Bot-Ross

Robotics Project

Lecturer:
Professor Talebi

Team Members:
Pouria Alimoradpor - Kasra Hassani – Vaniya Malakzahedi
Mojtaba Dehghani Firouzabadi - Seyed Ahmad AmirAhmadi BabaHeydari

Our Team

- Electronics and Mechanics
- Programming and UX Design
- Documentation
- Coordinations

Our Team

Electronics and Mechanics

Kasra Hassani

Programming and UX Design

Pouria Alimoradpor

Documentation

Vaniya Malakzahedi

Seyed Ahmad AmirAhmadi BabaHeydari

Mojtaba Dehghani Firouzabadi

Coordinations

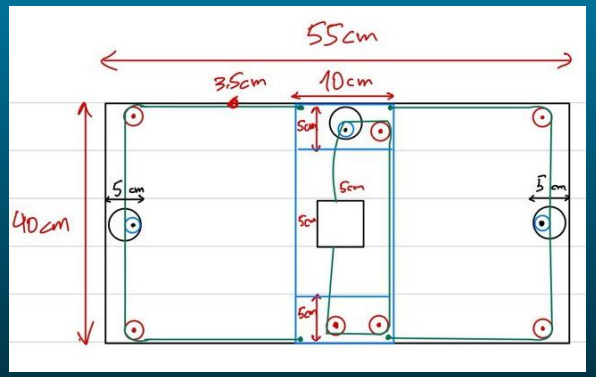
Vaniya Malakzahedi

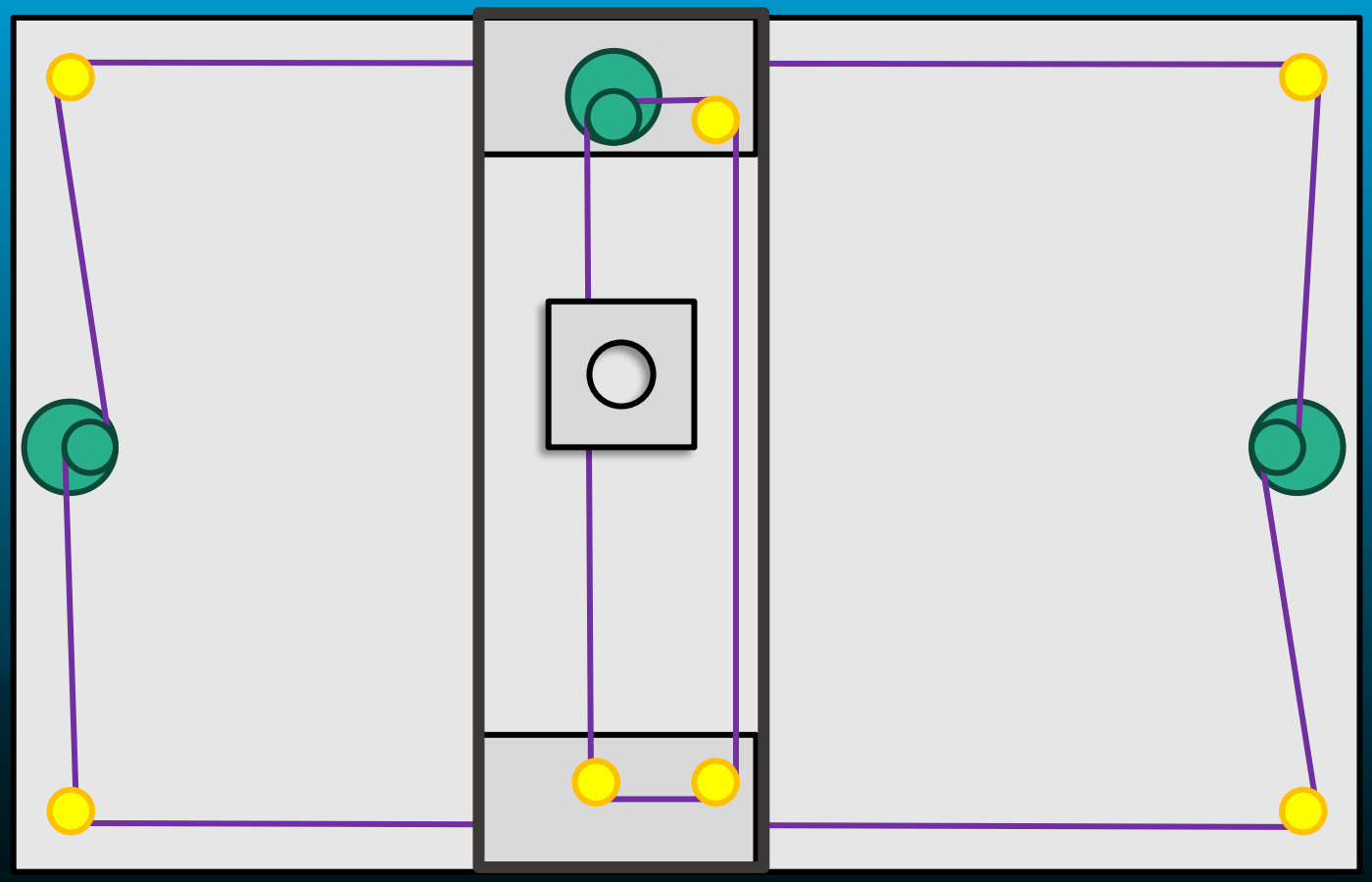
Mechanic & Electronics

- Full View
- Servo Marker (End-Effector)
- Up/Down (End-Effector)
- Left/Right Platform
- Servo Motor
- Stepper Motor
- Driver Motor

Mechanic & Electronics

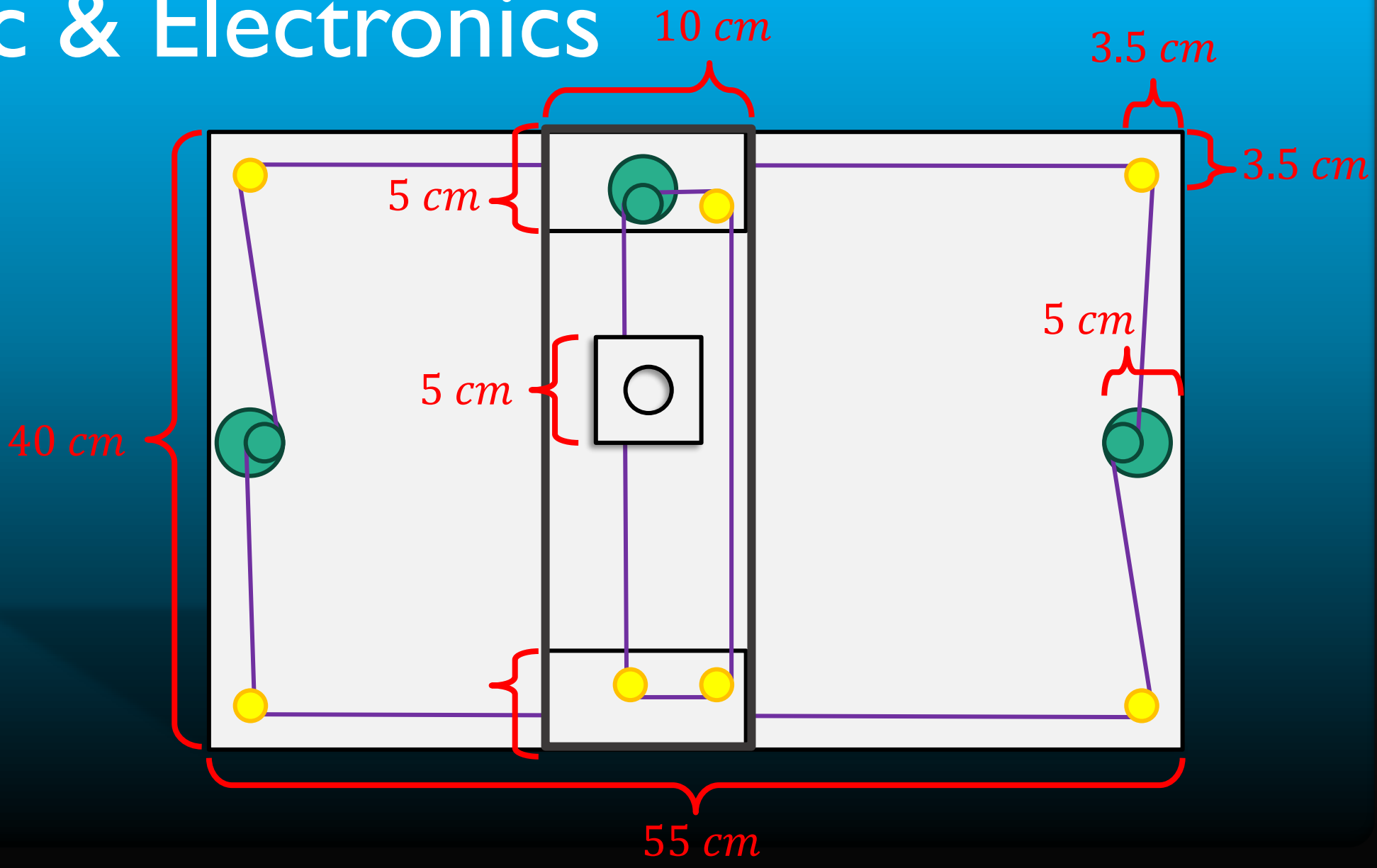
Full View





Mechanic & Electronics

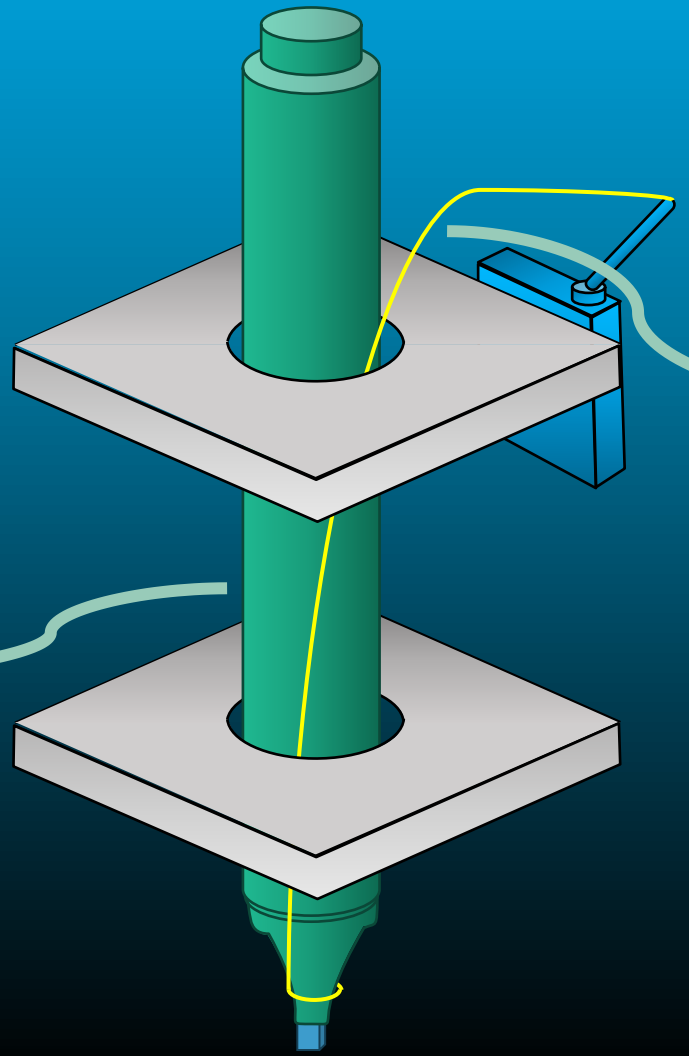
Full View



Mechanic & Electronics

Servo Marker (End-Effector)

Full View

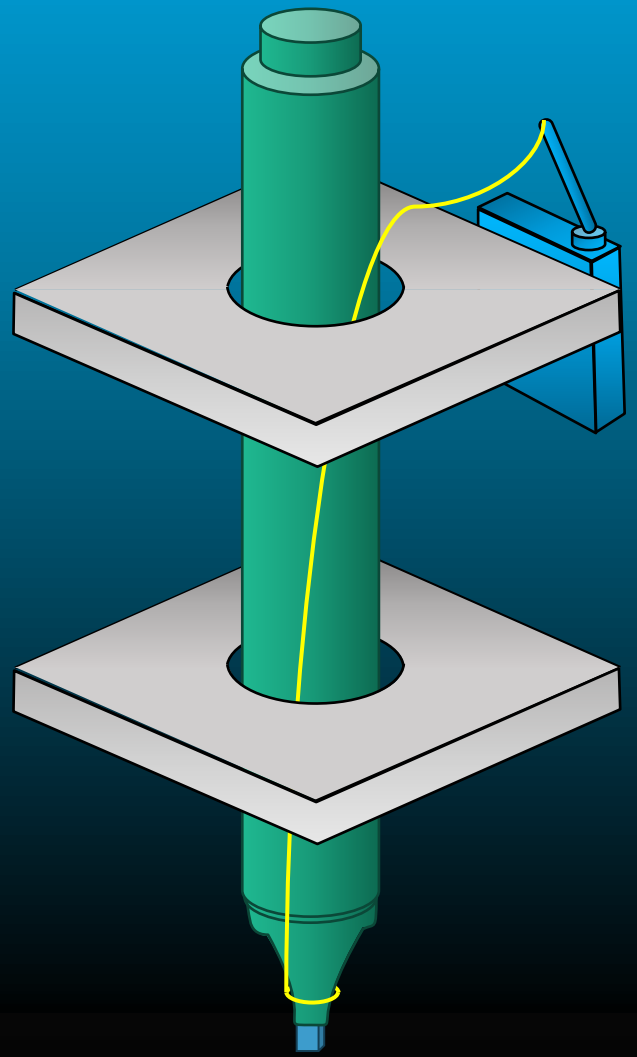


$$D = 2cm \Rightarrow P = \pi \times 2 \approx 6.28 \approx 8cm$$

$$len = 10cm + 8cm = 18cm$$

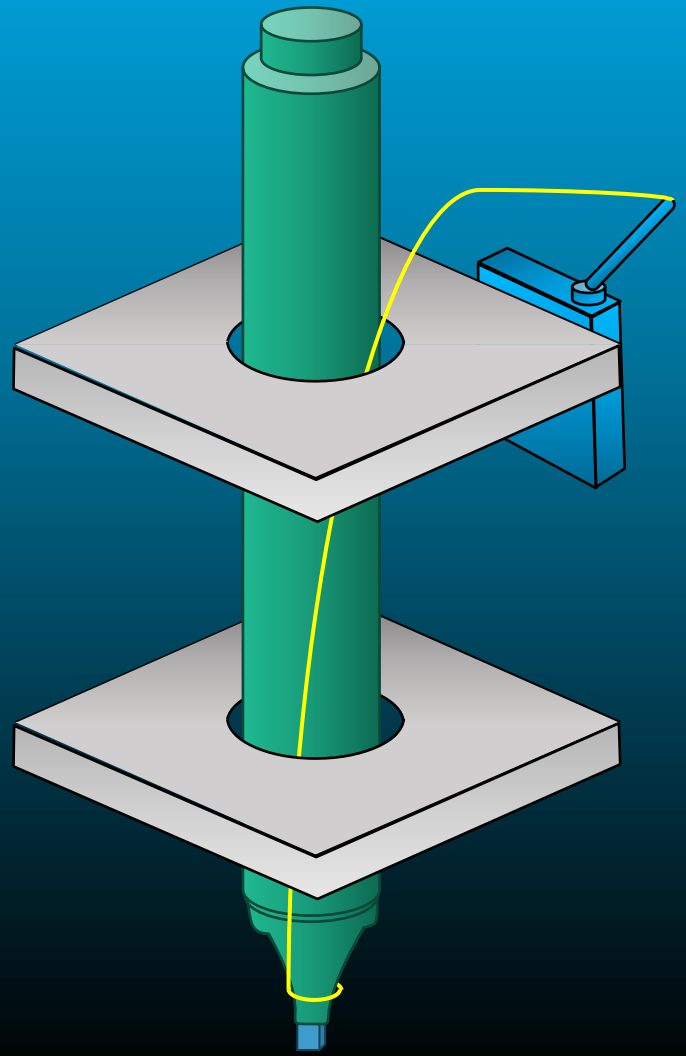
Mechanic & Electronics

Servo Marker (End-Effector)



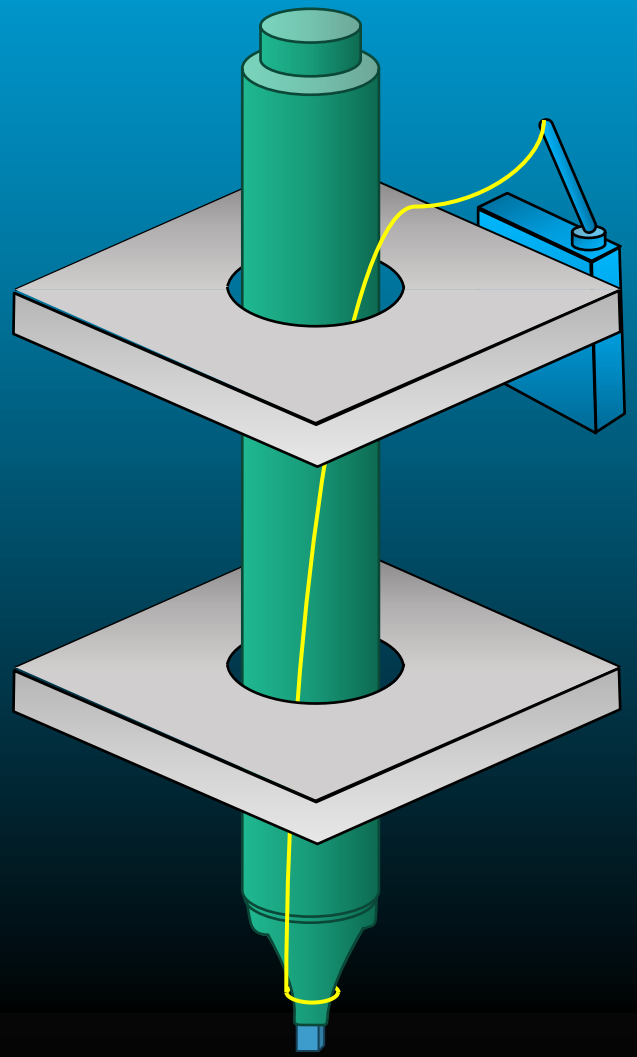
Mechanic & Electronics

Servo Marker (End-Effector)



Mechanic & Electronics

Servo Marker (End-Effector)

