BrightSign®

TECHNICAL NOTES

Using Javascript Objects for BrightScript

BrightSign, LLC. 16795 Lark Ave., Suite 200 Los Gatos, CA 95032
408-852-9263 | www.brightsign.biz

INTRODUCTION

Starting with firmware version 4.7, BrightSign’s Webkit implementation exposes several Javascript objects for BrightScript. These objects allow you to link many standard interactive events and hardware elements (serial, CEC, device info, etc.) to HTML5 pages. This tech note details the methods and parameters of each Javascript object. For more information about the BrightScript objects that the Javascript objects are linked to, see the BrightScript Object Reference Manual.

TABLE OF CONTENTS

BSControlPort

For more information about available methods, refer to the entry on *roControlPort*.

Methods

```
boolean SetOutputValue(in unsigned long Param);
boolean SetOutputValues(in unsigned long Param1,
                        in unsigned long Param2,
                        in unsigned long Param3,
                        in unsigned long Param4);
boolean SetPinValue(in unsigned long Pin,
                    in unsigned long Param);
```

Events

The following events are available on the *BSControlPort* object. Each event can receive a `ControlPortEvent` event.

```
oncontroldown
oncontrolup
oncontrolevent
```

Example

The following Javascript example causes the LEDs on a BP900 button board to twinkle.

```
function myFunction()
{
    var bp900_setup = new BSControlPort("TouchBoard-0-LED-SETUP");
    bp900_setup.SetPinValue(0, 11)

    var bp900 = new BSControlPort("TouchBoard-0-LED");
    bp900.SetPinValue(0, 0x07fe)
    bp900.SetPinValue(1, 0x07fd)
    bp900.SetPinValue(2, 0x07fb)
    bp900.SetPinValue(3, 0x07f7)
    bp900.SetPinValue(4, 0x07ef)
    bp900.SetPinValue(5, 0x07df)
    bp900.SetPinValue(6, 0x07bf)
    bp900.SetPinValue(7, 0x077f)
    bp900.SetPinValue(8, 0x06ff)
    bp900.SetPinValue(9, 0x05ff)
    bp900.SetPinValue(10, 0x03ff)

    var bp900_gpio = new BSControlPort("TouchBoard-0-GPIO");
    bp900_gpio.oncontroldown = function(e)
    {
        console.log('###### oncontroldown' + e.code);
    }
}
```

ControlPortEvent – Attributes

```
readonly attribute unsigned long code;
```

BSDeviceInfo

For more information about available methods, refer to the entry on *roDeviceInfo*.

Attributes

```
readonly attribute DOMString model;
readonly attribute DOMString version;
readonly attribute int deviceUptime;
readonly attribute int deviceLifetime;
readonly attribute int deviceBootCount;
readonly attribute DOMString bootVersion;
readonly attribute DOMString deviceUniqueId;
readonly attribute DOMString family;
```

Methods

```
int VersionCompare(in DOMString version);
int BootVersionCompare(in DOMString version);
boolean HasFeature(in DOMString feature);
```

Example

The following Javascript example posts device information on the page when the button is clicked.

```
function deviceInfo()
{
    var device_info = new BSDeviceInfo();

    document.getElementById("modelText").innerHTML = device_info.model;
    document.getElementById("versionText").innerHTML =
device_info.version;
    document.getElementById("bversionText").innerHTML =
device_info.bootVersion;
    document.getElementById("serialText").innerHTML =
device_info.deviceUniqueId;
    document.getElementById("familyText").innerHTML = device_info.family;

    document.getElementById("uptime").innerHTML =
device_info.deviceUptime;
    document.getElementById("lifetime").innerHTML =
device_info.deviceLifetime;
```

```
    document.getElementById("bootcount").innerHTML =
device_info.deviceBootCount;

    if(device_info.VersionCompare("4.7.36") > 0)
    {
        document.getElementById("version1").innerHTML = "Version > 4.7.36"
    }
    else
    {
        document.getElementById("version1").innerHTML = "Version <=
4.7.36"
    }

    if(device_info.HasFeature("Six Channel Audio"))
    {
        document.getElementById("feature").innerHTML = "6 Channel Audio
Available"
    }
    else
    {
        document.getElementById("feature").innerHTML = "6 Channel Audio
NOT Available"
    }
}
```

BSIRReceiver

This class receives IR events. For more information, refer to the entry on *roIRRemote*.

Events
These events are available on the BSIRReciever object. Each event can receive an `IRReceiverEvent` event.

Example
The following Javascript example displays messages on the log when receiving remote codes.

```
function myFunction()
{
    var ir_receiver = new BSIRReceiver();

    ir_receiver.onremotedown = function(e){
        console.log('############ onremotedown: ' + e.irType + " - " +
e.code);
    }

    ir_receiver.onremoteup = function(e){
        console.log('############ onremoteup: ' + e.irType + " - " +
e.code);
    }
}
```

IRReceiverEvent – Attributes

```
readonly attribute DOMString irType;
readonly attribute unsigned long code;
```

BSIRTransmitter

For more information about available methods, refer to the entry on *roIRRemote*.

