Hand Gesture Based Surveillance Robot

A thesis submitted in

partial fulfilment of the requirements

for the degree of

Bachelor of Technology

In

Electronics & Instrumentation Engineering

By

Jasmine Sahoo [109EI0341]

Chidananda Tarai [109EI0346]

Department of Electronics & Communication Engineering

National Institute of Technology, Rourkela

Rourkela-769008, Odisha, India

Hand Gesture Based Surveillance Robot

A thesis submitted in

partial fulfilment of the requirements

for the degree of

Bachelor of Technology

In

Electronics & Instrumentation Engineering

By

Jasmine Sahoo [109EI0341]

Chidananda Tarai [109EI0346]

Under the guidance of

Prof. Samit Ari

Department of Electronics & Communication Engineering

National Institute of Technology, Rourkela

Rourkela-769008, Odisha, India

Department of Electronics & Communication Engineering

National Institute of Technology, Rourkela

Rourkela -769008, Odisha, India

CERTIFICATE

This is to certify that the work in the thesis entitled **"Hand Gesture Based Surveillance Robot"** submitted by **Jasmine Sahoo** (Roll No. **109EI0341**) and **Chidananda Tarai** (Roll No. **109EI0346**) in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Electronics and Instrumentation Engineering at National Institute of Technology, Rourkela is an authentic work carried out by them under my supervision. To the best of my knowledge, the matter embodied in the research work has not been submitted to any other University / Institute for the award of any Degree or Diploma.

10th May, 2013
Rourkela

Prof. Samit Ari
Department of Electronics & Communication Engineering
National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

We would like to articulate our deep sense of gratitude and reverence to our supervisor **Prof. Samit Ari**, Department of Electronics & Communication Engineering, National Institute of Technology, Rourkela, for his constant encouragement, continuous monitoring and supervision throughout the period of this research work, for providing us with the department facilities. We are extremely thankful to him for having guided us in shaping the problem statement and providing the insight we required towards plotting the solution.

We would like to express our profound gratitude to **Prof. Lakshi Prasad Roy** for having taught us the basics of image processing in the course named "Digital Image Processing" in such depth and setting up the base of the project that enabled us achieve this success. We would like to thank all our faculty members and staff of the Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela for their revered help throughout the project period.

We would like to conclude while expressing our deepest gratitude to our parents, and all our loved ones. Our full dedication to the work was made possible with their blessings and moral support.

Jasmine Sahoo [109EI0341]

10th May, 2013

Chidananda Tarai [109EI0346]

ABSTRACT

In this work, a hardware and software based integrated system is developed for hand gesture based surveillance robot. The proposed system is a non-invasive technique and software part of the system uses gesture based image processing technique. The hardware part is developed based on AVR microcontroller platform. The captured image of hand is segmented and its contour is determined. The convexity defects are computed to detect the number of fingers used by the subject. The number of fingers directs the path to robot that is to be followed. The camera placed on the robot capture the images of its surrounding, wherever it travels and send it back to the PC for monitoring. In this way, it can be used as a surveillance system. Experimental results show that the overall accuracy obtained above 90% for gesture recognition by which robot will be directed to follow the path. The system can be directly applied to defence grounds for detection of enemy, for spying purpose where the human reach is avoided or not recommended. This unit can be used for overcoming physical handicaps by helping in development of gesture-based wheel chairs, for control of home devices and appliances for persons with physical handicaps and/or elderly users with impaired mobility.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 INTERFACES

This project composed of two types of interfaces:-

- Human – PC Interface
- PC – Robot Interface

1.1.1 Human – PC Interface

Noninvasive techniques for controlling are in high pace with the advancement of technology. A Webcam is connected to the PC. This webcam collects the data in the form of images and compute these data to get information. A person producing different gestures is captured by the webcam and PC process these images using various Image Processing algorithms [1]-[4].The processed images provide us the information what exactly the person want to direct through his gestures. Many works have already been done by Computer Vision experts that include Augmented Reality, Controlling PC mouse events through hand gestures that include selecting, opening, closing files. Both mouse and keyboard can replaced by virtual keyboard and mouse described by Tsang, W.-W.M [5]-[9] which reduces hardware components of a PC.

1.1.2 PC – Robot Interface

Every Hand gesture signifies a pre-defined command .The gestures received by the PC are analyzed by the PC and the command corresponds to it is generated by the computer. In this case computer provides different directions to the robot for its traversal. These commands are first transferred to the microcontroller through serial communication [10]-[11]. Then the microcontroller transfers the data to the Robot through wireless communication using different communication modules.

1.2 CHALLENGES

Real time Image/Video processing becomes a challenging task because it is highly environment dependent. The illumination of light highly affects the processing [12]-[13]. Lack of proper lighting condition, focusing on mobile subjects, interferences in signals causes presence of various type of noise in the image which makes processing not only difficult but also slower.

Presence of complex background makes segmentation, a tiresome task for computation. A Computer having high processing speed is preferable for this purpose. However, now-a- days many stand-alone development boards like Beagle Board, ARM9, ARM11 are available which is compatible for porting Image processing projects on it and make portable Image Processing projects.

These boards are much cheaper than our personal computer but a lot of hands-on exercises are need to be performed to have a deep knowledge about its architecture and functioning before switching to build Image Processing applications on it.

In our project, we have considered our Personal Computer as our processing device and we have developed all our applications which are compatible to Intel Core2duo processor. It is a platform dependent application so it may not work on other processors.

1.3 HARDWARE IMPLEMENTATION

1) 2 Mega Pixels Video Resolution Iball Webcam: For capturing the gestures produced by the user.
2) Wireless Webcam 2.4 GHz : For capturing the images while surveillance
3) ATMEGA 16L: For programming, communication and controlling the motor.
4) USB to Serial Converter: To transmit data serially from the USB port of the PC

5) MAX232 IC Board: It is used for level conversion from RS232 to TTL
6) A434 RF TX/RX Modules : To transmit and receive the data through wireless communication
7) HT 12E and HT12D ICs : For encoding and decoding of data
8) L293D Motor Driver : For driving the motor providing sufficient power
9) 12V / 200 RPM DC Motors: To make the robot move.

1.4 APPLICATIONS

It is seen that there has been rapid developments in the field of robot technology. Robots are being used in a variety of sectors. However, the use of robots must be easier for general users so that the scope of applications expand for intelligent robots in our day-to-day life.

1.4.1 In defence:

Here it is used as a surveillance robot. This unit can be of help and used for surveillance of enemies in an area in defence fields or to spy on the enemies where the human presence is avoided or not recommended. The unit is small in size, easy to handle, portable to move and can reach inconvenient places easily.

1.4.2 In gesture-based wheel chairs:

The unit can be configured onto a wheel chair for differently-abled persons with a camera on the hand rests. Signaling by hand finger count, they can travel easily from place to place. Thus it can be used for overcoming physical handicaps by helping in development of gesture-based wheel chairs, for control of home devices and appliances for persons with physical handicaps and/or elderly users with impaired mobility.

1.4.3 In healthcare environments:

Information is accessed while maintaining total sterility. Touchless interfaces are especially useful in healthcare environments. For patient monitoring, this unit can be used for increasing the efficiency of surveillance.

