UT61X Multi DMM Utility

This note describes a data acquisition system based on the UNI-T Multimeter UT61X models. Up to four UT61X multimeters can be used to log data at the same time. Data from the DMM (Digital Multi Meter) is taken via an opto coupled RS232 connection. Data can be shown in a real time graph and be saved to a CSV file or a database. The Utility is developed on Windows 7 SP1.

The intention of this utility application is to be able to collect data from up to four DMM's and simultaneously measure and collect data. Data for each DMM is presented on the PC and a graph can be shown of the collected data. Data can also be saved to a CSV file or database for later analyze. The main goal was to be able to quickly set up the system to catch several parameters from an experiment like volt, current, temperature, resistance etc. No programing is necessary, just plug and go. A schematic sketch is shown in Figure 1.

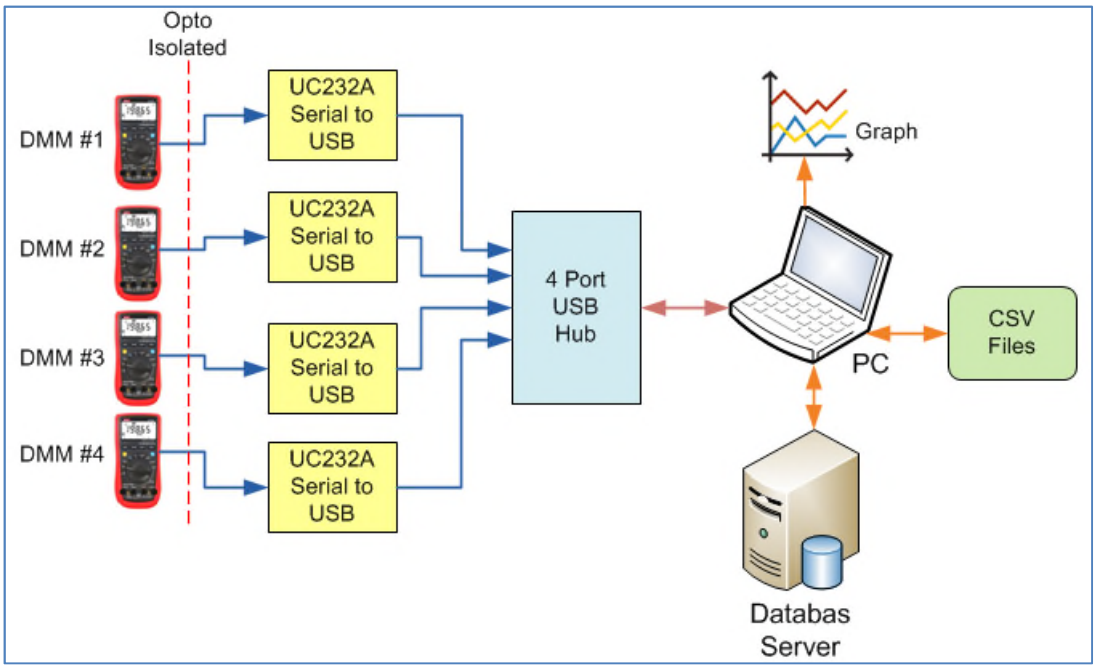

Figure 1, Block diagram of the system

The utility only works with UNI-T multimeters of type UT61X. There are several types of the UT61 multimeter model, named A, B, C, D and E. A general picture of the multimeter can be seen in Figure 3. This utility has been tested with the B, D and E model.

The UT61X DMM's are probably one of the most prices worthy DMM's on the market (reference 4). Some models can be seen in Figure 3. They have most of the functionality you expect including temperature measurement. They also have an opto coupled RS232C connection to collect data to a PC. Model A, B, C and D uses the same protocol for data transmission and uses 2400 baud. The E-model uses 19200 Baud and a slightly different protocol. The D model has 6000 counts but the E-model has 22000 counts and gives a better resolution. The sampling speed is about 0.6 seconds per sample.

