HAND GESTURE CONTROLLED ROBOT

(Bluetooth-Joystick and Gravity Sensor)

by

NAGAVISHNESH KS 21BRS1312
SUMANTH MN 21BRS1504
LAVANYA 21BRS1187
SIYAMALA 21BRS1710
RAHUL K 21BRS1619

Project Report

submitted

to

Dr. SWAGATIKA MOHANTY

SCHOOL OF MECHANICAL ENGINEERING

in partial fulfilment of the requirements for the course of

BCSE421L Robotics:Kinematics,Dynamics and Motion Control

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report entitled "**HAND GESTURE CONTROLLED ROBOT**" is a bonafide work of NAGAVISHNESH KS 21BRS1312 ,SUMANTH MN 21BRS1504 LAVANYA 21BRS1187 SIYAMALA 21BRS1710 RAHUL K 21BRS1619 carried out the Project work under my supervision and guidance for **BCSE421L Robotics:Kinematics,Dynamics and Motion Control.**

Dr. SWAGATIKA MOHANTY

Assistant Professor Senior Grade,

School of Mechanical Engineering (SME),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

The Bluetooth controlled robot using the RemoteXY app and gravity sensor is an exciting project that combines Bluetooth communication, smartphone-based control systems, and the use of a gravity sensor. The system allows users to control the movement of a robot using their smartphone, making it a user-friendly and engaging project.

The system uses Bluetooth communication to establish a connection between the robot and the smartphone. The RemoteXY app provides a graphical interface on the smartphone that allows the user to control the robot's movements. The gravity sensor is used to detect the orientation of the smartphone, which is used to control the robot's movements. If the user tilts the smartphone to the left, the robot will move left, and if the user tilts it to the right, the robot will move right.

The system is implemented using an Arduino board, a Bluetooth module, and two motors to control the robot's movements. The RemoteXY app is used to create a graphical interface on the smartphone that allows the user to control the robot's movements by tilting the phone in different directions.

Overall, the Bluetooth controlled robot using the RemoteXY app and gravity sensor is a fun and engaging project that demonstrates the potential of smartphone-based control systems. It is a great project for beginners who are interested in robotics and programming and want to learn about Bluetooth communication and sensor-based control systems.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Swagatika Mohanty,** Assistant Professor Senior Grade, School of Mechanical Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Ganesan R,** Dean of School of Computer Science and Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We express our thanks to our Head of the Department **Dr. Rekha D** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

TABLE OF CONTENTS

CHAPTER 1- Introduction

Introduction

Hand gesture-controlled robots are becoming increasingly popular due to their intuitive and user-friendly interface. These robots are controlled by detecting and interpreting hand gestures made by the user, making them an ideal candidate for use in a wide range of applications, including healthcare, manufacturing, and entertainment. In this robotics project, we have developed a hand gesture-controlled robot using Arduino microcontroller, which allows for precise and reliable control of the robot's movements.

The robot is equipped with sensors that detect hand gestures and transmit signals to the Arduino, which then processes the signals and sends commands to the robot's actuators. In this report, we will discuss the design and implementation of the hand gesture-controlled robot, including the hardware and software components, as well as the results of our testing and evaluation.

The goal of this project is to demonstrate the potential of hand gesture-controlled robots in real-world applications, and to provide a foundation for future research and development in this area.

1.1) Background

In recent years, the use of hand gesture recognition technology has emerged as a promising solution for controlling robots in a more intuitive and user-friendly manner. Arduino microcontroller is a popular choice for controlling robotic systems due to its flexibility and low cost. This report aims to explore the potential of combining these technologies by designing and implementing a hand gesture-controlled robot using Arduino.

The report will provide a detailed description of the hardware and software components used in the project, as well as the design and implementation of the hand gesture recognition system. The project's objective is to demonstrate the potential of this technology in real-world applications, providing valuable insights into the future of human-robot interaction.

1.2) Motivation

The motivation behind this report is to explore the potential of hand gesture-controlled robots as a more intuitive and user-friendly interface for controlling robotic systems. By utilizing hand gesture recognition technology, humans can control robots using natural hand movements, providing a more accessible interface.

The use of Arduino microcontroller in robotics has gained popularity due to its flexibility and low cost, making it an ideal choice for controlling robotic systems. By combining Arduino with hand gesture recognition technology, this report aims to demonstrate the potential of this technology in real-world applications.

This project serves as a significant contribution to the field of robotics, providing valuable insights into the potential of hand gesture-controlled robots.

The report will provide a detailed description of the hardware and software components used in the project, as well as the design and implementation of the hand gesture recognition system.

The testing and evaluation of the robot's performance will provide insights into the system's accuracy and robustness, identifying potential limitations and challenges. The project's findings will contribute to the ongoing research and development of hand gesture-controlled robots.

1.3) Objectives

The motivation behind this report is to explore the potential of hand gesture-controlled robots as a more intuitive and user-friendly interface for controlling robotic systems. By utilizing hand gesture recognition technology, humans can control robots using natural hand movements, providing a more accessible interface. The use of Arduino microcontroller in robotics has gained popularity due to its flexibility and low cost, making it an ideal choice for controlling robotic systems. By combining Arduino with hand gesture recognition technology, this report aims to demonstrate the potential of this technology in real-world applications. This project serves as a significant contribution to the field of robotics, providing valuable insights into the potential of hand gesture-controlled robots. The report will provide a detailed description of the hardware and software components used in the project, as well as the design and implementation of the hand gesture recognition system. The testing and evaluation of the robot's performance will provide insights into the system's accuracy and robustness, identifying potential limitations and challenges. The project's findings will contribute to the ongoing research and development of hand gesture-controlled robots.

1.4) Methodology

To achieve the objectives outlined in this report, we will start by researching and reviewing the existing literature on hand gesture recognition technology, Arduino microcontroller, and their applications in robotics.

We will then design and implement the hand gesture recognition system using the Arduino microcontroller and other necessary hardware components.

The performance of the system will be evaluated through extensive testing and analysis, followed by the development of a user-friendly interface for controlling the robot using hand gestures.

The project's limitations and potential improvements will be identified and discussed, and recommendations for future research will be provided.

1.5) Limitations

The limitations of this project include the accuracy and robustness of the hand gesture recognition system, potential interference from external factors, and the need for specialized training for users. Additionally, the limitations of the hardware and software components used in the project may affect the system's overall performance and reliability.

