

VISION BASED BALL COLLECTING ROBOT

Degree Project
Report



Muhammad Zohaib Aziz Khan
01-133072-189

SUPERVISOR: MR. BILAL ASHRAF AWAN

DEPARTMENT OF COMPUTER ENGINEERING
BAHRIA UNIVERSITY ISLAMABAD



BAHRIA UNIVERSITY ISLAMABAD

Dated: 30/03/2012

CERTIFICATE

We accept the work contained in the degree project report titled “Vision Based Ball Collecting Robot” as a confirmation to the required standard for the partial fulfillment the degree of BCE.

Project Coordinator
(Junaid Imtiaz)

Supervisor
(Mr. Bilal Ashraf Awan)

Internal Examiner
(Dr.Usman Akram)

External Examiner
(Mr. Akhtar Khan)

Head of Department
(Mr. Bilal Ashraf Awan)

ACKNOWLEDGEMENTS

First of all I would like to thank Allah the all Mighty and the most Merciful. Then I would like to thank all of my teachers who enabled me through the light of knowledge. I completed this project by utilizing different parts of that knowledge. All of them were always kind to me and willing to help out whenever needed. I would like to specially thank Mr. Bilal Ashraf Awan for his continuous and kind support. Other teachers who honored me with their valuable advices include Mr. Usman Akram and Mr. Haroon Rasheed.

As always, I would like to thank all of my family with the depth of my heart for their endless support, encouragement and prayers.

Muhammad Zohaib Aziz Khan

April 2012

DEDICATED

I would like to dedicate this project to all of my teachers, my family, my friends and other well wishers.

ABSTRACT

With the advancement in electrical and mechanical engineering, a new form of machines came into scene; called Robots. A Robot is a combination of electronics, mechanical design and software. If we see the modern world around us, we will find many machines/robots which are intelligent and make our life very easy. So are the vision based machines. In modern manufacturing plants we can see them working and checking the proper filling of bottles. If the bottle is not properly filled then that bottle is picked out automatically. Also the vision based robots are used to hover on the wings of Boeing aircraft during its manufacturing process in order to find cracks in it if there are any. The Robo Cup is also a world famous tournament in which robot teams from all over the world take part and the football is played solely by robots using vision. Hence there is very wide application of vision based robots in all the fields of life. Therefore I would like to work in this area of engineering for my final year project.

The **aim** of this project is to develop an automated robot that is capable of recognizing red balls using vision and be able to collect them in a totally automated way.

The **approach** used in for this project was first brain storm different system level solutions keeping in view of the requirements of the project. The problem is divided into two major categories, one is the development of the control system of the robot which include the embedded system, and design of robot in order to achieve required functionality. Second part is related to the acquisition of the video from the robot camera and performing the image processing for the required objects or according to requirements. In our case we need to recognize the red balls. After each image processing, control signals are generated accordingly that tells the embedded system of the robot what movement(s) should the robot perform next.

The **result** of the employed technique is that I have successfully designed and implemented the robot. The hardware and software work in perfect coordination.

It is **concluded** that the software and hardware work in coordination to achieve the needs of the robot. It is capable of performing the required operations of scanning the surroundings using vision and collecting the red balls.

Table of Figures

Figure 1: L7895 DC voltage regulator.....	15
Figure 2: System view of a microprocessor (LEFT) and microcontroller (RIGHT)..	16
Figure 3: AT89c51 microcontroller in PDIP package.	17
Figure 4: RS-232 port.	19
Figure 5: Simple Serial Communication.	19
Figure 6: Three form of implementation of serial communication according to the requirement.	20
Figure 7: Use of MAX232 with 8051 microcontroller.	21
Figure 8: Symbol of a voltage comparator.	21
Figure 9: LM393N voltage comparator IC.	21
Figure 10: L293D DC motor driver IC.....	22
Figure 11: Chaparral Servo Motor.....	23
Figure 12: Position of servo motor with respect to duty cycle of the PWM signal....	24
Figure 13: Infrared transmitter (BELOW) and infrared receiver (ABOVE).....	25
Figure 14: Pixel intensity values within the indicated red box in image.....	25
Figure 15: The three planes of red, blue and green combine to make a color image.	26
Figure 16: Effect of filtering on an image.	26
Figure 17: We can see a gray scale image on left and its binary image created using threshold value at right.....	27
Figure 18: the graph of thresholding.....	27
Figure 19: Effect of dilation using 3x3 structuring element on original image left and dilated image on right.	28
Figure 20: Filling of gaps by dilation.	28
Figure 21: Effect of erosion using 3x3 structuring element on original image left and eroded image on right.	29
Figure 22: Splitting the objects and removing extrusions by using erosion.	29
Figure 23: Opening operation. Original Image (LEFT). Eroded Image (Centre). Dilated Image (Right).	30
Figure 24: Break down of the project into its subprojects.....	32

Figure 25: Screen shots of robot design in AutoDesk Inventor.	34
Figure 26: Actual Robot.	35
Figure 27: Screen shot of the embedded design in Proteus.	36
Figure 28: Simulation of the electronic design.	36
Figure 29: Final schematic design.	37
Figure 30: the LM393N comparator circuit.	38
Figure 31: L7805CV regulation circuit.	38
Figure 32: Final PCB design.	39
Figure 33: Original Image.	48
Figure 34: Red Frame of the original image.	48
Figure 35: Green plane of the original image.	48
Figure 36: Blue frame of the original Image.	49
Figure 37: Binary Image of the original image.	49
Figure 38: Image after Erosion Morphological Operation.	49
Figure 39: Image after Closing Morphological Operation.	50
Figure 40: Image after Opening Morphological Operation.	50
Figure 41: Image after Close Morphological Operation.	50
Figure 42: Opening window of the GUI.	51
Figure 43: Control window for the robot, each box in the window shows real time processed images.	51

Table of Contents

1. Introduction	10
1.1. Project Motivation.....	10
1.2. Project Objectives	11
1.3. Project Aims.....	11
1.4. Project Requirements	12
1.5. Project Challenges.....	12
1.6. Thesis Organization.....	13
2. Literature Overview	14
2.1. Background	14
2.1.1. Embedded Control System	14
2.1.2. Digital Image Processing	25
3. Project Methodology	31
3.1. Methodology and Tool Support	31
3.1.1. Approach to Problem	31
3.1.2. Collection of Tools	32
4. Project Specifications	34
4.1. Robot Design Development	34
4.2. Embedded Control System Development	35
4.3. PCB Development.....	39
4.4. Embedded programming	39
4.5. Image Processing Program.....	43
5. Future Work	52
6. References.....	53

1. Introduction

1.1. Project Motivation

From the beginning of 20th century the world has seen a tremendous progress in the field of electronics. Hence is the case with the computers. They started from mechanical devices such as computers of Babbage, went through the usage of vacuum tube like Electronic Numeric Integrator and Computer (ENIAC) in 1946. It was the first electronic computer and required a significant amount of space of more than a normal room. The transistor revolutionized everything and electronic device became smaller, cheaper and less power consuming with a gain of more performance and speed. Finally the development of integrated circuits (IC) made those things possible which once could only be seen in fiction movies. It also contributed in the breakthrough of computers and their computing power. The result of this is that we see digital electronics and computing systems in almost every field of life. Especially the use of embedded computing systems is ubiquitous. Modern machinery and robots are controlled by such computing systems and have made the impossible to possible. These little computers are far more powerful than those built earlier in 1900's.

One of the trends in modern robots is the use of vision system. A lot of work is being carried out in this area. We also see many practical robots working in industry that are based on vision. Vision and image processing requires a lot of processing power. But thanks to the fabrication of circuits at micro level we are able to fit this power in tiny integrated circuits. With the advent of nano technology I am pretty sure that every field of science and technology will be revolutionized. In modern manufacturing plants we can see such robots them working and checking the proper filling of bottles. If the bottle is not properly filled then that bottle is picked out automatically. Also the vision based robots are used to hover on the wings of Boeing aircraft during its manufacturing process in order to find cracks in it if there are any. The Robo Cup is also a world famous tournament in which robot teams from all over the world take part. Football is played solely by robots using vision. Hence there is very wide application of vision based robots in many fields of life. Therefore I got

inspired by such modern developments and wanted to work in this part of engineering science. So I began to think for a project for my graduation and finally I came up with the idea of developing a vision based intelligent robot. I wanted to work in this area of engineering for my final year project with great interest and enthusiasm. Therefore after getting accepted to carry out this project, I started working on it. And at last by the grace of Allah, I successfully completed each and every part of this project in due time regardless of being alone to work on it.

1.2. Project Objectives

The objective of this project is to develop a working model robot which would be capable of identifying red balls in its surroundings. This identification will be done by the use of vision instead of any sensor. So the robot intelligently looks in to its environment for the required ball. Although it depends that what we want to be processed and in which object or objects we are interested in, but I have selected the red balls to be one to be the target of collection. The operation of identifying and collecting the balls (which are also the functional requirements of project) is listed below:

- The robot must be equipped with vision such that camera
- The robot must be able to look into its surroundings in order to sight a required ball
- When a ball is sighted it must be able to recognize it
- After recognition it must move exactly towards the ball
- When it approaches near enough to the ball, robot must stop
- Now in the next step the ball collection mechanism must be activated
- Then ball must be collected in a space present on or within robot.
- After the successful completion of above said actions, the robot must start again from the process of looking for another ball and again should follow the subsequent steps.

1.3. Project Aims

The aim of the project is implement an automated robot that can work independently mostly based on the vision capability rather than using different sensors for different purposes. The processing the vision we can become capable of performing many different advanced tasks as well in a more intelligent

way, which otherwise could be difficult to implement using sensors with decrease in efficiency. This project can form basis to other projects which require certain objects to be handled for example for robotic competition in NUST.

1.4. Project Requirements

In order to build this project according to the project objectives stated in [chapter 1.2](#), a number of different tools were required. First of all the structure of the robot had to be built which should be light in weight and strong enough. Therefore after examining a number of different possibilities I chose plastic sheet. So for cutting and joining the structure I required a number of different working tools for example Drill machine, screw drivers, cutters etc. The 3-D design of the robot was prepared in an early stage in order to avoid modifications requiring hand work in implementation stage. Also it is very difficult and costly to modify structure when it is already built. The software I used for this purpose is “Auto Desk Inventor”.

Then the electronics part (embedded system) was designed using “Proteus Professional”. This helped to simulate the electronic embedded design and remove any errors or modify the design according to the requirements. The image processing part is catered by “Mathworks Matlab”. The designing of the printed circuit board (PCB) was performed using “Eagle CAD”. Then for the actual making of PCB, I needed ferric Chloride (FeCl_3), PCB board, mini drill machine, photo paper and hot iron. To solder the components I required a soldering gun along with some extra things.

The project required an extensive knowledge of “Microprocessor Based Designing” and “Image Processing” along with some other engineering skills and knowledge.

1.5. Project Challenges

Many different challenges were faced from the start of project till the completion. Some of them are listed below:

- Brain stormed within myself for different design possibilities
- Economic hurdles
- Sorting out the best required tools
- Gathering the required equipment and tools

- Removing errors during actual implementation such that transition from software design simulations to actual hardware
- Time constraints for project completion
- Thinking of new ideas, then their implementation and at last analyzing results

1.6. Thesis Organization

The report has been organized in such a way that in the next [chapter 2](#), the literature reviewed. This includes the background of the project literature, related work and related book/articles or thesis. [Chapter 3](#) is dedicated to the methodology followed in the whole project. This also explains the tools that are used, description of the steps involved, different approaches to solve problems and diagrams/flowcharts etc. Finally this report will move to the [chapter 4](#) and it is about project specifications. In this chapter all the design diagrams, interconnections and the integration process is explained and showed with help of diagrams and pictures. There are also simulations results explained and indicated with help of screen shots. Finally the report is closed with some conclusions, appendices and references.

2. Literature Overview

2.1. Background

The project is mainly divided into two major parts. One is related to embedded control system. Some of the robotics details are present in embedded control part and remaining are explained in [chapter 4.1](#). Therefore I didn't find it necessary to dedicate a separate topic to robotics. Second is the image processing part. Hence it requires an extensive knowledge of both the subjects in order to carry out this project successfully. Now I am going to explain the related literature of the project.

