

PROJECT REPORT ON
Robot Navigation Using MATLAB'S Bounding Box
Algorithm

Submitted by:

Akshay Chidrawar (11)

Vaibhav Khedkar (34)

Ajay Patkar (56)

Vishal Thakur(71)

Under the guidance of

Prof. Mr. Shailesh Nandgaonkar

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF
BACHELOR OF ENGINEERING ELECTRONICS**



VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF THECHNOLOGY

DEPARTMENT OF ELECTRONICS ENGINEERING
UNIVERSITY OF MUMBAI
2014-15

VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled **“Robot Navigation Using MATLAB’s Bounding Box Algorithm”** is a bonafide work of **Akshay Chidrawar (11), Vaibhav Khedkar(34), Ajay Patkar (56), Vishal Thakur (71)** submitted to the University of Mumbai in partial fulfillment of the requirements of the B.E. Degree in Electronics under the guidance of **Prof. Mr. Shailesh Nandgaonkar** during the year 2014-15 as prescribed by Mumbai University.

(Name and sign)
Supervisor/Guide

(Name and sign)
Co-Supervisor/Guide

(Name and sign)
Head of Department

(Name and sign)
Principal

Project Report Approval for B. E.

This thesis / dissertation/project report entitled ***Robot Navigation Using MATLAB's Bounding Box Algorithm*** by *Akshay Chidrawar, Vaibhav Khedkar, Ajay Patkar, Vishal Thakur* is approved for the degree of **Bachelor of Engineering.**

Examiners

1. -----

2. -----

Date:

Place:

Declaration

I declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Vishal Thakur 71

(Name of student and Roll No.)

(Signature)

Date:

Abstract

Object detection and tracking are important and challenging tasks in many computer vision applications such as surveillance, vehicle navigation, and autonomous robot navigation. Object detection based techniques is the most popular choice to detect stationary foreground objects because they work reasonably well when the camera is stationary and the change in ambient lighting is gradual, and they also represent the most popular choice to separate foreground objects from the current frame. This paper illustrates the application of the color detection technique using MATLAB algorithms, to control the state of the output pins of an Arduino (AT mega 328) and also to control the movements of Bot. This gives an example of implementing Bounding Box algorithm and use of the simple color detection technique and other 'REGIONPROPS' parameters we can control the outputs of an Arduino board and also control the Bot. Service robots directly interact with people, so finding a more natural and easy user interface is of fundamental importance. While earlier works have focused primarily on issues such as manipulation and navigation in the environment, few robotic systems are used with user friendly interfaces that possess the ability to control the robot by natural means. To facilitate a feasible solution to this requirement, we are implementing a system through which the user can give commands to a robot using colored tip of finger. Through this method, the user can control or navigate the robot by using finger with color tip, thereby interacting with the robotic system. The command signals are generated using image processing. These signals are then passed to the robot to navigate it in the specified directions through the Arduino.

Contents

Chapter 1: Introduction **10-11**

- 1.1 Problem Statement 11
- 1.2 Product Design Overview 11

Chapter 2: Project Overview **12-14**

- 2.1 Initialization 13
- 2.2 Image and Video Processing 13
- 2.3 Detection 14
- 2.4 Motion 14
- 2.5 Serial Communication 14

Chapter 3: System Design **15-18**

Chapter 4: Hardware Design **19-34**

- 4.1 Arduino Board (Uno) 20
- 4.2 Motor Driver 21
- 4.3 Bot 23
 - 4.4.1. Chassis 23
 - 4.4.2. DC Motor 24
 - 4.4.3. Caster Wheel 26
- 4.4 Bluetooth Module 26
- 4.5 Power Supply 33

Chapter 5: Software Design **35-42**

5.1	Matlab	36
5.1.1	Why Matlab?	36
5.2	Arduino IDE	37
5.2.1	Why Arduino?	38
5.2.2	How to use Arduino?	38

Chapter 6: Algorithm **43-46**

6.1	Image Processing Algorithm	44
6.2	Bot Control Algorithm	45
6.2.1	Transmitter End	45
6.2.2	Receiver End	46

Chapter 7: Flowchart **47-48**

Chapter 8: Simulations and Results **49-52**

Chapter 9: Pros and Cons **53-56**

9.1	Advantages	54
9.2	Applications	55
9.3	Future Scope	56

Chapter 10: Conclusion **57-58**

References **59**

Acknowledgement **60**

List of Figures

SR. NO.	Figure No.	Title	Page No.
1	Figure 2.1	Block Diagram of Project	13
2	Figure 2.3.1	Object Detection	14
3	Figure 3.1	Transmitter Section	16
4	Figure 3.2	Receiver section	17
5	Figure 3.3	Layout of Chassis	18
6	Figure 4.1	Arduino Uno Pin Outs	20
7	Figure 4.2.1	Pin Configuration of L293D	22
8	Figure 4.2.2	Internal Circuit of L293D	22
9	Figure 4.3.1	Chassis Top View	23
10	Figure 4.3.2	Chassis Front View	23
11	Figure 4.3.3	DC Motor	24
12	Figure 4.3.4	Caster Wheel	26
13	Figure 4.4.1	Bluetooth Module	26
14	Figure 4.4.2	Various Pins of Bluetooth HC-06 and HC-05	28
15	Figure 4.4.3	Pin Out of Bluetooth	29
16	Figure 4.4.4	Connection of Microcontroller and Bluetooth	29
17	Figure 4.4.5	Output Screen while configuring Bluetooth	31
18	Figure 4.5	HW Batteries	33
19	Figure 5.1	Matlab Interface	36
20	Figure 5.2.1	Arduino IDE Interface	37
21	Figure 5.2.2	Arduino Board	39
22	Figure 5.2.3	USB Cable	39
23	Figure 5.2.4	Selecting an Arduino Uno	41
24	Figure 5.2.5	Compiling Sketch in Arduino	42
25	Figure 5.2.6	Uploading Sketch in Arduino	42
27	Figure 8.1	Result at Left Position	50
28	Figure 8.2	Result at Right Position	51
29	Figure 8.3	Result at Up Position	51

30	Figure 8.4	Result at Down Position	52
----	------------	-------------------------	----

List of Tables

SR. NO.	Figure No.	Title	Page No.
1	4.1	Arduino Uno Specifications	20
2	4.5	HW Battery Specification	34

Chapter 1

Introduction

1.1. Problem Statement

1.2. Product Design Overview

In today's age, the robotic industry has been developing many new trends to increase the efficiency, accessibility and accuracy of the systems. Basic tasks could be jobs that are harmful to the human, repetitive jobs that are boring, stressful etc. Though robots can be a replacement to humans, they still need to be controlled by humans itself. Robots can be wired or wireless, both having a controller device. Both have pros and cons associated with them. Many systems exist that are used for controlling the robot through gestures. Some gesture recognition systems involve adaptive colour segmentation, hand finding and labelling with blocking, morphological filtering, and then gesture actions are found by template matching and skeletonising. This does not provide dynamicity for the gesture inputs due to template matching. Another system uses machine interface device to provide real-time gestures to the robot. Analog flex sensors are used on the hand glove to measure the finger bending, also hand position and orientation are measured by ultrasonics for gesture recognition. This system though is not very cost effective.

Beyond controlling the robotic system through physical devices, recent method of object detection control has become very popular. The main purpose of using colored finger tip is that it provides a more natural way of controlling and provides a rich and intuitive form of interaction with the robotic system.

1.1. Problem Statement

Thus, we can summarize our project as follows: the main aim of this project is to design a prototype that can be used in robot vision system. Using the camera to detect the colored object and analyze its movements to drive the robot.

1.2. Product Design Overview

We use colored object movements for robot control. For the simplicity and to make a prototype, we are going to design a small, motorized platform. We will use serial communication to communicate between the camera and the microcontroller. The microcontroller will be placed on the robot which will be connected to the motors, driving the robot in the direction as per the movements of the colored object.

Chapter 2

Project Overview

- 2.1. Initialization**
- 2.2. Image Processing**
- 2.3. Detection**
- 2.4. Motion**
- 2.5. Serial Communication**

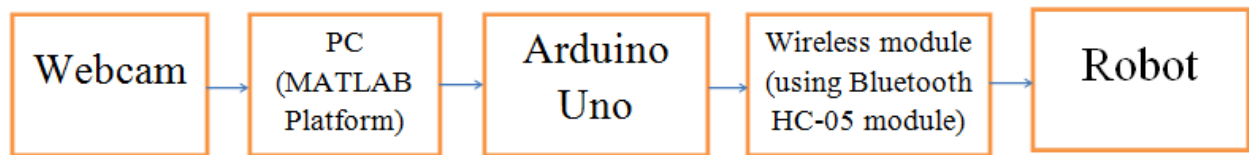


Figure 2.1 Block Diagram of Project

So, now we will have a look at the overall code structure of our algorithm and the logic behind the decision making.

There are two parts in the code structure. The first part is to detect the object movements and the other part is to drive the motors. The code structure can be explained in the following steps:

2.1. Initialization

Initially we set up the serial communication that will be used later for the interface between MATLAB and the controller, the video capture and the program variables.

2.2. Image Processing

The web cam is used as image capturing device. Here the device takes snapshots continuously after a predefined very short interval of time. Due to human's perception of vision these snapshots taken within a very short period of time and displayed continuously might look as a continuous image i.e. video. Depending on the processing speed of the computer being used, this interval for taking snapshots is adjusted.

2.3. Detection

Colored object is detected using Bounding Box algorithm.[1][2]Centroid of the detected object is calculated. Now after detecting the object movements, we have to come up with a decision algorithm that will help the controller to drive the motors accordingly.



Figure 2.3.1 Object Detection

2.4. Motion

Now after detecting the object movements, we have to come up with a decision algorithm that will help the controller to drive the motors accordingly.

2.5. Serial Communication

Now according to the detected command, the MATLAB application will transmit L, R, F or B for left, right, forward and backward respectively to the controller which will drive the motors.

