runlinc Intermediate Project I8: Servo Motor Buttons (E32W Version)

Contents

Introduction

Problem

Servo motor plays an important role in modern industrial production. And We want to simulate a simple application of the Servo motor in the real-life manufacture.

Background

Servo motors or “servos”, as they are known, are electronic devices and rotary or linear actuators that rotate and push parts of a machine with precision. Servos are mainly used on angular or linear position and for specific velocity, and acceleration. Most of the companies that use servos are manufacturing companies that need it to position control surfaces and rotate objects at precise angles and distances.

Figure 1: robotic arms in automobile production workshop

At home, the intelligent devices do the drudgery we hate—the cleaning and washing. At work, repetitive, dirty, and dangerous tasks are as easy as one-two-three for a robotic arm with servo motors. The type of engine provides accuracy, smoothness, and flexibility of motion comparable to a human limb, while excelling it at repeatability.

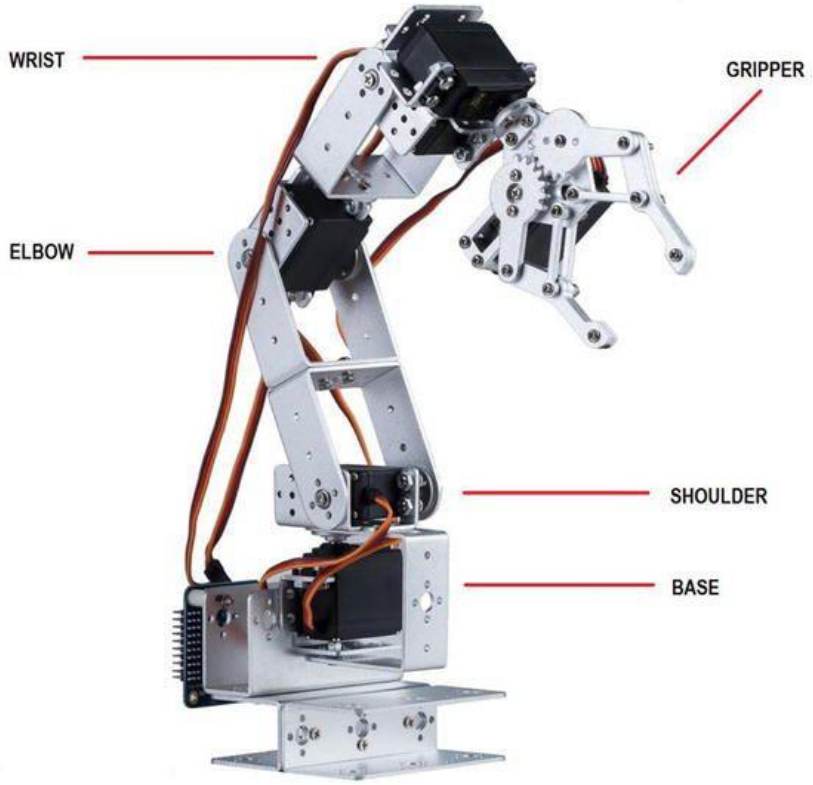

Figure 2 segments of robotic arm

A robotic arm is a programmable mechanism comprising two or more segments linked by means of joints into a kinematic chain. Each joint in the chain is a servo or another motor providing either rotational or linear displacement of the segments.

Ideas

What is the basic function that a servo motor can do in a common industrial production? How can we create buttons in runlinc webpage and how to use buttons to control a servo motor? How can we control the angle of rotation of the pendulum rod?

Plan

We will use the servo motor and pendulum rod to simulate the basic application of servo motor used in the industrial manufacture. We will create three buttons to control the rotation angle of the pendulum rod.

Figure 3: Block diagram of Microchip inputs and outputs

runlinc Background

runlinc is a web page inside a Wi-Fi chip. The programming is done inside the browsers compare to programming inside a chip. The runlinc web page inside the Wi-Fi chip will command the microchips to do sensing, control, data logging Internet of Things (IoT). It can predict and command.

Part A: Design the Circuit on runlinc

Note: Refer to runlinc Wi-Fi Setup Guide document to connect to runlinc

Use the left side of the runlinc web page to construct an input/output (I/O).

For port D13 name it Servo1 and set it as SERVO.

