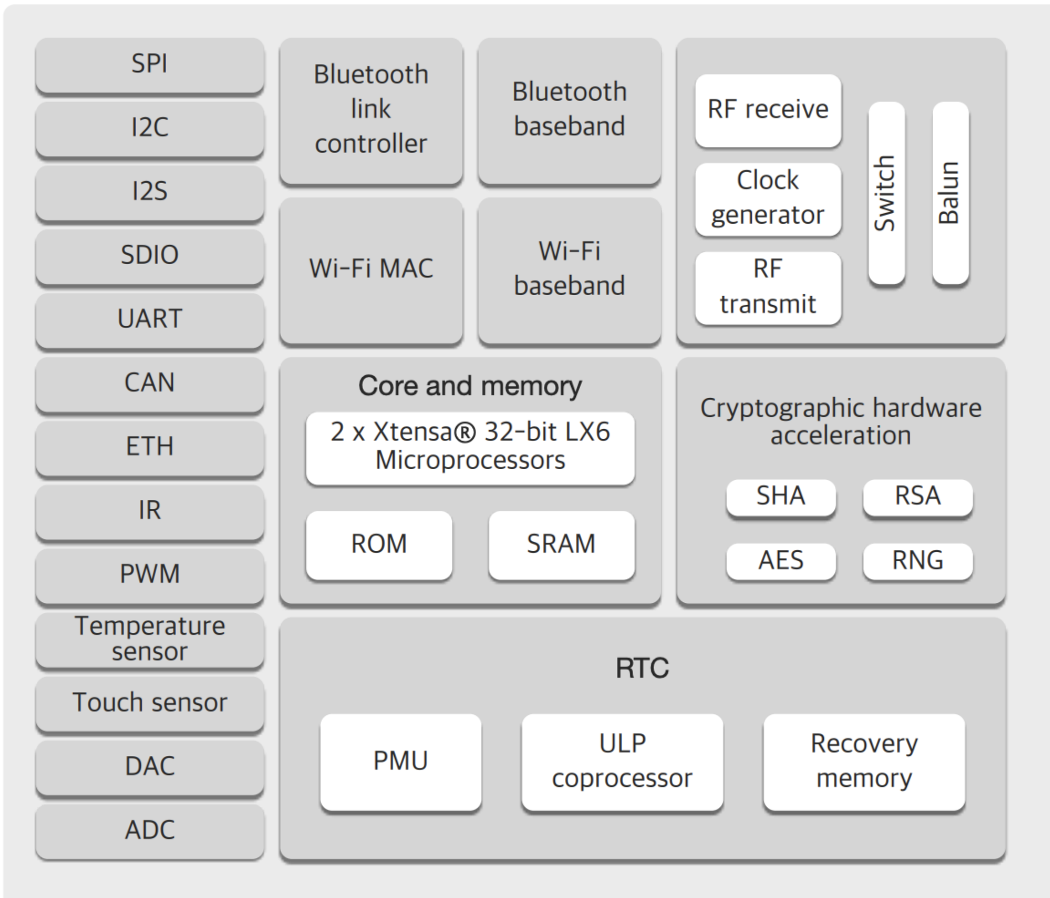


Figure 5: ESP32 Block Diagram.

For the microprocessor on the glove, we chose the Expressif ESP32 Development Board, with its block diagram shown in Figure 5 and pin layout shown in Appendix A. We made our decision based on three main considerations:

1. *Size and Weight*: With the microprocessor being on the glove, the size of the microprocessor should be relatively small and lightweight. The Expressif ESP32 Development Board's product dimension is 2.2" x 1.1" x 0.5" and weighs only 9.6g, which make it an optimal choice compared to the other microprocessor we have looked into (e.g. the 3.2'' x 2.1'' Arduino BT)

2. *Bluetooth Capability*: Given the nature of the project, the microprocessor must have Bluetooth capability to enable wireless communication between the glove and the computer. The Expressif ESP32 Development Board has both WiFi and Bluetooth

capability. Furthermore, it also supports both classic Bluetooth and also BLE (Bluetooth Low Energy) technology, which will provide even more flexibility in the development process and can also enhance energy efficiency of the device.

3. *Easy to Program*: The Expressif ESP32 Development Board is compatible with the Arduino IDE, which makes communication between different hardware components easy and convenient. On the other hand, during the planning and design phase, one of our major concerns is how do we pipeline the glove-generated inputs to the OS of the computer and whether or not the computer will recognize the generated input as a generic mouse or keyboard input. Although we have decided to process the raw data on the computer, it won't hurt to have a backup plan in case that initial design does not work. The Arduino IDE contains built-in mouse and keyboard API respectively. Nevertheless, we have also found other third-party APIs, such as the ESP32 BLE Mouse/Keyboard library that can provide similar functionalities using ESP32 and the Arduino IDE.

4. *Work with the IR Camera*: Since we are going to have the IR camera on the glove, the microprocessor will also have to be compatible with the IR camera we are using. And since the model of the IR camera we are using can be connected and controlled through the Arduino IDE, the Expressif ESP32 Development Board will also be an optimal choice for this purpose.

Flex sensors

A flex sensor made by Spectra Symbol was chosen for the glove because it has a long-range of resistance that allows for finer tuning of sensitivity. The Spectra Symbol flex sensor has a resistance range of 25kΩ-100kΩ, which is wider than the other flex sensor we looked at which only had a range of 10kΩ-20kΩ. This wider range allows for finer tuning which also allows for users to calibrate the glove to register a flex based on their hand dexterity. For a user that cannot fully bend their index finger, this would allow them to set the flexed/unflexed points with higher sensitivity. With the smaller range of resistances on the other flex sensor, there is less ability for the glove to be calibrated to meet a user’s needs.

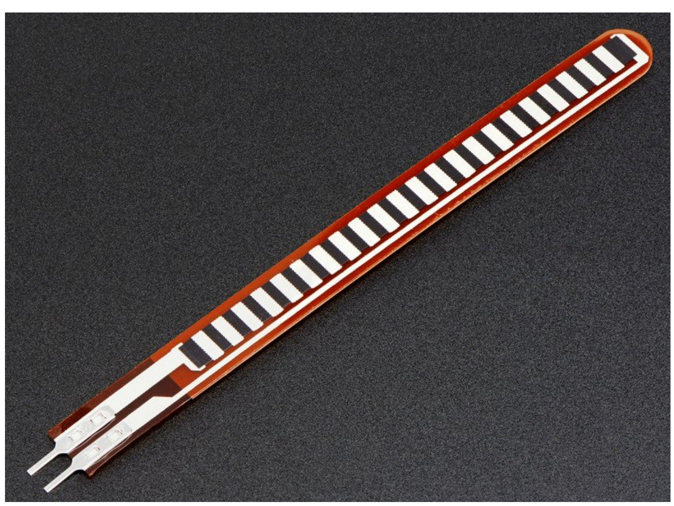


Figure 6: Flex sensor from spectra symbol,

Battery

A lithium ion polymer (LiPo) battery has been chosen for this prototype because of the range of options available and the ability to charge and use the device at the same time. A LiPo battery is not expected to be used for the final product after the prototype, as the battery presents a danger to the user and other components if overcharged. For this prototype, we are going to use a LiPo battery that comes with protections against these issues. When choosing the battery, a battery that could output 3.3V for the ESP32 microprocessor was required. Thus, we are going to use a 4.2/3.7V LiPo battery and connect it using a JST port.

