Hand Gesture following Firebird V

Course Project Report

Submitted in partial fulfillment of the requirements
for the course of

Embedded Systems

by

Group 18
Kumar Lav 06D05012
Milind Kothekar 06D07017
Prithvi Raj 06D05017
Surinderjeet Singh 06D05021

under the guidance of

Prof. Kavi Arya

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
November 2010

Contents

Chapter 1

Introduction

1.1 Problem Statement

One of the ways in which embedded systems have encroached into our lives is in the way we interact with machines. The ubiquitous computer isdoing away with the traditional modes of human-computer interaction, which were dependent on devices like keyboard and mice, and is becoming more user-centric introducing technologies like touch-screen and gesture recognition. Through this project we hoped to introduce ourselves to one such technology - accelerometer based gesture recognition and to build a robust hardware for the future designers to develop new and interesting projects on.

The aim of our project is to build a system which would allow a user to control the motion of a firebird bot by hand gestures. The hand gestures of a user will be captured by a hand glove worn by the user. The glove, which is mounted with an accelerometer, will transmit data wirelessly to a firebird bot. The microcontroller on the bot will sense the gestures and the motion will be executed as per a set of predefined norms.

Although we implemented this project specific to a firebird bot, the ramifications of this project are plenty. Many future applications are explored in the 'Future Scope' section.

1.2 Requirements specification

Following hardware were needed for this project.

- Firebird V
- ZigBee modules (2)
- Accelerometer MMA 7361LC
- External analog accelerometer interface
- ATMEGA8 microcontroller
- 7805 5V voltage regulator
- TPS7A4533 3.3V voltage regulator
- In System Programming Feature
- Mode control switch

1.3 System Design

Figure 1.1 presents the various states that FireBird V Bot can be in and transitions among them.

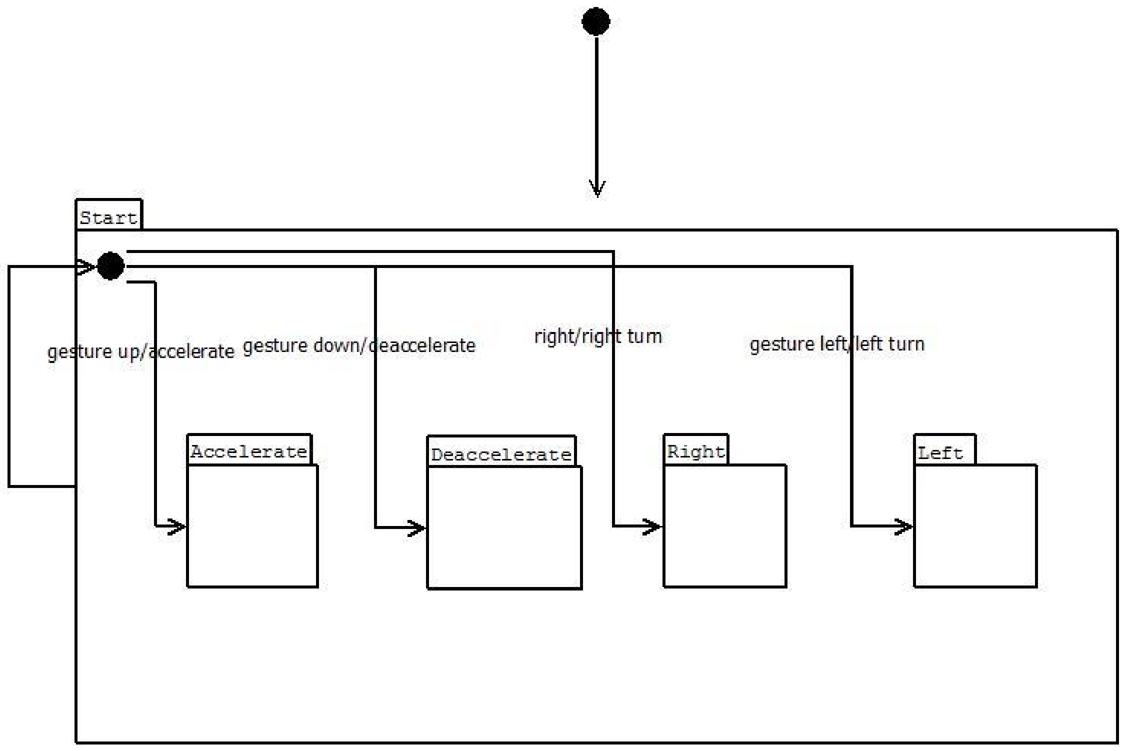


Figure 1.1: Sate Diagram

Figure 1.2 presents the use case for the system

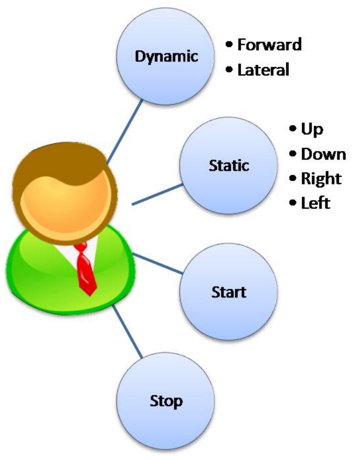


Figure 1.2: Use Case Diagram

Figure 1.3 presents the data flow schema for the system.
Figure 1.4 presents the process flow for the system.

1.4 Assumptions and Limitations

- An obstacle free path is assumed to be present for the robot to traverse.

