PC BASED MULTI-TOUCH INTERFACE SYSTEM

Project submitted in partial fulfillment of requirements

For the Degree of
Bachelor of Engineering

By
Prerak H. Desai
Labhesh Hase
Pallav Edwankar
Mihir Joshi

Under the guidance of
Internal Guide

Prof. K. T. Talele

Department of Electronics Engineering
Sardar Patel Institute of Technology
University of Mumbai
2009-2010

Bhartiya Vidya Bhavans
Sardar Patel Institute of Technology
Munshi Nagar, Andheri (w),
Mumbai - 400 058.
2009-10

PC BASED MULTI-TOUCH INTERFACE SYSTEM

Project submitted in partial fulfillment of requirements

For the Degree of
Bachelor of Engineering

By
Prerak H. Desai
Labhesh Hase
Pallav Edwankar
Mihir Joshi

Under the guidance of
Internal Guide

Prof. K. T. Talele

Department of Electronics Engineering

Bhartiya Vidya Bhavans
Sardar Patel Institute of Technology
Munshi Nagar, Andheri (w),
Mumbai - 400 058.
2009-10

CERTIFICATE OF APPROVAL

This is to certify that the following students

Prerak H. Desai
Labhesh Hase
Pallav Edwankar
Mihir Joshi

have successfully completed and submitted the project entitled

PC BASED MULTI-TOUCH INTERFACE SYSTEM

towards the fulfillment of Bachelor of Engineering course in Electronics of the Mumbai University

. . . .
Internal Examiner

. . . .
External Examiner

. . . .
Internal Guide

. . . .
Head of Department

. . . .
Principal

Contents

List of Figures

List of Tables

Acknowledgement

We would like to thank our internal guide, Prof K. T. Talele, for his overwhelming support during the entire phase of our project. His able guidance was instrumental in us achieving our goal. We would also like to thank him for providing us with the pre requisite components for the project.

We would also like to thank Prof Y.S.Rao for helping us make the PCBs and extending his knowledge of hardware electronics to us. We would also like to thank Prof S.T. Gandhe for providing us with the college projector whenever we needed it and the staff at the workshop for providing us with assistance related to carpentry and metal working. The project would not have shaped up the way it has without the support and constructive criticism of our professors that enabled us to complete our project with a high degree of competence.

We would also like to thank all the non teaching staff of Electronics division for their invaluable support during the duration of the project.

Abstract

Technology has evolved to a great extent and it promises to evolve further in the years to come. For instance, we have faster processors that can execute thousands of instruction within a second. These developments haven't stopped and scientist, developers and companies around the globe are continuously working on ways to improve the technology in every possible way.

Apart from speed, throughput and various other aspect of a computer there is one other important factor that is to be taken into account when designing a system. This aspect is the quality of interaction that takes place between user and computer. There are various ways in which quality of interaction can to improved.

The project aims at using one such low cost interaction method, a multi touch system, to show how the interaction between human and machine can be improved. This improved interaction is shown by controlling a microcontroller based robot according to the touch input given by the user.

The touch screen is not capacitive based screens that are used in the mobile phone these days. But the touch screen is implemented using Infra red LED and an Infra red camera. The robot is made using DC motors and a microcontroller. The feedback includes images captured by the over head camera. The communication is done using wireless module that implements IEEE 802.15.4 Zigbee protocol. The programming is done on a Java based platform and C language.

This project can be used in scenarios wherein the operator has to control multiple objects simultaneously and the environment cannot be viewed directly by the operator.

Chapter 1

Introduction

1.1 Brief Introduction

The traditional way of interacting with a computer is by a mouse and/or a keyboard. We provide the computer with inputs more or less by the use of buttons. Regardless of the input type, the computer can only handle one input at the time which makes the input handling and sorting very easy. However the way we use computers today will soon change. The tools we use today to communicate with the computer will slowly be replaced by tools which are more comfortable and natural for the human being to interact with. One such device is the Touch Screen. The touch screen can be single touch screen or multi touch screens. The single touch screens are widely used in small embedded device like mobile phones and have limited scope to which their implementation can extend. But with multi touch screen this is not the case, as the number as inputs are more.

A touch screen is an electronic visual output that can detect the presence and location of a touch within the display area. The term generally refers to touch or contact to the display of the device by a finger or hand. The touch screen has two main attributes. First, it enables one to interact with what is displayed directly with the hand. Secondly, it lets one do so without requiring any intermediate device. Such displays can be attaches to computers, terminals, or networks.

Touch screens emerged from corporate research labs in the second half of the 1940s. One of the first places where they gained some visibility was in the terminal of a computer-assisted learning terminal that came out in 1975 as part of the PLATO project . They have subsequently become familiar in kiosk systems, such as in retail and tourist settings, on point of sale systems, on ATMs and on PDAs where a stylus is sometimes used to manipulate the GUI and to enter data. The popularity of smart phones, PDAs, portable game consoles and many types of information appliances is driving the demand for, and the acceptance of, touch screens. Today there are many types [12] of touch screen technology available in the market:

1. **Resistive**: This type of screen uses the property of increased resistance when the touch is detected.

2. **Surface Acoustic wave**: This type of screen uses absorption of ultrasonic waves traveling over the touch surface to detect touch.

3. **Capacitive**: This type of screen uses the principle of change in capacitance to detect the touch.

4. **Strain Gauge**: This uses the principle of change in strain to detect the location of the touch.

5. **Optical Imaging**: This uses the reflection of IR light to detect the location of touch.

6. **Acoustic Pulse Recognition**: This type of screen uses more than two piezoelectric transducers located at some positions of the screen to turn the mechanical energy of a touch (vibration) into an electronic signal.

7. **Coded LCD**: In this method LCD display is turned into giant camera that provides gesture based control of objects on-screen.

1.2 Need for the Project

The primary reason of doing this project is to get to know and understand the touch technology, which is next innovation in consumer electronics. Though the reach of touch screen handsets is large in India, there is still a large scope for introducing touch screens in industrial and military applications. Our project aims to show one such application where we can remotely control multiple robots using a touch screen.

The touch screen has wide range of application and the project has helped us understand these applications and its varied implementations. One of the concepts that are currently under development is gesture recognition and their implementation using multi touch screen. The paper User-Defined Gestures for Surface Computing is one such example. Another application apart from controlling robot can be the use of this multi touch table in restaurants[8].

All robots, including those that do their tasks autonomously, do not work without an instruction by users. We therefore need interfaces for giving instructions to them. Single robots are normally controlled with joysticks, keyboards, and other pointing devices. However, with advances in robotics, variations of user interfaces for these purposes have become wider. Handling tasks with multiple robots is desirable, because they can do various tasks with greater efficiency than a single robot.

However, multiple robots substantially increase amount of information exchange with their users who have to maintain situational awareness and continue operation. It often makes the manner of operation complex and difficult. Users have a limitation in the capability of their attention, so they cannot see too much information displayed either at the same time or time-multiplexed. Therefore, many user interfaces for operating single robots do not work effectively.

Two factors are necessary for effective cooperation between people and robots. The first factor is that roles and responsibilities against tasks are clearly separated between users and robots. It is generally said that users should have responsibility for global tactics, and robots for local tasks. When the distinction between global and local tasks is unclear, problems occur for both the robot and the user. The second factor is that users can command robots as easily as possible. Here we need richness of user interfaces cultivated in the field of Human Computer Interaction (HCI). In teams consisting of people, robots and their supervisor, the use of graphical user interfaces (GUI) greatly affect the performance of their tasks.

In this project we shall demonstrate the efficiency and speed of the Frustrated Total Internal Reflection based system which can accept multiple inputs and use these inputs to move robots using wireless communication in an cost efficient way.

1.3 Scope of the Project

In this project, we have made an intuitive interface using a multi-touch display to control multiple mobile robots simultaneously. We shall be using FTIR technology to make the touch screen which is a relatively-modern development in touch screen technology. Here IR LEDs are placed along the edges of the screen and an IR camera is placed directly beneath the touch screen. A touch shows up as a shadow and camera is then used to locate the position and size of the touch. This human touch can then used to control multiple robots. Users get a top-down view on the touch screen from a ceiling camera in real time, which is virtually overlaid with a 2-dimensional vector field. The robots are denoted as icons on the touch screen. Users can manipulate these icons by touching and passing their hands on the display. As the icons are moved along the screen the new coordinates are sent to the robots by Zigbee wireless communication modules. The robots are fitted with Atmega 168 controllers which then direct the robot to the new location. However as the number of devices increase, the cost increase exponentially as each device has to support its own controller board, wireless modules, motors and power supply. Hence in the interest of making the project viable we shall demonstrate the controlling of one robot using the multi touch interface. However to demonstrate the multi touch capability of the touch screen various application have also been developed by the group. These include a program that generates a sine wave whose amplitude and frequency is controlled by the touch points and finger movements, a gesture based program that creates rectangular and circular objects, menu dragging similar to one used in cellphones, playing with a 3D box, and many others. In short the scope of this project includes:

1. Creating an FTIR based touch table
2. Creating an IR webcam from a normal one
3. Detecting multiple touches and drag on the touch screen
4. Detecting robots on the field
5. Establishing communication between the robots and CPU using XBee

6. Ensuring the synchronization between the motion of drag on the touch screen and the motion of robot in the real world.

7. Running Java applications on the touch screen having multiple inputs.

1.4 Block Diagram

The proposed system consists of four major blocks:

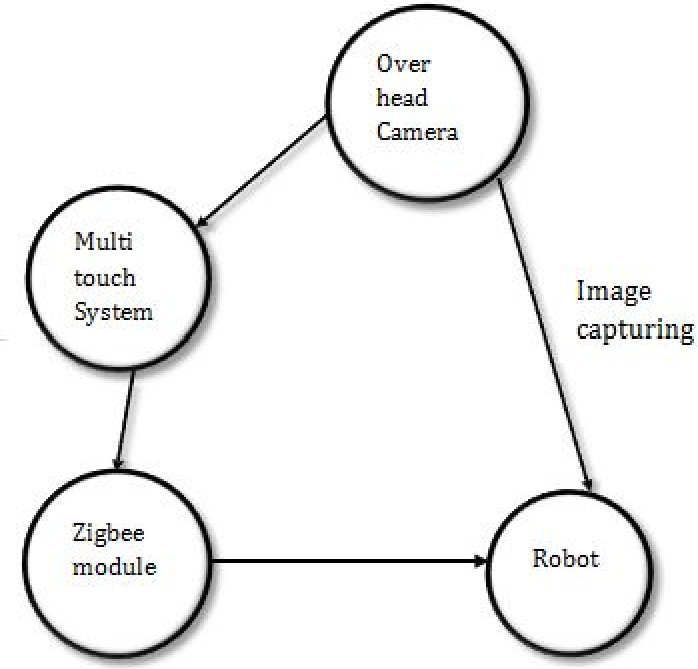


Figure 1.1: Block Diagram

- The Multi touch system

  The system acts as an interacting medium between the computer and the user. The idea of using this as the interacting means is to try and improve the interaction method. This section gets input from the over head camera. It further sends data to the zigbee module.

- Over head Camera

  The role of overhead camera is to capture the region within the workspace and send the data to the interfaced java application. The application then uses this captured image to search for the required marker. This marker is an indication of the presence of the robot.

- Zigbee module

  The zigbee module is the wireless module that communicates with robot. The communication involves transmission of the required parameters. The parameters involve the amount of distance to be travel and the angle through which the robot needs to be rotated.

- Robot

  The robot is a four wheeled robot and which is controlled by an Atmega 328 chip. The role of the microcontroller is to receive the data from the zigbee module and decode the data in order to find the distance and angle to be covered.

