“Wenchoi”, a 3DOF Robotic Arm

ME134 HW #2 Final Prototype

By: Jacob Choi, Wenchang Gao

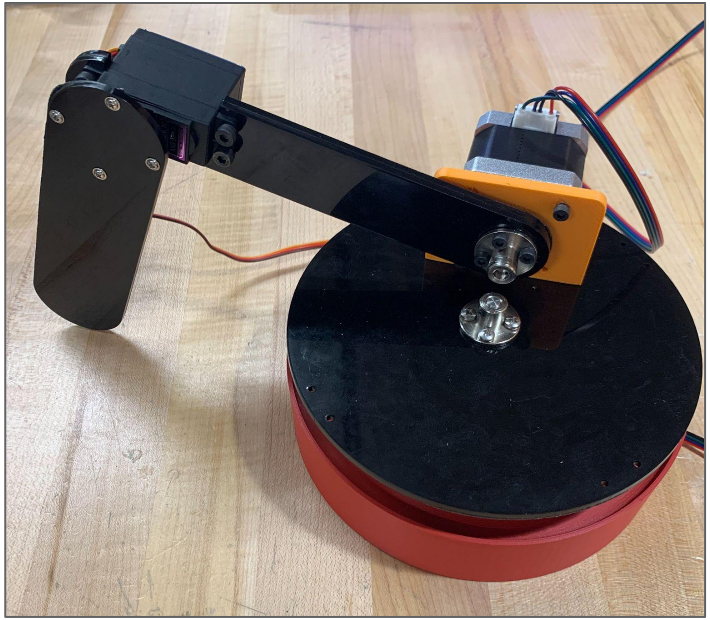
Sales Pitch

Functional Flexibility:

- 3 DOF and arm-like design
- 360° action space
- Potentials beyond writing letters

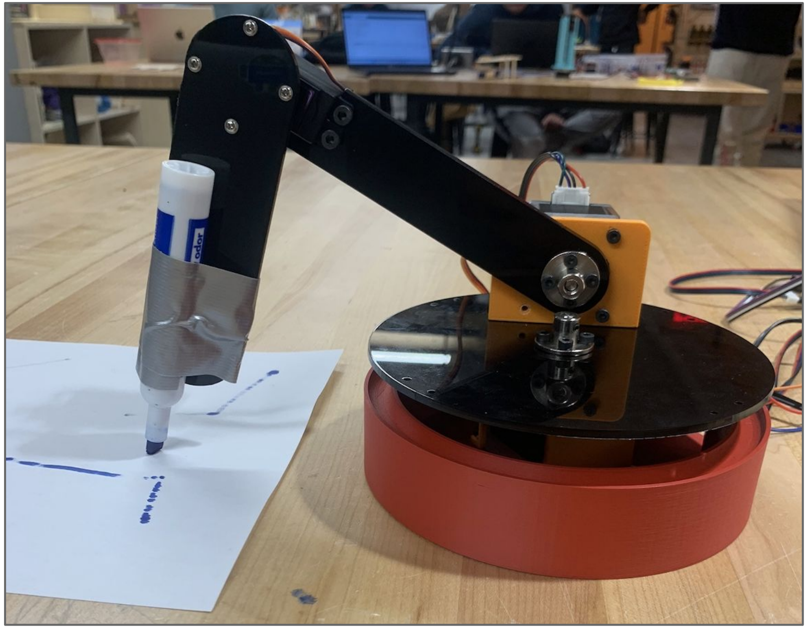
Computational Flexibility:

- Trajectory and inverse kinematics computed remotely
- Few data stored on ESP32



Wenchang Gao

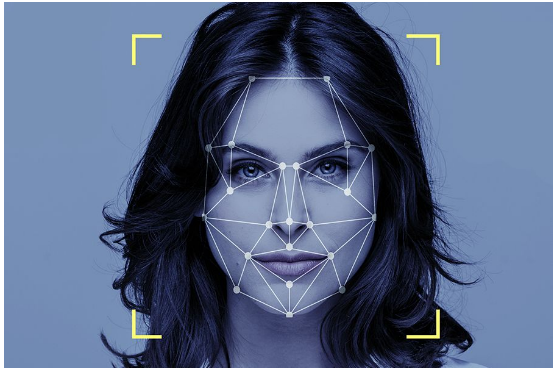
Sales Pitch



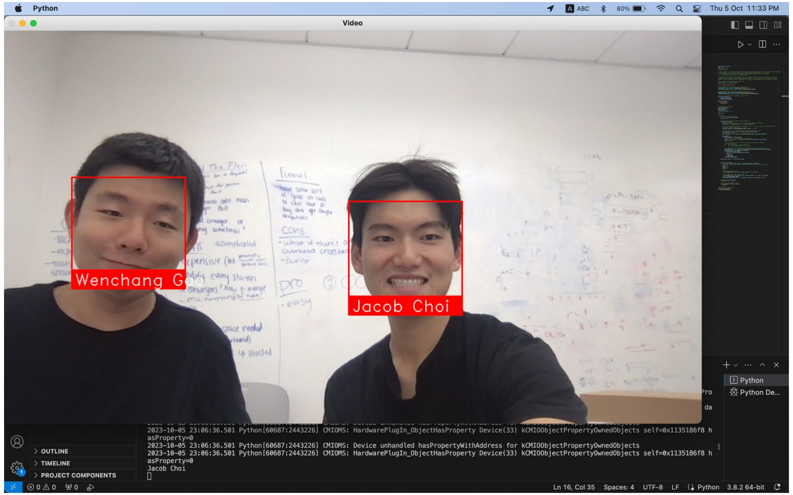
- Lightweight and portable design
- Multiple user interfaces
- Hands-free writing with facial recognition

Jacob Choi

Sales Pitch



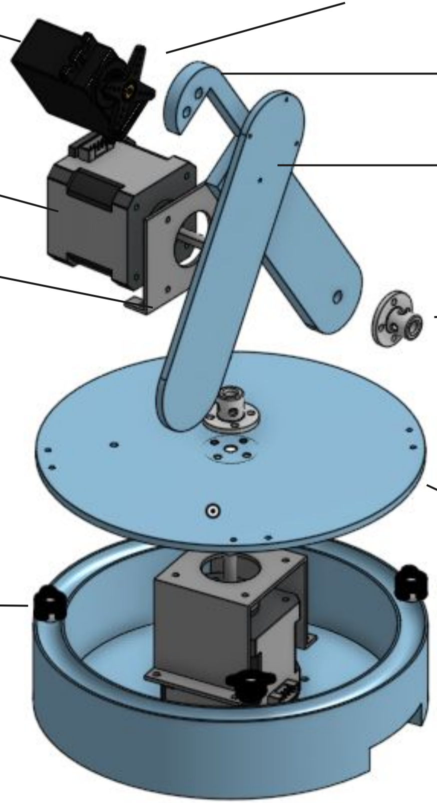
- Remembers your name and your face
- Automatically writes your initials when finds you

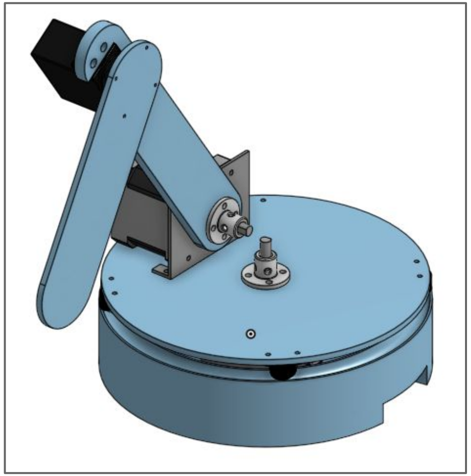


```
2023-10-05 23:06:36.501 Python[60687:2443226] CMIOMS: HardwarePlugIn_ObjectHasProperty Device(33) kCMIOObjectPropertyOwnedObjects self=0x1135186f8 h
asProperty=0
2023-10-05 23:06:36.501 Python[60687:2443226] CMIOMS: Device unhandled hasPropertyWithAddress for kCMIOObjectPropertyOwnedObjects
2023-10-05 23:06:36.501 Python[60687:2443226] CMIOMS: HardwarePlugIn_ObjectHasProperty Device(33) kCMIOObjectPropertyOwnedObjects self=0x1135186f8 h
asProperty=0
Jacob Choi
```

