



## Project Editor 1.6 User's Guide

### Getting Started

Project Editor Software from BitWise Controls is used by professional Installers to create and manage project files for BitWise Controls hardware installations.

In addition to this User Guide, there are numerous tutorial videos, tech notes, and other training materials located at [www.bitwisecontrols.com/dealers](http://www.bitwisecontrols.com/dealers)

Setting up a new project consists of five "phases".

**Phase 1: Add BitWise Hardware:** Add networked BitWise controllers (such as the BC1 BC4X1 Pro controller) to the project, via automatic discovery, or by manually entering the unit's address info.

**Phase 2: Set Up Project-Level Settings:** This includes location information (required for projects using BC1 scheduled events) and email settings (required for

**Phase 2: Add Connected Devices:** Create and test connected equipment, such as IR, RS232, and Socket (TCP/UDP) devices.

**Phase 3: Create Macros and other programming:** Create, test, and upload control macros (series of commands to be executed, such as “Watch DVD”). Other programming tasks for BC1 projects may include programming scheduled events, macro triggers, etc.

**Phase 4: Create/Edit GUIs:** Create and upload GUI (graphical user interface) pages. These files are created for use with our BitWise Touch app for iPad/iPhone/iPod Touch, and can also be generated as html files for use with other web enabled devices.

## Project Editor Main Window

The main Project Editor window has four main sections.

### Main Menu

The Main Menu bar runs across the top of the main window. The main menu contains items for project file management, project uploading, firmware updating and more.

### Project Tree

On the left hand side of the window, is the Project Tree. This tree contains all hardware, connected equipment, and GUI pages in the project. Items can be added, deleted, and edited by right-clicking the item and selecting the desired action from a pop-up menu.

### Properties Pane

The upper right-hand pane is the Properties pane. When you select an item in the Project Tree, the Properties pane will display any relevant properties for that item.

### Macro Editor

The lower right-hand pane is the Macro Editor. When creating or editing a macro, this is where you will drag/drop device action lines to create your control macros.

## Project Files

### Creating a New Project

In the main menu, click File->New Project. You may be prompted to save the previous file before creating a new one. Once the new file has been created, click File->Save As, and enter a new file name for this project.

Now you are ready to add your BitWise hardware to the project. In the Project Tree, expand the “PROJECT” node. You should see three child nodes, “BC4s”, “BC1s”, and “GUIs”

## Opening an Existing File

In the main menu, click File->Open. Browse to the desired project file.

## Opening a Recent File

In the main menu, click File->Open Recent. Select from the list of recently opened project files.

## Save a File

In the main menu, click File->Save to save the current project File. Select Save As to save the current project file under a new file name.

**Note\*\***Project Editor 1.6 supports a number of different project file formats. The “.bwe” file extension includes all required GUI image files inside the project file, which makes projects much more portable. This is the default New Project file type, and is the recommended file type for most new projects. The .bwp project file format is a legacy format used in earlier versions of Project Editor. A .bwp can be converted to the new .bwe format by doing a simple “Save As” and selecting the new extension. Note that all image file references must be intact for the images to be included in the .bwe file. The “.bws” extension is the same as a .bwe, but is password protected. Protected project files will not allow GUI or Device exporting without supplying the password.

# Adding BitWise Hardware

## Auto Discovery

Right-click on the “BC4s” or “BC1s” node in the Project Tree. You will be presented with the option to either add hardware from the network (auto-discovery), or to add hardware manually.

Click “Add Controller from Network” to use auto-discovery to find any BitWise controllers on the network. The Controller Discovery window will appear. To add a unit to the project, select it from the list of discovered hardware, and click “Add to Project”. Note that if the project already contains a unit with the same MAC address, you will not be able to add the unit. Note that the discovery window will discover both BC1 and BC4 hardware.

The Controller Discovery window has the following buttons:

**Discover:** Click this to refresh the list of hardware available on the network.

**Locate:** Click this to put the selected unit into “Locate” mode, which will start blinking the red Status LED on the selected unit. This is useful for identifying hardware in a multi-unit installation. Click again to stop the Locate mode.

**Web Config:** Click this to use the controller's internal web configuration pages. These pages can be used to set up the unit's IP settings. The username for the web configuration is "admin", and the password is "bitwise".

**Add to Project:** This button will add the selected controller to the project tree in the appropriate controller group ("BC4s" or "BC1s").

**Done:** After you have finished adding controllers to the project, click "Done" to close the BC4 Discovery window.

## Manual

Right-click on the "BC4s" or "BC1s" node in the Project Tree. You will be presented with the option to either add hardware from the network (auto-discovery), or to add hardware manually.

Click "Add Controller Manually". Enter the MAC and IP addresses of the controller you want to add to the project. Click "Add" to add the unit to the project and close the Manual window. If you are manually adding a BC1 controller, make sure the "Is BC1" box is checked before adding the controller.