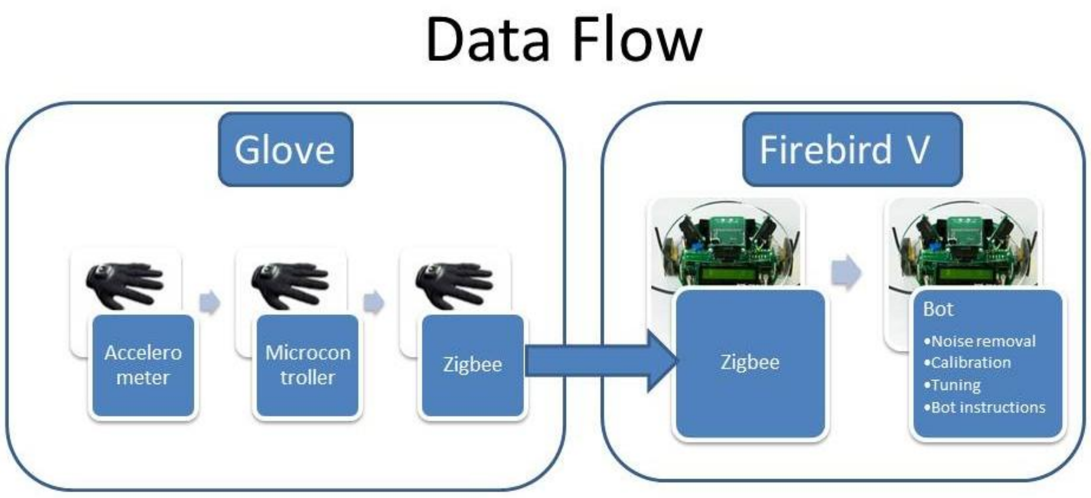


Figure 1.3: Data Flow Diagram

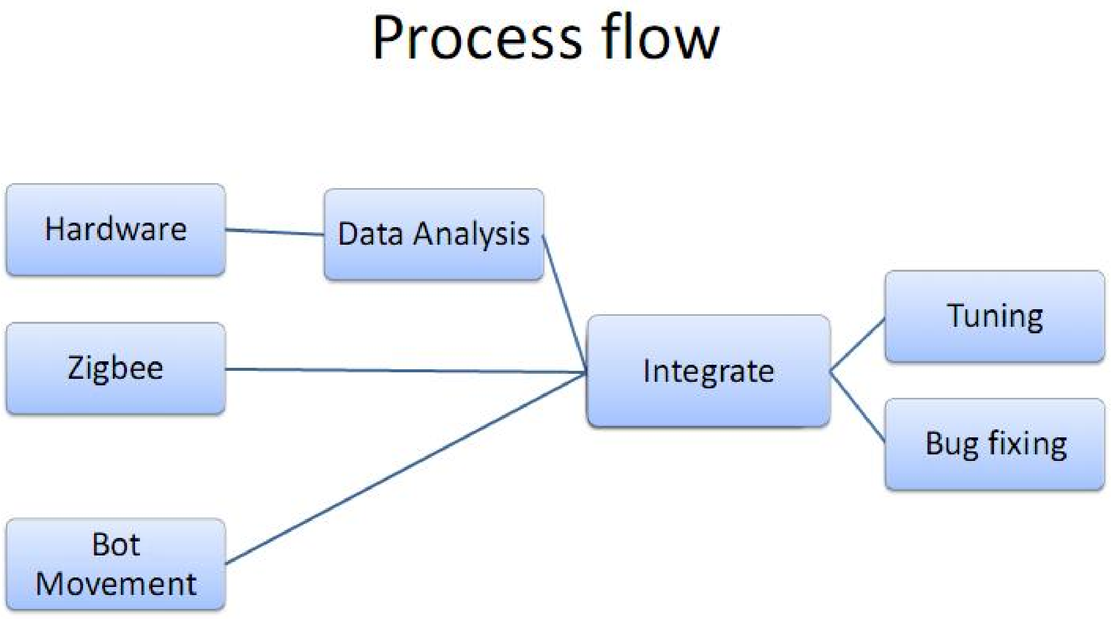


Figure 1.4: Process Flow Diagram

- No other zigbee transmission should be present
- he firebird bot and the glove should be within a suitable distance as allowed by the zigbee modules, for them to be in contact with each other.

1.5 Setup and extensions implemented on the robot

No extensions needed on the robot except a zigbee module for serial communication

1.6 Additional Hardware used

A PCB was designed for the glove with the following hardware.

- AccelerometerMMA 7361LC
- External analog accelerometer interface
- Zigbee communication module
- ATMEGA8 microcontroller
- 7805 5V voltage regulator
- TPS7A4533 3.3V voltage regulator
- In System Programming Feature
- Mode control switch

Chapter 2

Project

2.1 Status

2.1.1 Current project status

The goal of our final design project was to build a vehicle controlled using accelerometers, mounted on a glove that wirelessly transmits data to the vehicle using Zigbee technology, to move in any direction. The accelerometer is mounted on a glove, such that if the users want to move forward they can lean the glove forward and backward for reverse; to turn left or right they need to tilt the glove like a plane changing its direction all in static mode.

In addition there is a dynamic mode during which the vehicle goes in the direction requested by the user for the length of the distance requested by the user. The LCD located on the FIREBIRD displays the direction and mode (Static or dynamic), which is set using the switch.

We decided to go ahead with this project because of its ingenuity and similarity to game console systems games, which will appeal to users especially with the ease of use (physically moving the remote control). One of the main motives behind the project was to build a robust hardware for the future designers to develop very interesting projects on. Additionally, we all love playing with remote controlled vehicles and we wanted to take it to the next step therefore creating even more enjoyable form of entertainment.

2.1.2 Requirements Completed

The first step was to design the PCB of the Glove. The PCB was designed using EAGLE. It had 4 basic parts.

- Microcontroller
- Accelerometer
- Wireless Communication
- Power unit

The 1 design phase of the hardware was made keeping in mind all the difficulties that we might face that we could foresee. There was a provision of

- ACCELEROMETER

  - An onboard accelerometer
    * MMA7361LC
    * 3 Axis Analog Accelerometer
  - A backup digital accelerometer interface
    * MMA7455
    * 3 Axis Digital Accelerometer
    * SPI / I2C interface
    * Connected to microcontroller by 6 pin relimate connector
  - An extra analog accelerometer interface
    * Connect any analog 3 Axis accelerometer from freescale having compatible pin out

- COMMUNICATION
  - A Zigbee Interface
    * RF based wireless transceiver
    * Range of 100m outdoor
    * Reliable UART based transmission
    * Costs around Rs. 2000
  - A CC2500 chip
    * RF based wireless transceiver
    * Range of 10m outdoor
    * Requires a MAX232 to interface with microcontroller
    * Transmission Frequency: 2.4GHz
    * Costs Rs. 600
  - A wired control override
    * Direct connection to the Firebird using wires

- POWER
  - 5 V power supply
    * 7805 Voltage regulator
  - 3.3 V power supply
    * TPS7A4501: 1.5V to 20V, adjusted using resistors
    * TPS7A4533: 3.3 V fixed voltage Low Dropout Regulator (LDO)

- IN SYSTEM PROGRAMMING FEATURE

  The PCB designed is double sided PTH (Vias and the pads should be connected from top and bottom). The cost of the PCB was Rs.1100 for 3 PCBs.

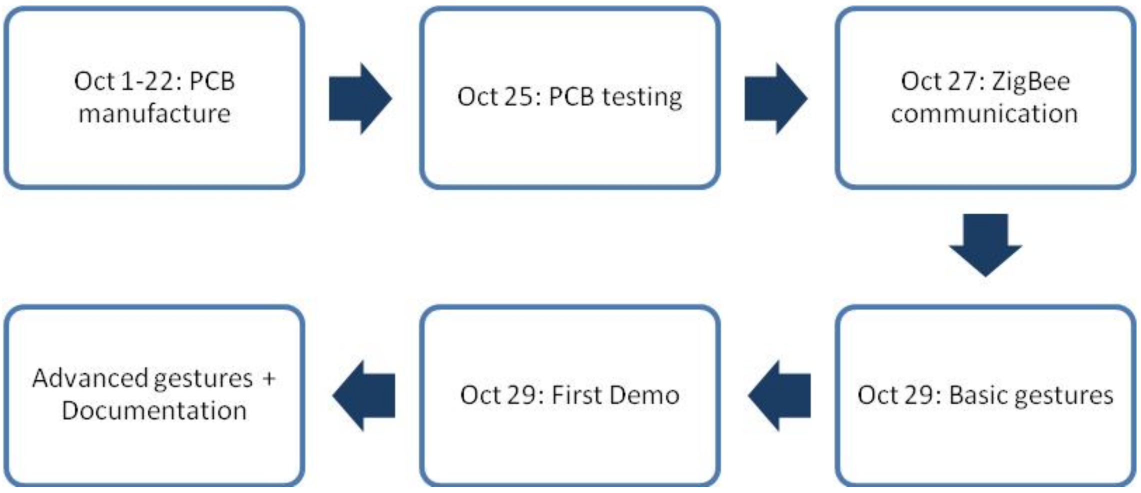

Oct 1-22: PCB manufacture
Oct 25: PCB testing
Oct 27: ZigBee communication
Oct 29: Basic gestures
Oct 29: First Demo
Advanced gestures + Documentation

Figure 2.1: Project Timeline

2.2 Delays, Issues and Critical Steps in the project

2.2.1 Issues

Making a flawless hardware was probably the toughest challenge of the whole project. There were a lot of problems that we faced that needed their root cause to be figured out and modify the hardware accordingly.

