# HAND GESTURE CONTROLLED ROBOT

Project report submitted in partial fulfillment of the requirement

for the degree of

Bachelor of Technology

in

**Electronics and Communication Engineering**

Batch: 2011-2015

Under the Supervision of

**Mr. Munish Sood**

By

**Venus Garg (111045)**

**Mohit Patiyal (111048)**

to

Jaypee University of Information Technology,

Waknaghat, Solan – 173234, Himachal Pradesh

# Certificate

This is to certify that project report entitled “Hand Gesture Controlled Robot”, submitted by Venus Garg(111045) and Mohit Patiyal(111048) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: Supervisor’s Name: Mr. Munish Sood

Designation: Assistant Professor

# Acknowledgement

We are very grateful and highly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide Mr. Munish Sood, Department of electronics and Communication Engineering, Jaypee University of Information Technology Waknaghat, in bringing this report to a successful completion.

We are grateful to Prof. Sunil Bhooshan, Head of the Department of Electronics and Communication Engineering, for permitting us to make the use of the facilities available in the department to carry out the project successfully. Last but not the least we express our sincere thanks to all of our teachers and friends who have patiently extended all sorts of help for accomplishing this undertaking.

Date: Venus Garg(111045)

Mohit Patiyal(111048)

# Table of Contents

# List of Figures

# List of Tables

# Abstract

Wireless Communication has announced its arrival on big stage and the world is going mobile. We want to control everything and without moving an inch. The remote control of appliances is possible through Embedded Systems.

The main aim of this project will be to design a robot which can be controlled by simple gestures. The user just needs to wear a gesture device which includes a sensor. The sensor will record the movement of hand in a specific direction which will result in the movement of the robot in the respective direction. The robot and the Gesture device are connected wirelessly via radio waves. The wireless communication enables the user to interact with the robot in a more friendly way.

# CHAPTER 1: INTRODUCTION

Recently, strong efforts have been carried out to develop intelligent and natural interfaces between users and computer based systems based on human gestures. Gestures provide an intuitive interface to both human and computer. Thus, such gesture-based interfaces can not only substitute the common interface devices, but can also be exploited to extend their functionality.

## 1.1 ROBOT

A robot is usually an electro-mechanical machine that can perform tasks automatically. Some robots require some degree of guidance, which may be done using a remote control or with a computer interface. Robots can be autonomous, semi-autonomous or remotely controlled. Robots have evolved so much and are capable of mimicking humans that they seem to have a mind of their own.

## 1.2 HUMAN MACHINE INTERACTION

An important aspect of a successful robotic system is the Human-Machine interaction. In the early years the only way to communicate with a robot was to program which required extensive hard work. With the development in science and robotics, gesture based recognition came into life. Gestures originate from any bodily motion or state but commonly originate from the face or hand. Gesture recognition can be considered as a way for computer to understand human body language. This has minimized the need for text interfaces and GUIs (Graphical User Interface).

## 1.3 GESTURE

A gesture is an action that has to be seen by someone else and has to convey some piece of information. Gesture is usually considered as a movement of part of the body, esp. a hand or the head, to express an idea or meaning.

## 1.4 MOTIVATION FOR PROJECT

Our motivation to work on this project came from a disabled person who was driving his wheel chair by hand with quite a lot of difficulty. So we wanted to make a device which would help such people drive their chairs without even having the need to touch the wheels of their chairs.

## 1.5 OBJECTIVE OF PROJECT

Our objective is to make this device simple as well as cheap so that it could be mass produced and can be used for a number of purposes.

## 1.6 GESTURE CONTROLLED ROBOT

Gesture recognition technologies are much younger in the world of today. At this time there is much active research in the field and little in the way of publicly available implementations. Several approaches have been developed for sensing gestures and controlling robots. Glove based technique is a well-known means of recognizing hand gestures. It utilizes a sensor attached to a glove that directly measures hand movements.

A Gesture Controlled robot is a kind of robot which can be controlled by hand gestures and not the old fashioned way by using buttons. The user just needs to wear a small transmitting device on his hand which includes a sensor which is an accelerometer in our case. Movement of the hand in a specific direction will transmit a command to the robot which will then move in a specific direction.

The transmitting device includes a microcontroller which feeds data to an Encoder IC as programmed which is used to encode the four bit data to serial data and then it will be transmitted by an RF Transmitter module.

At the receiving end an RF Receiver module will receive the encoded data and decode it by using a decoder IC and pass onto a motor driver to rotate the motors in a special configuration to make the robot move in the same direction as that of the hand.

## 1.7 APPLICATIONS

- ❖ Through the use of gesture recognition, remote control with the wave of a hand of various devices is possible.
- ❖ Gesture controlling is very helpful for handicapped and physically disabled people to achieve certain tasks, such as driving a vehicle.
- ❖ Gestures can be used to control interactions for entertainment purposes such as gaming to make the game player's experience more interactive or immersive.

# CHAPTER 2: EMBEDDED SYSTEMS

## 2.1 EMBEDDED SYSTEM

- An embedded system is a combination of computer hardware and software, either fixed in capability or programmable, that is specifically designed for a particular function.
- Industrial machines, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys (as well as the more obvious cellular phone and PDA) are the possible hosts of an embedded system.
- Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants.
- Embedded systems that are programmable are provided with programming interfaces and embedded programming is a specialized occupation.
- It is embedded as part of a complete device often including hardware and mechanical parts.
- The key characteristic is being dedicated to handle a particular task.
- The embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.
- Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.


Fig 2.1 "EMBEDDED" in our life

## 2.2 PARTS OF EMBEDDED SYSTEM

An embedded system consists of three parts:

I. Input
II. Processing Unit
III. Output

- The INPUT devices are responsible for providing input to the embedded system which is then processed by the processing unit to produce a desired output.
- In general, we use sensors as input devices while dealing with microcontrollers.
- A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument.
- PROCESSING UNIT process the input provided by the input devices and produces the output.

- All the decisions are taken by this devices depending upon the algorithm provided by the user.
- OUTPUT devices show the result of our algorithm. A number of devices can be used. Some of them include LEDs, Motors (DC, Stepper, Servo) etc.

IR SENSOR


KEYPAD

Fig 2.2 Input units

Fig 2.3 Processing unit

LEDs


LCD

Fig 2.4 Output units

## PROCESSING UNIT

- Microprocessor
- Microcontroller

## 2.3 MICROPROCESSOR

- A microprocessor incorporates the functions of computer's central processing unit (CPU) on a single integrated circuit (IC or microchip).
- It is a multipurpose, programmable, clock-driven, register-based electronic device that accepts digital data as input, processes it according to instructions stored in its memory, and provides results as output.
- It is an example of sequential digital logic, as it has internal memory, i.e. its registers.

