Universal Gesture Interface - Design Document

Introduction

Problem and Solution

Much of the way we interact with the world requires expressive, free-flowing modes of interaction. Activities like playing an instrument, martial arts, dancing, or sculpting often can't simply be described by a series of inputs in the correct order at the correct time. They take place in continuous, 3D space—yet, the most complex expression we typically get with a computer is the 2D plane that mouse movement provides. As it currently stands, there is no low-cost, low-fatigue, desk-friendly input device that allows continuous spatial interaction on PC.

We propose a wearable gesture-detecting glove that allows users to interface with computer applications through hand and finger motions. This glove will have a wired USB connection (though wireless would be ideal, we are omitting it for the sake of scope) with two interfaces. The first interface is an HID compliant mouse, allowing the glove to be generally used for regular applications, while the second interface streams live 3D movement data to be interpreted by specialized applications. This dual-interface approach allows the glove to stand on its own as a general-purpose tool while also granting the extensibility to be leveraged to its full potential by specialized applications. Such a device allows for greater immersivity, expression, and even use for both specialized applications and ordinary computer use.

Visual Aid

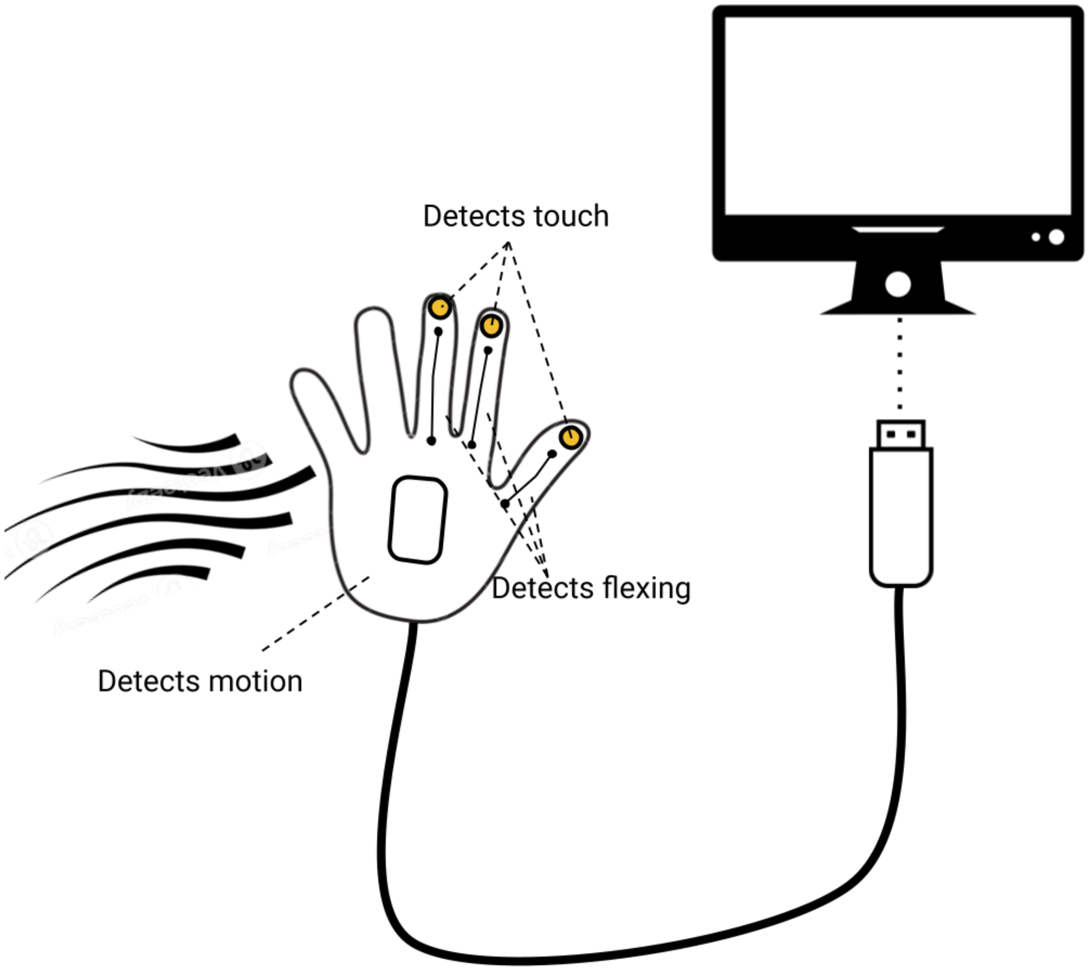

Figure 1: A broad visual overview of the UGI's functionality

High-Level Requirements

1. The system outputs cursor and orientation updates to the host computer with an end-to-end latency of **≤ 20 ms** for 95% of samples during continuous motion.
2. The system correctly classifies a predefined set of hand gestures with a **minimum accuracy of 90%**, measured over at least **N = 50 trials per gesture**, with a **false-positive rate ≤ 5%.**
3. A user can complete a predefined navigation task in the testbed application with a **success rate of at least 80%** and a **completion time no more than 1.5**× that of a standard mouse-and-keyboard baseline.