1. The VCC value provided to the accelerometer and zigbee was 3.3V. We used in system programming (ISP) feature. The computer programmed the microcontroller on the other hand at 5V. If any of the 3.3V VCC rated chips would have got 5V supply then those would have burnt right on the spot. To solve this problem we first cut the tracks that connected VCC of microcontroller to VCC of Accelerometer. We took a pin out from both VCC of microcontroller and VCC of the accelerometer and connected them after we programmed the microcontroller while disconnecting them at the time of programming.

2. We used TPS4501, a voltage regulator from TI to get a constant 3.3 voltage supply. But it did not give a fixed 3.3 volts as its voltage range was from 1.5V to 20V and depended upon the resistors used in the circuit very heavily. Hence we modified the circuit in such a way that we could use the TI's TPS4533 voltage regulator which gave us fixed 3.3 V voltage supply. We had to short one of the resistors and open the other to achieve this. This way we had our 3.3V fixed supply.

3. The next problem was to make the accelerometers work. There was an analog accelerometer on-board and a digital accelerometer that could be interfaced with a 6 pin connector. First we tried to make the analog accelerometer work. But it did not work. There was probably some problem with the soldering and the problems were not fixable as the accelerometer had a very difficult to test package. It was not really a plug and play kind of device. The probable causes were that the layout guidelines of making a symmetric layout were not strictly followed. Also the accelerometer was soldered at a local mobile shop and not by a using auto aligned machine soldering. Hence there were a lot of chances for errors which surely kicked in making our first on-board accelerometer fail.

4. Next we moved on to try and make the digital accelerometer work. Probably, due to similar soldering reasons and layout guideline problems the digital accelerometer also refused to work. The digital accelerometer had SPI as well as I2C interface. We made the hardware connections for the I2C interface. We had the code ready most of which was the standard I2C protocol. We still could not gather any data from the digital accelerometer. We tried 2 more digital accelerometer none of which provided any successful results. Here, both our digital and analog accelerometers which were the heart of our project were not working.

5. Next we bought the commercial off the shelf soldered accelerometers from Nex-Robotics and tried to make them work. They were quite easy to use, plug and play kind of devices that were made out of the same accelerometers that we used; MMA7361C to be precise. We made a circuit on the breadboard and worked with the same code which we wrote for the onboard accelerometers. Then we tested them by sending the data from the UART of the microcontroller to the serial port of the computer. Not surprisingly, the accelerometer worked giving us very precise data on the computer using the HyperTerminal. First we gathered the data for just one axis and tried to plot it using MATLAB. The serial input data could be plotted in MATLAB against time and showed us the actual variation of the acceleration graphically making it much easier to understand the data. The plot of data from all the three axes was a bit difficult to plot as it required a precise filtering and sequencing of the data.

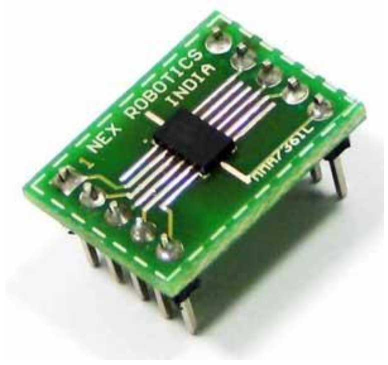


Figure 2.2: Accelerometer

6. The circuit on the breadboard was difficult to handle. We could not move it around nearly as freely as we could move the PCB. Hence we shifted the circuit to a General Purpose Circuit Board and designed a small convenient circuit to be interfaced with the original PCB. There were 3 ADC pins still available on the microcontroller, Pinouts of which were taken just in case we needed to use an external accelerometer. The design consideration worked and we could interface an external accelerometer to the microcontroller. That circuit gave no problem and is still working perfectly fine providing reliable data all the time.

7. The 5V-3.3V mismatch between the computer and the microcontroller was getting too tedious to deal with as we had to disconnect and reconnect all the 3.3V VCC supplies from time to time. That resulted in waste of time as well as left the probability for destruction of the hardware. Hence, we removed the 5V pin from the

computer to the microcontroller, taking 3.3V supply as our supply to microcontroller all the time. This resulted in an enormously simpler system to deal with and improved the work rate exceptionally.

2.2.2 Delays

- We had to get the PCB made from a manufacturer outside IIT
- The manufacturer needed multiple visits and took more time than expected, resulting in some delay in the beginning of the project but we were able to compensate for it by putting in late hours later on
- The PCB based on the final and improved design will take 12 days to manufacture. The design is ready but the final build will not be ready for the final demo.

2.2.3 Critical steps in the project

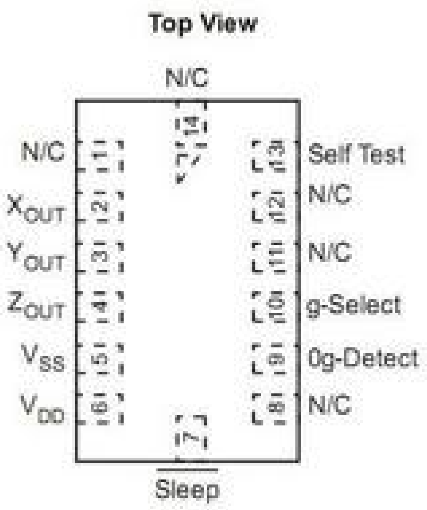
- Getting the PCB ready for the accelerometer module was the most crucial aspect of the project
- Also important was to come up with a basic design for the PCB and then the many reworks of it to reduce its complexity and improve efficiency
- Getting the accelerometer module (through the Zigbee on the PCB) to communicate with the Firebird was also an important aspect of the project

2.3 Components

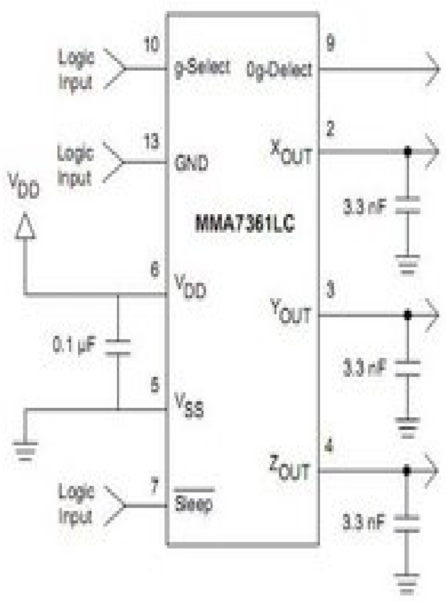
2.3.1 Accelerometer

The accelerometer we used was MMA7361LC which is a 3 axis analog accelerometer which comes in 3mm x 5mm x1mm LGA-14 package. It consumes very low current (400 μA) and works at 2.2-3.6 V. It has a high sensitivity of (800 mV/g @ 1.5g). The sensitivity can be selected ($\pm$1.5g, $\pm$6g). The 3 axis accelerometer contains an onboard single-pole switched capacitor filter. Because the filter is realized using switched capacitor techniques, there is no requirement for external passive components (resistors and capacitors) to set the cut-off frequency. Figure 2.3(b) shows the pin connections of accelerometer.

