

Open Robotic Board  
ORB-WebView (Android)  
Javascript Interface

Thomas Breuer  
Hochschule Bonn-Rhein-Sieg  
Datum: 27.09.2023

# 1 Interface Methods

## 1.1 JS to App

The App provides an Interface, which can be used by the Javascript to send messages to the App. The message must be a JSON-formatted string.

Usage in Javascript:

```
OpenRoberta.jsToAppInterface( msg );
```

## 1.2 App to JS

The Javascript must provide an interface, which is used by the App to send messages to the Javascript. The message must be a JSON-formatted string:

```
function wvController()
{
    this.appToJsInterface = function( msg )
    {
        // evaluate msg;
        return true;
    }
}

var webViewController = new wvController();
```

## 2 JSON-formatted Messages

### 2.1 Internal

#### 2.1.1 Identify

JS to App:

```
{"target": "internal", "type": "identify"}
```

App to JS:

```
{"target": "internal", "type": "identify", "name": "OpenRoberta",  
"app_version": "1.0", "device_version": 26, "model": "SM-A320FL" }
```

#### 2.1.2 setRobot

JS to App:

```
{"target": "internal", "type": "setRobot", "robot": "orb" }
```

App to JS: No response

### 2.2 ORB Connection

#### 2.2.1 startScan

JS to App:

```
{ "target": "orb", "type": "startScan" }
```

App to JS:

Multiple answers, one per detected device.

```
{"target": "orb", "type": "scan", "state": "appeared",  
"brickid": "00:06:66:69:38:69", "brickname": "ORB-2 3869" }
```

#### 2.2.2 connect

JS to App:

```
{"target": "orb", "type": "connect", "robot": "00:06:66:69:38:69" }
```

App to JS::

```
{"target": "orb", "type": "connect", "state": "connected",  
"brickid": "00:06:66:69:38:69", "brickname": "ORB-2 3869" }
```

## 2.3 ORB Data

### 2.3.1 configToORB

JS to App:

```
{ "target": "orb", "type": "configToORB",  
  "data": { "Sensor": [ { "type": 0, "mode": 0, "option": 0 },  
                        { "type": 0, "mode": 0, "option": 0 },  
                        { "type": 0, "mode": 0, "option": 0 },  
                        { "type": 0, "mode": 0, "option": 0 } ],  
    "Motor": [ { "tics": 0, "acc": 0, "Kp": 0, "Ki": 0 },  
               { "tics": 0, "acc": 0, "Kp": 0, "Ki": 0 },  
               { "tics": 0, "acc": 0, "Kp": 0, "Ki": 0 },  
               { "tics": 0, "acc": 0, "Kp": 0, "Ki": 0 } ] } }
```

Parameter	Bedeutung	typisch
Motor.tics	Anzahl Encoder-Impulse pro Umdrehung	144
Motor.acc	Beschleunigung bzw. Verzögerung im Modus "Move-To"	50
Motor.Kp	Proportional-Faktor der PI-Regelung	50
Motor.Ki	Integral-Faktor der PI-Regelung	30

The app starts sending `propFromORB` periodically as long as the Javascript is running.

### 2.3.2 propToORB

JS to App:

```
{ "target": "orb", "type": "propToORB",  
  "data": { "Motor": [ { "mode": 0, "speed": 0, "pos": 0 },  
                      { "mode": 0, "speed": 0, "pos": 0 },  
                      { "mode": 0, "speed": 0, "pos": 0 },  
                      { "mode": 0, "speed": 0, "pos": 0 } ],  
    "Servo": [ { "mode": 0, "pos": 0 },  
               { "mode": 0, "pos": 0 } ] } }
```

Parameter	Bedeutung
Motor.mode	0: POWER_MODE Die durch speed angegebene Spannung wird eingestellt
	1: BRAKE_MODE Bremsbetrieb durch Kurzschluss des Motors
	2: SPEED_MODE Die durch speed angegebene Drehzahl wird geregelt
	3: MOVETO_MODE

	Die durch pos angegebene Motor-Position wird angefahren, die in speed angegebene Geschwindigkeit wird dabei nicht überschritten.
Motor.speed	Spannung im Bereich -1000 bis 1000 (Einheit: 1/1000 der Versorgungsspannung) ODER Geschwindigkeit in 1/1000 Umdrehungen/Sekunde
Motor.pos	Absolute Position in 1/1000 Umdrehungen

The app starts sending `propFromORB` periodically as long as the Javascript is running.

### 2.3.3 propFromORB

App to JS:

```
{ "target": "orb", "type": "propFromORB",
  "data": { "Motor": [ { "pwr": 0, "speed": 0, "pos": 0 },
                      { "pwr": 0, "speed": 0, "pos": 0 },
                      { "pwr": 0, "speed": 0, "pos": 0 },
                      { "pwr": 0, "speed": 0, "pos": 0 } ],
    "Sensor": [ { "valid": false, "type": 0, "option": 0, "value": [ 0, 0 ] },
                { "valid": false, "type": 0, "option": 0, "value": [ 0, 0 ] },
                { "valid": false, "type": 0, "option": 0, "value": [ 0, 0 ] },
                { "valid": false, "type": 0, "option": 0, "value": [ 0, 0 ] } ],
    "Vcc": 0, "Digital": [ false, false ], "Status": 0 } }
```

Parameter	Bedeutung
Motor.pwr	Motor-Spannung im Bereich -100 bis 100 (Einheit: 1/100 der Versorgungsspannung)
Motor.speed	Gemessene Geschwindigkeit in 1/1000 Umdrehungen/Sekunde
Motor.pos	Gemessene absolute Position in 1/1000 Umdrehungen

## 2.4 ORB Settings

### 2.4.1 settingsToORB

JS to App:

```
{ "target": "orb", "type": "settingsToORB",
  "data": { "update": false, "clearMemory": false,
    "Name": "myORB", "VCC_ok": 7.5, "VCC_low": 7.1 } };
```

If update is true, the given setting is stored in ORB flash. In any case, the App answers with a `settingsFromORB` message.

## 2.4.2 settingsFromORB

App to JS:

```
{ "target": "orb", "type": "settingsFromORB",  
  "data": { "Version": [0,0], "Board": [0,0], "Name": "test",  
            "VCC_ok": 7.5, "VCC_low": 7.1 } };
```

## 2.5 Android-Sensorik

### 2.5.1 CommandToAS

JS to App:

```
{ "target": "orb", "type": "commandToAS", "data": { "cmd": "resetSensor" } }
```

### 2.5.2 configToAS

JS to App:

```
{ "target": "orb", "type": "configToAS",  
  "data": { "name": "Umgebungslicht", "type": 5 } }
```

### 2.5.3 sensorFromAS

App to JS:

```
{ "target": "orb", "type": "sensorFromAS",  
  "data": { "Umgebungslicht": [123], "Schwerkraft": [1,1,9.81] } }
```