Chapter 2

Literature Survey

1. Software driven Multi Touch input display as an improved, intuitive, and practical interaction device, Bradley Hayes, 2008 Boston college[9]

   This thesis examines and investigates the substitution of the mouse for a more natural means of human computer interaction, ranging from manipulating WIMP-based applications to developing post-WIMP interfaces and exploring their usefulness. The WIMP (Window, Icon, Menu, and Pointer) interface has been the standard paradigm for the personal computing era. Their use is optimized for the keyboard and mouse input device pair, tools that have remained fundamentally stagnant with regard to innovation for decades. Accomplished through the construction of a touchscreen with variable levels of contact detection, targeted demo applications not only show the effectiveness of such an input apparatus but introduce the potential for previously unexplored levels of interaction.

2. A FTIR based Multit Touch Table Using Pulsed Illumination for noise Suppression, Uwe Hahne, Marc Alexa, Bjorn Bollensdorff, Ingo Bressler, Stefan Elstner,2008, TU Berlin[1]

   Touch Sensing based on Frustrated Total Internal Reflection (FTIR) is capable of sensing multiple touch points on a projection screen. Infrared light is totally reflected within a screen, and on touch diffusely reflected from the finger tip. This diffuse reflection is captured with a camera and fingertips are localized by blob detecting image processing. This approach suffers from surrounding infrared light and a small amount of light leaving the screen and illuminating the user so far it can only be used in controlled lighting conditions. In our new approach, we obtain a dynamic background subtraction, which makes detection of fingertips possible even in the presence of non-constant, bright illumination conditions.

3. Multi Touch Interaction Wall, Jefferson Y. Han, 2006 Courant Institute of Mathematical Sciences,New York University[7]

   Touch is a very natural and intuitive way for people to interact. However, typical touchscreen technology, such as that commonly found in kiosks and interactive whiteboards are limited to detecting only a single point of contact at a time. This makes them completely unsuitable for enabling touch interaction on larger displays designed to accommodate multiple users, since any one touch will prevent any others from being registered at the same time. The Multi-Touch Interaction Wall is a 16-feet long by 3-feet high rear-projection surface that has the unique capability of detecting multiple points of contact simultaneously. At this Emerging Technologies installation, we demonstrate an assortment of innovative applications and user interfaces that leverage this unique device.

4. Arduino Programming Notebook ,By Brian W. Evans[16]

   This notebook serves as a convenient, easy to use programming reference for the command structure and basic syntax of the Arduino microcontroller. To keep it simple, certain exclusions were made that make this a beginners reference best used as a secondary source alongside other websites, books, workshops, or classes. This decision has lead to a slight emphasis on using the Arduino for standalone purposes and, for example, excludes the more complex uses of arrays or advanced forms of serial communication.

5. Building a multi-touch display based on computer vision techniques -By D.Michl, A.A. Argyos, D. Grammenos, X. Zabulis, T. Sarmis,2009 Institute of Computer Science, Greece[2]

   We present the development of a multi-touch display based on computer vision techniques. The developed system is built upon low cost, off-the-shelf hardware components and a careful selection of computer vision techniques. The resulting system is capable of detecting and tracking several objects that may move freely on the surface of a wide projection screen. It also provides additional information regarding the detected and tracked objects, such as their orientation, their full contour, etc. All of the above are achieved robustly, in real time and regardless of the visual appearance of what may be independently projected on the projection screen. We also present indicative results from the exploitation of the developed system in three application scenarios and discuss directions for further research.

6. Multi Touch Sensing using Frustrated Total Internal Reflection - By Jefferson Y. Han, 2006 Courant Institute of Mathematical Sciences, New York University[6]

   While touch sensing is commonplace for single points of contact, it is still relatively difficult and/or expensive to construct a touch sensor that can register multiple points of contact simultaneously by one or more users. We introduce here a new technique that enables robust multi-touch sensing at a minimum of engineering effort and expense. It relies on a phenomenon known as frustrated total internal reflection (FTIR), a technique well known in the biometrics community for fingerprint acquisition. It captures true touch information at high spatial and temporal resolutions, and is trivially scalable to large installations.

7. A Low Cost Infrastructure for Table Top Games -By Christopher Wolfe, J. David Smith and T.C. Nicholas Graham School of Computing, Canada[3]

   Tabletop games provide an intimate gaming experience where groups of friends can interact in a shared space using shared physical props. Digital tabletop games show great promise in bringing this experience to video game players. However the cost of developing tabletop games is high due to the need for expensive hardware and complex software. In this paper, we introduce EquisFTIR, a low-cost hardware and software infrastructure for digital tabletop gaming.

8. Synthesis and Control on Large Scale Multi-Touch Sensing Displays -By Philip L. Davidson, Jefferson Y. Han, New York University[4]

   In this paper, we describe our experience in musical interface design for a large scale, high-resolution, multi-touch display surface. We provide an overview of historical and present day context in multi-touch audio interaction, and describe our approach to analysis of tracked multi-finger, multi-hand data for controlling live audio synthesis. The musicians need to manipulate many simultaneous degrees of freedom in audio synthesis has long driven the development of novel interface devices. Touch sensors integrated with graphical display functionality can provide intuitively direct interactivity with richly dynamic context; however they are typically only able to respond to a single point of contact a time, making them quite limiting for musical input. Multi-touch sensors on the other hand permit the user fully bi-manual operation as well as chording gestures, offering the potential for great input expression. Such devices also inherently accommodate multiple users, which makes them especially useful for larger interaction scenarios such as interactive tables.

9. Zigbee: Wireless Control that Simply Works -By William C. Crai, Program Manager Wireless Communications, ZMD America[13]

   There are many wireless monitoring and control applications for industrial and home markets which require longer battery life, lower data rates and less complexity than available from existing wireless standards. These standards provide higher data rates at the expense of power consumption, application complexity and cost. What these markets need, in many cases, is a standards-based wireless technology having the performance characteristics that closely meet the requirements for reliability, security, low power and low cost. This standards-based, interoperable wireless technology will address the unique needs of low data rate wireless control and sensor-based networks. For such wireless applications, a standard has been developed by the IEEE: ”The IEEE 802.15 Task Group 4 is chartered to investigate a low data rate solution with multi-month to multi-year battery life and very low complexity. It is intended to operate in an unlicensed, international frequency band”. Potential applications are home automation, wireless sensors, interactive toys, smart badges and remote controls.

10. Low-Cost Malleable Surfaces with Multi-Touch Pressure Sensitivity -By J. David Smith, T.C. Nicholas Graham, School of Computing, Queens University[5]

    While touch sensitivity has today become commonplace, it is oftentimes limited to a single point of contact with a hard, rigid surface. We present a novel technique for the construction of a malleable surface with multi-touch sensitivity. The sensor is pressure sensitive and responds to near zero-force touch from any object. The technique is an extension of previous work based on frustrated total internal reflection.

Chapter 3

FTIR Concepts

There are various methods by which multi touch system can be made. Some of these are listed below:

1. Frustrated Total Internal Reflection(FTIR)
2. Diffused Illumination (DI)
3. Using Capacitive or Resistive screen
4. And various others

The concept of FTIR is in existing for a long time but a successful implementation of this concept was demonstrated by Jeff Han[6] at the TED in the year 2005. The FTIR concept uses a combination of optics and image processing in order to find out the touch points. As can be seen in the figure 1, the IR led are placed facing the edge of the acrylic sheet. This ensures that the IR light rays experience Total Internal Reflection at the boundary of the acrylic sheet and air.

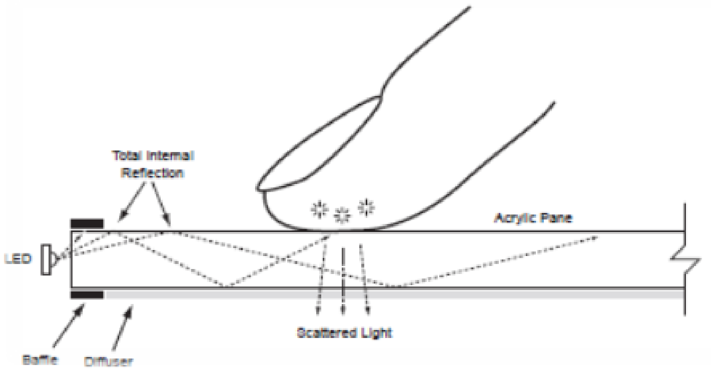


Figure 3.1: FTIR concept

As the IR light rays are experiencing total internal reflection the IR camera cannot detect these single light rays.

But when the user places his/her finger on the screen the total internal reflection is disturbed at that particular point. And thus at this point a lot IR light get concentrated. This along with the fact that the camera and light rays are almost parallel to each other makes the detection of the IR blob easier. These blobs are the area where the user makes contact with the surface of the acrylic sheet. The blobs are detected using IR camera placed below the acrylic sheet.

3.1 FTIR Layers

The FTIR layer are the layer which are placed one above the other to form the required touch sensor.These layer have been explained below:

Acrylic: It is necessary to use acrylic for the screen since it has the desired properties. The minimum thickness is 6 mm however large screens should use 1 cm to prevent the screen from bending.

Before a sheet of acrylic can be used for a multi-touch screen it needs to be prepared. Because acrylic often gets cut up roughly, it is required to polish the sides of the sheet. This is done to improve the illumination from the sides. To polish the sheet it is recommend using different kinds of sandpapers. We must first start with a fine sandpaper to remove most of the scratches, after that continue with very fine, super fine and even wet sandpaper. To make your sheets shine we can use Brasso.

Figure 3.2: Acrylic Sheet and Compliant Surface

Baffle: The baffle is required to hide the light that is leaking from the sides of the LEDs. This can be a border of any material (wood/metal).

Diffuser: Without a diffuser the camera will not only see the touches, but also all objects behind the surface. By using a diffuser, only bright objects (touches) will be visible to the camera. All other noise data will be left out.

Compliant layer: With a basic FTIR setup, the performance mainly depends on how greasy the fingertips of the user are. Wet fingers are able to make better contact with the surface. Dry fingers and objects wont be able to frustrate the TIR. To overcome this problem it is recommended to add a compliant layer on top of the surface. Instead of frustrating the total internal reflection by touch, a compliant layer will act as a proxy. The complaint layer can be made out of a silicon material[15] such as ELASTOSIL M 4641. To protect and improve the touch surface, rear projection material such as Rosco Gray can be used. With this setup it is no longer required to have a diffuser on the rear side.

3.2 Strength and Weaknesses

There are various method of making the touch sensor. These various methods have their respective advantages and disadvantages. The strength and weaknesses of Frustrated Total Internal Reflection method of making a touch sensor are as follows:

The strength includes:

1. High contrasting blob. Blobs are the touch points where the user of the system is making contact with the acrylic sheet. High contrasting blobs enables high touch sensitivity and avoids applying large amount of pressure.

2. Varying amount of pressure can be sensed and thus a new dimension can be added to the touch points.

3. With a compliant surface, it can be used with something as small as a pen tip

The weakness includes:

1. The inability of the system to recognize objects or markers which are also known as fiducials.

2. Requires a compliant surface (silicone rubber) for proper use - no glass surface

Even with these weaknesses FTIR method is simpler method to implement as compared to the other methods like the Diffused Illumination[14]. The FTIR method cannot recognize objects or fiducials, but if the sensitivity of the screen in improved using powerful IR led it is possible to recognize objects as well. The object that would be detected can result in ambiguous detection. Thus to avoid ambiguous detection using FTIR method, various objects need to listed along with the pattern of blobs that appear. For example, if a cellphone is placed on the acrylic sheet there is a possibility of obtain a rectangular blob. Or if a coin is placed there is a possibility of getting a circular blob.

In order to avoid this ambiguous decision making DI method should be used. The use of marker which has known definition for a particular object in that particular system makes the detection of object easier.

Chapter 4

Hardware Description

4.1 Introduction

There are two sections in the project. One is the software and the other is the hardware section. The hardware section involves the making of the robot and the construction of the touch table. This section of the report focuses on is the electronic hardware involved in the multi touch system and the hardware used to make the robot.