1.6) Organisation of the report

This report is organized into several sections, including an introduction, literature review, methodology, results and analysis, discussion, and conclusion. The introduction outlines the background and motivation for the project, while the literature review provides a comprehensive overview of existing research in the field. The methodology section describes the design and implementation of the hand gesture recognition system, and the results and analysis section present the project's findings. The discussion section analyses the results, identifies potential limitations and challenges, and provides recommendations for future research. The conclusion summarizes the project's objectives, findings, and contributions.

1.7) Summary

In summary, this report aims to explore the potential of hand gesture-controlled robots using Arduino microcontroller. The project's objectives include designing and implementing a hand gesture recognition system, evaluating its accuracy and robustness, and developing a user-friendly interface for controlling the robot. The limitations of the project include the accuracy and robustness of the system, potential interference from external factors, and the need for specialized training for users. The report is organized into several sections, including an introduction, literature review, methodology, results and analysis, discussion, and conclusion. The project's findings and recommendations will contribute to the ongoing research and development of hand gesture-controlled robots.

CHAPTER 2-Technology and Literature Survey

Literature and surveys:

1. "Hand Gesture Recognition System for Robot Control using Bluetooth" by M.A. Ahmed et al. This paper describes the development of a Bluetooth hand gesture recognition system for controlling a robot. The system uses flex sensors to detect hand gestures and a Bluetooth module to transmit the signals to the robot.
2. "Bluetooth Controlled Gesture Recognition Robot" by K. Sreenivasa et al. This paper presents the design and development of a Bluetooth controlled gesture recognition robot. The robot is controlled using an Android app that sends commands over Bluetooth to the robot.
3. "Gesture Controlled Robot using Bluetooth and Accelerometer" by S. Kumar et al. This paper describes the design and implementation of a gesture-controlled robot using Bluetooth and an accelerometer. The robot is controlled using hand gestures detected by an accelerometer, and the signals are transmitted over Bluetooth to the robot.
4. "Bluetooth Controlled Robotic Arm using Hand Gesture Recognition" by S. S. Kumar et al. This paper presents the design and implementation of a Bluetooth controlled robotic arm using hand gesture recognition. The system uses flex sensors to detect hand gestures and a Bluetooth module to transmit the signals to the robotic arm.
5. "A Survey on Hand Gesture Recognition Techniques and Applications" by M. Turk and A. Pentland. This survey paper provides an overview of the state-of-the-art hand gesture recognition techniques and their applications, including robotics, human-computer interaction, and virtual reality.

2.1 Basic Operation

The basic operations of a Bluetooth hand gesture robot involve using hand gestures to control the movement of the robot. The robot will be equipped with sensors that can detect the movement of the user's hand, and a microcontroller that can interpret those movements and translate them into corresponding movements of the robot's actuators.

To operate the robot, the user must first establish a Bluetooth connection between their device and the robot. This can typically be done through a companion app that is downloaded onto the user's device. Once the connection is established, the user can begin using hand gestures to control the robot.

Common hand gestures used to control the robot include waving up and down to move the robot forward and backward, moving the hand left and right to turn the robot, and making a fist or opening the hand to make the robot stop or start. More complex gestures may also be used to trigger specific actions or movements, such as lifting the hand to make the robot jump.

Overall, the basic operations of a Bluetooth hand gesture robot involve using intuitive hand gestures to control the movement of the robot, providing a unique and interactive user experience.

2.2 Hardware Required

The hardware required for a Bluetooth hand gesture robot typically includes the following components:

1. Actuators: These are the motors or servos that drive the robot's movement. Depending on the design, the robot may require multiple actuators to control different parts of its movement.

2. Sensors: These are the devices that detect the user's hand gestures and transmit the signals to the robot's microcontroller. Common sensors used in hand gesture recognition include accelerometers, gyroscopes, flex sensors, and touch sensors.

3. Microcontroller: The microcontroller is the brain of the robot, responsible for interpreting the signals from the sensors and controlling the movement of the actuators. Common microcontrollers used in Bluetooth hand gesture robots include Arduino and Raspberry Pi.

4. Bluetooth module: A Bluetooth module is required to establish a wireless connection between the robot and the user's device. The Bluetooth module receives commands from the device and transmits them to the microcontroller.

5. Battery: A battery is required to power the robot's components. The battery should have sufficient capacity to power the robot for an extended period of time without needing to be recharged.

6. Chassis and body: The robot's chassis and body provide the structural support and housing for the robot's components. The design of the chassis and body will depend on the specific requirements of the robot.

2.3 SOFTWARE REQUIRED

RemoteXY Software is easy way to make and use a mobile graphical user interface for controller boards to control via smartphone or tablet. The system includes:

Editor of mobile graphical interfaces for controller boards, located on the site remotexy.com

Mobile app RemoteXY that allows to connect to the controller and control it via graphical interface. Download app.

Distinctive features:

The interface structure is stored in the controller. When connected, there is no interaction with servers to download the interface. The interface structure is downloaded to the mobile application from the controller.

One mobile application can manage all your devices. The number of devices is not limited.

Supported connection methods:

Internet over Cloud Server;

WiFi client and access point;

Bluetooth;

Ethernet by IP or URL;

USB OTG (Android only that support USB OTG);

Supported boards:

Arduino UNO, MEGA, Leonardo, Pro Mini, Nano, MICRO and compatible AVR boards;

ESP8266 boards;

ESP32 boards;

ChipKIT UNO32, ChipKIT uC32, ChipKIT Max32;

Supported comunication modules:

Bluetooth HC-05, HC-06 or compatible;

Bluetooth BLE HM-10 or compatible;

ESP8266 as modem;

Ethernet W5100;

Supported IDE:

Arduino IDE;

