Institute Technical Summer Project 2018

ARMstrong (Team ID- 3) (Image Creating Robotic Arm)

Team members :-
Amartya Mishra
Kirtan Vora
Denil Pareshbhai Mehta
Ayush Agrawal

LINK TO PHOTOS AND VIDEOS :-
https://drive.google.com/drive/folders/1nUojiPF3lEoT3eml68zUPkw2B1RI3i8Z?usp=sharing

LINK TO BILLS :-
https://drive.google.com/file/d/0B6YNyt3FX4bASEhnQ2Z5T2ZGZnBmdFVRb294cmVXXzBLRFhV/view?usp=sharing

MOTIVATION :-

To be honest, the only true motivation we had during the entire project was purely to make something that does something quite interesting to look at and to have something to brag about.

WHAT DOES OUR BOT DO? :-

To put it in simple words, our robot draws down an image feeded to it on a piece of paper. Using image processing and an assembly of motors that make it look like an arm, this metallic artist draws down the drawing you want it to draw on paper. Basically we collect data on where to draw lines using Image Processing and using this data, an Arduino Uno board controls the motor movements to draw the image.

TECHNICAL ASPECTS OF THE PROJECT :-

The entire project's technical work can be broadly divided into 3 aspects/parts :-

- Mechanical aspect
- Electrical aspect
- Software aspect

Mechanical aspect :-

The mechanical aspect was mainly comprised of the **metallic members** connected to make the arm and the **wooden base**. This is what gives you the first impression. The metallic members were ordered from Robokits India. We ensured that the wooden base platform was given proper elevation by fixing **wooden supports** at the four corners of the bottom of the base on which the platform rests.

Electrical Aspect :-

The major components under this aspect were the **3 servo motors** causing the arm's motion and the **power source** (9V cell in series with six AA cells). For the electrical connections of individual servo motors with the arduino and the power source, the connections are shown in this video. For making the connections compact, we used connector jumper wires running from the bottom of the base stuck to it using black tapes and over it, we used a **breadboard**. (suggestion :- Duracell is preferred for batteries. Not a promotional suggestion :p)

Software aspect :-

The microprocessor used in our project is Arduino Uno and the platform we have used for writing the codes is MATLAB. The entire software part of the project can be categorized into

a) Edge detection of input image and creating matrix of the pixel values.
b) Feeding the data obtained after edge detection into the code for servo motors.
c) Code for movement of servo as per acquired data.

For edge detection, we used the **canny edge detection** method. The corresponding "edge image" obtained was read as a matrix of bit values to each pixel.

For linking the data of the matrix obtained to the servo code, we used the SimuLink package for Arduino hardware in MATLAB.

THEORY INVOLVED IN THE PROJECT :-

As discussed earlier, the significant theories involved in this project were **Canny edge detection** to convert the input .jpeg/.jpg or any similar file to a .bmp file that was achieved using MATLAB (We have screenshots available for the codes) and using the binary data to create a matrix in MATLAB to assign a pixel value (1 for white and 0 for black) to each pixel in the drawing space grid. After receiving the corresponding pixel values for each pixel, the data was used to move the servos. For writing the code for the servo movement, the algorithm behind the code can be found here. Some points kept in mind while designing the project in whole were :-

a) The minimum angle that our servo motors could rotate was 1 degree.

b) As the fixed servo had to rotate the entire mass of the arm, the maximum torque that the fixed servo could provide had to be higher than the remaining two servos.

c)The power source had to provide adequate potential difference for all the 3 Servo motors to run simultaneously.

