CS-308-2016 Final Report

Gaming Glove

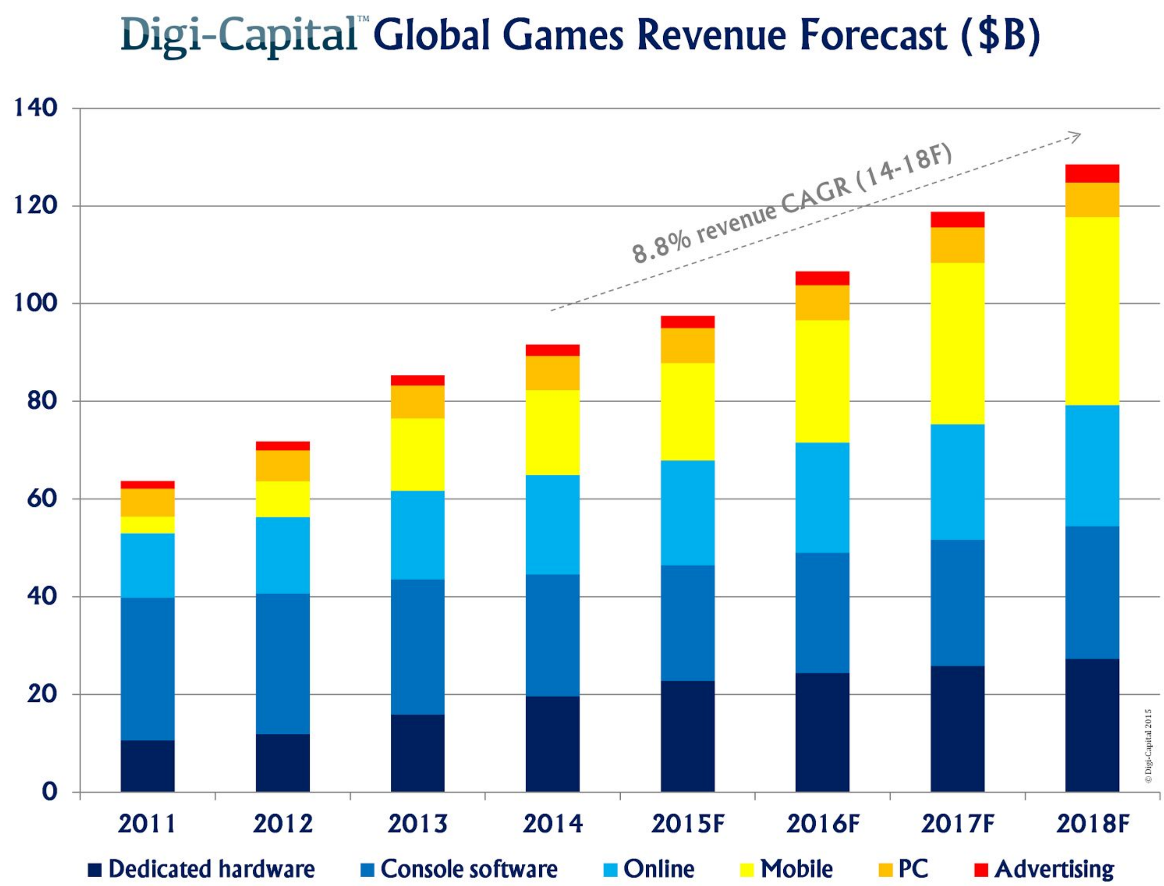
Team number 12

Pratik Fegade (120050004)
Devdeep Ray (120050007)
Haren Saga (120050072)

Table of Contents

Introduction

The video game industry has been growing by leaps and bounds in the last few years. Video games of a lot of genres have been coming up these days. The figure below emphasizes this fact. According to predictions the PC gaming industry will be a 120 billions dollar industry by 2018.



Joysticks and gaming pads are generally the preferred devices gamers use to play video games, on PCs or on gaming consoles. Often these might get a bit cumbersome to maintain and use. Also, a good gamepad costs around $30. This is expensive. In this project, we aim to come up with a more convenient and cost effective gaming device by mounting various sensors on a hand worn device.

Problem Statement

The aim of the project is to build a hand-worn, glove-like device that can be convenient use to play video games. The device should be use palm and finger gestures to provide inputs to the game/application being controlled. It should be easy to install and use, preferably plug and play. Last but not the least, it should be possible to manufacture this device in a cost effective manner.

