IT & BUSINESS COLLEGE

Department: COMPUTER SCIENCE

Project Title : IR REMOTE CONTROLLED ROBOTIC ARM

Group Name: MK Team - 3

Team members:

Maryam Kasymova: 238715063

Bayel Manapov: 238715069

Azim Albarsbekov: 238715053

Amangeldi Aidaraliev: 238715056

Instructor: Mirlan Nurbekov

Date: 16.12.2025

Fall Semester of 2025/2026

Table of contents

Table of figures........3

ABSTRACT........4

1.INTRODUCTION........5

1.1 The history of the appearance of the robot arm........5

1.2 Types of robots today........6

1.2.1 Humanoid robots........6

1.2.2 Industrial robots........6

1.2.3 Agricultural robots........7

1.2.4 Cobots........7

2. PROBLEM STATEMENT AND OBJECTIVES........8

2.1 Problem statement........8

2.2 Objectives........8

3. SYSTEM REQUIREMENTS & CONSTRAINTS........9

3.1 Functional requirements........9

3.2. Constraints........9

3.2.1 Non-functional requirements........9

3.2.2 Mechanical constraints........9

3.2.3 Safety........9

4.SYSTEM OVERVIEW........10

4.1 Hardware Design........10

4.1.1 Main hardware blocks........10

4.1.2 Signal flow........11

4.2 Software design........11

5.1 Prototype 0000 - Basic servo test........14

5.2 Prototype 0001 - IR communication setup........14

5.3 Prototype 0002 - IR control with servo........15

5.4 Prototype 0003 - Full multi-servo system with external power........16

6.IMPLEMENTATION........17

7.DISCUSSION........19

7.1 What worked well........19

7.2 Problems we faced........19

7.3 Trade-offs we made........20

8.CONCLUSION........21

9.REFERENCES........22

10.APPENDICES........23

Table of figures

Figure 1- Humanoid robot 6
Figure 2 - Industrial Robot 6
Figure 3 : Agricultural robot 7
Figure 4: Cobot 7
Figure 5: Infrared Remote → IR receiver → Microcontroller (Arduino) → Actuators (servomotors) → Robotic arm movements 10
Figure 6: Signal flow of robotic arm 11
Figure 7: Flowchart for robotic arm software 11
Figure 8 12
Figure 9 13
Figure 10 13
Figure 11 13
Figure 12: Circuit diagram of prototype 0000 14
Figure 13: Circuit diagram of prototype 0001 14
Figure 14: Circuit diagram of prototype 0002 15
Figure 15: Circuit diagram of prototype 0003 16
Figure 16 17
Figure 17 17
Figure 18 17
Figure 19 17
Figure 20 18
Figure 21 18
Figure 22 18
Figure 23: Power supply with powerbanks 19
Figure 24: Final project 20

ABSTRACT

This report presents the design and control of an Robotic arm using an Arduino microcontroller and infrared remote control, and receiver. .The aim of our project is to ensure safety for workers in industrial organizations, as robotic arms are suitable for performing repetitive, constant actions that require high precision, as well as for tasks that can be difficult for humans to perform safely, such as welding, metal cutting, working with chemicals, lifting heavy objects, and other tasks. Robotic arm, Arduino UNO R3, IR Remote control and SG90 servo motors are key components in the design of the robotic arm. The report describes process of creating our robotic arm and how it works. As part of the development we also learned how to work with an IR remote control and an IR receiver, and were able to design our robotic arm. Our project aims to show how useful technology can be in an industrial environment.

1.INTRODUCTION

A robotic arm is a machine designed to perform tasks that resemble those performed by a human hand. Many robotic arms are equipped with programmable controls and are highly precise in their actions. It consists of segments connected by hinges, which usually allows it to move in any direction.

Robotic arms are used in various industries, including manufacturing, medicine, and research institutions. These manipulators are used in production to assemble cars by painting their surfaces or joining various parts by welding.In manufacturing, they are used to assemble vehicles such as automobiles by welding parts together or painting surfaces. In surgery, they can help surgeons perform precise movements during low-trauma surgery that does not require large incisions and minimal intervention.

The key components of a robotic arm are actuators that actuate the arm, sensors that evaluate position or force, a controller that processes commands, and the final actuator, a tool attached to the tip of the arm such as a gripper or camera. The performance of a robotic arm depends on its control system and kinematic scheme.The controller operates according to a pre-written program, usually accepting high-order commands such as "move to X, Y, Z coordinates" and converting them into signals for actuators. This procedure requires solving kinematic equations, including the inverse kinematic problem: finding the connection angles at which the end actuator is installed in the desired position.

For example, when a robotic arm needs to lift a certain object, it must calculate how all its joints should rotate and stretch in order to reach that object. There are sensor systems such as encoders and torque sensors that provide instant feedback to correct inaccuracies in the selection of readings due to slippage or unforeseen obstacles. These robots are also usually coded by software developers using ROS interfaces, if any. They provide trajectory programming and collision avoidance logic, which increases efficiency and accuracy. The reliability of robotic manipulators is demonstrated in the following use cases: sorting packages in a warehouse using barcodes and then placing them on conveyor belts, performing repetitive tasks such as packaging liquids in test tubes. For a developer, integrating a robotic arm usually involves writing scripts for motion trajectories, calibrating sensors, and testing safety protocols to ensure reliable operation in a higher-quality environment.