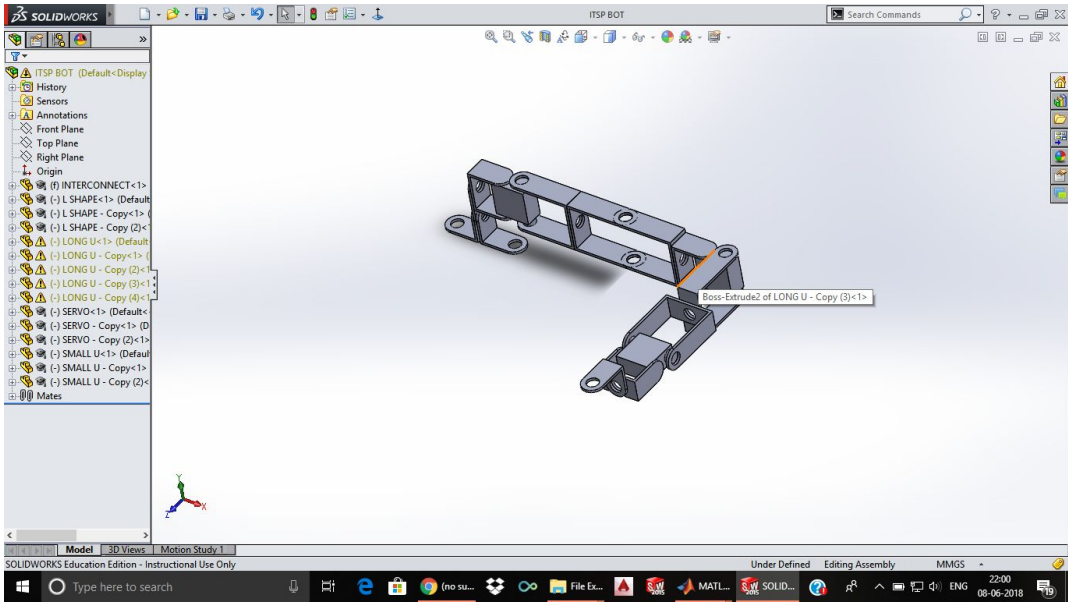
d)A castor wheel may be used to support the weight of the arm and prevent it from Tilting towards the base

PROBLEMS WE FACED AND WHAT WE CAME UP WITH :-

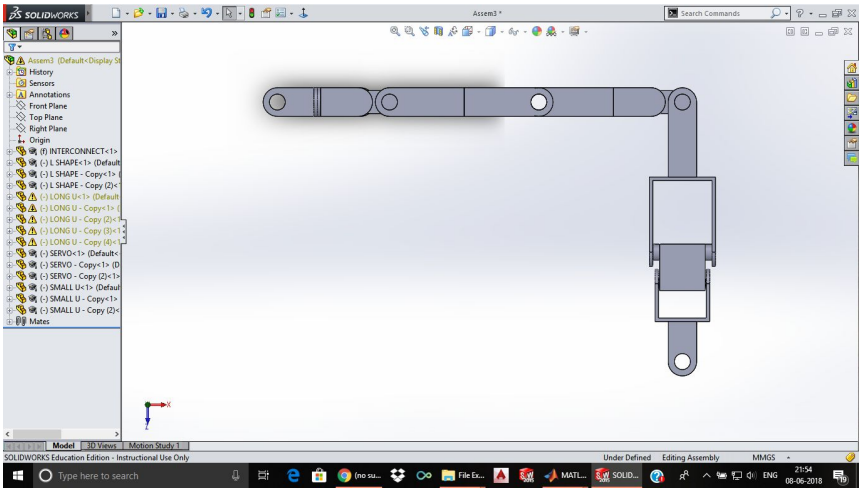
I) PROBLEMS RELATED TO MECHANICAL ASPECTS :-

a) While making the arm one needs to keep in mind the weight of the arms because the whole arm will be supported by the servo at the fixed end .

b) One also need to choose servo of sufficient torque since the degrees of freedom of the rm will be governed by the servos only . We after coming out with a 3D cad model noted the approximate length of l1 and l2 along with the weight of the bot . Through this we decided that we need dual shaft servo motors ; one with 35 kg cm torque and two with 16 kg cm torque . If the torque is not of sufficient magnitude the arm will not move .



