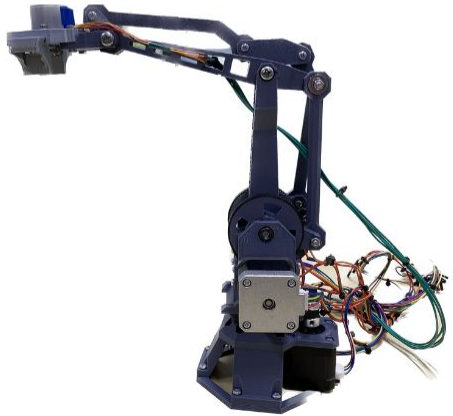
Introduction to Robot

3DOF Robot Arm

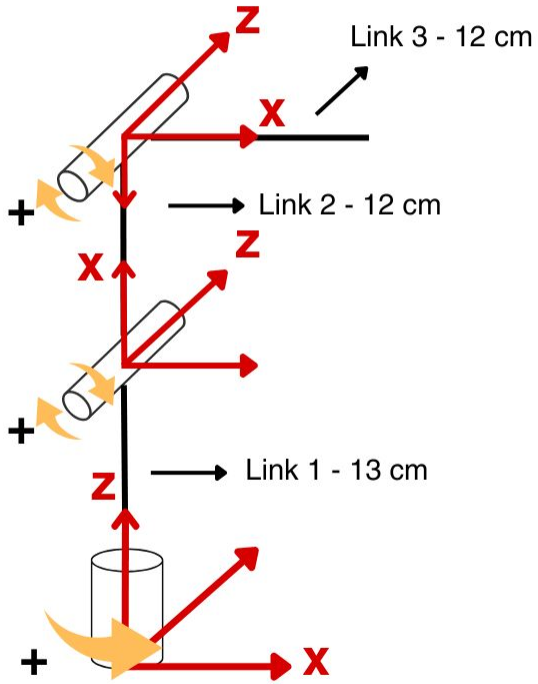
Controlled by Bluetooth



RAI 2 - Group 7

Paing Thet Ko - 65011416
Pyae Htoo Khant - 65011497
Thura Zaw - 65011685
Swan Pyae Sone - 65011683
Rachata Raktham - 65011500

Finding DH Parameters of 3DOF Robot Arm



DH Parameters

Joint	θ	d(cm)	α	a(cm)
1	θ1	13	-90	0
2	θ2-90	0	0	12
3	θ3+90	0	0	12

Calculating DH Matrix

DH_matrix.m

```
% DH parameters for a 3-joint robotic arm
a = [0, 12, 12];
alpha = [-pi/2, 0, 0];
d = [13, 0, 0];
theta = [(0*pi/180), (0*pi/180)-pi/2, (0*pi/180)+pi/2]; % theta 1, theta 2 - 90, theta 3 + 90

% Initialize overall transformation matrix as identity matrix
T = eye(4);

% Calculate DH transformation matrix for each joint and update overall transformation matrix
for i = 1:3
    % Calculate the DH transformation matrix for the current joint
    T_i = eye(4);  % Initialize identity matrix for current joint

    T_i(1,1) = cos(theta(i));
    T_i(1,2) = -sin(theta(i))*cos(alpha(i));
    T_i(1,3) = sin(theta(i))*sin(alpha(i));
    T_i(1,4) = a(i)*cos(theta(i));

    T_i(2,1) = sin(theta(i));
    T_i(2,2) = cos(theta(i))*cos(alpha(i));
    T_i(2,3) = -cos(theta(i))*sin(alpha(i));
    T_i(2,4) = a(i)*sin(theta(i));

    T_i(3,2) = sin(alpha(i));
    T_i(3,3) = cos(alpha(i));
    T_i(3,4) = d(i);

    % Update the overall transformation matrix
    T = T * T_i;
end

% Overall transformation matrix for the robotic arm
disp(T);
```