In Figure 2 we can see a setup using a laptop computer running Windows 10 using the application for data acquisition during a test of the application. In this case 3 UT61 DMM's where connected via a USB hub, which can be seen in the middle top of the picture, to the PC. A RS232 breakout monitor was connected to one of the DMM's to check the serial communication.

Figure 2, testing application with three DMM's

The RS232 data stream is synchronous. The communication is only one way, from the DMM to the PC. The PC sends no commands to the DMM. For the A, B, C and D model the user has to push a button to initiate the serial RS232C communication. On the E-model it is always on. The power consumption is low and a fresh battery works for several days for continuously data collection.

Figure 3, UT61E, UT61D and UT61B models

For more general information about the UT61X DMM's see reference 1.

The UT61X DMM models use an opto coupled RS232C connection. The connection and cable can be seen in Figure 4. This has the big advantage that we avoid ground current loops that can disturb the measurements.

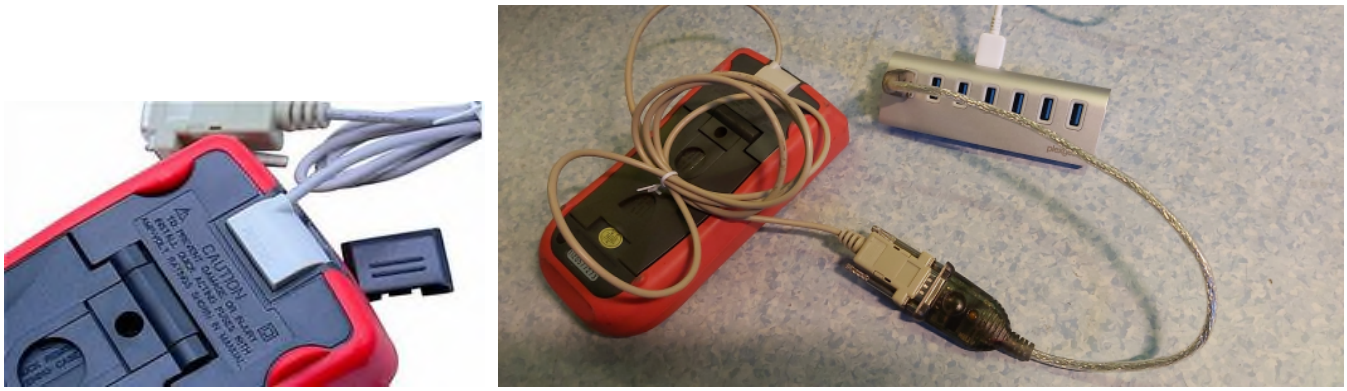

Figure 4, Opto coupling of the DMM and connection to US232A device and USB Hub

As can be seen in Figure 4 the opto cable from the DMM is connected to a US232A unit. This is a RS232C to USB converter. Due to that most laptops PC just have a few USB connectors a USB hub is used to connect up to four DMM's. The US232A USB converter and USB hub can be seen in Figure 5.

Figure 5, UC232A unit and four port USB hub

Application Layer

The PC application is written in Delphi XE2. Delphi is a powerful object oriented with a fully integrated IDE (Integrated Development Environment). You can easily add components to add functionality to Delphi, for example support for serial or Ethernet communication. The main screen of the application can be seen in Figure 6.

When you enable a DMM the current measured value for the DMM will continuously be displayed. The "*Settings/Info*" informs the user about the current configuration of the DMM. The "*Setup*" is used to configure the DMM for the Serial port (Com Port) that the DMM is connected to etc. Here you also set the sampling time for the collecting of data.

When the user checks the *Enable* the application starts to read values from the connected DMM. The Alarm and Set Alarm level will be described later. Zeros is a legacy function used to enable or disable leading zeros of the measurement values. For example the value 1.28 KOhm will be displayed as 001.28 KOhm when enabled.