# Controller Properties (BC4)

## BC4 Properties

To change BC4 IP settings, right-click the desired BC4 node in the Project Tree, and select "Properties". The BC4 Properties window will open. Here you can make changes to all of the unit's network settings.

Click "Get Settings from BC4" to poll the unit for the actual settings stored in the device, which may be different from the values stored in the project file.

Click "Apply Settings to BC4" to apply the project values to the unit.

Click "Compare Project with Network" to view the differences between the project values and the values stored in the unit. Any differing fields will be highlighted in yellow, and the differences will appear in the Warning list.

## Temperature Sensor

The BC4 hardware has a built-in temperature sensor. The state of this temperature sensor can be evaluated in macro lines. To modify the temperature sensor settings, expand the BC4 node in the Project Tree, and right-click the TEMPERATURE node, then select "Properties" to open the Temperature Properties window.

There are two settings which can be adjusted.

**Delay:** Can be set from 0-200 (tenths of a second). This affects how often the BC4 will send asynchronous temperature reports to a connected third-part hardware or software system.

**Offset:** Can be set from 0-255. The offset value affects the raw AD value of the temperature sensor, and can be used to make the reported value more closely reflect ambient temperature, as opposed to the internal temperature of the unit.

Click the “Apply” button to apply the currently displayed settings to the unit. Click the “Refresh” button to get the actual settings from the unit. Click the “Monitor Activity” button to view real-time incoming temperature reporting from the unit.

## Controller Properties (BC1)

### BC1 Properties

To change BC1 IP settings, right-click the desired BC1 node in the Project Tree, and select “Properties”. The BC1 Properties window will open. Here you can make changes to all of the unit’s network settings.

Click “Get Settings from BC1” to poll the unit for the actual settings stored in the device, which may be different from the values stored in the project file.

Click “Apply Settings to BC1” to apply the project values to the unit.

Click “Compare Project with Network” to view the differences between the project values and the values stored in the unit. Any differing fields will be highlighted in yellow, and the differences will appear in the Warning list.

### Time/Date Settings

The BC1 controller has an internal clock, which must be properly set in order for scheduled events to function.

These are the settings, which can be adjusted.

**NTP Server:** The address of a Network Time Protocol server, which can be used to synchronize the BC1’s clock.

**Time Zone:** Select the desired Time Zone profile to use. These are Unix-style “Zone Info” profiles, which contain information about a city or region. Selecting the proper profile will ensure that the BC1 uses the correct Time Zone and Daylight Savings Time information.

**Time:** The current time of the BC1’s clock. This entry will display the time of the programming PC until click the “Get” button.

**Date:** The current date of the BC1’s clock. This entry will display the date of the programming PC until click the “Get” button.

Click the “Set” button to apply the currently displayed settings to the unit. Click the “Get” button to display the current settings from the unit.

## Connected Devices

Each BC1 or BC4 controller can be connected to numerous external devices. All of these hardware connections are available in the Project Tree within the CONNECTED DEVICES node of the controller. To create and set up connected devices, expand the BC4 or BC1 node in the Project Tree, then expand the CONNECTED DEVICES node.

### Serial Port (BC4)

The BC4 has 1 serial port. To change serial port settings, right-click the SERIALPORT node in the Project Tree, then click “Properties”. The serial port can be configured to work in four different modes.

**Normal (Two-Way TCP):** This mode means that external systems can connect to the defined TCP port number, and operate in a pass-through two-way mode.

**Processes Host Commands:** In this mode, the unit will respond to the BitWise Command Protocol (normally reserved for TCP and UDP communications with external systems) over the serial port.

**Serial Matching:** In this mode, the unit will attempt to match incoming serial data against user-defined “match strings”. Upon a successful match, the unit will trigger the control macro associated with that match string.

**GUI Two Way(RS232):** In this mode, the unit will pass any incoming serial data to any client devices running BitWise Touch, for use with two-way Script Devices.

**GUI Two Way(TCP):** In this mode, the unit will pass any incoming TCP Client data to any client devices running BitWise Touch, for use with two-way Script Devices.

Choose the desired baud rate from the “Baud” drop down list. If applicable, enter the desired port number for TCP communications.

Click the “Apply” button to apply settings to the unit. Click the “Refresh” button to get settings from the unit.

## Serial Port (BC1)

The BC1 has 4 serial ports. To change serial port settings, right-click the SERIALPORT node in the Project Tree, then click “Properties”. The serial port can be configured to work in four different modes.

**Normal (Two-Way TCP):** This mode means that external systems can connect to the defined TCP port number, and operate in a pass-through two-way mode. This is also the desired setting for two-way Script Devices connected to the BC1.

**Processes Host Commands:** In this mode, the unit will respond to the BitWise Command Protocol (normally reserved for TCP and UDP communications with external systems) over the serial port.