Chapter 3

System Design

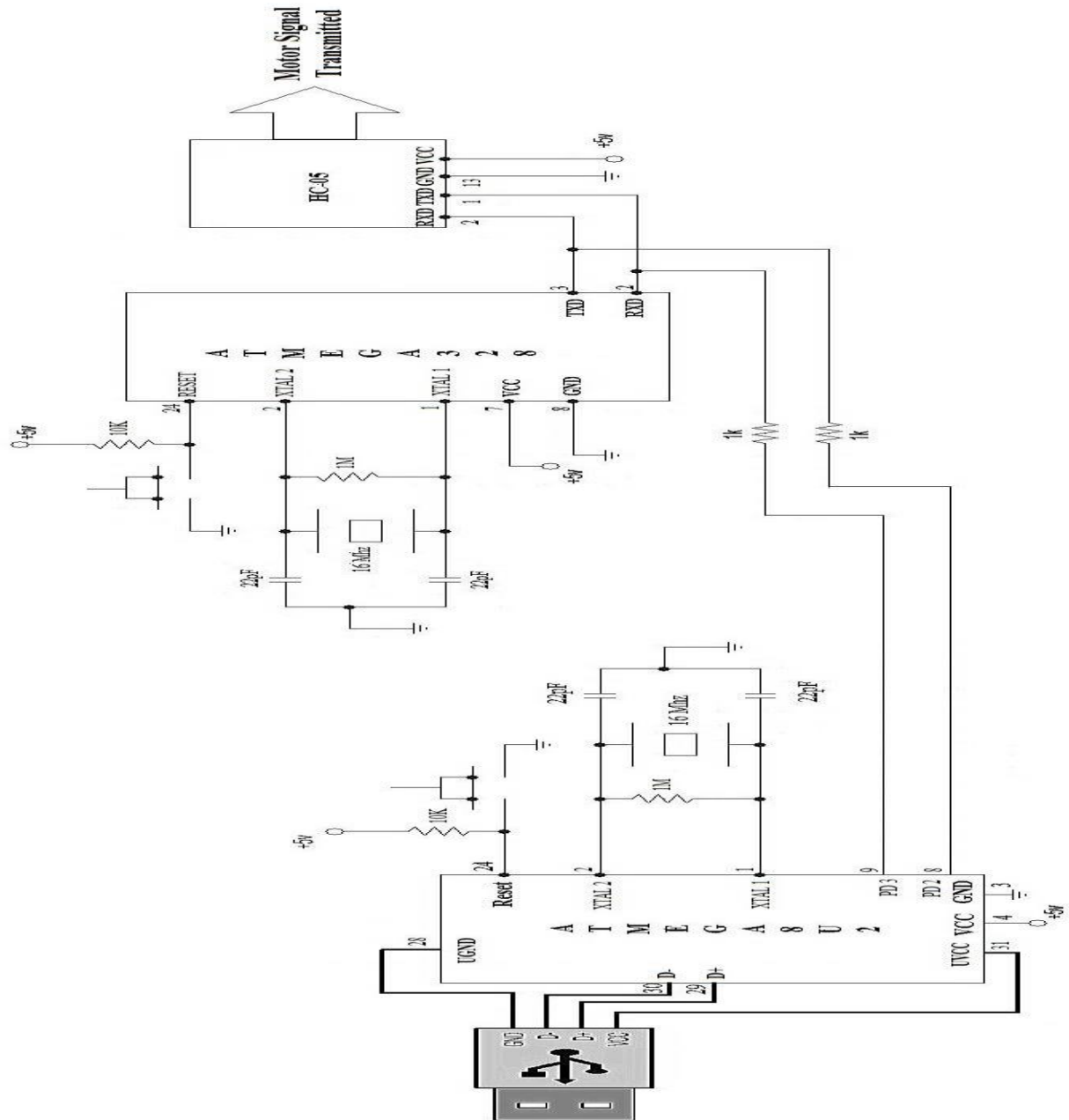


Figure 3.1 Transmitter Section

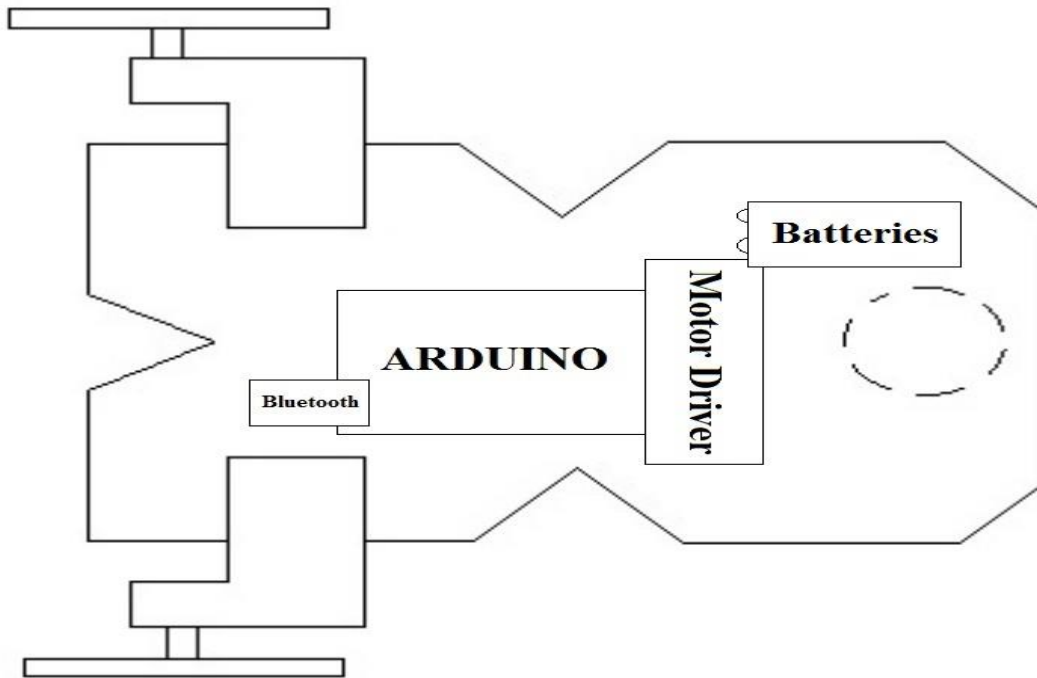


Figure 3.3 Layout of Chassis

Chapter 4

Hardware Design

4.1. Arduino Board (Uno)

4.2. Motor Driver

4.3. Bot

4.3.1. Chassis

4.3.2. DC Motor

4.3.3. Caster Wheel

4.4. Bluetooth Module

4.5. Power Supply

4.1 Arduino Board (UNO)

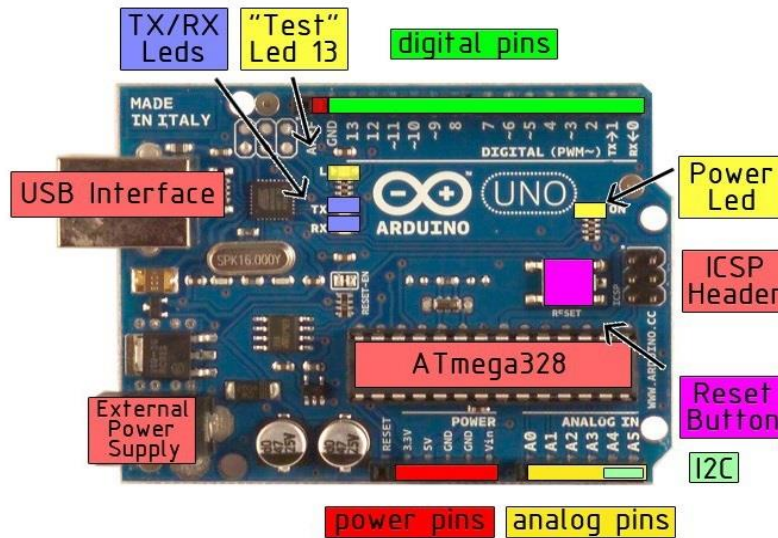


Figure 4.1 Arduino Uno Pin Outs ^[5]

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

Table 4.1 Arduino Uno Specifications

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by boot loader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 Hz

4.2. Motor Driver

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. Dual H-bridge Motor Driver integrated circuit (IC).

It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. As you know voltage need to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, hence H-bridge IC are ideal for driving a DC motor.

In a single l293d chip there two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors. Given below is the pin diagram of a L293D motor controller.

There are two Enable pins on l293d. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge you need to enable pin 1 to high. And for right H-Bridge you need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend working. It's like a switch.

L293D Pin Diagram

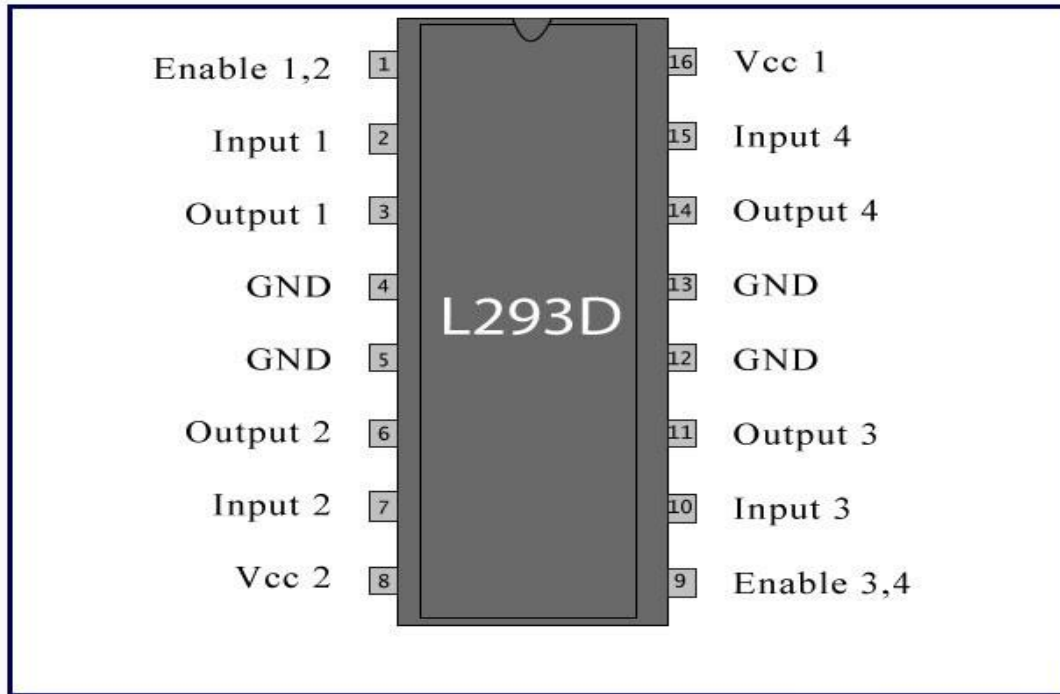


Figure 4.2.1 Pin Configuration of L293D ^[6]

