ISSN No:-2456-2165

# IR Navigator Postioning System

Prathamesh Jadhav[1]; Yash Dhekale[2]; Abhishek Joshi[3]; Darshan Achalkar[4]
Kolhapur Institute of Technology’s College of Engineering, Kolhapur

**Abstract:- The IR Navigator Positioning System shows how IR remote control and Arduino-powered servo motors can work together smoothly. It's a project that provides accurate control in different situations using IR communication and servo motors. It's a clever and easy-to-use solution for various applications.**

**The system uses IR receivers to pick up signals from a standard remote control. These signals are carefully handled by an Arduino microcontroller, the brain of the operation. What makes this system stand out is its ability to turn those IR signals into precise movements of a servo motor, enabling accurate angular positioning for various applications.**

**Combining these elements opens up a world of possibilities, enhancing robotics, automation, and enabling incredibly accurate remote-controlled devices. The Precision Positioning System, introduced here, invites a closer look into its technical intricacies and practical applications. This integration provides a foundation for advancing various projects, making them more precise and efficient. Whether it's refining the movements of robots or ensuring unparalleled accuracy in remote-controlled devices, the system offers versatile solutions. The following exploration will delve into the details of how this system works and its potential impact on a wide range of applications.**

## I. INTRODUCTION

In the realm of electronics and automation, the Precision Positioning System stands as a testament to the synergy between Infrared (IR) remote control technology and the precision control capabilities of Arduino-based servo motors. This project addresses the need for fine-tuned positional control in various applications, introducing a sophisticated yet accessible solution that leverages the power of IR communication and servo motor manipulation.

At its core, the system employs IR receivers to capture signals from a standard remote control device. These signals are then meticulously processed by an Arduino microcontroller, serving as the brain of the operation. The defining feature of the system lies in its ability to translate these IR signals into precise movements of a servo motor, allowing for accurate angular positioning.

The integration of these elements opens the door to a myriad of possibilities, from enhancing robotics and automation projects to facilitating remote-controlled devices with unparalleled accuracy. This introduction sets the stage for a closer exploration of the technical intricacies and practical applications that define the Precision Positioning System.

## II. REVIEW OF RELATED LITERATURE

➢ *IR Remote Control Technology:*

- Explore the advancements and applications of infrared communication in remote control systems.
- Investigate protocols commonly used in IR remote communication and their integration with electronic devices.

➢ *Arduino-based Control Systems:*

- Review existing literature on Arduino microcontrollers and their applications in robotics and automation.
- Explore studies on interfacing Arduino with various sensors and actuators for precise control.

➢ *Servo Motor Control:*

- Examine literature on servo motor technology, focusing on its principles of operation and applications in precision control systems.
- Investigate control algorithms and methodologies used for accurate positioning of servo motors.

➢ *Integration of IR with Arduino and Servo Motors:*

- Identify studies that have successfully integrated IR remote control with Arduino based systems.
- Explore research on servo motor control through Arduino and assess the challenges and solutions presented in previous works.

➢ *Applications in Robotics and Automation:*

- Survey literature related to the use of similar systems in robotics, automation, and mechatronics.
- Identify successful case studies and applications where precise positioning systems have been instrumental.

## III. RESEARCH GAP

➢ *The Potential Research Gaps for this Project Include Areas where Current Studies Might be Lacking:*

- *Advanced IR Protocols:*

Research may not cover more advanced IR protocols for improved remote control.

- *Integration Challenges in IR with Arduino:*

There might be a gap in addressing challenges and solutions for seamlessly combining IR with Arduino.

- *Servo Motor Control Algorithms:*

Specific algorithms for precise servo motor positioning may be inadequately documented.

- *Case Studies in Robotics and Automation:*

Real-world case studies demonstrating the impact of precise positioning systems in robotics and automation could be lacking.

➢ *Objective :*

What we plan to accomplish is a precise control of servo motor varying due to the signals received by IR receiver. It can form a small application in various big/mega projects. This type of project is needed as a small contribution for the big projects taking place. Applying this method and implementing on a large scale will surely help industrial projects to move/plan according to the given data. Heavy Machinery can be rotated due to the implementation of this idea.

➢ *Scope :*

The research focuses on combining IR remote control, Arduino technology, and servo motors for precise positioning. It includes studying advancements in IR communication, Arduino's role in robotics, servo motor control, and how these elements can be successfully integrated. The practical applications in robotics and automation will be explored, emphasizing case studies. The goal is to contribute to understanding and improving the use of these technologies together. The research will cover both theoretical and practical aspects, addressing challenges and potential future developments within this scope.