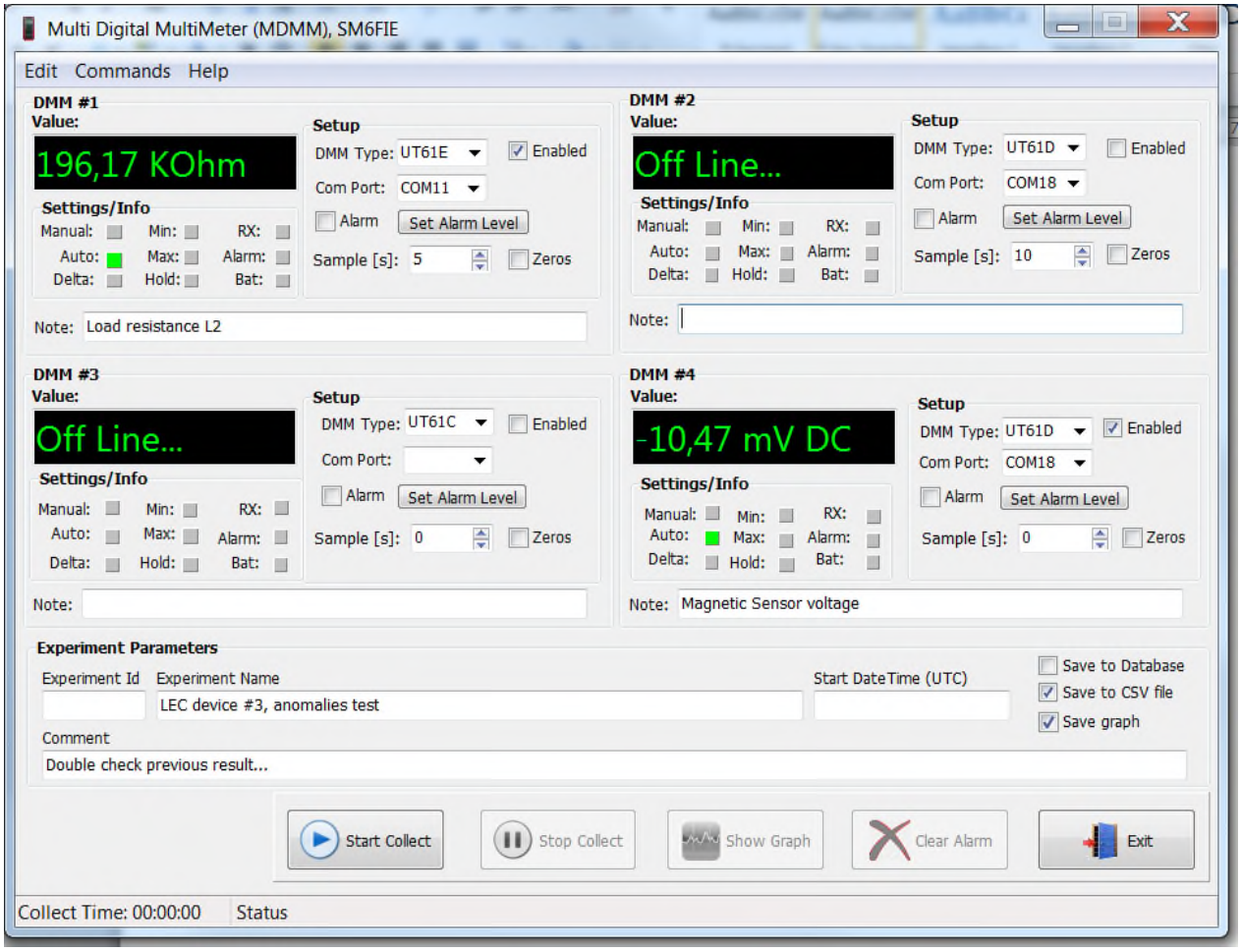

Figure 6, main screen of the UT61X DMM application

When the user clicks on the "*Start Collect*" button the system starts to make a graph of the incoming data. If "*Save to Database*" and/or "*Save to CSV file*" is checked data are saved to database and/or CSV file (text file). Then Collection is started an information header is saved.