1.1 The history of the appearance of the robot arm

In 1961, Unimate introduced the very first industrial robotic arm. Joseph Engelberger and George Devol are to thank for this. Devol developed the patent that was granted in 1961. Engelberger functioned as the entrepreneurial half of the duo who founded Unimation. Thus, while Devol was the one who conceived the idea for the arm, Engelberger was thought to be the “father of the robot”. This revolutionary invention featured a 3-axis arm that could handle parts up to 226 kg. Equipped with hydraulic actuators, the Unimate was able to hold force constant due to the incompressibility of fluids (Inst Tools, 2019). The robot promised the ability to execute multiple tasks from material handling to welding. However, the invention proved to be a double-edged sword. While it alleviated the work hazards for humans, it created another problem of mass unemployment if adopted. Thus, companies hesitated to move forward with the technology.

1.2 Types of robots today

1.2.1 Humanoid robots

Humanoid robots created in a way that they resemble humans both in their shape and functionality. They usually contain features similar to those of human beings sort of the head, torso, arms, and legs. Such robots apply actuators for the reproduction of muscle and joint movements and other sensors, which enable them to perceive and interact with the environment autonomously.Algorithms are based on a fusion of engineering and cognitive sciences, enabling them to further mimic human capabilities. Humanoid robots are used in scientific research, healthcare, customer service, entertainment, and other fields where human-like interaction is required.

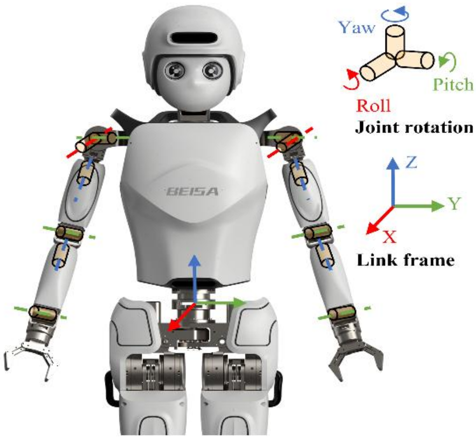

Figure 1- Humanoid robot

1.2.2 Industrial robots

Industrial robots are designed to automate large scale production operations. They are usually installed in factories and are used to perform repetitive tasks quickly and accurately. Their large-sized devices usually consist of at least one arm-like manipulator capable of moving along three or more axes. They are programmed to perform industrial functions such as welding, painting, assembly, packaging, and handling.

Figure 2 - Industrial Robot

1.2.3 Agricultural robots

Agri-bots or agricultural robots are automated machines built to assist in various agriculture and farming tasks They are usually employed in fields to automate repetitive, labor intensive agricultural activities.For instance harvesting, watering, planting, weeding, fertilizing, and crop monitoring. These robots are

equipped with advanced technologies such as sensors, GPS and artificial intelligence to perform various tasks in precision farming.

Figure 3 : Agricultural robot

1.2.4 Cobots

Cobots, also referred to as collaborative robots designed to work alongside humans in a common working environment. Unlike traditional robots, cobots are fitted with high tech sensors that provide a safer working environment allowing robots and human beings to work side by side. Some of the tasks that can be performed by cobots include assembly, equipment choice, quality checks, and product testing. And also these robots can be designed to facilitate a flexible manufacturing system that eliminates mundane, repetitious, or heavy tasks. Their flexibility in working with industrial technological processes makes them ideal for small scale production and a dynamic industrial setting.

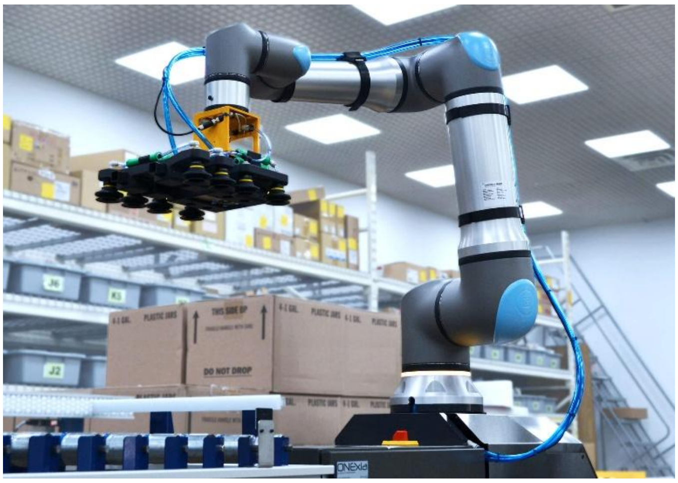

Figure 4: Cobot

2. PROBLEM STATEMENT AND OBJECTIVES

2.1 Problem statement

Our project is aimed at a more simplified version of an industrial-type robotic arm. Throughout the period of study, all the parts of robots name it the body structure, controller, joints, actuators for movement. Power supply and the end-effector are designed and finally integrated into one working robot. The robot is developed part by part to ensure all parts are working before the components are combined together. This project is significant as it develops the fundamentals of robotic systems that is needed by the university to further the study on this exciting area of electrical and electronics engmeenng.A robotic arm can be implement and used it in small industrial enterprises in Kyrgyzstan and scale our project as much as possible. opportunities to masterstab our project.The robotic arm can also be used to help people in critical situations, which will help people with disabilities or the elderly in their daily lives.

