GESTURE CONTROL USING ARDUINO IN MECHANICAL SYSTEMS

A B.Tech Project Report

Submitted by

ADARSH JAIN (B17ME002)

AJAY GOEL (B17ME004)

Under the Supervision

Of

Dr. B.Ravindra

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Department of Mechanical Engineering

Indian Institute of Technology Jodhpur

April, 2019

CONTENTS

Details

List of figures…….3

Acknowledgements……4

Abstract……...…5

1. Introduction

1.1 Objective…….……6

2. Work done

2.1 Methodology……7

2.2 Theoretical Work……...…...7

2.3 Experimental Analysis……....13

2.4 Implementation……...16

3. Result and Conclusion…….……..20

4. Prototype…….…….22

5. Future work……..……..23

Appendix……..……..24

References……..……29

List of figures

Figure 1. MPU-6050 Gyro + Accelerometer Sensor …….8

Figure 2. The Orientation of 3-axis in Gyroscope……..8

Figure 3. The Orientation of 3-axis in Accelerometer……9

Figure 4. L293D Motor driver IC……...9

Figure 5. RF Receiver and Transmitter 434Mhz……..10

Figure 6. HT-12E Encoder……11

Figure 7. HT-12D Decoder……...12

Figure 8. IC 7805……...……...12

Figure 9. DC Motor……...……...13

Figure 10. Block diagram of transmitter section…...…….……..17

Figure 11. Circuit diagram of transmitter section…….…....18

Figure 12. Block diagram of receiver section...……18

Figure 13. Circuit diagram of receiver section…….19

Figure 14. Stop……..20

Figure 15. Forward Motion…….....20

Figure 16. Backward Motion……20

Figure 17. Left Turn……..20

Figure 18. Right Turn……20

Figure 19. Transmitter circuit Prototype…….……..22

Figure 20. Receiver circuit Prototype……22

Acknowledgements

The completion of any project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge. This report acknowledges a number of guidance, supervision, stimulation and lot of inspiration from numerous people.

It is our privilege to express our sincerest regards to our project advisor **Prof. Dr. B.Ravindra** for his valuable inputs, able guidance, encouragement, wholehearted cooperation and constructive criticism throughout the duration of our project.

We take the opportunity to thank all our lectures who have directly or indirectly helped our project. Last but not the least we express our thanks to our friends for their cooperation and support.

Mr. Adarsh Jain

Mr. Ajay Goel

Abstract

This project presents a Hand Gesture Controlled Mechanical system using Arduino, which can be controlled by simple hand wrist movement. According to the movement of the person's hand wrist, the accelerometer starts moving. It is based on 3axis of accelerometer and Mechanical system moves in four direction forward, backward, left and right.

For sensing Human motion, we used MPU-6050 sensor which consists of 3-axis Gyroscope Accelerometer with Micro Electro Mechanical System (MEMS) technology. This type of Gesture controlled systems can be used by a person using a wheelchair, military application, industrial robotic, purpose of surgery, construction field, and entertainment purpose also.

1. Introduction

1.1 Objective

The objective of this project is to design a hand gesture control mechanical system using Arduino board, controlled by Flexion, Extension, Pronation, and Supination exhibit Forward, Backward motion along with Left and Right turning.

This project finds applications in:

The wheelchair: Gesture controlled wheelchair can be used by elders and disabled people for easy driving and controlling of wheel chair. They will not be dependent on others.

Surveillance: When you want to control the motion of the CCTV cameras, we can use the gesture control on that system

Industrial grade robotic arms: Since we know, while doing welding a lot of harmful gases and radiations are released which is harmful for the person doing job. So, using this gesture control system, we can perform MMAW without affecting the human resource.

Purpose of Surgery: These systems are used in medical applications for the purpose of surgery, using gesture control we can optimized the movement of medical tools matching the level of sensitivity required.

Entertainment purposes: Gestures can be used to control interactions for entertainment purposes such as gaming to make the game player's experience more interactive or immersive.

2. Work done

2.1 Methodology

A gesture-controlled system is controlled by using hand wrist movement in place of any other manual method of controlling like buttons or joystick. Here someone needs to move hand wrist only to control the system. In our hand, a communication device is used, which has RF transmitter and accelerometer meter. This will send the command to the system so that it can perform the necessary tasks such as Forward Motion, backward motion, turning left, turning right, and stopping. All these tasks will be done using hand wrist movements.

Hand gesture control work system

As mentioned earlier, gesture-controlled system is a wireless powered system and has two parts: transmitter and receiver. When the system is turned on, the transmitter part, which contains Arduino, MPU6050, Encoder and RF transmitter, will constantly monitor the MPU6050 sensor.

This data has been captured by Arduino, which then transmits a related data to the encoder based on the orientation of the MPU6050 sensor. Parallel data received by the encoder is converted into serial data and this serial data is transmitted by the RF transmitter.

In the receiver section, the RF receiver receives the serial data and transmits it to the decoder IC. Decoder will convert serial data into parallel data and this parallel data is given to motor driver IC. Depending on the data, speed of motors and therefore the speed of the system is defined.

2.2Theoretical

MPU-6050 Accelerometer + Gyro

The MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefore, it captures the x, y, and z channel at the same time. The sensor uses the I2Cbus to interface with the Arduino.

The MPU-6050 is not expensive, especially given the fact that it combines both an accelerometer and a gyro.