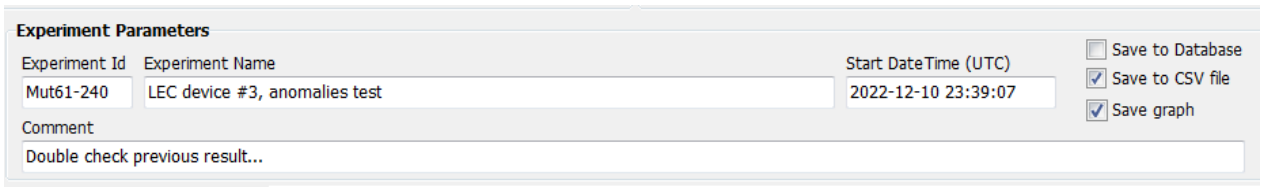

Figure 7, Experiment ID and Start date are updated by the system

Also included will be a short note for each DMM about what the DMM is measuring (e.g. temperature at outlet, power current etc.). Each collection run is given a unique "*Experiment Id*". It will also save the "*Experiment Name*", "*Start Date*" and "*Comment*".

As can be seen in Figure 7 the "*Experiment Id*" and "*Start Date*" is automatically updated by the application when the "*Start Collect*" button is clicked.

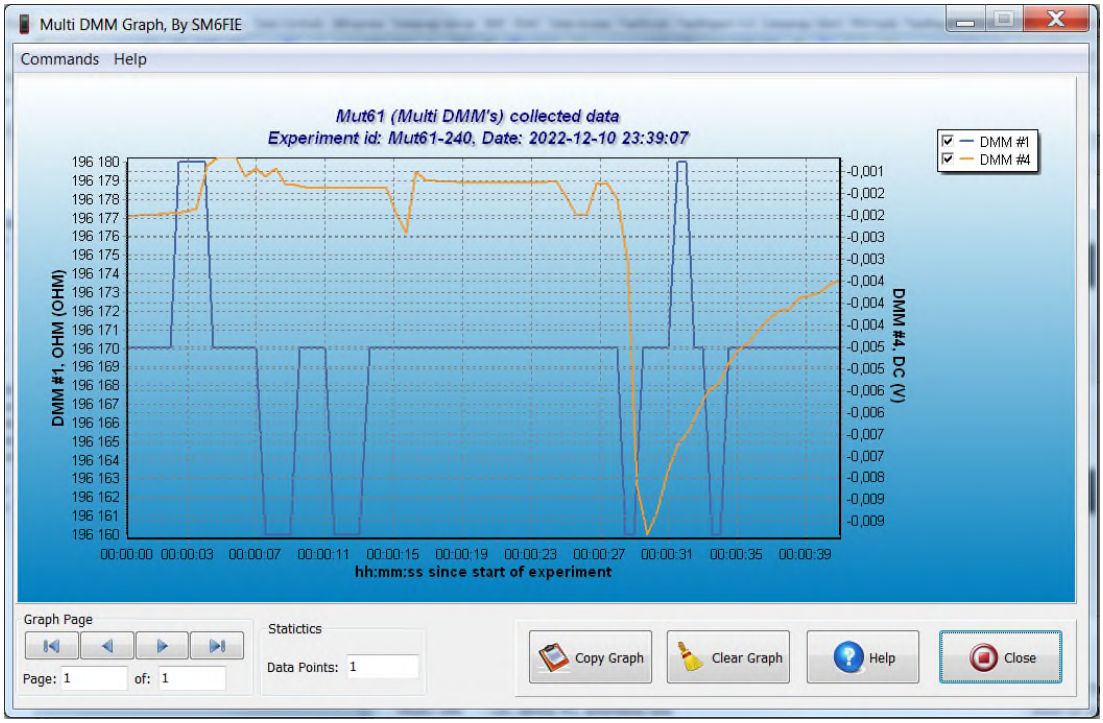

Figure 8, Graph of collected data

In Figure 8 we can see the generated graph based on the collected data. The left vertical axis is for the DMM #1 that is measuring resistance; the right vertical axis is for DMM #4 that measures DC voltage. The X-Axis shows the Time in “hh:mm:ss” since the start of the experiment.