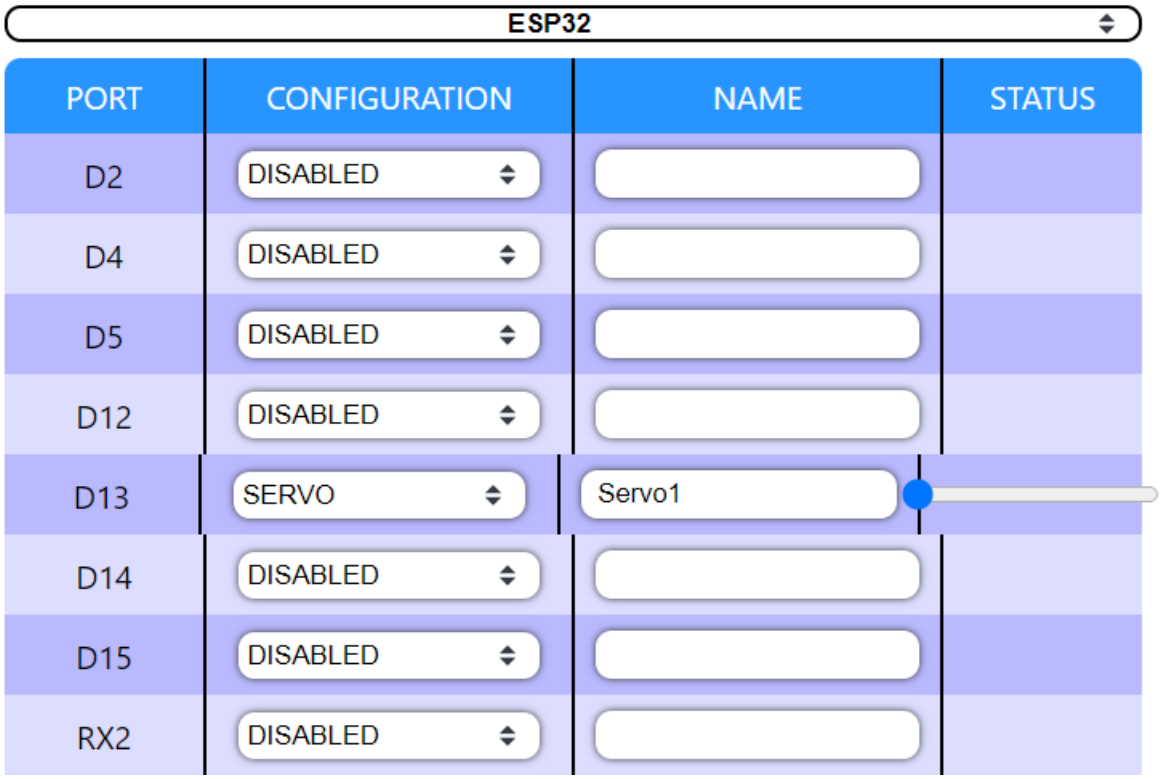

Figure 4: I/O configurations connections

Part B: Build the Circuit

Use the STEMSEL E32W board to connect the hardware. For this project we are using on only the left I/O ports, with **negative port (-ve)** on the outer side, **positive port (+ve)** on the middle and **signal port (s)** on the inner side (as shown below).

