ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018

Design and Implementation of Robot Arm Control Based on Matlab with Arduino Interface

[1] T.Rajesh, [2] M. Karthik Reddy, [3] Afreen Begum, [4] D.Venkatesh
[1][2] Asst. Professor, [3][4] Student
Balaji Inistitute of Technology and Science, Warangal, Telangana.

***Abstract*: -- In the present days, a number of situations exist where it is not possible for a human operator to do an activity on his/her own, due to a level of danger or difficulty involved. They may involve taking readings from an active volcano, entering a building on fire, diffusing a bomb, or collecting a radioactive sample. Rather than compromising on human lives, it is better to employ robotic systems for performing difficult tasks. Robotic systems are far superior in ensuring the accuracy of the system under adverse circumstances wherein a human operator may lose his/her composure and focus. Here we propose to build a robotic arm controlled by Matlab/Simulink interfacing with Arduino Uno. The development of this arm is based on Arduino platform and Matlab. A servo motor is a combination of DC motor, position control system, gears. The position of the shaft of the DC motor is adjusted by the control electronics in the servo, based on the duty ratio of the PWM signal. Servo is proposed for low speed, medium torque and accurate position application. These motors are used in robotic arm machines, flight controls and control systems. This project presents an interactive module for learning both the fundamental and practical issues of servo systems interface with ARDUINO UNO. This project, developed using Matlab coding tool, is used to control robotics applications. The objective of this project is to control the servo by using ARDUINO UNO with MATLAB & SIMULINK.**

***Keywords*- Arduino UNO, Servo motors, ATmega 328, matlab, pwm signal , robotic arm.**

I. INTRODUCTION

Nowadays, robots are increasingly being integrated into working tasks to replace humans, especially to perform repetitive tasks. In general, robotics can be divided into two areas, industrial and service robotics. International Federation of Robotics (IFR) defines a service robot as a robot which operates semi or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations. These robots are currently used in many fields of applications including office, military tasks, hospital operations, dangerous environment and agriculture. Besides, it might be difficult or dangerous for humans to do some specific tasks like picking up explosive chemicals, defusing bombs or to pick and place a bomb somewhere for containment, and for repeated pick and place action in industries. Therefore, a robot can replace a human to do work.
A robotic arm by definition is a robot manipulator, usually programmable, with functions similar to a human arm. The links of such a manipulator are connected by joints all owing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The business end of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand. The end effectors can be designed to perform any desired task such as welding, gripping, spinning, dropping etc., depending on the application. The robotic arm can be autonomous or controlled manually, which imparts to it the characteristic to be used to perform a variety of tasks with great accuracy. The robotic arm can be fixed or mobile (i.e. wheeled) and can be designed for industrial or home applications. [1][2]. There are various ways in which a robotic arm may be controlled. In the past, many researchers have worked to control a robotic arm through computer terminals, joysticks, even interfacing them with the internet so that they can be controlled from anywhere in the world [1],[2].

Typically, the following types of robotic arms are defined [3]:
- Cartesian/Gantry Robot
- Cylindrical Robot
- Spherical/Polar Robot
- SCARA Robot
- Articulate Robot
- Parallel Robot

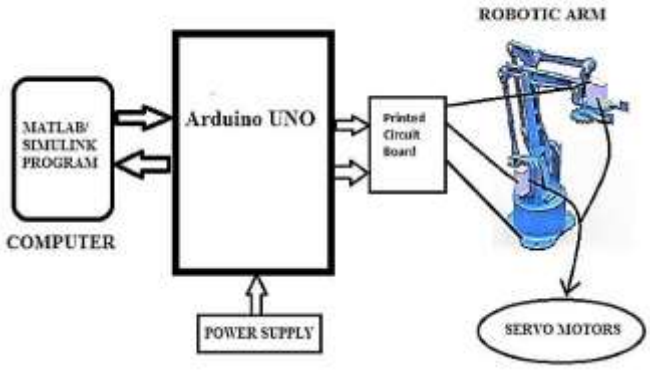
The proposed robotic arm is an Articulated Robot. Usually most of the robotic arms are controlled by a central controller which makes use of values taken in from the terminal that are entered by the user at the terminal to move the arm to particular coordinates in space. This makes the control very difficult as the control values of the motors are very difficult to predict to achieve a particular movement. This is easily achieved by our project.