The hardware section is further segregated in the following sub-sections:

1. Multi touch table
2. XBee communication
3. Robot Movement

The hardware apparatus exists to transform the task of collecting and interpreting screen contact data into a binary blob tracking problem. The actual display presentation is accomplished via rear projection, using a translucent material affixed to the screen. By utilizing rows of infrared LEDs attached to the periphery of the screen directed inwards toward the center of the glass, any object in direct contact will reflect the infrared light. This light is then processed by the webcam behind the screen, isolated by an infrared-pass filter.

The hardware construction of the project can be mainly divided into 3 parts: construction of the Touch Screen, construction of the Atmega boards atop the robots and communication between the robot and the touch table.

4.2 Touch Table

4.2.1 Touch Screen

A projecting surface is needed as a screen to display the interface and to provide appropriate transmission characteristics for the IR LED to pass through and to reflect to the IR camera. The acrylic glass has the following characteristics making it the ideal one to use in a multi-touch interface system:

- Easy handling and processing
- High boiling and melting point.
- Has a density of 1,1501,190 kg/m3,hence it is light weight.
- Allows infrared light of up to 2800 nm wavelength to pass. IR of longer wavelengths, up to 25,000 nm, is essentially blocked.
- Transmits up to 92% of visible light (3 mm thickness), and gives a reflection of about 4% from each of its surfaces on account of its refractive index of 1.4893 to 1.4899.

A **diffuser** is placed behind the screen so that only bright surfaces are visible to the camera. With a basic FTIR setup, the performance mainly depends on how greasy the fingertips of the user are. Wet fingers are able to make better contact with the surface. Dry fingers and objects wont be able to frustrate the TIR. To overcome this problem it is recommended to add a compliant layer on top of the surface. Instead of frustrating the total internal reflection by touch, a compliant layer will act as a proxy. The complaint layer can be made out of a silicon material such as ELASTOSIL M 4641. To protect and improve the touch surface, rear projection material such as Rosco Gray can be used.

The **compliant surface** or compliant layer is simply an additional layer between the projection surface and the acrylic. It enhances the finger contact and gives you more robust blobs, particularly when dragging as your finger will have less adhesion to the surface. In the FTIR technique, the infrared light is emitted into side the acrylic waveguide, the light travels inside the medium (due to total internal refection much like a fiber optic cable), when you touch the surface of the acrylic (you frustrate this TIR effect) causing the light to refract within the medium on points of contact and creating the blobs (bright luminescent objects).

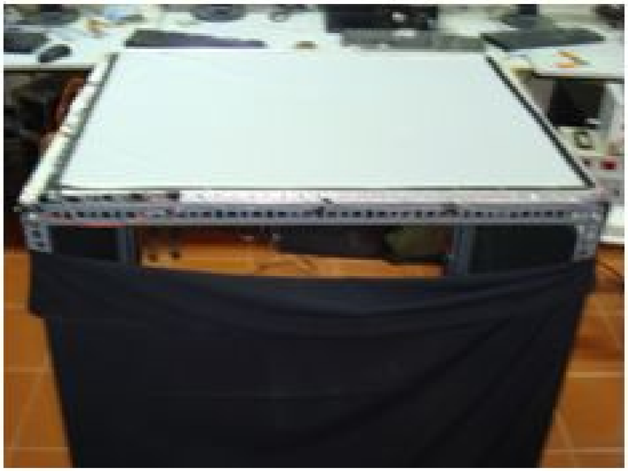


Figure 4.1: Touch table

4.2.2 IR Led Array

We prefer IR LEDs over normal LEDs as there is a difference in their electrical properties. Infrared LEDs have a lower forward voltage, and a higher rated current compared to visible LEDs. This is due to differences in the material properties of the junction. A typical drive current for an infrared LED can be as high as 50 milliamps. This is essential for the FTIR as the intensity on every point on the surface must be constant. If there are distortions on the surface the camera may detect it as a touch and this may lead to an error.

Common infrared LED that emits infrared rays has the same appearance with visible light LED. Its appropriate operating voltage is around 1.4v and the current is generally smaller than 20mA. Current limiting resistances are usually connected in series in the infrared LED circuits to adjust the voltages, helping the LEDs to be adapted to different operating voltages.

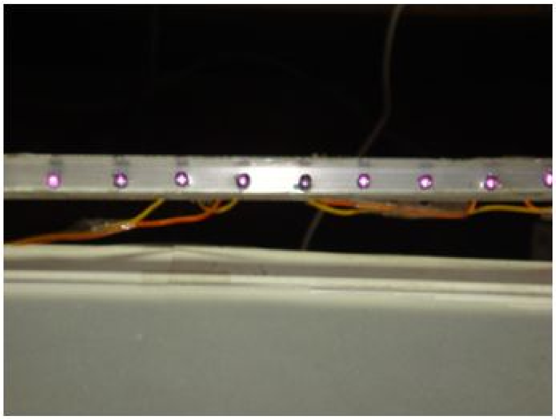


Figure 4.2: IR Array in working condition. The red points indicate that IR LED is working. They are not invisible to naked eyes.

Table 4.1: IR led Specifications

Parameter	Symbol	Rating	Units
Forward Current	I_F	50	mA
Pulse Forward Current	I_{FP}	1	A
Reverse Voltage	V_R	5	V
Power Dissipation	P_D	95	mW
Operating Temperature	T_{OPR}	-20→ +70	°C
Half Power Beam Angle	δ	10	degree
Peak Emission Wavelength	λP	850	nm
Spectral Bandwidth	$\delta\lambda$	30	nm

Infrared LED chips with different wavelengths can be applied in extensive devices, for example, Infrared LED chip with wavelength of

1. **940nm**: suitable to be used in remote controller, such as remote controllers for household appliances.

2. **808nm**: suitable to be used in medical treatment appliances, space optical communication, infrared illumination and the pumping sources of the solid-state lasers.

3. **830nm**: suitable to be used in the automated card reader system in freeway.

4. **840nm**: suitable to be used in colored zoom infrared waterproof video camera.

5. **850nm**: suitable to be used in video cameras that are applied in digital photography, monitoring system, door phone, FTIR and so on.

6. **870nm**: suitable to be used in video cameras in marketplace and crossroad.

For our project, we have used IR0510. Its datasheet and description is as follows:

Infrared Emitting Diode: IR0510

General Description: IR0510 is a high output power and high speed GaAlAs infrared light emitting diode mounted in a clear epoxy looking package. It emits narrow band of radiation peaking at 850nm. The general specifications of the LED IR0510 are:

4.2.3 IR Detection Camera

We need a webcam which detects only IR light and filters out the visible light. The webcam must also have high resolution and high fps so that the blob motion can be detected and identified. The camera which fulfills all the above requirements is the PS3 Eye .Its properties are:

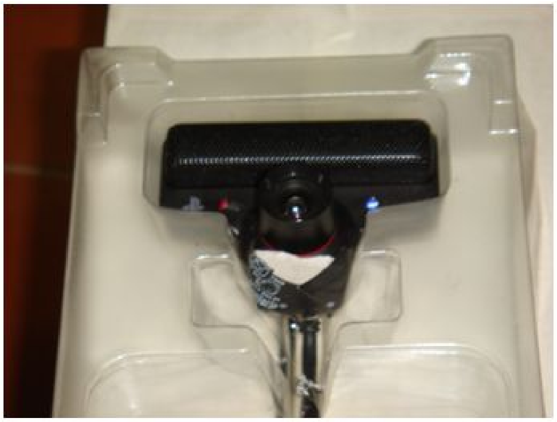


Figure 4.3: PS3Eye camera converted to IR camera

Frame capturing: Frames are captured from the webcam at its native resolution. As we are dealing with filtered infrared light, a single channel image is sufficient to preserve all relevant data. As it is typical for the frame rate of the camera to be limited by the efficiency of the blob identification algorithms, significant increases in performance can be obtained by scaling the incoming image by a factor of one half or one quarter. It has fps of 60.

Camera calibration: Camera calibration is crucial to the usability of the device. This project accomplished calibration by utilizing an optional 8-point touch exercise on launch, and calculating a homography based on the identified blob locations and on-screen coordinates of the target areas. The application saves the homography to its local folder until the next calibration is run. This provides a more user friendly experience, as it quickly becomes tedious to calibrate each time the application is launched. This step utilizes its own routine for identifying blobs, as precision is of paramount importance.

4.3 Microcontroller-Atmega328

The direction and distance is sent to the Atmega168 microcontroller by the CPU via the XBee communication modules. The microcontroller converts this data into precise rotations of the DC motors using L293D. The component which is absolutely essential to this conversion is the Atmega168 controller. The pin diagramis as shown above. The technical features are as given below:

Figure 4.4: Atmega328 microcontroller chip

Features of Atmega168

- High Performance, Low Power AVR 8-Bit Microcontroller
  - Advanced RISC Architecture
  - 131 Powerful Instructions Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
- High Endurance Non-volatile Memory Segments
  - 4/8/16K Bytes of In-System Self-Programmable Flash progam memory (ATmega48P/88P/168P)
  - 256/512/512 Bytes EEPROM (ATmega48P/88P/168P)
  - 512/1K/1K Bytes Internal SRAM (ATmega48P/88P/168P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85C/100 years at 25C
  - Optional Boot Code Section with Independent Lock Bits
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler , Compare, and Capture Mode
  - Real Time Counter with Separate Oscillator

  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
  - 6-channel 10-bit ADC in PDIP Package
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change

- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

- Operating Voltage:
  - 1.8 - 5.5V for ATmega48P/88P/168PV
    - 2.7 - 5.5V for ATmega48P/88P/168P
  - Temperature Range:
  - -40°C to 85°C

- Speed Grade:
  - ATmega48P/88P/168PV: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
  - ATmega48P/88P/168P: 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V

4.3.1 Programming Robot Board

The **programming board** that is used is called a Arduino. Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing)[18]. Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

The boards can be built by hand or purchased preassembled; the software can be downloaded for free. The hardware reference designs (CAD files) are available under an open-source license.

The **robot specific board** is designed by the group using the Eagle software. The board is single sided board.All the ports of the microcontroller has been pulled out. The Zigbee(Xbee modules) modules are interfaced to RXD and TXD pin of the microcontroller. The supply to the board is a 12V supply. This supply is converted to 5V using 7805 voltage regulater. 5V suply can also be directly give to the board. The supply and and reset led indicate is microntroller is properly powered or the microcontroller is properly reset. The board has provison to interface four independent motors at time. The motor are interfaced to the PORTB and PORTC.

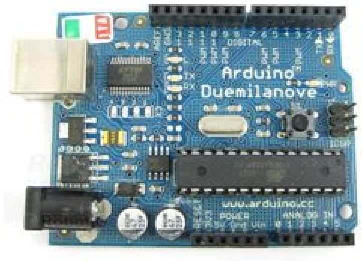


Figure 4.5: USB board used to program the microcontroller

A motor driver daughter board was also designed which can be directly interface to the programming board. Along with this a Xbee module based daughter board was also designed, which is compatible to the Arduino programming board.

4.4 XBee Communication Modules

The XBee modules are used to send distance and angle from old position to the new position to the robot. The transmitter is connected to the CPU whereas the receiver is connected on the robot. The XBee RF Modules were implement IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other. The key features of XBee are:

Figure 4.6: The Zigbee modules used for wireless communication

Table 4.2: Specifications

Specifications	
Indoor Range	Upto 100 ft
Outdoor RF line of sight range	Upto 300 ft
Transmit Power Output	1mW
RF Data Rate	250,000 bps
Serial Interface Data Rate	1200 bps-250 kbps
Receiver Sensitivity	-92 dBm
Power Requirement	
Supply Voltage	2.8-3.4 V
Transmit Current	45 mA
Receive current	50 mA
Networking and Security	
Supported Network Topologies	Point to point, point to multipoint, peer to peer
Number of Channels	16 Direct Sequence Channels
Addressing Options	PAN ID, Channels and Addresses

XBee RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device. Devices that have UART interface can directly to the pins of the RF module as shown in the figure:

Data enters the module UART through the DI pin as an asynchronous serial signal. The signal should idle high when no data is being transmitted. Each data byte consists of a start bit, 8 data bits and a stop bit. The following figure illustrates the serial bit pattern of data passing through the module.