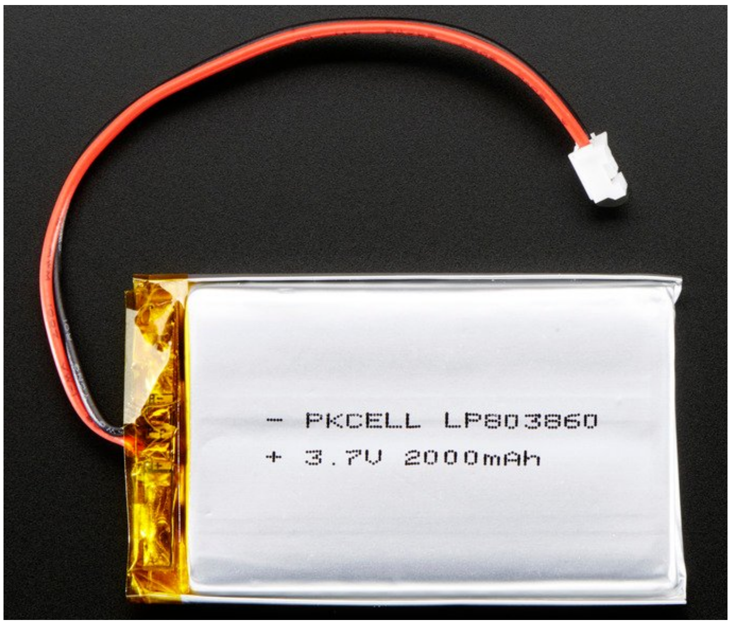


Figure 7: Lithium Ion Polymer Battery

IR Camera

The IR camera selected is the IR Positioning Camera found on dfrobot.com.

Figure 8: IR Camera

This device was chosen due to the following criteria:

1. *Size and Form Factor*: With the IR tracking camera having a relatively small size and weight, the camera can easily be integrated onto the glove. The long wiring included, along with the pre-installed female sockets, makes it easy to connect to the microcontroller on the glove as well. The camera is also relatively cheap, at $23.

2. *Tracking Capability*: This device has the ability to track up to four heat/IR sources. This fully completes our requirements as we need an IR Tracking camera with the ability to track four IR sources. By tracking four IR lights arranged in a square and determining the distance between them, we can accurately determine the distance between the glove and the base station. This counteracts any issues such as a user performing a wrist twist or other issues with the LEDs not being parallel with the camera.

3. *Easy to Program*: This IR Tracking Camera is fully compatible with Arduino, using only four wires, and is therefore easy to program. Any AVR development board can be used to connect the device, and the camera is connected to the microcontroller through the I2C interface.

Other Hardware Components

- On/Off Switch
  - The on/off switch will allow the user to turn the glove off when not in use.
- Buttons
  - Tactical buttons will be placed on the fingertips to respond to touch input.

Hardware Diagram

A schematic diagram of the glove hardware components is shown in Figure 9. The connector will be used to attach the single-cell lithium polymer battery to power the glove.

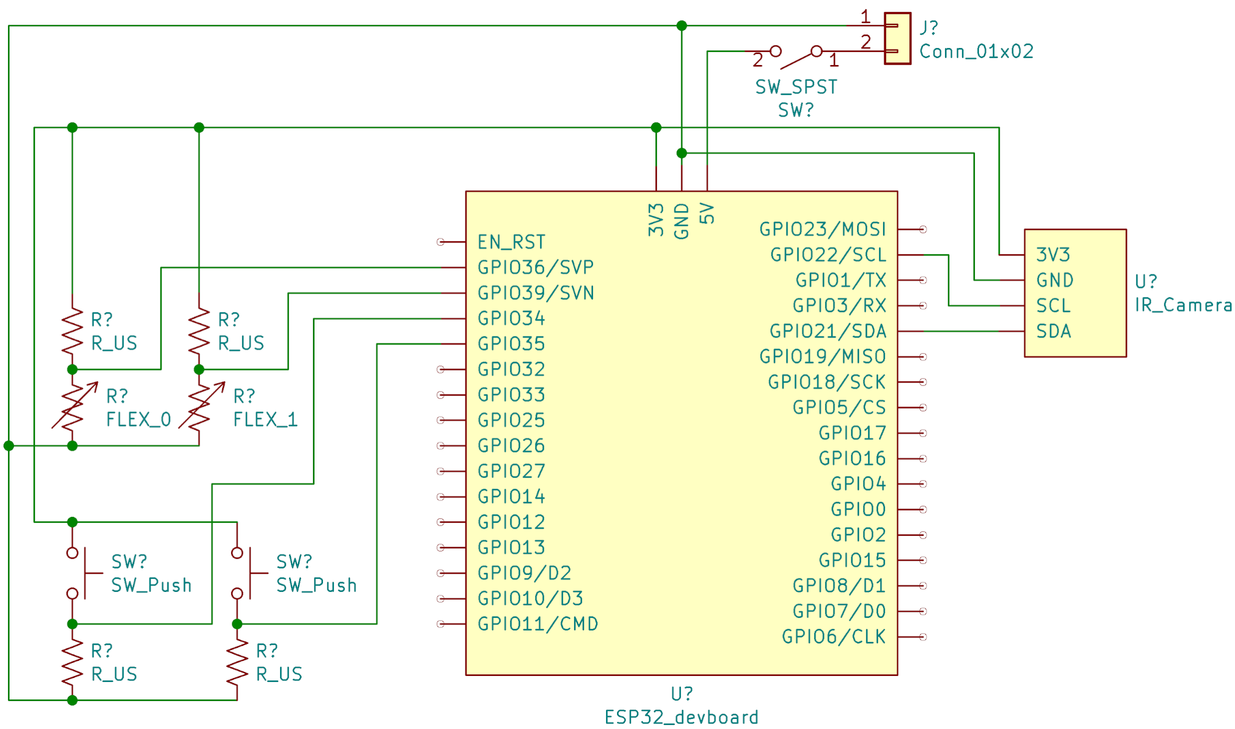


Figure 9: Glove hardware schematic diagram.

Base Station

LED Power Supply

Figure 10: Micro-USB power supply module.

For supplying power to the base station LEDs, a Micro USB power supply module was chosen. This simple component will provide power to the LEDs from the same Micro-USB power supply as used by the battery charger, so the user can simply unplug and replace the power supply cord to facilitate charging or using the glove.

Battery Charger

For charging the battery, we are going to use a pre-made Adafruit charger which will connect the battery's JST connector to a Micro-USB which will provide power. The charger is built to prevent overcharging of the battery and by fully disconnecting the battery, we can ensure that if something does go wrong while charging the battery, the microprocessor will not be harmed.