2.1.1. Embedded Control System

The embedded control system can be divided into following main parts:

- Regulated DC Voltage Supply
- Microcontroller based design
- Embedded C
- Serial Communication
- Comparator
- DC Motor Driver
- Servo Motor
- Infrared Sensor

All of the above mentioned parts are explained below:

Regulated DC Voltage

For the proper working of any electronic device, a proper voltage supply is needed which is according to the power requirements of the device. Otherwise the circuit and whole device will not run in a required manner. The improper voltage supply also results in the damage of electronic components. It had been measured through experimentation that my robot needs a peak 0.3A current hence I have used L7805CV for the regulation of DC voltage. This is a very decent voltage regulator IC with three pins. It is small in size, very reliable and regulates the DC supply at input greater than or equal to 7 volts very efficiently. The [Figure 1](#) on the next page shows the L7805CV and its schematic symbol:

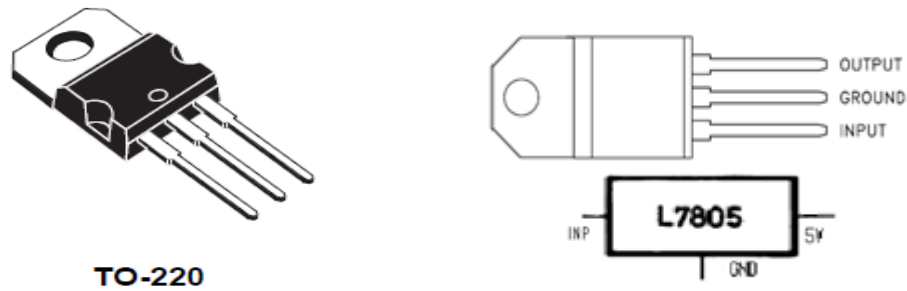


Figure 1: L7805 DC voltage regulator. [Ref 3]

Following are the salient features of the L7805CV:

- The L7805CV has three terminals as shown in Figure 1. Pin number one is the input terminal. We can apply an input voltage value from 7 volts to 20 volts normally.
- Pin number 2 is the ground pin. An adequate heat sink is required to be attached to grounded back side of the L7805CV voltage regulator to transfer heat into atmosphere and for better performance of the device and current requirement of around 1A (our design has been measured to use 0.3A at maximum during its operation).
- The pin number 3 is for output. Hence we get a clean and regulated voltage supply from this pin. As we are using L7805CV so will get a regulated 5V. This voltage can swing between 4.75 to 5.25 volts for the above stated voltage and current ranges.
- There are also some other capabilities of this voltage regulator, such that:
 - Thermal overload protection
 - Short circuit protection
 - Output transition safe operating area (SOA) protection

It comes in different packages but I have used the TO-220 package

Microcontroller Based Designing

Microcontroller is the heart of any modern embedded computing system. It is just like the brain of the whole electronics. Difference between microprocessor and a microcontroller is that microprocessor is used for general processing and requires external components like memory (ROM, RAM etc), communication peripherals, inputs/outputs (I/O), timers etc. But a microcontroller is used for specialized tasks and

normally has almost everything on chip like CPU (microprocessor), ROM, RAM, and also on chip peripherals like serial communication (UART), timers, input/outputs (I/O) ports, Analogue to digital converters (ADC) etc as shown in [Figure 2](#) below.

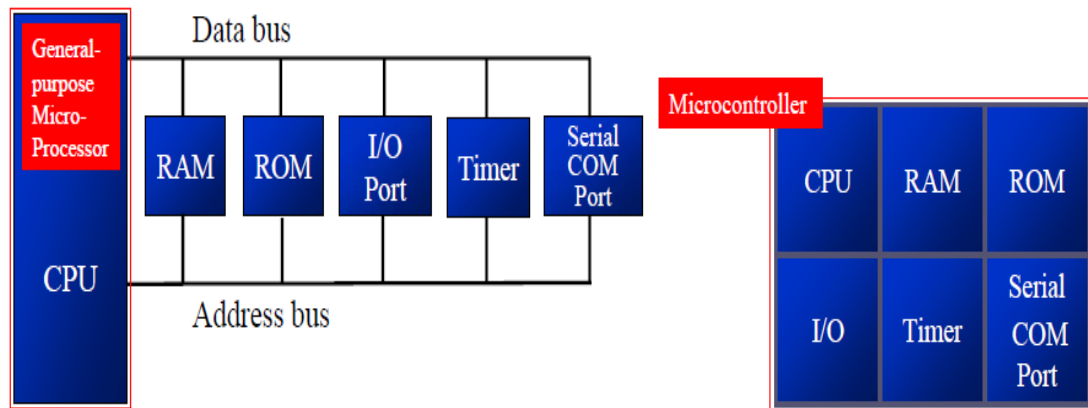


Figure 2: System view of a microprocessor (LEFT) and microcontroller (RIGHT). [Ref 4]

The microcontroller I have selected for this project is Atmel 89c51 because this microcontroller fulfills all the requirements needed in this project and is also cost effective choice. It is also commonly available and its development software and programmer hardware is easily available. The salient features of AT89c51 are:

- It is an 8-bit microcontroller with low power consumption and high performance
- It has 4 Kilo Bytes of in system Flash programmable and erasable read only memory (PEROM)
 - 1000 write/erase cycles
- Compatible with industry standard MCS-51 instruction set and pin out
- 32 programmable input/output (I/O) pins divided into four 8-bit ports
- Static operation from 0Hz to 24 MHz
- 128 x 8-bit internal RAM
- Six interrupt sources
- Programmable Serial Channel
- Cost effective and highly flexible solution for embedded control applications
- It is available in different packages but the one I am using is in PDIP (Plastic Dual In-Line Package) as shown in [Figure 3](#) on the next page:

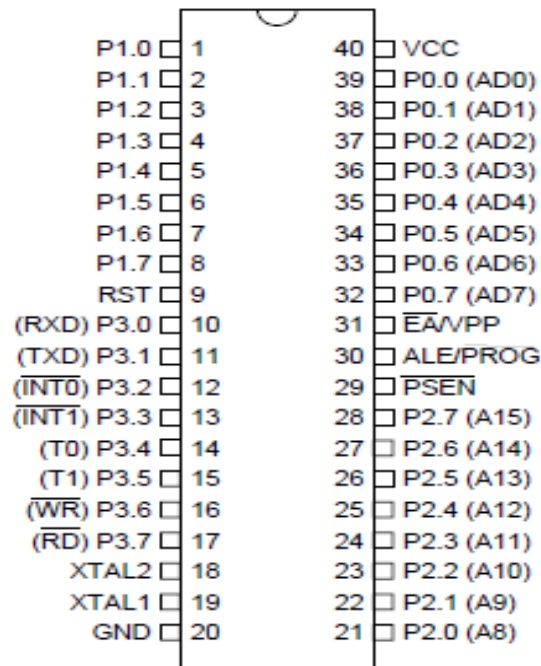


Figure 3: AT89c51 microcontroller in PDIP package. [Ref 3]

The pin description of the AT89c51 is given below:

- V_{CC} is the pin 40 at which 5 volt supply is provided to the microcontroller
- Pin 20 is dedicated to **ground**
- **Port 0** ranges from pin number 32 to 39 with pin 39 as least significant. It makes it an 8-bit port. It may be used for input and out purpose but also to address the lower byte of the data or program memory. For making it an I/O port, external pull up resistors of 10K ohm are required.
- **Port 1** is also an 8-bit port and is used for I/O purposes. It already has internal pull up resistors so no need for external resistors. It contains the pin numbers 1 till 7.
- **Port 2** is again 8-bit as well with internal pull up resistors. It is accessed through the pin numbers 21 to 28. So can be used either for I/O or to address high byte of the external program and data memory.
- **Port 3** ranges from pin 10 to 17 and has many functions as well along with usage of 8-bit I/O. It has internal pull up resistors. The other functions that it can perform are
 - **P3.0** can be used for RXD (serial input port)

- **P3.1** TXD (serial output port)
- **P3.2** INT0 (external interrupt 0)
- **P3.3** INT1 (external interrupt 1)
- **P3.4** T0 (timer 0)
- **P3.5** T1 (timer 1)
- **P3.6** WR (external data memory write strobe)
- **P3.7** RD (external memory read strobe)
- **RST** stands for reset and is on pin number 9. If the signal on this pin remains high for two machine cycles, then the microcontrollers gets reset.
- **EA/VPP** must be connected to Vcc in order to use the internal program memory of the microcontroller. Otherwise if attached to the ground, it enables the microcontroller to fetch code from the external program memory. Pin number 31 is dedicated to it.
- **XTAL1** and **XTAL2** are used for crystal clocking circuit. They are present on pin numbers 18 and 19 respectively.

Embedded C

It is subset of C language and is used for programming the embedded systems. It also consists of some extensions according to the needs of the embedded system. The syntax of embedded C is same as of normal C we use for making programs on the computer.

Serial Communication

Communication is very critical in modern technologies and has seen tremendous growth and advancement in recent years. One of the techniques used in digital computer communications is “Serial Communication”. In this method bits are sent one by one in a queue. RS232 is the port which is primarily used for this purpose. The desktop computers still come with this port while modern laptops don't have this port anymore. The RS-232 connector has nine pins as shown in [Figure 4](#) on the next page:

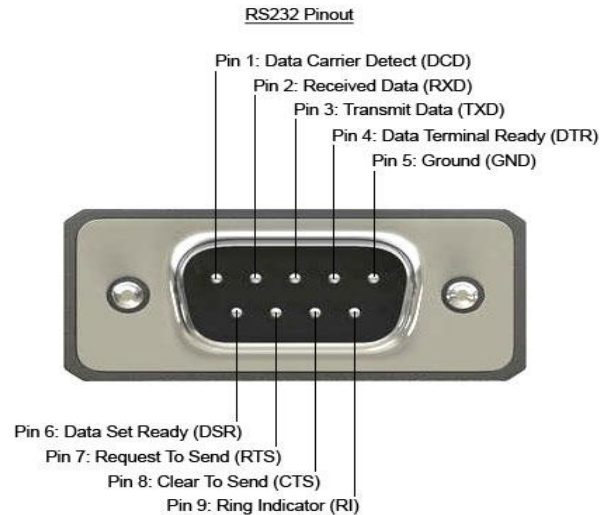


Figure 4: RS-232 port. [Ref 4]

The three important pins are RXD, TXD and GND. RXD receives data serially, TXD which transmits data serially and ground pin as shown in Figure 5 below:

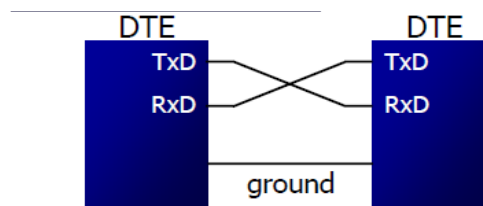


Figure 5: Simple Serial Communication. [Ref 4]

Serial communication can be of two types. One which sends data in form of blocks of data along with synchronization information, this is called “Synchronous Communication”. This is clock oriented transmission format. Other is the one which sends one byte at a time and is termed as “Asynchronous Communication”. It is character oriented and includes start and stop bits. Softwares can be written for both types of methods. But this is very hard work to do and is very lengthy as well; therefore special integrated circuits are made for this purpose and are capable of serial communication in any of above way. The two commonly used are:

- **UART** stands for universal asynchronous receiver and transmitter
- **USART** stands for universal synchronous and asynchronous receiver and transmitter

Asynchronous method is mostly implemented for communicating characters. As ASCII characters are mostly widely used, hence each 8-bit character is placed between start and stop bit and this process is called “Framing”. The data that can be sent or received can either be simplex (meaning that one is transmitter and other is receiver only), half-duplex (means that one can be receiver and other can be transmitter at an instant but both can send and receive data) and the at last may also be full-duplex meaning that both can send and receive data at the same time. Following Figure 6 illustrated these modes:

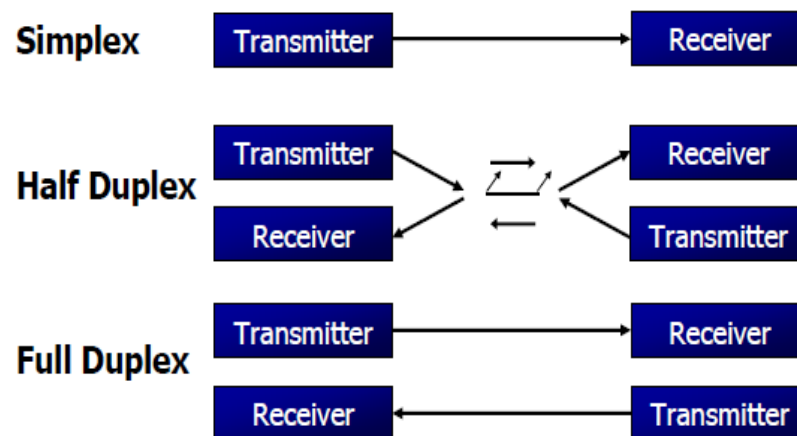


Figure 6: Three form of implementation of serial communication according to the requirement. [Ref 4]

Also there are some important terms used in communication. These are:

- **Data rate** means how many bits can be transmitted and received in one second. The unit used for this measure is “bits per second” (bps)
- **Baud rate** means that how many signal levels can be changed in one second.

