CS-308-2014 Final Report

TU - 06 (Transformers)

<u>MoodiFire</u>

Aman Mangal, 100050015

Jayanth (leader), 100050041

Pulkit Maheshwari, 100050043

Kumar Rahul Ranjan, 100050038

Table of Contents

1. Introduction

Human face recognition have been conducted vigorously. Such studies are aimed at the implementation of an intelligent man-machine interface. Especially, studies of facial expression recognition for human machine emotional connection are attracting attention. The shape (static diversity) and motion (dynamic diversity) of facial components such as the eyebrows, eyes, nose and mouth manifest expressions. This project involves design and implementation of facial expression recognition bot. Based on the facial expressions and gestures, the bot performs certain tasks.

An interesting use case would be in a hospital where all the patients can be monitored in real time using such a system. We can get updates using a Kinect sensor. The video feed can be processed to figure out the mental state of the patient. Based on the state, various medicine, anti-depressant can be given to the patient. Such a system can also be used in vehicles as well where the state of the driver can be analysed using a Kinect sensor. Based on the state of the driver, various settings of the car can be controlled in an autonomous fashion. For example, a drunk driver is not supposed to drive the car. An angry driver can be stopped from using horn too frequently to avoid noise pollution etc.

In this report, we present an embedded system which can deliver object placed in an arena at known co-ordinates based on the the emotions of a person residing in front of a Kinect sensor. The report is structured as follows. Section 2 defines the problem we are trying to solve, more formally. Section 3 contains all the technical, functional requirements. We describe detailed design and working of the system in Section 4 and 5 respectively. We wrap up in Section 7, 8 with future work and conclusion.

Definitions, Acronyms and Abbreviations

- *SDK : Software Development Kit*
- *USB: Universal Serial Bus*
- *SVN: Apache Subversion software*
- *GUI: Graphical User Interface*

2. Problem Statement

We divide our problem into two parts. First part of the problem is to capture human face expressions and detect the emotions after processing captured video feed. This part of the problem takes input a video feed containing only one person in the video and output the emotions of the person in the video. To decrease the scope of the problem and remove ambiguity in different emotions, we focus on detecting only 6 emotions - default (neutral), happy, sad, anger, surprised, fear.

The second part of the problem is to transfer the detected emotion to a bot, which is free to move in an arena. The bot will pick up objects from a predefined location based on the received emotion and deliver them to the initial position of the bot. The input to this part of the problem is an emotion (or a command representing the emotions of a person) and an initial position in the predefined arena. The bot will, then, move autonomously on the arena, pick object and return back to the initial position. The challenge in solving this part is the autonomous movement of the bot on the arena.

2.1. Product Perspective

This project is a small step in the direction of intelligent interactive bots. The automated bot understands the feelings and expressions of the user.

2.2. Product Function

A Kinect is connected to the laptop and is kept in front of the user while a bot is kept in a line follower 2D space. Based on the facial expressions and gestures, the bot performs different tasks.

2.3. User Characteristics

The user is sitting in front of Kinect, giving different expressions or gestures either based on his need or as per his emotions.

2.4. Constraints

The expressions should be clear enough to be captured by Kinect.

2.5. Assumptions and Dependencies

We kept following assumption in mind while designing the system-

- No wireless signal interference in the arena
- Xbee wireless module has a range over the entire arena
- Expressions are consistent for certain time period

3. Requirements

3.1. Functional Requirements

The requirement for the system is to identify different expressions and based on the expressions, move to specific positions and perform tasks.

3.2. Non-Functional Requirements

The tasks to be performed are suited to different expressions like picking and giving chocolate, if the person is happy, picking and giving anti-depressants, if the person is sad and so on.

