



Using the Vivisys EasyConnect board to connect a Garage door opener

Introduction

This note describes the retrofitting of the C-Bus control system to an existing garage door opener. The operation is to be as follows:

- Pressing the remote transmitter button in the car operates the garage door, and turns on the garage C-Bus controlled fluorescent lights with a 5 minute auto-off timer.
- Pressing the DLT button inside the garage door does the same thing.
- Pressing another button on the DLT cancels the 5 minute auto-off timer (so the lights stay on).
- Finally, an independent sensing of whether the garage door is open or closed is to be implemented.

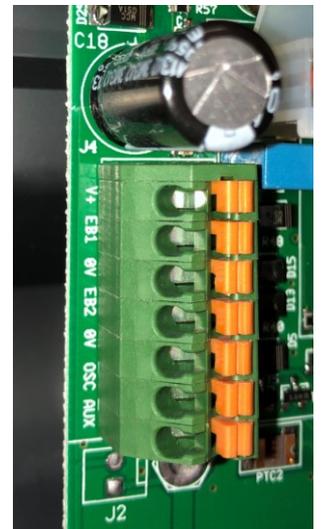
This will be achieved using the Vivisys EasyConnect board.

Hardware

The garage door opener needs to have the following features:

- A button on the remote transmitter closes a contact in the main opener unit.
- An input in main opener unit accepts a contact closure to operate the garage door.

The Automatic Technology GDO-9v2 overhead garage door opener being retrofitted has these features.



The internal interfacing connector had the following terminals needed for this exercise:

V+ = Positive supply – nominally 30 V but measured at 34 V.

0V = Supply return (zero volts).

OSC = Used for the connection of a wired switch (momentary contact). Sits at 5 V, and can be switched or pulled to 0V to open, stop or close the door.

AUX = Open collector output. Pulls to 0V when remote transmitter AUX button pressed.

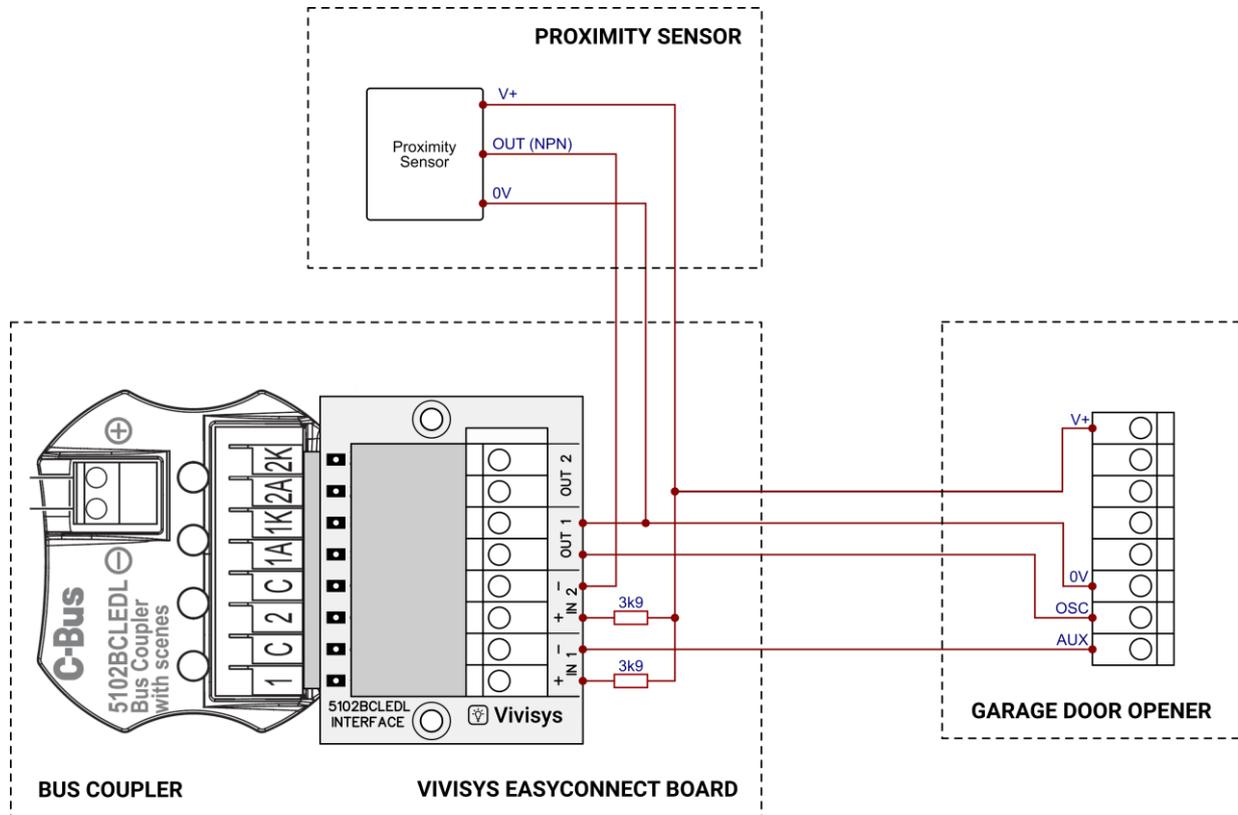
The remote transmitter unit was programmed so that pressing the button operated the AUX contact.