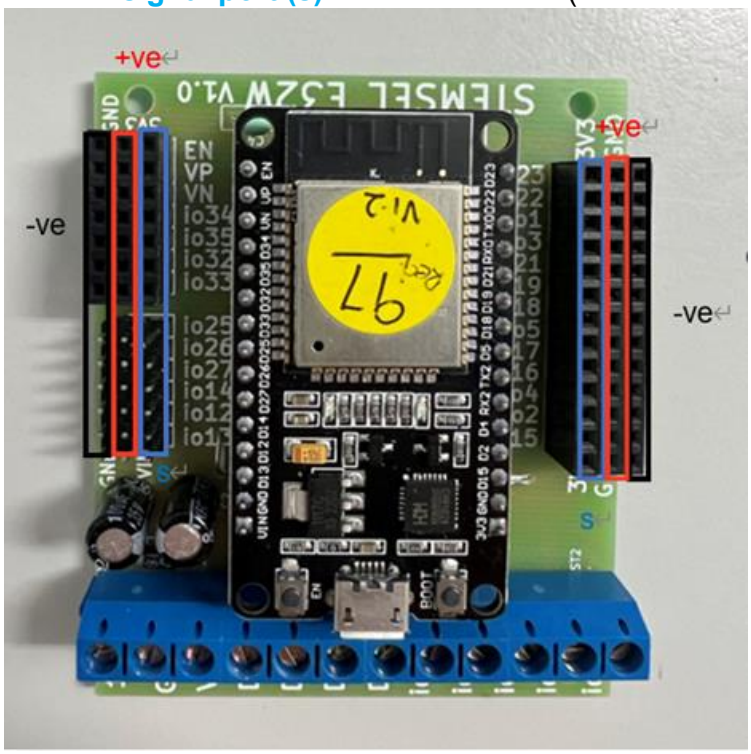

Figure 5: Negative, Positive and Signal port on the E32W board

There is one I/O part we are using for this project, a servo motor, their respective pins are shown in the figure below.

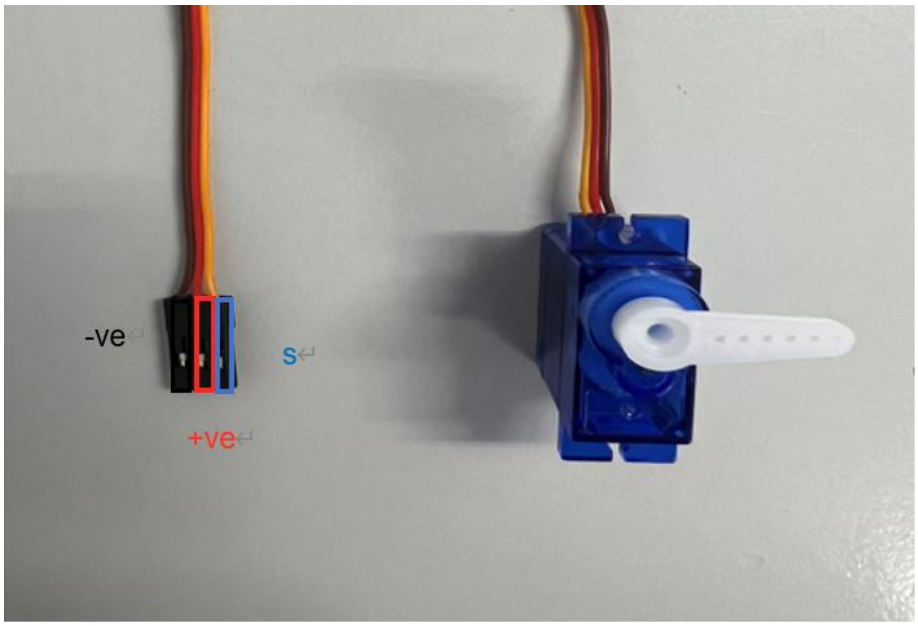

Figure 6: I/O part with negative, positive and signal pins indicated

Wiring instructions

a.) Plug in the servo motor to io13 on the E32W board.

b.) Make sure the (-ve) pin are on the GND (outer) side of the I/O ports.