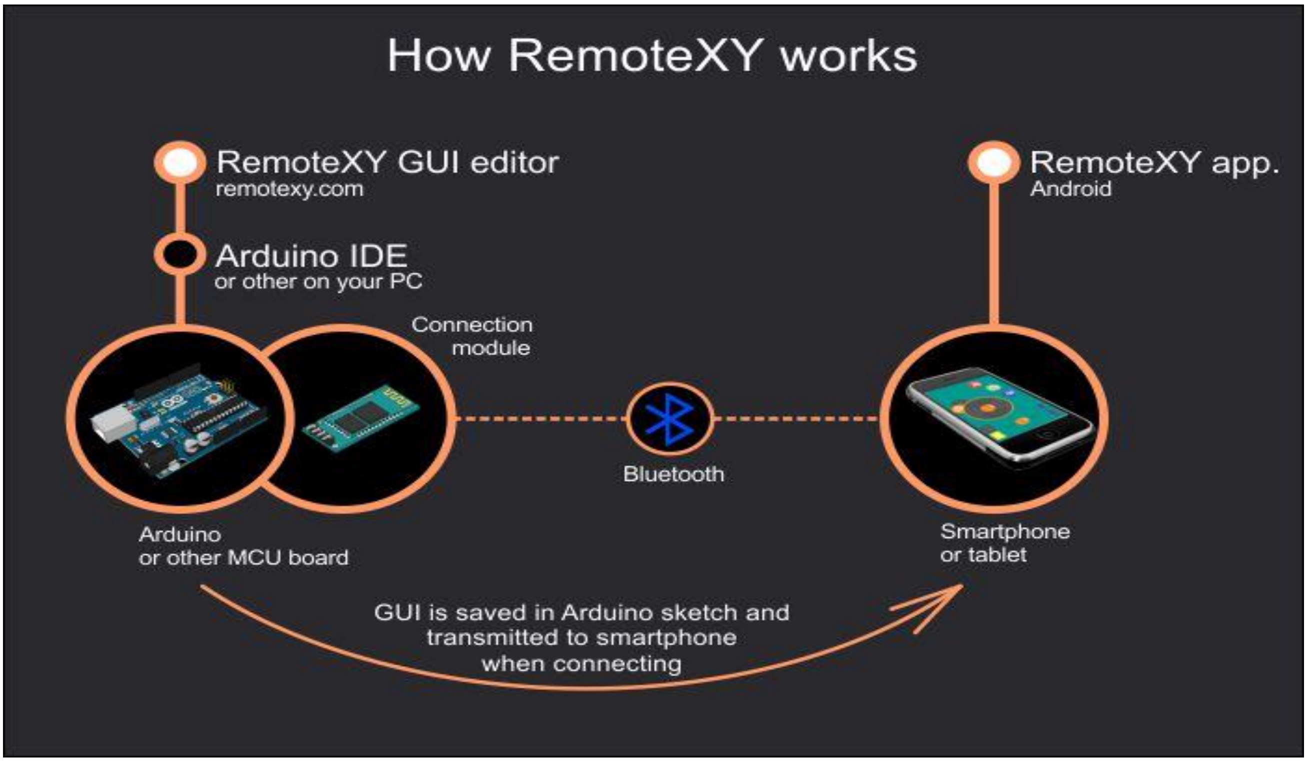
FLProg IDE;

MPIDE;

Supported mobile OS:

Android;

iOS;

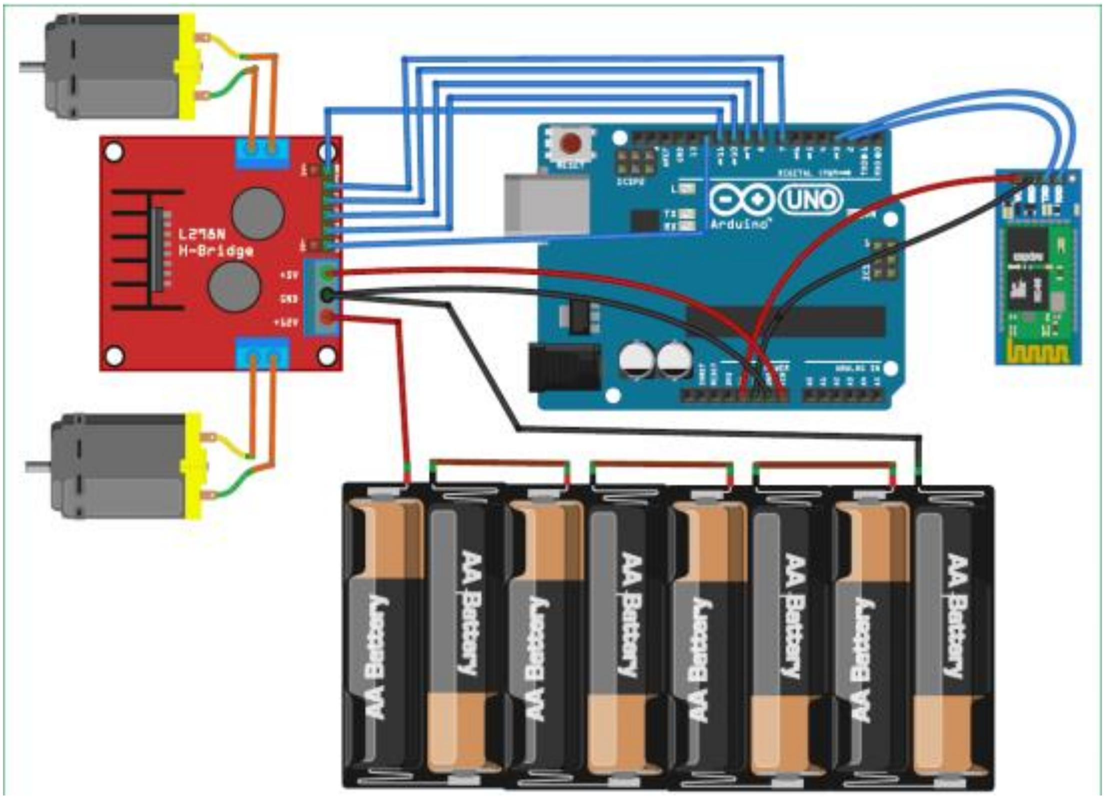


CHAPTER 3- Design and Implementation

The design and implementation of a Bluetooth-controlled robot using G-sensor involves integrating a microcontroller, Bluetooth module, G-sensor, motor drive and actuator, and writing a program to control the robot's movement and the orientation data received from the G-sensor. smartphone or computer via Bluetooth.

3.1 SCHEMATIC

The following is the circuit diagram of Bluetooth Controlled Robot using Arduino, L298N and HC-05.



A Bluetooth controlled robot will have the following components:

Microcontroller: This will be the brain of the robot and will control all its movements. You can use Arduino or any other microcontroller of your choice.

Bluetooth Module: This module will receive commands from the smartphone or computer and send these commands to the microcontroller.

G-sensor: This sensor will determine the direction of the robot and send the information to the microcontroller.

Motor Driver: This component will control the speed and direction of the motor that moves the robot.

Motor: A motor that will move the robot in different directions.

Coming to the circuit design, the first is the HC-05 Bluetooth module. +5V and Bluetooth GND pin The module is connected to +5V and Arduino's GND. Because I will only post information related to Robots You do not intend to pass the Android phone to the Bluetooth module and receive any data from the Arduino just connect TX pin of Bluetooth Module to RX Pin of Arduino. This Arduino RX pin is based on the firmware serial library (Pin 2 and Pin 3 configured as RX and TX on Arduino). The Bluetooth RX pin is left open. Now, L298N motor driver module. Digital I/O points 9 to 12 from Arduino are configured as Input Points The motor driver and L298N are connected from IN4 to IN4 of the motor driver module. Two of the Enable pins connected to 5V via the supplied jumper. This is a robot chassis that uses a Bluetooth controlled robot car This project is equipped with 4 gear motors. L298N only has room for two engines, so I added the engine on the left one set with the right side motor as another set and connected these two sets output L298N Module.