In RS-232 serial communication, 1 is represented by voltage levels ranging from -3 to -25 volts whereas a 0 is represented from +3 to +25 volts. The voltage values from -3 to +3 volts are undefined. As RS-232 was designed before transistor transistor logic (TTL), hence these voltage levels are not compatible with TTL. To make it compatible we need a line driver IC. The most widely used IC is MAX232. This makes it compatible with microcontrollers and as we are dealing with 8051 microcontroller in our project, therefore we would use its reference. Figure 7 on the next page illustrates the use of MAX232 with microcontroller:

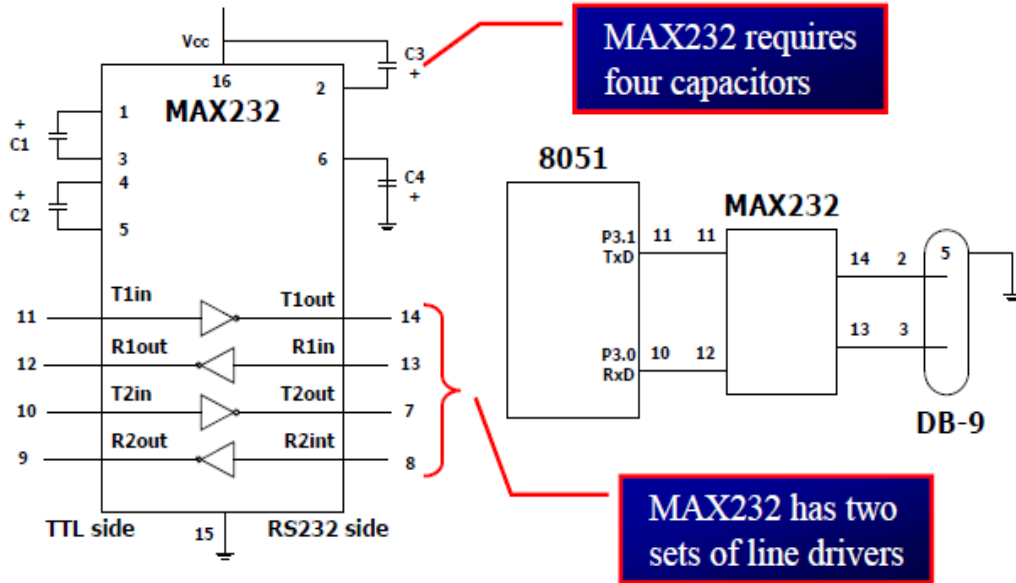


Figure 7: Use of MAX232 with 8051 microcontroller. [Ref 4]

The supported baud rates by 8051 are 1200, 2400, 4800 and 9600.

Comparator:

A comparator is a circuit which compares the two voltage levels at its input and outputs the one that is higher or lower than other, depending upon the need and implementation. It has been used to drive TTL compatible logic in this project for R sensor. The symbol of a simple comparator is shown in Figure 8 below:

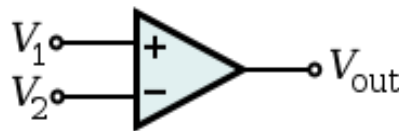


Figure 8: Symbol of a voltage comparator. [Ref 4]

One of the most widely used comparator IC's is LM393N. It is an 8 pin IC and comes in different packages. I am using the PDIP package for my project and looks like as shown in Figure 9 below:

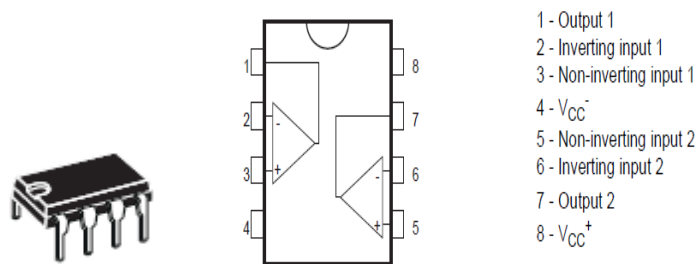


Figure 9: LM393N voltage comparator IC. [Ref 3]

Salient features of LM393N include:

- Can be used for single voltage supply comparison or for dual voltage supply comparisons
- The input voltage ranges from +2 to +36 volts in case of single supply comparison. For dual supply comparison the voltage range is ± 1 to ± 18 volts
- Requires very low input current independent of voltage that is supplied. It is normally 0.4mA
- Differential input voltage range is equal to the supplied voltage.
- TTL, DTL, ECL, MOS and CMOS compatible outputs

DC Motor Driver

An important thing for any mechatronics or robotics design is the drive and control of different actuators. These may include DC motors, stepper motors, linear motors, or any other actuators. These drivers help in a number of ways, such that simplicity of design and control, reliable working, proper safety against back EMF from the actuators that can destroy the circuit and replacing H-Bridge circuits etc. As I am using two geared DC motors of 12 volts, therefore the best choice for me was to choose L923D motor driver IC. It is commonly available and comes in different packages. I am using the PDIP package. Following [Figure 10](#) shows the L923D:

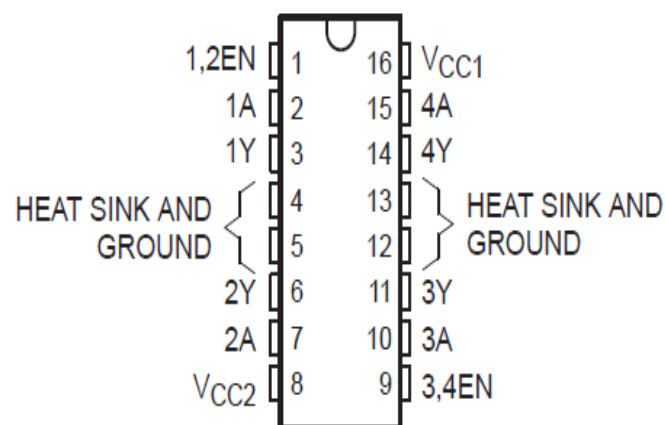


Figure 10: L293D DC motor driver IC. [Ref 3]

Salient features of L293D include:

- Voltage supply range from 4.5 to 36 volts, hence a number of DC motors can be easily driven

- Separate input logic supply signals for controlling motors
- Thermal shutdown
- Two motors can be driven bi-directionally simultaneously
- 1.2A peak output current per channel
- Output clamp diodes for inductive transient suppression. Hence no external diodes needed.

Servo Motor

Servo motor is very commonly used in different robotic needs and for variable position stability and control. These motors can only rotate to different angles with high torque rather than full circular revolutions. They are available with different range of possible angles like -45 to $+45$ degrees or -90 to $+90$ degree servo motors are commonly used. They are actually DC motors but included with essential gears for producing high torque and control circuitry for different possible positions. Different companies manufacture these motors but I have used the one that is made by Chaparral Company and is used in Satellite antennas. Following [Figure 11](#) gives us the look of the servo motor used:

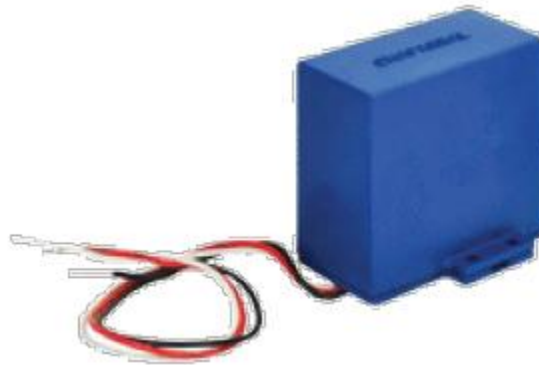


Figure 11: Chaparral Servo Motor. [Ref 3]

The working principle of these motors is present behind the use of pulse width modulation (PWM) technique. As discussed earlier that these motors can only rotate to a certain range of degrees, therefore their angle (position) depends upon the width of PWM wave. Following [Figure 12](#) on the next page illustrates the mechanism of a servo motor depending upon the duty cycle of the PWM waveform:

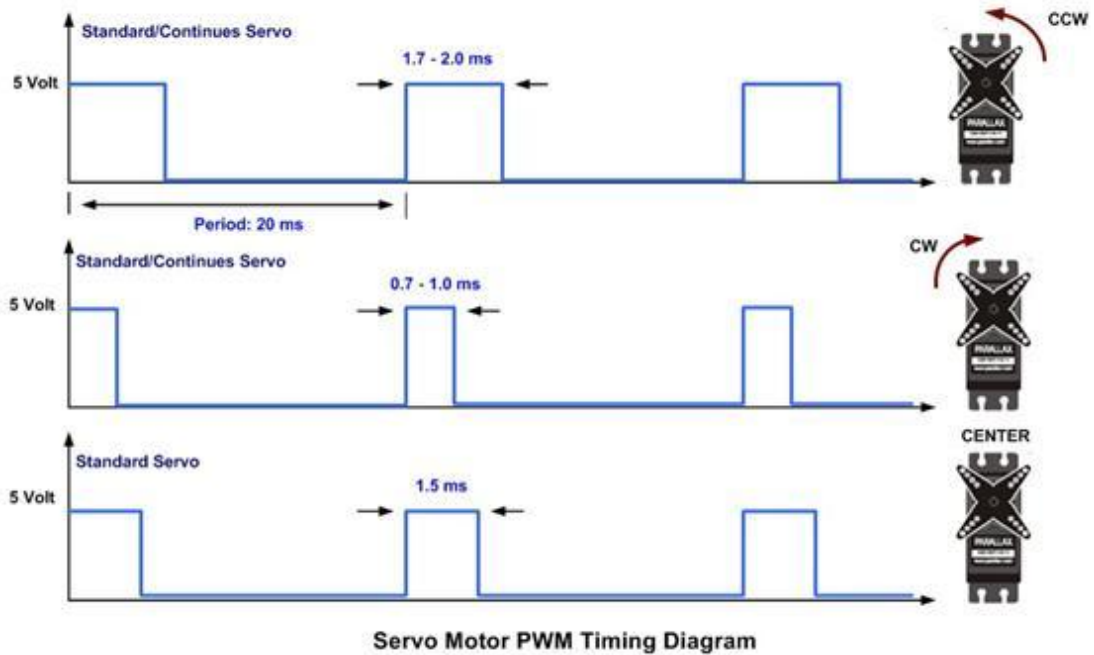


Figure 12: Position of servo motor with respect to duty cycle of the PWM signal. [Ref 4]

It can clearly be interpreted the working of a servo motor from Figure 12 above. In case of -90 to $+90$ degree servo motor, the 50Hz signal is used and the pulse width of 1.5 milli-seconds keeps the position of the servo motor at centre position. The width of 0.7ms moves the servo motor to extreme counter clockwise position of suppose -90 degrees and the pulse width of 2.0ms moves it to the extreme left position of $+90$ degrees. It is important to note here that the pulse widths of intermediate values from 0.7ms till 2.0ms will position the servo motor in between the range of -90 to $+90$ respectively.

Important features of the Chaparral motor we used are:

- Low current, high torque motor
- High reliability
- Working voltage of 5 volts required
- A total of 180 degree movement such that -90 to $+90$.
- Pulse width of 0.8ms to 2.2ms
- Time period 17ms to 21ms
- DC current 500mA

Infrared Sensor

LED's are used in many applications for multiple purposes. One of them is infrared LED. This LED is used to send signals in infrared band (around 760nm) and hence is invisible to human eye. They are made from gallium arsenide or aluminum gallium arsenide. It has many applications. The most common one is the use in remote controls. The LED that transmits infrared light is called infrared transmitter and there is also an infrared receiver. Both are shown in the [Figure 13](#) below in order to have a visual look:

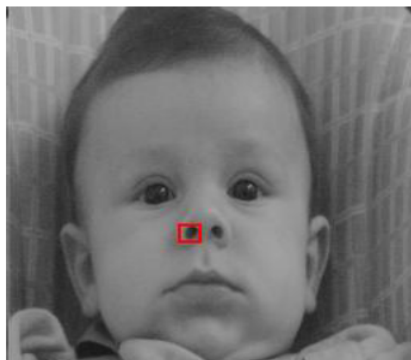


Figure 13: Infrared transmitter (BELOW) and infrared receiver (ABOVE). [Ref 4]

2.1.2. Digital Image Processing

A picture is worth 1000 words. So this is very true. Information conveyed through a video is more informative than just simple telling. Image processing requires a lot of computations but thanks to the advancement in digital technology and computers, the image processing now can be carried out on any reasonable computer or even at personal computers with appropriate application or software.