Choose the desired baud rate, parity, databits and stopbits selections from the appropriate drop down lists. If applicable, enter the desired port number for TCP communications. In most cases, this port number should be left as-is.

Click the “Apply” button to apply settings to the unit. Click the “Refresh” button to get settings from the unit.

## GPIOs (BC4)

The BC4 has three GPIO (General Purpose Input/Output) ports. GPIO1 and GPIO2 are accessible through the terminal blocks on the unit. GPIO0 is internally connected to GPIO1. To edit GPIO properties, right-click the desired GPIO node in the Project Tree, then click “Properties”.

GPIO1 and GPIO2 can be configured to behave in a number of modes of operation:

**Disabled:** Activity will not be monitored or reported.

**Digital Input:** The supplied voltage/signal will be reported as a 0 or 1, depending on whether the voltage level is above the Threshold setting.

**Analog Input:** The supplied voltage will be monitored by a 10 bit 0-3.3VDC DAC. The signal can be evaluated by its VAL (current value), MIN (minimum value since last report), and MAX (maximum value since last report).

**Digital Output:** The GPIO can be set to switch a 10ma 3.3VDC signal on or off.

GPIO0 is tied internally to GPIO1 and can be configured to report GPIO1 activity in two additional modes:

**Counter:** Keeps a running count of off/on transitions on GPIO1.

**Frequency:** Reports the frequency (in Hz) of off/on transitions on GPIO1.

The GPIOs have two adjustable parameters to help tailor the monitoring to the supplied signal.

**Delay:** The amount of time (in tenths of a second) between asynchronous status reports when a GPIO state is changed. Set to 0 to disable async reporting.

**Threshold:** For a digital input, this is the level that will determine the difference between reporting the signal as off (0) or on (1).

Click the “Apply” button to apply settings to the unit. Click the “Refresh” button to get settings from the unit.

Click the “Monitor Activity” button to monitor real-time activity on the selected GPIO. It can be useful to watch the behavior of the GPIO when connected to a signal, to make evaluating the input during macro setup easier.

## ADs (BC1)

The BC1 has four AD (Analog/Digital) input ports. To edit AD properties, right-click the desired AD node in the Project Tree, then click “Properties”.

AD inputs 1-4 can be configured to behave in a number of modes of operation:

**Disabled:** Activity will not be monitored or reported.

**Digital Input:** The supplied voltage/signal will be reported as a 0 or 1, depending on whether the voltage level is above the Threshold setting.

**Analog Input:** The supplied voltage will be monitored by a 10 bit 0-3.3VDC DAC. The signal can be evaluated by its VAL (current value), MIN (minimum value since last report), and MAX (maximum value since last report).

The ADs have two adjustable parameters to help tailor the monitoring to the supplied signal.

**Delay:** The amount of time (in tenths of a second) between asynchronous status reports when a AD state is changed. Set to 0 to disable asynchronous reporting.

**Threshold:** For a digital input, this is the level that will determine the difference between reporting the signal as off (0) or on (1).

Click the “Apply” button to apply settings to the unit. Click the “Refresh” button to get settings from the unit.



Click the “Monitor Activity” button to monitor real-time activity on the selected Ad input. It can be useful to watch the behavior of the input when connected to a signal, to make evaluating the input during macro setup easier.

The AD Properties window also allows you to select a macro to be triggered when the state of the input changes.

## Relays

The BC4 has two SPDT relays. The BC1 has four SPDT relays. These relays are rated for 2A @ 30VDC.

Relays can be controlled via control macros or GUI button actions. To edit Relay properties and test Relay functions, right-click the desired Relay node in the Project Tree, then click “Properties”.

You can give the Relay a friendly name, such as “Projector Screen Motor” to make macro creation and editing easier. You can also use the supplied control buttons to control/test the Relay in real time. The Relay Properties window for BC1 relays also allows you to select a macro to be triggered when the state of the relay changes.

## Adding Devices

### IR Devices

To Create a new IR Device, right-click the “IRDEVICES” node (this node is a child of the CONNECTED DEVICES node) in the Project Tree, and select “Add New IR Device”. The IR Device Properties window will open. Provide a name for the new device, and select which IR port this device’s emitter is connected to. To browse the built-in IR database, begin by choosing the applicable Device Type from the droplist. Then choose the desired Manufacturer. Each Manufacturer in the list will have one or more Codesets available for the selected Device Type. To test a codeset, double-click any IR command in the “Default Keys” list. That code will be sent from the selected IR port.

If there are some IR keys not found in the built-in database, you can add learned IR keys to the IR device file. Enter a name in the “New User Key” text box, and click “Add”. This will add a new key to the User Keys list. To learn a User Key, select it from the list, and click the “Learn” button. When the controller is ready to learn, the red “Status” LED will be flashing rapidly. Hold the original remote directly in front of the Learn LED on the controller, and press the desired button. After the IR key has been learned, you can double-click it to test.

