©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.

A SUSTAINABLE APPROACH TO A CARTESIAN ROBOTIC ARM

R Swarnalatha[1], Swetha Perial[2], Samar Kulsoom[3] and Saaima Anwar[4]
[1]Department of EEE Birla Institute of Technology and Science, Pilani Dubai Campus, Academic City, Dubai, United Arab Emirates
[2]Department of Electronics and Communication Engineering, Birla Institute of Technology and Science, Pilani Dubai Campus, Academic City, Dubai, United Arab Emirates
[3]Department of Computer Science Engineering, Birla Institute of Technology and Science, Pilani Dubai Campus, Academic City, Dubai, United Arab Emirates
[4]Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani Dubai Campus, Academic City, Dubai, United Arab Emirates
E-Mail: swarnalatha@dubai.bits-pilani.ac.in

ABSTRACT

This research deals with the construction and designing of a sustainable robotic arm with five degrees of freedom and memory. The entire setup was simulated using the Arduino UNO microcontroller and Arduino IDE software. The arm designed can perform basic useful functions including but not limited to, picking up and placing objects in desired locations. The arm has great flexibility and is highly adjustable owing to the five degrees of freedom it possesses. This investigation deals with the designing and implementation process of a fully functioning robotic arm. The robotic arm is created to move around with five degrees of freedom across a plane in order to increase the range of activities it can perform. The movements of the arm can be recorded and replayed making it useful to carry out redundant functions present in the daily routine of man. The arm is made up of environment friendly materials so as to attain sustainability and not further lead to environmental degradation. The robotic arm and its features are explained in detail along with the pertinent need for sustainability. The construction and circuitry involving the internal and external connections concerned are discussed along with the program code executed on Arduino.

Keywords: arduino UNO, servo motors, sustainability, robotic arm, memory, degrees of freedom.

1. INTRODUCTION

A robotic arm can be defined as a part of an entire robotic system but as an entity on its own. It can be programmed to carry out or execute a multitude of tasks of varying complexities. It can be used to perform tasks which are found to be too redundant or time consuming from the human perspective. Thus, the arm was used as an assistive measure to aid working professionals in many fields that eventually led to robotic arms replacing full-fledged employees which served as the turning point in the field of automation. Currently, robotic arms are deployed in numerous demanding fields by profession. From drilling, welding, thermal spraying to handling items, packaging and even painting, the robotic arm is used for various applications on the industrial front by storing the exact sequence of movements in its memory and performing it again and again every time a new component comes down the assembly line. At present in the marketplace, many products are being assembled or managed by robots [1].

This research includes an industrial robotic pick-and-place arm that could perform a variety of assembly, material handling and transition, joining, inspection, and palletising tasks [2]. This research involves a Cartesian robot, a type of robot used for pick and place work, the handling of machine tools and the application of arc welding for various purposes, such as assembly operations [3].

1.1 Features

A robotic arm essentially entails a single board microcontroller which is easily available and accessible. These microcontrollers are very open ended in terms of extension and customization.

A robotic arm can be categorized based on an array of features which include the number of axes, the degrees of freedom, the source of power, drive i.e., how the motors of the arm are attached to the joints and working freedom i.e., the spatial region that can be reached by the arm along with the carrying capacity/ pay load of the arm. This revolutionized technology essentially provides a humanized capability in a multitude of fields along with speed, reliability, acceleration, repeatability and accuracy of the highest standard [3].

1.2 Sustainability

Sustainability refers to the ability to answer the needs of the environmental needs and demands at present without compromising or affecting the ability of future generations to meet such needs when encountered [4].

Robotic arms and their deployment have played a key role in assuring the requirements to attain sustainability are met at large. They are helping manufacturers and large corporations on the sustainable front by assuring reduction in waste and in the amount of energy expended which in turn leads to the preservation of current resources. The robotic arm puts forward many advantages as it can be quite cost effective and friendly to the environment thus proving to be sustainable in the long run. It can be used in profusion in physical environments that can be either dangerous or not accessible [3].