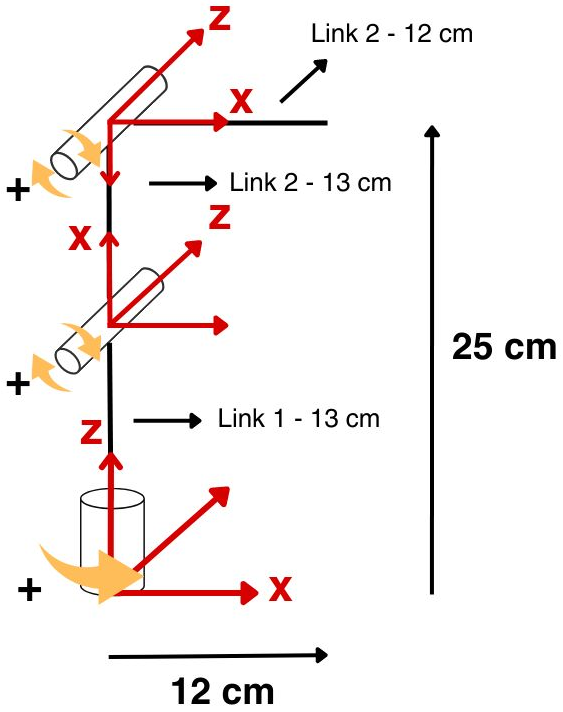
Command Window

```
>> DH_matrix
    1.0000         0         0   12.0000
         0    0.0000    1.0000   -0.0000
         0   -1.0000    0.0000   25.0000
         0         0         0    1.0000

fx >>
```

DH Matrix

```
>> DH_matrix
    1.0000         0         0   12.0000
         0    0.0000    1.0000   -0.0000
         0   -1.0000    0.0000   25.0000
         0         0         0    1.0000

fx >>
```

Home Position



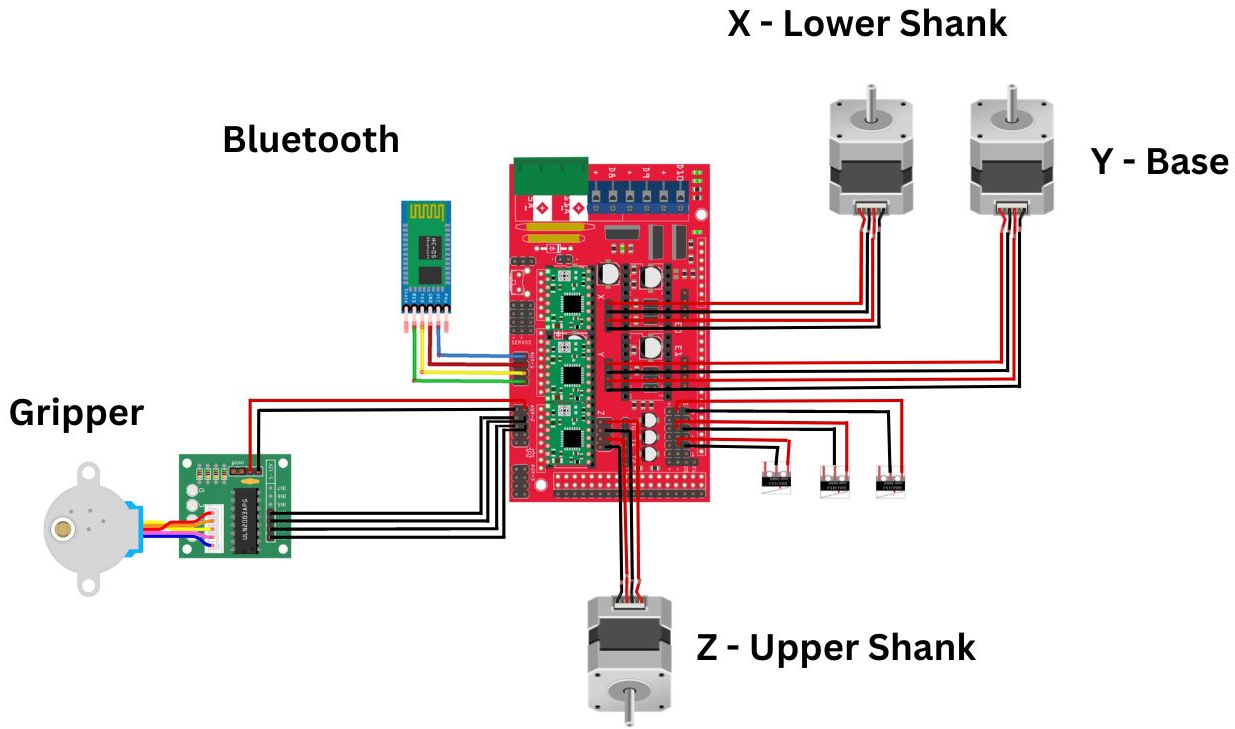
DH Matrix

```
>> DH_matrix
    1.0000         0         0   12.0000
         0    0.0000    1.0000   -0.0000
         0   -1.0000    0.0000   25.0000
         0         0         0    1.0000

fx >>
```