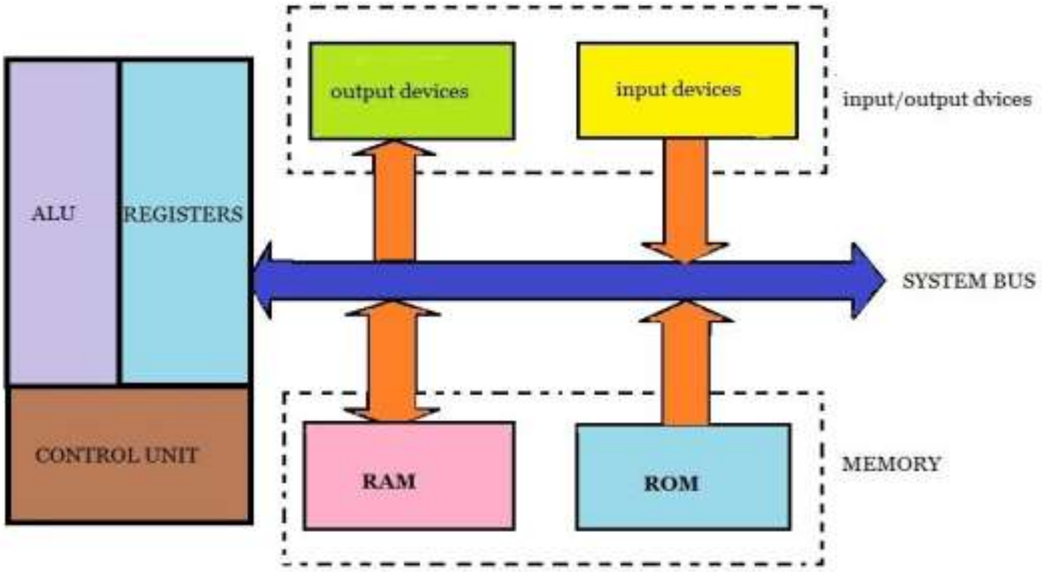


Fig 2.5 Microprocessor Block Diagram

## 2.4 MICROCONTROLLER

- A digital computer having microprocessor as the CPU along with I/O devices and memory is known as microcomputer.
- The microcontroller could be called a "one-chip solution".
- It typically includes:

- CPU (central processing unit),
- RAM (Random Access Memory),
- EPROM/ PROM/ROM
- I/O (input/output)

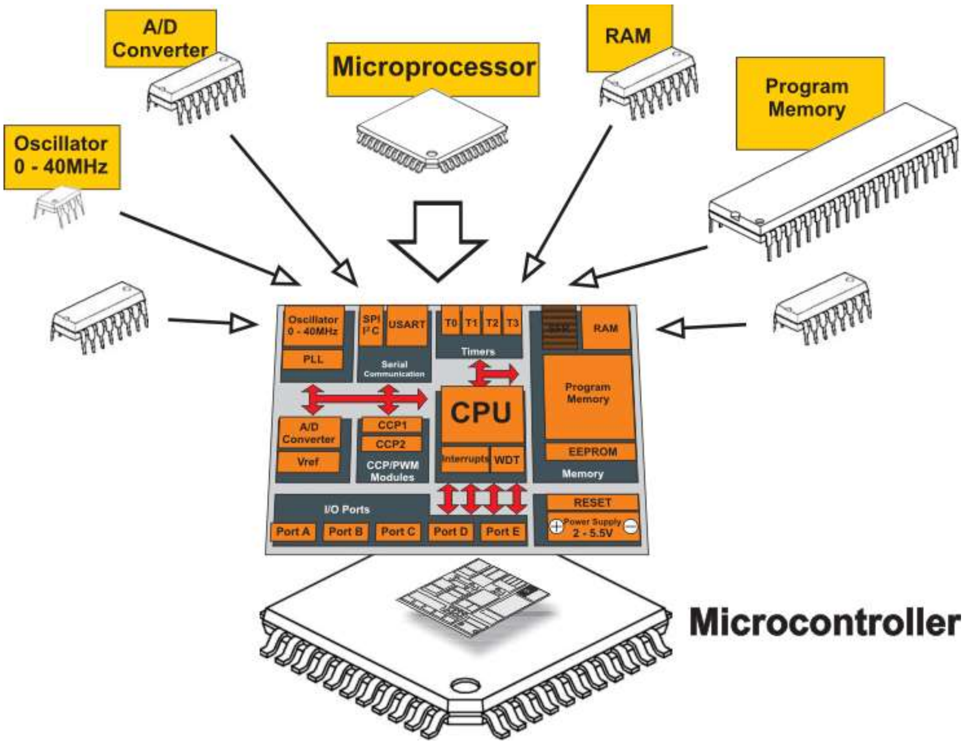


Fig 2.6 Basic Microcontroller

### 2.4.1 Different families of microcontrollers

- There are number of popular families of microcontrollers which are used in different applications as per their capability and feasibility to perform the desired task.
- Most common of these are
  - 8051 microcontroller family
  - AVR microcontroller family
  - PIC microcontrollers family

| | 8051 | PIC | AVR |
|---|---|---|---|
| **SPEED** | Slow | Moderate | Fast |
| **MEMORY** | Small | Large | Large |
| **ARCHITECTURE** | CISC | RISC | RISC |
| **ADC** | Not Present | Inbuilt | Inbuilt |
| **TIMERS** | Inbuilt | Inbuilt | Inbuilt |
| **PWM CHANELLS** | Not Present | Inbuilt | Inbuilt |

Table 2.1 Comparison of Different Microcontroller families

## 2.5 MICROPROCESSOR VS MICROCONTROLLER

| MICROPROCESSOR | MICROCONTROLLER |
|---|---|
| • CPU is stand-alone, RAM, ROM, I/O, timer are separate.<br>• Designer can decide on the amount of ROM, RAM and I/O ports.<br>• Expensive<br>• Versatility<br>• General-purpose | • CPU, RAM, ROM, I/O and timer are all on a single chip.<br>• Fix amount of on-chip ROM, RAM, I/O ports.<br>• For applications in which cost, power and space are critical<br>• Not versatile<br>• Single-purpose |

Table 2.2 Comparison between Microprocessor and Microcontroller

## 2.6 INTRODUCTION TO AVR

- **AVR** was developed in the year 1996 by Atmel Corporation.
- The architecture of **AVR** was developed by Alf-Egil Bogen and Vegard Wollan. AVR derives its name from its developers and stands for Alf-EgilBogen VegardWollan RISC microcontroller.
- They are also known as **A**dvanced **V**irtual **R**ISC.
- The AT90S8515 was the first microcontroller which was based on **AVR architecture** however the first microcontroller to hit the commercial market was AT90S1200 in the year 1997.

AVR microcontrollers are available in three different categories:

- **Tiny AVR** – Less memory, small size, suitable only for simpler applications.
- **Mega AVR** – These are the most popular ones having good amount of memory (up to 256 KB), higher number of inbuilt peripherals and suitable for moderate to complex applications.
- **Xmega AVR** – Used commercially for complex applications, which require large program memory and high speed.

| Series Name | Pins | Flash Memory |
|---|---|---|
| Tiny AVR | 6-32 | 0.5-8 KB |
| Mega AVR | 28-100 | 4-256KB |
| Xmega AVR | 44-100 | 16-384KB |

Table 2.3 Comparison of major AVRs Series

# CHAPTER 3: HARDWARE DESCRIPTION

We have divided our task into two parts to make the task simple and to avoid errors and reduce complexity.

❖ The first is the transmitting section which includes the following components:

- Accelerometer
- Microcontroller
- Encoder IC
- RF Transmitter Module

❖ The second is the receiving end which comprises of following main components:

- RF Receiver Module
- Decoder IC
- Motor Driver IC
- DC Motors

## 3.1 ACCELEROMETER (ADXL335)

An Accelerometer is an electromechanical device that measures acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic – caused by moving or vibrating the accelerometer. It is a kind of sensor which record acceleration and gives an analog data while moving in X, Y, Z direction or may be X, Y direction only depending on the type of the sensor.

