Robotic Arm Using Arduino

Submitted

By

Mehul Raj – 18BCE1146

Parth Choudhary – 18BCE1039

ABSTRACT

A 4 Degree of Freedom (DOF) robotic arm has been developed. It is controlled by an Arduino Uno microcontroller which accepts input signals from a user by means of a set of potentiometers. The optimized part used the electrical signal input via Numpad. The arm is made up of four rotary joints and an end effector, where rotary motion is provided by a servomotor. The Arduino has been programmed to provide rotation to each servo motor corresponding to the amount of rotation of the potentiometer shaft. For a different mechanism, hand controlled by a numpad has also been implemented with some changes in architecture.

INTRODUCTION

A robot is a machine made by humans that can not only execute the tasks which we are capable of doing but also those tasks which are not in our hands and that too with high accuracy repeatedly. That is the reason that one of the most common applications of robots is collecting information and experiments on hazardous sites which are difficult and dangerous for humans. It is a mechanical arm, a manipulator designed to perform many different tasks. To perform its assigned tasks, the robot moves parts, objects, tools, and special devices by means of programmed motions and points. The robotic arm performs motions in space. Its function is to transfer objects or tools from point to point, as instructed by the controller. Robots have reduced human interference by nearly 50 percent and have become an integral part of almost all industries. They are employed for different tasks including welding, trimming, picking and placing etc. These robots can be controlled in different ways like keypads, voice control etc. Our project portrays the control of a 4-Degree of Freedom robotic arm using Arduino Uno, potentiometers and Numpad, which are more precise and reduces time delay and is comparatively better than other programming languages in the field of monitoring and control. A robot can be defined according to the nature of the relative movements between the links that constitute it. There are five important types of Robot Arm Configurations. They are as follows:

- Cartesian (3P)
- Cylindrical (R2P)
- Spherical (Polar) (2 RP)
- Articulated (3R)
- SCARA (2R in horizontal + 1P in vertical plane)

Where P- Prismatic joint, R – Rotary joint. The robotic arm presented in this paper uses only rotary joints. Rotary joints allow a full range of motion, as they rotate through multiple planes and they increase the capabilities of the robot considerably. With multiple rotary joints, a robot can engage in very exact movements and utilize their flexibility to manipulate a greater work volume. The robotic arm is constituted of five revolute joints where its mobility is five. The actuator used is a servomotor.

Being new to this field, our aim is to learn the basic concepts of Robotics and Arduino so we have tried to show a simulation of Robotic Arm using Arduino with 5 servo-motors as well as explored using a single servo-motor.

EXISTING SYSTEMS

Various projects have been made by different people for developing this project. However, they serve a different application and have different technologies implemented. Some of those papers are mentioned below stating their technology and application.

Ranjith Kumar Goud and B. Santosh Kumar [1] have invented a pick and drop robot. They wanted it to be used for diffusing a bomb remotely with safety. For the robotic arm, they used a pair of motors and another pair as the wheels of the robot for controlling the movement. Connectivity is established using Bluetooth. The microcontroller used is LPC2148. They had also attached wireless cameras for remote surveillance. They have worked on this project mainly for industrial and military applications.

M. Selvam [2] in his paper has designed to develop a robotic system which has a wireless camera attached to the surveillance. Bluetooth was implemented in his project for providing connection between robot and smartphone. Wireless night vision camera was used for providing the robot surveillance. The video which is recorded by camera is then transmitted to the TV unit through radio-frequency signal. He used 8051 microcontrollers for the robotic unit.

Jorge Kazacos Winter [3] has developed android controlled robot automation. Main aim of his project was the transfer of information wirelessly between a Smartphone and the robot and developing the robot and its communication system underneath a low price and open-source philosophy. He used 3D design techniques to style the structure of the robot with the facilitation of parametric modeling software. The style, when fed to the 3D printer can print the parts of the robot in a layered manner one by one and can then use these parts to assemble the robot simply. He has used an Arduino microcontroller and Wi-Fi technology in this robot.

PROBLEM STATEMENT

Robotic Arm using Arduino has been a very interesting and common topic for us as students. In this pandemic period, the hardware components required to build a robotic arm are not available. To learn the concepts of Robotics and Arduino, a Robotic Arm simulation has to be developed using the components like Arduino, Servo Motors, Potentiometer, Numpad, etc.

INTRODUCTION TO PROPOSED MODEL

Using the basic concepts of Robotics, we have implemented a simulation of a Robotic Arm controlled by Arduino programming. A different approach for reducing the number of servo-motors has also been explored and implemented.

PROPOSED SOLUTION

The basic components remain the same for the proposed solution. But we have developed a system in which user will be able to move the servo motors by the angle as less as 22.5 degree.