2.2 Objectives

The aim of project work was developing and applying skills necessary for creating a functioning prototype of a robotic arm capable of executing basic as well as more intricate commands. We have short-listed the tasks relating to the project, and these are:

- Develop a robot controlled by an infrared remote control.
- Develop a simple layout design that includes a base, joints, and a grip.
- Design the robot in such a way that it is convenient to use at an affordable cost.

3. SYSTEM REQUIREMENTS & CONSTRAINTS

3.1 Functional requirements

The functional requirements describe what the robotic arm must be able to do: The system shall receive instructions from the infrared remote, go through processing via the Arduino controller, and execute the prescribed movements. It has to further enable the robotic arm to make rotational movements at the joints, elevate/lower its segments, and perform end-effector movements that would be instructed by the user. Execute these movements in a smooth and articulate manner, respond accurately to the transmitted signal, and be stable during operation.

3.2. Constraints

3.2.1 Non-functional requirements

These are conditions that are not related to functions, but are important:

Range - IR remote controls require a clear line of sight to the receiving device and their range maxes out at about 30 feet

Latency- The response time from sending a command to the arm performing the movement is less than 0.5 seconds, ensuring smooth and accurate operation

Power limits - A 9v battery or USB power supply is sufficient for normal operation.

Cost limits -The total cost of the mini robot arm remains low, in the range of 200-300 KGS, due to the use of affordable components such as cardboard for the manufacture of structures, micro servos and an Arduino board.

3.2.2 Mechanical constraints

Mechanical constraints is defines the limitations and characteristics of robotic arm. Total height of the robotic arm is 20 cm, and the total weight of the structure is 500 grams. It stable while making movements. The cardboard used here light and stable. The structure designed in such a manner that all the components will be stable.

3.2.3 Safety

The safety constraints make sure that there are no damages caused by the robotic arm. The maximum voltage level allowed for safety considerations is 5V for the arduino and 6V for the servos. The current restricted within safety limits so as not to cause overheating. All components involving motion made with safety in mind so as not to harm anyone, and it will be capable of automatically shutting down with overload and short circuits.

4.SYSTEM OVERVIEW

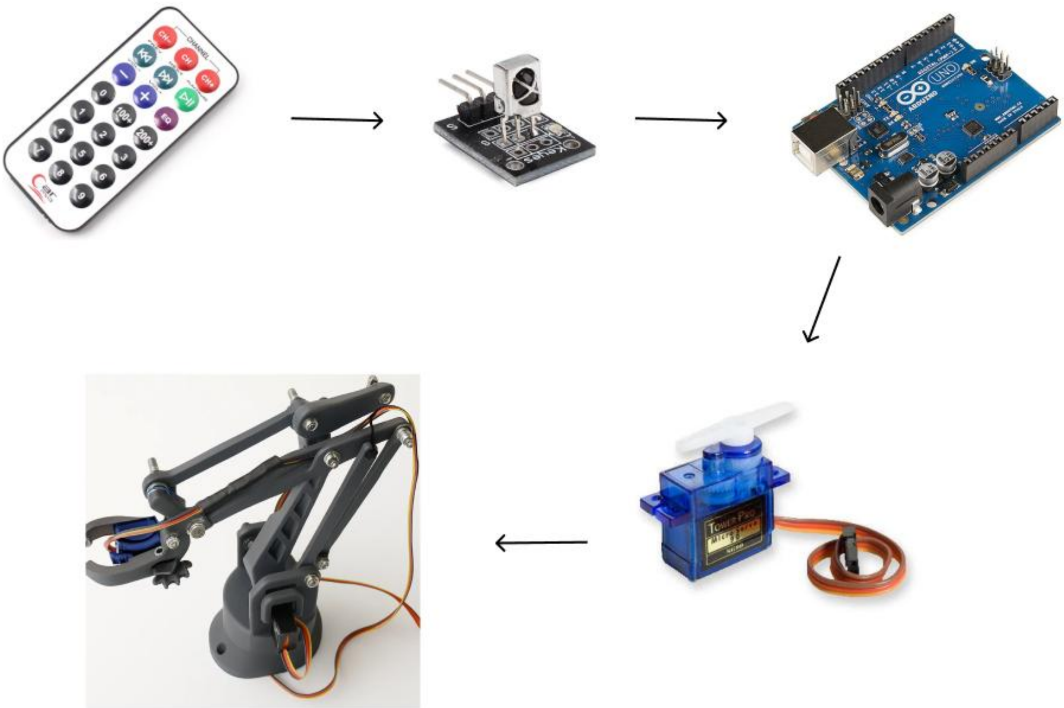

Figure 5: Infrared Remote → IR receiver → Microcontroller (Arduino) → Actuators (servomotors) → Robotic arm movements

4.1 Hardware Design

4.1.1 Main hardware blocks

1. Controller - Arduino
2. Communication module - Infrared receiver
3. Input device - Infrared remote
4. Actuators- Servo motors
5. Power supply - 9V battery