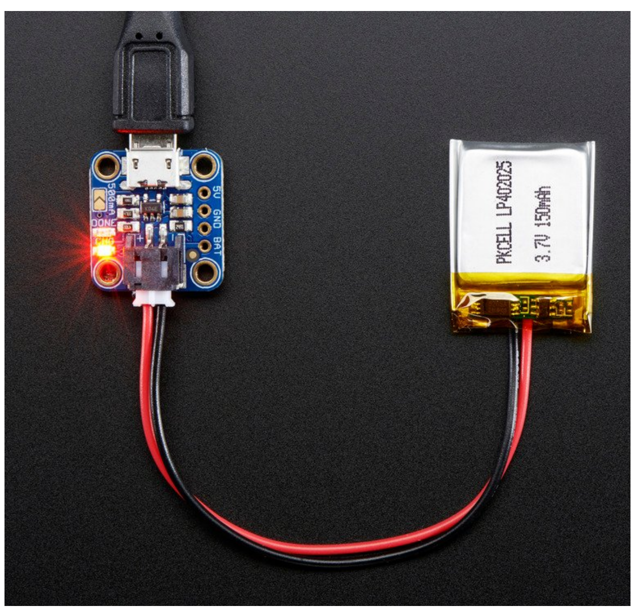


Figure 11: Charging setup for battery.

Other Components

- Infrared LEDs
- LED resistors
    - Small resistors will be placed in series with each LED to provide current limiting.

Hardware Diagram

A schematic diagram of the LED connections is shown in Figure 12. Not shown is the battery charger, which will simply plug into the micro USB power supply cord to charge the battery.

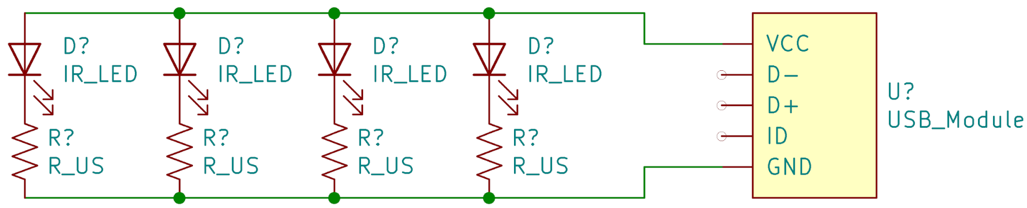


Figure 12: Camera hardware schematic diagram.

Software Design

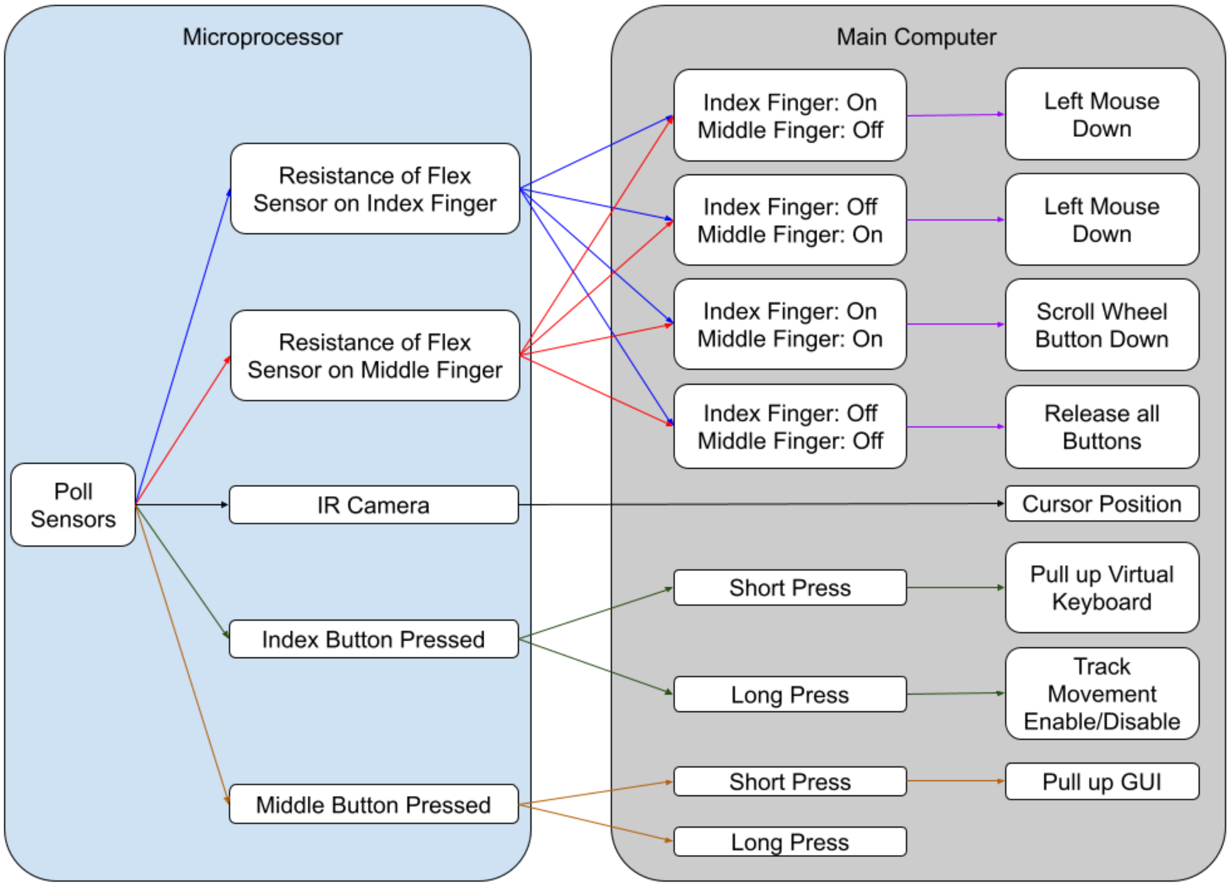


Figure 13: Overall software diagram.

The Glove

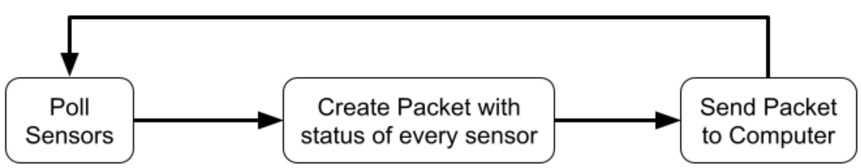


Figure 14: Glove software summary.

The software for the microprocessor connected to the glove connects the glove sensors to the main PC. The glove will poll the sensors, creating a summary of the states to send to the main PC. The microprocessor will send those states to the main computer for further processing and then resume updating the sensor states as they change.

For the button sensors, a single bit will be sent indicating if the button is pressed or not. However, the flex sensors provide an analog input. The microprocessor will send the full analog information to the main PC, where that information will be used to determine if the finger is “flexed” enough to perform the assigned action. The IR camera however has a separate module for computing its information before passing it on to the microprocessor and then the main computer.