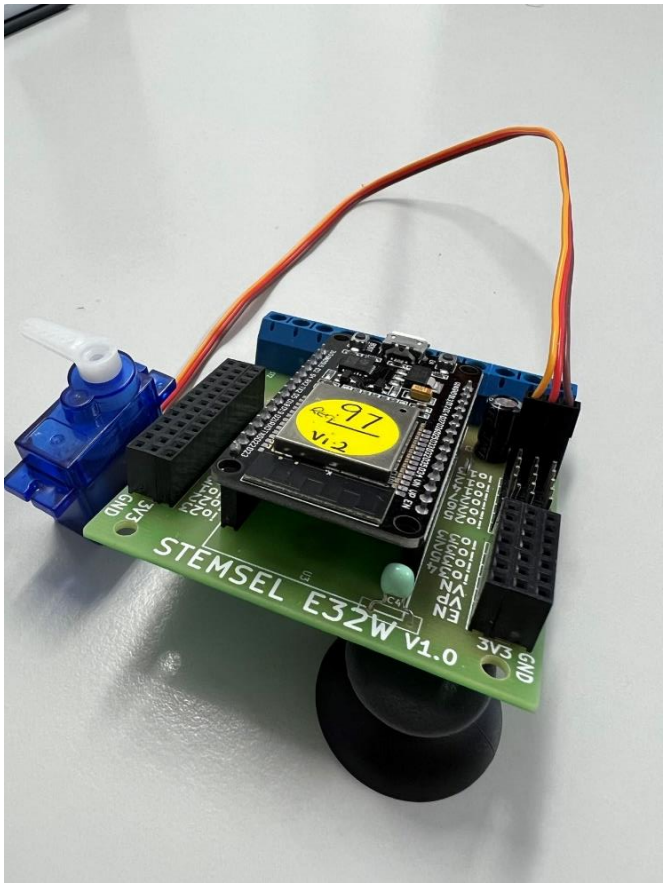

Figure 7: Circuit board connection with I/O part (side view)

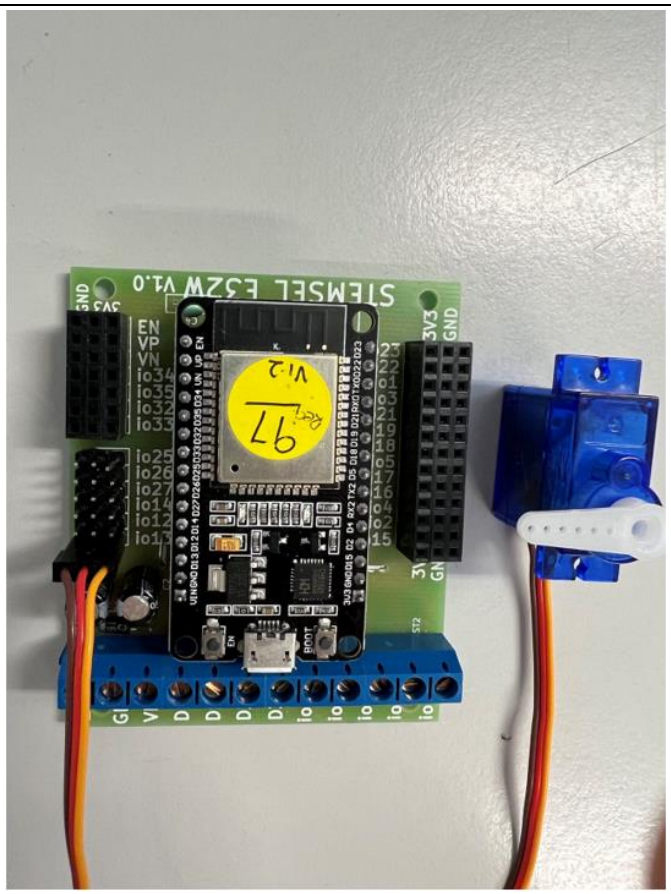

Figure 8: Circuit board connection with I/O part (top view)

Part C: Program the Circuit

HTML:

1. Under HTML, we will be displaying the title of the webpage, Servo Direction Buttons

```
<h2>Servo Direction Buttons</h2>
```

2. After creating the title, we will start to design three buttons. Firstly, we will create the button named 'SERVO 0'. The function of this button is to remain the angle between the pendulum rod and the horizontal line be 0 degree.

```
<button onclick="setServo(Servo1,0)">SERVO 0</button>
```

3. Then, the second button named 'SERVO 90' is to remain the angle between the pendulum rod and the horizontal line be 90 degrees(vertically).

```
<button onclick="setServo(Servo1,90)">SERVO 90</button>
```

4. Lastly, we need to generate a button named 'SERVO 180' to remain the angle between the pendulum rod and the horizontal line be 180 degrees

Final Code:

The final code for HTML block:

```
<h2>Servo Direction Buttons</h2>
<button onclick="setServo(Servo1,0)">SERVO 0</button>
<button onclick="setServo(Servo1,90)">SERVO 90</button>
<button onclick="setServo(Servo1,180)">SERVO 180</button>
```