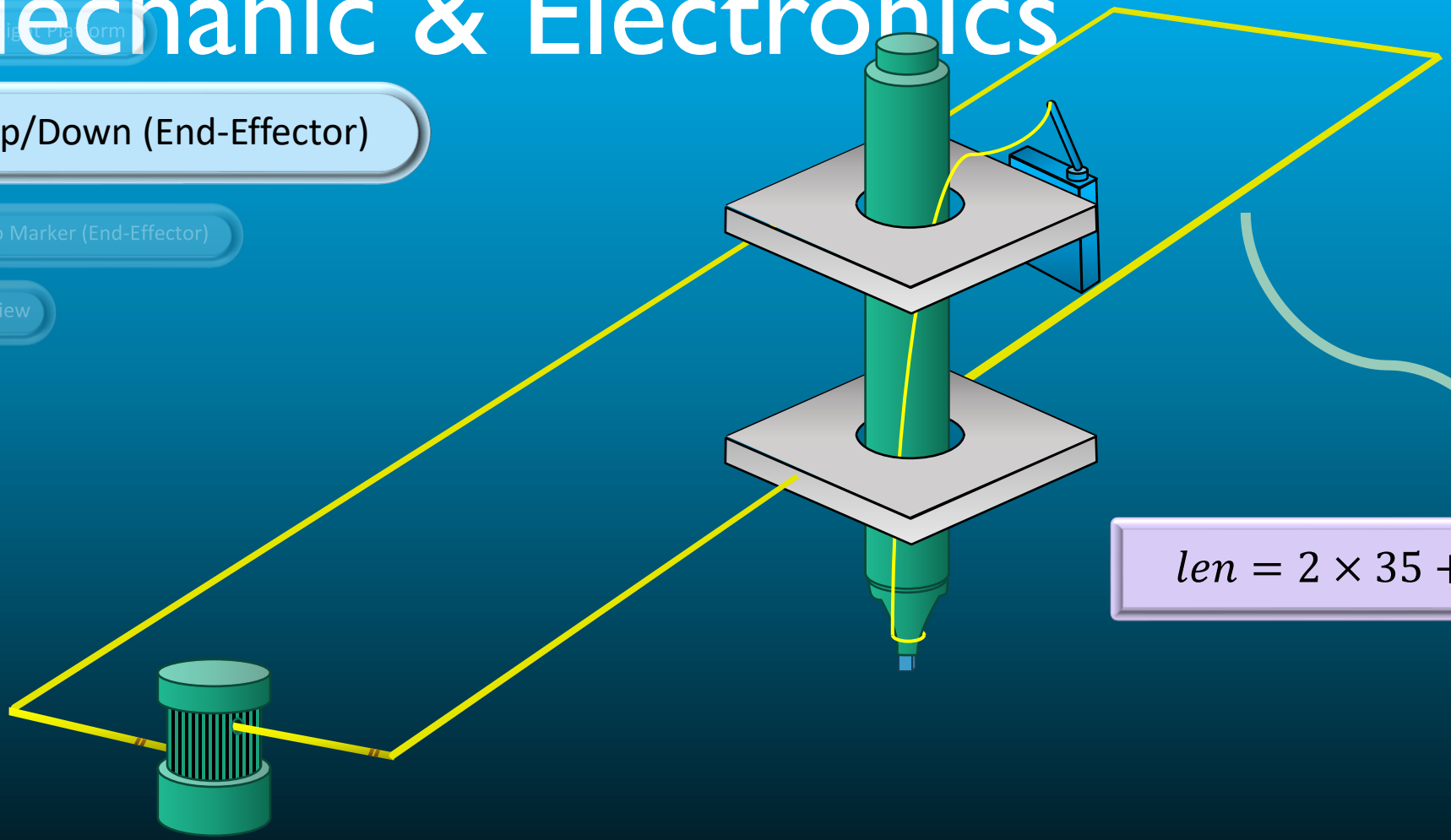


Mechanic & Electronics

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



$$len = 2 \times 35 + 2 \times 5 + 20 = 100cm$$

Servo Motor

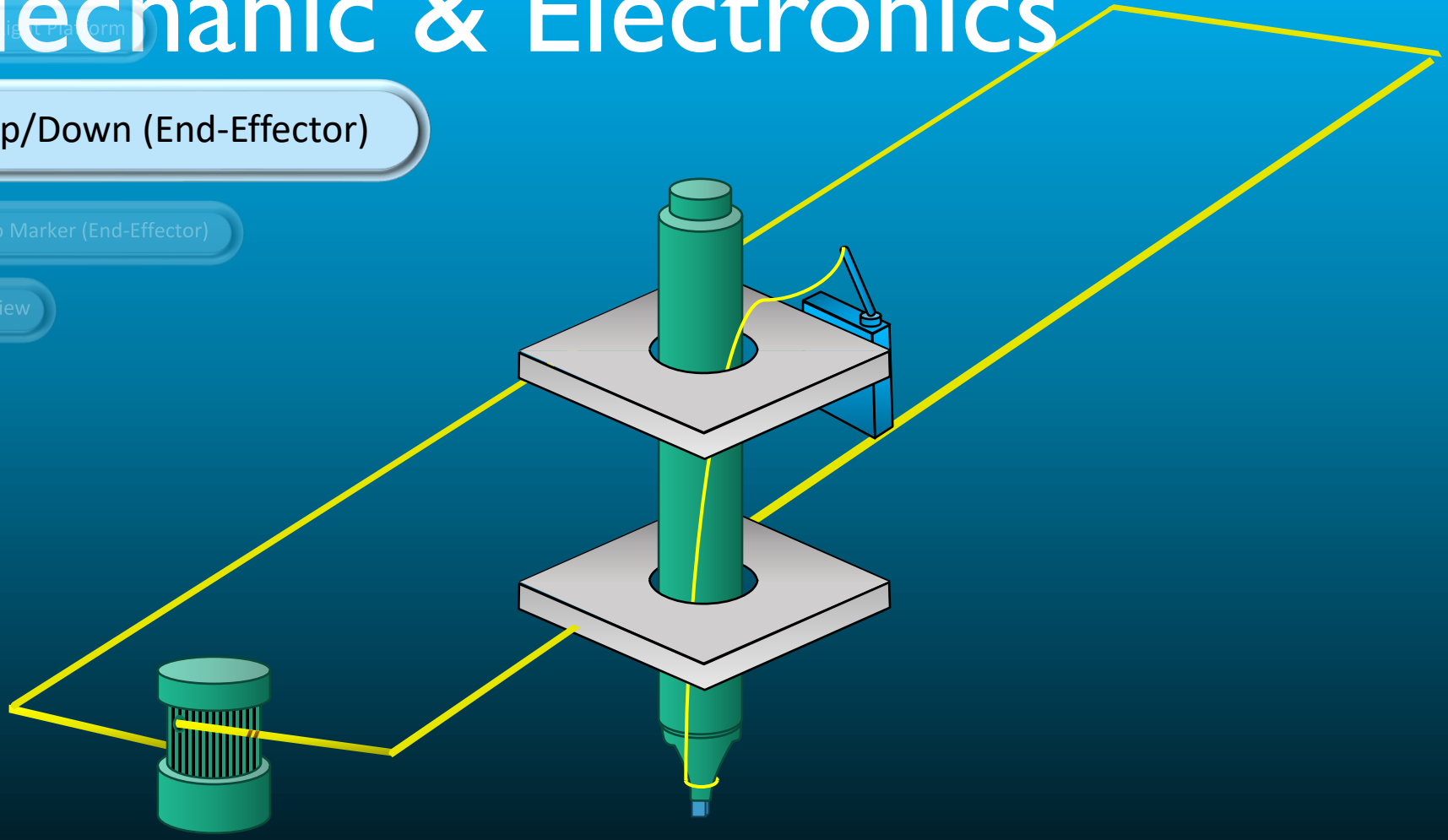
Left/Right Platform

Mechanic & Electronics

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Servo Motor

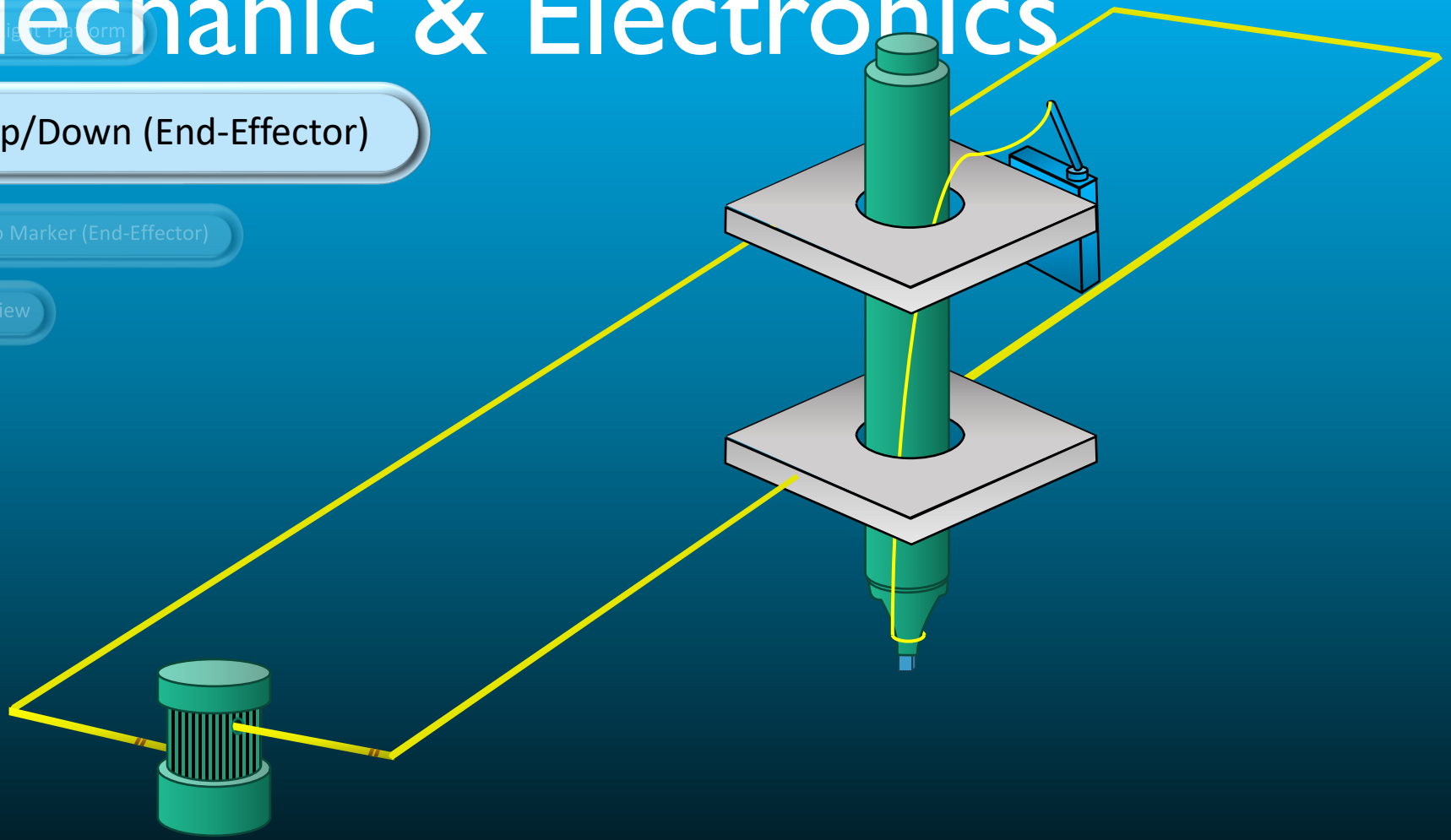
Left/Right Platform

Mechanic & Electronics

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Servo Motor

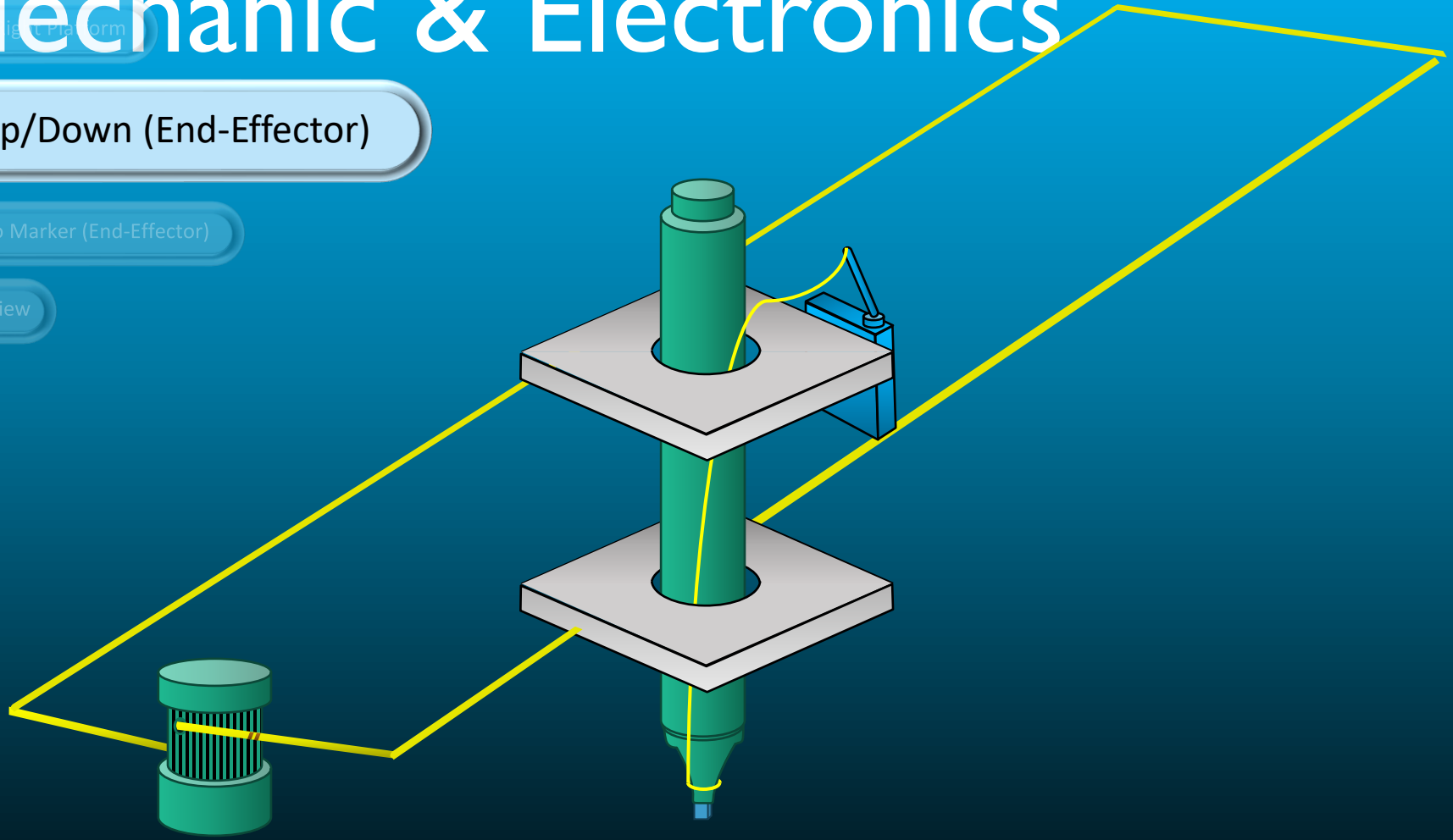
Left/Right Platform

Mechanic & Electronics

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Servo Motor

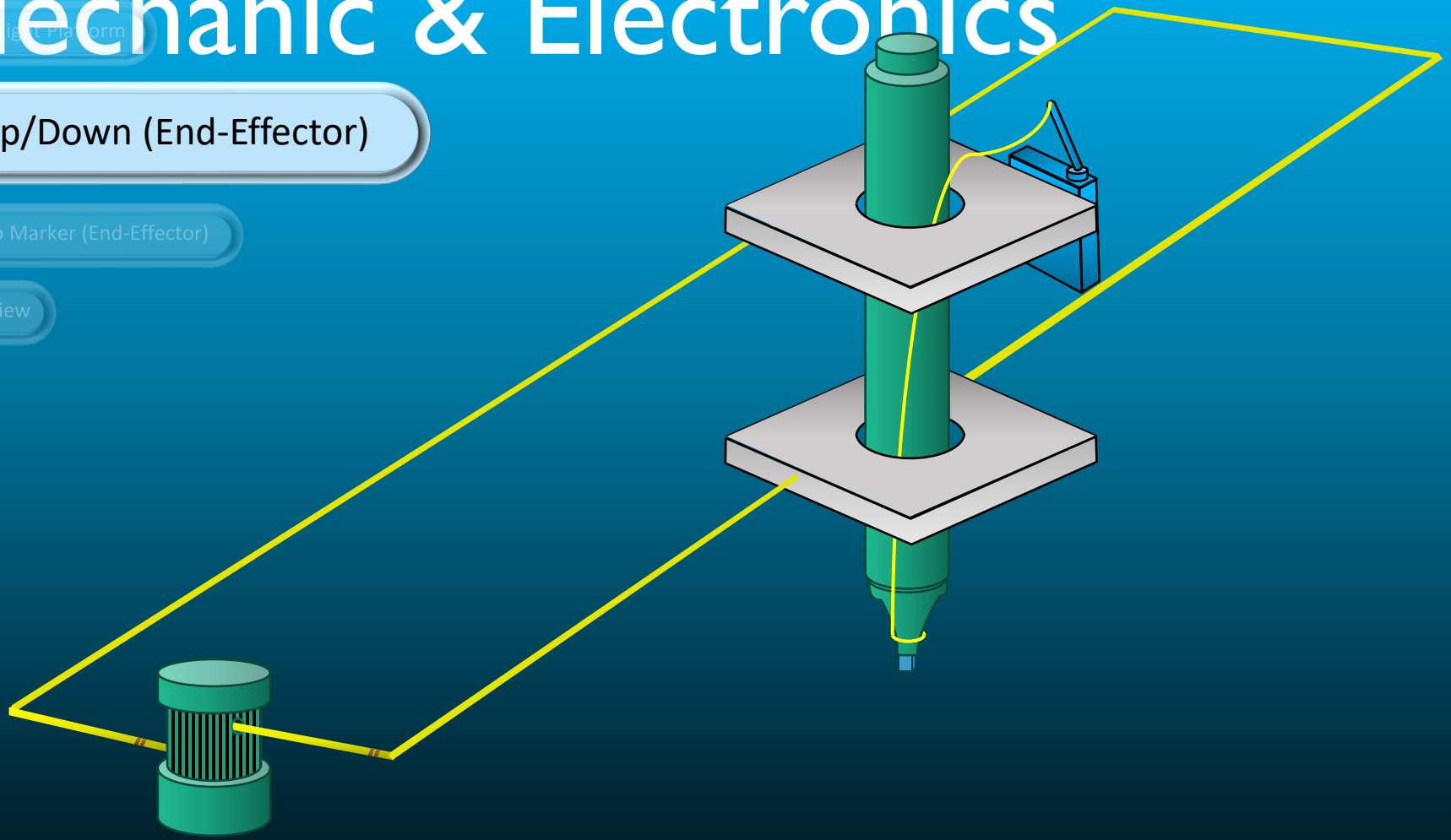
Left/Right Platform

Mechanic & Electronics

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Servo Motor

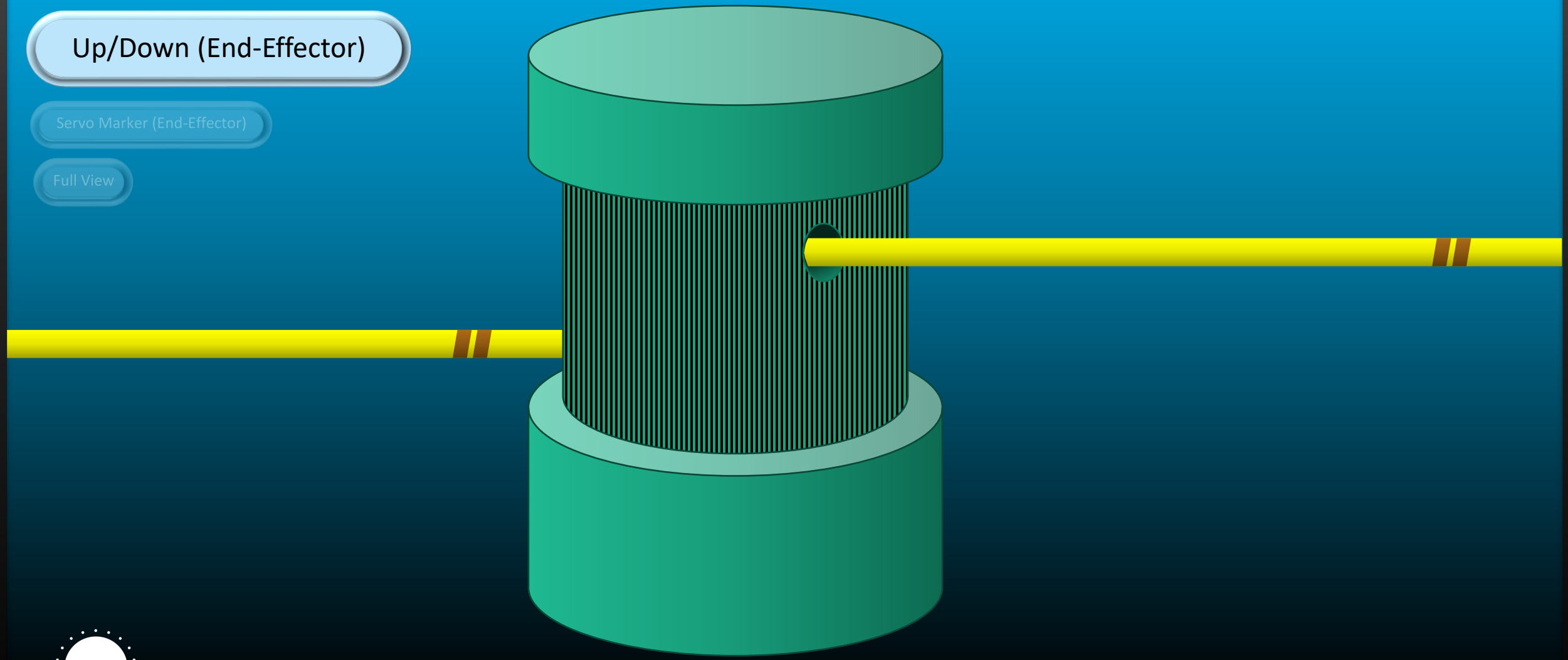
Left/Right Platform

Mechanic & Electronics

Up/Down (End-Effector)

