CHAPTER 1: INTRODUCTION

1.1) Robotic Arm:

A robot arm is a type of robot consisting of parts linked together in the same way as those of a human arm, mounted on a stand. The most common manufacturing robot is the robot arm which is usually made up of several metal segments. Sensors were fixed to the robot arm to detect whether there were humans close to the robot. A robot arm is a type of robot consisting of parts linked together in the same way as those of a human arm, mounted on a stand. A typical robotic arm is made up of seven metal segments, joined by six joints. The computer controls the robot by rotating individual step motors connected to each joint (some larger arms use hydraulics or pneumatics). Unlike ordinary motors, step motors move in exact increments, this allows the computer to move the arm very precisely, repeating exactly the same movement over and over again. The robot uses motion sensors to make sure it moves just the right amount. Industrial robots are designed to do exactly the same thing, in a controlled environment, over and over again. The robot stores the exact sequence of movements in its memory, and does it again and again every time a new unit comes down the assembly line.

Your arm's job is to move your hand from place to place. Similarly, the robotic arm's job is to move an end effecter from place to place. You can outfit robotic arms with all sorts of end effectors, which are suited to a particular application. One common end effector is a simplified version of the hand, which can grasp and carry different objects. Robotic hands often have built-in pressure sensors that tell the computer how hard the robot is gripping a particular object. This keeps the robot from dropping or breaking whatever it's carrying. Other end effectors include blowtorches, drills and spray painters.

Most industrial robots work in auto assembly lines, putting cars together. Robots can do a lot of this work more efficiently than human beings because they are so precise. They always drill in the exactly the same place, and they always tighten bolts with the same amount of force, no matter how many hours they've been working. Manufacturing robots are also very important in the computer industry. It takes an incredibly precise hand to put together a tiny microchip.

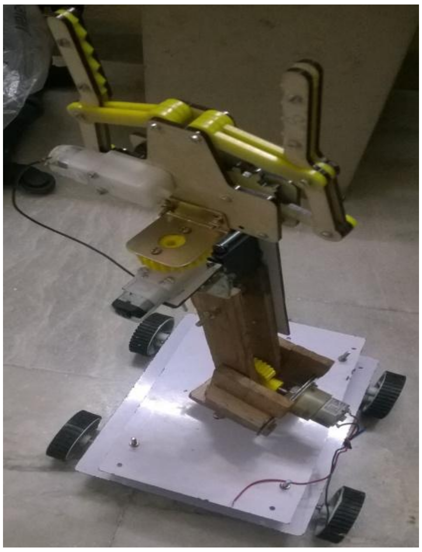

Figure 1.1: Model-1

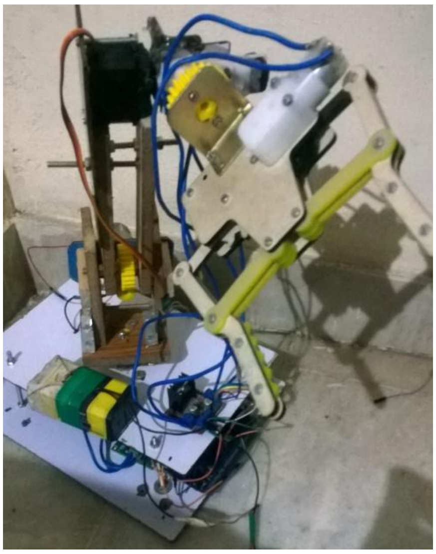

Figure 1.2: Model-2

1.2) Need of Robotics in Human Life:

- Robotics is an interesting topic of research.
- Basically it is an engineering field that is concerned with research and creation of robots for various applications.
- Robots are machines that consist of electronic and mechanical parts such as gears and cogs put together for performing tasks in place of humans.
- They can be programmed to perform a whole range of tasks with ease.
- They are most desired for certain functions because they never tire, they can endure harsh physical conditions that is possibly life threatening and they never get bored or distracted from repetitious work.
- The number of robots has begun to increase in numbers everywhere as they make human labour almost non-existent with their efficiency and throughput.
- They can work with the simplest of materials to the most dangerous such as radioactive materials.
- They can be found more commonly in industrial use from production lines of factories to harvesting of fruits in orchards.
- In the more modern military use, robots play an important role to reduce human casualties from dangerous jobs such as bomb defusal and not forgetting space exploration where it might not be possible for humans to explore and for collecting terrain sample from foreign planets.
- For exploring enemy territories unmanned aerial vehicles are used from which these pilotless drones can search terrains for hostiles and fire on targets.
- Robots are meant to complete tasks that it is programmed to do.

1.3) Purpose:

- The aim of this project is to build a robotic arm for the purpose of lifting and moving small objects.
- It can be used in industrial applications such as gripping objects from conveyor belts or it can be used in a more advanced role such as bomb defusal, where it would be dangerous for a human to interact. With camera attachments, humans can control these machines from a safe distance while completing the task at hand in a safe and efficient manner.
- In the medical field where a more delicate touch is required, a robotic arm can perform tiny incisions for a less invasive method.
- With a robotic arm jobs can be done with ease and efficiency and thus eliminate human errors and the costs that come with those errors.

CHAPTER 2: LITERATURE SURVEY

Like the growth of Apple two decades later, the birth of the industrial robotics needed two men to succeed. And both were, at first, hesitant to call the thing a robot.George Devol was the man who invented the robotic arm and whose name is on the patent that was filed for in 1954 and granted in 1961. But it was Joseph Engelberger, the man who cofounded the company Unimation, who sold that invention, the Unimate, to the industrial world. The patent Devol came up with, under the unsuspecting name "programmed article transfer," was effectively the first robotic arm. This passage in its patent filing, which does not use the word "robot" once but does use variations on the term "universal transfer devices," makes clear that the device was a robotic arm that would soon change the world—though the language, admittedly, is a bit on the dry side of things. In the 1940s, per *Robotics Age*, this automation led Devol to the idea of a "teachable machine," a device that would function based on a list of commands saved magnetically into the machine. The success of that early patent eventually led Devol to come up with the idea of the robotic arm, or as he called it, a "manipulator."

There exist some works in the field gesture recognition in which instruments are designed and build for man-machine interface using a video camera to interpret the American one-handed sign language alphabet and number gestures (plus others for additional keyboard and mouse control). Humans communicate mainly by vision and sound, therefore, a man machine interface is also available which is intuitive. It makes greater use of vision and audio recognition. Another advantage is that the user not only can communicate from a distance, but need have no physical contact with the computer. However, unlike audio commands, a visual system will be preferable as in noisy environments or at situations where sound would cause a disturbance. There is a simplification used in this project, which was not found in any recognition methods researched. The number of different gestures recognized and the recognition accuracy are amongst the best found.

The essential aim of building hand gesture recognition system is to create a natural interaction between human and computer where the recognized gestures can be used for controlling a robot or conveying meaningful information How to form the resulted hand gestures to be Understood and well interpreted by the computer considered as the problem of gesture interaction. Human computer interaction (HCI) also named Man-Machine Interaction (MMI) refers to the relation between the human and the computer or more precisely the machine, and since the machine is insignificant without suitable utilize by the human. There are two main characteristics should be deemed when designing a HCI system as mentioned in functionality and usability. System functionality referred to the set of functions or services that the system equips to the users, while system usability referred to the level and scope that the system can operate and perform specific user purposes efficiently. The system that attains a suitable balance between these concepts considered as influential performance and powerful system. Gestures used for communicating between human and machines as well as between people using sign language. Gestures can be static (posture or certain pose) which require less computational complexity or dynamic (sequence of postures) which are more complex but suitable for real time environments. Different methods have been proposed for acquiring information necessary. For recognition gestures system. Some methods used additional hardware devices such as data glove devices and colour markers to easily extract comprehensive description of gesture features. Other methods based on the appearance of the hand using the skin colour to segment the hand and extract necessary features, these methods considered easy, natural and less cost

CHAPTER 3: Design Methodology

3.1).General purpose PCB:

The general purpose PCB builds a connection between Encoder IC (HT12E), microcontroller and RF module of 433 MHz's frequency. It takes the parallel input from the microcontroller and supplies it to the data pin of Encoder IC which convert it to a serial bit and then passes it to the RF module which transmit it to the receiver section.

