WebBrick User and Reference Manual

Andy Harris, Lawrence Klyne

October 30, 2009

July 2007 Document Version 6.6

Firmware Versions 6.6

http://www.WebBrickSystems.com for company information

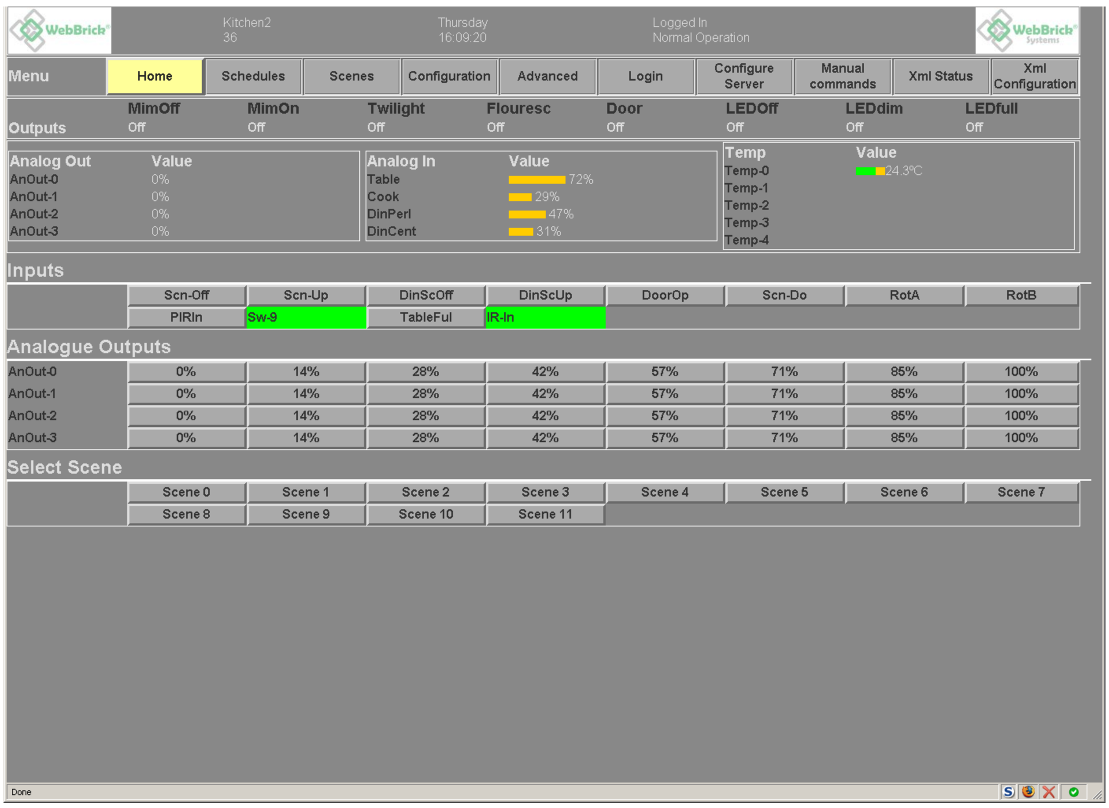
http://community.WebBrickSystems.com for WebBrick information

Contents

1 Overview

What is a WebBrick? It is a network connected control and automation product designed around the principle of 'local control - global intelligence'. This means that most of the time control is handled locally but a system as a whole can be made more intelligent through the use of some central control (for example the WebBrick Gateway) that provides the global intelligence. i.e. Lights for a room are a local issue, but heating is more of an overall control issue. This also means that you are not dependant on a single control system for the complete system, you can always switch on the lights in this room while someone is making changes to a global control system.

Control can be done through a WebBrick's web interface, through its physical inputs or using remote commands.



Kitchen2
36

Thursday
16:09:20

Logged In
Normal Operation

Outputs	MimOff	MimOn	Twilight	Flouresc	Door	LEDOff	LEDdim	LEDfull
	Off	Off	Off	Off	Off	Off	Off	Off

Analog Out	Value
AnOut-0	0%
AnOut-1	0%
AnOut-2	0%
AnOut-3	0%

Analog In	Value
Table	72%
Cook	29%
DinPerl	47%
DinCent	31%

Temp	Value
Temp-0	24.3°C
Temp-1	
Temp-2	
Temp-3	
Temp-4	

AnOut-0	0%	14%	28%	42%	57%	71%	85%	100%
AnOut-1	0%	14%	28%	42%	57%	71%	85%	100%
AnOut-2	0%	14%	28%	42%	57%	71%	85%	100%
AnOut-3	0%	14%	28%	42%	57%	71%	85%	100%

Figure 1: Example of WebBrick Home Page

1.1 History

WebBrick started life as a bespoke project and in that incarnation there are 50-100 of them installed handling tasks from building security to heating control, the largest setup uses 15+ WebBrick to manage a single house. The latest incarnation (WebBrick6) is a commercial project with added facilities. The V6 WebBrick has an added IO board which provides screw terminals for all connections. It also has 4 Mains switching triacs and 2 Double pole changeover relays as well. We can also produce bespoke versions of the IO board for specific uses.

1.2 The Version 6.4 WebBrick

The V6.4 WebBrick has the following connections:

1. 8 Digital Outputs available as a combination of TTL and Open Collector drives, with an option for 8 more on bespoke developments. In the default configuration 4 digital outputs are connected to mains triacs and can handle up to 500W per channel with a maximum of 1500W across all 4 channels. 2 Of the Open collector outputs are connected to double pole changeover relays. 4 channels are available as TTL outputs.

2. 8 Mimic outputs. These are available at TTL level are used to indicate the state of the WebBrick outputs. The main functions of these mimics is to drive indicator LEDs found on push buttons. They are PWM modulated so that the brightness can be modified for the on and off phases. The internal analogue and digital outputs can be configured to set the mimics to the on or off brightness(xx) levels or they can be directly controlled from a system external to the WebBrick.

3. 12 Digital Trigger Inputs, TTL level - These can be treated as de-bounced push-buttons or toggle [MK] style inputs. The actual trigger point can be the rising edge, falling edge or both. All inputs have weak pullups so that switches can be connected with no further components involved.

4. 4 Analogue Outputs 0-10V. When the output level is changed, a new target value is set, these outputs adjust the output value over a time period (Fade Rate). Fade Rate is adjustable and can be as short as 200mS for full range change or as long as 50 Seconds.

5. 4 Analogue Inputs This is a 0-5V input, high impedance, care must be taken not to exceed 5V on this pin, the web interface displays 0-100.

6. One Wire Bus for up to 5 off Dallas DS18B20 temperature sensors.

7. RJ45 connection for 10Mb/s Ethernet

8. 12V power connection

Note that the Rotary Encoder connections from the pre 6.2 WebBrick have been incorporated into standard Digital Inputs, up to 6 rotary encoders can be supported.

2 Architecture Description

2.1 Hardware

The WebBrick V6 is based around the PIC 18F6x22 chip and a NetMedia SitePlayer to create both HTTP and UDP command channels. The base 8 digital outputs are buffered through a Open collector drivers that can handle 12V 500mA drive with reasonable protection.

Some provision is made for input protection from excessive voltage by a combination of a current limiting resistor and the clamp on the PIC chip inputs, edge triggered inputs are de-bounced in software.

WebBrick uses several properties of the SitePlayer:

- SitePlayer can send commands to the PIC ship from its own web interface.
- SitePlayer can send commands to the PIC chip received over HTTP and UDP.
- The WebBrick can get the SitePlayer to send a UDP packet, the contents of which are managed by the PIC chip.

2.2 Software Architecture

The WebBrick is based around a configurable state machine, this is driven by a variety of triggers. The triggers can be generated by digital inputs, analogue inputs, temperature sensors and network command. These state machines are configured on what to do when they receive a trigger, this can be as simple as switch an output on or may involve changing the current scene and sending a network message to the home gateway. Although there are direct output control commands, in general operation the user configures the inputs to perform *actions* on outputs.

actions can be activated by:

- Physical input pins, for example taking digital input 0 low momentarily would set the actions configured for state engine 0 in motion. There is a de-bounce time on these circuits.
- HTTP request.
- Analogue or temperature inputs above or below configured thresholds.
- UDP packet
- Scheduled Event

2.3 WebBrick Indicator LEDs

The WebBrick has three indicators, which can been seen through the top translucent window. They are:

- Red This indicator shows that the power to the process control and web server chips is healthy.
- Green This indicator shows that the web server chip is seeing a network link
- Blue This is the WebBrick heartbeat, it blinks a lot, it has different blinks at start-up, but once a WebBrick is running it will settle to a 1 sec blink rate. *seesectionbelow*

2.4 WebBrick Start-Up

When power is applied to the WebBrick, the SitePlayer chip and the PIC chip will start.

It is important that any WebBrick has deterministic behaviour, therefore there is a settling period before the configuration details from the EEPROM are transferred to the SitePlayer and hence the outside world.

The sequence is:

- Power applied to WebBrick
- Internal web server boots
- PIC boots waits 1-2 seconds for web server to settle
- PIC starts to transfer configuration to web server, this takes about 3 seconds
- If the PIC contains an IP address definition, this is set into the web server. Just before this point the WebBrick will have the default address of 10.100.100.100.
- Operational Status set to 'Normal Operation' Interrupts enabled
- One Wire Bus searched for temperature sensors
- At 30 seconds after power up the internal real time clock is read, it is checked once a minute thereafter.
- Schedules are evaluated to see if any actions need to be executed. i.e. if the WebBrick is started at 10 o'clock and there is a scheduled action at 9 o'clock this action will be executed. Therefore if a WebBrick was controlling a central heating system it would recover from a power failure and carry on where it left off.

2.5 WebBrick Heartbeat

A LED connected to the PIC chip is used as a Heartbeat to show that the system is working. It switches on and off at approx 1 second intervals with an equal off and on time once the WebBrick has completed its startup. This LED is driven from the main program loop in the PIC chip.

If it stops flashing then the software has encountered an error, this would be a very dire state and if you ever see it you should let us know. We've only ever seen this happen with very noisy power supplies.

During start-up the Heartbeat will flash quickly. If the factory reset button is held in at power-up the Heartbeat will remain on for two seconds before starting the load defaults and normal start-up operations.

[Future functionality] If the web interface is logged in then the LED will flash faster spending more time on and less time off, if the web home page has its controls disabled it will flash slower spending more time off than on. [Not yet implemented]

3 Command Structure

The WebBrick has a command interface that accepts command strings, these can be generated by the Web interface and can also be sent to the WebBrick over the network. For network delivery the commands can be sent using an HTTP URL with the command encoded in the parameters or a UDP transmission, the HTTP use is preferable as you get an indication of whether the command was received at the WebBrick.

3.1 General

All commands start with a 2 character identifier, and are followed by any required parameters and terminated by a colon (':') character after the parameters, which means end-command. Each parameter to a command is terminated by a semi-colon (;). In this document items in <is a user parameter> and should be replaced with real world values. Items in [...] are optional parameters, therefore are not essential to replace. Generally the first character within the 2 character identifier identifies the command group and the second identifies the entity type. No extra blank spaces should be inserted into the command as unexpected things may happen or the command be rejected.

These commands can be sent to the WebBrick in two ways:

1. By encoding the parameters into an HTTP URL and accessing that URL
2. or by sending a UDP packet to the WebBrick with a correct Siteplayer header on it for delivery to the PIC chip.

The embedded UI uses the HTTP URL approach and it is recommended that this approach is used for external systems. If you want to use UDP packets look at the python code in wbUdp.py for details.

The HTTP URL to be used is http://<ip Address>/cfg.spi?com=<commandString> Where <ip address> is the WebBricks IP address or host name. and <commandString> is one of the command strings documented here. Note some commands will need you logged in for them to be processed.

To make it easier to understand the command strings the first character is generally used as follows:

3.1.1 command groups

1st Letter	Command Group
C	Configure
S	Set
N	Name
D	Digital
A	Analogue
I	Infra Red
T	Thresholds

And the second character generally identifies the target channel type, as follows:

3.1.2 entity type

2nd Letter	entity type
D	Digital input
A	Analogue Output
I	Analogue input
O	Digital Output
T	Temperature
C	Scene
R	Serial comms or rotary encoder

3.1.3 Trigger Configuration

A lot of the commands take a sequence of parameters referred to as a Trigger sequence, this defines what is to happen when a trigger event occurs, for example a digital input or a temperature threshold being crossed.