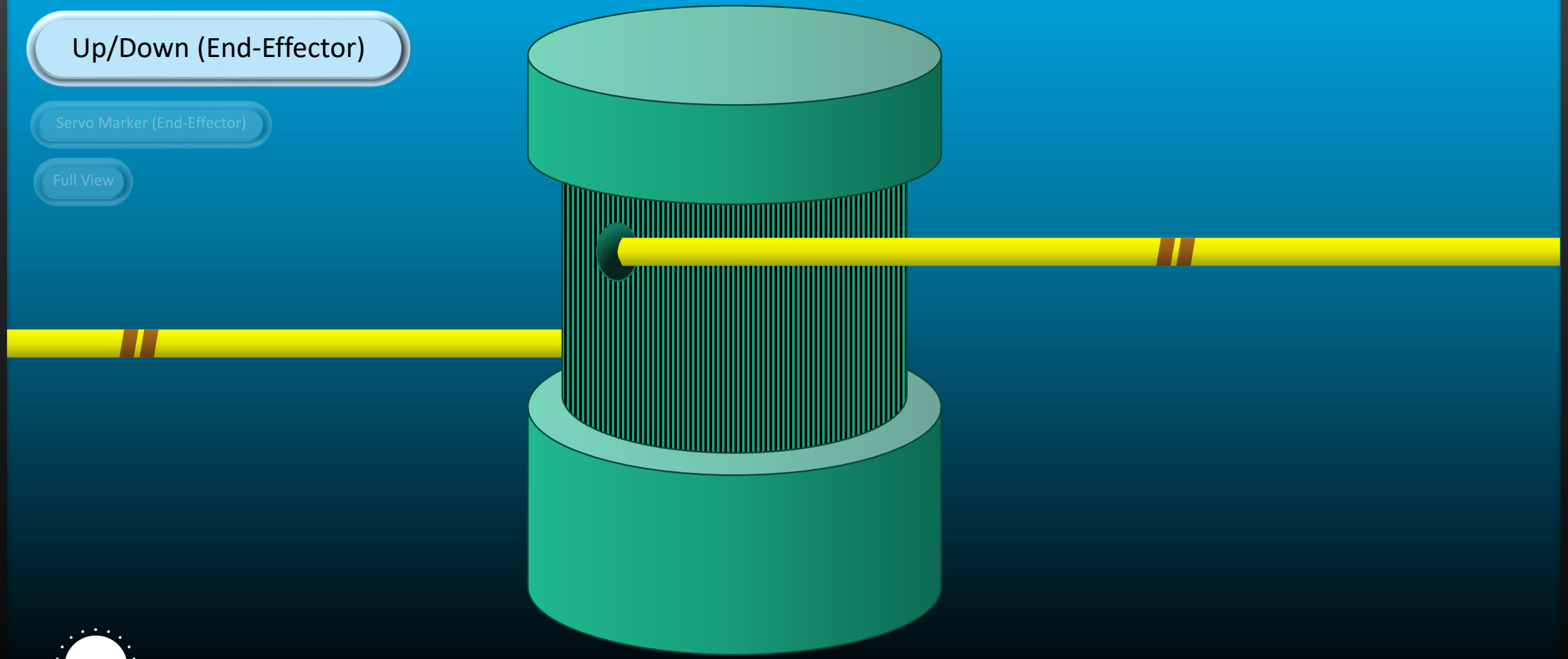
Servo Marker (End-Effector)

Full View



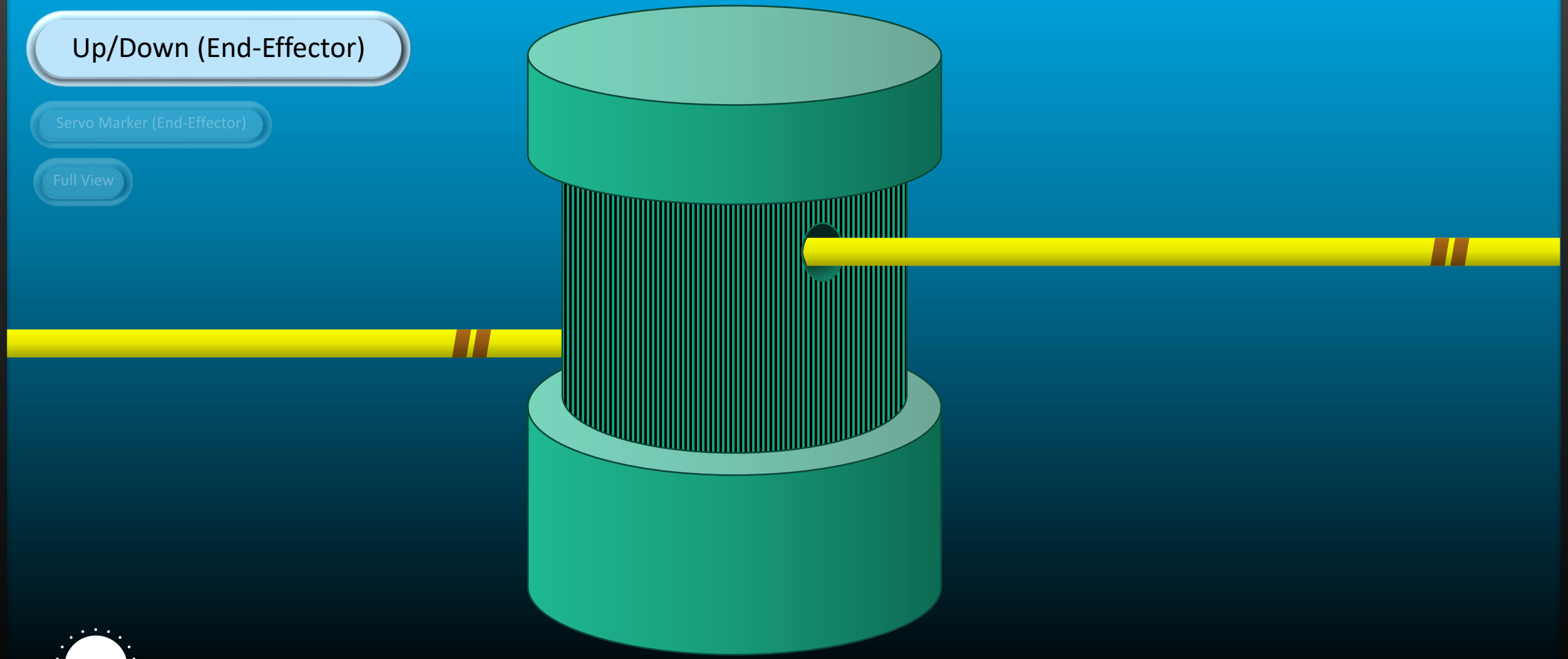
Mechanic & Electronics

Up/Down (End-Effector)



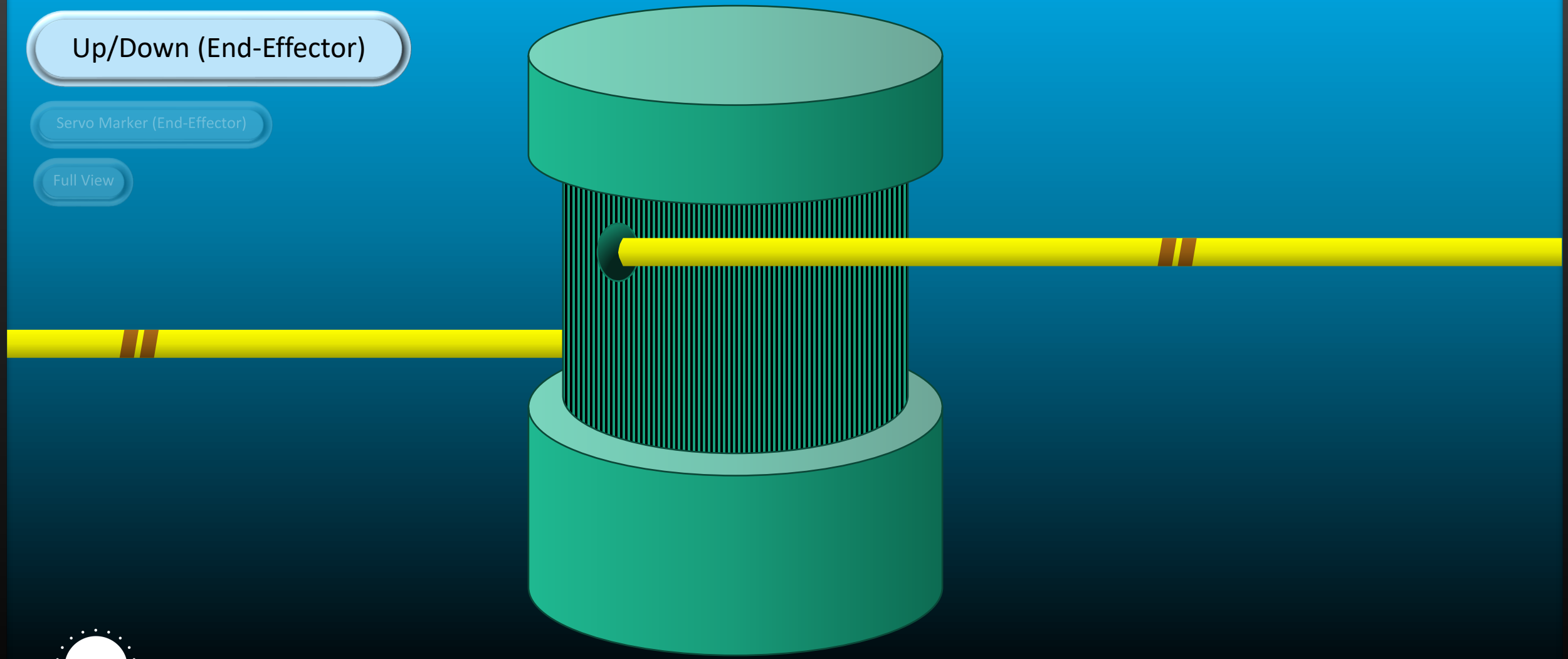
Mechanic & Electronics

Up/Down (End-Effector)



Mechanic & Electronics

Up/Down (End-Effector)



Mechanic & Electronics

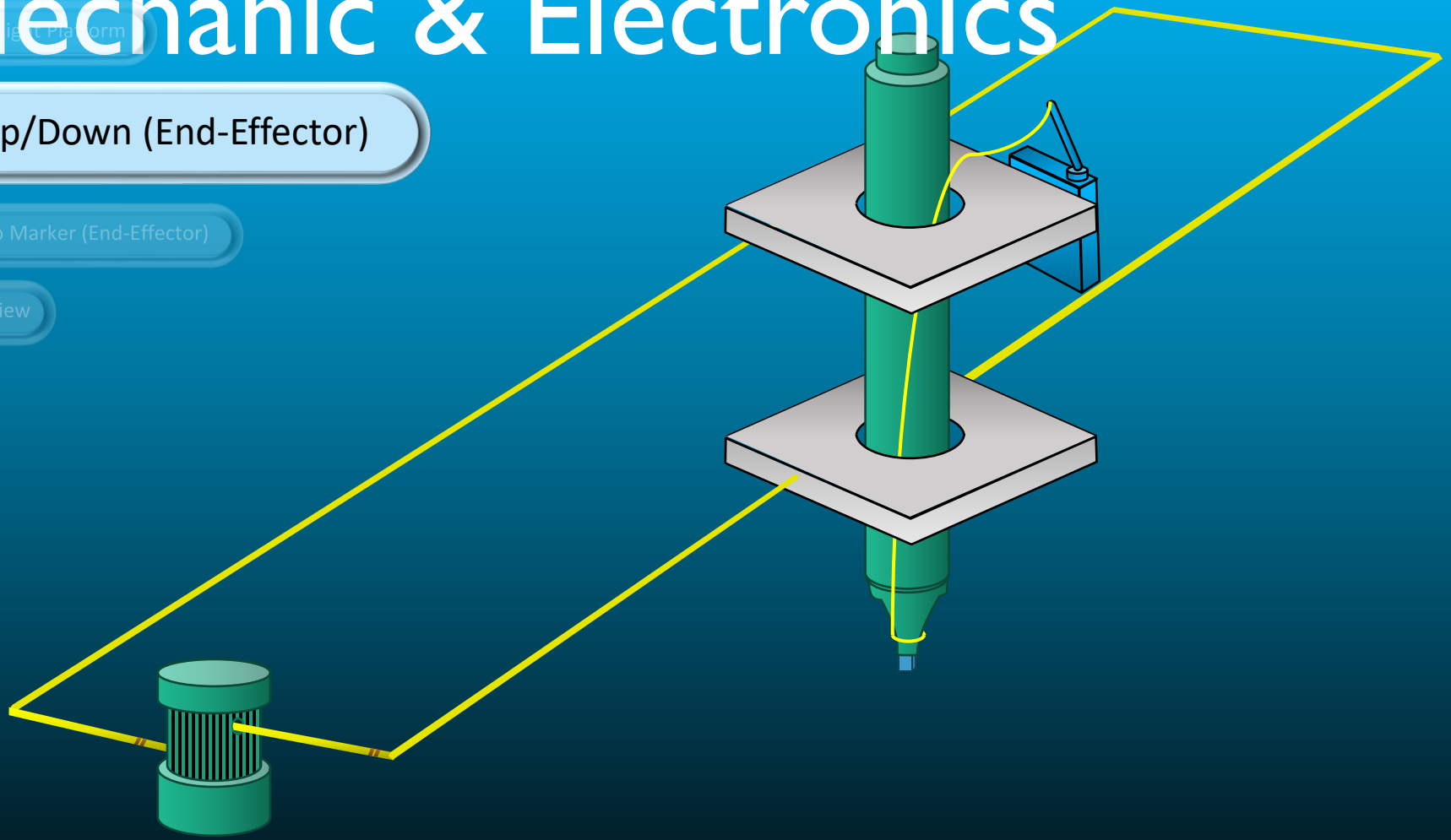
Servo Motor

Left/Right Platform

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Servo Motor

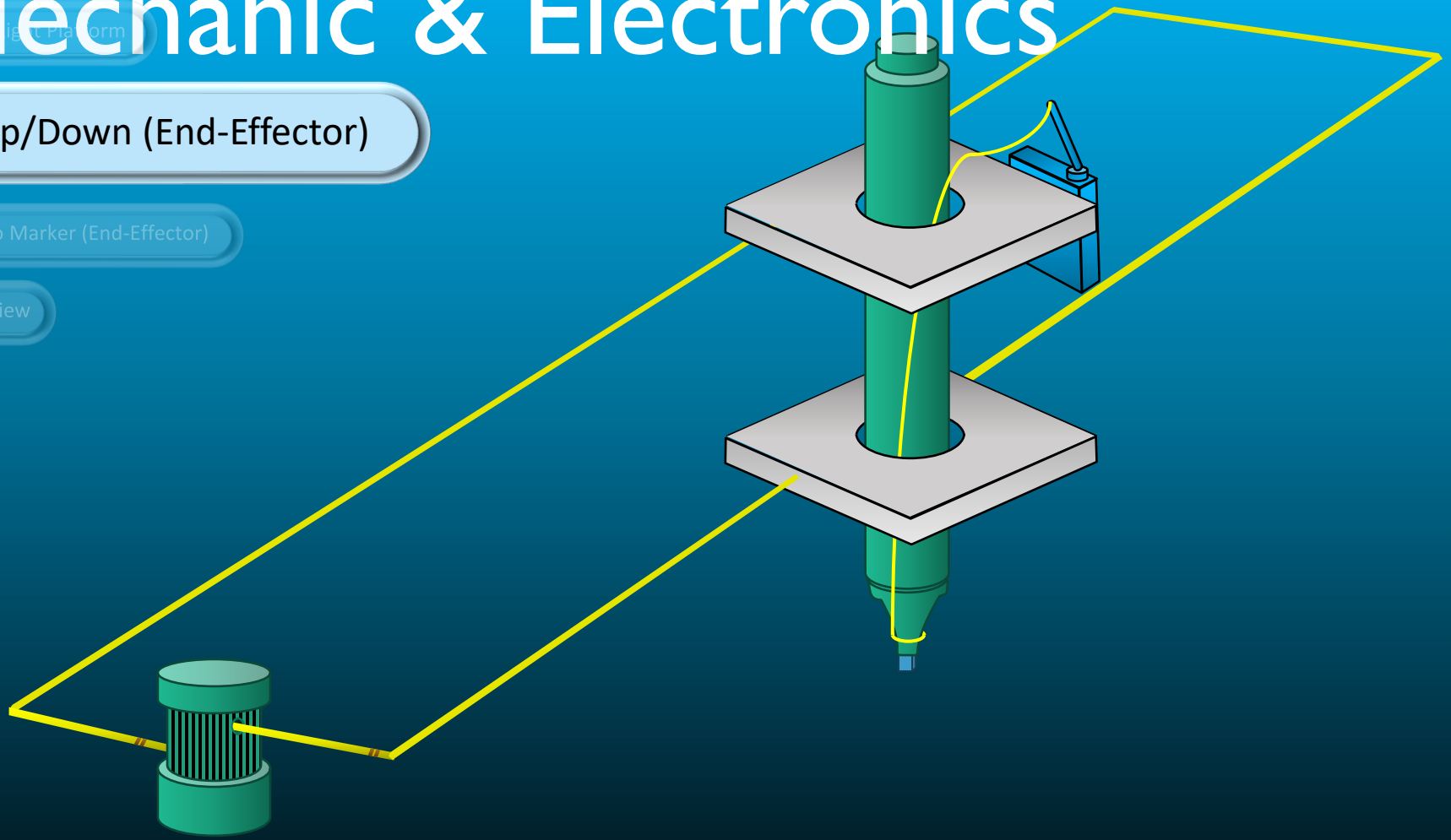
Left/Right Platform

Mechanic & Electronics

Up/Down (End-Effector)

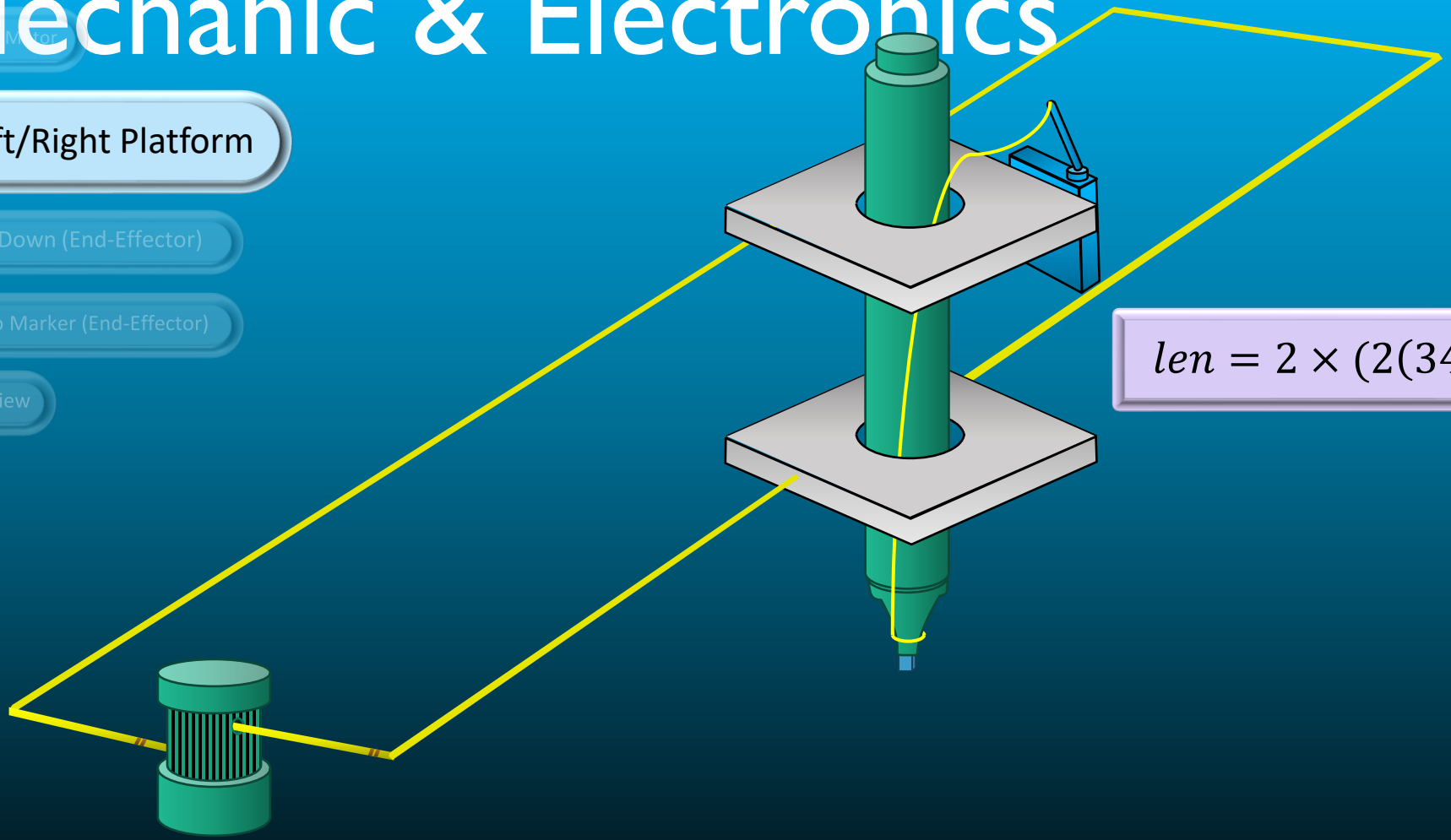
Servo Marker (End-Effector)