*You may need experiment a bit when learning a tricky remote. Try moving the original remote closer or farther away from the Learn LED. Try a longer press/hold of the original button, or a very quick press/release.*

You can also import Pronto Hex formatted IR codes into a User Key. Right-click the desired user key, and select “Import/Convert Pronto Hex”. Paste the Pronto Hex formatted code into the provided text box. Click “Convert” to convert the code into the BitWise format and store the code in the project file.

## Serial Devices

To create a new Serial Device, right-click the “SERIALDEVICES” node (this node is a child of the MODULE: BCX1 node) in the Project Tree, then select “Add New Serial Device”. The Serial Device Properties window will open. Provide a name for the new device.

To add a command, enter a name for the new command, define the command data, and click “Add”. To edit an existing command, right-click the command and select “Edit”.

Serial command data can be defined in one of three methods.

**ASCII String:** Command data is entered as plain ASCII text.

**Hex String:** Command data is entered as a series of hex bytes. For example, “0D0AFF01” (without quotation marks) would send the hex bytes 0x0D, 0x0A, 0xFF, 0x01.

**ASCII String with Hex Terminator:** Command data is entered as a plain ASCII string, to be terminated with one or more hex characters.

## Socket Devices

To create a new Socket Device, right-click the “SOCKETDEVICES” node (this node is a child of the MODULE: BCX1 node) in the Project Tree, then select “Add New Socket Device”. The Socket Device Properties window will open. Provide a name for the new device.

To add a command, enter a name for the new command, define the command data, and click “Add”. To edit an existing command, right-click the command and select “Edit”.

Socket command data can be defined in one of three methods.

**ASCII String:** Command data is entered as plain ASCII text.

**Hex String:** Command data is entered as a series of hex bytes. For example, “0D0AFF01” (without quotation marks) would send the hex bytes 0x0D, 0x0A, 0xFF, 0x01.

**ASCII String with Hex Terminator:** Command data is entered as a plain ASCII string, to be terminated with one or more hex characters.

## Script Devices

Script Devices are a powerful addition to Project Editor, which provide the ability to add two-way feedback for BitWise Touch GUIs. A Script Device can use either RS232 or TCP /UDP. Script Devices are made up of Javascript Functions (for controlling the device), and Expressions (using RegEx, or Regular Expressions, for capturing status feedback from the device). Script Devices are edited with the Script Device Properties Window. To add a new Script Device to your project, right-click the “SCRIPTDEVICES” node in the CONNECTED DEVICES node of the desired BC4 in your Project Tree, then select “New Script Device”.

**Note that each BC4 controller in an installation can support ONE two-way Script Device.**

**BC1 controllers can support many two-way Script Devices.**

Script Devices are a powerful way to add two-way feedback and advanced functionality to your installation. As such, creating new script devices can be more complicated than simple one-way Serial or Socket Devices. Check the “Tech Notes” section of the installer forum downloads area for more detailed information about working with Script Devices.

**(BC4 ONLY) Make sure to select the proper Two-Way mode in the Serial Port properties if your Script Device uses RS232 or TCP.**

## Device Import/Export

Once you have created your devices, you may wish to export the device to share with others, or to use in a future project. To export an IR, Serial, or Socket device, simply right-click the device in the project tree, and select “Export Device”. You will be prompted to choose the name and location to export the device file.

To import a device, right-click the desired device group (“IRDEVICES”, “SERIALDEVICES”, or “SOCKETDEVICES”) in the Project Tree, and choose “Import Device”. Browse to the device file you wish to import.

## Device Copy/Paste

You can copy and paste devices from one controller to another, or within the same controller. To copy a device, right-click the device in the Project Tree, then click “Copy”. To paste the device, right-click the device group you want to paste the device into, then click “Paste”.

## Device Properties

To edit a device you have already created or imported, right-click the device in the Project Tree, then click “Properties”.

## Online Device Library

Users can submit exported device files to share with others via our online device library. In the main menu bar, click Tools->Online Device Library. You can browse the library for IR, Serial, or Socket device files supplied by BitWise Controls, or submitted by other users. To download a file from the library, use the list boxes to choose a device file, then click “Download”. The file will be saved to the “Devices” folder in the application directory. These files can then be imported to your projects.

To submit a device you have created, click “Submit a Device”. You will be taken to a web page you can use to enter the device information and upload the device file you have created and exported.

Click “Request a Device” to enter a request for a device file you are looking for. BitWise support will attempt to find any codesets or command data and assist you in creating a device file for the requested device.