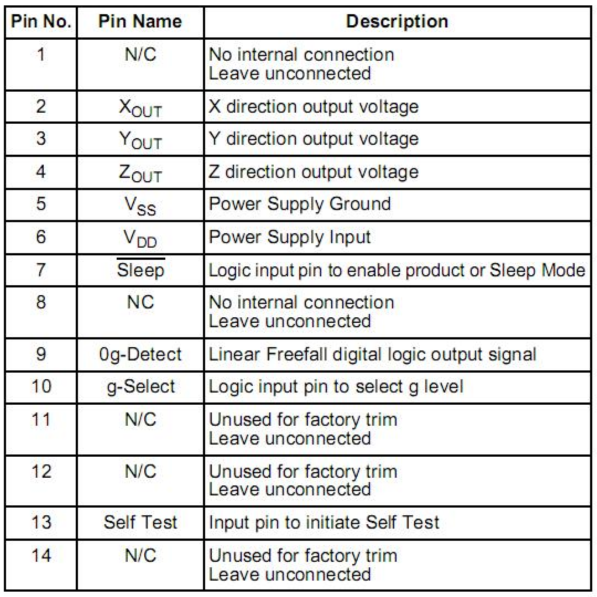
The accelerometer gives output as analog voltage of 800mV/g when configured at 1.5g sensitivity. We convert the analog voltage into digital values using the microcontroller ADCs. We sample all three axes one after the other in the same order at the frequency of approximately 100Hz. We feed the data to the ADC of the microcontroller thus getting the digital output that we can work with. The data thus generated is sent to the firebird robot via Zigbee. A photograph of the final accelerometer which was soldered on a circuit board is shown in Figure 2.4.



(a) Pin Connections



(b) Accelerometer with recommended connection



Pin No.	Pin Name	Description
1	N/C	No internal connection Leave unconnected
2	X_{OUT}	X direction output voltage
3	Y_{OUT}	Y direction output voltage
4	Z_{OUT}	Z direction output voltage
5	V_{SS}	Power Supply Ground
6	V_{DD}	Power Supply Input
7	$\overline{\text{Sleep}}$	Logic input pin to enable product or Sleep Mode
8	NC	No internal connection Leave unconnected
9	0g-Detect	Linear Freefall digital logic output signal
10	g-Select	Logic input pin to select g level
11	N/C	Unused for factory trim Leave unconnected
12	N/C	Unused for factory trim Leave unconnected
13	Self Test	Input pin to initiate Self Test
14	N/C	Unused for factory trim Leave unconnected

(c) Pin Description

Figure 2.3: Accelerometer

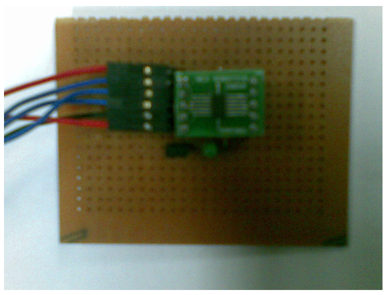
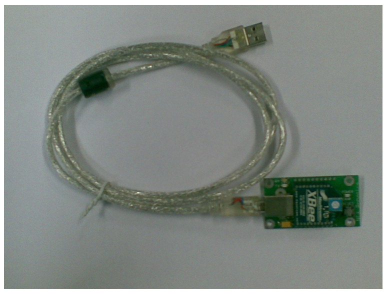


Figure 2.4: Accelerometer soldered on the board

2.3.2 Zigbee Communication Module

(a) Zigbee module

(b) Zigbee Adaptor Board

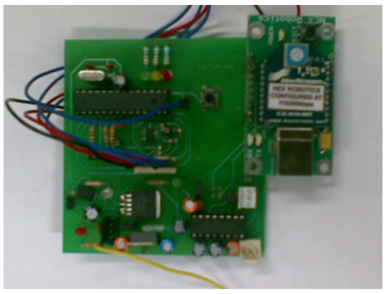


(c) Zigbee for testing

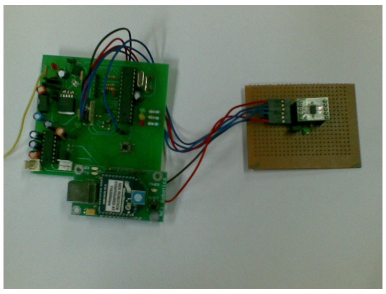
Figure 2.5: Zigbee

2.3.3 The Glove Hardware

Figure 2.6 shows the photographs of the hardware of the glove. In the top left corner is the microcontroller and its peripherals. Top right corner is zigbee module. Bottom part is the power circuitry and an interface for CC2500. Next shown is the hardware of glove interfaced with the accelerometer making it a complete Glove.



(a) Glove Circuit



(b) Glove circuit integrated with accelerometer

Figure 2.6: Glove harwdare

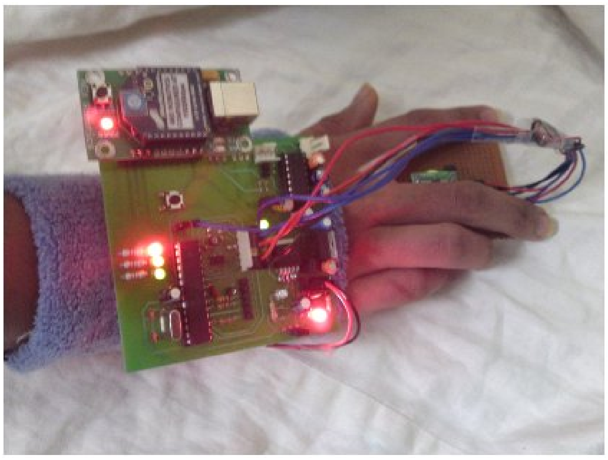


Figure 2.7: Glove

2.3.4 The Programmer

Figure 2.8 Shown here is the Programmer that we used to program the microcontroller ATMEGA8 on the glove. It is a standard USBASP programmer, schematic and PCB of which are easily available in the open source domain.

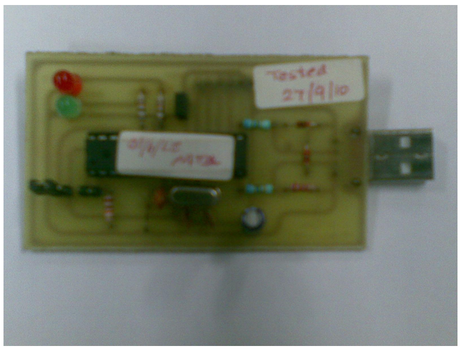


Figure 2.8: Programmer

2.3.5 The Microcontroller ATMEGA8L

We used the microcontroller ATMEGA8L for the glove control. The function of the microcontroller was to sample the data and to send the data to the firebird using zigbee. The microcontroller has 6 ADC channels. We already used three channels for the on board accelerometer. Hence we were left with the other three channels ADC3 – ADC5. We interfaced the accelerometer with these 3 channels and sampled each channel at the frequency of 33 Hz. In the final PCB, we are using all the 6 channels using 3 channels for on board accelerometer and 3 for another accelerometer that can be interfaced externally. The microcontroller was low power consuming hence gave more time of operation without needing recharging. It was AVR based microcontroller, a family of microcontrollers with which we are familiar. Also it worked on 3.3V-5V a range which was suitable for operation with the accelerometer as well as to use In System Programming. All these were reason enough for us to choose the ATMEGA8L as our Glove microcontroller.