➢ *List of Materials :*

Hardware Components:

- *IR Remote Control:*

Fig 1 Standard Infrared Remote-Control Device for Sending Commands to the System.[1]

- *IR Receiver:*

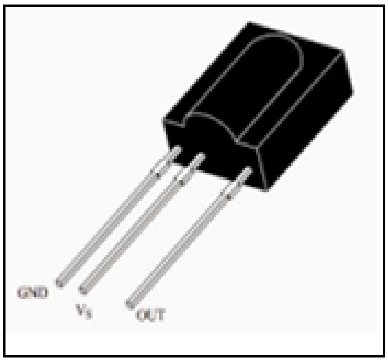


Fig 2 Receives and Decodes Infrared Signals from the Remote Control

- *Arduino Microcontroller:*

Fig 3 Acts as the Central Processing Unit for Interpreting IR Signals and Controlling the Servo Motor

- *Servo Motor:[2]*

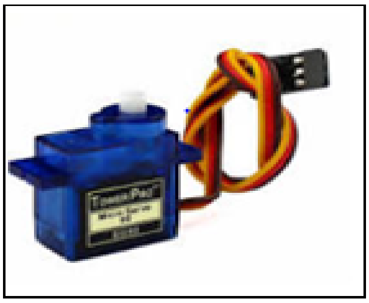


Fig 4 Output Actuator Responsible for Physical Rotation to the Specified Angle.

- *Power Supply:*

Provides the necessary power to the Arduino, servo motor, and other components.