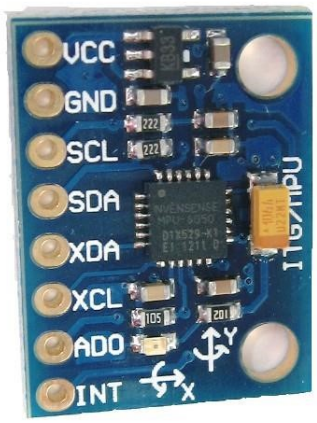


Figure 1. MPU-6050 Gyro + Accelerometer Sensor

3-Axis Gyroscope

The MPU6050 consist of 3-axis Gyroscope with Micro Electro Mechanical System (MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.

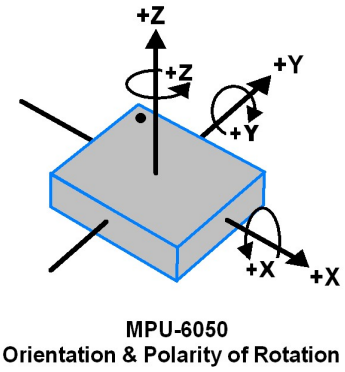


Figure 2. The orientation of 3-axis in Gyroscope

3-Axis Accelerometer

The MPU6050 consist 3-axis Accelerometer with Micro Electro Mechanical (MEMs) technology. It used to detect angle of tilt or inclination along the X, Y and Z axes as shown in below figure.

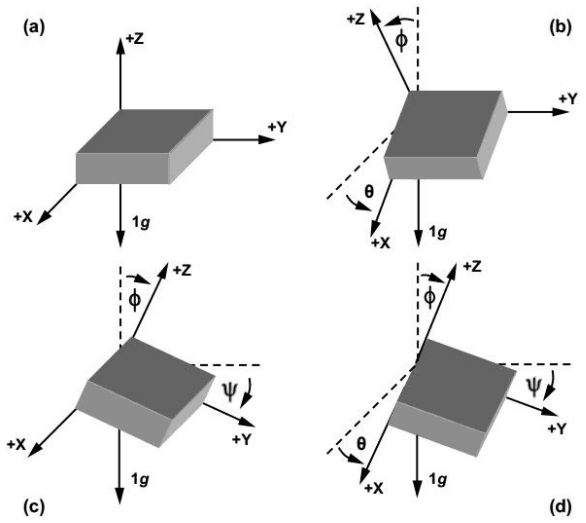


Figure 3. The orientation of 3-axis in Accelerometer

L293D motor driver IC

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

The L293D is a 16 pin IC, with eight pins, on each side, dedicated to the controlling of a motor. There are 2 INPUT pins, 2 OUTPUT pins and 1 ENABLE pin for each motor. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor.

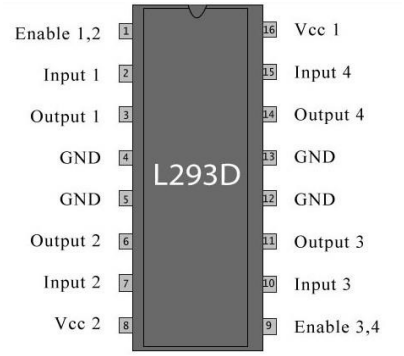


Figure 4. L293D Motor driver IC

Working

The L293D IC receives signals from the microprocessor and transmits the relative signal to the motors. It has two voltage pins, one of which is used to draw current for the working of the L293D

and the other is used to apply a voltage to the motors. The L293D switches its output signal according to the input received from the microprocessor.

RF RECIEVER AND TRANSMITTER MODULE

The communication between transmitter and receiver is using RF modules. A 434 MHz transmitter and receiver pair are used in this project.

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 stronger and more 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.

Figure 5. RF Receiver and Transmitter 434MHz

HT-12E

It is an encoder IC that converts the 4-bit parallel data into serial data in order to transmit over RF link.

An encoder of the 212 series of the HT12E encoder is the integrated circuit. They have been connected with 212 series of decoders for use in the remote-control system application. This is mainly used to intercept the RF and infrared circuit. The selected pair of encoder / decoder should have the same number of addresses and data formats.

Simply put, HT12E converts parallel input into a serial output. It encodes 12-bit parallel data in the serial for transmission via RF transmitter. These 12 bits have been divided into 8 address bits and 4 data bits.

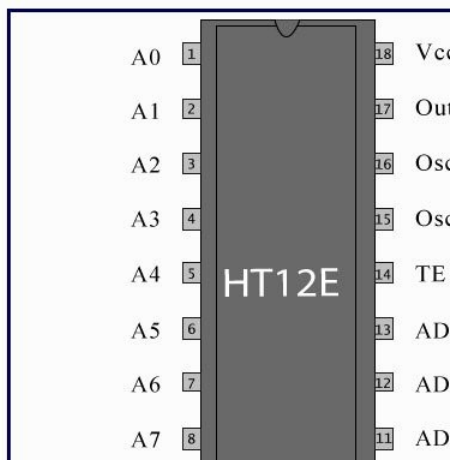


Figure 6. HT-12E Encoder

HT-12D

It is a decoder IC that converts the serial data received by the RF Receiver into 4-bit parallel data. This parallel data can be used to drive the motors.

HT12D is a decoder integrated circuit that relates to the 212 series of decoders. This series of decoders is mainly used for remote control system applications, such as burglar alarm, car door controller, security system etc. It is mainly provided to interface RF and infrared circuit. They have been connected with 212 series of encoder. The selected pair of encoder / decoder should have the same number of addresses and data formats.

In simple words, HT12D converts serial input into parallel output. This serial address and data, which decodes the data received by an RF receiver, into parallel data and sends the output data to the pin.

Serial input data is compared to local addresses three times in a row. Input data code is decoded if no error or unmatched code is found. A valid transmission indicated by a high signal on the VT pin.