Requirements

Functional Requirements

The product should satisfy the following functional requirements

1. Should get detected as a joystick on a PC
2. Must work on USB power
3. Needs to be real time and no lags
4. Should support the following gestures
    a. Pinches between a finger and the thumb
    b. Bending of fingers
    c. Orientation of the palm

   Each of these should be mapped to some of the buttons/axes of the joystick

Non-Functional Requirements

The product should satisfy the following non-functional requirements

1. Should be natural to use the device
2. Should be easy to wear
3. Should be comfortable for long use
4. Should be durable and robust
5. Should not hinder mouse usage or keyboard usage

Hardware Requirements

The device uses the following hardware components.

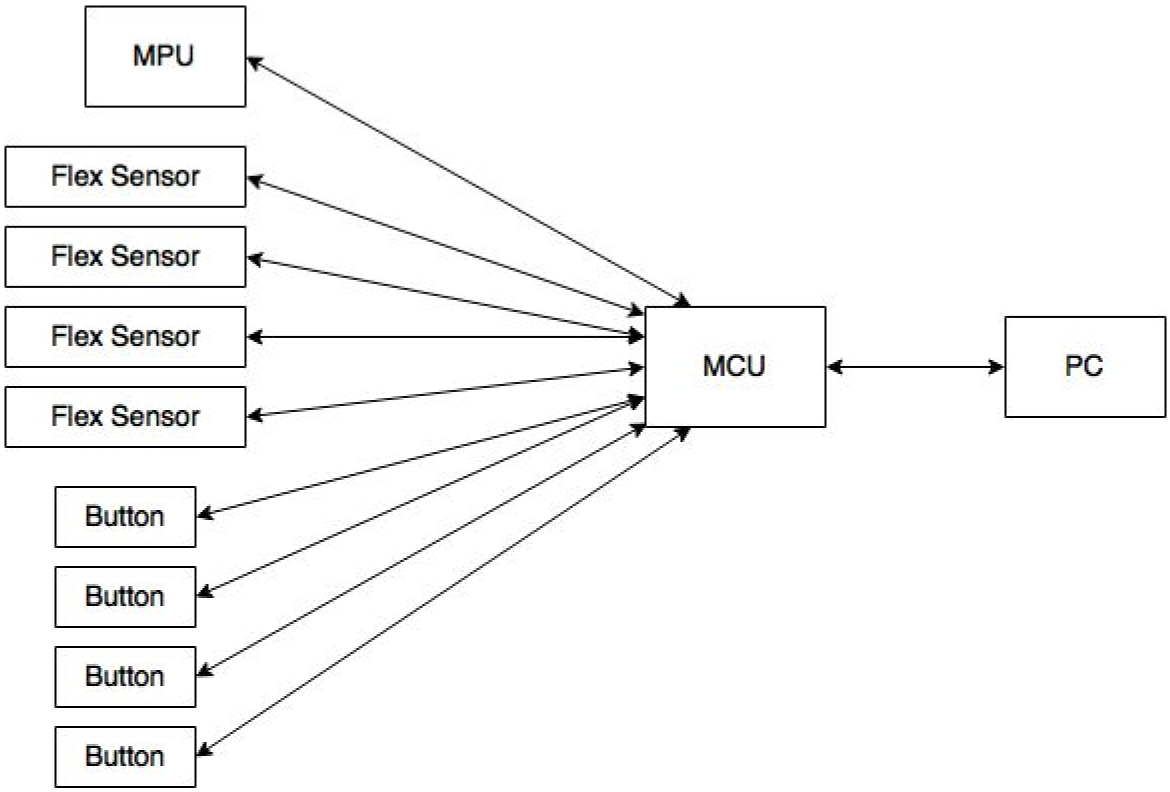
1. TIVA C-series board
2. A 6-axis MPU
3. Flex sensors (4 in number)
   Each flex sensors needs
    a. Some flexible rubber tubing
    b. A white LED
    c. An LDR
4. Mechanical buttons (4 in number)
5. Other small components like PCBs, jumper cables, velcro strips etc.

Software Requirements

The device software has been coded by using the TIVA usb library collection and it needs a USB HiD driver installed on the PC to be used.

