



Hardware Graduation Project

ARC

Author: Ali Moalla ,Baker Dweikat ,Hussien Jabali
Supervisor:Dr.Manar Qamhieh

May 21, 2022

Acknowledgement:

”We would like to express our thanks and gratitude to the many people who gave us moral support and contributed to the achievement and success of this work. and We appreciate and thank our teacher, Dr.Manar Qamhieh, for her support and advice and guidance during all stages of this work. Finally We thank our parents,families and friends for their role in supporting and making this work a success.”

DISCLAIMER:

”This report was written by students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned. ”

Contents

1 Abstract:	1
2 Introduction:	2
3 Constraints & Earlier Coursework:	3
3.1 Constraints Limitations:	3
3.2 Earlier Coursework:	3
4 Methodology:	3
4.1 Robotic Arm:	3
4.1.1 Component:	4
4.1.2 The 3D module:	6
4.1.3 Circuit:	8
4.1.4 Standards/Codes:	9
4.2 Hand Gestures Glove:	13
4.2.1 Components:	13
4.2.2 Hand Gestures Glove Circuit:	17
4.2.3 Hand Glove Code:	18
5 Conclusion	20
5.1 Summary	20
5.2 Future Work	21

1 Abstract:

A robotic arm that can be helping in carrying heavy objects or helping in reaching places that can't be reached or are dangerous to be reached by humans , this robotic arm can be controlled by mobile application or by using hand gestures.

the robotic arm can handle the received commands from both mobile application and the hand gesture glove using Arduino UNO , as for the hand gesture glove we use Arduino nano alongside flex sensors and gyroscope sensors to detect the movement of the hand and be able to send the commands to the robotic arm via Bluetooth.

2 Introduction:

The world is in a nonstop evolution progress especially in the industrial section , which made some of our inventions and buildings more complected such as some of the inventions become bigger or need a high precision while creating it, so it become difficult for people to handle and create these inventions thus so people made tools that can help them and make it easier to make such inventions an example on these tools are robotic arms.

These robotic arms can help carrying items and can make a very high precision actions (if the robotic arm was well made) that the humans can't achieve. This project will provide the users to be able to control the robot arm wirelessly via mobile application or by the Hand Gestures, these commands will be handled by the Arduino UNO for both the mobile application and the robotic arm movement, as for the Hand Gestures glove the commands will be generated by Arduino nano based on the hand movement and will be transmitted to the robotic arm via Bluetooth.

This Work will be very useful for people to carry heavy objects or reaching places that are inaccessible or dangerous for people.

3 Constraints & Earlier Coursework:

3.1 Constraints Limitations:

we have faced some constraints making "ARC", it was related to the limitations and the low quality of the resources.

3.2 Earlier Coursework:

there were many courses that helped us building this project , for example we have been benefited by the C++ course to be able to write the code , moreover the Microcontroller Lap helped us to handle the Arduino Chip and testing our code on it, and the electronic and circuit courses helped us in understanding the design and build our circuit also to know how to use lab instruments.

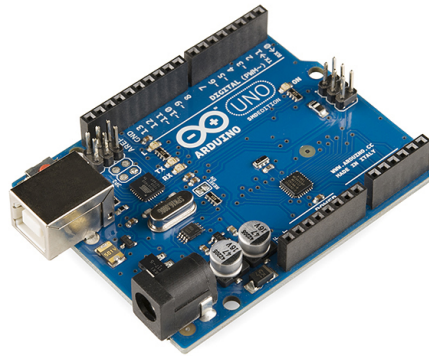
4 Methodology:

4.1 Robotic Arm:

To begin with, we have designed a Robotic Arm that have a 5 degrees of freedom, in general we have used 6 servo motors 3 of them are MG996R we used them for the shoulder , the elbow and the waist , as for the other 3 servo motors are smaller servo motors modules (SG90 micro servos) were used for the the wrist roll and wrist pitch, as well as the gripper.

4.1.1 Component:

- Arduino Uno :



it's an open source Microcontroller developed by Arduino.cc[1] the board contains a set of Analog and digital i/o pins that can be interfaced to other circuits , in general it has 14 digital i/o pins and 6 analog pins that can be programmed with the Arduino IDE[2].