OUR 3D CAD MODEL



II) PROBLEMS RELATED TO ELECTRICAL ASPECTS :-

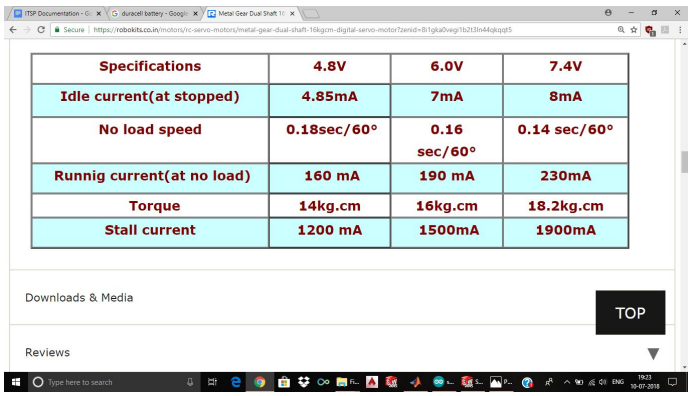
a) The major problem with servo is that its torque is directly proportional to the amount of voltage applied across the servos. One of our servo which was marked **" 18.2 kg cm at 7.4v "** was working properly at **9 volts** .

b)One cannot power all the motors with a single power source by connecting them in parallel as the desired motion of the arm is not observed .

c)Using DURACELL could seem to be solving above problems but soon after using it regularly will drain it and the above problems start to prevails .

d)Using lead acid battery(12V) could be an alternative but then it becomes expensive to buy 3 lead acid batteries . Since it is chargeable , it wont give pain in future .

e)We have ARDUINO UNO as our controller and haven't found any problem in using it . We have only faced the power issues .One can also use RASPBERRY PI if they are comfortable with it .



Specifications	4.8V	6.0V	7.4V
Idle current(at stopped)	4.85mA	7mA	8mA
No load speed	0.18sec/60°	0.16 sec/60°	0.14 sec/60°
Runnig current(at no load)	160 mA	190 mA	230mA
Torque	14kg.cm	16kg.cm	18.2kg.cm
Stall current	1200 mA	1500mA	1900mA