We have implemented this using simple Numpad where all the digits have been utilized that helps to shift the angle by 22.5 degree, this will help user to rotate all the four fingers simultaneously with précised angle based on the type of material needs to picked and placed.

The number of servo-motors is 4 and to give them signal to rotate has been reduced from 4 to 1 and that via Numpad.

Sample model diagram is given below:

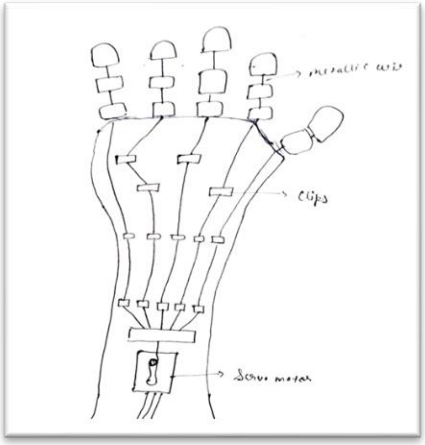

Figure 1: Sample Working Model

COMPONENTS OF THE MODEL

Arduino Uno: The Arduino UNO is the best board to get started with electronics and coding. The UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.

Potentiometer: This changes the resistance by rotating the knob. This will help to change the resistance that will help us to rotate the servo motor.

Servo Motor: Simple to DC motor, here we can define the rotational speed.

For our current development of the project, we need to 4 server motors.

Numpad: Simple Numpad is being used, which is connected with Arduino Uno and is used to give the electrical signal input for the all the 4 servo motors to be rotated simultaneously.

Bread Board & Connecting Wires: These are used to bring all the parts of the project together.

BLOCK DIAGRAM

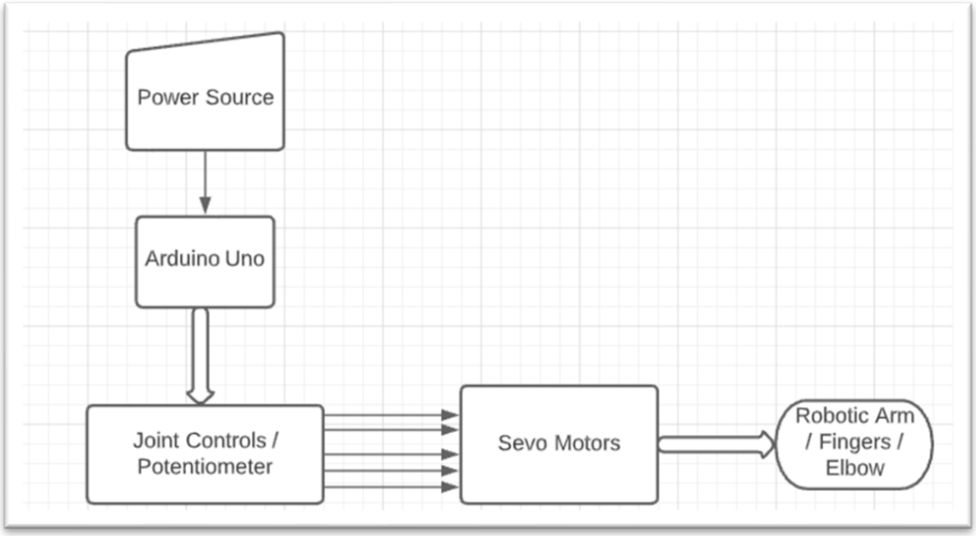

Figure 2: Block Diagram

The Arduino Uno is powered by an external power source. Its pins are then mapped/connected to a potentiometer which is responsible for the movement of mechanical fingers of the robotic arm, by changing its resistance, servo-motor can be controlled as it is connected with the servo motor which acts as actuator and results in the movement of the fingers of the arm.

After this, we have used the Numpad to give the inputs in place of potentiometer. This helps to reduce the 4 potentiometers that we were earlier using for analog input signal for the servo motors and help them to rotate them together. We have given the base degree as 22.5 and thus user can rotate the 4 servo motors simultaneously using the numpad digits and that too, very precisely and rotate based on the size of the object it needs to pick and place.

DESCRIPTION OF THE COMPONENTS (HW / SW / TECH. SPECIFICATIONS)

Technical Specifications:

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts

- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- UART: 1
- I2C: 1
- SPI: 1
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

WORKING MECHANISM

In this, we are developing an Articulated Arm that will help users to pick and place different objects of different sizes. We will be moving our Articulated arm on the basis of: End effector movements. Since our main objective is to implement pick and place therefore, we have optimized our work so that our user will be able to rotate all the fingers simultaneously with simple input device and will be able to do the operations. Elbow moment for the Arm to have more helping hand for the user.

