

Jon00 LED Message Sign



I've have been looking for a LED Message screen for a number of years now. My interest was rekindled when SteveA published his [Snevl Betabrite Sign Control](#) script for the Betabrite signs on the Homeseer message board. This display was predominantly available in the USA/Canada and not readily available in the UK.

After looking around, I found a relatively cheap single line LED message sign from [CPC](#) in the UK. This was 7 X 80 colour display but had absolutely no details on the protocols used. I took a bit of a gamble and purchased this display and tried SteveA's script. As you can guess, this failed as it uses a different protocol from the BetaBrite. I then found some code to control the JetLite II protocol but this again this failed.

In the end, I decided to reverse engineer the protocol using the supplied software and used Homeseer to sniff the RS232 port and also display on the message sign.

The supplied script is therefore the fruits of my labour and made available to assist anyone else who wants to use these displays with Homeseer. I have researched the Internet and would appear you can get these displays in both Europe and USA/Canada:

The model number within the sign is M500N-7X80RG and made in China. It has the following specifications:

- Available in red or tri-colour (Red, Orange & Green)
- 7 x 80 dots resolution
- 2" Character height & 13 characters per line
- Full programmable via remote control or PC
- Program your own message using characters, numbers & special symbols
- Construct your message from the built in words & phrase library
- 8 Display speed & 24 display/transition methods
- 8 Built-in animations & 7 font styles
- 16 Font colours & 100 (99) File storage areas
- Displays current time & date
- 180° Adjustable viewing angle
- Desk/wall mountable
- Power: 5V DC (PSU supplied)
- Dimensions (WxHxD): 660 x 98 x 35 mm [26" x 3.8" x 1.4"]
- Weight: 1.34 kg

I have found the specification of this sign with numerous model numbers including the [NS-500RG2](#) which seems to be the model number used in North America.

I'm sure that if you find a LED sign with this specification, then the protocol generated by this script will work, however purchasing a sign is at your own risk!

Please note that the sign is connected via RS232 and the script will only work with connection type. It will also only support one sign. I do not propose to make this multi-sign addressable at this time. You should be able to parallel up signs on the same COM port and they will all show the same information.

The script provides access to most of the signs capabilities including changing colours, fonts, transitions, symbols, cartoons, graphics, time, date etc.

It also allows direct access to a large number of Homeseer data methods and therefore you can operate the display just by using events. This alleviates the need for any scripting knowledge.

Installation

Place the following scripts in to your Homeseer 2/Scripts directory:

Jon00MessageSign.vben
Jon00MessageSignControl.vben
Jon00MessageSignInput.vben

Place Jon00MessageSign.ini in to your Homeseer 2/Config directory

Please note that the directory 'Homeseer 2' may be different on your system depending on which version of Homeseer you are running. Examples include Homeseer HS2, Homeseer HSPro etc.

Connect your 7 x 80 LED Message sign into a spare serial port on your Homeseer server PC.

You now need to configure the port number in the [Jon00MessageSign.ini](#) (see later).

General

It would appear that the sign has no method to send data to the display's memory and then send a command to show this data. In other words, whatever you send to the display will be stored but it will also be displayed at the same time.

The message sign has 99 memories; however for the majority of the functionality I have provided, this only uses one of these. This memory can display around 8190 characters which is sufficient for most uses.

Understanding the script calls

The LED sign is controlled via a script call (Jon00MessageSignInput.vben) using optional parameters to define the output.

This can be controlled via an event directly using the run script entry or via another script using the call hs.runex.

The optional parameters are defined as:

("Main", "<argument 1>|<argument 2>|<argument 3>|<argument 4>|<argument 5>")

You will notice that the second parameter is split into 5 arguments separated with a pipe (vertical bar) command.

<argument 1> (Optional – providing argument 2 and/or argument 3 are used)

This defines preset message configurations, which are defined under the [Text] section in the Jon00MessageSign.ini. The naming of presets is entirely up to the user but sample numerical presets have been included in the ini file.

<argument 2> (Optional – providing argument 1 and/or argument 3 are used)

This is used as a direct input for data when running the script via the hs.runex command via another script.

<argument 3> (Optional – providing argument 1 and/or argument 2 are used)

This is identical to <argument 1> and has been included in case you wish to use <argument 2> but adding additional information to it from preset configurations.

<argument 4> (Optional)

This is used to set the priority of the message using 3 methods:

xx (where xx is a numerical value in seconds)

If you enter a numerical value in this field, the message will be placed on the priority queue list. If this is the only message in the queue, it will be shown on the display immediately for the number of seconds defined as the numerical value. If subsequent messages are sent in this manner, they will be shown in order that they arrive.