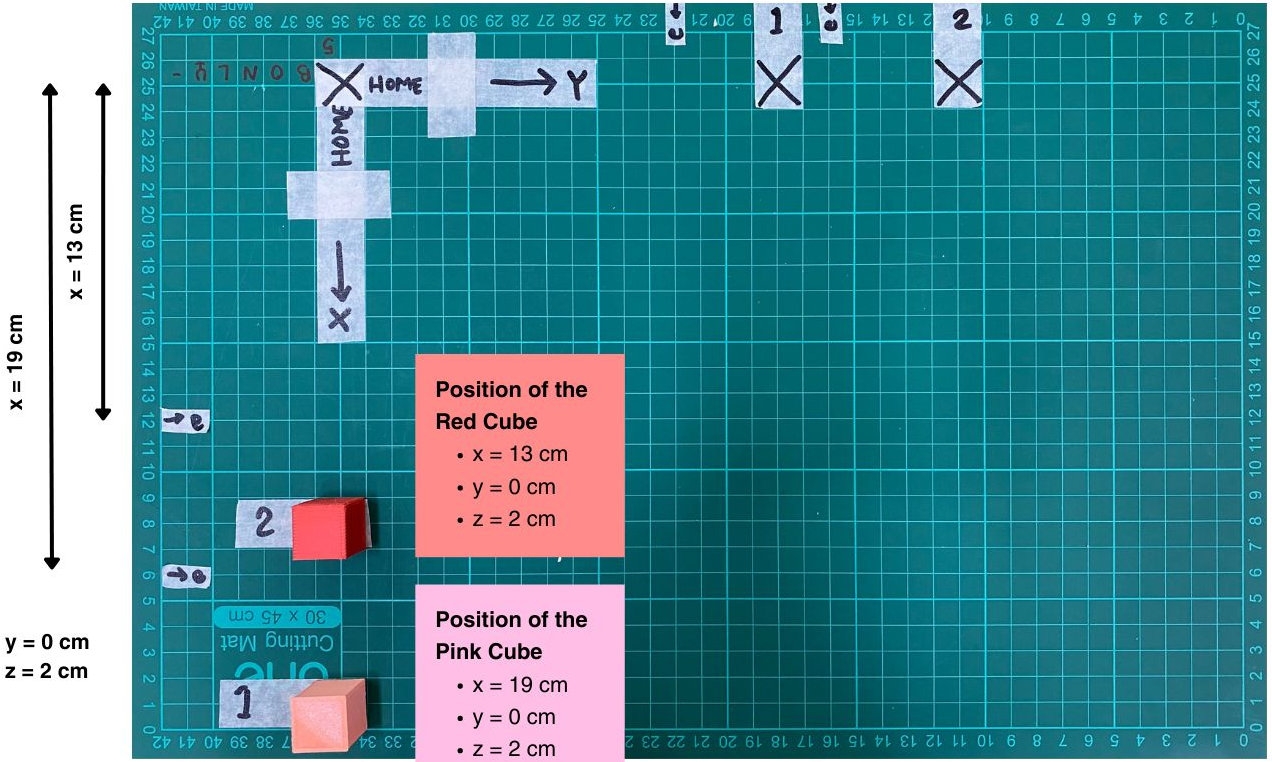
Position of the End Effector

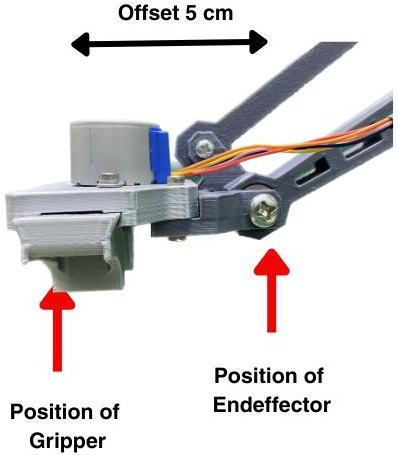
- x = 12 cm
- y = 0 cm
- z = 25 cm

Circuit Diagram

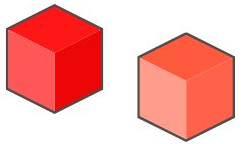


Pick Position

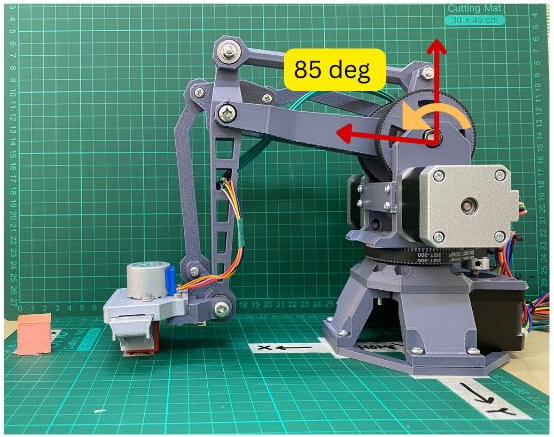




2 cm cubes

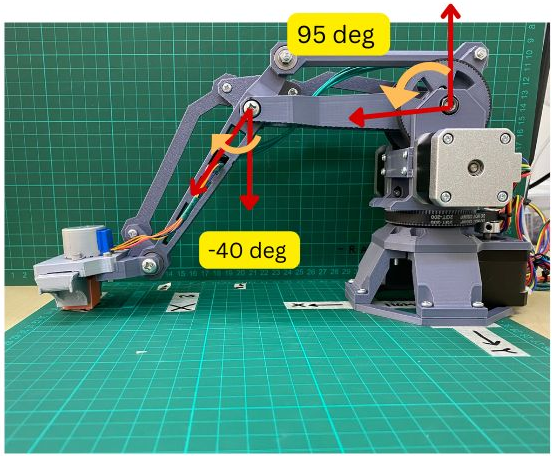


Finding joint angles for Pick Position



To get the pick position of Red cube, joint 2 (θ2) has to rotate about 85 degree.

- θ1 = 0
- θ2 = 85 deg
- θ3 = 0 deg



To get the pick position of Pink cube, joint 2 (θ2) has to rotate 95 degree and joint 3 (θ3) has to rotate -40 degree.

- θ1 = 0
- θ2 = 95 deg
- θ3 = -40 deg

Comparing with Calculated Positions from DH matrix

DH_matrix.m

```
% DH parameters for a 3-joint robotic arm
a = [0, 12, 12];
alpha = [-pi/2, 0, 0];
d = [13, 0, 0];
theta = [(0*pi/180), (85*pi/180)-pi/2, (0*pi/180)+pi/2]; % theta 1, theta

% Initialize overall transformation matrix as identity matrix
T = eye(4);