Design

Block Diagram

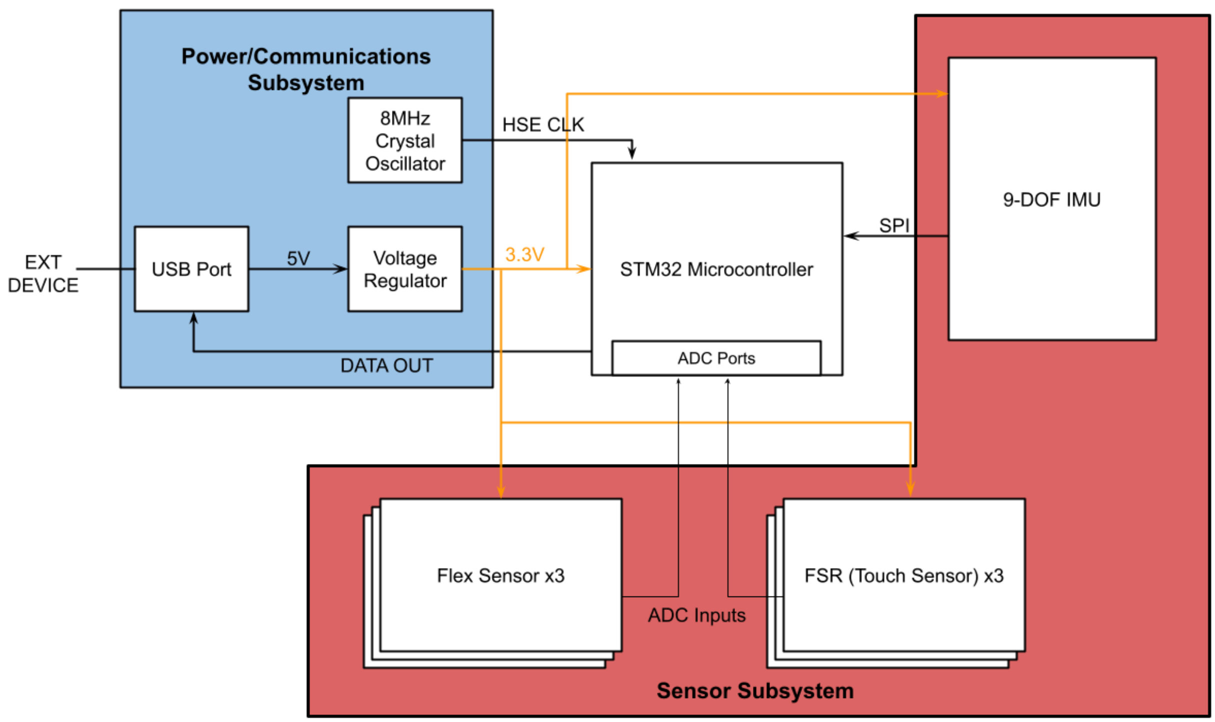

Figure 2: Block diagram layout of the overall system

Physical Design

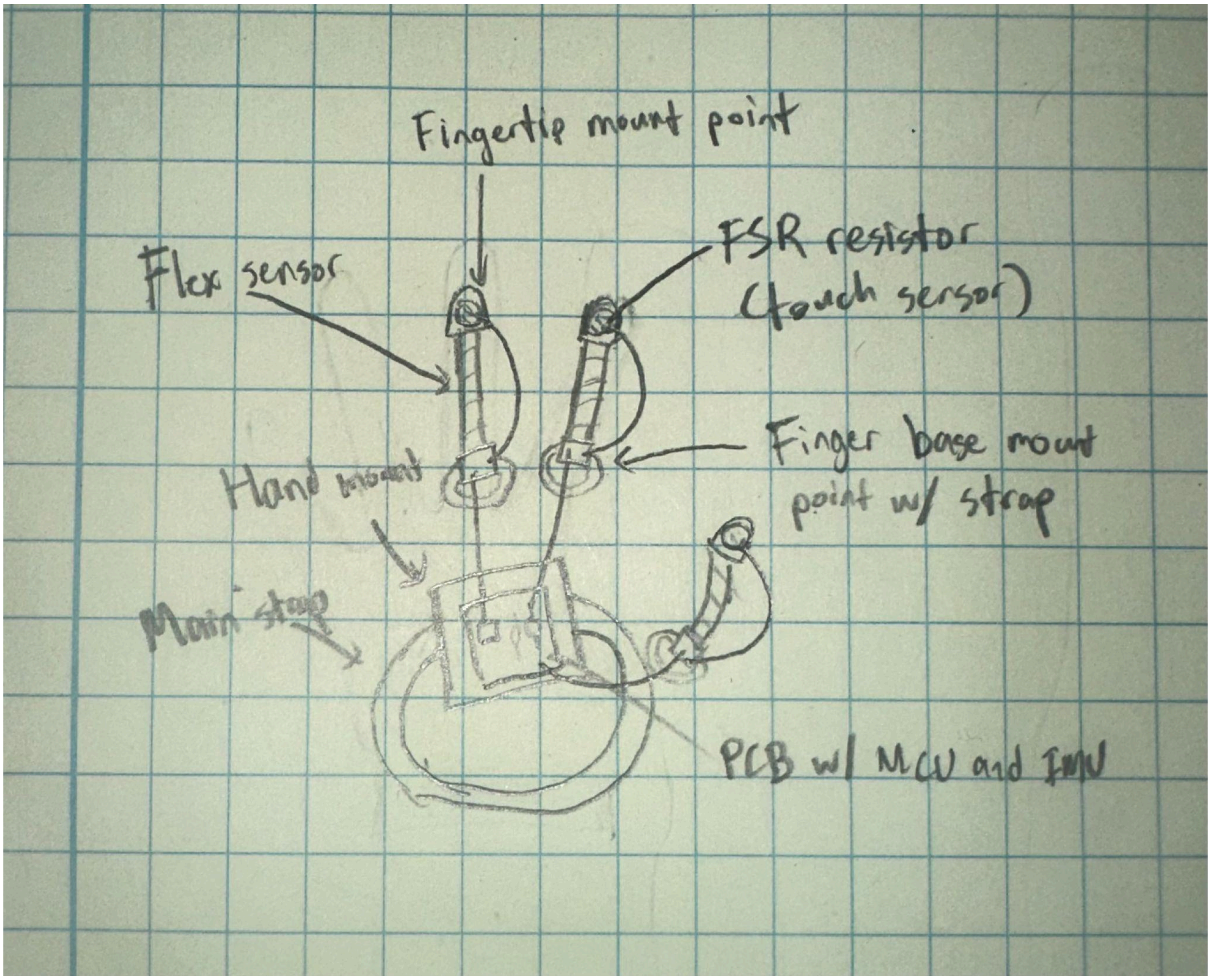

Figure 3: Rough draft of physical layout

For our first round of prototyping, we are prioritizing ease of design, iteration, and construction. We are choosing to disregard factors such as comfort and appearance in favor of getting something that allows us to get usable sensor data for testing. Instead of a glove, our initial design is composed of a series of hand mounts secured by adjustable straps. The mounts themselves will be 3D printed and fit specifically for our hands for ease of testing. The PCB will be mounted to the main hand mount, which secures to the back of the hand, and the flex sensors will bridge the finger-base mounts and fingertip mounts. The FSRs will be attached to the fingertip mounts, which are essentially cone-like pieces that fingertips can slot into.

Subsystems:

IMU Unit

Overview:

This IMU will be a 9-axis accelerometer and will be used for detecting broad-phase translational and rotational movements of the hand. It will be mounted to the back of the palm, and raw sensor data will be sent over SPI to the MCU for processing. The goal is to get an accurate idea of how the glove is positioned in space with minimal drifting, if possible.

Subsystem Requirements:

Components: ICM-20948

This subsystem contributes to the overall design by providing information on the glove's translational and rotational positioning. In order for the glove to be useful as a Human Interface Device (HID), such as a mouse, this information is essential for providing tracking behavior and would be one of the main methods the user would be able to command translational movements of whatever software they are controlling. The unit can communicate with SPI and will be connected to the microprocessor.

Requirements:

- Must provide high frequency data reporting at 7MHz through SPI
- Must operate with an acceptable level of drift. Must not drift further than 1 inch per 5 minutes of active usage. Recalibration techniques can remedy minor drift.