The circuit diagram is as shown:

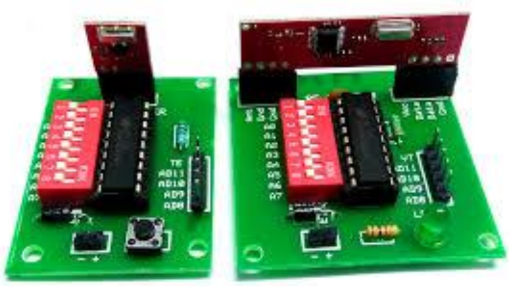

Figure 3.1: 433MHz's/315MHz's/434MHz's/866 RF Encoder & Decoder Wireless Communication Board Module

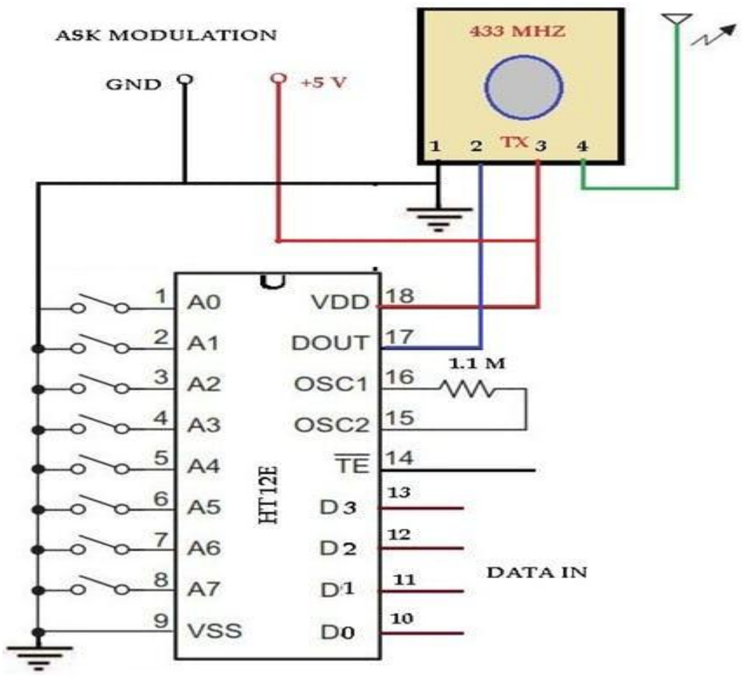

Figure 3.2: Encoder HT12E interaction with RF Transmitter

3.1.1). Encoder IC (HT12E):

The encoder consists of 4 data pins whose data can be sent. We will use these 4 data pins to represent the four motions, a HIGH on these pins individually will represent one of the four motions and a LOW on all represents stand still.The left hand pins (A0-A7) are the address pins and define the pair which will exchange data (Transmitter and Receiver having same addresses will only share data). We will set A0-A7 as LOW (Grounded).The transmission Enable (TE) pin should always high but here inverter is added at the TE pin so that's why we always connect it to the ground.

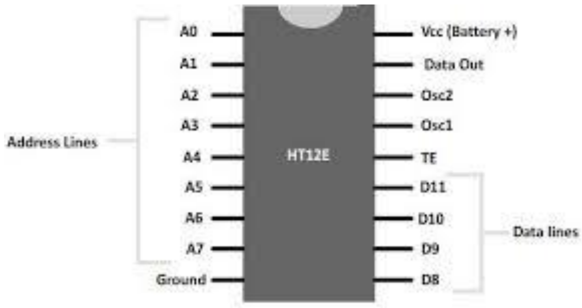

Figure 3.1.1 Encoder IC (HT12E)

3.1.2). Decoder IC (HT12D):

HT12D is a 2^{12} series decoder IC (Integrated Circuit). It is commonly used for radio frequency (RF) wireless applications. By using the paired HT12E encoder and HT12D decoder we can transmit 12 bits of parallel data serially. HT12D simply converts serial data to its input (may be received through RF receiver) to 12 bit parallel data. These 12 bit parallel data is divided in to 8 address bits and 4 data bits.

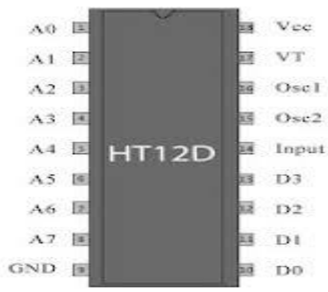

Figure 3.1.2 Decoder IC (HT12D)

3.1.3). RF Transmitter and Receiver (433 MHz's):

The function of RF module is simple: to transmit command data from wrist Arduino Uno to the motor controlling Arduino Mega. The RF module uses Radio waves at 433 Hz frequency and thus the name RF-433. They use Amplitude Modulation to send data; they will be used to transmit command to the robot.

i.e. move forward, backward, right or left. And in case of no data, stand still. They work well up to 10 meter range.

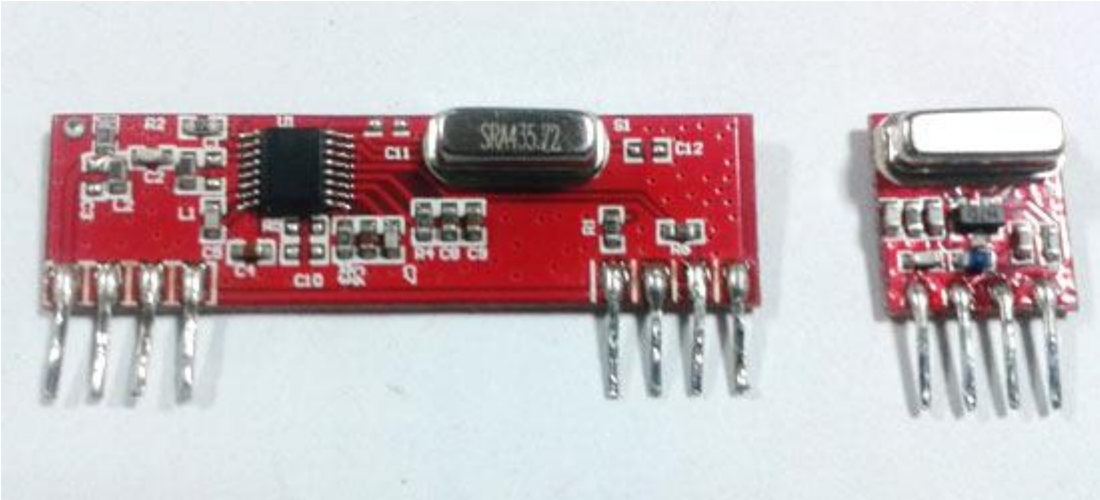

Figure 3.1.3: RF Transmitter and Receiver

3.2). Accelerometer (ADXL 335):

The function of the accelerometer is simple: to sense the orientation of the wrist. The accelerometer measures acceleration including the acceleration due to gravity 'g' as well. Thus we can use the accelerometer to sense the orientation of the wrist by measuring the component of 'g' in any particular axis of ADXL335 as shown in figure below:

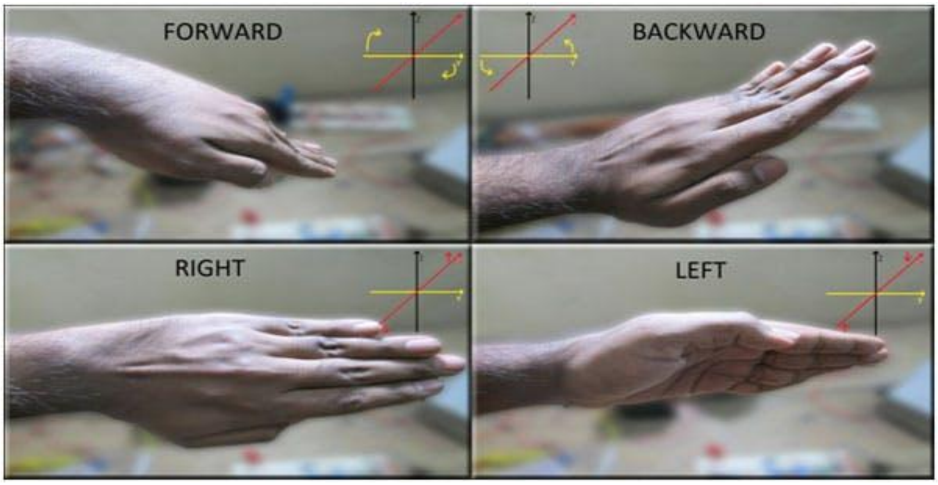

Figure 3.2.1: Command through Hand Gstures.