Moreover, the presence of robotic arms can reduce work related injuries in the manufacturing sector which can be brought upon by stress or other dangerous hazards to a human being. The ability of robotic arms to

©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.

be programmed to perform a task without burnout guarantees increased efficiency and productivity in the work field and as a result, increased product quality. The research deals with a robotic arm being made out of environmentally friendly items such as cardboard, wool and wood for its construction.

1.3 Hardware Specification

- Two breadboards
- Jumper Wires
- 5 LEDS
- 5 10K ohm resistors and 2 221-ohm resistors
- Push Buttons
- Power Supply of 4.5 Ah, 6V
- 5 Potentiometers
- MG996R servo motors and 2 MG90S servo motors

To provide the necessary torque at each joint position, servo motors were used. The servo will precisely control the angular position of the output shaft [1]. There is a combination of gears and a potentiometer with circuitry within the servo, which allows its location to be set reasonably quickly and accurately and within a travel range of 180 degrees [5]. The 'MG' added to the name of the servo is an abbreviation for 'Metal Gear' which means that the drive gears of the servo are of metallic nature. By sending a series of pulses to it, a servo motor is operated. This signal is known as Pulse Width Modulation (PWM) signal. Every 20 milliseconds, a pulse is forwarded to the servo. The motor spline or shaft moves to a location within a 180-degree range depending on the pulse width [1]. The motor shifts to a position correspondingly between 0 and 180 degrees by any pulse between 900µs and 2100µs [6].

- Arduino Uno microcontroller

The primary component of this research is the Arduino Uno I/O board, which interfaces with the Graphical User Interface motors. The Arduino Uno is an ATmega328 based microcontroller board [7]. It is composed of -

a) 14 digital input/output pins where 6 of these can be used as PWM outputs
b) 16 MHz crystal oscillator
c) USB connection
d) Power jack
e) ICSP header
f) Six analog inputs
g) Reset button

2. MATERIALS AND METHODS

2.1 Pin Connection of the Arduino with Circuit Components

- Potentiometer - A0, A1, A2, A3, A4 (Analog Input)
- Pushbuttons - Pin 12, Pin 13 (Digital PWM)
- LEDs - Pin 2, Pin 3, Pin 4, Pin 7, Pin 8 (Digital PWM)
- Servos - Pin 5, Pin 6, Pin 9, Pin 10, Pin 11(Digital PWM)

Figure-1 below shows the pin configuration of the Arduino Uno board.

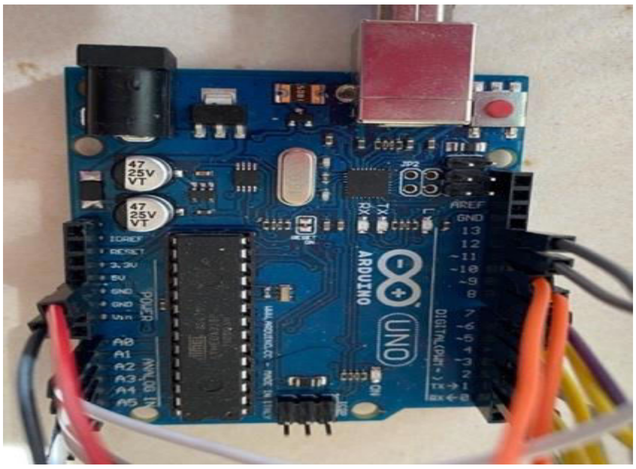

Figure-1. Arduino UNO Pins.

2.2 Circuitry

Figure-2 below shows the complete arrangement of the arm including the placement of the robotic arm along with circuitry.

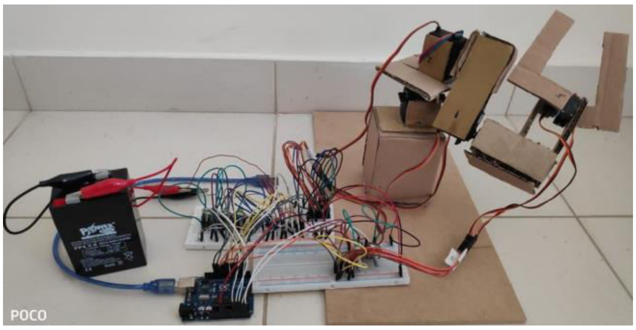

Figure-2. Arrangement of the Robotic Arm.