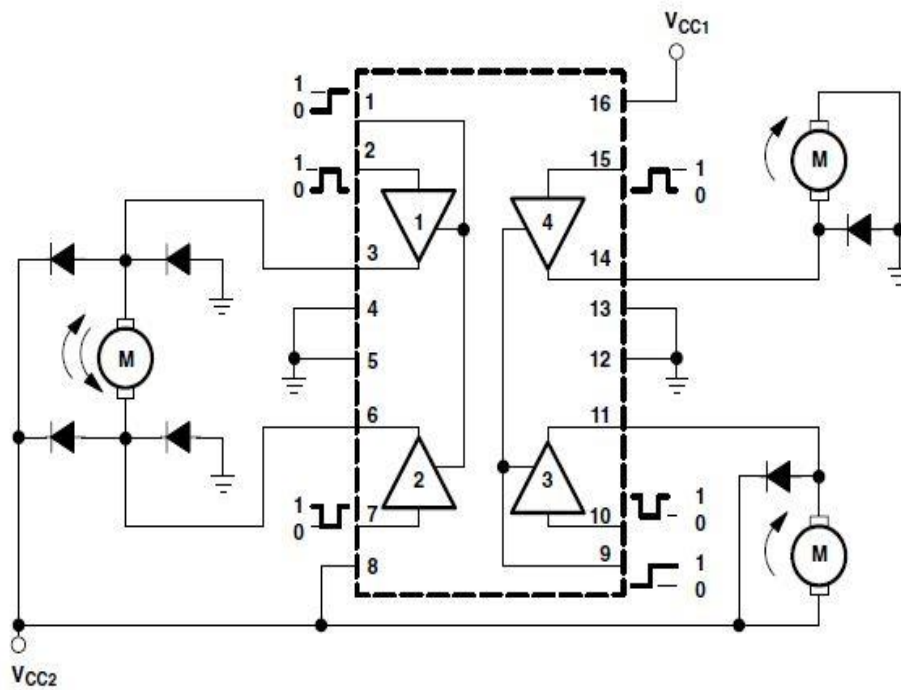


Figure 4.2.2 Internal Circuit of L293D ^[7]

4.3. BOT

4.3.1. Chassis

Fiberglass is an outstanding material for creating specific shapes that would be nearly impossible to make from metal or wood. It is also extremely strong and rigid once set, as well as waterproof. The process involves laying a fiberglass cloth and then applying a two-part resin on top of the cloth. It only takes about 1 hour to harden, but it does make some strong fumes.



Figure 4.3.1 Chassis Top View

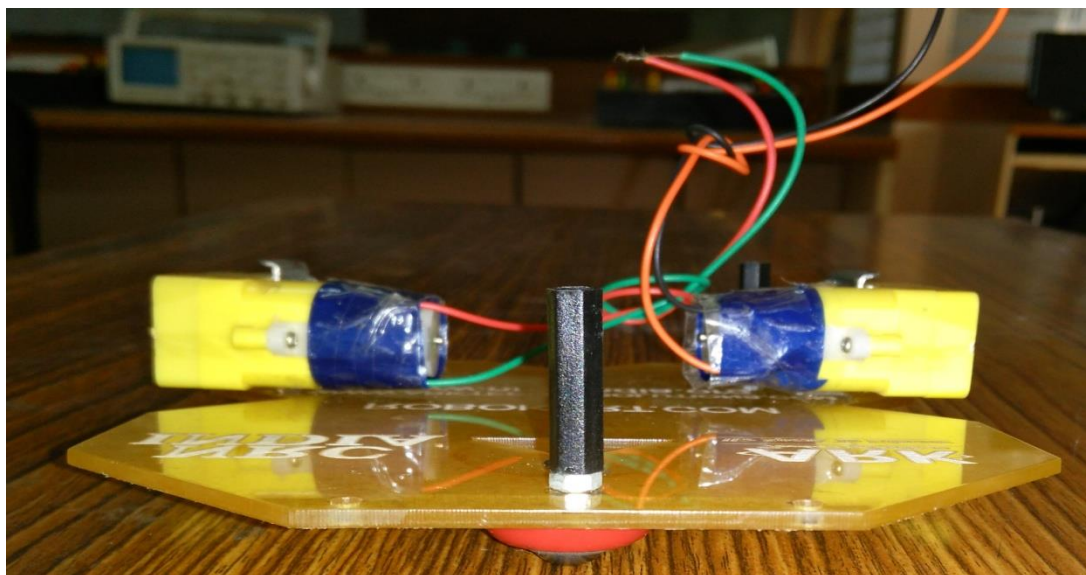


Figure 4.3.2 Chassis Front View

4.3.2. DC Motor



Figure 4.3.3 DC Motor

This is a 300 RPM low cost single shaft straight DC geared motor. It is most suitable for light weight robot requiring small power. This motor can be used with 69mm Diameter Wheel for Plastic Gear Motors and 87mm Diameter Multipurpose Wheel for Plastic Gear Motors.

Drive shaft has clutch for non-continuous protection from overload which protects gears from the sudden overload. Motor runs smoothly from 2V to 12V and gives wide range of RPM, and torque. Table below gives fairly good idea of the motor's performance in terms of RPM, no load current as a function of voltage and stall torque, stall current as a function of voltage.

Features:

1. Cost effectiveness of the injection-molding process.
2. Elimination of machining operations.
3. Low density: lightweight, low inertia.
4. Uniformity of parts.
5. Capability to absorb shock and vibration as a result of elastic compliance.
6. Ability to operate with minimum or no lubrication, due to inherent lubricity.

7. Relatively low coefficient of friction.
8. Corrosion-resistance; elimination of plating, or protective coatings.
9. Quietness of operation.
10. Tolerances often less critical than for metal gears, due in part to their greater resilience.
11. Consistency with trend to greater use of plastic housings and other components.

Limitations:

1. Less load-carrying capacity, due to lower maximum allowable stress; the greater compliance of plastic gears may also produce stress concentrations.
2. Plastic gears cannot generally be molded to the same accuracy as high-precision machined metal gears.
3. Plastic gears are subject to greater dimensional instabilities, due to their larger coefficient of thermal expansion and moisture absorption.
4. Reduced ability to operate at elevated temperatures; as an approximate figure, operation is limited to less than 120°C. Also, limited cold temperature operations.
5. Initial high mold cost in developing correct tooth form and dimensions.
6. Can be negatively affected by certain chemicals and even some lubricants.
7. Improper molding tools and process can produce residual internal stresses at the tooth roots, resulting in over stressing and/or distortion with aging.
8. Costs of plastics track petrochemical pricing, and thus are more volatile and subject to increases in comparison to metals.

4.3.3. Caster Wheel

A **caster** (or **castor**) is an undriven, single, double, or compound **wheel** that is designed to be mounted to the bottom of a larger object (the "vehicle") so as to enable that object to be easily moved. They are available in various sizes, and are commonly made of rubber, plastic, nylon, aluminum, or stainless steel.



Figure 4.3.4 Caster Wheel

4.4 Bluetooth Module HC-05

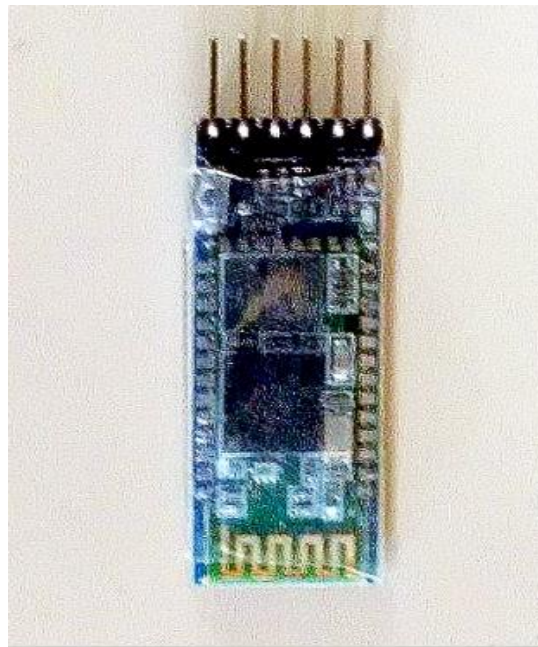


Figure 4.4.1 Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications:

Hardware features

- Up to +4dBm RF transmit power.
- Low Power 1.8V Operation, 3.3 to 5 V I/O.
- PIO control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

Software features

- Slave default Baud rate: 9600, Data bits: 8, Stop bit: 1, Parity: No parity.
- PIO9 and PIO8 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval,
- While disconnected only blue led blinks 2times/s.
- Auto connect to last device on power as default.
- Permit pairing device to connect as default.
- Auto pairing **PINCODE:"1334"** as default.
- Auto reconnect in 30 min when disconnected as a result of beyond the range of operation.

Information of Package

The PIN definitions of HC-03, HC-04, HC-05 and HC-06 are kind of different, but the package size is the same: 28mm * 15mm * 2.35mm. The following figure 1 is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs.