Due to tilt of hand, the X and/or Y axis' angle with vertical changes and hence a component of 'g' acceleration acts upon them as well which can be measured and thus indicates the orientation of the hand.The ADXL335 can measure up to 3g of acceleration and is interfaced with Arduino by connecting it's axis pins to Analog pins of Arduino. The accelerometer outputs voltage values proportional to acceleration.In this project, Accelerometer is connected to Arduino Uno and is attached to the palm. The ADXL335 outputs voltage in range from 0 to Vcc (Applied voltage usually 3.3V) and is read by Arduino's analog pins. Thus for the user, we get a value in range from 0 to 1024 (10-bit ADC). The different orientation yields a different analog value for each axis which is then mapped to different robot movements.

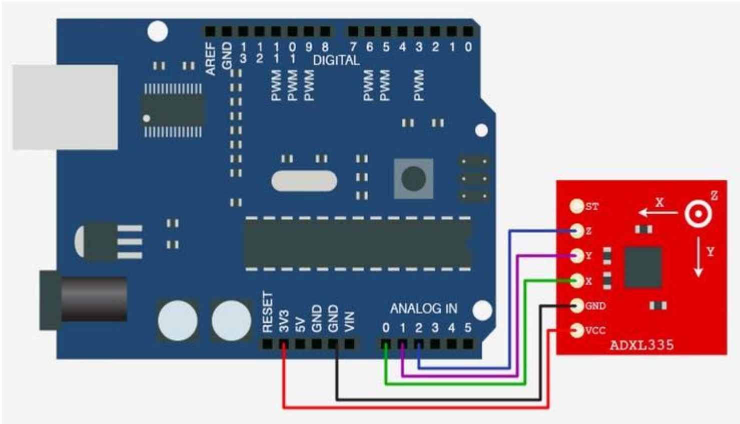

Figure 3.2.2 Accelerometer interaction with Arduino Uno.

3.3). Microcontrollers:

3.3.1). Arduino Uno (AT-Mega16 IC):

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328Pmicrocontroller and developed by Arduino.cc.The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

Figure 3.3.1: Arduino Uno (AT-Mega 16)

3.3.2). Arduino Mega 2560:

- Arduino Mega 2560 is a Microcontroller board based on Atmega2560. It comes with more memory space and I/O pins as compared to other boards available in the market.
- There are 54 digital I/O pins and 16 analog pins incorporated on the board that make this device unique and stand out from others.
- Out of 54 digital I/O, 15 are used for PWM (pulse width modulation).
- A crystal oscillator of 16MHz frequency is added on the board.

Figure 3.3.2: Arduino Mega 2560

3.4). Jumper Wires:

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

Figure 3.4.1: Jumper wires

3.5). Motor Driver (L298N):

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and

35V, with a peak current up to 2A.Let take a closer look at the pin out of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.

Next, the Input 1 and Input 2 pins are used for controlling the rotation direction of the motor A, and the inputs 3 and 4 for the motor B.

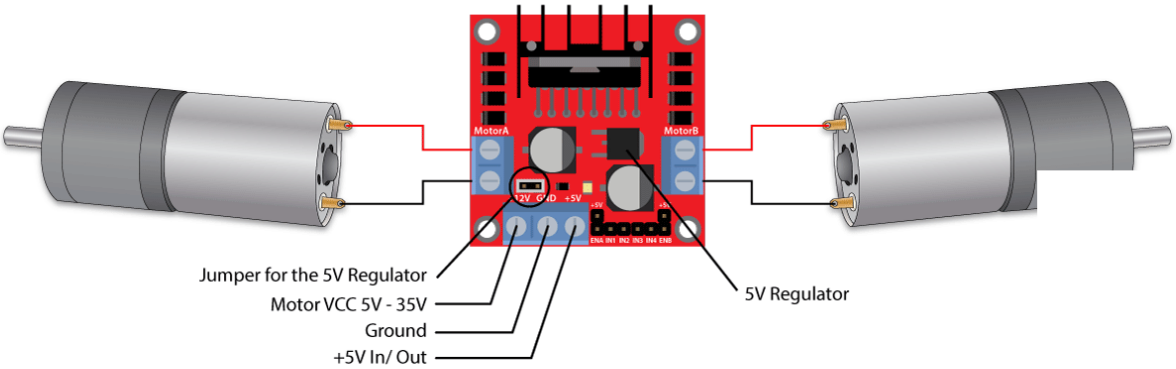

Figure 3.5.1: Motor Driver (L298N)

Using these pins we actually control the switches of the H-Bridge inside the L298N IC. If input 1 is LOW and input 2 is HIGH the motor will move forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor will move backward. In case both inputs are same, either LOW or HIGH the motor will stop. The same applies for the inputs 3 and 4 and the motor B.

3.6). 12V Power supply:

Here we are using a 4v 1.0Ah sunca battery .But our circuit requires a 12v supply hence we will connect 3 batteries of 4v in series so as to make a 12v power supply.

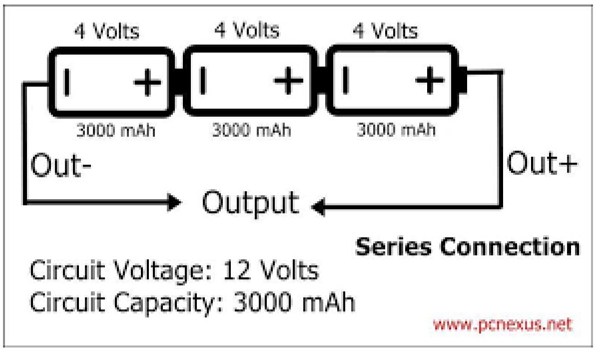

Figure 3.6.1: 3 Batteries of 4V

Connected in series.

Figure 3.6.2: Sunca batteries of

4V – 1.0Ah.

3.7). Gear motors:

3.7.1). DC motors:

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic to periodically change the direction of current flow in part of the motor.

Figure 3.7.1: DC toy motors of 100rpm.

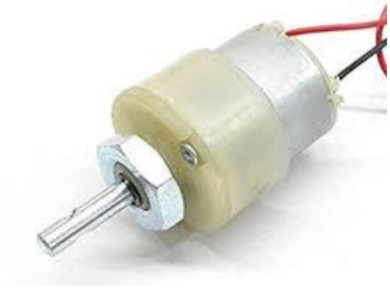

Figure 3.7.2: DC plastic gear motors.

3.7.2).Servo motor (Tower pro MG995):

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

MG995:

MG995 is a digital metal gear high torque servo for airplane, helicopter, RC-cars from 10 to 6-th scale truggy and monster and many RC models. This is the most famous servo made by tower pro.

Motors application in the project:

- Here we are using 2 DC toy motor of 100 rpm for gripping purpose.
- 2 DC motors of 50 rpm for different positions of robotic arm.
- 2 DC motors of 200 rpm for the movement of the machine.
- 1 servo motor for the up and down movement of gripper.

Figure 3.7.3: Servo motors (metal gears) – Tower pro mg995

Parameters	Values
Weight	55g
Stall torque	9.4kg/cm (4.8v); 11kg/cm (6v)
Gear type	Metal gear
Operating speed	0.20sec/60degree (4.8v); 0.16sec/60degree (6.0v);
Servo Plug	JR (Fits JR and Futaba)
Dimension	40.7×19.7×42.9mm

3.8). Gripper:

A gripper is a device which enables the holding of an object to be manipulated. The easier way to describe a gripper is to think of the human hand. Just like a hand, a gripper enables holding, tightening, handling and releasing of an object.