The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits). There are two modes of operation are: Transparent Mode and API mode.

4.4.1 ZigBee v/s Bluetooth

ZigBee is broadly categorized as a low rate WPAN, and its closest technology is Bluetooth. They are two different technologies with very different areas of application and different means of designing for those applications. While

- ZigBee is focused on control and automation, Bluetooth is focused on connectivity between laptops, PDAs, and the like, as well as more general cable replacement.
- ZigBee uses low data rate, low power consumption, and works with small packet devices; Bluetooth uses a higher data rate, higher power consumption, and works with large packet devices.
- ZigBee networks can support a larger number of devices and a longer range between devices than Bluetooth.
- ZigBee is designed to respond quickly, while Bluetooth takes much longer and could be detrimental to the application.
- Bluetooth must rely on fairly frequent battery recharging, while the whole goal of ZigBee is for a user to be able to put a couple of batteries in the devices and forget about them for long time.

Because of these differences, the technologies are not only geared toward different applications, they don't have the capability to extend out to other applications. Thus, a user could easily use both technologies as a wireless solution in a PAN to suit all types of applications within that network. The technical details and differences between the two are listed below:

- **Modulation technique:**
  Bluetooth: Frequency Hopping Spread Spectrum (FHSS)
  ZigBee: Direct Sequence Spread Spectrum (DSSS)
- **Protocol stack size:**
  Bluetooth: 250 kbyte
  ZigBee: 28 kbyte
- **Battery:**
  Bluetooth: Intended for frequent recharging
  ZigBee: Not rechargeable (one reason batteries will last for up to 10 years)

- **Maximum network speed:**
  Bluetooth: 1 Mbit/s
  ZigBee:250 kbit/s

- **Network range:**
  Bluetooth: 1 or 100m, depending on radio class
  ZigBee: Up to 70m

- **Typical network join time:**
  Bluetooth: 3 seconds
  ZigBee: 30 milliseconds

Chapter 5

Software Description

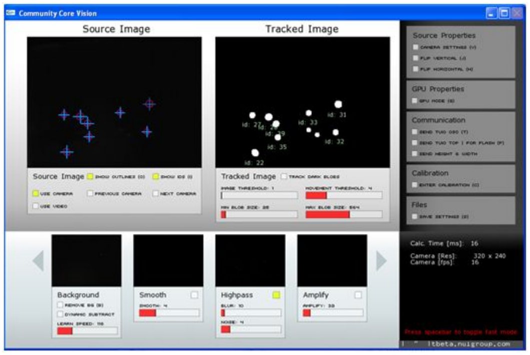
The main tasks in the entire process are to detect robots lying on ground and the blobs of our fingertips. This requirement can be fulfilled by :

1. Object detection algorithm and
2. Blob detection algorithm

Object Detection Algorithm In edge detection we get only edge of the object. So limitation of edge detection is that object of same edge might also get detected as robot. So to avoid it we go for template matching method. In this technique we go by following algorithm:

1. Capture images of objects to be detected.
2. Make its entry into the database
3. Now capture current image.
4. Perform edge detection
5. Check length and breadth of the successive edges
6. Compare length and breadth with database image parameters
7. Then find correlation coefficient of array elements of current and database image.
8. If correlation coefficient a threshold indicate the presence of the object using a circle(say)
9. Display circle corresponding to our robot on the multi-touch screen.

Blob detection algorithm



Source Image ☑ SHOW OUTLINES (O) ☑ SHOW IDS (I)
☑ USE CAMERA □ PREVIOUS CAMERA □ NEXT CAMERA
□ USE VIDEO

Tracked Image □ TRACK DARK BLOBS
IMAGE THRESHOLD: 1
MOVEMENT THRESHOLD: 4
MIN BLOB SIZE: 25
MAX BLOB SIZE: 564

Source Properties
□ CAMERA SETTINGS (V)
□ FLIP VERTICAL (J)
□ FLIP HORIZONTAL (H)

GPU Properties
□ GPU MODE (G)

Communication
□ SEND TUIO OSC (T)
□ SEND TUIO TCP | FOR FLASH (F)
□ SEND HEIGHT & WIDTH

Calibration
□ ENTER CALIBRATION (C)

Files
□ SAVE SETTINGS (S)

Background
□ REMOVE BG (B)
□ DYNAMIC SUBTRACT
LEARN SPEED: 118

Smooth □
SMOOTH: 4

Highpass ☑
BLUR: 10
NOISE: 4

Amplify □
AMPLIFY: 33

Calc. Time [ms]: 16
Camera [Res]: 320 x 240
Camera [fps]: 16

Figure 5.1: Blob detection

For constructing and designing a interactive tables mirrors commonly have to be used to allow a camera and projector configuration which results in a usable table height. Instead of showing detection of IR Light in GUI of our program we convert image obtained from our CAM into binary image. This binary image after inverting we displayed it on PC. Tracking touches on a surface involves setting up a pipeline of image processing operators that transform a camera image into user interface events.

Images captured by a camera are first pre-processed to remove any unchanging parts using history subtraction. A connected components algorithm finds bright regions in the pre-processed image. These are the areas where something is touching the surface. Post processing involves finding corresponding touches in different camera frames (temporal correlation) and transforming the camera coordinates to screen coordinates.

When given the processed image, the blob detection module performs a sequential pixel comparison to find edges of potential intentional contact points. Upon detection of a nonzero value, the algorithm begins a radial expansion about the coordinate. Each subsequent nonzero value has its coordinates added to a list as the expansion progresses, stopping only if a zero or value with a delta of greater than a constant,

$$\delta = |\text{ current coordinate value} - \text{previous coordinate value}|$$

As coordinates are evaluated, they are zeroed out to prevent overlapping points and infinite recursion. Upon return from the expansion function, the centroid is calculated along with the mean intensity. Blobs found with radii outside the specified acceptable range are discarded.

Once a list of all candidate points are computed and filtered, they are further pared down through a process of small neighborhood elimination. If two centroids are found to be within a predetermined constant distance that with a weaker mean intensity is discarded. This neighborhood is optimally set to be a small region. The value used for this constant is primarily affected by the delta used in the expansion function.

For touch programming, it is required that we identify different touch input. Along with identifying different touch position it is required that the programming environment also updates the position of the corresponding touch inputs. These requirements are required for all the multi touch system that uses TUIO protocol to communicate the touch data.

5.1 Community Core Vision(CCV)

Community Core Vision or CCV for short (also known as tbeta), is a open source/cross-platform solution for computer vision and machine sensing. It takes a video input stream and outputs tracking data (e.g. coordinates and blob size) and events (e.g. finger down, moved and released) that are used in building nui aware applications. CCV can interface with various web cameras and video devices as well as connect to various TUIO/OSC/XML enabled applications and supports many multi-touch lighting techniques including: FTIR, DI, DSI, and LLP with expansion planned for the future vision applications (custom modules/filters).CCV is released under the LGPL License. The main characteristics of CCV 1.2 are:

- Cameras:
  - Full PS3 camera support on windows
  - Support for Firefly cameras on windows
- Fixes:
  - Background Removal on startup
  - GPU speed slightly faster than before
  - New calibration technique - Can now calibrate even when there are false blobs
- Features:
  - Simple GUI - The new interface is more intuitive and easier to understand and use.
  - Filters (dynamic background subtraction, high-pass, amplify/scaler, threshold) - This means it will work with all optical setups (FTIR, DI, LLP, DSI). More filters can be added as modules.
  - Camera Switcher - Have more than one camera on your computer? Now you can press a button and switch to the next camera on your computer without having to exit the application.
  - Input Switcher- Want to use test videos instead of a live camera? Go ahead, press a button and it will switch to video input.
  - Dynamic Mesh Calibration - For people with small or large tables, now you can add calibration points (for large displays) or create less points (smaller displays) while maintaining the same speed and performance.
  - Image Reflection- Now you can flip the camera vertical or horizontal if it is the wrong way.
  - Network Broadcasting - You can send OSC TUIO messages directly from the configapp for quick testing.

  - Camera and application FPS details viewer - Now you can see the frame rate of both the tracker and camera that you are getting.
  - GPU Mode - Utilize your GPU engine for accelerated tracking.
  - Cross-platform - This works on windows, Mac, and Linux.
  - Min/max blob tracking size sliders - Sets the minimum and maximum size blobs to look for
  - Dynamic background subtraction 'learn speed' slider - Sets the rate dynamic background subtraction occurs
  - Movement threshold to eliminate jittery blobs - Sets the minimum amount a blob must move to consider it as 'moving'. This helps with jitter
  - Option to track dark or light blobs only (no more absolute background subtraction since it causes inconsistencies)
  - Startup in mini mode
  - Logging system - In bin/logs you can find a text log file of what happened when running the application. This should be useful in debugging errors

- Low Level features:
  - Add-on support for openframeworks - Option to not use TUIO and use directly with c++
  - Classes rewritten to comply with openframeworks addon standards Communication
  - TUIO direct to Flash sending through TCP - Can now send TUIO to flash without FLOSC (currently doesn't work on Windows Vista for unknown reason)
  - Option to turn on/off width/height TUIO sending to comply with the TUIO 1.0 protocol
  - Now sending correct fseq, dX, dY, and maccel TUIO values.

5.2 Processing

Processing is an open project[19] initiated by Ben Fry and Casey Reas. It evolved from ideas explored in the Aesthetics and Computation Group at the MIT Media Lab. It is an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool. Processing is free to download and available for GNU/Linux, Mac OS X, and Windows. Processing is a simple programming environment that was created to make it easier to develop visually oriented applications with an emphasis on animation and providing users with instant feedback through interaction. The developers wanted a means to sketch ideas in code. As its capabilities have expanded over the past six years, Processing has come to be used for more advanced production-level work in addition to its sketching role. Originally built as a domain-specific extension to Java targeted towards artists and designers, Processing has evolved into a full-blown design and prototyping tool used for large-scale installation work, motion graphics, and complex data visualization. The latest version of Processing can be downloaded at http://processing.org/download.

Processing consists of:

- The Processing Development Environment (PDE). This is the software that runs when you double-click the Processing icon. The PDE is an Integrated Development Environment (IDE) with a minimalist set of features designed as a simple introduction to programming or for testing one-off ideas.
- A collection of functions (also referred to as commands or methods) that make up the core programming interface, or API, as well as several libraries that support more advanced features such as drawing with OpenGL, reading XML files, and saving complex imagery in PDF format.
- A language syntax, identical to Java but with a few modifications.
- An active online community, hosted at http://processing.org. For this reason, references to Processing can be somewhat ambiguous. Are we talking about the API, the development environment, or the web site? We'll be careful in this text when referring to each.

5.3 TUIO Protocol

TUIO is an open framework that defines a common protocol and API for tangible Multitouch surfaces. The TUIO protocol allows the transmission of an abstract description of interactive surfaces, including touch events and tangible object states. This protocol encodes control data from a tracker application (e.g. based on computer vision) and sends it to any client application that is capable of decoding the protocol. There exists a growing number of TUIO enabled tracker applications and TUIO client libraries for various programming environments, as well as applications that support the protocol. This combination of TUIO trackers, protocol and client implementations allow the rapid development of table based tangible interfaces. TUIO is mainly designed as an abstraction for interactive surfaces, but also has been used in many other related application areas. Technically TUIO is based on Open Sound Control - an emerging standard for interactive environments not only limited to musical instrument control - and can be therefore easily implemented on any platform that supports OSC.

Since the initial publication of the TUIO protocol specification to the public domain the protocol has been also adopted by several other projects related to tangible and interaction, such as the NUI group and several other tangible interaction platforms. The TUIO protocol has been widely adopted and used for multi touch interaction.