The sequence of parameters is:

<A|D|S|T|I><targetChn>; <SetPointNr>; <actionType>; <DwellNr>; <UDPType>; <AssociatedValue> [; <OptValue>]:

Where:

Parameter	Description
<A|D|S|T|I>	is a single character that identifies whether the target channel is an analogue output(A), a digital output (D), a Scene (S), a Temperature sensor (T), or Infra Red Send
<targetChn>	is the channel number or Scene being targeted
<actionType>	is one of the values from the Action table (range 0-15) In the case of a Scene this is used as the action for all analogue channels that are not marked to be ignored and for digital channels marked for On in the scene configuration.
<DwellNr>	is a dwell number and is ignored if the actionType is not a dwell. Note for DT command dwell is passed in seconds and not as a dwell index.
<SetPointNr>	is a setpoint number and is ignored if the target channel is not analogue, Note a Scene is not an analogue channel and the SetPoint is taken from the Scene.
<UDPDo>	identifies whether a UDP packet type is to be sent
<AssociatedValue>	is a number associated with the action.
<OptValue>	is an optional reserved value.

3.1.4 Actions

Each trigger can cause one of the following actions to be performed on an output channel.

Nr	Action	Description
0	None	no action
1	Off	switch off
2	On	switch digital channels on or an analogue channel to a setpoint.
3	Momentary	Switch a channel on for a small time period circa 200mS.
4	Toggle	Turn an on channel off and an off channel on. For analogue channels if the current setting is greater than 0 it is deemed to be on.
5	Dwell Always	Always Switch a channel on for a configured time period, if lamp already on this will switch off after a time period.
6	Dwell-Cancel	If a channel is Off Switch the channel on for a configured time period. If the channel is already On then switch the channel Off immediately.
7	Next	Move a channel to its next higher state, for analogue channels this is to the next set point, for digital channels this is equivalent to a toggle. If the target is any Scene change the current scene at the web brick up by one.
8	Prev	Reverse of Next, Down Setpoint, Toggle, Previous Scene.
9	SetLowThreshold	Change the low threshold for one of the analogue input or temperature sensor
10	SetHighThreshold	Change the high threshold for one of the analogue input or temperature sensor
11	AdjustLowThreshold	Move the low threshold for one of the analogue input or temperature sensor
12	AdjustHighThreshold	Move the high threshold for one of the analogue input or temperature sensor
13	SendIR	Send a command over the Infra Red emitter, RC5 only, the target RC5 channel is the command code and associated value is the RC5 address.
14	Up	Take a channel up a step, generally used for analogue outputs.
15	Down	Take a channel down a step, generally used for analogue outputs.
16	Set Dmx	Set a DMX channel to a level.
17	Count	Sets the channel to count mode.
18	Dwell On	Switch a channel on for a configured time period if it is not already on.
19	Dwell Off	Switch a channel off after a configured time period.

Notes

- Toggle, On and Off actions all override a current Dwell command.
- A Dwell command issued during a current Dwell period will reset the Dwell time.
- A Dwell Cancel command issued during a current Dwell period will end the dwell. A Dwell Cancel issued outside a Dwell period will switch the output on for the Dwell

time.

- At the end of an analogue dwell the output will return to the level prior to the dwell, this is useful for setting lights to a low level and bringing them high on some trigger, e.g. security PIR.
- Channel numbers are zero based for the internal commands.
- Scenes are in two banks, one bank of 8 and one bank of 4, when performing Next and Prev Scene the scene change stays within the bank that the trigger definition identifies by selecting one of the scenes within the chosen bank.
- String parameters cannot contain any of the characters '<>&
- Analogue channels go up and down setpoints with Next/Prev and step the level the rotary step value with Up/Down &
- Some commands are meaningless with some outputs, where possible reasonable actions are chosen. i.e. for digital channels Next/Prev/Up/Down are treated as a toggle command

3.2 Error Codes

These may be displayed in the header of the webbpage and may appear in the xml status.

Value	Description
0	No Error.
1	Bad Command.
2	Bad Parameter.
3	Not Logged In.
4	Address Locked, attempt to change IP address on address locked webbrick.
5	In Startup state.
6	No Command yet issued, post startup state.

3.3 Command summary

3.3.1 Static configuration commands

Command	Function
ND	Name digital Input
NO	Name digital output
NI	Name analogue input
NA	Name analogue output
NT	Name temperature sensor
NN	Name node
CD	Configure digital input
CI	Configure analogue input
CT	Configure temperature input
CS	Configure set point
CW	Configure dwell
CR	Configure serial interface.
CE	Configure scheduled event
CC	Configure a scene
SN	Set node number
SF	Set fade rate
SM	Set mimic brightness levels for on and off signals, and fade rate
CM	Configure mimic channels for analogue and/or digital outputs
ST	Set Time
SD	Set Date
SR	Set rotary encoder step
SI	Set Internet address
SA	Set IP address with verified MAC address; the !WebBrick sets its IP address to the specified value only if its 48-bit MAC address matches that supplied. This command can be used with UDP broadcasts to set IP addresses when several WebBricks on a network have the same initial IP address.
SP	Set password for indicated security level
SO	Set option on (1) or Off (0) (Use with care: currently deprecated, and not reflected in XML configuration)
IR	Enable infrared receive
IT	Enable infrared transmit
IA	Sets the RC5 Infrared address for receiving

3.3.2 Dynamic status setting commands

Command	Function
DI	Trigger digital input
DO	Set digital output
DT	Invoke a trigger event action using the supplied parameters. The parameters are a full trigger definition as can be attached a local event source, i.e. digital input, but are acted upon when received. Note dwell is passed in seconds and not as a dwell index.
DM	Set one or more mimic outputs to specified levels
DA	Requests web brick to send a small number of UDP attention packets within a few seconds (currently: one immediately, and one more within a second). This is used for WebBrick discovery. (cf. factory reset causes attention packets to be sent every few seconds for a minute.)
AA	Set analogue output to one of the setpoints or to an absolute level.
SC	Set scene
SS	Set operational state
TA	Modify dynamic threshold for analogue input
TT	Modify dynamic threshold for temperature sensor

3.3.3 Miscellaneous WebBrick control commands

Command	Function
LG	Login to security level associated with password - also logout.
RT	Re scan 1-wire bus for new devices
RU	Refresh web user interface data
RS	Reboot Siteplayer, PIC chip delays a while and then rebuilds the UI data.
RB	Reboot PIC
RI	Set RS485 driver off, input mode
RO	Set RS485 driver on, output mode
RD	Send serial data
IS	Sends an RC5 infrared command (channel) over a configured infrared emitter

3.4 Commands

All command are listed here.

3.4.1 AA - Analogue output control

AA<chn>;[S<nn>|<nn>] Set Analogue output setpoint or absolute

Set an analogue output to a specific level. chn is the analogue output channel number. If the output value is S<nn> then <nn> is the setpoint, otherwise <nn> is the output level as an absolute value of between 0-100%.

e.g. AA1;S1:

3.4.2 CC - Configure Scene

CC<nr>;[NFI][NFI][NFI][NFI][NFI][NFI][NFI][NFI]; [I|S<nn>]; [I|S<nn>]; [I|S<nn>]; [I|S<nn>]:

Configure a scene. A scene consists of optional settings for all the digital and analogue channels. Any channel may be marked as Ignore (do not change). Digital channels also be marked as On or Off, whilst analogue channels can be given a set point. The first group [NFI] is for the digital channels and is repeated as many times as required to cover the digital channels where a change is required, i.e. if you only want to modify digital channels 0, 1, 2 then you only need list 3 entries. If you want to only affect channel 8 (or 16) then you must send all the preceding ignore's (I). Similarly for the analogue channels where each later entry is optional if you have already specified the required changes.

e.g. CC0;NFNFNFNF;I;S1;S2;S3: CC1;NF: CC3;NF;S0;S1;S2;S3:

3.4.3 CD - Configure Digital Input

CD<chn>; <A|D|S><targetChn>; <sp>; <actionType>; <dwell>; <udpType>; <AssociatedValue> [; <OptValue> [; <Options>]]:

Configure one of the digital inputs. chn is the digital input channel number from 0 to 7 (or more). Remaining parameters are a Trigger Sequence. Options is a set of flags that controls the digital input event generation.

Bit	Value	Description
0	1	generate trigger event on rising edge, i.e. 0v to 5v transition.
1	2	generate trigger event on falling edge (Default) i.e. 5v to 0v transition, a normally open switch is pressed.
2	4	This input and the next is connected to a rotary encoder, when the rotary encoder is turned 'down' this trigger is actioned
3	8	
4	16	
5	32	
6	64	
7	128	

To create the option value add the required Value's from the above table together. You can set both rising and falling edge flags and this is useful if you have a latching switch of some sort, i.e. standard MK light switch.

e.g. CD7;A1;2;2;0;1;1: Configure Digital in 7 to target analogue output 1. setpoint 2, action On(2), dwell (0) ignored, UDP

3.4.4 CE - Configure scheduled event

CE<chn>; <Days>; <hours>; <Min>; <A|D|S><targetChn>; <sp>; <actionType>; <dwell>; <udpType>; <AssociatedValue>[; <OptValue>];

Configure scheduled event. chn is event number, currently 16 scheduled events are catered for. Days is a set of day numbers from 0-6 to identify which days the event occurs on, e.g. "06" is Sunday and Saturday. hours (0-23) and min (0-59) identifies the time within the day when it occurs. The remaining parameters constitute a Trigger set as described else where,

e.g. CE6;12345;12;30;A1;2;2;0;1;1: Configure event 6, at 12:30 on Mon to Fri to target analogue output 1. setpoint 2, action On(2), dwell (0) ignored, UDP.

3.4.5 CI - Configure Analogue Input

CI<chn>; <L|H>; <Threshold>; <A|D|S><targetChn>; <sp>; <actionType>; <dwell>; <udpType>; <AssociatedValue>[; <OptValue>];

Configure one of the analogue inputs. L or H identify whether this is setting a High or Low threshold and Threshold is the analogue threshold as a value from 0-100%. The low level hardware is 0-5V but so as to avoid issues when some signal conditioning is added the input is scaled.

e.g. CI0;L;20;A1;2;2;0;1;1:

3.4.6 CM - Configure Mimics

CM<A|D><source>; <mimicChn>[...]:

Configure mimic channels for one or more analogue and/or digital outputs.

A target indicates an analogue output number target for a which a mimic channel is specified

D target indicates an digital output number target for a which a mimic channel is specified

mimicChn indicates a mimic channel number that will be associated with the corresponding analogue or digital output, to break the connection between a channel and a mimic use a mimic channel number of 15.

Pairs of outputs and corresponding mimic channels may be repeated for each association that is required to be specified. (Or, a separate command may be used for each.)

The default setting is that Digital outputs 0 to 7 are associated with mimic output 0-7. The off level and on level can be reconfigured with the SM command.

e.g. CMA0;0;A1;1;A2;2;A3;3: target mimics 0 to 3 from analogue 0 to 3.

3.4.7 CS - Configure Preset point

CS<chn>; <val>

Configure one of the set points (Preset point) used for the analogue outputs. chn is the setpoint number and val is the setpoint value in the range 0-100%. The low level hardware generates 0-10V but this may be passed through some signal conditioning to suit the application.

3.4.8 CT - Configure Temperature Sensor

CT<chn>; <L|H|B>; <Threshold>; <A|D><targetChn>; <sp>; <actionType>; <dwell>; <udpType>; <AssociatedValue>[; <OptValue>];

Configure one of the temperature inputs. L or H or B identify whether this is setting a High or Low or Both threshold(s) and Threshold is the temperature threshold as a decimal temperature between -50 and +150 degrees Celsius

3.4.9 CR - Configure Serial

CR<mode>; <baudIdx>:

The following modes exist:

mode	Description
0	No mode change.
2	RS232.
4	RS485
3	DMX mode

The following baud rates exist:

index	Speed
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	19200
7	38400
8	57600
9	115200
10	250000
11	9600
12	9600
13	9600
14	9600
15	9600

See commands RO, RI, RD

3.4.10 CW - Configure Dwell

CW*[0-3];DwellValue*:

There are 4 Dwell values that may be set, 0-3. DwellValue is a number between 2 and 32767, it is measured in 'near seconds'. Dwell is measured in more or less seconds. Note that it is the transition between 1 and 0 that marks the end of a Dwell countdown, therefore a Dwell value of 10 really gives a Dwell of between 9 and 10 seconds, depending on when it was started.