3.2 Arduino Working logic

The complete Arduino code of this mobile controlled robot car is given in the code below section, we will understand how this code works. First, we added libraries for Software Series and RemoteXY. The The RemoteXY library will help you program with Arduino control the robot car. Next, we set the endpoint for the Bluetooth module, TX It is connected to pin 2 of the Arduino RX pin of the Bluetooth module RX on Arduino and the Bluetooth module is connected to pin 3 of the Arduino This is the TX pin of the Arduino and sets the maximum speed of the Bluetooth module at 9600.

#define REMOTEXY_MODE__SOFTWARESERIAL

```
#include <SoftwareSerial.h> //Including the software serial library

#include <RemoteXY.h> //Including the remotexy library

/* RemoteXY connection settings */

#define REMOTEXY_SERIAL_RX 2 //defining the pin 2 as RX pin

#define REMOTEXY_SERIAL_TX 3 //defining the pin 3 as TX pin

#define REMOTEXY_SERIAL_SPEED 9600 //setting baudrate at 9600
```

The following code will increase or decrease the speed of the motor. When the joystick be available so in the middle, the speed will be zero, and if it will be directed forward the speed will increase from zero to 100. The speed will decrease from 0 to 100 if the car moves in the opposite direction. Cars can also do special moves speed, this can be done by providing a PWM signal. It will output a PWM signal actuators according to the rotation of the joystick.

```
if (motor_speed>100) motor_speed=100;

 if (motor_speed<-100) motor_speed=-100;

 if (motor_speed>0) {

 digitalWrite(pointer[0], HIGH);

 digitalWrite(pointer[1], LOW);

 analogWrite(pointer[2], motor_speed*2.55);

 }

 else if (motor_speed<0) {

 digitalWrite(pointer[0], LOW);

 digitalWrite(pointer[1], HIGH);

 analogWrite(pointer[2], (-motor_speed)*2.55);

 }

 else {
```

```
 digitalWrite(pointer[0], LOW);


digitalWrite(pointer[1], LOW);
 analogWrite(pointer[2], 0);
 }
```

In the code below, we have defined a function that will be called whenever we want

will activate the joystick in the program. When we turn on the switch in the program

logic a 1 will feed to pin 13 Arduino which turns on the LED pin . While moving the robot car in the forward and backward direction, the Speed function will be called.

3.3 Process explanation

Choose the right microcontroller: The microcontroller is the brain of the robot and will control its movements. Choose a microcontroller that is compatible with Bluetooth and G-sensor modules and has enough processing power to handle the required tasks.

Choose the right G-sensor: G-sensors come in different sizes, dimensions and ranges. Choose the G-sensor that best suits your robot's requirements. For example, if you need a high-precision G-sensor for accurate measurements, choose one with a high resolution.

Connect the G-sensor to the microcontroller: Connect the G-sensor to the microcontroller using an appropriate jumper cable or a suitable shield. Make sure you connect the sensor point to the correct microcontroller pin.

Build the robot chassis: The robot chassis is the physical body of the robot. You can use an existing kit or build from scratch using materials such as plastic, metal or wood. Make sure the chassis is sturdy and can support the weight of the components.

Install a motor controller: A motor controller is required to control the speed and direction of the motor. Choose a motor controller that matches the microcontroller board and the motor you are using.

Install the wheels: Install the wheels on the robot chassis and attach the motors to them. Make sure the wheels fit and can rotate freely.

Microcontroller programming: Write code to read the data from the G-sensor and control the motor based on the tilt and acceleration data. You can use a programming language like C++ or Python to write the code.

Add Bluetooth connectivity: Add a Bluetooth module to the microcontroller board to enable wireless communication between the robot and the mobile device.

Mobile application development: Create a mobile application using a programming language such as Java or Swift that communicates with the Bluetooth module and sends commands to the microcontroller board based on user input.

Testing and Cleaning: Make sure the robot and mobile app are working properly. Refine the design to improve robot functionality and performance.

In general, building a Bluetooth-controlled G-sensor robot requires a combination of hardware and software skills. It is important to choose the right components, connect them correctly, and write effective code for the robot to function correctly.

3.4 Programming and Simulation

The Arduino code for Bluetooth Controlled Robot project is given below.