TUIO protocol definition is used to provide a general and versatile communication interface between tangible tabletop controller interfaces and underlying application layers. It was designed to meet the needs of tabletop interactive multi-touch surfaces, where the user is able to manipulate a set of objects and draw gestures onto the table surface with the finger tips. The objects are tracked by a sensor system and can be identified and located in position and orientation on the table surface.

The programming language is required to understand the TUIO protocol. This is because the touch data from the TUIO tracker application is communicated to the client application. The client application is required to understand this TUIO protocol. The data that is sent involves information like the touch position, touch ID, session ID. The touch tracker also notifies the programming environment when a new touch input is detected, a existing touch input is removed or an existing touch input is updated.

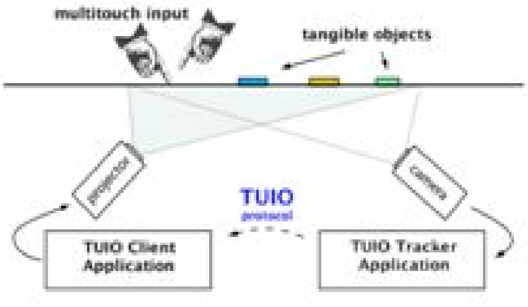


Figure 5.2: TUIO Protocol

Open Sound Control

The TUIO protocol is encoded using the Open Sound Control format, which provides an efficient binary encoding method for the transmission of arbitrary controller data. Therefore the TUIO messages can be basically transmitted through any channel that is supported by an actual OSC implementation. The default transport method for the TUIO protocol is the encapsulation of the binary OSC bundle data within UDP packets sent to the default TUIO port number 3333. This default transport method is usually referred as TUIO/UDP, and most implementations are based on this method due to its simplicity and speed when sent over a local or wide area network. Since OSC is not directly bound to a dedicated transport method, alternative transport channels such as TCP can be employed to transmit the OSC encoded TUIO data. As introduced with the TUIO 1.1 implementations, there are already several alternative transport methods available, such as TUIO/TCP and TUIO/FLC (flash local connection via shared memory) to interface with Adobe Flash applications for example.

Efficiency and mode of Commnication In order to provide low latency communication our implementation of the TUIO protocol uses UDP transport. When using UDP the possibility exists that some packets will be lost. Therefore, our implementation of the TUIO protocol includes redundant information to correct possible lost packets, while maintaining an efficient usage of the channel. An alternative TCP connection would assure the secure transport but at the cost of higher latency. For efficiency reasons set messages are packed into a bundle to completely use the space provided by a UDP packet. Each bundle also includes a redundant alive message to allow for the possibility of packet loss. For larger object sets, a bundle with a series of set messages, each including an alive message, are transmitted. When the surface is quiescent, alive messages are sent at a fixed rate dependent on the channel quality, for example once every second, to ensure that the receiver eventually acquires a consistent view of the set of alive objects. The state of each alive but unchanged object is periodically resent with additional set messages. This redundant information is resent at a lower rate, and includes only a subset of the unchanged objects at each update. The subset is continuously cycled so that each object is periodically addressed.

Attributes The Session ID is a unique identifier for each individual TUIO object, cursor or blob, which is maintained during its individual presence during a session. This unique ID allows to distinguish various markers with the same symbol ID or to identify individual unmarked blobs or cursors. The session ID usually is unique for all profiles, although using the same session ID within the cursor and blob profile for example indicates a reference to the same individual instance. This allows the additional transmission of the cursor or object geometry using the blob profile if desired.

The attributes defined in this section reflect the object properties we considered important for an interactive surface interface. Some of these attributes (ID, position and angle) are retrieved directly by the sensor. Others (speed, acceleration) are derived from these primary attributes using timing information. Computing these attributes on the low level side of a tangible user interface system is more reliable, since the necessary timing information does not need to be available to clients.

5.4 Ardunio IDE

The Arduino IDE[18] is a cross-platform application written in Java which is derived from the IDE made for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit Makefiles or run programs on the command line. The Arduino IDE comes with a C/C++ library called ”Wiring” (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++, although users only need to define two functions in order to make a runnable program: setup() a function run once at the start of a program which can be used for initializing settings, and loop() a function called repeatedly until the board is powered off. A typical first program for a microcontroller is to simply blink a LED (light-emitting diode) on and off. In the Arduino environment, the user might write a program like this:

```
#define LED_PIN 13

void setup () {
    pinMode (LED_PIN, OUTPUT);     // enable pin 13 for digital output
}

void loop () {
    digitalWrite (LED_PIN, HIGH);  // turn on the LED
    delay (1000);                  // wait one second (1000 milliseconds)
    digitalWrite (LED_PIN, LOW);   // turn off the LED
    delay (1000);                  // wait one second
}
```

The above code would not be seen by a standard C++ compiler as a valid program, so when the user clicks the ”Upload to I/O board” button in the IDE, a copy of the code is written to a temporary file with an extra include header at the top and a very simple main() function at the bottom, to make it a valid C++ program. The Arduino IDE uses the GNU toolchain and AVR Libc to compile programs, and uses avrdude to upload programs to the board.

Chapter 6

Hardware Implementation

There are two section of hardware implementation:

1. Touch table related hardware
2. Robot Hardware related hardware

This chapter on hardware implementation includes how the touch table is constructed, how the led circuit is designed, the modification of the visible light camera and designing of microcontroller compatiable motor controller daughter board.

6.1 Touch Table

The material required for the construction of the touch table and the material used are tabulated below:

Table 6.1: Hardware Components

Sr. no.	Component	Material used
1.	Screen	Acrylic Glass
2.	Projector	Epson EB-X6
3.	Projection Surface	Gateway Paper/ Tracing Paper
4.	IR Camera	Modified PS3eye
5.	Frame	Metal railings
6.	IR leds	IR 0510
7.	Adherents	Screws, Nuts, Bolts

6.1.1 Construction of Touch Sensor

The touch sensor is the surface on which the user interacts with the system. The touch sensor is an acrylic sheet of 10mm thickness. The thick is so chose to ensure that all the

IR light emitted by the IR led enter into the acrylic sheet and experiances total internal reflection within it. The acrylic was first cleaned wih soap water in order to avoid the effect of any fingerprint or dust present on the surface of the acrylic sheet.

This acrylic sheet is surrounded by an array of IR leds. The leds are sepearted by a distance of 1 inch. The sensor was first test with an array of five leds of different types. After analysing the result of these leds, IR 0510 was selected.

6.1.2 Construction of I.R Circuitry

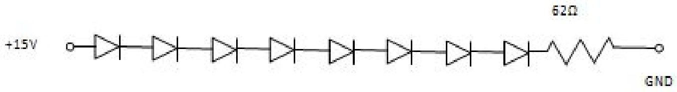


Figure 6.1: IR circuitry for 8 IR leds

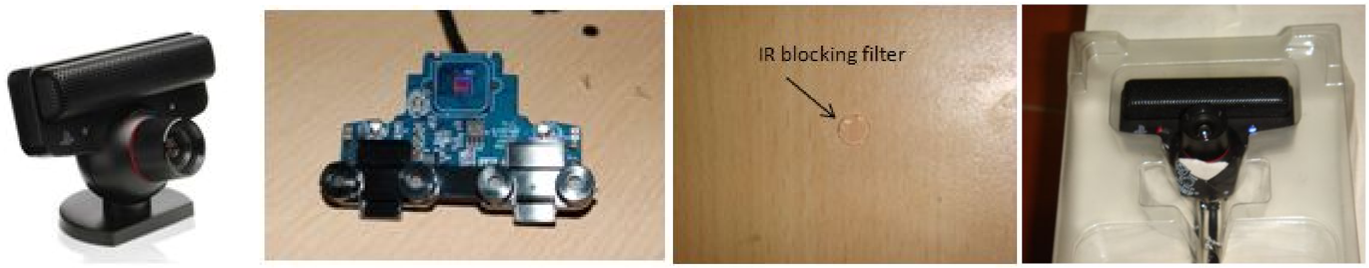
The above circuit was designed for eight IR led with the following specification:
Power supply voltage=15V
Power supply voltage=1.5V
LED current rating=50mA
Number of LEDs: 8
Each side of the acrylic glass has an accompanied IR LED circuitry so that the illumination throughout the screen is uniform. There are thus 4 strips of I.R LED, one for each side. Each strip has as many LEDs as the length of the strip, one for each inch of the strip. To keep the required current to a minimum, the LEDs on each side were separated into three groups of five. Now we had to give current and voltage to all LEDs using only one 15V supply. We calculated the current and voltage which each LED will get for different serial parallel combinations and then we chose the best one which required resistance of about 24ohms.

These groups were wired in series, connected in parallel to the power source. One circuit is then completely soldered and the adjustments are evaluated so that the LEDs fit into the enclosure. We then repeat this process for all the remaining sides then. In order to provide a convenient source of power and ground, cable is run up each side of the insides of the frames with the plastic coating stripped at key locations.

Once the LED circuits are wired and fixed in place, the circuit is powered and examined using a cell phone camera or other I.R sensitive image capture device. Once is verified that all the LEDs are working, the solder joints are double checked for stability. The webcam is then attached orthogonally oriented with respect to the acrylic directly beneath the centre point of the acrylic.

6.1.3 Modification of Visible Light Camera

We followed the following steps to convert a very good working camera into an Infra red Camera.



(a) Original Camera (b) Opened camera (c) IR blocking Filter (d) Modified camera

Figure 6.2: Camera modification process

1. Locate the small screws on the back of the device. Remove it and set it aside.
2. Use a flat-head screwdriver or any thin, sturdy implement on hand to pry the camera open. Make sure to do this evenly around the unit, or you risk cracking the plastic. It is necessary to apply some force for the device to open.
3. Find the lens apparatus and unscrew it until you can safely remove it. It is critical to avoid getting dust on the (soon to be exposed) photo sensor. If small dark dots appear on your image, it is likely caused by dust on the sensor.
4. Find the glass disc on the rear of the lens apparatus with a red hue. This is the infrared filter, and is very hard to remove intact. It is not necessary to remove in one piece as it will not be used again, but one must be wary of scratching the lens underneath.
5. Once the filter is removed, simply reassemble the camera.
6. Using the IR-pass filter, affix a stack of 5 to 10 layers to the outer lens of the camera. This will only allow infrared light to pass through. After preparing IR webcam when it is placed below acrylic sheet it only detects IR Light.

6.2 Robot Hardware

The robot related hardware include the following:

1. Motor: Four mototrs are used such that two motor on each side are shorted. the motor are 100 rpm motor
2. Motor driver: The motor controller used is L293D. The motor driver is used to supply the required current to the motor.
3. Programming board and daughter board
4. Zigbee module

6.2.1 Designed board

The desingned board as described in the hardware description section is interfaced to the programming board. The interfacing is done by using the input from the microcontroller PORTB and PIN number 2,3,4 and 5 to the motor driver input pins. The output of the motor driver input pins is connected to the respective motor. The Xbee modules require a 3.3V supply. The Xbee is placed on the board that converts 5V at its input to 3.3V. The XBee module is interfaced to the microcontroller using four pins: RXD, TXD, VCC and GND.

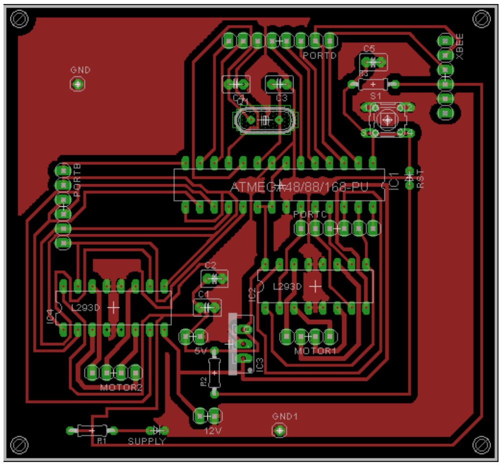


Figure 6.3: Designed microcontroller board

6.2.2 ZigBee Modules

The zigbee communication is implemented using xbee modules. There are two modules that are used:

1. One is at the computer end which transmits the data to the robot
2. Second is on the robot which receives the data and decodes it to get the distance and angle of rotation

As the system has only two zigbee communication modules, the transmission and reception of data can be done using simple serial communication method and using the Serial library. If there are number of Zigbee modules then it is required to configure the xbee module for there particular address. This is to be done in order to ensure that the data is received by the addressed xbee module. In case of multiple modules the configuration and communication is done by AT commands.