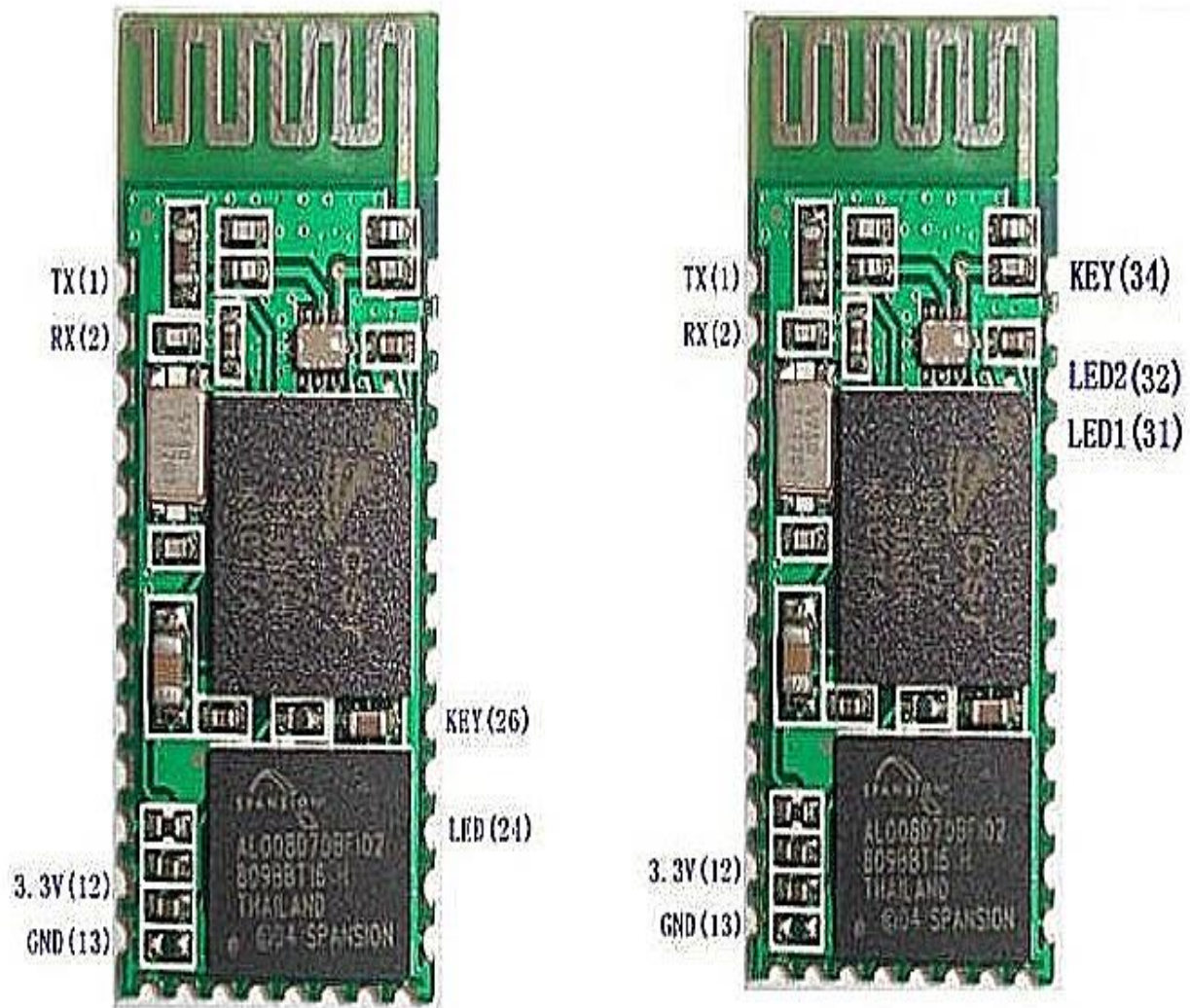


Figure 4.4.2 Various Pins of Bluetooth HC-06 and HC-05^[8]

Pin out Configuration

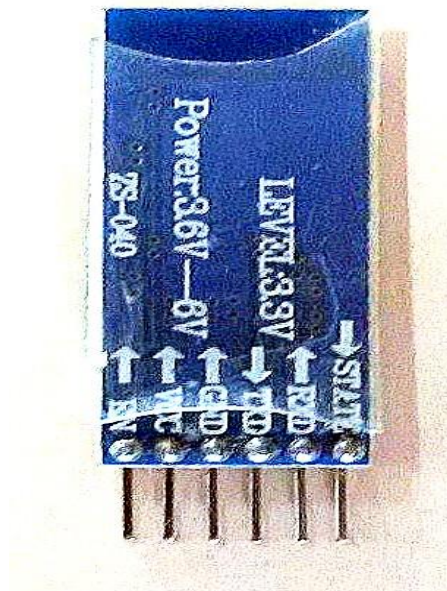


Figure 4.4.3 Pin Out of Bluetooth

Application Circuit

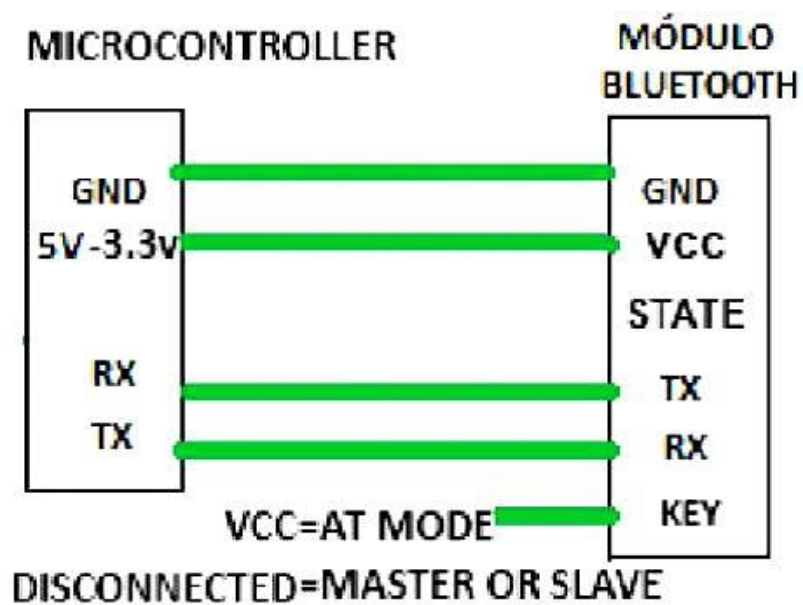


Figure 4.4.4 Connection of Microcontroller and Bluetooth^[9]

4.4.1 How to Configure Bluetooth HC-05

For our project we have used two HC-05 Bluetooth modules. One will be located at Transmitter Section and other will be place on BOT to receive the data. For this we have configured both HC-05. One is configured for master mode which will transmit the data and other is configured in slave mode which will receive the data. The hc-05 placed at Transmitter section is act as master and the one on BOT is act as Slave.

Now let us understand how we have configured the HC-05.

Most probably for any HC-05 module any following pins are so much imp for working.

1. **GND**
2. **VCC**
3. **RX**
4. **TX**
5. **KEY**

For any HC-05 to configure we have to put it AT command mode. We have done this by following steps.

- 1) First of all downloaded the code from following link and burned it in the arduino
<http://www.instructables.com/id/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/step2/The-Arduino-Code-for-HC-05-Command-Mode/>
- 2) Then make connection as follows
 - HC-05 GND --- Arduino GND Pin
 - HC-05 VCC (5V) --- Arduino 5V
 - HC-05 TX --- Arduino Pin 10 (soft RX)
 - HC-05 RX --- Arduino Pin11 (soft TX)
 - HC-05 Key (PIN 34) --- Arduino Pin 9

- 3) If key pin out is not there then there will one switch which will act as key.
- 4) Now to put the HC_05 in command mode first remove the VCC Pin. Then press the button on HC-05 which is key button. Keep it pressed then apply VCC to HC-05 then release the button. So due to this HC-05 will enter in AT command mode this will be indicated by LED which will start blinking slowly
- 5) By making these connections open the serial monitor on arduino IDE. Select the baud rate 9600. And send serially AT. If connections are correct then we will get reply OK.

This means we have successfully putted the HC-05 in AT command mode.

- 6) Now there are so many AT commands available for HC-05. But Some of the important that we have used are as follows
 - a. AT+NAME
 - b. AT+ADDR
 - c. AT+ROLE
 - d. AT+PSWD
 - e. AT+BIND
 - f. AT+CMODE
 - g. AT+UART

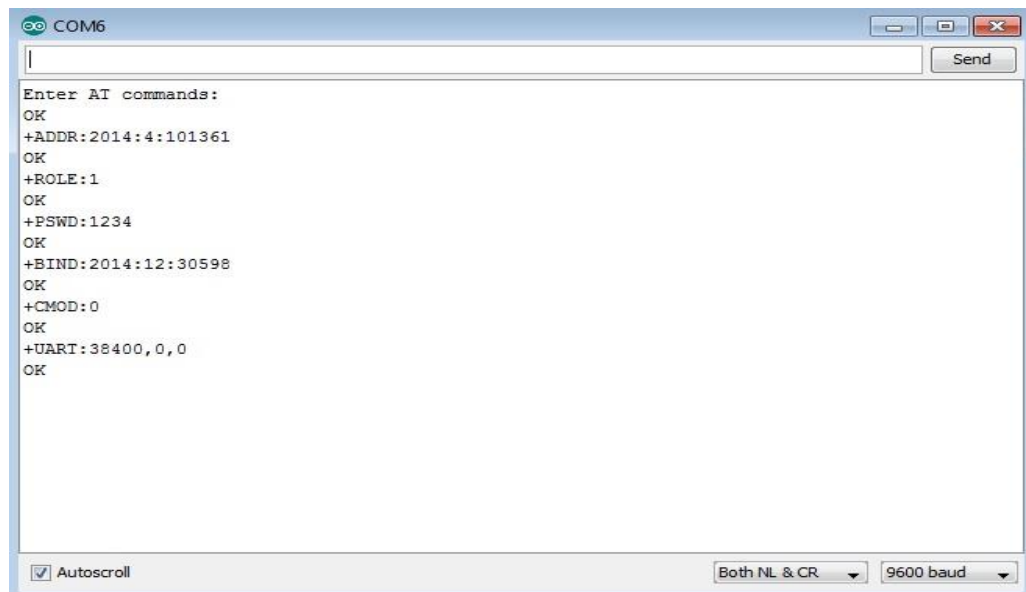


Figure 4.4.5 Output Screen while configuring Bluetooth

- a. AT+NAME
 - i. BY using this command we can change the name of HC-05.
 - ii. E.g. AT+NAME=emocbot

- b. AT+ADDR
 - i. By using this command we can find the 48 bit MAC address of HC-05. This MAC address is very important for auto pairing
 - ii. E.g. AT+ADDR
 - iii. Response- 12: 34: 56: ab: cd: ef

- c. AT+ROLE
 - i. Now this will decide the role of HC-05 whether it is Master or Slave
 - ii. AT+ROLE=0; HC-05 will act as Slave
 - iii. AT+ROLE=1; HC-05 will act as Master

- d. AT+PSWD
 - i. By using this command we can set the password for pairing. For auto pairing password for both modules should be same. Default it '1234'.
 - ii. E.g. AT+PSWD=0000

- e. AT+BIND
 - i. By using this command we can tell the master that to which slave u have to connect. For this we have to add slave's mac address in master. This is done by using this command.
 - ii. E.g. AT+BIND=1234, 56, abcdef

- f. AT+CMODE

- i. This will ensure the function of master. This will tell master whether to connect only specified mac address which is putted using BIND or to connect any module.
- ii. AT+CMODE=0; connect to the specified address
- iii. AT+CMODE=1; connect to any available slaves

g. AT+UART

- i. By using this command we can change the baud rate of data transmission of HC-05.
- ii. E.g. AT+UART= 38400

Thus By using all this steps we have configured HC-05 Bluetooth Module.