Full View



Mechanic & Electronics

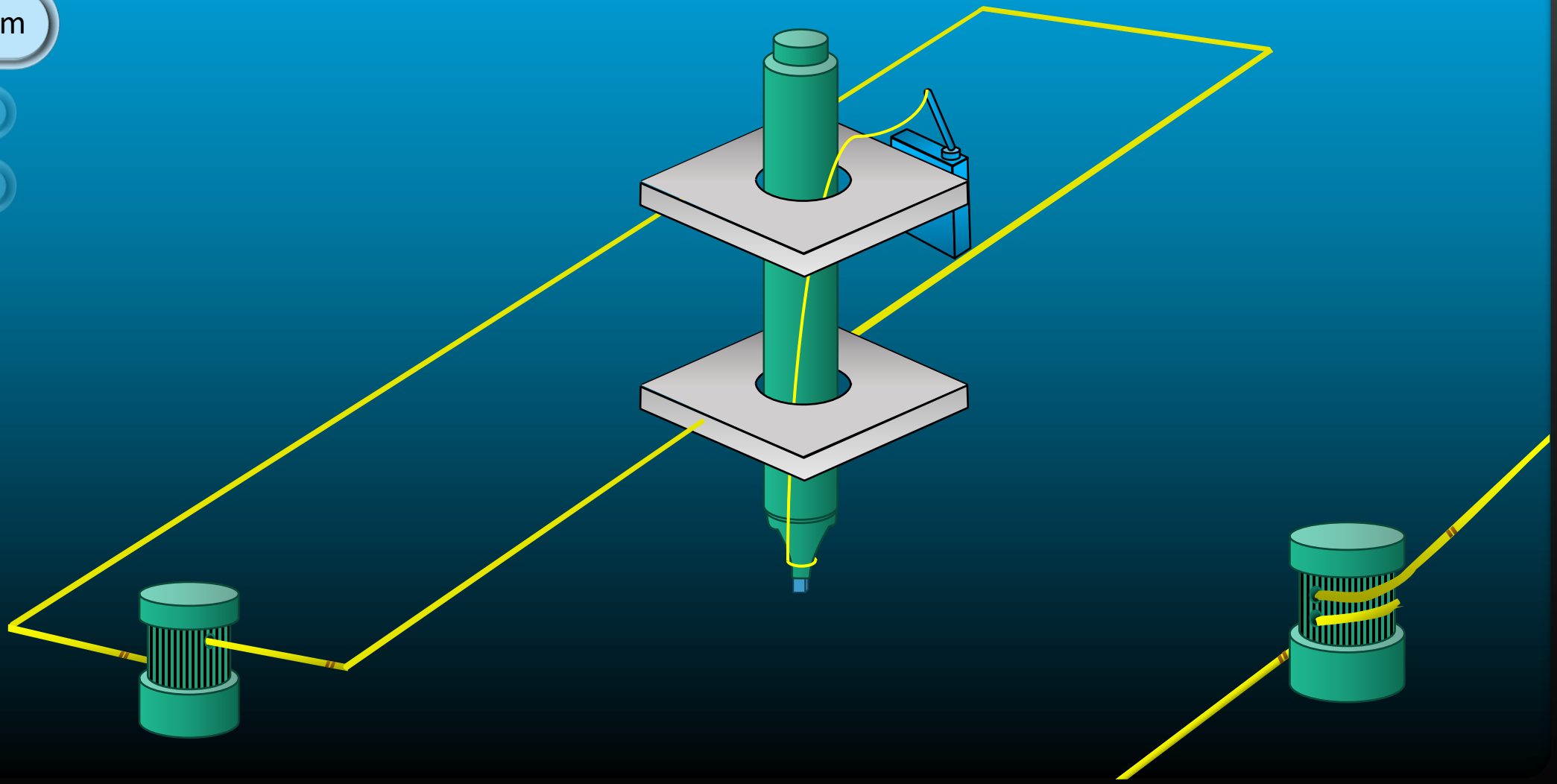
- Stepper Motor
- Servo Motor (...)
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



$$len = 2 \times (2(34 + 16) + 20) = 240cm$$

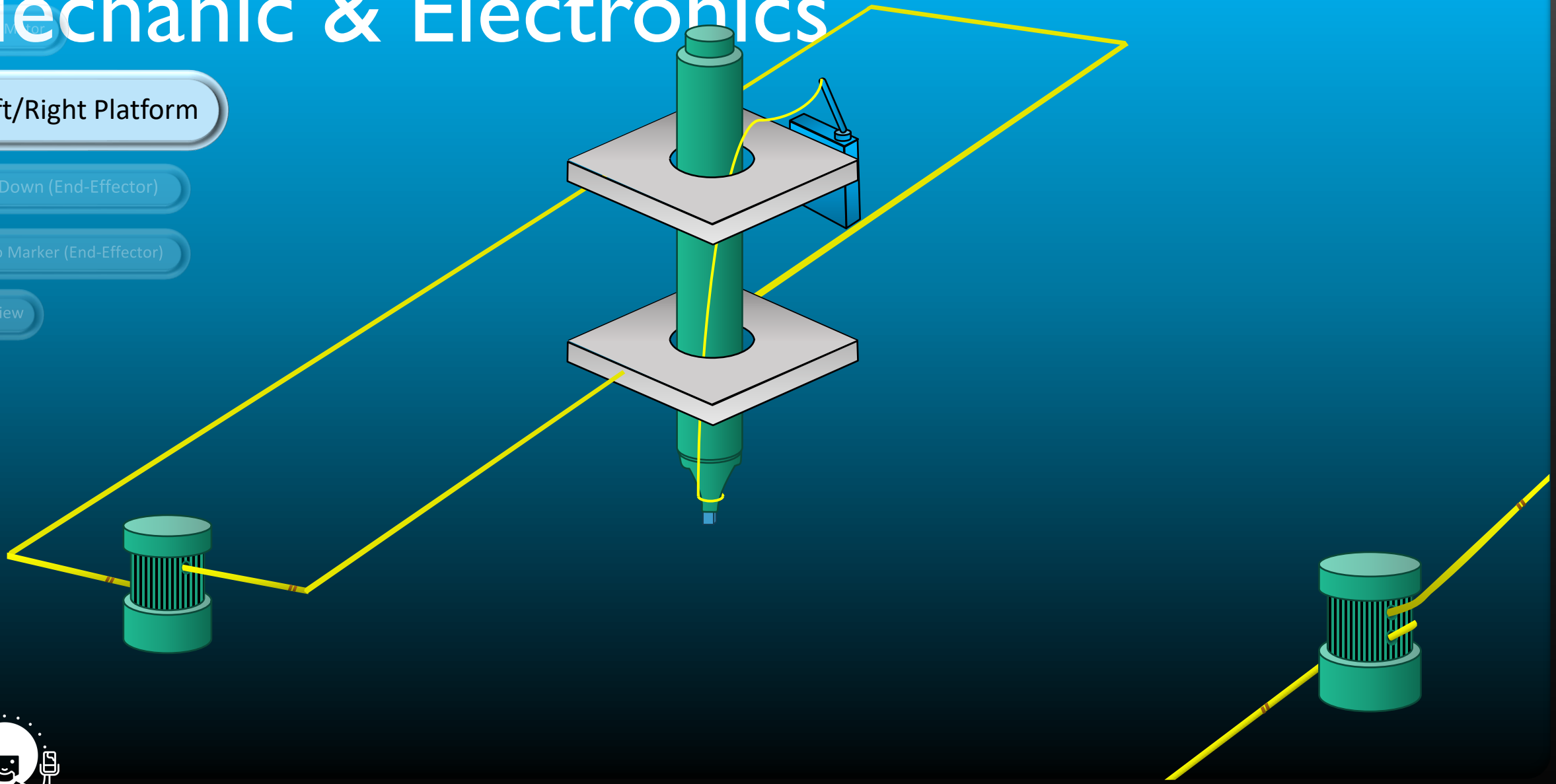
Mechanic & Electronics

- Stepper Motor
- Servo Motor (...)
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



Mechanic & Electronics

- Stepper Motor
- Servo Motor (...)
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



Stepper Motor

Servo Motor (...)

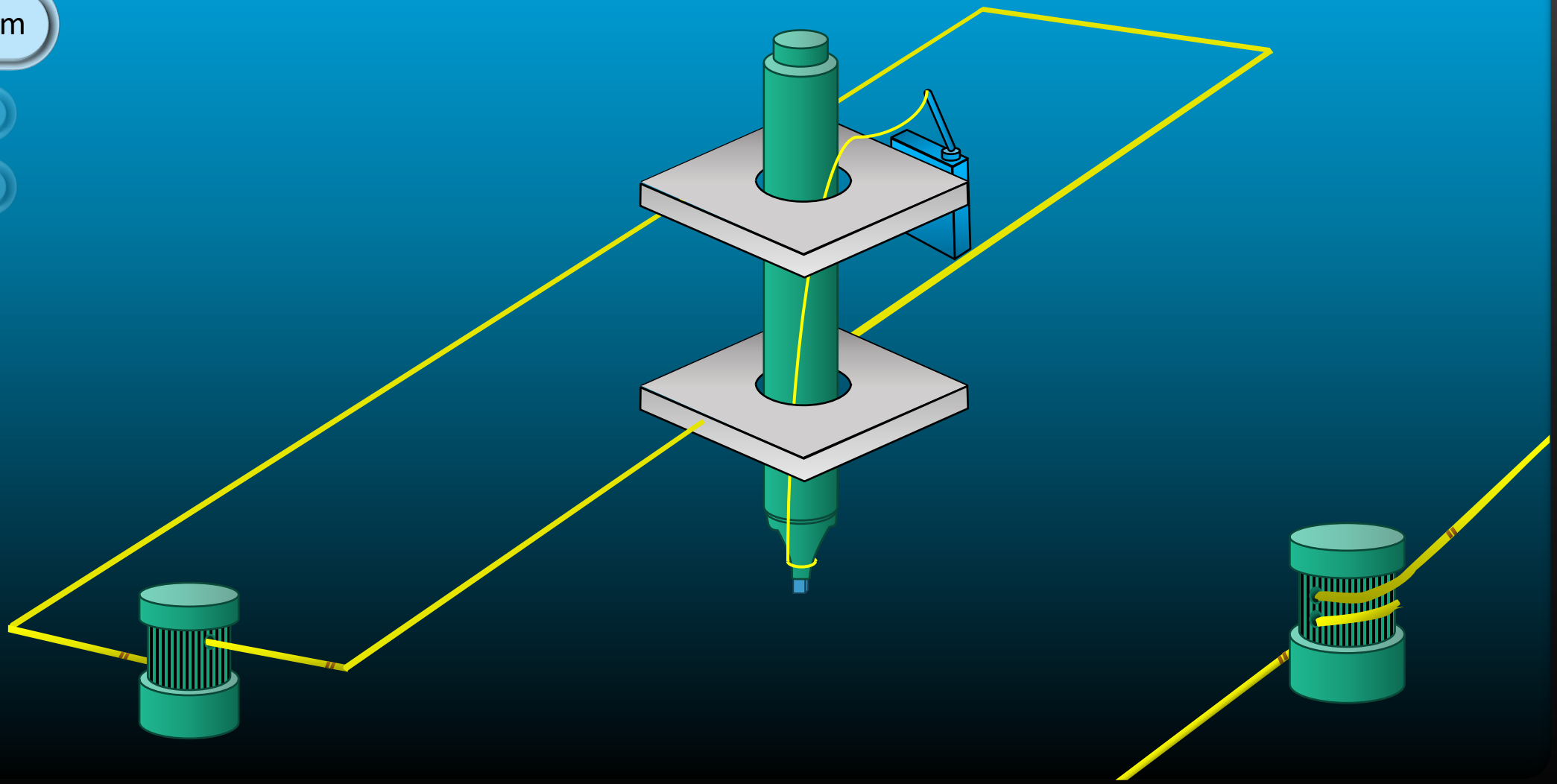
Mechanic & Electronics

Left/Right Platform

Up/Down (End-Effector)

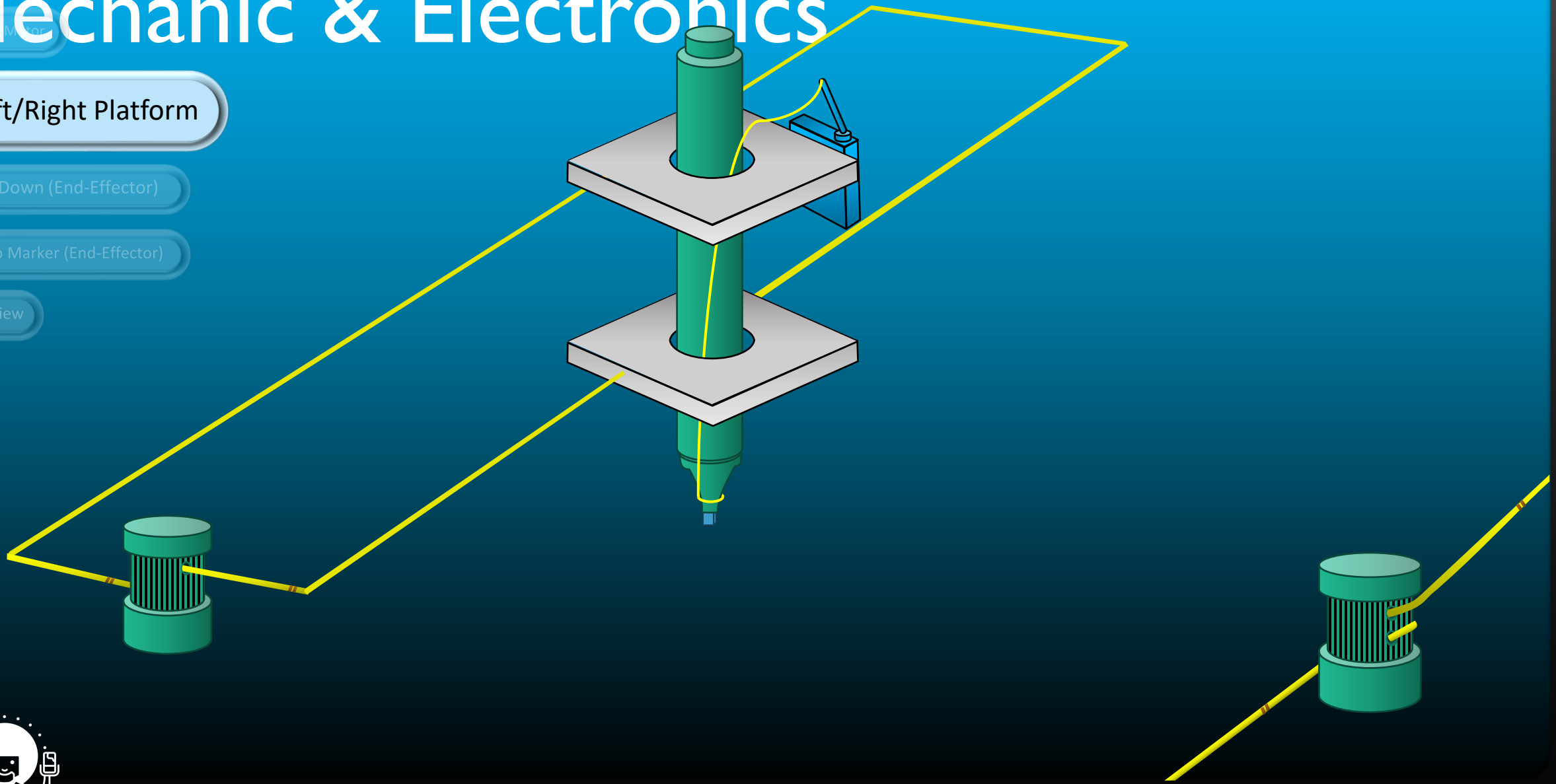
Servo Marker (End-Effector)

Full View



Mechanic & Electronics

- Stepper Motor
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



Mechanic & Electronics

Stepper Motor

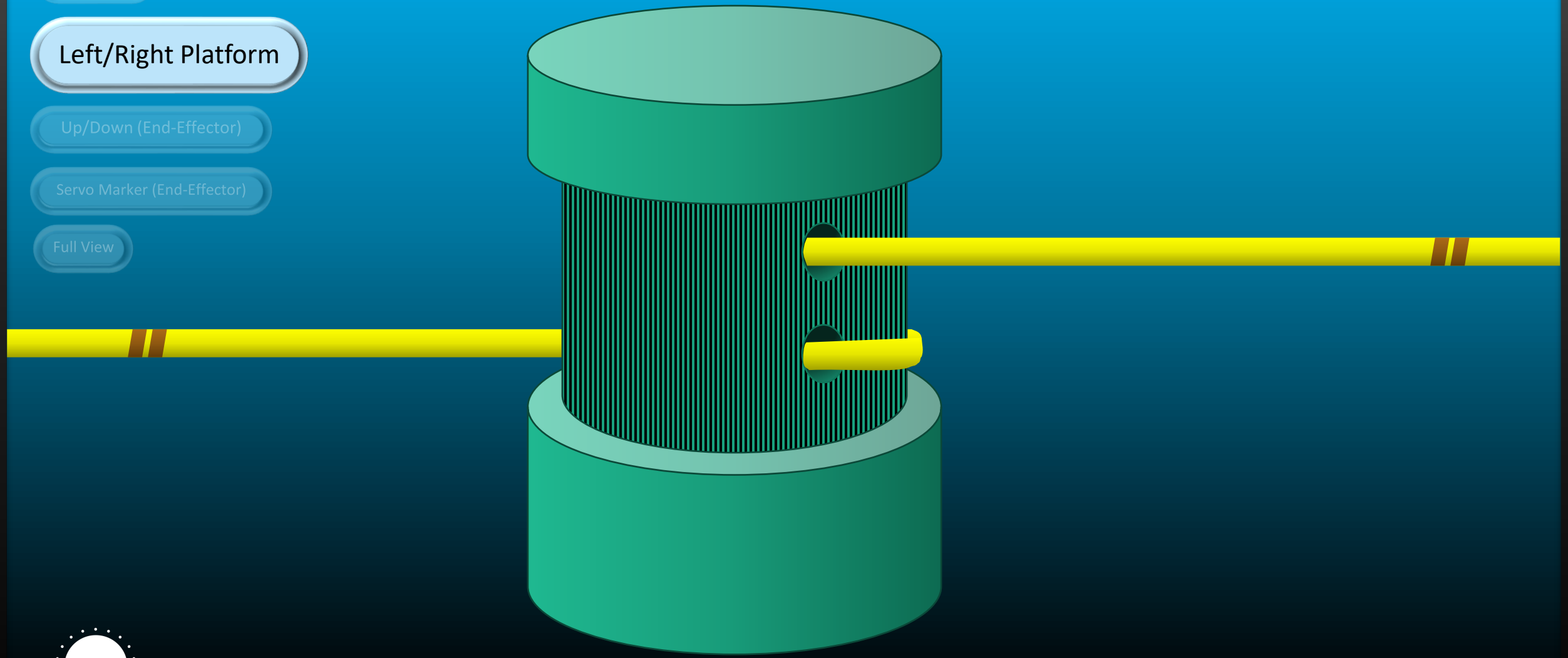
Servo Motor

Left/Right Platform

Up/Down (End-Effector)