Chapter 7

Software Implementation

7.1 Libraries

The programming of the system is done in Java based processing environment and using libraries that use the pure java language. The libraries include the use of the following:

1. **TUIO**
   This library is required so as to decode the touch points and the data related to it. The library is capable of decoding the data from many blob tracking environment. The library can identify touch points and fiducial objects. Whenever a touch point(Touch cursor) is added, removed or updated corresponding callback method is called by the system. The callback methods are listed below:

   ```
   void addTuioObject(TuioObjecttobj)
   void removeTuioObject(TuioObjecttobj)
   void updateTuioObject (TuioObjecttobj)
   void addTuioCursor(TuioCursortcur)
   void updateTuioCursor (TuioCursortcur)
   void removeTuioCursor(TuioCursortcur)
   void refresh(TuioTimebundleTime)
   ```

2. **Video library**
   The video library is used to capture the data from the over head camera. The video library has a constructor that creates the object of this library. The construction is as follows:

   ```
   Capture(parent, width, height)
   Capture(parent, width, height, fps)
   Capture(parent, width, height, name)
   Capture(parent, width, height, name, fps)
   ```

```
Parameters  Width int: width of the frame
Height int: height of the frame
Fps int: number of frames to read per second
Name String: name of the camera
```

The video library allows Processing to display QuickTime video files, grab video data from a camera, and make QuickTime videos directly from a running program. Video can be captured from USB Cameras, IEEE 1394 (Firewire) Cameras, or Video Cards with composite or S-video input devices connected to the computer. Video can be loaded from QuickTime files located on your computer or anywhere on the Internet. For this library to work, it is necessary to have QuickTime installed on your computer. Download and install QuickTime (or iTunes, which includes QuickTime). If you are using a Windows operating system, you will also need a video driver that translates from your hardware to the video functions used by QuickTime. This software may be included in the software provided by the maker of your hardware, or you can download the software. Search for ”vdig” using an online search engine to find the software.

3. **Artoolkit library**
   This library is used to detect the robot in the workspace of the over head camera. As the programming language used is Java, it is required to create an object of the class defined in the Artoolkit library. The Artoolkit library is derived from the original Artoolkit library written in C language.

```
public Nyarboard (Processing.core.papplet parent, I_Widthint, intI_Htight,
I_Cparamjava.lang.String, java.lang.StringI_Patt, intI_Patt_Width)
```

This constructor is used to initialize the artoolkit. The various parameters involved are:

```
Parameters:
parent- Specify processing instance.
i_width- Width of source image size for "detect ()".
i_htight- Height of source image size for "detect ()".
i_cparam- The file name of the parameter of ARToolKit format Camera.
i_patt- The file name of the file marker pattern of ARToolkit.
i_patt_width- The length of marker in Millimeter Square One Side of a unit.
```

These are the major libraries used to make the project work.

7.2 Algorithm

The software algorithm is divided into 6 major steps. These steps are as follows:

1. Interface the camera to the running program and obtain an image of the area within its work space. This is done using the video library explained in the previous section. The image is processed to check and find the position of the marker, if it is present. Once the position is obtained, the center point of the marker is found out by taking the average of all point vertices of the marker.

2. The next step is to create an robot. The robot is a class that has been created by us. The class has various parameters. These parameters include the actual coordinate of the robot in the workspace. The parameters also include the robot ID. With this class it is possible to add number of robot. The number of robot corresponds to the number of different markers in the workspace.

3. Once the robot has been created, the path of the robot has been created; the path on which the robot has to be travelled is to be defined. This path is similar to the path of the touch point or touch cursor. Thus it necessary to get the touch path. This path is then stored so that it can be analyzed later.

4. The next step is to analyze the path, so as to obtain the direction in which the robot has to move and the distance it has to travel. This step is a bit complex as compared to the other steps. The direction is obtained by dividing the workspace into four quadrants and comparing the current angle to the previous angle.

5. Once angle and distance have been obtained, the next step is to send these data to the robot using the Zigbee modules.

6. Once the data has been sent the data is analyzed at the robot to obtain the actual direction and distance.

7.3 Detection of robot

We are using ARToolKit software library to detect robot on ground. ArToolKit is a C and C++ language software library that lets programmers easily develop Augmented Reality applications. Augmented Reality (AR) is the overlay of virtual computer graphics images on the real world, and has many potential applications in industrial and academic research. One of the most difficult parts of developing an Augmented Reality application is precisely calculating the user's viewpoint in real time so that the marker mounted on our robot can detected easily with fast and precise tracking. ARToolKit uses computer vision techniques to calculate the real camera position and orientation relative to marked cards, allowing the programmer to overlay virtual objects (circle in our case) onto these cards.

7.3.1 Camera Marker Relationship

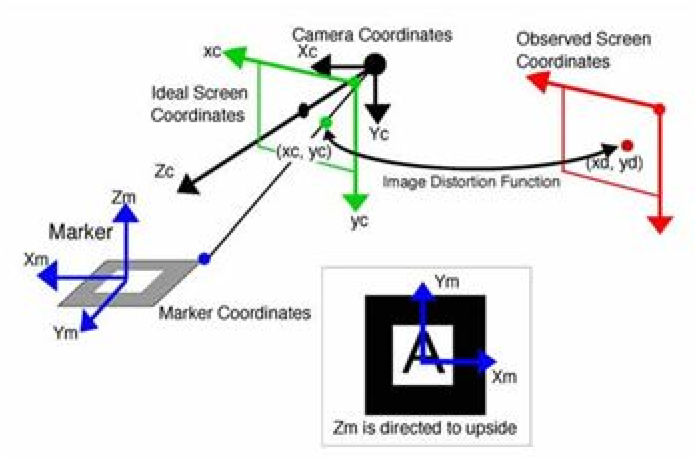


Figure 7.1: Camera Marker Relationship

The coordinates of the robot in the real world and the coordinates obtained by processing the image taken by the camera are not same. The relation between them is called as camera marker relationship. For this reason camera calibration has to take place. The software library used in this project, ARToolkit, provides with certain function that help in finding this relationship. This function makes use of the transformation matrix to get the actual coordinates of the robot. The tranformation matrix gives the relationship between the camera coordinate system and the marker or robot coordinate system.

ARToolKit defines different coordinate systems mainly used by the computer vision algorithm and the rendering. One needs to notice that arGetTransMat give the position of the marker in the camera coordinate system (not the reverse). If one want the position of the camera in the marker coordinate system you need to inverse this transformation (arMatrixInverse()).

7.3.2 Marker Recognition- Basic Principle

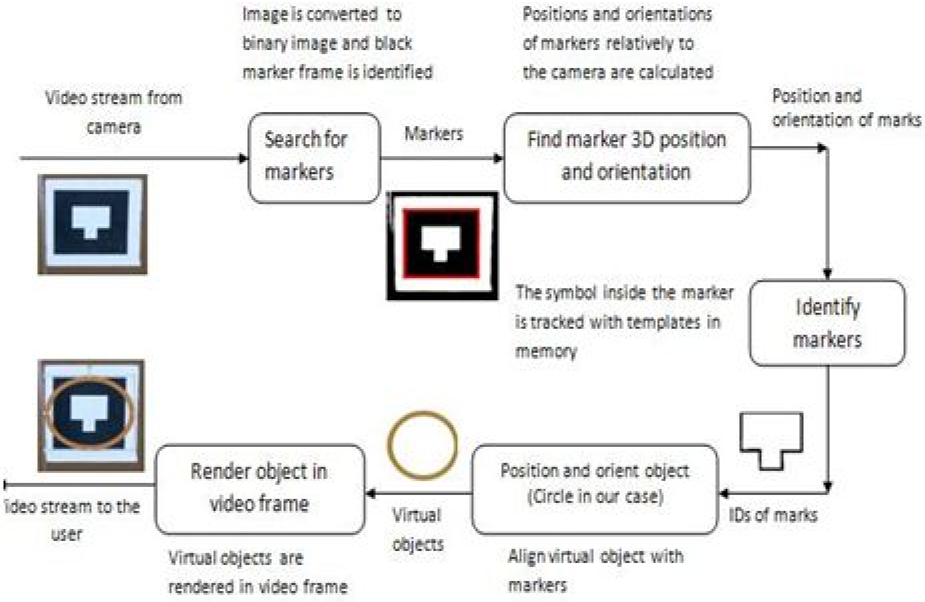


Image is converted to binary image and black marker frame is identified
Positions and orientations of markers relatively to the camera are calculated
Video stream from camera
Search for markers
Markers
Find marker 3D position and orientation
Position and orientation of marks
The symbol inside the marker is tracked with templates in memory
Identify markers
Render object in video frame
Position and orient object (Circle in our case)
ideo stream to the user
Virtual objects
IDs of marks
Virtual objects are rendered in video frame
Align virtual object with markers

Figure 7.2: Marker Recognition

ARToolKit applications allow virtual imagery to be superimposed over live video of the real world. Although this appears magical it is not. The secret is in the black squares used as tracking markers. The ARToolKit tracking works as follows:

1. The camera captures video of the real world and sends it to the computer.

2. Software on the computer searches through each video frame for any square shapes.

3. If a square is found, the software uses some mathematics to calculate the position of the camera relative to the black square.

4. Once the position of the camera is known a computer graphics model is drawn from that same position.

5. This model is drawn on top of the video of the real world and so appears stuck on the square marker.

6. The final output is shown back in the handheld display, so when the user looks through the display they see graphics overlaid on the real world.

The figure below summarizes these steps. ARToolKit is able to perform this camera tracking in real time, ensuring that the virtual objects always appear overlaid on the tracking markers.

7.3.3 Marker Recognition- Program Description

The algorithm is to detect the desired marker in the captured image and indicate the same to the user when the marker has been detected. Once the marker has been detected the next step is to draw the virtual object on top of the marker. The object we chose here is an ellipse. The center of the ellipse is calculated with the help of the four corner points that are obtained. The coordinates of the vertices are averaged to obtain the center of the marker. With this center and ellipse is drawn which has the diameter equal to the width of the marker. This step helps in indicating the position of the marker on the screen. Once the center point of the marker has been found a robot object is created. The robot is created is of type class Robot. The advantage of creating a special class for the robot is that it enables to create any number of robot objects. The number of robot class is equal to the number of markers in the area that can be captured by the over head camera. The following code snippet indicates how we have created a robot object:

```
r=new Robot(x,y,id);
where, r if of type Robot,
       x and y are the current robot coordinates,
       id is the id of the robot that has been created.
```

The id of the robot can be dependent on the current number robot in the workspace. In this case we have kept the id constant to 1, as there is only one robot involved.

7.4 D.C. Motor Calibration

7.4.1 Need for using DC Motors

In the maneuvering of the robot, it is imperative that robot moves a distance which corresponds to the dragged path on the touch screen. For this purpose, we may use either stepper motors or calibrate dc motors such that they move a specific distance when supplied a specific voltage for a given period of time. There are two reasons why stepper motors where not used in the project:

1. The high torque unipolar motors are usually costlier than the dc motors having the same torque.

2. In bipolar motors, the internal resistance between two terminals of the motor is shorted when the data between the terminals changes from 0 to 1 and vice versa very quickly. If we add a resistor between the two terminals , the current greatly reduces. As an example, for a 100Ω resistor that was connected between the terminals reduced the current from 1A to a very low value. To supply a higher value of current, we would require a more powerful battery. This again increases the cost.