2.3.6 Schematics

This is the board with component specifications and placements of the hand glove that we initially designed as our first prototype. Figure 2.9 and Figure 2.10 shows the Layout and Schematic of the same.

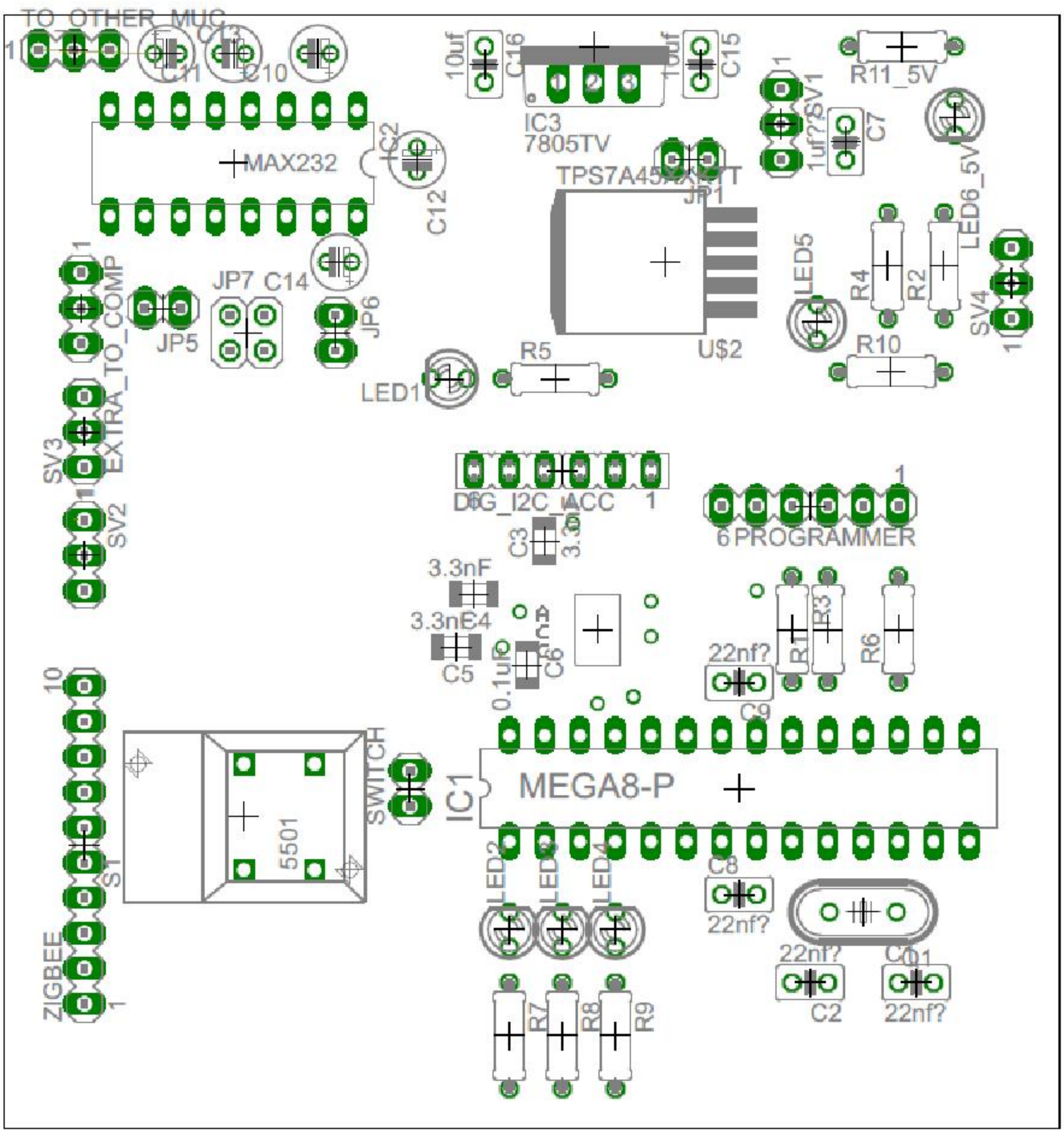


Figure 2.9: Design-1 Layout

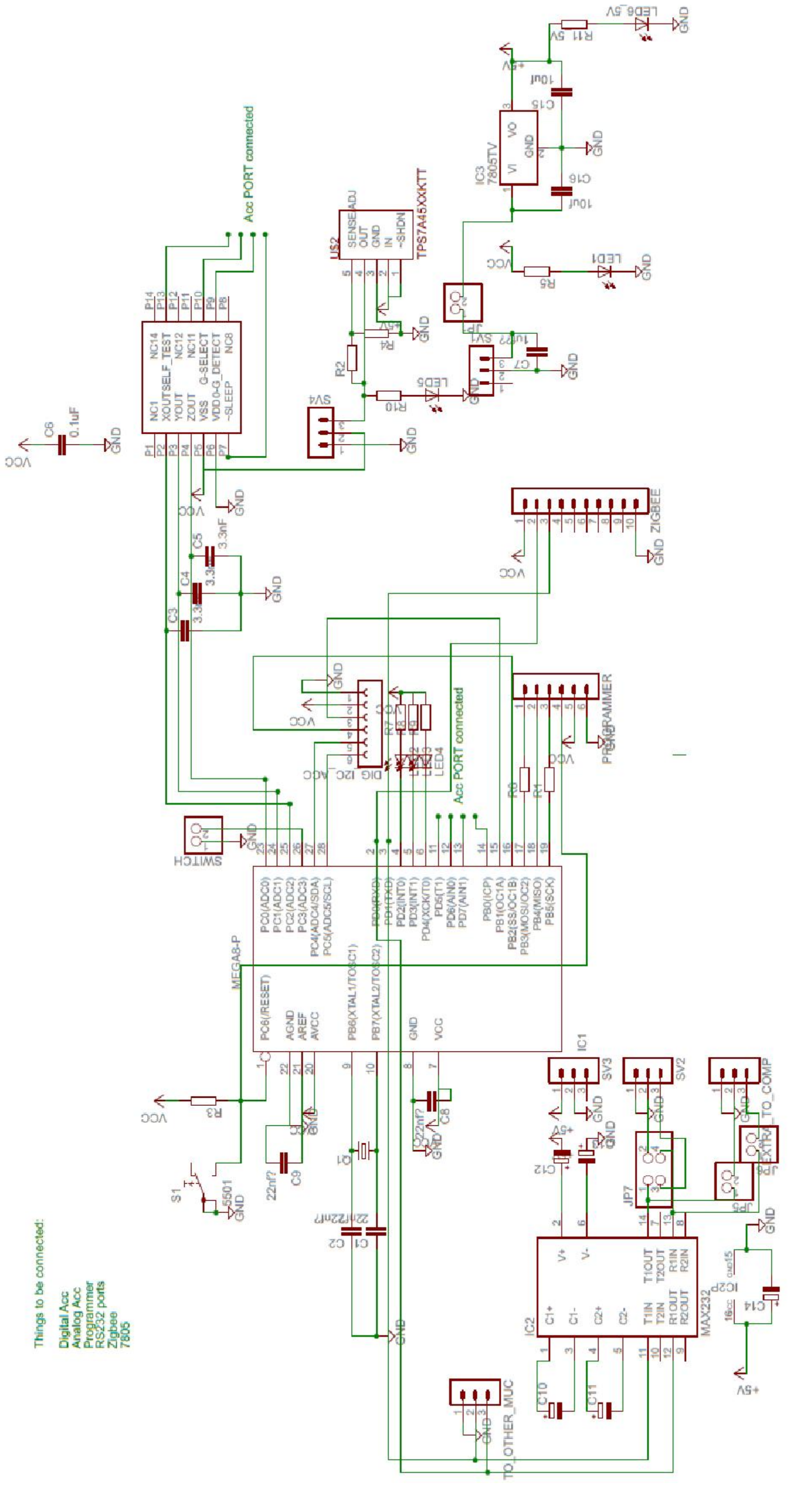


Figure 2.10: Design-1 Schematics

2.3.7 FINAL HARDWARE

The final hardware is much simpler, much smaller version of initial hardware. It is just a (2.5'x2.5') board that can easily be mounted on our hand glove. It consists of

- Analog accelerometer interface (MMA7361LC)
- External analog accelerometer interface
- Zigbee communication module
- ATMEGA8 microcontroller
- 7805 5V voltage regulator
- TPS7A4533 3.3V voltage regulator
- In System Programming Feature
- Mode control switch

Figure 2.11 shows the final PCB of the board that will soon be fabricated with the component placements