A regular 3-wire proximity sensor was used as described later.

A Clipsal 5102BCLEDL Bus coupler in conjunction with a Vivisys EasyConnect board was used to provide the C-Bus interfacing.

Schematic diagram

The schematic diagram for the connections was as follows:



The following design notes and calculations apply:

Input 1: The EasyConnect board inputs have a Minimum and Maximum input current for the ON state of 5 mA and 25 mA respectively. The AUX output pulls down to 0V, and so a 3k9 0.5 W resistor was added as a pullup to the 34 V supply, resulting in a current of about 7.0 mA into IN 1 when the remote transmitter button is being pressed (comfortably in the required range of 5 to 25 mA). The required 3k9 resistor was determined from the following table.

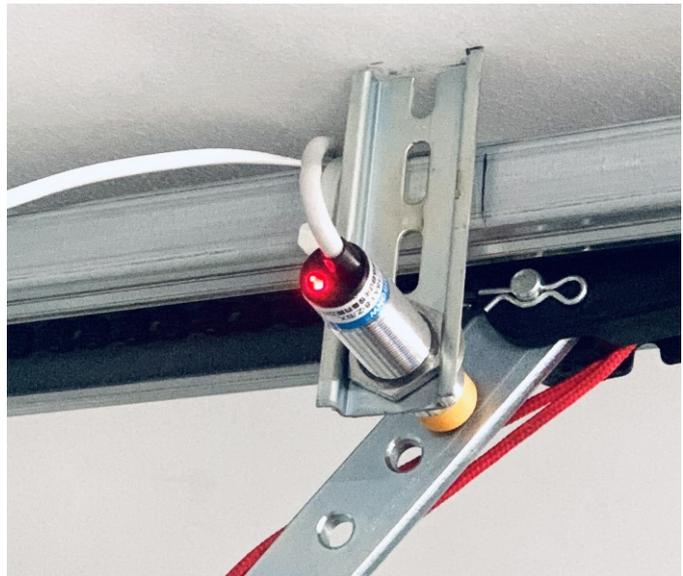
Series resistance required to limit current for voltages greater than 18 Vdc						
Voltage source (V+)	5 - 18 Vdc	19 - 24 Vdc	25 - 30 Vdc	31 - 36 Vdc	37 - 42 Vdc	43 - 48 Vdc
External Resistor(s)	Not needed	2.2 kΩ	2.7 kΩ	3.9 kΩ	2 x 2.2 kΩ in series	2 x 2.7 kΩ in series

Input 2: The proximity sensor used has a maximum supply voltage of 36 V and the output is an NPN transistor pulling down to 0V. This is the same situation as for Input 1, and so again a 3k9 0.5 W resistor was added into the circuit.

Output 1: A check was made of the current flow when shorting the OSC terminal to the 0V terminal (i.e. to activate the door operation). A multimeter set to the 10A current range was connected from OSC to 0V, and indicated approximately 1 mA, and operated the door. This was well less than the maximum 300 mA current rating for the EasyConnect board outputs, so direct connection of OUT 1 to the garage door opener was acceptable.