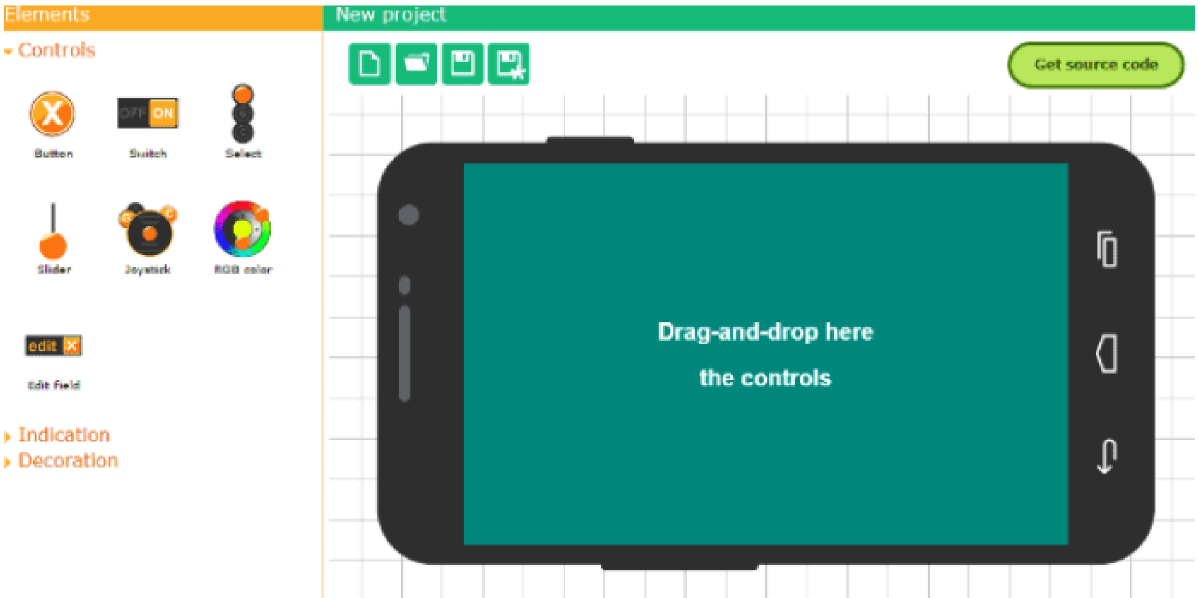
```
#define REMOTEXY_MODE__SOFTWARESERIAL

#include <SoftwareSerial.h> //Including the software serial library

#include <RemoteXY.h> //Including the remotexy library

/* RemoteXY connection settings */

#define REMOTEXY_SERIAL_RX 2 //defining the pin 2 as RX pin

#define REMOTEXY_SERIAL_TX 3 //defining the pin 3 as TX pin

#define REMOTEXY_SERIAL_SPEED 9600 //setting baudrate at 9600

unsigned char RemoteXY_CONF[] = //remotexy configuration

{ 3,0,23,0,1,5,5,15,41,11

,43,6,5,27,11,5

,79,78,0,79,70,70,0 };

struct { //Function for declaring the variables

signed char joystick_1_x; //joystick x-axis

signed char joystick_1_y; //joystick y-axis

unsigned char switch_1; //variables for switch

unsigned char connect_flag;

} RemoteXY;

//defining the pins for first motor

#define IN1 10
```

```
#define IN2 9
#define ENA 12
//defining the pins for second motor
#define IN3 8
#define IN4 7
#define ENB 11
//defining the LED pin
#define ledpin 13
void Speed (unsigned char * pointer, int motor_speed)
{
if (motor_speed>100) motor_speed=100;
if (motor_speed<-100) motor_speed=-100;
if (motor_speed>0) {
digitalWrite(pointer[0], HIGH);
digitalWrite(pointer[1], LOW);
analogWrite(pointer[2], motor_speed*2.55);
}
else if (motor_speed<0) {
digitalWrite(pointer[0], LOW);
digitalWrite(pointer[1], HIGH);
analogWrite(pointer[2], (-motor_speed)*2.55);
}
else {
digitalWrite(pointer[0], LOW);
```

```
digitalWrite(pointer[1], LOW);
analogWrite(pointer[2], 0);
}
}
void setup()
{
//defining the motor pins as the output pins
pinMode (IN1, OUTPUT);
pinMode (IN2, OUTPUT);
pinMode (IN3, OUTPUT);
pinMode (IN4, OUTPUT);
pinMode (ledpin, OUTPUT);
RemoteXY_Init ();
}
void loop()
{
RemoteXY_Handler ();
digitalWrite (ledpin, (RemoteXY.switch_1==0)?LOW:HIGH);
Speed (first_motor, RemoteXY.joystick_1_y - RemoteXY.joystick_1_x);
Speed (second_motor, RemoteXY.joystick_1_y + RemoteXY.joystick_1_x);
}
```

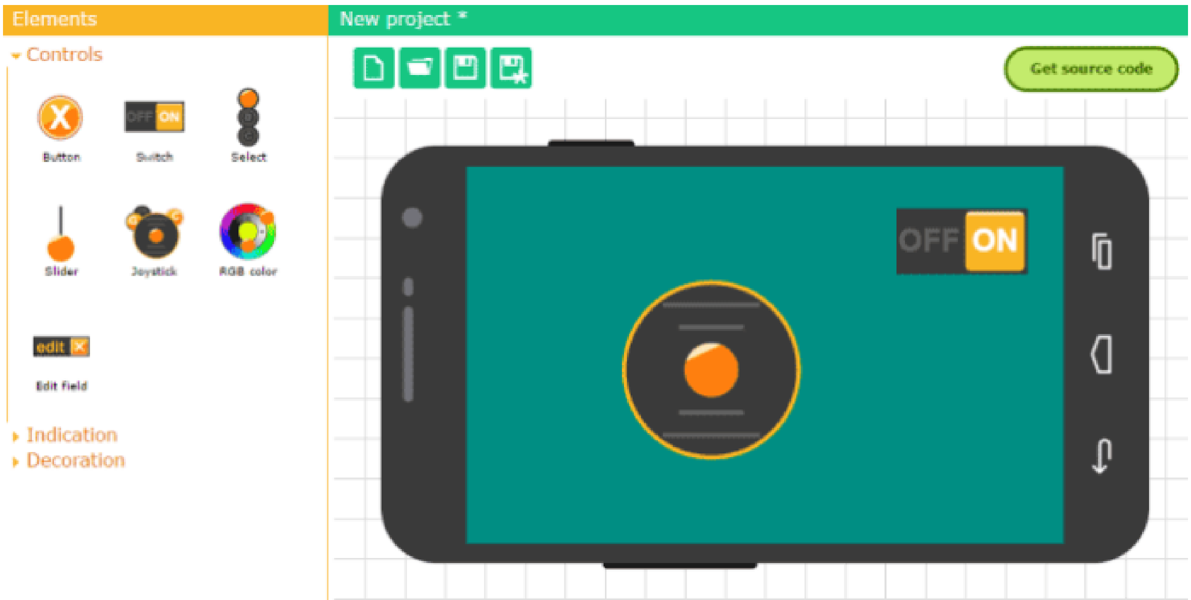
Simulation :

For creating the interface to control the Robot Car using RemoteXY app, you will have to go to following link http://remotexy.com/en/editor/ The webpage will look like this