Downloads & Media

TOP

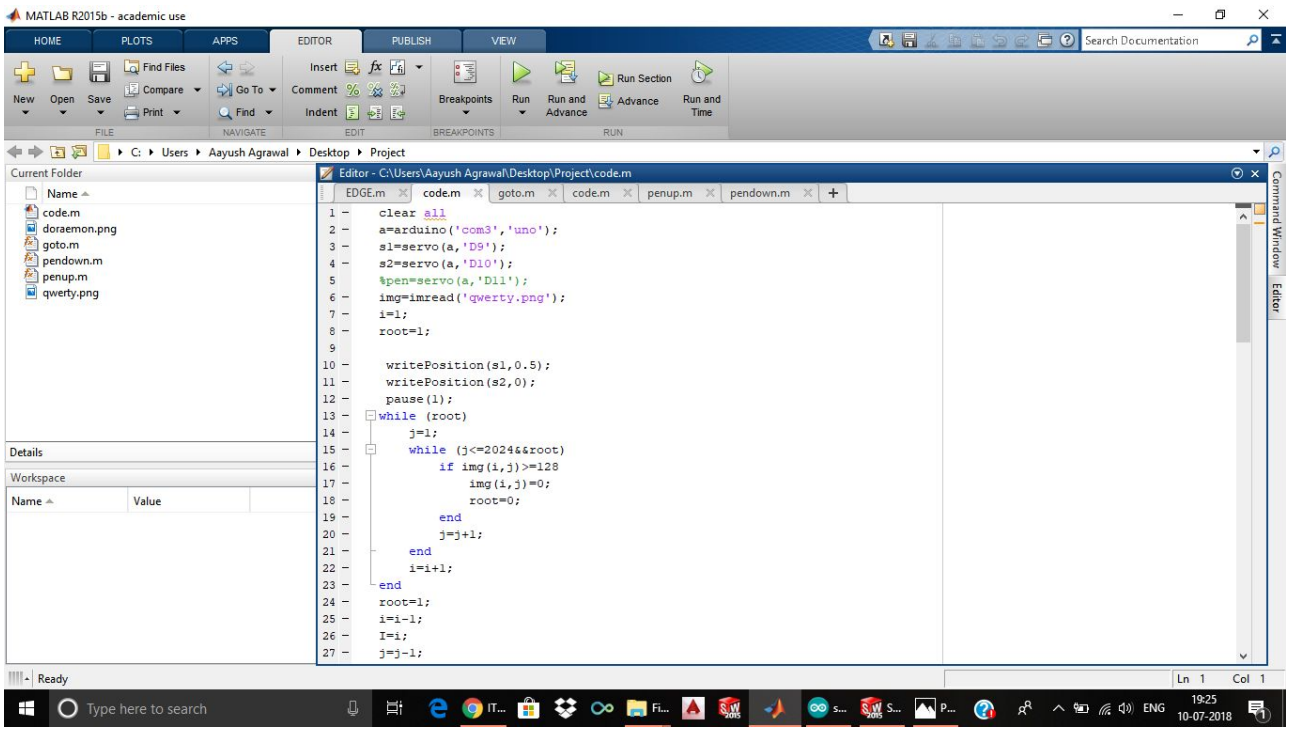
Reviews