<u>The IR Camera</u>

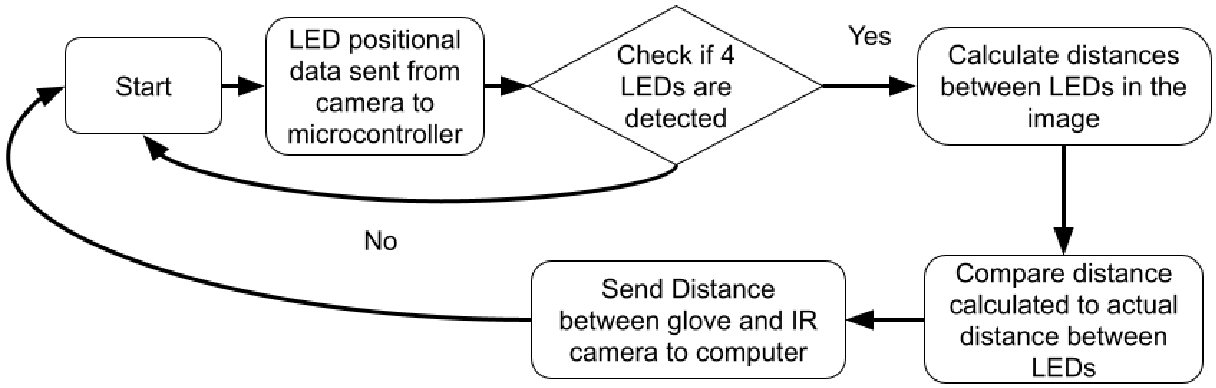


Figure 15: IR Camera Software Summary

The IR Tracking camera connects to the microcontroller through the I2C interface. This means that Baud Rates of 100kHz and 400kHz are generally supported. This I2C interface can be set up between the camera and the microcontroller by setting up the microcontroller as an I2C master device and the camera as an I2C slave device. TWI (Two-Wire Interface) can be used as an alternative, which is very similar with I2C and can use the same wires as I2C.

Once the camera detects a signal, the camera will do some image processing on the image and send the coordinates of the IR light source to the microcontroller. Since the camera can track up to four sources, the camera will send to the microcontroller that the others are empty (returns 1023,1023) if there is only one source. If the camera detects several objects, it will arrange them according to the detecting order. If any of the light sources goes out of view of the camera, that position will update and send that it is empty (return 1023,1023) to the microcontroller. In our case, we require all four of the LEDs to be in the image for our device to work, so the microcontroller will not track/ send positional data to the computer if one or more LEDs are not visible.

The microcontroller can use the positional data sent from the IR camera to find the distance between the microcontroller and the IR camera. An initial distance is determined by measuring the physical distance between the two LEDs. This distance is then compared with later distances between the LEDs in the images from the IR camera to determine the physical distance between the IR LEDs and the IR camera. The microcontroller continuously polls for this data to make sure the data is as accurate as possible.

The Computer

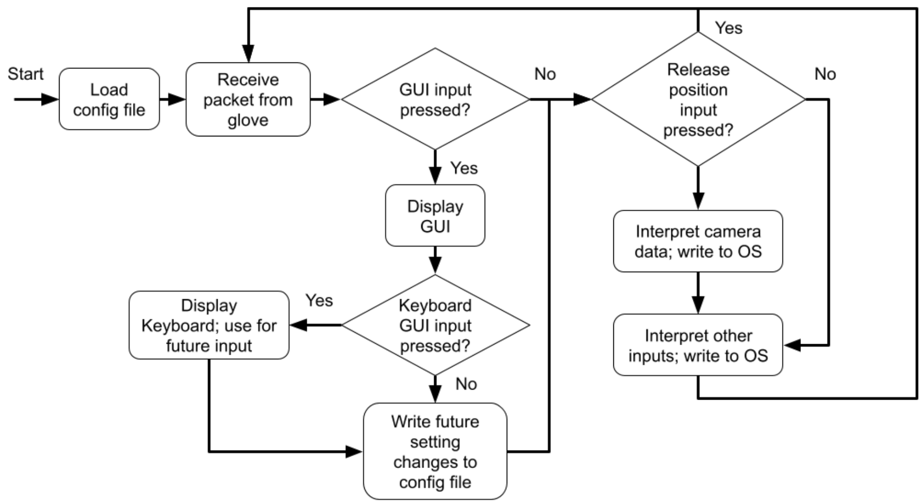


Figure 16: Computer software summary.

The computer software is shown in Figure 16. It will be used to synthesize inputs from the glove to produce the desired response. The glove camera will provide input for mouse position, while the flex sensors and buttons will provide all other inputs. The software will additionally track the state of the GUI and on-screen keyboard, and provide the appropriate response when each is interacted with (based on the interpretation of the input data).

Finally, the computer software will be in charge of long-term storage of user configuration settings. Changes will be stored in a JSON file on the user's computer and will be recalled by the software and written to with each use.

The Graphic User Interface (GUI)

We decided to go with Python as the programming language for building the GUI due to the various libraries and resources available online. Below are some of the most popular GUI framework in Python:

- PYQT/PYQT5
  - Not free, but very convenient
  - Best Styling
  - Have QTDesigner interface: can drag & drop elements and attach code to the GUI easily
- Tkinter
  - The most basic and commonly used built-in GUI library of python
  - No installation needed
  - A lot of resources online
  - Simple and easy to use

- WxPython
    - Python extension module of the wxWidgets GUI API
    - Provide native support
- PyGUI
    - Simple, cross platform library
    - Most lightweight and simple of all
- Kivy
    - Based on OpenGL (mostly used in game development)
    - Can create modern, fancy GUIs
    - Heavily focus on mobile App development

After careful consideration, we decided to use the ***Tkinter*** library as the GUI framework for our user interface. Figure 17 below shows the overall design of the user interface.

The top-most dropdown menu allows the user to toggle between different modes, whereas the rest of dropdowns allows the user to change function mapping of each button or gesture. The device will come with two predefined modes:

- "*Default*": This mode will have basic mouse functionalities, which include cursor movement, clicking, drag/select, and scrolling.

- "*Accessibility*": In addition to the basic mouse functions provided in "Default", a virtual keyboard pull-up function will be added and allows users to type using the virtual keyboard.

Changes made to the function mappings can be saved as a new mode by clicking on the "Save" button. For all the dropdown menus present, the last item will all be the "Customize" option where the user can add new functions such as hotkeys or shortcuts accordingly. The mapping will be saved and appear as a new mode in the top dropdown menu with a user-defined name.

All user information, including mode information, will be saved in a JSON file as a JSON object, which can be accessed and modified by the user accordingly through the Python script attached to the GUI.