4.1.2 Signal flow

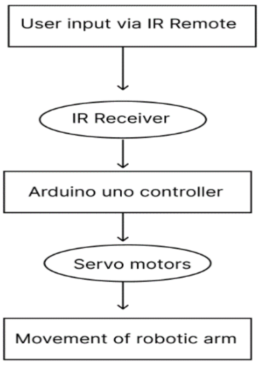

Figure 6: Signal flow of robotic arm

4.2 Software design

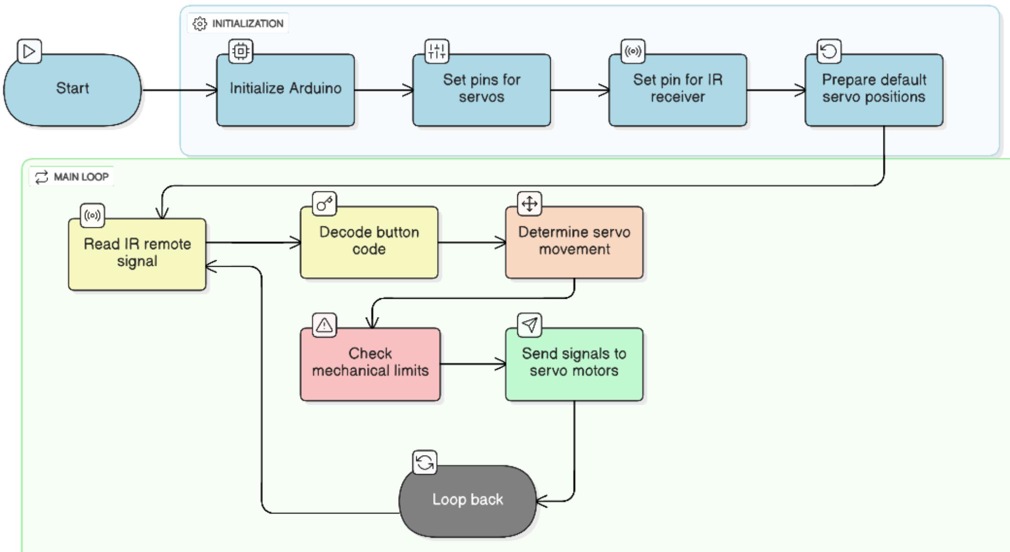

Figure 7: Flowchart for robotic arm software

Explanation to Figure 7

The code for our robotic arm enables it to listen to commands from an IR remote and consequently drive servo motors. At the start, Arduino initializes the system by setting up pins for servos and IR receiver, preparing default positions of the arm joints. The Arduino continuously reads the signal from the IR remote through the IR receiver. Each button has a unique code, which, upon being decoded, will tell the software what movement is required.

The software, after decoding, decides which servos to move and in which direction. Safety checks are carried out in order to prevent damage by restricting the angles the servos will reach within their mechanical limit. This finally sends the signals from the Arduino to the servo motors in order to execute the movement. It enters into this loop continuously so that it shall be able to serve the arm instantaneously based on commands from the IR remote**.**

Breaf explanation to code:

Our mobile application is created to control a robotic arm and it developed with Flutter library for Dart language. Application works as remote controller to our Arduino project(robotic arm) and it allows user to control it by sending signals-commands to Bluetooth module that connected to Arduino. The goal of application is to let user easily control robotic arm with 2 joysticks and switcher that enables or disables control. After application's start, Flutter initializes user interface and displays main screen of application. There are 2 joysticks and 1 switcher with text: "Turn on/off". Our application now waits for user signal such as finger movement on any of joysticks and clicking on switch button.

Switch's role in application is to prevent any misclicks of user. When switch is turned off all joysticks are inactive and any of their movement are being ignored, that means that it will not send any signal to Arduino Bluetooth module. Otherwise if it is turned on it allows joysticks transfer data through connection to Arduino Bluetooth module.

Now, lets talk about Bluetooth connection between mobile phone and Arduino Bluetooth module such as HC-05 and HC-06. When it connected application sends signals, that transfers data of control in real time. Every movement of joysticks is converting into digital data, that transmitted by data packets to Arduino, then in Arduino it processes and gives commands to servo motors.

Left joystick is used to control one group of movements: forward, backward and left, right; while right one is used to control other group: up, down and open, close. Each joystick detects position of user's finger from center. Then this positions convert to X and Y axis values, which range from -1 to 1.

Each time user moves a finger on joystick, application updates joysticks positions and repeatedly sends new values, that happen multiple times per second. When user is not touching screen or just released finger from joystick, it automatically returns joystick position to its center and sends new zero values, that indicates as a stop command.

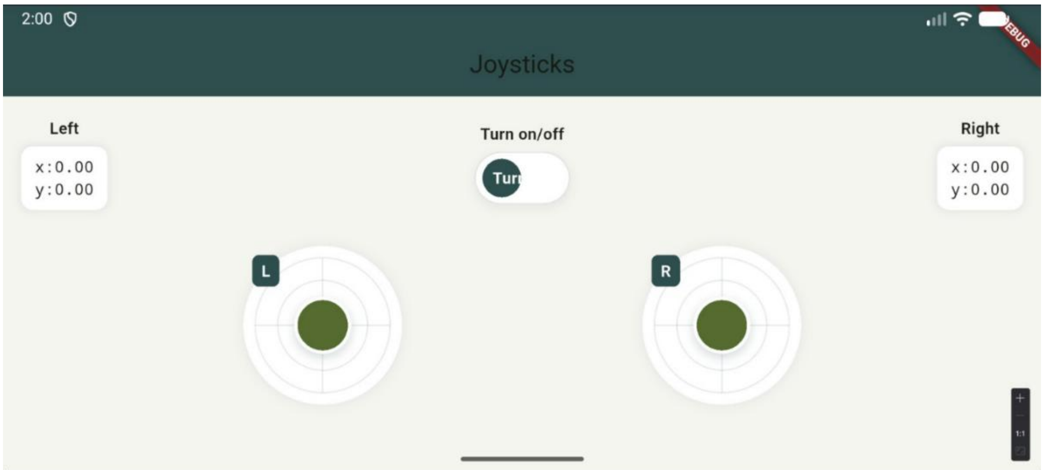

Figure 8

This screenshot shows us the overall layout of application.