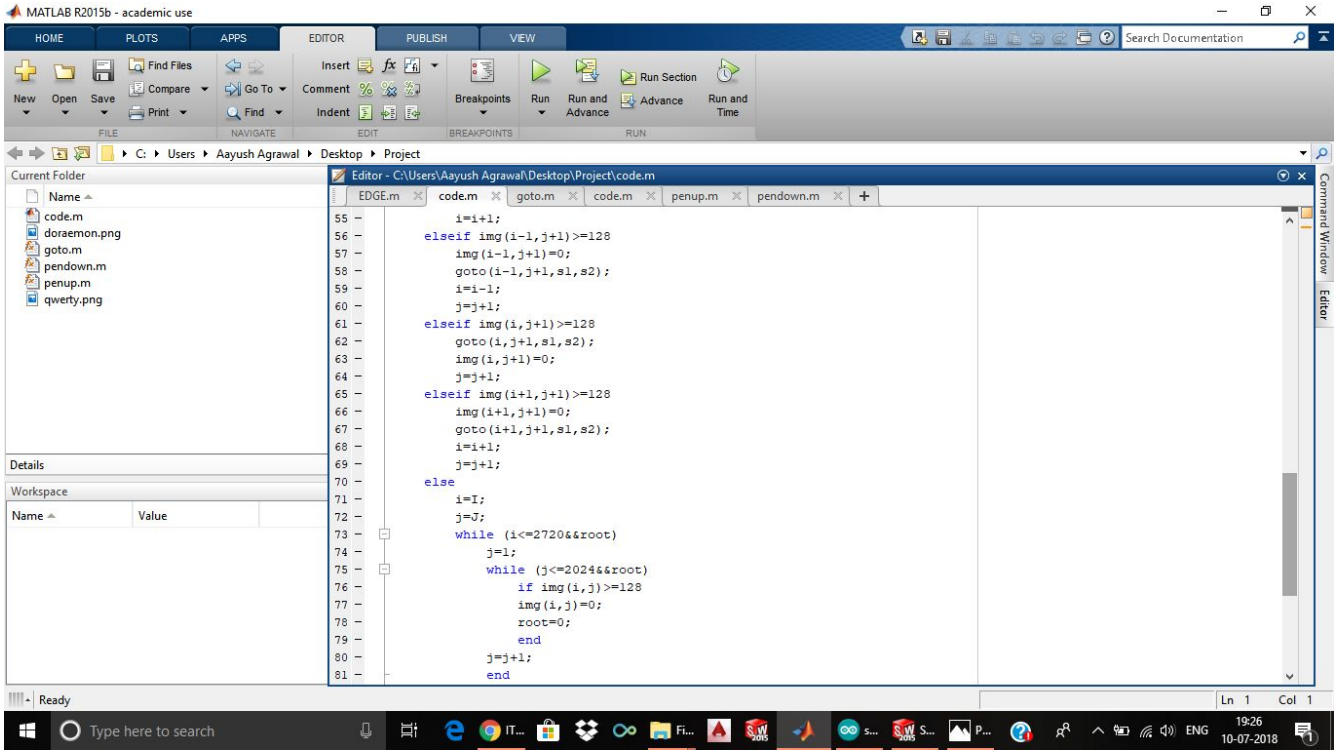
III) **PROBLEMS RELATED TO SOFTWARE ASPECTS** :-