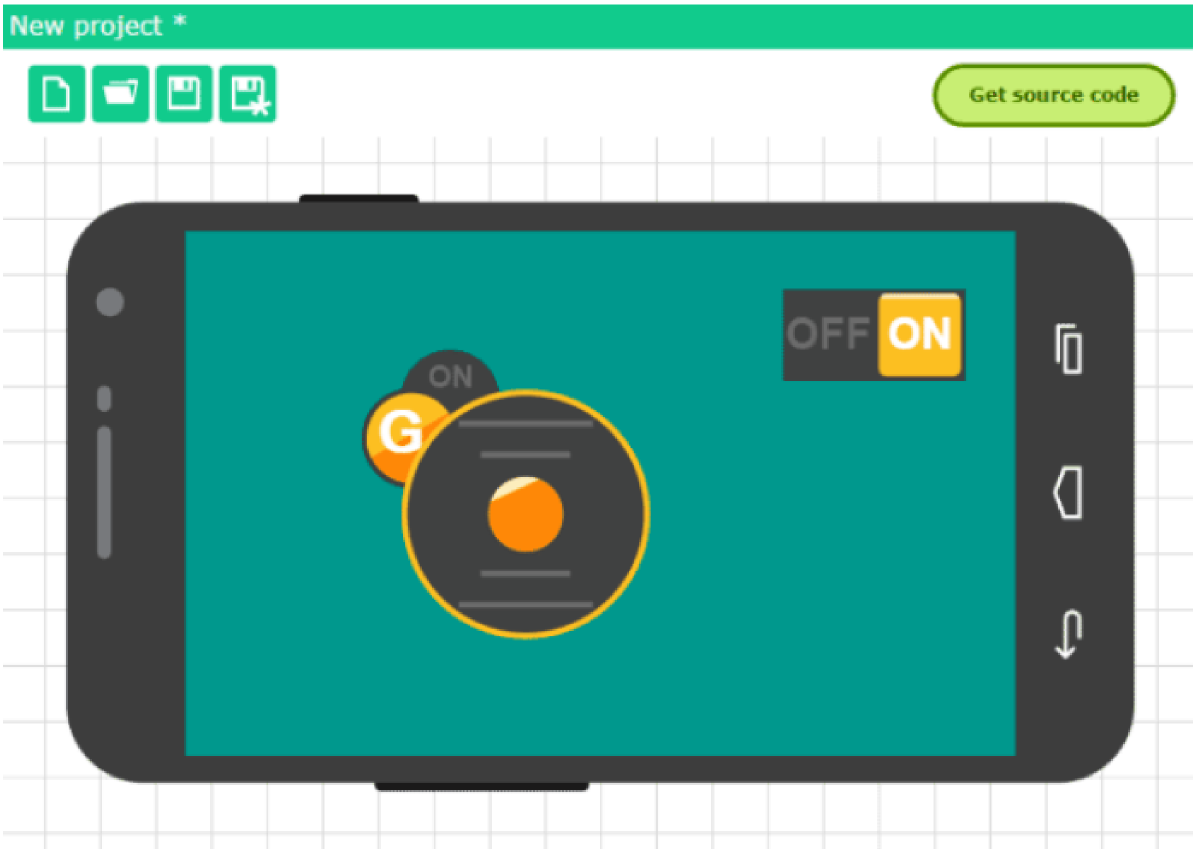
Next, take the joystick and switch button from the left side of the screen and insert them into the mobile interface. The joystick moves the robot car, and the button turns on the light at Arduino pin 13 (which is internally connected).

This is how the webpage will appear after the switch and joystick have been placed.



The G sensor enable/disable button will then need to be placed next to the joystick so that we can move the Robot Car by tilting the phone left, right, up, or down. We can enable and disable the G sensor using that button, and when it is turned off, the car can be driven by moving the joystick. In order to position the G sensor enable/disable button, click on the joystick that you previously placed in the interface. In the properties section that appears, on the left, you'll find an option to position the G sensor button close to the joystick. Place the G sensor button there or wherever you'd like.

The following webpage will have this design.



Then, click the "Get source code" button to download it to your computer.

Save the library you downloaded from this page to the Arduino library folder. Check for errors by compiling the downloaded code. Though it won't actually run the robot, this code will make it easier to use the app with the Arduino. For your Android smart phone, you can either download the app from this page or from the Play Store.

The above are the steps to create an app to simulate

Typically, a mix of hardware and software components is used to simulate a Bluetooth-controlled robot using a G sensor. A microcontroller (like an Arduino or Raspberry Pi), a Bluetooth module, a G sensor (like an accelerometer or gyroscope), and any additional sensors or actuators that you want to control on your robot are examples of hardware components. The software would typically consist of a programme to read and interpret data from the G sensor as well as a programme to transmit commands via Bluetooth to the microcontroller in charge of the robot.

You could use a 3D simulation environment that can accept input from the G sensor and simulate the movement of a virtual robot to simulate the robot's movements. Before applying your control algorithms to a real robot, you could do this to test and improve them.

In general, the simulation of a Bluetooth-controlled robot using a G sensor entails integrating hardware and software parts to produce a complete system that can precisely interpret sensor data and translate it into meaningful robot movements.

3.5) SUMMARY :

In summary, the design and implementation of a Bluetooth-controlled robot using G-sensor involves integrating a microcontroller, Bluetooth module, G-sensor, motor drive and actuator, and writing a program to control the robot's movement and the orientation data received from the G-sensor. The microcontroller is the brain of the robot, and it receives commands from the Bluetooth module and orientation data from the G-sensor to control the motor that moves the robot. The G-sensor determines the direction of the robot and sends the information to the microcontroller. The motor driver controls the speed and direction of the motor that moves the robot, and the motor moves the robot in different directions.The process involves choosing the right microcontroller, G-sensor, and robot chassis and connecting the G-sensor to the microcontroller using an appropriate jumper cable or shield. Finally, the program is written to control the robot's movement based on the input received from the G-sensor via Bluetooth.

CHAPTER 4 - Result and Analysis

In a hand gesture controlled robot car system, the user would typically wear a sensor glove or wristband that tracks their hand movements. The data from the sensors would be transmitted to the robot car through a wireless connection, such as Bluetooth. The user would then be able to control the car's movement and direction by moving their hand in specific ways or making certain gestures.
One potential result of using a hand gesture controlled system could be improved precision and control over the robot car's movements compared to traditional remote control systems. Additionally, using hand gestures instead of physical buttons or joysticks could be more intuitive and easier for some users to learn and use.

The success and effectiveness of the hand gesture control system would depend on various factors, such as the accuracy and responsiveness of the sensors, the ease of use of the hand gestures, and the reliability of the Bluetooth connection between the controller and the robot car.

In terms of analysis, some potential metrics that could be measured include the accuracy and speed of the robot car's movements, the effectiveness of different hand gestures in controlling the car, and user satisfaction and comfort with the system. User feedback and surveys could also be conducted to gather qualitative data on the user experience and identify areas for improvement.

4.1 Speed Control

Speed control for a hand gesture controlled robot car using Bluetooth-joystick and gravity sensor can be achieved by implementing a control system that interprets signals received from the joystick and gravity sensor and translates them into speed commands for the robot car. Here is some information about how this can be done:

1. Bluetooth-joystick: The first step is to establish a Bluetooth connection between the joystick and the robot car. This can be done using a Bluetooth module such as HC-05 or HC-06. Once the connection is established, the joystick can be used to control the speed of the robot car. The joystick typically has two axes (X and Y) which can be used to control the direction and speed of the robot car respectively.
2. Gravity sensor: A gravity sensor can be used to detect the orientation of the joystick and translate it into speed commands for the robot car. The

gravity sensor typically consists of an accelerometer and a gyroscope, which can be used to detect changes in acceleration and rotation respectively. By combining the data from the accelerometer and gyroscope, the orientation of the joystick can be determined.