Wenchang Gao

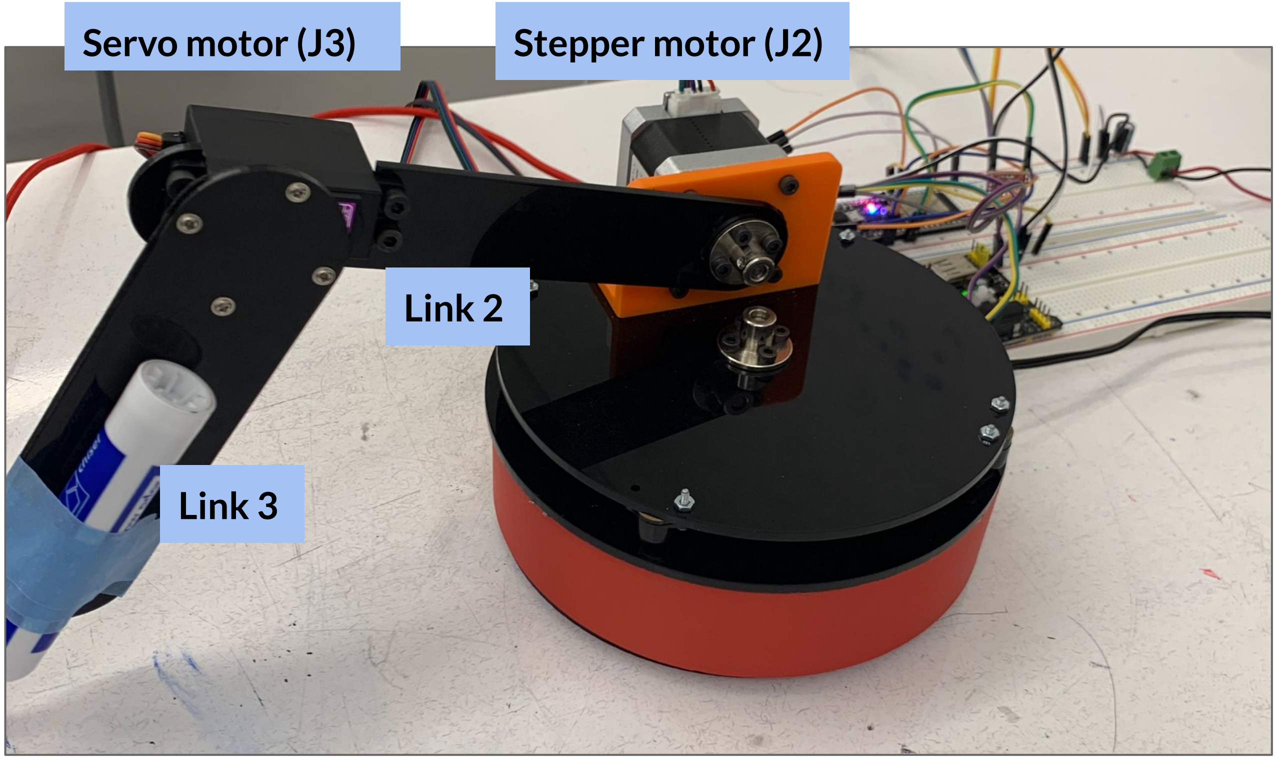
CAD Review

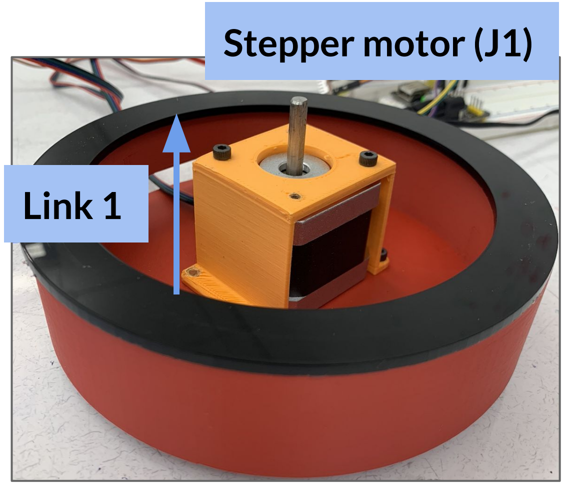




Jacob Choi

Overview





Jacob Choi

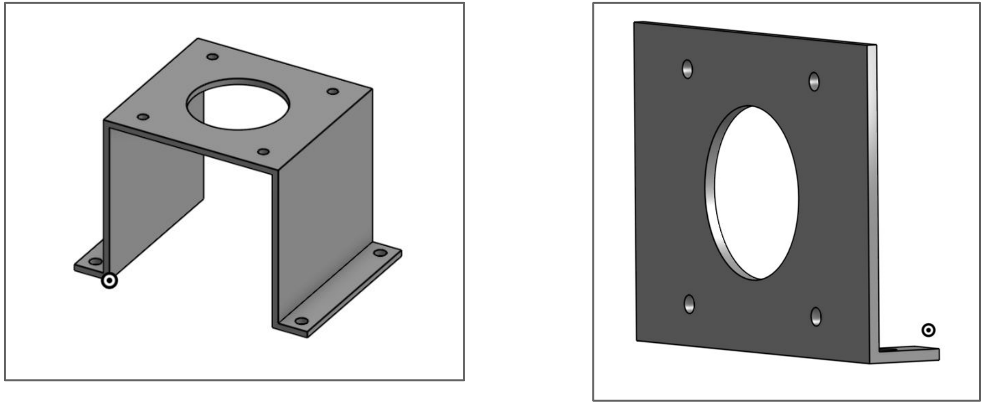
