Department of Computer Engineering and Applications

GLA University, Mathura

Declaration

We hereby declare that the work which is being presented in the Mini Project “**Robotic Arm"** in partial fulfilment of the requirements for Mini-Project, is an authentic record of our own work carried under the supervision of **Mr. Amir Khan,** Trainer Training and placement department**, GLA University, Mathura**.

Name of Students with signature

CERTIFICATE

This is to certify that the project entitled **“Robotic Arm”** carried out in Mini Project is a Bonafede work done by **Udit Nayyar (161500591), Ravindra Faujdar (161200173), Rishabh Mishra (161500494) and Prakhar Verma (161500393)** and is submitted in partial fulfilment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor:

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to **Dr. (Prof). Anand Singh Jalal,** Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe special debt of gratitude to **Mr. Amir Khan,** Assistant Professor Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Abstract

Today, technology is developing in the same direction in line with rapidly increasing human needs. The work done to meet these needs makes life easier every day, and these studies are concentrated in robotic arm studies. Robot arms work with an outside user or by performing predetermined commands. Nowadays, the most developed field of robot arms in every field is the industry and medicine sector. Designed and realized in the project, the robot arm has the ability to move in 4 axis directions with 4 servo motors. Thanks to the holder, you can take the desired material from one place and carry it to another place, and also mix it with the material it receives.

Table of Contents

Declaration
Certificate
Acknowledgments
Table of Contents

- **Introduction**
  - Objective
  - Need for Robotic Arm

- **Requirement Analysis**
  - Arduino
  - Servo Motor
  - Breadboard
  - Potentiometer

- **Testing**
  - Generate test cases for Black box
  - Generate test cases for white box

References/Bibliography
- **Appendices**

Coding /Code Templates

INTRODUCTION

These days people always needed additional help systems. With the rapid increase in the flow of information, people are now guided to search for different markets and people have entered the competition to manufacture quality products cheaply. Automation systems are also needed to minimize errors as well as to have experienced and well trained employees for quality products. Because of their physical characteristics, people needed to use auxiliary machines in places where their strength was not enough. These machines, which are operated with the need for human assistance in advance, have been made to operate spontaneously without the need of human power with the progress of technology. One of the most components of automation systems in robots. Robotic systems; Mechatronics Engineering, Mechanical Engineering, Electrical Engineering and computer Engineering have all come together to work.

In the Project, researchers have been done and implemented in order to have knowledge about mechanics and software during the operations carried out by the robotic arm which is designed to fulfill the tasks determined in accordance with predetermined commands.

First, it was determined what function the robotic arm would be and what movements it could make. Robotic arm made of Arduino or servo motors; it can carry the desired material, mix it up and perform the commands previously determined by a user. If this project is also a designated task; the robotic arm takes a piece of material and brings it to the desired positions and then records it movement and lets it do the same action until we stop it. The servo motor is preferred in order to order to be able to perform these operations properly since the motor to be selected must operate precisely and must be at high torque. The robotic arm is composed of 4 servo motors and can move in 4 axis directions with these motors.

OBJECTIVE

The main objectives of the project are:

1. To be able to design and construct a robotic arm.

2. To be able to control the robotic arm using a potentiometer with simple commands.

The first objective is very straight forward and it requires modern designing capacities. The complete robotic arm was first designed and assembled in designing software. We have used wildfire

NEED OF ROBOTIC ARM

1. In the field of medical robotic arms are used in the operations.
2. In the food processing industry. They use them as case packers.
3. In the field of automotive for the assembling of vehicles.

Requirement Specification-

Our robotic arm is having following features and specification.

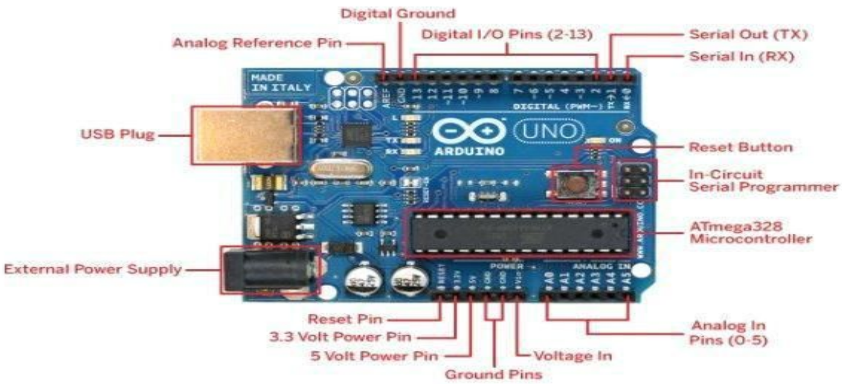
1. Degree of Freedom:5
2. Payload Capacity(Fully Extended):100 gm
3. Maximum Reach(Fully Extended):25 cm
4. Rated Speed(Adjustable): 0-0.3 m/s
5. Joint Speed(Adjustable): 0-60 rpm
6. Shoulder Base Spin:180 degree
7. Shoulder Pitch:180 degree
8. Elbow Pitch:180 degree
9. Wrist Pitch:180 degree
10. Wrist Spin:180 degree
11. Gripper Opening(Max):8 cm