All the connections are done through the breadboard. The servo motors, potentiometers, pushbuttons and LEDs along with their resistors are all connected in parallel. The breadboard consists of one bus strip on either side. The positives and negatives are connected using these strips. The bus strip on the left side of the breadboard connects the Arduino UNO with the pushbuttons, LEDs, resistors and potentiometers while the bus strip on the right connects the servo motors and the power supply.

2.2.1 Right terminal strip of breadboard

VOL. 17, NO. 8, APRIL 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.
www.arpnjournals.com

- **Power supply**

The positive and negative end of the power supply is connected to the top of the positive and negative rails of the breadboard bus strip.

- **Servo motors**

The red wires of the 5 servo motors are connected to the positive rail and the brown wires of the 5 servo motors are connected to the negative rail. The orange wires of the 5 servo motors are connected to the digital PWM pins 5, 6, 9, 10 and 11 of the Arduino UNO.

2.2.2 Left terminal strip of breadboard

- **Arduino UNO**

The 5V pin of the Arduino and ground pin is connected to the top of the positive and negative rails of the left breadboard bus strip.

- **Potentiometer**

Each potentiometer has three pins - first pin is for positive; second pin is for angle output and third pin is for negative. The first pins of all the 5 potentiometers are connected to the positive rail and the third pins of all the 5 potentiometers are connected to the negative rail (grounded). The middle pins of all the 5 potentiometers are connected to the analog pins A0, A1, A2, A3 and A4 of the Arduino UNO.

- **Pushbuttons**

Each pushbutton has two pins on either side which are interconnected. The first pins of both the pushbuttons are connected to the positive rail. The second pins of both the pushbuttons are connected to pin 12 and 13 of Arduino UNO and are also connected to the negative rail (grounded) through 10k ohm resistors.

- **LEDs**

The anodes of the 5 LEDs are connected the digital pins 2, 3, 4, 7 and 8 of the Arduino UNO and are also connected to the positive rail through the 221-ohm resistors. The cathodes of the 5 LEDs are connected to the negative rail (grounded).

2.3 Arm Construction

The robotic arm consists of five degrees of freedom (DOF) and four links. The placement of motors has been done keeping the major joints of a human arm in consideration. The three MG996R motors replicate a human shoulder, elbow and wrist respectively. The two SG90 motors replicate human fingers which are used to pinch and lift objects.

For the MG996R motors, the star shaped fitting is used which has 6 blades since this fitting provides more support. For the SG90 motors, the twin blade fitting is used. The blades of Servo 3, 4 and 5 are placed horizontal to the plane and the blades of servo 1 and 2 are placed vertical to the plane. For constructing the arm, only sustainable environment friendly materials like wood, cardboard and wool are used.

Figure-3 below shows the entire set up involved in constructing the robotic arm where the first and second servo motors are placed horizontal to the plane and the third, fourth and fifth servo motors are placed vertical to the plane.

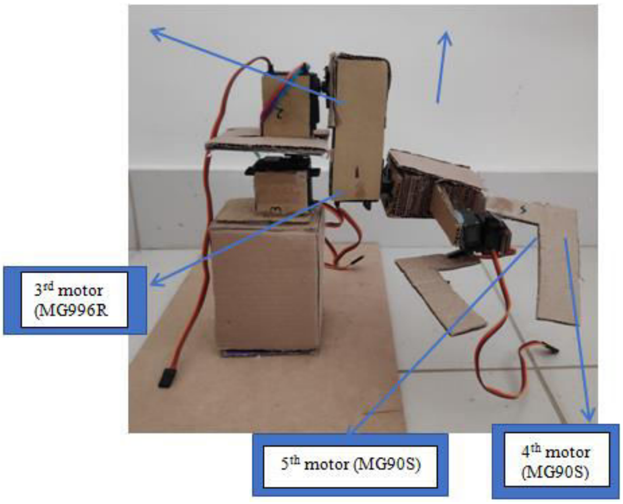

Figure-3. Construction of robotic arm.