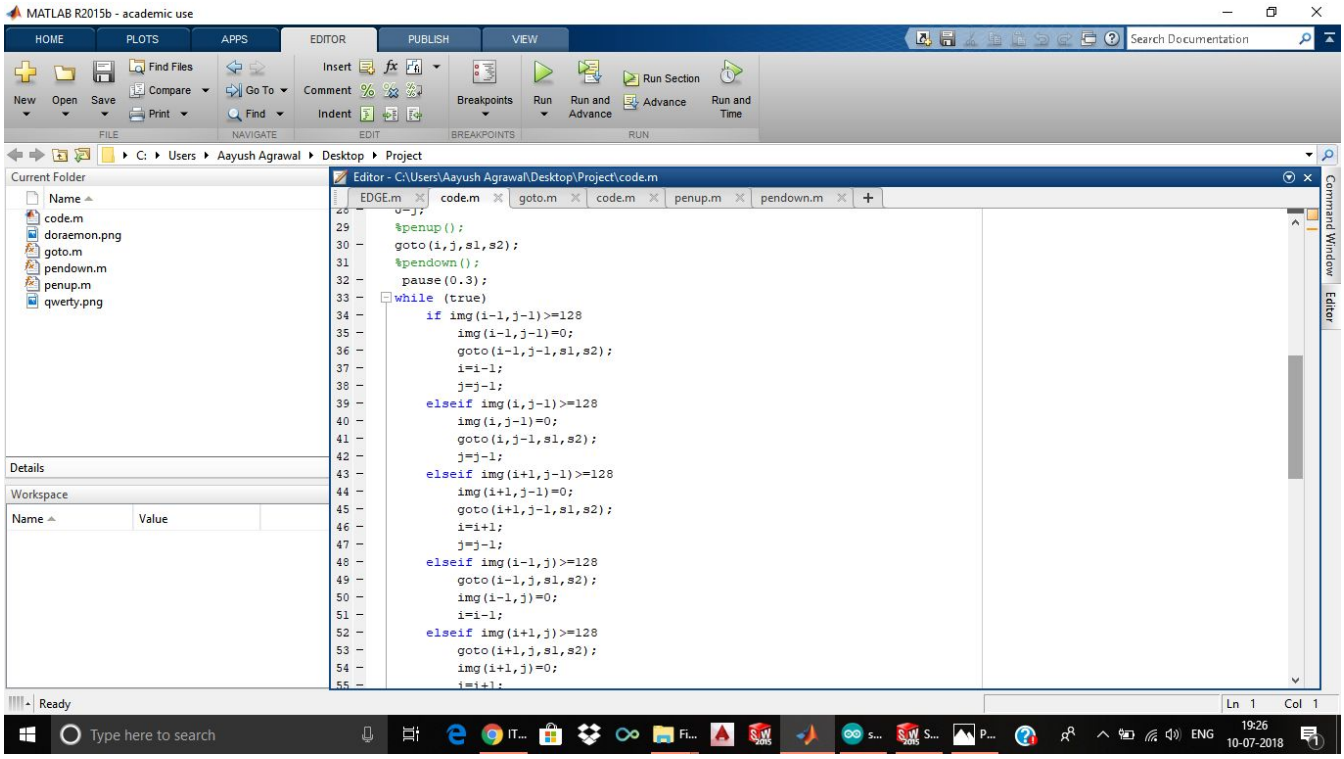
a) One needs to frame a algorithm for getting the job done .