Salient Features

- The Arm has four servos which are controlled through potentiometer.
- The arm could grab things approximately in a hemisphere of 50 cm and is robust made completely with an aluminum sheet of 2.5mm.
- This arm is very user friendly.
- It can lift objects up to weight of 100 gm.
- The base is equipped with the high torque servo motor.
- Keeping the design of robotic arm gripper simple, as well as implementing the gripping mechanism without using gears and with one servo motor

Arduino

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces,including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. Arduino R3 Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack,an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.



- **Power USB**

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection

- **Power (Barrel Jack)**

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

- **Voltage Regulator**

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

- **Crystal Oscillator**

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

- **Arduino Reset**

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled

RESET.

- **Pins (3.3, 5, GND, Vin)**

3.3V (6) − Supply 3.3 output volt

5V (7) − Supply 5 output volt

Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

- **Analog pins**

The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

- **Main microcontroller**

Each Arduino board has its own microcontroller (11). You can assume it as the brain of you board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

- **Power LED indicator**

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

- **Digital I/O**

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled can be used to generate PWM.

Servo Motor

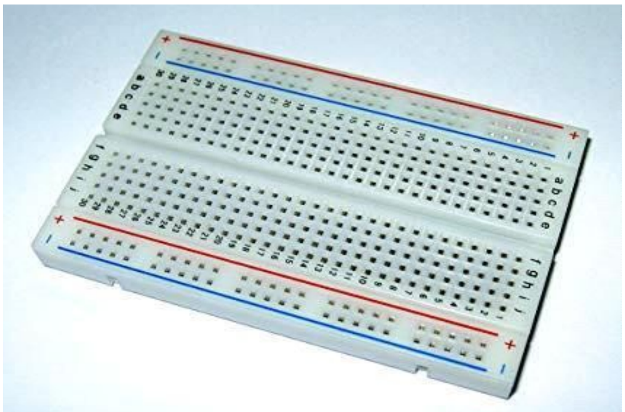
A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. If the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, the angular position of the shaft changes. In practice, servos are used in radio-controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio-controlled cars, puppets, and of course, robots.

Servo Motor

Servos are extremely useful in robotics. The motors are small, have built-in control circuitry, and are extremely powerful for their size. A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is strong for its size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, does not consume much energy. The guts of a servo motor is shown in the following picture. You can see the control circuitry, the motor, a set of gears, and the case. You can also see the 3 wires that connect to the outside world. One is for power (+5volts), ground, and the white wire is the control wire.

Breadboard

A breadboard is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also extremely popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Vero board) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).



Breadboard

Potentiometer

A potentiometer is a three terminal resistor with a sliding or rotating contact that forms an adjustable voltage driver. If only two terminals are used, one end and the wiper, it act as variable resistor or rheostat.

The measuring instrument called a potentiometer is essentially a voltage driver used for measuring electrical potential (voltage); the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducer for example in joystick. Potentiometer are rarely used to directly control significant power, since the power dissipated in the potentiometer would be comparable to the power in controlled load.

Power

The Power supply selected for feeding the control circuits of the servo motors is capable of delivering the same current even if all the synchronous servo motors are operating. When all servo motors are operated at same time, they draw 0.5A current. In addition 5V was needed for robotic movement in project. This requirement is powered by 5V power supply.

TESTING TECHNIQUES

Black Box Testing

Black box testing is mainly used to execute the requirements of the system. Black-Box testing examines functionality without knowledge of internal implementation. It mainly concentrates on whether the input is accepted, and output is generated or not. It concentrates on functional requirements, so it is also called as functional testing. Functional testing evaluates the correctness of the program without any knowledge of how the software is implemented. In black box testing, testers test software through user interfaces, data structures, data base, or the application programming interfaces at the later stages of software development.

White Box Testing

White box testing known as clear box testing, glass box testing, transparent box testing and structural testing. In White box testing an internal perspective of the system, as well as programming skills are used to design test cases. White box testing is detailed examination of the code. Code coverage criteria is defined using segment coverage, branch coverage, node testing, statement coverage, condition coverage, basis path testing, data flow testing, path testing and loop testing. White box testing can be applied at the unit integration and system levels of software testing process. The test procedure attempts to execute every part of the source code using the test data.