% Calculate DH transformation matrix for each joint and update overall tra
for i = 1:3
    % Calculate the DH transformation matrix for the current joint
    T_i = eye(4);  % Initialize identity matrix for current joint

    T_i(1,1) = cos(theta(i));
    T_i(1,2) = -sin(theta(i))*cos(alpha(i));
    T_i(1,3) = sin(theta(i))*sin(alpha(i));
    T_i(1,4) = a(i)*cos(theta(i));

    T_i(2,1) = sin(theta(i));
    T_i(2,2) = cos(theta(i))*cos(alpha(i));
    T_i(2,3) = -cos(theta(i))*sin(alpha(i));
    T_i(2,4) = a(i)*sin(theta(i));

    T_i(3,2) = sin(alpha(i));
    T_i(3,3) = cos(alpha(i));
    T_i(3,4) = d(i);

    % Update the overall transformation matrix
    T = T * T_i;
end

% Overall transformation matrix for the robotic a
disp(T);
```

Joint angles

- θ1 = 0
- θ2 = 85 deg
- θ3 = 0 deg

Command Window

```
>> DH_matrix
    0.0872   -0.9962         0   13.0002
    0.0000    0.0000    1.0000    0.0000
   -0.9962   -0.0872    0.0000    2.0915
         0         0         0    1.0000
```