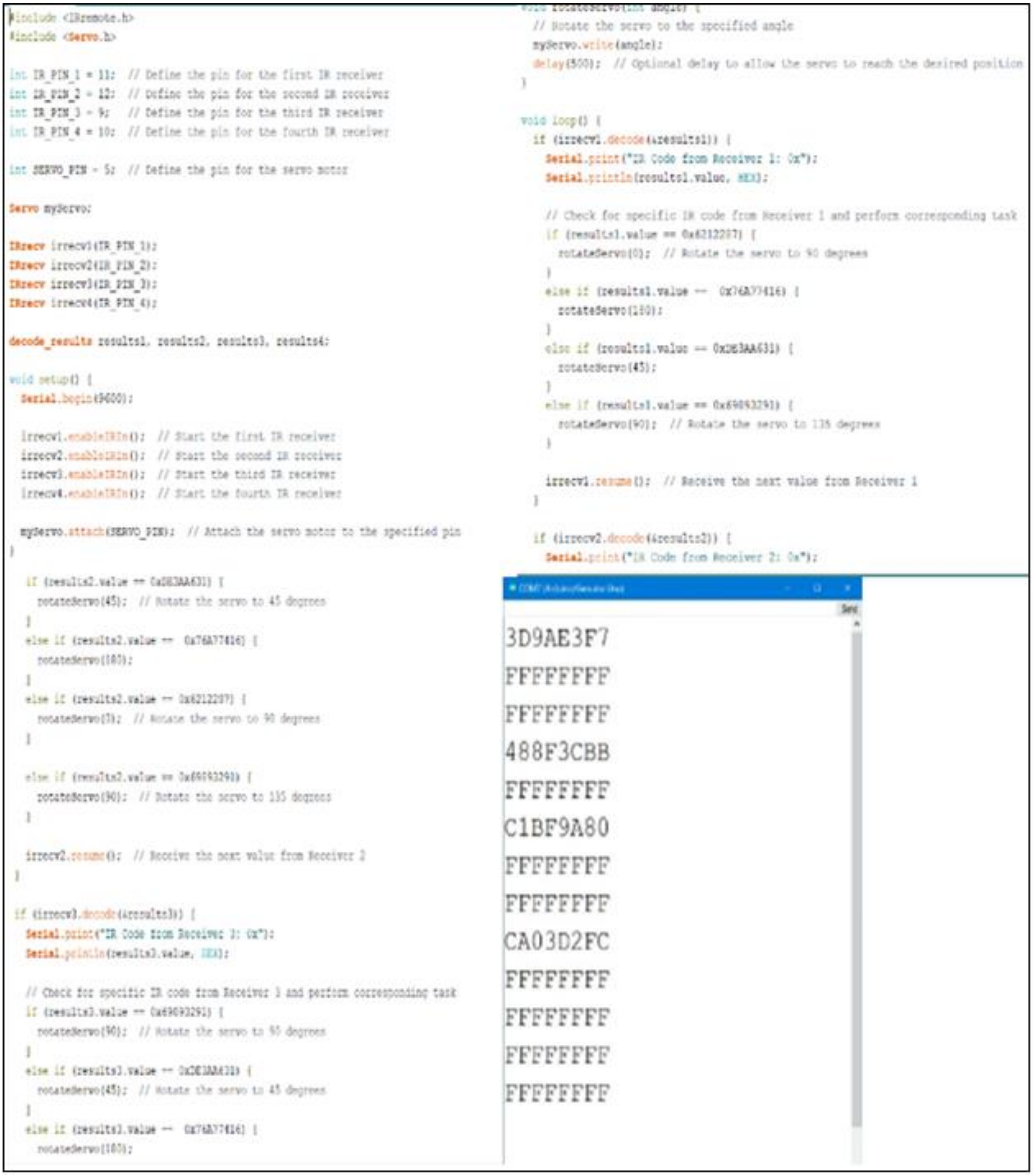
- *Connecting Cables:*

Wires and cables for connecting the various components, ensuring proper communication and power distribution.

- *Software Components:*
- *Arduino IDE:*

Fig 5 Integrated Development Environment for Writing, Compiling, and Uploading the Arduino Code.

- *Arduino Code:[5]* https://drive.google.com/file/d/1aUtl4A2o9e30XbbVUJr4hqYlmzckE3VH/view?usp=drive_link



```
#include <IRremote.h>
#include <Servo.h>

int IR_PIN_1 = 11;  // Define the pin for the first IR receiver
int IR_PIN_2 = 12;  // Define the pin for the second IR receiver
int IR_PIN_3 = 9;   // Define the pin for the third IR receiver
int IR_PIN_4 = 10;  // Define the pin for the fourth IR receiver

int SERVO_PIN = 5;  // Define the pin for the servo motor

Servo myServo;

IRrecv irrecv1(IR_PIN_1);
IRrecv irrecv2(IR_PIN_2);
IRrecv irrecv3(IR_PIN_3);
IRrecv irrecv4(IR_PIN_4);

decode_results results1, results2, results3, results4;

void setup() {
  Serial.begin(9600);

  irrecv1.enableIRIn();  // Start the first IR receiver
  irrecv2.enableIRIn();  // Start the second IR receiver
  irrecv3.enableIRIn();  // Start the third IR receiver
  irrecv4.enableIRIn();  // Start the fourth IR receiver

  myServo.attach(SERVO_PIN);  // Attach the servo motor to the specified pin
}

   if (results2.value == 0xDE3AA631) {
     rotateServo(45);  // Rotate the servo to 45 degrees
   }
   else if (results2.value ==  0x76A77416) {
     rotateServo(180);
   }
   else if (results2.value == 0x6212207) {
     rotateServo(0);  // Rotate the servo to 90 degrees
   }

   else if (results2.value == 0x69093291) {
     rotateServo(90);  // Rotate the servo to 135 degrees
   }

   irrecv2.resume();  // Receive the next value from Receiver 2
 }

 if (irrecv3.decode(&results3)) {
   Serial.print("IR Code from Receiver 3: 0x");
   Serial.println(results3.value, HEX);

   // Check for specific IR code from Receiver 3 and perform corresponding task
   if (results3.value == 0x69093291) {
     rotateServo(90);  // Rotate the servo to 90 degrees
   }
   else if (results3.value == 0xDE3AA631) {
     rotateServo(45);  // Rotate the servo to 45 degrees
   }
   else if (results3.value ==  0x76A77416) {
     rotateServo(180);
```

```
void rotateServo(int angle) {
  // Rotate the servo to the specified angle
  myServo.write(angle);
  delay(500);  // Optional delay to allow the servo to reach the desired position
}

void loop() {
  if (irrecv1.decode(&results1)) {
    Serial.print("IR Code from Receiver 1: 0x");
    Serial.println(results1.value, HEX);

    // Check for specific IR code from Receiver 1 and perform corresponding task
    if (results1.value == 0x6212207) {
      rotateServo(0);  // Rotate the servo to 90 degrees
    }
    else if (results1.value ==  0x76A77416) {
      rotateServo(180);
    }
    else if (results1.value == 0xDE3AA631) {
      rotateServo(45);
    }
    else if (results1.value == 0x69093291) {
      rotateServo(90);  // Rotate the servo to 135 degrees
    }

    irrecv1.resume();  // Receive the next value from Receiver 1
  }

  if (irrecv2.decode(&results2)) {
    Serial.print("IR Code from Receiver 2: 0x");
```

```
3D9AE3F7
FFFFFFFF
FFFFFFFF
488F3CBB
FFFFFFFF
C1BF9A80
FFFFFFFF
FFFFFFFF
CA03D2FC
FFFFFFFF
FFFFFFFF
FFFFFFFF
FFFFFFFF
```

