



Glovesy

BY

Alan Devine - 17412402

Sean Moloney - 17477122

A functional specification document

As a requirement for CA400

Dublin City University (DCU)

PLAGIARISM DECLARATION

I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I/We engage in plagiarism, collusion, or copying. I/We have read and understood the Assignment Regulations. I/We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

I/We have read and understood the referencing guidelines found at
<https://www.dcu.ie/info/regulations/plagiarism.shtml>,
<https://www4.dcu.ie/students/az/plagiarism>,
and/or recommended in the assignment guidelines.

Project Title	Glovesy
By	Alan Devine - 17412402 Sean Moloney - 17477122
Field of Study	Computer science
Project Advisor	David Sinclair
Academic Years	2020/2021

ABSTRACT

Glovesy is a wearable computer interfacing device in the form of glove which will allow the user to interface with their computer by using custom macros, or use the device for hand-tracking in VR or AR applications.

Keywords: : Wearables, human-computer interfacing, VR, AR, Arduino

GLOSSARY

- **VR Headset** A head-mounted device for use in consuming Virtual Reality Content.
- **AR Headset** Similarly to a VR Headset, an AR Headset is a head-mounted device which aims to provide a User with content that is built around their surroundings rather than replace them in the case of a VR Headset.
- **Arduino** An Open-Source electronic prototyping platform.
- **ISR**

TABLE OF CONTENTS

PLAGIARISM DECLARATION	i
ABSTRACT	ii
GLOSSARY	iii
Introduction	1
1.1 Overview	1
General Description	2
2.1 Product / System Functions	2
2.2 User Characteristics and Objectives	2
2.3 Operational Scenarios	3
2.4 Constraints	3
Functional Requirements	5
3.1 Data Transfer	5
3.2 Hand Tracking	5
3.3 Finger Tracking	6
3.4 Gesture Recognition	6
System Architecture	7
4.1 Arduino	8
4.2 Flex sensor Array	8
4.3 IMU	8
4.4 Macro Database	8

High-Level Design **9****Preliminary Schedule** **13**

6.1	Hardware Requirements	13
6.2	Software/ Wetware Requirements	13
6.3	Major Tasks	13
6.4	Gantt Chart	14

Introduction

1.1 Overview

Glovesy is an Arduino based wearable device which will allow the user to interface with their computer, either by using user-defined macros, which will be set up using our program which will allow a number of gestures do be defined to certain actions within the PC, or by allowing the user accurate hand and finger tracking for use in Virtual and Augmented Reality.

Gestures will be defined by a user and mapped to some action on their PC. Actions could include entering a combination of keystrokes, opening an application, raising/ lowering system volume, etc. The will be executed upon the user repeating their chosen gesture.

General Description

2.1 Product / System Functions

Glovesy is a wearable human/computer interacting device, which will allow users to interact with their PC in a number of different ways, for different scopes. The primary focus of the device, will be to allow the user to set up macros, or certain movements or gestures, which the computer will recognise as a specific command, thereby allowing ease of use. Another function of the device will be to track user hand and finger movements for increased accuracy and control in VR applications, since the device is so low profile, as opposed to current VR controllers which tend to be bulky, handheld devices.

2.2 User Characteristics and Objectives

Glovesy is designed for users who engage in VR/AR applications. This includes, but is not limited to, Gaming, Product prototyping across remote teams, Creating virtual art, etc. In addition to VR/AR users, given that Glovesy is designed to be a standard input device, we also envision it being used in environments such as, a presenter navigating a stage, users who for whatever reason may not feel comfortable/can use traditional input devices, etc. We hope to make Glovesy as accessible as possible to the vast majority of users, that being said, here is our idea of the "Ideal User".

- Experience with gesture-based input devices
 - e.g. Nintendo Wii, or an Air mouse.
- Experience with defining Macros

- Keyboard and Mouse Combinations that are present with devices with extra additional keys/ buttons.

2.3 Operational Scenarios

1. Gaming in a VR setting:

- The User would see a virtual representation of their hand (this is entirely dependent on the Game itself). The user would be able to interact with their surroundings using Gloves rather than a traditional controller.

2. Using an AR Device:

- The User would be able to control their AR Device using hand gestures.