In this study we are implementing it using MATLAB to track the human arm using different X and Y axis to control the robotic arm. For each position of hand there is a color to detect its positions that is from shoulder to elbow the color let us say

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018

Green, from elbow to wrist red color and then ahead of wrist a blue color is used. These colors will be detected by MATLAB and gives position to each and every color depending upon the axis defined for the video screen. These axis locations will be captured and interfaced with the controller. The controller will gives the command to the robot that, at movement of particular Location the particular robotic servo motor will move which is placed in the robotic arm



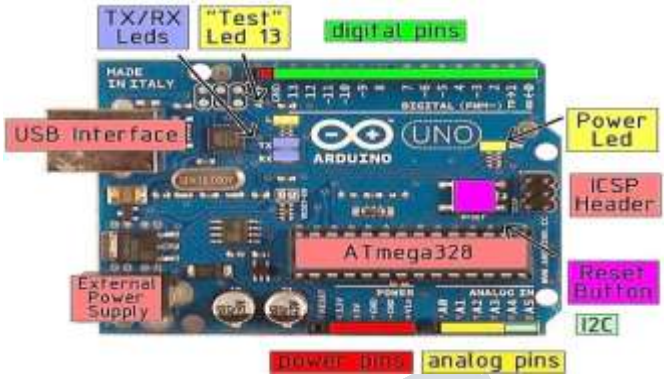
Block Diagram

II ARDUINO

The Arduino Uno [5] is a microcontroller board based on the ATmega328 (datasheet)[12]. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, 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. The Uno differs from all preceding boards in that it does not use the FTD USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform;

UNO BOARD

A micro-controller is a small computer on a single integrated circuit Containing a processor core, memory, and programmable input/output peripherals The important Part for us is that a micro-controller contains the processor (which all computers have) and memory, and some input/output pins that you can control. (Often called GPIO – General Purpose Input Output Pins)[13].



Arduino uno board

Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, 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. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version1.0 will be the reference versions of Arduno, moving forward.

III. IMPLEMENTATION

Robotic Arm Design Steps

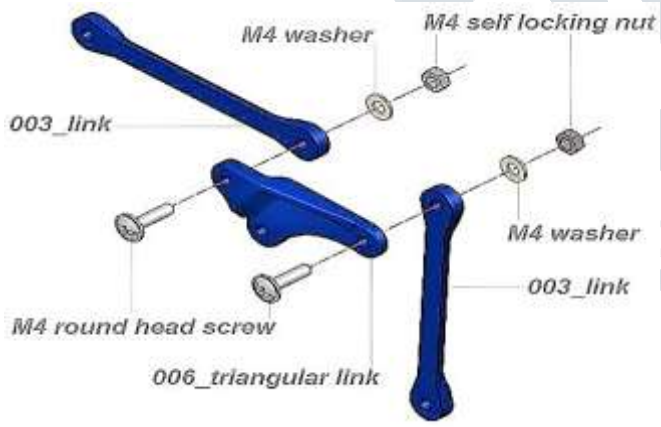
This ia a 3DPrinted robotic arm [10]. The design intent was to make something "easy" to build and quite cheap It uses MG90S small servos for driving the kinematics linkage and a Pololu mini maestro 12 to control the servos (but this is my choice any other methods are valid) [7]. The Arm can be

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018

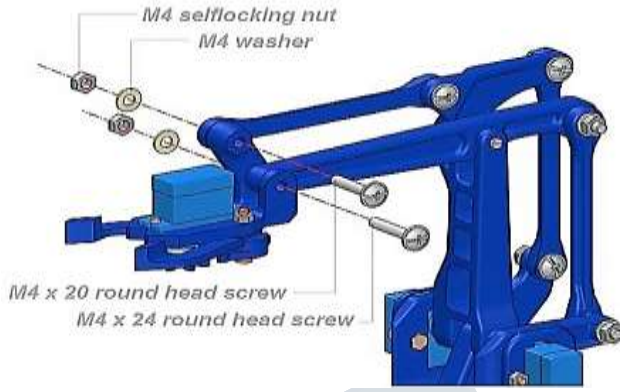
driven in several different ways: sketches, potentiometers, joystick, and WII nun chuck. After several trials the paper found very "easy" to use a controller from Pololu: Mini Maestro USB Servo Controller. You can attach up to 6 – 12 – 24 servos depend of the controller type. It is provided with a free configuration and control program for Windows and Linux that give you the power to drive the servo in manual moving slides; in the mean time you are able to set the values of speed and acceleration for any singular item. You can also build sequences of servo movements and run scripts stored in the internal script memory that can be automatically played back without any computer or external microcontroller connected.