- The MG996R Servo Motor:



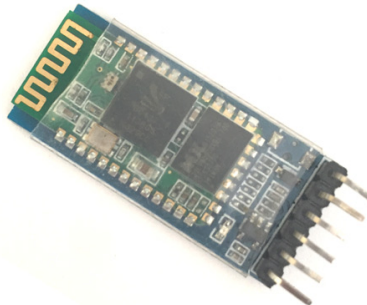
it's a is a metal gear servo motor Like the other servos the motor rotates from 0 to 180 degree counting on the duty cycle of the PWM that is supplied to it's signal pin

- SG90 Micro Servo Motor :



it's a carbon fiber gear which makes the servo motor much lighter than the metal gear servo motors, it rotates from 0 to 180 degree counting on the duty cycle of the PWM that is supplied to it's signal pin, it is used for small load applications.

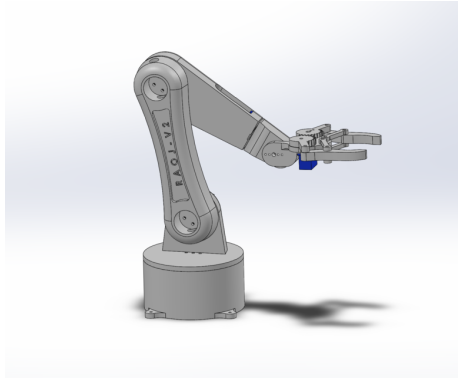
- HC-05 Bluetooth Module:



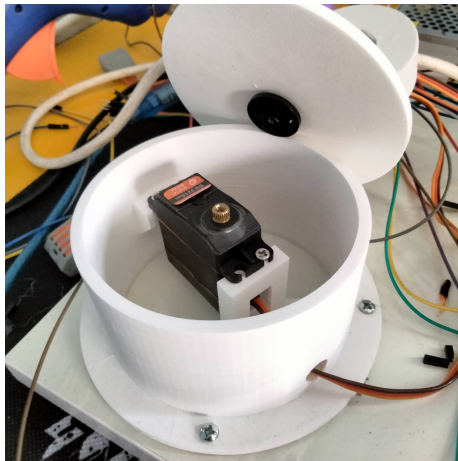
HC-05 is a very popular Bluetooth module which adds two-way (full-duplex) wireless functionality. this module can be used to make communication between two microcontrollers or communication between the micro controller with any device with Bluetooth functionality like a Phone or Laptop.It communicates with the help of USART at a baud rate of 9600.

4.1.2 The 3D module:

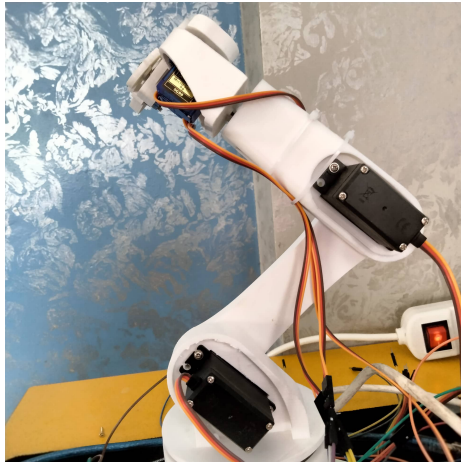
We got the 3D module of the arm online and 3D printed it



and after the printing was done we started to assemble it starting with the base (the waist) we attached it with an MG9967R servo motor



the same servo motor module as the shoulder and the elbow as shown in the figure below:

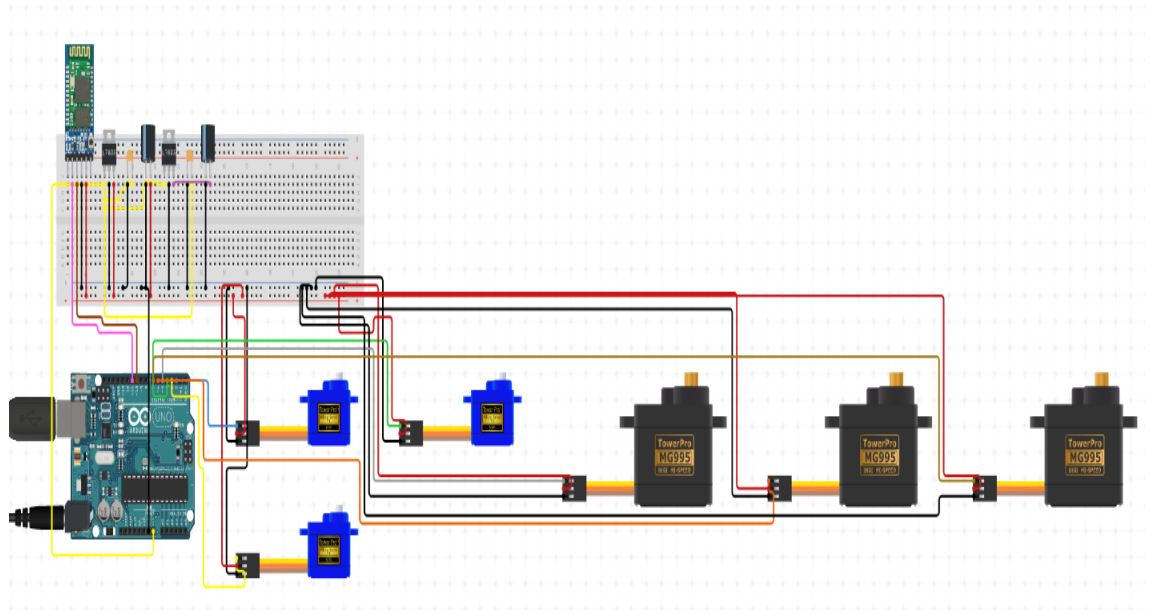


As for the gripper mechanism we attached them to the last three servo motors of module(SG90 micro servos)



4.1.3 Circuit:

As for the circuit it was as shown as the figure below



we have replaced the capacitors and the regulators with a 5V, 5A DC Power Supply (SMPS)



4.1.4 Standards/Codes:

The "ARC" project was built using Arduino and Arduino programming languages.

For the robotic arm first we included the servo library for servo motors, and we also include the SoftwareSerial library for the serial communication of the Bluetooth. then we define six servo motors that we have in the robotic arm and we also define the Bluetooth model at pins (3,4) we also define variables for storing the positions of the servo motors current and previous, as well as arrays for storing the positions for the automatic mode, the code was as following:

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

Servo servo01;
Servo servo02;
Servo servo03;
Servo servo04;
Servo servo05;
Servo servo06;
|
SoftwareSerial Bluetooth(3, 4);
int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos;
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos;
int servo03WP[8], servo04WP[8], servo05WP[8], servo06WP[8];
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50];
int speedDelay = 20;
int index = 0;
String dataIn = "";
int servo02WP [8] = {90, 180, 90, 180, 90, 180, 90, 180};
```

Then in the setup section we initialized the motors and the Bluetooth Module as well. and to make that possible we used the write() function which you simply give it the degree in the parameter and the motor moves to any position from 0 to 180 degrees. I also program the robotic arm to wave to us when it starts, this was done using a for-loop and an array of fixed positions for the elbow servo.

```

void setup() {
  servo01.attach(5);
  servo02.attach(6);
  servo03.attach(7);
  servo04.attach(8);
  servo05.attach(9);
  servo06.attach(10);
  Bluetooth.begin(9600);
  delay(1000);
  servo3PPos = 50;
  servo03.write(servo3PPos);
  servo4PPos = 180;
  servo04.write(servo4PPos);
  servo5PPos = 85;
  servo05.write(servo5PPos);
  servo6PPos = 180;
  servo06.write(servo6PPos);
  delay(20);
  for (int i = 0; i < 7 ; i++) {
    if (servo02WP[i] > servo02WP[i + 1]) {
      for ( int j = servo02WP[i]; j >= servo02WP[i + 1]; j--) {
        servo02.write(j);
        delay(10);
      }
    }
    if (servo02WP[i] < servo02WP[i + 1]) {
      for ( int j = servo02WP[i]; j <= servo02WP[i + 1]; j++) {
        servo02.write(j);
        delay(10);
      }
    }
  }
  delay(20);
}

```

In the Loop Section we used `Bluetooth.available()` function , to check whether that we are receiving data from the glove or the Smartphone. if it is true, we use the `readString()` function to read the data as string and store it into the `dataIn` String variable. And depending on the data that we receive we make the suitable moves on the servo motors.