3. Mouse Emulation:

- Within a normal desktop, a User would be able to control their mouse cursor by moving their hand in a way similar to the WiiMote. The user would be able to left click by curling their index finger and right click using their middle finger. The threshold for when a click is actuated will be set by the user.

4. Gesture Control:

- The User would define some sequence of hand movement combined with finger position, and create a mapping to certain actions on their computer, e.g. raising and lowering system volume, advancing slides, etc.

2.4 Constraints

There are a number of constraints that we foresee will have some impact on the development process of this project.

- **Distinguishing Gestures:** It may be challenging to distinguish hand movements between general movement and purposeful gestures.

- **Application Support:** It could be difficult to set up programs to use the device, as, particularly in games, there may be different controls that are pre-defined.
- **Cross-platform:** a different driver will need to be made for different operating systems, which could be challenging.

Functional Requirements

3.1 Data Transfer

The device must be able to transfer data to the PC reliably.

- **Criticality** This Aspect is critical, since, even if the rest of the data is accurately measured, if the data isn't transferred reliably, the device is useless, meaning the entire functionality of the device is dependent on this.
- **Technical Issues:** One of the issues with this will be getting the bluetooth module on the board we are using to be reliable, otherwise we can use the device with

The device must be able to track the overall hand's orientation and position in 3D space.

3.2 Hand Tracking

- **Criticality:** This aspect is critical to the overall accuracy of the device, especially for tracking the hand for VR and for Gesture Recognition, as well as for general use, as it can be very frustrating to move a mouse cursor with a device that doesn't accurately read user inputs.
- **Technical Issues:** A significant issue with this is that the accuracy is almost entirely dependent on the hardware, meaning if the IMU device we use is inaccurate, we won't be able to do anything to improve the tracking.

3.3 Finger Tracking

The device must be able to track the fingers' individual joints accurately.

- **Criticality:** This aspect is also critical, as it is one of the main functions of the device.
- **Technical Issues:** One of the main issues with this is that the flex sensors that we are using are made by us, so there will need to be some calibration when first using the device.

3.4 Gesture Recognition

The system must be able to distinguish between random movements and intentional gestures.

- **Criticality:** This requirement is important for a specific use-case scenario which is not focused on gaming or vr, but rather on workflow.
- **Technical Issues:** Due to the nature of the device, we only have a certain number of inputs, and thereby limited on the amount of information we get from the device, meaning that we will have to focus on some way to distinguish between gestures and non-gestures through pattern recognition.