3.4.11 DA - Do Attention

DA

Request the WebBrick to send 1 or 2 attention packets, command is generally sent using UDP and broadcast to all WebBricks on the network.

3.4.12 DI - Trigger Digital input

DI<chn>

Trigger input, generates a trigger just as if digital input chn had been triggered.

3.4.13 DM - Do Mimic

DM<mimicChn>; <mimicLevel>[; <mimicChn>; <mimicLevel>]

Set one or more mimic channels to a set level. The mimic channel and level can be repeated to send more than one request in a single command. This overrides the values set by any recent output change targeted to the mimics at the WebBricks. mimicLevel can be between 0 and 63.

3.4.14 DO - Switch Digital output

DO<chn>;N|F|T|D[; <dwell>]:

Set digital output. Sets the state of one of the digital outputs. chn is the channel number to operate.

Action	Description
N	On
F	Off
T	Toggle
D	Dwell

If Dwell is specified then this can either be a DwellNumber <0-3> or if greater than 3 is a dwell in seconds.

3.4.15 DT - Trigger from External

DT; <A|D|S><targetChn>; <sp>; <actionType>; <dwell>; <udpType>; <associated-Value>; <optValue>[; <options>]]:

The packet contains trigger configuration which is actioned on receipt and not stored for later use. This enables an external source to do what an internally generated event can do.

NOTES:

Parameter	Notes
dwell	This is set in seconds and not an index into the dwell table
udpType	this is either 0,None or 1,Send
associatedValue	If this is relevant to the action then this can be set
optValue	This is not used and should be left at 0
options	This is not used and should be left at 0

3.4.16 FR - Factory Reset

FR Configuration Factory reset.

FR1 Full Factory reset.

Perform a factory reset of the WebBrick, the base version only resets user configuration, the later version also resets all options and the IP address (Unless IP address locked at Factory).

3.4.17 IA - InfraRed Address

IA<address>:

Sets the RC5 infra red address to be recognised. The command values 1-8 are mapped to generating digital triggers, i.e. another soft key input.

3.4.18 InfraRed On Off

IR<N|F>:

Switch on/off infra red reception, uses Digital input 11 and disables any other use for this connection.

Action	Description
N	On
F	Off

3.4.19 IS - InfraRed Send

IS<address>; <channel>:

Sends the IR command using RC5 and the address (0-31) and channel (0-63) given. Allows a remote system to send RC5 infra red commands.

3.4.20 IT - InfraRed emitter

IT<N|F>:

Switch on/off infra red transmission. Uses DigOut 7 and disables any other use for this connection.

Action	Description
N	On
F	Off

3.4.21 LG - Login and Logout

LG; <password>

Try to login to the WebBrick to enable the command interface. Up to 3 passwords may be set Level 1 allows access to the Home page controls, its default is blank so the WebBrick automatically enters this state. Level 2 is for reconfiguration Level 3 is full reconfiguration access for installers. Note Login times out 5 minutes after the last valid configuration command (Level 3 is 1 Hour timeout).

Entering an invalid password will logout the user interface.

3.4.22 NA - Name Analogue Output

NA<chn>; <nameStr>:

Give a name to an analogue output, chn is the channel number from 0 to max analogue outputs-1.

3.4.23 ND - Name Digital Input

ND<chn>; <nameStr>:

Give a name to a digital input, chn is the channel number from 0 to max digital inputs-1. The name string cannot contain any of the characters '<>&

3.4.24 NI - Name Analogue Input

NI<chn>; <nameStr>:

Give a name to an analogue input, chn is the channel number from 0 to max analogue inputs-1. The name string cannot contain any of the characters '<>&

3.4.25 NN - Name Node

NN<NodeName>:

Give a name to the WebBrick node. NodeName is limited to 10 characters The name string cannot contain any of the characters '<>&

3.4.26 ND - Name Digital Output

NO<chn>; <nameStr>:

Give a name to digital output, chn is the channel number from 0 to max digital outputs-1. The name string cannot contain any of the characters '<>&

3.4.27 NT - Name Temperature sensor

NT<chn>; <nameStr>:

Give a name to a temperature input, chn is the channel number from 0 to max temperature inputs-1. The name string cannot contain any of the characters '<>&

3.4.28 RB - Reboot

RB Reboot.

Hardware reboot/reset of the PIC chip and Siteplayer.

3.4.29 RS - Reboot Siteplayer

RS Reboot of Siteplayer.

Hardware reboot/reset of the Siteplayer.

3.4.30 RT - Re scan 1 Wire

RT Re scan 1 wire bus.

Scan the one wire bus for new sensors.

3.4.31 RU - Refresh User Interface

RU Refresh User Interface.

Resends all data from the PIC chip to the Siteplayer, for use if the PIC chip and Siteplayer are out of step.

3.4.32 RI - RS485 driver off

RI: Set RS485 driver off, input mode

3.4.33 RO - RS485 driver on

RO: Set RS485 driver on, output mode

3.4.34 RD - Serial send data

RD <databyte as Decimal ASCII>: Send serial data, the databyte parameter can be repeated multiple times.

Example, send 'A' character - RD65:

3.4.35 SI - Set IP Address

Set Internet protocol address SI<n>; <n>; <n>; <n>:

Where each <n> is an element of the IP address.

SA<m>; <m>; <m>; <m>; <m>; <m>; <n>; <n>; <n>; <n>:

Where each <m> is an element of the network MAC address. Where each <n> is an element of the IP address. This is to enable a bunch of WebBricks to be added to a network, identified and addresses set by a discovery process.

3.4.36 SC - Set Scene

SC<nr>:

Set the output channels to match a specific scene. The result is the equivalent of issuing On, Off or SetScene for any channel not marked as Ignore. NOTE there is a slight difference when a trigger is used to set a scene in that instead of the ON action for digital channels marked as On and the analogue channels being sent the action from the trigger will be sent i.e. Dwell, some possible actions will not make sense.

3.4.37 SD - Set Date

SD<years>; <mon>; <date>

Set Date for the WebBrick. Not currently implemented or used.

3.4.38 SF - Set FadeRate

SF<rate> Set Fade rate

Set the rate at which the analogue channels are adjusted to meet the desired output value. The smaller the number the quicker the analogue output channel swings.

With a setting of 1 the analogue will swing full range in approx 200mS, with a value of 255 the swing full range will take circa 50 seconds.

3.4.39 SM - Set Mimic

SM<offLevel>; <onLevel>; <fadeRate>[;<0|1>]:

Set Mimic high and low level and the fade rate between them. These high and low values are used when connected as mimics for analogue and digital outputs, by use of an off level that is not quite off we have a seek light in the dark. onLevel and offLevel can be between 0 and 63. The fade rate controls the speed that the mimic shifts from one level to the next, the next level could be selected by the DM command. The final optional parameter sets the mimic output to low voltage (approx 4V) or high voltage (approx 10.5V).

3.4.40 SN - Set NodeNumber

SN<NodeNumber>:

Set node number NodeNumber should be between 1-254, 0 is reserved for 'new' WebBricks that will be configured by a remote server before they go into production.

3.4.41 SO - Set options flag

Set an option flag value, some are bit mapped and others just on/off. SO<num>; <value>:

Options control some small bits of a WebBrick's operation. Most options are not intended for general use and are undocumented.

Option	Description
1	Manages UDP event transmission.
2	Manages Analogue input options.
3	Manages Analogue output options.
4	Manages Digital input options.
5	Manages Digital output options.
6	Manages Temperaure sensor options.
7	Manages Scene options.

UDP event transmission option.

Bit	Value	Description
0	1	Enable sending temperature changes.
1	2	Enable analogue input changes.
2	4	Enable analogue output changes.
3	8	Enable Infra red reception UDP packets.
4	16	Enable RTC debug UDP packets.
5	32	Enable Digital output UDP packets.

Analogue input options.

Bit	Value	Description
0	1	None.

Analogue output options.

Bit	Value	Description
1	2	Update DMX channels with analogue output changes.

Digital input options.

Bit	Value	Description
0	1	None.

Digital output options.

Bit	Value	Description
0	1	None.

Temperature sensor options.

Bit	Value	Description
0	1	None.

Scene options.

Bit	Value	Description
0	1	Scenes setpoints target analogue outputs.
1	2	Scenes setpoints target DMX channels.

Note: scenes can target analogue outputs and/or dmx channels.

3.4.42 SP - Set password

Set a password. SP<level>; <new password>:

level is the password level number 1-3. password is the new password to set. If the same password is set at multiple levels then login using that password will set itself to the highest of the levels using that password string. If Level 1 password is blank the WebBrick will default to being logged in at level 1 at start and after login timeout. This enables the Home page controls.

3.4.43 SR - Rotary Encoder Step

Configure rotary encoder. SR<chn>;<Steps>:

Steps should be between 2-254, Analogue outputs are set in the raw range 0-1023, where 1023 is 5V, Steps controls how far the output is indexed 'up' or 'down' for each step turn of the rotary encoder. The configuration value sets the step for all rotary encoders and chn should be 0. Older webbricks had dedicated pins for rotary encoders and this only targetted analogue 0, Since 6.4 a rotary encoder may be connected to any even/odd pair of digital inputs and target any analogue output or what ever the trigger definition can target.

3.4.44 SS - Set Operating Mode

Set operational state SS<ToD>:

ToD is a value between 0-255 that lets the WebBrick know a bit about its operating environment, the current recognised values are:

Value	Description
0	This locks out any commands from the WebBrick holding any outputs at their current state (any Dwell in progress will complete).
1	Holiday state, digital inputs ignored.
2	This is normal operation.

3.4.45 ST - Set Time

ST<dd>; <hh>; <mm> Set Time

Set the WebBrick clock. dd is day number from 0-6. hh and mm are the 24 hour time. The WebBrick will send out starting packets until the clock is set.

3.4.46 TA - Adjust Analogue Input Threshold

TA<chn>; <L|H|B>; <Threshold>;

Configure the active threshold on one of the analogue inputs. This does not update the persistent configuration only the active configuration. L or H or B identify whether this is setting a High or Low or Both threshold and Threshold is the analogue threshold as a value from 0-100%. The low level hardware is 0-5V but so as to avoid issues when some signal conditioning is added the input is scaled.

3.4.47 TT - Adjust Temperature Sensor Threshold

TT<chn>; <l|H|B>; <Threshold>;

Configure the active threshold on one of the temperature sensors. This does not update the persistent configuration only the active configuration. L or H or B identify whether this is setting a High or Low or Both threshold and Threshold is the temperature threshold as a decimal temperature between -50 and +150 to 1 decimal place degrees Celsius. i.e. these are valid 50, 43.3, -43, -54.9. These are not valid 34.95, -43.12.

4 IO Board

The V6 webbrick comes with an IO board that provides screw terminal access to all connections other than the network cable. It also provides 4 switched mains outputs and 2 double pole changeover relay outputs.

Layout diagram In here.

4.1 Mains Switching

The mains outputs are digital channels 1 to 4 and can each switch up to 500W, with a maximum of 1500W for the four channels. There is a 6.3 Amp fuse in the unit that will blow if you excede this loading. These are zero voltage switched and intended for simple on off actions. To handle dimming use the analogue outputs connected to external dimmers.

4.2 Relays

Digital channels 4 and 5 are connected to relays that provide a doubel pole changeover capability and are rated for 4A at 230V. Due to internal spacing these should not be used for low voltage applications if the mains switching capability is used on the Triacs.

4.3 Connections

The IO board has a total of 6 connectors on it, 2 are 9 way and 4 are 12 way. The 9 way connectors are used for the mains outputs and the 12 way for inputs and low voltage outputs.