3.3. Hardware Requirements

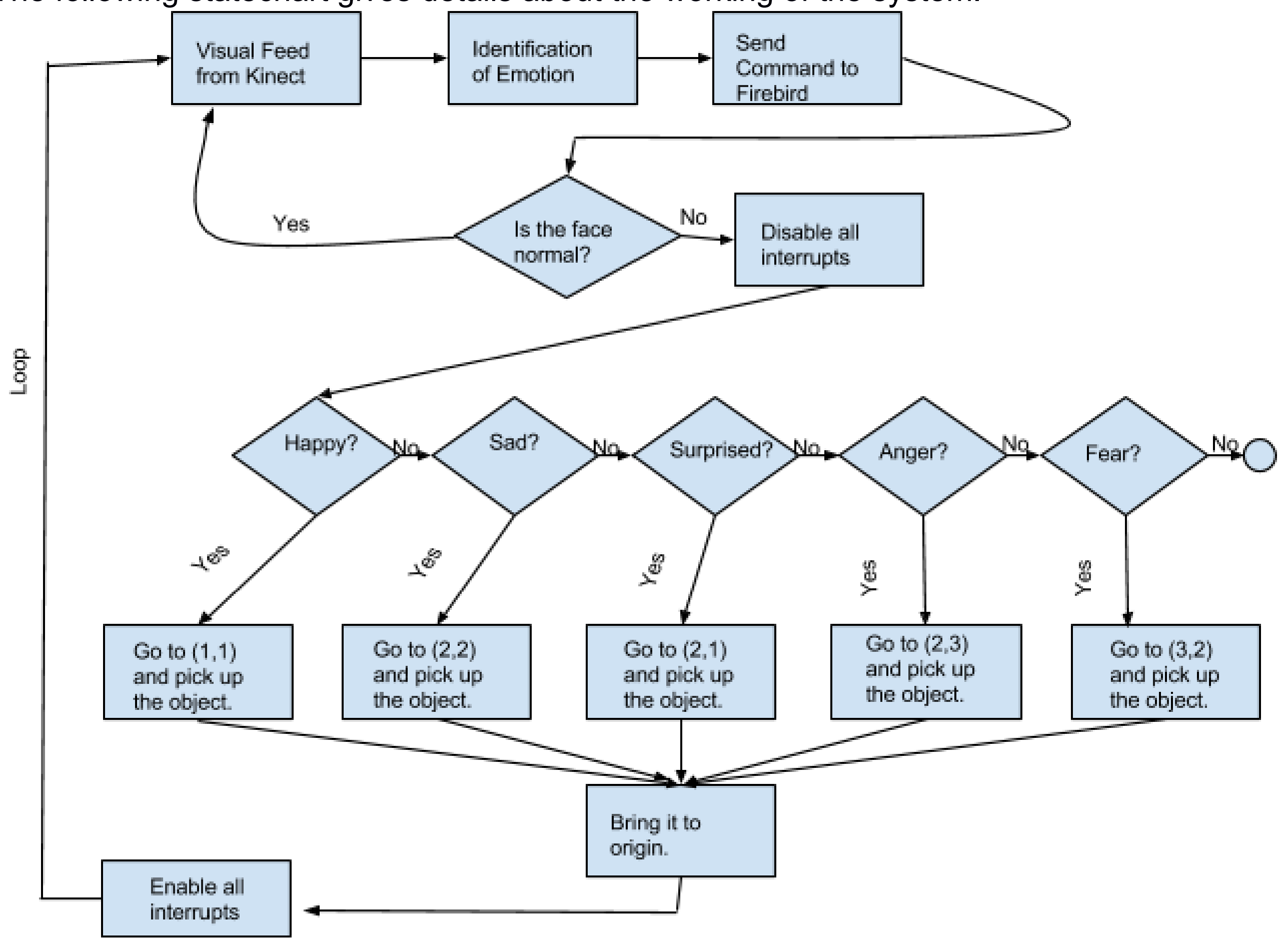
➔ One Firebird V Atmega 2560 robot
➔ A robotic arm attached to Firebird V robot
➔ One Microsoft XBox Kinect connected to a computer through USB
➔ Two XBee modules: one for the robot and another for the computer
➔ Line follower 2D space
➔ Laptop

3.4. Software Requirements

- Keil
- Kinect SDK v1.8
- Microsoft Visual Studio with SP I
- AVR Studio
- WinAVR

