



Clipsal C-Bus Control of Systemline 7

Rev 1.0
23rd May 2018

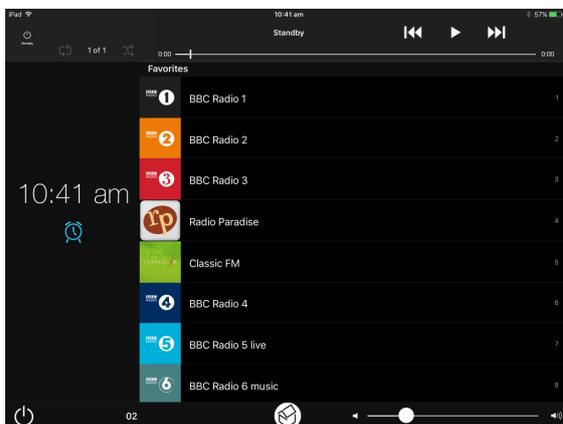
1.0 Introduction

The Schneider Electric C-Bus Network Automation Controller allows a C-bus system to control S7 using a network connection. It is possible to configure the Controller to map button presses from C-Bus keypads to output S7 TCP/IP commands. The Network Automation Controller must be configured correctly using Scripting to make a connection to an S7 Netamp and send the correct command.

S7 has been designed so that other control systems such as C-Bus can connect to S7 and send commands. All the control protocol to achieve this is published on the Systemline7 web page, but these are included in the example scripts that can be copied into the Controller

http://ecom.armourhome.co.uk/files//systemline/Systemline_7/Netamp/S7-NetAmp-IP-Control-Specification-V4.2.pdf

It is important to configure the appropriate type of control at each device to give a simple and reliable end user experience. C-Bus control panels usually have from 6 to 12 buttons that can be labelled with its function, such as 'Scene 1' or 'Off' for lighting control. The ideal audio function on these hard button keypads is to simply select a radio station or playlist and adjust the volume. Radio stations can be stored as Favourites on an S7 system using the S7 App, and these are listed numerically on the App as shown below.



In order for the C-Bus system to select a favourite in a zone, it needs to communicate with the Netamp controlling the zone. Netamps with software 0004-0011 or 0003-0008 or higher have commands to select stored favourites. If the Netamp has a lower software version, you should perform a software update from the internet.

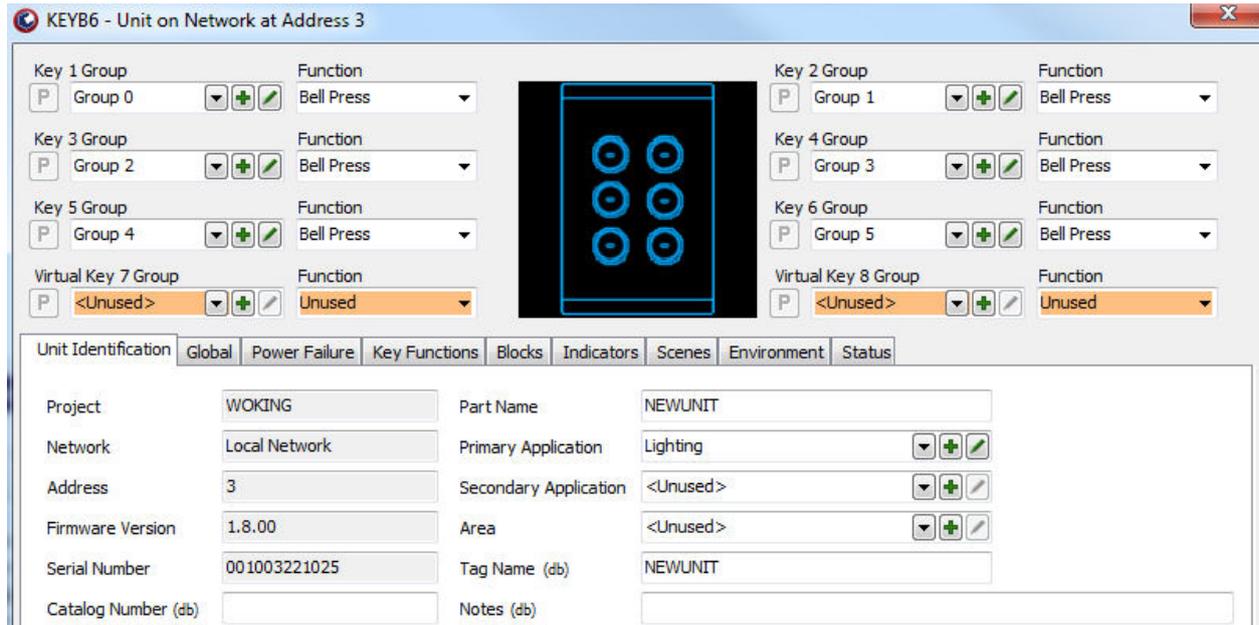
Favourites can also be playlists of music as stored by the end user, in which case you may also consider adding in play, stop, pause, and skip commands as button functions.