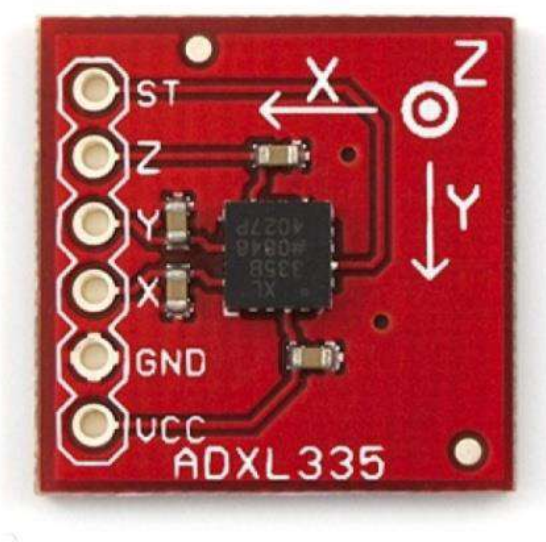


Fig3.1: Pin configuration of Accelerometer

| PIN NO. | SYMBOL | FUNCTION |
|---|---|---|
| 1 | Z | Records analog data for Z direction |
| 2 | Y | Records analog data for Y direction |
| 3 | X | Records analog data for X direction |
| 4 | GND | Connected to ground for biasing |
| 5 | VCC | +5 volt is applied |

Table 3.1: Pin description of Accelerometer

## 3.2 ATMEGA 8

The Atmel AVR ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

The Atmel AVR core combines a rich instruction set with 32 general purpose working registers .All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The device is manufactured using Atmel's high density non-volatile memory technology. The

Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core.

The ATmega8 is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program simulators, and evaluation kits.

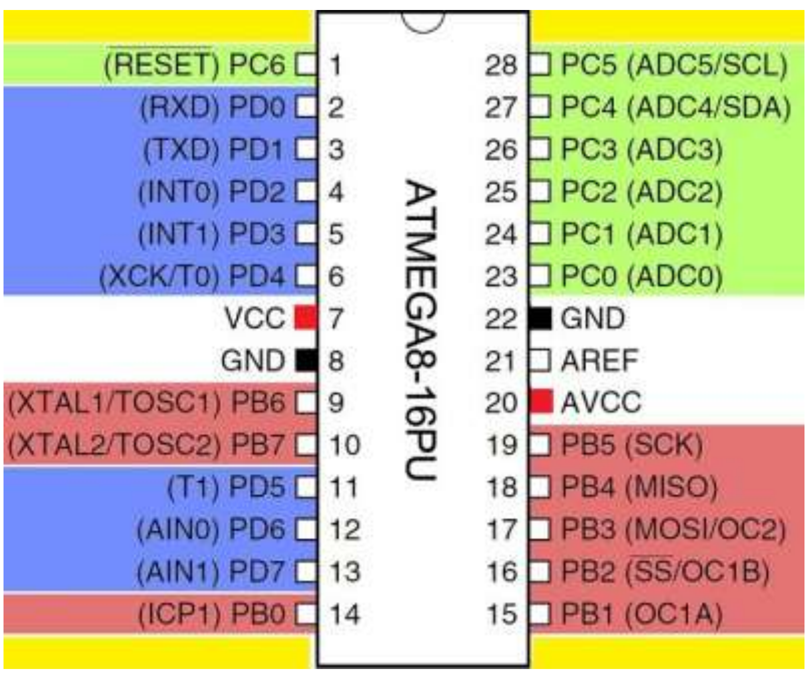


Fig3.2: Pin Configuration of Atmega8

### 3.2.1 FEATURES

- High-performance, Low-power Atmel AVR 8-bit Microcontroller
- Advanced RISC Architecture
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 8Kbytes of In-System Self-programmable Flash program memory
  - 512Bytes EEPROM
  - 1Kbyte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture.
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save and Power-down.
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
  - 4.5V - 5.5V (ATmega8)

### 3.2.2 PIN DESCRIPTION

- **VCC**

  Digital supply voltage. Magnitude of the voltage range from 4.5 to 5.5 V for the ATmega8 and 2.7 to 5.5 V for ATmega8L.

- **GND**

  Ground. Zero reference digital voltage supply.

- **PORTB (PB7… PB0)**

  PORTB is a port I / O two-way (bidirectional) 8-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTB pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

- **PORTC (PC5... PC0)**

  PORTC is a port I / O two-way (bidirectional) 7-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTC pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

- **PC6/RESET**

  If RSTDISBL Fuse programmed, PC6 then serves as a pin I / O but with different characteristics. PC0 to PC5. If Fuse RSTDISBL not programmed, then serves as input Reset PC6. LOW signal on this pin with a minimum width of 1.5 microseconds will bring the microcontroller into reset condition, although the clock is not running.

- **PORTD (PD7... PD0)**

  PORTD is a port I / O two-way (bidirectional) 8-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTD pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

- **RESET**

  Reset input pin. LOW signal on this pin with a minimum width of 1.5 microseconds will bring the microcontroller into reset condition, although the clock is not running. Signal with a width of less than 1.5 microseconds does not guarantee a Reset condition.

- **AVCC**

  AVCC is the supply voltage pin for the ADC, PC3… PC0, and ADC7... ADC6. This pin should be connected to VCC, even if the ADC is not used. If the ADC is used, AVCC should be connected to VCC through a low-pass filter to reduce noise.

- **AREF**

  Analog Reference pin for the ADC.

- **ADC7 … ADC6**

  ADC analog input.

## 3.3 ENCODER IC (HT12E)

HT12E is an encoder integrated circuit of $2^{12}$ series of encoders. They are paired with $2^{12}$ series of decoders for use in remote control system applications. It is mainly used in interfacing RF and infrared circuits. The chosen pair of encoder/decoder should have same number of addresses and data format. Simply put, HT12E converts the parallel inputs into serial output. It encodes the 12 bit parallel data into serial for transmission through an RF transmitter. These 12 bits are divided into 8 address bits and 4 data bits.

HT12E has a transmission enable pin which is active low. HT12E begins a 4-word transmission cycle upon receipt of a transmission enable. This cycle is repeated as long as TE is kept low. As soon as TE returns to high, the encoder output completes its final cycle and then stops.