1.4.4 In industries:

The primary aims of these systems are to provide an automatic interpretation of scenes and to understand and predict the actions and interactions of the observed objects based on the information acquired by sensors. It can detect any unwanted movement and also if detect the absence of wanted supervision at a particular unit by continuous vigilance. It will help to collect the large data. Exploration of large complex data volumes and manipulation of high-quality images through intuitive actions benefit from 3D interaction, rather than constrained traditional 2D methods.

1.4.5 In entertainment field:

Gesture interfaces for gaming based on hand/body gesture technology is designed to achieve social and commercial success.

1.5 COLOR SPACES

Color space can be defined as a set of rules that allows representing colours with a specific range of numbers. A computer always searches for a colour space defined by the characteristics identifiable to that image only while displaying a color in an image or the image itself.

Color is defined as that aspect of a thing which is caused by differing qualities of the light reflected from the thing or emitted by them, identifiable in terms of the observer or with respect to the light. Colour is considered to be the thing that one can see and is able to distinguish with its (human) eyes.

1.5.1 RGB: The **R**ed-**G**reen-**B**lue color space is the simplest and most natural color space available. Each color in this space is represented by a set of 3 numbers. The numbers in this set of three indicate the intensities of the 3 primary colors - red, green, and blue in range from 0% to 100%.

Say the presence in a dark room which has no windows, we cannot see anything. But we know that the walls are painted white. We are having 3 colored flashlights. One red, one green and the third one is blue. The light intensities can be adjusted with a value starting from 0 to a maximum value. Turning on the red flashlight and we see a red circle on the white wall. This comes as surprise. Now we turn the green flashlight on and see a green circle. Then we do adjust the flashlights such that the circles overlap. Now a new color – yellow is seen. After the last flashlight is turned on, it is aimed it at the same spot, the color of the coincident area changes to white. By varying the intensities of the individual flashlights, we are able to get any color virtually.

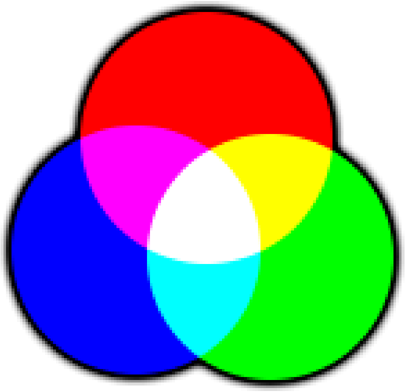


Fig: 1.1

The 3 basic colors having varying intensities are added up to obtain the target color. RGB is, therefore, considered as an additive color space.

RGB color space defined as a natural color space. It was designed to represent the inner working of the human eye. The eye contains three types of cells which are sensitive to the three primary lights. If light rays of all three colors hit human eyes, they appear as white.

RGB is used mostly in LCDs and old CRTs.

1.5.2 YCrCb: Here the color space represents each color also with 3 numbers. The Y component, here, represents the intensity of the light. The Cb and Cr components represent the intensities of the blue and red components of light relative to its green component. This color space makes use of the properties of the human eye. The eye is supposed to be more sensitive to light intensity changes and less to hue changes. When the information amount is to be minimized, the intensity component is stored with higher accuracy content than the Cb and Cr components here. The JPEG file format is making use of this color space to reject unimportant information.

This is one of two important color spaces used for representing digital component video, the other being RGB. The difference between the two is that the former represents color as brightness and two different color signals, while the latter represents color as red, green and blue. In YCbCr model [14], the Y stands for the brightness (luma), Cb is for blue minus luma (B-Y) and Cr for red minus luma (R-Y).

1.6 VISION BASED TECHNOLOGY

In recent years of research, efforts are being done seeking to provide more natural and human-centered means of interaction with computers. A particular direction of importance is that of a perceptive user interface, where the computer is endowed with cognitive capabilities that allow it to acquire information about the user as well as the environment. Vision has got the ability to carry a vast amount of information in a non-invasive method [15]-[16] and at a lower price; therefore it constitutes a very interesting sensing model for developing perceptive user-interfaces. There are proposed approaches for vision-driven interactive user-interfaces referring to technologies such as head tracking, face expression recognition, eye and iris tracking and gesture recognition.

1.7 OpenCV

Fig: 1.2

OpenCV [17] is an open source library mainly used for developing real time image processing applications. It can be freely distributed under the open source BSD license. OpenCV was officially launched by Intel in the year 1999.

Areas including the applications of OpenCV:

- Machine Learning
- Facial Recognition
- Gesture Recognition
- Augmented Reality
- Stereo Vision
- Hidden Language Enhancement
- Tracking Objects
- Object Identification
- Segmentation and Recognition

OpenCV is written in C, C++. It has interfaces with Python, Java and Matlab. In the official site of OpenCV online documentations are available which contains the API for these Interfaces.

In order to encourage the adaptation of this library wrappers for many other language like C#, Ch and Ruby have been developed

OpenCV runs on Windows, Android, Mac, Linux, Blackberry and many more; however setup files are different for different Operating systems.

Chapter 2
Literature Review

2.1 LITERATURE REVIEW

There has been many research works in the field of Hand Gesture based Human Computer Interaction following different algorithms to develop a fast and reliable procedure for gesture recognition. In Paper [18] by Francisco Arce, José Mario García Valdez a three axis accelerometer has been used to read different types of hand gestures. But carrying extra circuitry on the hand involves attaching a number of accelerometers with the hand, this causes irritation to the user, there may be loose connection in the system which may result in abnormal outcomes. In Paper [19] by Anala Pandit, Dhairya Dand , Sisil Mehta , Shashank Sabesan , Ankit Daftery used a combination of accelerometer and gyroscope and the readings are taken into for analyzing the gesture. Here accelerometer is dedicated for collecting translational dynamic and static change in positional vector of hand and infer it to the movement of mouse whereas gyroscope has been used for rotation of virtual object. There are many papers where gestures are being analyzed using color gloves [20]. A data glove is a type of glove that contains fiber optics sensor embedded in it to recognize the fingers movement.

Hand gesture recognition using image processing algorithms many time involve use of color gloves . By tracking this color glove different hand gestures can be interpreted as described by Luigi Lamberti1 and Francesco Camastra in their paper [21].Here they have modeled a color classifier performed by Learning Vector Quantization.

In paper [22] by J.S. Kim, C.S. Lee, K.J. Song, B. Min, Z. Bien, a pattern recognizing algorithm has been used to study the features of hand. There are many papers where training of hands using a large database of near about 5000-10000 positive and negative images are considered. But this procedure is very tiring and time taking. But the proposed algorithm of our project is a fast method of gesture recognition which neither needs neither any extra circuitry nor it requires any color gloves.

CHAPTER 3

Hand segmentation and finger counting

3.1 CAPTURING THE IMAGE THROUGH THE WEBCAM

Fig: 3.1

Presently Laptops are available with high quality webcams embedded in it. We have used iball external USB webcam for capturing better quality of image .Its better if we get a camera having night vision system in it for having proper view and calibration of the image. By varying the light focused on the subject varies its color components so proper care must be taken while fixing the color model and values. The movement of hand is hence recorded by here presented camera. This input video will be decomposed into multiple static images and is captured into multiple frames.

3.2 SEGMENTATION OF HAND, REMOVAL OF BACKGROUND AND NOISE

Obtained color image is a three channel image composed of R [0-255], G [0-255], and B [0-255]. This colored image is first converted into YCrCb model using function available in OpenCV library; refer OpenCV online documentation or Learning OpenCV by Gary Bradski and Adrian Kaehler [17]. This color space allocates each color with a set of 3 numbers, similar to RGB space.