## Macro Creation

A macro is a series of command actions. For example, you might create a “Watch DVD” macro, which would set the theater Projector and AV Receiver to the correct inputs, set the correct surround mode, and maybe even dim the lights. In Project Editor, you can define these macros and upload them to the BC4X1 Pro. To start working with macros, expand the “BC4MACROS” node (this node is a child of a BC\$ node) in the Project Tree. Inside this node, you will find three child nodes, “VARIABLES”, “MACROS”, and “MATCHSTRINGS”

## Variables

Each BC4X1 Pro contains 30 numeric user variables, which can be used to track equipment states, and help create complex logic for control macros. User variables can hold numeric values from 0-255. You can rename a variable to something more user friendly, like “Theater Power State”. To do this, expand the “VARIABLES” node, then right-clicking the desired variable, then click “Rename”. User variable values can be set and evaluated in control macros.

## Macros

There are two System Macros present in every BC4. Macro #1 is the “Startup Macro”, which runs every time the hardware is reset. Macro #2 is the “Looping Macro”, which runs in a continuous loop anytime the unit is not doing anything else. The Looping Macro is a good place to evaluate the state of GPIO inputs, in order to trigger actions based on an input state change.

The BC1 only contains one dedicated System Macro. Macro #1 is the “Startup Macro”. The BC1 does not utilize a looping macro, since actions can be triggered directly from relay and AD input state changes.

To add a new macro, right-click the “MACROS” node in the Project Tree, then click “Add New Macro”.

## Edit a Macro

To edit a macro, select the desired macro from the Project Tree, and click “Edit” in the Macro Editor pane of the Main Window. You can add macro lines in one of two ways. You can right-click the macro line list and choose “Insert Command”. You can also drag any command (such as an IR command, or a Relay Action) from the Project Tree into the Macro Editor. Once the command has been added to the macro, you can right-click any macro line, then click “Edit”. This will bring up the Macro Line Editor window.

## Macro Lines

Each macro line is made up of three components. The Macro Line Editor window provides a tab to edit each component.

**Delay:** The amount of time to delay before executing the command action. The delay value is in tenths of a second, for example 0.5, or 1.2 .

**Evaluations:** Each macro line can execute its command action based on the evaluation of user variables or hardware states.

**Action:** This is the actual command to be executed.

## Evaluations

The following items can be part of a macro line evaluation.

**Temperature AD Val:** The current AD reading of the built-in temperature sensor.

**Temperature C Val:** The Celsius value of the built-in temperature sensor.

**Temperature F Val:** The Fahrenheit value of the built-in temperature sensor.

**GPIO Count:** The current state of the counter. Only available if GPIO0 is set to Counter Mode.

**GPIO Frequency:** The Frequency of input activity on GPIO1. Only available if GPIO0 is set to Frequency Mode.

**GPIO Input:** The current value of the GPIO. Only available if a GPIO has been set to Digital or Analog Input mode.

**GPIO MIN:** The Minimum value of the GPIO since the last report. Only available if a GPIO has been set to Analog Input mode.

**GPIO MAX:** The Maximum value of the GPIO since the last report. Only available if a GPIO has been set to Analog Input mode.

**Relay State:** The current state of the Relay. Will be 0 if the Relay is deactivated, 1 if the Relay is activated.

**User Variable:** The current value of the User Variable.

**Host TCP Socket Status:** The status of the main TCP socket, which is used for BitWise Command Protocol communications. See Command Protocol Documentation for more information.

**Serial TCP Socket Status:** The status of the BC4X1 serial port's dedicated TCP socket. See Command Protocol Documentation for more information.

**Client TCP Socket Status:** The status of the Client TCP Socket, which is used to communicate with TCP Socket Devices. The value will be 0 if the Client Socket is disconnected, 1 if the Client Socket is connected.

## Actions

Each macro line must have a Command Action. Actions can be dragged into a macro from the Project Tree. The following Action types are available:

**Output Action:** Only available if a GPIO has been configured as a Digital Output.

**Relay Action:** The selected Relay can be set On or Off, or can be pulsed with a user-defined period.

**IR Command Action:** Send an Infrared command to a connected IR device. IR Command Actions can be defined as a single pulse, or a continuous send, which must be canceled. Continuous/Cancel mode IR commands can be useful for things like volume controls, where you want an IR command to start sending continuously when a GUI button is pressed, and canceled when the GUI button is released.