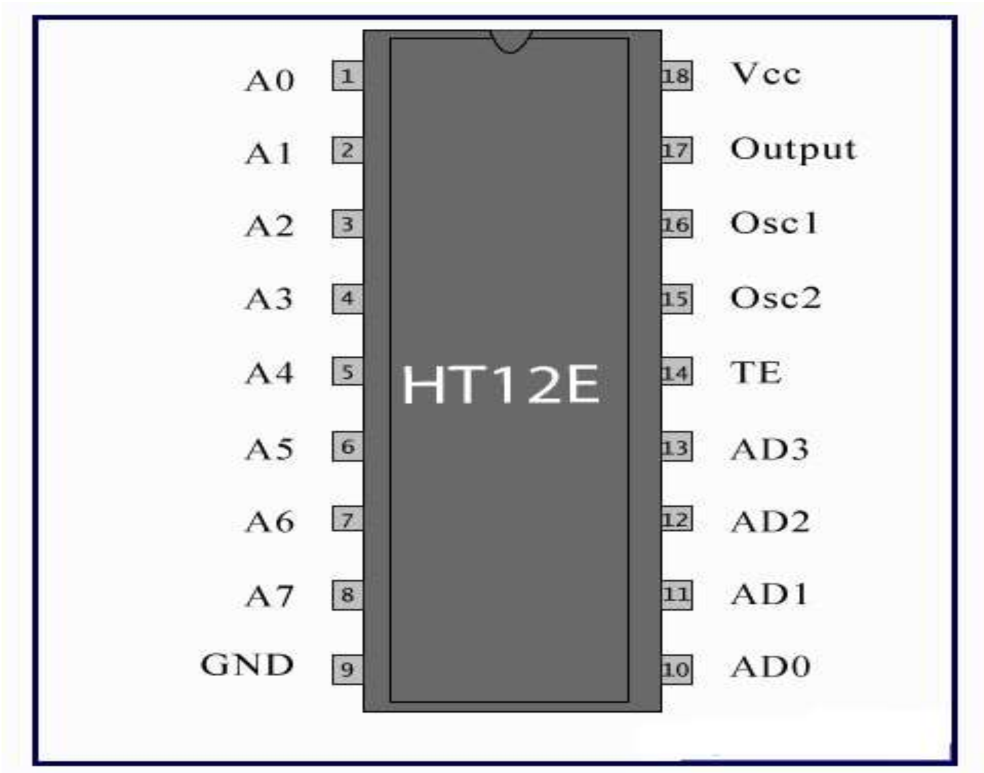


Fig 3.3: Pin configuration of Encoder IC

| Pin No. | Name | Function |
|---|---|---|
| 1-8 | A0-A7 | Address Pins |
| 9 | GND | Ground Pin |
| 10-13 | AD0-AD3 | Data Pins |
| 14 | TE | Transmission Enable |
| 15-16 | Osc2-Osc1 | Oscillator Pins |
| 17 | Output | Serial Output Pin |
| 18 | Vcc | Voltage Supply |

Table 3.2: Pin description of Encoder IC

## 3.4 RF MODULE (Rx/Tx)

The RF module, as the name suggests, operates at Radio Frequency. The corresponding frequency range varies between 30 kHz & 300 GHz. In this RF system, the digital data is represented as variations in the amplitude of carrier wave. This kind of modulation is known as Amplitude Shift Keying (ASK).

Transmission through RF is better than IR (infrared) because of many reasons. Firstly, signals through RF can travel through larger distances making it suitable for long range applications. Also, while IR mostly operates in line-of-sight mode, RF signals can travel even when there is an obstruction between transmitter & receiver. Next, RF transmission is more strong and reliable than IR transmission. RF communication uses a specific frequency unlike IR signals which are affected by other IR emitting sources.

This **RF module** comprises of an **RF Transmitter** and an **RF Receiver**. The transmitter/receiver (Tx/Rx) pair operates at a frequency of **434 MHz** An RF transmitter receives serial data and transmits it wirelessly through RF through its antenna connected at pin4. The transmission occurs at the rate of 1Kbps - 10Kbps.The transmitted data is received by an RF receiver operating at the same frequency as that of the transmitter.

The RF module is often used along with a pair of encoder/decoder. The encoder is used for encoding parallel data for transmission feed while reception is decoded by a decoder.HT12E-HT12D; HT640-HT648, etc. are some commonly used encoder/decoder pair ICs.

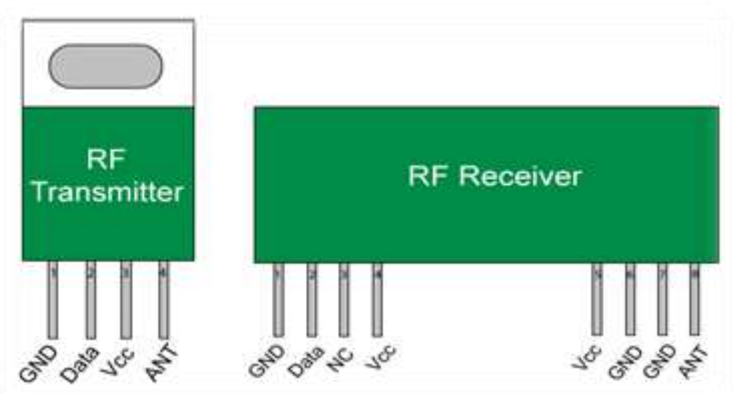


Fig 3.4: Pin configuration of RF Module

| Pin No | Function | Name |
| --- | --- | --- |
| 1 | Ground (0V) | Ground |
| 2 | Serial data input pin | Data |
| 3 | Supply voltage; 5V | Vcc |
| 4 | Antenna output pin | ANT |

| Pin No | Function | Name |
| --- | --- | --- |
| 1 | Ground (0V) | Ground |
| 2 | Serial data output pin | Data |
| 3 | Linear output pin; not connected | NC |
| 4 | Supply voltage; 5V | Vcc |
| 5 | Supply voltage; 5V | Vcc |
| 6 | Ground (0V) | Ground |
| 7 | Ground (0V) | Ground |
| 8 | Antenna input pin | ANT |

Table 3.3: Pin description of RF Module

## 3.5 DECODER IC (HT12D)

HT12D is a decoder integrated circuit that belongs to $2^{12}$ series of decoders. This series of decoders are mainly used for remote control system applications, like burglar alarm, car door controller, security system etc. It is mainly provided to interface RF and infrared circuits. They are paired with $2^{12}$ series of encoders. The chosen pair of encoder/decoder should have same number of addresses and data format.

In simple terms, HT12D converts the serial input into parallel outputs. It decodes the serial addresses and data received by, say, an RF receiver, into parallel data and sends them to output data pins. The serial input data is compared with the local addresses three times continuously. The input data code is decoded when no error or unmatched codes are found. A valid transmission in indicated by a high signal at VT pin.

HT12D is capable of decoding 12 bits, of which 8 are address bits and 4 are data bits. The data on 4 bit latch type output pins remain unchanged until new is received.