Installation

The proximity sensor could be installed in many possible locations but was installed as follows:

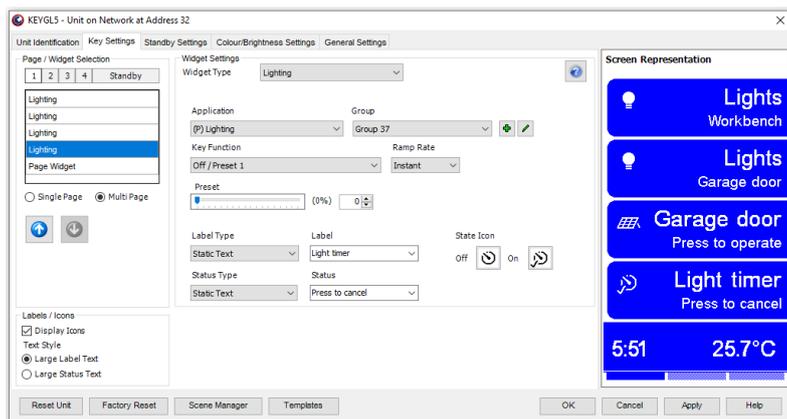


The remaining items were installed as per the following photos. The C-Bus cable came down through the ceiling into the plastic enclosure, with the proximity sensor wiring coming in through the back. The wiring to the garage door opener passed through the flexible conduit.



Software setup

The garage had two sets of fluorescent lights - Lights (Workbench), and Lights (Garage door), assigned to the first two eDLT buttons. The third button activates the garage door, with the fourth button cancelling the 5 minute delay if desired. The Toolkit setup for these four buttons is shown here.



- (1) Lights – Workbench
Group 30
Toggle/Toggle
- (2) Lights – Garage door
Group 29
Toggle/Toggle
- (3) Garage door
Group 36
Bell Press/Bell Press
- (4) Light timer
Group 37
Off/Preset 1
(Preset 1 = 0)



The C-Bus Bus Coupler 5102BCLEDL has the following Toolkit setup. The Bell Press settings ensure that the Group Addresses follow the state of the two inputs, while the LEDs are assigned separate Group Addresses allowing them to be independent of the inputs.

Block Assignments		Recall Lvl (%)		Timer		Keys Using Block								LED Assignment	
Application	Group	Recall 1	Recall 2	Timer 1	Expiry	1	2	V3	V4	V5	V6	V7	V8	1	2
1	Primary Input 1 (BCLED)	100	100	0h0m0s	Off Key	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Primary Input 2 (BCLED)	100	100	0h0m0s	Off Key	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Primary Output 1 (BCLED)	100	100	0h0m0s	Off Key	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Primary Output 2 (BCLED)	100	100	0h0m0s	Off Key	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Primary <Unused>	100	100	0h0m0s	Off Key	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Primary <Unused>	100	100	0h0m0s	Off Key	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	Primary <Unused>	100	100	0h0m0s	Off Key	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	Primary <Unused>	100	100	0h0m0s	Off Key	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The actual decimal and hexadecimal Group Addresses are shown here, but these will vary from project to project.

Application (56) - Lighting				
Group Name	Address	Hex Address	Level	
Input 1 (BCLED)	080	50	Off	
Input 2 (BCLED)	081	51	Off	
Output 1 (BCLED)	082	52	Off	
Output 2 (BCLED)	083	53	Off	

These two input groups are used in this project to trigger scripts in a 5500SHAC and details of this follow. Of course, for simpler requirements an Automation Controller is not required, and the inputs could directly control output groups or scenes to produce the desired outcomes.

5500NAC/SHAC Settings and Coding

The 5500NAC/SHAC is not needed to use the EasyConnect board. However, to round out this project description, the LUA scripts written to support the desired operation are included.