2.4 Working Principle

The working of the robotic arm can be logically divided into two phases: storing the positions and replaying the positions.

2.4.1 Storing the positions

The first phase of the circuit which contributes to storing the positions consists of Potentiometers, LEDs, servos motors and the first pushbutton. The arm has been programmed to hold five set of positions. Once the program is compiled and uploaded into the Arduino UNO board, the positions can then be stored one at a time. The potentiometer input angles given, go as analog inputs into the Arduino board. The first pushbutton is connected to pin 12 which performs the function of Master in Slave Output (MISO). The Arduino UNO consists of an in-built Analog to Digital Converter (ADC) [8]. So, once the potentiometer knobs are rotated to the desired angles and press the first pushbutton, the Arduino takes in the analog input, converts and gives out a digital 8-bit Pulse Width Modulation (PWM) output through the digital pins where the LEDs are connected. Hence the LED glows indicating that a set of angles have been received and are being stored.

- The knobs of the 5 potentiometers are adjusted to the desired first position of the arm. Once the position is achieved, the first pushbutton is pressed and then the first LED will glow and, in the output, serial monitor it will display “Position 1 saved” indicating that the first set of positions have been saved. Figure-4 below shows the first example position set along with the LED output indicating that it has been saved.

VOL. 17, NO. 8, APRIL 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.
www.arpnjournals.com

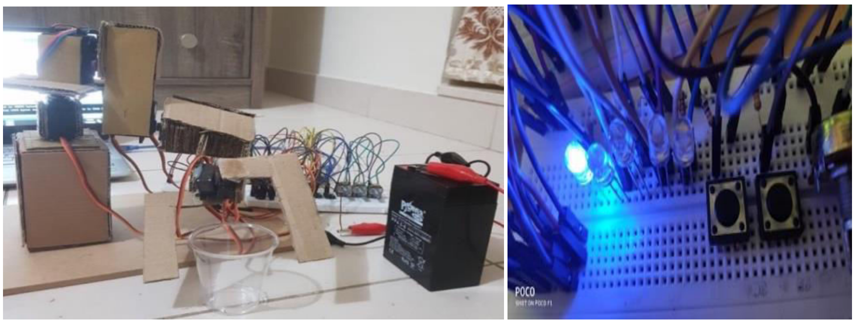

Figure-4. Position 1 and LED output.

- Similarly, the knobs of the 5 potentiometers are adjusted to the desired second position of the arm. Once the position is achieved, the first pushbutton is pressed and then the second LED will glow and, in the output, serial monitor it will display "Position 2 saved" indicating that the second set of positions have been saved. Figure-5 below shows the second example position set along with the LED output indicating that it has been saved.

Figure-5. Position 2 and LED output.

- Similarly, the knobs of the 5 potentiometers are adjusted to the desired third position of the arm. Once the position is achieved, the first pushbutton is pressed and then the third LED will glow and, in the output, serial monitor it will display "Position 3 saved" indicating that the third set of positions have been saved. Figure-6 below shows the third example position set along with the LED output indicating that it has been saved.

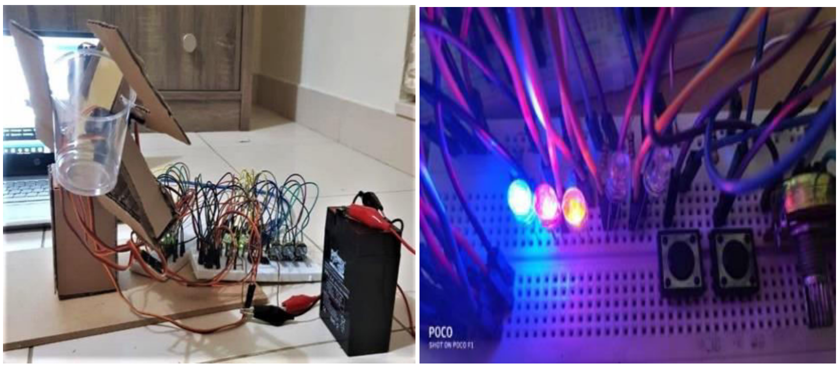

Figure-6. Position 3 and LED output.