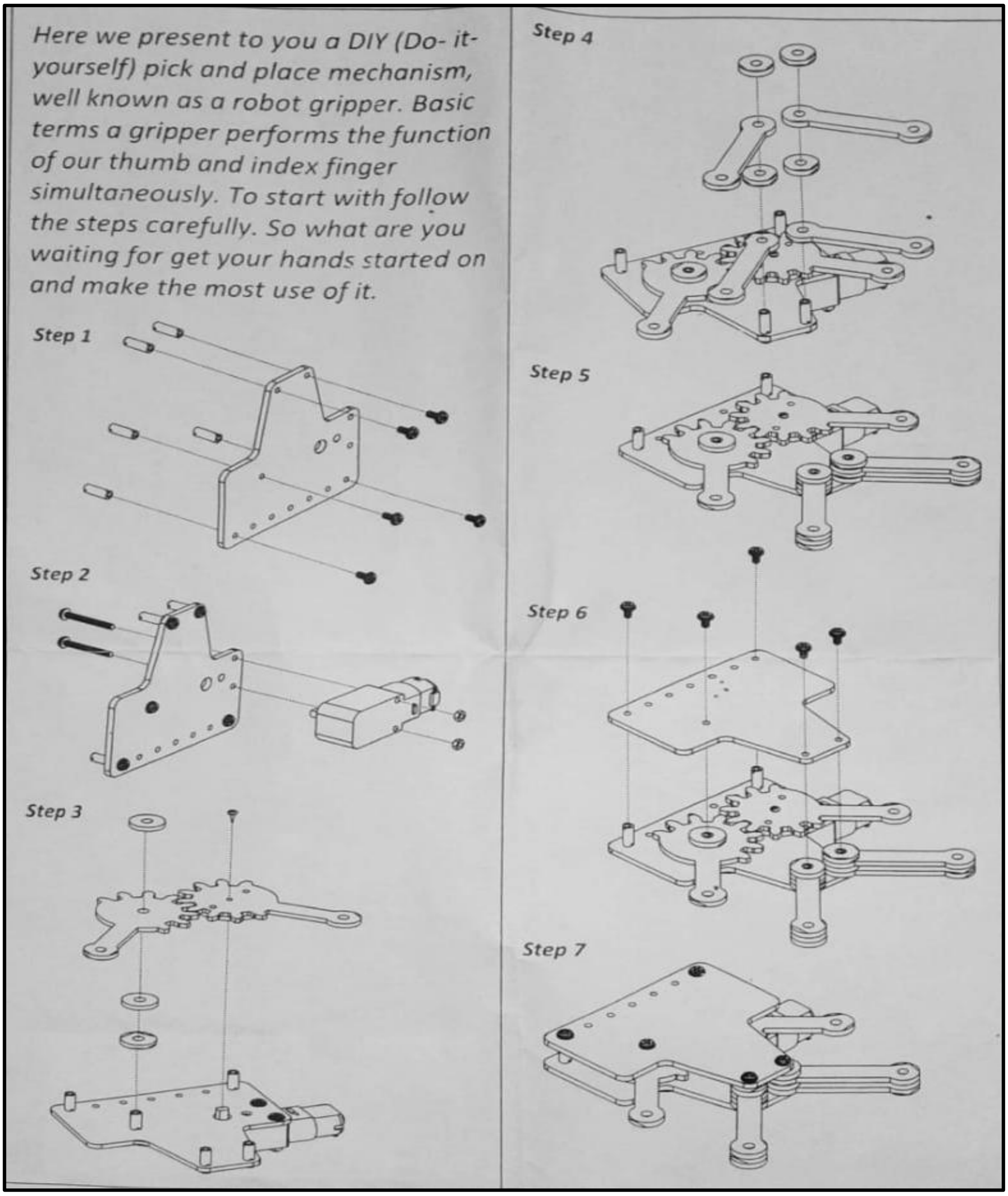

Figure 3.8.1: Gripper Mechanism-1

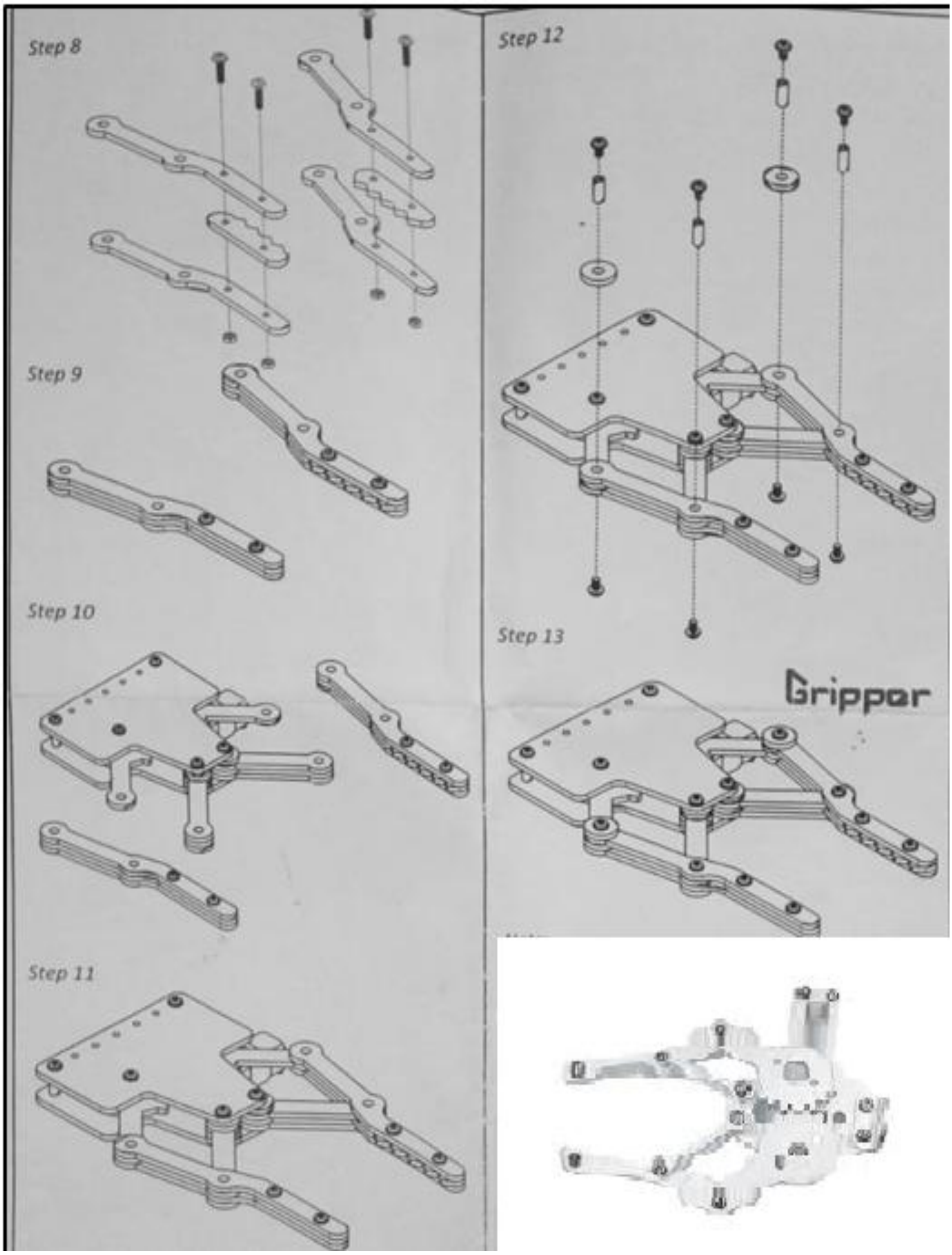

Figure 3.8.2: Gripper Mechanism-2

Chapter 4 : Tools Used

4.1) Software Tools:

Software Used: Arduino IDE (Integrated Development Environment)

Working language: Embedded C.