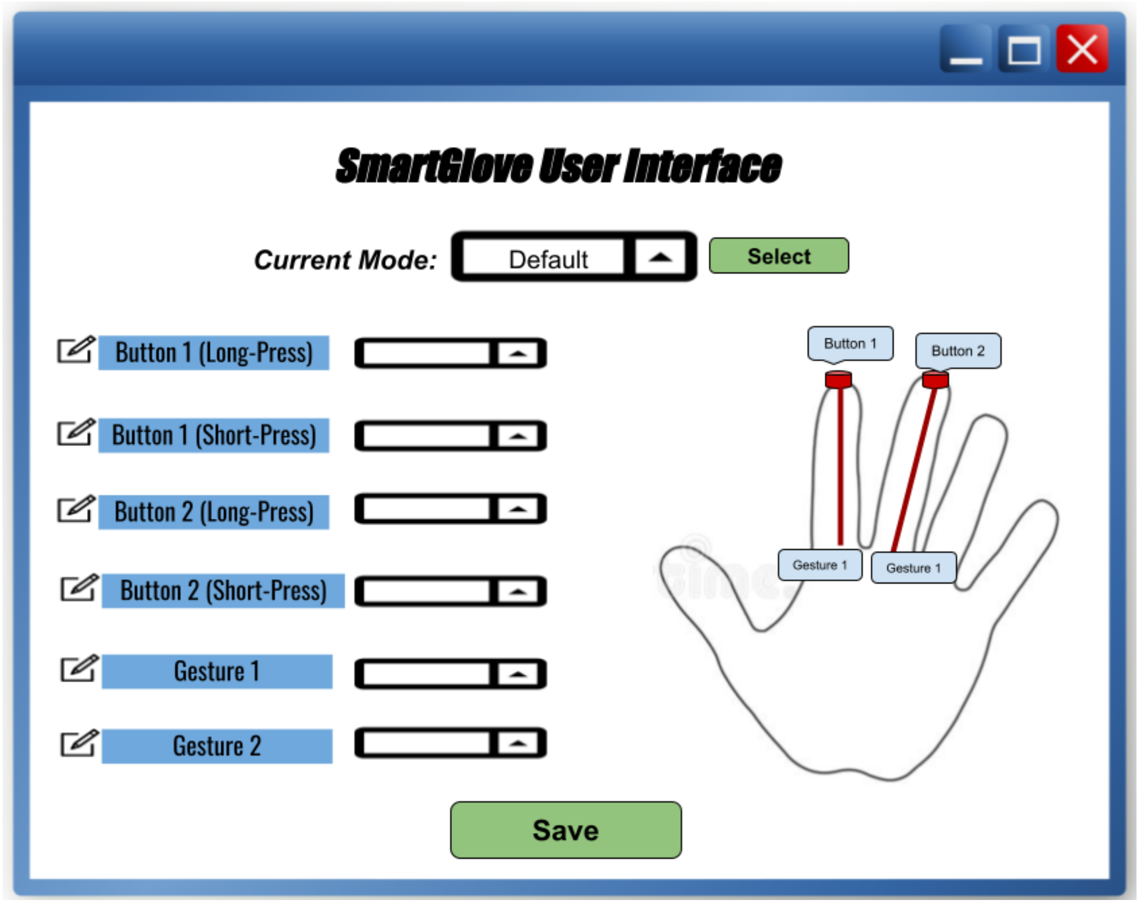
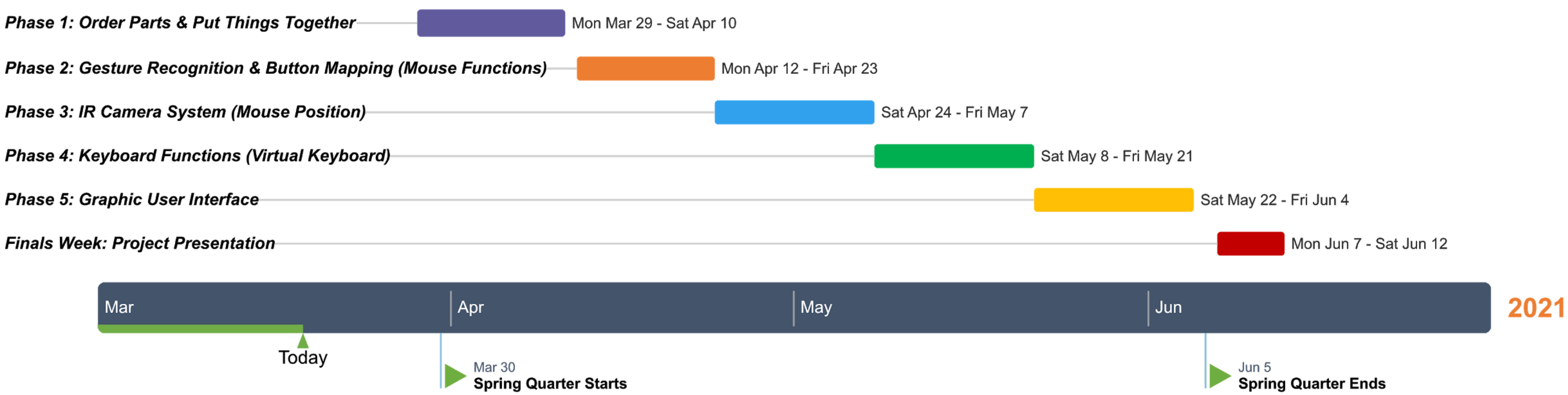


Figure 17: Graphic User Interface Sample Design

Completion Schedule



Parts List

Item	Type	Qty	Unit Cost	Extended Cost
Flex Sensor		2	$7.95	$15.90
LiPo Battery	42/3.7V	1	$12.50	$12.50
Expressif ESP32 Development Board		1	$15.00	$15.00
IR Tracking Camera		1	$23.55	$23.55
On/Off Button		1	$0.95	$0.95
LiPo Battery Charger		1	$6.95	$6.95
MicroUSB with Wall Adapter		1	$7.50	$7.50
Synthetic Glove	Pack of 2 pairs of gloves	1	$3.25	$12.99
Micro-USB power module		1	$0.40	$0.40
Infrared LEDs		4	$0.75	$3.00
Misc. Resistors		1	$8.99	$8.99
Gooseneck Bracket Holder		1	$12.99	$12.99
			Prototype Cost:	**$120.72**