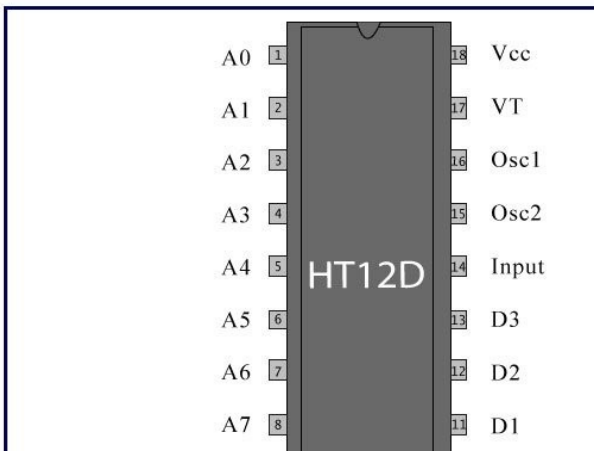


Figure 7. HT-12D Decoder

IC 7805

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

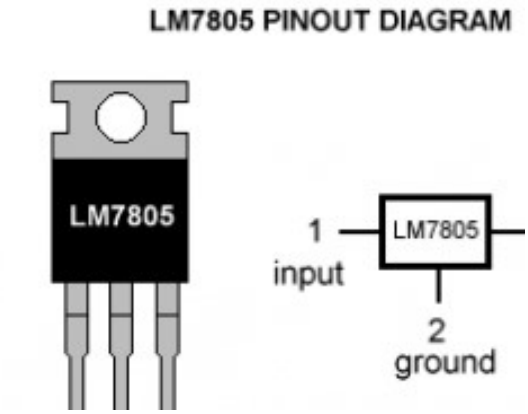


Figure 8. IC 7805

Pin Details of 7805 IC

Pin No.	Pin	Function	Description
1	INPUT	Input voltage (7V-35V)	In this pin of the IC positive unregulated voltage is given in regulation.
2	GROUND	Ground (0V)	In this pin where the ground is given. This pin is neutral for equally the input and output.
3	OUTPUT	Regulated output; 5V (4.8V-5.2V)	The output of the regulated 5V volt is taken out at this pin of the IC regulator.

DC Motor

DC motor is machines that converts direct current electrical energy into mechanical energy. here we have used 300RPM 12v DC motor which is very easy to use and available in standard size Nut and threads on shaft to easily connect and internal threaded shaft for easily connecting it to wheel.

`

Figure 9. DC Motor

2.3 Experimental Analysis

For setting the angle range corresponding to different motion of gesture control mechanical system, we took **comfortable wrist movement data** of people **age range (17-22)** and have considered generalized

possible minimum and maximum values of Sin(angle with X) and Sin(angle with Y) to estimate optimal range of gForceX and gForceY of a specific motion.

Forward Motion

S.No	Name	Age		Accel(Forward motion)			
				SinX	**Angle with X**	SinY	**Angle with Y**
1	Shivang	17		-0.65	-40.54160187	-0.02	-1.145991998
2	Pravar Gupta	18		-0.75	-48.59037789	0.11	6.315315569
3	Rameshwar	18		-0.48	-28.68540201	0.03	1.719131321
4	Dev Meena	19		-0.48	-28.68540201	0.21	12.12235224
5	Vinay	19		-0.42	-24.83458749	-0.01	-0.572967345
6	Adarsh Jain	20		-0.5	-30	0.01	0.572967345
7	Ajay Goel	20		-0.6	-36.86989765	-0.02	-1.145991998
8	Aniket	21		-0.57	-34.75022575	0.01	0.572967345
9	Hrishiraj	21		-0.63	-39.05012254	0.01	0.572967345
10	Pranjal Jain	22		-0.6	-36.86989765	0.08	4.588565736
			Max	-0.48	-28.68540201	0.21	12.12235224
			Min	-0.75	-48.59037789	-0.02	-1.145991998

Backward Motion

S.No	Name	Age		Accel(Backward motion)			
				SinX	**Angle with X**	SinY	**Angle with Y**
1	Shivang	17		0.75	48.59037789	0.15	8.626926559
2	Pravar Gupta	18		0.84	57.14011962	0.11	6.315315569
3	Rameshwar	18		0.78	51.2605754	0.03	1.719131321
4	Dev Meena	19		0.73	46.88639405	0.22	12.70903299
5	Vinay	19		0.75	48.59037789	0.1	5.739170477
6	Adarsh Jain	20		0.55	33.36701297	0.07	4.013987218
7	Ajay Goel	20		0.48	28.68540201	0.25	14.47751219
8	Aniket	21		0.6	36.86989765	0.19	10.9527842
9	Hrishiraj	21		0.76	49.46419789	0.09	5.163607091
10	Pranjal Jain	22		0.95	71.80512766	0.25	14.47751219
			Max	0.95	71.80512766	0.25	14.47751219
			Min	0.48	28.68540201	0.03	1.719131321

Turning Left

S.No	Name	Age		Accel(Left turn)			
				SinX	**Angle with X**	SinY	**Angle with Y**
1	Shivang	17		-0.12	-6.892102579	-0.48	-28.68540201
2	Pravar Gupta	18		-0.1	-5.739170477	-0.62	-38.31613447
3	Rameshwar	18		-0.15	-8.626926559	-0.56	-34.05579774
4	Dev Meena	19		-0.22	-12.70903299	-0.47	-28.03429653
5	Vinay	19		0.15	8.626926559	-0.55	-33.36701297
6	Adarsh Jain	20		-0.1	-5.739170477	-0.58	-35.45054264
7	Ajay Goel	20		-0.08	-4.588565736	-0.58	-35.45054264
8	Aniket	21		0.21	12.12235224	-0.33	-19.26877549
9	Hrishiraj	21		-0.18	-10.36975981	-0.5	-30
10	Pranjal Jain	22		0.1	5.739170477	-0.46	-27.3871075
			Max	0.21	12.12235224	-0.62	-38.31613447
			Min	-0.22	-12.70903299	-0.33	-19.26877549