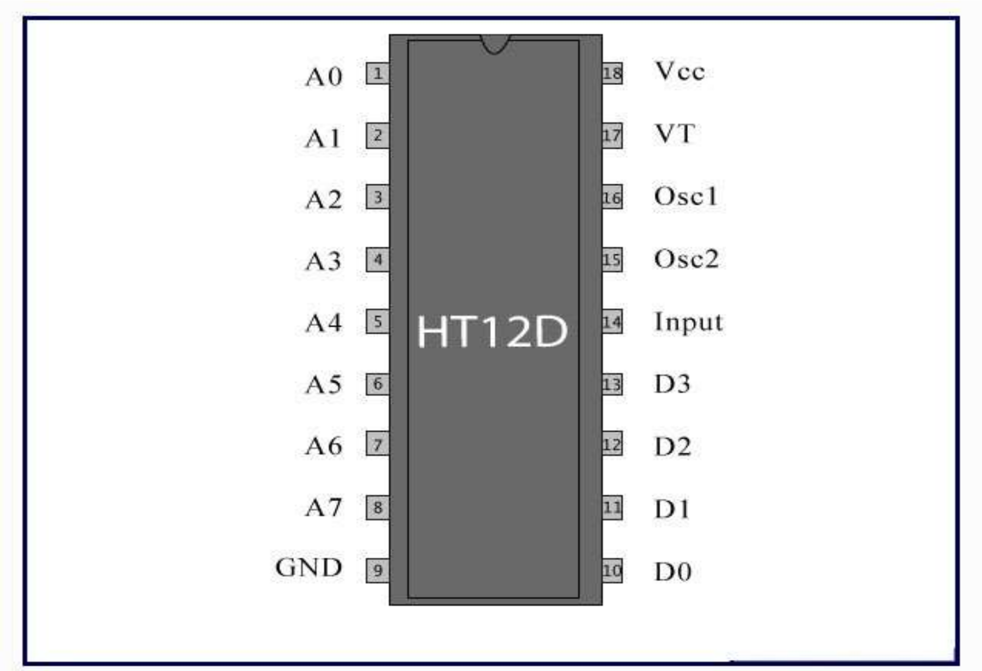


Fig 3.5: Pin configuration of Decoder IC

| Pin No. | Name | Function |
|---|---|---|
| 1-8 | A0-A7 | Address Pins |
| 9 | GND | Ground Pin |
| 10-13 | D0-D3 | Data Output Pins |
| 14 | Input | Serial Input Pin |
| 15-16 | Osc2-Osc1 | Oscillator Pins |
| 17 | VT | Valid Transmission Pin |
| 18 | Vcc | Voltage Supply |

Table 3.4: Pin description of Decoder IC

## 3.6 MOTOR DRIVER IC (L293D)

It is also known as H-Bridge or Actuator IC. Actuators are those devices which actually gives the movement to do a task like that of a motor. In the real world there are different types of motors available which work on different voltages. So we need a motor driver for running them through the controller.

The output from the microcontroller is a low current signal. The motor driver amplifies that current which can control and drive a motor. In most cases, a transistor can act as a switch and perform this task which drives the motor in a single direction.

Turning a motor ON and OFF requires only one switch to control a single motor in a single direction. We can reverse the direction of the motor by simply reversing its polarity. This can be\ achieved by using four switches that are arranged in an intelligent manner such that the circuit not only drives the motor, but also controls its direction. Out of many, one of the most common and clever design is a H-bridge circuit where transistors are arranged in a shape that resembles the English alphabet "H".

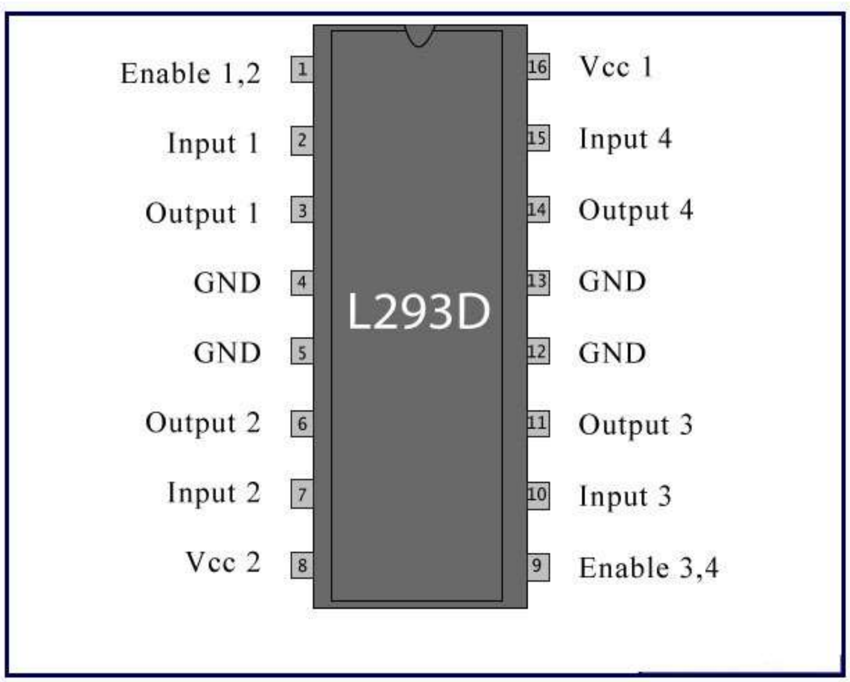


Fig 3.6: Pin configuration of L293D IC

| Pin No | Function | Name |
|---|---|---|
| 1 | Enable pin for Motor 1; active high | Enable 1,2 |
| 2 | Input 1 for Motor 1 | Input 1 |
| 3 | Output 1 for Motor 1 | Output 1 |
| 4 | Ground (0V) | Ground |
| 5 | Ground (0V) | Ground |
| 6 | Output 2 for Motor 1 | Output 2 |
| 7 | Input 2 for Motor 1 | Input 2 |
| 8 | Supply voltage for Motors; 9-12V (up to 36V) | Vcc2 |
| 9 | Enable pin for Motor 2; active high | Enable 3,4 |
| 10 | Input 1 for Motor 2 | Input 3 |
| 11 | Output 1 for Motor 2 | Output 3 |
| 12 | Ground (0V) | Ground |
| 13 | Ground (0V) | Ground |
| 14 | Output 2 for Motor 2 | Output 4 |
| 15 | Input2 for Motor 2 | Input 4 |
| 16 | Supply voltage; 5V (up to 36V) | Vcc1 |

Table 3.5: Pin description of L293D IC

## 3.7 DC MOTORS

A machine that converts DC power into mechanical power is known as a DC motor. Its operation is based on the principle that when a current carrying conductor is placed in a magnetic field, the conductor experiences a mechanical force.

DC motors have a revolving armature winding but non-revolving armature magnetic field and a stationary field winding or permanent magnet. Different connections of the field and armature winding provide different speed/torque regulation features. The speed of a DC motor can be controlled by changing the voltage applied to the armature or by changing the field current.