A digital image is a two dimensional plane which is divided into pixels. If it is a **binary image** then each pixel may either have a value 0 (black) or 1 (white). In case of **gray scale image** of 8-bits, each pixel value may range from 0 to 255. This is called intensity value. Following [Figure 14](#) illustrates this point:



PIXEL VALUES IN HIGHLIGHTED REGION

99	71	61	51	49	40	35	53	86	99
93	74	53	56	48	46	48	72	85	102
101	69	57	53	54	52	64	82	88	101
107	82	64	63	59	60	81	90	93	100
114	93	76	69	72	85	94	99	95	99
117	108	94	92	97	101	100	108	105	99
116	114	109	106	105	108	108	102	107	110
115	113	109	114	111	111	113	108	111	115
110	113	111	109	106	108	110	115	120	122
103	107	106	108	109	114	120	124	124	132

Figure 14: Pixel intensity values within the indicated red box in image. [Ref 2]

Conversion from Gray Image to Binary Image

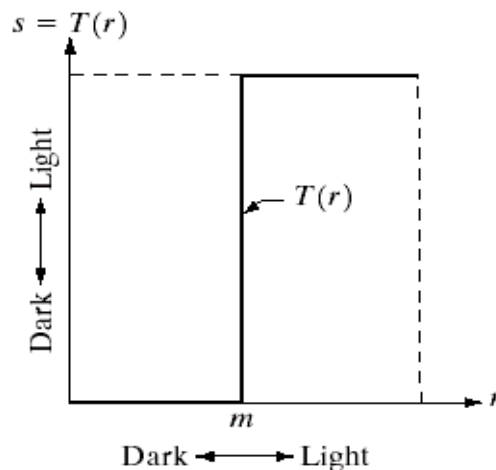
We can convert any image into its bi-level (binary) image using different techniques. One of them is thresholding. In this technique we set a constant intensity value and check it against each pixel value in the original image. If it is less than or equal to the threshold value, then that pixel is set to 0 (black) in the new image, otherwise it is set to 1 (white). [Figure 17](#) provides the visual concept:



[Figure 17](#): We can see a gray scale image on left and its binary image created using threshold value at right.

[Ref 2]

The graphical representation of this technique is shown in [Figure 18](#) below:



[Figure 18](#): the graph of thresholding. [Ref 2]

Morphological Operations

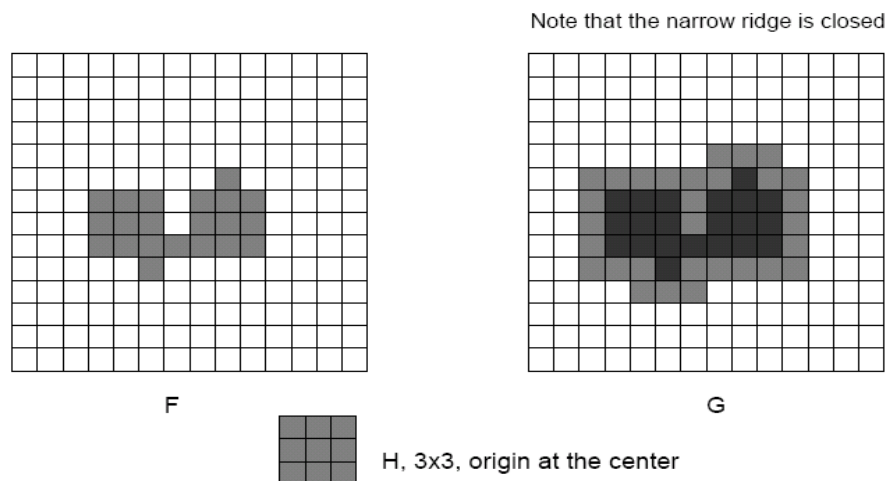
It is an image processing technique which deals with the shape of the features of an image. It is basically similar to the special filtering as discussed above. A structuring element is moved across every pixel in the original image to give the pixel a new value in a new processed image. The value of this new pixel depends on the operation performed. Morphological operations are originally developed for bi-level

images for shape and structural manipulations. Basic functions are dilation and erosion. Combination of dilation and erosion in different orders result in more high level operations, including closing and opening.

The shape, size, and orientation of the structuring element depend upon the application. A symmetrical one will enlarge or shrink the original set in all directions. A vertical one will only expand or shrink the original set in the vertical direction.

Dilation

In dilation the origin of template is stepped over every element in the whole of the image. The place where the origin of the template corresponds to a 1 in the image, the template is ‘unioned’ with that part of the image. The resulting template-sized matrix is then unioned with all the other template-sized results using their original image positions and giving a resulting image that is larger than the original. Following [Figure 19](#) shows the effect of dilation:



[Figure 19](#): Effect of dilation using 3x3 structuring element on original image left and dilated image on right. [Ref 2]

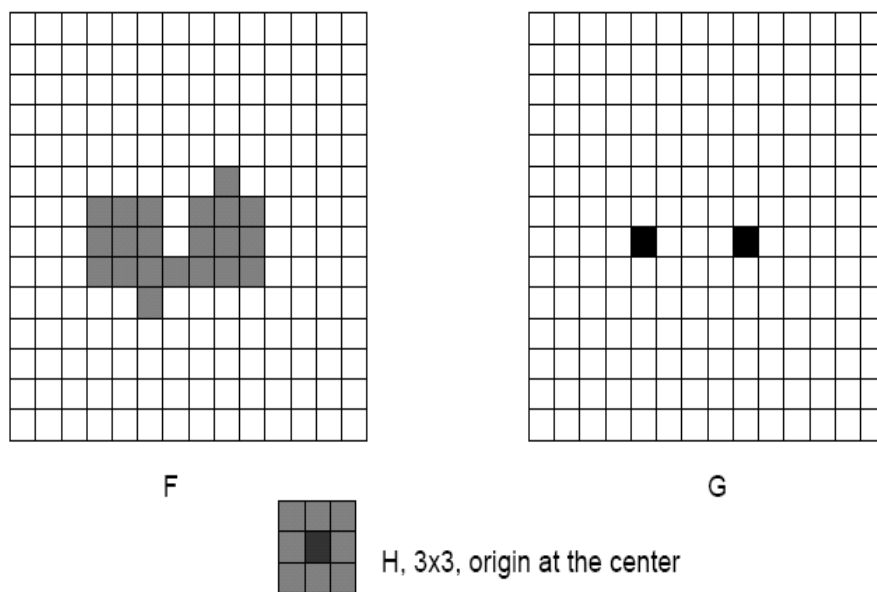
Dilation is used to repair breaks in an image. It is also used to repair intrusions. They also cover the broken gaps in text or other objects. But keep in mind that dilation enlarges the object from its normal size as shown in [Figure 20](#):



[Figure 20](#): Filling of gaps by dilation. [Ref 2]

Erosion

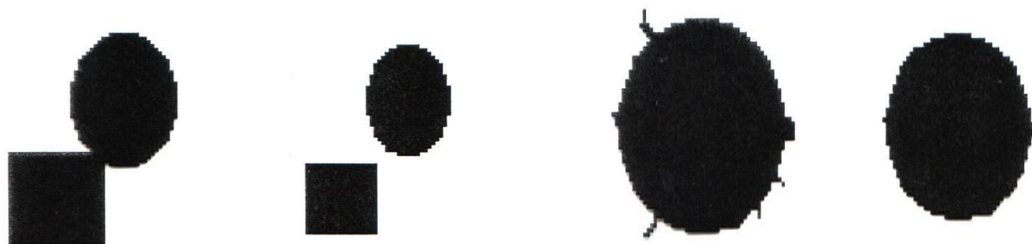
In the technique of erosion, the template is stepped over every element in the whole of image but this time only in positions where the whole of the template lies on top of the image, that is, it is not allowed to go off the edge. For every stepped position in the image the template is compared with the corresponding window to the image. If the template is exactly the same as the image window, then the element corresponding to the template origin in the resulting matrix is set to a 1. Following [Figure 21](#) illustrate the concept of erosion:



[Figure 21](#): Effect of erosion using 3x3 structuring element on original image left and eroded image on right.

[Ref 2]

This technique is used to split the joined objects. It is also used to strip away the extrusions. But remember that erosion makes the objects shrink from their normal size. Following [Figure 22](#) clarifies the concept:



[Figure 22](#): Splitting the objects and removing extrusions by using erosion. [Ref 2]

More interesting morphological operations can be performed by performing combinations of erosions and dilations. The most widely used of these compound operations are “Opening” and “Closing”.

- An erosion followed by a dilation using the **same structuring element** for both operations is called “Opening”.
- A Dilation followed by an erosion using the *same structuring element* for both operations is called “Erosion”.

Opening smoothes the contour of an image and eliminates thin ridges and branches whereas closing smooth the contour of an image, fuses narrow breaks and long thin gulfs and fills the small gaps and holes. [Figure 23](#) shown below also illustrates one the usage of opening and closing. In this image only the big squares are desired so they are extracted using the opening technique:



Figure 23: Opening operation. Original Image (LEFT). Eroded Image (Centre). Dilated Image (Right). [Ref 2]

3. Project Methodology

3.1. Methodology and Tool Support

After the proposal of the project, it was the biggest task to determine that how the objectives of the project can be achieved and what methodology to be used. It required significant knowledge and brain storming. Finally I made an initial concept of the project. The idea/concept of the working model is given below:

- The robot must be equipped with vision such as camera
- The robot must be able to look into its surroundings in order to sight a required ball
- When a ball is sighted it must be able to recognize it
- After recognition it must move exactly towards the ball
- When it approaches near enough to the ball, robot must stop
- Now in the next step the ball collection mechanism must be activated
- Then ball must be collected in a space present on or within robot.
- After the successful completion of above said actions, the robot must start again from the process of looking for another ball and again should follow the subsequent steps.

3.1.1. Approach to Problem

Therefore the work on project started with this very first step. Then I divided the project according to its requirements and finally made a tree diagram of the requirement of the whole project in terms of its working areas and knowledge required. [Figure 24](#) given on the next page shows the subprojects of it:

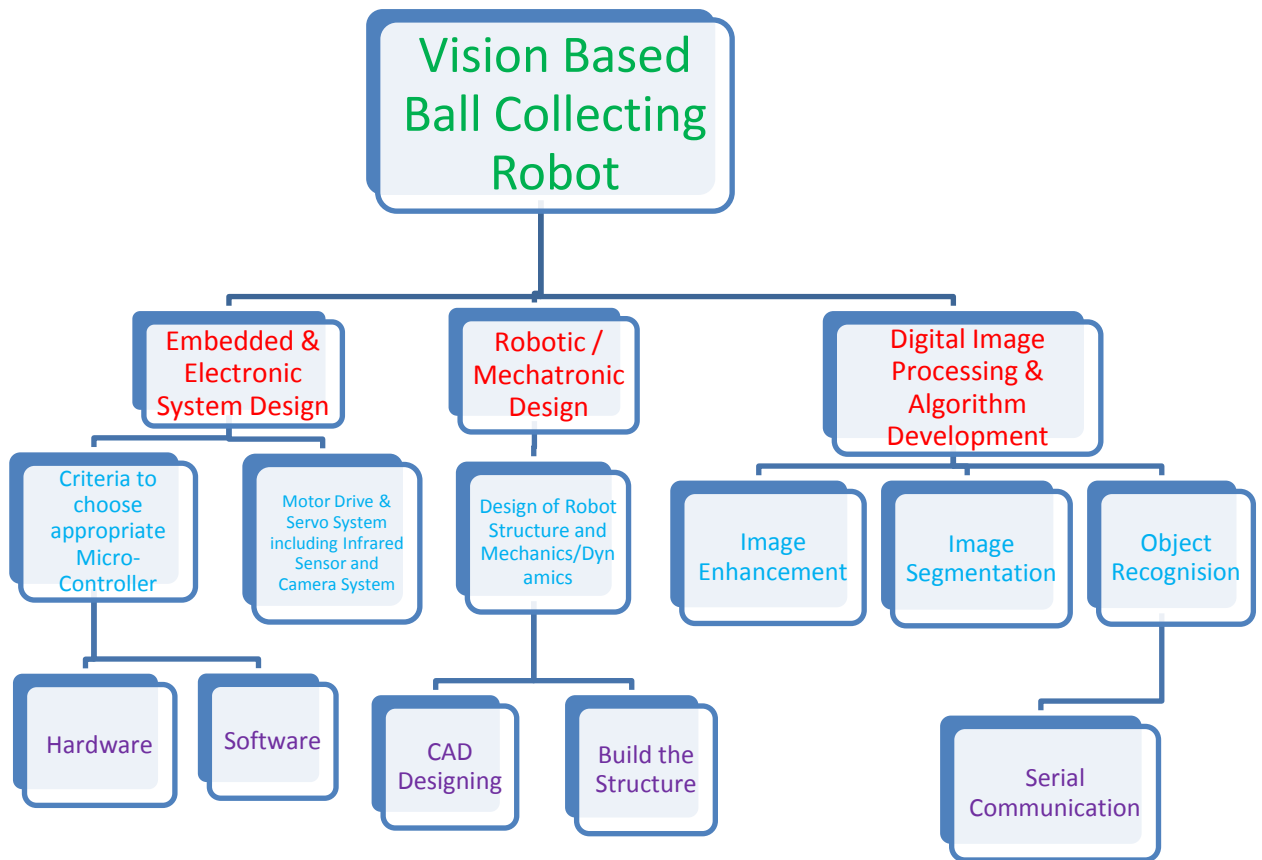


Figure 24: Break down of the project into its subprojects.