Appendix

Appendix A - ESP32 Microprocessor Specifications

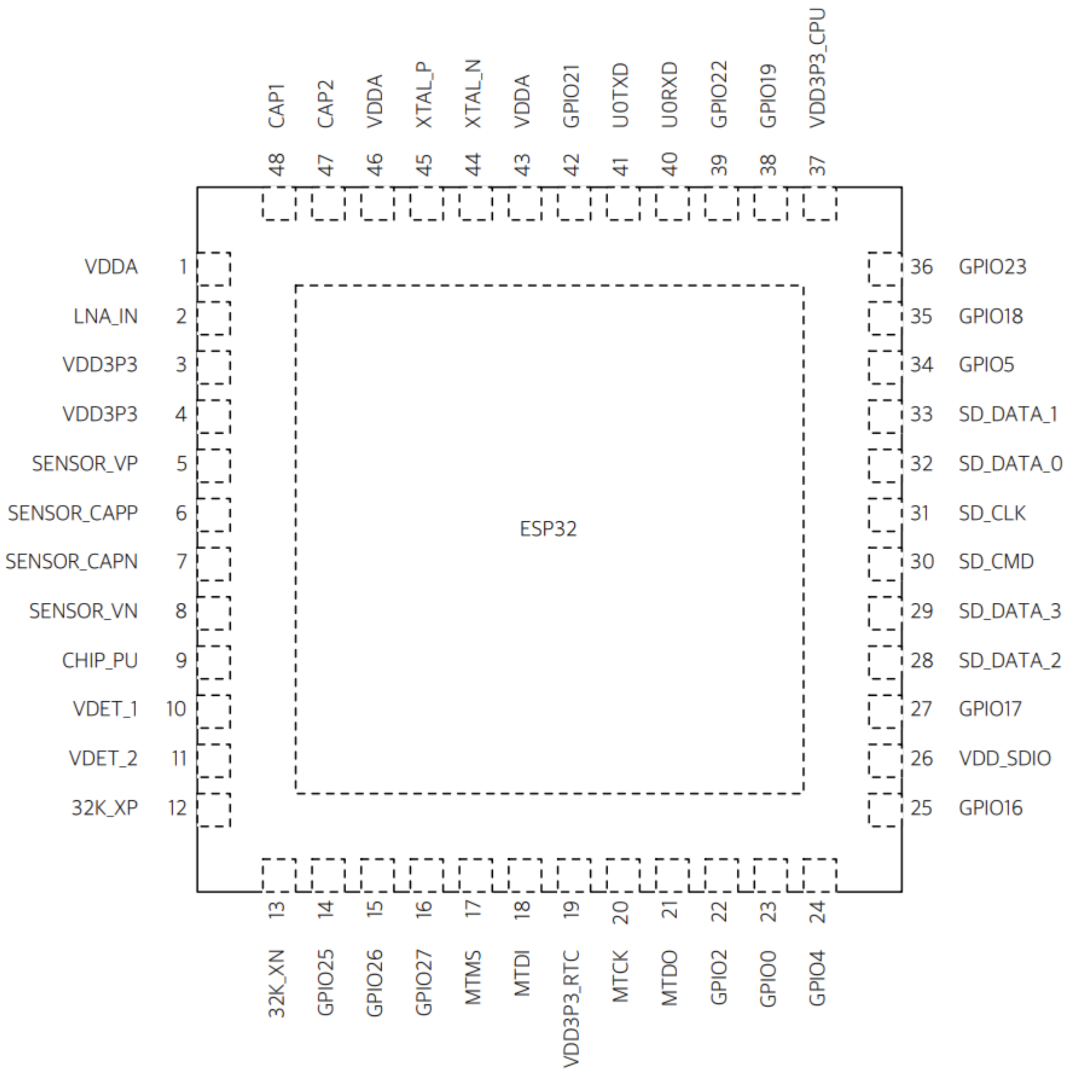


Figure 18: ESP32 Microprocessor Pin Layout Diagram

Table 1: ESP32 Pin Description

Name	No.	Type	Function
Analog			
VDDA	1	P	Analog power supply (2.3V ~ 3.6V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Amplifier power supply (2.3V ~ 3.6V)
VDD3P3	4	P	Amplifier power supply (2.3V ~ 3.6V)
VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36, ADC_PRE_AMP, ADC1_CH0, RTC_GPIO0 Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.
SENSOR_CAPP	6	I	GPIO37, ADC_PRE_AMP, ADC1_CH1, RTC_GPIO1 Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, ADC_PRE_AMP, RTC_GPIO2 Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.
SENSOR_VN	8	I	GPIO39, ADC1_CH3, ADC_PRE_AMP, RTC_GPIO3 Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.
CHIP_PU	9	I	Chip Enable (Active High) High: On, chip works properly Low: Off, chip works at the minimum power Note: Do not leave CHIP_PU pin floating
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, 32K_XP (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
32K_XN	13	I/O	GPIO33, 32K_XN (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
GPIO25	14	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
GPIO26	15	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2

MTDI	18	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
VDD3P3_RTC	19	P	RTC IO power supply input (1.8V - 3.3V)
MTCK	20	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
MTDO	21	I/O	GPIO15, ADC2_CH3, TOUCH3, RTC_GPIO13, MTDO, HSPICS0, HS2_CMD, SD_CMD, EMAC_RXD3
GPIO2	22	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
GPIO4	24	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
VDD_SDIO			
GPIO16	25	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	P	1.8V or 3.3V power supply output
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SD_DATA_3	29	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SD_CMD	30	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SD_CLK	31	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SD_DATA_0	32	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SD_DATA_1	33	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, VSPICLK, HS1_DATA7
GPIO23	36	I/O	GPIO23, VSPID, HS1_STROBE
VDD3P3_CPU	37	P	CPU IO power supply input (1.8V - 3.3V)
GPIO19	38	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
GPIO22	39	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPIHD, EMAC_TX_EN
Analog			
VDDA	43	I/O	Analog power supply (2.3V - 3.6V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Digital power supply for PLL (2.3V - 3.6V)
CAP2	47	I	Connects with a 3 nF capacitor and 20 kΩ resistor in parallel to CAP1
CAP1	48	I	Connects with a 10 nF series capacitor to ground

Appendix B - Specification Document

Mission Statement

The goal of this is to develop an alternative computer input device based on gesture and motion recognition. The core of the system is a glove with sensors embedded on each finger to detect motion and also a number of customizable buttons to provide additional functionalities. The user should be able to control the computer from afar, thus the device will need to be wireless.

Constraints

The overall cost of the device should not exceed the price of other high-end computer input devices such as game pads, game mouse, or other controllers, which would be in the range of $100- $120. Since the screen size of most computers are relatively small, although the computer can be controlled from a distance, the user will have to be within the 30ft (10m) range of the computer device. In addition, the device must also comply with the Electronic Code of Federal Regulations and any federal regulations that are applicable.

Users and Stakeholders

The primary users are people with disabilities relating to hand dexterity which hinders their ability to use a keyboard and mouse. We are targeting users who still have at least three digits, two fingers and a thumb, all of which have partial functionality. The level of functionality that we expect from users is the ability to grasp an object loosely and/or close their fingers into loose fist. Additionally, users can touch each finger individually to their thumb and touch at least two fingers to their palm along with being able to move their thumb along the index finger at least from tip to middle knuckle.

The secondary users are presenters, gamers, and people who want a different way to use a mouse. We assume these people will be willing to experiment with what works best for them.

For both groups of users, we assume they have an average amount of knowledge on how to use a computer, such as connecting a simple device to bluetooth and searching the Internet. We also assume that users are not doing tasks that are extremely mouse/keyboard intensive such as using Photoshop or typing a long email.