- Similarly, the knobs of the 5 potentiometers are adjusted to the desired fourth position of the arm. Once the desired position is achieved, the first pushbutton is pressed and then the fourth LED will glow and, in the output, serial monitor it will display "Position 4 saved" indicating that the fourth set of positions have been saved. Figure-7 below shows the fourth example position set along with the LED output indicating that it has been saved.

ARPN Journal of Engineering and Applied Sciences
©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.

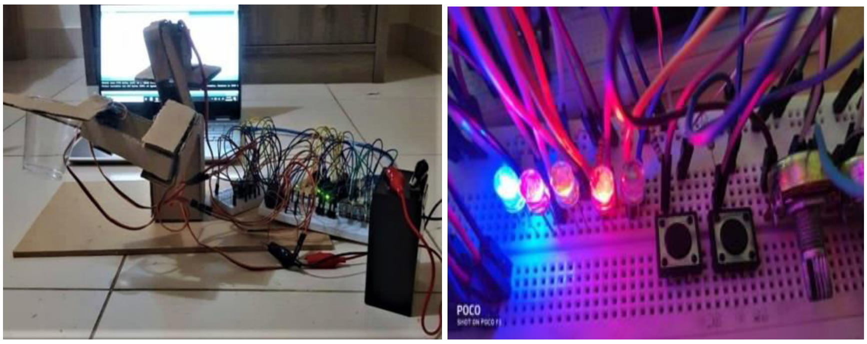

Figure-7. Position 4 and LED output.

- Similarly, the knobs of the 5 potentiometers have been adjusted to the desired fifth position of the arm. Once the position is achieved, the first pushbutton is pressed and then the fifth LED will glow and, in the output, serial monitor it will display “Position 5 saved” indicating that the fifth set of positions have been saved. Figure-8 below shows the fifth example position set along with the LED output indicating that it has been saved.

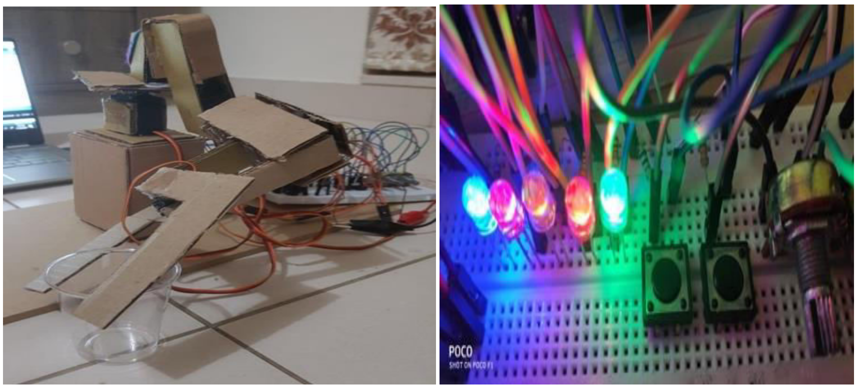

Figure-8. Position 5 and LED output.

2.4.2 Replaying the positions

Once the five sets of five servo motor positions of the arm are saved, then comes the second phase of the circuit. The second phase of the circuit which contributes to replaying the positions consists of the second pushbutton and the servo motors. Here, the second pushbutton which is connected to pin 13, is pressed, in order to replay the 5 positions saved in loop. Pin 13 is the serial clock pin. So once its pressed it makes sure that the saved sets of angles are performed by the servo motors in order for each clock pulse until the Arduino UNO is disconnected from the system.

2.5 Arduino IDE

The simulation of the robotic arm is done using Arduino IDE which is an open-source cross platform application used to write and upload programs to the Arduino board arranged [9].