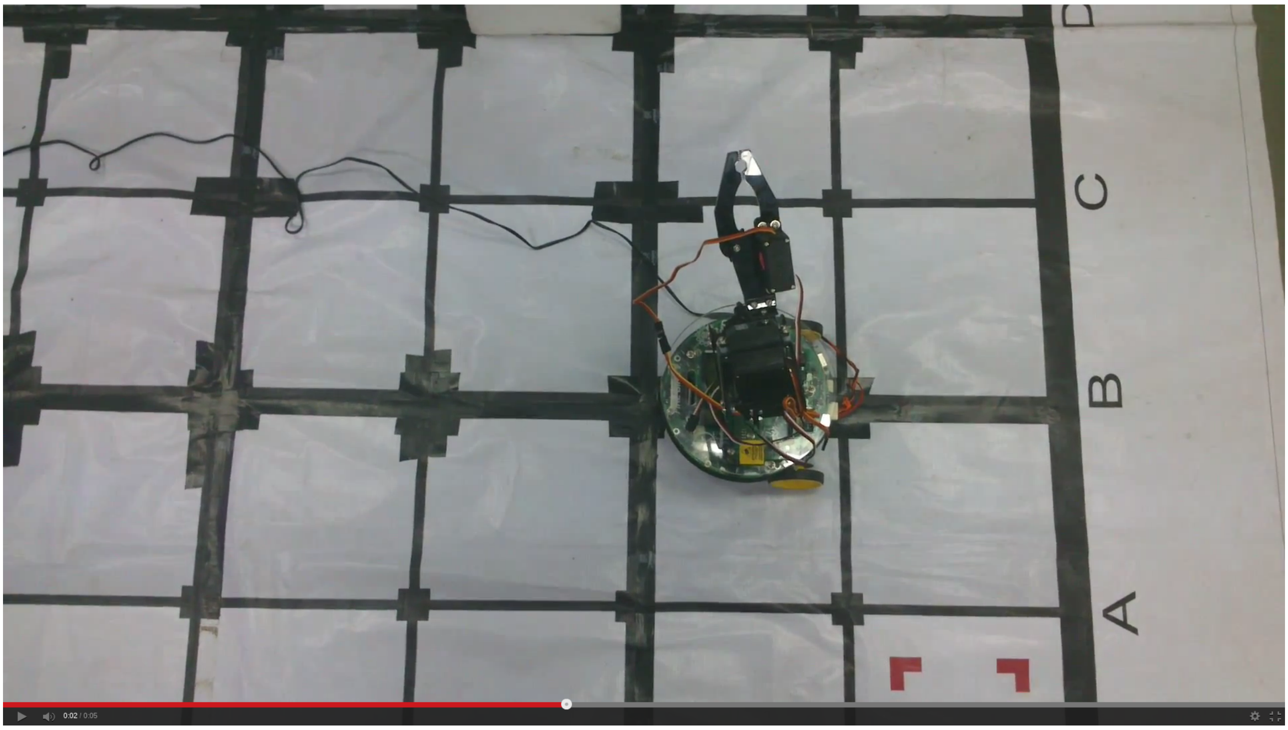
4. System Design

The following statechart gives details about the working of the system.

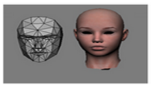
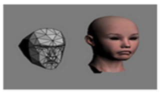
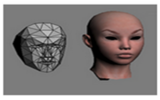


The bot used was Firebird V with ATMega 2560 as microcontroller. The bot was connected to the 6-axis Nex Robotics gripper arm. The gripper servo wire is connected to the servo 1 pin of the bot, the next axis for limb elongation is connected to the servo 2 pin and the last axis for limb elongation is connected to the servo 3 pin.

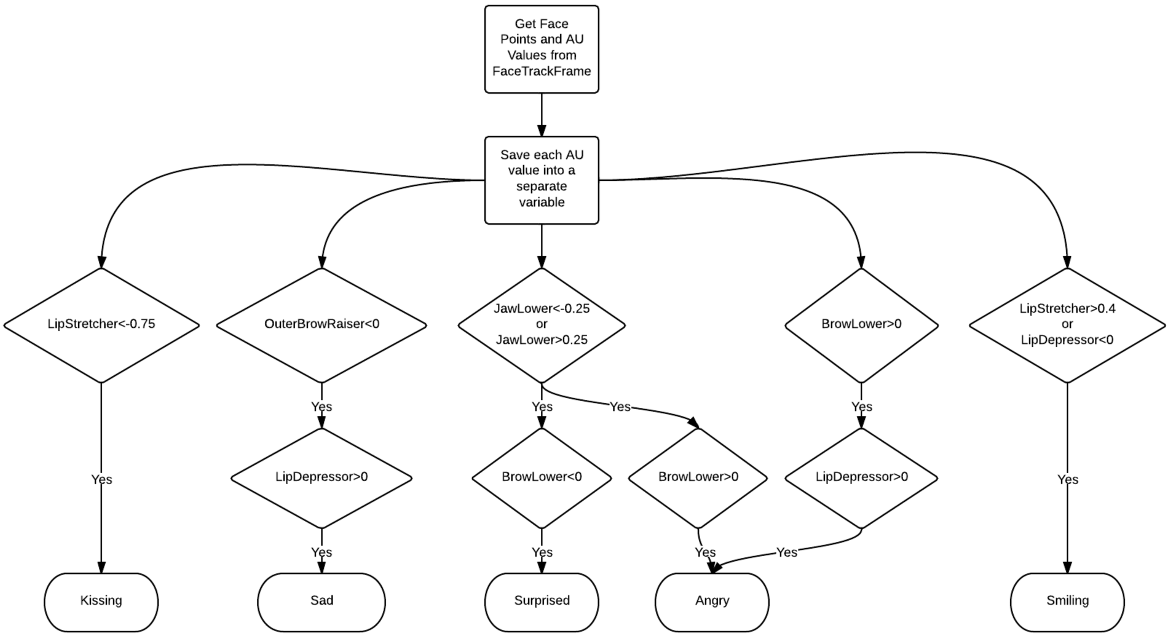
The bot in white line arena was set up like this:



The system uses the AU coefficients to detect the different facial features. The Kinect Face Tracking SDK results are also expressed in terms of weights of six AUs and 11 SUs. The SUs estimate the particular shape of the user's head: the neutral position of their mouth, brows, eyes, and so on. The AUs are deltas from the neutral shape that you can use to morph targets on animated avatar models so that the avatar acts as the tracked user does.
The Face Tracking SDK tracks the following AUs. Each AU is expressed as a numeric weight varying between -1 and +1.

AU Name and Value	Avatar Illustration	AU Value Interpretation
Neutral Face (all AUs 0)		
AU0 – Upper Lip Raiser (In Candid3 this is AU10)		0=neutral, covering teeth 1=showing teeth fully -1=maximal possible pushed down lip
AU1 – Jaw Lowerer (In Candid3 this is AU26/27)		0=closed 1=fully open -1= closed, like 0
AU2 – Lip Stretcher (In Candid3 this is AU20)		0=neutral 1=fully stretched (joker's smile) -0.5=rounded (pout) -1=fully rounded (kissing mouth)
AU3 – Brow Lowerer (In Candid3 this is AU4)		0=neutral -1=raised almost all the way +1=fully lowered (to the limit of the eyes)
AU4 – Lip Corner Depressor (In Candid3 this is AU13/15)		0=neutral -1=very happy smile +1=very sad frown
AU5 – Outer Brow Raiser (In Candid3 this is AU2)		0=neutral -1=fully lowered as a very sad face +1=raised as in an expression of deep surprise

The following algorithm based on different AUs was used to detect the emotions:

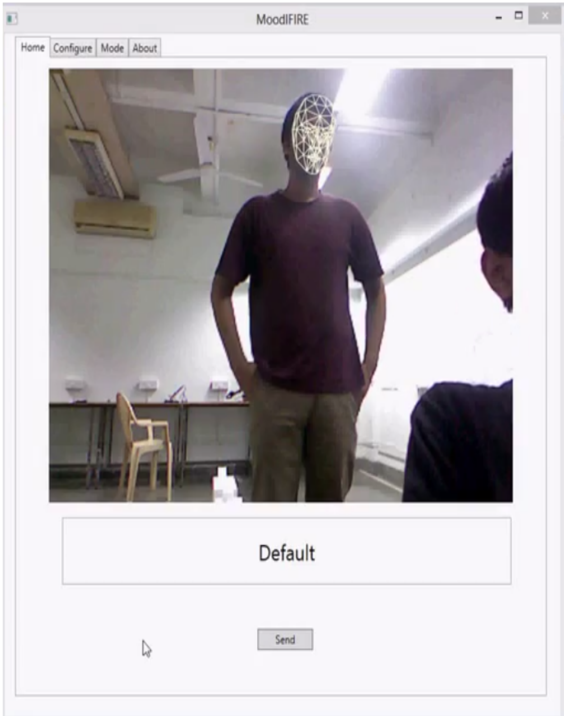


The emotion recognized is also displayed on the lcd screen along with sensor readings.
For path planning in the whiteline arena, we created our own module for coordinate based locomotion developed over the white line following module. In the arena, multiple nodes were created. All the whiteline sensors would indicate black in the node area. In path other than nodes, any two of whiteline sensors would indicate black. We moved firebird from one coordinate to another using the count of nodes it has traversed and by moving right or left such that it would not leave its path. We always ensured the movement of firebird on the path by continuously checking whiteline sensors and moving it right if left whiteline sensor was less than right whiteline sensor or moving it left if right one is less or moving it straight if both are equal.

