CAN-RS An interface module between CBUS and Serial RS-232

Introduction

CAN-RS is one of a number of 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 this website.

CAN-RS is unique amongst the CBUS modules in that it is neither a producer nor consumer of events but a simple interface between CBUS and a standard serial port. Its primary function is to link CBUS with a computer (PC) or other device using RS-232 serial protocol. It acts as a two way message passing node.

Please refer to the schematic CAN_RS_sch.pdf

Power supply.

CAN-RS is a very basic node which requires an external 5 volt supply. This may be taken from other CBUS modules via the CBUS connector pins 1 and 4 or supplied directly to the two pin terminal block on the board. When such an external 5 volt supply is used, it may be connected to the CBUS terminal bock for use by other modules by fitting a link across R1 on the PCB. Pads are available for this. Care must be taken not to fit this link if the CBUS is already powered from other modules or it will result in two 5 volt supplies being wired in parallel.

The serial interface.

The connection is via a standard 9 way D socket requiring a ‘one to one’ cable to the PC. The BAUD rate is fixed at 115.2Kbps. This is the highest common rate available on PCs although even this rate does not allow continual transmission of the CBUS frames which use a CAN bit rate of 125 Kbps. At present, no handshake is used although the PCB layout allows for this facility in the future.

The serial protocol.

There are several published CAN to serial protocols but we adopted the one by ‘Gridconnect’ to whom full acknowledgement is made. The full protocol is available from

http://site.gridconnect.com/docs/CAN/can-rs232.pdf

although we have only used the basic message format. This is described below.

The information on the serial side uses ASCII characters. This simplifies message parsing by the PC and is compatible with most software, e.g. Visual Basic. However, the structure of the ASCII string follows that of a CAN frame so there is direct correspondence between the CAN frame and the serial string.

The header.

Following the 'Gridconnect' scheme, the ASCII string starts with a ":" followed by an "S" to indicate a Standard CAN frame. CBUS only uses Standard frames. The next 4 chars are the ASCII version of the two header bytes in HEX.. This is departure from the Gridconnect format as CBUS uses a 7 bit node ID and 4 priority bits rather than just an 11 bit number. These two bytes map directly into the bytes sent and received by the CAN processor as SIDH and SIDL. (Standard IDentifier High byte and Standard IDentifier Low byte)

An example would be where the CBUS priority bits are 1011 and the CAN ID number is 0000001. These bits become the two bytes of the CAN header as follows

10110000 00100000 or in HEX form, B020. SIDH is B0 and SIDL is 20. This gives the string so far as :SB020 or in ASCII,

3A 53 42 30 32 30

The frame type

The next character is either "N" or "R" signifying a Normal or a RTR frame (RTR is Remote Transfer Request). Except during the self enumeration process, CBUS only uses Normal frames.

The data segment

A CBUS frame has up to 8 data bytes and the remainder of the string is the data bytes in ASCII (HEX) form. The string is concluded by a ";" Note, there is no value indicating the number of data bytes. This is worked out by the firmware in the CAN-RS module. If a frame has all 8 data bytes then the format for a normal frame is as follows.

:ShhhhNd0d1d2d3d4d5d6d7;

Where hhhh is the two byte header and d0 to d7 are the 8 data bytes. If the header is B020 as above and the data is 1,2,3,4,5,6,7,8 then the ASCII string becomes

3A 53 42 30 32 30 4E 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 3B

Exactly the same format is used for data to or from the CAN-RS module.

A software program (CAN_RS2) is available which allows entry of the various bytes in either HEX or binary and then sends the appropriate ASCII string. The same software also displays incoming CBUS frames in HEX and binary and also whether the frame is N or R and the number of data bytes. This program is written in Visual Basic 5 but a compiled (installable) version is also available. See the MERG website.

Notes.

While the CAN-RS module can buffer two successive CBUS frames, it cannot keep up with a continuous stream of CBUS frames without any gaps. It is not a true 'sniffer' but is satisfactory for all normal layout control activities.

The PC used must be capable of supporting a BAUD rate of 115.2Kbps. A reasonably fast PC is also recommended. The PC should also have a genuine serial port. Attempts to use USB to RS-232 adapters have not always proved reliable as some of these cannot run at 115.2 Kbps.

The pushbutton S1 on the PCB is only used for the 'self enumeration' scheme where there are multiple CAN-RS modules and can be ignored if there is only one on the bus. The self-enumeration sequence using the pushbutton and the two LEDs is still under development and not presently implemented in the SLiM modules.

Although a CAN ID number is included in the ASCII string sent to the CAN-RS module, it is not actually used by the module which has its own CAN ID number. The priority bits are sent so the PC can determine the priority of the outgoing message. (The CAN ID is not part of the CBUS message). However, for received CBUS messages, the CAN ID sent to the PC is that of the received frame, not the CAN-RS module. This allows for monitoring other node CAN IDs as well as their message priority.

The SLiM PCBs include provision both for in-circuit serial programming and debugging (ICSP) and two LEDs driven off the programming pins. These LEDs and associated resistors are for diagnostic purposes and may be omitted. Where fitted, low current LEDs are recommended. If 2mA LEDs are used, the series resistors can be 1K8 or 2K2.

A diagnostics and test program is being developed for CBUS. This will entail additions to the microcontroller firmware on the CAN-RS module which will be posted when available.

Resistor R4 determines the rise and fall times of the CAN waveform. Until such time as an optimal value has been determined, R4 should be just a wire link (zero ohms) which sets the maximum available rates.

Mike Bolton and Gil Fuchs 8th February 2008 ©

Corrected 2nd March 2008