Position of the Red Cube

- x = 13 cm
- y = 0 cm
- z = 2 cm

DH_matrix.m

```
% DH parameters for a 3-joint robotic arm
a = [0, 12, 12];
alpha = [-pi/2, 0, 0];
d = [13, 0, 0];
theta = [(0*pi/180), (95*pi/180)-pi/2, (-40*pi/180)+pi/2]; % theta 1, thet

% Initialize overall transformation matrix as identity matrix
T = eye(4);

% Calculate DH transformation matrix for each joint and update overall tra
for i = 1:3
    % Calculate the DH transformation matrix for the current joint
    T_i = eye(4);  % Initialize identity matrix for current joint

    T_i(1,1) = cos(theta(i));
    T_i(1,2) = -sin(theta(i))*cos(alpha(i));
    T_i(1,3) = sin(theta(i))*sin(alpha(i));
    T_i(1,4) = a(i)*cos(theta(i));

    T_i(2,1) = sin(theta(i));
    T_i(2,2) = cos(theta(i))*cos(alpha(i));
    T_i(2,3) = -cos(theta(i))*sin(alpha(i));
    T_i(2,4) = a(i)*sin(theta(i));

    T_i(3,2) = sin(alpha(i));
    T_i(3,3) = cos(alpha(i));
    T_i(3,4) = d(i);

    % Update the overall transformation matrix
    T = T * T_i;
end

% Overall transformation matrix for the robotic a
disp(T);
```