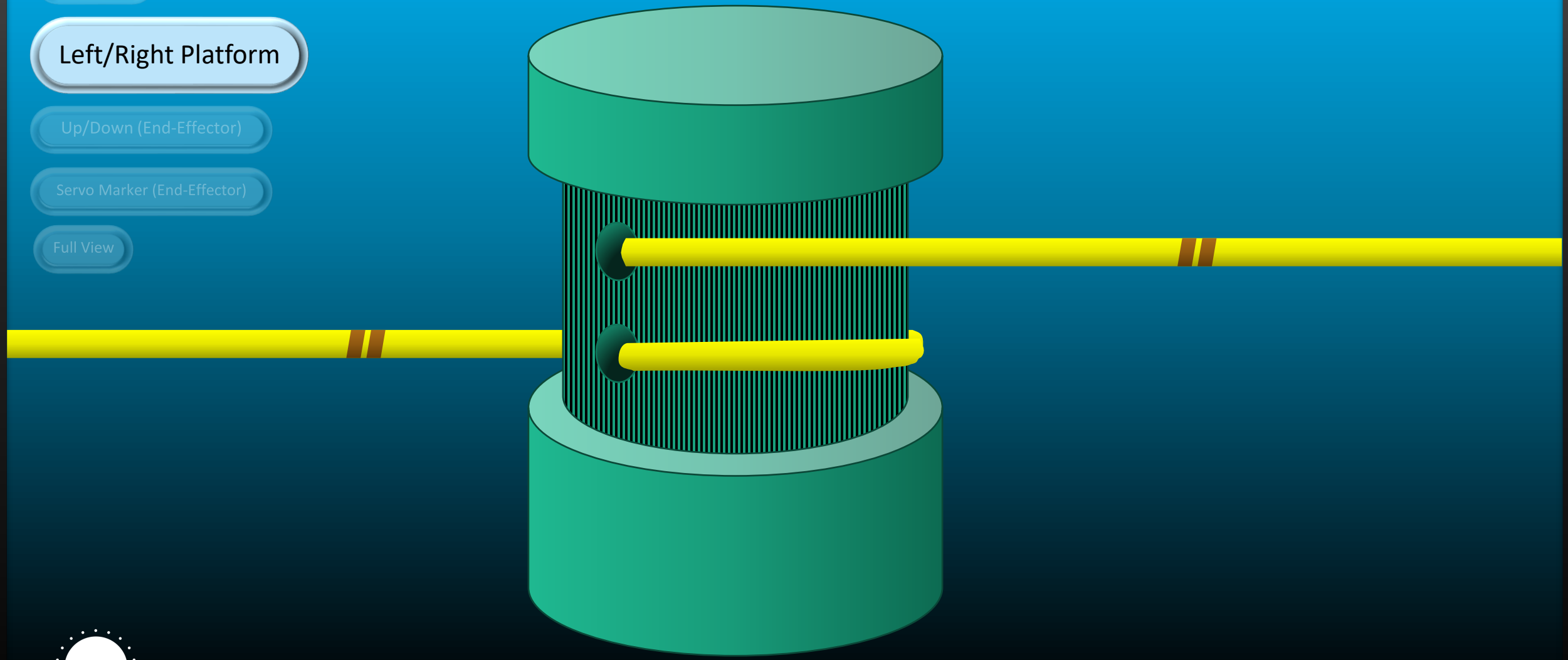
Servo Marker (End-Effector)

Full View



Mechanic & Electronics

- Stepper Motor
- Servo Motor (...)
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



Mechanic & Electronics

Stepper Motor

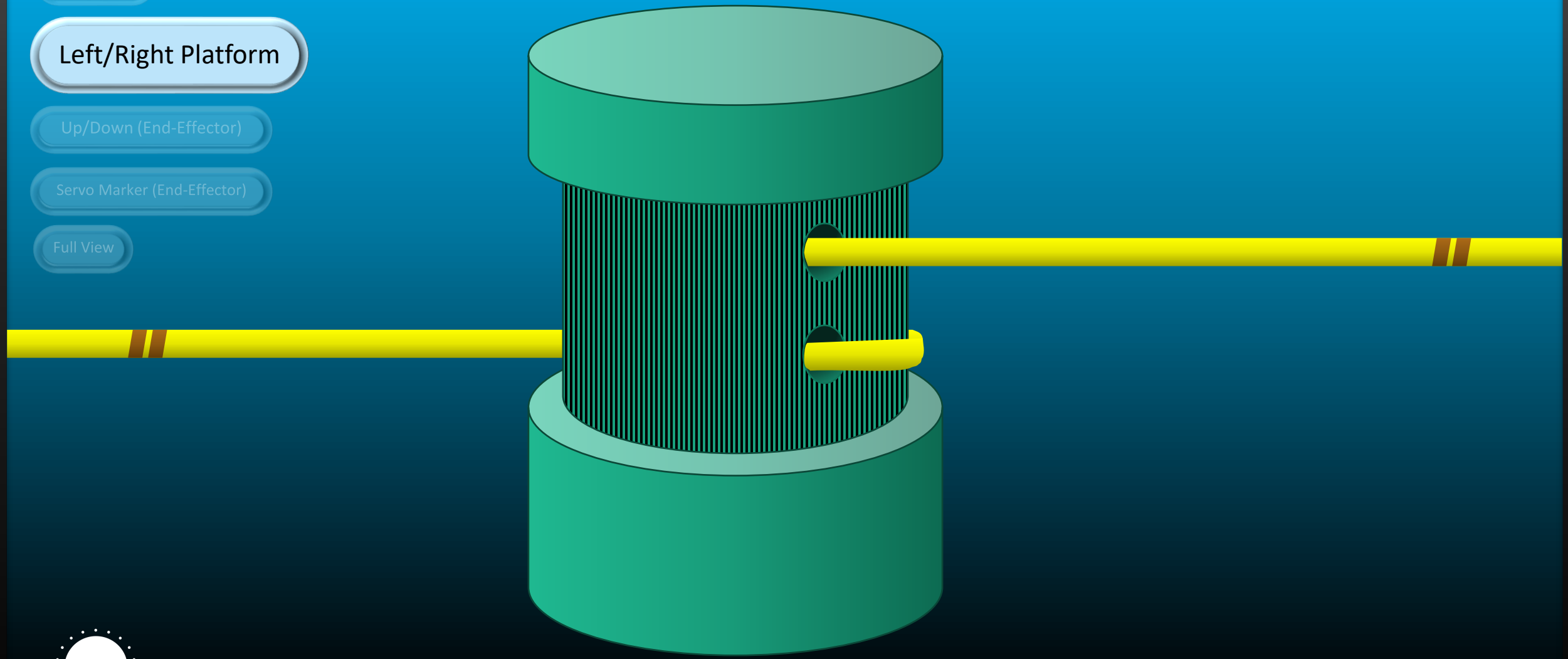
Servo Motor (...)

Left/Right Platform

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Mechanic & Electronics

Stepper Motor

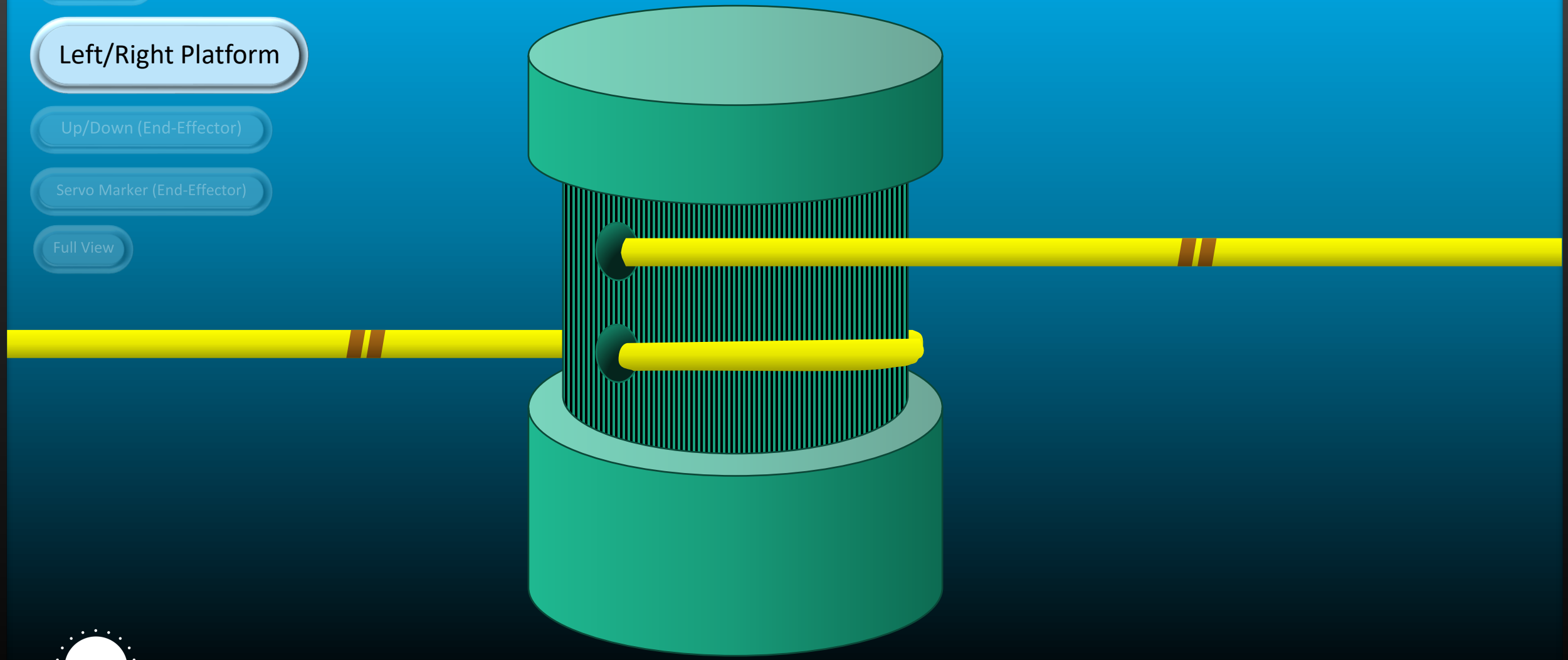
Servo Motor (...)

Left/Right Platform

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



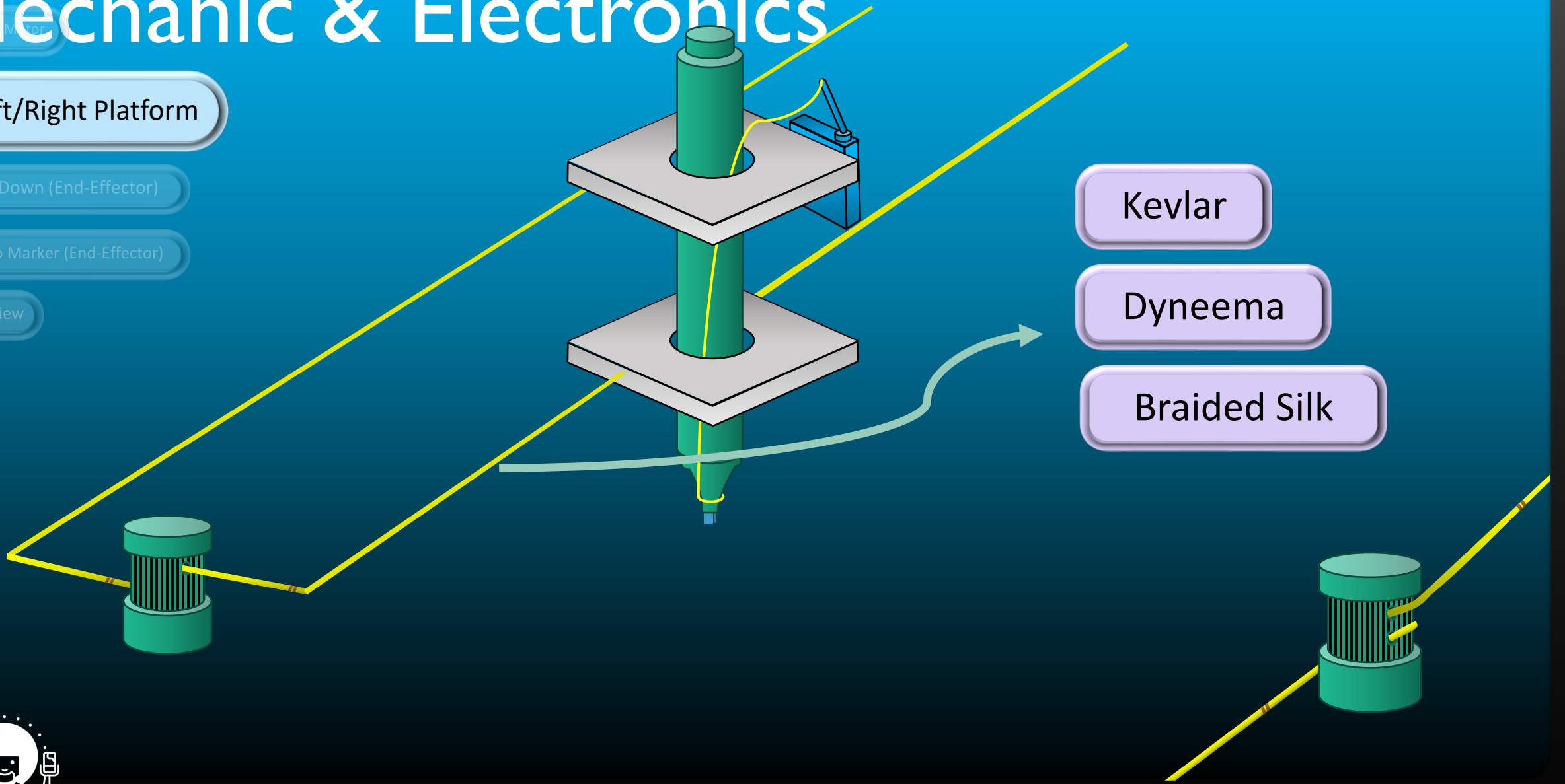
Mechanic & Electronics

Left/Right Platform

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Kevlar

Dyneema

Braided Silk

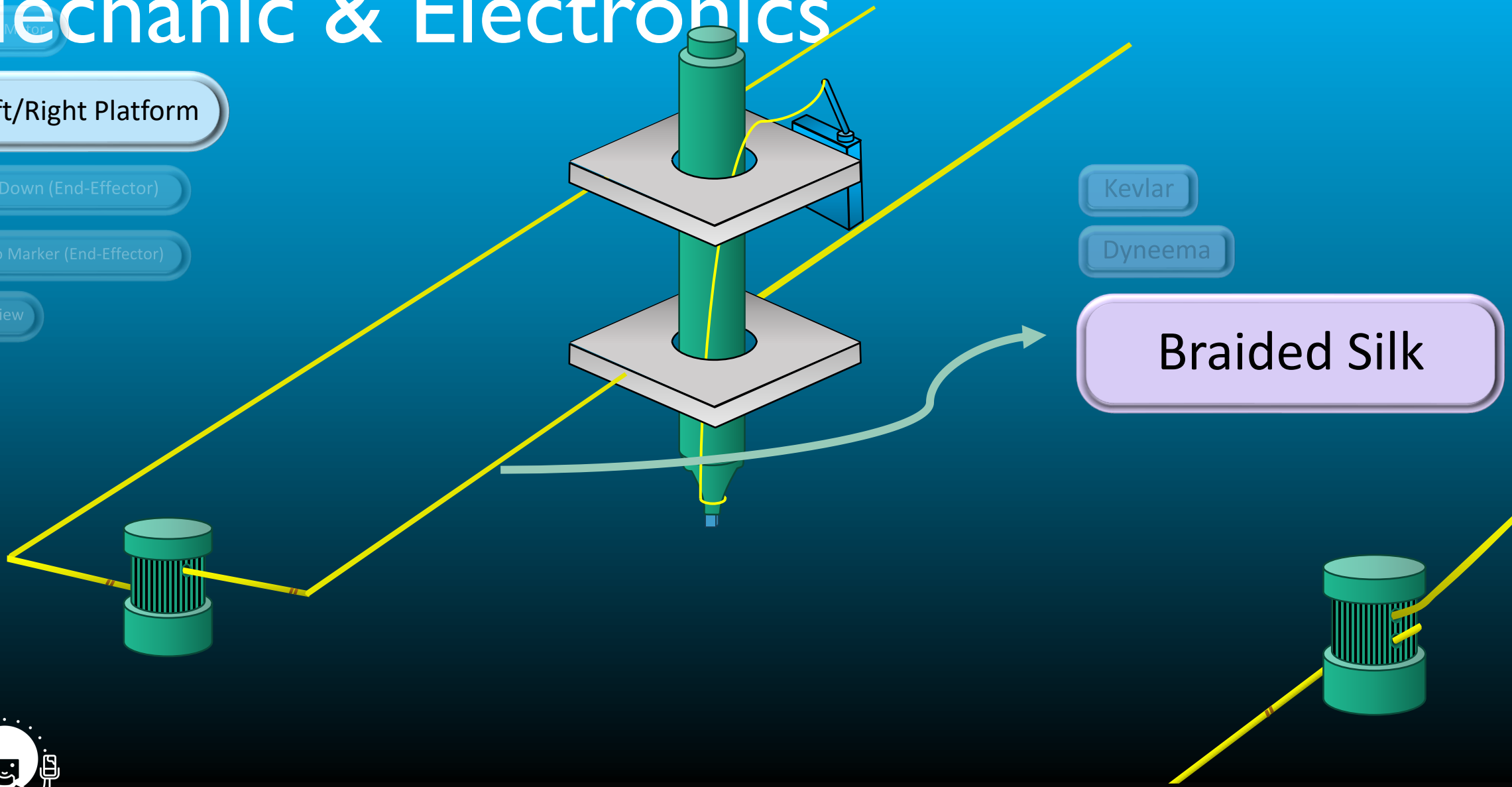
Mechanic & Electronics

Left/Right Platform

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View



Kevlar

Dyneema

Braided Silk

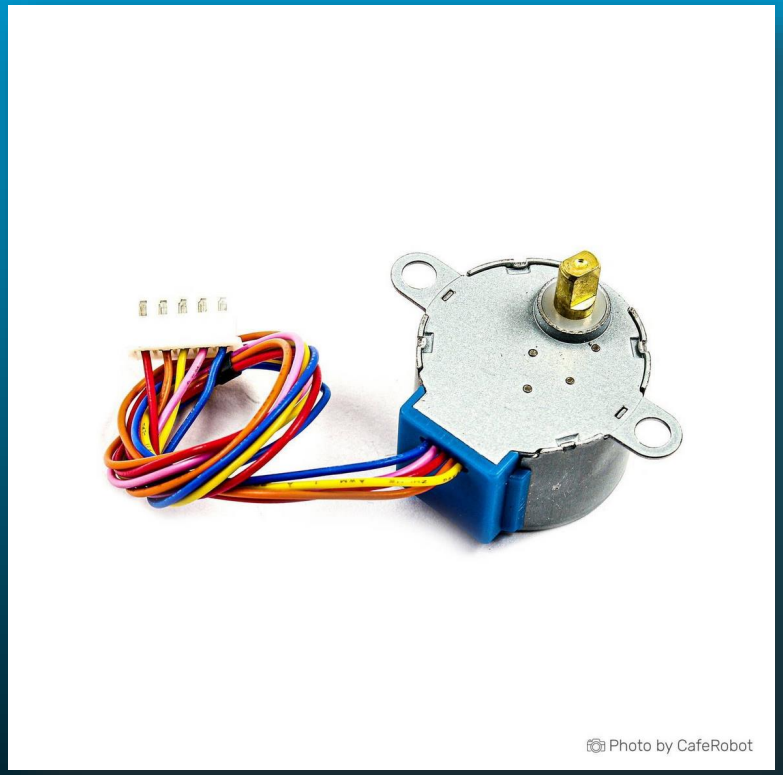
Mechanic & Electronics

- Driver Motor
- Stepper Motor
- **Servo Motor**
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View