Fig: 3.2

The Y component denoting the intensity of the light. The Cb and Cr components here are the intensities of the blue and red components ratio to the green component.

This color space makes use of human eye properties. The eye is likely to be more sensitive to light intensity variation and less to hue variations.

To minimize the amount of information, the intensity component is stored with the Cb and Cr components. JPEG format use this method to remove unrequited in format.

The values of skin color can be got easily through various databases available in the internet. In the program it has used the following range of values for thresholding the skin portion on the hand [16]:

Y: 20 -250

Cb: 140-210

Cr: 90-135

In Thresholding, all the values present in this range are brightened while every pixels not lying out of this range are removed by darkening. This converted the image from three channels to single channel. To increase invariance against illumination [12] variability some methods operate in the HSV, YCrCb or YUV color spaces, in order to approximate the "chromaticity" of skin (that is look into its absorption spectrum) rather than just looking into its apparent color value. They eliminate the luminance component first, thus to be removing the effect of shadows, now for illumination changes, as for modulations of orientation of hand's skin surface with relation to the light source(s). To reduce noise different variety of morphological operations are carried out like smoothening, dilating, erosion. In our algorithm, Gaussian smoothing is carried out.

Fig: 3.3

The Gaussian smoothing operator is applied to blur the images. It removes noises and detail present in intensity variation form. It is like a mean filter but the kernel taken here is quite different. It is like the shape of Gaussian function bell-shaped hump.

3.3 CONTOUR FINDING OF THE HAND

Contour is nothing but boundary of a blob detected in a single channel image. Here the single channel image consists of the segmented hand, removed from the background. Many objects having color components values similar to the hand present within the frame also get segmented. Many noises get added with original image and increase the presence of unnecessary components in the image. All these non-required objects cause formation of unnecessary contours which are filtered out. The area of the contours formed due to the noise is very small comparative to other so we reject those contours [13] which are very small in area. We assume that the contour of the hand covers maximum area so we will collect contours frame by frame, find the contour of maximum area and store it for further processing.

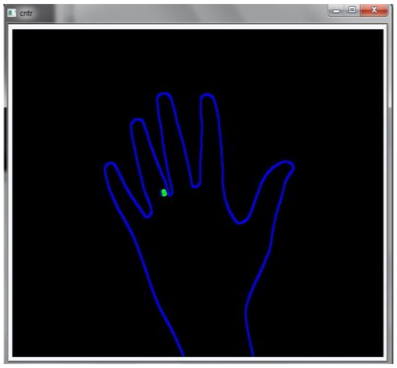


Fig: 3.4

Theo Pavlidis' Algorithm is one of the more recent contour tracing algorithms and was proposed by Theo Pavlidis [23]. Pavlidis chooses to trace the contour in a counterclockwise direction. Here we are taking a digital pattern, a set of black pixels on a sheet of white pixels which makes the grid here; a black pixel is found and it is declared as being the "start" pixel. Locating it is done mostly at the bottom left corner of the grid. That pixel is called the "start" pixel.

Let us assume that we have located the start pixel facing frontwards.

There will be three pixels P1, P2 and P3 in the front direction.

Case-I:

Checking is done if P1 is black, if yes, P1 is declared as the current position and one step is moved forward followed by left to land on P1.

Case-II:

If P1 is white, proceed to P2.

If P2 is black, then P2 is declared as current boundary pixel and one step is moved forward to land on P2.

Case-III:

If both P1 and P2 are both white, then proceed to P3.

If P3 is black, P3 declared as current boundary pixel and one step moved to right then to left. If none of the three in front are black, the facing is rotated 90° clockwise and new three pixels are taken.

The process terminates when starting point is reached again [23].

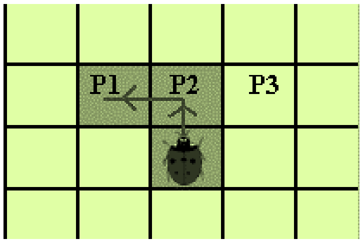


Fig: 3.5.a

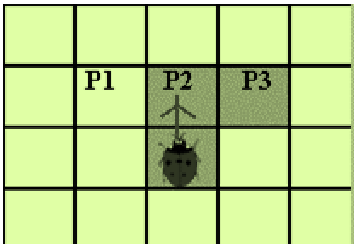


Fig: 3.5.b

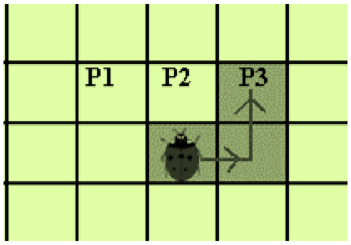


Fig: 3.5.c [17]

3.4 CONVEX HULL

Convex hull [17] is the smallest polygon made with a set of points such that all the points are either inside the or on the polygon mad.

Mathematically , Convex hull of a set A of points where all points are located in an Euclidean space is the smallest set containing A. Convex hull can be assumed to be a rubber band containing the set A.

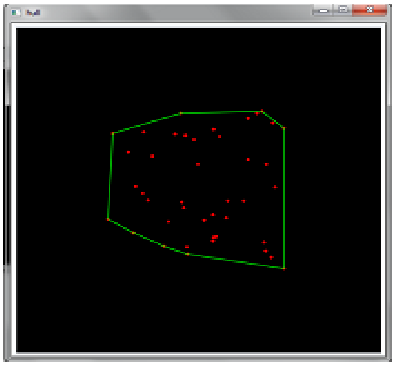


Fig: 3.6

Here the Figure describes that the set contains thirteen points i.e. P0 to P12. Convex Hull is formed by connecting the points P0, P1, P3, P1 and P12.

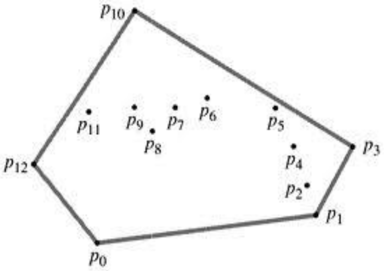


Fig: 3.7

All rest points remain within the polygon. Here the points forming the contour of the hand is taken into account. So the convex hull is formed by the lines joining each finger tip of the hand.

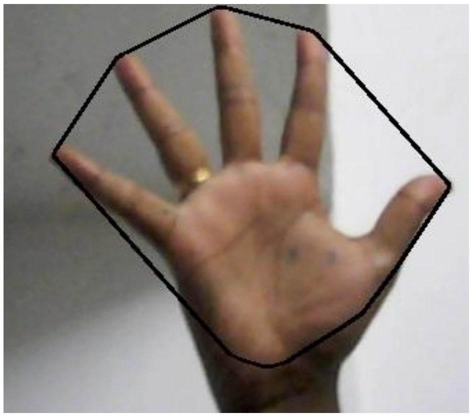


Fig: 3.8

3.5 CONVEXITY DEFECTS

This convex hull is applied on the hand contour causing defects at different region around the hand. Defects are the void region surrounded by the fingers of the hand and the convex hull formed by joining the finger tips of the hand.