Pin Configuration:

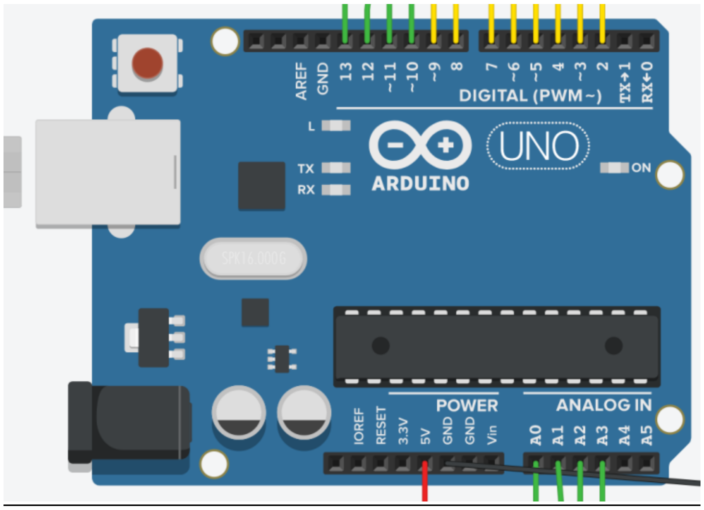

Figure3: Pin Configuration Arduino Uno

- We have given 5V as input for the Arduino-Uno Model.
- We have made sure to use grounding unit of Arduino Uno to keep the circuit intact.
- A0-A3: Analog pins that we have used first to rotate the servo motors using potentiometer. These pins help to get the analog input from the potentiometer.
- L3-(~10): These are used to give the electrical signals to the servo motors to rotate.
- (~9)-2: Numpad Connection and this helps to use the digits of the numpad to give the electrical input to the servo motors in place of potentiometer.

Working demo: https://youtu.be/oXG4NbL2OFM

CODE (INPUT / EXECUTION / OUTPUT)

```
#include <Keypad.h>

#include <Servo.h>

// Servo configuration

Servo servo0;

Servo servo1;
Servo servo2;
Servo servo3;
int sensorvalue0;
int sensorvalue1;
int sensorvalue2;
int sensorvalue3;

// Keypad configuration
const byte numRows= 4; //number of rows on the keypad
const byte numCols= 4; //number of columns on the keypad

//keymap defines the key pressed according to the row and columns just as appears on the keypad
char keymap[numRows][numCols]= {
 {'1', '2', '3', 'A'},
 {'4', '5', '6', 'B'},
 {'7', '8', '9', 'C'},
 {'*', '0', '#', 'D'}
};

//Code that shows the the keypad connections to the arduino terminals
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);

void setup()
{
  Serial.begin(9600);
  pinMode(A0,INPUT);
  pinMode(3,OUTPUT);
  servo0.attach(13);

  pinMode(A1,INPUT);
  pinMode(5,OUTPUT);
  servo1.attach(12);

  pinMode(A2,INPUT);
  pinMode(6,OUTPUT);
  servo2.attach(11);

  pinMode(A3,INPUT);
  pinMode(9,OUTPUT);
  servo3.attach(10);
}


void setAngle(int angle) {
		sensorvalue0 = angle;
    sensorvalue1 = angle;
    sensorvalue2 = angle;
    sensorvalue3 = angle;
    servo0.write(sensorvalue0);
    servo1.write(sensorvalue1);
    servo2.write(sensorvalue2);
    servo3.write(sensorvalue3);
}


void loop()
{
  // 1,2,3,6,9,8,7,4 -> 22.5 degree
  // 0 -> 180 degree and 5 to reset -> 0 degree


  //If key is pressed, this key is stored in 'keypressed' variable
  //If key is not equal to 'NO_KEY', then this key is printed out

  //if count=17, then count is reset back to 0 (this means no key is pressed during the whole keypad scan process



  char keypressed = myKeypad.getKey();

  if (keypressed != NO_KEY){

      Serial.println(keypressed);

     if(keypressed== '1' || keypressed == '5'){

       setAngle(180);

     }else if(keypressed == '0') {

       setAngle(0);

     }

    else if(keypressed=='2'){

      setAngle(22.5);

     }else if(keypressed=='3'){

      setAngle(45);

     }else if(keypressed == '6') {

      setAngle(67.5);

             }else if(keypressed == '9') {

      setAngle(90);

             }else if(keypressed == '8') {
```

```
    setAngle(112.5);

        }else if(keypressed == '7') {

    setAngle(135);

        }else if(keypressed == '4') {

    setAngle(157.5);

        }

}

}
```