The four Event triggered scripts:

Script name ▲	Group address / keyword	Description
Garage AUX button	Local Network/Lighting/Input 1 (BCLED) (0/56/80)	Garage remote control AUX button press
Garage door status	Local Network/Lighting/Input 2 (BCLED) (0/56/81)	Update the garage door status string text (OPEN or CLOSED)
Garage eDLT button	Local Network/Lighting/Group 36 (0/56/36)	Garage door button press (on the wall eDLT)
Garage light delay routine	Local Network/Lighting/Group 37 (0/56/37)	Turn on both garage inside lights for 5 minutes

Event-based: Garage AUX button

```
1 -- This routine activates if the garage Door remote control AUX button is pressed (in the car usually)
2
3 if (event.getvalue() == 255) then -- only do this routine on the initial edge of the bell press
4
5   SetLightingState(37, true) -- activate Garage light delay routine
6
7 end
```

Event-based: Garage eDLT button

```
1 -- This routine activates if the garage Door button is pressed on the garage eDLT
2
3 if (event.getvalue() == 255) then -- only do this routine on the initial edge of the bell press
4
5   SetLightingState(37, true) -- activate Garage light delay routine
6
7 end
```

Event-based: Garage door status

```
1 -- Garage door status (update string text)
2
3 if event.getvalue() == 0 then
4
5   SetUserParam(0, 'Garage_door_status_string', 'OPEN')
6
7 else
8
9   SetUserParam(0, 'Garage_door_status_string', 'CLOSED')
10
11 end
```

Resident: Garage Light Delay Timer

```
1 -- Checks to see if Garage Lights should be automatically turned off
2
3 if ((GetUserParam(0, 'Sleeping') == 1) and ((os.time() - GetUserParam(0, 'GLDST')) > 300)) then -- first time delay is 5 minutes
4
5   SetLightingState(29, false) -- Nearest the panel door (turn it off)
6
7   SetUserParam(0, 'Sleeping', 2) -- start the second delay
8   SetUserParam(0, 'GLDST', os.time())
9
10 elseif ((GetUserParam(0, 'Sleeping') == 2) and ((os.time() - GetUserParam(0, 'GLDST')) > 60)) then -- second time delay is 1 minutes
11
12   SetLightingState(30, false) -- Nearest the workbench (turn it off)
13
14   -- Reset flag that started the delay script
15   -- This triggers the script that then sets Sleeping = 0
16   -- and, of course, turns off the switch led and Delay icon on the visualisation
17   SetLightingState(37, false)
18
19 end
```

These routines are quite straightforward. The `os.time()` is unix epoch time in seconds (seconds since midnight 1/1/1970), and UserParameter 'GLDST' is Garage Light Delay Sleeping Time, being the time for the lights to turn off.

Originally, this was coded so that both sets of garage lights turned off at the same time, plunging the garage into darkness – and too bad if someone was still in there! The code above turns the first set of lights off after 5 minutes, and the second set off after a further minute (so some warning is given!). This staggered turn-off behaviour is coded in the Resident script.

The single Resident script:

Script name ▲	Sleep interval (seconds)	Description
Garage Light Delay Timer	5	Checks to see if Garage Lights should be automatically turned off

```
Resident: Garage Light Delay Timer
1  -- Checks to see if Garage Lights should be automatically turned off
2
3  if ((GetUserParam(0, 'Sleeping') == 1) and ((os.time() - GetUserParam(0, 'GLDST')) > 300)) then -- first time delay is 5 minutes
4
5      SetLightingState(29, false) -- Nearest the roll-a-door (turn it off)
6
7      SetUserParam(0, 'Sleeping', 2) -- start the second delay
8      SetUserParam(0, 'GLDST', os.time())
9
10 elseif ((GetUserParam(0, 'Sleeping') == 2) and ((os.time() - GetUserParam(0, 'GLDST')) > 60)) then -- second time delay is 1 minutes
11
12     SetLightingState(30, false) -- Nearest the workbench (turn it off)
13
14     -- Reset flag that started the delay script
15     -- This triggers the script that then sets Sleeping = 0
16     -- and, of course, turns off the switch led and Delay icon on the visualisation
17     SetLightingState(37, false)
18
19 end
```

The UserParameter 'Sleeping' is 0 if there is no light delay active, 1 if the first 5 minute delay is active, and 2 if the last 1 minute delay is active.

Conclusion

The purpose of this application note was to demonstrate the use of the Vivisys EasyConnect board. The explanations, settings, and scripts illustrate how it can be used with confidence to make connecting miscellaneous items easier.