Pxx (where xx is a numerical value in seconds)

This is identical to the previous method; however, this will push the message to the top of the priority queue. The current priority message will continue to display but will then switch to this message once complete. Other priority messages will then display in order.

Ixx (where xx is a numerical value in seconds)

This is immediate mode and will push the message to the top of the priority queue. In addition this will discard the current priority message showing and display it on the sign immediately. Other priority messages left in the queue will then display in order.

Please note that once priority messages have timed out, the original idle message will continue to be displayed.

<argument 5> (Optional)

This defines the file number where the message is to be stored. This will overwrite the default FileNumber= setting in the jon00LEDMessage.ini. See [working with files](#) later.

This is an optional value for more advanced control of messages.

Formatting the look of messages

The LED message sign has lots of built in functionality to display messages in different colours, fonts, speed etc and includes a number of built in cartoons, symbols and transitions. In addition, there is a built in clock, which can be used to display the time/date in either 12-hour or 24-hour mode.

To allow control of these functions, the script makes use of formatting tags. A list of all formatting tags for the display can be found at [Appendix I](#).

These tags can be defined in the preset configuration area under the [Text] section in the Jon00LEDMessage.ini (used in argument 1 and/or argument 3) or directly in argument 2.

Displaying data from Homeseer

The script includes a number of built-in formatting tags to provide data from Homeseer, This includes things such as text from device strings, device values, status etc. A full list of supported tags is shown at [Appendix 2](#).

These tags can be defined in the preset configuration (scratch pad) area under the [Text] section in the jon00LEDMessage.ini (used in arguments 2 and/or argument 4) or directly in argument 3.

Getting Started – Hello World Examples

In the jon00LEDMessage.ini under [Text] you will find the following entries:

```
1=Hello World
2=Hello
3=Goodbye
4=Cruel
5=World
6=Hello{n}World
7={orange}Hello{n}{green}World
8={trans24}{orange}Hello{n}{trans19}{green}World
9={trans24}{orange}Hello{pause5}{n}{trans19}{green}World{pause5}
```

Let's get the sign to show Hello World.

Under the Events action tab, add the Jon00MessageSignInput.vben script (Select Run script from the drop down). Press the green 'edit' button and then press the pink switch to advanced view' button. You will see a field marked 'Optional Parameters (function,param)'

Add the following line to the optional parameters field:

```
("Main","1")
```

When you run this event you should see "Hello World" on the display with random transitions.

This simply uses argument 1 to display this message. The "1" in this argument is simply referenced to the data contained in the ini file under the [text] (scratchpad) section. As you can see this has "hello world" assigned to it as a text string.

Let's now get the screen to show "Goodbye Cruel World", this time showing each word separately.

The optional parameters would be:

```
("Main","2+3+4")
```

When you run this event you should see "Goodbye" then "Cruel" then "World" alternating on the display with random transitions. This demonstrates how you can obtain different data from the scratchpad area and then combine them on your display.

Let's now show Hello World as the word "Hello" and then "World" but this time as a single definition from the scratchpad area.

You can see that I have defined 6 as "Hello{n}World". The {n} is a sign tag meaning new line.

You now just need optional parameters of ("Main","6")

When running this you should see the word 'Hello' and then 'World'.

Let's now change the colour of each word. From [Appendix 1](#), orange uses the tag {orange} and green as {green}

You now just need optional parameters of ("Main","7") as 7 have been defined with additional colour information.

When running this you should see the word 'Hello' in orange and then 'World' in green. Without any transition tags, the screen will apply random effects between the words.

Let's now add transition effects between the words. I have chosen transitions of Pac Man and slide-in for this example.

You now just need optional parameters of ("Main","8") as 8 have been defined with additional transition effects.

When running this you should see the word 'Hello' in orange and then 'World' in green. This will show the different transitions defined.

Finally, let's now also display each word on the display for 10 seconds. Without any guidance, the display will show the message for around 2 seconds. Looking at Appendix 1, a 10 second pause uses the {pause5} tag.

You now just need optional parameters of ("Main","9") as 9 have been defined with additional pause information.

Using Homeseer Tags

Providing information from Homeseer is just as easy as using the sign formatting tags. These are defined at [Appendix 2](#)

Let's define a simple example of showing the version of Homeseer that you are running.