The above project division was not a simple task to be done. First of all the whole concerned fields of engineering were consulted and a detailed study was carried out. Therefore, it resulted in the above hierarchy of the project, as a top to bottom model. This model is very helpful in time management of the whole project and provided a true picture throughout the progress of the project that where did I lie at the given moment. Also it helped me to choose a starting point for my project. I started from designing the embedded system of the robot.

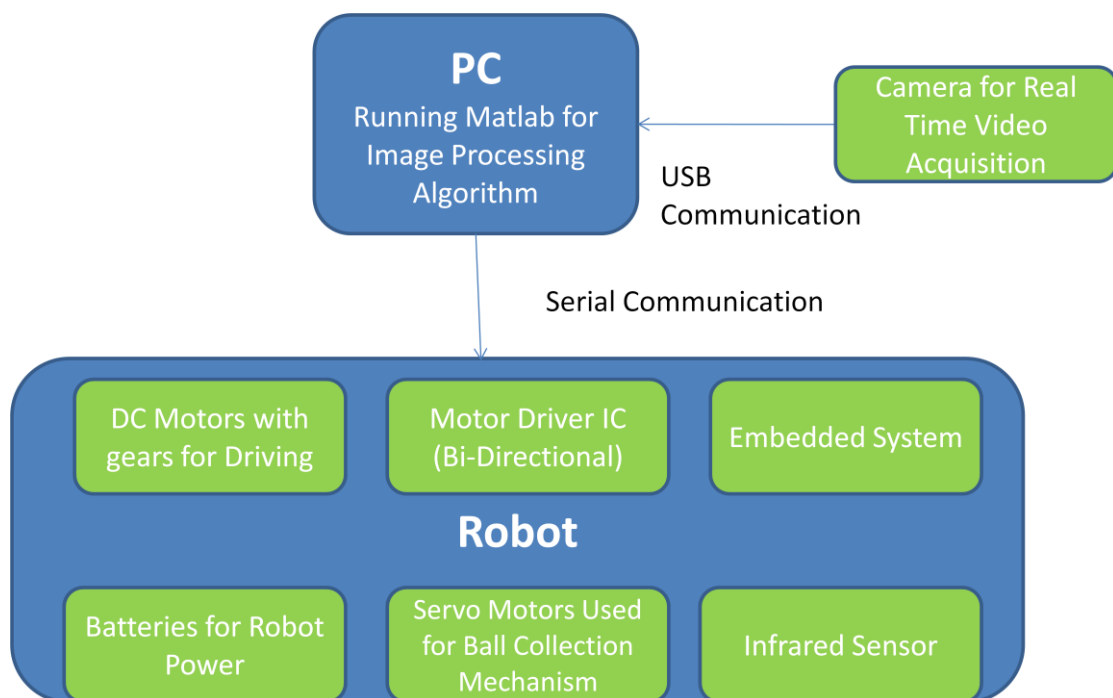
3.1.2. Collection of Tools

Furthermore I collected the required tools which would be needed to build this project. It includes all the necessary software. The softwares I am using are given on next page:

- Keil μ Vision 4 (Used for Embedded C programming)
- Matlab 2010b (Image Processing)
- Autodesk Inventor 2009 (Designing of Robot)
- Eagle CAD, PCB Design 2011 (PCB Design Tool)
- Proteus Electronic Design 7.5 sp2 (Electronics Design and Simulation Tool)

Other than that I already had a number of different tools in order to build most part of the project at home, which includes soldering equipment, digital multi-meter, and other tools which would be helpful in building the robot structure. Some other I bought by myself when I needed. This includes the mini drill for PCB making etc. Also I bought the electronic components from the “College Road” situated in Rawalpindi.

System Level Design was also constructed as shown below in [Flow Diagram 1](#) for better understanding of working system.



Flow Diagram 1: System Level Design.

4. Project Specifications

4.1. Robot Design Development

The designing of the robot was carried out using the AutoDesk Inventor. Different ideas were brain stormed and after analyzing different tradeoffs, I came up with a design to collect balls into the body of a three wheeled robot. I made a concept and image in my brain and visualized it in reality using the above said CAD tool. The robot will be driven by two front wheels and would be capable of turning left or right on the same axis of rotation. This helped out to eliminate and understand different design possibilities, technicalities to be kept in mind which could arise model was started to build directly, design constraints etc. following [Figure 25](#) shows the screen shots taken when the design was finalized:

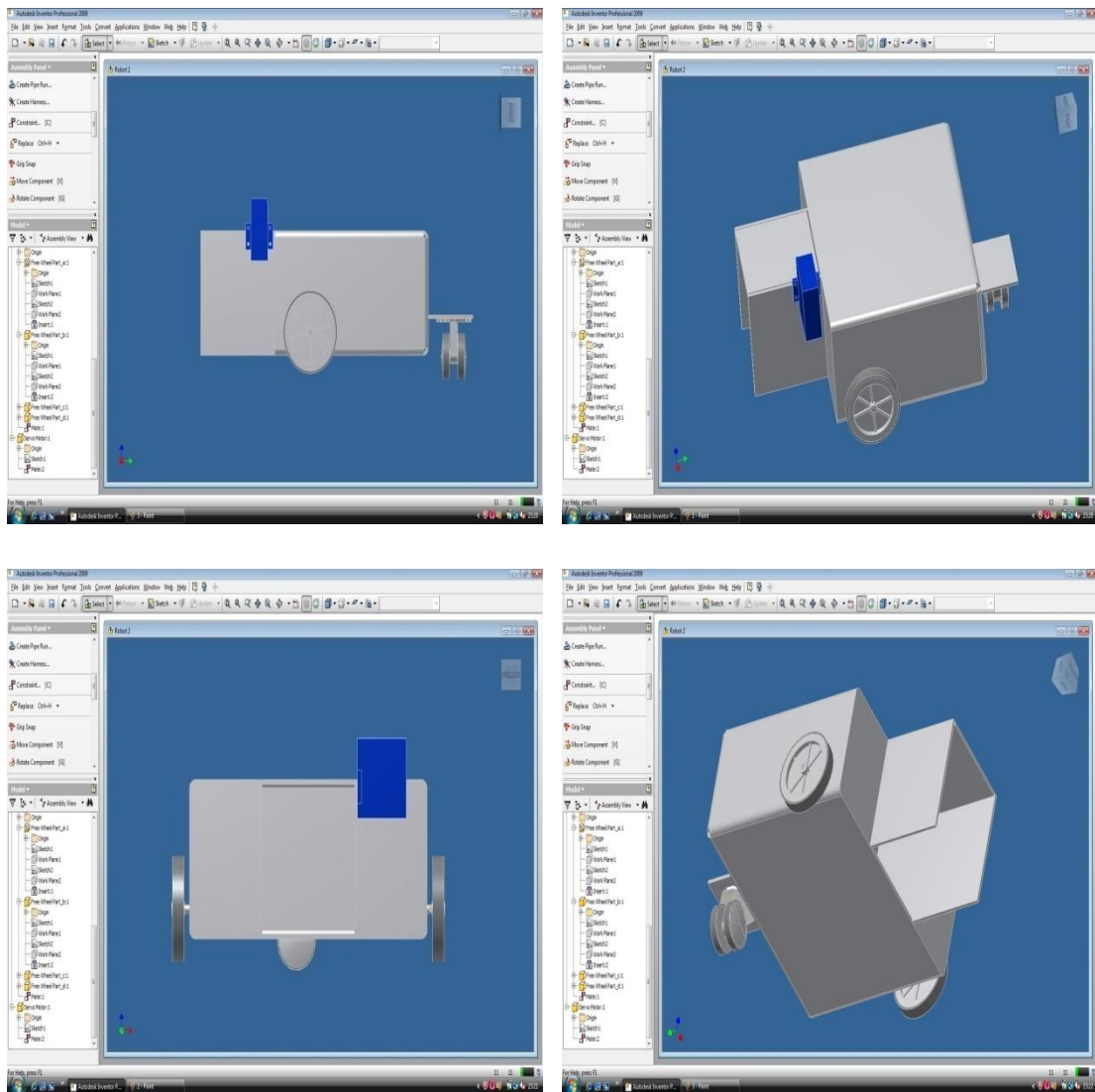


Figure 25: Screen shots of robot design in AutoDesk Inventor.

Let us have a look at the actual model as well. The actual one is made exactly the same as it was designed in AutoDesk Inventor as shown in [Figure 26](#) below:



Figure 26: Actual Robot.

4.2. Embedded Control System Development

The embedded Control system was designed using the Proteus software. Different design and component possibilities were designed and tested for different design point of views. It helped to save a lot of effort and costs if had to be carried out practically instead of software design and simulations. Finally the design was made as shown in the [Figure 27](#) on the next page. It is worth mentioning here that the design and verification of the infrared sensor was carried out practically in the university lab because its simulation was not possible in Proteus software. Also sensor design required reliability and other factors hence I thought it is better to design it directly and practically rather than simulating it.

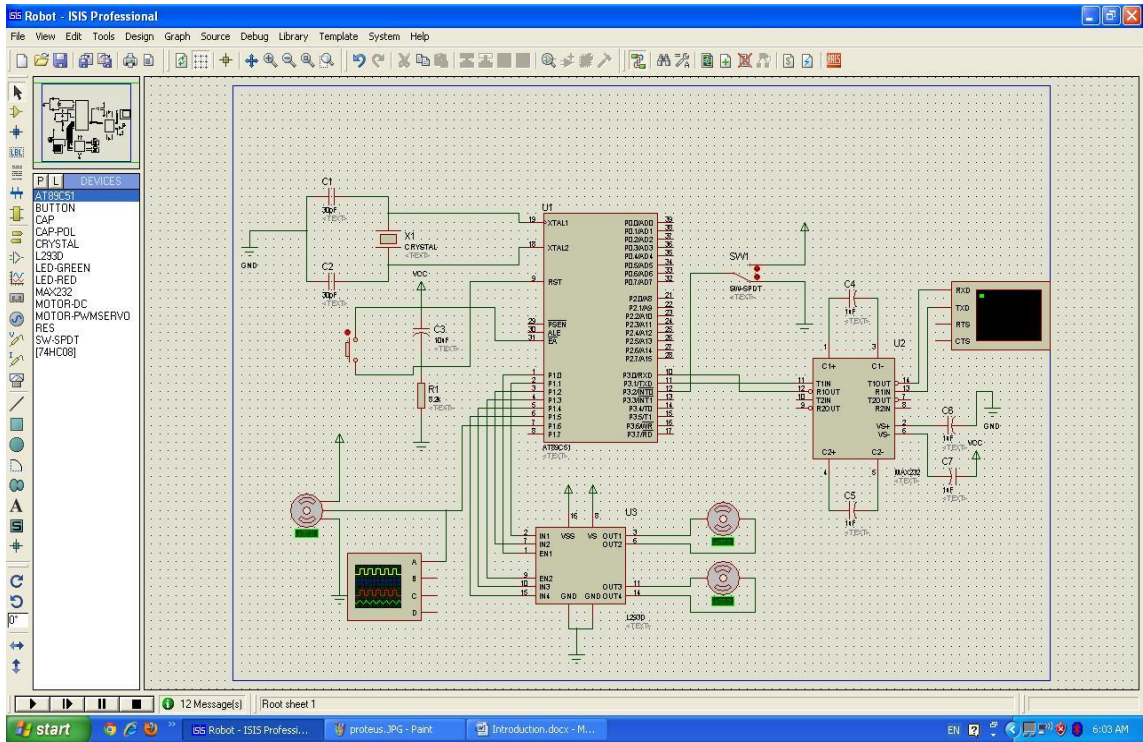


Figure 27: Screen shot of the embedded design in Proteus.