Flex sensors

Overview:

We will mount three flex sensors to the thumb, index finger, and middle finger. They will be connected each to an ADC port by voltage divider with a 50kOhm resistor. 0.1uF capacitors will be used for noise reduction. Input from these flex sensors will be used for detecting specific hand positions.

Subsystem Requirements:

Components: Adafruit Industries Short Flex/Bend Sensor

The flex sensors must show detectable change in resistivity for the entire human range of extension. This need not be a specific value, but must be enough that the firmware can reasonably detect it with the right mapping of ADC input to flex amount. After hardware de-noising, fluctuations from noise must be less than 5% of the total detection range.

Touch sensors

Overview:

Three force-sensitive resistors will be attached to the tips of the thumb, index finger, and middle finger. Similar to the flex sensors, they will be wired to ADCs with voltage dividers (22kOhm) to be read by the MCU. Used for detecting pinching, tapping, and pressing.

Subsystem Requirements:

Components: Geekcreit FSR402 Force Sensitive Resistor

The force sensitive resistors serve the purpose of capturing the data required to sense how the user utilizes their fingers to gesture. They must provide an accurate reading that should avoid hitting false positives. They are an analog component so will require the use of an Analog-to-Digital converter on our Microprocessor. They provide a range of 0.2N-20N which should be sufficient for our use case. Ultimately the challenge would be positioning them mechanically in a way that best captures the user's intended motion. If that positioning is not done precisely, it will be hard to guage what the user's intended gesture is regardless of how good the software is.

Microprocessor

Overview

This microprocessor takes as input all of the aforementioned sensor data and sends USB as output. The processor itself has been chosen for its DSP capabilities, as processing sensor inputs and identifying them as gestures will constitute a considerable portion of this project. Attached to the PCB will be a USB port for connecting to a computer, over which identified gestures are sent as inputs to the computer.

This is also where most of our design decisions will be integrated. For example, the IMU is prone to drift, meaning we'll have to make UX decisions that mitigate its influence, i.e. only moving the mouse when a finger is down on the desk.

Subsystem Requirements:

Components: STM32F405 Microprocessor

The microprocessor's requirements are primarily in terms of its specs. The microprocessor we have chosen has sufficient RAM for DSP (**> 32kb**), appropriate operating speeds (**8MHz** external clock from crystal oscillator) and appropriate peripherals (> **6** ADC ports, SPI port, excess GPIO pins).

USB Connection (Power and Serial)

Overview:

The USB connection will be integrated directly into the PCB of the device. It will have a USB-C port and all the standard considerations for a USB connection. This includes decoupling capacitors, a pull-up on the D+ pin (done in software), and an external 8MHz crystal oscillator. It will act as the 5V power supply to the device, and an LDO will be used to convert this into the expected 3.3V for the chip.

Subsystem Requirements:

Components: 8MHz crystal oscillator, REG103GA-3.3 voltage regulator

The USB port must reliably supply 5V to the whole device, and the voltage regulator must be able to supply 3.3V to the rest of the device. The USB connection must be able to communicate at full speed (**12 Mbps**) within 2 seconds of connection to the host device. It must be recognizable by windows/mac/linux without the installation of external drivers. Reports sent to the host device must be transmitted at a rate that latency is undetectable by humans, minimum **100Hz**.

Physical Frame

Another important aspect of the project will be the physical design itself. In order for our project to be even moderately successful, it has to be wearable. This presents the unique challenge of designing a glove that is both comfortable and can house the electronic components in a way that does not impede movement.

Associated Software

This is not a part of the actual project, but a testbed to demonstrate its capabilities. We will use Unreal Engine 5 to create a very basic flight simulation that allows for controlling the plane with orientation of the user’s hand.

For basic testing, we will also have a barebones program that receives gesture inputs and prints them to the screen when received over serial.

Tolerance Analysis

The area of the project we foresee being the greatest risk factor in terms of proper functionality is the drift and noise of the IMU. It connects to the very core functionality of the device and characteristically tends to have data that is very difficult to work with.

For the sake of testing, we will quantify this through a “drift per gesture” statistic, given by *final - initial - measured difference*. We will aim to minimize this statistic for both translation and rotation, and will account for it with periodic re-zeroing and user calibration.