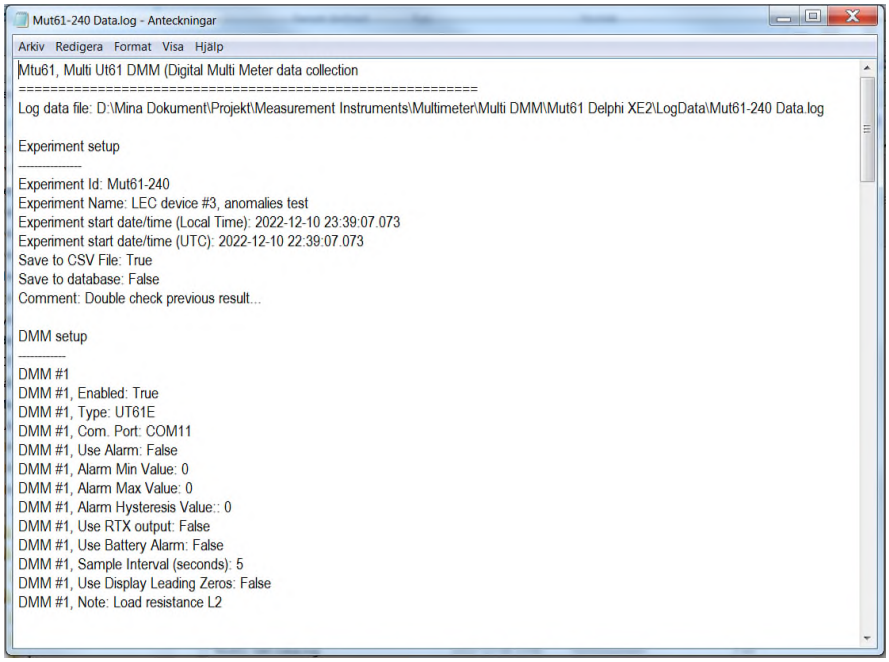

Figure 9, part of the generated CSV file

The user can zoom in and out of the graph by using the mouse buttons. The button “Copy Graph” makes a copy of the graph to the clipboard. This makes it easy to insert the graph to an experiment document. The user can also via the menu system save the graph to a file for later use.

In Figure 9 we can see part of the generated CSV file for the experiment. Figure shows the just a part of the header in the file. The data values collected can be seen in Figure 10.

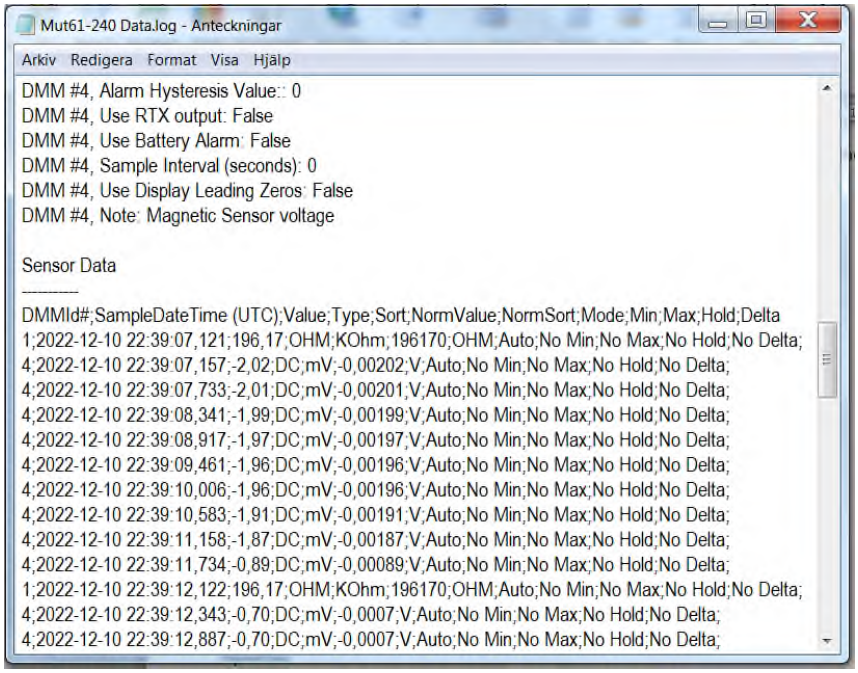

Figure 10, collected data values

As we can see the sampling time is lower for DMM ID #1 (5 seconds) compared to DMM ID #4 that samples with the maximum speed (about 0.6 seconds per sample). The sampled data can easily be imported to Excel for further calculations and analyze.