4.5 Power Supply

HIGH-WATT (HW) 9v Batteries



Figure 4.5 HW Batteries

Description:

- Use them for Toys, Electronic Equipment, Radio's and other devices that requires this battery size!
- This battery is a high capacity & low cost solution for many electronic devices.
- It is used with its specific battery snap, connector or clip.
- Sealed in original package

Table 4.5 HW Battery Specification

Model	6F22M
System	Zinc Carbon
Nominal Voltage(V)	9V
Discharge Resistance(Ω):	620
Cut-off Voltage(V)	5.4
Discharge time	270HM,9HOURS
Jacket	Metal
Operating Temperature Range	(deg. C)** -20 to +85
Weight	(g)*** 2.0
Dimensions	26. 5 H x 48.5 W x 17. 5 L (mm)
CERTIFICATION	ULCEROHSISO9001~2000

*Nominal capacity indicates duration until the voltage drops down, when discharged at a nominal discharge current at 20 deg. C.

** When using these batteries at temperatures outside the range of 0 to +40 deg. C, please check in advance for conditions of use.

*** Dimensions and weight are for the battery itself, but may vary depending on the shape of terminals or other factors.

Chapter 5

Software Design

5.1. Matlab

5.1.1. Why Matlab?

5.2. Arduino

5.2.1 Why Arduino?

5.2.2 How to use Arduino?

5.1 Matlab

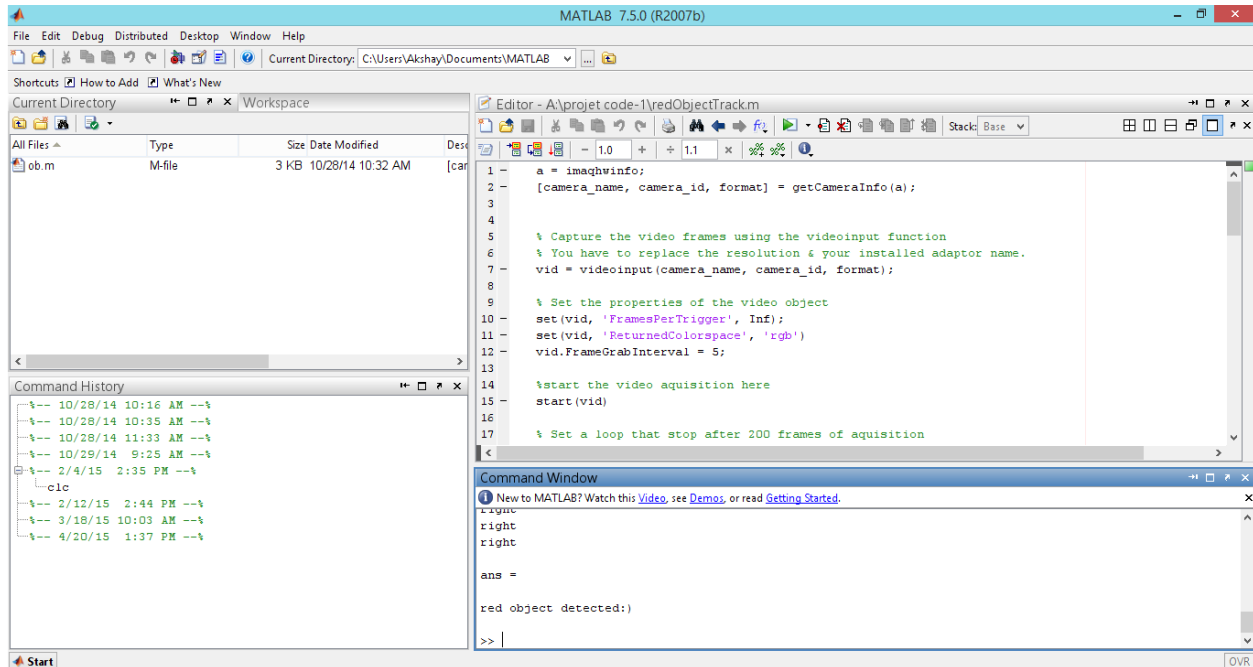


Figure 5.1 Matlab Interface

5.1.1. Why Matlab?

- A very large (and growing) database of built-in algorithms for image processing and computer vision applications
- MATLAB allows you to test algorithms immediately without recompilation. You can type something at the command line or execute a section in the editor and immediately see the results, greatly facilitating algorithm development.
- The MATLAB Desktop environment, which allows you to work interactively with your data, helps you to keep track of files and variables, and simplifies common programming/debugging tasks.
- The ability to read in a wide variety of both common and domain-specific image formats.
- The ability to call external libraries, such as OpenCV
- Clearly written documentation with many examples, as well as online resources such as web seminars ("webinars").
- Bi-annual updates with new algorithms, features, and performance enhancements
- If you are already using MATLAB for other purposes, such as simulation, optimization, statistics, or data analysis, then there is a very quick learning curve for using it in image processing.

- The ability to process both still images and video.
- Technical support from a well-staffed, professional organization (assuming your maintenance is up-to-date)
- A large user community with lots of free code and knowledge sharing
- The ability to auto-generate C code, using MATLAB Coder, for a large (and growing) subset of image processing and mathematical functions, which you could then use in other environments, such as embedded systems or as a component in other software.

5.2. Arduino IDE:

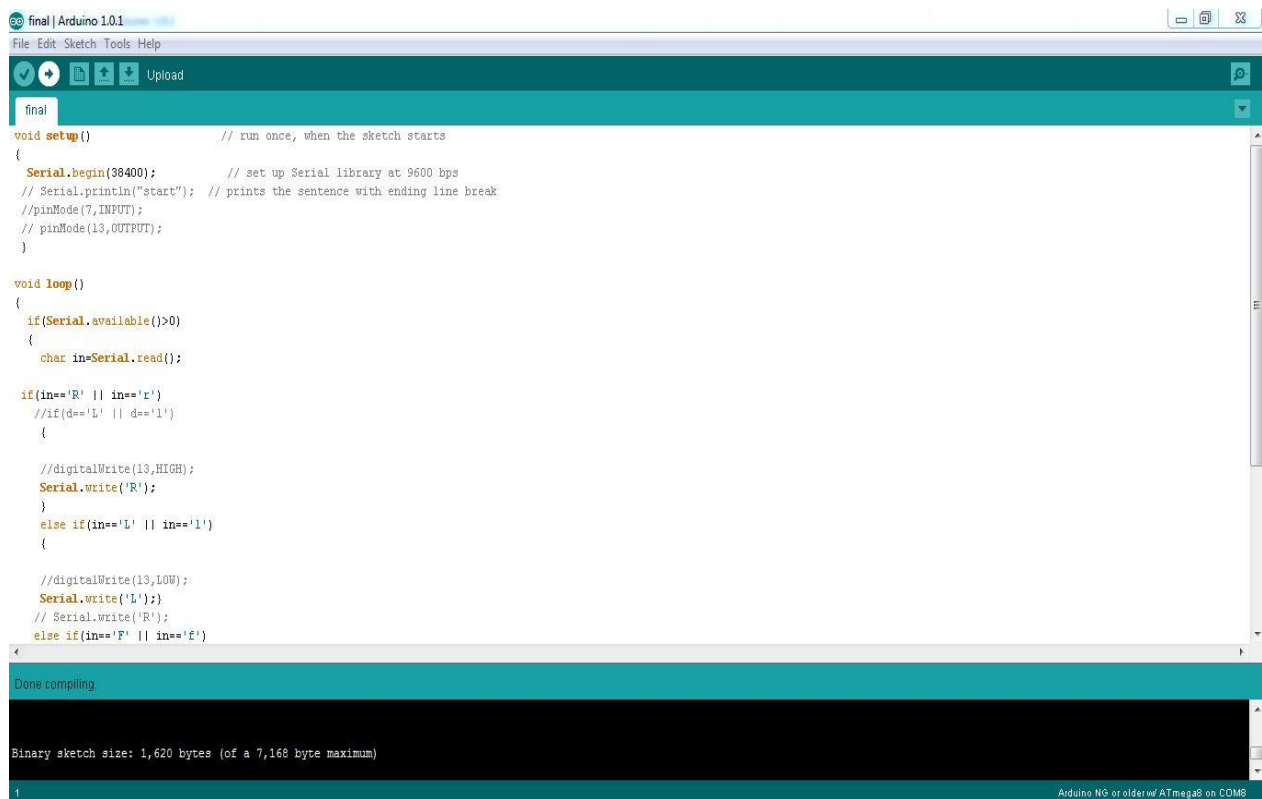


Figure 5.2.1 Arduino IDE Interface

The open-source *Arduino software (ide)* makes it easy to write code and upload it to the board. It runs on windows, mac os x, and Linux. The environment is written in java and based on processing and other open-source software. This software can be used with any Arduino board.

5.2.1. Why Arduino?

- It is an **open-source project**, software/hardware is extremely **accessible** and very flexible to be customized and extended.
- It is **flexible**, offers a variety of digital and analog inputs, *SPI* and serial interface and digital and *PWM* outputs.
- It is **easy to use**, connects to computer via USB and communicates using standard serial protocol, runs in standalone mode and as interface connected to PC/Macintosh computers.

5.2.2. How to use Arduino?

- 1) Get an Arduino board and USB cable
- 2) Download the Arduino environment
- 3) Connect the board
- 4) Install the drivers (CP210x)
- 5) Launch the Arduino application
- 6) Open the code to upload
- 7) Select your board
- 8) Select your serial port
- 9) Upload the program

1) Get an Arduino board and USB cable

You can use an Arduino Uno, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila.

You also need a standard USB cable (A plug to B plug): the kind you would connect to a USB printer, for example. (For the Arduino Nano, you'll need an A to Mini-B cable instead.)

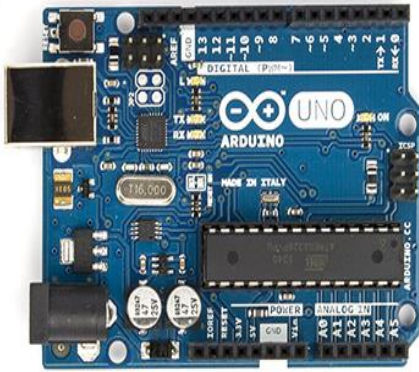


Figure 5.2.2 Arduino Board



Figure 5.2.3 USB Cable

2) Download the Arduino environment

When the download finishes, unzip the downloaded file. Make sure to preserve the folder structure. Double-click the folder to open it. There should be a few files and sub-folders inside.

