Project Editor 1.4 Javascript API

Revision History

1.3 4/19/11
Added “getScriptDeviceRefByName”
Added Controller objects and methods
Added “didRotate” events

1.2 3/25/11
Added “changeWebImgSource”
Added “doPleaseWait”
Added “debugPrint”

1.1 3/10/11
Added “getStoredFeedbackData”
Added “getJSON”

Introduction

With the release of Project Editor 1.4, users can add custom functionality to BitWise Touch GUIs via our new Javascript API. There are many resources available online to learn about writing javascript. One great place to start is http://www.w3schools.com/js/default.asp , which contains a complete reference to all standard javscript objects and methods, and provides some useful tutorials for common javascript tasks.

Dynamic GUI Content

These functions can be used to easily modify GUI appearance at runtime.

setFeedbackData(FBID,DATA)
Stores and displays data (string or int)DATA on any GUI elements with Feedback ID (string) FBID .

getStoredFeedbackData(FBID)
Returns stored (string)data on GUI elements with Feedback ID (string) FBID

setButtonState(BUTTON,STATE)
Sets GUI button with name (string)BUTTON to state (bool)STATE

changeButtonLabelText(BUTTON,TEXT)
Sets GUI button text with name (string)BUTTON to text (string)TEXT

changeButtonLabelColor(BUTTON,COLOR)
Sets GUI button text color with name (string)BUTTON to color (string)COLOR
changeLabelText(LABEL,TEXT)
Sets GUI label text with name (string)LABEL to text (string)TEXT
changeLabelColor(LABEL,COLOR)
Sets GUI label text color with name (string)LABEL to color (string)COLOR
changeWebImgSource(IMAGE,SOURCE)
Sets GUI Web Image with name (string)IMAGE to have source(string)SOURCE
setSliderValueByName(NAME,VALUE)
Sets GUI Slider with name (string)NAME to value(string or int)VALUE)
setSliderValueByFBID(FBID,VALUE)
Sets GUI Slider with Feedback ID (string)FBID to value (string or int)VALUE

Commands & Macros

These functions can be used to execute BC4X1 commands and macros.
sendCommand(IP,CMD)
Sends command protocol formatted command(string)CMD to BC4 at IP Address (string)IP. See Command Protocol documentation for more information about properly formatted bwc commands.
doMacroByName(IP,NAME)
Run macro with name (string)NAME on BC4 at IP Address (string)IP
doMacroByNumber(IP,NUMBER)
Run macro with number (string)NUMBER on BC4 at IP Address (string)IP

Script Devices

Script Devices can also be referenced by user javascript, giving the user direct access to any functions defined in the Script Device.

getScriptDeviceRefByName(NAME)
Returns a reference to the Script Device with name (string)NAME. This reference can be used to directly call Script Device functions from user javascript, using the defined function names and parameters

BC4 Controllers
Generating GUI files will result in an array of "Controller" objects. Each Controller object relates to a BC4 controller in the project file. The following functions can be used to obtain a reference to a Controller object.

getControllerWithMAC(MAC)
Returns a reference to the Controller object with MAC address (string) MAC.

getControllerWithIP(IP)
Returns a reference to the Controller object with IP address (string) IP.
After obtaining a reference to a Controller object, you can make use of the following methods it contains. Controllers objects are contained in the zero-based "Controllers" array, with the first Controller in the project file having index 0. Thus, each controller in the project file could be also be accessed by its index in that array, such as "Controllers[0]" for the first controller in the project. Each Controller object has the following methods:

.activateRelay(RELAYNUM)
Activates Relay number (string) RELAYNUM

.deActivateRelay(RELAYNUM)
Deactivates Relay number (string) RELAYNUM

.setUserVariable(VARNUM,VALUE)
Sets User Variable (string) VARNUM to value (string) VALUE on the controller

.incrementUserVariable(VARNUM)
Increments User Variable (string) VARNUM on the controller

.decrementUserVariable(VARNUM)
Decrements User Variable (string) VARNUM on the controller

.queryUserVariableState(VARNUM)
Tells controller to report current value of User Variable (string) VARNUM

.queryRelayState(RELAYUM)
Tells controller to report current value of Relay (string) RELAYNUM

.queryOutputState(GPIONUM)
Tells controller to report current value of GPIO (string)GPIONUM that has been configured as a digital output

.queryDigitalInputState(GPIONUM)
Tells controller to report current value of GPIO (string)GPIONUM that has been configured as a digital input

.queryAnalogInputState(GPIONUM)
Tells controller to report current value of GPIO (string)GPIONUM that has been configured as an analog input.

.getRelayFBID(RELAYNUM)
Returns the dynamically generated Feedback ID of Relay (string) RELAYNUM

.getGPIODigitalInputFBID(GPIONUM)
Returns the dynamically generated Feedback ID of GPIO (string)GPIONUM that has been configured as a digital input

.getGPIODigitalOutputFBID(GPIONUM)
Returns the dynamically generated Feedback ID of GPIO (string)GPIONUM that has been configured as a digital output

.getGPIOAnalogInputValFBID(GPIONUM)
Returns the dynamically generated “VAL” Feedback ID of GPIO (string)GPIONUM that has been configured as an analog input

.getGPIOAnalogInputMinFBID(GPIONUM)
Returns the dynamically generated “MIN” Feedback ID of GPIO (string)GPIONUM that has been configured as an analog input

.getGPIOAnalogInputMaxFBID(GPIONUM)
Returns the dynamically generated “MAX” Feedback ID of GPIO (string)GPIONUM that has been configured as an analog input

.getGPIO0CountFBID()
Returns the dynamically generated Feedback ID of GPIO0 that has been configured as a counter

.getGPIO0FreqFBID()
Returns the dynamically generated Feedback ID of GPIO0 that has been configured as a frequency counter

.getTemperatureADValueFBID()
Returns the dynamically generated raw AD value Feedback ID of the BC4's built-in temperature sensor

.getTemperatureCValueFBID()
Returns the dynamically generated Celcius value Feedback ID of the BC4's built-in temperature sensor

.getTemperatureFValueFBID()
Returns the dynamically generated Fahrenheit value Feedback ID of the BC4's built-in temperature sensor

Events
Each Event function call must only be implemented once per GUI Group. If implemented, these functions will be called by the app when the corresponding event occurs.

onAppStart()
Called when the app is first launched and a page has been loaded. Also called when app wakes from sleep. This is a useful place to call a macro which will poll two-way systems for their current status.

onPageLoad(PAGENAME)
Called when any page is loaded, page loaded was page with name (string)PAGENAME

didRotateToPortrait()
Called when the device has been rotated to the portrait orientation.

didRotateToLandscape()
Called when the device has been rotated to the landscape orientation.