Also the simulations of the electronic design and software written in Embedded C were carried out in this software. Following Figure 28 shows the verification of electronic design and embedded software in form of waveforms and rotating drive DC motors:

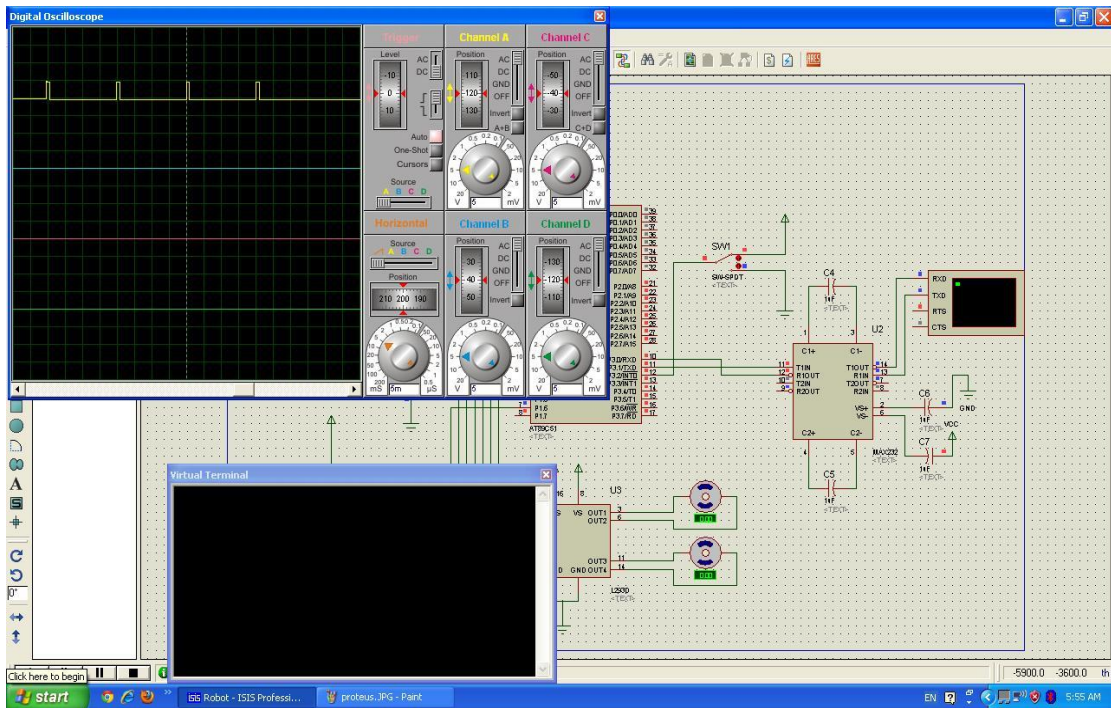


Figure 28: Simulation of the electronic design.

The final embedded design is shown in the [Figure 29](#) below:

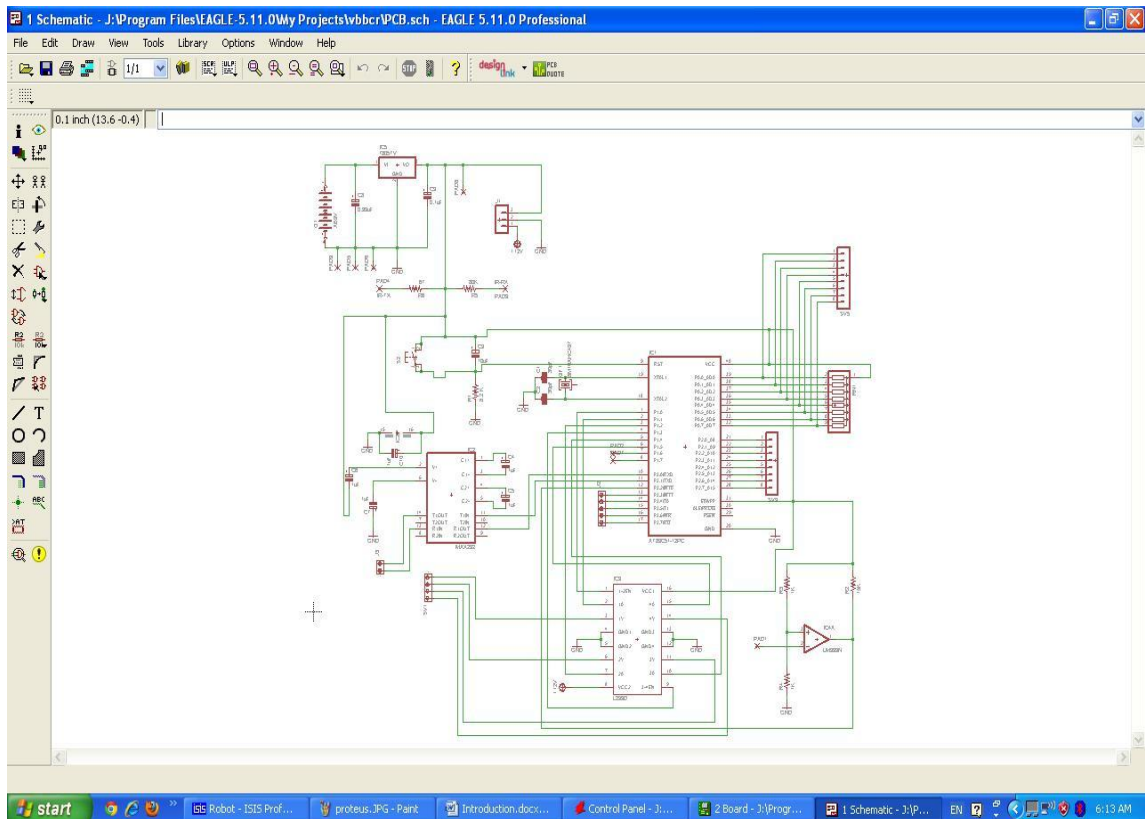


Figure 29: Final schematic design.

The final design consists of five major parts. These parts are explained below:

AT89c51 Microcontroller

It is the brain of the whole embedded system. It coordinated the operations between PC and the robot. It controls all the actuators of the robot which include the geared DC motors which drive the robot and the servo motor for collection mechanism. Also it keeps a check on the sensors and as soon as an interrupt occurs from the infrared sensors it activated the ball collection mechanism, because an interrupt from IR sensor means a ball in bound.

MAX232CPE

It is the line driver. As the RS-232 serial communication is not compatible with TTL voltage levels that are used by microcontrollers, hence this IC converts the voltage levels for both RS-232 and microcontroller for a two way communication. It

required four external capacitors. These are of 1uF value and can be operated till 50V. The baud rate used is 4800 bps.

L293D

It is used to drive the geared DC motors. It can enable the motors to rotate in both clockwise and anticlockwise directions. It also provides significant protection against back EMF. Also the microcontroller can logically control the motors by sending logic signals to L293D only.

LM393N

This is the comparator IC. The purpose of using this IC was to give strength to the voltage level taken from the IR sensors. It keeps an output of 5V continuously so that TTL logic of microcontroller can interpret it. As soon as the voltage level from the IR sensors exceeds 2.5V at its negative input terminal, it drops the voltage to almost 0V so that microcontroller can easily get the interrupt signal. 2.5V is set as threshold value for comparator using two resistors of 1K ohm in voltage divider configuration. [Figure 30](#) shows the circuit diagram of comparator:

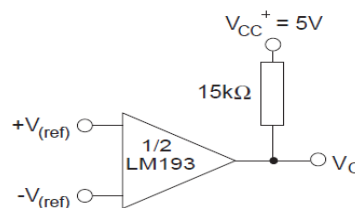


Figure 30: the LM393N comparator circuit. [Ref 3]

L7805CV

It regulated the voltage to 5V and can provide peak current of 1.5A. It uses only two capacitors of 0.1uF and 0.33uF as shown in [Figure 31](#) below.

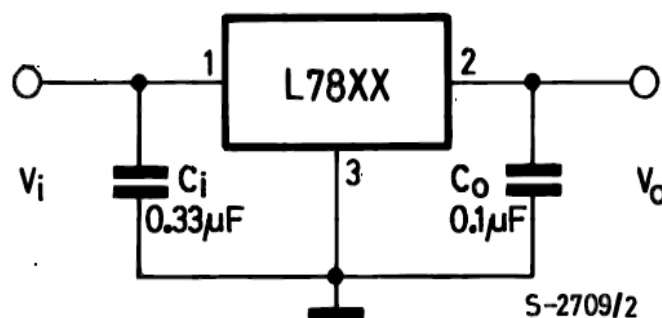


Figure 31: L7805CV regulation circuit. [Ref 3]

4.3. PCB Development

Eagle CAD is used to develop the printed circuit board. The final schematic as shown in Figure 29 is used to create the PCB. The final design of the PCB developed and used is shown in Figure 32 below:

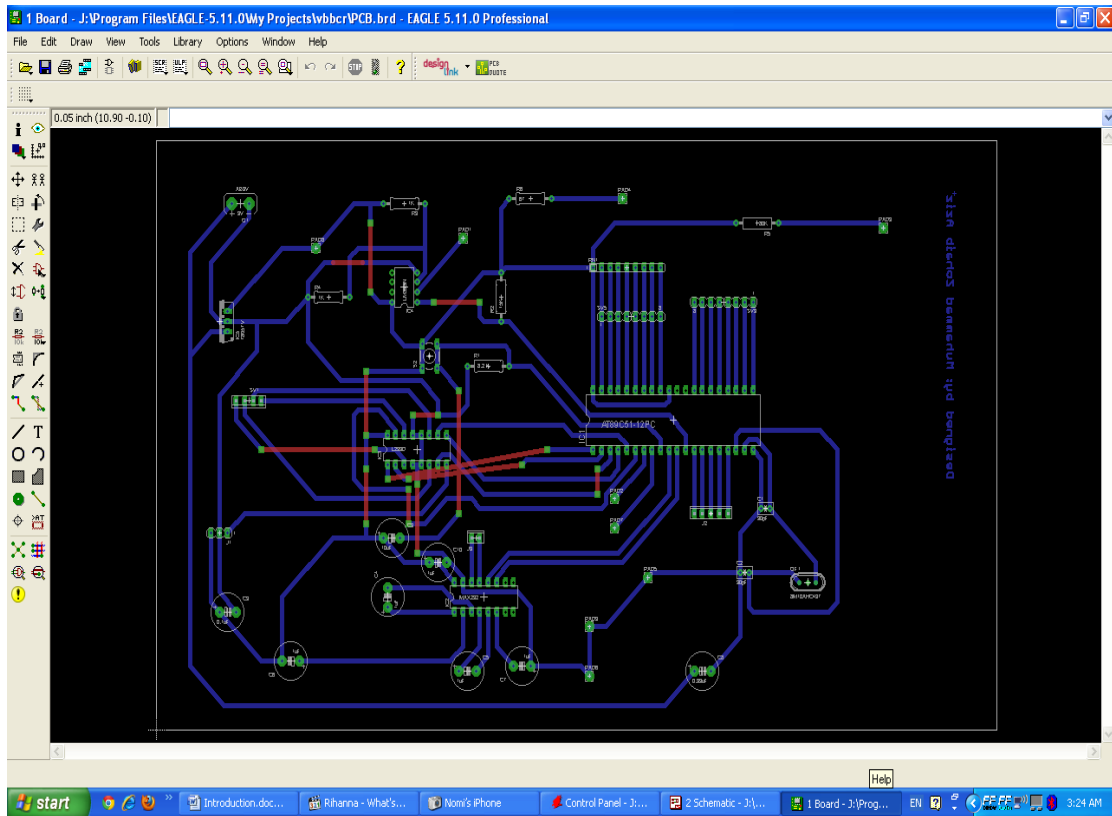


Figure 32: Final PCB design.