**Serial Command Action:** Send a RS232 command to a Serial Device.

**Socket Command Action:** Send a command to a Socket Device.

**Report Variable:** Force the BC4X1 Pro to send a User Variable state report packet to a connected system.

**Set Variable:** Set the value of a User Variable.

**Increment Variable:** Increase the value of a User Variable by one.

**Decrement Variable:** Decrease the value of a User Variable by one.

**Report Counter Value:** Force the BC4X1 Pro to send a Counter report packet to a connected system.

**Report Frequency Value:** Force the BC4X1 Pro to send a Frequency report packet to a connected system.

**Set Counter:** Manually set the Counter value.

**Manual Command:** Manually define a command to execute based on the BitWise Command Protocol.

**Run Macro:** Run another Macro.

(BC4 Note ) When running one macro from another macro, the last macro called will only return to the macro that called it. For example, if Macro #3 calls Macro #4, which then calls Macro #5, after Macro #5 completes, it will return to Macro #4, but Macro #4 will not return to Macro #3. BC1 controllers do not have this limitation, macros can be repeatedly nested. It is important to use caution with nested macros, since it is possible to create an “infinite loop” in this manner.

## Matchstrings (BC4)

If the serial port has been configured for Serial Matching, you can define serial “Match Strings” which will execute a macro when received. Incoming serial data is matched according to two configurable parameters, Match Length, and Match Terminator.

If Match Length is set to 0, it will attempt to match the entire incoming string, until the Match Terminator is received. Otherwise, it will attempt to match the number of characters defined as Match Length.

Match Terminator is the decimal representation of the character you expect to be received as a terminator for incoming serial data packets.

To Edit these parameters, expand the “BC4MACROS” node in the Project Tree, right-click the “MATCHSTRINGS” node, then click “Serial Match Settings”.

To add a new Matchstring, expand the “BC4MACROS” node in the Project Tree, right-click the “MATCHSTRINGS” node, then click “Add New Serial Match”.

When the Serial Match Editor window appears, you can enter the string of incoming serial data you expect to receive. Then select the macro you want to trigger when that MATCHstring is received.

You can click “Get Match String From Input” to monitor the incoming serial data, and choose the desired Matchstring from that data.

## IR Triggers (BC1)

The BC1 has a built in IR receiver module, which provides the bulk of the functionality of our BC-IRI IR to RS232 adapter. This means that third-party IR control systems can trigger BC1 macros. There are 255

named IR codes which can be used for IR triggers. These codes can be programmed into third-party remote control systems. The IR codes are available in a number of formats and can be downloaded from the BitWise Controls website.

IR Triggers can be matched to BC1 macros via the IR Triggers setup window.

## Macro Uploading

After you have created your control macros, they must be uploaded to the BC4X1 Pro controller.

Macros can be uploaded by clicking “Upload Project (All BC4s)” in the “Program” menu of the Main Window. This will compile and upload control macros for ALL BC4 hardware in the project.

In projects containing multiple BC4X1 Pro controllers, you may wish to only upload one unit at a time. To upload a single BC4 unit, right-click the BC4 node in the Project Tree, then click “Upload This BC4”.

## Scripts (BC1)

The BC1 has a built-in javascript rendering engine, which can run user scripts stored inside the BC1. BC1 Script functions can be called from macros or Scheduled tasks. More information about BC1 scripting can be found in the Javascript API documentation available from the BitWise Controls website.

## Schedule(BC1)

Macros and script functions can be scheduled to occur at predefined times using the Schedule Properties window. Tasks may be scheduled based on any combination of time and date. Tasks can also utilize the built-in astronomical clock functionality, defining an offset number of minutes before or after sunrise or sunset. **In order for scheduled tasks to occur at the right times, the BC1 clock and project location settings must be defined properly.**

**Check the dealer section of the BitWise Controls website for tutorial videos explaining scheduling features .**

## GUI Creation

Project Editor includes a visual GUI (Graphical User Interface) editor, which is used to create GUI pages which control the BC4X1 Pro and its connected devices. These GUI files can be used in conjunction with



the BitWise Touch App for iPhone, iPod Touch, and iPad, and can also be generated as complete HTML files which can be used for browser-based control via PCs and mobile devices. The basic components are listed below, but be sure to watch all available tutorial videos for additional information and time-saving tips and tricks.

## GUI Groups

GUI Groups contain one or more GUI Pages. All GUI Pages that you wish to include for a given display device (such as an iPad) must be contained in a GUI Group. A Project file can contain as many GUI Groups as you like. A single GUI Group can be uploaded to multiple instances of the BitWise Touch App.

To add a new GUI Group, right-click the “GUIs” node of the Project Tree, then click “Add GUI Group”.

## GUI Pages

Each GUI Page can contain labels, images, and buttons, and is completely customizable. To add a new GUI Page, right-click the GUI Group you want to add a page to, then click “Add GUI Page”.

## Editing a GUI Page

To customize a GUI page, double-click the GUI page, or right-click the GUI Page in the Project Tree, then click “Edit”. The GUI Page Editor will appear.

## GUI Menu

The GUI Menu contains the following page-specific options.

**Target Screen Resolution:** This is where you define the size of the GUI Page. Changing this setting will resize the canvas to the boundaries of the specified resolution. Note that you can choose from preset screen resolutions, or enter a custom value. Presets are provided for iPhone/iPod Touch and iPad, in both Landscape and Portrait orientations.