Stretch Goal | Creativity & Innovation

Stretch goal: to automatically write initials based on facial recognition and pre-trained object detection models.

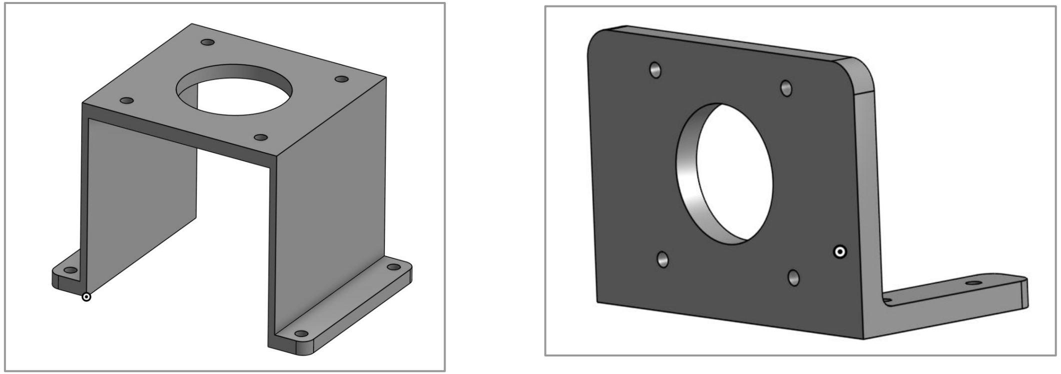
Demo video:

Jacob Choi

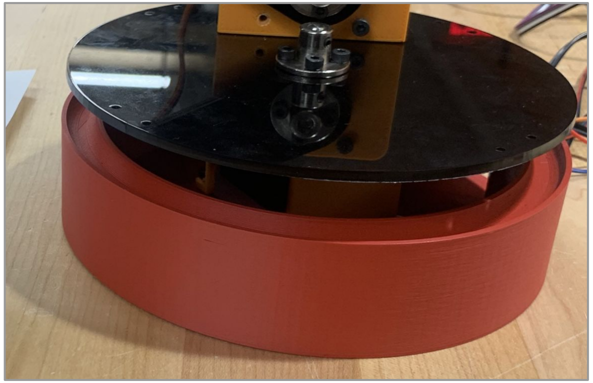
Mechanical Updates



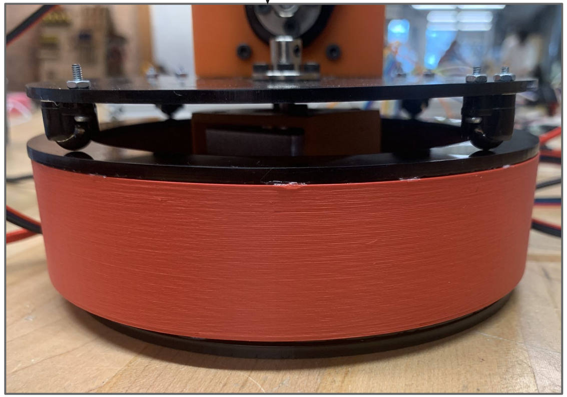
Structurally weak mounts



Thicker mounts



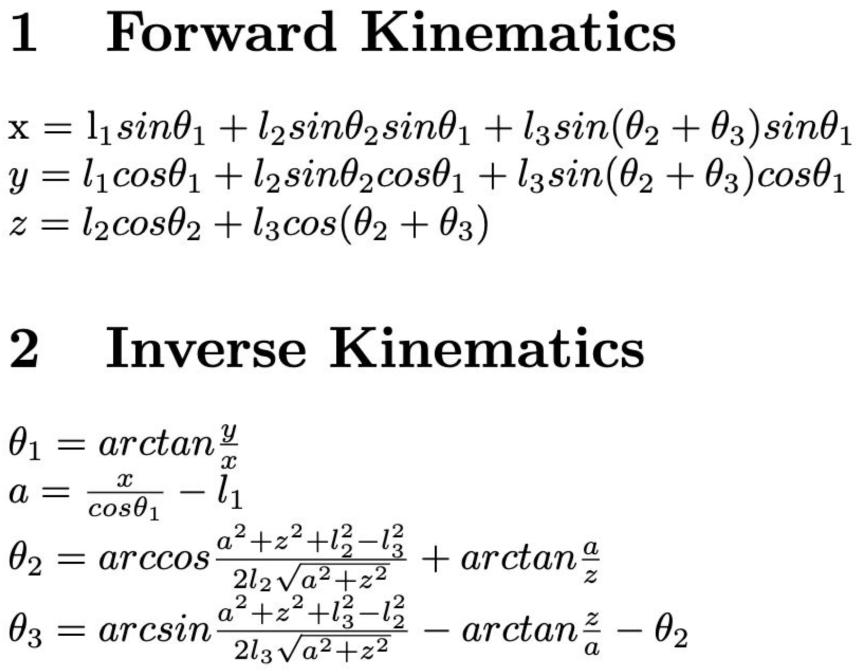
Unstable, unsupported base rotation



Reduced deflection, controlled base rotation

Jacob Choi

Mathematical Reasoning



1 Forward Kinematics

$$\mathrm{x} = \mathrm{l}_1 sin\theta_1 + l_2 sin\theta_2 sin\theta_1 + l_3 sin(\theta_2 + \theta_3) sin\theta_1$$
$$y = l_1 cos\theta_1 + l_2 sin\theta_2 cos\theta_1 + l_3 sin(\theta_2 + \theta_3) cos\theta_1$$
$$z = l_2 cos\theta_2 + l_3 cos(\theta_2 + \theta_3)$$

2 Inverse Kinematics

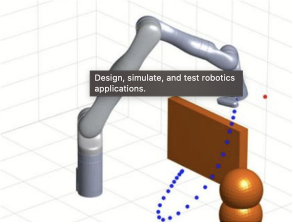
$$\theta_1 = arctan\tfrac{y}{x}$$
$$a = \tfrac{x}{cos\theta_1} - l_1$$
$$\theta_2 = arccos\tfrac{a^2+z^2+l_2^2-l_3^2}{2l_2\sqrt{a^2+z^2}} + arctan\tfrac{a}{z}$$
$$\theta_3 = arcsin\tfrac{a^2+z^2+l_3^2-l_2^2}{2l_3\sqrt{a^2+z^2}} - arctan\tfrac{z}{a} - \theta_2$$

Scientific Reasoning

$$x = x_1 + t(x_2 - x_1) \qquad (Line)$$

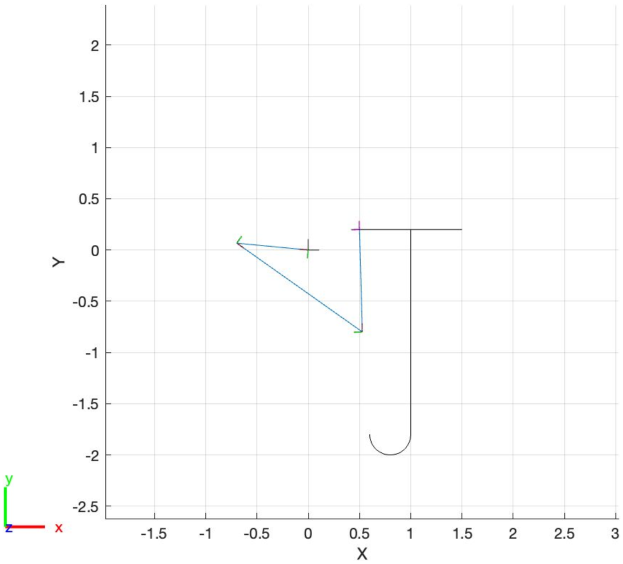
$$x = o + \begin{bmatrix} cos\theta \\ sin\theta \\ z \end{bmatrix} \qquad (Circle)$$

Using parametric equations, we can discretize lines and circles in Cartesian.