System Design

As described above, the system consists of the sensors and the TIVA board. A schematic of the system is shown in the diagram below.



Flex Sensors

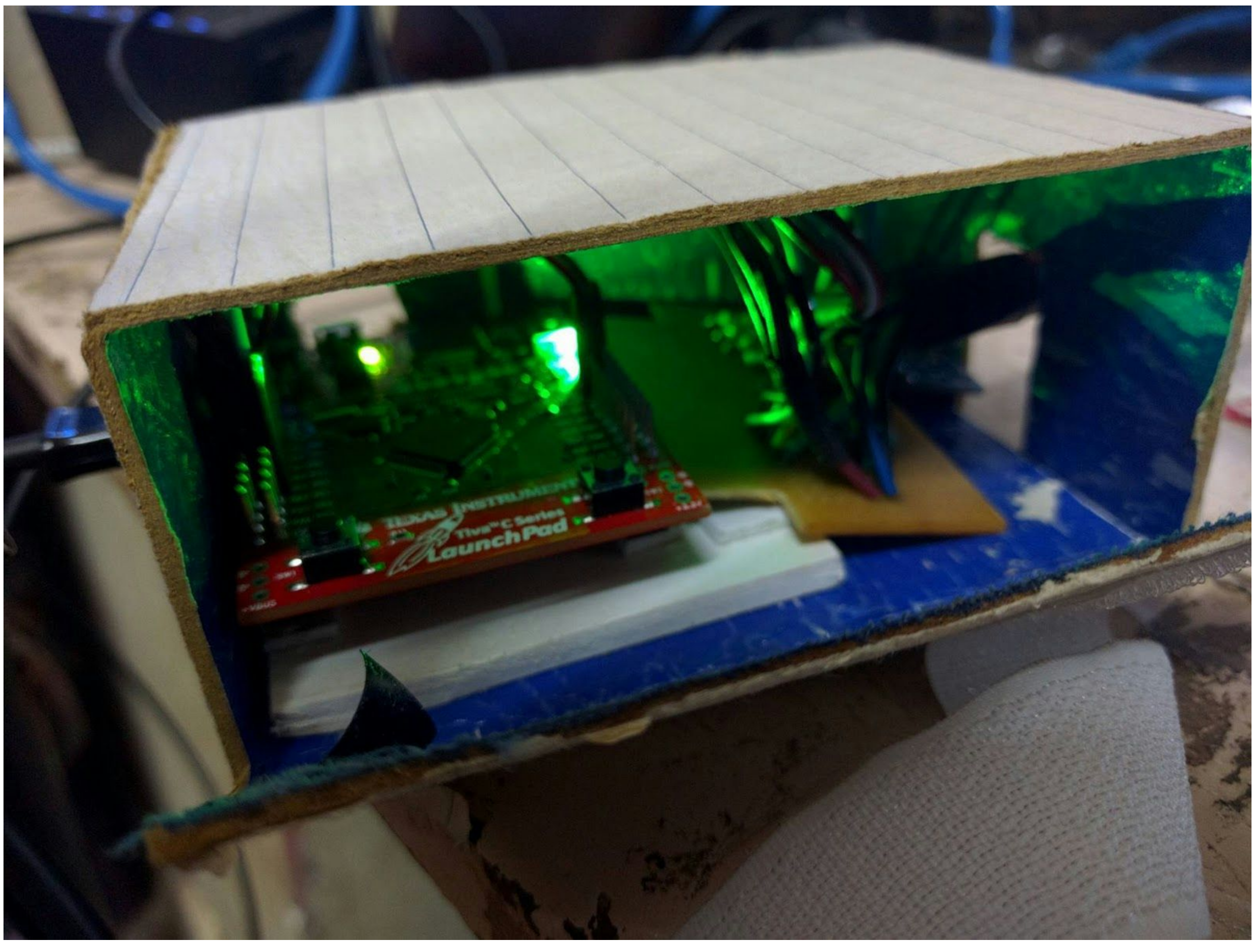
Flex sensors cost about $10 each in the market. With the objective of lowering the production cost of the device, we decided to fabricate flex sensors ourselves using LEDs, LDRs and some flexible rubber tubing. We attach an LED at one end of the rubber tubing and an LDR at the other end. The intensity of the light incident on the LDR changes as the rubber tube is bent. This leads to a change in the resistance of the LDR and this thus can be used to detect bending of fingers. The total of one flex sensor thus comes down to $1 from $10 and this leads to huge savings in the fabrication cost.