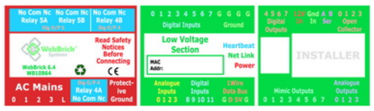


Figure 2: WebBrick IO connectors

Connector X1	Relay connections
Pin 1	Mains Triac 0
Pin 2	Mains Triac 1
Pin 3	Mains Triac 2
Pin 4	Mains Triac 3
Pin 5	Mains In
Pin 6	Relay1 NO1
Pin 7	Relay1 C1
Pin 8	Relay1 NC1
Pin 9	Mains Earth

Connector X2	Triac/Mains and more Relay connections
Pin 1	Relay1 NO2
Pin 2	Relay1 C2
Pin 3	Relay1 NC2
Pin 4	Relay2 NO1
Pin 5	Relay2 C1
Pin 6	Relay2 NC1
Pin 7	Relay2 NO2
Pin 8	Relay2 C2
Pin 9	Relay2 NC2

Connector U1	
Pin 1	0V-Digital Ground
Pin 2	0V-Digital Ground
Pin 3	0V-Digital Ground
Pin 4	0V-Digital Ground
Pin 5	Digital In 7
Pin 6	Digital In 6
Pin 7	Digital In 5
Pin 8	Digital In 4
Pin 9	Digital In 3
Pin 10	Digital In 2
Pin 11	Digital In 1
Pin 12	Digital In 0

Connector U2	
Pin 1	Analogue In 0
Pin 2	Analogue In 1
Pin 3	Analogue In 2
Pin 4	Analogue In 3
Pin 5	Digital Input 8 - may be labelled Digital Monitor 1
Pin 6	Digital Input 9 - may be labelled Digital Monitor 2
Pin 7	Digital Input 10 - may be labelled Digital Monitor 3
Pin 8	Digital Input 11 - may be labelled Digital Monitor 4& IR In
Pin 9	0V-Digital Ground
Pin 10	1 Wire data bus
Pin 11	1 Wire 5V Supply
Pin 12	0V-Digital Ground

Connector U3	
Pin 1	Open Collector Digital 0
Pin 2	Open Collector Digital 1
Pin 3	Open Collector Digital 2
Pin 4	Open Collector Digital 3
Pin 5	Special Connection A
Pin 6	Special Connection B
Pin 7	0V-Digital Ground
Pin 8	12V Supply In
Pin 9	Digital 4
Pin 10	Digital 5
Pin 11	Digital 6
Pin 12	Digital 7 & IR Out

Connector U4	
Pin 1	Mimic 0
Pin 2	Mimic 1
Pin 3	Mimic 2
Pin 4	Mimic 3
Pin 5	Mimic 4
Pin 6	Mimic 5
Pin 7	Mimic 6
Pin 8	Mimic 7
Pin 9	Analogue Out 0
Pin 10	Analogue Out 1
Pin 11	Analogue Out 2
Pin 12	Analogue Out 3

5 Xml Structure

A WebBrick can serve up two Xml files, one of these (WbStatus.xml) provides the current status of dynamic information, i.e. current output state and the other (WBCfg.xml) contains the complete WebBrick configuration.

The element names are chosen to identify the commands used to modify attributes within the PIC chip.

The XML is intended to be consumed by other computer software more than used to display to the end user in its raw form. If you try and look at it in a normal text editor you will find some very long line lengths.

5.1 Trigger Encoding,XML

When a trigger sequence is sent out from the webbrick it is encoded into 3 bytes encoded as follows.

ConfigByte1 This uses the following Bit fields

- Bits 0-4 The encoding of these bits depends on the command group.
- Bits 5-6 These bits select a command group.
  - 0 General commands
  - 1 Not currently used
  - 2 Not currently used
  - 3 Dwell commands.
- Bits 7 This indicates whether to send a UDP packet.
  - 0 No UDP packet will be sent in response to a input trigger
  - 1 General UDP Packet, see packet contents for further details.

General Commands - these use 5 bits as a command or action.

- 0 No Action
- 1 Off
- 2 On
- 3 Momentary - short duration
- 4 Toggle
- 5 Unused
- 6 Unused
- 7 Next

8	Prev
9	Set Low Threshold
10	Set High Threshold
11	Adjust Low Threshold
12	Adjust High Threshold
13	Send Infra Red
14	Up
15	Down
16	Set DMX channel
17	Count

Dwell Commands - these use 2 bits as a command and 3 bits to select a dwell number.

0-2	Dwell number or index
3-4	Dwell command

0 - 00	Dwell On
1 - 16	Dwell Off
2 - 32	Dwell Cancel
3 - 48	Dwell Always

ConfigByte2 This uses the following bit fields.

Bit 7 If this bit is set then the target channel is analogue and the following analogue format applies.

Analogue format.

Bit 7	bit is set
Bits 0-3	This is the setpoint to be used in analogue commands
Bits 4-6	This is the analogue channel on which the action is to be performed.

Non analogue format.

Bit 7	bit is clear
Bit 6	If Clear then it is a digital channel, if set then it is a Scene.
Bits 0-3	This is the digital channel or scene number.
Bits 4-5	Reserved for other possible options, keep as 0.

ConfigByte3 This is the remote node to be triggered for remote actions. It will be set in all UDP packets so can be used for other purposes when type is General or Alarm.

ConfigByte4 This is a reserved byte not yet used.

5.2 WbStatus.xml

Element	Description
WebbrickStatus	Overall Element in WbStatus.xml
SN	Node number
Error	Response to last command
Context	Operational State
LoginState	Login level
DI	Digital input state
DO	Digital output state
Clock	Current webbrick clock
Date	Not implemented
Time	Current webbrick time
Day	Current webbrick day of week
OWbus	One wire bus status
Tmps	Temperature elements
Tmp	Temperature id nn values
AOs	Analogue output elements
AO	Analogue output id nn values
AIs	Analogue input elements
AI	Analogue input id nn values

For example of using this see wb6Status.py in the python API or wb6Status.php in the PHP API.

5.2.1 Error

The element contents are 0 or an error code if the last webbrick command was invalid.

Value	Description
0	No Error
1	Bad Command
2	Bad Parameter
3	Not Logged In
4	Address Locked, attempt to change IP address on address locked webbrick.
5	In Startup
6	No Command yet issued

5.2.2 Context

Operating context.

Value	Description
0	Startup

5.2.3 LoginState

Login state.

Value	Description
0	Logged out
1	controls enabled
2	configuration
3	installer

5.2.4 DI

Digital input state, bit mapped integer where bit 0 is the current state of digital input 0 (when using a zero base).

5.2.5 DO

Digital output state, bit mapped integer where bit 0 is the current state of digital output 0 (when using a zero base).

5.2.6 Clock

Contains the current time as known by the WebBrick, the XML includes full date but this is not implemented within the system at present. The element Date within Clock contains date string and Time contains the time in 24 hour format.

5.2.7 OWBus

Contains the status of the one wire bus. The value of 255 shows no sensors detected and a small integer is a bit mapped value with a bit for each temperature sensor.

5.2.8 Tmps

Contains one entry for each possible temperature sensor.

5.2.9 Tmp

Contains an id attribute, the current low threshold and high thresholds as attributes and the current temperature reading. Note these are in 16ths of a degree so convert to float/real number and divide by 16 to get value.

5.2.10 AO

Contains the current analogue output value for an analogue output channel, this is a number from 0-100 where 100 is 10V output.

5.2.11 AI

Contains an id attribute, the current low threshold and high thresholds as attributes and the current analogue reading. Note these are in the range 0 - 100 where 100 is 5V at the terminal.

5.2.12 Sample wbStatus.xml

The following is a sample copy of a webbrick Xml status file.

5.3 WbCfg.xml

Element	Description
WebbrickConfig	Overall Element in WbStatus.xml
NN	Node Name
SN	Node number
SRs	Rotary encoder elements
SR	Rotary encoder step value
IR	Infra red address received by webbrick on IR reception.
SF	Analogue fade rate
CDs	digital input elements
CD	digital id nn input
Trg	Trigger setting
TrgL	Low threshold and trigger setting
TrgH	High threshold and trigger setting
CCs	Scene elements
CC	Scene id nn settings
CWs	Dwell elements
CW	Dwell id nn setting
CSs	Set point elements
CS	Set point id nn
CTs	Temperature elements
CT	Temperature input id nn value,name,thresholds
CIs	Analogue input elements
CI	Analogue input id nn value,name,thresholds
CEs	Scheduled event elements
CE	Scheduled event id nn
NOs	Digital output elements
NO	Digital output id nn name
NMs	Monitor input elements
NAs	Analogue output elements
NA	Analogue output id nn name
MM	Mimic configuration

5.3.1 WebbrickConfig

Contains the firmware version number

5.3.2 NN

Contains the configured node name.

5.3.3 SI

Contains the IP address and MAC address of the webbrick.

5.3.4 SN

Contains the configured webbrick node number.

5.3.5 SR

Contains attributes id and the Value for the rotary step value.

5.3.6 SF

Contains the fade rate counter, this controls how fast the analogue channels fade up and down.

5.3.7 CDs

Container element for all the digital input channel elements.

5.3.8 CD

Configuration details for a digital input. Contains an id, the name and the options for the input and a nested trigger element. See start of section for trigger details.

5.3.9 CCs

Container element for all the scene elements.

5.3.10 CS

Configuration details for a scene. Contains an id attribute and controls for the digital outputs and analogue outputs. The Dm attribute identifies which digital channels are affected by this scene and Ds for those channels whether to switch the channel off or on. Am identifies which analogue channels are affected and Av the set points for those channels. Note when a scene is selected by a trigger the trigger action will override the On action here, so that a scene can be dwelled etc. If the scene says Off then Off it goes.

5.3.11 CWs

Container element for all the dwell values.

5.3.12 CW

Contain the configured dwell values in seconds.

5.3.13 CSs

Container element for all the analogue set point values.

5.3.14 CS

Contains each of the set point values in the range 0-100.

5.3.15 CTs

Container element for all the temperature sensor configuration.

5.3.16 CT

Contains an id and a name for a temperature sensor and an embedded trigger along with the threshold at which it occurs. TrgL is the low threshold amd TrgH is the high threshold for the sensor. Threshold values are in 16ths of a degree.

5.3.17 CIs

Container element for all the analogue input configuration.

5.3.18 CI

Contains an id and a name for an analogue channel and embedded triggers along with the threshold at which they occurs. TrgL is the low threshold amd TrgH is the high threshold for the sensor. Threshold values are in the range 0-100.

5.3.19 CEs

Container element for all the scheduled event configuration.

5.3.20 CE

Each element contains an id, Days, Hours, Minutes attribute. The Days attribute is bit mapped for days 0-6 and identifies on which days of the week the event occurs. The embedded Trg element is the action to perform.

5.3.21 NOs

Container element for all the digital output elements.

5.3.22 NO

Each element contains an id and a name for a digital output.

5.3.23 NAs

Container element for all the analogue output elements.

5.3.24 NA

Each element contains an id and a name for a analogue output.

5.3.25 IR

Contains the Infra red RC5 address recognised by the webbrick.

5.3.26 MM

Contains the current mimic settings. The 'lo' and 'hi' attributes are the off an on mimic settings, this value is in the range 0-63. The 'dig' attribute is a 32 bit number with 4 bits for each digital output that maps the output channel to a mimic, as there are only 8 mimics a channel number greater than 7 is basically No Mimic. The 'an' attribute provides the same mapping for the analogue outputs. Finally the 'fr' attribute controls how fast the mimics fade from one setting to the next.

5.3.27 Sample wbCfg.xml

The following is a sample copy of a webbrick Xml config file.

6 Setting the IP address

In this section we deal with setting the IP address.

The site player has been given a default IP address of 10.100.100.100 so it initially starts on 10.100.100.100. The IP address can also be set by the PIC chip configuration to overide this value.

The IP address may also of been set to another value at factory and then locked down, to change it the unit will need returning to factory.

6.1 Setting the IP address using Web Interface

To set the IP address you need to be logged into the WebBrick. This is done by selecting the login page , from the menu and then entering the level 2 or 3 password into the login field.

Select the Configure server and enter the new IP address into the correct 4 boxes and then click Save.

See section on Configure Server in Web Interface section.

If you are not sure what IP address the device is using, run either the WebBrickMon application which is a windows application that can be downloaded from http://community.webbricksystems.co or use the python wbMonitor program. Both of these should pick up UDP events being sent by the webBrick and display them. The IP address is listed after the time stamp on the Windows application.