Expected Result

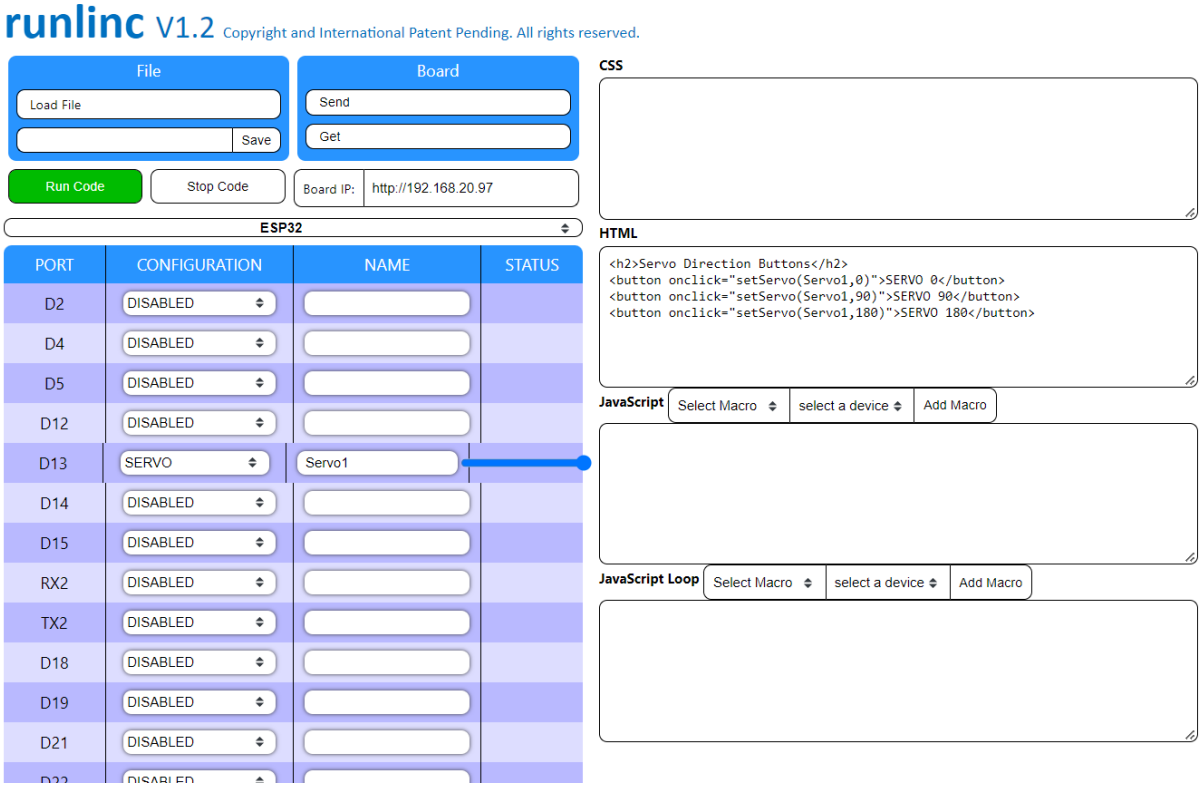

Figure 9: Expect runlinc result screenshot

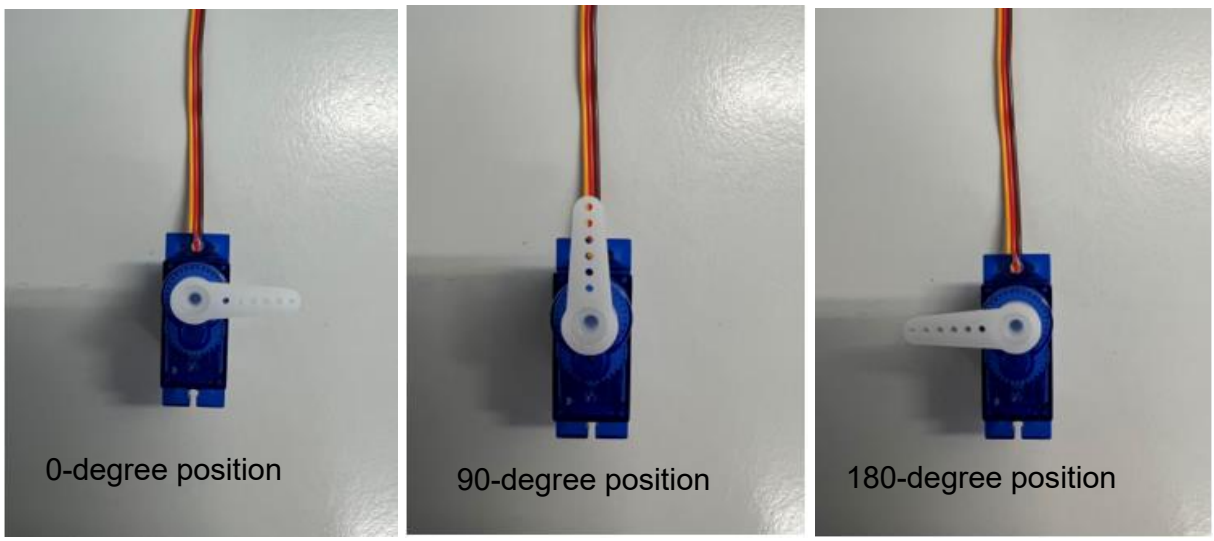

Figure 10: Expect performance of servo motor after clicking three buttons

Challenge

What if we want to control the servo motor using a Light Dependent Resistor (LDR) automatically rather than clicking buttons manually.

Part A: Design the Circuit on runlinc

Use the left side of the runlinc web page to construct an input/output (I/O).

For port D33 name it Sensor and set it as ANALOG_IN.

In our circuit design, we will be using the Light Dependent Resistor (LDR). We happen to have this in our kits, so these can be used inn our circuit design, as per the plan.

