CAN_RS test software package for CBUS.

Introduction

The software for the CAN_RS module is the same as used for the CAN_USB module and is installed in the same way using the CAN_USB3 code and installation procedure.

CAN_USB3 is a software utility for sending and receiving CBUS messages. It connects to the CBUS via a CAN_RS or CAN_USB module and a serial port on a PC. Operation of the CAN_RS is at a fixed rate of 115.2 Kbps. No handshake is used and while a conventional RS-232 serial port is recommended, successful operation has been achieved using a USB to serial converter although operation with all such converters is not guaranteed.

This program is purely a test utility. It is not for operating layouts. The serial bit rate of 115.2Kbps is the highest normally available on a PC. However, the CBUS CAN rate is 125Kbps and the serial data to the PC is in ASCII format so the program will not transfer CBUS frames at the continuous CAN rate. It is not a true 'sniffer'. (Use a CAN_USB for this)

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 except for the 'bootloader'. 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 CAN_USB3 version of the software, when used with the latest firmware in the CAN_RS module (rev d) now allows extended frames as well as the standard ones. Instead of the letter 'S' for a standard frame, the extended frame starts with an 'X'. The CAN header now

has 4 bytes and the extended identifiers are displayed or entered in the EIDH and EIDL boxes.

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 PC can send RTR frames if you change the N to R in the ‘send’ box.

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

corresponding to : S B0 20 N 01 02 03 04 05 06 07 08 ;

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

The utility program.

Installation

The program is available as an installable .exe from the MERG website. Download the CAN_USB3.CAB, SETUP.LST and SETUP.EXE into a suitable folder. All three files must be in the same folder. Then run setup.exe. This uses the Microsoft Installation Wizard so just follow the instructions. You will be prompted for a folder to install the program into. You can use the default or any other you choose – but remember what it was. However, you can’t install it into the same folder as the downloaded files. After the installation is complete, locate the installation folder where you will find CAN_USB3.EXE. You can then run the program or create a ‘shortcut’ (mouse right button) and drag this onto the desktop for future use.

Running the program

Open the program from the EXE file or the shortcut. The program form will appear along with a box prompting for the serial port number. The default is 4. Generally the serial COM port is a lower value. If you are using a USB converter, you will need to change the number to the one for your converter. Set the correct number and click on OK. The PC must be connected to a CAN_RS interface which requires its own 5v supply. See the documentation on the CAN_RS module.

The form is arranged in two sections. The left hand set of boxes is for incoming frames from the CBUS. The right hand set is for entering CBUS frames to send.

Incoming frames.

At the top is a frame counter. This simply displays the number of received frames. The adjacent 'clear' button clears the counter. The maximum number is 1000. You will then get a 'Buffer full' message.

Below the frame counter is a text box which displays the incoming frames as character strings. This has the same format as the outgoing messages. Messages are displayed 'top down' with the most recent message at the top of the display box.

Below this text box are a set of boxes which display the bytes of the last incoming frame using CAN nomenclature. The first two boxes show the SIDH and SIDL bytes in both HEX and binary and the second two boxes show the EIDH and EIDL of extended frames (if any). The DLC box (Data Length Code) gives the number of data bytes in the frame and the N / R box shows whether the frame was a normal (N) or a RTR (R) frame. The next 8 boxes show the data bytes in both HEX and binary. The 'erase' button clears all the incoming frame boxes. Note that any incoming frame will overwrite any existing frame data.

Setting a frame to send

The required byte data can be entered in the boxes in either HEX or binary. The SIDH box allows you to set a priority for the message in accordance with the CBUS protocol. These priority bits are the first 4 bits or the first nibble of the SIDH byte. A HEX value of B (1011) is the lowest priority. A value of 0 is the maximum. The remaining 4 bits of SIDH and the top 3 bits of SIDL make up the 7 bit CAN ID. The latest revision of the CAN_USB or CAN_RS module code uses this CAN_ID as set by the PC. There is no preset CAN_ID for the module. This allows for multiple PC interfaces and multiple PCs on a layout. Generally for accessory messages, SIDH can be set to BF hex where the priority bits are 1011 or 'B'. In binary this is 10111111. SIDL May be set from 00 to E0 giving a CAN_ID range of 120 to 127. This range allows for 8 PC interfaces and avoids conflict with other layout modules. However, no two PC interfaces should have the same CAN_ID.

Box d0 is the 'command' byte and must be set for whatever the message or event is to do. For example, a normal accessory event is 90 for 'on' and 91 for 'off. Boxes d1 to d4 are for the event number. For RTR frames used in the 'enumeration' process, all data boxes should be left blank.

When entering values in HEX, you can enter one or two digits. These must be numbers from 0 to 9 and upper case characters A to F. When entering in binary, only the numbers 1 and 0 are allowed. Binary values must have at least three digits to avoid the conflict between, say, 11 (binary 3) and 11 hex (17). Binary numbers such as 10 can be entered as 010, 0010 etc. If you enter more than 8 binary digits, the least significant are ignored.

If you are satisfied with the entered values in the boxes, click on 'set'. This will transfer the values with the correct parsing to the 'string to send' box at the top. You can check if this is the CAN frame you want. If not, change the boxes and click 'set' again.

When satisfied and the 'system' is ready, click on 'send' and the frame will be sent as soon as the CAN_RS node has access to the CAN bus. You can send this frame as many times as you like.

If you want to send the same frame as a received one, you can copy and paste the string from the 'received frame' box to the 'string to send' box directly. You can also edit this string before sending.

This program is for test and diagnostics. It may not trap all possible errors. It was written in Visual Basic 5 and the source code is available if anyone wants to modify or expand it.

Mike Bolton 14th January 2010 ©