Alarm

For each DMM you have the possibility to set an alarm level, see Figure 11, Alarm level. You first have to click the "*Set Alarm Level*" button; this will show you the dialog in Figure 12, Set alarm level.

In the "*Set Alarm Level*" dialog you can set the "*Min Value*", the "*Max Value*" and the "*Hysteresis*" for the alarm. Furthermore you can also check the "*Use RTS Output*" check box to activate the RTS signal of the serial output when an alarm is activated. This signal could be used for example to turn off a heater, stirrer etc. You must enable the alarm by check the "Alarm" check box; see Figure 11, Alarm level. When an alarm is triggered an audio signal will start to alert the operator that an alarm has been triggered.

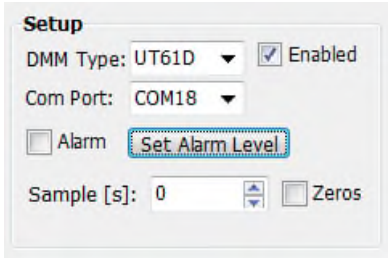

Figure 11, Alarm level

The "*Battery Alarm*" will make an alarm if the battery of the DMM becomes low.

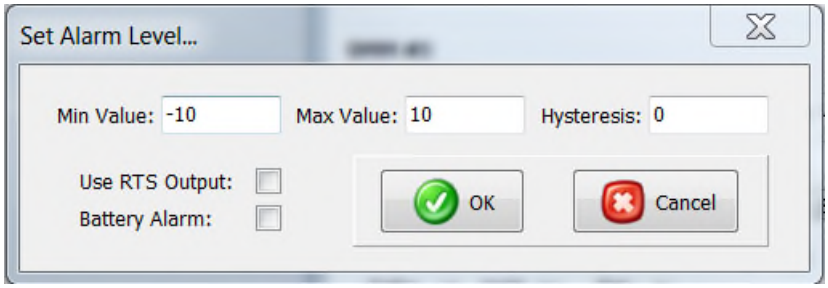

Figure 12, Set alarm level

The current implementation of the Alarm functionality is rather rudimentary and should be refined in any further release of the application. However the basic functions work as described.

Serial Ports

To know what serial port numbers to be used you can open up the device manager and find the serial port numbers, see Figure 13, device manger shows the serial ports As can be seen in the Figure 13, device manger shows the serial ports. In this case we have two ATEN USB ports, COM11 and COM18 available.

Figure 13, device manger shows the serial ports

Low level engineering notes

This section is more to document and remember some of the issues that I had during the development of the system.

One big problem was the issues I had with TurboPower Async Professional Data (TAPD). I got Packet Access Violation when using the Com Port component together with the Data packet component. The setup was the following: I use windows 7, 64 bit SP1 and TP Async V4.07 and hade the following problem:
The system was configured with two US232 Serial to USB converters using Com ports, Com11 and Com18.

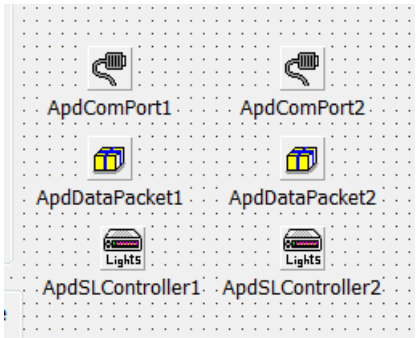

Figure 14, TAPD components

I add the following components to my form, see Figure 14:

I open a Com port on ApdComPort2 (Com18) and it works without problem. The ApdDataPacket2 detects the packet terminator and the result displayed is what is expected. Both Com11 and Com18 works fine standalone.