Connect two link arms (003) to the Triangular link (006). Keep the M4 round heads screws to the inner side like shown on image and self-locking nuts to the outer side. The design involves holes of joints quite exact to allow to make them more precise using a drill bit and nuts are to be tightened till the locking of the joint, then consequently you must lose them until you obtain a smooth movement with the lower clearance between components. This rule is valid and is to be applied also for the following joint that involve use of self-locking nuts.



Attaching the Gripper

Attach the latest link to the fixed arm on the rear side of the base using a M4x20 a washer and a self-locking nut The last assembly step is to join the gripper to the horizontal arm as shown on the picture. At the end of the last step the ARM is ready to work. As an optional, in the 3D model downloadable from Thing hiverse, add a round ramp that allow to easy obtaining a loop test with a ball (3D printed, obviously!). It mean to make this tool to work you have to attach another servo (cheap SG90) to the end of the ramp. Keep the ramp center at a distance of about 180mm from the base vertical axis. There is also a 3D model of a support dedicated to the Pololu USB servo

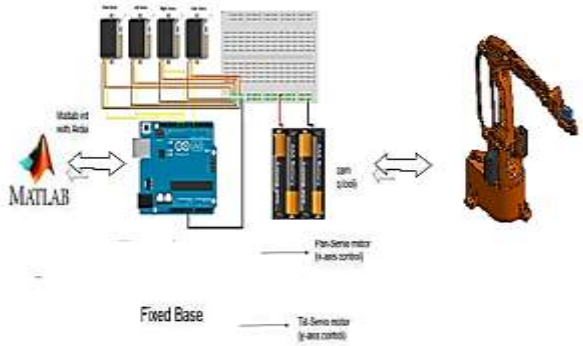


The ways to drive the servo are several. To explain it will take too much and this instructable is big enough .probably I'll make a new instructables dedicated, if get time. Anyway if want to explore there are quite enough material around the web. As told at the beginning, very easy using a Pololu USB servo Mini Maestro, it is not very cheap but solve a lot of problems. You have to install drivers, software and when connected to usb you're immediately able to drive the servos choosing their speed and acceleration also [14]. You can store the servo position to a sequence and when ready it can be played once or in a loop. Can also be stored in the internal script memory and it can be automatically played without computer connected.

Wiring Diagram

The best thing about a servo motor [4][6] is that it can be connected directly to an Arduino. Connect to the motor to the Arduino as shown in the table below:

Servo red wire – 5V pin Arduino
servo brown wire – Ground pin Arduino
Servo yellow wire – PWM(9) pin Arduino



Connection of arduino uno and servo motors

Servo Motor Working

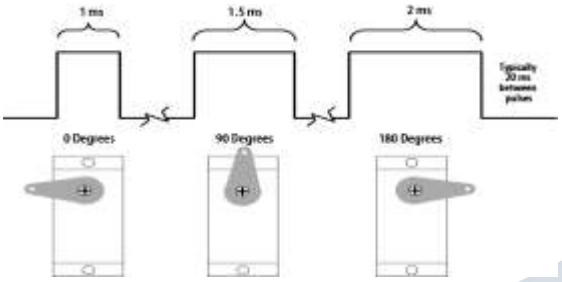
The third pin accepts the control signal which is a pulse-width modulation (PWM) signal. It can be easily produced by all micro- controllers and Arduino board. This accepts the signal

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018

from your controller that tells it what angle to turn to. The control signal is fairly simple compared to that of a stepper motor. It is just a pulse of varying lengths. The length of the pulse corresponds to the angle the motor turns to.

For detailed and simplified explanation of Pulse Width Modulation (PWM):
The pulse width sent to servo ranges as follows:
Minimum: 1 millisecond ---> Corresponds to 0 rotation angle.
Maximum: 2millisecond ---> Corresponds to 180 rotation angle.