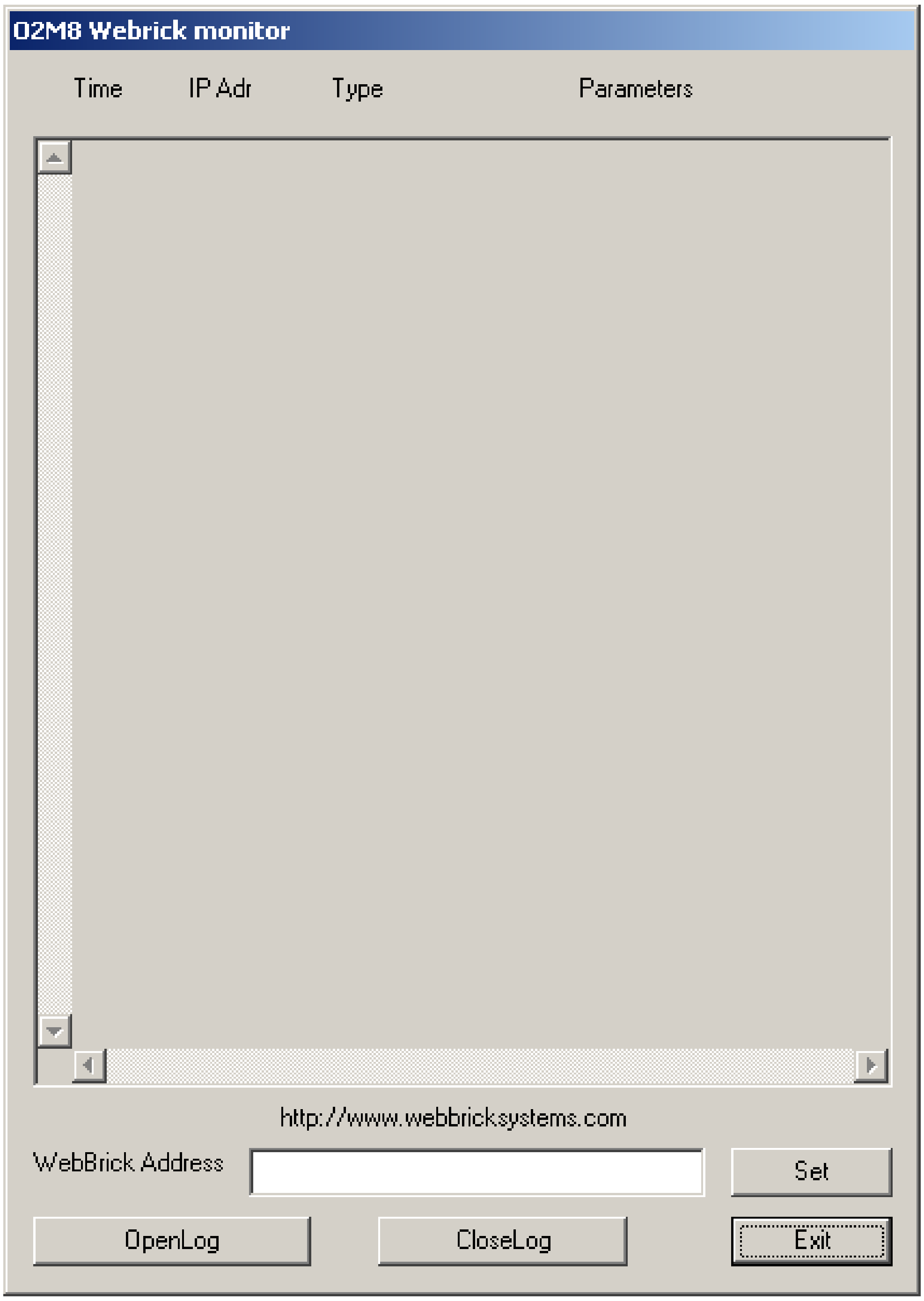
6.2 Setting the IP address using Python

There is a python fragment setIp.py that can be edited and used to set the IP address, it needs the MAC address and new IP address installing in it. It is only intended to be used where the WebBrick is on its default of 10.100.100.100, possibly with other new WebBricks. This is the basis of the home gateway Discovery activity.

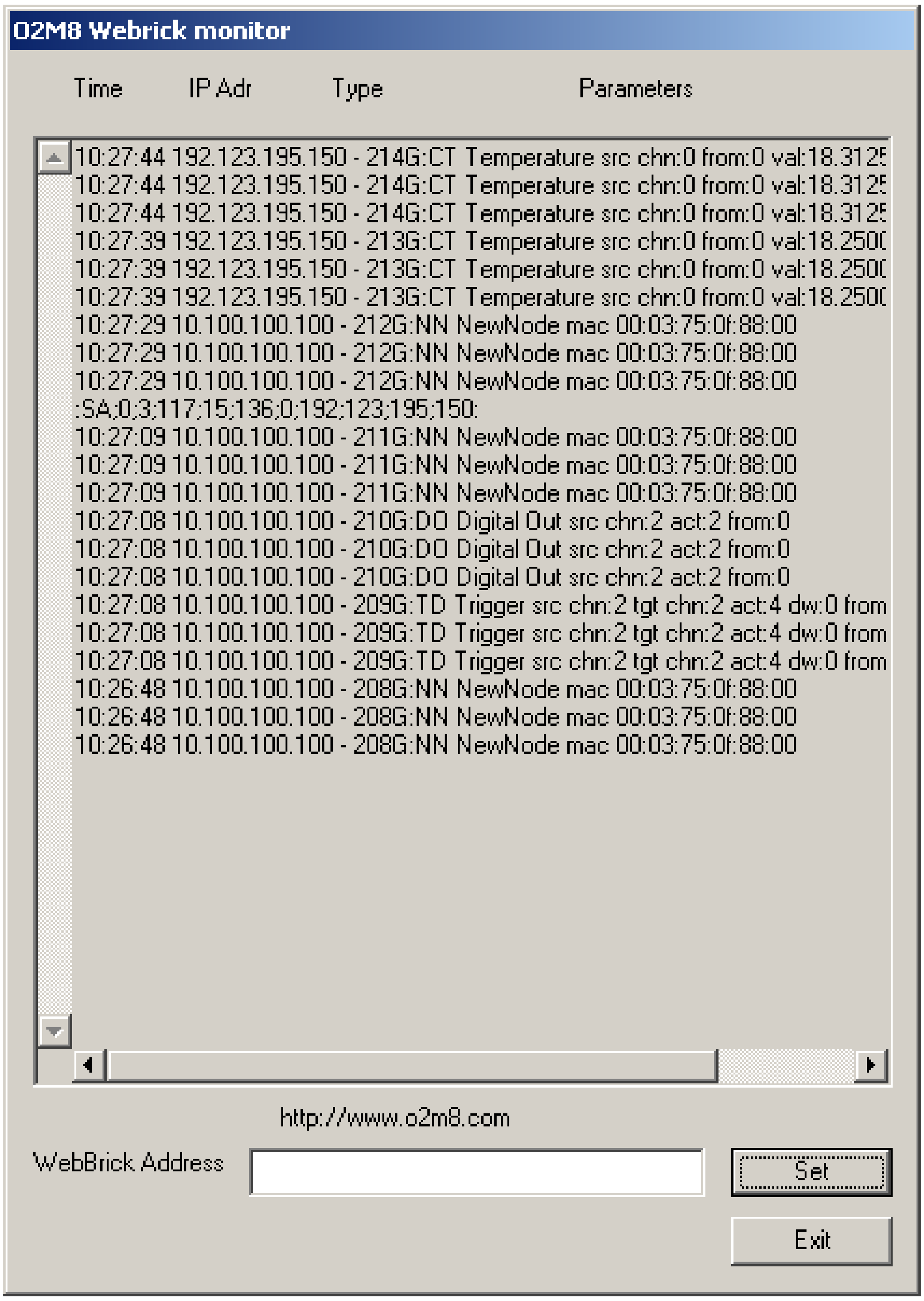
6.3 Setting the IP address using WebBrickMon

WebBrickMon is a standalone windows application, it's primary purpose is for use in initial setup to locate your webbrick on your local network. It communicates with a webbrick to set the IP address using broadcast messages and therefore can work when you caannot access the web interface.

When the monitor is started you will see something similar to this:



The top section shows UDP events coming from the webbrick, if your network is capable of connecting to the IP address listed in the events then just use your web browser to connect to the IP address. If your system uses different IP address ranges then enter aa new IP address in the box at the bottom labelled WebBrickAddress and click SET. The monitor will then wait for the next NewNode event and change its webbrick address. You should then see the IP address in the events change.



```
10:27:44 192.123.195.150 - 214G:CT Temperature src chn:0 from:0 val:18.3125
10:27:44 192.123.195.150 - 214G:CT Temperature src chn:0 from:0 val:18.3125
10:27:44 192.123.195.150 - 214G:CT Temperature src chn:0 from:0 val:18.3125
10:27:39 192.123.195.150 - 213G:CT Temperature src chn:0 from:0 val:18.2500
10:27:39 192.123.195.150 - 213G:CT Temperature src chn:0 from:0 val:18.2500
10:27:39 192.123.195.150 - 213G:CT Temperature src chn:0 from:0 val:18.2500
10:27:29 10.100.100.100 - 212G:NN NewNode mac 00:03:75:0f:88:00
10:27:29 10.100.100.100 - 212G:NN NewNode mac 00:03:75:0f:88:00
10:27:29 10.100.100.100 - 212G:NN NewNode mac 00:03:75:0f:88:00
:SA;0;3;117;15;136;0;192;123;195;150:
10:27:09 10.100.100.100 - 211G:NN NewNode mac 00:03:75:0f:88:00
10:27:09 10.100.100.100 - 211G:NN NewNode mac 00:03:75:0f:88:00
10:27:09 10.100.100.100 - 211G:NN NewNode mac 00:03:75:0f:88:00
10:27:08 10.100.100.100 - 210G:DO Digital Out src chn:2 act:2 from:0
10:27:08 10.100.100.100 - 210G:DO Digital Out src chn:2 act:2 from:0
10:27:08 10.100.100.100 - 210G:DO Digital Out src chn:2 act:2 from:0
10:27:08 10.100.100.100 - 209G:TD Trigger src chn:2 tgt chn:2 act:4 dw:0 from
10:27:08 10.100.100.100 - 209G:TD Trigger src chn:2 tgt chn:2 act:4 dw:0 from
10:27:08 10.100.100.100 - 209G:TD Trigger src chn:2 tgt chn:2 act:4 dw:0 from
10:26:48 10.100.100.100 - 208G:NN NewNode mac 00:03:75:0f:88:00
10:26:48 10.100.100.100 - 208G:NN NewNode mac 00:03:75:0f:88:00
10:26:48 10.100.100.100 - 208G:NN NewNode mac 00:03:75:0f:88:00
```

7 UDP Output

Any trigger (Digital, analogue threshold, temperature threshold, scheduled event, Infra Red receive) can be configured to send a UDP packet with a configured type indicator, one of General, Remote, Alarm. Output channels also send UDP packets to indicate a change has occured this can be used to confirm the operation of remote commands.

The UDP packet format is the same on most occasions (exception being unconfigured node) although some fields may not be used and may contain garbage data.

7.1 General UDP Packet format

The UDP Packet Format is formed as follows:

Field	Description
Len (byte)	The overall size of the UDP packet.
udpType (byte)	The packet type one of the characters 'A', 'G', 'R'.
Source 0-1 (2bytes)	These are two characters to identify the Udp source see later table.
srcChannel (byte)	The channel/event index.
tgtChannel (byte)	The target channel when sending remote commands, if the target channel type is analogue then the top bit is set, i.e. values are 128+channel.
action (byte)	The action being triggered (low 4 bits), includes Dwell number (high 4 bits).
fromNodeNr (byte)	Source webrick number 1-253.
toNodeNr (byte)	Target webrick number for remote commands.
setPointNr (byte)	Where relevant for packet type.
curValH (byte)	High byte of any value being sent
curValL (byte)	Low byte of any value being sent

7.1.1 UDP Packet Types

The UDP Packet Types are:

Type	Description
G	General format UDP Packet generated

There used to be a Remote and Alarm type in pre 6.6 firmware, this was never used in any applications and the space used to define these has been recovered for other uses.

7.1.2 UDP Packet Sources

The UDP Packet Sources are:

Code	Description
Ta	Low analogue threshold trigger
TA	High analogue threshold trigger
Td	Trigger from remote DI command
TD	Trigger from local digital input
Tt	Low temperature threshold trigger
TT	High temperature threshold trigger
TS	Trigger from scheduled event
TR	Trigger from infra red remote control
TX	Trigger from external source, i.e. WebBrick Gateway
IR	infra red remote control
AI	New analogue input value 0-100
AO	New analogue output value 0-100
CT	Current temperature in 1/16ths degree
DO	Digital output/monitor state.
NN	Unconfigured node.
SS	Node starting, clock not set.
AT	Atention button pressed on webbrick, AKA Factory Reset button.
AA	Dicovery triggered response, Alert.
CC	Configuration changed.

