WinTAUSB DEMO Program : Operating Manual

The WinTAUSB Demo Program allows to communicates to the TAUSB board and its main purpose is to take effect on the operation with the board.

The TAUSB Board

The TAUSB board is constantly transmitting data and send a measure when it has new value. The data transmission values are in division unit.
The TAUSB board is tuned to have +/-20000 divisions full scale in case of 2mV/V strain gauge input while for amplified signal it is tuned to have 20000 divisions for all signal range .

The input connector is a standard 9 pin SUB-D female connector.

For the 2mVV strain gauge input version the pin assignment is the following:

Pin Number	Description	AEP Color
1	Signal -	yellow
2	Signal +	white
3	V+	red
4	V-	black

For the potentiometer input the pin assignment is the following

Pin Number	Description
7	V+ (3V)
8	GND
9	Potentiometer Output

The V+(3V) power supply for the potentiometer can manage up to 10mA so the TAUSB board can manage potentiometer resistance as low as 500 ohm.

Communication Protocol

Although the communication is via the USB interface application software sees the board like a virtual serial channel with the following parameters

Baud rate : 38400
Parity : NO
Data bit :8
Stop bit :1

The TAUSB board transmits a data packet at a frequency that can vary from 400Hz to 40Hz depending to the selected digital filter (see the discussion in the chapter dedicated to the digital filter).
Each data packet is composed of 5 bytes in bynary format.
To synchronize the incoming data the application software must check the pattern 1 1 1 1 on the most significant nibble of a received character.
Decoding the next four received bytes it is possible to get the TAUSB data

BYTE N.	B7	B6	B5	B4	B3	B2	B1	B0	Comment
1	1	1	1	1	HMSB				First 4 bits are the SYNC for the data packet
2	0	0	0	0	LMSB				
3	0	0	0	0	HLSB				
4	0	0	0	0	LLSB				
5	0	0	0	0	CheckSum				

The data is transmitted is a 16bit 2' complement binary format. The high bit of HMSB nibble specify the sign.
To compute the transmitted data please use the following formula.

DIVISION = HMSB*4096+LSMB*256+HLSB*16+ LLSB

If bit B3 of HMSB=1 DIVISION = -(65536-DIVISION)

To guarantee the integrity of the computed value the 5° byte is a checksum.

The checksum is calculated with the following formula

Checksum= (HMSB+LMSB+ HLSB+LLSB) AND 15

Where "AND" is the logical operator AND used to clear the most significant nibble of the calculated checksum.

C++ program example

```
bool LeggiTx()
{
DWORD cerror,nbytes;
DWORD Timer;
COMSTAT ComStat;
bool cont,result;

          Timer=GetTickCount();
          cont=true;
          result=false;
          do
          {
                    ClearCommError(hPort, &cerror, &ComStat);
                    if (ComStat.cbInQue>0)
                    {
                              ReadFile(hPort,RxBuffer, 1,&nbytes, NULL);
                              if ((RxBuffer[0] & 0xf0)== 0xf0)
                              {
                                        ClearCommError(hPort, &cerror, &ComStat);
                                        if (ComStat.cbInQue>=4)
                                        {
                                                  ReadFile(hPort,&RxBuffer[1], 4,&nbytes, NULL);
                                                  result=true;
                                                  cont=false;
                                                  PurgeComm(hPort,PURGE_RXCLEAR);

                                        }
                              }
                    }
                    else
                    {
                              if ((GetTickCount()-Timer)>200) cont=false;
                    }
          }while (cont);
          return result;
}


void xxxx::OnTimer(UINT nIDEvent)
{
BYTE CheckSum;
int i;
int Carico;
                    CheckSum=0;
                    for (i=0;i<4;i++)
                              CheckSum+=RxBuffer[i] & 0xf;

                    if (RxBuffer[4] == (CheckSum & 0xf))
                    {
                              if (RxBuffer[0] & 0x8)
                              {
                                        Carico=(RxBuffer[0] & 0xf)*4096;
                                        Carico+=(RxBuffer[1] & 0xf)*256;
                                        Carico+=(RxBuffer[2] & 0xf)*16;
                                        Carico+=RxBuffer[3] & 0xf;
                                        Carico=-(65536-Carico);
                              }
                              else
                              {
                                        Carico=(RxBuffer[0] & 0xf)*4096;
                                        Carico+=(RxBuffer[1] & 0xf)*256;
                                        Carico+=(RxBuffer[2] & 0xf)*16;
                                        Carico+=RxBuffer[3] & 0xf;
                              }
                    }
……
……
```

TAUSB Commands

There are a few commands that allows to operates on the TAUSB board.

The commands are performed transmitting to the board a command code. It consists of just a character . No answer is sent back from the TAUSB board.

DIGITAL FILTER

The TAUSB board filters the converted signal by using a two stage digital filter.

The first digital filter is just a pure average filter. Before to give a valid value the converted data are summed up and at the end divided by the number of data taken in account. So greater is the number of used samples longer is the time to output a valid value.

The number of samples for the average filter is programmable by software from 0 (no filter at all) to 99 (average on 100 samples).

In case of 0 it is possible to reach a 400Hz data rate for the transmitting data.

In case of 100 the transmitting rate is about 40Hz.

The output of the first average filter is the input for a moving average filter.

The moving average filter is a FIFO (First In First Out) buffer and it is possible to enable (to have a stabler value) or disable (to have a faster but a noisier value). In case it is enabled the first stage digital filter is performed using a fixed number of samples (100 samples). It is possible to define the length of the moving average filter from 1 to 100. In case of 1 the digital filter will be faster but noisier. In case of 100 the digital filter output will be stabler but slower.

Digital Filter Commands

Code 145 : Enable the second stage digital filter

Code 147 : Disable the second stage digital filter

Code from 0 to 99 : Filter parameter.

In case of the second stage digital filter is disabled rappresents the number of samples for the first stage filter

In case of the second stage digital filter is enable rappresents the length of moving average filter. The number of samples for the first stage filter is fix to 100.

Zero On /Zero Off Commands

Code 129 : perform the zero of the signal

Code 130 : clear the zero signal

Peak +On / Peak-On / Peak Off Commands

Code 132 : set the Peak+ Mode : The positive maximum value will be transmitted

Code 133 : set the Peak- Mode : The negative maximum value will be transmitted

Code 131 : Reset The Peak Mode.

Saving Data Row and Export Data in Microsoft Excel

By pressing the **SaveData** button the WinLoadCell Program start to record the data acquisition.
By Stopping the data recording the data will be stored in the “Prova.txt” file in the program installation folder.
An example of the format of the file is the following: the first value is the measured value the second is the acquisition time

-8181,0.0
-8182,0.3
-8180,0.4
-8185,0.5
-8182,0.6
-8182,0.7
-8177,0.8
-8183,0.9
-8181,1.0
-8186,1.1
-8179,1.3

These data can be exported in a Microsoft Excel file by pressing the “Export Last Saved Data in Excel” Button. This button is enabled if a valid recording test has been performed.