Under the [Text] heading in the ini file, define a line with something like:

Version=The version of Homeseer is {version}

You now just need optional parameters of ("Main","Version") to display this.

In this example you will notice that I did not use a number as a reference and shows you can choose any label you wish to store your definitions.

Another example may be to display the outside temp. This may be kept in a device string, which for example could be located in virtual device w12.

To display this on the message screen, define a line with something like:

OT=The temperature outside is currently {devicestring w12} °F

Now run the script with optional parameters of ("Main","OT") to display this message.

Providing external information to the display

The information provided by Homeseer tags may not be suitable in some instances so you can use argument 2 to provide this input. To provide this functionality, you must call the script from another script using the hs.runex command.

Lets create a vbscript to show "Goodbye Cruel World" via this direct call

```
Sub Main()  
Text="Goodbye Cruel World"  
hs.runex "jon00LEDMessageInput.vben","Main","|" & Text & "||"  
end sub
```

This is the same in VB.NET

```
Sub Main(ByVal Parm As Object)  
Dim Text As String  
Text="Goodbye Cruel World"  
hs.runex ("jon00LEDMessageInput.vben","Main","|" & Text & "||")  
end sub
```

Running either script will display the message. You can add sign tags and/or homeseer tags to the string if you wish. Obviously the text string would ultimately contain data from your script that you want to display and not just text!

To demonstrate arguments 1, 2 and 3, let's now show "Goodbye Cruel World" using all three via a vbscript call (the script is saved with a .txt extension).

```
Sub Main()  
Text=" Cruel "  
hs.runex "jon00LEDMessageInput.vben","Main","2|" & Text & "|4|"  
end sub
```

From this example you will see that argument 1 is set to '2' (which is defined as "Goodbye" in the ini file), the text is set to ' Cruel ' in argument 2 and argument 3 set to '4' (which is defined as "World" in the ini file).

Using argument 1 and/or 3 in this manner allows you to provide pre-defined formatting or text if you wish.

Working with Files

The sign has storage for 100 files each with a capacity to hold around 8190 characters. File00 is reserved for the demo, which leaves File01 to File99 for user input. For normal usage, the script just updates a single file (as defined by the FileNumber= entry in the ini file).

You will notice that I have provided a 5th argument in the script call, which allows you to assign a file number to store and show the data. This overrides the FileNumber= entry setting.

You can use this to set and store different messages if you wish.

Let's set File20 to "Hello" and File21 to "World"

From previous examples, we can use the data in the [Text] scratch pad area to provide this info.

First run the script with the following Optional Parameters:

```
("Main","2|||20")
```

This will set File 20 to "Hello"

Then run it again with the following Optional Parameters:

```
("Main","5|||21")
```

This will set File 21 to "World"

Finally, run the script for a third time with the following Optional Parameters

```
("Main","3")
```

This will set the default File number to show "Goodbye" and should be displayed on the display.

To switch the display to use any file or files, you use the sequence tag. This takes the form of {SeqXXYYZZ} where XX is the first file and YY is the second etc. You can keep adding file names to this command as necessary. Each file name must be 2 digits.

To show the contents of File20, you would use:

```
("Main","|{seq20}")
```

To show a recurring sequence of File 20 and 21 you would use:

```
("Main","|{seq2021}")
```

Running this example would show "Hello World" but this time using the two files. Obviously, you can define the sequence tags in the scratch pad area if you wish and call the reference instead.

This method is really is no good for dynamic data as this command is cancelled each time you send new data to a file.

Special Characters

The sign has a number of special characters, which are mixture of regional character sets. This means that they are not referenced to the normal ASCII codes for a particular region. As the majority of people use the ANSI character set, I have re-referenced the special characters back to their correct ASCII code positions if available.

Certain characters have not been referenced and so you will need to access these using the {&HXX} tags to pick out individual characters.

Jon00LEDMessage.ini

[Settings]

Comport=1 (Values 1 upwards)

This setting defines the COM port that you have used to connect the sign on your Homeseer server/pc.

PortSetupType=0 (Values 0 or 1)

This defines how the script opens the port for use with the sign. If this is set to 0 then the port will be opened when the sign needs updating and then closed again.

If you are having COM port lockups, you can set this to 1. This will not open or close the COM port when the screen needs updating, however you will need to permanently open the defined COM port when Homeseer starts. To achieve this, add the following to your Startup.txt script:

```
hs.runex "Jon00LEDMessage.vben","Startup",""
```

As Homeseer automatically closes all ports on shutdown, there is no need to add an entry to the shutdown.txt

FileNumber=1 (1-99)

The LED sign has 99 Files where messages are stored. This defines the Filenumber that is assigned for normal updating of messages. For normal usage, keep on file 1.