3) Connect the board

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you're using an Arduino Diecimila, you'll need to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it's on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labelled **PWR**) should go on.

4) Install the drivers (CP210x)

Installing drivers for the Arduino Uno or Arduino Mega 2560 with Windows 7, Vista, or XP:

- Plug in your board and wait for Windows to begin its driver installation process. After a few moments, the process will fail, despite its best efforts
- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
- Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "**arduino.inf**", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "**Arduino UNO.inf**".
- Windows will finish up the driver installation from there.

5) Launch the Arduino application

Double-click the Arduino application. (Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog.

6) Open the code to upload

File > Open > Code.ino

7) Select your board

You'll need to select the entry in the **Tools > Board** menu that corresponds to your Arduino.

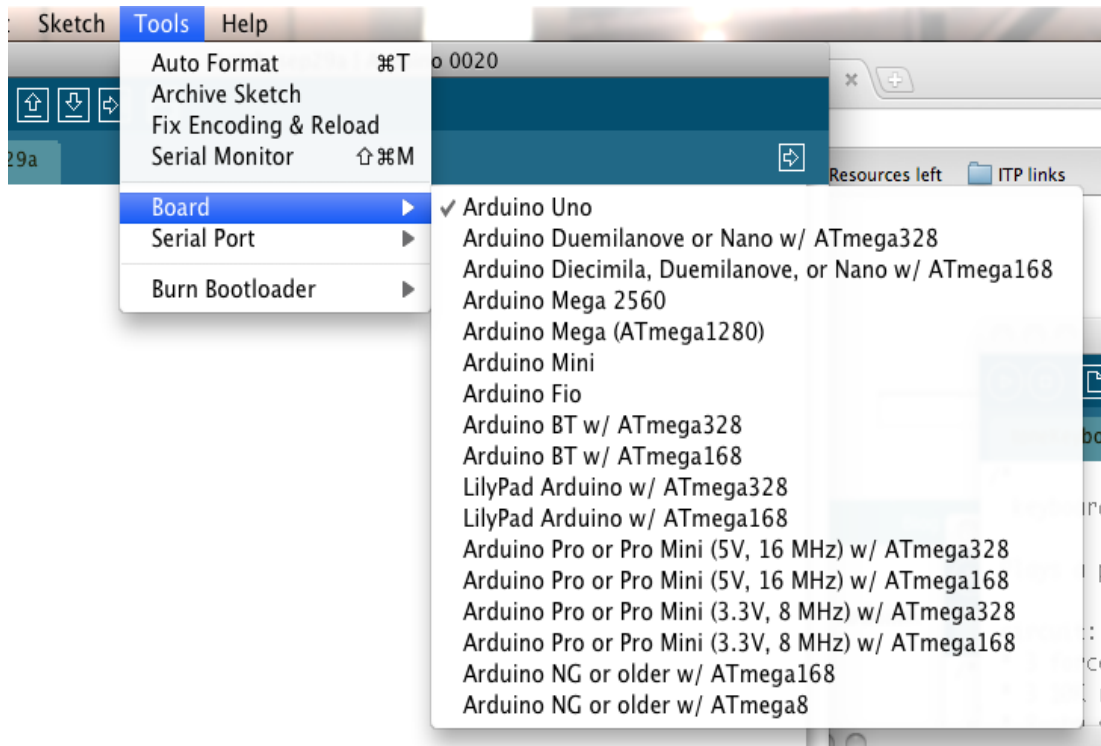


Figure 5.2.4 Selecting an Arduino Uno

For Duemilanove Arduino boards with an ATmega328 (check the text on the chip on the board), select **Arduino Duemilanove or Nano w/ ATmega328**. Previously, Arduino boards came with an ATmega168; for those, select **Arduino Diecimila, Duemilanove, or Nano w/ ATmega168**.

8) Select your serial port

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be **COM3** or higher (**COM1** and **COM2** are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

9) Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading."

will appear in the status bar. (*Note:* If you have an Arduino Mini, NG, or other board, you'll need to physically present the reset button on the board immediately before pressing the upload button.)

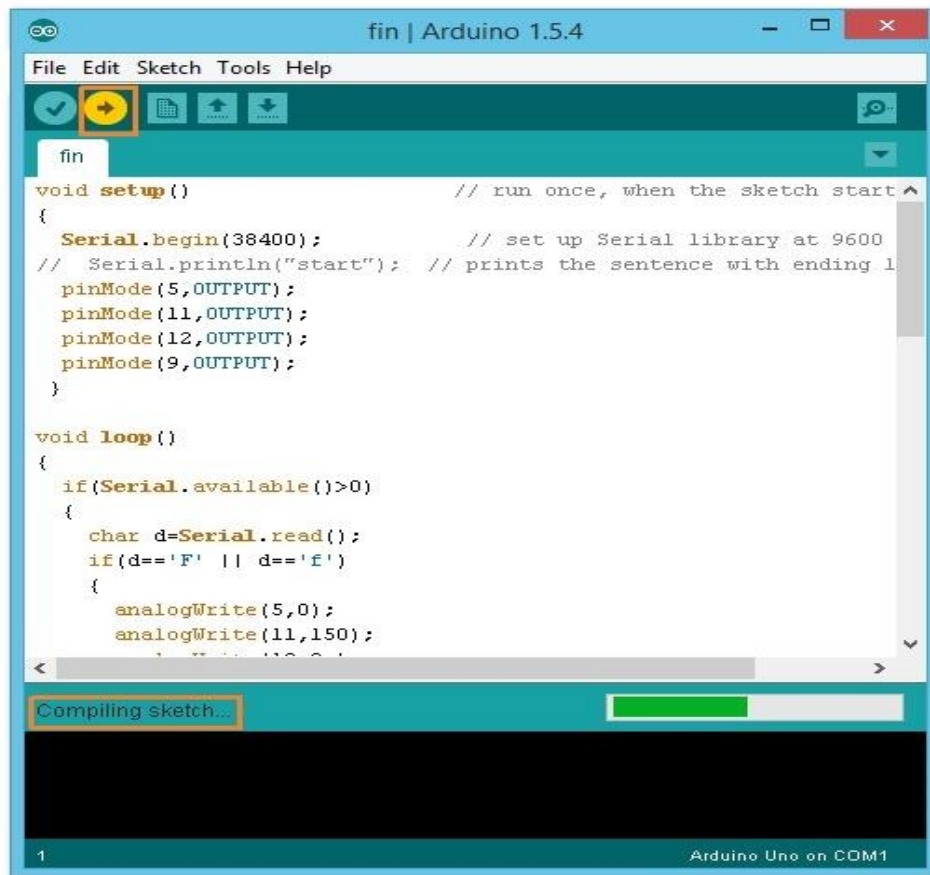


Figure 5.2.5 Compiling Sketch in Arduino

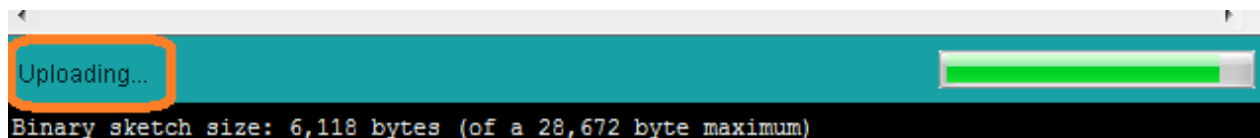


Figure 5.2.6 Uploading Sketch in Arduino

Chapter 6

Algorithm

6.1. Image Processing Algorithm

6.2 Bot control Algorithm

6.2.1 Transmitter End

6.2.2 Receiver End

6.1 Image Processing Algorithm

Step 1. Start

Step 2. Get the details of the hardware device interfaced with system for imaging.

Step 3. Declare the Serial COM and start the serial communication and webcam.

Step 4. Start the Video input.

Step 5. Run image grabbing function inside while loop.

Step 6. Take snapshot.

Step 7. From required color index, perform image subtraction.

Step 8. Apply median filter for noise cancelation and convert the image into black and white image.

Step 9. Extract the interested color region and label them.

Step 10. Calculate the “regionprops”[4] of each of the labelled components of image.

Step 11. Calculate the Centroid of the detected region.

Step 12. . Track the Centroid path as per the movement of the colored object and generate the corresponding movement signals. e.g.Left, Right, Forward, Backward etc.

Step 13. If count of the snapshots reached then Stop else go to step 6.

6.2 Bot control Algorithm

6.2.1 Transmitter End

Step1. Initialize and Start the Serial Communication.

Step2. Check for if Serial Data is available

Step3. If available follow step 4 else go to step 8

Step4. Declare char in to Read the Serial Data

Step5. Check If serial data is “R” if yes Serially Write ‘R’

Step6. If Serial Data is “L” , then serially write ‘L’

Step7. Or if Serial Data is “F” , then serially write ‘F’

Step8. if Serial Data is “B”, then serially write ‘B’

Step9. Repeat

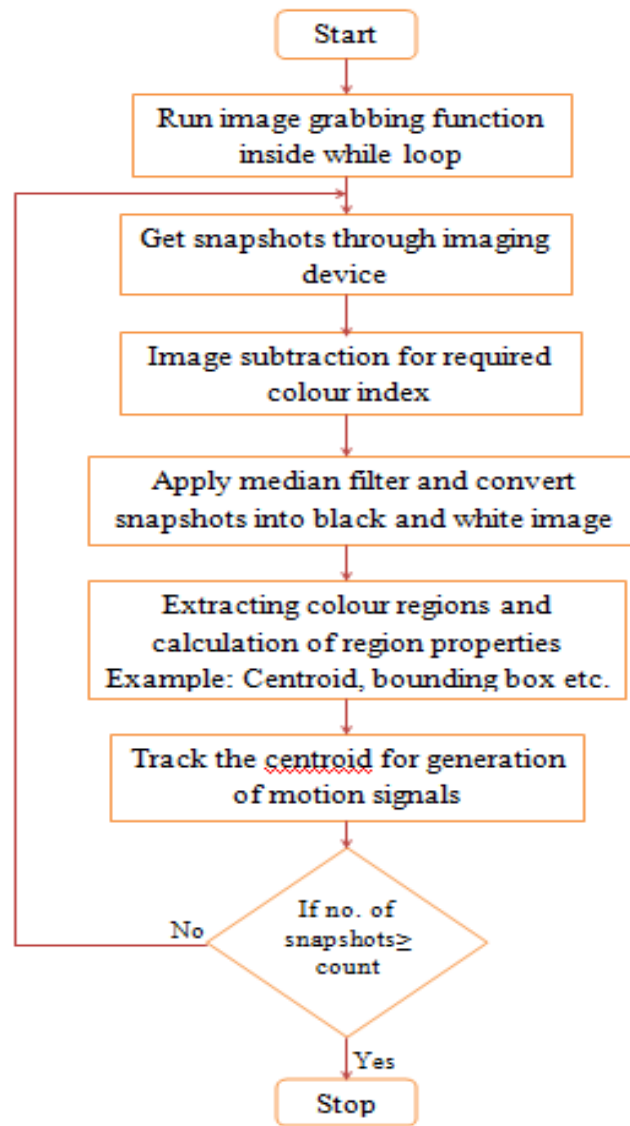
6.2.2 Receiver End

- Step1.** Initialize and Start the Serial Communication.
- Step2.** Declare Pin 7, 8, 12 and 13 as output pins for Bot connection
- Step3.** Check for if Serial Data is available
- Step4.** If available follow step 5 else go to step 11
- Step5.** Declare char d to serially read the data
- Step6.** If serial data is “F” analog write 1 on pins 7, 8, 12 and 13 to move bot forward.
- Step7.** If serial data is “B” analog write 0 on pins 7, 8, 12 and 13 to move bot backward.
- Step8.** If serial data is “L” analog write 0 on pins 12, 13 and write 1 on pins 7 and 8 to move bot
Left.
- Step9.** If serial data is “R” analog write 1 on pins 12 , 13 and write 0 on pins 7 and 8 to move
Bot to right.
- Step10.** Repeat.