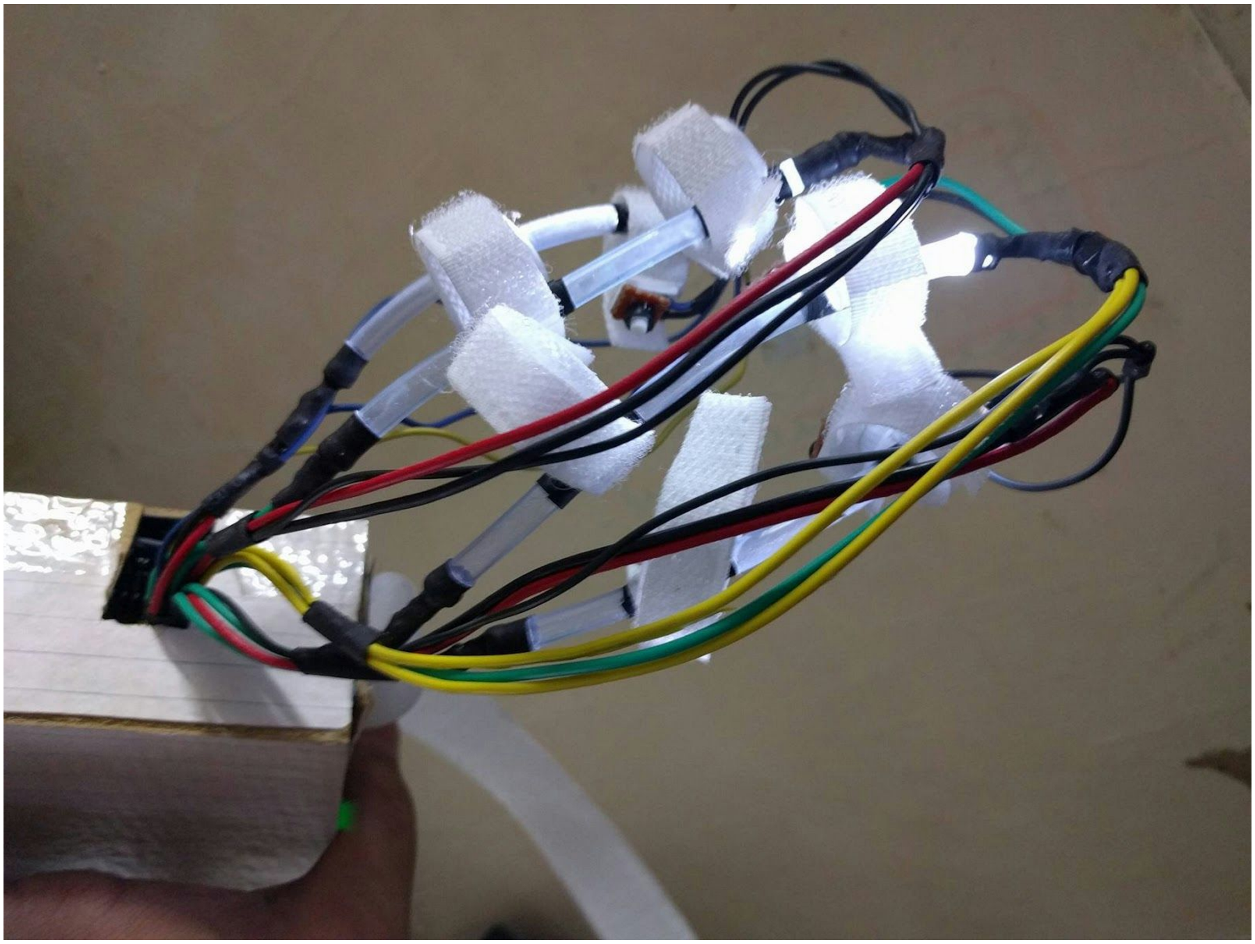
Pinch Sensors

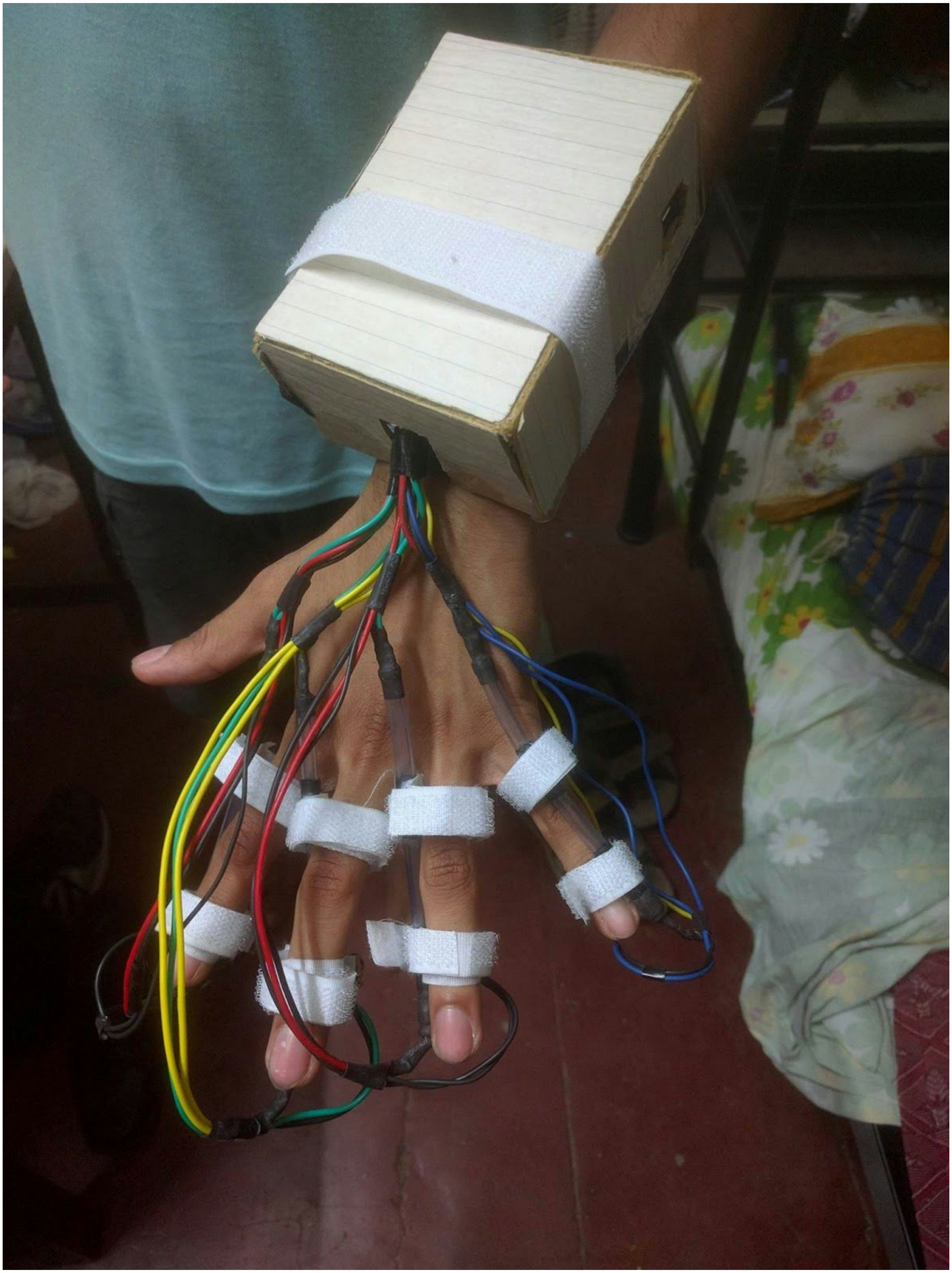
Simple mechanical buttons were used to detect pinch instead of the more costlier option of force sensing resistors

Pictures

A few pictures of the device are shown below.