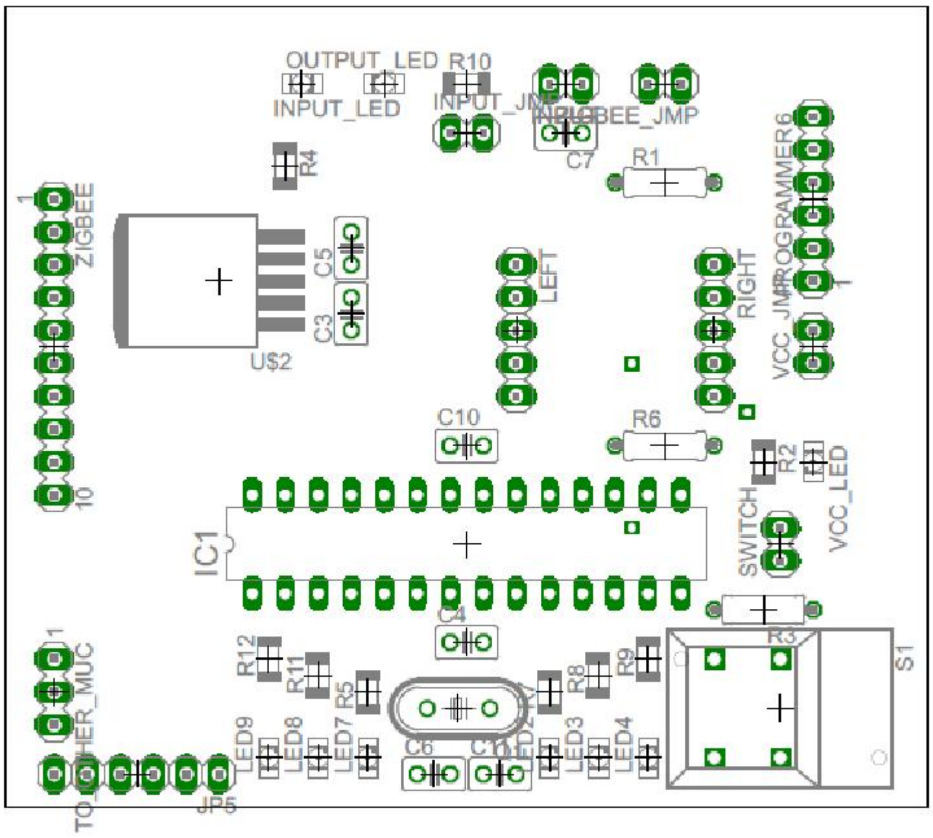


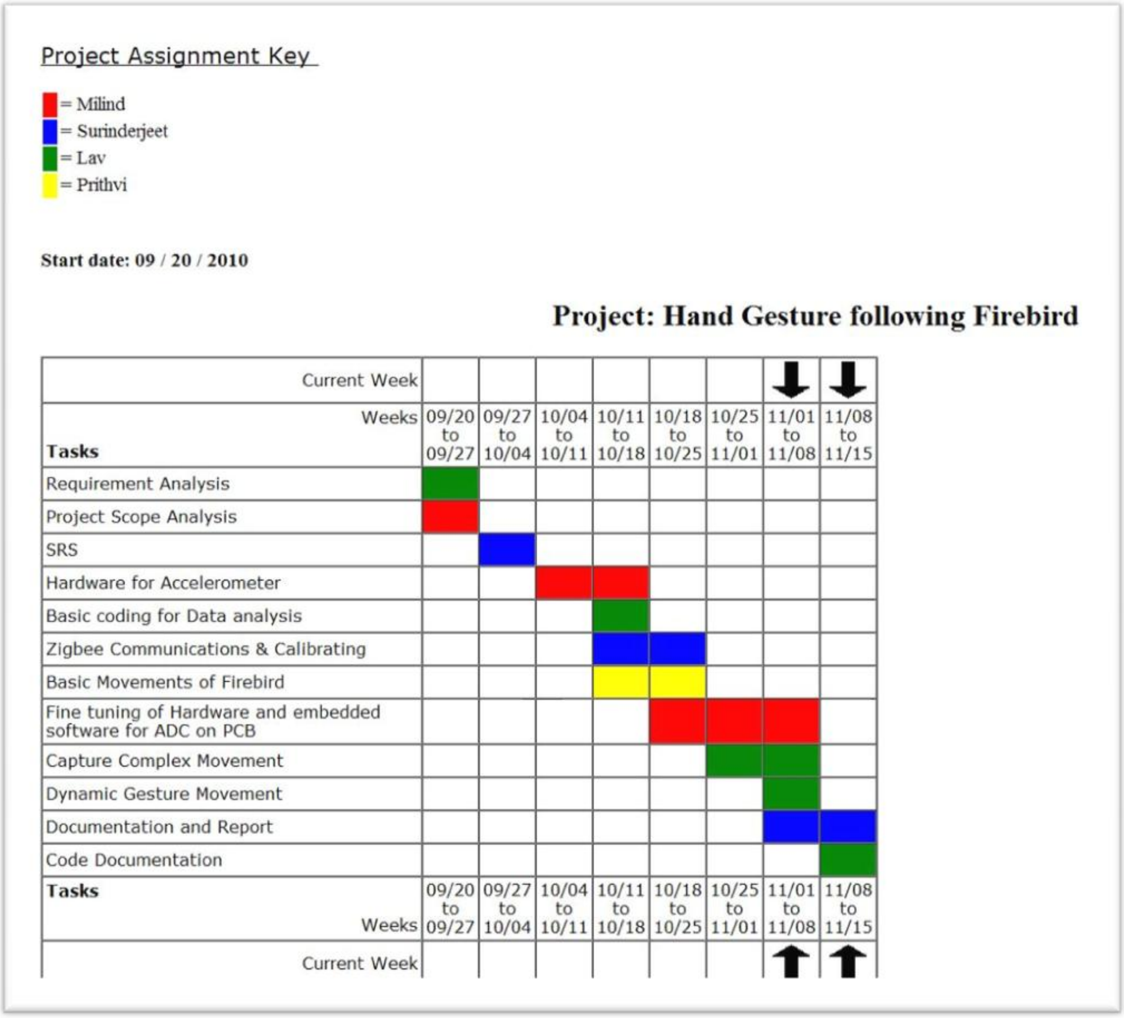
Figure 2.11: Final Schematics

2.4 Individual roles and contributions

Ours was a balanced team with a good mix of students from the Elec and CS department, which was very useful to come up with a good design and implementation for a project meaningful to an Embedded Systems course.

- The idea, design and the hardware implementation of the project were the effort of Milind whose knowledge of electronics were very useful.
- Besides helping in the project discussions, Surinder was involved in the Zigbee communication setup, and writing the code for the same
- Lav was involved in making sense of the data received over Zigbee and classifying them as various gestures meant by the user
- Prithvi took on the task to implement various movements corresponding to various gestures defined

Figure 2.14 shows the project plan and progress.



Project Assignment Key

🟥 = Milind
🟦 = Surinderjeet
🟩 = Lav
🟨 = Prithvi

Start date: 09 / 20 / 2010

Project: Hand Gesture following Firebird

Current Week							⬇	⬇
Tasks / Weeks	09/20 to 09/27	09/27 to 10/04	10/04 to 10/11	10/11 to 10/18	10/18 to 10/25	10/25 to 11/01	11/01 to 11/08	11/08 to 11/15
Requirement Analysis	🟩							
Project Scope Analysis	🟥							
SRS		🟦						
Hardware for Accelerometer			🟥	🟥				
Basic coding for Data analysis				🟩				
Zigbee Communications & Calibrating				🟦	🟦			
Basic Movements of Firebird				🟨	🟨			
Fine tuning of Hardware and embedded software for ADC on PCB					🟥	🟥	🟥	
Capture Complex Movement						🟩	🟩	
Dynamic Gesture Movement							🟩	
Documentation and Report							🟦	🟦
Code Documentation								🟩
Tasks / Weeks	09/20 to 09/27	09/27 to 10/04	10/04 to 10/11	10/11 to 10/18	10/18 to 10/25	10/25 to 11/01	11/01 to 11/08	11/08 to 11/15
Current Week							⬆	⬆

Figure 2.12: Gantt Chart

2.4.1 Time

Each member of the team put in an effort of 10-12 man days

2.5 Demonstration - Live demo + Video

The team will record its final video demo on Wednesday as per the schedule announced.

2.6 Final Roadmap of Project