T-Pro Mini Servo S690

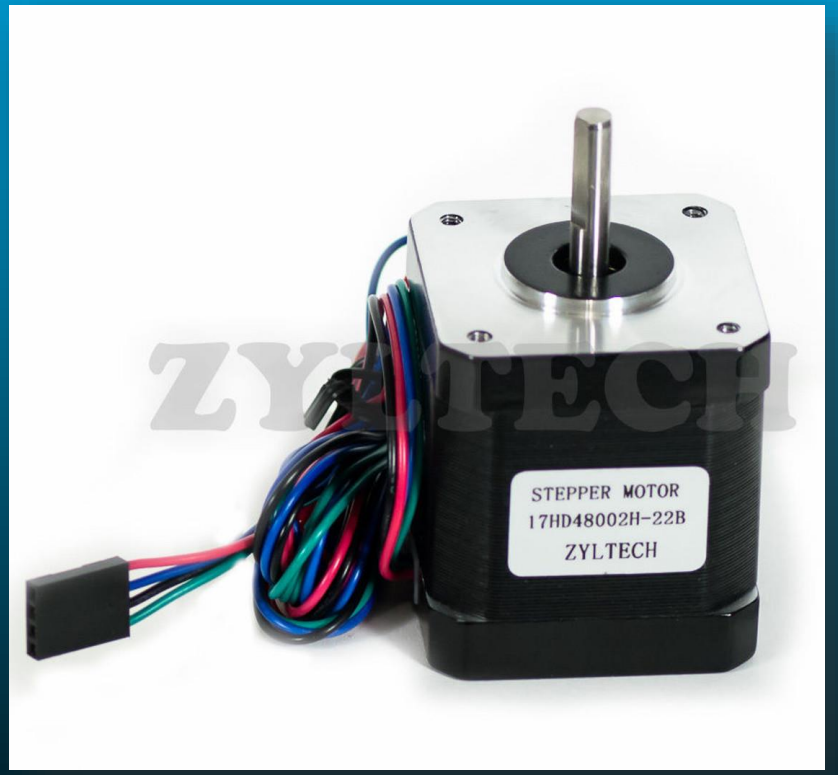
Mechanic & Electronics

- Driver Motor
- Stepper Motor
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



28BYJ-48

bipolar (5.625)

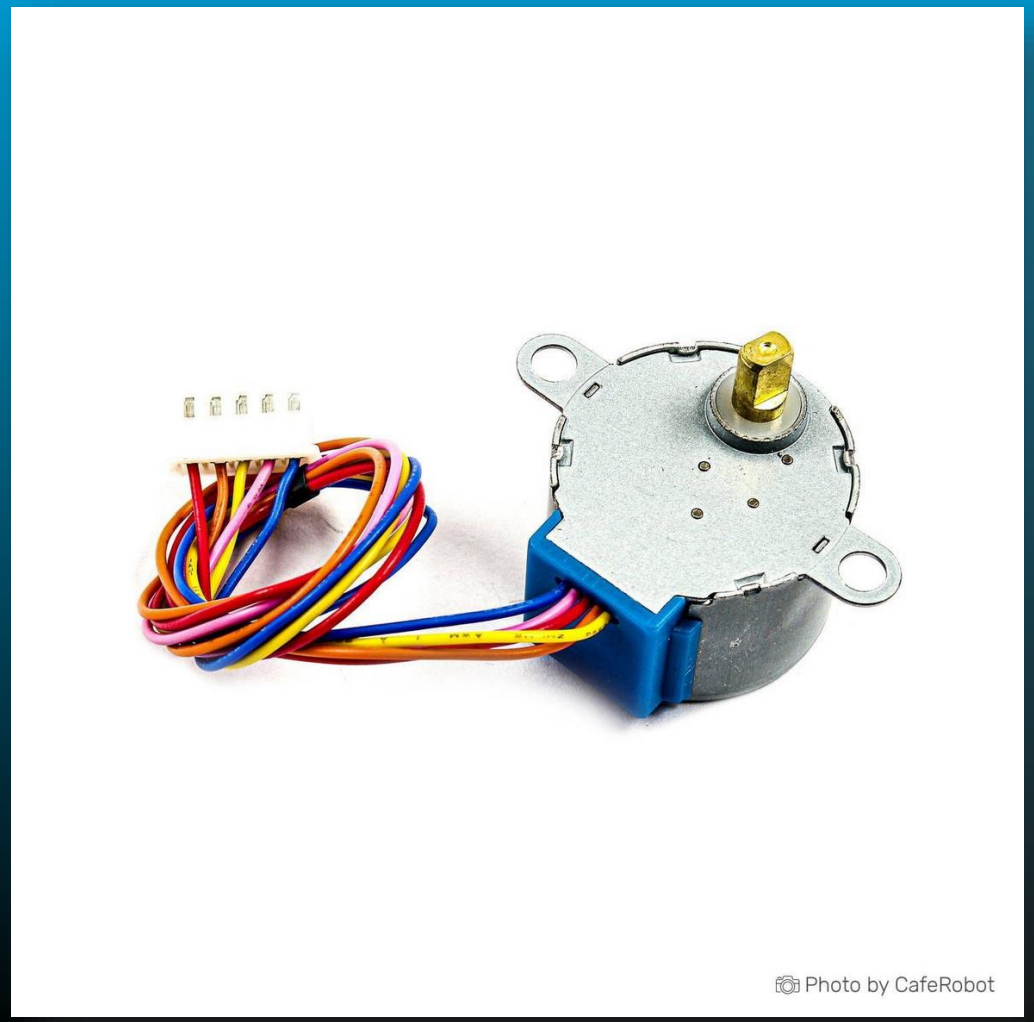


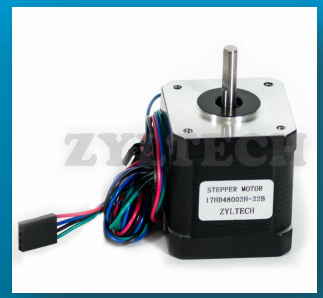
Nema 17

unipolar (1.8)

Mechanic & Electronics

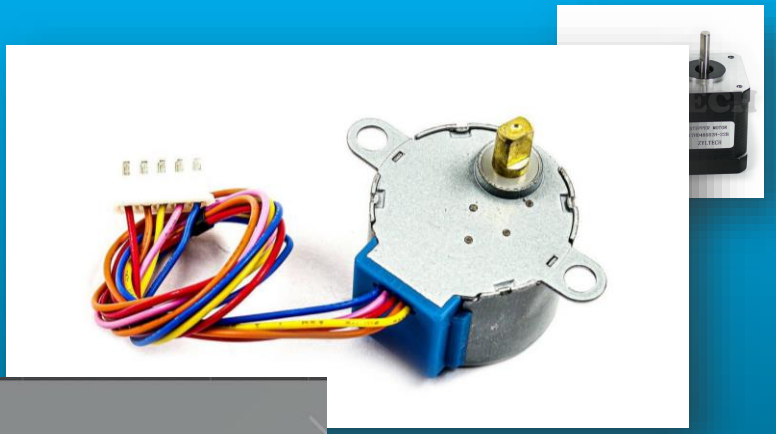
- Driver Motor
- Stepper Motor
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View

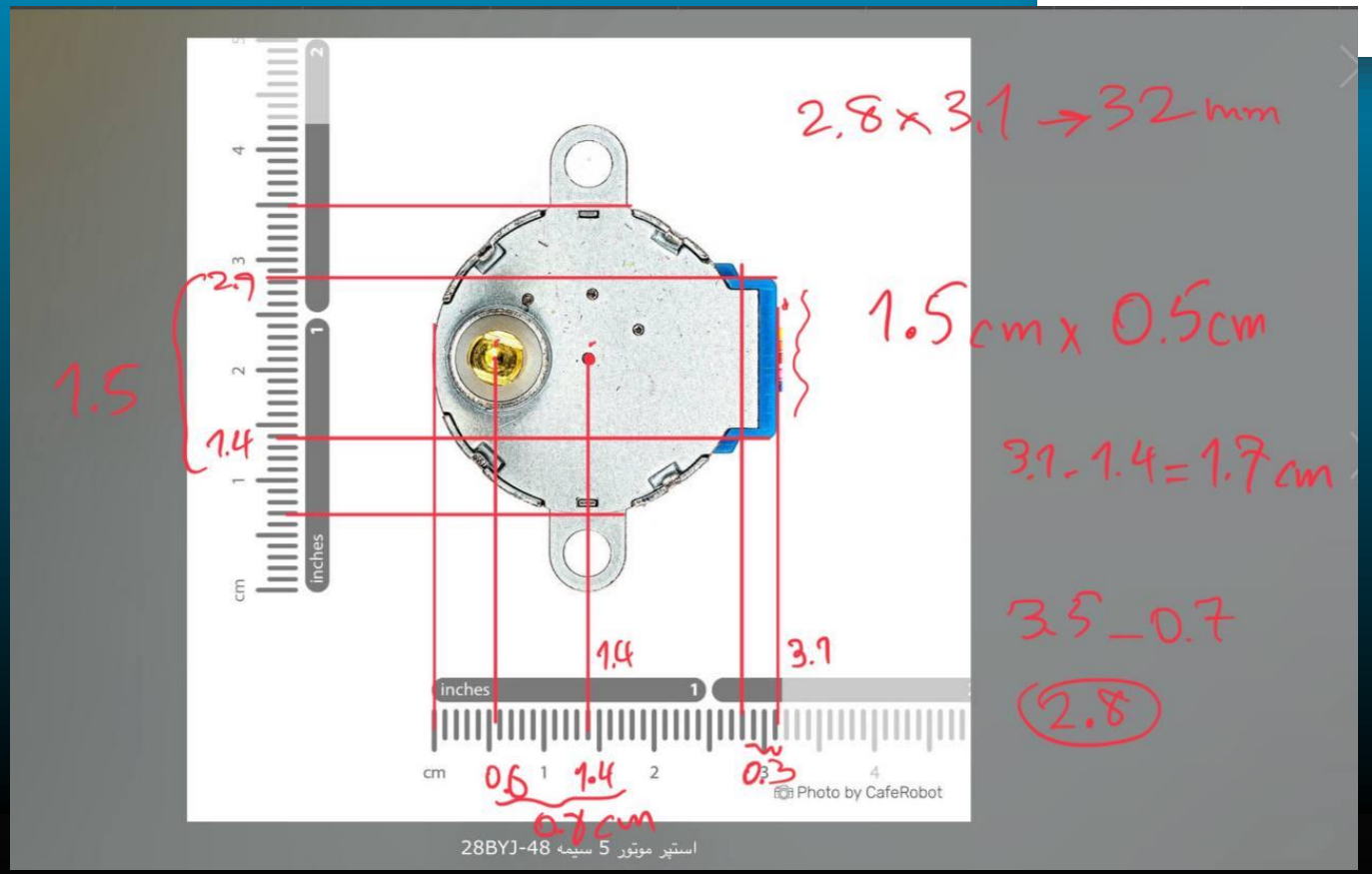




Mechanic & Electronics

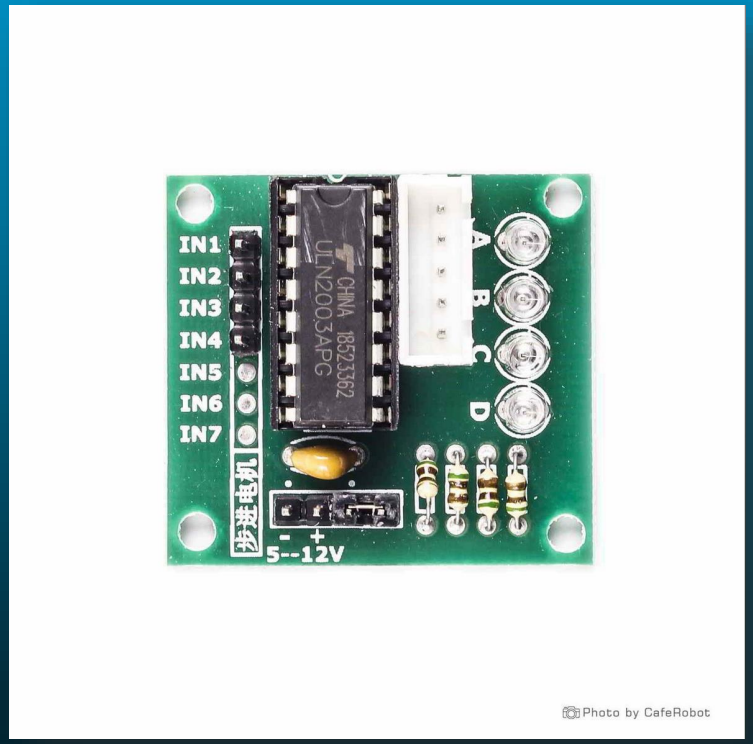
- Driver Motor
- Stepper Motor
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



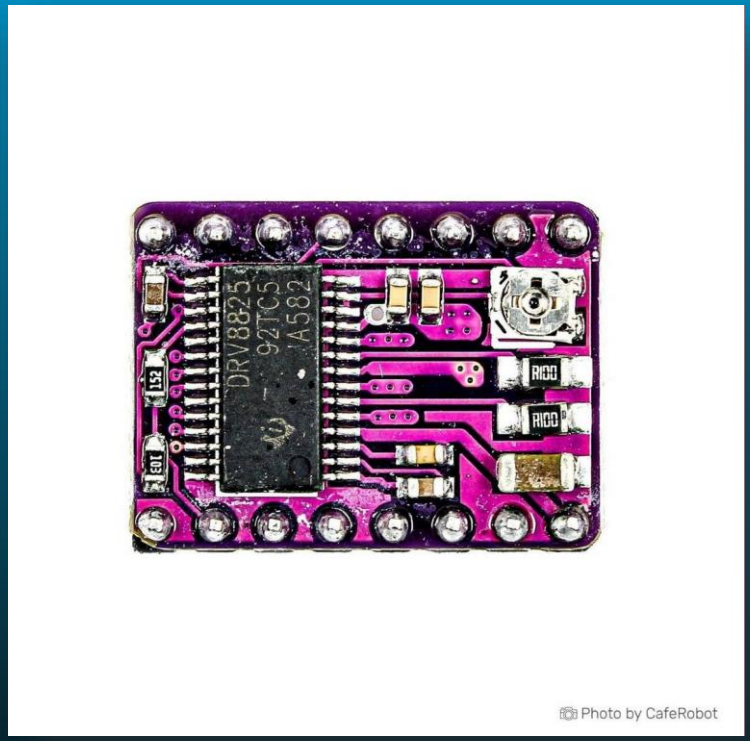


Mechanic & Electronics

- Driver Motor
- Stepper Motor
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



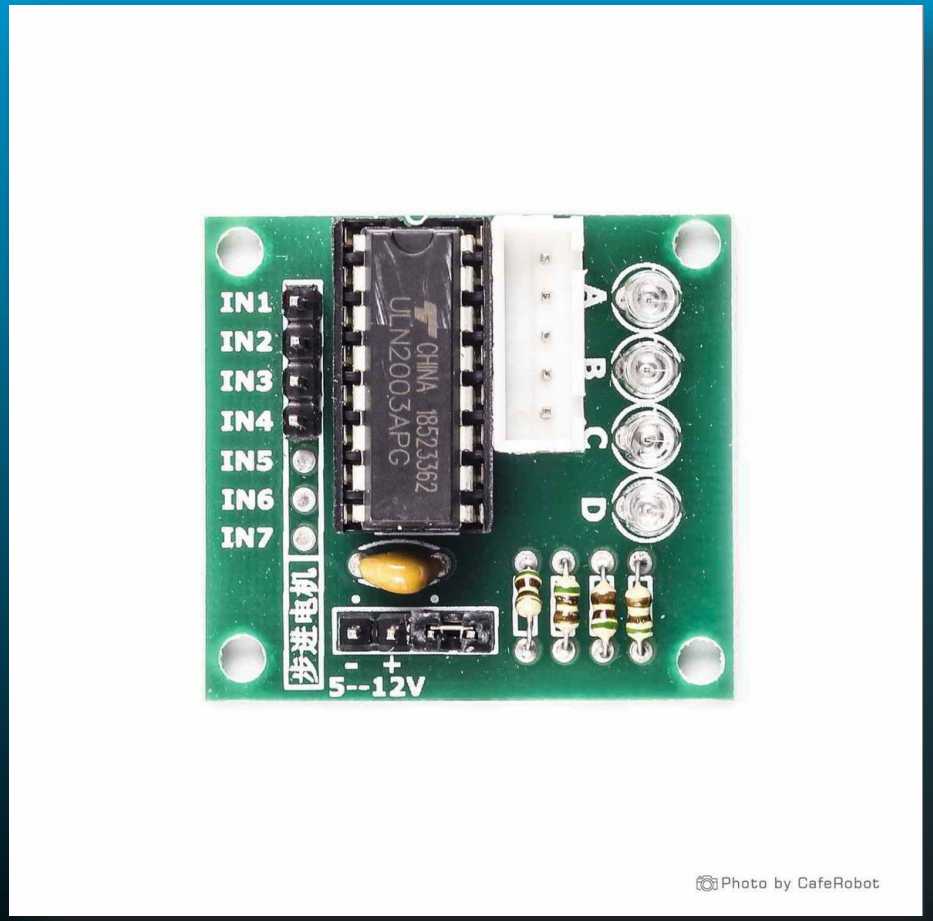
ULN2003

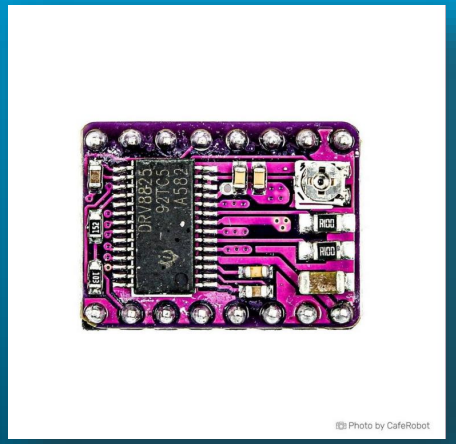


DRV8825

Mechanic & Electronics

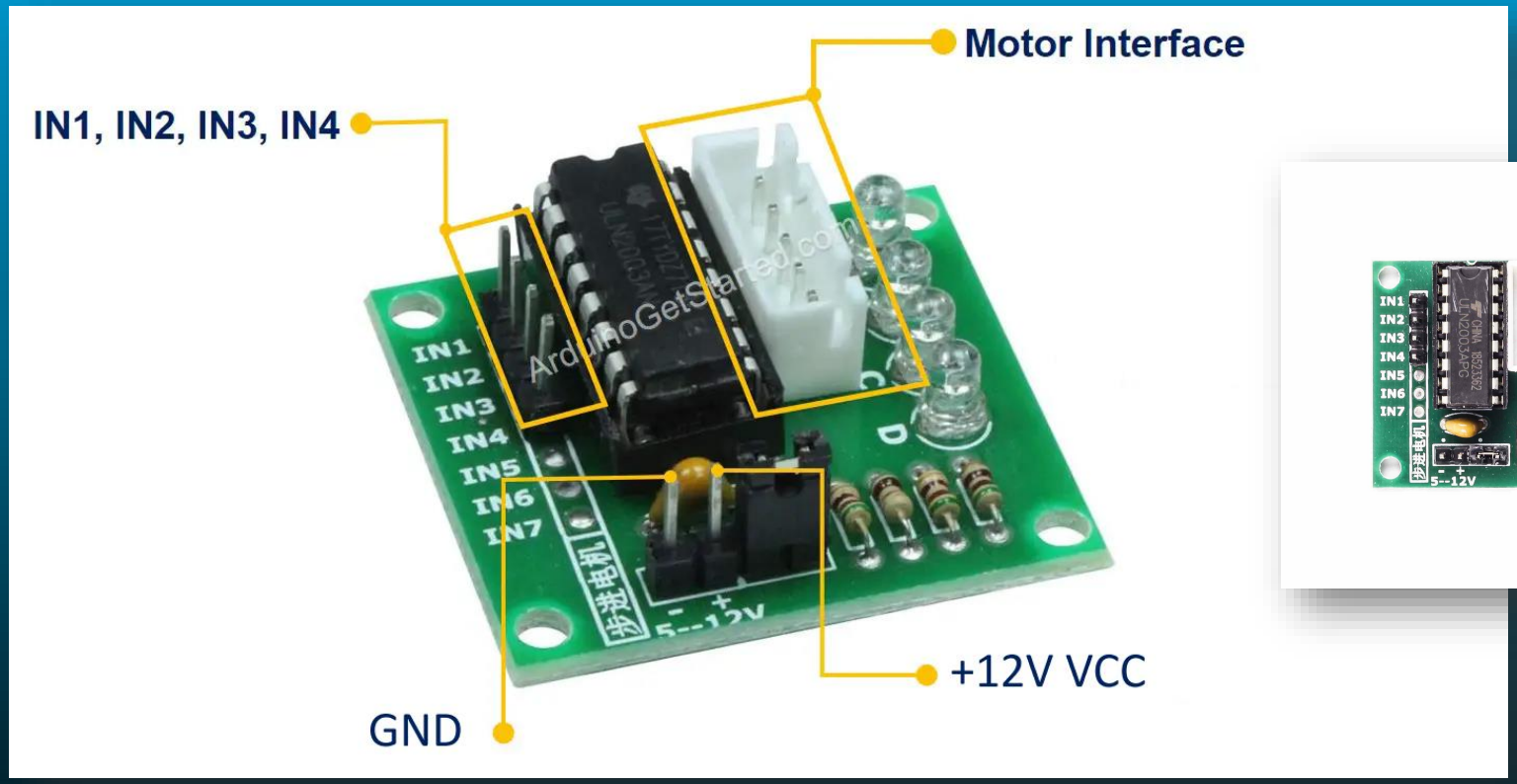
- Driver Motor
- Stepper Motor
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View