System Architecture

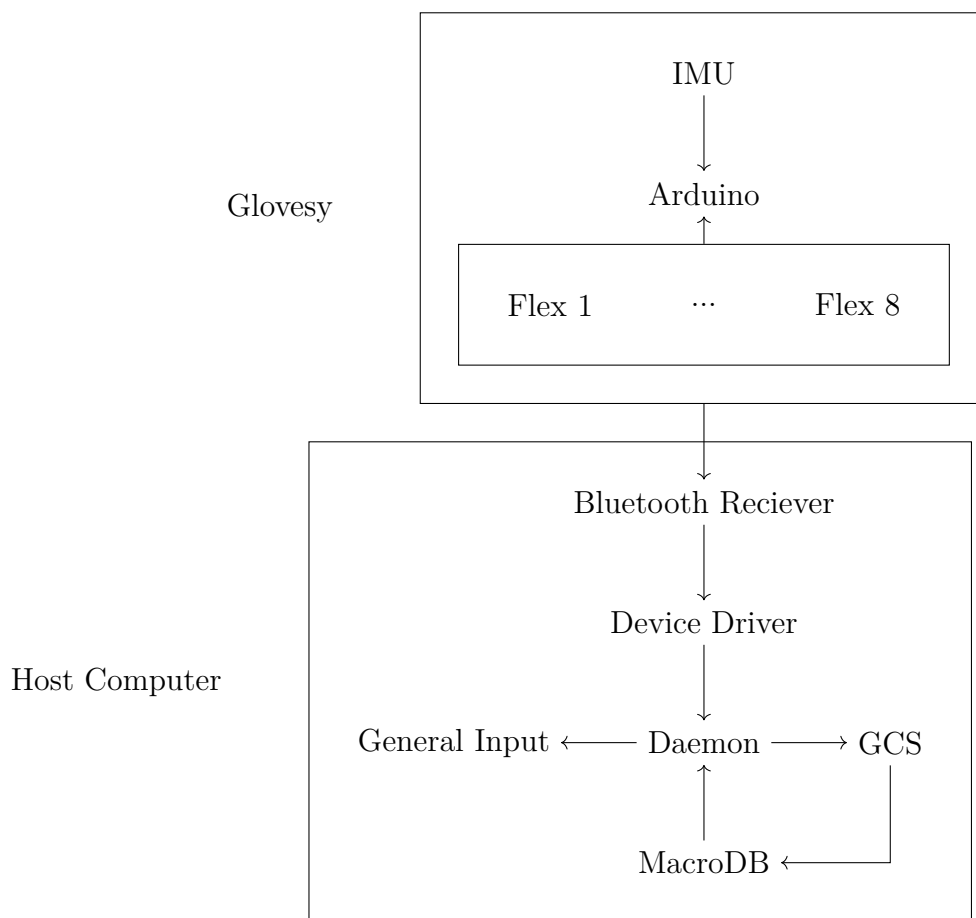


Figure 1: High-Level System Architecture Diagram

4.1 Arduino

The arduino in use is an *Adafruit Feather M0 Bluefruit*. This has an integrated Bluetooth controller which allows us to transfer data to and from the host computer. This Arduino has 10 analog pins which we can use to measure resistance on our flex sensors.

4.2 Flex sensor Array

A flex sensor is a variable resistor that changes resistance based on how much the sensor is bent. For this project we will be making use of 8 flex sensors which will measure the position of each knuckle as well as each finger joint. With the exception of the Thumb and pinky fingers, which will only have one flex sensor each. To help mitigate the costs associated with this project, we will be constructing our own flex sensors using several layers of velostat and conductive thread which will act as our positive and negative terminals.

4.3 IMU

The IMU, or Inertial measurement unit is the device which we will use to measure the device in 3D space. Our device in question, measures Acceleration and Gyroscopic data on the X, Y and Z Axes in meters per second and radians respectively. Our IMU also measures temperature, however that will not be used for this project.

4.4 Macro Database

User defined macros will be stored in a database implemented in MongoDB. We chose MongoDB as conceptually, it makes more sense to group these actions as an object rather than in a relational database.