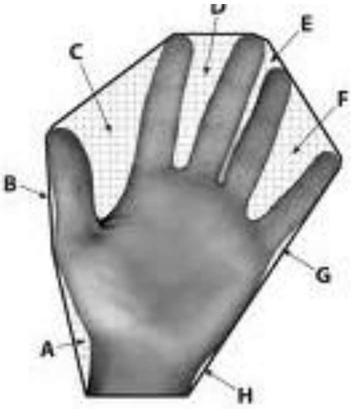


Fig: 3.9 [1]

A, B, C, D, E, F, G, H represents the convex defects of the hand formed by the convex hull. Convex defects finding function of OpenCV library provides three important parameters:

- Starting point of defect
- End point of defect
- Depth of the defect

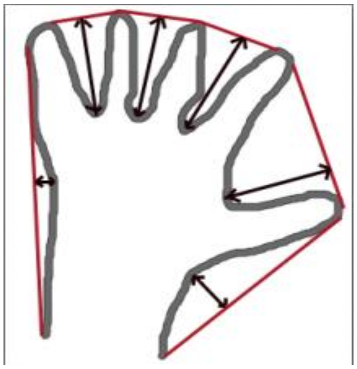


Fig: 3.10

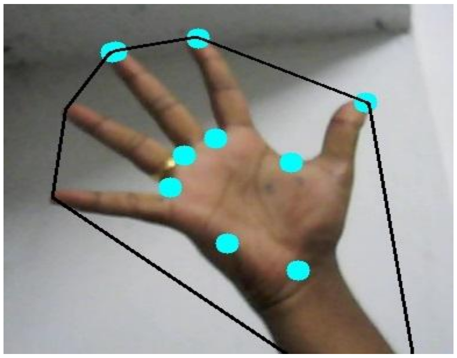


Fig: 3.11

Defects: C, D, E and F are formed in between the fingers. The defect depths of these defects are usually much large compared to other defects (A, B, G, and H). We need to consider the defects present in between the fingers since they provide the information about the number of fingers shown. The depths of the defects formed in between the fingers are noted. The least depth is taken as a threshold value to detect the actual depth in the real time. Defects present in the region A, B, G, H have depth of the defects much less than the threshold value hence rejected.

So the high depth defects are considered rejecting all other defects.

Through this formula, number of fingers shown by the user is counted.

Number of Fingers=Number of Defects+1

Then the computer searched for the command corresponds to it and send the commands to the microcontroller through serial communication.

CHAPTER 4

Communication: Between Robot and PC

4.1 INTRODUCTION TO AVR MICROCONTROLLER

ATMEGA 16L of AVR (Advanced Virtual RISC) family is used for the project. The point of choosing this microcontroller is it is cheap and all the resources are available as open source applications.

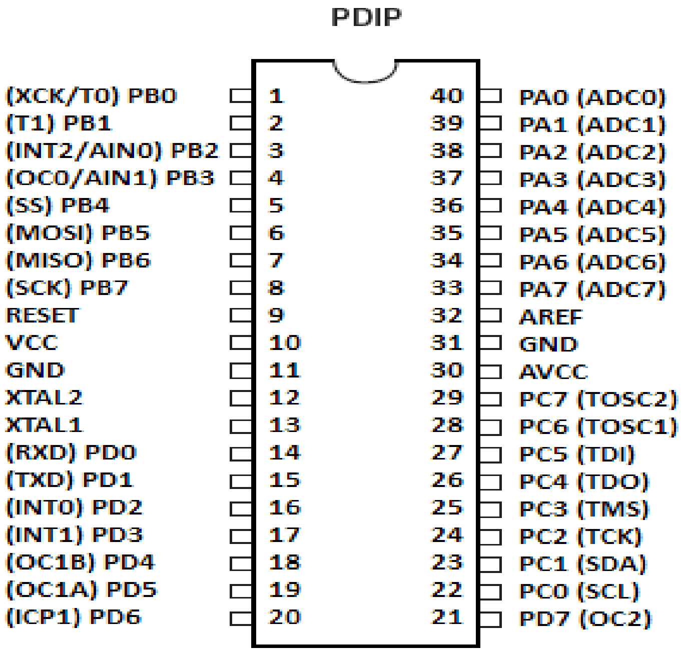


Fig: 4.1

ATmega32 & ATmega32L:

AtMega32 is available in varieties .The '**L**' denotes **Low Voltage** device. Operating Voltages is (**2.7 - 5.5V**) for ATmega32L (**4.5 - 5.5V**) for ATmega32. I prefer '**L**' because it gives me more flexibility to use a single lithium battery (3.7V) to power the microcontroller **8-bit microcontroller**: It means the processor's ALU can handle 8-bits of data at a time.

• **32K Bytes In-System Programmable Flash:** It serves as storage for the programs that are being written by the programmer.

• **2K Byte Internal SRAM**: It has 2Kb Static RAM

• **Four PWM Channels**: four different pins which can independently produce PWM. PWM stands for Pulse Width Modulation which is used for producing variable speed in a DC motor.

• **8-channel, 10-bit ADC**: Normally the input/output pins deal with Logic1/Logic0 (digital data) but the PORTA could be configured to receive analog data. There are inbuilt 8 different channels of Analog to Digital Converter inside the microcontroller

• **Internal Calibrated RC Oscillator**: All the microcontrollers need clock pulse to process instruction. This clock pulse defines the speed of the processor. The maximum speed of AVRs is **16 MHz** There is an inbuilt RC Oscillator which supplies the clock pulses. The internal RC Oscillator can give a maximum speed of 8 MHz but initial factory setting is 1MHz. AtMega32 has four 8-bit (I/O) PORTS which act as general purpose Input Output Port.

• **Pin No (**1-8): **PORTB**

• **Pin No (**9): **Reset**: On applying ground (**logic0**) to this port restarts the microcontroller. It means the program that you have written starts all over again if **logic0** is given to it

• **Pin No (**10): **VCC**: This is usually **5 Volts for AtMega32**, but for **AtMega32L** it is as low as 2.7 Volts Because of this I can supply **AtMega32L** with **3.7 Volts** directly from Li-ion battery (Rather than using a IC 7805/LM317 and a 6-Volt Lead-Acid battery). The **VCC** along with the **GND** forms the **power supply**, which provides current to drive the microcontroller.

• **Pin No (11): GND**: The reference/ ground is connected to it.

• **Pin No (12-13): XTAL2-XTAL1**: Here external crystal oscillator is connected. The crystal oscillator provides clock cycle for the system to run.

• **Pin No (14-21):** PORT D

• **Pin No (22-29):** PORTC

• **Pin No (30): AVCC**: This along with the GND forms the **power supply** for the **ADC** (Analog to Digital Comparator). Whenever we will use the ADC only then we shall give this 5 volts or 3.7 volts for AtMega16L

• **Pin No (31): GND**: This ground is connected to the other ground internally but it is advisable to externally them too, while using the **ADC** we must connect this **GND** to the system ground.

• **Pin No (32): AREF**: This is actually the reference voltage required for **Analog to Digital Conversion**

• **Pin No (33-40): PORTA**

Apart from the normal port function the other important functions are listed below.

• **Pin No (2)**: **INT2**: This pin can be configured to receive interrupt

• **Pin No (3)**: **OC0**: At this pin 8-bit PWM (Pulse Width Modulation) can be produced

• **Pin No (6-11)**: **MOSI, MISO, SCK, RESET, VCC, and GND**: All these pins are together used to program the microcontroller by connecting to the suitable pins of the parallel ports.