Thus, to make the project as cost effective as possible, we calibrate the dc motors so that we can control the distance that they traverse.

7.4.2 Calibration Process

In order to calibrate the DC motor, it is required to plot the distance traversed by the dc motors versus the time for which a given voltage is supplied. It is important to note here that the voltage across the motors and the current given to it must be constant with respect to time. For the dc motors, calibration of two parameters we required for successful maneuvering.

1. Distance
2. Angle

Angle Calibration

Angle calibration involves calibrating the angle by which the robot rotates. Here, we had given the dc motors a constant supply of 8.5 volts. Then, for various values of time in seconds, the corresponding distance covered by the motors was measured in centimeters. The readings were tabulated as follows: Constant voltage V=9.0V

Table 7.1: Angle readings

Time(ms)	Angle (degrees)
15	2
20	2.2
35	3
50	10
100	20.5
250	31.429
465	56.976
600	87.825
1050	147.72
1400	208.69
2010	298.30
2500	378

The equation obtained after curve fitting is: $-0.001559x^2 + 7.186x + 1.569$

Distance Calibration

Here, similar to the angle calibration, a specific voltage was given for a specific time and the robot was rotated by a corresponding angle. The tan inverse of the angle was then calculated and then the corresponding angle was noted down. The tabulation, formula derivation and curve fitting was similar to process in done in distance measurement. The readings table, formula and the snapshot of the curve fitting GUI are as follows:

Constant voltage V=9.0V

Table 7.2: Distance readings

Time(ms)	Distance travelled(cm)		Final value(cm)
15	0.4	0.4	0.4
35	1.2	1.1	1.15
50	2.5	2.5	2.5
75	2.8	2.9	2.85
200	6	6.2	6.15
495	16	16	16
600	19.5	19	19.25
1100	33.5	34	33.75
1502	48	48	48

The equation thus obtained after curve fitting is: $-0.02495x^2 + 32.97x - 12.52$

After the measurements were tabulated, they are used as input for the purpose of curve fitting. The distance was taken on the x axis and the time on the y axis . The curve was fitted for the degree 2 that is a quadratic polynomial. The higher orders were neglected because their coefficients were negligible.

7.5 Program Description

Once the relation between angle of rotation and time and the relation between distance covered and time has been found, these relation are to be implemented into the program. The relation is implemented in the form of function which returns an integer according to the relations obtained in the section DC calibration (Section 7.3). The user defined functions that are used in the function are as follows:

1. For distance

   ```
   int di(int d);
   where d is the distance to be calibrated
   returns a calibrated value which is an  integer
   ```

2. For angle

   ```
   int an(int a);
   where a is the distance to be calibrated
   returns a calibrated value which is an  integer
   ```

7.6 Embedded Programming

The robot is controled by the microcontroller, Atmega328. The programming of the microcontroller is done in Arduino IDE(developed on Java). The programming of Arduino is similar to C language and is also similar to Codevision environment. The syntax is not the same as many embedded programming environment. The syntax is explained in section 5.4. The programming is then converted to C language by adding main() to the created program. The IDE also allows to programming in the manners used by other Embedded Programming Environment like CodeVision.

7.6.1 Programming the Microcontroller

The microntroller is programmed using USB interface. The programming board called Arduino USB board has a FTDI chip which assists in USB communication with the microcontroller. The FTDI chip helps to migrate to USB by combining USB-Serial communication using FT232 chip.

The facility of USB programming was one of the reason why Arduino board was chosen for the project.

7.6.2 Robot Programming

The program of the microcontroller need to take care of the following major operation that robot has to perform:

- Ensuring data sent by the touch system is received
- Decoding the data
- Turning the motor on or off

Data communication

As there is only a pair of Zigbee modules involved in the system, the communication between them can be carried out using usual serial transmission and reception commands. If the number of modules involved are more then it may become necessary to use AT commands to ensure proper communication.

The data that is transmitted by the touch system is byte-by-byte communication. For example, if a string ”ab” is to be transmitted, first ’a’ will be received and the ’b’ will be received. The data that received can be stored as an integer or a char. In case of integer an ASCII value is stored.

The data, distance and angle, was transmitted using the following format:

```
dDistance,aAngle.
where Distance is an integer value
      Angle is an integer value
```

When the data is received is received following steps are taken:

1. The byte is checked for the type, that is, numder or alphabet or special character
2. If it is an alphabet, then number following it is stored as in respective variable to indicate angle or distance
3. A comma(,) indicates the current data, distance or angle, has been completely transmitted
4. A fullstop(.) indicates the a complete set, distance and angle, has been transmitted and the motor should run in that direction.

Motor control

Once data has been received the motor is turned on for a particular period of time. This value is same as the alue received by the microcontroller. After every forward, right or left operation the motor is stopped for a minimum of 100ms to nullify the effect of the momentum gained by the motor.

Chapter 8

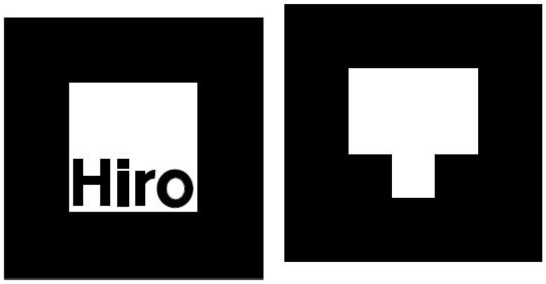
Results and Conclusion

This chapter discusses the various results related to marker detection, wireless communication, Multi touch table and its components. The final section draws a conclusion based on the proposed system and the final outcome of the project.

Results

The result of the final year project are as follows:

1. **Marker Detection**:
   During the course of the project we have tried different markers. Some of them have been shown below: All these marker gave satisfactory result for the purpose of robot detection. The problem in detection arises when the lighting condition are not good. The detection process is affected by very low intensities and very high light intensities.



(a) Original Camera (b) Opened camera

Figure 8.1: Different markers

   The quality of camera also plays an important role in the marker detection process. The camera we tested includes Microsoft VX-3000, HP camera, Sony PS3Eye and Logitech QuickCam. Out of all these PS3Eye gave the best result. It was least affected by lighting condition and the calrity was superior as compared to other camera. The

result of marker detection with respect to various angle are shown below:
As can be seen from the above list of snapshots that the marker is detection gives a satisfactory results.

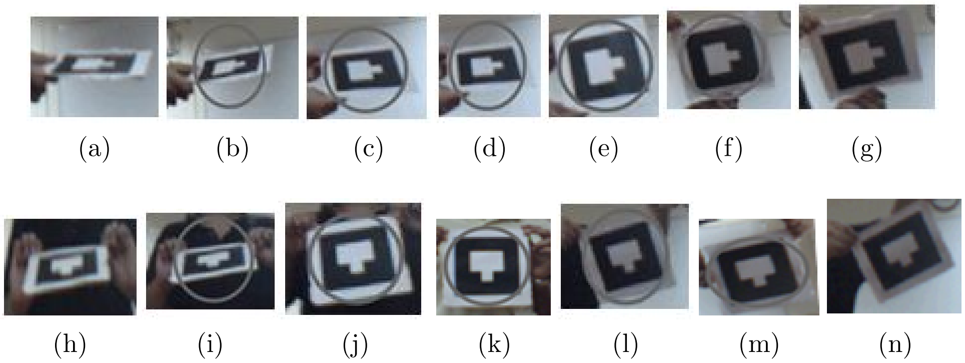

(a) (b) (c) (d) (e) (f) (g)
(h) (i) (j) (k) (l) (m) (n)

Figure 8.2: Marker detection in different orientations

2. **Wireless Communication:**
   The communication between Multi touch System and the robot involved continuous transmission of data set, which contain distance to be covered and angle through which the robot is to be rotated.

   The process of implementing(rotating the motor) the decoded data requires maximum of around 2 seconds. Thus, there is a possibility that the transmission or reception buffer will overflow. During the experimentation with different sets of data, we found that if number of data sets is greater than 10, then there is a possibility that the data set and the consecutive data sets will not be implementated. This happens because the default buffer size in the serial library is 128 bytes. The format described in section 7.6.2 can lead to data set of maximum size 12 byte which inturn corresponds to maximum of 10 data set at a time.

   The solution to this problem was to either increase the buffer size or involves hand shaking signal between robot and multi touch system.

   The Xbee communication was successfully tested to send and received data to and from about more than 10m.

3. **Multi Touch System**

   (a) IR Led
   We have tested led of wavelength 850nm and 940nm. The IR camera uses floppy disk as a visible light filter. During the testing process, the 940 nm led did not give satisfactry result as the detection of blob was difficult and a lot of pressure had to be applied in order to create a touch point. But as compared to 940nm led 840nm led requires less pressure to create a blob.

(b) Compliant surface

In order to reduce the presure required to reduce a touch point compliant surface is used. The compliant surface is a projection surface(tracing paper)coated with mixture of clear silicon sealent and thinner. As the number of coating of this mixture increases, the sensitivity of the touch surface improves.

(c) Mirror

The projection from the projector is first incident on the mirror. This step helps in reducing the height of the touch table. But the thickness of the mirror plays an important role in the quality of image displayed on the projection surface. Because of reflection taking place from surface of the mirror the quality of the image on the projection screen is distorted. Thus, greater the thickness better is the visibility of two distinct images and thus weaker is the quality of image.

We tested the quality with a 5mm and 3mm thick mirror. The 3mm thick mirror gave clear and best result among the two.

4. **Robot movement**

(a) To maintain the calibration of DC motors it is imperative to give constant DC supply to the motors. If the DC supply is varied $\pm 0.5V$ then the calibration is not affected.

(b) During the testing, robot was moved through various angle and we found that for angle above 90° the robot did not obey the calibration equation. The error in calibration increased as the angle increased toward 360°. We found that this happened due to the momentum gained by the DC motor during the on time. Thus, the movement of the robot was restricted to a maximum of 180°.

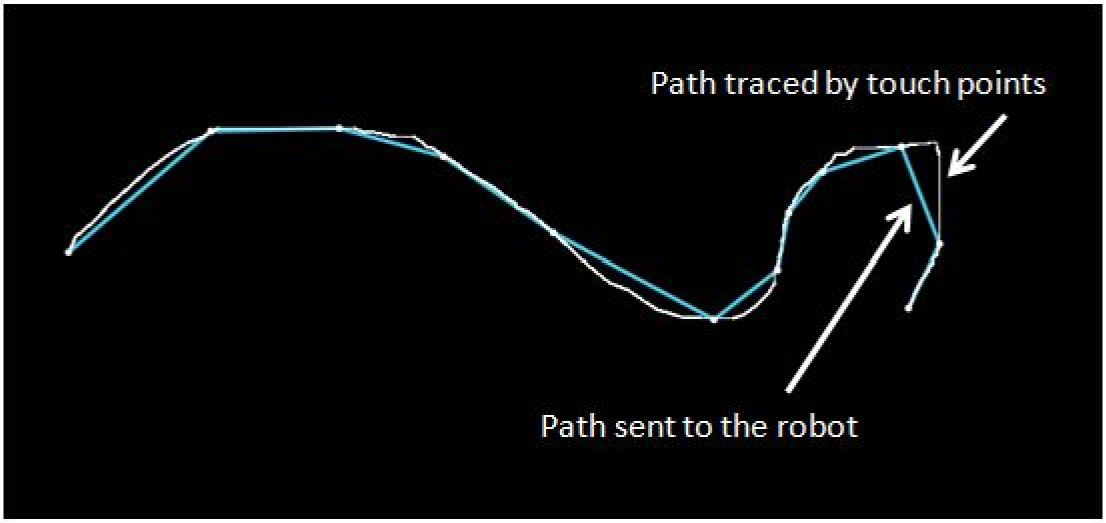


Figure 8.3: Difference between traced and original path

(c) The smooth path of our fingers is broken into series of straight line motion for easy dc motor control. Hence we do not get a smooth path. The path is created by sampling the position of our fingers. If we take low sampling period we get more points and hence better reconstruction, however the Atmega328 has limited buffer space and hence large number of points will result in data overflow and hence, data loss. Hence we use the maximum sampling points that shall not cause buffer overflow. The figure 8.3 illustrates the idea. This figure is a snapshot of the program that the group has created.