```
void loop() {
  if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString();
```

The incoming data from either the smartphone or the glove will be starting with letters and the degree that we need to move the specified servo motor to. So we indicate which servo that we need to move by using the `startsWith()` function on the string variable `dataIn` for example if we have in the `dataIn` variable the value of ("s2150") the function `startsWith("s2")` will indicate the servo that needs to be moved and we also use the function `substring()` to get the remaining string which will be the degree and by converting it to integer value and by using the `servo.write()` function we can move the servo to the given degree.

```
if (dataIn.startsWith("s1")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servolPos = dataInS.toInt();
  if (servolPPos > servolPos) {
    for ( int j = servolPPos; j >= servolPos; j--) {
      servo01.write(j);
      delay(10);
    }
  }
  if (servolPPos < servolPos) {
    for ( int j = servolPPos; j <= servolPos; j++) {
      servo01.write(j);
      delay(10);
    }
  }
  servolPPos = servolPos;
}
```

```

    if (dataIn.startsWith("SAVE")) {
        servo01SP[index] = servo1PPos;
        servo02SP[index] = servo2PPos;
        servo03SP[index] = servo3PPos;
        servo04SP[index] = servo4PPos;
        servo05SP[index] = servo5PPos;
        servo06SP[index] = servo6PPos;
        index++;
    }
}

```

As for the Save method first we want to indicate if the save button was pressed then we save each servo position in the arrays as the above code.

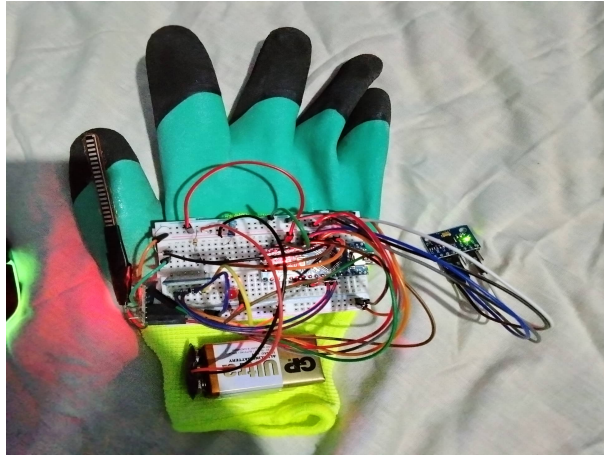
Then if we want the repet function we press the RUN button in the application by that we call the runservo() function which runs the stored steps in the arrays. And by this our code for the robot arm was done!