Arduino IDE: The Arduino **integrated development environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2.The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub **main()** into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Developer	**Arduino software**
Stable release	1.8.9 / 15 March 2019; 25 days ago
Written in	Java,C,C++
Operating System	Windows,MacOS,Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment

4.2) Electronic Components:

Electronic Component	No. of Components	Values
Resisters	8	10K,150K,1M,47K and 4 resisters of 220K ohm.
RF Transmitter	1	433 MHz's
RF Receiver	1	433 MHz's
HT12E	1	4 data and 8 address pins
HT12D	1	4 data and 8 address pins
capacitors	5	0.1uf,0.33uf,22pf,22pf,10uf
7805 voltage regulator	1	12v to 5v
L298N	3	5v to DC motors
Accelerometer	2	Analog output(250-450)
Servo Motor	1	180 degree rotation
DC motors	6	360 degree rotation
Switch	1	ON-12V, OFF-0V
Battery	3	4V each connected in series = 12V
Arduino Uno and Mega Board	1	UNO: ATMega-16 IC Mega: ATMega-2560 IC

Chapter 5: Design, Implementation and Working:

5.1) Design and Implementation:

- Transmitting Circuit:

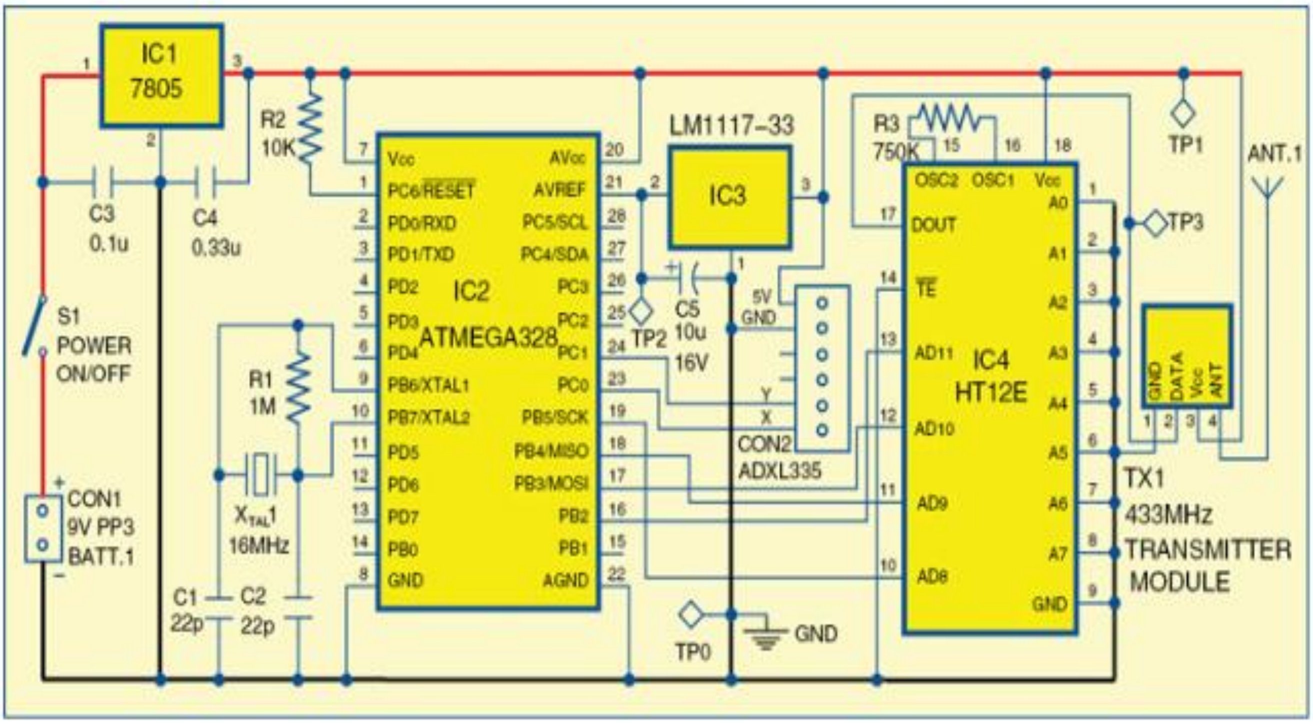

Figure 4.1: Transmitter Design Circuit

- Receiving Circuit:

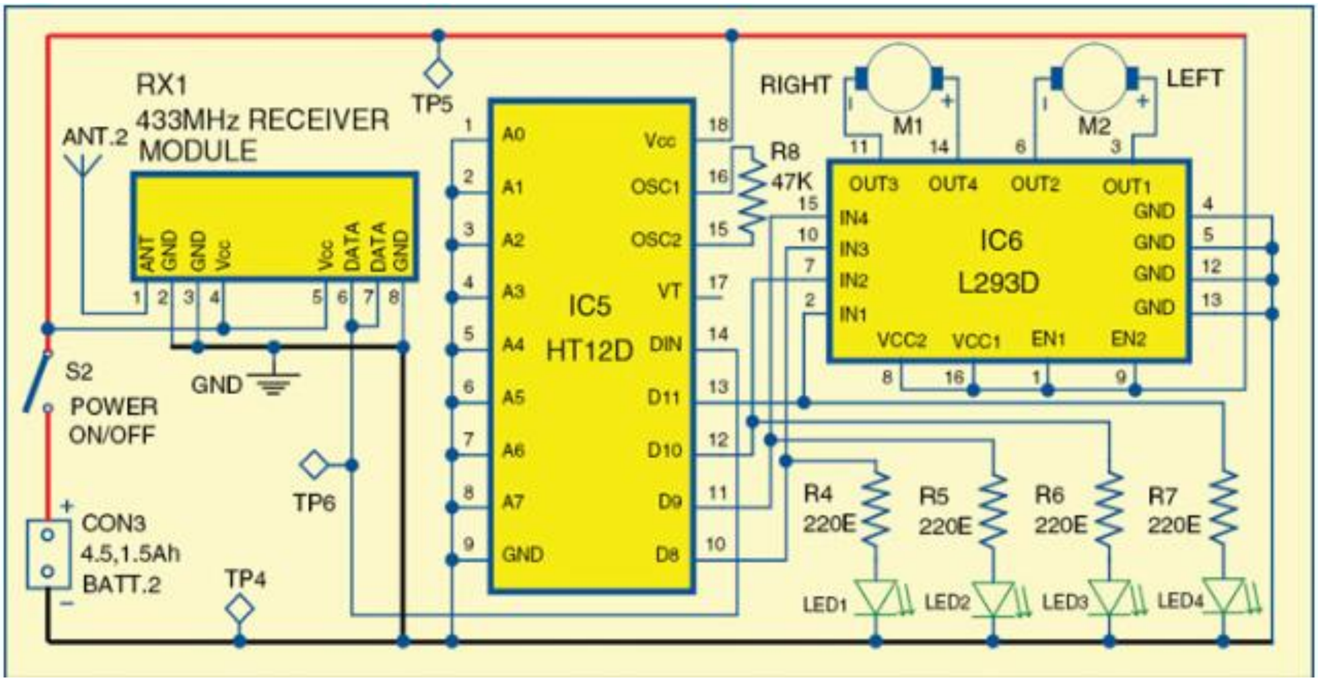

Figure 4.2: Receiver Design Circuit

The mechanical design of a robotic arm is based on a robotic manipulator with similar function like a human arm. In order to establish a generalized operating system and the technological systems for the analysis, design, integration and implementation of a humanoid robotic arm.

The main focus of the project was to design and develop the mechanism for robotic arm for lifting. The robotic arm is equipped with 6 DC and 1 servo motors to link the parts and bring arm movement. Arduino, an open source computer hardware and software is applied to control the robotic arm by driving servo motors to be capable to modify the position.

