Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

Course: EEE 318

Course Title: Control System I Laboratory

Project Report on

GESTURE CONTROLLED ROBOT

Group 2

Level 3, Term 2

Section: A2

Submitted By:

Name	ID
Ashikur Rahman	1806050
Fardin Ahmed	1806056
Sabbir Ahmed	1806057
Yeaz Mahmud	1806065

Contents

ABSTRACT

This project aims to find out if hand gestures can reasonably be used to control the movement of a robot or car. This is an easy, user-friendly way to interact with robotic systems and robots. There are two main components of the system: The accelerometer depends upon the gestures of the hand. Through accelerometer, a passage of data signal is received and it is processed with the help of Arduino microcontroller. The microcontroller gives command to the robot to move in the desired direction.

INTRODUCTION

Robots are playing an important role in automation across all the sectors like construction, military, medical, manufacturing, etc. The integration of more and more functionality into the human machine interface (HMI) of robots increases the complexity of device handling. Thus, optimal use of different human sensory channels is an approach to simplify the interaction with robotic devices. Hence, instead of using joystick or physical controller with buttons, hand gestures can be used to control a robot. This device can be very useful for surveillance, military operations and industrial grade robotic arms for physically challenged individuals.

COMPONENTS USED

- Arduino Uno
- Two Arduino Nano
- Two SX1278 LoRa Module
- Two 18650 lithium-ion battery
- Battery storage box
- MPU-6050 triple axis accelerometer and gyro breakout
- Two half size breadboards
- L298N DC motor driver
- Two DC motors
- Wheels
- Ball caster wheel
- Toggle Switch
- PVC board
- Jumper Wires

METHODOLOGY

The whole project consists of three major parts:

1. Transmitter
2. Receiver
3. Car Driving Module

Transmitter

The transmitter module consists of one Arduino Nano, one MPU-6050 triple axis accelerometer and one LoRa module. This circuit can be attached to human hand and it tracks the movement of the hand. Then using the microprocessor, it processes the sensor data and send necessary commands with the help of LoRa communication module. LoRa stands for Long Range Radio and is mainly targeted for M2M and IoT networks. We are using 433MHz range of LoRa for our communication.

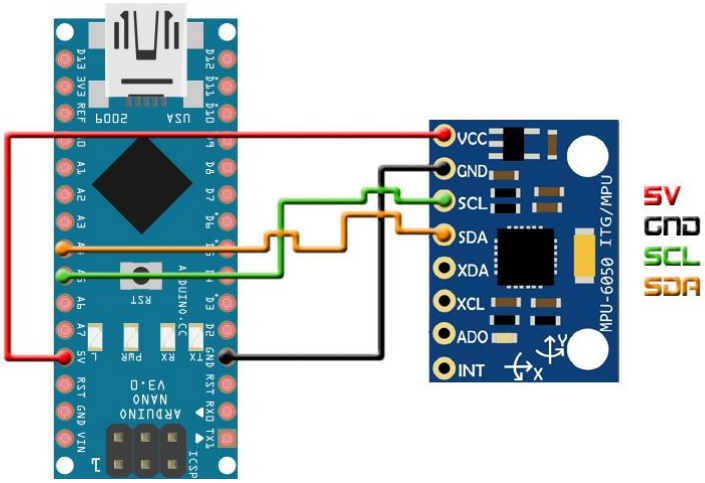


Fig 1: Interfacing MPU-6050 sensor module with Arduino Nano

The MPU-6050 accelerometer and gyro sensor work with I2C communication protocol. Hence, using SCL and SDA pins all sensor data can be accessed using microcontroller. Here we are using accelerometer reading of two-axis (X and Y). In which X axis controls the forward and reverse speed of the car and Y axis controls the turning mechanism.