Using Robotics System Toolbox in MatLab, we can safely calculate the inverse kinematics.

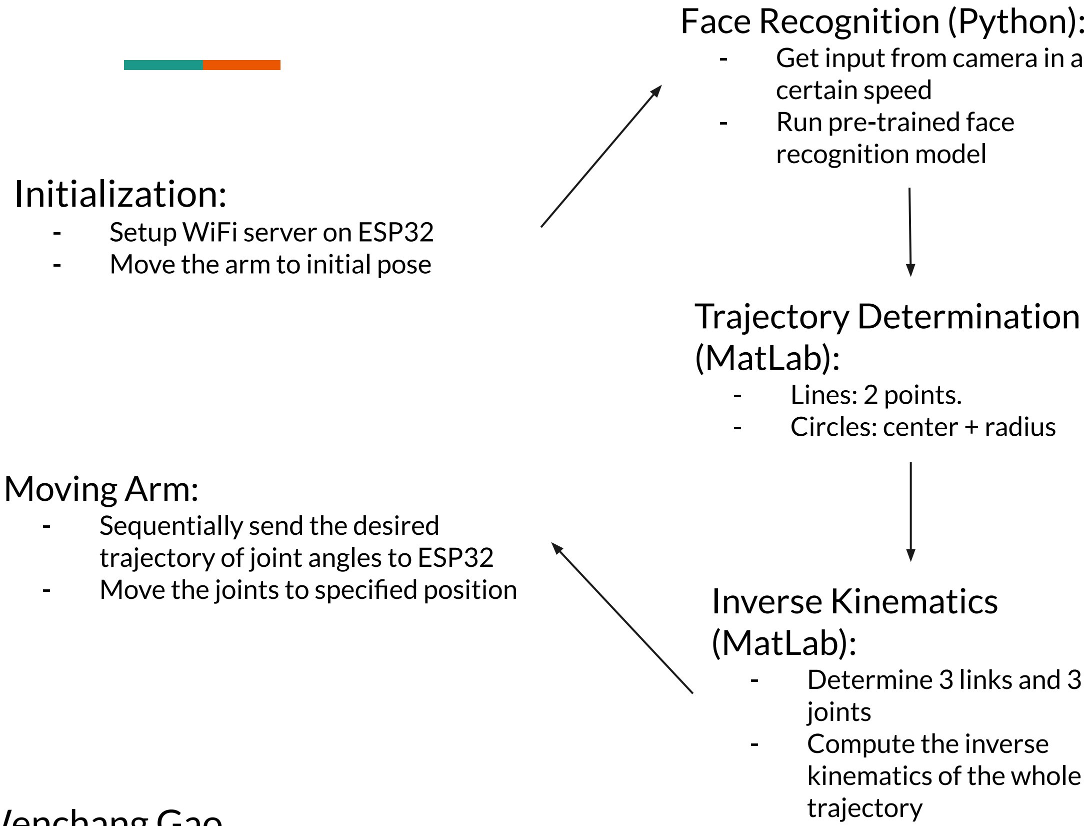
Wenchang Gao



Calculating the inverse kinematics trajectory of letter ‘J’ in matlab.

This letter is combined with 3 lines and 1 half circle

Code Flow



Initialization:

- Setup WiFi server on ESP32
- Move the arm to initial pose

Face Recognition (Python):

- Get input from camera in a certain speed
- Run pre-trained face recognition model

Trajectory Determination (MatLab):

- Lines: 2 points.
- Circles: center + radius

Inverse Kinematics (MatLab):

- Determine 3 links and 3 joints
- Compute the inverse kinematics of the whole trajectory

Moving Arm:

- Sequentially send the desired trajectory of joint angles to ESP32
- Move the joints to specified position

Wenchang Gao

Thank you for listening!

Any questions?