

## 17. Working with Signalk

OpenCPN process certain Signalk messages. If you wish, you can process Signalk messages in a JavaScript, which has a natural affinity with JSON.

The following script shows how to use the OpenCPN message interface to receive Signalk messages. This illustration script simply prints out certain information.

```
// Listen out for Signalk messages and display

Position = require("Position");
OCPNNonMessageName(received, "OCPN_CORE_SIGNALK"); // listen for message

function received(message){
    signalK = JSON.parse(message);
    // uncomment next line to pretty-print object
    // print(JSON.stringify(signalK, null, "\t"), "\n\n");
    update = signalK.update[0];
    timeStamp = update.timestamp;
    sentence = update.source.sentence;
    values = update.values;
    switch (sentence){
        case "GLL":
            position = new Position(values[0].value);
            print("Position at\t\t", timeStamp, " is\t",
                position.formatted, "\n");
            break
        case "VTG":
            cog = values[1].value;
            sog = values[2].value;
            print("Over ground at\t\t", timeStamp,
                " is\tCOG:", cog, "\tSOG:", sog, "\n");
            break;
        case "VHW":
            hdt = values[0].value;
            stw = values[1].value;
            print("Through water at\t", timeStamp,
                " is\tHDT:", hdt, "\tSTW:", stw, "\n");
            break;
        default:
    }
    OCPNNonMessageName(received, "OCPN_CORE_SIGNALK"); // listen again
};
```

Sample output:

```
Position at      2023-11-13T13:39:36.000Z is 49° 42.903'N 003° 35.039'W
Over ground at   2023-11-13T12:51:05.808Z is COG:123.4 SOG:2.34
Through water at 2023-11-13T12:51:05.814Z is HDT:125.2 STW:2.86
```