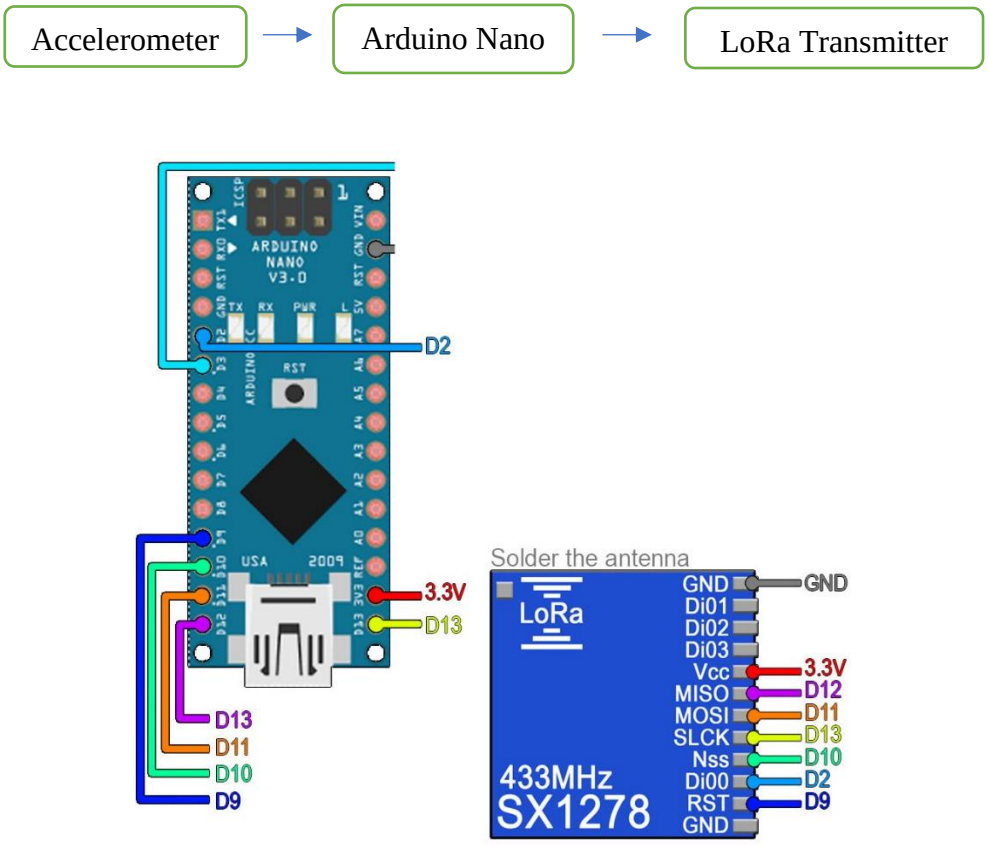


Fig 2: Interfacing LoRa with Arduino Nano

Code for transmitter module

```
#include <MPU6050_tockn.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <LoRa.h>

int counter = 0;
char *data;
float accX=0;
float accY=0;
MPU6050 mpu6050(Wire);
long timer = 0;

void setup() {
  Serial.begin(9600);
  while (!Serial);
  Serial.println("LoRa Sender");
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
```

```
  LoRa.setTxPower(20);
  Wire.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);
}

void loop() {
  mpu6050.update();
  if(millis() - timer > 300){
    accX = mpu6050.getAccX();
    accY = mpu6050.getAccY();
    if(accX>0.50)
    {
      data="B";
      Serial.write('B');
      LoRa.beginPacket();
      LoRa.print("B");
      LoRa.endPacket();
    }
    else if(accX<-0.50)
    {
      data="F";
      Serial.write('F');
      LoRa.beginPacket();
      LoRa.print("F");
      LoRa.endPacket();
    }
    else if(accY<-0.50)
    {
      data="R";
      Serial.write('R');
      LoRa.beginPacket();
      LoRa.print("R");
      LoRa.endPacket();
    }
    else if(accY>0.50)
    {
      data="L";
      Serial.write('L');
      LoRa.beginPacket();
      LoRa.print("L");
      LoRa.endPacket();
    }
    else
    {
      data="S";
      Serial.write('S');
      LoRa.beginPacket();
```

```
      LoRa.print("S");
      LoRa.endPacket();
    }
    delay(200);
  }
}
```

Receiver

The receiver circuit is an intermediate part which receives the signals from the transmitter and sends it using wires to the card driving microcontroller(Arduino Uno). It consists of an Arduino Nano and a LoRa receiver module. Interfacing procedure is similar to the one in the transmitter circuit. After receiving and processing the incoming commands, necessary data is sent to the car driving circuit using three pins/wires.