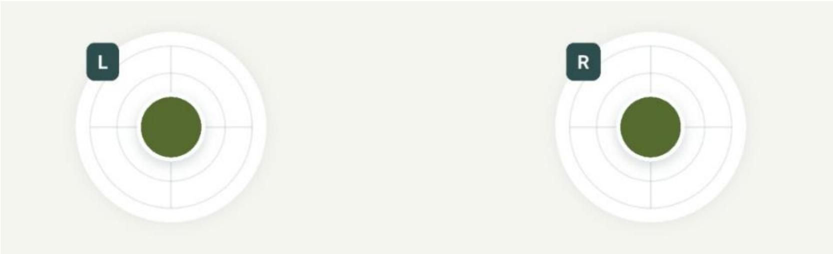

Figure 9

This screenshot shows us joysticks, that is used to control movement of robotic arm.

Figure 10

This screenshot shows us current coordinates of joystick(in example is the left one), but if switch is disabled it don't show use any coordinates except 0.00.

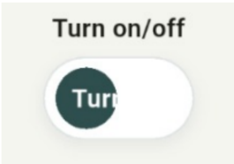

Figure 11

This screenshot shows us switch, that currently disabled.

In total, the application functions as a real-time Bluetooth controller for Arduino based robotic arm. Flutter is used to show the interface and receive input data, while Bluetooth communication is used to transfer received data to Arduino. The use of switch, joystick's centralized control and range of axis makes system reliable and scalable, that makes it good for controlling robotic arm.

5.PROTOTYPE VERSIONS & PROGRESS

5.1 Prototype 0000 - Basic servo test

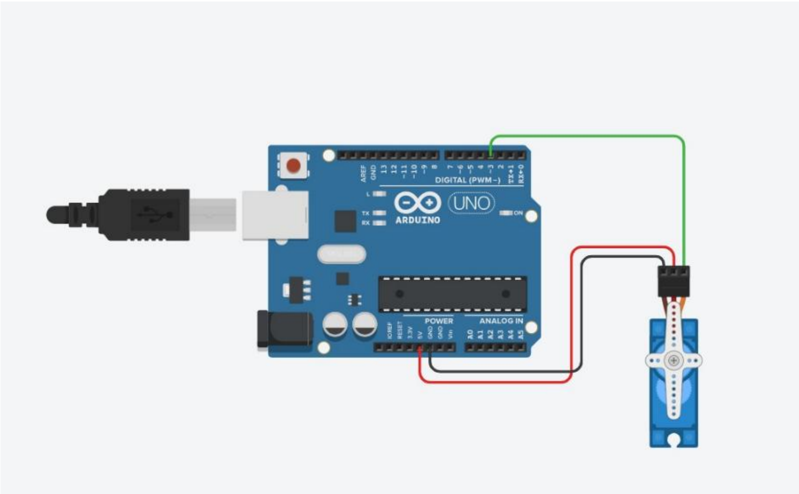

Figure 12: Circuit diagram of prototype 0000

Portuse: The purpose of this version was to verify that the Arduino board could successfully control a single servo motor.

Changes: In this prototype, Arduino was connected with one servo motor in order to test its basic operation. The initial movement tests showed that the microcontroller and the servo were functioning correctly, establishing a stable foundation for the next stages of integration.

5.2 Prototype 0001 - IR communication setup

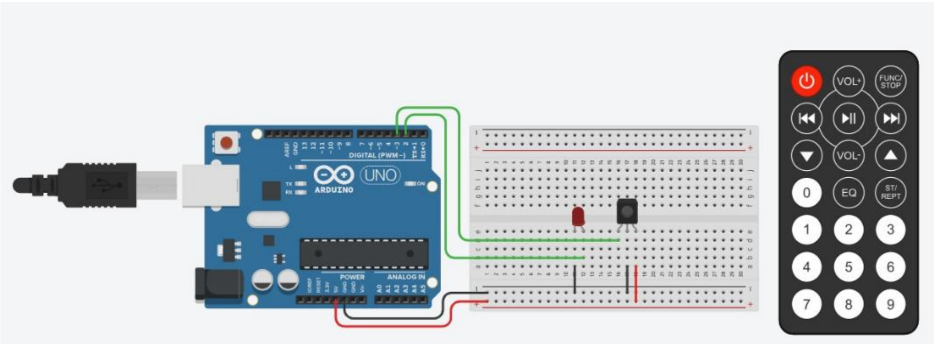

Figure 13: Circuit diagram of prototype 0001

Purpose: This version aimed to set up infrared communication for future remote-controlled operation.

Changes: The IR remote and the IR receiver were connected to the Arduino, and the first IR signals were successfully read and decoded. Although no actuators were attached yet, this prototype confirmed that remote commands could be reliably received, preparing the system for combined control.