# CHAPTER 4: IMPLEMENTATION

## 4.1 DESIGN OVERVIEW

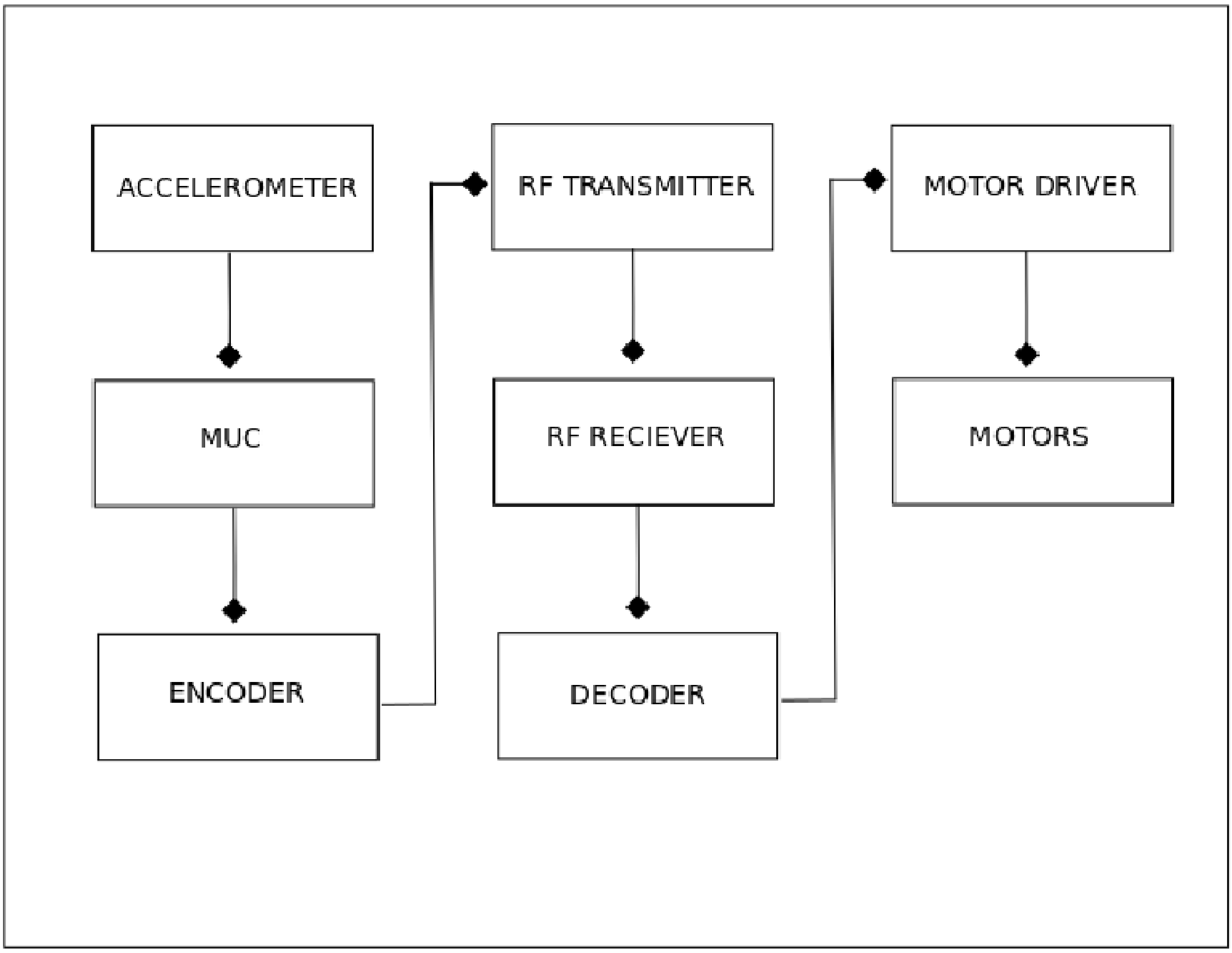


Fig 4.1: Overview

| TRANSMITION SIDE | RECEIVER SIDE |
|---|---|
| Accelerometer | RF receiver |
| MUC | Decoder |
| Encoder | Motor Driver |
| RF transmitter | Motors |

Table 4.1 Overview

## 4.2 WORKING

The accelerometer records the hand movements in the X and Y directions only and outputs constant Analog voltage levels. These voltages are fed to the internal ADC (Analog to Digital Converter) of microcontroller Atmega8.

The output from the microcontroller is the input to the encoder IC. The input to the encoder is parallel while the output is a serial coded waveform which is suitable for RF transmission. The RF transmitter modulates the input signal using Amplitude Shift Keying (ASK) modulation. It is the form of modulation that represents digital data as variations in the amplitude of a carrier wave.

The following figure shows the modulated output of the RF module:

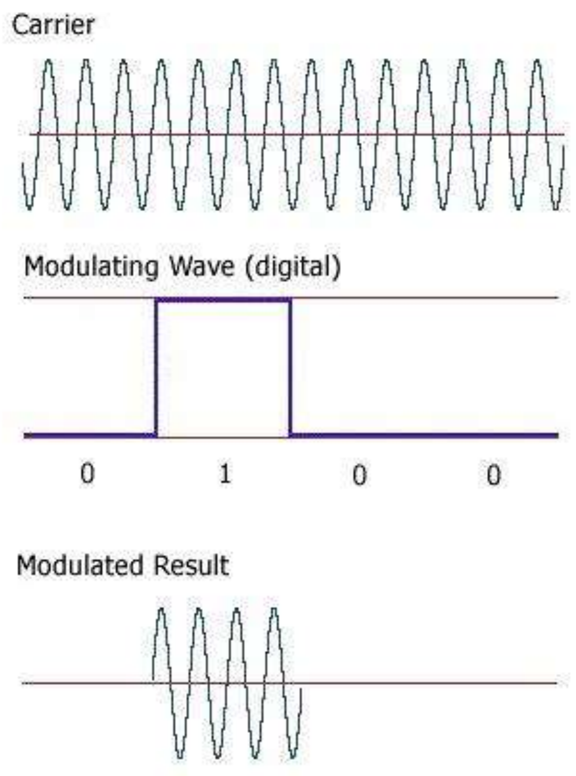


Fig 4.2: ASK Modulation

The RF modules works on the frequency of 433 MHz .It means that the carrier frequency of the RF module is 433MHz. The RF module enables the user to control the robot wirelessly and with ease.

The schematic of transmitting end can be seen below:

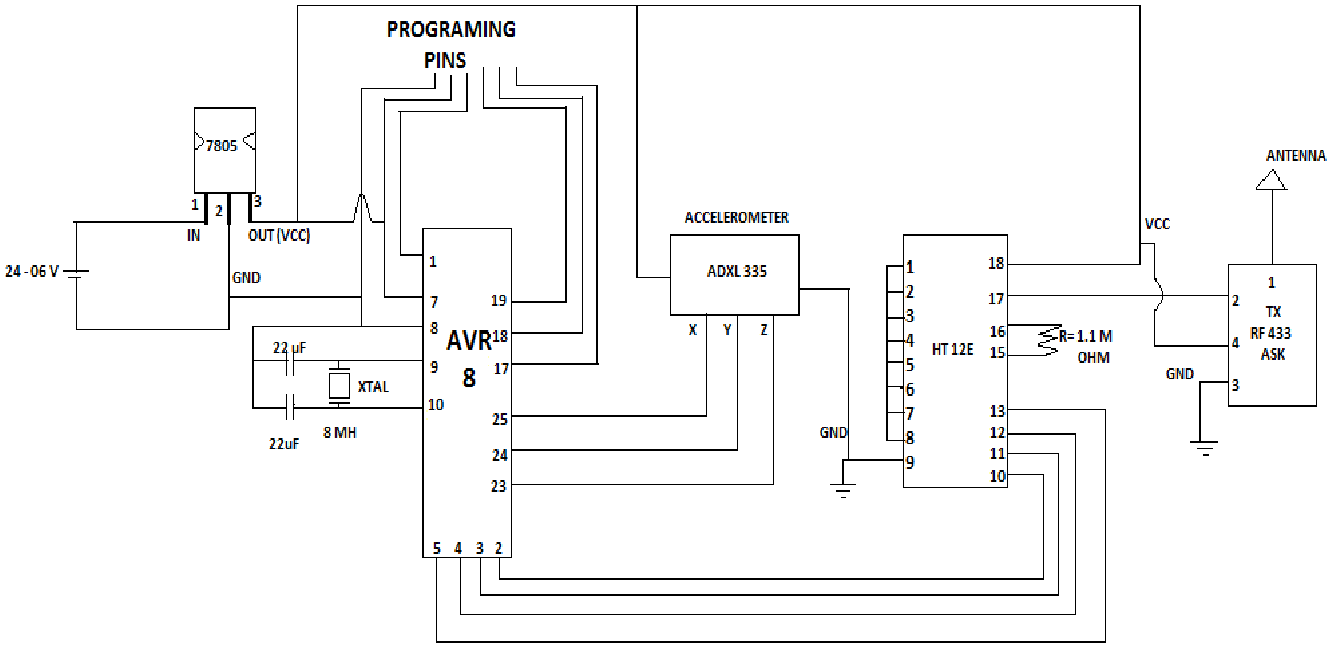


Fig 4.3 Transmitting Circuit

This transmitted signal is received by the RF receiver, demodulated and then passed onto the decoder IC. The decoder IC decodes the coded waveform and the original data bits are recovered. The input is a serial coded modulated waveform while the output is parallel. The pin 17 of the decoder IC is the Valid Transmission (VT) pin. A led can be connected to this pin which will indicate the status of the transmission. In the case of a successful transmission, the led will blink.