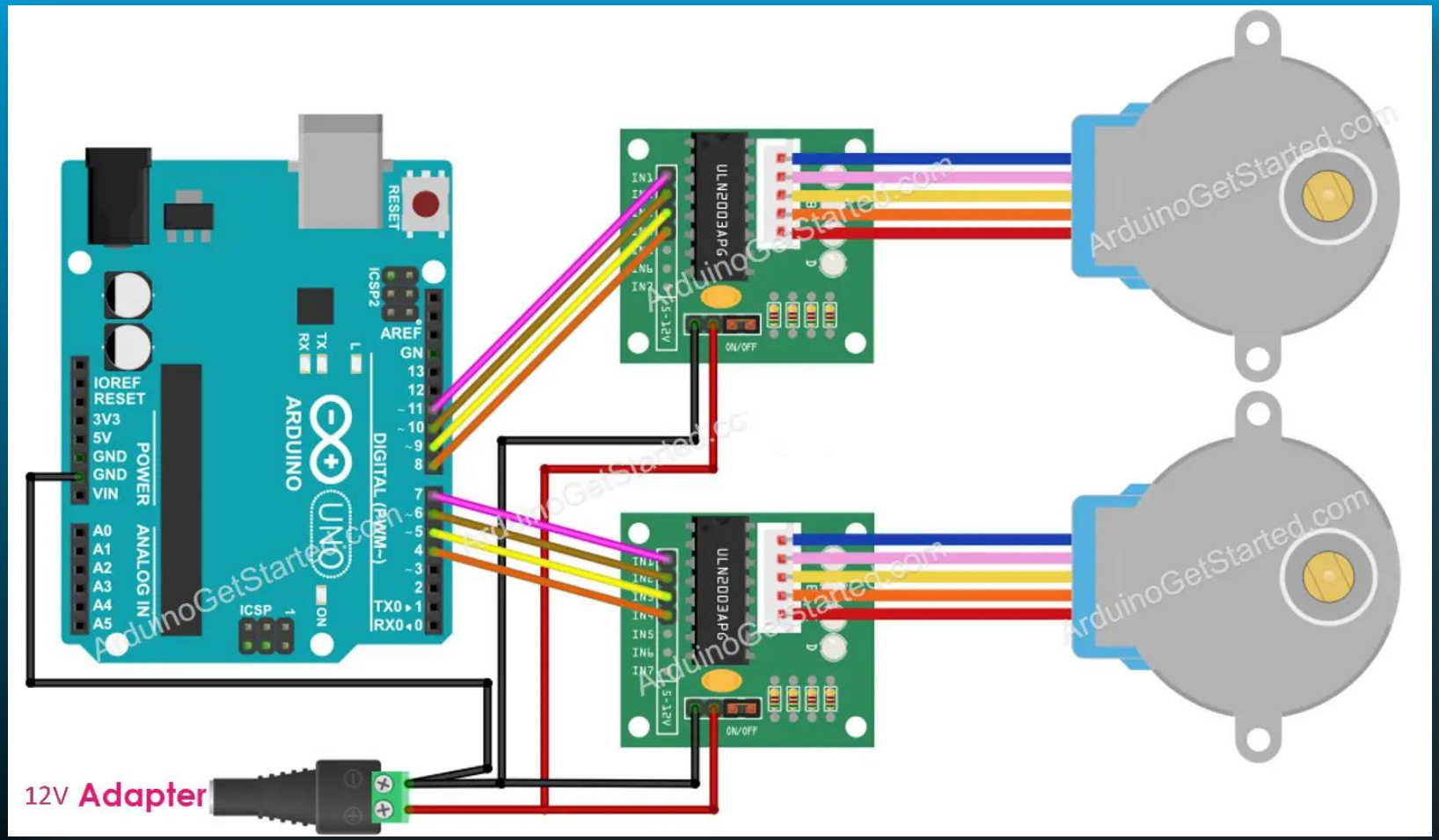
Mechanic & Electronics

- Driver Motor
- Stepper Motor
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



Mechanic & Electronics

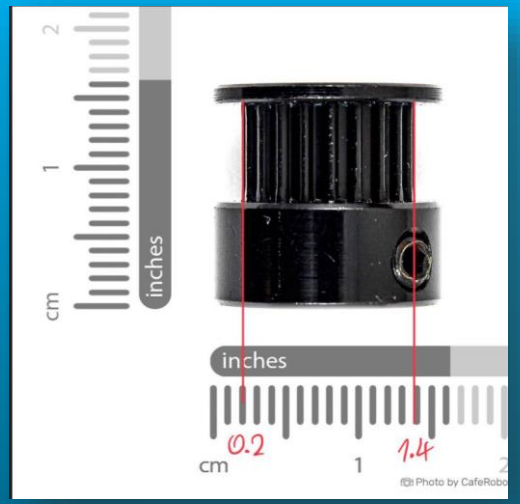
- Driv
- Wiring
- Stepp
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View



Mechanic & Electronics

- Driver
- Stepper
- Servo Motor
- Left/Right Platform
- Up/Down (End-Effector)
- Servo Marker (End-Effector)
- Full View

Equations



$$D = 1.2\ mm$$

$$\frac{5.625°}{360°} \times \pi \times 1.2 = 0.58\ mm \approx 0.5mm$$

$in\ real-world$

$1\ mm$ → *equal to* → $100\ ms$

$linear(\times 10)$

$10\ mm = 1cm$ → *equal to* → $1000\ ms = 1s$

Mechanic & Electronics

Driver

Step

Servo Motor

Left/Right Platform

Up/Down (End-Effector)

Servo Marker (End-Effector)

Full View

Equations

$$D = 1.2\ mm$$

$$\frac{5.625°}{360°} \times \pi \times 1.2 = 0.58\ mm \approx 0.5mm$$

$in\ real - world$

$$1\ mm \quad \text{equal to} \quad 100\ ms$$

$linear(\times 10)$

$$10\ mm = 1cm \quad \text{equal to} \quad 1000\ ms = 1s$$

$$\Delta y = \Delta t$$

$$\Delta x = \Delta t$$

Programming

- Embedded Arduino
- Image Processing
- Algorithm
- Capabilities

Capabilities

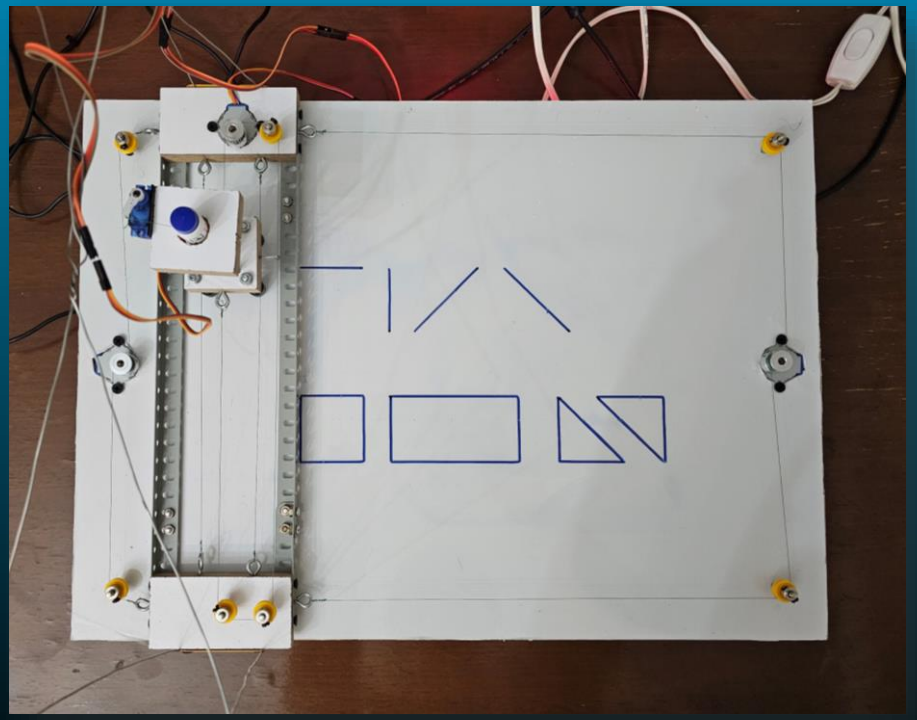
Algorithm

Image Processing

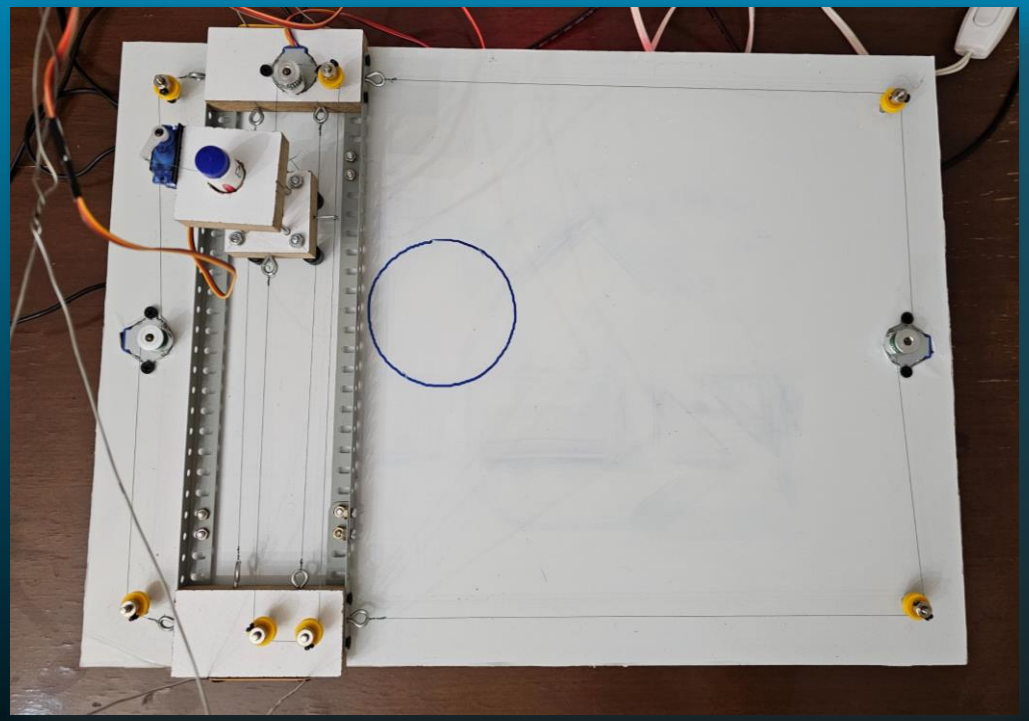
Programming

Embedded Arduino

Preset



Creative



Capabilities

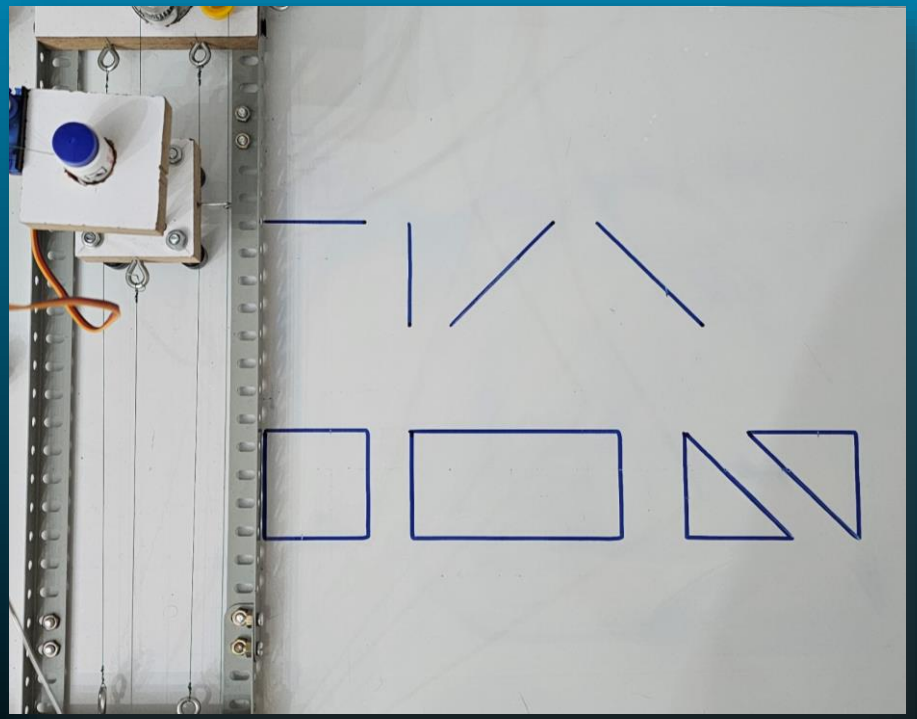
Algorithm

Image Processing

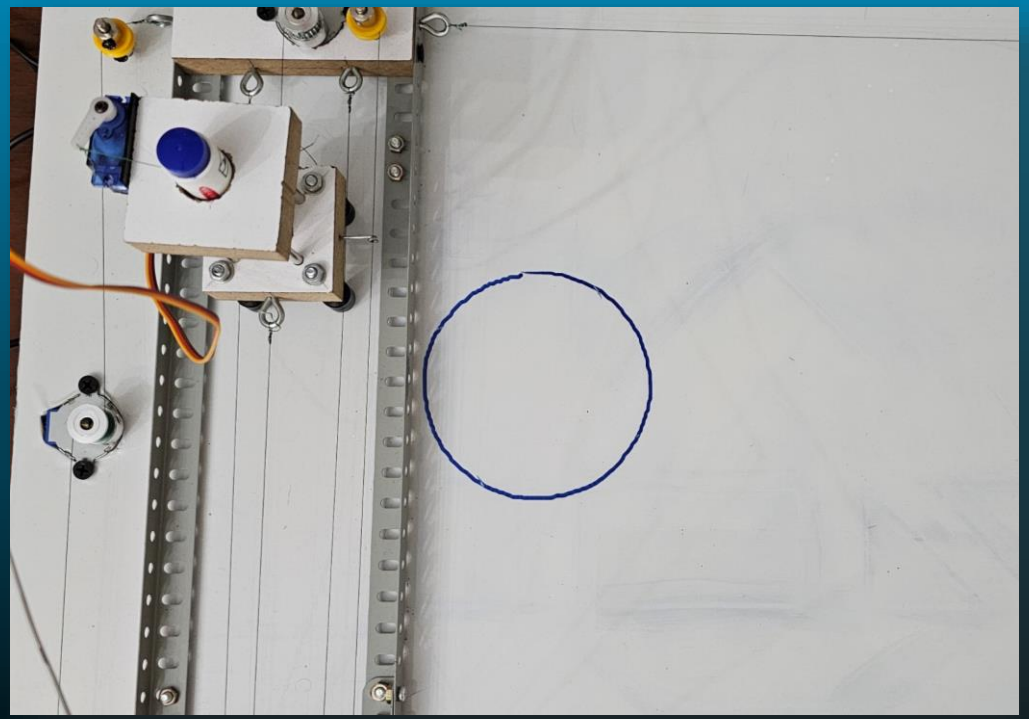
Programming

Embedded Arduino

Preset



Creative



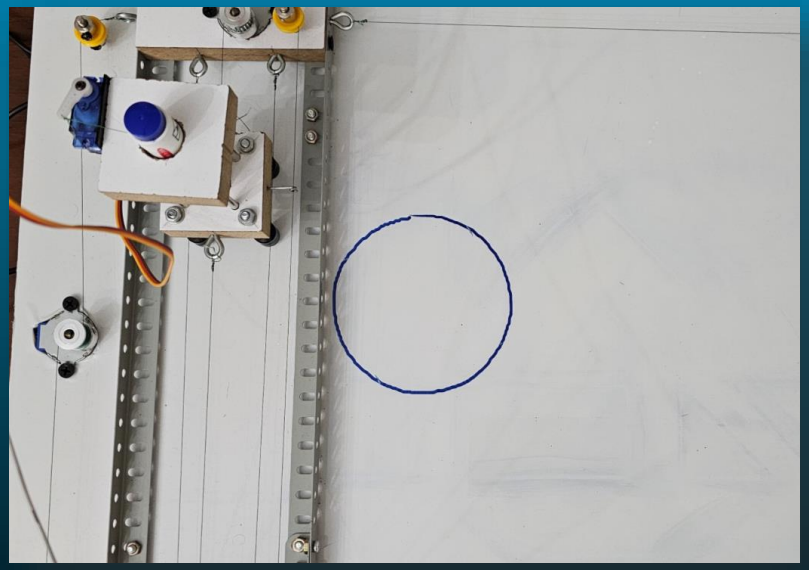
Capabilities

Algorithm

Image Processing

Programming

Embedded Arduino



Command = “D, Y, X”

Creative

Draw { 0 1

Direction { 0 + -

{"1-0", "10-", "1+-", "1-0", "1-+", "10+",
"1-0", "1-0", "0+0", "0+-", "10-", "1-0",
"1-0", "00-", "00-", "0+0", "0+0", "0+0",
"0+0", "1-0", "1-0", "1-0", "1-0", "0++",
"0++", "0++", "0++", "000"}

Capabilities

Algorithm

Image Processing

Programming

Embedded Arduino

Creative

0 0 0 → ❌

x x x → 🪢

Command = “D, Y, X”

Draw { 0, 1 }

Direction { 0, +, - }

- 0 → ⬇️

+ 0 → ⬆️

0 - → ➡️

0 + → ⬅️

- - → ↘️

+ - → ↗️

- + → ↙️

+ + → ↖️

```
{"1-0",
"10-",
"1+-",
"1-0",
"1-+",
"10+",
 "1-0",
"1-0",
"0+0",
"0+-",
"10-",
"1-0",
 "1-0",
"00-",
"00-",
"0+0",
"0+0",
"0+0",
 "0+0",
"1-0",
"1-0",
"1-0",
"1-0",
"0++",
 "0++",
"0++",
"0++",
"000"}
```

```
1   Painting Segment 1:
2   Dr fr (0, 0) to (1, 0)
3   Dr fr (1, 0) to (1, 1)
4   Dr fr (1, 1) to (0, 2)
5   Dr fr (0, 2) to (1, 2)
6   Dr fr (1, 2) to (2, 1)
7   Dr fr (2, 1) to (2, 0)
8   Dr fr (2, 0) to (3, 0)
9   Dr fr (3, 0) to (4, 0)
10  Mv fr (4, 0) to (3, 0)
11  Mv fr (3, 0) to (2, 1)
12  Dr fr (2, 1) to (2, 2)
13  Dr fr (2, 2) to (3, 2)
14  Dr fr (3, 2) to (4, 2)
15  Painting Segment 2:
16  Mv fr (4, 2) to (4, 3)
17  Mv fr (4, 3) to (4, 4)
18  Mv fr (4, 4) to (3, 4)
19  Mv fr (3, 4) to (2, 4)
20  Mv fr (2, 4) to (1, 4)
21  Mv fr (1, 4) to (0, 4)
22  Dr fr (0, 4) to (1, 4)
23  Dr fr (1, 4) to (2, 4)
24  Dr fr (2, 4) to (3, 4)
25  Dr fr (3, 4) to (4, 4)
26  Mv fr (4, 4) to (3, 3)
27  Mv fr (3, 3) to (2, 2)
28  Mv fr (2, 2) to (1, 1)
29  Mv fr (1, 1) to (0, 0)
30  EOF
```

Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino

Input

Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino

Array (Matrix) | Text (string) | Image (.jpg, .png, …) | Input

Capabilities

Algorithm

Programming

Image Processing

Embedded Arduino

Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

“Hello, I am Bot-Ross.”

Image (.jpg, .png, …)

Input

Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino

Array (Matrix)

Text (string)

Image (.jpg, .png, …)

Input

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

"Hello, I am Bot-Ross."