Joint angles

- θ1 = 0
- θ2 = 95 deg
- θ3 = -40 deg

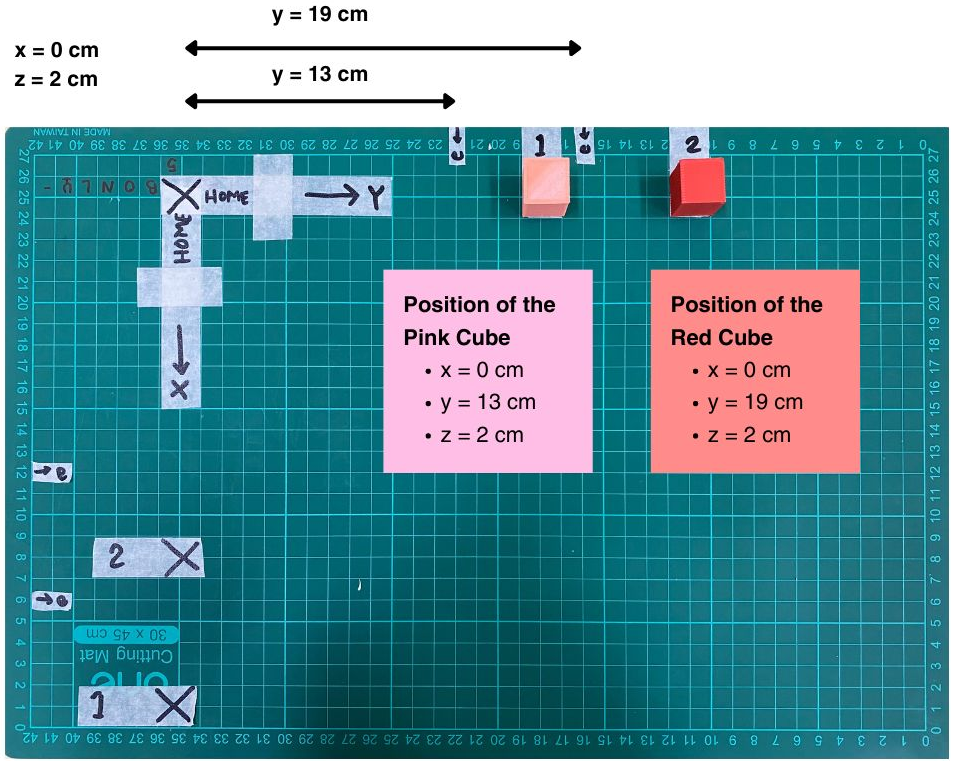
Command Window

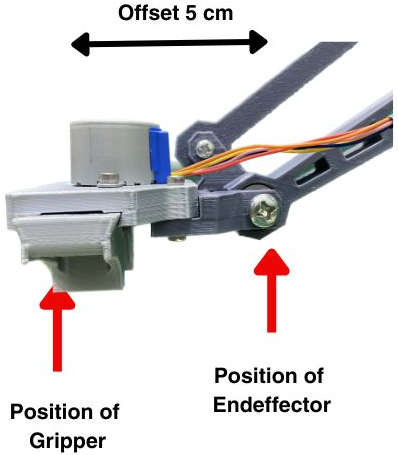
```
>> DH_matrix
    0.5736   -0.8192         0   18.8373
    0.0000    0.0000    1.0000    0.0000
   -0.8192   -0.5736    0.0000    2.1243
         0         0         0    1.0000
```

Position of the Pink Cube

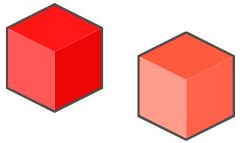
- x = 18.8 cm
- y = 0 cm
- z = 2.1 cm

Place Position

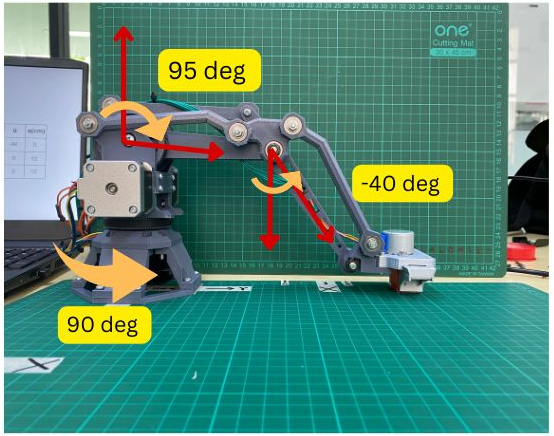




2 cm cubes

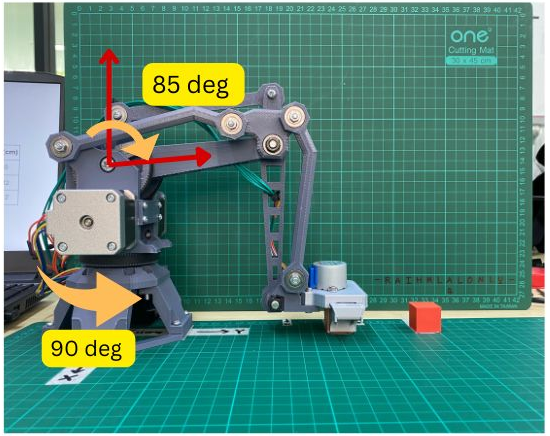


Finding joint angles for Place Positionion



To get the place position of Red cube, joint 1(θ1) has to rotate 90 degree, joint 2 (θ2) to 95 degree and joint 3 (θ3) to -40 degree.

- θ1 =90 deg
- θ2 = 95 deg
- θ3 = -40 deg



To get the pick position of Pink cube, joint 1 (θ1) has to rotate 90 degree and joint 2 (θ2) has to rotate 85 degree.

- θ1 = 90 deg
- θ2 = 85 deg
- θ3 = 0 deg

Comparing with Calculated Positions from DH matrix

DH_matrix.m

```
% DH parameters for a 3-joint robotic arm
a = [0, 12, 12];
alpha = [-pi/2, 0, 0];
d = [13, 0, 0];
theta = [(90*pi/180), (95*pi/180)-pi/2, (-40*pi/180)+pi/2]; % theta 1, the

% Initialize overall transformation matrix as identity matrix
T = eye(4);

% Calculate DH transformation matrix for each joint and update overall tra
for i = 1:3
    % Calculate the DH transformation matrix for the current joint
    T_i = eye(4);  % Initialize identity matrix for current joint

    T_i(1,1) = cos(theta(i));
    T_i(1,2) = -sin(theta(i))*cos(alpha(i));
    T_i(1,3) = sin(theta(i))*sin(alpha(i));
    T_i(1,4) = a(i)*cos(theta(i));

    T_i(2,1) = sin(theta(i));
    T_i(2,2) = cos(theta(i))*cos(alpha(i));
    T_i(2,3) = -cos(theta(i))*sin(alpha(i));
    T_i(2,4) = a(i)*sin(theta(i));

    T_i(3,2) = sin(alpha(i));
    T_i(3,3) = cos(alpha(i));
    T_i(3,4) = d(i);

    % Update the overall transformation matrix
    T = T * T_i;
end

% Overall transformation matrix for the robotic a
disp(T);
```