Cost and Schedule

Cost Analysis

Labor:

- Estimated salary per engineer: $35/hr
- Estimated work: 9 hours/week for 11 weeks ~ 100 hours
- Number of engineers: 3

$35/hr x 100 hrs x 3 engineers = **$10,500.00**

Parts:

Part	Unit cost	Quantity	Total cost
STM32F405 Microprocessor	$5.24	5	$26.20
Geekcreit FSR402 Force Sensitive Resistor	$4.98	5	$24.9
Adafruit Industries Short Flex/Bend Sensor	$8.99	4	$35.96
LSM6DSLTR 6-DOF IMU	$3.58	5	$17.90
Total:			**$104.96**

Note: Quantities include redundancy purchases in case parts break

Total sum:
$10,500.00 + $104.96 = **$10604.96**

Schedule

Week	Tasks
2/23	**All:** - Order dev board - Order flex sensors, IMU, FSRs - Finalize initial PCB design and place order
3/2	**Kobe:** - Begin testing DSP algorithms in Python - Find way to send simulated IMU data to board until PCB comes in

	Kenobi: - Experiment with flex sensor and FSR circuits, get usable data sent to the board **Connor:** **-** Start researching/testing USB interface
3/9	*Assuming PCB has arrived - May need to adjust accordingly* **Kobe:** **-** Start getting real sample data - Create live visualizations for data for debugging **Kenobi:** - Begin prototyping makeshift physical layout. Get something that can reasonably attach to the hand **Conor:** - Get raw sensor data sent over USB - Help with data visualization software
3/16	*Spring break*
3/23	**Kobe:** - Research viable DSP methods - Write python code for DSP methods, run on recorded pre-recorded sensor data **Kenobi:** - Create round 2 PCB design, designing for attachment to the glove - Iterate on physical design if needed **Connor:** - Extra time to work on USB data transfer - Begin working on HID-compliant mouse interface
3/30	**Kobe:** - Additional week for researching/testing DSP methods **Kenobi:** - Mount the flex sensors and force sensitive resistors securely to the glove - Experiment with different mounting strategies to minimize signal noise **Connor:** - Continue work on HID interfacing over USB
4/6	**Kobe:**

	- Begin translating DSP code to STM32 **Kenobi:** - Assemble a hardware prototype of the entire system - Adjust as necessary to maximize comfort and reliability **Connor:** - Assist Kobe in integrating DSP on the STM32 - Finalize the HID device interfacing for the mouse - If feasible, begin work on developing Xinput device for use with other devices
4/13	**Kobe:** - Extra time for translating DSP code - If extra time, expand gesture set **Kenobi:** - Ensure discreet wire management allowing for the glove to flex - Fine tune sensor sensitivities for the most intuitive performance **Connor:** - Assist Kobe in integrating DSP on the STM32 - Extra time for issues encountered with HID interfacing - If feasible, continue work on developing Xinput device for use with other specialized programs.
4/20	**Kobe:** - Continue refining gesture set - Begin work on testbed program **Kenobi:** - Assemble a final prototype for the demo **Connor:** - Assist Kobe in developing the testbed program. - If feasible, finish work on the Xinput device
4/27	**Kobe:** - Finish testbed program **Kenobi:** - Extra time to continue refining the final prototype **Connor:** - Extra time for refining user experience with USB protocol/device integration in software.

	- Assist Kobe with finishing testbed program
5/4, 5/11	*Extra time allocated for delays. If everything goes according to schedule, use this time for additional functionality and polishing*

Societal Impact, Engineering Standards, Ethics, and Safety Considerations

Section II subsection 7 of the IEEE standard states that engineers must "treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression." Two characteristics that would greatly affect our users' ability to use this product are hand size and handedness. We will need to take care in our design decisions to make sure our product is accessible to as many people as possible. Exact solutions to this issue will require iteration through the design process, but possible avenues include making multiple hand sizes, making left-handed and right-handed gloves, or making the components mount to the hand in a way that accounts for all ranges of hand sizes instead of a typical fabric glove.

The safety concerns of this device are a subset of those of a typical virtual reality headset. Any technology that requires the user moving around physically in a room poses a risk to themselves, their environment, and those around them. This is particularly true when the application involves a high degree of activity or fast, sudden motions. Not only could this result in harm from accidentally striking the environment or a nearby person, but individuals with cardiovascular issues may also be put at risk of heart complications by high activity. These are risks inherent to any physical activity and we cannot completely nullify them through our design, but we intend to include appropriate warnings with our product to encourage safe usage. It is also worth noting that these risks will be slightly magnified for this first iteration of our design due to the wired connection, which increases the risk of damage to surroundings and poses a potential strangling hazard. As this design decision has been made out of necessity for the sake of completing the project within course deadlines, we will not be addressing it this semester, but the goal for a finished product would be to have a wireless connection to improve both usability and safety.