High-Level Design

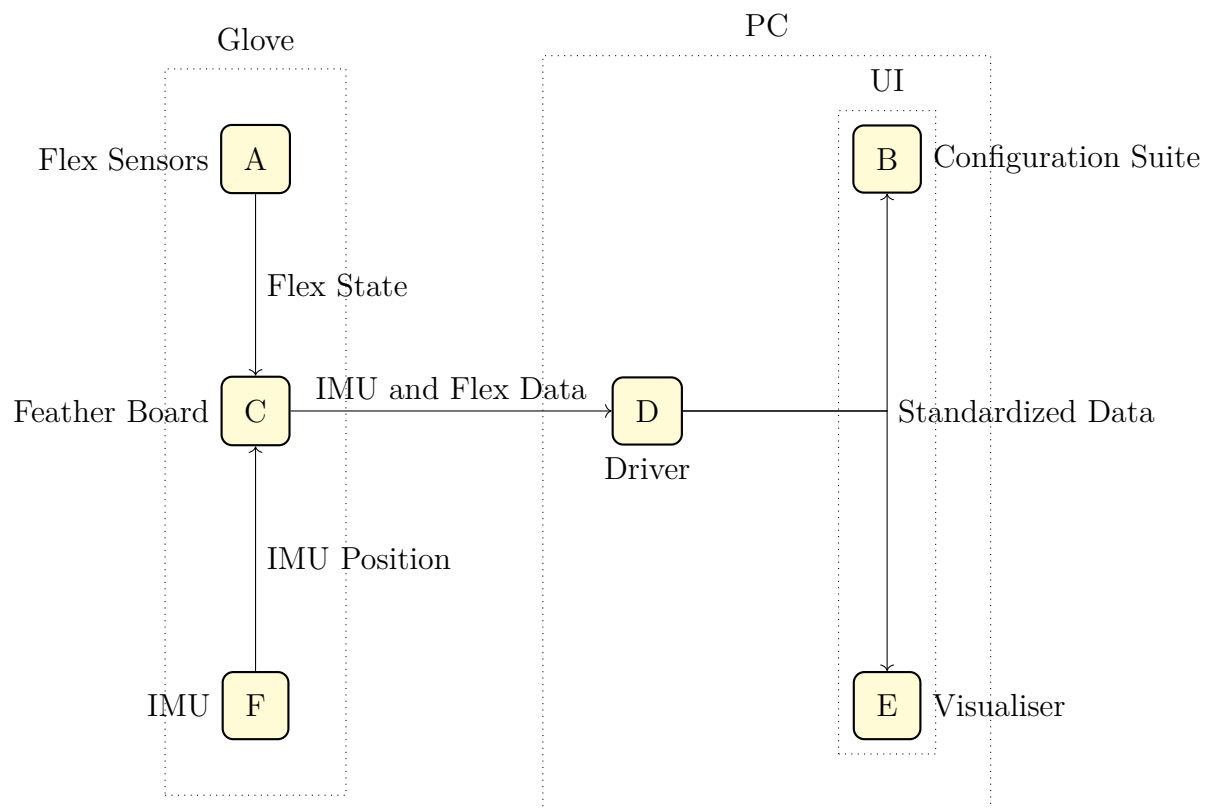


Figure 2: Data Flow Diagram

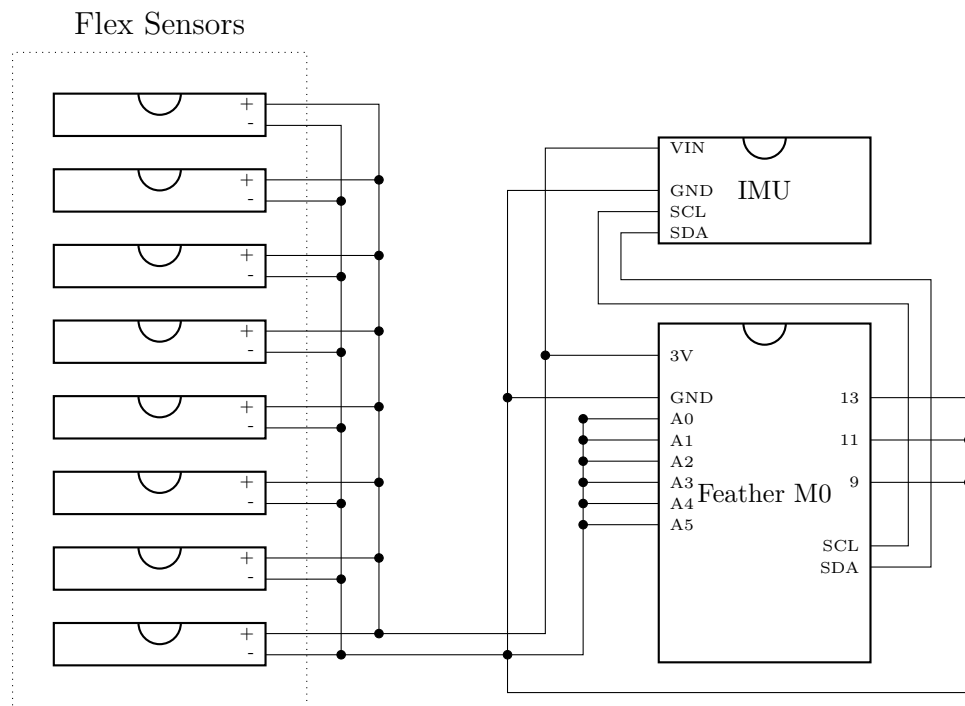


Figure 3: Circuit Diagram

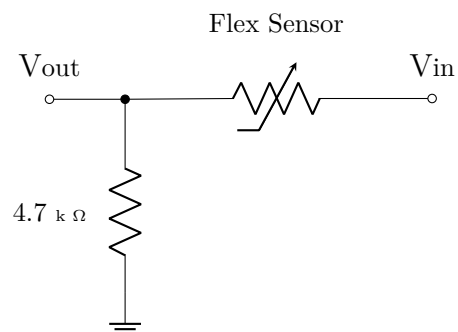


Figure 4: Flex Sensor Circuit

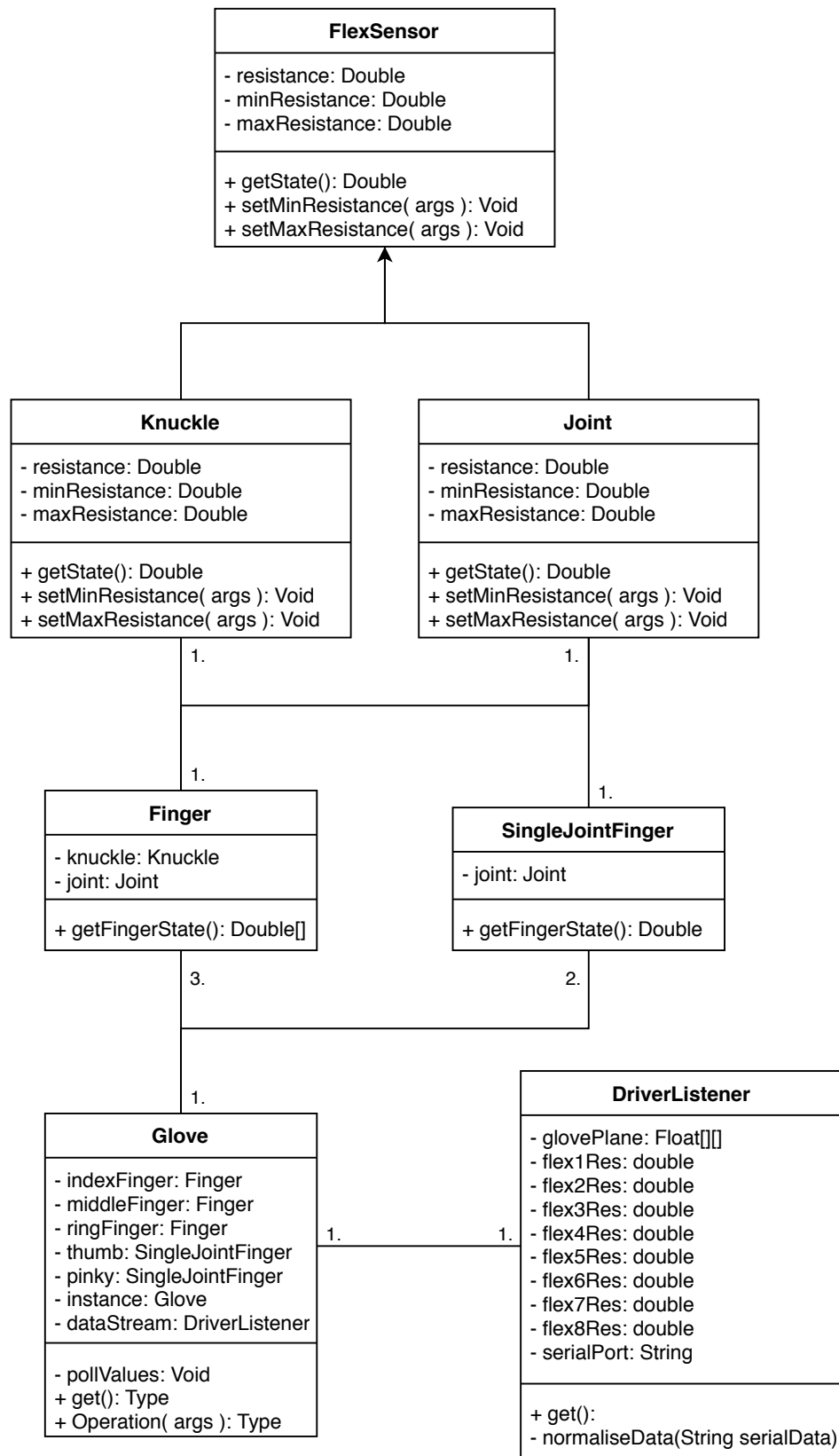


Figure 5: UML Class Diagram

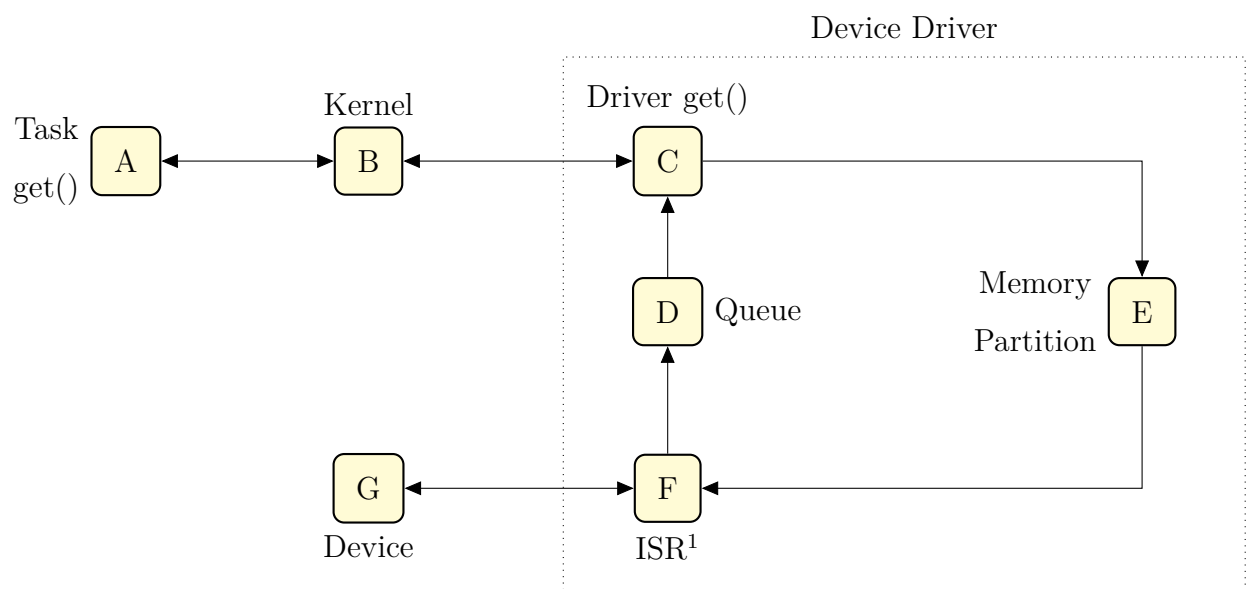


Figure 6: Device Driver

The task calls the device driver via a Kernel device call, the driver receives the message from the queue and returns the data to the task and the buffer to the memory partition. When new data arrives, the ISR¹ allocates a buffer from the memory partition, puts the data in the buffer, and puts the pointer in a message which is then sent to the queue