Turning Right

S.No	Name	Age		Accel(Right turn)			
				SinX	**Angle with X**	SinY	**Angle with Y**
1	Shivang	17		-0.04	-2.292442776	0.87	60.4586395
2	Pravar Gupta	18		0.01	0.572967345	0.47	28.03429653
3	Rameshwar	18		-0.05	-2.865983983	0.81	54.09593142
4	Dev Meena	19		-0.07	-4.013987218	0.85	58.21166938
5	Vinay	19		0.02	1.145991998	0.68	42.84364304
6	Adarsh Jain	20		-0.07	-4.013987218	0.9	64.15806724
7	Ajay Goel	20		-0.05	-2.865983983	0.55	33.36701297
8	Aniket	21		-0.07	-4.013987218	0.84	57.14011962
9	Hrishiraj	21		-0.08	-4.588565736	0.74	47.73141557
10	Pranjal Jain	22		0.018	1.031379731	0.92	66.92608193
			Max	0.02	1.145991998	0.92	66.92608193
			Min	-0.08	-4.588565736	0.47	28.03429653

Stop

S.No	Name	Age		Accel(Stop)			
				SinX	**Angle with X**	SinY	**Angle with Y**
1	Shivang	17		0.12	6.892102579	0.11	6.315315569
2	Pravar Gupta	18		-0.2	-11.53695903	0.11	6.315315569
3	Rameshwar	18		-0.02	-1.145991998	0.05	2.865983983
4	Dev Meena	19		0.04	2.292442776	0.18	10.36975981
5	Vinay	19		0.21	12.12235224	0.12	6.892102579
6	Adarsh Jain	20		-0.04	-2.292442776	0.07	4.013987218
7	Ajay Goel	20		-0.08	-4.588565736	0.16	9.206896221
8	Aniket	21		0.06	3.439812768	0.11	6.315315569
9	Hrishiraj	21		0.01	0.572967345	0.15	8.626926559
10	Pranjal Jain	22		0.12	6.892102579	0.2	11.53695903
			Max	0.21	12.12235224	0.18	10.36975981
			Min	-0.2	-11.53695903	0.05	2.865983983

2.4 Implementation

Transmitter Section

The transmitter section of the system includes the Arduino Nano Board, MPU6050 sensor, HT-12E encoder IC and an RF transmitter.

Transmitter section basically receive data of AngX and AngY from MPU sensor and on the basis of ranges predefined in our Arduino code decide and transfer motion corresponding data to the encoder IC which further transfer encoded data to the RF transmitter system such that it can be transmitted and eventually able to reach receiver section.

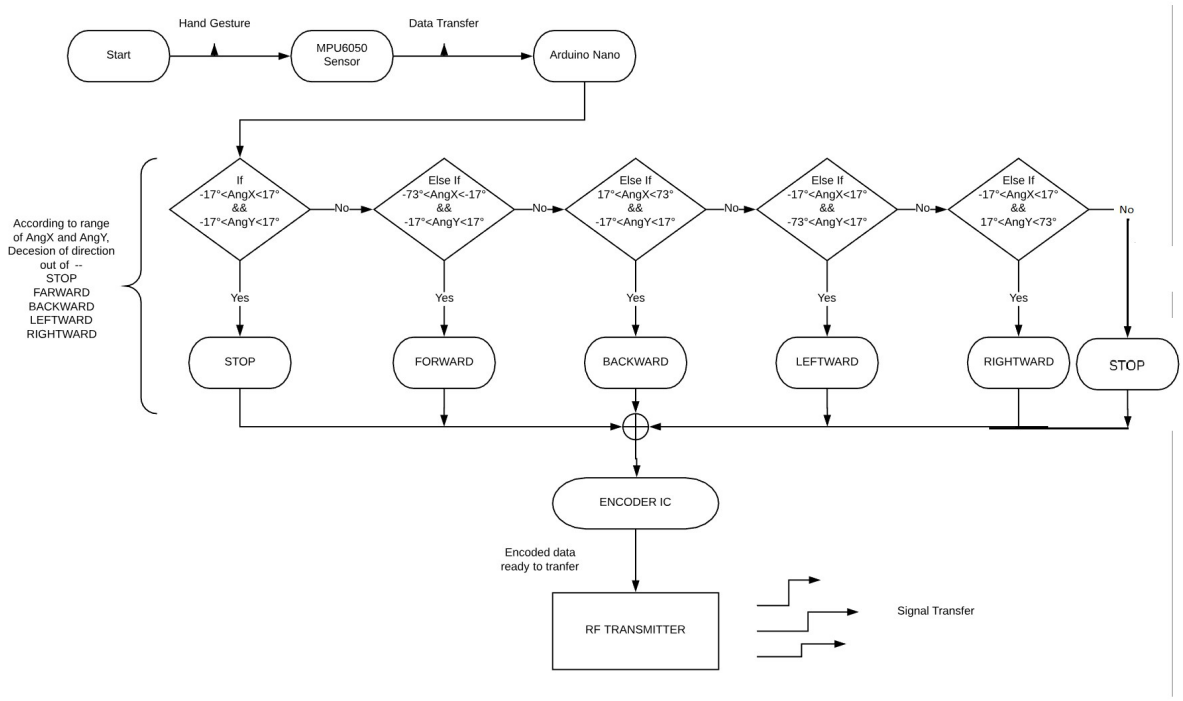


Figure 10. Block diagram of Transmitter section

The communication between Arduino and MPU6050 sensors happens through the I2C interface. Therefore, the SCL and SDA pin of MPU6050 sensor are connected to Arduino Nano's A5 and A4 pins. In addition, we will use the interrupt PIN of MPU6050 and therefore, it is connected to the D2 of Arduino Nano.