1. **The roadmap to completion** A detailed roadmap and documentation on deliverables is presented in the above sections and project documentation.

2. **Deadlines for each activity** A detailed timeline for the project is presented in section 2.

2.7 Innovation, Creativity and Reusability Index of your Project

2.7.1 Innovations in project

The very idea of the project is very innovative. It provided for the functionality that state of the art technology products are delivering. Our team has designed, coded and provided for documentation of a gesture following module which could be basis for numerous future projects.

2.7.2 How you have enhanced reusability in project

1. We have documented in detail the design of our accelerometer circuit

2. We have also documented in detail the obstacles we faced and decisions we took to resolve those mistakes, giving an insight to the new user as to how we came about the decisions we made

3. We have commented and documented all the code we have written and explained in detail how to go about executing it

4. We have taken measures to keep our code modular, so that a user can reuse bits of code he finds useful

2.8 Help us in improving the process

2.8.1 What you think can be improved in terms of project activities

he students can be given the idea about the project in the introductory lectures itself and they can thus get more time for ideating the project.

2.8.2 Any comments on the current schedule of events

Keep the students aware of the nest stage deadline, after one stage finishes, so that they don't become complacent.

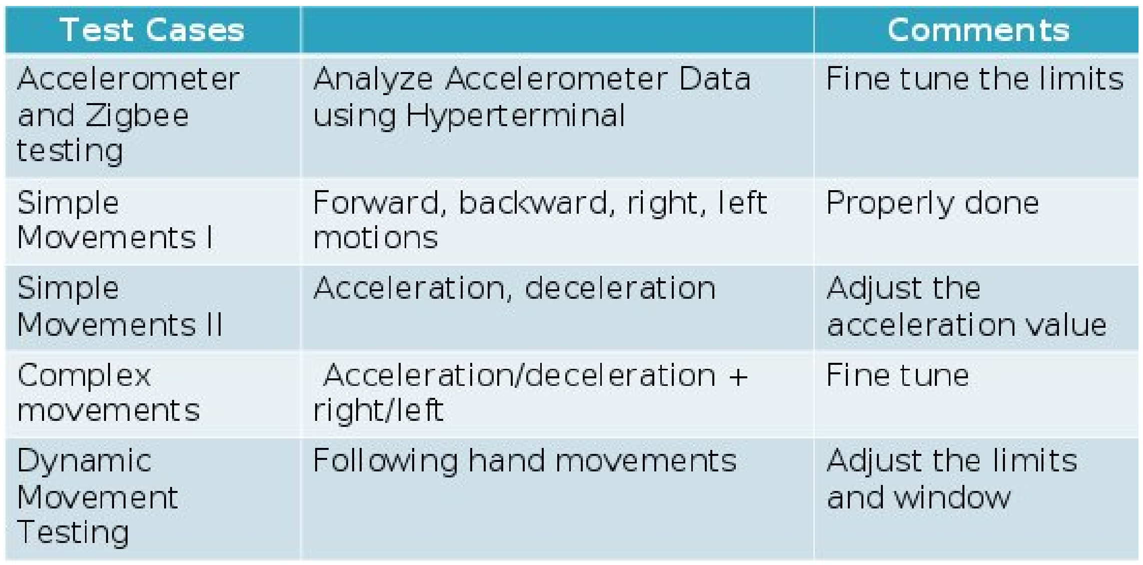
2.8.3 Are you satisfied with the way the course activities have gone – specially the project?

Yes. The project was conducted very methodically making us first acquainted to all the basics of firebird and microcontrollers. The firebird allowed us to work smoothly without too many difficulties and precious time being spent in the hardware of the car. Overall it was a very rigorous and refreshing experience to build something innovative of our own that worked!

