Encoders/Sensors

[Encoders/Sensors](#)

[FAQs](#)
- [What are encoders?](#)
- [Where are the sensors?](#)

[Two-Way Communication](#)

[From Electron App -> To Unity Webgl Build](#)

[From To Unity Webgl Build -> Electron App](#)
- [Calling the Function](#)

FAQs

What are encoders?

Encoders are essentially the interface used for communication between the Electron App and the Unity Webgl App embedded within the Electron App.

Motors and sensors use these communication pathways.

Where are the sensors?

All sensors are individual prefabs placed onto the robot prefabs.

Two-Way Communication

Refer to this Unity Doc for more details

The encoders use different interfaces depending on the direction of communication and the code is found in different locations.

From Electron App -> To Unity Webgl Build

Important Files

- UnityInstanceHandler.js (root folder of electron app)
- EncoderActionManager.cs (unity script)

The Encoder Action Manager script defines functions that the electron app can call. All the instance handler script needs to do is know the name of the game object that holds the script to call the functions.

EXAMPLE

In this example, the instance handler is sending the first motor of the robot a float value.

EncoderActionManager.cs

```
public void SetFrontLeft(float driveAmt)
{
      frontLeftWheel.driveAmount =
            new Vector3(-driveAmt * forceMultiplier,0,0);
}
```

unityInstanceHandlers.js

```
UnityInstance.SendMessage("JSAppIntegration","SetFrontLeft",motors[0]);
```

NOTE: Using this format

```
UnityInstance.SendMessage("GameObject Name","FunctionName", parameters);
```

From To Unity Webgl Build -> Electron App

Sensors send data from Unity to the Electron app. We use the following files to define the functions.

Important Files

- ElectronFunctions.jslib (found in Unity Assets/Plugins/)
- Index.html (found in Electron App /webgl/)

Index.html

The index.html defines the methods that the Unity app can call.

```
function setDistanceSensorData(distance) {
	setTouchSensorData(distance < 5);
	localStorage.setItem('distanceSensorReadings', "[[" + distance + "]]");
}
```

ElectronFunctions.jslib

The jslib file wraps the js functions with a c# function that Unity can call.

```
mergeInto(LibraryManager.library, {
	updateDistanceSensorData: function (Distance) {
	    setDistanceSensorData(Distance);
	}
});
```

Calling the Function

In order to call the function, you need to import the jslib function into a c# script. For the DistanceSensor.cs we do it this way:

```
using System.Runtime.InteropServices;

[DllImport("__Internal")]
private static extern void updateDistanceSensorData(float distance);
```

…then call the function as normal

```
private void FixedUpdate()
{
#if UNITY_WEBGL && !UNITY_EDITOR //do not call while testing in editor
	try
	{
		updateDistanceSensorData(distanceSensed);
	}
	catch { }
#endif
}
```