HT-12E is an encoder IC which is often connected to the RF transmitter module. It converts 12-bit parallel data to serial data. 12-bit data is divided into addresses and data bits. A0 to A7 (Pin 1 to pin 8) are address bits and they are used for secure transmission of data. These pins can either be left open or connected to the ground (VSS). In this circuit, pin 9 (A0 - A7 and Vss) from pin 1 of HT-12E are connected to the ground.

Pin 10 to 13 (AD8, AD9, AD10 and AD11) are the data pins of HT-12E. They receive 4 words parallel data from the external source, such as a microcontroller (in this case Arduino Nano). They are connected to the Arduino Nano pin D12, D11, D10 and D9, respectively. TE 'transmission is a built-in pin and it is an active lo pin. The data is transmitted until TE 'is less. Therefore, pin 14 (TE ') is also connected to the ground.

Encoder IC has an internal oscillator circuit between pins 16 and 15 (OSC1 and OSC2). A 750K to resistor

is connected between these pins so that the oscillator can be enabled. Dout (pin 17) is the serial data out pin. It is connected to the data in the pin of the RF transmitter. The Arduino Nano and MPU6050 both have 3.3V regulators. Therefore, all VCC pins are connected to a regulated 5V supply.

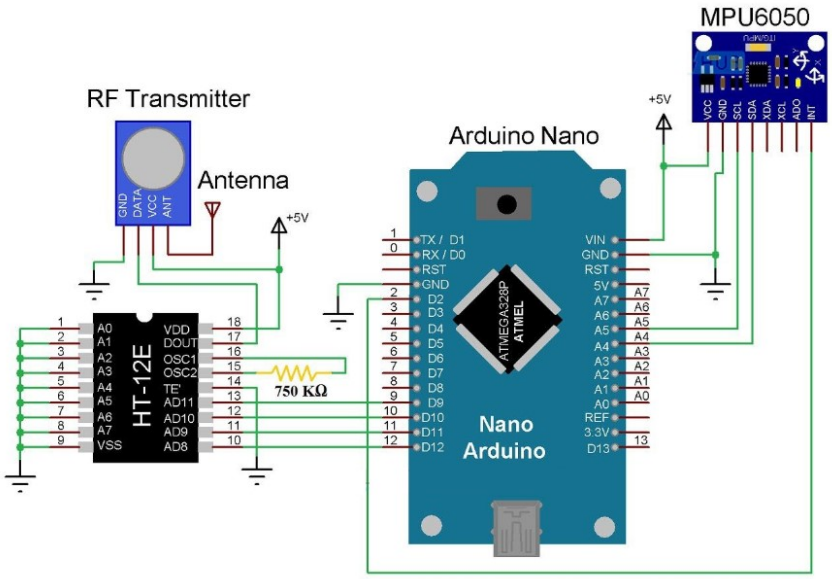
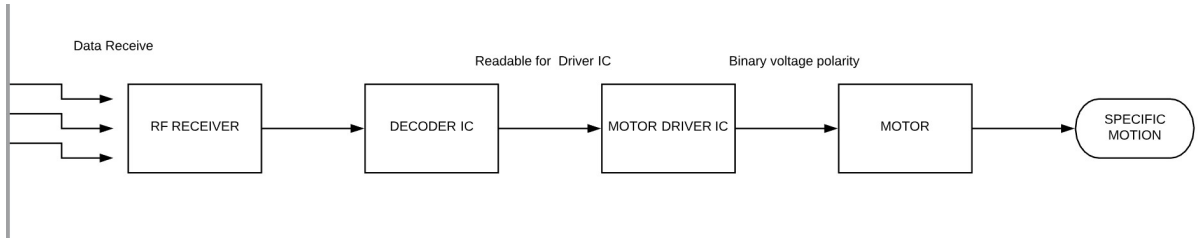


Figure 11. Circuit diagram of Transmitter section

Receiver Section

In the receiver section of the system, there is an RF receiver, HT-12D decoder IC, L293D motor driver IC and a chassis attached to the wheels.

Receiver antenna receives the encoded data transmitted by RF transmitter described above and Forward the data to HT-12D decoder IC which convert data in the readable compatible for L293D motor driver IC which commands and controls our left and right DC motor by providing binary (HIGH , LOW) potential difference input which eventually corresponds to specific require motion of hand gesture controlled mechanical protype.



.

Figure 12. Circuit diagram of Receivr section

The HT-12D decoder is IC which is often connected to the RF receiver. It converts the serial data received by the RF link into parallel data. A0 to A7 (PIN 1 to PIN8) is the address PIN and the encoder must match the address PIN.

Since encoder (HT-12E) address PINs are grounded, the address pin of decoder will also be grounded. Therefore, pins 1 to 9 (A0 - A7 and Vss) are connected to the ground. The serial data decoder IC is given to the DIN (PIN 14) from the RF receiver.

The HT-12D has an internal oscillator and an external blocker of 33K connected is connected between OSC1 and OSC2 (pins 16 and 15). Pin 17 (VT) indicates a valid transmission of data and this PIN will be more when there is a valid data on the data pin. An LED in the series with a resistor, is connected to this pin so that a valid data transmission can be pointed out.