Methods

```
boolean Send(in DOMString Type, in unsigned long Code);
boolean SetSendPolarity(in boolean Polarity);
```

Example

The following Javascript example sends the indicated IR codes when the corresponding functions are called.

```
<script>
    var irTransmitter = new BSIRTransmitter();

    function irCode1()
    {
        console.log('############ irCode1');
        irTransmitter.Send("NEC", 65284);
    }

    function irCode2()
    {
        console.log('############ irCode2');
        irTransmitter.Send("NEC", 65288);
    }

    function irCode3()
    {
        console.log('############ irCode3');
        irTransmitter.Send("NEC", 65290);
    }
</script>
```

BSVideoMode

For more information about available methods, please refer to the entry on *roVideoMode*. If you'd like to change the video mode of the player, you will need to use BrightScript instead of this Javascript class.

Methods

```
readonly attribute int resX;
readonly attribute int resY;
readonly attribute int safeX;
readonly attribute int safeY;
readonly attribute int safeWidth;
readonly attribute int safeHeight;
readonly attribute DOMString mode;

boolean IsAttached(in DOMString connector);
DOMString GetBestMode(in DOMString connector);
boolean SetBackgroundColour(in unsigned long rgb);
boolean SetBackgroundColour(in unsigned long r,
                            in unsigned long g,
                            in unsigned long b);
boolean HdmiAudioDisable(in boolean disable);
```

Example

The following Javascript example illustrates how to retrieve information about the current video mode.

```
    function fillInVideoData()
    {
        var videomode_info = new BSVideoMode();
        document.getElementById("resX").innerHTML = videomode_info.resX;
        document.getElementById("resY").innerHTML = videomode_info.resY;
        document.getElementById("safeX").innerHTML = videomode_info.safeX;
        document.getElementById("safeY").innerHTML = videomode_info.safeY;
        document.getElementById("safeWidth").innerHTML =
videomode_info.safeWidth;
        document.getElementById("safeHeight").innerHTML =
videomode_info.safeHeight;
        document.getElementById("videoMode").innerHTML =
videomode_info.mode;
        document.getElementById("bestMode").innerHTML =
videomode_info.GetBestMode("hdmi");

        document.getElementById("connectedFlag").innerHTML =
videomode_info.IsAttached("vga");
    }

    function changeBackground()
    {
        var videomode_info = new BSVideoMode();
        videomode_info.SetBackgroundColour(0xFF0000);
    }
```

BSCECTransmitter

For more information about available methods, refer to the entry on *roCecInterface*. Note that you can only use this Javascript class to send CEC messages.

Methods

```
boolean SendRawMessage(in ArrayBuffer data);
boolean SendRawMessage(in ArrayBufferView data);
boolean SendRawMessage(in DOMString data);
```

Example

The following Javascript example shows how to send a set of CEC messages.

```
    function cecDisplayOn()
    {
        console.log("### cecDisplayOn ###");

        var cec_control = new BSCECTransmitter();

        var buffer = new Uint8Array(2);
        buffer[ 0 ] = 0x40;
        buffer[ 1 ] = 0x0D;

        cec_control.SendRawMessage(buffer);
    }
    function cecDisplayOff()
    {
        console.log("### cecDisplayOff ###");

        var cec_control = new BSCECTransmitter();

        var buffer = new Uint8Array(2);
        buffer[ 0 ] = 0x40;
        buffer[ 1 ] = 0x36;

        cec_control.SendRawMessage(buffer);
    }
```

BSSerialPort

For more information about available methods, refer to the entry on *roSerialPort*.

Methods

```
void SetBaudRate(in unsigned long baudRate) raises(DOMException);
void SetDataBits(in unsigned long dataBits) raises(DOMException);
void SetStopBits(in unsigned long stopBits) raises(DOMException);
void SetParity(in DOMString parity) raises(DOMException);

boolean SetEcho(in boolean flag);
boolean SetInverted(in boolean flag);

void SetGenerateByteEvent(in boolean flag) raises(DOMException);
void SetGenerateLineEvent(in boolean flag) raises(DOMException);

void SetLineEnding(in DOMString eol) raises(DOMException);

boolean SendByte(in unsigned long byte);
boolean SendBytes(in ArrayBuffer data);
boolean SendBytes(in ArrayBufferView data);
boolean SendBytes(in DOMString data);

boolean SendBreak(in long duration_ms);
void Flush() raises(DOMException);
```

Events

The following events are available via the *BSSerialPort* object. Each event can receive a `SerialPortEvent` event.

```
onserialbyte
onserialline
```

Example

```
function serialOut()
{
    console.log("*** serialOut **");

    // '2' is the first externally connected USB  port on Cheetah
    var serial_out = new BSSerialPort(2);

    serial_out.SetBaudRate(115200);
    serial_out.SetDataBits(8);
    serial_out.SetStopBits(1);
    serial_out.SetParity("none");
    serial_out.SetEcho(true);

    serial_out.SetGenerateByteEvent(true);
    serial_out.SetGenerateLineEvent(true);

    serial_out.onserialbyte = function(e){
        console.log('### onserialbyte: ' + e.byte);
    }

    serial_out.onserialline = function(e){
        console.log('### onserialline: ' + e.data);
    }

    serial_out.SendByte(89);
    serial_out.SendByte(90);
    serial_out.SendByte(91);

    serial_out.SendBytes('Hello World!');
    serial_out.SendBytes(String.fromCharCode(64, 27, 66, 67))
```

SerialPortEvent – Attributes

For the `onserialbyte` event:

```
readonly attribute unsigned long byte;
```

For the `onserialline` event:

```
readonly attribute DOMString data;
```