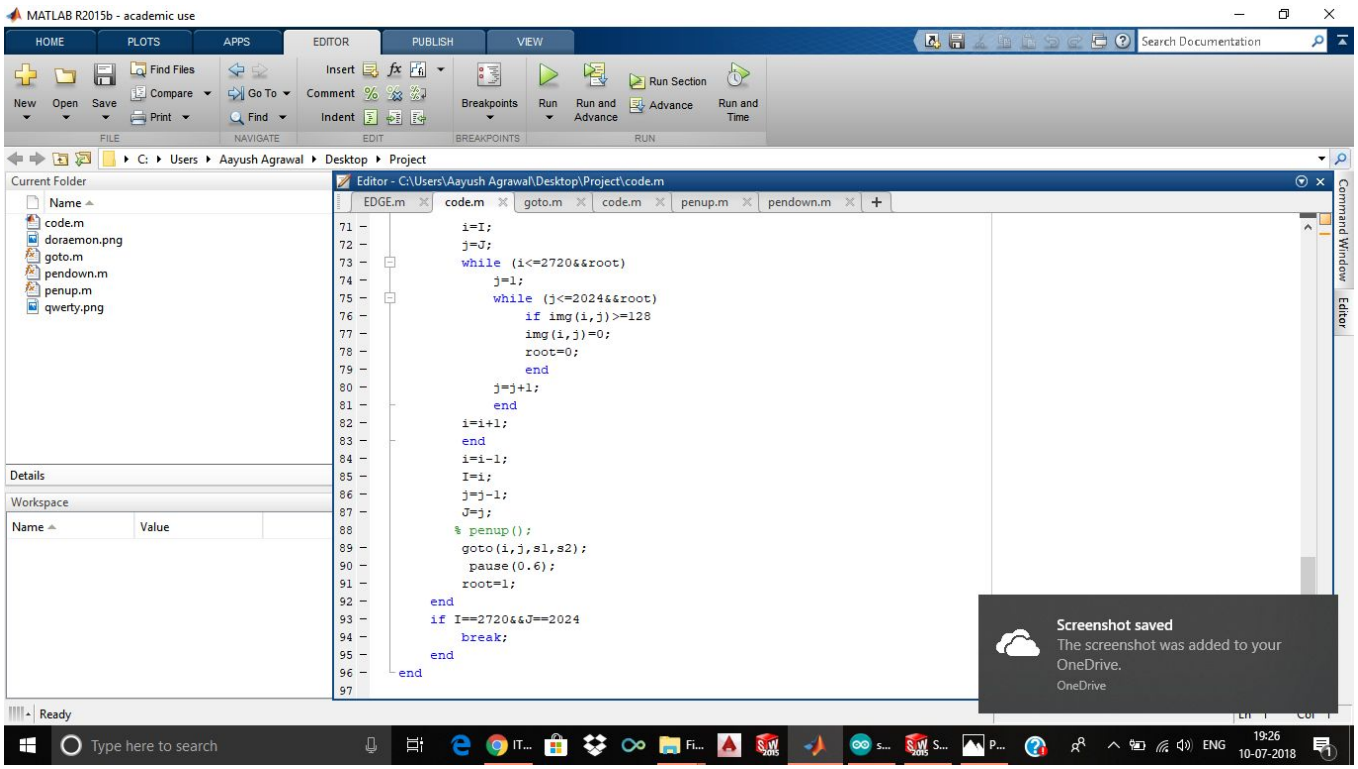
b) **Our algorithm** :- Every rgb image is made of three layers . Each layer has a matrix .We first converted the rgb image into grayscale image and then extracted the edges of the image .Now we scanned the final image pixel by pixel .Our image contained only black(0) and white(255) pixel .The time we find the first white pixel the pointer of the arm will align itself as per the position of that pixel in that matrix .We used inverse kinematics ,through this one can find the angle of rotation of the servo using coordinates of the pixel in the matrix .Then we scan the pixels adjacent to the pixel found , if a white pixel is found then the bot start moving the pointer in that direction . As the pointer moves the previous pixels set to ZERO . If the line seems to be terminating then we scan the page starting from the first pixel found . The same thing repeats until all pixels of matrix are set to zero.
c)It is difficult to write the code independently so we chose to write it in MATLAB using SIMULINK in it . Through Matlab also , one can run the code in Arduino Uno .