[Text]

This is used as a scratchpad area to define pre-programmed messages.

```
1=Hello World
2=Hello
3=Goodbye
4=Cruel
5=World
6=Hello{n}World
7={orange}Hello{n}{green}World
8={trans24}{orange}Hello{n}{trans19}{green}World
9={trans24}{orange}Hello{pause5}{n}{trans19}{green}World{pause5}
```

[Data]

This section is used by the script for storing information.

Version=1.00

This defines the version of the script and is not user configurable.

Status=1

This defines the on/off status of the sign when using the {signon} and {signoff} tags. If you cannot get a response from the sign, check that this value is 1 (on).

IdleData=

QueueData=

ImmediateData=

This is used by the script to reference the various messages by priority. This is not user configurable.

History1=

History2=

History3=

These just stores the last 3 configurations sent to the display and are not configurable.

LED Message Sign Annoyances

Unlike the Betabrite displays, there is no means to store variables directly to the display memory, which allows updating data without the display flashing. To provide updated information, you need to refresh the entire data string every minute or so which will cause the screen to flash. The exception to this is displaying the internal time and/or date.

In addition, sending any information to the display via the serial port will cause the internal sounder to 'chirp'. This is obviously annoying and so I have disabled mine by removing the sounder.

To do this, unscrew the end of the display with the power and COM port sockets and pull out the PCB until a daughter board is located. Undo the two screws holding the daughter board to the main board and then pull vertically upwards. This will separate the two connect by DIL headers. Unsolder the circular beeper for complete silence or place something over top to reduce the volume. Place the daughterboard back on the main board and re-screw. Slide the PCB back with the main casing and re-screw the end cap in place

You may have noticed 8 DIP switches. I assume this is to set the address of each sign but have no other information on this.

Revision

V1.00 – January 1st 2009

Original Version

Author

Jon Simmons (Jon00)

Jon00@btinternet.com

Newer versions may be found at <http://www.jon00.me.uk/hs.html>

Support


























Via my forums on my Homeseer site or via email.






APPENDIX 1

SIGN TRANSITION TAGS

Command	Description	Examples & Info
{trans1}	Cyclic	
{trans2}	Immediate (static)	{trans2>Hello{trans19}World
{trans3}	Open from right	
{trans4}	Open from left	
{trans5}	Open from centre	
{trans6}	Open to centre	
{trans7}	Cover from centre	
{trans8}	Cover from right	
{trans9}	Cover to left	
{trans10}	Cover to centre	
{trans11}	Scroll up	
{trans12}	Scroll down	
{trans13}	Interlace to centre	
{trans14}	Interlace cover	
{trans15}	Cover up	
{trans16}	Cover down	
{trans17}	Scan line	
{trans18}	Explode	
{trans19}	Pac Man	
{trans20}	Fall & stack	
{trans21}	Shoot	
{trans22}	Flash	
{trans23}	Random	
{trans24}	Slide In	
{trans25}	Automatic	
{trans26}	Random	

SIGN SYMBOL TAGS

Command	Description	Examples & Info	
{sym1}	Sunny		
{sym2}	Cloudy		
{sym3}	Rainy		
{sym4}	Clock		
{sym5}	Phone	{sym5} 123 4567 890	
{sym6}	Spectacles / Glasses		
{sym7}	Faucet / Tap		
{sym8}	Rocket		
{sym9}	Alien / Bug		
{sym10}	Key		
{sym11}	Shirt / Sweater		
{sym12}	Helicopter		
{sym13}	Car		
{sym14}	Tank		
{sym15}	House		
{sym16}	Teapot		
{sym17}	Trees		
{sym18}	Duck		
{sym19}	Motorcycle		
{sym20}	Bicycle		
{sym21}	Crown		
{sym22}	Hearts		
{sym23}	Arrow right		
{sym24}	Arrow left		
{sym25}	Arrow down Left		

Command	Description	Examples & Info	
{sym26}	Arrow up left		
{sym27}	Mug		
{sym28}	Chair		
{sym29}	Shoe		
{sym30}	Martini Glass		

SIGN FONT TAGS

Command	Description	Examples & Info
{font1}	5 X 6 Short	
{font2}	5 X 11 Short & Wide	
{font3}	7 X 6 (Default)	{font3}Normal text {font4}Wide text
{font4}	7 X 11 Wide	
{font5}	7 X 9	
{font6}	7 X 17 Extra Wide	
{font7}	Small font	{font7}This text is small