The pins of HT-12D are parallel data from pin 10 to 13 (D8, D9, D10 and D11). They are connected to the input pin of the L293D motor driver IC (Pins 2, 7, 10 and 15 respectively).

The L293D motor driver IC is used to provide the motors for the required current (both back and forth). Pins 1 and 9 are enabled PINs and are connected to VCC (+ 5V) with pin 16 (which is the logic supply). Pins are 3 - 6 and 11 - 14 outputs and are connected to four motors.

Pin 8 motor supply is pin and connected to a separate power supply. Therefore, you will need two batteries in the receiver section; One for the other motors for the circuit.

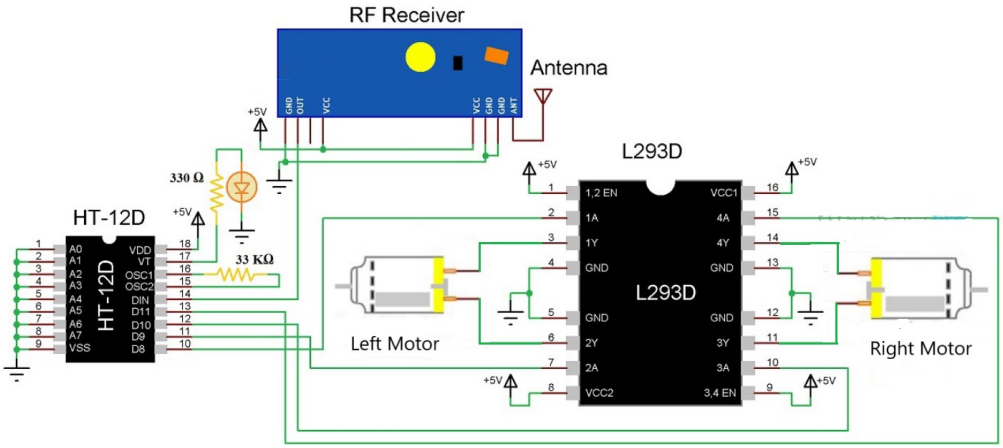


Figure 13. Circuit diagram of Receiver section

3. Results and Conclusion

We achieved our objective without any hurdles i.e. the control of a system using gestures. The system is showing proper responses with a minimum time response of 1/1000 seconds whenever we move our hand wrist. Different Hand gestures to make the system move in specific directions are as follow:

Figure 14. Stop

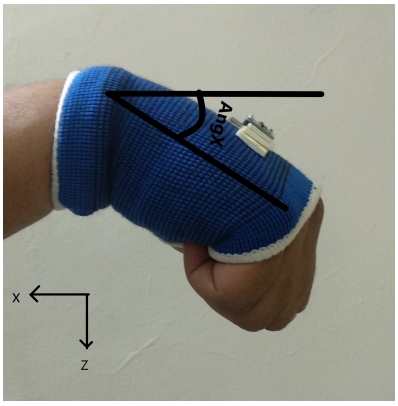


Figure 15. Forward Motion

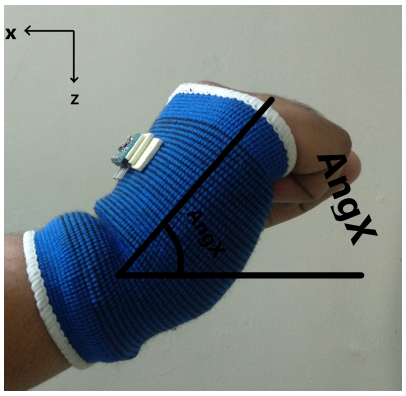


Figure 16. Backward Motion

Figure 17. Left Turn

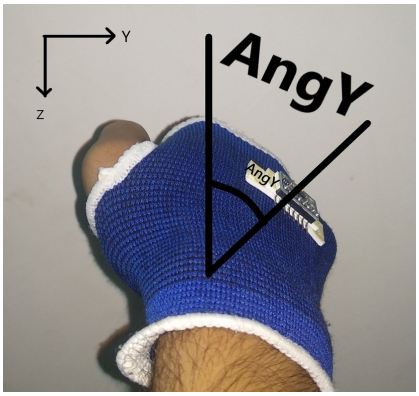


Figure 18. Right Turn

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

Direction	MPU-6050 Orientation
Forward	X-
Backward	X+
Left	Y-
Right	Y+
Stop	REST

On the Basic of the experimental analysis of the wrist movement of the persons belonging to 17-22 age, we have concluded the angle corresponds to the motion of our gesture control mechanical system. We have taken +10° to -10° extra, as if someone has another comfortable position of his/her hand.

- For **Stop**, we have set the angle with the X axis between -17° to 17°, and with Y axis it is -17° to 17°
- For **forward motion**, we have set the angle with the X axis between -73° to -17°, and with Y axis it is -17° to 17°
- For **Backward Motion**, we have set the angle with the X axis between 17° to 73°, and with Y axis it is -17° to 17°
- For **Turning Left**, we have set the angle with the X axis between -17° to 17°, and with Y axis it is -73° to -17°
- For **Turning Right**, we have set the angle with the X axis between -17° to 17°, and with Y axis it is 17° to 73°
- Any angle crossing max angle limit (±73 deg) also lead to **Stop**.