4.4. Embedded programming

Then embedded programming is carried out in Keil uVision 4. Given below is the explanation of important sections of embedded C code developed for the robot. First of all there is code information heading then important ports and variables are declared for use in the rest of the program. “sbit” is used to declare a pin of a port and “unsigned char” is one byte data type. Relevant code is shown below:

```
// Header: Embedded Code for the project of "Vision Based Ball
Collecting Robot"
// File Name: C Code
// Author: Muhammad Zohaib Aziz Khan
// Date: 11/11/2011

#include<reg51.h>
```

```

sbit EN12 = P1^0;
sbit A1 = P1^1;
sbit A2 = P1^2;
sbit EN34 = P1^3;
sbit A3 = P1^4;
sbit A4 = P1^5;
sbit PWM = P1^6;

unsigned char getByte;
unsigned char i;
unsigned char j;

```

Then there are delay functions declared and written. The delay is used to generate PWM for the servo motor for collection mechanism. The reason for using timer is that it produces accurate delay timing and this delay is used to produce PWM wave for the control of position of servo motor. Relevant code is shown below:

```

void us_1500_delay(void)           // 1500uS Delay
{
    TMOD = 0x01;
    TH0 = 0xFC;
    TLO = 0xA0;
    TR0 = 1;
    while(TF0 == 0);
    TR0 = 0;
    TF0 = 0;
}

void us_2200_delay(void)           // 2200uS Delay
{
    TMOD = 0x01;
    TH0 = 0xF8;
    TLO = 0x1F;
    TR0 = 1;
    while(TF0 == 0);
    TR0 = 0;
    TF0 = 0;
}

```

Two delay functions are used for positioning the servo at center and left position. Center is referred to as right in this program. Relevant code is shown below:

```

void right(void)
{
    PWM = 1;
    us_2200_delay();

    PWM = 0;
    for(i=0 ; i<7 ; i++)

```



```
        {
            us_2200_delay();
        }
    }

void left(void)
{
    PWM = 1;
    us_1500_delay();

    PWM = 0;
    for(i=0 ; i<20 ; i++)
    {
        us_1500_delay();
    }
}
```

I have written an external interrupt service routine. This means that as soon as the ball is detected by the infrared sensor, the external interrupt is called and the robots stops and performs the ball collection steps. Relevant code is shown below:

```
void external_interrupt0() interrupt 0
{
    EN12 = 0;
    EN34 = 0;

    A1 = 0;
    A2 = 0;
    A3 = 0;
    A4 = 0;
    for(j=0 ; j<100 ; j++)
    {
        right();
    }
    for(j=0 ; j<100 ; j++)
    {
        left();
    }
    TMOD = 0x20;           // 8-Bit, Auto-Reload
    TH1 = 0xFA;           // 4800 Baud Rate using Timer1
    SCON = 0x50;
    TR1 = 1;
}
```

In the main body of the program, I have used the serial communication programming and reason for this is that the image processing software (Matlab) issues the control signals depending upon the results of its image processing through serial port. Hence the microcontroller coordinates with the image processing software and

issues the control logic to the DC motor driver IC which is L923D. the while loop keeps the robot active all the time and the whole program keeps on executing. The robot can be moved forward, backward, left or right depending upon the need using this embedded C program. Relevant code is shown below:

```
void main(void)
{
    IE = 0x81;           // Enable Interrupt 0

    EN12 = 0;
    EN34 = 0;
    A1 = 0;
    A2 = 0;
    A3 = 0;
    A4 = 0;

    for(j=0 ; j<50 ; j++)
    {
        left();
    }

    TMOD = 0x20;        // 8-Bit, Auto-Reload
    TH1 = 0xFA;        // 4800 Baud Rate using Timer1
    SCON = 0x50;
    TR1 = 1;           // Set Timer1

    while(1)
    {
        while(RI == 0);
        getByte = SBUF; // Read Buffer

        switch(getByte)
        {
            case 'L':
            {
                EN12 = 1;
                EN34 = 1;

                A1 = 1;
                A2 = 0;
                A3 = 1;
                A4 = 0;

                break;
            }
            case 'R':
            {
                EN12 = 1;
                EN34 = 1;

                A1 = 0;
                A2 = 1;
                A3 = 0;
            }
        }
    }
}
```

```
        A4 = 1;
        break;
    }
    case 'F':
    {
        EN12 = 1;
        EN34 = 1;

        A1 = 1;
        A2 = 0;
        A3 = 0;
        A4 = 1;

        break;
    }
    case 'B':
    {
        EN12 = 1;
        EN34 = 1;

        A1 = 0;
        A2 = 1;
        A3 = 1;
        A4 = 0;

        break;
    }
    default:
    {
        EN12 = 0;
        EN34 = 0;

        A1 = 0;
        A2 = 0;
        A3 = 0;
        A4 = 0;

        break;
    }
}
RI = 0;
}
```

4.5. Image Processing Program

Image processing is carried out in Matlab. Matlab is a technical language in order to facilitate engineers and scientists to develop complex algorithms and other computations. The Matlab computations are based upon the matrices. Hence enables it for 2D domains as well. Now I am going to explain the important sections

of the Matlab algorithm I developed for red ball detection and generating control signals respectively. First of all is as usual the information part and clearing all the variables and data already in the Matlab environment. Then there is code to acquire video from the video device into Matlab, and images are extracted out from the video for further image processing. The redundant video data is flushed out to keep memory available. Also the serial port of the computer is declared and activated using the object created for serial port. The baud rate and data length is also set for serial communication. Relevant code is shown below:

```
// Header: Matlab Code for the project of "Vision Based Ball
Collecting Robot"
// File Name: ImgeProcessingVisionBasedRobot
// Author: Muhammad Zohaib Aziz Khan
// Date: 11/12/2011
clear all;
clc;

% vid = videoinput('winvideo',1,'RGB24_640x480');
vid = videoinput('winvideo',1,'YUY2_640x480');
preview(vid);
vid.ReturnedColorspace = 'rgb';
pause(5);

% ser=serial('COM1'); % Defining the specified COM Port to be used
% fopen(ser); % starting serial Communication,opening serial port

% while(1)
    im = getsnapshot(vid);
    flushdata(vid);
% im = imread('2.jpg');
figure('Name','Original Image');
imshow(im);
```

Then the red, blue and green planes of images are viewed to get a comparative view along with the original image taken. Further the color range of the required object is defined through trivial experimentation and its respective binary image is formed using threshold technique for further processing. One can also do some pre processing on the image before doing all above said procedures for example sharpening of image etc but is not mandatory if a good quality imaging device is used. Relevant code is shown below:

```
% sharpFilter = fspecial('laplacian');
% im_sharp = imfilter(im, sharpFilter, 'replicate');
```

```

% figure('Name','Sharpened Image');
% im_sharp = im + im_sharp;
% imshow(im_sharp);

% im_sharp = imfilter(im, sharpFilter, 'replicate');
% figure('Name','Sharpened Image');
% imshow(im_sharp);

im_red = im(:,:,1);
figure('Name','Red Frame');
imshow(im_red);

im_green = im(:,:,2);
figure('Name','Green Frame');
imshow(im_green);

im_blue = im(:,:,3);
figure('Name','Blue Frame');
imshow(im_blue);

im_bin = (im_red >= 150) & (im_blue <= 80) & (im_green <= 80);
figure('Name','Binary Image');
imshow(im_bin);

```

Now in the next step an important processing is carried out and that is called “Morphological Operations”. The theory behind these operations is already explained in this report. Important to mention here is that I am using a disk shaped structuring element for my morphological operations and reason behind using such structuring element is that we are interested in red balls and balls are always round shaped. Hence is suitable for me according to my requirements. First of all the image is eroded using disk shaped structuring element in order to remove unnecessary red details in the image and elimination of noise. After that the close operation is carried out on the eroded image. This helps to fill the gaps in the obtained image of red ball after erosion. In the next step the opening operation is performed in order to remove any branches and ridges if they exist. So after that finally in order to get only the smooth image of ball, the closing operation is performed again. Relevant code is shown below:

```

se = strel('disk',2);

im_erode = imerode(im_bin,se);
figure('Name','Image after Erosion morphological operation');
imshow(im_erode);

% im_open = imopen(im_bin,se);
% figure('Name','Image after Open morphological operation');

```

```

% imshow(im_open);

se2 = strel('disk',2);

im_close = imclose(im_erode,se2);
figure('Name','Image after Close morphological operation');
imshow(im_close);

im_open = imopen(im_close,se2);
figure('Name','Image after Open morphological operation');
imshow(im_open);

se3 = strel('disk',20);

im_close = imclose(im_erode,se3);
figure('Name','Image after Close morphological operation');
imshow(im_close);

% im_final = im_bin & im_open;
% figure('Name','Anding Open and Close morphological operations');
% imshow(im_final);

im_label = bwlabel(im_close);

```

Up till here the required red ball has been verified and extracted if it is in sight. If it is not in sight then what to do? And if it has been extracted then what is the position of the ball in the image so that robot can make a decision which direction (left or right) to turn. So this needs some sort of intelligent processing and has been done in the code below. The centroid of the ball is computed if it does exist and respective control signal is sent to robot through serial communication. If the centroid doesn't exist then it means that there was no red ball in view. Hence the robot will take a slight right turn in order to look for one if there. The pause functions provide the delays that we require on different occasions according to needs of proper functioning of robot. Relevant code is shown below:

```

im_props = regionprops(im_close, 'Centroid');

if(size(im_props,1) ~= 0)

    im_area = regionprops(im_close, 'Area');    %Finds the areas of
the images accordign to their labels
    im_area_no_of_structs = length(im_area);
    max_area = 0;
    im_area_index = 1;
    for n = 1:im_area_no_of_structs
        if((im_area(n,1).Area) >= max_area)
            max_area = im_area(n,1).Area;
            im_area_index = n;
        end
    end
end

```

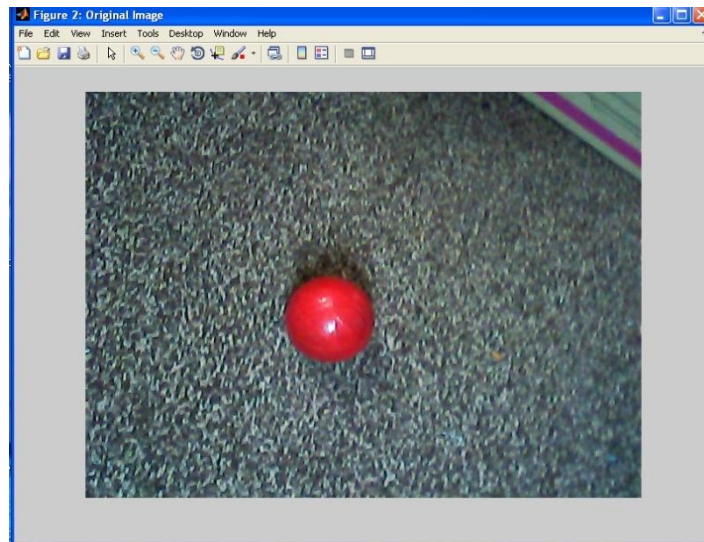



Figure 33: Original Image.

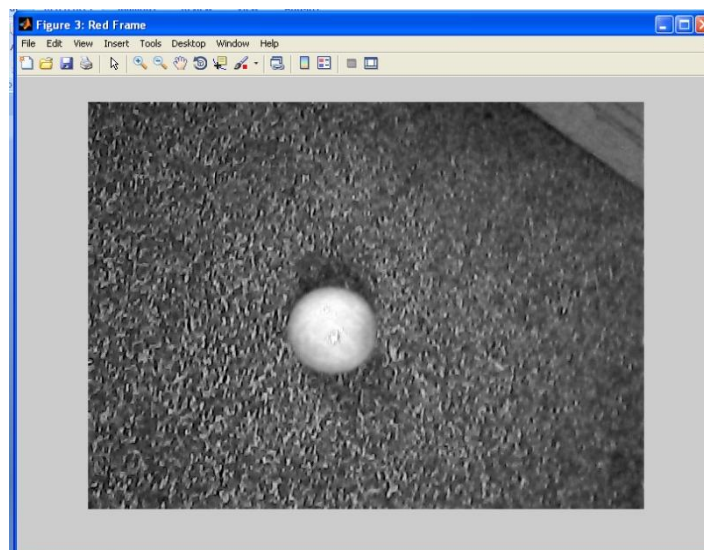


Figure 34: Red Frame of the original image.

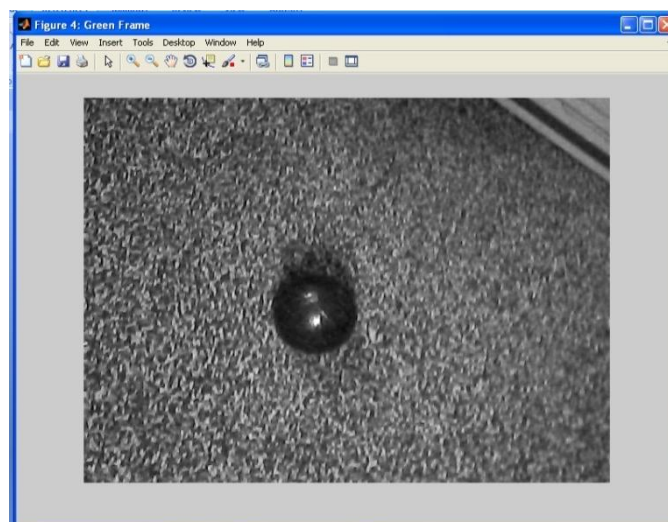


Figure 35: Green plane of the original image.

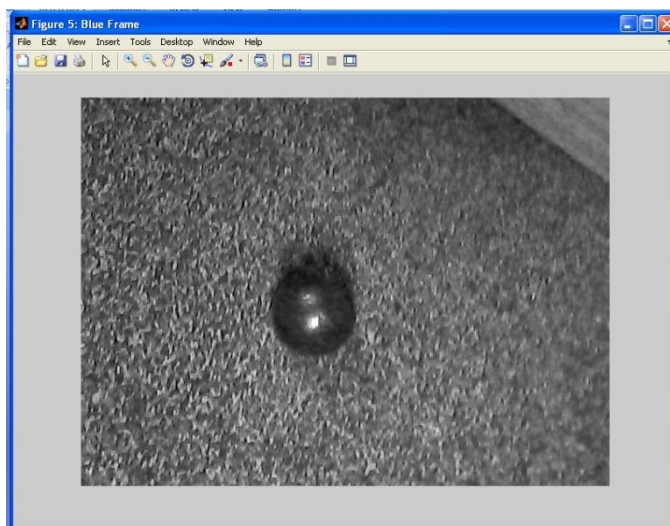


Figure 36: Blue frame of the original Image.