```

void runservo() {
    while (dataIn != "RESET") {
        for (int i = 0; i <= index - 2; i++) {
            if (Bluetooth.available() > 0) {
                dataIn = Bluetooth.readString();
                if (dataIn == "PAUSE") {
                    while (dataIn != "RUN") {
                        if (Bluetooth.available() > 0) {
                            dataIn = Bluetooth.readString();
                            if (dataIn == "RESET") {
                                break;
                            }
                        }
                    }
                }
            }
            // Servo 1
            if (servo01SP[i] == servo01SP[i + 1]) {
            }
            if (servo01SP[i] > servo01SP[i + 1]) {
                for (int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
                    servo01.write(j);
                    delay(speedDelay);
                }
            }
            if (servo01SP[i] < servo01SP[i + 1]) {
                for (int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
                    servo01.write(j);
                    delay(speedDelay);
                }
            }
        }
    }
}

```

4.2 Hand Gestures Glove:



The Hand Gestures Glove consists of the following components:

4.2.1 Components:

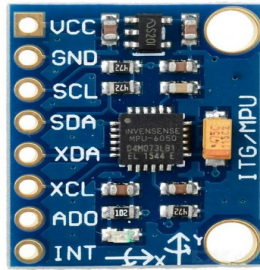
- Flex Sensors (used 3)



FLEX SENSOR terminal resistance changes when it is bent, when the surface of FLEX SENSOR is completely linear it will be having its nominal resistance. When it is bent 45° angle the FLEX SENSOR resistance increases to twice as before. And when the bent is 90° the resistance could go as high as four times the nominal resistance. So the resistance across the terminals rises linearly with bent angle thus so it is used to change

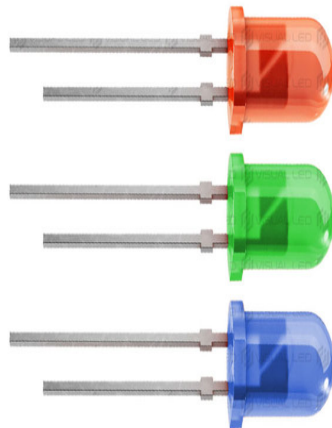
the current when the finger is bent and send the current to the Arduino, which in turn sends the command to the robotic arm.

- Accelerometer MPU6050 Module (Used 2)



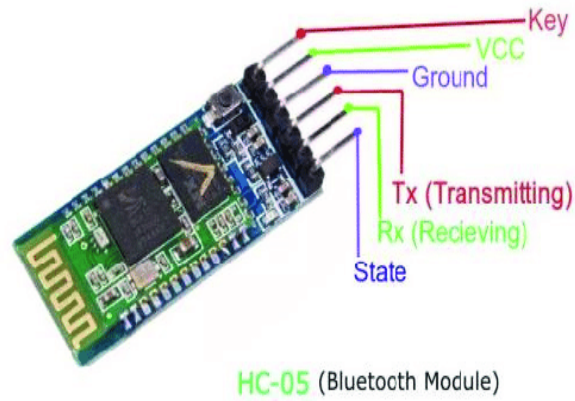
The MPU6050 is a Micro Electro-Mechanical Systems (MEMS) it consists of a 3-axis Accelerometer and 3-axis Gyroscope inside it. It is used to know the rotational speed on the axes and is transmitted to the robotic arm by assigning X, Y, Z hand gestures.

- LED



The LED indicator is used to ensure that the signals sent to the Arduino are received by hand glove.

- Bluetooth (HC-05) Module



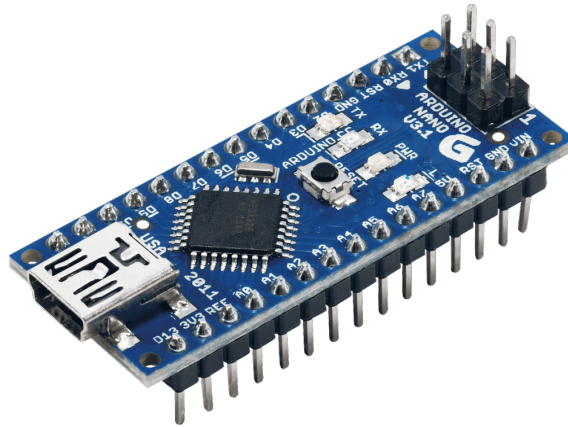
as explained above it's used to connect the microcontroller with another microcontroller or any device that uses the Bluetooth module , here we use it to send commands from the Arduino Nano to the HC-05 robotic arm receiver.

- 9 Volts battery



To provide the glove with the power it needs to function properly.

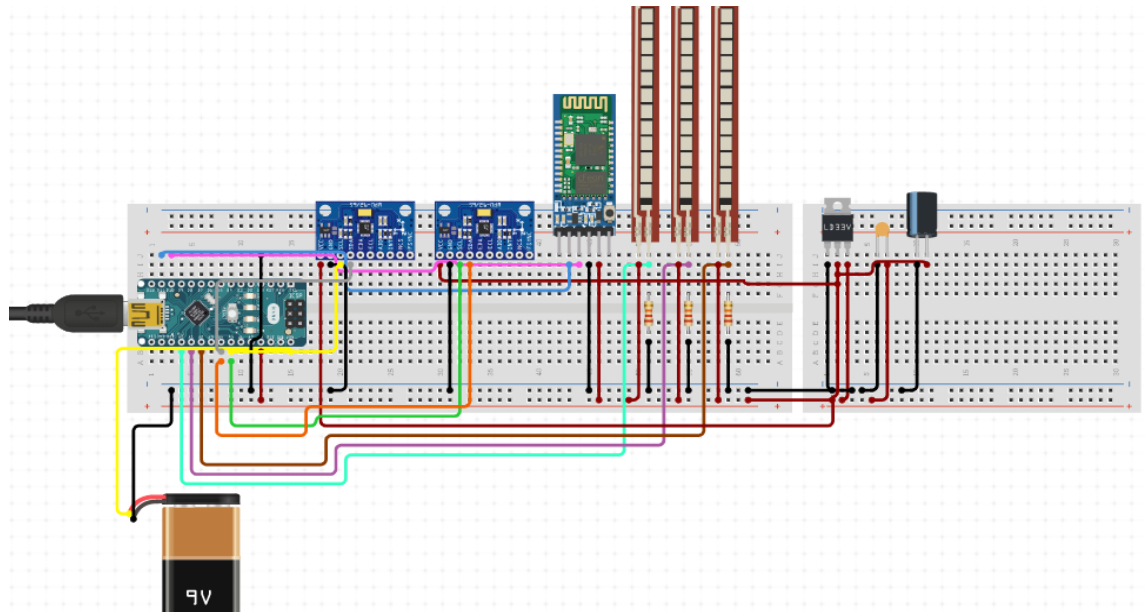
- Arduino Nano



The microcontroller that is used to control the sensors in the glove and send them to the main controller. It's very similar to the Arduino UNO in fact they have the same Processor (Atmega328p) thus so they both can share the same program, but the major difference between them is the size and the number of pins.

4.2.2 Hand Gestures Glove Circuit:

The Circuit was as shown in the figure below:



4.2.3 Hand Glove Code:

As for the Hand Glove code we started by connecting the 2 Bluetooth Modules with each others by making the hc-05 module on the Hand Glove as master and the other hc-05 module as slave this was done by using attention (AT) command for each module. Then we started writing our code by including the SoftwareSerial library again for the Bluetooth module and define some variables to help us read values from the sensors and send the command to the robotic arm :