Conclusion

The proposed system for the final year project was to design a touch based system that enables to control one or more robot using wireless communication.

We have succesfully implemented a PC based Multi Touch System and a Wireless robot. We have test and interfaced wirelessly one robot with the multi touch system. Though we have not interfaced more than two robots to the Mult touch System but we have been able to detect two distinct marker at the same time.

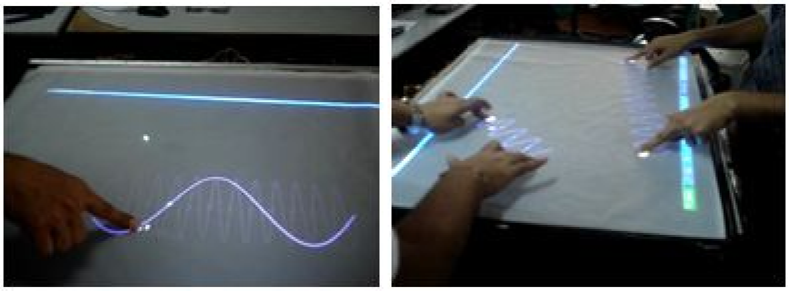
We have also created some basic and intermediate level touch applications and a touch presentation in Java.

Though the proposed system is partially fulfilled but the PC based Multi Touch System has been succefully implemented. The system is called as low cost because total cost involved in the creating this system excluding the projector is around Rs. 6000. This is cheap compared to the capacitive touch screens available in the market of the same dimension.

Appendix A

Additional Programs and Project Pictures

1. **Sine wave**

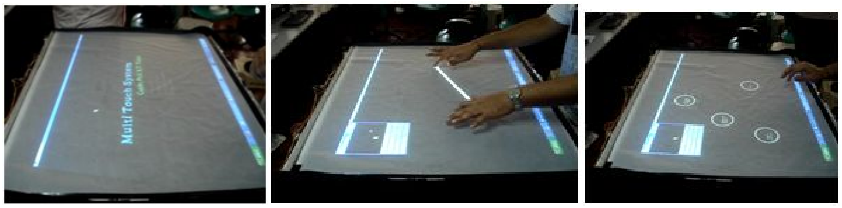


(a) (b)

Figure A.1: Snapshot of two different sine wave related programs

This program was created with an intention to show how the touch interface can improve the quality of interaction with a compute system. The program recognizes a straight line gesture and creates a sine wave whos amplitude and frequency can be changed using touch gestures. The natural user gesture like moving the finger upwards on the screen increases the amplitude or moving it down decrease the amplitude. Similarlly, moving the finger toward right or left increases or decreases the frequency

2. **Touch Presentation**
This was the program created specifically for the project presntation. The presentation is controlled by single touch and multi touch gesture[11]. These geture are designed to help the user move from one slide to another. The presentation gives an idea of the various section of the project.



(a) (b) (c)

Figure A.2: Snapshot of the Presentation

3. 3D
 This program can be used to control the orientation of a box. The box can be rotated in x,y or z direction using the touch points.

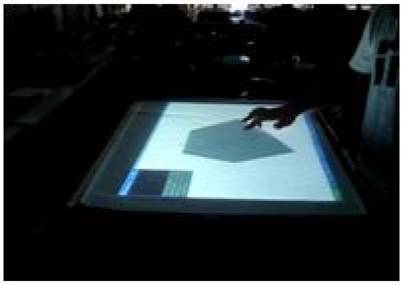


Figure A.3: Snapshot of program playing with box

4. **Photo application**
 The photo application perform the drag and drop operation. It also ha the ability to zoom in and zoom out of a particular photo using two fingers.

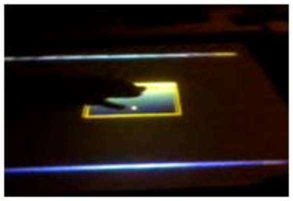


Figure A.4: Snapshot of photo application

There are various possibilites and application possible with multi touch system[9]. Developers have been developing games like pong which is a multi player game. pong is a flash game. Multi touch programming and game development can be done on various platforms and using verious textual and visual programming languages like C, C++, Java, vvvv and many more.

5. **Some Project Pictures**

These snapshot indicates the actual robot, the overhead camera and user drawing the path for the robot to move on. These snapshots are the inside and the outside view of the multi touch table. It also shows the arrangment of various componenets described in the report like mirror, IR camera, XBee module and Touch sensor.

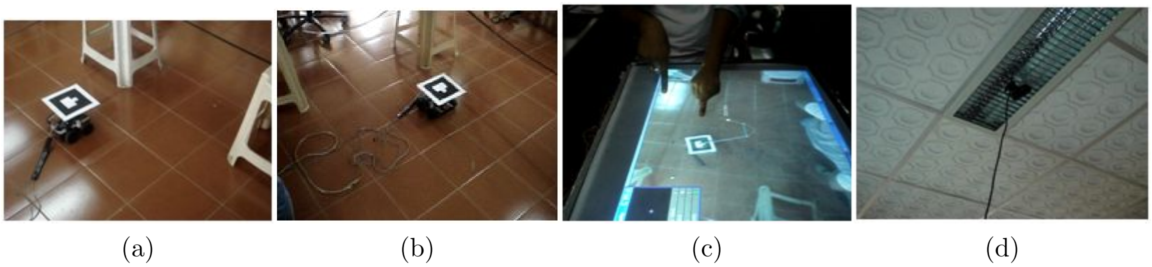
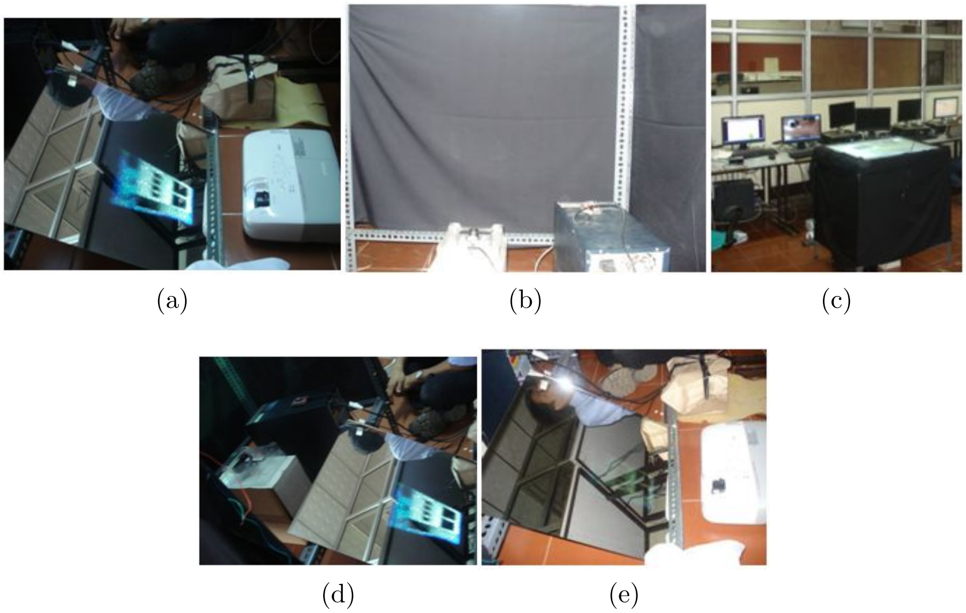

(a) (b) (c) (d)

Figure A.5: Snapshot of the Project


(a) (b) (c)

(d) (e)

Figure A.6: Snapshot of Arrangement of Touch Table

Appendix B

Designed Atmega328 board

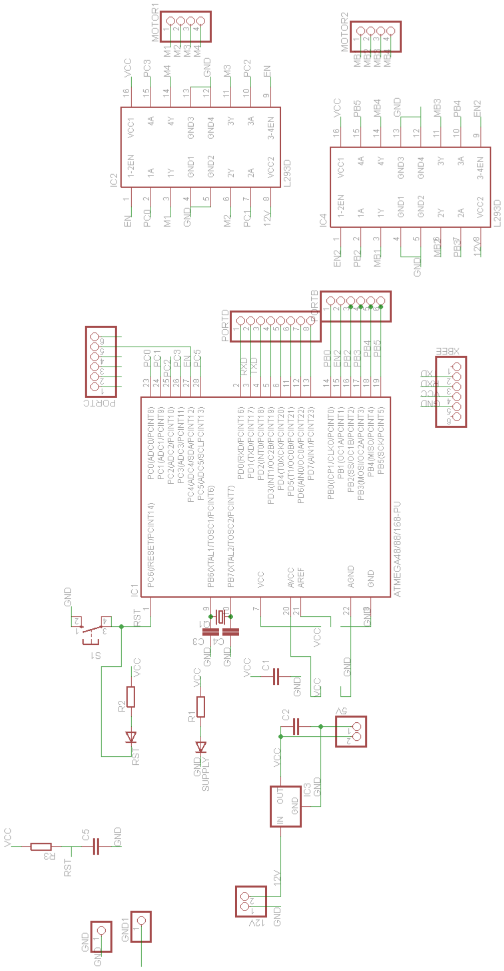


Figure B.1: Designed Atmega328 board

Bibliography

[1] Uwe Hahne, Marc Alexa, Bjorn Bollensdorff, Ingo Bressler, Stefan Elstner,2008,A FTIR based Multit Touch Table Using Pulsed Illumination for noise Suppression, TU Berlin

[2] D.Michl, A.A.Argyos, D. Grammenos, X. Zabulis, T. Sarmis, 2009, Building a multi-touch display based on computer vision techniques Institute of Computer Science,Greece

[3] Christopher Wolfe, J. David Smith and T.C. Nicholas, A Low Cost Infrastructure for Table Top Games Graham School of Computing, Canada

[4] Philip L.Davidson, Jefferson Y. Han,Synthesis and Control on Large Scale Multi-Touch Sensing Displays , New York University

[5] J. David Smith,T.C. Nicholas Graham,Low-Cost Malleable Surfaces with Multi-Touch Pressure Sensitivity , School of Computing, Queens University

[6] Jefferson Y. Han, 2005, Low-cost multi-touch sensing through frustrated total internal reflection, UIST '05

[7] Jefferson Y. Han, 2006, Multi-touch interaction wall, ACM SIGGRAPH 2006 Emerging technologies, Boston, Massachusetts

[8] TODD BISHOP, May 2007,Microsoft Surface brings computing to the table,
$http : //www.seattlepi.com/business/317737_{m}sftdevice30.html$

[9] Bradley H. Hayes, 2008, Software Driven Multi Touch Input Display as an Improved, Intuitive, and Practical Interaction Device, Undergraduate Honors Thesis ,Computer Science Department, Boston College

[10] Microsoft Co., 2008, Microsoft Surface, http://www.surface.com

[11] Micheal Thornlund, 2007, Gesture Analysizing for Multi touch screen interface, Luea University of Technology

[12] B. Buxton, Multi-Touch Systems that I Have Known and Loved, http://www.billbuxton.com/multitouchOverview.html

[13] William C. Crai, Spring 2008, Zigbee: Wireless Control that Simply Works, Program Manager Wireless Communications, ZMD America

[14] NUI Group Authors,1st edition [Community Release]: May 2009, Multi-Touch Technologies

[15] Sudhanshu, Rahulmotiyar, CHEAP AND EASY COMPLIANT SURFACE, Team Sparsh,www.sparsh-i.com,India

[16] Brian W. Evans, Arduino Programming Notebook

[17] OSRAM Opto Semiconductors, SFH 485P Datasheet, http://www.osram-os.com

[18] www.ardunio.cc, Official website for Open Source Arduino Hardware

[19] www.processing.org, Offical website Processing environment and language