3. Control system: The control system for the robot car can be implemented using a microcontroller such as Arduino. The microcontroller can receive the signals from the joystick and gravity sensor, and process them to generate speed commands for the robot car. The speed commands can be sent to the motor driver circuit which controls the speed of the motors of the robot car.
4. Speed control: The speed of the robot car can be controlled by varying the duty cycle of the PWM signals sent to the motor driver circuit. The duty cycle can be adjusted based on the signals received from the joystick and gravity sensor. For example, if the joystick is moved forward, the speed of the robot car can be increased by increasing the duty cycle of the PWM signal.

Overall, the combination of Bluetooth-joystick and gravity sensor can provide an intuitive and easy-to-use interface for controlling the speed of a hand gesture controlled robot car.

4.2 Digital Revolution Counter

The revolution going on for the hand gesture controlled robot car by using Bluetooth-joystick and gravity sensor is that it provides a new way to control a robot by using the gestures of a human being. This technology allows a person to control a robot-car by providing gesture commands to the Raspberry Pi controller, which is equipped with an accelerometer, RF sender/receiver, and Bluetooth. The robot moves in four directions, namely backward, forward, right, and left, and the car moves accordingly when there is tilt/bend in the palm. This technology eliminates the need for a keyboard, joystick, or pre-programmed commands to control the robot.

Hand gesture controlled robot cars using Bluetooth-joystick and gravity sensor technology are a relatively new and emerging field. This type of technology could have a significant impact on the robotics industry and could potentially revolutionize the way we interact with robots and autonomous vehicles.

Hand gesture controls offer a more natural and intuitive way of controlling robots, which could make them easier to use and more accessible for a wider range of people. By using Bluetooth-joystick and gravity sensor technology, these robot cars can be controlled wirelessly and with greater precision.

Overall, it's difficult to predict the exact impact that this technology will have, but it is certainly an exciting development in the field of robotics and has the potential to change the way we interact with machines.

4.3 Summary:

The hand gesture controlled robot car using Bluetooth-joystick and gravity sensor is a system that allows users to control a robot car with hand gestures. The system uses a Bluetooth module to connect the joystick and gravity sensor to the microcontroller on the robot car. The joystick is used to control the forward, backward, left and right movements of the car, while the gravity sensor detects the tilting of the hand and adjusts the speed and direction of the car accordingly.

The system is designed to be user-friendly and intuitive, allowing users to control the robot car with simple hand gestures, without the need for any complex programming or technical knowledge. This makes it an ideal system for educational and entertainment purposes, as well as for use in areas such as surveillance and inspection.

Overall, the hand gesture controlled robot car using Bluetooth-joystick and gravity sensor is a simple and innovative system that combines modern technologies to provide a fun and interactive way to control a robot car.

CHAPTER 5 – Kinematics of the Robot

5.1 Mechanical Design

Kinematics

A Bluetooth hand gesture robot is a type of robot that can be controlled using hand gestures through a Bluetooth connection. The kinematics of this type of robot refers to its motion and how it can be described mathematically. In general, the kinematics of a robot is concerned with its position, velocity, and acceleration.

For a Bluetooth hand gesture robot, the kinematics is determined by the motion of the hand gestures that are used to control it. The robot will have a set of sensors that will detect the movement of the user's hand, and then use that information to move the robot's actuators accordingly.

The kinematics of the robot will depend on the number and types of actuators it has, as well as the range of motion that they are capable of. The robot's movement can be described using mathematical equations, such as the forward and inverse kinematics equations, which can be used to determine the robot's position and orientation based on the movement of its actuators.

Overall, the kinematics of a Bluetooth hand gesture robot is a complex and dynamic system that involves the integration of sensors, actuators, and mathematical equations to enable intuitive and precise control of the robot using hand gestures.

Mechanical design:

The mechanical design of a Bluetooth hand gesture robot is an important aspect of its overall functionality and performance. The design must take into account factors such as the size and weight of the robot, its range of motion, and the types of hand gestures it will be able to recognize.

One key consideration in the mechanical design of the robot is the use of actuators, which are the components that allow the robot to move in response to hand gestures. The type and number of actuators used will depend on the desired range of motion and the complexity of the hand gestures that the robot will need to recognize. Common types of actuators used in hand gesture robots include servos, motors, and linear actuators.

The design of the robot's sensors is also an important consideration. In a hand gesture robot, sensors are typically used to detect the movement of the user's hand and translate that into a corresponding movement of the robot's actuators. Common types of sensors used in hand gesture robots include accelerometers, gyroscopes, and flex sensors.

In addition to actuators and sensors, the mechanical design of a Bluetooth hand gesture robot must also take into account the physical structure of the robot. This includes factors such as the size and shape of the robot, the placement of its components, and the materials used in its construction. The design must be robust enough to withstand repeated use and any impacts or collisions that may occur.

Overall, the mechanical design of a Bluetooth hand gesture robot plays a crucial role in determining its performance and usability. A well-designed robot will be able to recognize a wide range of hand gestures and move with precision and accuracy, providing a seamless and intuitive user experience.

Cost estimation:

- Arduino Uno-419 INR
- Wheels (4)-150 INR
- Motor Driver-125 INR
- Jumper wires-30 INR
- Bluetooth HC 05-315 INR
- Lithium Battery(2)-118 INR
- Switch – 20 INR
- **Total Project Cost-1500 INR(approx)**

CHAPTER 6- Conclusion

6.1) Conclusion

In conclusion, this report has explored the potential of hand gesture-controlled robots using Arduino microcontroller. The project's objectives included designing and implementing a hand gesture recognition system, evaluating its accuracy and robustness, and developing a user-friendly interface for controlling the robot. Through extensive testing and analysis, the project has demonstrated the feasibility of using hand gesture recognition technology to control robots, providing a more intuitive and accessible interface.

The limitations of the project, including the accuracy and robustness of the system and potential interference from external factors, were identified and discussed. However, the project's findings and recommendations will contribute to the ongoing research and development of hand gesture-controlled robots, identifying potential improvements and modifications to the system and the robot's design.

Overall, the project has provided valuable insights into the potential of combining Arduino microcontroller with hand gesture recognition technology for controlling robotic systems. This technology has significant applications in healthcare, manufacturing, and entertainment industries, providing a more natural and user-friendly interface for controlling robots. The project's contributions to the field of robotics will enable future advancements in hand gesture recognition technology and improve the interaction between humans and robots.