Now if I open another Com port (Com11), while Com18 is active with ApdComPort1 I get an Access violation. See Figure 15.

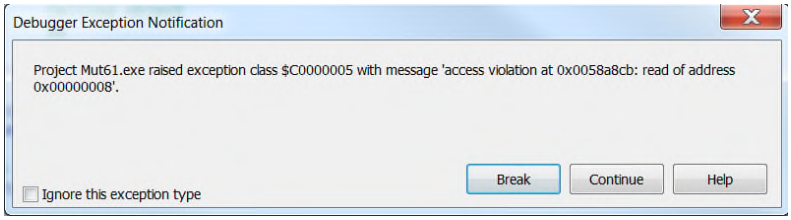

Figure 15, TAPD Access Violation

The code that generates the error is this in the AdPacket module:

```
procedure TApdDataPacketManager.EnablePackets;
var
  i : integer;
begin
  for i := 0 to pred(PacketList.Count) do
    with TApdDataPacket(PacketList[i]) do
      if Enabled then
        Enable;
end;
```

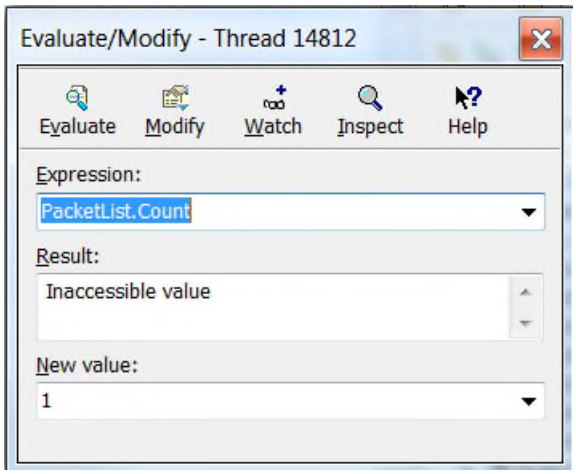

Figure 16, TAPD Packet list count

It is the "*PacketList.Count*" that seems to be the problem when it iterates through the list but I can't catch why, see Figure 16.

Note that ApdComPort2 works without problem with both Com11 and Com18.

If I remove the Apd2 components then Apd1 works as expected. The problems surfaces when I try to use two (or more) TAPD components on the same time.

After much effort I finally found a solution, see below.

Solution to Turbo Power Async Professional components access violation
When using the TAPD Async components it is very important on how you add the components to the form. If you don't does it in the right order and in the correct way it will not work if you use more than one serial port. You will actually get an access violation. For example, if you add the components below you have to do it in this way, see Figure 17.

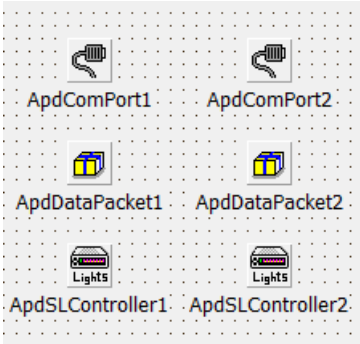

Figure 17, TAPD Async components added by copy/past

1. Add one ApdComPort to the form, it will become ApdComPort1
2. Now copy and paste this component to the form, it will become ApdComPort2
3. Add one ApdDataPacket component to the form, it will become ApdDataPacket1
4. Now copy and paste this component to the form, it will become ApdDataPacket2
5. Add one ApdSLController component to the form, it will become ApdSLController1
6. Now copy and paste this component to the form, it will become ApdSLController2

When doing it, as described above, it works to use two serial ports with ApdDatapacket. Now I don't get any getting Access violations. I have tested it up to 4 ports and it works as well.

Application load on PC
The application load on the PC is small and about 1 % of the available resources. See Figure 18, MUT61 processor load on PC. This was measure on a PC with an i7 CPU @ 1.6 GHz. The system was using Windows 7, SP1, 64 bits with 14 GByte of RAM. The application was running with 3 DMM's and was saving data to CSV file. The application was in the "Collecting" mode and was constantly updating the Graph in real time.