5.3 Prototype 0002 - IR control with servo

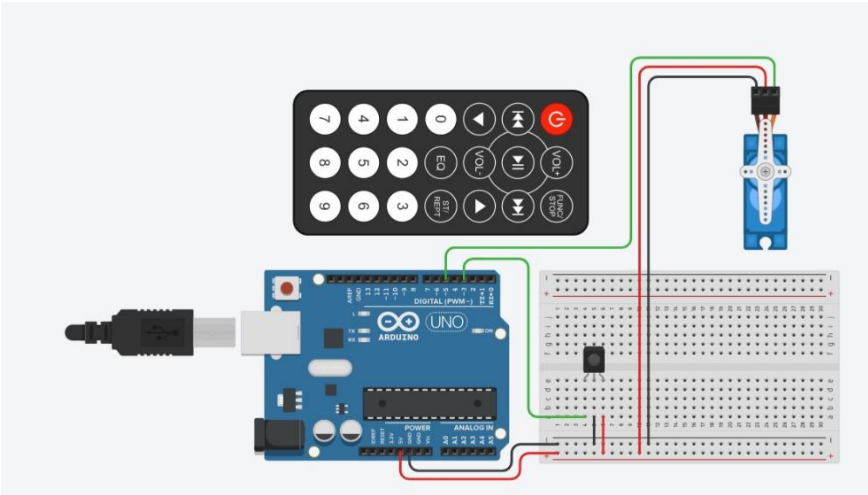

Figure 14: Circuit diagram of prototype 0002

Purpose: This version focused on integrating IR communication with mechanical actuation.

Changes: IR remote and IR receiver were connected together with a single servo motor, and Arduino was programmed to respond to IR commands by moving the servo. This prototype achieved the system’s first fully controlled movements, demonstrating stable communication and accurate servo response.

5.4 Prototype 0003 - Full multi-servo system with external power

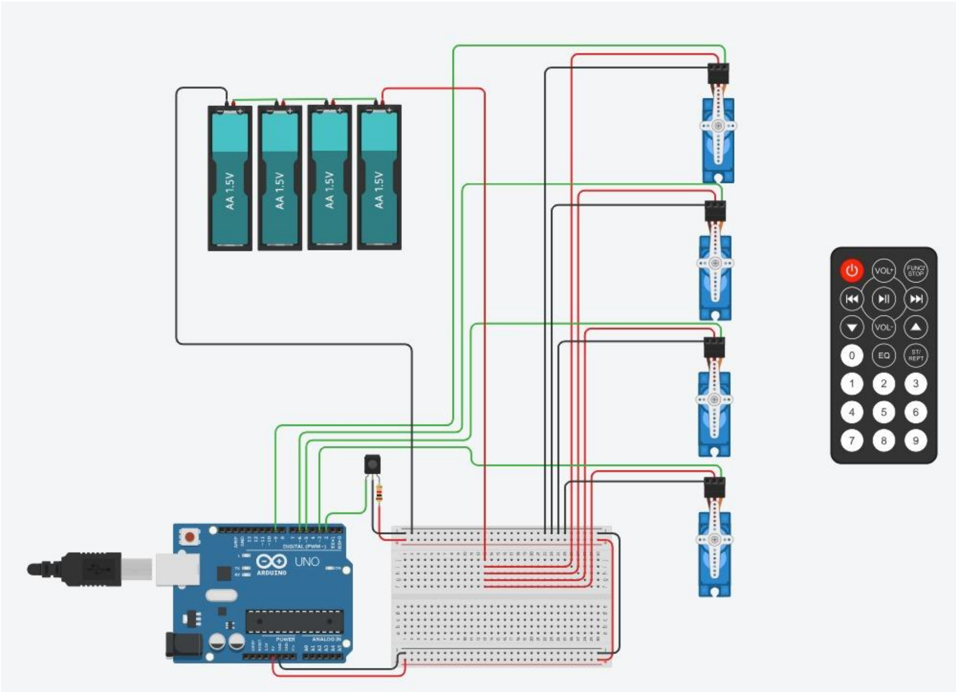

Figure 15: Circuit diagram of prototype 0003

Purpose: The goal of this version was to build a complete multi-servo robotic arm and ensure stable power delivery.

Changes: In this version, IR remote and receiver were integrated with four servo motors, creating a fully functional control system. An external power supply of four 1.5V batteries was added because Arduino could not safely power multiple servos and the receiver simultaneously. This upgrade significantly improved system stability and allowed all four servos to operate smoothly without voltage drops.

6.IMPLEMENTATION

```
if (command == 0x9) {   // кнопка "EQ(turn ON/OFF)"
  isActive = !isActive;
  delay(500);
}
```

Figure 16

This function changes state. By using this function we turn on/off control.

```
if (command == 0x43) {   // кнопка "pause"
  isRunning = false;
  stepIndex = 0;
}
```

Figure 17

This function stops another function that do some commands repetitively.

```
if (command == 0x18) {   // кнопка "2"
  pos1 = pos1 - step;
  pos1 = constrain(pos1, 0, 180);
  myServo1.write(pos1);
}
if (command == 0x5E) {   // кнопка "3"
  pos1 = pos1 + step;
  pos1 = constrain(pos1, 0, 180);
  myServo1.write(pos1);
}
```

Figure 18

This function turns servo №1 right and left. Function “constrain()” is used to prevent servo motor’s tries to turn more than they can. We also have another 3 same functions which turn other 3 servo motors.

```
if (command == 0x16) {   // кнопка "0"
  pos1 = 90;
  pos2 = 90;
  pos3 = 90;
  pos4 = 90;
  step = 7;
  myServo1.write(pos1);
  myServo2.write(pos2);
  myServo3.write(pos3);
  myServo4.write(pos4);
}
```