SIGN COLOR TAGS









Command	Description	Examples & Info
{color1} or {dimred}	Dim Red	
{color2} or {dimgreen}	Dim Green	
{color3} or {dimorange}	Dim Orange	
{color4} or {dimyellow}	Dim Yellow	
{color5} or {dimlmix}	Dim Layer Mix	
{color6} or {red}	Red	{red}This text is red!
{color7} or {green}	Green	
{color8} or {orange}	Orange	{orange}Orange text {green} Green Text

Command	Description	Examples & Info
{color9} or {yellow}	Yellow	
{color10} or {lmix}	Layer Mix (Rainbow)	
{color11} or {verticalmix}	Vertical Mix	
{color12} or {sawtoothmix}	Saw tooth mix	
{color13} or {greenonred}	Green on red	
{color14} or {redongreen}	Red on Green	
{color15} or {orangeonred}	Orange on red	
{color16} or {yellowongreen}	Yellow on green	
{rc1}	Random solid colours & layer mix dimmed	Range: {color1} {color2} {color3} {color4} {color5}
{rc2}	Random solid & mix colours	Range: {color6} {color7} {color8} {color9} {color10} {color11} {color12}
{rc3}	Random colours with background	Range: {color13} {color14} {color15} {color16}
{rc4}	Random solid colours	Range: {color6} {color7} {color8} {color9}
{rc5}	Random mix colours	Range: {color10} {color11} {color12}
{rc6}	All colours	Range:{color1} to {color16}

SIGN CARTOON TAGS

Command	Description	Examples & Info
{cartoon1}	MERRY X'MAS	
{ cartoon2}	HAPPY NEW YEAR	
{ cartoon3}	4 TH JULY	
{ cartoon4}	HAPPY EASTER	
{ cartoon5}	HAPPY HALLOWEEN	
{ cartoon6}	DON'T DRINK & DRIVE	

SIGN GRAPHICS TABS

Command	Description	Examples & Info
{graphic1}	User Defined 1	Defaults to Cityscape if not programmed
{graphic2}	User Defined 2	Defaults to Cars if not programmed
{graphic3}	User Defined 3	Defaults to Teapot & cups if not programmed
{graphic4}	User Defined 4	Defaults to Telephone if not programmed
{graphic5}	User Defined 5	Defaults to Nature if not programmed
{graphic6}	User Defined 6	Defaults to Ship if not programmed
{graphic7}	User Defined 7	Defaults to Swimming if not programmed
{graphic8}	User Defined 8	Defaults to Cat if not programmed
{graphic9}	Cityscape	
{graphic10}	Cars	
{graphic11}	Teapot & cups	
{graphic12}	Telephone	
{graphic13}	Nature	
{graphic14}	Ship	
{graphic15}	Swimming	
{graphic16}	Cat	

SIGN PAUSE TAGS

Command	Description	Examples & Info
{pause1}	pauses for 2 seconds	
{pause2}	pauses for 3 seconds	
{pause3}	pauses for 4 seconds	
{pause4}	pauses for 6 seconds	Hello{pause4}World{pause4}
{pause5}	pauses for 10 seconds	
{pause6}	pauses for 20 seconds	
{pause7}	pauses for 30 seconds	
{pause8}	pauses for 60 seconds	

SIGN CONTROL TAGS

Command	Description	Examples & Info
{signoff} *	Puts the sign to sleep	The sign can only be woken by sending a {signon} tag.
{signon} *	Wakes up the sign	Wakes the sign after being put to sleep by the {signoff} tag. Data previously shown is restored.
{signonclr} *	Wakes up the sign	Wakes the sign after being put to sleep by the {signoff} tag and clears the display.
{settime12}	Sets the time/date	Sets the time and date of the sign from the PC clock to show in 12 hour format i.e. 04:43 PM
{settime24}	Sets the time/date	Sets the time and date of the sign from the PC clock to show in 24 hour format i.e. 16:43
{houralarmon}	Enable on the hour alarm beep	
{houralarmoff}	Disable on the hour alarm beep	
{signtimeonHHMM}	Turns on the sign at specified time	Times must be in 24 hour format Only use if you do not update the sign
{signtimeoffHHMM}	Turns off the sign at specified time	Times must be in 24 hour format Only use if you do not update the sign
{test} *	LED display test	All LED's will illuminate. 2 seconds red, 2 seconds green and 2 seconds orange
{Seq[file names]} *	Display a file or sequence of files	File numbers under 10 must be preceded with a 0 {seq01021524} will sequence files 1,2, 15 & 24