Figure 11: I/O configurations connections

Part B: Build the Circuit

Based on the E32W board equipped in the previous steps which is shown in the figure below.

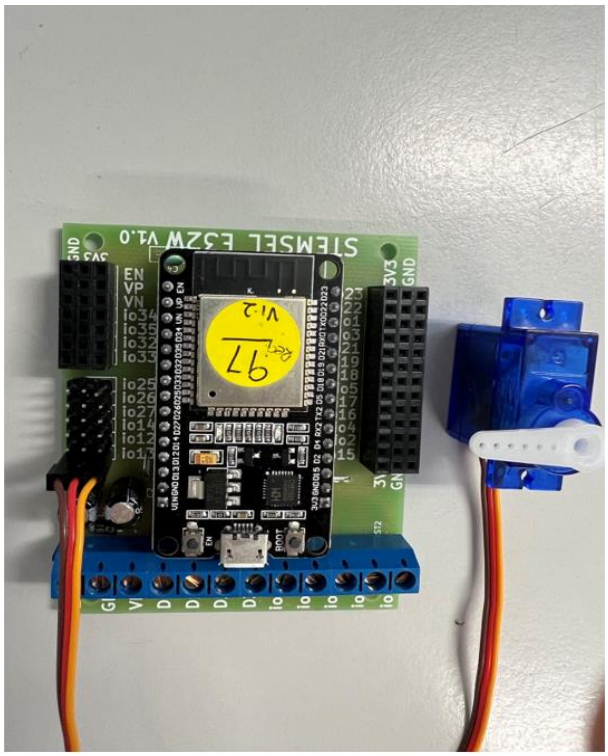

Figure 12: Circuit board connection with I/O part (top view)

We need a Light Dependent Resistor (LDR) module (KY-018) and their respective pins are shown in the figure below.

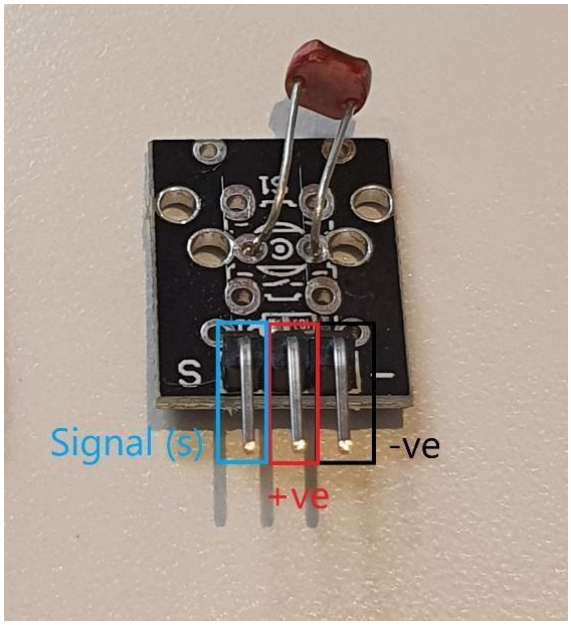

Figure 13: I/O part with negative, positive and signal pins indicated

Wiring instructions

a.) Plug in the Light Dependant Resistor (LDR) to io33 on the E32W board.
b.) Make sure the (-ve) pin are on the GND (outer) side of the I/O ports.