Joint angles

- θ1 = 90 deg
- θ2 = 95 deg
- θ3 = -40 deg

Command Window

```
>> DH_matrix
   -0.0000   -0.0000   -1.0000    0.0000
    0.5736   -0.8192    0.0000   18.8373
   -0.8192   -0.5736    0.0000    2.1243
         0         0         0    1.0000
fx >>
```

Position of the Red Cube

- x = 0 cm
- y = 18.8 cm
- z = 2.1 cm

DH_matrix.m

```
% DH parameters for a 3-joint robotic arm
a = [0, 12, 12];
alpha = [-pi/2, 0, 0];
d = [13, 0, 0];
theta = [(90*pi/180), (85*pi/180)-pi/2, (0*pi/180)+pi/2]; % theta 1, theta

% Initialize overall transformation matrix as identity matrix
T = eye(4);

% Calculate DH transformation matrix for each joint and update overall tra
for i = 1:3
    % Calculate the DH transformation matrix for the current joint
    T_i = eye(4);  % Initialize identity matrix for current joint

    T_i(1,1) = cos(theta(i));
    T_i(1,2) = -sin(theta(i))*cos(alpha(i));
    T_i(1,3) = sin(theta(i))*sin(alpha(i));
    T_i(1,4) = a(i)*cos(theta(i));

    T_i(2,1) = sin(theta(i));
    T_i(2,2) = cos(theta(i))*cos(alpha(i));
    T_i(2,3) = -cos(theta(i))*sin(alpha(i));
    T_i(2,4) = a(i)*sin(theta(i));

    T_i(3,2) = sin(alpha(i));
    T_i(3,3) = cos(alpha(i));
    T_i(3,4) = d(i);

    % Update the overall transformation matrix
    T = T * T_i;
end

% Overall transformation matrix for the robotic a
disp(T);
```

Joint angles

- θ1 = 90 deg
- θ2 = 85 deg
- θ3 = 0 deg

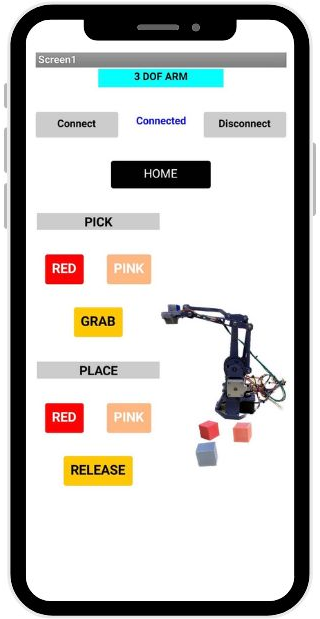
Command Window

```
>> DH_matrix
   -0.0000   -0.0000   -1.0000    0.0000
    0.0872   -0.9962    0.0000   13.0002
   -0.9962   -0.0872    0.0000    2.0915
         0         0         0    1.0000
fx >>
```