• **Pin No (14-15)**: **RXD, TXD:** These two pins can be configured for serial data communication with another microcontroller, a computer, or any suitable device supporting serial data communication

• **Pin No (16-17)**: **INT0, INT1**: This pin can be configured to receive interrupt. As you can note there is only three interrupt pins (INT0, INT1, and INT2) available on AtMega16.

• **Pin No (18-19)**: **OC1A, OC1B**: At these pins 10-bit PWM (Pulse Width Modulation) can be produced

• **Pin No (21)**: **OC2**: At this pin 8-bit PWM (Pulse Width Modulation) can be produced. Now you can note there are **two 8-bit PWM pins (OC0, OC2)** and **two 16-bit PWM pins (OC1A, OC1B).**

• **Pin No (33-40)**: **8-ADC Channels**: If configured PORTA acts as 8-channel ADC. It receives analog Voltage and converts them into **10-Bit/8-**bits **binary numbers**

The **PORT's** function is to communicate outside the microcontroller. These ports take input and produce output, whatever they are configured as (input/output). Except the ADC operation in all other cases there exists only two levels **High/Low** i.e. **logic 1/logic 0**. Any voltage above 1.5 Volts is **logic 1** and voltage below 0.7 Volts is **logic 0.** Voltage in-between leads to ambiguous results.

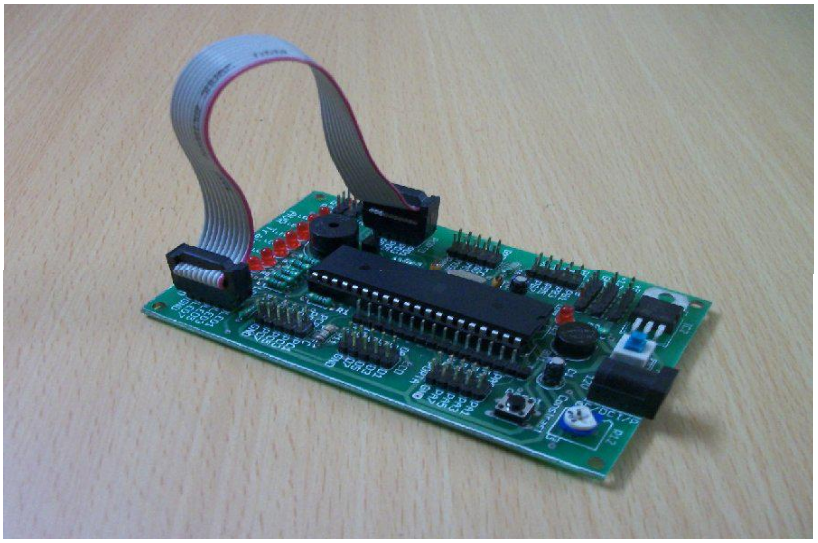


Fig: 4.2 [11]

4.2 PROGRAMMING USING AVR MICROCONTROLLER

For General Purpose I/0 programming 3 types of registers are used:

- DDR : Data Direction Register
- PIN
- PORT

DDR is used to configure the pins of a port in input or output mode according to our requirements.

PIN register is used to get the input values from the external sensors

PORT register is used to provide signals to the actuators connected to the microcontroller.

4.3 MOTOR DRIVER

On giving a LOGIC 1 at INPUT 1, INPUT 2, INPUT 3, INPUT 4 we obtain VC at OUTPUT 1, OUTPUT 2, OUTPUT 3, and OUTPUT 4.

On giving LOGIC 0 at INPUT 1, INPUT 2, INPUT 3, INPUT 4 we obtain GND at OUTPUT 1, OUTPUT 2, OUTPUT 3, and OUTPUT 4.

So using various combination of (INPUT 1 and INPUT 2) or (INPUT 3 and INPUT 4) we do different operation on the motor. When we give CHIP INHIBIT 1 LOGIC 1 then the CHANNEL 1 produces output if LOGIC 0 is given it doesn't produce any output, same goes for CHANNEL 2. So when connecting the circuit connect pin 1 and pin 9 to VSS.

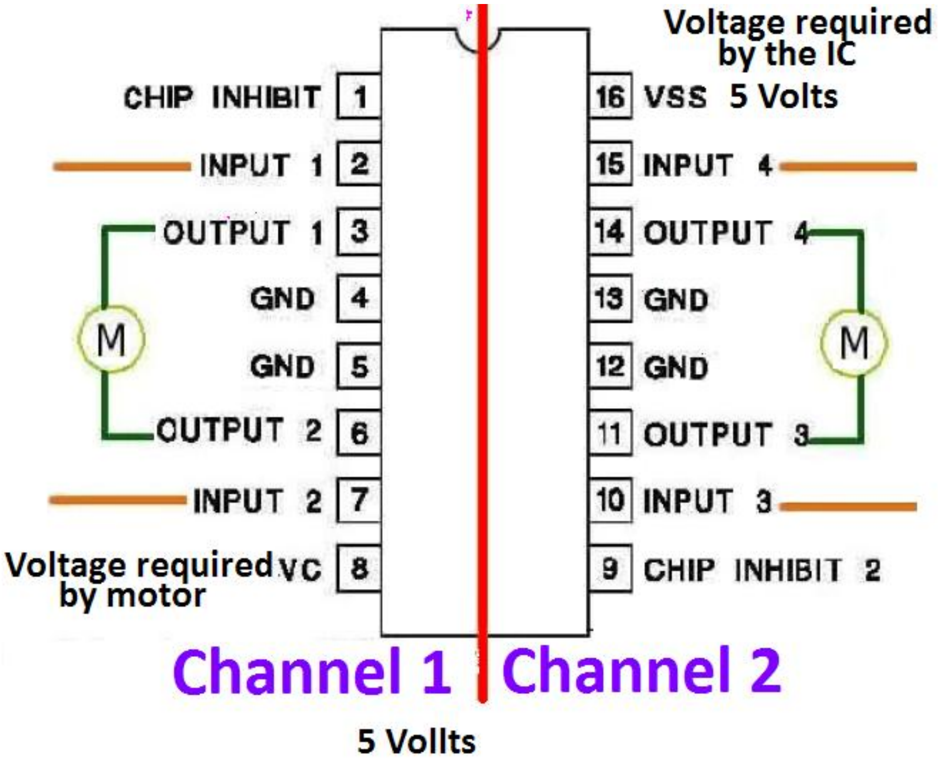


Fig: 4.3

Tabulation of Motor Response:

I1	I2	A	B	Motion
0	0	0	0	STOP
1	0	VCC	0	Clockwise
0	1	0	VCC	Anti Clockwise
1	1	VCC	VCC	Brake

Table: 4.1

4.4 INTRODUCTION TO RS232 LOGIC LEVEL

Serial Communication:

Serial Communication is the method adopted in which data, say; one byte of data is transmitted one bit at a time where as in parallel communication, whole data is to be transmitted at a time i.e. all eight bits are transmitted through 8 data lines at a time.

In serial communication, only one wire is used for transmission of information between two devices. The information data is in the form of logic levels which changes levels according to the bits being transferred i.e. 1 or 0. Serial communication, generally, follows two protocols:

USART: Universal Synchronous Asynchronous Receiver Transmitter

UART: Universal Asynchronous Receiver Transmitter

In USART, a “clock line” is used for synchronizing the communication between the two devices. It is usually used for shorter distance’s communication. This type of communication is followed by serial buses like SPI and I2C.

In USART, there is a start bit and also a stop bit which acts as an important tool for synchronization in communication between the two devices serially.