The location of the motor shaft, which in turn regulates the movement of the joints, is controlled by the Arduino I/O board upon receiving the IDE commands [10].

The programs can be written in various languages including but not limited to, Java, C++ and C. This investigation uses an Arduino Uno, and the program is written using Java. The Arduino Servo library is used in order to control the servo motors and is accessed using the header file <Servo.h>.

2.5.1 Program identifiers and functions

The program makes use of many variables and functions to get the various components involved to work together in unison and give rise to a functioning robotic arm.

Program Identifiers

a) To identify the Servos
b) To identify the LEDs connected to pins 2, 3, 4, 7, 8 of the Arduino board
c) To identify the Push Buttons connected to pins 12 and 13 of the board
d) The binary values of the push buttons
e) To identify the values of Potentiometers connected to pins A0, A1, A2, A3 and A4 of the Arduino board
f) To identify the Potentiometer Values
g) To store the Potentiometer Angles
h) To store the Servo Position Array

ARPN Journal of Engineering and Applied Sciences
©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.

Program Functions

- void setup()

Servos are attached to a pin on the Arduino board using attach() which is used to set up everything and is run once as to initiate. The LEDs and push buttons are assigned as input or output in the Arduino board.

- void loop()

Potentiometer values are read and stored using analogRead() and mapped to the servo values using map(). Servos are then made to move to the mapped angle using write(). Using digital Read(), we determine if button1 was pressed to store the positions, storing the count. A switch case is then opened determined by value of button1 count leading to 5 cases. An if statement is then opened determined by the Boolean value of button2 being pressed to play the stored positions.

3. RESULTS AND DISCUSSIONS

3.1 Storing the Positions

Figure-9 below shows the code output on the serial monitor after the first position is saved.

Figure-9. Position 1- Code output.

Figure-10 below shows the code output on the serial monitor after the second position is saved.

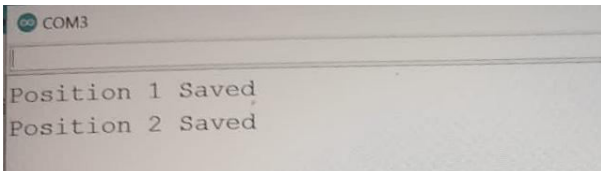

Figure-10. Position 2- Code output.

Figure-11 below shows the code output on the serial monitor after the third position is saved.

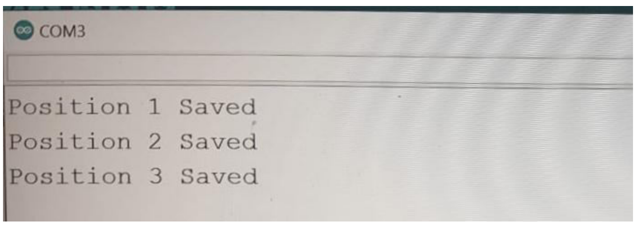

Figure-11. Position 3- Code output.

Figure-12 below shows the code output on the serial monitor after the fourth position is saved.

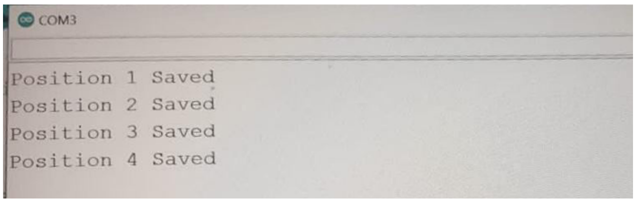

Figure-12. Position 4- Code output.

Figure-13 below shows the code output on the serial monitor after the fifth position is saved.