Position of the Pink Cube

- x = 0 cm
- y = 13 cm
- z = 2 cm

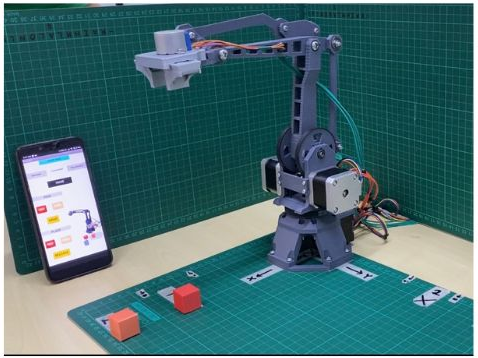
Control Methodology



Wireless serial communication

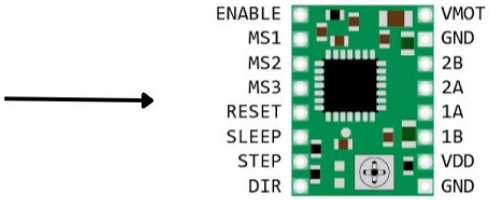
Arduino Mega

Created control software using MIT App Inventor Platform



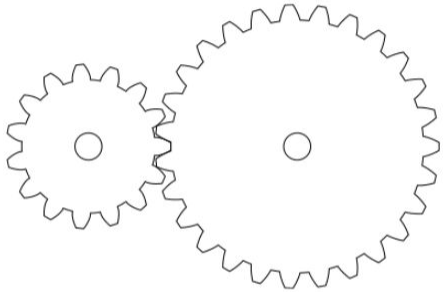
Motor Steps Calculation

200 steps per revolution
NEMA 17 HS4401



Full 16 microsteps resolution
3200 steps per revolution

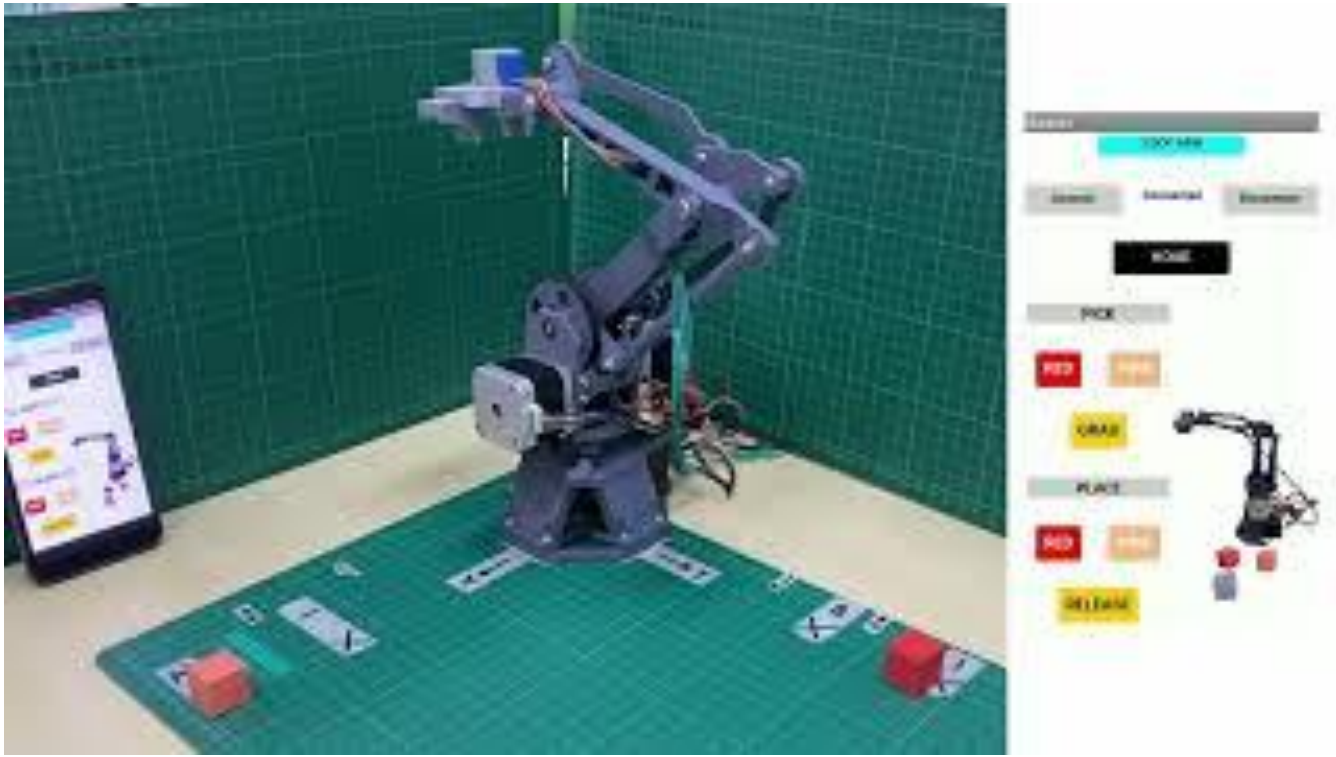
A4988 Motor Driver



Gear ratio = 4.5
14,400 steps per revolution

1 degree = 40 steps

Result Video



https://youtu.be/D6i2JJkeEAg