4.4.1 RS232:

RS232 is a synchronous serial communication protocol used widely in computers and also digital systems. There is no separate synchronizing clock signal as there is in other serial protocols like SPI or I2C. The protocol is arranged in such a way that it automatically synchronizes itself. We use RS232 to create a data link between our microcontrollers based projects and standard personal computer.

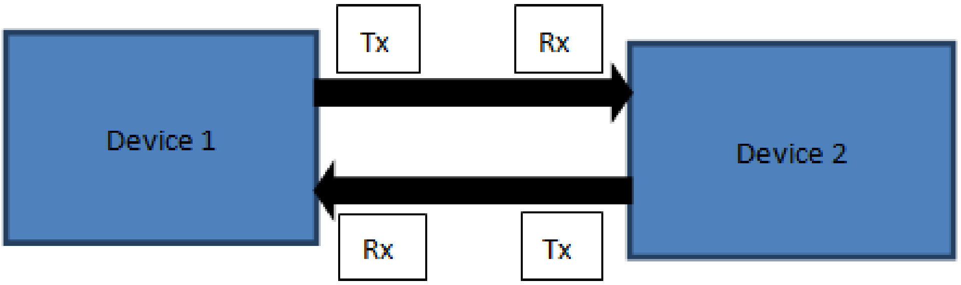


Fig: 4.4

RS232 follow asynchronous serial communication protocol. RS232 is highly used in computers and other electronics system. This protocol automatically synchronizes itself so there is no requirement of separate clock signal as in SPI and I2C. We use RS232 for data transfer between the two.

In RS232 there exist two data lines RX and TX. TX is the wire through which data is sent to other device. RX is the line through which other device put the data it is required to send to the device.

We have: HIGH= +5 V, LOW= 0 V (in TTL / MCU circuits) But whereas in RS232: HIGH= -12 V, LOW= +12 V. There is no “clock line” for synchronization. But accurate timing is required for transmissions to be carried out with standard speeds only. The speed is measured in bits per second (bps). Number of bits transmitted herein is also known as the baud rate. Some standard baud rates are as follows: 1200, 2400, the common ones as 4800, 9600 and other rates as 19200, 38400, 57600, 115200, etc. For transmission a single byte, one needs to have two extra bits: the START BIT and the STOP BIT. Thus for sending a byte, a total number of ten bits are required, we send 960 bytes per second. It is to be noted that the number of stop bits can be one or two. There also exists one more bit; the parity bit, seeing into simplicity we are using it.

4.5 DATA TRANSMISSION THROUGH RS232

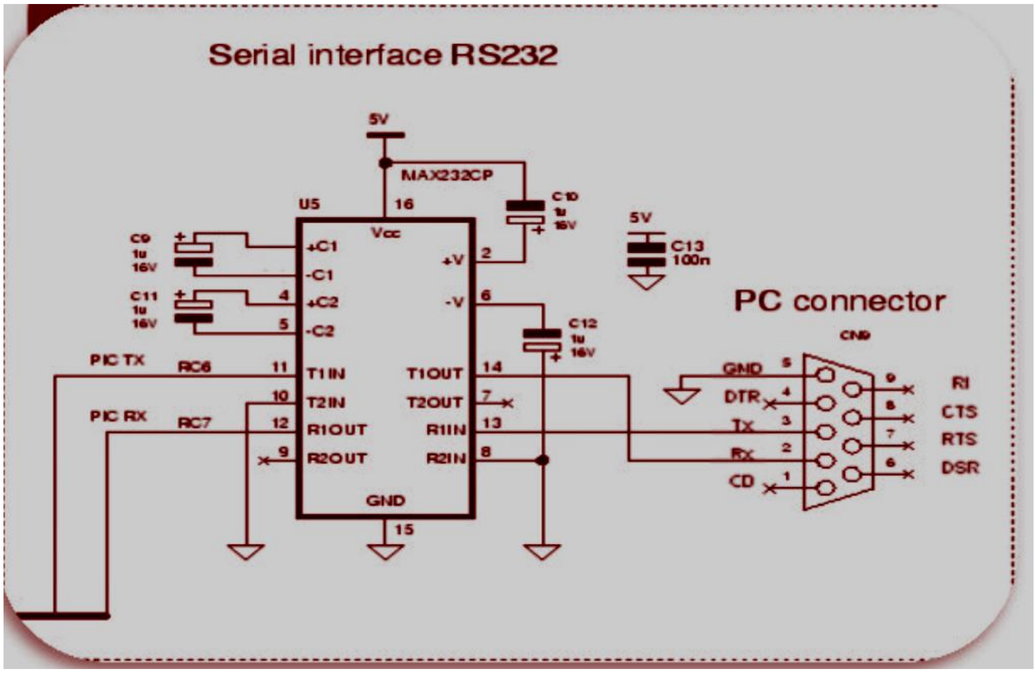


Fig: 4.5 [11]

The data transfer can be done in following two methods:

Transmission:-

1. Whenever no transmission, the TX line is set to HIGH (STOP CONDITION).
2. When the device has to send data, it drops the TX line low for 104us (start bit 0).
3. Then it is sending each bit with duration = 104us.
4. At the end, it gets TX line to HIGH for at least 104us (stop bit 1)

Reception:

1. The receiving instrument waits for the start bit i.e. the RX line goes LOW (+12 V).
2. When it has start bit, it waits half bit time i.e. 104/2 = 51 us. In middle of start bit, it reads it again to see if it is a valid start bit and not a spike.
3. It waits for 104us, in middle of first bit; it reads the value of RX line.
4. Similarly,, it reads all the 8 bits.
5. The receiver has now got the data.

The interfacing between RS232 signals and MCUs is done by using a Level Converter. The level converter job is to convert the RS232 level signals which is (HIGH = -12V LOW = +12V) from PC to the TTL level signal which is (HIGH = +5V LOW = 0V) to be the input to MCU and also the opposite. RS232 is such a common protocol. There is a dedicated and specific IC designed for this “Level Conversion”. This IC is called MAX232 from Maxim. By making use of charge pumps, it generates high voltages of (12V) and negative voltages of (-12V).

4.6 USART OF AVR MICROCONTROLLER

The AVR's USART will be connected to the processing unit by the six registers listed as follows.

- UDR - USART Data Register: Actually this is not one but two register but when you read it you will get the data stored in receive buffer and when you write data to it goes into the transmitter's buffer. This is important to remember it.

- UCSRA - USART Control and status Register A: As the name suggests it is used to configure the USART and it also stores some amount of status of the USART. There are two more in use of similar types: one is the UCSRB and the other is UCSRC.

- UBRRH and UBRRH: Here is presented the USART Register for Baud rate, 16 bits in width so UBRRH will be the High Byte and UBRRL will be the Low byte. But in C language it is directly available for use as UBRR and compiler is configured to manage the 16BIT access.

UCSRA: This stands for USART Control and Status Register A.

- RXC: This is the bit which is set when the USART has completed a byte reception from the host (PC) and the program reads it from UDR
- TXC This bit is set (1) when the USART has completed transmitting a byte to the host and your program can write new data to USART via UDR
- RXCIE: Receive Complete Interrupt Enable - When this bit is written one the the associated interrupt is enabled
- TXCIE: Transmit Complete Interrupt Enable - When this bit is written one the the associated interrupt is enabled.