Code for receiver module

```
#include <SPI.h>
#include <LoRa.h>
int A=3;
int B=4;
int C=5;
int D;

void setup() {
  pinMode(A,OUTPUT);
  pinMode(B,OUTPUT);
  pinMode(C,OUTPUT);
  digitalWrite(A,LOW);
  digitalWrite(B,LOW);
  digitalWrite(C,LOW);
  Serial.begin(9600);
  while (!Serial);
  Serial.println("LoRa Receiver");
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
```

```
}

void loop() {
  int packetSize = LoRa.parsePacket();

  if (packetSize) {
    while (LoRa.available()) {
      D=LoRa.read();
      Serial.print(D);
      Serial.print("\n");

      if(D==83) //Stop 000
      {
        digitalWrite(A,LOW);
        digitalWrite(B,LOW);
        digitalWrite(C,LOW);
      }
      else if(D==70) //Forward 111
      {
        digitalWrite(A,HIGH);
        digitalWrite(B,HIGH);
        digitalWrite(C,HIGH);
      }
      else if(D==66) //Backward 011
      {
        digitalWrite(A,LOW);
        digitalWrite(B,HIGH);
        digitalWrite(C,HIGH);
      }
      else if(D==76) //Left 110
      {
        digitalWrite(A,HIGH);
        digitalWrite(B,HIGH);
        digitalWrite(C,LOW);
      }
      else if(D==82) //Right 101
      {
        digitalWrite(A,HIGH);
        digitalWrite(B,LOW);
        digitalWrite(C,HIGH);
      }
      else
      {
      }
    }
  }
  // Stop = 83, Forward = 70, Backward = 66, Left = 76, Right = 82
}
```

Car Driving Module

This circuit receives the incoming data from the receiver circuit and drives the car according to those data. This circuit consists of one Arduino Uno, one L298N motor controller board and two DC motors.

L298N is a two channel motor driver which can control the speed and direction of two motors according to the controlling signals coming from Arduino Uno. It uses H-bridge mechanism to control the direction of the motors.

