ACC8_2 CBUS module for driving LEDs, lights, relays etc.

Introduction

ACC8_2 is one of a number of layout control modules for use with the CBUS system. This is a general purpose layout control bus (LCB) using the industry standard CAN bus. For more information on CBUS, see the introductory article on the website (www.cbus.org.uk).

ACC8_2 is a ‘consumer’ only module which accepts and learns events sent over the CBUS by other (producer) modules. It drives 8 separate outputs using an ‘open collector’ Darlington driver IC (ULN2803A). This module provides a wide range of supply voltages and the driver IC has built-in diodes which allow use with inductive loads such as relays and motors without the need for external diodes.

ACC8_2 defaults to the SLiM (Small Layout interface Model) of CBUS which allows it to be set up and taught without any need for a programming device or computer. However, like all other SLiM modules, it responds with its CAN-ID when interrogated by ‘nodes’ which implement the self-enumeration scheme so is compatible with modules running in the Full Model (FLiM). The firmware allows for running in either mode.

Please refer to the schematic ACC8_2_sch.pdf.

Power supply.

This module requires a 12V DC supply. Note that all other _2 CBUS modules require a similar 12V DC supply and the same supply can be used for all.

The DC output available for driving relays, lights etc can be adjusted using the potentiometer VR1 over the range 5 to 10 volts. If an output of 12V is needed the regulator can by bypassed. This is a useful facility as it allows for relays or lamps of 12V as well as covering the 5V needed by some logic devices. It is not possible to have different voltages for each output though unless external resistors or regulators are used. The output terminal block has the +ve supply and its 0v available on separate terminals.
(Note, if the 12V supply is to be used for output the LM317 and associated components can be omitted).

There is a separate voltage regulator (U2) which supplies the rest of the circuit with a fixed 5v. Diode D1 protects the 5V regulator and the rest of the low voltage circuit from connection to the wrong polarity. The maximum current available from this regulator is 1 amp and even with no external load, a small heatsink is necessary.

In SLiM mode, the green LED (LD2) will illuminate to show the circuit is working correctly. This is not just a power on indicator but confirms correct working of the processor.

When in FLiM mode, the yellow LED (LD1) shows correct running.

Output drive capability.

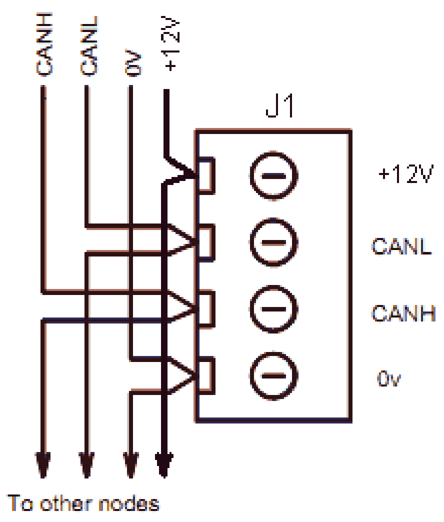
Each of the outputs can sink a current of 500mA maximum but the total current is limited to 1.25 amps, or an average of 150ma each if the chip is not to overheat. The regulator (U5) will limit the current to about 1.5 amps except when the regulator is bypassed to give a 12V output. Hence short duration pulses of 500mA per output can be used for solenoids or motors but the maximum continuous current with all outputs active must be less than 1 amp. Depending on the voltage setting with VR1, the regulator U5 will need substantial heatsinking if heavy current loads are expected.

When driving conventional LEDs, an external series resistor is required for each LED.

The output is designed to allow a choice of a darlington driver for relays etc. or a direct link from the PIC via a 1K resistor for driving 5V logic devices such as the Servo4. Selection can be made for individual outputs as required.

ACC8_2 is designed for continuous on or off outputs and does not have a capability to generate short pulses of itself. However, short bursts can be effected by sending an ON event followed by an OFF event to the same output.

Connecting the module



The CANH and CANL wires go to all modules. They are polarity sensitive so CANH must go to CANH and CANL to CANL. These wires should preferably be a twisted pair but it is not essential, especially for short distances.
The 0V must also be connected to all modules.

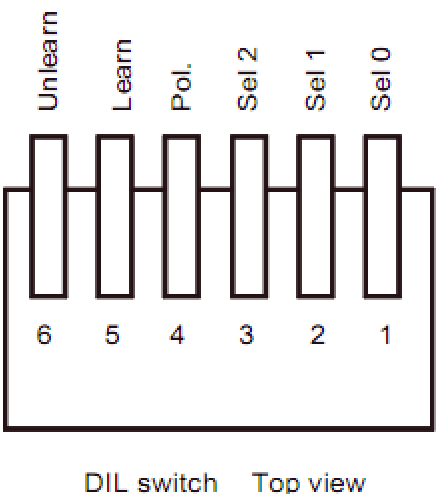
Where a number of modules are powered off the same 12 V DC supply, then their 12V input terminals should all be connected together.

While it would be usual to wire the bus sequentially round the various modules, it is not essential and individual nodes can be 'star' connected if this is more convenient.

The CAN bus requires 'termination' resistors at some point in the network. If the bus is wired sequentially round the modules, then a resistor of 120 ohms should be fitted across the bus at each end. For small layouts, it is sufficient to have a resistor across the bus at one point. The value is not critical and a 68 ohm resistor will suffice.

Note: Where the CBUS contains a mixture of the older AC powered modules and the current DC powered modules the 12V DC should be independent of the AC supply, **not** derived from the same transformer. Only the CANH, CANL and 0V should be connected between AC and DC powered modules.

Training the module in SLiM mode.