OUR CODE :

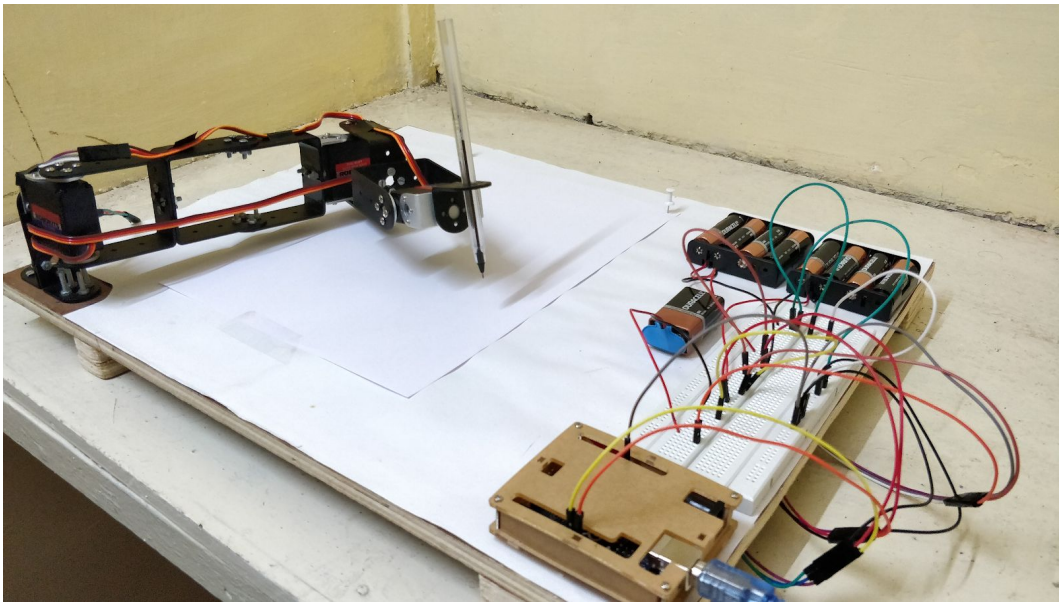



MATLAB R2015b - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
C: ▸ Users ▸ Aayush Agrawal ▸ Desktop ▸ Project
Current Folder
Name
code.m
doraemon.png
goto.m
pendown.m
penup.m
qwerty.png
Details
Workspace
Name
Value
Editor - C:\Users\Aayush Agrawal\Desktop\Project\code.m
EDGE.m
code.m
goto.m
code.m
penup.m
pendown.m
i=i+1;
elseif img(i-1,j+1)>=128
img(i-1,j+1)=0;
goto(i-1,j+1,s1,s2);
i=i-1;
j=j+1;
elseif img(i,j+1)>=128
goto(i,j+1,s1,s2);
img(i,j+1)=0;
j=j+1;
elseif img(i+1,j+1)>=128
img(i+1,j+1)=0;
goto(i+1,j+1,s1,s2);
i=i+1;
j=j+1;
else
i=I;
j=J;
while (i<=2720&&root)
j=1;
while (j<=2024&&root)
if img(i,j)>=128
img(i,j)=0;
root=0;
end
j=j+1;
end
Command Window
Editor
Ready
Ln 1
Col 1
Type here to search
ENG
19:26
10-07-2018


MATLAB R2015b - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
C: ▸ Users ▸ Aayush Agrawal ▸ Desktop ▸ Project
Current Folder
Name
code.m
doraemon.png
goto.m
pendown.m
penup.m
qwerty.png
Details
Workspace
Name
Value
Editor - C:\Users\Aayush Agrawal\Desktop\Project\code.m
EDGE.m
code.m
goto.m
code.m
penup.m
pendown.m
%penup();
goto(i,j,s1,s2);
%pendown();
pause(0.3);
while (true)
if img(i-1,j-1)>=128
img(i-1,j-1)=0;
goto(i-1,j-1,s1,s2);
i=i-1;
j=j-1;
elseif img(i,j-1)>=128
img(i,j-1)=0;
goto(i,j-1,s1,s2);
j=j-1;
elseif img(i+1,j-1)>=128
img(i+1,j-1)=0;
goto(i+1,j-1,s1,s2);
i=i+1;
j=j-1;
elseif img(i-1,j)>=128
goto(i-1,j,s1,s2);
img(i-1,j)=0;
i=i-1;
elseif img(i+1,j)>=128
goto(i+1,j,s1,s2);
img(i+1,j)=0;
Command Window
Editor
Ready
Ln 1
Col 1
Type here to search
ENG
19:26
10-07-2018


MATLAB R2015b - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
C: ▸ Users ▸ Aayush Agrawal ▸ Desktop ▸ Project
Current Folder
Name
code.m
doraemon.png
goto.m
pendown.m
penup.m
qwerty.png
Details
Workspace
Name
Value
Editor - C:\Users\Aayush Agrawal\Desktop\Project\code.m
EDGE.m
code.m
goto.m
code.m
penup.m
pendown.m
i=I;
j=J;
while (i<=2720&&root)
j=1;
while (j<=2024&&root)
if img(i,j)>=128
img(i,j)=0;
root=0;
end
j=j+1;
end
i=i+1;
end
i=i-1;
I=i;
j=j-1;
J=j;
% penup();
goto(i,j,s1,s2);
pause(0.6);
root=1;
end
if I==2720&&J==2024
break;
end
end
Command Window
Editor
Screenshot saved
The screenshot was added to your OneDrive.
OneDrive
Ready
Type here to search
ENG
19:26
10-07-2018

FINALLY WHAT WE HAVE MADE



COMPONENTS' LIST :-

Sl.No.	Component	Quantity
(1)	Wooden Plank	1
(2)	L-Shaped Interconnect	3
(3)	Linear Interconnect	1
(4)	U-Shaped Aluminium Servo Bracket	5
(5)	Servo pack	3
(6)	Jumper wires	Plenty
(7)	Pb-Acid battery	1
(8)	Bread-board	1

(9)	**Arduino Uno board**	**1**
(10)	**AA cells**	**6**
(11)	**9V cell**	**1**
(12)	**Castor wheel (was not eventually required)**	**1**

LINKS:-

Reference -
http://ijesc.org/upload/bc241f3e3974ff5cf18d5f3a907f103a.Drawing%20Robot%20using%20Inverse%20Kinematic%20Algorithm.pdf

Video related to our project -
https://www.youtube.com/watch?v=bbdQbyff_Sk