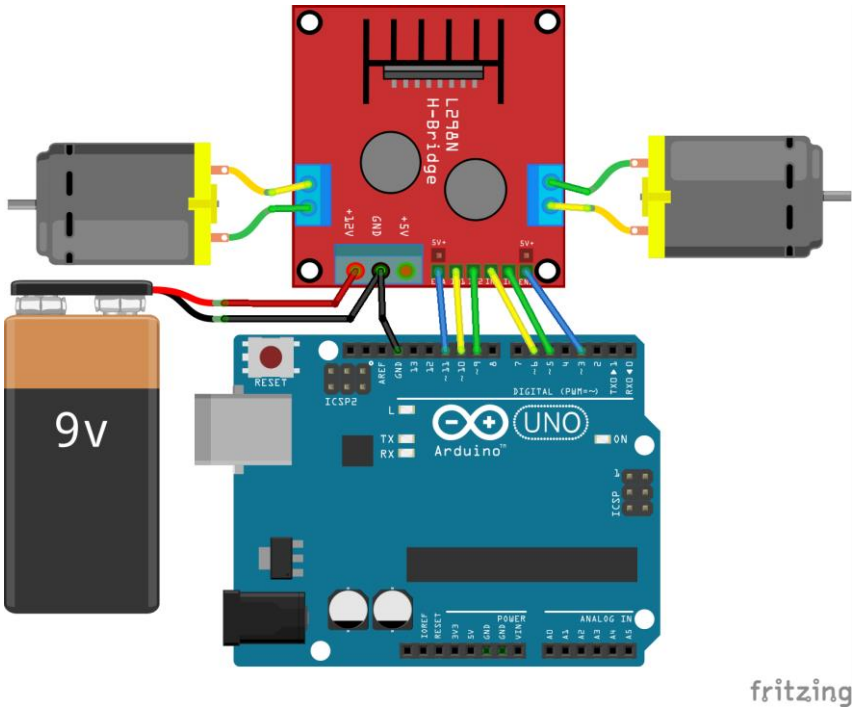


Fig 3: Car driving circuit

Code for car driving module

```
#include <SoftwareSerial.h>

#define ENA 10
#define IN1 9
#define IN2 8
#define IN3 6
#define IN4 5
#define ENB 3
#define A 4
#define B 7
#define C 11
void wheel(int LEFT, int RIGHT);
void setup() {
```

```
  // set the data rate for the SoftwareSerial port
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(A, INPUT);
  pinMode(B, INPUT);
  pinMode(C, INPUT);
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}

void loop() {
    if ((digitalRead(A)==
HIGH)&&(digitalRead(B)==HIGH)&&(digitalRead(C)==HIGH)) //Forward 111
      wheel(1, 1);
    else if ((digitalRead(A)==
LOW)&&(digitalRead(B)==HIGH)&&(digitalRead(C)==HIGH))  //Backward 011
      wheel(-1, -1);
    else if ((digitalRead(A)==
HIGH)&&(digitalRead(B)==HIGH)&&(digitalRead(C)==LOW))  //Left 110
      wheel(-1, 1);
    else if ((digitalRead(A)==
HIGH)&&(digitalRead(B)==LOW)&&(digitalRead(C)==HIGH))  //Righ 101
      wheel(1, -1);
    else if ((digitalRead(A)==
LOW)&&(digitalRead(B)==LOW)&&(digitalRead(C)==LOW))  //Stop 000
      wheel(0, 0);
}
```

Code of wheel.ino function

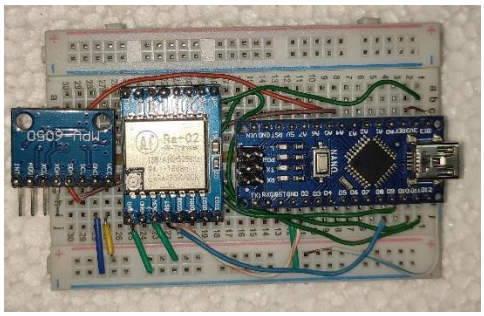
```
void wheel(int left, int right)

{ if (left == 1) {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
  }
  else  if (left == -1) {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
  }
```

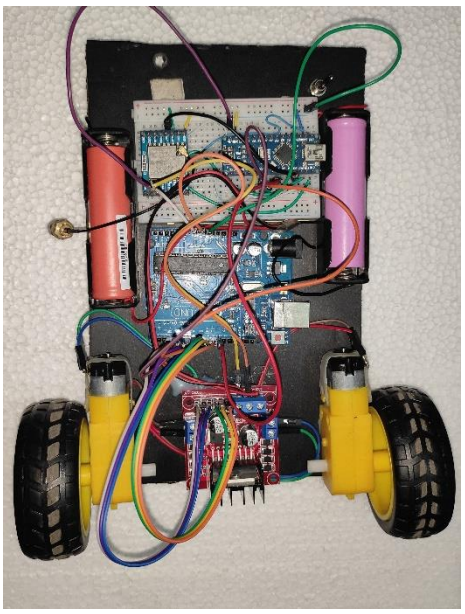
```
  else if (left == 0) {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
  }
  if (right == 1) {
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
  }
  else  if (right == -1) {
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
  }
  else if (right == 0) {
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
  }
}
```

PROJECT OUTPUT

The car was able to follow all the hand movements without any error and with minimum amount of delay.



(a)



(b)

Fig 4: Final circuits for (a) Transmitter and (b) Car including receiver

APPLICATIONS

On practical cases, gesture control method can be implemented in many sectors to make human life easier. Already this method is used in some sectors like

- In military applications to operate robots
- In medical applications for the purpose of surgery with the help of robots
- In the construction field
- In the industries to control robotic hands, trolly and lifts

As more precise controlling is possible using hand gesture compared to the other currently used methods, this technology can be used in every sectors where precision is needed in control system.

LIMITATIONS

- Our car can only move in four major directions: Left, Right, Forward and Backward. Other diagonal directions like forward-left, forward-right couldn't be implemented as the sensor's data isn't that precise.
- Due to hardware limitations, there is a delay of 200 milliseconds between two consecutive commands.
- As MPU-6050 is an entry-level, low-cost accelerometer sensor, readings are not that precise.

CONCLUSION

The project "Gesture controlled robot" was difficult yet very interesting project to carry on. Our results in the project are very promising and we were able to control the robot using hand gesture. With more powerful equipment and more precise sensors, this can be successfully implemented in different sectors. As implementations of robotics is increasing enormously with time, this kind of controlling system can offer a great alternative for other available manual controls.