Stakeholders:

1. Tech Companies/Video Game Companies
    a. The glove may impact the programs that are used on the computer, favoring ones with simpler interfaces and accessibility modes.
2. Medical Insurance Companies
    a. Depending on the designation of the glove, it may be considered a device covered by HSA. This may impact insurance pricing which affects the insurance company and it’s customers.

3. Other users of shared computer
    a. The glove may alter the immediate settings of the computer affecting other users of the computer if they choose not to use the glove. The other computer users would have to disconnect the glove if it remained connected after its last session.

Universal Access

To accommodate our primary users, we intend to focus on making the glove usable with only a few fingers. The glove can be configured to match the user's personal preferences/requirements. Additionally, the buttons and sensors are intended to be laid out in a way that allows for easy access. Furthermore, during set up, there will be a sensitivity configuration so that gestures can be adjusted to match the user's abilities.

Specification

- The user must be able to set up the device with minimal computer knowledge:
    - The glove and computer will communicate using Bluetooth to eliminate wires and ensure security.
    - The camera and base will activate upon plugging into the USB port of the user's computer.
    - During normal usage, the device should be fully operational after plugging in the base and connecting to bluetooth.
        - The initial setup will require some minimal calibration, where the user is asked to point to and click on a few points on the screen.
    - The device may be initially configured with an intuitive GUI, accessible with the device or a traditional mouse and keyboard.
    - The camera will draw power from the computer USB port.
    - The glove may be charged by plugging into the base when not in use.
- The device must function as a mouse and keyboard replacement:
    - The device must provide mouse pointing functionality.
        - The glove will track the user's position in 2-D space parallel to the monitor
        - This will be performed using input from the camera base station.
    - The device must support scrolling, left click, and right click actions.
    - The device must provide an intuitive method of keyboard input.
    - The device will function with a maximum input delay of 20 ms between user input and computer response.
- The device must be easy to control:
    - The user's hand position will be tracked to provide mouse input.
    - The flexing of the user's index and middle fingers will be used to provide input.
    - Pushbuttons will be used to provide additional input.
        - The index and middle fingers will each include one push button at their tip that can be activated using the thumb.

    - The device GUI will provide intuitive options for controlling the operation of the glove.
        - This GUI will be activated by a pushbutton on the glove to ensure it is always available during device operation.
    - The intuitive device GUI will be used to switch between different operating modes.
        - The default operating mode will provide basic mouse functionality for light computer usage, including web browsing, organizing files, or utilizing the device GUI.
        - The accessibility mode will include more controls to function as a full mouse and keyboard replacement, allowing the user to perform such tasks as typing emails or writing small documents.
        - The presentation mode will provide controls for presentation-specific actions, such as changing slides, drawing annotations, or using a virtual laser pointer. This mode assumes the user is working with Microsoft PowerPoint.
        - The configuration mode will provide the user with freedom to remap glove inputs to suit specific applications, including gaming and virtual reality.
    - The device will allow mapping of inputs to common user actions.
        - These will include at a minimum copy and pasting, volume adjustment, and brightness adjustment.
    - The GUI will be used to adjust sensitivity and input settings.
- The device must be safe for the user:
    - The glove will fully shield the user's hand from all electronics.
    - The glove will act as a heat or flame retardant in the event of electronics failure.
    - Any failure will not harm the user's computer or peripherals.
    - The software will be secure to ensure no user data is leaked.
    - The device will not cause user strain or fatigue beyond that of a regular mouse and keyboard.
    - The rechargeable battery will be located in the camera base, away from the glove and user during normal usage.

Use Cases and Scenarios

Installation: Connecting the glove to the PC
Summary:

The end user connects the glove to their PC.

Preconditions:

- The PC must have bluetooth in order to interface with the device.

- The user has a basic knowledge about computers and wireless connections.
- The user has access to a mouse or other device that can interact with the PC.

Course of events:

1. The user logs in to their PC.
2. The user goes to their computer settings and enables Bluetooth connection.
3. The user turns on the pairing mode for the glove.
4. The user finds the glove in the list of available devices that are Bluetooth compatible.
5. The user connects the glove to their PC.

Exceptions:

1. The PC doesn't find the glove as an available device (drivers are outdated, etc)
2. The Bluetooth chip on the glove does not work as intended.

Installation: Charging the base and glove

Summary:

The end user connects the base to a power outlet or PC. Glove is connected to the base.

Preconditions:

- The user must have a power outlet OR empty USB port on PC.

Course of events:

1. The user connects the power adapter to the base.
2.
    a. The user connects the power adapter to the power outlet.

    OR

    b. The user connects the base to the PC using a USB cord.
3. The user sees that LED light on the base turns on, indicating that it is connected.
4. The user connects the glove to the base.
5. The user sees that LED light on the glove turns on, indicating that it is connected.
6. LED flashes once the glove finishes charging, indicating that charging is complete.

Exceptions:

1. LED for the base or the glove is burnt out.
2. The user should make sure the USB cord is properly attached to the base and power outlet.
3. The user should make sure the glove is securely attached to the base.

Installation: Installing the Software (GUI)

Summary:

The end user executes an executable file which installs the GUI on their PC.

Preconditions:

- The user has a functional PC with proper user privileges.
- The user has an input device (mouse or the glove) that can interact with the PC.
- The PC must have Internet connection or a USB port to install the software.

Course of events:

1. The user logs in to the PC.
2. The user acquires the installation pack (USB key or download).
3. The user executes the executable file.
4. The executable unpacks the installation pack giving user options such as setting the GUI as a shortcut on the Desktop, etc.
5. The installation ends and shows if it has succeeded or failed.
    a. In the case of failure, the computer sends out an error message along with potential fixes.
    b. If the error can't be fixed, cancel the installation.

Maintenance: Reconfiguring Sensitivity of Glove Movements

Summary:

The end user reconfigures the glove sensitivity to make mouse movements more natural.

Preconditions:

- The GUI must be installed on the PC.
- The device is set up and connected to the PC via bluetooth.
- The user has access to the PC in use.

Course of events:

1. The user presses a button on the glove, opening up the GUI.
2. The user uses the glove OR another computer interactive device (mouse) to open up the settings menu.
3. The user clicks a slider that adjusts the sensitivity of the glove movements.
    a. The user repeats this step until mouse movement is comfortable for the user.
4. The user exits the GUI once sensitivity is satisfactory.

Normal: Cursor Movements

Summary:

The microcontroller tracks mouse movements using a variety of sensors.

Preconditions:

- The GUI must be installed on the PC.
- The device is set up and connected to the PC via bluetooth.
- The user has access to the PC in use.

Course of events:

1. The user removes the glove from the base.
2. The user turns on the glove via switch and the glove connects to the PC.
3. Data is polled and collected.
    a. The data for absolute positional data is collected using gyroscopes/ sensors on device.
    b. An infrared sensor tracks the distance/ position of the glove.

4. Data is sent to the microcontroller to be interpreted.
5. A visual indicator (cursor) pops up on the PC showing where the glove is pointing.
6. Microcontroller continues to track hand movements and translate that to cursor movements on the PC.

Normal: Mouse Clicking

Summary:

Mouse clicking is enabled through a combination of gestures and positional sensors.