The parallel data from the encoder is fed to the L293D Motor Driving IC which drives the motors in a special configuration based on the hand movements. This data is in the form of bits.

The schematic of receiving end can be seen below:

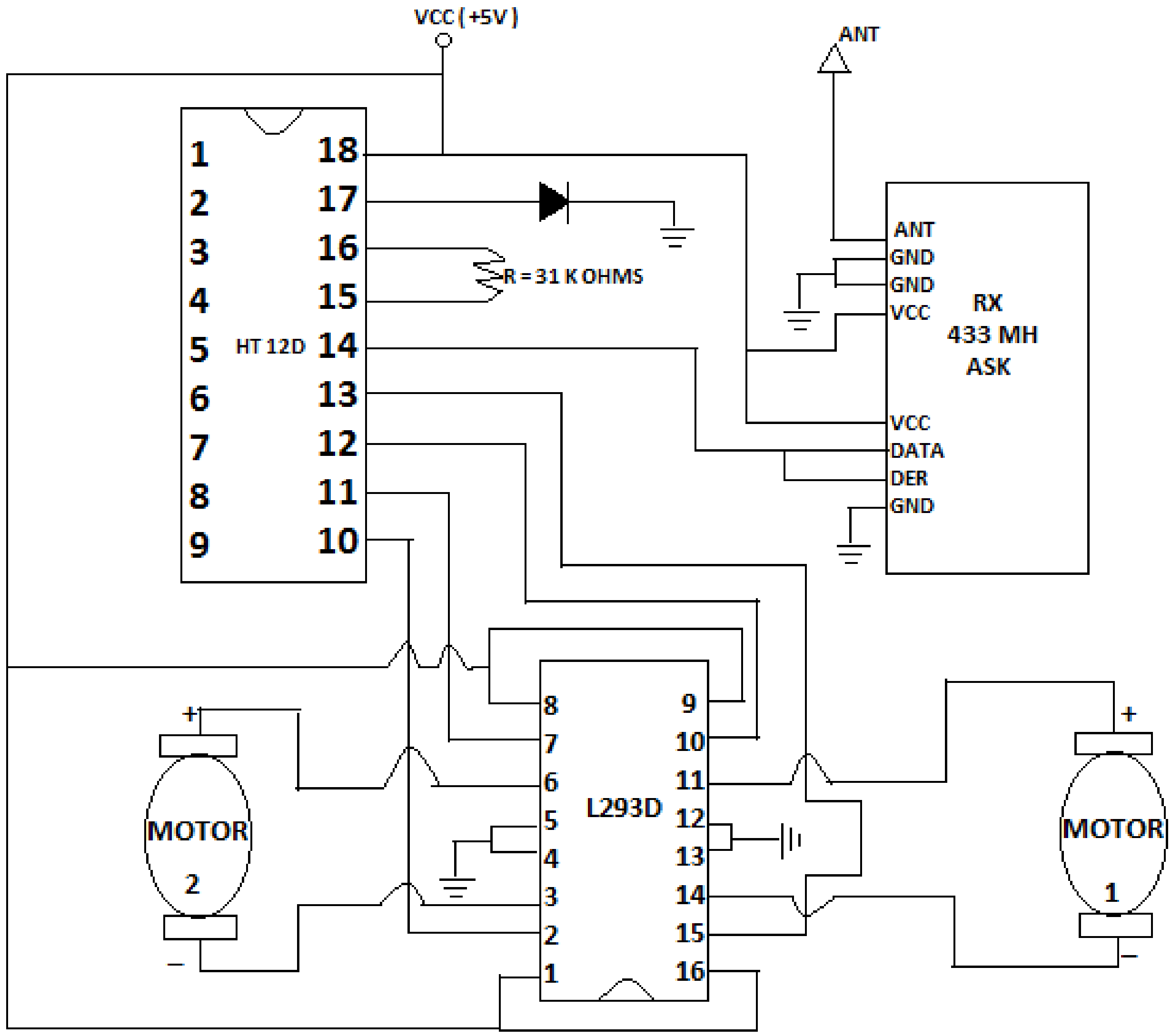


Fig 4.4: Receiving Circuit

A Crystal Oscillator is attached to the pins 9 and 10 of the microcontroller. The oscillator creates an electrical signal of a very precise frequency which is used to keep track of time. Two capacitors are connected in parallel with the oscillator to remove unwanted frequencies.

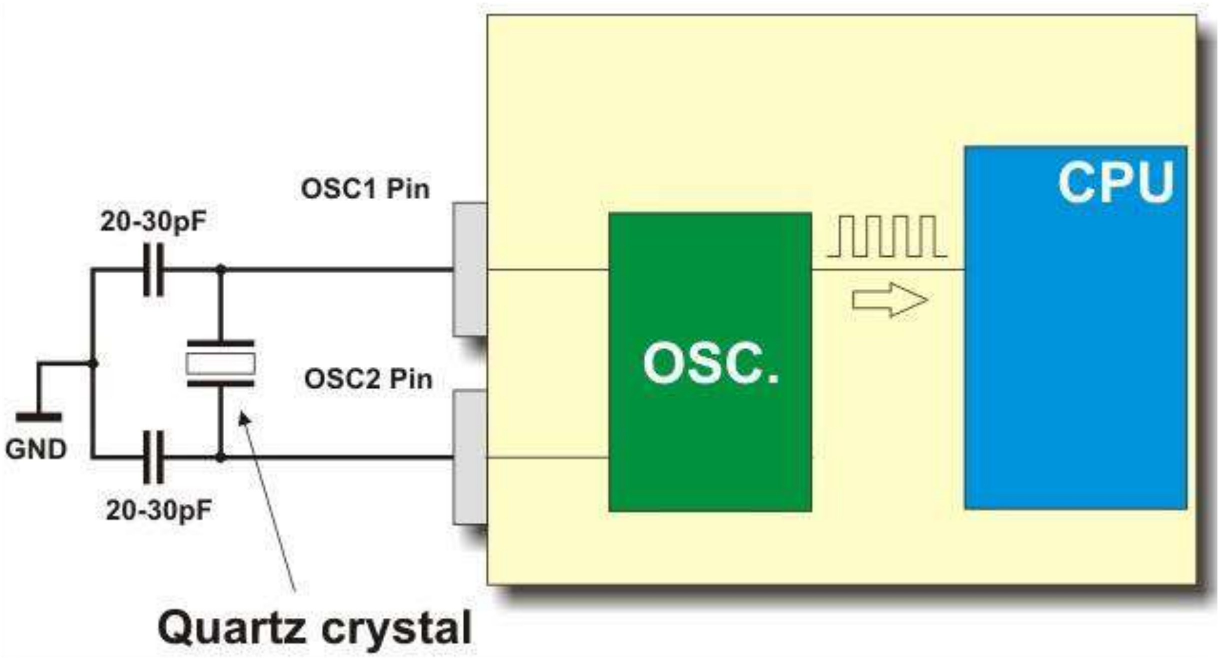


Figure 4.5: Crystal Oscillator

# CHAPTER 5: SOFTWARE DESCRIPTION

## 5.1 EMBEDDED C

Embedded C is language for programming the microcontroller for embedded applications. There is a large and growing international demand for programmers with embedded skills and many desktop developers are starting to move in this developed area.