Motion	Max. SinX	Min. SinX	Range of gForceX	Angle with X	Max. SinY	Min. SinY	Range of gForceY	Angle with y
Stop	0.21	-0.2	(-0.3,0.3)	(-17°,17°)	0.18	0.05	(-0.3,0.3)	(-17°,17°)
Forward	-0.48	-0.75	(-0.95,-0.3)	(-73°,-17°)	0.21	-0.02	(-0.3,0.3)	(-17°,17°)
Backward	0.95	0.48	(0.3,0.95)	(17°,73°)	0.25	0.03	(-0.3,0.3)	(-17°,17°)
Left Turn	0.21	-0.22	(-0.3,0.3)	(-17°,17°)	-0.62	-0.33	(-0.95,-0.3)	(-73°,-17°)
RightTurn	0.02	-0.08	(-0.3,0.3)	(-17°,17°)	0.92	0.47	(0.3,0.95)	(17°,73°)

4. Prototype

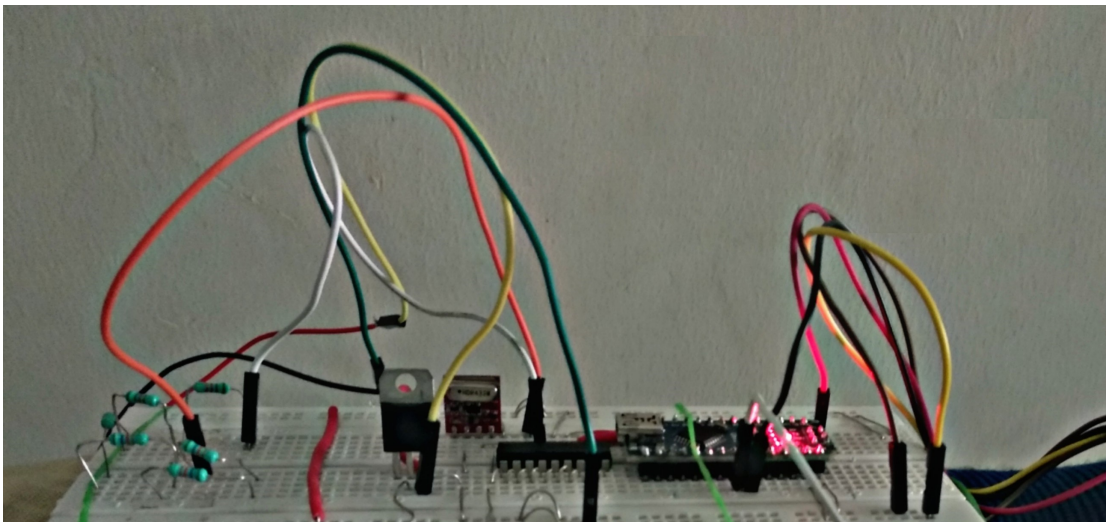


Figure 19. Transmitter circuit Prototype

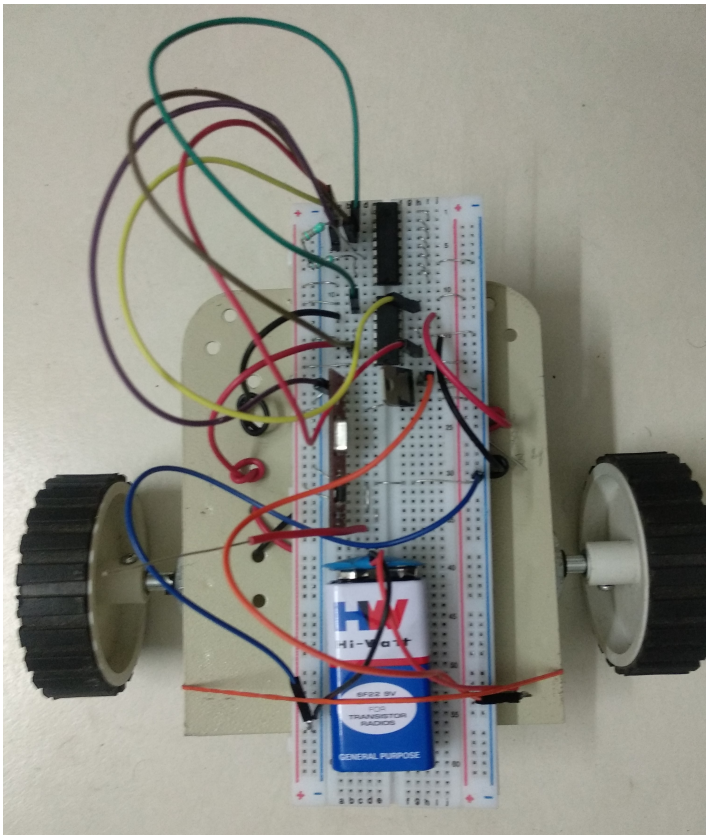


Figure 20. Receiver circuit Prototype

5. Future Work

- Speed of motors according to the degree of angle with the axis as now we have only given binary output of HIGH and LOW corresponding to move to stop.
- Security alarm system as if a person suddenly tilts his hand to the maximum degree, the speed will be increased immediately and may cause to the chances of mishap.
- Make the circuit small and compatible one
- Flyable system control using gesture.