Figure 19

This function sets all servo motors to default position and also sets step to default value.

```
if (command == 0x15) {   // кнопка "+"
  step++;
}
if (command == 0x7) {   // кнопка "-"
  step--;
}
```

Figure 20

This function increases and decreases step. Step is servo motor's move speed.

```
if (command == 0x19) {   // кнопка "100+"
  save1pos1 = pos1;
  save1pos2 = pos2;
  save1pos3 = pos3;
  save1pos4 = pos4;
}
if (command == 0xD) {   // кнопка "200+"
  pos1 = save1pos1;
  pos2 = save1pos2;
  pos3 = save1pos3;
  pos4 = save1pos4;

  myServo1.write(pos1);
  myServo2.write(pos2);
  myServo3.write(pos3);
  myServo4.write(pos4);
}
```

Figure 21

This function saves and loads servo motors's positions. We also have another one same function in order to have additional slot for saved postion.

```
if (command == 0x45) {   // кнопка "CM-"
  seqChoise = ((seqChoise + (maxChoises - 2)) % maxChoises) + 1;
}
if (command == 0x47) {   // кнопка "CM+"
  seqChoise = (seqChoise % maxChoises) + 1;
}
if (command == 0x46) {   // кнопка "CM (without any symbol)"
  startSequence(seqChoise);
}
```

Figure 22

This function changes repetitive command number and starts this repetitive command.

7.DISCUSSION

7.1 What worked well

Several components of our system functioned consistently and exceeded our expectations. First, the IR communication between the remote controller and the IR receiver worked reliably. Button presses were correctly detected and decoded, allowing smooth interaction between the user and the robotic arm. The servo motors also responded accurately to the received commands, maintaining stable movement without calibration issues.

The modular testing approach also proved effective. Starting from testing only one servo and then adding the components step by step helped us avoid complex debugging later on. Additionally, sharing the ground between all power sources and the Arduino improved system stability and prevented unexpected resets. After switching to a proper battery supply, the entire setup operated smoothly and showed consistent performance.

7.2 Problems we faced

The most serious problem which occurred in the prototype preparation was related to power supply. In the first attempts, we tried to power four servo motors by using two different powerbanks. Each powerbank powered two servo motors, sharing its ground with Arduino. That was somehow working but the servos were moving very slowly and did not have much torque. The powerbanks could not provide the required current spikes, resulting in a weak and inconsistent performance.

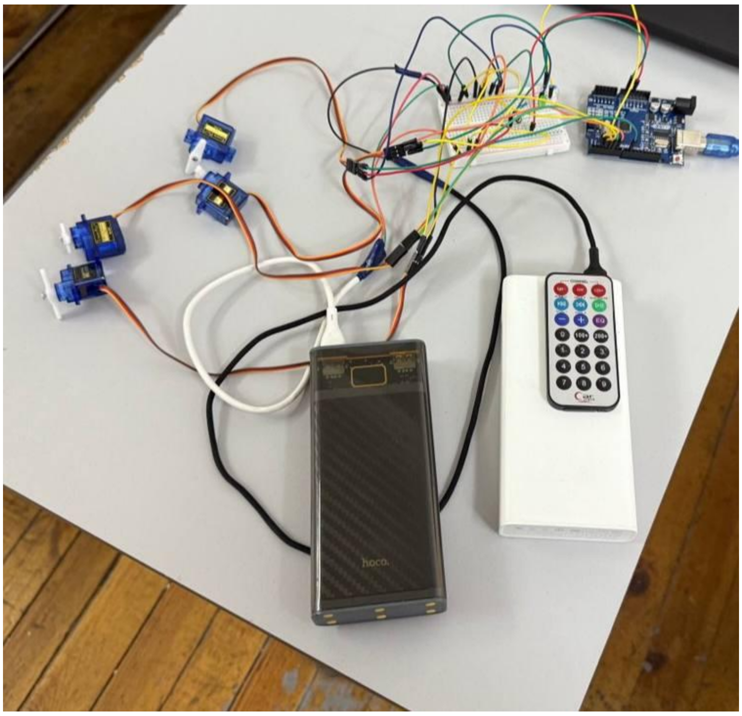

Figure 23: Power supply with powerbanks

Eventually, we changed the powerbanks with a 9V battery. The servo speed was much greater and more stable. After using a battery, the robotic arm operated normally; motors were able to move without any problems. We did not encounter serious problems with electrical noise nor sound interference. We did not face major delays: the system responded promptly to most of the IR commands.

7.3 Trade-offs we made

1. First phase - Single servo test

We started by connecting only one servo motor to Arduino. This was just to verify that the microcontroller would handle basic servo movement and angle setting.

2. Adding the IR receiver

Then, we added IR receiver and tested that it could read and decode the signals from the IR remote successfully. Then we tested the communication in isolation before hooking it up to a servo motor.

3. Integration of IR control with servo movement

Once both parts were verified to work independently, we combined them. The single servo was responding according to IR button presses, providing for full basic functionality.

4. Going to four servos

We then attached all four servo motors and tried powering them with two powerbanks, one powerbank per two servos. That worked fine but yielded very slow and weak motor movement due to a small value of current supply.

5. Conversion to a 9V battery

After replacing two power banks with a 9V battery servo performance became much better. From now on, robotic arm became fully functioning.

Final stage - Complete prototype

We combined our test's final version: Arduino, IR receiver, IR remote, four servos, and the 9V battery into a full robotic arm system that is complete and stable. All modules communicated correctly; the arm executed its pre-set movements.