The reasons for writing programs in C are

- It is easier and less time consuming to write in C than assembly.
- C is easier to modify and update.
- We can use code available in function libraries.
- C is portable to other microcontrollers also.

## 5.2 AVR STUDIO

It is often believed that without target hardware it is difficult, if not impossible, to develop and test software for a microcontroller project. This is often not the case as many of the microcontroller manufacturers offer software simulators that exist for just this purpose. Not only does ATMELs free IDE, AVR studio, provide the framework for compiling programmes and downloading them to the microcontroller, but it also comes with the ability to simulate programmes for most of their AVR microcontrollers .This simulator has the ability to not only execute AVR instructions but also to simulate limited I/O. AVR studio is relatively similar to other IDE’s like ECLIPSE or NET BEAMS.

## 5.2.1 USBasp (Programmer)

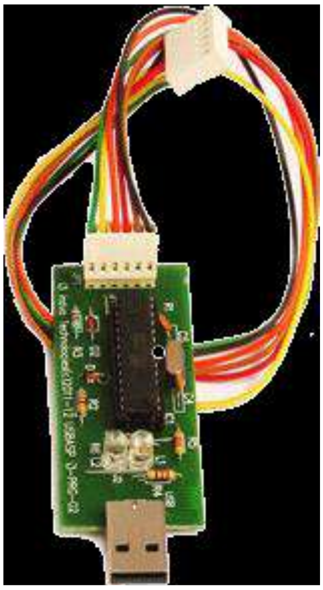

Fig 5.1 USBasp

To connect microcontroller with PC for programming we need an AVR programmer. We are using USBasp. The programmer will have a USB cable that can be used to connect with PC and an SPI cable which can be connected to the MUC.

We need programmer drivers for connecting our programmer with PC.

## 5.2.2 Steps to install Drivers for Programmer

- Insert USBasp programmer, then open Device Manager. There you will find USBasp in yellow mark.
- Right click on the yellow marked USBasp, and then click on Update Driver.
- Now update the driver using “Browse my computer for Driver Software”
- Provide the location where the Driver for USBasp exists.
- Select “install this driver software anyway” option.
- After installation close the window.

### 5.2.3 Getting the microcontroller ready for programming.

- After installation open “win32_executable” folder.
- Burn the program (.HEX file) on “Flash.”

## 5.3 WRITING THE PROGRAM

- Open the installed AVR Studio.
- Select AVR GCC and name the project.
- Select the simulator (AVR simulator in our case)
- Now start writing your code in the editor window
- After writing your code, click on “Build” Option in the menu.
- After successful building operation select “Build and Run.”

## 5.4 CODE FOR ATMEGA 8

```
#include<avr/io.h>
#include<util/delay.h>
void adc_init();
unsigned int adc_read (char channel);
void main()
	{
	unsigned int a;
	DDRD=0xFF;
	unsigned int x,y;
	adc_init();
	while(1)
		{
		x=adc_read(0);
		y=adc_read(1);
		if((x>370)&&(x<430))
			{
			if((y>370)&&(y<430))
				{
				PORTD=0x00;
				_delay_ms(5);
				}
				else if(y>430)
					{
					PORTD=0x0A;
					_delay_ms(5);
					}
					else if(y<370)
```

```
                                        {
                                        PORTD=0x05;
                                        _delay_ms(5);
                                        }
        }
        if((y>370)&&(y<430))
        {
        if((x>370)&&(x<430))
                {
                PORTD=0x00;
                _delay_ms(5);
                }
                        else if(x>430)
                        {
                        PORTD=0x04;
                        _delay_ms(5);
                        }
                                else if(x<370)
                                {
                                PORTD=0x01;
                                _delay_ms(5);
                                }
                        }
                }
}
```

```
void adc_init()
	{

		ADMUX = 0b01000000;
		ADCSRA = ((1<<ADEN) | (1<<ADPS2)| (1<<ADPS1)| (1<<ADPS0));
	}



unsigned int adc_read (char channel)
	{
	ADMUX=0x40|channel;
	ADCSRA |= (1<<ADSC);
		while(!(ADCSRA & (1<<ADIF)));
		delay_ms(1);
		ADCSRA |= (1<<ADIF);
	return ADC;
	}
```

# CONCLUSION

We achieved our objective without any hurdles i.e. the control of a robot using gestures. The robot is showing proper responses whenever we move our hand. Different Hand gestures to make the robot move in specific directions are as follow:

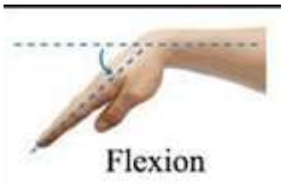


Fig 5-2 Move Forward

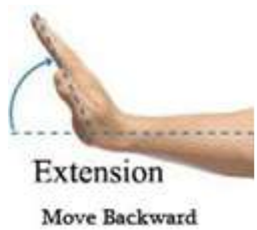


Fig 5-3 Move Backward


Fig 5-4 Move Right


Fig 5-5 Move Left

The robot only moves when the accelerometer is moved in a specific direction. The valid movements are as follows:

| DIRECTION | ACCELEROMETER ORIENTATION |
|---|---|
| Forward | +y |
| Backward | -y |
| Left | +x |
| Right | -x |
| Stop | Rest |

## LIMITATIONS AND IMPROVEMENTS

- The on-board batteries occupy a lot of space and are also quite heavy. We can either use some alternate power source for the batteries or replace the current DC Motors with ones which require less power.
- Secondly, as we are using RF for wireless transmission, the range is quite limited; nearly 50-80m. This problem can be solved by utilizing a GSM module for wireless transmission. The GSM infrastructure is installed almost all over the world. GSM will not only provide wireless connectivity but also quite a large range.
- Thirdly, an on-board camera can be installed for monitoring the robot from faraway places. All we need is a wireless camera which will broadcast and a receiver module which will provide live streaming.

# REFERENCES

[1] "Embedded Systems Guide (2002)"
http://www.webstatschecker.com/stats/keyword/a_hand_gesture_based_control_interface_for_a_car_robot

[2] "Robotic Gesture Recognition (1997)" by Jochen Triesch and Christoph Von Der Malsburg
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.5427

[3]< http://www.robotplatform.com/howto/L293/motor_driver_1.html>

[4]< http://en.wikipedia.org/wiki/Gesture_interface>

[5]< http://www.wisegeek.com/what-is-a-gear-motor.htm>

[6]< http://en.wikipedia.org/wiki/DC_motor>

[7]<http://electronics.stackexchange.com/questions/18447/what-is-back-emf-counterelectromotive-force>

[8]< http://en.wikipedia.org/wiki/Robots>

[9]www.alldatasheet.com

[10]http://letslearnelectronics.blogspot.in/