Fig 6 Arduino Code

- ✓ (Mail us while requesting for access at ydextreme5@gmail.com)
- ✓ (Change the HEX Codes as per displayed in serial monitor from your device)
- ✓ The program that runs on the Arduino microcontroller, handling IR signal processing and servo motor control. Assign individual code for every button of your remote.)

- *IR Signal Decoding Library (e.g., IRremote):*

A library for Arduino that facilitates the decoding of infrared signals from the remote control.

- *Servo Motor Control Library:[2]*

A library for Arduino that simplifies the control of servo motors.

- *User Interface (Optional):*

If applicable, software or code for a user interface that may be displayed on a computer or mobile device.

- *Mechanical Components (if needed):*
- *Mounting Hardware:*

Brackets, screws, and other hardware for securing the servo motor in place.

- *Mechanical Linkage (If Applicable):*

Connecting elements to translate the rotational movement of the servo motor into a specific mechanical action.

➢ *Procedure :*

- *Mount the Hardware as per below figure.*

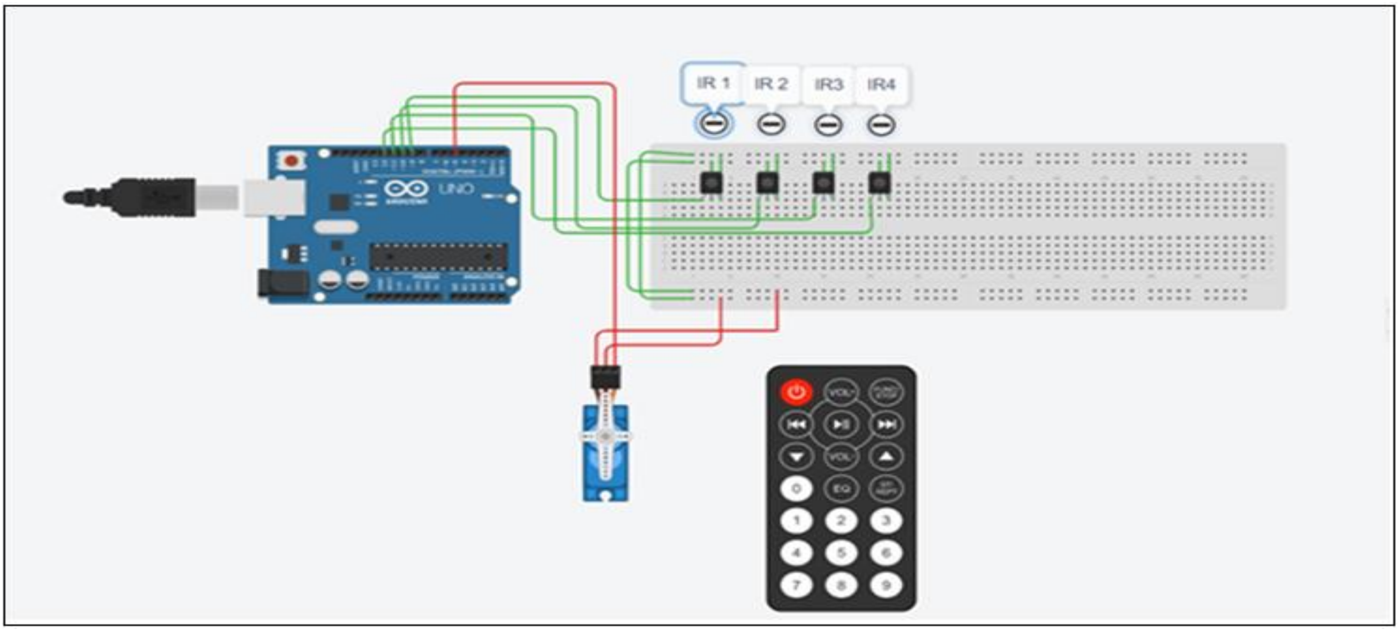


Fig 7 Mount the Hardware

- *Upload the Above Code in Arduino (Make changes as per the HEX code you get into the code, provide separate HEX codes for every IR receiver, then the motor will rotate as per the HEX code and the button you press on the remote. Additional: You can even use Mobile application of remote to control the motor.)*
- *You can make changes according to your choice for angles, baud rate, etc in the code.*