Once the C-Bus Network Automation Controller has been configured to map S7 commands to button keypad presses, the end user will be able to use C-Bus keypads to select favourites and adjust the volume without the need to open up an App.

A Controller project titled "Systemline 7-2018.05.21-14.30" is available to download from the Systemline7 website downloads section. This already has an S7 Netamp with the commands in 'Scripting' mapped to C-Bus group addresses on a room as an example

2.0 Adding the Netamp Commands to the Controller

The Network Automation Controller LSS5500NAC needs to be plugged into the Local Area Network and also the C-Bus system. You must know the application and group addresses of the keypad buttons you wish to use to control S7, for example below is a keypad in Toolbox:



It has a primary application of lighting (56) so has 6 buttons with addresses:

0/56/0 (Key 1) 0/56/1 (Key 2)
0/56/2 (Key 3) 0/56/3 (Key 4)
0/56/4 (Key 5) 0/56/5 (Key 5)

2.1 Open up the Controller webpage, select 'Configurator' and import the Toolkit CGL file, so that it has a record of all applications and group addresses which can then be easily selectable.

2.2 Select the 'Scripting' tab



2.2 Press 'Add a new script' on the bottom left corner



You can name the script, and select the group address from the drop down option (if you imported the CGL file in step 1).

2.3 Once the new script is added you can edit it pressing the edit icon



2.3.1 Script for selecting a favourite

You will now need to know the fixed IP address of the Netamp associated with the zone you wish to control, also if the zone is connected to zone 1 or zone 2. Below is the script needed to control Netamp with IP address 192.168.1.130, selecting favourite 1 on zone 1 for a button on address 0/56/0. The favourite should be selected when you press the button each time, it is a single action type.

```

Event-based: Fav1
1
2 value = GetCBusState(0, 56, 0)
3 local socket = require("socket")
4 local host, port = "192.168.1.130", 9760
5 local tcp = assert(socket.tcp())
6
7 tcp:connect(host, port);
8
9 if (value) then
10     tcp:send("$s1fav1\r\n");
11     value = GetCBusState(0, 56, 0)
12 end
13
14 local s, status, partial = tcp:receive()
15 tcp:close()

```

Line 2 requests the state of group address 0/56/0, which is either 1 or 0. If you have the bell press button pressed down then this state should equal 1.

Lines 3,4,5,7 makes the TCP/IP connection to the Netamp

Line 9 determines if the action shall take place dependent on the state of the group address (so only send the command if the button is pressed, do not send the command when the button is released).

Line 10 sends the command to the netamp

Line 11 requests the state of the button again

Lines 14,15 closes the TCP/IP connection

To select a different keypad button group address simply change the address in lines 2 and 11:

Group address 0/30/1
value = `GetCBusState(0, 30, 1)`

To select a different IP address simply change the host number in line 4:

IP address 192.168.0.210:
local host, port = `"192.168.0.210", 9760`

To select a different favourite simply change the tcp:send command in line 10:

Favourite 3:
tcp:send(`"$s1fav3\r\n"`);

Favourite 9
tcp:send(`"$s1fav9\r\n"`);

Favourite 13
tcp:send(`"$s1fav13\r\n"`);

2.3.2 Script for adjusting the volume

Volume can be adjusted when you press and then hold down on a button, the volume continuing to ramp up or down whilst the button is pressed. The volume adjustment ceases as soon as the button is released. The script includes a loop which repeats the volume command again and again whilst the button is being held down.

```
Event-based: Vol+
1
2 local socket = require("socket")
3 local host, port = "192.168.1.130", 9760
4 local tcp = assert(socket.tcp())
5
6 tcp:connect(host, port);
7
8
9
10 value = GetCBusState(0, 56, 5)
11 -- loop while condition is met
12 while (value) do
13     tcp:send("$s1vol+\r\n");
14     os.sleep(0.2);
15     value = GetCBusState(0, 56, 5);
16
17 end
18
19 local s, status, partial = tcp:receive()
20 tcp:close()
```

Line 10 obtains the state of the button being pressed, so the actions are only sent whilst the button is pressed.

Line 12 repeats the commands below whilst the button is pressed

Line 13 sends the volume + command

Line 14 adds a 0.2s delay

Line 15 checks the status of the button again

To decrease the volume simply change the command on line 13

```
tcp:send("$s1vol-\r\n");
```

2.3.3 Other commands in place of favourites

You can substitute other commands in place of favourite that work well on a single press:

Play: `$s1trn1`

Stop: `$s1trn2`

Pause: `$s1trn3`

Skip >I: `$s1trn4`

Skip I<: `$s1trn5`

2.4 Save each script and sync project data. You can now test your button presses and make sure the correct operation is carried out.

END