Preliminary Schedule

6.1 Hardware Requirements

See below a table of the required hardware to complete this project.

Name	Description
Adafruit Feather M0 Bluefruit LE	Arduino board with built in Bluetooth
Adafruit LSM6DS33 + LIS3MDL	9-DoF IMU
Pressure-Sensitive Conductive Sheet	Velostat for Flex Sensors
Stainless Thin Conductive Yarn	30ft of conductive thread
Adafruit Flex Perma-Proto	Flexible Bread board

The above hardware was ordered upon receiving project approval.

6.2 Software/ Wetware Requirements

There are no specific Software/ Wetware requirements to complete this project.

6.3 Major Tasks

- Construct Glove
- Develop Linux Driver
- Develop OpenGL Environment
- Implement Gesture Recognition

Task ID	Task	Dependency
1	Construct Glove	n/a
2	Develop Linux Driver	1
3	Develop OpenGL Environment	n/a
4	Implement Gesture Recognition	1, 2

6.4 Gantt Chart

In this section we will outline our provisional Gantt chart depicting how work will be completed through the duration of this project. The document is layed out as follows:

- Items which are Blue will be completed by Sean Moloney.
- Items which are Red will be completed by Alan Devine.
- Items which are purple will be completed by both of us.

Note: Time to write unit tests is factored in with each development task presented below.