The training process is a unique aspect of CBUS and provides a very powerful method of configuration without the need for any programming device or knowledge of how the system works. The ACC8_2 module has a 6 way DIL switch for training.

The three 'Sel' switches select which output the 'event' will apply to. There are 8 outputs numbered 1 to 8. The three switches allow a selection of one of the 8 outputs using a binary sequence. When the switch is 'down' (ON as written on the switch) this represents a logic 0. A switch in the up (OFF) position is a logic 1. With all three switches down, this gives a value of binary 000 and selects output 1

The following table gives all the possible combinations.

Sel 2	Sel 1	Sel 0	Output
on	on	on	1
on	on	off	2
on	off	on	3
on	off	off	4
off	on	on	5
off	on	off	6
off	off	on	7
off	off	off	8

To train the ACC8_2 module, you need a CBUS 'producer' module which creates events. This could be an ACE3_2 (CANACE3) control panel scanner, an ACE8C_2 (CANACE8C) switch input module or a PC program which creates events via a CAN-USB module.

Connect the various modules and apply power. Connect the 'devices' you want the ACC8_2 to operate to the output terminals. Remember to adjust the output voltage with VR1 to the desired value. This supply is available on J3 pins 1 (+ve) and pin 10 (0v). Select the output you want the event to operate with the Sel switches according to the above table. Put the 'learn' switch to ON (down). Send the event. If it is an ON event, then the corresponding output will turn on. Try it with the same event number but as an OFF event. The output should now go off. Move the learn switch to OFF (up). Test the event again. The output should go on or off as expected.

Now comes the interesting bit. A single event can activate more than one output. Set the Sel switches to another output and put into learn mode. Send the same event as previously. Now both outputs will be activated. Repeat if wanted for more outputs. This process allows a single event (like a switch change on a control panel) to create a combination of outputs (256 possibilities) for setting routes, signal aspects or alpha-numeric displays.

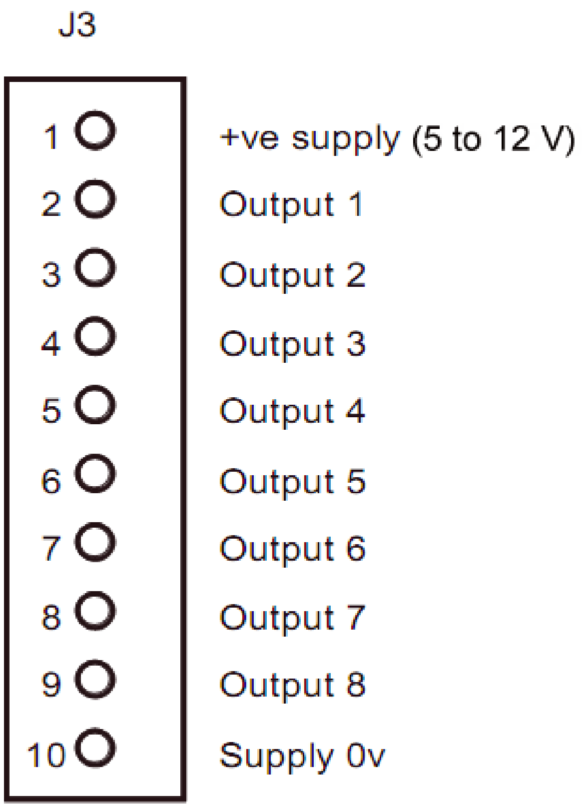
The next option is the use of the ‘Pol’ switch. With the Pol switch off, an ON event will turn an output on and vice versa. However, it is possible to reverse this by putting the Pol switch ON when learning the event. Consequently a single event can set some outputs ON and some OFF at the same time. For example, if output 1 is set normally and output 2 is set with the polarity reversed but with the same event, sending that event will cause outputs 1 and 2 to ‘toggle’ so an ON event will put output 1 on and output 2 off. An OFF event will reverse the outputs. This can be applied to any or all of the outputs.

If you want the module to forget an event it has learned, (remove it altogether), set both the learn and unlearn switches ON and send that event.

Note that different events can set the same combination of outputs. This can be useful if you want switches on different control panels or a combination of control panel and PC events to have the same effect.

Notes.

The present firmware sets the number of stored events to 32. If you try to set more than 32, no more will be added but the yellow LED (LD1) will flash to indicate the event stack is full.



The output connector

You can add more outputs to an event and alter the polarity for an existing output but you cannot remove an output from an event once set. If you want to remove an output, you need to erase the whole event and teach it again.

If you have forgotten which events the module has been taught, then you cannot tell it to unlearn an event that you don’t know it has!. In this case there is a ‘clear all’ option. This removes completely all stored events. To do this, set the unlearn switch to ON with the learn switch OFF. Remove power from the module and then reapply it. This will clear the memory.
Remember to switch off the unlearn switch after this process.

The PCBs include provision for in-circuit serial programming and debugging (ICSP). The firmware also contains code for use with the CBUS bootloader so code can be updated over the CBUS. (see document ‘CBUS_Bootloader.pdf)

The small pushbutton S1 is for switching between the SLiM mode and the Full (FLiM) mode. For this feature code version **ACC8p** or higher is required.
For operation in the Full mode (FLiM), see the document 'Full_mode_operation.pdf

Resistor R1 determines the rise and fall times of the CAN waveform. The value should be 100K to minimise fast edges and possible EMI.

The full schematic, a PCB layout which is in .PDF form and can be printed to the exact size for making masks and the PIC assembly and HEX code are available on the MERG website.
These can be freely used for non-commercial purposes. Copyright to the designs is held by the authors.