2.8.4 Any other suggestions?

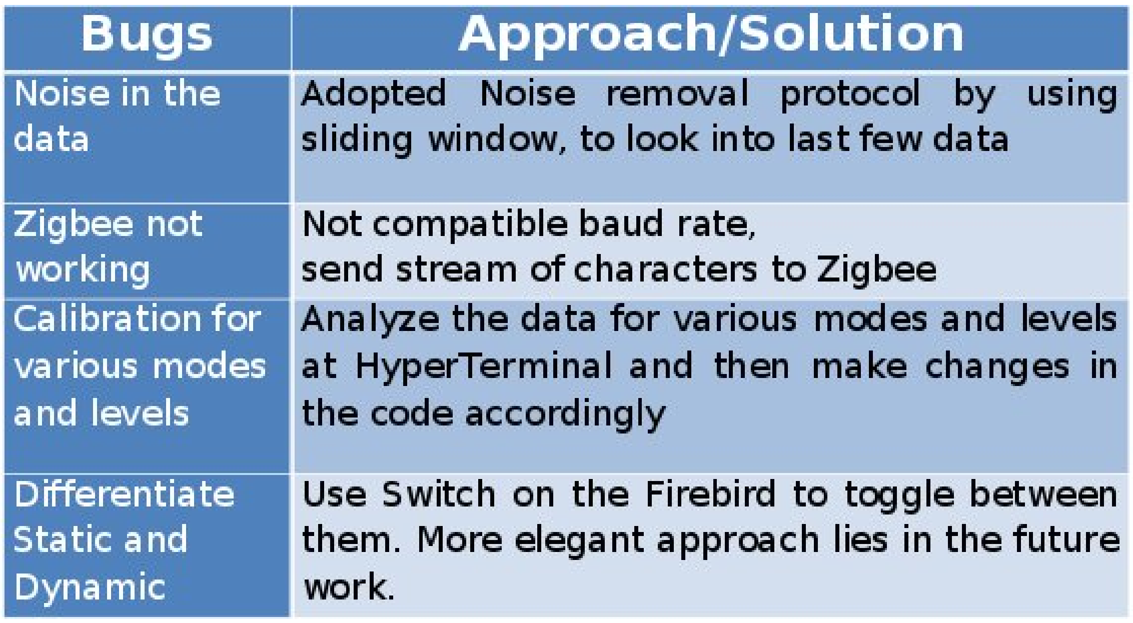
Be more flexible with deadlines instead of giving near hard deadlines for uploading.

2.9 Test and Bug Report



Test Cases		Comments
Accelerometer and Zigbee testing	Analyze Accelerometer Data using Hyperterminal	Fine tune the limits
Simple Movements I	Forward, backward, right, left motions	Properly done
Simple Movements II	Acceleration, deceleration	Adjust the acceleration value
Complex movements	Acceleration/deceleration + right/left	Fine tune
Dynamic Movement Testing	Following hand movements	Adjust the limits and window

Figure 2.13: Test Plan and Report



Bugs	Approach/Solution
Noise in the data	Adopted Noise removal protocol by using sliding window, to look into last few data
Zigbee not working	Not compatible baud rate, send stream of characters to Zigbee
Calibration for various modes and levels	Analyze the data for various modes and levels at HyperTerminal and then make changes in the code accordingly
Differentiate Static and Dynamic	Use Switch on the Firebird to toggle between them. More elegant approach lies in the future work.

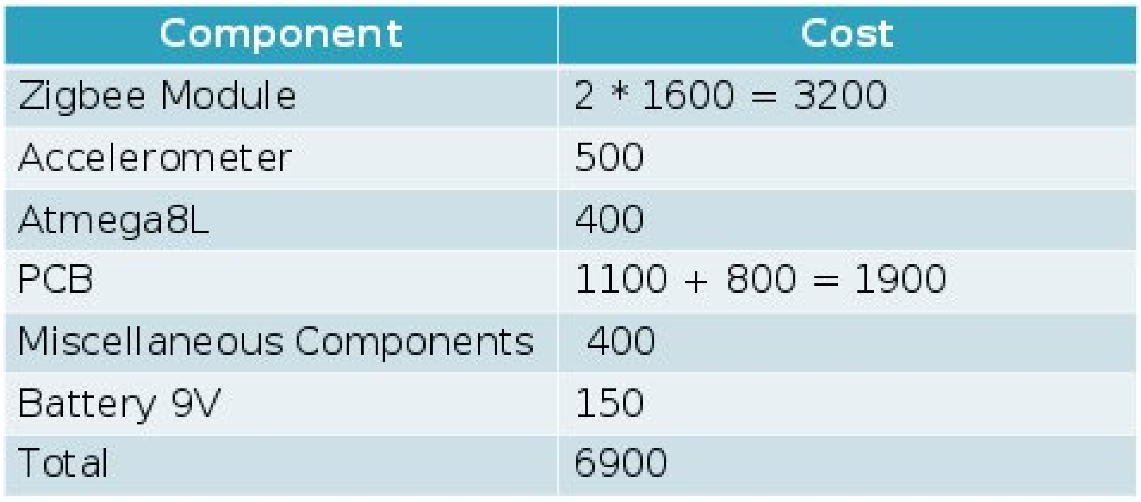
Figure 2.14: Bug Report

2.10 Future Scope

The glove can be a base to many different innovative ideas to be implemented on the Firebird or on computer by future teams. This is as intuitive as robot controlling technology can get. We suggest a few ideas that can be implemented using this hardware

1. Robotic hand
2. Pen that can write in thin air
3. Hand Motion controlled Firebird (Different from Gesture Controlled)
4. Gesture Controlled Hexapod
5. Inertial Navigation System
6. Intuitive Spy Bots
7. Intuitive Surveillance Bots
8. Hand Gesture based Computer Games
9. Hand Gesture controlled mouse
10. Hand motion based Aircraft Simulators

2.11 Cost



Component	Cost
Zigbee Module	2 * 1600 = 3200
Accelerometer	500
Atmega8L	400
PCB	1100 + 800 = 1900
Miscellaneous Components	400
Battery 9V	150
Total	6900

Figure 2.15: Cost

2.12 Code Overview

Figure 2.16 shows the high level overview of the code.

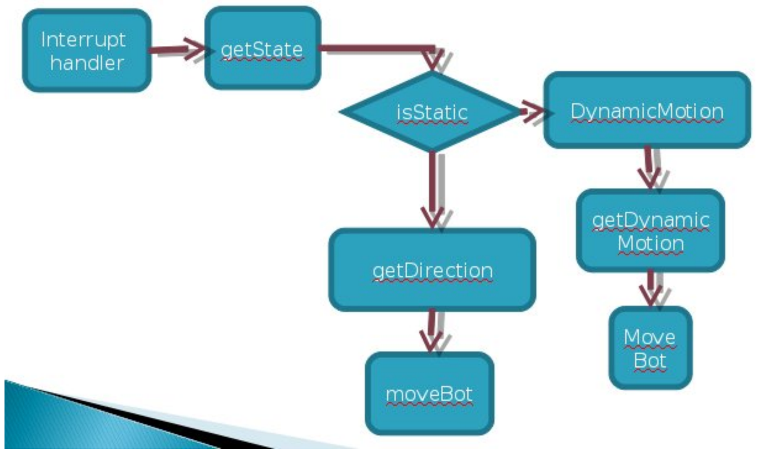


Figure 2.16: Cost

2.13 Learnings

- Hardware interfacing
- Zigbee transmission
- To describe behavior of systems using State charts, and other state diagrams