**Image Folder:** You can define a custom path for images associated with this GUI page. By default the “Images” folder in the application directory will be used.

**Render Mode:** In most cases this will be left in “Normal” mode. If you are storing GUI files as HTML to be stored on a webserver for use with the Mobile Safari browser, choose “Mobile Safari”.

**iPhone/iPad Rotation:** If you would like to create both landscape and portrait versions of each GUI Page (when used with BitWise Touch), you can have them automatically jump when the device is rotated. For example, let’s say you’ve created a landscape page called “DVD-Landscape”, and a portrait version of the same page called “DVD-Portrait”. When editing DVD-Landscape, you would set the On Portrait Rotation setting to the DVD-Portrait page, and leave the On Landscape setting to None. When editing DVD-Portrait, you would set the On Landscape setting to the DVD-Landscape page, and leave the On Portrait setting set to None.

**NEW\*\*Page Macros:** Project Editor 1.4 introduces Page Macro and App Macro triggers, which allow macros to be triggered when the BitWise Touch app is launched, or when a specific page is loaded. App Macros and Page Macros can be selected via the GUI menu in the GUI Editor, or by drag/dropping the desired macro onto the GUI page or GUI Group in the Project Tree.

## Grid Menu

You can use a customizable grid to assist you in lining up GUI page elements, such as Buttons, Labels, and Images. The Grid Menu contains the following items:

**Show/Hide:** Click to show or hide the grid.

**Spacing:** Used to adjust the spacing size of the grid.

**Color:** Choose what color you want the grid to be displayed in.

**Turn Snap On/Off:** Used to turn Grid Snapping on or off. When snapping is enabled, the GUI element being moved will snap to the nearest grid corner that lines up with the top left corner of the GUI element.

## Background

You can choose to use a solid color background for your GUI Page, or to use an image file for the background.

To change the Background color, right-click anywhere on the background, then click “Change Background Color”. If you have selected a Background Image, you will need to remove it before you will be able to see the Background color.

To change the Background Image, right-click anywhere on the background, then click “Change Background Image”. Browse to the image file you wish to use. Note that the image size should match the GUI Page’s Target Screen Resolution.

To remove the Background Image, right-click anywhere on the background, then click “Remove Background Image”.

## GUI Buttons

GUI Buttons are used to control devices and/or navigate between GUI pages.

To add a new button, right-click anywhere on the background, then click “Add Button”.

To edit an existing button, right-click the button, then click “Properties”, or double-click the button.

In the Button Properties window, you can select image files to use for the “Normal” and “Pressed” states of the button. You can define multi-line button text, including left/center/right alignment, font, and size.

Use the Navigation drop list to choose a GUI page you would like the button to jump to. Other specialized Navigation options include URL Schemes (for launching other Apps), and the new “Wait Page” functionality, which allows you to jump to a “Please Wait” page during activity setup, then jump to the activity page after the desired delay. GUI Button navigation can also be set by dragging a GUI page from the Project Tree onto a button in the GUI Editor.

Use the Actions lists to define button actions. The action list will be populated with any BC4X1 hardware and connected devices you have defined in the project. Note that you can define separate actions for button press and release.

GUI Button actions can also be added to buttons via drag/drop from the Project Tree. You can drag individual commands from a device in the Project Tree onto a GUI button to add commands one at a time. If you hold Shift while dropping the command, it will be added to the button’s “Release” action, instead of the “Press” action.

Macros can also be added to button actions via drag/drop.

Entire pages of commands can be populated by dropping a Device node from the Project Tree onto the GUI page background. You will be prompted to choose which command types to populate. Commands are populated on buttons based on the button’s Command Tag, which can be defined in the button’s properties window. If a button has no Command Tag, it will be matched based on button text.

**NEW\*\***Beginning with Project Editor 1.4, you can also drag/drop Devices onto GUI Groups and Pages inside the project tree. This allows the installer to populate pages quickly, without even needing to open the page in the GUI Editor. Dropping a device onto a single page will populate only that page. Dropping onto a GUI Group will populate the entire Group. This is especially useful for volume commands.

**NEW\*\***GUI Buttons can also be assigned a Feedback ID (FBID) from a script device. This will allow two-way feedback to be displayed as the button’s text (using the Feedback Text ID) or the button’s Pressed state (using the Feedback State ID) at runtime. To assign a FBID to a label, either drag/drop the feedback item from the Script Device in the Project Tree onto the Label in the GUI Editor, or manually select the FBID in the GUI Label’s properties window. Dropping the feedback item onto a button in the GUI Editor will set the button’s Text ID. Dropping while holding Shift will set the button’s State ID. Note that a button that has a defined State ID will not change its press/release appearance based on physically touching the button, it will only change state when the associated feedback value changes.