Further details on each packet follow.

Trigger

For this packet format:

1. the srcChannel identifies the channel number or scheduled event number. In the case of the monitor inputs this is the monitor number + number of digital inputs. In a later revision of the firmware the monitor inputs will just be other digital inputs and can be a normal trigger. In the case of external source this will be zero.
2. the tgtChannel is the remote webrick node number from the trigger configuration
3. udpType is from the trigger configuration
4. action is from the trigger configuration action and dwell number if the trigger type is analogue
5. fromNodeNr is my node number
6. toNodeNr is from the trigger configuration and is only relevant when the udpType is 'R' remote.
7. SetPointNr comes from trigger configuration.
8. CurValH is not used.

9. CurValL is used for local digital inputs and is an estimate of the number of times the input is triggered in the last second.

7.1.3 UDP - Analogue Values

Analogue values

For this packet format:

1. the srcChannel identifies the analogue channel number
2. the tgtChannel is not used
3. udpType is 'G' General
4. action is not used
5. fromNodeNr is my node number
6. toNodeNr is not used
7. SetPointNr is not used
8. CurValH is not used.
9. CurValL is the new value for the input or output in the range 0-100.

7.1.4 UDP - Digital Output and Input States

Digital/Monitor State

For this packet format:

1. the srcChannel identifies the digital output channel number.
2. the tgtChannel is not used
3. udpType is 'G' General
4. action is either DINACTIONON or DINACTIONOFF
5. fromNodeNr is my node number
6. toNodeNr is not used

7. SetPointNr is not used
8. CurValH is not used.
9. CurValL is not used.

7.1.5 WebBrick Starting

WebBrick node starting
For this packet format:

1. fromNodeNr is my node number

All other fields are not used.

7.1.6 UDP - Unconfigured Node

Unconfigured
For this packet format the payload does not follow the standard format, the first 6 bytes of the payload are the MAC address of the network interface.

7.2 WebBrick Time and UpTime format

This packet has the following format:

1. the srcChannel identifies the digital output channel number.
2. i the tgtChannel is not used
3. udpType is ’G’ General
4. ’S’
5. ’T’
6. WebBrick Time - HOURS, taken from the Real-Time Clock
7. WebBrick Time - MINUTES, taken from the Real-Time Clock
8. WebBrick Time - SECONDS, taken from the Real-Time Clock

9. WebBrick Time - DAY, taken from the Real-Time Clock
10. WebBrick Uptime, upper byte
11. WebBrick Uptime, lower byte
12. ResetCode

ResetCode can have the following values:

1. '0' This means that the SitePlayer (WebServer) was reset by either the internal watchdog, or a manual 'RS' command
2. 'value of RCON', this is the internal value of the Reset Register of the WebBrick processor (Normally '92' for 6 and 7 series WebBricks)

8 WebBrick and Python

8.1 Introduction

Python is a very useful langauge. it can be used by almost anyone and allows for a full range of programming from simple scripting to full on object orientated programming. Webbrick Systems have used Python to implement the interface service between the webbrick and ITunes (Apple).

For the WebBrick we provide a number of Python resources, these fall into the following catergories:

Class Libraries	These allow the user to create programs that interact with WebBricks using routines that have been fully tested and debugged. This allows the programmer to get on with building the automation they desire rather than getting bogged down in the details of network protocols.
Utilities	These allow for common operatios such as uploading and downloading WebBrick configuration files.
Example Code	This allows people to quickly build on some of the ideas we already had and implemented.

8.2 Class Libraries

8.3 wb6.py the WebBrick class library

This python module provides basic commands that can be sent to a WebBrick.

To import this library use the following in your code:

```
import wb6
```

From this point you can use this module to interact with WebBricks, for example if you wanted to switch on a particular output channel you might use somethine like

wb6.DigOn('10.100.100.100', 3)

8.4 Functions implemented by wb class

8.4.1 getStatusXml

getStatusXml(<adrs>)

Retrieve the Xml blob that describes the current status of the addressed WebBrick <adrs> is the Ip address/Dns Name of the WebBrick being targetted.

8.4.2 getConfigXml

getConfigXml(<adrs>)

Retrive the Xml blob that describes the current configuration of the addressed WebBrick <adrs> is the Ip address/Dns Name of the WebBrick being targetted.

8.4.3 DigTrigger

DigTrigger(<adrs>, <chn>)

Generate a digital trigger on a digital input

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <chn> is the channel on the webBrick being targetted

8.4.4 DigOn

DigOn(<adrs>, <chn>)

Set a digital On

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <chn> is the channel on the webBrick being targetted

8.4.5 DigOff

DigOff(<adrs>, <chn>)

Set a digital Off

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <chn> is the channel on the webBrick being targetted

8.4.6 DigToggle

DigToggle(<adrs>,<chn>)

Toggle a digital Off

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <chn> is the channel on the webBrick being targetted

8.4.7 DigDwell

DigDwell(<adrs>,<chn>,<DwellNr>)

Set a digital channel on for Dwell Time

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <chn> is the channel on the webBrick being targetted <DwellNr> is the dwell number

8.4.8 AnOutSp

AnOutSp(<adrs>,<chn>,<sp>)

Set an analogue channel to preset sp.

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <chn> is the channel on the webBrick being targetted <sp> is the preset number

8.4.9 AnOutPercent

AnOutPercent(<adrs>,<chn>,<val>)

Set an analogue channel to a specific per-cent level.

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <chn> is the channel on the webBrick being targetted <val> is a value 0-100 where 100 if 10V output

8.4.10 Send

Send(<adrs>,<cmd>)

Send a specific WebBrick command.

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <cmd> is the command to be sent, see the section on Commands.

8.4.11 GetXml

result = GetXml(<adrs>,<xmlName>)

Retrieve a specific Xml blob from a webBrick.

<adrs> is the Ip address/Dns Name of the WebBrick being targetted. <xmlName> The name of the Xml blob, currently xmlStatus.xml and xmlCfg.xml. <result> the return value is the Xml if the parameters are valid

8.4.12 wbUdpEvents.py

This is a class that captures and delivers Udp events to an event target. To use this you create an event target derived from wbUdpEvents.udpPacket for the events to be delivered to and pass this when you create wbUdpEvents.wbUdpEvents.

wbMonitor is an example using this, See under Python Utilities.

8.4.13 wbXmlEvent.py

This is a class that builds upon wbUdpEvents to turn the UDP events from a webbrick into an Xml format, another example of an Event target for wbUdpEvents. wbXmlEventTest.py (see under Python Utilities) uses this class.

8.4.14 WebBrickConfig.py

This class contains methods to read a WebBrick XML configuration file, and convert them into WebBrick configuration commands. There is a command interface to this class that uses it to save and restore configuration to/from disk. See under Python Utilities.

8.4.15 wbStatus.py

This class retrieves a copy of the current WebBrick Status that can then be asked for specific details. Note the object reads and caches the values, so you get a current snapshot. To get updated values create a new object. There is no point retrieveing tha values at more a 1 second interval and it is recommended that you do as much as possible with a single retrieved snapshot.

1. wbStatus(<adrs>) Construct a wbStatus object. <adrs> is the Ip address/Dns Name of the WebBrick from which the status should be retrieved.
2. digOutState = digOutState(<chn>) return current status of a digital output. <chn> the digital output channel number. <result> One of True or False for On or Off.
3. digInState = digInState(<chn>) return current status of a digital input or monitor input (channels 8-11). <chn> the digital input channel number. <result> One of True or False for On or Off.
4. anOutVal = anOutVal(<chn>) return current setting of an analogue output channel. <chn> the analogue output channel number. <result> 0-100 where 100 is an output of 10V

5. anInVal = anInVal(<chn>) return current level of an analogue input channel. <chn> the analogue input channel number. <result> 0-100 where 100 is an input of 5V
6. tempVal = tempVal(<chn>) return current temperature. <chn> the analogue input channel number. <result> a signed temperature value multipled by 16, temperarure resolution is to 1/16 of a degree celcius.
7. getTime = getTime() return current time at webBrick. <result> The time as a formatted string.
8. getHour = getHour() return current hour at webBrick. <result> The hour in 24 hour format 0-23
9. getMinute = getMinute() return current minute at webBrick. <result> The minute as 0-59
10. getSecond= getSecond() return current second at webBrick. <result> The second as 0-59
11. getDay = getDay() return current day at webBrick. <result> The day as a number 0-6 for Sunday through to Saturday.
12. getVersion = getVersion() return firmware version for WebBrick. <result> String version number.
13. getOpState = getOpState() return operational state of webBrick. <result> See comnmand section for values.
14. getLoginState = getLoginState() return login level at webBrick. <result> 0-3.
15. getLastError = getLastError() return result of last command to webbrick. <result> 0 was no error.

8.4.16 wbCurrentCfg.py

This class retrieves a copy of the current WebBrick configuration that can then be asked for specific details. Not yet implemented.

1. wbCfg(<adrs>) Construct a wbCfg object <adrs> is the Ip address/Dns Name of the WebBrick from which the configuration should be retrieved.
2. NodeName = NodeName()Retrive node name
3. NodeNumber = NodeNumber()Retrive node number

8.5 Python Utilities

8.5.1 wbMonitor.py

Uses wbUdpEvents to display received webbrick events.

8.5.2 wbXmlEventTest.py

Uses wbXmlEvent to display received webbrick events as Xml blobs.

8.5.3 WbCfg.py

This program is used to configure a WebBrick from a configuration file and to save there configuration. It uses the class WebBrickConfig.

Usage wbCfg.py prog [options] <WbAdrs>

option	option parameters	Description
-d,–display		Display WebBrick configuration (default)
-s,–save	(<fileName>) <fileName>	save webbrick configuration name of file to save configuration to.
-u,–update	(<fileName>) <fileName>	Update WebBrick configuration name of file to read configuration from.
-p,–password	(<password>) <password>	Access password Access password, default password.
-t,–trace		Trace operation

You should note that during configuration we set the operating state - to '1' (startup) to disable all the functions of the WebBrick.We do this to ensure that there are no extraneous actions or commands in progress that would interrupt or corrupt the configuration process.

The save format at present is the wbCfg.xml format.

8.6 Python Examples

There are some Python examples that you can use to generate the command packets that WebBricks will respond to.

8.6.1 wbDigOut.py

This program turns a single digital channel on off or toggles it.

usage: python wbDigOut.py <ipAdrs> <channel> <action>

8.6.2 wbExcercise.py

This program just repeatadly toggles all the digital channel in turn. If you have LEDs attached to the outputs you will get a NightRider type scan.

9 WebBrick and PHP

9.1 Introduction

PHP is a web application langauge that allows you to mix business logic and user interface logic in a single file, this file is processed on a web server to mix fixed and variable parts of the user interface. http://www.php.net

For the WebBrick we provide a number of PHP resources, these fall into the following catergories:

Libraries	These allow the user to create programs that interact with WebBricks using routines that have been fully tested and debugged. This allows the programmer to get on with building the user interface they desire rather than getting bogged down in the details of network protocols.
Example Code	This allows people to quickly build on some of the ideas we already had and implemented.

THESE PHP resources are not actively maintained and are provided AS IS with no support, they are similar to the python code so refer to the python section for details. Your PHP will need to have an XML parser as part of it's install for the status and configuration classes to work.

9.2 Libraries

9.2.1 wb.php basic comms libray

This php module provides basic communication with the WebBrick and is mainly intended to be a building block.

To use this library use the following in your code:

```
include("../API/wb.php") ;
```

From this point you can use this module to interact with WebBricks, for example if you wanted to switch on a particular output channel you might use somethine like

wbSendCmd('10.100.100.100', 'DO3;N')

9.2.2 wb6.php the WebBrick 6 library

This module provides basic commands that can be sent to a WebBrick.

To import this library use the following in your code:

```
include("../API/wb6.php") ;
```

From this point you can use this module to interact with WebBricks, for example if you wanted to switch on a particular output channel you might use somethine like

wb6DigOn('10.100.100.100', 3)

9.2.3 wb6Status.php

This class retrieves a copy of the current WebBrick Status that can then be asked for specific details. Note the object reads and caches the values, so you get a current snapshot. To get updated values use the load method. There is no point retrieveing tha values at more a 1 second interval and it is recommended that you do as much as possible with a single retrieved snapshot.