## IV. DATA ANALYSIS TOOLS

➢ *For the Data Analysis of the Project Involving the Integration of IR Remote Control with Arduino-based Systems and Servo Motors, Several Tools and Instruments can be Employed:*

- *Arduino IDE:*

The Arduino Integrated Development Environment is essential for programming and uploading code to the Arduino microcontroller. It allows for the development and debugging of the software controlling the system.

- *IR Protocol Analyzers:*

Specialized tools for analyzing and decoding IR signals can assist in understanding the communication protocols between the IR remote control and the system.

- *Oscilloscope:*

An oscilloscope can be used to visualize and analyze electronic signals, ensuring the proper functioning of the communication between components.

- *Servo Motor Tester:*

A servo motor tester allows for the calibration and testing of servo motors to ensure accurate and controlled movements.

- *Sensors and Actuators:*

Various sensors and actuators (e.g., position sensors, temperature sensors) may be used to collect data on the performance and environmental interactions of the system.

- *Multimeter:*
  Multimeter helps measure voltage, current, and resistance, providing crucial information about the electrical characteristics of the components.

- *Data Logging Tools:*
  Software or hardware tools for data logging can be employed to record and analyze data over time, allowing for the assessment of system behavior and performance.

- *Simulation Software:*
  Simulation tools can be used to model and simulate the behavior of the system before actual implementation, helping in predicting and understanding potential issues.

- *Statistical Software:*
  Statistical analysis software (e.g., R, Python with NumPy and Pandas) can be employed for analyzing and interpreting data, especially in cases where extensive data sets are involved.

## V. DISCUSSION

➢ *Arduino IDE:*

- *Role:*
  Programs and fine-tunes Arduino for controlling servo motors.

- *Contribution:*
  Essential for customizing how Arduino behaves to achieve precise servo motor movements.

➢ *IR Protocol Analyzers:*

- *Role:*
  Decodes IR signals, ensuring proper communication with the remote.

- *Contribution:*
  Ensures the system understands and responds accurately to IR commands.

➢ *Oscilloscope:*

- *Role:*
  Checks electronic signals for smooth communication.

- *Contribution:*
  Helps identify and fix signal issues, ensuring reliable communication.

➢ *Servo Motor Tester:*
Role: Calibrates and tests servo motors.
Contribution: Ensures servo motors move accurately, validating their performance.

➢ *Sensors and Actuators:*

- *Role:*
  Collects real-world data on system performance.
- *Contribution:*
  Helps understand how the system reacts to different conditions.

➢ *Multimeter:*

- *Role:*
  Measures electrical characteristics.

- *Contribution:*
  Checks components to make sure they're working within limits.

➢ *Data Logging Tools:*

- *Role:*
  Records and analyzes data over time.

- *Contribution:*
  Helps spot trends and potential issues in how the system behaves.

➢ *Simulation Software:*

- *Role:*
  Models system behavior before actual tests.

- *Contribution:*
  Predicts and understands potential problems, making design better.

➢ *Statistical Software:*

- *Role:*
  Analyzes data for better understanding.

- *Contribution:*
  Provides numbers to support conclusions about how well the system works.

➢ *Documentation Tools:*

- *Role:*
  Records everything about the project.

- *Contribution:*
  Keeps clear and organized records for others to understand and replicate the work.

## VI. CONCLUSION

The IR Navigator Positioning System, blending IR remote control with servo motors, offers a flexible solution for precise motion. It's useful in robotics, automation, and interactive settings, highlighting its adaptability and significant potential.

## REFERENCES

[1]. Wikipedia Remote Control https://en.wikipedia.org/wiki/Remote_control

[2]. Wikipedia Servo Motor https://en.wikipedia.org/wiki/Servomotor

[3]. Instructables Webpage https://www.instructables.com/Controlling-servo-motor-using-IR-remote-control/

[4]. Sai, C.P.R., Datta, V.S. and Sudheera (2016) Design of a Smart Remote. 2016 Inter-national Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, 18-19 March 2016, 1-4.

[5]. ChatGPT for Code

[6]. Automatic Antenna Positioning system/researchgate https://www.researchgate.net/publication/343570017_Automatic_Antenna_Positioning_System.