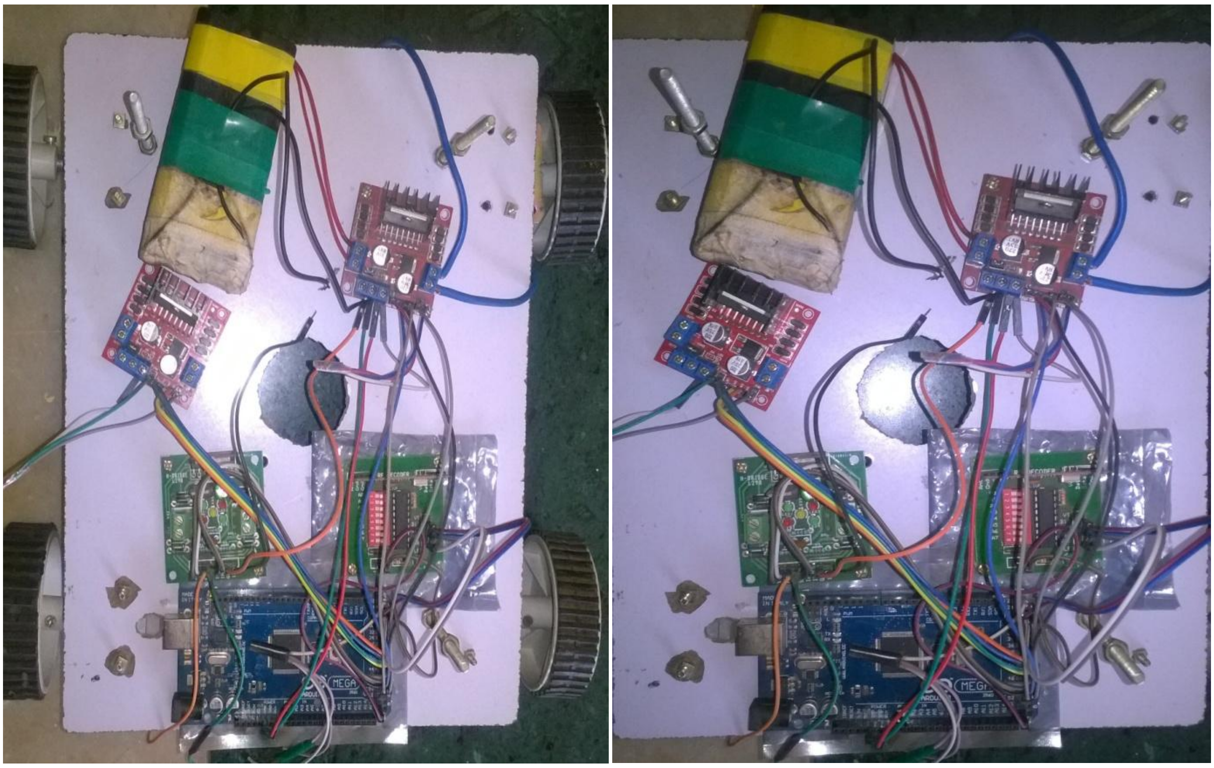

Figure 4.3: Model Circuit

Robotic arm system often consists of links, joints, actuators, sensors and controllers. The links are connected by joints to form an open kinematic chain. One end of the chain is attached to the robot base, and another end is equipped with a tool (hand, gripper, or end-effectors) which is analogous to human hand in order to perform assembly and other tasks and to interact with the environment. There are two types of joint which are prismatic and rotary joints and it connect neighbouring link.

A robotic arm with only four degrees of freedom is designed because it is adequate for most of the necessary movement. Based on the moment calculated, a suitable servo motor was selected, is, which has a torque of 152.8 oz-in to comply with the torque requirement of the robotic arm. They can turn 60 degrees in 15 million second and weight of 55.0 gram each. For the gripper, a micro servo with 9 grams and 0.12sec/60 degrees is used because not much torque needed to operate the gripping process due to the light weight of gripper. In the robotic arm, 6 DC and 1 servo motors are used.
Motors served different purposes at each part of the robotic arm.

5.2) Working Description:

❖ How does it work and recognize the gestures?

An accelerometer is a three-axis acceleration measuring device. The accelerometer used here is ADXL335 and it has 3 axes (X Y Z).

Here the brain of the robot is Arduino Uno (Atmega16) it is fed with set of code. The gestures/motion made by hand is recognized by an acceleration measuring device called accelerometer (ADXL335).

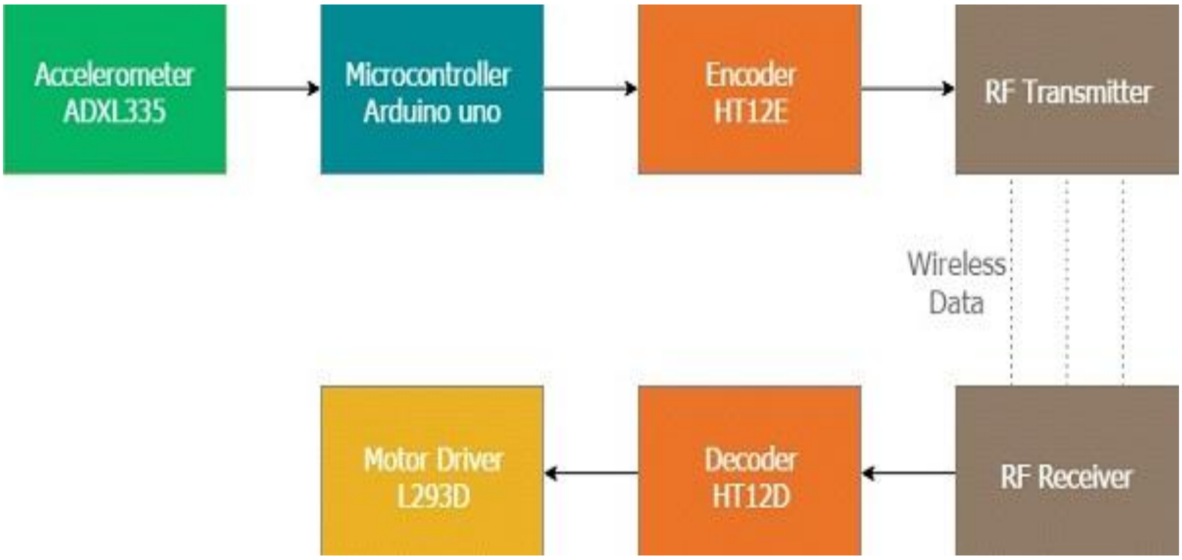

Figure 5: Control flow block diagram

Here the accelerometer reads the X Y Z coordinates when we make gestures by hand and send the X Y Z coordinates to the Arduino (here we don’t need the Z axis we need only two coordinated X and Y So neglect the Z coordinate). Accelerometer provides the analog output which is get served at analog pins of Arduino.

The Arduino checks the analog values of coordinates and then these analog values get converted into digital values by the inbuilt ADC (Analog to digital converter) and then send a 4 bit code to the Encoder IC (HT12E).

The Encoder encodes this four bit (parallel data/Binary value) to the single bit (serial data/decimal value).

The Encoder passes the data to RF transmitter which has a transmitting frequency of 433 MHz’s, the information from the encoder is received at the data pin of RF transmitter and then get transmitted through the antenna terminal.

This transmitted data is received by the RF receiver. This data is send to the data pin of decoder IC (HT12D) the decoder decodes this information into a series of parallel 4 bit data.

The decoder then passes it to Arduino Mega 2560, it contains a set of code which analyse this input from the decoder and then make appropriate actions (making a digital pin high which is connected to the motor driver) according to the conditions provided in the code.

Motor Driver (L298N) receives input from these pins only at its 4 INPUT ports. L298N has two enable pins which will always be at 5v potential and 4 IN(Input) pins .

Later the motor driver makes the decision to turn the two motors in the required direction according to the input which is provided at its 4 IN pins.

The servo motor which we are using in our case is Tower Pro 950 metal gear motor which only provides the rotation of 180 degrees.

Also gripper contains two gears in it which always rotate in the opposite direction, if one is rotating in the clockwise direction then other will rotate in the anticlockwise direction and thus creates a torque which is used to lift the wait through gripper.

The bits which we are encoding and decoding is in the no. of 4 bits thus by the concept 2^n combinations, where n is no. of bits, which is 4, so here we have 2^4 i.e. 16 combinations possible.

The data which is send follows the concept of 8421 coding scheme:

For ex:

1101 should be get transmitted, then as per 8421 coding scheme.

8 4 2 1

1 1 0 1 = 8 + 4 + 1 = 13 (will get transmitted).

0 1 0 1 = 4 + 1 = 5

0 1 1 1 = 4 + 2 + 1 = 7

1 0 1 0 = 8 + 2 = 10

1 0 1 1 = 8 + 2 + 1 = 11

Chapter 6: Result

As we have studied above about the gesture control robot, by demonstrating it we have observed that it can successfully lift light weight, capable to grab any desired thing and also capable to displace the object from one place to another which it has previously grabbed.

Transmission through RF (Radio frequency) 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 433MHz 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 transmitted.

Your arm's job is to move your hand from place to place. Similarly, the robotic arm's job is to move an end effector from place to place. You can outfit robotic arms with all sorts of end effectors, which are suited to a particular application. One common end effector is a simplified version of the hand, which can grasp and carry different objects. Other end effectors include blowtorches, drills and spray painters.