## GUI Labels

GUI Labels are used to add text to GUI Pages.

To add a new label, right-click anywhere on the background, then click “Add Label”.

To edit an existing label, right-click the label, then click “Properties”.

In the Label Properties window, you can define multi-line label text, including left/center/right alignment, font, and size.

GUI Labels can also be assigned a Feedback ID (FBID) from a script device. This will allow two-way feedback to be displayed as the label’s text at runtime. To assign a FBID to a label, either drag/drop the feedback item from the Script Device in the Project Tree onto the Label in the GUI Editor, or manually select the FBID in the GUI Label’s properties window.

## GUI Sliders

Sliders are used in conjunction with Script Devices or custom Javascript functions. They provide a convenient method for controlling (and displaying) the current level of things like device volume, lighting levels, etc.

To add a new slider, right-click anywhere on the background, then click “Add Slider”.

To edit an existing slider, right-click the slider, then click “Properties”.

Sliders consist of a background image (the slider’s base) and images for the handle’s pressed and released states. Changing slider images is handled the same way as a GUI button. Sliders can be assigned functions and feedback via the same methods as GUI buttons.

## GUI Panels

GUI Panels are a simple static image, which can be used to create a layered appearance for backgrounds. Panels do not have actions, or display feedback.

To add a new panel, right-click anywhere on the background, then click “Add Panel”.

To edit an existing panel, right-click the panel, then click “Properties”.

## Web Images

Web Images can be used to add any network-accessible image file (including most MJPEG IP Camera streams) to GUI Pages.

To add a new Web Image, right-click anywhere on the background, then click “Add Web Image”.

To edit an existing Web Image, right-click the Web Image, then click “Properties”.

In the Web Image Properties window, you can define the path to the image file, as well as the desired display size of the Web Image.

## GUI Page Copy/Paste

You can copy and paste GUI Pages from one GUI Group to another, or within the same GUI Group. To copy a GUI Page, right-click the GUI Page in the Project Tree, then click “Copy”. To paste the GUI Page, right-click the GUI Group you want to paste the GUI Page into, then click “Paste GUI”.

## Import/Export

Once you have created your GUI Pages and Groups, you may wish to export them to share with others, or to use in a future project. To export a GUI Group, simply right-click the Group in the project tree, and select “Export Group”. You will be prompted to choose the name and location to export the GUI file.

To import a GUI file, right-click “GUIs” node in the Project Tree, then click “Import GUI Group”. Browse to the device file you wish to import.

Note that exported GUI files will not retain button actions, but Page Navigations assigned to GUI Buttons will be maintained.

Exported GUI files include all associated image files. When a GUI file is imported, the image files will be copied into Project Editor’s default image folder.

## GUI Scripts

You can use custom Javascript files to extend the functionality of GUI pages. Project Editor 1.4 introduces the ability for GUI Groups to utilize embedded scripts, which eliminates the need to maintain separate Javascript files. These embedded scripts can utilize the new built-in script editor, which includes syntax highlighting for Javascript.

Javascript functions can be called from GUI button actions, using the “External Javascript” button action type. Specify your function and parameters in the following format:

```
funcName:parameter1:parameter2:parameter3:etc.....
```

See the Javascript API reference document for information about using some built-in javascript functions to create your own desired functionality.

## GUI Generation (as HTML)

GUI Groups can be generated as HTML files, which can be used with most modern desktop browsers such as Internet Explorer, Firefox, Safari, and Chrome. To generate a GUI group as HTML, right-click the GUI Group in the Project Tree, then click “Generate Group as HTML”. You will be prompted to select the

directory you would like to generate the files into. After the files have been generated, you will be prompted to view the generated folder.

Note that Two-Way communication with Script Devices requires native socket support. At this time this is not possible with GUIs running in desktop browsers as html.

## GUI Uploading (to BitWise Touch App)

After you have created a GUI Group, you can upload the files to BitWise Touch for iPhone/iPod Touch/iPad. Right-click the GUI Group you want to upload, then click “Upload Group to BitWise Touch”. The Upload Server window should appear. On the Apple device, start the BitWise Touch app. When prompted, click “Yes” to begin the file transfer. After the files have been transferred, you will be prompted to select which GUI page you want to be your Startup page (the page that gets loaded when the app is started).

## Firmware Updates

Occasionally, BitWise Controls may release updated firmware for the BC4 and BC1 hardware to introduce new or enhanced functionality. Project Editor has a built-in Firmware Updater you can use to update the hardware. The Firmware Updater can be accessed from the Tools Menu of the Main Window.

To update the firmware, use the “Discover” button to discover any BitWise hardware on the network. Select the desired unit from the list. Use the “Browse” button to browse to the desired firmware file. Click “Update Firmware” to apply the update. The Firmware Updater window can also be used to update the unit’s internal web pages.