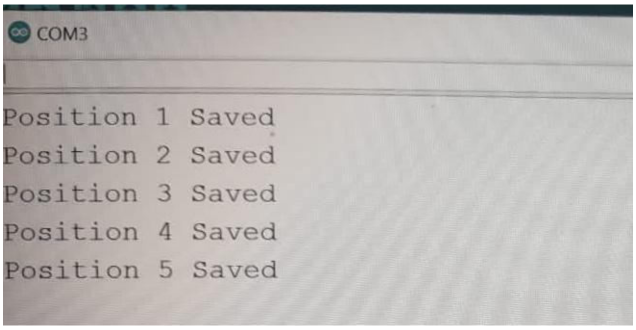

Figure-13. Position 5- Code output.

3.2 Replaying the Positions

Figure-14 below shows the first two sets of angles during the replaying phase in the serial monitor.

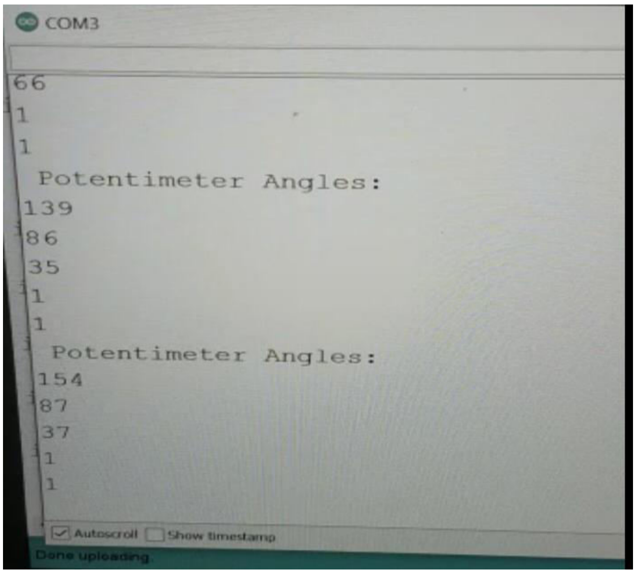

Figure-14. First and Second Set of Angles.

Figure-15 below shows the next two sets of angles during the replaying phase in the serial monitor.

ARPN Journal of Engineering and Applied Sciences
©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.

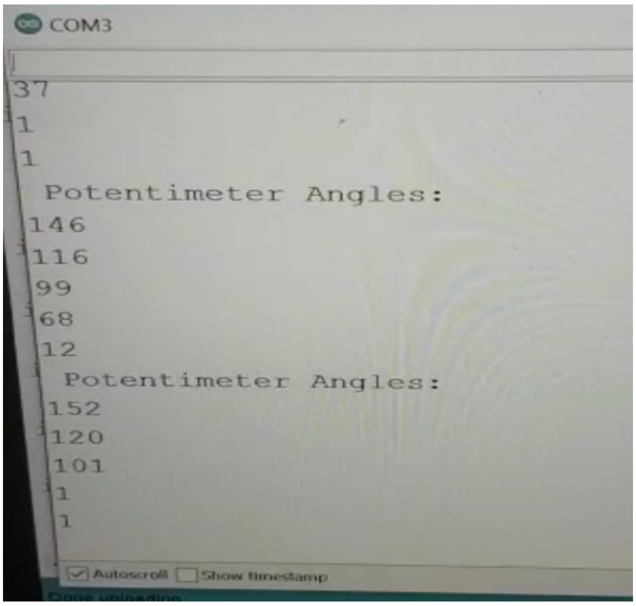

Figure-15. Third and fourth set of angles.

Figure-16 below shows the last two sets of angles during the replaying phase in the serial monitor.