Chapter 7

Flowchart

FLOWCHART:



Chapter 8

Simulations and Results

Simulation

The simulation was carried out on Matlab, the code was executed and the simulation results were as shown as follows:

Left Position

- When the object is moving **left**.

Result: The command window displays the message “**Left**”.



Figure 8.1 Result at Left Position

Right Position

- When the object is moving **right**.

Result: The command window displays the message “**Right**”



Figure 8.2 Result at Right Position

Forward Position

- When the object is moving **up**.

Result: The command window displays the message “UP”

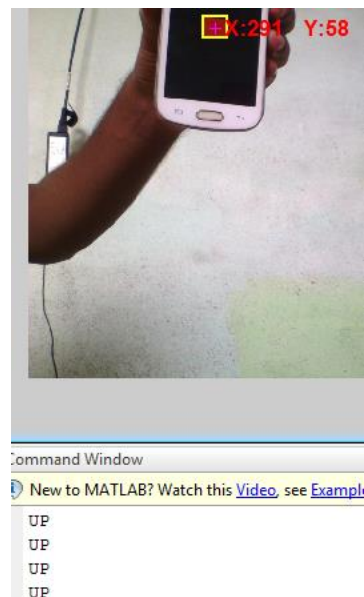


Figure 8.3 Result at Up Position

Backward Position

- When the object is moving **down**.

Result: The command window displays the message “Backward”

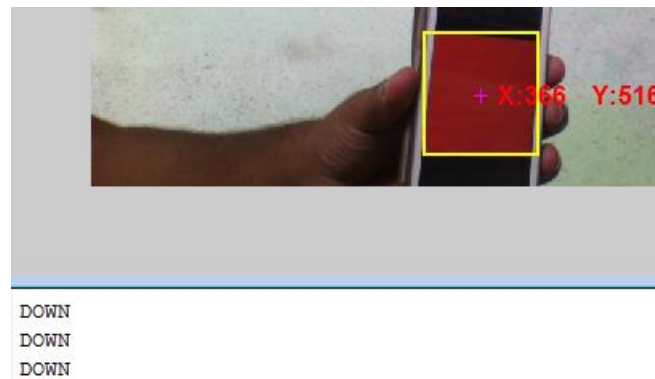


Figure 8.4 Result at Down Position

Chapter 9

Pros and Cons

- 9.1. Advantages**
- 9.2. Applications**
- 9.3. Future Scope**

9.1. Advantages

1. The project requires only *one time investment* of components.
2. Cost of the project is also very *cheap and affordable*.
3. *Maintenance* cost is very *low*.
4. Light weight and can be made *portable*.
5. Any color object can be used for generation of *movement signals*.

9.2 Applications

1. To control the *movements of robotic arm* in unmanned regions.
2. Object Detection algorithm can be used in *robot vision*.
3. In *pick and place robot* for particular color objects.

9.3 Future Scope

1. Further the object detection can be improved by eliminating unwanted detection of color while execution of Bounding Box algorithm except the target object, multiple objects detection.
2. The *wireless communication* can be made from the camera to the main controller board.
3. The entire project can be burned into a Raspberry Pi so that the need of always carrying a laptop or Personal Computer can be avoided, and project becomes easy to carry.
4. The robot can move left, right, front and backwards, but more detailed studies is to be done to rotate it in intermediate angles using the object movements more accurately.
5. The idea can be implemented to *control objects at remote places*.

Chapter 10

Conclusion

Conclusion:

The main aim of the project was to detect the colored object's movement and how that can be applied to control the movement of the Robot. It also gives us an opportunity to control objects in Remote places.

The project is successfully running in real-time and further scope is added to improve the accuracy of the object detection system.

References:

- [1] Raquib Buksh, Soumyajit Routh, Parthib Mitra, Subhajit Banik, Abhishek Mallik, Sauvik Das Gupta, "Implementation of MATLAB based object detection technique on Arduino board and iROBOT CREATE", International Journal of Scientific and Research Publications, Volume 4, Issue 1, January 2014.
- [2] Shubham Srivastava, Pratibha Singh, "Real Time Object Tracking Using Color Feature", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-8, January 2014
- [3] Aji Joy, Ajith P Somaraj, Amal Joe, Muhammed Shafi, Nidheesh T M, "Ball Tracking Robot Using Image Processing and Range Detection", International Journal of Innovative Research in Computer and Communication Engineering Vol. 2, Issue 3, March 2014.
- [4] Region Properties :<http://www.mathworks.in/help/images/ref/regionprops.html>
- [5] Arduino Uno Board –
<http://docs-europe.electrocomponents.com/webdocs/0e8b/0900766b80e8ba21.pdf>
- [6] LM293D Pin diagram
<http://www.engineersgarage.com/electronic-components/l293d-motor-driver-ic>
- [7] Internal Circuit of L293D
<http://www.ti.com/lit/ds/symlink/l293.pdf>
- [8] http://www.rcscomponents.kiev.ua/datasheets/hc_hc-05-user-instructions-bluetooth.pdf
- [9] <http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>

ACKNOWLEDGEMENT

The success of the project depends largely on the encouragement and guidelines of the Project Guide. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. First and foremost we would like to express our deep sense of gratitude towards our Project Guide, Mr. Shailesh Nandgaonkar whose priceless guidance and advice helped us to complete this Project. We sincerely thank her for her invaluable motivation during the course of project work. Her constant probe into the finest details enabled us to understand the topic and thereby helped us to complete the project in stipulated amount of time. Her foresightedness and organized skills have really helped us in the completion of this project.

We specially thank to Mr. Parmeshwar Birajdar for his grateful guidance and advice for our project. He was ready whenever we had asked him for his help in our project. We are highly indebted to him for his timely assistance and expertise in the field of wavelets.

We thank our Principal, Dr. (Mrs.) J. M. Nair and the Head of the Department (Electronics), Mrs. Kavita Tewari for the use of college and departmental facilities for our project. We would also like to thank members of the Electronics department for their direct and indirect help at opportune times.

We take this opportunity to thank our friends and family for their technical help and moral support. Finally, we would like to thank everybody who was important to the successful realization of the thesis.

Akshay Chidrawar
Vaibhav Khedkar
Ajay Patkar
Vishal Thakur

Robot Navigation Using MATLAB'S Bounding Box Algorithm

SUBMITTED BY:

- | | |
|---------------------|------|
| 1. AKSHAY CHIDRAWAR | (11) |
| 2. VAIBHAV KHEDKAR | (34) |
| 3. AJAY PATKAR | (56) |
| 4. VISHAL THAKUR | (71) |

GUIDED BY:

MR. SHAILESH NANDGAONKAR

CONTENTS

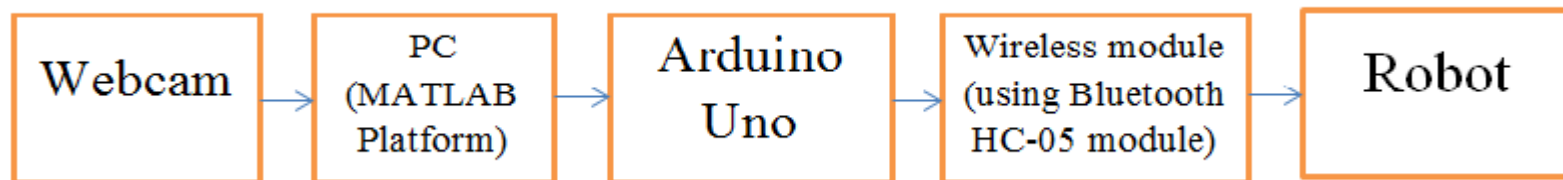
- ❑ Introduction
- ❑ Block Diagram
- ❑ Hardware Tool
- ❑ Software Tool
- ❑ Algorithm
- ❑ Application
- ❑ References

INTRODUCTION

- In today's age, the robotic industry has been developing many new trends to increase the efficiency, accessibility and accuracy of the systems. Basic tasks could be jobs that are harmful to the human, repetitive jobs that are boring, stressful etc.
- Robots can be wired or wireless, both having a controller device.
- Many systems exist that are used for controlling the robot through gestures involve adaptive colour segmentation , hand finding and labelling with blocking, morphological filtering.
- Not very cost effective.

INTRODUCTION

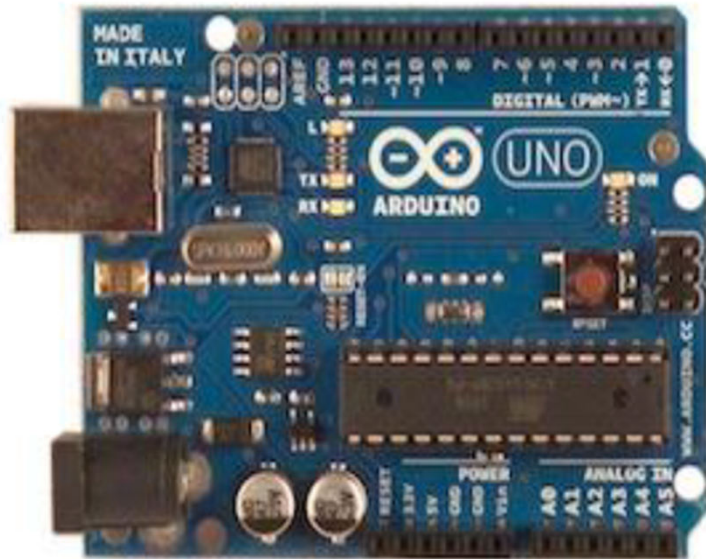
In this we are going to take advantage of the Bounding Box technique, present in image processing tool box to detect the centroid of a particular colored image, and the change of the centroid co-ordinates will be detected which will be applied to generate different commands.