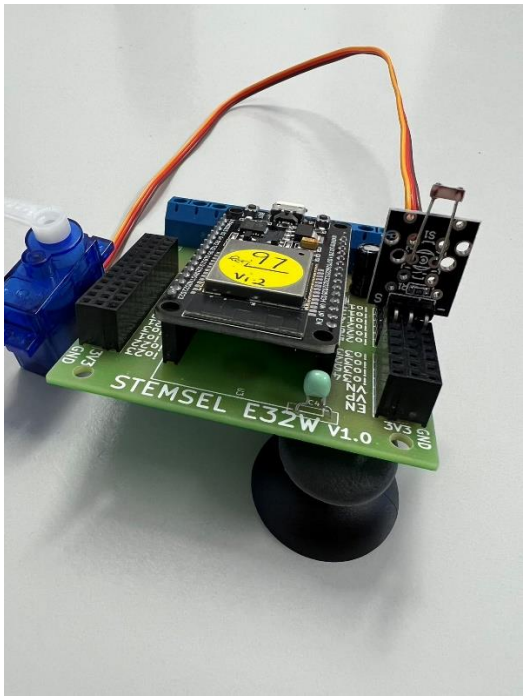

Figure 14: Circuit board connection with I/O part (side view)

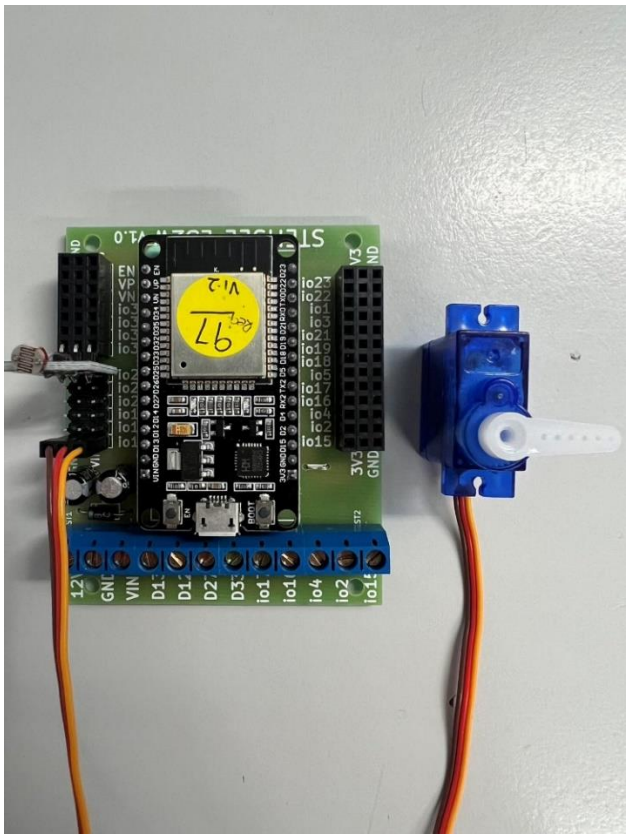

Figure 15: Circuit board connection with I/O part (top view)

Part C: Program the Circuit

Now delete all the codes in the HTML section. We don't need any buttons created before.

JavaScript Loop:

For JavaScript Loop type the following code:

```
if(analogIn( Sensor )>80)
setServo(Servo1,0);
else
setServo(Servo1,180);
await mSec( 1000 );
```

Then an 'if statement loop' is added to the Light Dependent Resistor (LDR) for it to decide when to rotate the pendulum rod. If the reading is above 80, the angle between the pendulum rod and the horizontal line remains 0 degree. However, if the reading is below 80, the angle between the pendulum rod and the horizontal line remains 180 degrees. Remember our Light Dependent Resistor (LDR) have a higher reading when its surrounding is darker, and a lower reading when its surrounding is brighter.