Black Box Testing	White Box Testing
the main focus of black box testing is on the validation of your functional requirements.	White Box Testing (Unit Testing) validates internal structure and working of your software code
Black box testing gives abstraction from code and focuses testing effort on the software system behaviour.	To conduct White Box Testing, knowledge of underlying programming language is essential. Current day software systems use a variety of programming languages and technologies and its not possible to know all of them.
Black box testing facilitates testing communication amongst modules	White box testing does not facilitate testing communication amongst modules

REFERENCES:-

- S.S. Navghane, M.S. Killedar, Dr.V.M. Rohokale, || IoT Based robotic arm May 2016.
- Ghose, M.K., Dikshit, A.K., Sharma, S.K. A GIS based transportation model for robotic arm.
- Guerrero, L.A., Maas, G., Hogland, W.: Robotic Arm for cities in developing countries.
- Kasliwal Manasi H., Suryawanshi Smitkumar B, A Novel Approach to Robotic Arm Using Internet of Things.
- Monika K A, Nikitha Rao, Prapulla S B, Shobha G, Robotic Arm using Arduino.

Future Work

1. We can interface sensor to this robot so that it can monitor some parameters.
2. We can add wireless camera to this robot.

Conclusion

An autonomous robot with adjustable gripper that perform pick and place operation has been successfully designed and developed. The robot has been able to pick the object and place it effectively. The robot is also able to perform lifting upward and downward smoothly. By using AVR microcontroller, the robot has performed the task perfectly according to the program that being made. Beside than that, the adjustable gripper with sensors is able to open its grip according to the size of the object. Due to this advantage, the robot can pick. This system can be used in various applications like in gripper, fabrication process, and inspection, processing, spraying, stamping and welding for work piece

Appendices:-

- Coding /Code Template

Source code:

Servo Code to perform robotic arm movement:

```
#include <Servo.h>
#include <EEPROM.h>
bytepackage_order=0; package
byteIncomingdata[200];
byteMMC_packageOrder=0;
byteData_size=0;
byteRead_Data_size=0;
intdata_size=0;
boolsec_btn_up = 0;
boolsec_btn_down = 0;
boolup_btn =0;
booldown_btn=0;
boolsave_btn =0;
boolplay_btn  =0;
String hmisend;
byteSecili_Servo=1;
bytehiz=75;
Servo Alt_servo;
Servo Alt_shoulder;
Servo Ust_elbow;
Servo G_wrist;
Servo Gripper;
Page | 27
intAlt_Servo_Degree=90;
intAlt_shoulder_Degree=90;
intUst_elbow_Degree=90;
intG_wrist_Degree=90;
```

```
intGripper_Degree=90;
charbt_data;
void setup() {
Serial.begin(9600);
pinMode(2,OUTPUT);
digitalWrite(2,1);
pinMode(13,OUTPUT);
digitalWrite(13,0);
Alt_servo.attach(5);//3
Alt_shoulder.attach(6);  //6
Ust_elbow.attach(9);//11
G_wrist.attach(10);//5
Gripper.attach(11);//9
Alt_servo.write(90);
Alt_shoulder.write(90);
Ust_elbow.write(90);
G_wrist.write(90);
Gripper.write(90);
}
voidDegree_updown()
{
delay(hiz);
if(bt_data=='B')
```



```
  {
//    Serial.print("Alt_servo: ");Serial.println(Alt_Servo_Degree);
Alt_Servo_Degree++;
if(Alt_Servo_Degree> 180) Alt_Servo_Degree=180;
Alt_servo.write(Alt_Servo_Degree);
  }
else if(bt_data=='D')
  {
//     Serial.print("Alt_shoulder: ");Serial.println(Alt_shoulder_Degree);
Alt_shoulder_Degree++;
if(Alt_shoulder_Degree> 180) Alt_shoulder_Degree=180;
Alt_shoulder.write(Alt_shoulder_Degree);
  }
else if(bt_data=='F')
  {
//     Serial.print("Ust_elbow: ");Serial.println(Ust_elbow_Degree);
Ust_elbow_Degree++;
if(Ust_elbow_Degree> 180) Ust_elbow_Degree=180;
Ust_elbow.write(Ust_elbow_Degree);
  }
else if(bt_data=='H')
{
//     Serial.print("G_wrist: ");Serial.println(G_wrist_Degree);
G_wrist_Degree++;
```

```
if(G_wrist_Degree> 180) G_wrist_Degree=180;
G_wrist.write(G_wrist_Degree);
  }
else if(bt_data=='Y')
Page | 29
  {
//     Serial.print("Gripper: ");Serial.println(Gripper_Degree);
Gripper_Degree++;
if(Gripper_Degree> 180) Gripper_Degree=180;
Gripper.write(Gripper_Degree);
  }
if(bt_data=='A')
  {
//    Serial.print("Alt_servo: ");Serial.println(Alt_Servo_Degree);
Alt_Servo_Degree--;
if(Alt_Servo_Degree< 0) Alt_Servo_Degree=0;
Alt_servo.write(Alt_Servo_Degree);
  }
else if(bt_data=='C')
  {
//   Serial.print("Alt_shoulder: ");Serial.println(Alt_shoulder_Degree);
Alt_shoulder_Degree--;
if(Alt_shoulder_Degree< 0) Alt_shoulder_Degree=0;
Alt_shoulder.write(Alt_shoulder_Degree);
  }
else if(bt_data=='E')
  {
//     Serial.print("Ust_elbow: ");Serial.println(Ust_elbow_Degree);
Ust_elbow_Degree--;
if(Ust_elbow_Degree< 0) Ust_elbow_Degree=00;
Ust_elbow.write(Ust_elbow_Degree);
  }
else if(bt_data=='G')
  {
//     Serial.print("G_wrist: ");Serial.println(G_wrist_Degree);
Page | 30
G_wrist_Degree--;
if(G_wrist_Degree< 0) G_wrist_Degree=00;
G_wrist.write(G_wrist_Degree);
  }
else if(bt_data=='X')
  {
//     Serial.print("Gripper: ");Serial.println(Gripper_Degree);
Gripper_Degree--;
if(Gripper_Degree< 0) Gripper_Degree=00;
Gripper.write(Gripper_Degree);
  }
}
```

```
intSaved_Location=0;
intstep=1;
voidSave()
{
if(bt_data=='K')
  {
EEPROM.write(step, Alt_Servo_Degree);
step++;
EEPROM.write(step, Alt_shoulder_Degree);
step++;
EEPROM.write(step, Ust_elbow_Degree);
step++;
EEPROM.write(step, G_wrist_Degree);
step++;
EEPROM.write(step, Gripper_Degree);
step++;
Saved_Location++;
```

```
 // Serial.print("LocationKaydedildi.!
ToplanLocationDizini:");Serial.println(Saved_Location);
}
EEPROM.write(0,Saved_Location);
}
bool play=0;
intplaystep=0;
voidplay()
{
if(bt_data=='P')
 {
  //Serial.println("Saved Is yapiliyor...");
play=1-play;
 }
if(play==1)
   {
playstep=1;
Saved_Location= EEPROM.read(0);
for(inti=0;i<Saved_Location;i++)
     {
intDegree = EEPROM.read(playstep);
while(Alt_Servo_Degree!=Degree)
       {
if(Alt_Servo_Degree>Degree) Alt_Servo_Degree--;
else if(Alt_Servo_Degree<Degree) Alt_Servo_Degree++;
delay(10);
Alt_servo.write(Alt_Servo_Degree);
       }
```

```
playstep++;
```

```
delay(100);
Degree = EEPROM.read(playstep);
while(Alt_shoulder_Degree!=Degree)
      {
if(Alt_shoulder_Degree>Degree) Alt_shoulder_Degree--;
else if(Alt_shoulder_Degree<Degree) Alt_shoulder_Degree++;
delay(10);
Alt_shoulder.write(Alt_shoulder_Degree);
      }
playstep++;
delay(100);
Degree = EEPROM.read(playstep);
while(Ust_elbow_Degree!=Degree)
      {
if(Ust_elbow_Degree>Degree) Ust_elbow_Degree--;
else if(Ust_elbow_Degree<Degree) Ust_elbow_Degree++;
delay(10);
Ust_elbow.write(Ust_elbow_Degree);
      }
playstep++;
delay(100);
Degree = EEPROM.read(playstep);
while(G_wrist_Degree!=Degree)
      {
if(G_wrist_Degree>Degree) G_wrist_Degree--;
else if(G_wrist_Degree<Degree) G_wrist_Degree++;
delay(10);
G_wrist.write(G_wrist_Degree);
```

```
      }
playstep++;
delay(100);
Degree = EEPROM.read(playstep);
while(Gripper_Degree!=Degree)
      {
if(Gripper_Degree>Degree) Gripper_Degree--;
else if(Gripper_Degree<Degree) Gripper_Degree++;
delay(10);
Gripper.write(Gripper_Degree);
      }
playstep++;
    }
  }
}
void loop()
{
if(Serial.available()>0)
 {
```

```
bt_data=Serial.read();
 }
Degree_updown();
Save();
play();
if(bt_data=='K' || bt_data=='P')
  {
digitalWrite(13,1);
```

```
bt_data=' ';
  }
else  digitalWrite(13,0);
```