OUTPUT SCREENSHOTS

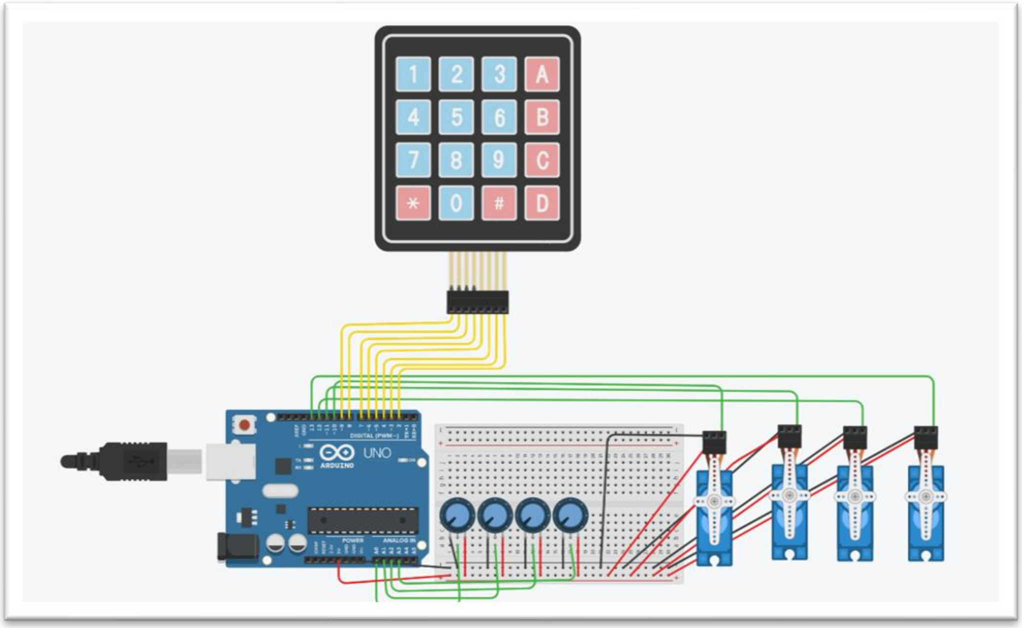

Figure4: Main Model

CONCLUSION

A pick & place Robotic arm simulation has been done using Arduino programming. The driver mechanism comprising of an Arduino microcontroller in conjunction with a numpad has been successfully used to control the arm as per the inputs given by the user. We have used the Numpad instead of potentiometer for input with base degree of 22.5. 4 servo motors can be rotated simultaneously using the numpad digits with precision and rotation can be done based on the size and shape of the object for pick and place operation. The robot so programmed for pick and place operation can be made versatile and more efficient by providing the feedback and making it to work on own than any human interventions.

REFERENCES

1. Goud, R.K. and Kumar, B.S., 2014. Android based robot implementation for pick and retain of objects. *Int. J. Eng. Trends Technol.(IJETT)*, *16*(3), pp.0974-2174.
2. Selvam, M., 2014. Smart phone based robotic control for surveillance applications. *International Journal of research in Engineering and Technology*, *3*(3), pp.229-232.
3. Kazacos Winter, J., 2013. Android controlled mobile robot| (Master's thesis).

APPENDIX

- Microcontroller: Microchip Atmega328P:
  - ATmega328 is a single chip microcontroller in the megaAVR family.
  - Specifications:
    - ➔ 8-bit AVR RISC-based microcontroller combines 32 KB ISP flash memory with read-write capabilities, 1 KB EEPROM, 2 KB SRAM, 23 general purpose i/o lines, 32 general purpose working registers.
    - ➔ Three flexible timer/ counters with compare modes, internal and external interrupts.
- SRAM:
  - Static Random-Access Memory
  - Volatile memory, data is lost when power is removed.
- EEPROM:
  - Electrically erasable programmable read only memory.

- Non-volatile memory, integrated in microcontrollers for smart cards and remote keyless systems, and other devices to store relatively small amounts of data by allowing individual bytes to be erased and programmed.

- Bootloader:
  - Software that is responsible for booting a system.
  - When the system is turned off, its software – including operating systems, application code and data remains stored on non-volatile memory.
- UART
  - A hardware communication protocol that uses asynchronous serial communication with configurable speed.
- Flash Memory
  - Also known as Flash storage
  - Non-volatile memory that erases data in units called blocks and rewrites data at the byte level.
  - Widely used for storage and data transfer in consumer services, enterprise systems and industrial applications.
- SPI
  - Serial Peripheral Interface.
  - A synchronous serial data protocol used by microcontrollers for communicating with one or more peripherals devices quickly over short distances.
  - With SPI, there is always one master device which controls the peripheral devices.
- I2C
  - Serial communication protocol, so data is transferred bit by bit along single wire.
  - Synchronous, so the output of bits is synchronized to the sampling of bits by clock signal shared between master and slave.