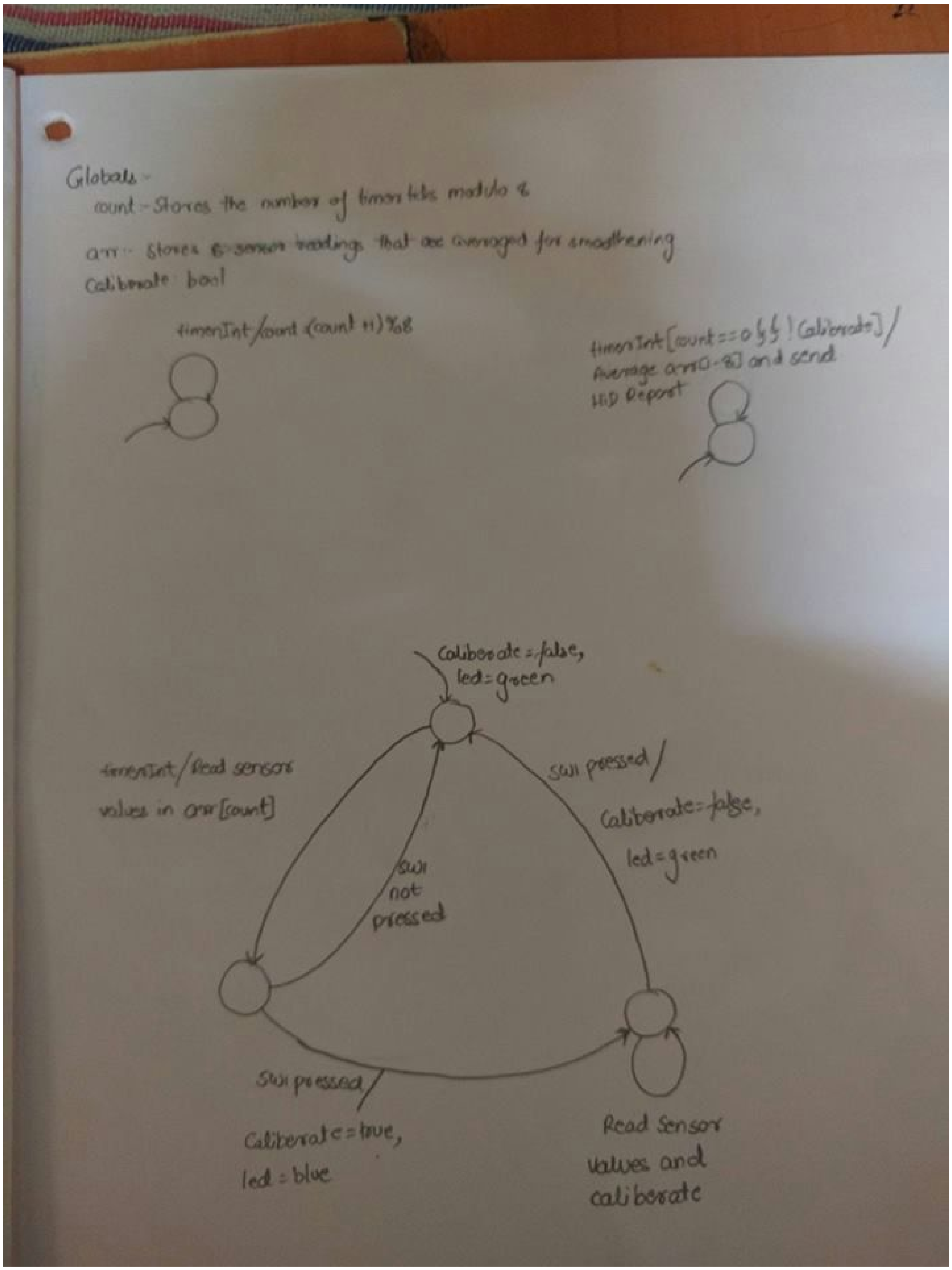






Working of the System

A statechart summarising the working of the system is given below



Sensors

The following modes of communication are used to communicate with the various sensors and buttons.

1. Mechanical buttons: The buttons when pressed ground the digital pins on TIVA board.
2. Flex Sensors: As we want to actually measure the resistance of the LDRs, we build a voltage divider circuit by adding a resistor in series with the LDR and measuring the voltage across the LDR to detect the amount of bending in the sensor and hence in the finger. This voltage is then converted into a digital value using the on-board ADC.
3. MPU: The MPU provides six readings, one for each of the axes. This readings are read using the I2C protocol. The MPU acts as the slave and the TIVA board acts as the master in this protocol. Using this protocol, one reads the 6 values from the registers on the MPU.