Expected Result

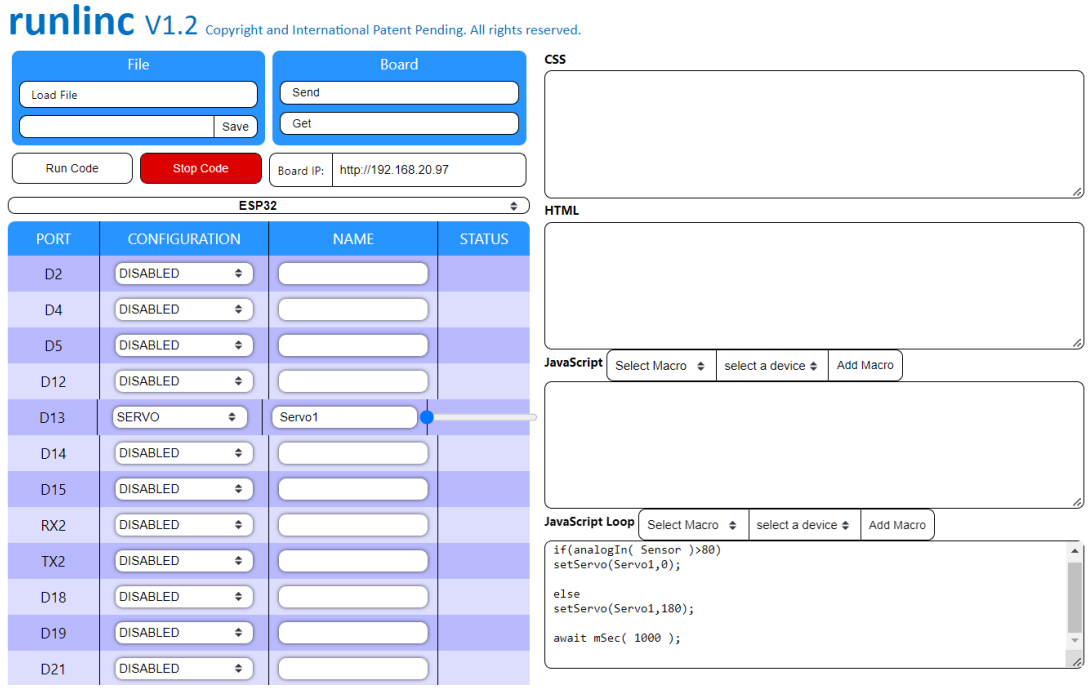

Figure 16 Expect runlinc result screenshot

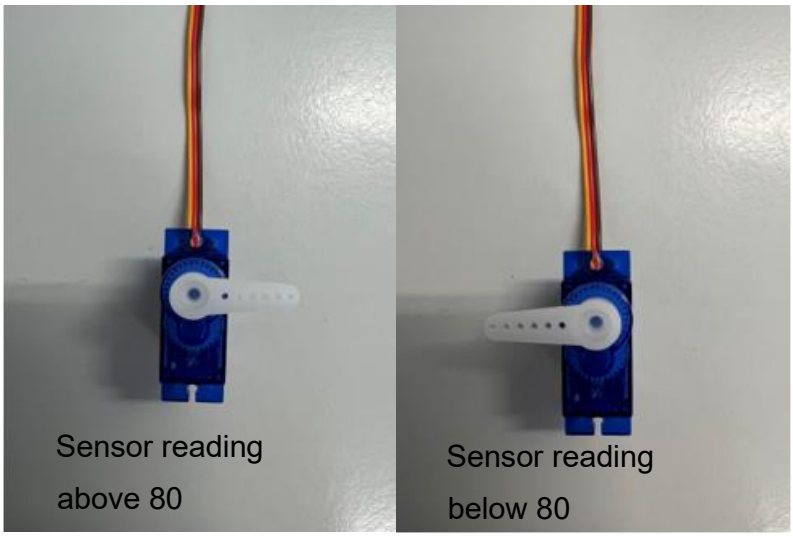

Figure 17 Expect pendulum rod position under different light intensity

Summary

In modern industrial manufacture, automation has been highly applied to Industrial production. In this process, servo motor is playing a more and more crucial role. In this project, we simulate a simple application of servo motor in terms of controlling the angle rotation of the pendulum rod.