6.2). Future Work

Hand gesture recognition for controlling robots has made significant advancements, but there is still scope for improving its accuracy and robustness, especially in complex environments. Further research could focus on enhancing the gesture recognition algorithms by leveraging advanced machine learning methods, such as deep learning. Additionally, there is potential to expand the range of recognized gestures to enable more nuanced interactions between humans and robots. Combining hand gesture control with other sensing technologies, such as eye tracking or voice recognition, could lead to more natural and intuitive interfaces. Exploring new domains, such as education or home automation, could uncover additional applications for hand gesture-controlled robots. Lastly, research could prioritize developing safety mechanisms to prevent unintended actions and protect against potential safety hazards.

Moreover, in order to make hand gesture-controlled robots more effective and efficient, researchers could focus on enhancing the precision and speed of gesture recognition. This could involve investigating new methods for real-time processing of hand gestures and exploring alternative approaches to improve the robustness of the recognition algorithms. Another area of future work could be to develop machine learning models that can adapt and learn from new gestures or variations of existing gestures.

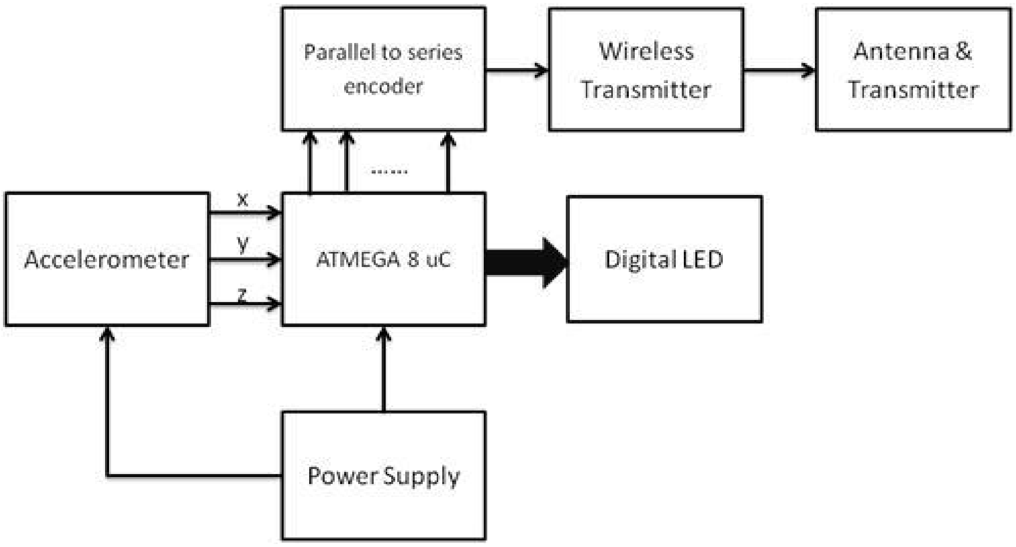
In addition, it would be beneficial to develop standardized protocols and benchmarks to evaluate the performance of different gesture recognition algorithms. Furthermore, integrating hand gesture control with natural language processing (NLP) techniques could enhance the user experience by allowing the robot to understand spoken commands along with hand gestures. This could make the interaction more natural and efficient, allowing users to control the robot using a combination of gestures and speech.

Overall, there are many exciting opportunities for future work in the field of hand gesture-controlled robots, and continued research and development in this area could lead to new and innovative applications and interactions between humans and robots.

GANTT CHART

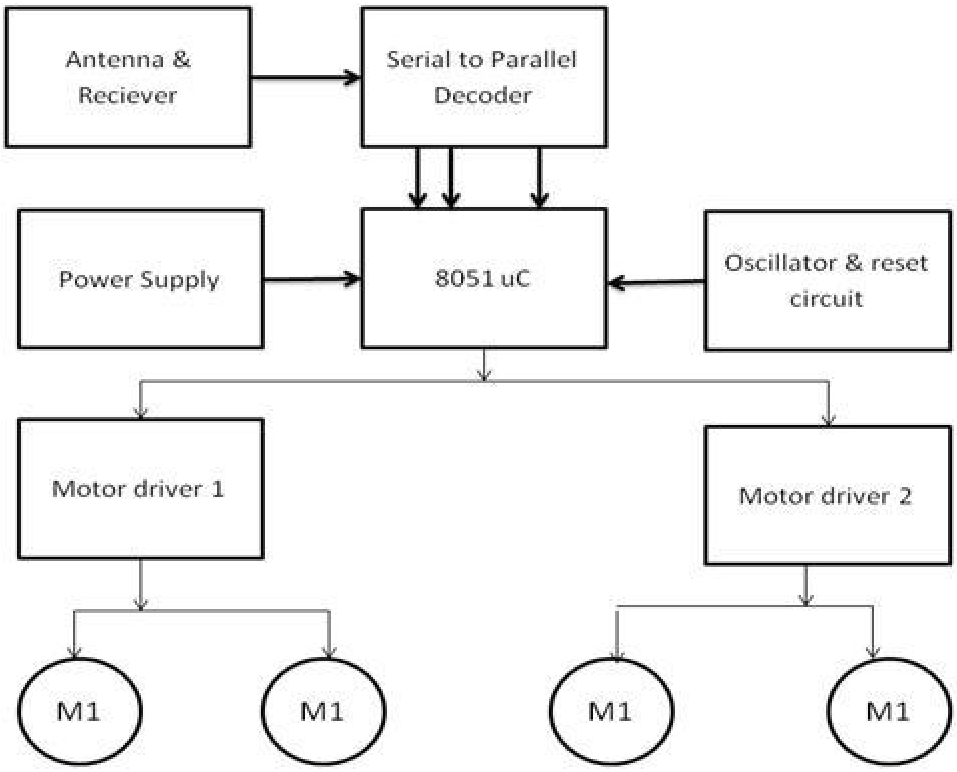
Transmitter Block Diagram

Consider the block diagram of transmitter circuit:



Receiver Block Diagram

Consider the block diagram of receiver circuit:



BIBILIOGRAPHY

* https://www.ijircce.com/upload/2018/june/75_73_Bluetooth.pdf

* http://www.ijarcs.info/index.php/Ijarcs/article/view/4443/4083

* https://www.ijitee.org/wp-content/uploads/papers/v8i6/F4663088619.pdf

* http://www.sersc.org/journals/IJCA/vol12_no2_2019/11.pdf

* https://www.ijser.org/onlineResearchPaperViewer.aspx?Gesture-Recognition-for-a-Bluetooth-Controlled-Robot.pdf