Most industrial robots work in auto assembly lines, putting cars together. Robots can do a lot of this work more efficiently than human beings because they are so precise. They always drill in the exactly the same place, and they always tighten bolts with the same amount of force, no matter how many hours they've been working

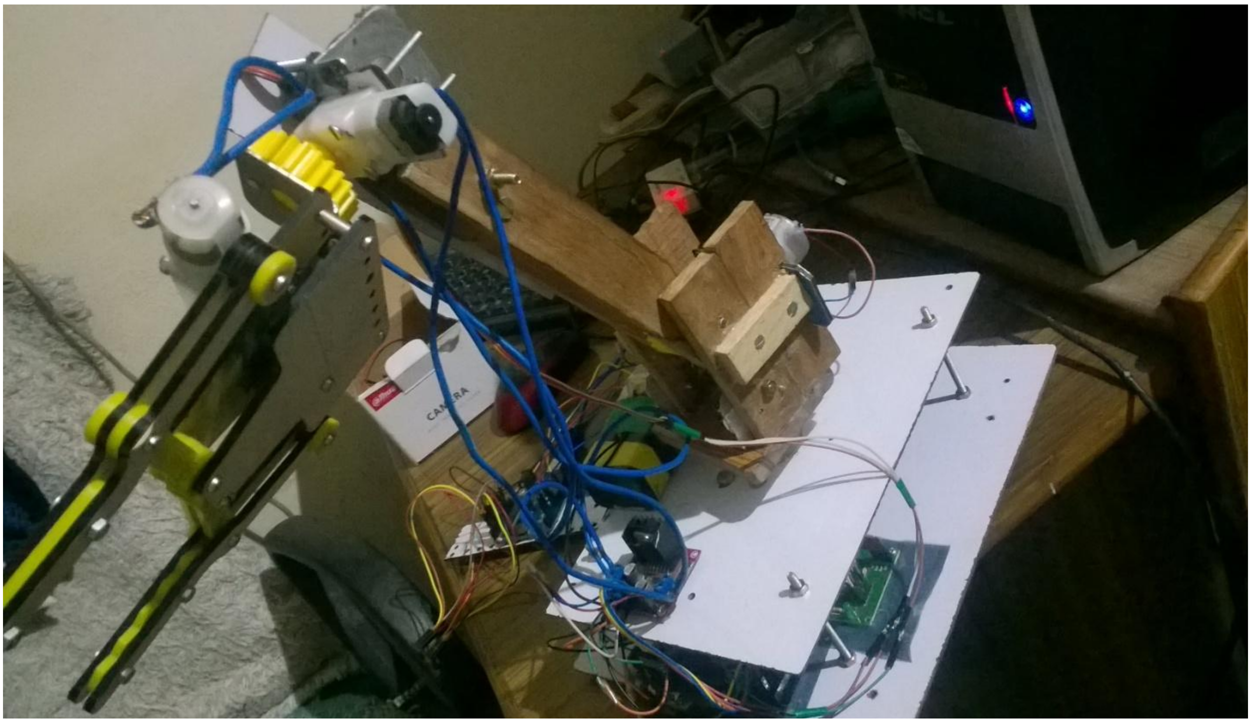

Figure 6

Chapter 7: Conclusion

Robotic arms, many areas are developable. Thanks to the gesture control robotic arms, many tasks are made easier and the resulting error level has been reduced to a minimum.

For example: some pharmacy-based drug-giving robots and a projected robot arm have been developed. In addition to this, the ability to move the robot arm is further increased, and when the camera is placed in the finger area and the sensitivity is increased, it can be used in a wide range of applications from the medical sector to the automation systems.

With the gesture control robotic arms developed in this way, the risk of infecting the patient in the medical sector is minimized, while the human errors are minimized during the surgical intervention.

Despite the fact that the robotic arm made by this project is of prototype quality, it has a quality that can be improved for more robotic systems.

Besides these, robotic arm sector, which is open to development, will keep its importance in the future. The purpose of the project is to provide control of 4 axes moving robot arm design and this robot arm with a suitable microcontroller and RF module . The necessary theoretical and practical information for this purpose has been obtained and the necessary infrastructure has been established for the project. During the process of making and developing the project, a lot of theoretical knowledge has been transferred to the practice and it has been ensured that it is suitable for the purpose of the project.

It provides a better way to control a robotic arm using accelerometer which is more intuitive and easy to work, besides offering the possibility to control a robot by other wireless means.

Using this system non Experience robotic arm controller can easily control robotic arm quickly and in a natural way.

Also, many applications which require precise control and work like human beings can be easily implemented using this approach.

And it provides more flexible control mechanism. Accelerometer equipped with gyro sensors can help to make movement smoother.

Although, the gesture control is achieved but problem of noise and jerks can be there which can be further removed by calibrating & taking more observations and using a much precise smoothing algorithms.

Chapter 8: Applications

1. Scientific:

Gesture control mobile robotic arm can help the scientist performing hazardous experiment in safer way by using robotic arm to pick the Dangerous liquids. This can also help scientist to perform the liquids which are highly flammable.

Figure 7.1: Gesture control arm in hazardous experiment.

2. Military:

Gesture controlled robotic arm can also be used for military purpose to perform operation on explosives as gesture controlled robotic arm can also help the bomb squad to detonate or defuse the bomb without involving risk to their life.

Figure 7.2: Gesture control arm in defusing explosive devices.

3. Space:

Gesture Controlled Robotic arm can also help the astronauts to repair or pick up objects in zero gravity without going outside the space craft, where movement of human being is quite difficult and is dangerous for human life.

Figure 7.3: Gesture control arm repairing space craft.

Chapter 9: Future scope

In the further development, the robotic arm should allow navigation. Design of a universal gripper is interesting because it can lift different shapes of objects. Robotic arm should have sensors to detect the position of the objects and the whole process is automated and it can also communicate with user through networking.

In the receiver section, a wireless camera is placed to monitor the performance of robot arm along with patient side (Robot arm side) 5 vital parameters (ECG, Respiration rate, Pulse rate, Temperature, Heart beat) of patient are monitored. This is a preventive measure for any imbalance in victim's metabolism (temperature, pressure, heart rate), ALARM in transmitter's section (physician side) will be ringing, which in turn brings into notice of physician that patient is in some critical situation, so that the physician immediately going to stops the action of robotic arm and he will inform the nearby doctors to take care of patient. This robotic arm developed is to reduce man power in medical field, take care of patient in absence of specialist/surgeon and to impart the robotic in medical areas.

Some future advancement of gesture technology:

Figure 8.1: Ease to access.

Figure 8.2: Microsoft Kinect

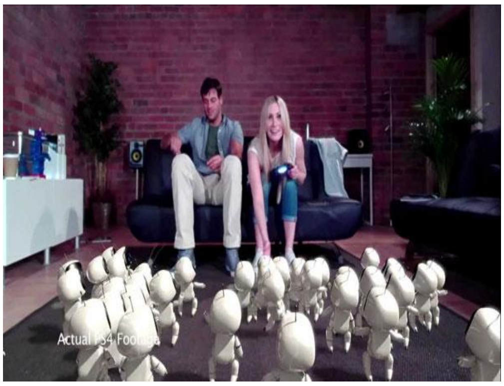

Figure 8.3: Sony PlayStation Eye/Camera

Figure 8.4: Leap Motion

Chapter 10: Cost Analysis

Components	cost
Gripper	900/-
6 clamp	15*6 = 90/-
servo motor	420/-
2 DC motors of 200 RPM	325*2 = 650/-
2 DC motor of 100 RPM	100*2 = 200/-
Mica board	30*2 = 60/-
Wheels	60*4 = 240/-
Jumper Wires	240/-
3 Batteries	40*3 = 120/-
Motor Driver	250*3 = 750/-
Arduino(Uno+Mega)	410+880 = 1290/-
Transmittion and Receiving Circuit Board	370/-
RF Module	180/-
Accelerometer	300*2 = 600/-
Plastic gears	180/-
Nut-bolts	100/-
50 RPM -DC Motors	200*2 = 400/-

Total cost = 6740/-

REFERENCES:

[1]Pattnaik, Ashutosh and Ranjan, Rajiv (2013) Robotic Arm Control through Human Arm Movement using Accelerometers. B.Tech thesis,109EI0297

[2]J. Aleotti, A. Skoglund and T. Duckett, "Position teaching of a robot arm by demonstration with a wearable input device," in International Conference on Intelligent Manipulation and Grasping (IMG04), Genoa, Italy, July 1-2, 2004

[3]I. Mihara, Y. Yamauchi, and M. Doi, "A real-time vision-based interface using motion processor and applications to robotics," in Systems and Computers in Japan, vol. 34, pp. 10-19, 2003.

[4]Pedro Neto, J. Norberto Pires, AP Moreira RO-MAN -2009 "Accelerometer based control of and industrial robotic arm."