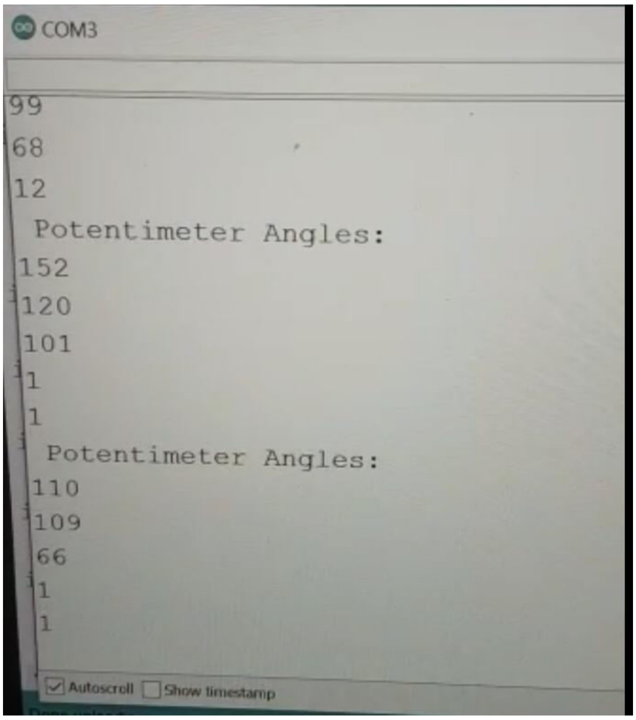

Figure-16. Fourth and fifth set of angles.

4. CONCLUSIONS

The revolutionized technology of the robotic arm has been a driving force in the field of automation. The human like capabilities is brought upon by making use of the servo motors to replicate human joints on the robotic arm and attempt to manoeuvre those using potentiometers. The entire configuration is simulated on an Arduino Uno microcontroller using the Arduino software. Each position of the arm movement can be recorded and then further played on a loop, making it an indispensable asset in manufacturing industries where repetitive and redundant tasks are present at large. The technological innovations in the field of assistive robots must factor in the crucial need of sustainability to ensure the longevity of this industry and in turn, the future of our environment.

4.1 Future Scope

The robotic arm can be improved further to meet application requirements in real time by the following future modifications.

- A 360-degree rotary servo motor can be introduced in order to make the setup more stable and in turn prevent singularities that occur in robotic arms i.e., when two or more links are lined up or when their geometric limits are crossed by two or more joints.
- In order to extend its workspace, the number of degrees of freedom of the robotic arm can be increased, thereby making it more flexible.
- It is possible to use motors with higher torque ratings to power the joints to ensure that the robotic arm stays in place even when the motors are not supplied with electrical current.
- Object detection and avoidance of collisions can be carried out by attaching proximity sensors [1].

ACKNOWLEDGEMENTS

The authors would like to thank Prof. R.N Saha, Director, BITS Pilani- Dubai for his constant encouragement and support.

REFERENCES

[1] Oluwajobi Akinjide & Oridate A.A. 2019. Design and Development of an Educational 5-DoF Robotic Arm. International Journal of Robotics and Automation Technology. 6. 55-65. 10.31875/2409-9694.2019.06.7.

[2] Angelo JA. 2007. Robotics: A Reference Guide to the New Technology. Westport: Greenwood Press.

[3] Priyambada Mishra, Riki Patel,Trushit Upadhyaya, Arpan Desai. 2017. Development of robotic arm using Arduino Uno.

[4] Author, Mitchell Grant. 2020. Sustainability. Retrieved November 5, from http://www.investopedia.com/terms/s/sustainability.asp

[5] Sullivan DO, Igoe T. 2004. Physical Computing: Sensing and Controlling the Physical World with Computers. Boston: Thomson Course Technology PTR.

©2006-2022 Asian Research Publishing Network (ARPN). All rights reserved.

[6] KUKA. History. 2002. Retrieved November 5, from http://www.kukarobotics.com/usa/en/company/

[7] Upadhyaya TK, Kosta S, Jyoti R, Palandoken M. Negative refractive index material-inspired 90-deg electrically tilted ultra-wideband resonator. Opt. Eng. 0001; 53(10): 107104. doi:10.1117/1.OE.53.10.107104.

[8] Author Nate. 2020. Analog to Digital Conversion. Retrieved November 5, from https://learn.sparkfun.com/tutorials/analog-to-digital-conversion/all

[9] What is An Arduino. 2020. Retrieved December 10, from https://opensource.com/resources/what-arduino

[10] Kruthika K, B M, Kiran and Lakshminarayanan Sanjay. 2016. Design and development of a robotic arm. 1-4. 10.1109/CIMCA.2016.8053274.