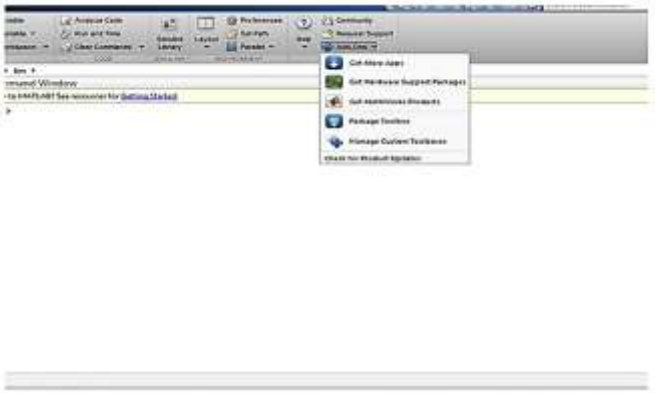
Any length of pulse in between will rotate the servo shaft to its corresponding angle. For example, 1.5 ms pulse corresponds to rotation angle of 90 degree. This is will explained in figure above

Steps for uploading the program into Arduino Uno Board:
Connect your arduino to the usb port of your computer. This may require a specific usb cable. Every arduino has a different virtual serial-port address, so you will need to reconfigure the port if you're using different arduinos. As in this step, let us try to control the servo motor using a signal generated from Matlab. To achieve this, first need to connect the servo motor appropriately to the Arduino. The servo motor will have three wires: power, ground, and signal [9]. Connect the wires as described below:

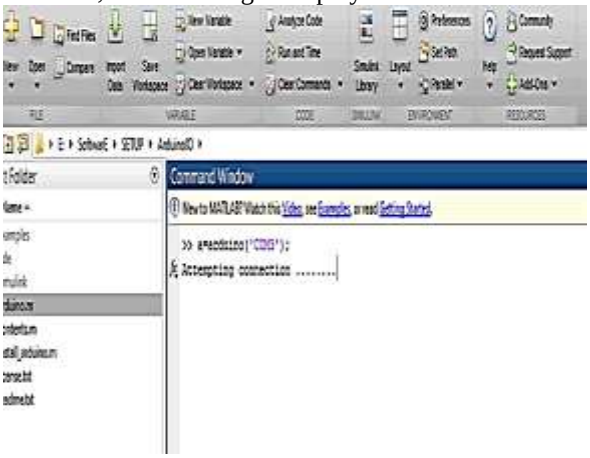
1. Connect the power wire (usually) red) to the 5V pin.
2. Connect the ground wire (usually black) to the ground pin.
3. Connect the signal wire (usually orange) to digital pin 4.

The most important thing is to select the Arduino IO folder as your current folder in Matlab. You can do it by clicking on 'browse the folder' Now your first command will be to create a variable a=arduino("COM3"); , it will create a variable 'a' which will be used to communicate between Arduino and Matlab. After that it will take few second to connect your Arduino to Matlab. You can see it in picture given below. In () you have to write about that COM port on which our Arduino is connected.

MATLAB Interface
Matlab is a high-performance language for technical computing. The name mat lab stands for matrix laboratory. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.
MATLAB program to control a servo motor connected to an Arduino.
The MATLAB Support Packages for Arduino makes use MATLAB to write programs for the Arduino. Support packages are available for 32-bit and 64-bit Windows, 64-bit Mac OS, and 64-bit Linux.

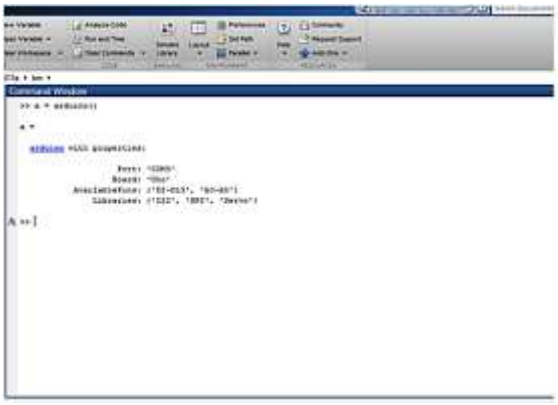


Installing Packages
To install packages MATLAB Arduino support packages, start MATLAB and click Add-Ons>Get Hardware Support Packages. Select the package you want to install and follow the installation instructions in the installer window.
Once the packages are installed, Arduino board is connected to the PC with a USB cable and following command is run.
>> a = arduino()
Now, MATLAB attempts to communicate with the Arduino.
If successful, the following is displayed in command window.



ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018



This display shows that my Arduino Uno is connected to COM3. If MATLAB was unable to connect to the Arduino, an error message will be displayed.

There are my steps and code in order to control and interface.
1. Dowload the arduino i/o support package and installed it.
2. Connect the arduino to PC via ubs port
3.Dowload and install the arduino software and 'upload' the adiosrv into arduino duemilanove (also try in romeo) board
4. Type arduino in the matlab and it connect successful.
5 .and then type "a=arduino('COM8')", a.servoAttach(1)"and "a.servoWrite(1,50)" to move the servo motor 1(pin 9) from 0 to 50 degree.

Write and read Servo position

Change the shaft position of the servo motor from 0(minimum) to 1(maximum) with 0.2, e.g 36 degrees, increment. Display the current position each time the position changes.

```
for angle = 0:0.2:1
    writePosition(s, angle);
    current_pos = readPosition(s);
    current_pos = current_pos*180;
    fprintf('Current motor position is %d degrees\n', current_pos);
    pause (2);
end
```

IV. ROBO JOINTS

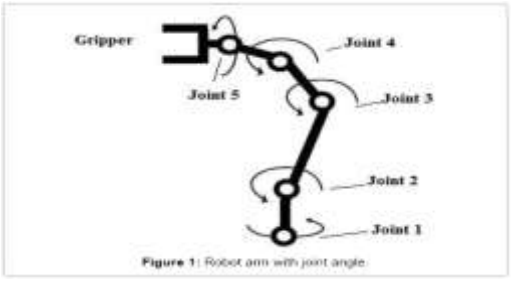
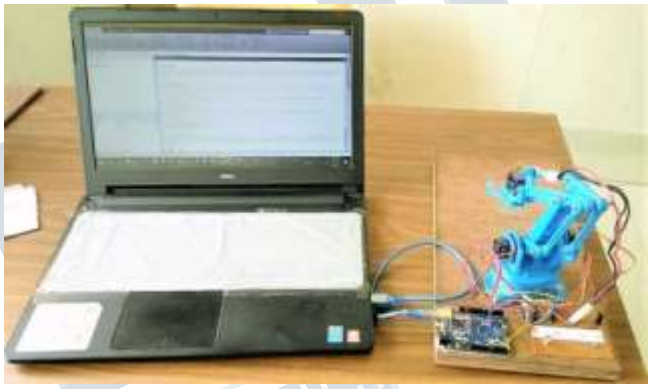


Figure 1: Robot arm with joint angle

V. RESULTS & OBSERVATIONS

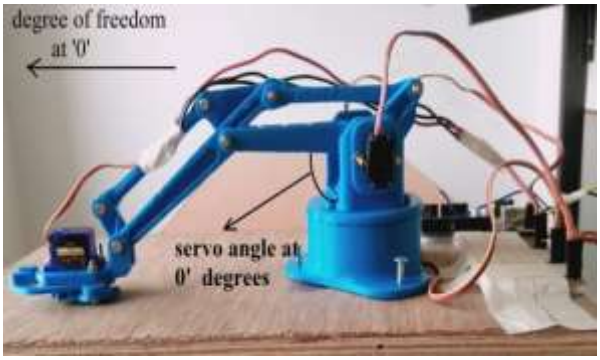
A Servo motor can be controlled with an Arduino UNO development board using the hardware and the software approach off MATLAB /SIMULINK. With the program described running, and connections properly made, the Servo motor will continuously rotate 180'' [8]. The Servo motor can be coupled to a shaft which can be used for numerous applications .i .e robotic arm .These methods can be modified slightly to move a motor in a variety of ways to fit another particular scenario. The ease of use of these 2 pieces of hardware offer a simple way to control a motor with a MATLAB program.