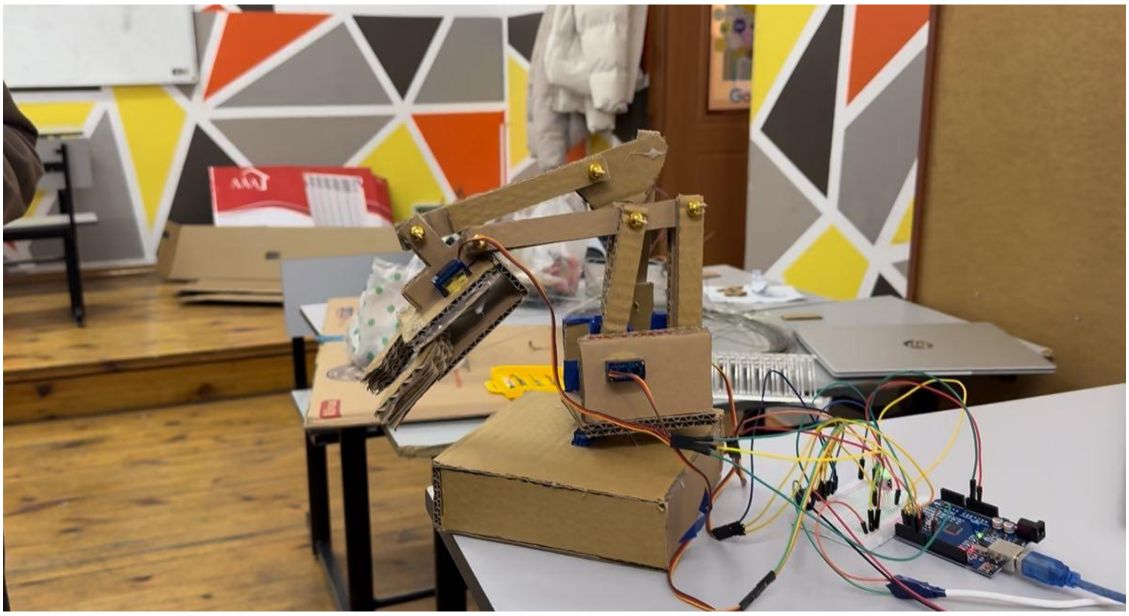

Figure 24: Final project

8.CONCLUSION

The design of a robotic arm, which could be used by an individual, was successfully demonstrated as part of this project. In this, the aim is to design a robot arm which is easy to fabricate, feasible, and of low cost. The project involved the use of both hardware and software in implementing the functions and controls of the robotic arm. System architecture for the robotic arm is designed in a manner that makes interaction between its components smooth. Cardboard was the material of the mock-up robotic arm. The robotic arm consists of six main parts: base ,shoulder,elbow,wrist, end effector.

Some coding was done through programming the Arduino by means of the Arduino IDE along with C-based language. In this code, the IR remote control interacts with, resulting in simplification of interaction and further communication. Arduino takes control by ensuring that the servos move in the appropriate fashion according to the given instructions. It allowed controlling each servo individually in real time and the execution of predefined actions. the sequence of motions the servo went through. It also had the feature of saving and memorizing certain positions of the servo were enabled, allowing custom motion pattern creation for the robotic arm. The project as a whole has really demonstrated design, construction, and management. Robotic arm for individual use. Such a robotic arm can be used in many different ways: this could include educational purposes, minor automation tasks, and hobby projects.

Following are the improvements that can be done in order to improve the working and dependability of the system. It is necessary first of all to improve the mechanical layout of the robotic arm. The prototype so far is made of cardboard material, which seriously constrains the strength, accuracy, and durability of the structure. Creating a new layout using 3D printing will increase the stability of the structure, improve joint alignment, and increase the service life of the device.

Moreover, it's possible to develop more complex and comprehensive control commands to make the movement of the robotic arm even smoother and more accurate. Besides, it is recommended to change SG90 servomotors for MG995 ones, which have more power, allowing you to work with bigger and heavier objects.

Finally, it's possible to enhance the management application by adding administrator functionality to view all users and manage the access rights for the control of the robotic arm, thus enhancing security and usability for this system.

9.REFERENCES

[1] Craig, J. J. *Introduction to Robotics: Mechanics and Control*. Pearson Education, 2018. https://www.pearson.com/en-us/subject-catalog/p/introduction-to-robotics-mechanics-and-control/P200000003295

[2] Arduino. *Servo Motor Control*. Arduino Official Website, 2023. https://www.arduino.cc/reference/en/libraries/servo/

[3] Monk, S. *Programming Arduino: Getting Started with Sketches*. McGraw-Hill Education, 2017. https://www.mheducation.com/highered/product/programming-arduino-monk/M9781259641633.html

[4] Augmentus. When Was the First Robotic Arm Made? The Evolution of Industrial Robots. Augmentus Blog, March 31, 2022. https://www.augmentus.tech/blog/the-history-of-robotic-arms/

[5] Instructables. *Controlling Servo Motor Using IR Remote Control*, 2021. https://www.instructables.com/Controlling-servo-motor-using-IR-remote-control/

10.APPENDICES

Github Link (https://github.com/AlbarsbekovAzim/Robot_arm_team.git) **and QR**

YouTube Link (https://youtu.be/xFCzmPis_qo?si=ejwiZYM4_fFpb9hU) **and QR**