[5]Megalingam, R.K.,Saboo, N. , Ajithkumar, N. Unny, S. Menon, D. "Kinect based gesture controlled Robotic arm" in MOOC Innovation and Technology in Education (MITE), 2013 IEEE International Conference, 20-22 Dec. 2013

[6]Aakash k. Sancheti, " Gesture Actuated Robotic Arm" , International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012 1 ISSN 2250-3153

[7]Jiang H , Wachs JP , Duerstock BS. "Integrated vision-based robotic arm interface for operators with upper limb mobility impairments" IEEE IntConfRehabil Robot. 2013 Jun;2013:6650447

[8]Avinash,http://extremeelectronics.co.in/avr-tutorials/pwm-signal-generation-by-using-avr-timers/.

Chapter 11: APPENDICES

Complete code of project:

Transmitting Arduino Code:

TRANSMITING_ARDUINO | Arduino 1.8.2

File Edit Sketch Tools Help

TRANSMITING_ARDUINO

```
void setup()
{
Serial.begin(9600);
pinMode(2,OUTPUT);
pinMode(4,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
}

void loop()
{
int xvalue = analogRead(A0);
int yvalue = analogRead(A1);
int zvalue = analogRead(A2);
int pvalue = analogRead(A3);
int qvalue = analogRead(A4);

int a2     = digitalRead(2);
int a4     = digitalRead(4);
int a7     = digitalRead(7);
int a8     = digitalRead(8);

Serial.print("|x:");
Serial.print(xvalue);
Serial.print("|y:");
Serial.print(yvalue);
Serial.print("|p:");
```

Figure 9.1

```
Serial.print(pvalue);
Serial.print("|q:");
Serial.print(qvalue);
Serial.print("\n");

 if(pvalue<290&&yvalue<320)
{
digitalWrite(2,LOW);
digitalWrite(4,HIGH);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
}
else if(pvalue>380&&yvalue>380)
{
digitalWrite(2,LOW);
digitalWrite(4,HIGH);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
}
 else if(qvalue<290&&xvalue<320)
{
digitalWrite(2,HIGH);
digitalWrite(4,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,LOW);
}
 else if(xvalue>385&&qvalue>380)
{
digitalWrite(2,HIGH);
digitalWrite(4,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
}
```

Figure 9.2

```
else if(xvalue<320)
{
digitalWrite(2,LOW);
digitalWrite(4,HIGH);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
 }
 else if(yvalue<320)
{
digitalWrite(2,LOW);
digitalWrite(4,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,LOW);
 }
 else if(yvalue>380)
{
digitalWrite(2,LOW);
digitalWrite(4,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
 }
 else if(qvalue<290)
{
digitalWrite(2,HIGH);
digitalWrite(4,HIGH);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
}
 else if(qvalue>380)
{
digitalWrite(2,LOW);
digitalWrite(4,HIGH);
digitalWrite(7,HIGH);
```

Figure 9.3

```
digitalWrite(8,HIGH);
}
 else if(pvalue<290)
{
digitalWrite(2,HIGH);
digitalWrite(4,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
}
else if(xvalue>385)
{
digitalWrite(2,HIGH);
digitalWrite(4,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
 }
else
{
digitalWrite(2,LOW);
digitalWrite(4,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
}
 Serial.print(a2);
Serial.print(a4);
Serial.print(a7);
Serial.print(a8);
Serial.print("\n");

 delay(10);
 }
```

Figure 9.4

Receiving Arduino Code:

receiver_arduino | Arduino 1.8.2

File Edit Sketch Tools Help

receiver_arduino

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position
void setup()
{
Serial.begin(9600);
pinMode(22,INPUT);
pinMode(23,INPUT);
pinMode(24,INPUT);
pinMode(25,INPUT);
pinMode(36,OUTPUT);
pinMode(37,OUTPUT);
pinMode(38,OUTPUT);
pinMode(39,OUTPUT);
pinMode(40,OUTPUT);
myservo.attach(9); // attaches the servo on pin 9 to the servo object
}
void loop()
{
int a2=digitalRead(22);
int a3=digitalRead(23);
int a4=digitalRead(24);
int a5=digitalRead(25);
Serial.print(a2);
```

Figure 10.1

```
Serial.print(' ');
Serial.print(a3);
Serial.print(' ');
Serial.print(a4);
Serial.print(' ');
Serial.print(a5);
Serial.print("\n");

if(a2==LOW&&a3==LOW&&a4==LOW&&a5==HIGH)
{
  myservo.write(180);
   delay(15);
 }
 else if(a2==LOW&&a3==LOW&&a4==HIGH&&a5==LOW)
{
    myservo.write(pos);
    delay(15);
 }
 else if(a2==HIGH&&a3==LOW&&a4==LOW&&a5==LOW)
{
digitalWrite(36,HIGH);
digitalWrite(37,HIGH);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
}

else if(a2==LOW&&a3==HIGH&&a4==LOW&&a5==LOW)
{
digitalWrite(36,LOW);
digitalWrite(37,LOW);
digitalWrite(38,HIGH);
digitalWrite(39,HIGH);
}
```

Figure 10.2

```
  else if(a2==HIGH&&a3==HIGH&&a4==LOW&&a5==LOW)
{
digitalWrite(36,LOW);
digitalWrite(37,LOW);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(40,HIGH);
digitalWrite(41,LOW);
digitalWrite(42,LOW);
digitalWrite(43,LOW);
}
  else if(a2==LOW&&a3==HIGH&&a4==LOW&&a5==HIGH)
{
digitalWrite(40,LOW);
digitalWrite(41,LOW);
digitalWrite(42,HIGH);
digitalWrite(43,LOW);
digitalWrite(36,LOW);
digitalWrite(37,LOW);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
detach();
}
 else if(a2==HIGH&&a3==LOW&&a4==HIGH&&a5==LOW)
{
digitalWrite(40,LOW);
digitalWrite(41,HIGH);
digitalWrite(42,LOW);
digitalWrite(43,LOW);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(36,LOW);
```

Figure 10.3

```
digitalWrite(37,LOW);
detach();
}
 else if(a2==LOW&&a3==LOW&&a4==HIGH&&a5==HIGH)
{
digitalWrite(40,LOW);
digitalWrite(41,LOW);
digitalWrite(42,LOW);
digitalWrite(43,HIGH);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(36,LOW);
digitalWrite(37,LOW);
detach();
 }

 else if(a2==LOW&&a3==HIGH&&a4==LOW&&a5==HIGH)
{
digitalWrite(40,LOW);
digitalWrite(41,LOW);
digitalWrite(42,LOW);
digitalWrite(43,LOW);
digitalWrite(44,HIGH);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(36,LOW);
digitalWrite(37,LOW);
detach();
 }

 else if(a2==LOW&&a3==HIGH&&a4==HIGH&&a5==HIGH)
{
digitalWrite(40,LOW);
```

Figure 10.4

```
 else if(a2==LOW&&a3==HIGH&&a4==HIGH&&a5==HIGH)
{
digitalWrite(40,LOW);
digitalWrite(41,LOW);
digitalWrite(42,LOW);
digitalWrite(43,LOW);
digitalWrite(44,LOW);
digitalWrite(45,HIGH);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(36,LOW);
digitalWrite(37,LOW);
detach();
 }

 else if(a2==HIGH&&a3==LOW&&a4==HIGH&&a5==LOW)
{
digitalWrite(40,LOW);
digitalWrite(41,LOW);
digitalWrite(42,LOW);
digitalWrite(43,LOW);
digitalWrite(44,LOW);
digitalWrite(45,LOW);
digitalWrite(46,HIGH);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(36,LOW);
digitalWrite(37,LOW);
detach();
 }

 else if(a2==HIGH&&a3==LOW&&a4==HIGH&&a5==HIGH)
{
```

Figure 10.5

```
{
digitalWrite(40,LOW);
digitalWrite(41,LOW);
digitalWrite(42,LOW);
digitalWrite(43,LOW);
digitalWrite(44,LOW);
digitalWrite(45,LOW);
digitalWrite(46,LOW);
digitalWrite(47,HIGH);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(36,LOW);
digitalWrite(37,LOW);
detach();
 }
  else
{
digitalWrite(36,LOW);
digitalWrite(37,LOW);
digitalWrite(38,LOW);
digitalWrite(39,LOW);
digitalWrite(40,LOW);
digitalWrite(41,LOW);
digitalWrite(42,LOW);
digitalWrite(43,LOW);
}
 delay(10);
 }
 void detach()
 {
 myservo.detach();
 }
```

Figure 10.6