Appendix

```
#include <Wire.h>

long accelX, accelY, accelZ;

float gForceX, gForceY, gForceZ;

float Max_ang, Min_ang ;

long gyroX, gyroY, gyroZ;

float rotX, rotY, rotZ;

int out1=9;     //output1 for HT12E IC

int out2=10;     //output1 for HT12E IC

int out3=11;    //output1 for HT12E IC

int out4=12;    //output1 for HT12E IC

void setup() {

  Serial.begin(9600);

  Wire.begin();

  setupMPU();

  Min_ang = 0.30;   // Min is Minimum angle (17 DEG)

  Max_ang = 0.95;   // Max is Maximum angle (73 DEG)

  pinMode(out1,OUTPUT);

  pinMode(out2,OUTPUT);

  pinMode(out3,OUTPUT);

  pinMode(out4,OUTPUT);

}

void loop() {

  recordAccelRegisters();

  recordGyroRegisters();
```

```
  printData();
  delay(100);
}
void setupMPU(){
  Wire.beginTransmission(0b1101000); //This is the I2C address of the MPU (b1101000/b1101001 for
AC0 low/high datasheet sec. 9.2)
  Wire.write(0x6B); //Accessing the register 6B - Power Management (Sec. 4.28)
  Wire.write(0b00000000); //Setting SLEEP register to 0. (Required; see Note on p. 9)
  Wire.endTransmission();
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x1B); //Accessing the register 1B - Gyroscope Configuration (Sec. 4.4)
  Wire.write(0x00000000); //Setting the gyro to full scale +/- 250deg./s
  Wire.endTransmission();
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x1C); //Accessing the register 1C - Acccelerometer Configuration (Sec. 4.5)
  Wire.write(0b00000000); //Setting the accel to +/- 2g
  Wire.endTransmission();
}
void recordAccelRegisters() {
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x3B); //Starting register for Accel Readings
  Wire.endTransmission();
  Wire.requestFrom(0b1101000,6); //Request Accel Registers (3B - 40)
  while(Wire.available() < 6);
  accelX = Wire.read()<<8|Wire.read(); //Store first two bytes into accelX
```

```
  accelY = Wire.read()<<8|Wire.read(); //Store middle two bytes into accelY
  accelZ = Wire.read()<<8|Wire.read(); //Store last two bytes into accelZ
  processAccelData();
}
void processAccelData(){
  gForceX = accelX / 16384.0;
  gForceY = accelY / 16384.0;
  gForceZ = accelZ / 16384.0;
}
void recordGyroRegisters() {
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x43); //Starting register for Gyro Readings
  Wire.endTransmission();
  Wire.requestFrom(0b1101000,6); //Request Gyro Registers (43 - 48)
  while(Wire.available() < 6);
  gyroX = Wire.read()<<8|Wire.read(); //Store first two bytes into accelX
  gyroY = Wire.read()<<8|Wire.read(); //Store middle two bytes into accelY
  gyroZ = Wire.read()<<8|Wire.read(); //Store last two bytes into accelZ
  processGyroData();
}
void processGyroData() {
  rotX = gyroX / 131.0;
  rotY = gyroY / 131.0;
  rotZ = gyroZ / 131.0;
}
```

```
void printData() {
 if ( (gForceX>-Min_ang && gForceX<Min_ang ) && ( gForceY>-Min_ang && gForceY<Min_ang )) //stop
 {
    digitalWrite(out1,LOW);
    digitalWrite(out2,LOW);
    digitalWrite(out3,LOW);
    digitalWrite(out4,LOW);
 }
 else
 {
  if ((gForceX>-Min_ang && gForceX<Min_ang ) && (gForceY>Min_ang && gForceY<Max_ang)) //right
   {
     digitalWrite(out1,HIGH);
     digitalWrite(out2,LOW);
     digitalWrite(out3,HIGH);
     digitalWrite(out4,LOW);
   }
  if ((gForceX>-Min_ang && gForceX<Min_ang ) && (gForceY>-Max_ang && gForceY<-Min_ang )) //left
   {
     digitalWrite(out1,LOW);
     digitalWrite(out2,HIGH);
     digitalWrite(out3,LOW);
     digitalWrite(out4,HIGH);
   }
```

```
   if ( (gForceX>-Max_ang && gForceX<-Min_ang ) && (gForceY>-Min_ang && gForceY<Min_ang ))  //
farward
    {
      digitalWrite(out2,LOW);
      digitalWrite(out1,HIGH);
      digitalWrite(out4,HIGH);
      digitalWrite(out3,LOW);
    }
       if ( (gForceX>Min_ang && gForceX<Max_ang) && ( gForceY>-Min_ang && gForceY<Min_ang )
)//back ward
    {
      digitalWrite(out2,HIGH);
      digitalWrite(out1,LOW);
      digitalWrite(out4,LOW);
      digitalWrite(out3,HIGH);
    }
   if ( gForceX>Max_ang || gForceX<-Max_ang || gForceY>Max_ang || gForceY<-Max_ang )
    {
      digitalWrite(out1,LOW);
      digitalWrite(out2,LOW);
      digitalWrite(out3,LOW);
      digitalWrite(out4,LOW);
    }
 }
 Serial.print("Gyro (deg)");
 Serial.print(" X=");
```

```
  Serial.print(rotX);
  Serial.print(" Y=");
  Serial.print(rotY);
  Serial.print(" Z=");
  Serial.print(rotZ);
  Serial.print(" Accel (g)");
  Serial.print(" X=");
  Serial.print(gForceX);
  Serial.print(" Y=");
  Serial.print(gForceY);
  Serial.print(" Z=");
  Serial.println(gForceZ);
  delay(500);
}
```

References:

1. Gesture control: - https://www.youtube.com/watch?v=rejZmqRrKMc

2.Arduino Basics: -

https://www.youtube.com/watch?v=d8_xXNcGYgo&list=PLGs0VKk2DiYx6CMdOQR_hmJ2NbB4mZQn-

3. MPU-6050: - https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module

4. L239D: - https://www.rakeshmondal.info/L293D-Motor-Driver

5. HT-12D: - https://www.engineersgarage.com/electronic-components/ht12d-datasheet

6. HT-12E: -https://www.engineersgarage.com/electronic-components/ht12e

7. MPU-6050 Standard Library: - https://github.com/jarzebski/Arduino-MPU6050