Robot Arm Controlling By Servomotors Using MATLAB/ SIMULINK Interfacing with Arduino

Servo motor At Right to Forward and Backward

In this step the command is given to servo-1,it's start at initial position ,after small time delay it's degree of freedom at 90' .The process of step-1 follows as ; MATLAB Library Browser -> MATLAB Support Package for Arduino Hardware -> Common Window -> Write program -.> Run

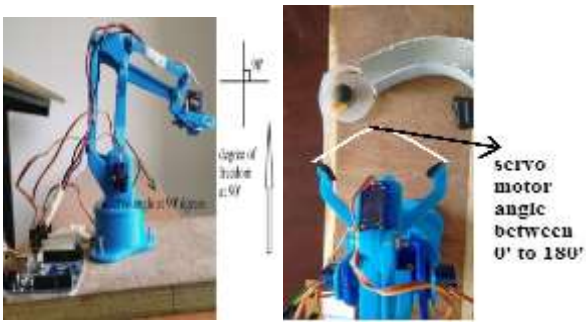


Servomotor at handling clamp to pick and place

In this step the command is given to servo-4, it's start at initial position ,after small time delay it's degree of freedom at 180' .The process of step-4 follows as ;

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018



MATLAB Library Browser -> MATLAB Support Package for Arduino Hardware -> Common Window -> Write program -.> Run[11]
Servomotor at handling clamp to pick and place

VI. CONCLUSION

In this paper the implementation of robo arm control with matlab based and auduino interface. The arm is controlled by the matlab and it will listen all the instructions through auduino. This is a very easily implemented straight forward method to coupled man-machine interface. The purpose of this article was to make an arduino UNO control a servo motor accurately. The servo motor reached the specific angles of movement that we pre-programmed and it repeated it over and over. With this new found Knowledge of simple servo motor control, now we can go out into the world and control things for more autonomously With this new found knowledge simple user interfaces in processing to control motors connected to an Arduino board with IP5 library to create a simple GUI. (graphical user interface) of MATLAB/SIMULINK .After setting up the Arduino circuit ,control servo's rotation angle with a slider in processing. In this project we can control the multiple servos with small time delay to robotic arm .The MATLAB /SIMULINK process following as MATLAB Library Browser -> MATLAB Support Package for Arduino Hardware -> Common Window -> Write program -.> Run

VII. FUTURE SCOPE

In basic robotics we design machines to do the specified tasks and in the advanced version of it robots are designed to be adaptive, that is, respond according to the changing environment and even autonomous, that is, capable to make decisions on their own. While designing a robot the most important thing to be taken in consideration is, obviously, the function to be performed. Here comes into play the discussion about the scope of the robot and robotics. Robots have basic levels of complexity and each level has its scope for performing the requisite function. The levels of complexity of robots is defined by the members used in its limbs, number of limbs, number of actuators and sensors used and for advanced robots the type and number of microprocessors and microcontrollers used. Each increasing component adds to the scope of functionality of a robot. With every joint added, the degrees of freedom in which a robot can work increases and with the quality of the microprocessors and microcontrollers the accuracy and effectiveness with which a robot can work is enhanced.

REFERENCES

[1] Mohd A. K. Yusoffa, R. E. Saminb, B. S. K. Ibrahim Wireless mobile robotic arm", International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), July 2012

[2] W. M. H. W. Kadir, R. E. Samin, B.S. K. Ibrahim, " Internet controller robotic arm", International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), July 2012.

[3] R. J. Wang, J. W. Zhang, J. Xu, H. Liu, "The multiple-function intelligent robotic arms", IEEE International Conference on Fuzzy Systems, 2009.

[4] www.servodatabase.com ›

[5]https://www.arduino.cc/ TowerPro Servo

[6] www.ieee.servo.motor.conrol

[7] C. F. Olson, Lab. JP, Technol. CIo, Pasadena, ―Probabilistic self-localization for mobile robots‖; IEEE Transactions on Robotics and Automation 16,1, 55-66, 2000.

[8] J. C. Trinkle and R. James Milgram, ―Complete Path Planning for Closed Kinematics Chains with Spherical Joints‖; SAGE International Journal of Robotic Research 21, 9,773-789, 2002.

[9]. ―Six-servo Robot Arm‖ from ww.arexx.com.cn on 13-11-2011.

[10]. ―Robot software‖ from Wikipedia, the free encyclopedia on 04-3-2012

[11]. ―ATMEL – short notes‖, www.atmel.com

[12]. ―Arduino MCU‖, - www.arduino.com