- RXEN: Receiver Enable - When you write this bit to 1 the USART receiver is enabled. The normal function of the port with RX pin is overridden. So the linked I/O pin now is switched to its secondary functionality, i.e. RX in there for USART.
    - TXEN: Transmitter Enable
    - UCSZ2: USART Character Size
    - URSEL: USART register select.
    - UMSEL: USART Mode Select - This bit here selects the mode: asynchronous and synchronous. Asynchronous mode is more effective with USART.
    - USBS: USART Stop Bit Select - This bit selects the number of stop bits in the data transfer.
    - UCSZ: USART Character size - These three bits (one in UCSRB) selects the number of bits of data that is being transmitted per frame. The unit of data in MCU is 8BIT (C type "char") and is most widely used, so preferred here.
    - UBRR: USART Baud Rate Register: This is the USART Baud rate register, it is 16BIT in width so UBRRH has to be the High Byte and UBRRL will be the Low byte. But in C language it is available as UBRR and compiler has the 16BIT access. This register is made useful by the USART to generate the transmission of data at specified speed (let's say 9600Bps).

$$\text{UBRR}= f_{osc} /(16 \text{ X baud rate}) – 1$$

4.7 WIRELESS Module :

Transmitter circuit

The transmitter circuit is made up of HT12E (encoder), which packages the data to be sent with an address. A wireless module named A434 transmit the packet through a mechanism called Amplitude Shift Keying. Data is sent in 4 bits format.

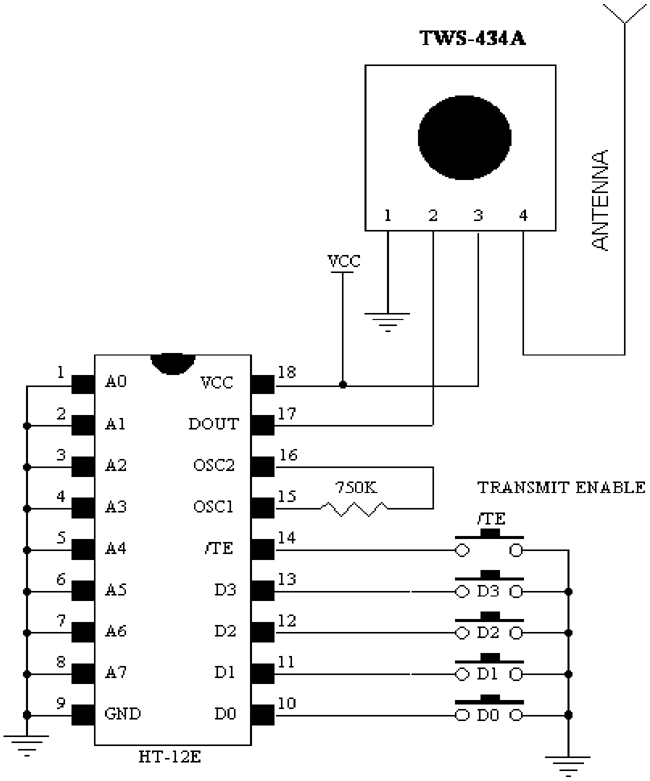


Fig: 4.6 [5]

Receiver Circuit

Receiver circuit is made up of HT12D IC. If the address of the data transmitted matches with the address of the decoder then decoding is being done. The four bits received by the receiver is

provided to the motor driver to run the motor.

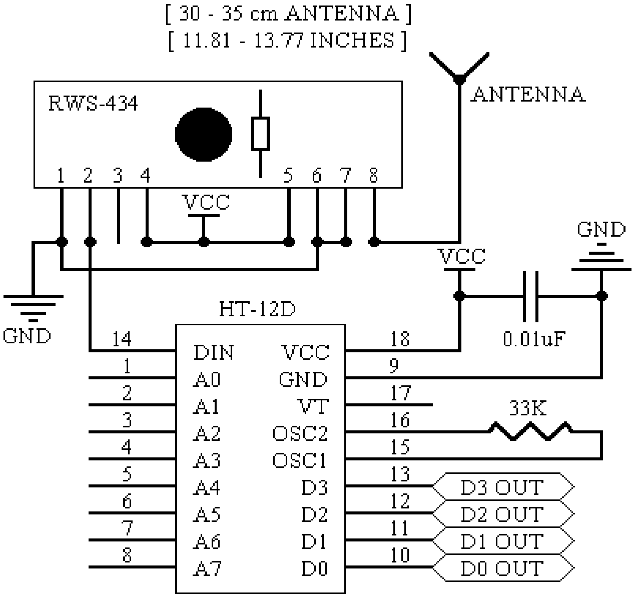


Fig: 4.7 [5]

CHAPTER 5

Results and Discussions

5.1 EXPERIMENTAL RESULTS

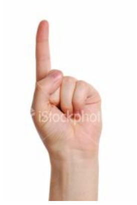
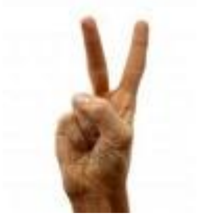
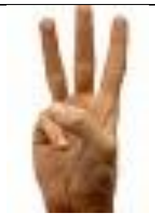
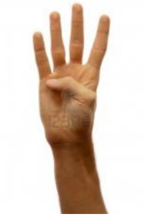
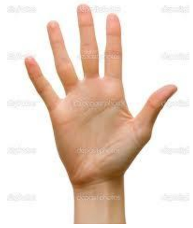
Hand Posture	No. of defects	Finger count	Robot Actualization	Observation
	0	1	⊘	Stop
	1	2	↑	Go Straight
	2	3	↰	Turns Left
	3	4	↱	Turns Right
	4	5	↓	Go backward

Table: 5.1

5.2 OBSERVATIONS

Observation	**No. of trials**	**No. of successful outcomes**	**Accuracy (%)**
Go Straight(0)	30	29	96.67
Go straight(1)	30	29	96.67
Turn left	30	26	86.67
Turn right	30	27	90.00
Go Backward	30	28	93.33
Stop	30	28	93.33
		Overall accuracy	92.78

Table: 5.2

5.3 SURVEILLANCE ROBOT

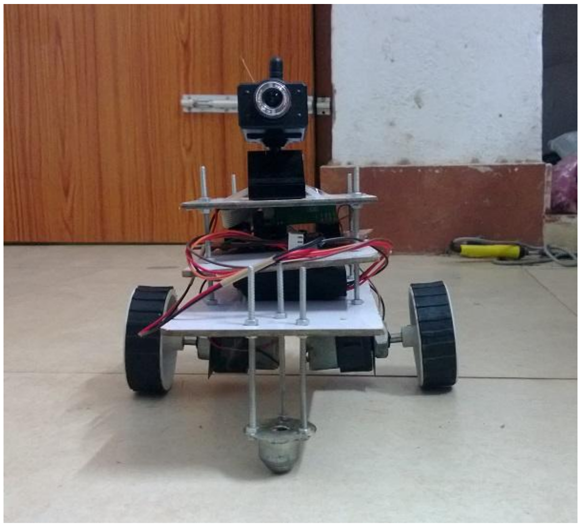


Fig: 5.1

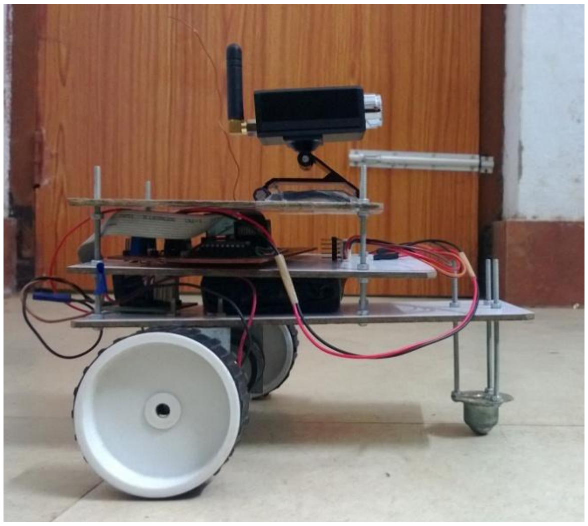


Fig: 5.2

These two above pictures show the model of surveillance robot. The system is designed for maneuvering the places, avoided to access directly.

