**Final Progressive Report (HAND GESTURE-CONTROLLED BOT)**

1. TASKS ACCOMPLISHED: -
2. I bought all the required components and checked them for proper functioning.
3. I assembled the components for the receiver part i.e., the bot.
4. I assembled the components for the transmitter part i.e., the hand-gesture part.
5. I completed the coding for both the transmitter and the receiver sections using Arduino IDE using the libraries “Wire.h” and “RH\_ASK.h”.
6. I fixed some of the problems(listed below) and completed the Hand Gesture-Controlled Bot using MPU6050.
7. CURRENTLY WORKING ON: -

The bot is complete in all its functions.

* 1. I am trying to boost the range of transmission.
  2. I am learning other ways than the one I used to read the values from MPU6050.

1. TASKS TO BE TAKEN-UP: -

The bot is completely made. I will try to make an antenna to boost the range of transmission for the rf transmitter receiver module. Also, for smoother power, I will use 4xAA batteries or DC adapter in place of 9V battery I currently use.

1. TECHNICAL PROBLEMS FACED: -
   * + 1. The Arduino IDE wasn't recognizing the arduino uno and nano boards respectively. It was solved by installing the proper drivers and updating the driver for the boards.
       2. After uploading the final code, the transmitter section was working fine but the receiver bot wasn’t working. I used the transmitter receiver example sketch from Arduino IDE to find out my module wasn’t working. So, I had to buy a new one.
       3. Again, my code was correct but the bot was still not moving, I re-checked my connections and found the motor driver wasn’t grounded with the Arduino and fixed that.
       4. The Arduino IDE gave the error “avrdude: programmer is not responding” but on disconnecting and reconnecting multiple times it got fixed on its own.
2. Working Video: - <https://drive.google.com/drive/folders/1FLY44BMYSc05NYKM9rP482HY-5_lNU0U?usp=sharing>
3. Code: -

**Transmitter (Uploaded on Arduino Nano board): -**

//Transmitter Code

#include <Wire.h> //Include the Wire library

#include <RH\_ASK.h> //Include the RH\_ASK library

RH\_ASK rf\_driver; //create object

char\* msg="100"; //Store the msg to be sent

void setup() {

Serial.begin(9600); //For debugging

Wire.begin();

Wire.beginTransmission(0x68); //0x68 is I2C slave address of MPU6050

Wire.write(0x6B); //Accessing register 6B - Power Management(comes up in SLEEP mode upon power-up)

Wire.write(0x00); //Setting SLEEP register to 0

Wire.endTransmission(true); //true sends a stop msg

if (!rf\_driver.init()) //Check initialisation of object

Serial.println("init failed");

}

void loop() {

Wire.beginTransmission(0x68); //Starts transmission with I2C slave address of MPU6050

Wire.write(0x3B); //Starting register for Accel Readings

Wire.endTransmission();

Wire.requestFrom(0x68, 6); //Request 6 Accel Registers(3B-40)

//In 16 bit variables, Wire.read reads 8 bits from the starting register

//We use bitwise left shift and bitwise OR to merge the value read from the next register

//The merged values are divided by 163.84 which is Accel senstivity to get meaningful data

int16\_t accX = ((Wire.read()<<8)|Wire.read())/163.84;

int16\_t accY = ((Wire.read()<<8)|Wire.read())/163.84;

int16\_t accZ = ((Wire.read()<<8)|Wire.read())/163.84;

Wire.endTransmission();

//Print the accelerometer values on the serial monitor

Serial.print(" a-x :");

Serial.print(accX);

Serial.print(" a-Y :");

Serial.print(accY);

Serial.print(" a-Z :");

Serial.println(accZ);

delay(500);

//Using the accelerometer values to store instruction in msg

if(accY>=20 && accY<=60)

{

msg="Backward";

Serial.println("Backward"); //Prints on serial monitor

}

else if(accY<=-20 && accY>=-60)

{

msg = "Forward";

Serial.println("Forward"); //Prints on serial monitor

}

else if(accX>=20 && accX<=60)

{

msg="Left";

Serial.println("Left"); //Prints on serial monitor

}

else if(accX<=-20 && accX>=-60)

{

msg="Right";

Serial.println("Right"); //Prints on serial monitor

}

else

{

Serial.println("Stop"); //Prints on serial monitor

msg="Stop";

}

delay(200);

rf\_driver.send((uint8\_t \*)msg, strlen(msg)); // Transmits the msg to receiver

rf\_driver.waitPacketSent(); //Waits for the full msg to be sent

delay(200);

}

**Receiver (Uploaded on Arduino Uno board):** -

//Receiver Code

#include <RH\_ASK.h> //Include the RH\_ASK library

//Defining motor driver pins

int m1 = 2;

int m2 = 3;

int m3 = 4;

int m4 = 5;

RH\_ASK rf\_driver; //create object

String str; //To store the received msg

void setup()

{

Serial.begin(9600); //For debugging

if (!rf\_driver.init()) //Check initialisation of object

Serial.println("init failed");

//Configures the pins to behave as output

pinMode(m1, OUTPUT);

pinMode(m2, OUTPUT);

pinMode(m3, OUTPUT);

pinMode(m4, OUTPUT);

}

void loop()

{

uint8\_t buf[1]; //max size of the buffer

uint8\_t buflen = sizeof(buf); //Create a 1 byte char buffer

if (rf\_driver.recv(buf, &buflen)) //returns true if valid msg copied to buf

{

str=(char\*)buf; //stores the received msg

Serial.println(str); //Prints the msg on serial monitor

}

delay(200);

if(str=="F")

{

//Writing the motor driver pins to move forward

digitalWrite(m1,LOW);

digitalWrite(m2,HIGH);

digitalWrite(m3,HIGH);

digitalWrite(m4,LOW);

Serial.println("Forward"); //Prints on serial monitor

}

else if(str=="B")

{

//Writing the motor driver pins to move backward

digitalWrite(m1,HIGH);

digitalWrite(m2,LOW);

digitalWrite(m3,LOW);

digitalWrite(m4,HIGH);

Serial.println("Backward"); //Prints on serial monitor

}

else if(str=="L")

{

//Writing the motor driver pins to move left

digitalWrite(m1,HIGH);

digitalWrite(m2,LOW);

digitalWrite(m3,LOW);

digitalWrite(m4,LOW);

Serial.println("Left"); //Prints on serial monitor

}

else if(str=="R")

{

//Writing the motor driver pins to move right

digitalWrite(m1,LOW);

digitalWrite(m2,LOW);

digitalWrite(m3,LOW);

digitalWrite(m4,HIGH);

Serial.println("Right"); //Prints on serial monitor

}

else if(str=="S")

{

//Writing the motor driver pins to stop

digitalWrite(m1,LOW);

digitalWrite(m2,LOW);

digitalWrite(m3,LOW);

digitalWrite(m4,LOW);

Serial.println("Stop"); //Prints on serial monitor

}

}

THANK YOU.

RITIK RAJ

BTECH/10396/20