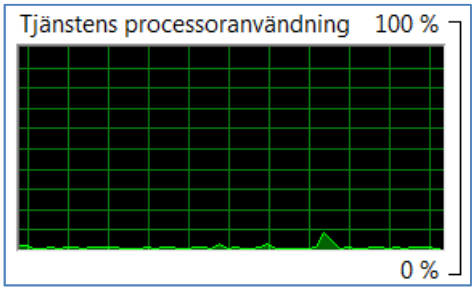

Figure 18, MUT61 processor load on PC

UT61B Current consumption
I did a measurement of the power consumption on the UT61B model. The battery voltage was about 8.9 volt. The current drain was 2.13 mA. The current drain varies slightly between different measurement types. When activating the RS232 communication the battery current did increase. The maximum value I could see was about 2.7 mA and an average about 2.3 mA. The nominal capacity of an Alkaline 9 volt battery is about 550 mAh. If we assume that the average current is 2.3 mA this would imply that the DMM will work for about 10 days before the battery voltage has dropped to a point that the DMM will stop to work. If you use a Lithium Primary battery type

the capacity is about 1200 mAh. This will give about 23 days of measurements. I suspect that the other models of the UT61X series give about the same figures.

Version info
The current version has not implemented Database support. The Alarm functions are rather limited. An update of the graph was done to enable to easily switch between the DMM's values to be displayed on the graph. This makes it very easy to check different relations between measured parameters. Note that only two parameters can be shown at the same time with one having the scale on the left y-axis and the other on the right y-axis. The author have found this to very valuable. The application was developed on Windows 7 with SP1. However it should run on later versions of Windows. The application was developed with Delphi XE2. This documentation is more an overview; there is more functionality in the application that not is described in this document. No installation script has been developed. However it's a rather simple task to just copy the EXE image and related files like the INI file. You also have to setup the folder structure. See the README file for more info.

README Installation instructions:

The flow to install the MUT61X Multi DMM's on a Windows system is as follows:

1. Make a folder named "*MUT61X Application*", this is the "App folder"
2. Copy the EXE file "*Mut61.exe*" to the "App folder"
3. Copy the file "*Mut61.ini*" to the "App folder"
4. Copy the file "*Mut61Sequences.ini*" to the "App folder"
5. Copy the file "*Alarm.wav*" to the "App folder"
6. Copy the file "*Multi_UNI-T_DMM_Utility.pdf*" to the "App folder"
7. Copy the file "*Mut61CodeTable.txt*" to the "App folder"
8. Copy the file "*Mut61DbgData.txt*" to the "App folder"
9. Copy the file "*Mut61IndexTable.txt*" to the "App folder"
10. Make a new folder in the "App folder" with the name "*LogData*"
11. Then click on the "*Mut61.exe*" to start the application
12. Change privileges on "*Mut61.ini*" to read and write for application/user
13. Change privileges on "*Mut61Sequences.ini*" to read and write for application/user
14. Change privileges on folder "*LogData*" to read and write for application/user

References

1. Delphi XE2
   https://www.embarcadero.com/products/delphi
2. UT61X Models in UT61+ series: UT61B+, UT61D+, UT61E+
   https://meters.uni-trend.com/product/ut61plus-series/#Specifications
3. USB-to-Serial Adapter
   https://docs.rs-online.com/9db6/0900766b815d6df5.pdf
   https://www.aten.com/global/en/products/usb-solutions/converters/uc232a/
4. TurboPower Async Professional
   https://sourceforge.net/projects/tpapro/
5. EEVblog 1378 - NEW Uni-T UT61E+ Multimeter - Still bang-per-buck king?
   https://www.youtube.com/watch?v=ZxzQZFRznp0

Date:	*2023-04-28 20:26*
File Name:	*Multi_UNI-T_DMM_Utility.docx*
Last update:	*2022-12-12 22:38*
Check sum:	*1024901*
Edit Time:	*9407*
Save Time:	*22:38:00*
Reversion #:	*15*
File Size:	*1024901*
Total Edit Time:	*9407*