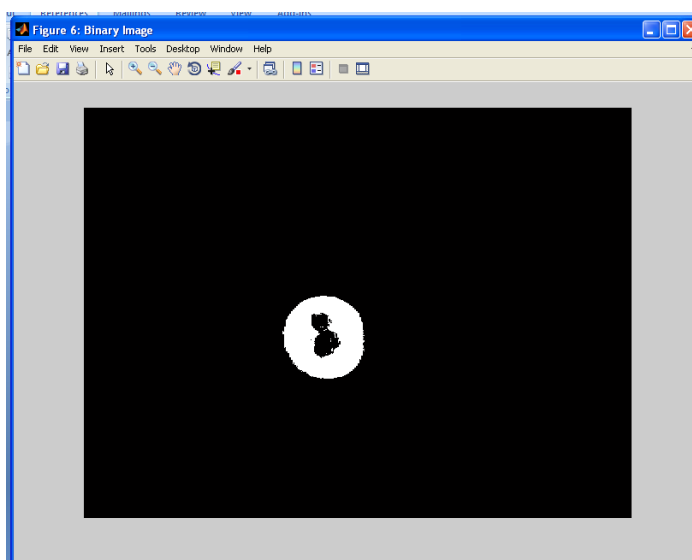


Figure 37: Binary Image of the original image.

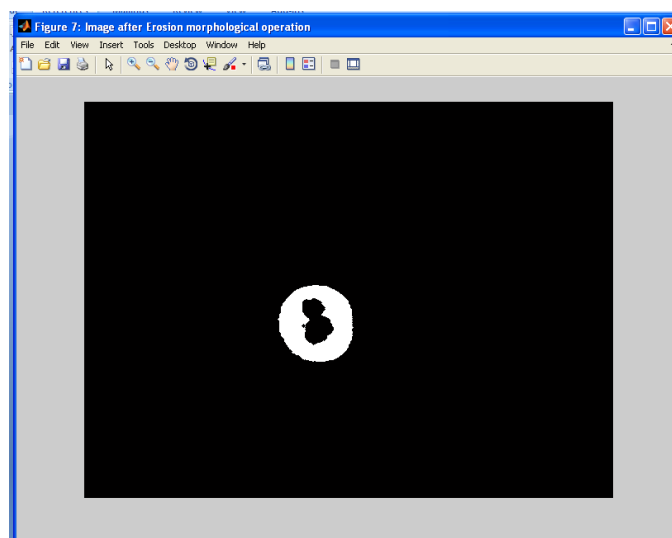


Figure 38: Image after Erosion Morphological Operation.

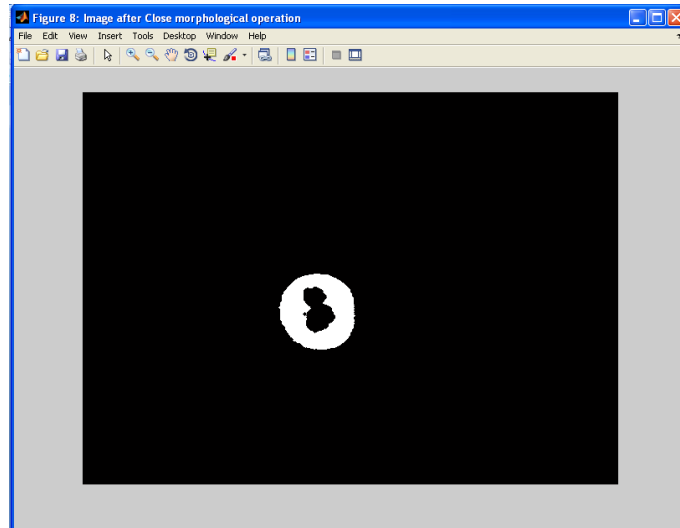


Figure 39: Image after Closing Morphological Operation.

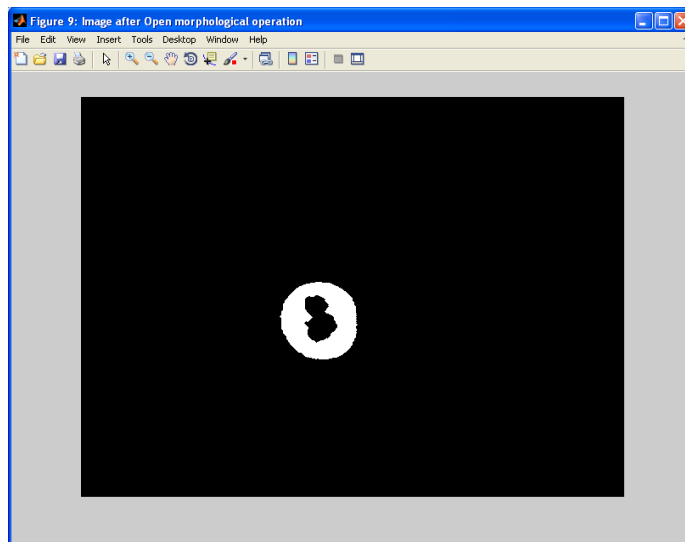


Figure 40: Image after Opening Morphological Operation.

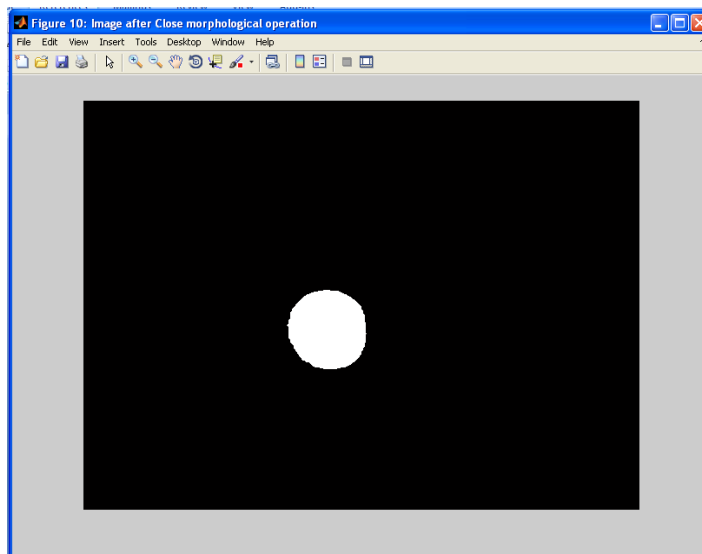


Figure 41: Image after Close Morphological Operation.

At the end, as part of finalizing the project, GUI has been developed gain by using MATLAB. This GUI window contains control buttons for the robot e.g. Start, Stop etc. Also the GUI shows the images taken and their resulting images after required processing. The Control signal issued to the robot is also displayed on the GUI window. Figure 42 and Figure 43 shows the GUI developed for the robot.

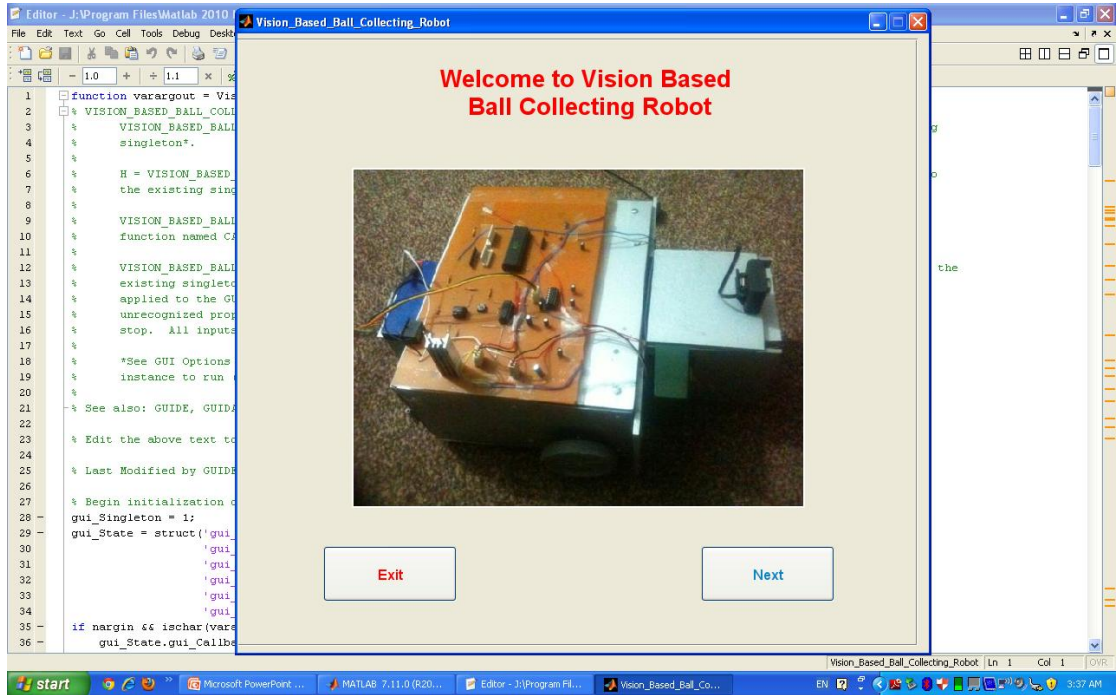


Figure 42: Opening window of the GUI.

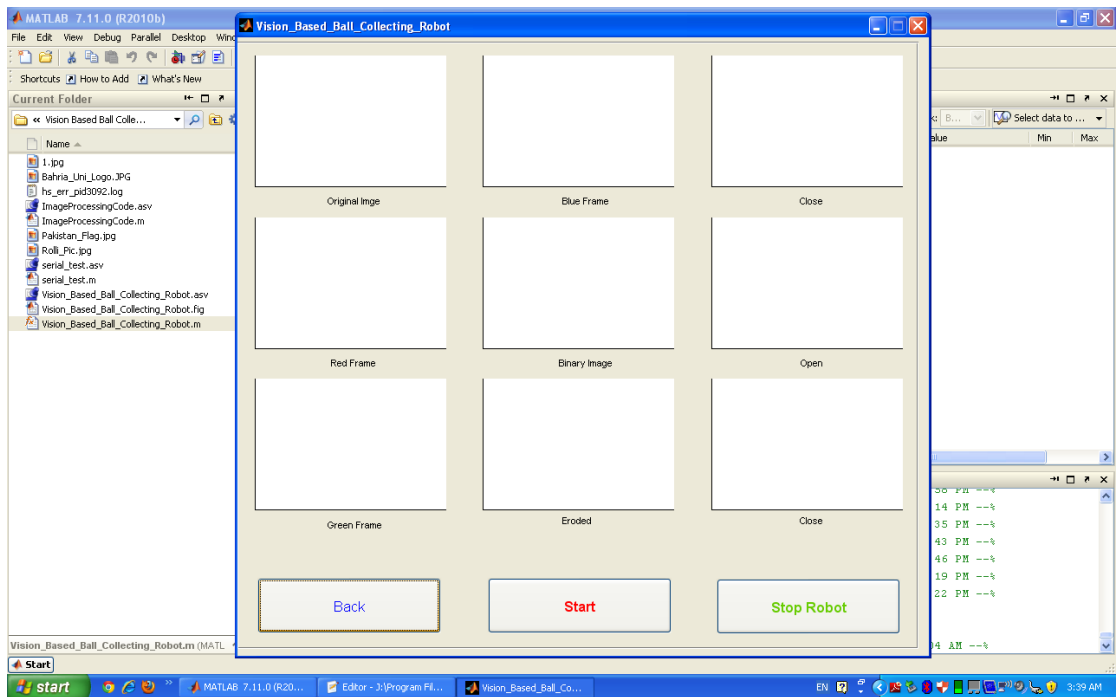


Figure 43: Control window for the robot, each box in the window shows real time processed images.

5. Future Work

There is great room for further improvements as this project provides a solid foundation. The robot can be upgraded to wireless communication rather than serial communication and by using wireless camera. Also the algorithm developed can be implemented in Digital Signal Processor by converting the Matlab code written into the C code. Also FPGA is another good option. This project can be very helpful to other intelligent robot related projects and competitions and can provide them new and efficient solutions e.g. for the robotics competition held every year at NUST.

6. References

[Ref 1] M. A. Mazidi, 8051 Microcontroller and Embedded Systems, 2nd Edition

[Ref 2] Rafael C. Gonzalez, Digital Image Processing, 3rd Edition

[Ref 3] Device data sheet

[Ref 4] www.google.com