9.2.4 wb6Cfg.php

This class retrieves a copy of the current WebBrick configuration that can then be asked for specific details.

Trigger results are returned as an asociative array with some of the following parameters in it, refer to commands section for details on these values.

attribute name	Description
Name	channel name
Value	threshold value
Hours	scheduled time
Minutes	scheduled time
Days	scheduled days
actionNr	action type number
action	action type string
dwell	dwell number
UDPRemNr	UDP packet type number
UDPRem	UDP packet type string
RemNode	target remote node.
typeNr	channel type number
type	channel type string
setPoint	setpoint number
pairChn	target channel number

Scene results are returned as an asociative array with the following parameters in it.

attribute name	Description
Digital<nn>	Ignore—On—Off
Analogue<nn>	Ignore—SetPoint<mm>

where

<nn> is a channel number and <mm> is a setpoint number.

9.3 PHP Examples

There is a PHP example that you can modify to produce your own central user interface.

9.3.1 PanelLib.php

This provides a library of code for a user interface panel that is capable of displaying details from one or more webBricks and issuing commands to them. This uses an Xml file standard.xml to define a user interface that consists of a number of columns that contain status information and/or commands.

The xml file has a top level element called status with a single attribute cols that specifies the number of columns on the display. There are then a set of elements called item. Each item has a Type element and typically a channel number and an IP address to identify the webbrick and the data to access from it. Each entry may also contain a Trig element that is used to specify a command to be issued.

item Type may be one of:

type	Description
Header	generates a text of Cols width.
Temp	displays a temperature.
inAnalogue	displays an analogue input value.
Analogue	displays an analogue output.
Switch	displays a digital output state.
Monitor	displays a monitor state.

9.3.2 standard.php

This is the start page for the standard panel.

10 Applications

Here we provide some examples of use of the WebBrick. The great thing with the WebBrick is that a single device can provide more than one activity. For example PIRs, can trigger video recording at all times. Can switch lights at night. Sound warning tones. In typical monitoring systems you need multiple devices because they do not interlink.

10.1 Heating

The notes in this section mostly consider that a WebBrick is taking the whole controller role. Of course it's possible to add extra global intelligence using a general purpose host computer, for example the WebBrick Gateway.

10.1.1 Thresholds as set points

In the WeBrick architecture we use temperature sensors rather than thermostats. We do this for a variety of reasons:

- Controls - there are no external controls for a temperature sensor. This means that it can be sited where it both makes sense and is convenient. For example under a kitchen worktop or directly bonded to a heated floor.
- Accuracy and repeatability - Old fashioned bi-metal strip thermostats close at one temperature and open at another. Adjusting the set-point is a mechanical operation. Whereas the sensors we use have resolution to 0.1 deg C and an accuracy of 0.5 deg C.
- Remote Control - Using sensors to read the temperature means that the actions on thresholds are defined within the WebBrick. Therefore the set points can be adjusted on a web page

With version 6.1 of the WebBrick it is now possible to control thresholds as an action. Actions can be triggered from Digital Inputs and from Schedules. We can also raise actions from multiple temperature sensors, so we can create zones and frost-stat functionality.

Therefore if we build a heating control scheme using WebBricks we can schedule changes of set-points using thresholds. We can also create *Boost* buttons that increase set-points.

10.1.2 Modern Combi boiler, single Zone

These generally have power applied all the time which is used to provide hot water and the central heating is controlled by a mixture of a thermostat and timed events. In this case we would use scheduled events to control the triacs and relays. For example the output a triac feeding a relay that is controlled by one of the temperature sensors using under and over thresholds. In effect the triac replaces the time switch and the relay replaces the standard thermostat. The great benefits of this are:

- that the control can then be overridden by local buttons for boost (e.g. Dwell 1 hour on boiler output)
- the use of a Home gateway system can use its calendar to switch heating off when home is empty.
- a home gateway can access local weather feeds to adjust switch on times dependant on temperature forecasts.
- using the home gateway the home owner can control and make changes whilst away from the house, e.g. going home early.

10.1.3 Two zone heating system

In this case we drive the boiler off a mains triac that is driven from a schedule entry. And drive two zone valves off the two relay outputs, these are triggered by high and low thresholds on two temperature sensors. Advantages are as for single Zone.

10.1.4 Electric under floor heating

These are great ways of heating floors for things like bathrooms but electricity is not the cheapest heating source if over used. So it would be nice for them to come on for timed periods and be able to boost at other times. In this case a single output can control the power to the floor controller, this can be switched by timed events for getting up time and a push button to boost it at other times.

Note it is not always best to chain Triac based controllers so if the underfloor heating controller uses a Triac we recommend using a Relay to power the floor from the WebBrick. Also underfloor heating can need power greater than the abilities of the onboard Triacs so an external relay is a good idea.

10.1.5 Air conditioning

Controlled by an external relay as the power consumption of these is higher than can be handled directly. Control by a mix of time schedules and temperature thresholds.

10.2 Outside lighting/Security Lighting

This could be as simple as lighting that is timed to come on at dusk until dawn or it could be controlled by PIR sensors that switch on all the lighting around a house. The PIR sensors may not necessarily be on the same WebBrick as the lights. For example on a big building there may be lights on all 4 sides of the building driven by different webBricks, on sensing movement on one side you may bring that side up to full brightness and all the other sides to half brightness.

10.3 Internal lighting

10.3.1 Night Lights

Especially with the very young and very old it is great to have some low level lighting around a house at night, modern LEDs are very good for this drawing very little power. For one or two devices they can be switched directly from the digital outputs for heavier loads 3 in series with a resistor can be driven from a 12V supply and the open collector driver. If the power supply for a WebBrick is a sealed lead acid (SLA) battery then this could also provide power for the LED lights so that they stay on during power cuts. With the low power requirements the system could stay running off a 7 Amp Hour SLA battery for hours if not days.

Maplin (www.maplin.co.uk) 12V 7Ah battery (MG47B) 25, Charger (LL30H) 13. TLC-direct (www.tlc-direct.co.uk) 12V 6AH battery 10+VAT

10.3.2 Dimming lights

For this: use dimmers that accept a 0-10V DC control voltage and wire these to the analogue outputs from the WebBrick. For example the soundlab G018VA 4 channel dimmer can handle 4 channels of 5A each (Max 16A for the Unit). (See also section on Outside World).

10.4 Alarm Monitoring

By connecting dry contacts from fire and burglar alarms to the WebBrick monitor inputs the WebBrick in association can display the state of the alarm on a central display.The home gateway could then send SMS or email messages to alert the home owner. E.g. via www.textanywhere.com.

10.4.1 Water sensors

In the case of property in low lying areas water sensors could be installed in basements and under floor voids and configured to generate alarms.

10.5 Household services

10.5.1 Water softener

A sensor could be added to a water softener to detect low salt levels and alert the household through a central display, especially useful when the device is installed in a basement where it is easily forgotten!

Capacitive sensors are particularly useful for sensing salt levels where the softener has GRP or plastic side walls.

10.6 Home Cinema

A home cinema setup can include more than just the audio and video equipment, you may also have (for example) roller screens and blackout curtains.

10.7 Access control

10.7.1 Entry gates and doors

With the use of RF remote systems connected to a WebbRick control of electric gates and garage doors can be handled through a single system, the WebBrick. You can open gates and doors from the house before braving the elements to get access to your car and can then close them from the car as you leave the premises. When video is used as well you can open doors from work when your cleaner arrives as you will get a message when the door bell is pushed, you then check the camera system verify the identity and let release the front door.

10.7.2 Camera systems

A PIR trigger can be set up to start a camera system recording video images which are made available to you on a web interface.

11 Interfacing to the outside world

Interfacing to the outside world

Here we deal with the basic principles of getting the WebBrick to control and listen to the outside world.

11.1 Basics

The WebBrick treats the Digital Inputs as push to make inputs. There is a software option that can allow the use on off switches.

To action, the push button should ground the input, pullups are enabled at the PIC chip. In out test applications we use CAT5e cable and have runs of up to 100m between the WebBrick and the switches, use a twisted pair for each input to reduce noise pickup. We have seen pickup cause problems, in our case we grounded unused cores which cured the issues.

The outputs at Digital Output Low (DOL) are TTL and should be treated as such, they have enough current sourcing to drive two LEDs and a Solid State Relay. In our test applications we drive one LED and an SSR simultaneously.

The outputs at Digital Output High (DOH) are open collector capable of sinking 500mA, this means the device being switched is connected to a supply rail and the outputs go low on activation to switch on the device. Do not use more than 12v as the supply to the device.

The monitor inputs have no pullups are very sensitive, so if not connected to anything or connected to a source that does not actively drive high or low tie down to 0V with a 4.7K resistor. Valid input range is 0-5V

The analogue outputs are buffered and provide a 0-10V output good enough to drive a pair of LEDs. The User interface is calibrated as 0-100% so 10% corresponds to 1V unless some other signal conditioning is inserted. Do NOT try to drive an LED and an opto isolator at the same time, the forward drop on the LED's will be different and the results not much fun (you could use separate resistors to drive each).

11.2 Cabling

The webBrick has been designed to use standard CAT 5 networking or structured cabling. CAT 5 cable has consists of 4 pairs of wires in a single sheath. These are colour coded with the first colour being the main colour and the later being just narrow stripes:

- Brown with White and White with Brown.

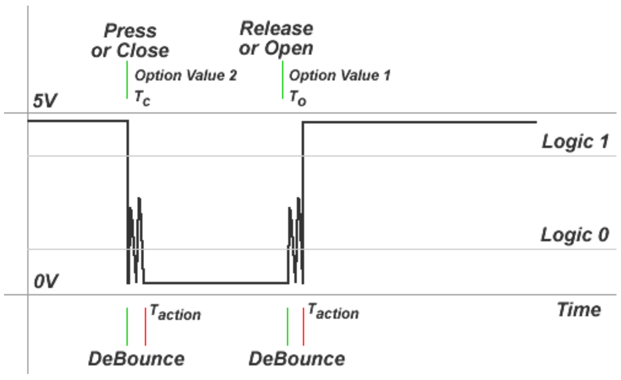
- Green with White and White with Green.
- Orange with White and White with Orange.
- Blue with White and White with Blue.

To reduce any possible interference the pairs should be used properly with a pair for each input or output. Even when multiple inputs are run on the same cable do not use a single ground wire and multiple input wires, use a ground wire for each input wire. Similarly for outputs.

11.3 Digital Inputs

11.3.1 Digital Input Characteristics

Before looking at the components that can be connected to a WebBrick is useful to look at the timing and voltage characteristics of the Digital inputs:



The diagram shows an example of a simple pushbutton connected between a Digital Input and WebBrick Ground.

Whilst the button is open, the voltage on the Digital Input will measure near 5V (typically 4.9). When the button is pressed this will fall to zero or near zero (anything below 0.6V is considered zero).

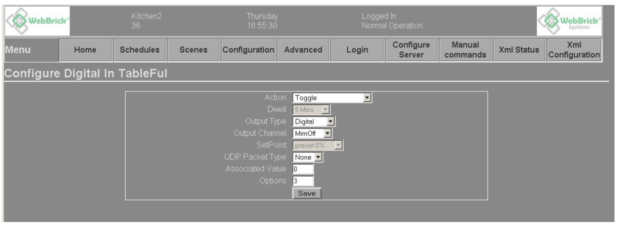
Note that in the diagram we have represented some switch bounce. The WebBrick has internal debouncing of around 20mS.

Because the WebBrick is designed to be used with a range of input devices, it has an option that sets when the state engine get triggered:

Rising Edge Option is set to 1

Falling Edge Option is set to 2 - this is the default behaviour

Both Edges Option is set to 3



Kitchen2
36

Thursday
16:55:30

Logged In
Normal Operation

Configure Digital In TableFul

Action	Toggle
Dwell	5 Mins
Output Type	Digital
Output Channel	MimOff
SetPoint	preset 0%
UDP Packet Type	None
Associated Value	0
Options	3

11.3.2 Switches

For real use, we've found that push buttons with integral LEDs not only look good, but can be very practical.

A background glow is helpful for locating the button in the dark.