Integrator(Input, method)

OUTPUT = An Image with specific dimensions and format

Capabilities

Algorithm

Programming

Image Processing

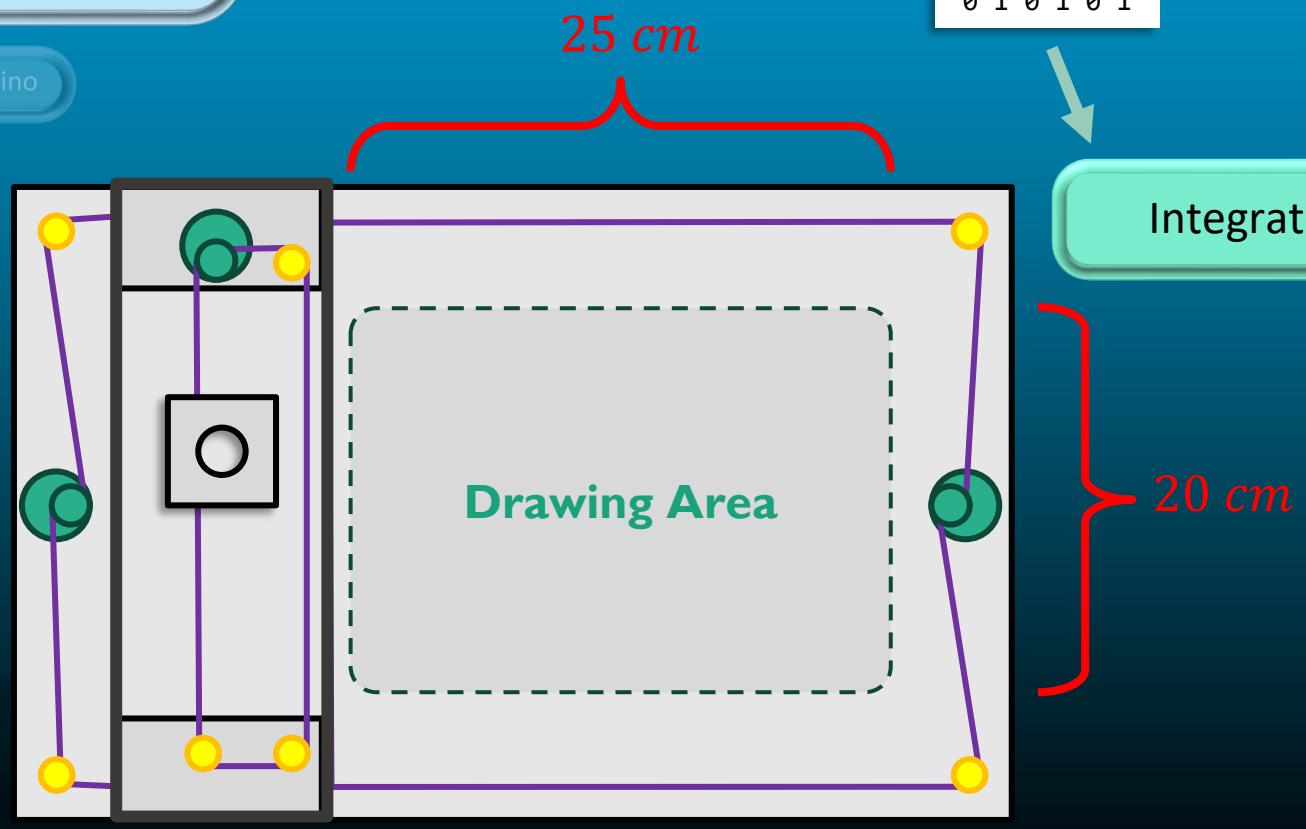
Embedded Arduino

Array (Matrix) | Text (string) | Image (.jpg, .png, …) | Input

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

"Hello, I am Bot-Ross."

Integrator(Input, method)



Capabilities

Algorithm

Programming

Image Processing

Embedded Arduino

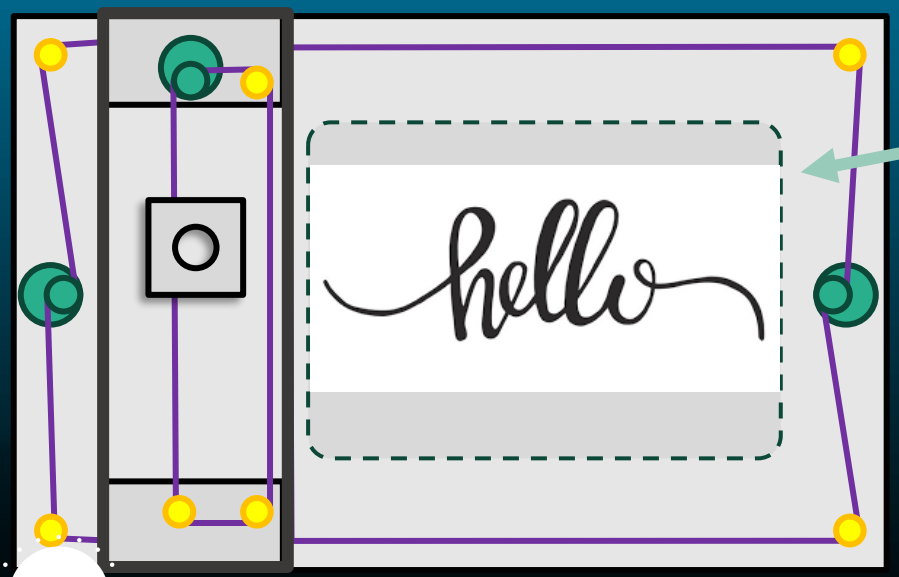
Array (Matrix) | Text (string) | Image (.jpg, .png, …) | Input

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

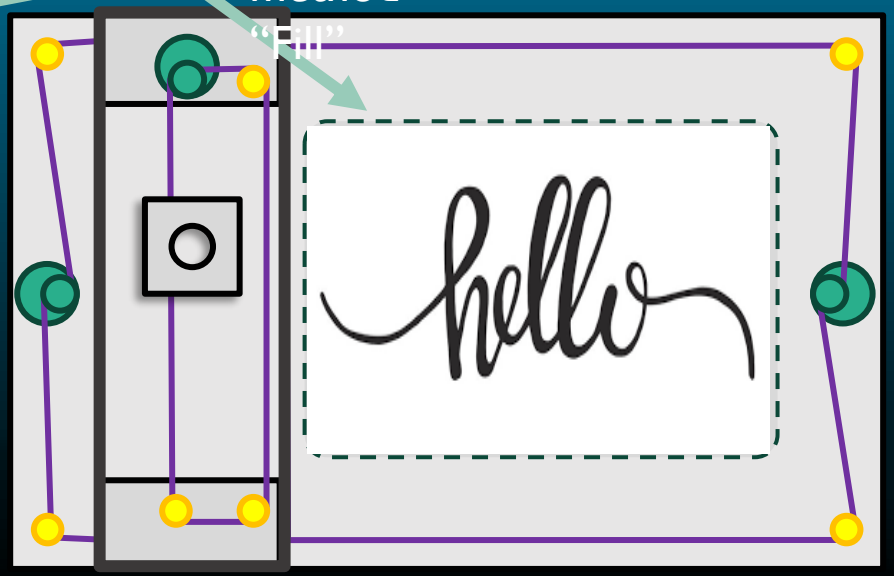
"Hello, I am Bot-Ross."

Integrator(Input, method)

method = "Fit"



method = "Fill"



Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino

Input

Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

"Hello, I am Bot-Ross."

Image (.jpg, .png, …)

Integrator(Input, method)

convert_to_binary(image)

OUTPUT = An Image with {0, 1} pixel values based on cv2 Python library.

Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino

Input

Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

"Hello, I am Bot-Ross."

Image (.jpg, .png, …)

Integrator(Input, method)

convert_to_binary(image)

painter(binary_image)

OUTPUT = File in .txt format for simulating drawing process.

Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino

Input

Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

"Hello, I am Bot-Ross."

Image (.jpg, .png, ...)

Integrator(Input, method)

convert_to_binary(image)

painter(binary_image)

Image_to_graph(binary_image)

OUTPUT = A graph which extracted from binary_image

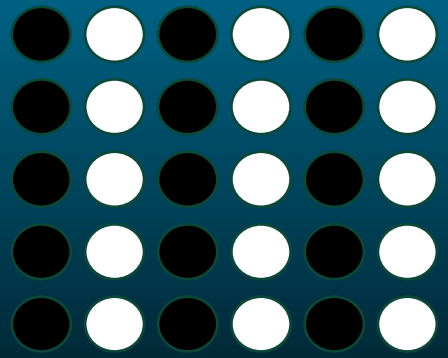
Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino



```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Image_to_grapg(binary_image)

painter(binary_image)

convert_to_binary(image)

Integrator(Input, method)

Input

Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

"Hello, I am Bot-Ross."

Image (.jpg, .png, …)

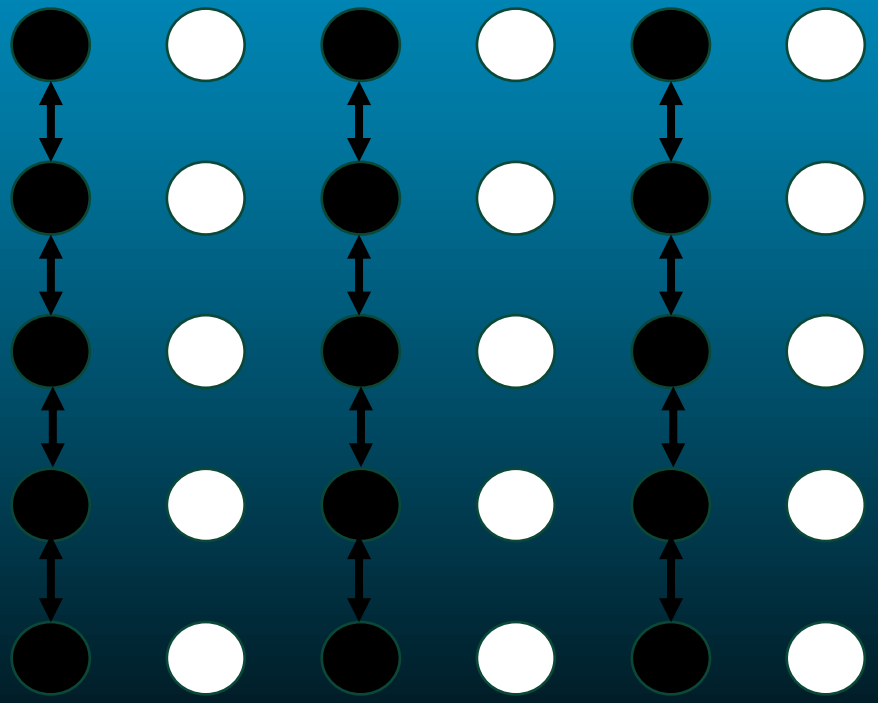
Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino



Array (Matrix) | Text (string) | Image (.jpg, .png, …) | Input

0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1

"Hello, I am Bot-Ross."

Integrator(Input, method)

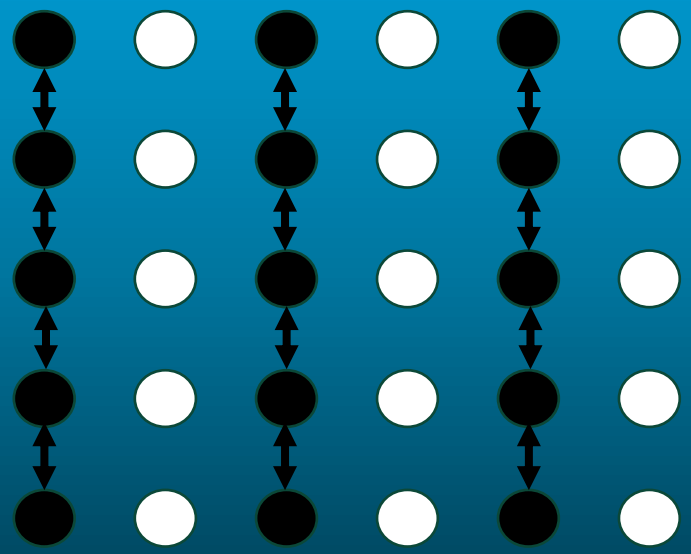
convert_to_binary(image)

painter(binary_image)

Image_to_grapg(binary_image)

Programming

- Capabilities
- Algorithm
- Image Processing
- Embedded Arduino



Array (Matrix) | Text (string) | Image (.jpg, .png, …) | Input

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

"Hello, I am Bot-Ross."

Integrator(Input, method)

convert_to_binary(image)

painter(binary_image)

Image_to_graph(binary_image)

find_spanning_trees(graph, method)

OUTPUT = Spanning Trees
(Spaning Jungle)

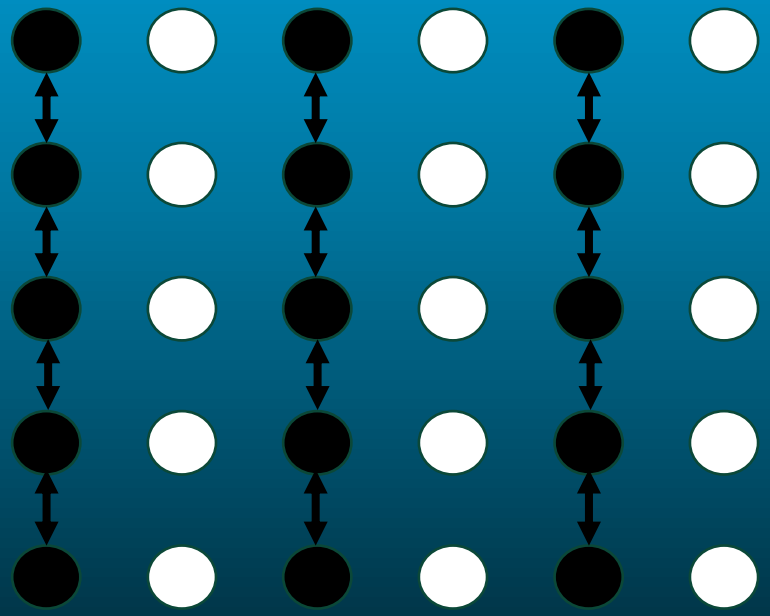
Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino



Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

"Hello, I am Bot-Ross."

Image (.jpg, .png, ...)

Input

Integrator(Input, method)

convert_to_binary(image)

painter(binary_image)

Image_to_graph(binary_image)

find_spanning_trees(graph, method)

A*

TSP

BFS

DFS

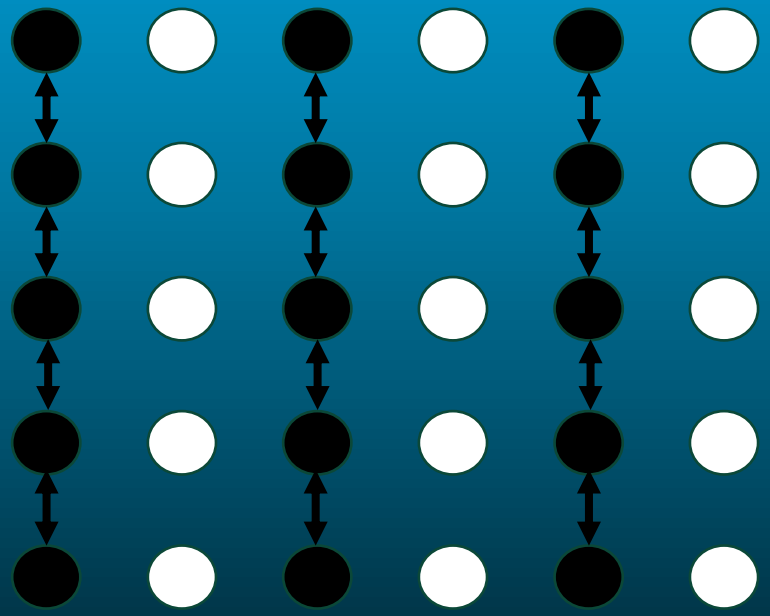
Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino



Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

"Hello, I am Bot-Ross."

Image (.jpg, .png, ...)

Input

Integrator(Input, method)

convert_to_binary(image)

painter(binary_image)

Image_to_graph(binary_image)

DFS

find_spanning_trees(graph, method)

A*

TSP

BFS

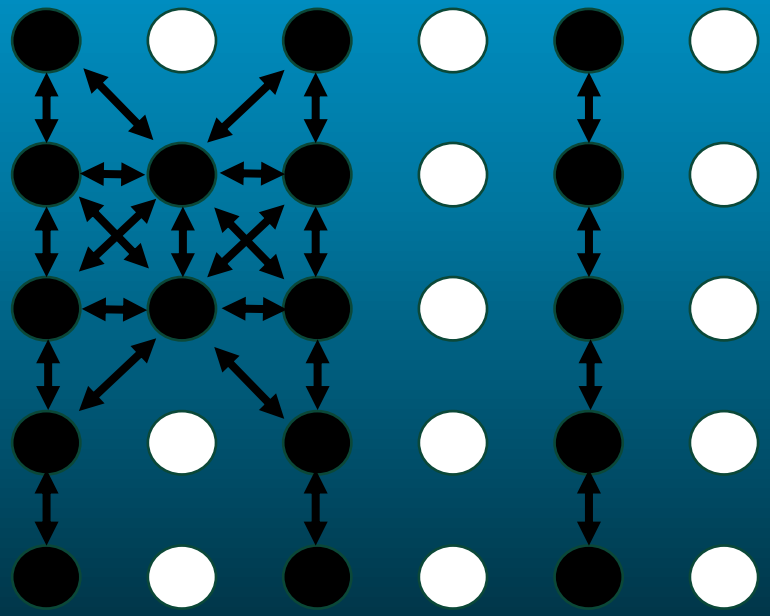
Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino



Array (Matrix)

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

Text (string)

"Hello, I am Bot-Ross."

Image (.jpg, .png, ...)

Input

Integrator(Input, method)

convert_to_binary(image)

painter(binary_image)

Image_to_graph(binary_image)

DFS

find_spanning_trees(graph, method)

A*

TSP

BFS

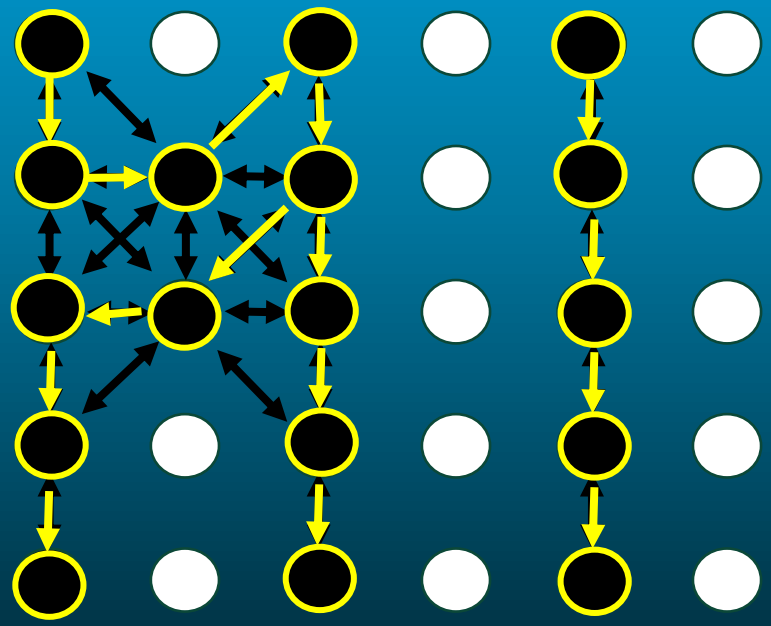
Programming

Capabilities

Algorithm

Image Processing

Embedded Arduino



Array (Matrix) | Text (string) | Image (.jpg, .png, …) | Input

```
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
0 1 0 1 0 1
```

"Hello, I am Bot-Ross."

Integrator(Input, method)

convert_to_binary(image)

painter(binary_image)

Image_to_graph(binary_image)

DFS

find_spanning_trees(graph, method)

A* | TSP | BFS

Programming

- Capabilities
- Algorithm
- Image Processing
- Embedded Arduino

1. Scalability (Any Board Size)
2. Efficiency (Drawing and Coming back Algorithm)
3. Affordability (Minimum Costs)

Challenges and Future Development

- Big Platform
- Adding Eraser
- AI Visual Learning
- Graph Pruning
- Input UI
- Adding Torque Meter Sensor
- Adding Trajectory Planning and Other Brushes

Special thanks to:

Dr. Mehran Hassani

Dr. Heidar Ali Talebi

Mr. Amir Ali Setayeshi

Dr. Khorsandi

ChatGPT & BingAI

Bot-Ross

Robotics Project

THANKS FOR YOUR ATTENTION