```
#include <SoftwareSerial.h>
int thumb = 0; //Index thumb
int thumb_Data = A3;
int response_time = 100;
int thumb_high = 0;
int thumb_low = 0;
bool bool_caliberate = false;
SoftwareSerial Bluetooth(5, 6);
```

For the setup section we only use it to define the baud rate for the Bluetooth module:

```
void setup() {
  Bluetooth.begin(9600);
}
```

As for the loop section here we calibrate the upper and lower values of the Flex Sensors then We check continuously the current value of the sensor if it has exceeded the defined upper or lower values, if the value that we are reading from the flex sensor exceeds the upper limit or goes below the lower limit we send a specified data to the robotic arm to move a certain motor in a specific direction:

```
void loop() {  
  thumb = analogRead(thumb_Data);  
  if (bool_caliberate == false) {  
    delay(1000);  
    thumb_high = (thumb * 1.15);  
    thumb_low = (thumb * 0.9);  
    bool_caliberate = true;  
  }  
  delay(response_time);  
  if (thumb >= thumb_high) {  
    Serial.print("T");  
    Bluetooth.write("T");  
    delay(response_time);  
  }  
  if (thumb <= thumb_low) {  
    Serial.print("t");  
    Bluetooth.write("t");  
    delay(response_time);  
  }  
}
```

5 Conclusion

5.1 Summary

This work is an overview of how to build up a robotic arm by using servo motors and how to program it using Arduino UNO and make it able to be controlled by mobile application via Bluetooth or by hand gestures using Flex sensors and Accelerometers. the final version of this project will be widely used in industrial applications that requires speed and accuracy in manufacturing, because it will make many things easier and faster.

5.2 Future Work

There are many things that we can do to upgrade our project some of these ideas that we can do as follows:

- connect a camera to the arm and upgrade it to be able to control it via WiFi thus so you can control the arm from far away places.
- Work on the design of the arm to increase its strength and length to reach much greater distances
- Using larger motors to increase the ability to carry larger weights
- Developing the arm by making it able to move from one place to another
- Arms of different sizes and capacities can be made in many fields, for example, the use of a small-sized and high-precision arm in the field of medicine for surgical operations

References

- [1] Yusuf Abdullahi Badamasi. “The working principle of an Arduino”. In: *2014 11th international conference on electronics, computer and computation (ICECCO)*. IEEE, 2014, pp. 1–4.
- [2] Michael McRoberts. *Beginning arduino*. Apress, 2011.