We either use a 10K resistor to provide a 'background' glow for the 'Off' condition and a 330ohm or 1k resistor from the corresponding digital output to confirm the 'On' condition.

Note It is quite possible to use simple toggle switches with the WebBrick. However it is likely that you'll want to trigger the state engine on both the 'close' and 'open' action. In this case the option value should be set to '3'. See the section on Digital Input Characteristics for more detail.

11.3.3 PIR Passive Infra Red sensors

These typically have a dry contact that closes for a period of time on detection of movement and can be connected to a WebBrick input.

11.3.4 Light sensor

These can also be connected to a digital input unless you have a type that provides a voltage dependant on the light level then it should be connected to an analogue input.

11.3.5 Security switches

Magnetic reed relays for door and window detection. These are typically used for burglar alarms but can just as easily be used to connect to WebBricks.

11.4 Digital Outputs

11.4.1 Driving LEDs

LEDs are now affordable for just about every kind of use and available in all sorts of colours and power. Therefore its worth a section here describing how they can be used with the WebBrick.

For simple indication purposes a 300R resistor in series with the TTL level digital outputs will give sufficient current for a standard 3/5mm LED. There is still sufficient drive to operate solid state relays.

11.5 Open Collector Outputs

These can sink up 500mA on a channel with a limit of 800mA across all 8 outputs at any one time.

We have also found that modern high brightness white LEDs make very good night lighting that can be switched from a WebBrick using the Open collector driver.

11.5.1 Solid state relays

We've been using the Crydom EZ240D5 solid state relays to great effect. These can be connected directly to the Digital outputs with no other components.

11.6 Analogue Inputs

These can be used to interface anything that provides a 0-5V output or can be set to produce a 0-5V input. You could also connect an ordinary potentiometer to this input. You should not exceed 5V but higher voltages could be reduced using a two resistor potential divider.

11.7 Analogue Outputs

11.7.1 Dimmers

Any dimmer that accepts a 0-10V control signal can be connected to the analogue outputs.

We have used the Velleman K8003 kits now replaced by the K8064.

CPC have some 4 channel dimmers.

`http://cpc.farnell.com/jsp/endecaSearch/partDetail.jsp?SKU=DP26289&N=411`

`http://cpc.farnell.com/jsp/endecaSearch/partDetail.jsp?SKU=DP27530&N=411`

11.8 Mains Outputs

These are zero voltage switched triacs and can handle loads up to 500W, with a total of 1500W across all 4. They should be able to drive all types of load. Note that there is an internal 6.3A fuse.

11.9 Relay Outputs

These are Double pole changeover relays. Particularly useful for curtain motors and other drapery controls. These are rated at 2A per contact.

11.10 Rotary encoder

This is a 3 wire device with a common connection and two signal wires. It is connected to an even/odd pair of digital inputs with the even trigger definition controlling the action when turned one way and the odd trigger defintion the action when turned the other way, i.e. up/down. The target is typically one of the analogue output or next/prev scene. It is possible to configure some weird actions like one way is brighter and the other way is off.

11.11 Using Temperature Sensors

11.12 Temperature Sensors

Here we deal with the basic principles of using the temperature sensors.

11.12.1 Hardware

The temperature sensors are accessed using the DALLAS one wire interface protocol with the master functionality handled by the PIC chip. The Webbrick6 supports up to 5 DS18B20 temperature sensors. These sensors have a 64 bit address that can be used to uniquely identify them and they can read to a resolution below 0.1 degrees C.

11.12.2 Software

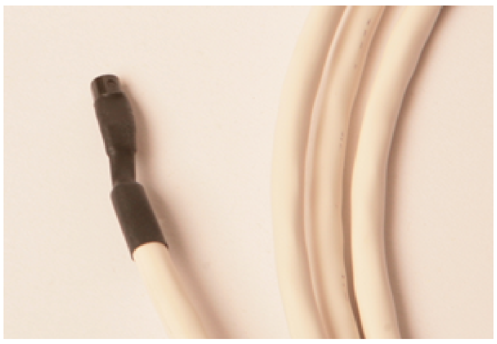
Every second, an attempt is made to read the temperature from one of the sensors and update the status and values. At start up and on command (RT manual command) the bus will be scanned for new devices. When a new sensor is found it is read to see whether it has been tagged previously, if it is untagged or the tag corresponds to a currently active sensor then a new tag number is generated and stored within the sensor. When a web brick is switched on these tags keep the sensors in the same order within the temperature table. The values are all available in the status XML. [NOTE this will change now we have room to store the device addresses in the WebBrick]

When a successful reading is taken it is recorded. Temperature sensor 1 will be tested against the Low and High trigger threshold. If the value is outside these values then the relevant triggers are processed, these are identical to the digital input trigger configuration.

11.12.3 Installing

So as to get sensors to display in your desired order in the list of temperatures. You should connect them one at a time with the first sensor getting tagged to display in 1st row and each subsequent being in subsequent rows. After connecting a sensor issue manual command RT and wait for each it to register before adding the next sensor.

12 Temperature Sensors



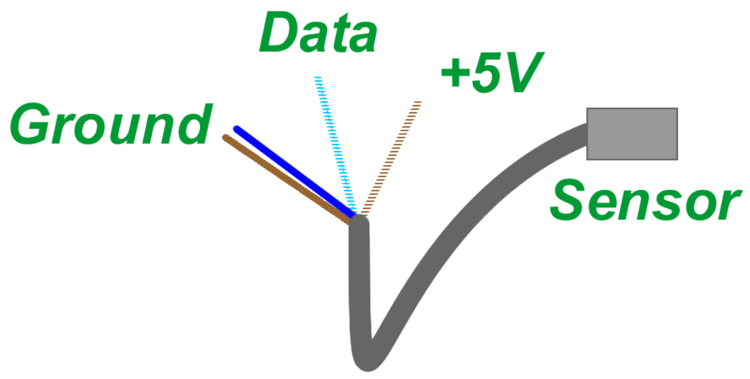
12.1 WebBricks and Temperature Sensors

We supply and recommend Dallas DS18B20 temperature sensors. These are connected to the 1 Wire bus on the WebBrick within the following limits:

- Cable: Cat 5E cable is recommended for temperature sensor connections. Webbrick Systems supply temperature sensors with 2M Cat 5E pre-attached.
- Bus Length: We recommend a maximum bus length of 90M for a single sensor and 50M for multiple sensors. Maximum spur length 2M.
- Unused cores: We recommend that unused cores are connected to ground. In particular the brown core should be connected to ground.
- Sensor Accuracy: Is within 0.5 Deg C.
- Sensor Resolution: Is 0.125 Deg C, although this is less than the accuracy of the temperature sensor it is very useful for determination of whether a temperature is rising or falling

12.2 Connections

The following connections should be made:



- Brown-White: +5V
- Blue-White: Data
- Blue: Ground
- Brown: Ground, this core is not connected to the temperature sensor, grounding this core satisfies the EMC conditions of CE marking.

Note that multiple temperature sensors are connected in parallel.

13 Power Requirements

This section deals with the power budget required to run a WebBrick.

13.1 Power Supply

WebBricks should be run from a power supply that delivers 12.6V to 18V. 12.6V is a minimum to achieve a full 0-10V range on the analogue outputs.

A quiescent WebBrick will consume 55-60mA at 12.6V, however the WebBrick in real use will be supplying power to a range of external items including:

- **LEDs** The WebBrick may drive up to 8 mimic LEDs, each consuming 5mA. *40mA*
- **Relays** There are two relays, each using 30mA to hold the contacts closed. *60mA*
- **Temperature Sensors** There may be up to 5 sensors, these are driven periodically. *5mA peak*
- **Analogue Outputs** There are four buffered 0-10V outputs each capable of supplying 20mA. *80mA*
- **General Output Drive** The digital outputs can each supply 5mA, although only four are presented as TTL. if a rotary encoder is connected it will use 2mA. Driving the triac and open collector gates requires 2mA each. *40mA*

Total power budget, all outputs driven, all sensors connected 280mA.

13.1.1 Back Feed

Because the WebBrick consumes so little current, it is possible to inadvertently power it through the coils of external relays connected to the open collector outputs. This only becomes an issue if one needs to fully power down a WebBrick. To avoid this situation we suggest that the +ve side of relay coils are driven from the same power supply as drives the WebBricks.

14 Revision

This section highlights changes to WebBricks and WebBrick firmware.

14.1 6.3

1. Added Real Time clock support.
2. All 12 digital inputs can support rotary encoders. Connecting rotary encoder to dedicated terminals no longer supported.

14.2 6.4

1. Added hardware and command to switch mimic output from 4V to 10V, this enables using some more generic switches that are designed for higher voltages.
2. Added hardware to suppoprt RS232 and RS485 serial interfaces, only one of which can be active at any time.
3. Added minimal support in firmware to configure the serial interface and access it from the Gateway over http.
4. Added support in firmware to use the serial port for DMX control.
5. Add Dwell On and Dwell off commands. This extends the dwell options. Dwell Always will always dwell the output, Dwell On only actions the dwell if the output is currently off, dwell off starts a dwell timer to switch an output off if it is currently on and dwell cancel switches an output off if currently on or switches it on for dwell time if off.

14.3 7.0

1. Changed connections for Relay 4 so that all connections go to 'Relay 4B'.
2. Introduced dual power supplies so that noise or shorts on the temperature sensor bus do not compromise the operation of the WebBrick.
3. Changed the watchdog scheme so that the web server is power cycled on network disruption.
4. Upgraded the open collector outputs to handle 3 Amps.

15 CE Marking

WebBrick WB10B60 is in conformity with the essential requirements of the directive 1999/5/EC. The product has been tested to the standard EN55022:1998 "EMC emmissions and immunity", using the limits described on EN55022 Class B CISPR22(B).

15.0.1 General Description

15.0.2 Risk - High Temperatures

The WebBrick has 4 triacs each rated at 2A with an overall rating of 6.3A. If 6.3A of 240VAC is drawn through the WebBrick the triacs will dissipate around 7W of heat. There is sufficient heat sinking in the WebBrick to cope with this. Internal temperatures can rise to 85 Deg C. It is important that the WebBrick is not housed in small totally enclosed areas. Air circulation is best when the WebBrick is mounted in the vertical plane.

15.0.3 Risk EMI

Since the WebBrick uses a number of microprocessors, it may be prone to electromagnetic emissions if it is not installed properly. The current standards are stringent and require that a typical installation radiates less that 2 nano watts of power in the RF range.

We recommend that CAT 5/5E/6/7 type cables are used and one core of each pair is connected to ground.

Further we suggest that you use a good quality clean 12V supply to the WebBrick to ensure that supply borne noise is radiated via the ground plane.

It is not a requirement to connect the mains ground to the electronic ground of a WebBrick. However, please note that if you use the triacs for mains loads then the protective mains ground should be connected. There is no internal connection between this protective ground and the electronic ground.

15.0.4 Risk Software Failures

The WebBrick contains about 40K assembler code, whilst this has been thoroughly tested, software errors cannot be ruled out.

WebBricks should not take the place of safety systems or override safety systems.

16 Gotchas, Futures

Here we layout all the unimplemented features and current known bugs for versions 6.0 through 6.6 Releases of the firmware

Web Site use http://community.webbricksystems.com/ for all WebBrick news.

Split Scene Banks On first power up, buttons that do a 'Next' from scene 8 onwards will actually do a next from scene 0 until scene 8 is selected. The simple workaround is to press any 'off' buttons just after WebBrick power up.

This has been fixed in WebBrick firmware 6.6 onwards.

Remote Actions Implemented at the send side, but not yet the receive side.

Analogue 0utputs Version 6.1 onwards From version 6.1 of the WebBrick we have included a diode to protect against reverse polarity at the supply rails. This effectively drops the internal 12v supply rail by 0.6V. This means that for WebBricks that are supplied with 12.0V or less, the top end of the analogue output drops to about 9.67V. The simple solution is to ensure that the WebBrick are supplied with a little more than 12V. 12.6-18.0V is fine. Note that 13.6-13.8V power supplies are particularly suitable for float charging sealed lead acid batteries.

Configuring Digital Inputs In version 6.1 it is possible to configure things like a high threshold for a digital channel. This is a meaningless configuration. It does no harm, but conversely does nothing at all.

udp Types From 6.6 onwards there is only one UDP type for Digital Inputs.