Preconditions:

- The GUI must be installed on the PC.
- The device is set up and connected to the PC via bluetooth.
- The device is in use.

Course of events:

1. The clicking gesture is recognized by the flex sensors, interrupting the system.
2. Positional sensors show where the cursor is pointing to on the screen.
3. Microcontroller processes and sends the relevant data to the main computer.
4. The item on the computer display is clicked.

Normal: Typing on Keyboard (Single Click)

Summary:

Single letter typing is enabled through cursor movements and mouse clicking.

Preconditions:

- The GUI must be installed on the PC.
- The device is set up and connected to the PC via bluetooth.
- The device is in use.

Course of events:

1. Dedicated button on the glove is pressed, sending a signal to the main computer.
    a. Pulls up the virtual keyboard on the computer display.
2. Positional sensors move the cursor to the desired letter on the virtual keyboard.
3. The clicking gesture is recognized by the flex sensors.
4. Letter on the virtual keyboard is pressed.
    a. Visual feedback which key was pressed.
5. This information is sent to the computer program.

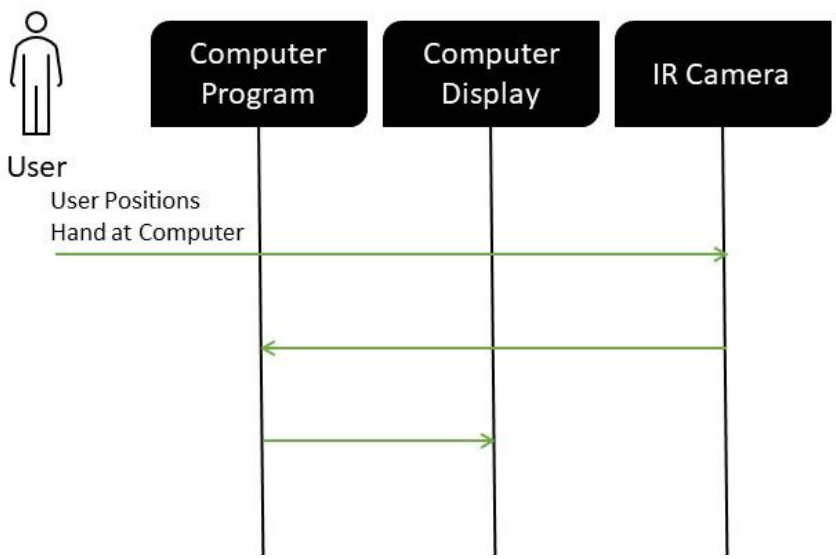
Failure Handling

- If the position tracking system fails, the GUI will alert the user and mouse functionality will be disabled.

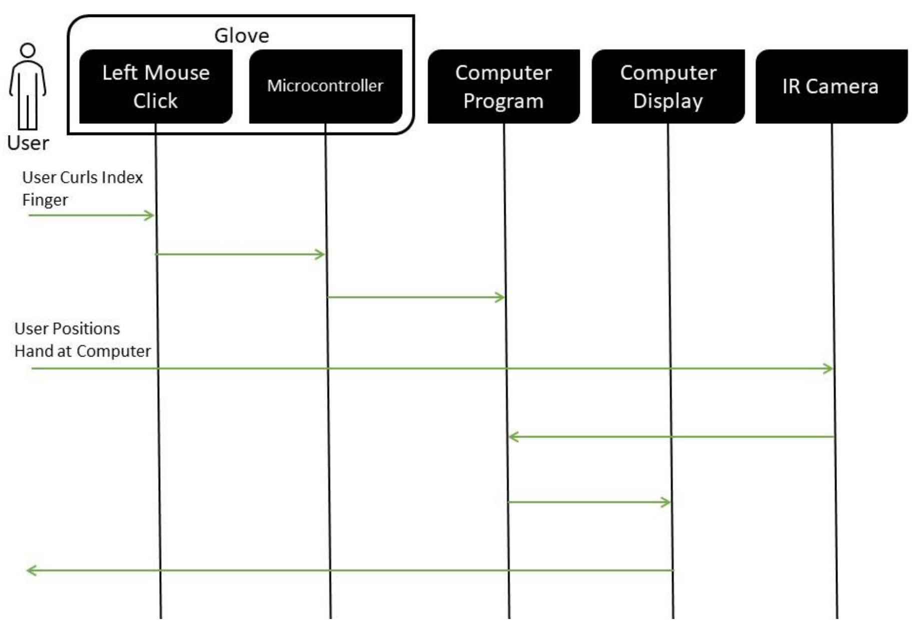
- If the electronics fail or short circuit, the system will attempt to shut off and the glove will protect the user from any burns.
- If the software fails, the system may be restarted by removing and re-inserting the connector, or by pressing the microcontroller reset button.
- The finger position sensors and pushbuttons will be monitored for irregular input. If this is detected, the user will be alerted via the GUI and the input will be disabled.

Appendix

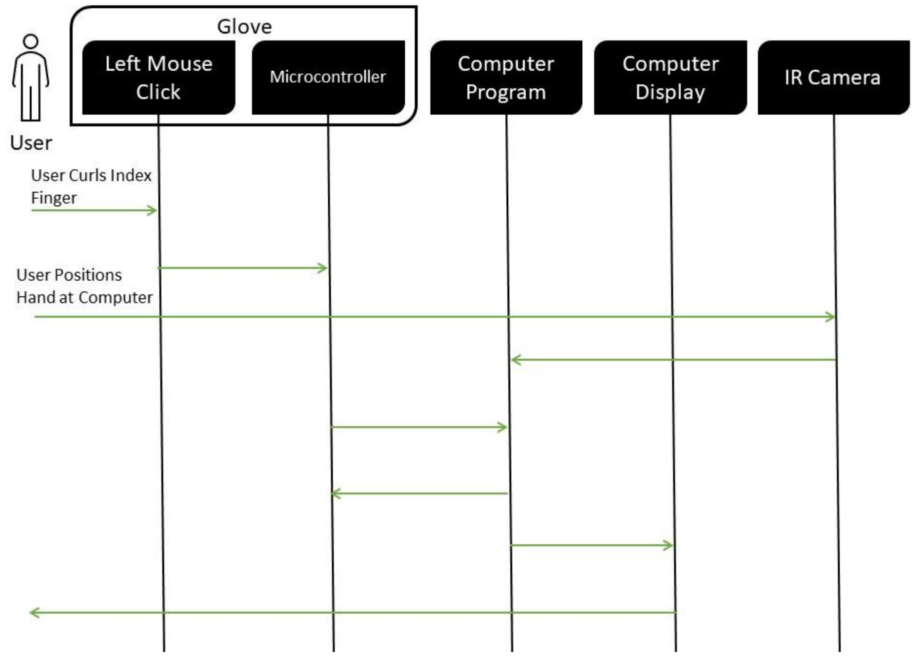
Normal Function (Mouse Position)



Clicking



<u>Mode Change</u>



<u>Typing on Keyboard</u>