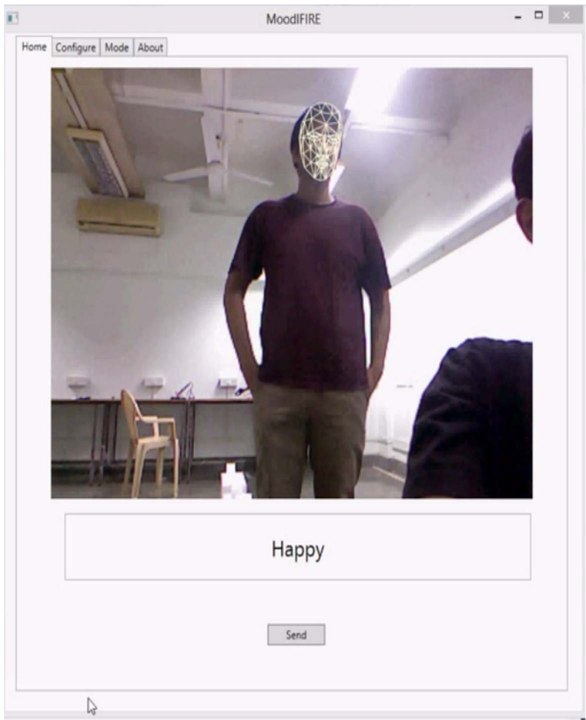
5. Working of the System and Test results

The following six expressions were detected using the Kinect input.

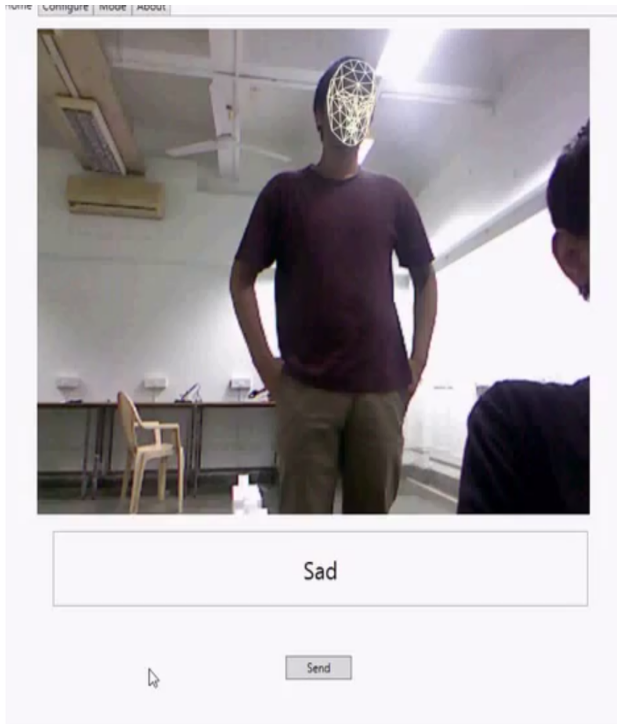
Default Face:



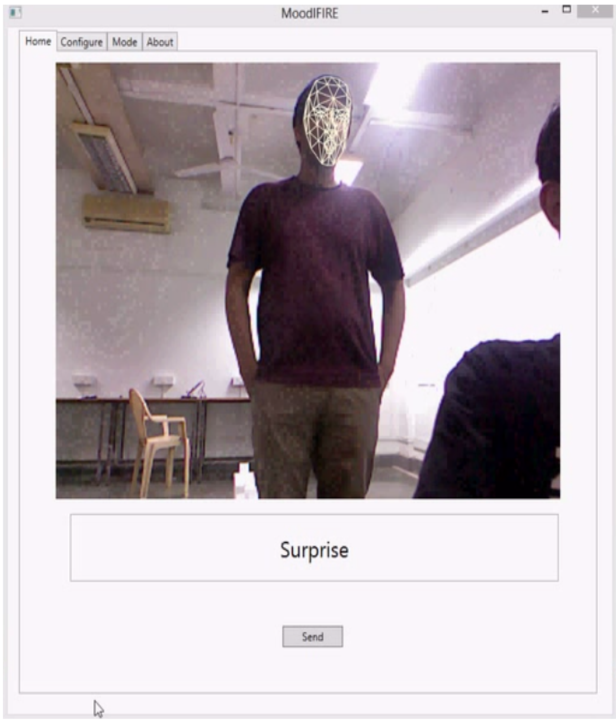
Happy Face:



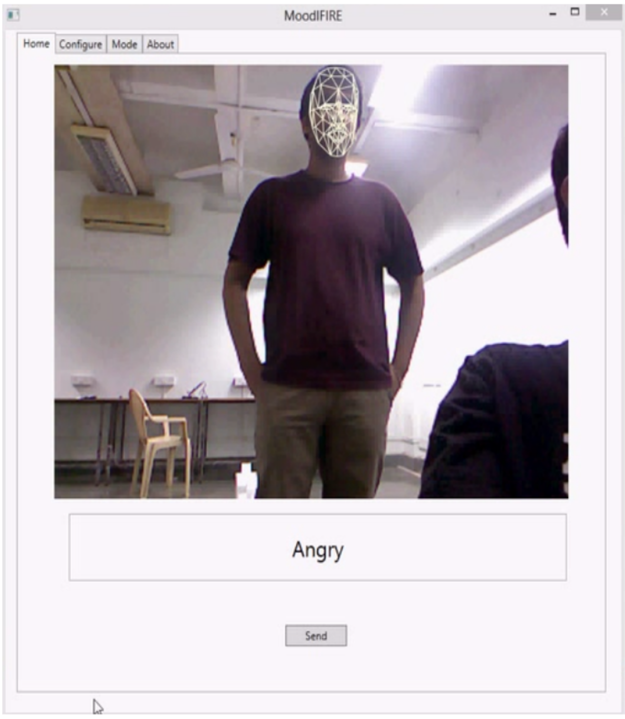
Sad Face:



Surprised Face:



Angry Face:



Fear Face:

We tested these emotions by their identification. The major functionalities and corresponding testing can be divided into the following blocks:

1. **Mood Analysis**: The Kinect analyzes the expressions to determine the mood of the user. It can be tested through input of user's facial expressions. The expressions detected are displayed on GUI.
2. **Wireless transfer**: The computer then transmits the encoded emotion to the automatic

robot using XBee module. It has been tested through the bot's LCD screen.

3. **Path-Planning**: The bot then, traverses the line-follower 2D space based on different inputs. It has been tested based on whether the bot moves to different ends or not, based on different expressions. The bot goes to following locations based on different expressions:

- Default - (0,0)
- Happy - (1,1)
- Sad - (2,2)
- Surprised - (2,1)
- Angry - (2,3)
- Fear - (3,2)

4. **Robotic Arm Control**: The bot at the junction end of the line-follower 2D space picks up objects and brings them back to user. It has been tested based on whether different objects like chocolates are picked if the user is happy, antidepressants are picked if user is sad, stabilizer medicines are picked, if user is angry and so on.

6. Discussion of System

a) What all components of your project worked as per plan?
Most of the components worked as planned except the sensors. The sensors in the bot were damaged and thus, they decreased accuracy of the locomotion.

b) What we added more than discussed in SRS?
We added the coordinate based locomotion module for bot. We made a robust module to help the bot to move from one co-ordinate point to another.

c) Changes made in plan from SRS:

1. Use of Remote - We discarded the remote as we intoroduced a switch in the GUI as master control for emotion send.

2. Gesture Recognition - It was already done by a previous team in past and thus, instead of spending our time and resources on this, we focussed more on increasing the accuracy of expressions identified.

3. Use of Firebird V LPC bot - Firebird V LPC bot doesnot have a RC Servo motor module. So, we had to switch back to Firebird V Atmega 2560 bot, which had servo motor module. This created a lot of new challenges as the Firebird V Atmega 2560 bot was old and its sensors were damaged. Also, we had to start the previously written code for path planning all over from scratch for ATMEGA 2560 microcontroller.

7. Conclusions & Future Work

We gave two modules for future use:
1. Expression Detection
2. Cordinate based locomotion

In future, the project can be modified to increase accuracy in emotion detection by trying supervised learning approaches. The possible projects are on this project:
1. Using supervised learning for more accuarcy by training on the SFEW SPI datasets and SFEW PPI datasets
2. Integrating Expression, Gesture and Voice based control using Kinect in a single project
3. Maintaing and improving the co-ordiante based path planning module

We also realized the need of a servo module for Firebird V LPC bot, which can be taken up as future project.

8. References

- Wyrembelski Adam, Detection of the selected, basic emotions based on face expression using Kinect
- Vineetha G R, Sreeji C, Lentin Joseph, Face Expression Detection Using Microsoft Kinect with the Help of Artificial Neural Network, Trends in Innovative Computing 2012 - Intelligent Systems Design