5.4 IMAGES RETRIEVAL FROM THE SURVEILLANCE ROBOT

The wireless camera placed on the surveillance robot captures the image of its surroundings and sends the information back to the PC for monitoring.

Fig: 5.3

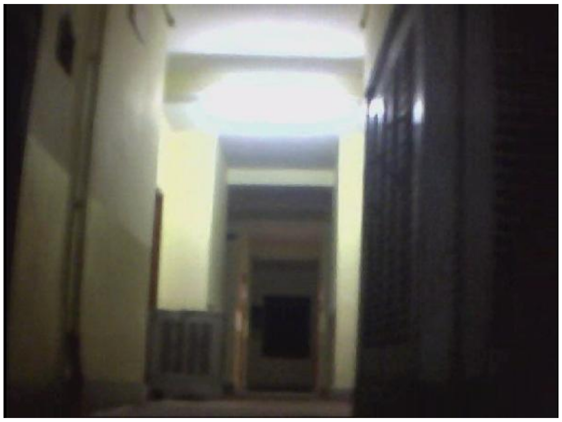


Fig: 5.4

CHAPTER 6
Conclusions

6.1 CONCLUSIONS

This project proposed work here is an authoring method capable of creating and operation controlling motions of surveillance and industrial robots based on hand-robot interaction. The proposed methodology adopted is user- friendly and eliminates intrusion and facilitates motion control of robots using finger signals, which is supplement only to language in the sense of means of communication. The proposed robot motion controlling method is expected to provide effective and implementable solutions for not only just industrial robots, but also for higher intelligence embedding robots like humanoids.

In this research work, we have presented a real-time algorithm to track and recognize hand gestures for human-computer interaction with the surveillance robot in context. It has been proposed an algorithm based on hand segmentation, then its tracking and posture recognition from extracted features of hand by making use of the concept of convexity hull. The system's performance evaluation results have concluded that this low-cost interface can be used by the researchers or industrial workers to substitute traditional machinery. The experiments have confirmed that with environmental factors in control it results in higher efficiency and, thus, better product development and application oriented concept.

6.2 KEY FEATURES OBSERVED IN THE PROJECT

1. Effective in implementation.

2. Low power consumption and compact size.

3. Long control range due the usage of RF devices.

4. Robot monitored from a remote area (no need of 'line-of-sight' arrangement).

5. Maneuvering its path avoiding obstacles by own.

6. Land mine detection.

7. Wireless video surveillance.

REFERENCES

[1] Rafael C. Gonzalez, *Digital Image Processing*, Pearson Education, 3rd ed., 2008.

[2] C. Wang, and K. Wang, "Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction" Recent Progress in Robotics, LNCIS, vol. 370, pp. 317-329, 2008.

[3] A. Malima, E. Özgür, and M. Çetin, "A fast algorithm for vision-based hand gesture Recognition for robot control," in *Signal Processing and communication application,* 17-19 April 2006, pp. 1-4.

[4] P. Garg, N. Aggarwal and S. Sofat, "Vision Based Hand Gesture Recognition," World Academy of Science Engineering and Technology, vol. 25, pp. 972-977, 2009.

[5] F. Quek, "Towards a Vision Based Hand Gesture Interface," in *Proceedings of Virtual Reality Software and Technology,* Singapore, 1994, pp. 17-31.

[6] A. A. Argyros and M. I. A. Lourakis, "Vision-based interpretation of hand gestures for remote control of a computer mouse," in *ECCV Workshop on HCI*, Graz, Austria, May, 2006, pp. 40-51.

[7] C. A. Burande, R. M. Tugnayat and N. K. Choudhary, "Advanced recognition techniques for human computer interaction," *Computer and Automation Engineering (ICCAE),The 2nd International Conference,* vol. 2, pp. 480-483, 2010

[8] J. Lee and T. Kunii, "Model-based analysis of hand posture," *IEEE Comput. Graph. Appl.,* vol 15. no. 5, pp. 77-86, 1995

[9] Tsang, W.-W.M. ," A finger-tracking virtual mouse realized in an embedded system," *IEEE Intelligent Signal Processing and Communication Systems(ISPACS)*, 2005,pp. 781-784

[10] www.alldatasheet.com

[11] http://extremeelectronics.co.in http://www.ablab.in/

[12] Dr. John W. V. Miller, Malayappan Shridhar," Illumination-Invariant Image Processing," *University of Michigan-Dearborn, Dearborn, MI,* 48128, USA,pp 543-564,2012

[13] Zhai Ming, Fu Shan," Noise Reduction for Tiny Contours in Image Sequence" in *12th International Conference on Information Fusion Seattle, WA, USA*, pp 1310-1316 , July 6-9, 2009

[14] B. D. Zarit, B. J. Super and F. K. H. Quek, "Comparison of five color models in skin pixel classification," in *ICCV International Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems*, 1999, pp. 58- 63

[15] G. Kukharev, A. Novosielski, "Visitoridentification elaborating real time face recognition system", *In Proc. 12th Winter Schoolon Computer Graphics (WSCG), Plzen, Czech Republic*, pp. 157 – 164, Feb. 2004.

[16] D. Chai, and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. on Circuits and Systems for Video Technology*, 9(4): 551-564, June 1999.

[17] G. Bradski, and A. Kaehler, *Learning OpenCV*, OReilly, 1st ed., September 2008.

[18] F. Arce, J. M. G. Valdez," Accelerometer-Based Hand Gesture Recognition Using Artificial Neural Networks" in *Soft Computing for Intelligent Control and Mobile Robotics Studies in Computational Intelligence*, vol. 318, pp 67-77, 2011

[19] A. Pandit , D. Dand , S. Mehta , S. Sabesan , A. Daftery," A Simple Wearable Hand Gesture Recognition Device using iMEMS," *International Conference of Soft Computing and Pattern Recognition* , pp 592-597,2009

[20] R. Wang, J. Popovic, "Real-time hand-tracking with a color glove," *ACM Transactions on Graphics*, vol. 28, pp 461-482 , 2009

[21] "Luigi Lamberti1 and Francesco Camastra", Real-Time Hand Gesture Recognition using a Color Glove," *Department of Applied Science*, *University of Naples Parthenope*, 2010, pp.321-344

[22] J.S. Kim, C.S. Lee, K.J. Song, B. Min, Z. Bien, "Real-time hand gesture recognition for avatar motion control," *Proceedings of HCI'97*, pp. 96-101,February 1997

[23] T. Pavlidis., *Algorithms for graphics and image processing*, Computer Science Press, Rockville, Md, 1982.