*** NOTE: These tags can only be used singularly in their own call to the script.**

SIGN MISCELLANIOUS TAGS

Command	Description	Examples & Info
{n}	New Line	Provides a break between messages
{clr}	Clear	Clears all data from the screen
{time}	Shows the time	Time maintained by the sign
{date}	Shows the date	Date maintained by the sign
{temp}	Shows the temperature	May not be supported i.e. hardware not installed
{&HXX}	Displays a single character	Allows you to display any character in the sign's character set using a hex code 00-FF

SIGN BEEP TAGS

Command	Description	Examples & Info
{beep1}	3 short beeps	
{beep2}	7 very fast short beeps	
{beep3}	1 short beep	

APPENDIX 2

HOMESEER TAGS

Command	Description	Examples & Info
{devicestring XX}	Shows the device string text for a virtual device	{devicestring w22}
{devicetext XX}	Shows the text from any device (alternative to devicestring)	{devicetext A5}
{devicevalue XX}	Shows the device value for a virtual device	{devicevalue G64}
{devicestatus XX}	Shows the device status (as a number) for a virtual device	{devicestatus C6}
{devicelastchange XX}	Shows the date & time of the virtual device last changed status	{devicelastchange D5}
{devicetime XX}	This will show the time in minutes since the device status last changed.	{devicetime t29}
{hstime}	Shows the current time via Homeseer	
{hsdate}	Shows the current date via Homeseer	
{day}	Shows the day as a number	
{month}	Shows the month as a number	
{year}	Shows the year as a number	
{hour}	Shows the current hour in 24-hour format	
{hour12}	Shows the current hour in 12-hour format	
{hour12a}	Shows the current hour in 12-hour format with leading 0 i.e. 06:32 or 10:44	
{minute}	Shows the current minute	

{second}	Shows the current second	
{ampm}	Shows either 'AM' or 'PM' depending on the time of day	
{monthname}	Shows the month as a name i.e. June.	
{weekdayname}	Shows the day of the week as a name i.e. Monday.	
{uptime}	This shows Homeseer's uptime	
{version}	Shows the current version of Homeseer.	
{webviews}	Shows the current web views as a number.	
{lastip}	Shows the last IP that viewed a page.	
{webuser}	Show the current/last web user	
{lastx10}	Shows the last X10 command sent.	
{lanip}	Shows the IP address of your Homeseer server.	
{wanip}	Shows the external (internet) IP address.	
{threads}	Shows the number of Homeseer threads.	
{hsmemory}	Shows the amount of memory that Homeseer is using.	
{scriptsruntime}	Shows a list of scripts currently running.	
{osversion}	Shows the Operating System version number.	
{licensed}	Shows if the copy of Homeseer is currently licensed.	
{plugincount}	Shows the number of plugin's installed on your system.	

{pluginlist}	Shows a list of plugin's installed on your system.	
{sunrise}	Shows the sunrise time for the day.	
{sunset}	Shows the sunset time for the day.	
{devicecount}	Shows the number of devices on your system.	
{eventcount}	Shows the number of events on your system.	
{linestatus}	Shows the line status if you have Homeseer phone installed.	
{replace {string to find},{string to replace}}	This can replace existing strings in a Homeseer tag with another. It cannot replace text in sign tags	<p>{devicestring w1} could display the following text: Temperature: 65F</p> <p>{devicestring w1}{replace Temperature,temp}</p> <p>This would display the following text: temp: 65F</p>
{getini {section},{key},{ini name}}	This will extract data from any HS ini file so that it can be displayed.	<p>HS web server port:</p> <p>{getini settings,svrport,settings.ini}</p>
{hslog {line number from end}}	This will extract and show a line from the Homeseer log starting from the end of the log.	<p>{hslog 0} - This will show the last line in the Homeseer log.</p> <p>{hslog 1} - This will show the second to last line in the Homeseer log.</p>
{bivarXXXX}	This will show the contents of a homeseer variable bivarXXXX where XXXX is any string of your choice.	<p>This Homeseer variable will need to be created by another script i.e.</p> <p>hs.createvar ("bivartest1")</p> <p>hs.savevar ("bivartest1","Hello World")</p>
{lastx10}	Shows the last X10 command sent.	
{lanip}	Shows the IP address of your Homeseer server.	
{wanip}	Shows the external (internet) IP address.	
{threads}	Shows the number of Homeseer threads in use.	