

## **New CAN232 version #3 is now available from LAWICEL AB**

### **The Story behind CAN232**

The original CAN232 was originally developed as a custom project back in 1993, at that time we used an 8051 with the 82C200. In 1997/1998 it was changed over to an Atmel AT90S8515 and an SJA1000 and this is the design which has been available since then even though the AVR has changed from AT90S8515, to ATmega161 and then finally ATmega162. It has been the same physical hardware (PCB) since 1997/1998 and with over 14,000 units sold since December 1997 we thought it was time to upgrade and enhance the CAN232 a bit.

### **The Success**

The success behind the CAN232 is the 2 chip solution where we have one fast MCU and an external CAN controller making the MCU perform at 0.0% error when it comes to RS232 communication plus a simple and robust ASCII protocol. Since then a lot of companies has cloned, copied our CAN232, they even used our protocol which later is referred to as the LAWICEL CAN protocol. We won an award 2003 in an US magazine that entitled us to have the best CAN<->RS232 converter available at that time (out of 8 tested).

### **What's new inside?**

The new CAN232 is housed in the same type of blue plastic enclosure, we have made it 100% backwards compatible with the existing and original CAN232, but enhanced it in several ways, both with new improved hardware design and many new commands and functions. The new firmware does not work in the CAN232 design version #1 and #2, so for customers to take advantage of the new functions, they need to use the new CAN232 version #3 as we refer it to.

The hardware has changed to using 4-layer PCB instead of 2-layer PCB, we also added a CAN choke filter for better improved EMI/EMC behavior, we also changed the reset device in it from an old device to a new more cost effective device, then also changed from a simple 78L05 5V regulator to a more efficient LM2937 LDO regulator, so we can now power it from as low as 6VDC. Top power is increased a bit from 15VDC to 16VDC, so nominal is still 12VDC. We also changed to SMD LED's (for more cost effective production) and added two more LED's to display more information when trouble shooting and getting started. There is a new green LED indicating RS232 activity for incoming RS232 characters and a yellow LED to indicate how the CAN channel is opened.

The software has been optimized to be faster and some new commands/functions have been added as described in the section below of new commands and functions.

## **New commands and functions (V1324)**

During the last two years we have gotten feedback and requests on new functionalities for the CAN232 which has been incorporated into custom versions of the CAN232 which now are permanent in the new CAN232, below is a summary of new functions.

RTR frames are now fully supported with DLC which is required in some higher layer protocols and this is how it works in CANUSB for some years. So when sending an RTR frame, this is done by sending r or R depending on standard or extended frame plus ID, then also DLC and [CR]. Example r1002[CR] sends a standard 11 bit RTR frame with ID=0x100 and DLC set to 2 data bytes. Before the DLC was skipped and that didn't work in many standard protocols.

A new command Wn[CR] (where n is 0 or 1) is for filter mode, this command has to be sent if you want to change from the standard dual filter mode (0) to the single filter mode (1). Command can only be sent when CAN is in pre-operational state, i.e. when s or S command has been sent and before the CAN channel is opened. The value is stored in EEPROM and remembered at next startup.

Auto startup is also supported, this command should be sent when CAN channel is opened and initiated with CAN speed and filters etc. The command is Qn[CR] where n is 0, 1 or 2. 0 turns off the Auto startup function (normal CAN232 behavior), 1 saves CAN speed and filters into EEPROM. It will automatically initiate CAN232 with filters settings and CAN speed when power is turned on thus enabling it to receive CAN frames without any setup from the RS232 side. Note that Auto send is only possible, so CAN frames received will automatically send them out on the RS232 port. 2 does exactly the same as 1, but opens the CAN channel in "Listen Only Mode" meaning the CAN232 will not act as a real CAN node sending ACK's, instead just listen on bus. When opened in "Listen Only Mode" it is not possible to send CAN frames!

"Listen Only Mode" is also supported with normal operation as well with a new command L[CR]. It works as Open command, but opens the CAN channel in a "Listen Only Mode", the CAN232 can receive CAN frames but not ACK them, but it is not possible to send frames. "Listen Only Mode" is indicated with a blinking yellow LED when it is opened.

When opening CAN232 in "Listen Only Mode" there must be at least 2 other active normally opened CAN nodes in the network since CAN232 is then not count as a CAN node.

## **The future**

We are also working on a more robust industrial version to be released Q4/2009 which can work with 12-24VDC and a binary protocol to be able to shuffle more CAN frames, i.e. increase bandwidth, but will not leave the traditional ASCII protocol.

Lars Victorsson, LAWICEL AB, Sweden, May 2009