These commands would be fed to an Arduino to change the states of its output pins, and, to control the movement of the Bot in a particular direction.

HARDWARE TOOL

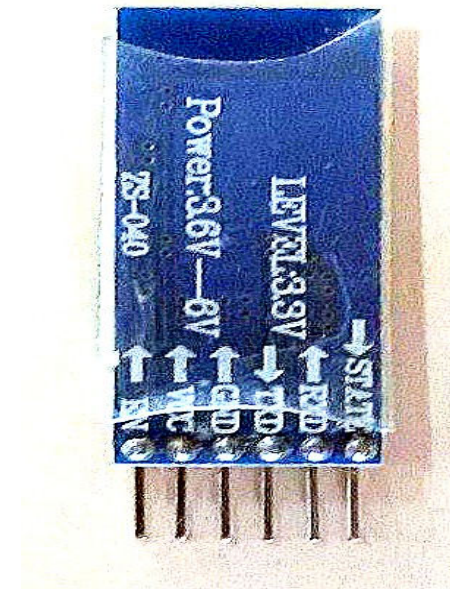
ARDUINO ATMEGA328



	ATmega328
Flash Memory	32KB; 0.5KB used by boot loader
SRAM	2KB
EEPROM	1KB
Clock speed	16MHz
Digital I/O pins	14(of which 6 provides PWM output)
Analog Input pins	6

HARDWARE TOOL

Bluetooth Module HC-05



Pin Out of Bluetooth

SOFTWARE TOOL

1. MATLAB

2. Arduino Programming

3. Bluetooth Module HC-05



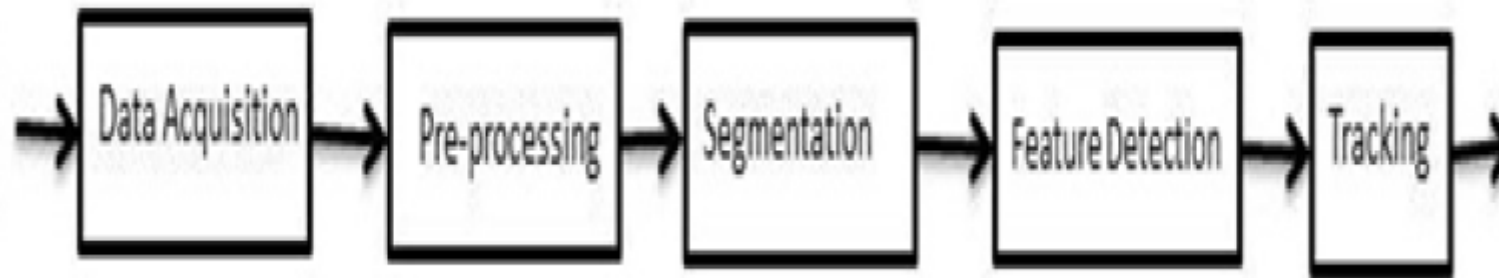
ALGORITHM

Movement detection using MATLAB:

Image acquisition is a crucial part for movement detection. To get the detail of the hardware device interfaced for imaging imaqhwinfo command is used. Snapshots are taken using imaging device and Bounding Box algorithm is applied on captured images.

1. Grab an image using Image Acquisition toolbox.
2. Run image grabbing function inside a loop(while loop is advised).
3. From the required color index, perform image subtracting function.
4. Apply median filter for noise cancelation and convert the image into black and white image.
5. Calculate the 'regionprops' of each of the component of the image.

ALGORITHM



Now tracking the centroid of the bounded region is done.

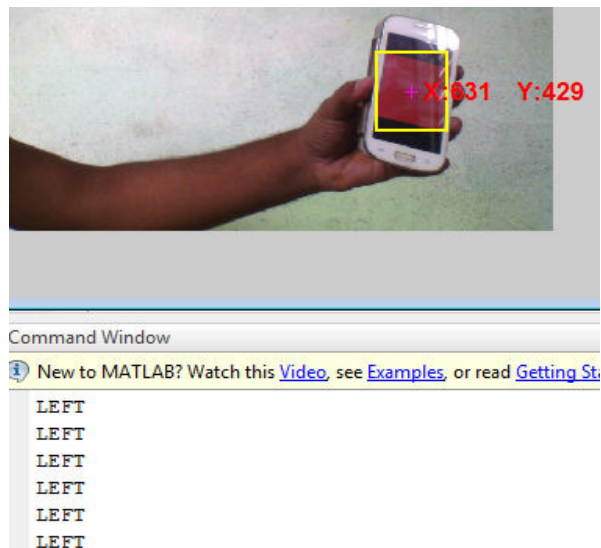
1. The difference or change in centroid of the object between two consecutive snapshots is noted.
2. After recognizing the axis in which the movement is intended, the direction of change is noted.
3. Command corresponding to it is generated and fed to the Arduino thereby to Robot.

RESULT

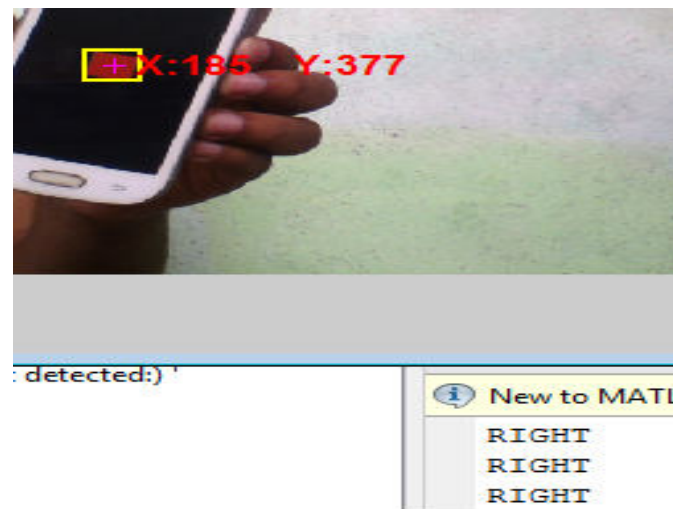
The program for RED object detection and CENTROID calculation have been simulated and following result was obtained.



RESULT

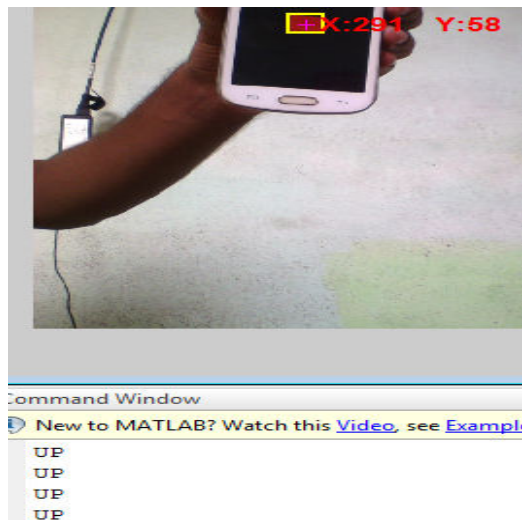


Result at Left Position

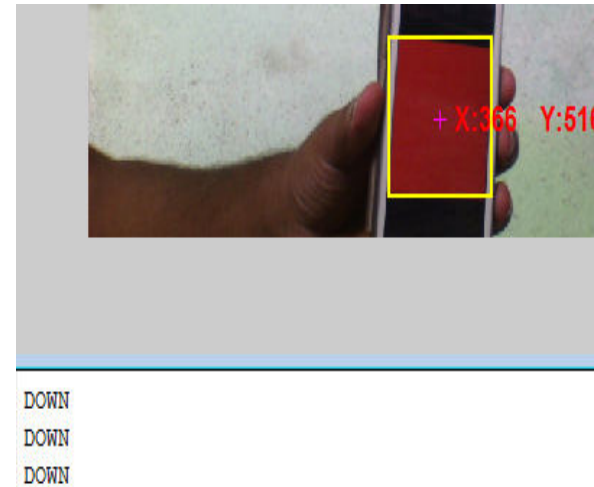


Result at Right Position

RESULT Cont..



Result at Forward Position



Result at Backward Position

Advantages

- The project requires only one time investment of components.
- Cost of project is less expensive and affordable
- Maintenance cost is very low.
- Light weight and can be made portable.

APPLICATION

- 1) To control the *movements of robotic arm* in unmanned regions.
- 2) Object Detection algorithm can be used in *robot vision*.
- 3) In *pick and place robot* for particular color objects.

Future scope

- 1) Further the object detection can be improved by eliminating unwanted detection of color while execution of Bounding Box algorithm except the target object, multiple objects detection.
- 2) The *wireless communication* can be made from the camera to the main controller board.
- 3) The entire project can be burned into a Raspberry Pi so that the need of always carrying a laptop or Personal Computer can be avoided, and project becomes easy to carry.
- 4) The robot can move left, right, front and backwards, but more detailed studies is to be done to rotate it in intermediate angles using the object movements more accurately.
- 5) The idea can be implemented to *control objects at remote places*.

Conclusion:

The main aim of the project was to detect the colored object's movement and how that can be applied to control the movement of the Robot. It also gives us an opportunity to control objects in Remote places.

The project is successfully running in real-time and further scope is added to improve the accuracy of the object detection system.

REFERENCES

- [1] Raquib Buksh, Soumyajit Routh, Parthib Mitra, Subhajit Banik, Abhishek Mallik, Sauvik Das Gupta, "Implementation of MATLAB based object detection technique on Arduino board and iROBOT CREATE", International Journal of Scientific and Research Publications, Volume 4, Issue 1, January 2014.

- [2] Shubham Srivastava, Pratibha Singh, "Real Time Object Tracking Using Color Feature",
International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-8, January 2014

- [3] Region Properties :<http://www.mathworks.in/help/images/ref/regionprops.html>