Reading the Sensor Values

The sensor values are read at regular intervals of time and sent to the PC as described above. We debounce the sensor values for smooth operation of the sensor and to iron out any minor variations in them. This is achieved by sending the computer values of the sensors averaged over a certain interval of time instead of the raw values. The following code snippet does this averaging.

```
uint32_t tmp = 0;
for (j = 0; j < MAX_AVG_TICK; ++j) {
        tmp += ui32ADC0Value[j][i];
}
tmp /= MAX_AVG_TICK;


int32_t tmp = 0;
for (j = 0; j < MAX_AVG_TICK; ++j) {
        tmp += ui32MPUValue[j][i];
}
tmp /= MAX_AVG_TICK;
```

Deadzone in the Flex Sensors

The resting position of the fingers is not fully straight. As a result, the flex sensors are a bit bent when the fingers are in rest position. To take this into account, we have added a deadzone for each of the flex sensors. Thus, the flex sensors do not work unless they have been bent to a significant extent.

USB Subsystem

The device identifies to the PC as a normal joystick. This is achieved by using the USB HiD protocol. The device sends the PC the values of the various sensors and buttons using this protocol in the form of the USB Report at regular intervals. The structure of this sensor data in the report is as follows.

1. 1 byte each for the four flex sensors
2. 3 bytes each for the MPU data
3. 4 bits for the four buttons on the fingers

Calibration

The device has an inbuilt calibration mode that can be used to calibrate the device in the absence of a calibration software in the PC. As shown in the statechart below, the mode can be activated by pressing a switch on the TIVA board. The LED indicates the mode the device is operating in.

Discussion of System

Components of the Project that Went as Decided

The following components of the project went as decided earlier during the planning stage.

1. Fabricating of the flex sensors: This was an ambitious task as we had no idea what practical problems would be encountered in the fabrication. As an example, we used transparent rubber tubing for the flex sensors and had thought that we would need to provide some sort of reflective coating on the tubing or at elast coat them black. However, the flex sensors worked very well despite us not doing any such thing due to sufficient internal reflection from the tubing.
2. Low fabrication cost of the device: We could successfully keep the fabrication cost of the device below $20. If the product is mass manufactured the cost would further drop to $12-13.

Changes from SRS

The following changes were made during the fabrication of the device from the SRS.

1. Using velcro strips instead of a glove: We could not find a rigid enough glove so that we could attach our sensors and the TIVA board on the it. As as result we decided to use velcro strips to wear the device instead of a glove.
2. Placement of the flex sensors: We had decided to place the flex sensors on the top of each finger. However, as we could not find a elastic rubber tubing of the necessary dimensions, we had to place the flex sensors on the sides of the fingers instead of the top thus sacrificing a bit on user comfort.
3. No flex sensor on the thumb: Having a flex sensor on the thumb would have made it very difficult and cumbersome to put on the device. As a result we had to not have a sensor on the thumb of the hand. Also, the thumb being short, it would have been difficult to bend that short a length of the rubber tubing.

Future Work

The following can be improved upon in the device in the future work.

1. Coming up with better design that improves user experience. For example, this includes, finding better materials and making the glove more robust to wear and tear.
2. Add haptic feedback for the buttons
3. Make the device wirelessly communicate with the PC, possibly via bluetooth.
4. Improve the calibration of the flex sensors for more natural usage
5. Possibly have a similar device on both the hands to provide a lot more controls to the gamer.
6. Use the device for helping physically challenged people to use a cpmputer

Conclusions

Our device thus makes it easy and cheap for gamers to have a better gaming experience. It also opens a gamut of possibilities to improve upon our prototype.

References

1. USB.org - HiD: http://www.usb.org/developers/hidpage/
2. I2C Communication with the TI Tiva TM4C123GXL: https://eewiki.net/display/microcontroller/I2C+Communication+with+the+TI+Tiva+TM4C123GXL
3. MPU 9250 Datasheet: https://store.invensense.com/datasheets/invensense/MPU9250REV1.0.pdf
4. MPU 9250 Register Map: https://strawberry-linux.com/pub/RM-MPU-9250A-00.pdf
5. https://github.com/Rhys79/Launchpad-Mame-Control
6. http://gamasutra.com/db_area/images/blog/234638/Games%20Revenue%20Forecast.PNG
7. Optical flex sensors: http://hackaday.com/2011/10/21/building-optical-flex-sensors/
8. Optical flex sensor (US 4542291 A): http://www.google.co.in/patents/US4542291