DEVELOPMENT OF A SANS DATA ACQUISITION GPIB interface USING WAVEMETRICS IGOR

Muhammad Akmal Asraf Mohamad Sharom[1], Zainol Abidin Ibrahim[1], Wan Ahmad Tajuddin Wan Abdullah[1], Megat Harun Al Rashid Megat Ahmad[2], Faridah Mohamad Idris[2], and Abdul Aziz Mohamed[2]

[1]*Department of Physics, Faculty of Science,University of Malaya, 50603 Kuala Lumpur, Malaysia*

[2]*Materials Technology Group, Industrial Technology Division, Malaysian Nuclear Agency, 43000 Kajang, Selangor, Malaysia*

*aziz_mohd@nuclearmalaysia.gov.my

ABSTRACT

Small angle neutron scattering (SANS) is used for probing the microstructure of materials in the range between 1-100 nm in dimension. The scattered neutrons from the target material were detected by a 128 x 128 array area sensitive, Helium gas-filled proportional counter, which is known as Position Sensitive Detector (PSD). The small angle neutron scattering (SANS) facility in Malaysian Nuclear Agency has been developed since 1995. The data acquisition system of this prototype facility consists of the two-dimensional Position Sensitive Detector (2D-PSD) and neutron monitor as a data grabber, TDC Histogram as a memory processing processor, two units of ORTEC 994 as a counter and timer and a computer as a data acquisition controller via GPIB interfacing protocol. This paper will describe on the development of GPIB interface for data acquisition of the SANS instrument on Windows based platform. The GPIB device interface and graphical user interface (GUI) for this data acquisition is developed using WaveMetrics Igor software.

ABSTRAK

Serakan neutron bersudut kecil ini digunakan untuk mengkaji bahan mikrostruktur dalam saiz dimensi antara 1-100nm. Nuetron yang terserak dari bahan akan dikesan oleh 128 x 128 jalur kawasan sensitif, pembilang yang berisi gas helium yang dikenali sebagai pengesan posisi sensitif (PSD).Fasiliti serakan neutron bersudut kecil telah dibangunkan sejak tahun 1995 di Agensi Nuklear Malaysia. Prototaip fasiliti sistem pengambilalihan data merangkumi dua dimensi pengesan posisi sensitif (PSD) and monitor neutron sebagai pemunggut data, TDC Histogram sebagai prosessor memproses memori, dua unit ORTEC 994 sebagai pembilang dan pemasa dan kompouter sebagai pengawal pengambilalihan data menerusi protokol antaramuka. Kertas ini akan menerangkan mengenai pembangunan antaramuka GPIB untuk pemgambilalihan data instrumen SANS pada Windows. Antaramuka alat GPIB dan antaramuka pengguna grafik yang dibangunakan untuk pengambilalihan data dengan mengunakan perisian WaveMetrics Igor.

Keywords: Small angle neutron scattering data acquisition program, Small angle neutron scattering data acquisition system, neutron scattering, GPIB

INTRODUCTION

Small angle neutron scattering (SANS) is one the technique that belongs to the broad range scattering techniques. This technique is wide used for the characterization of materials (Hammouda, 2009) and to study microstructure and microscopic properties of matter on the nanometer length scale between 1 to 100 nm (Rétfalvi, 2003).

The SANS facility in Malaysian Nuclear Agency (Fig. 1) has been operating since 1995 (Fang, 2006). The data acquisition system consists of the two-dimensional Position Sensitive Detector (2D PSD) and neutron monitor as a data grabber, TDC Histogram as a memory processing processor, two units of ORTEC 994 as a counter and timer, GPIB and a computer as a data acquisition controller.

The 2D PSD gas-filled proportional counter is with an active area of 128 x 128 pixels and resolution element dimension of 0.5 x 0.5 cm^2 is used for neutron detection. In front of the detector, a beam stopper is placed to prevent a direct beam and the detector efficiency is 83% (Sufi, 1996). To measure the incoming neutron flux before entering the main collimator, the neutrons monitor is placed after the monochromator assembly. The monitor which is consists of a uranium fission chamber and NIM modules are used and the measurement of incoming neutron flux from the neutron monitor is displayed by the ORTEC 994 dual counter and timer.

The current SANS data acquisition program is written in QuickBasic Language and run in Disk Operating System (DOS). For users who are already accustomed to working through a GUI, running SANS in DOS command line interface is a difficult, and requires additional training. Thus, the OS in the computer is being upgraded from DOS to Windows XP. WaveMetrics Igor Pro software version 6 was used as the instrument software to develop the graphical user interface (GUI) and device interface.

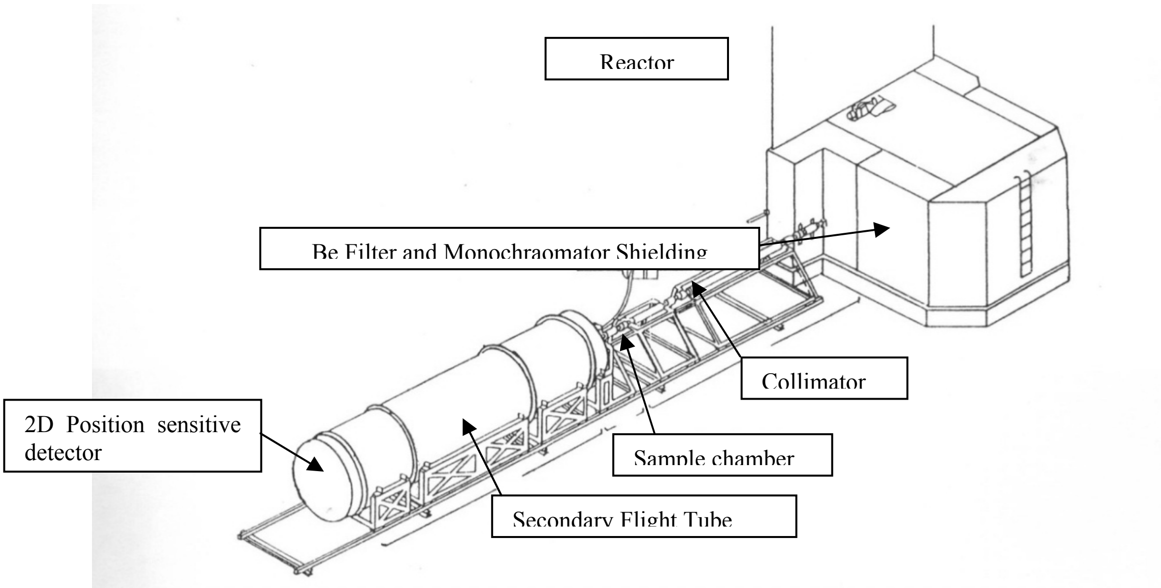


FIG. 1: General layout of SANS instrument at Malaysia Nuclear Agency (Fang.2005)

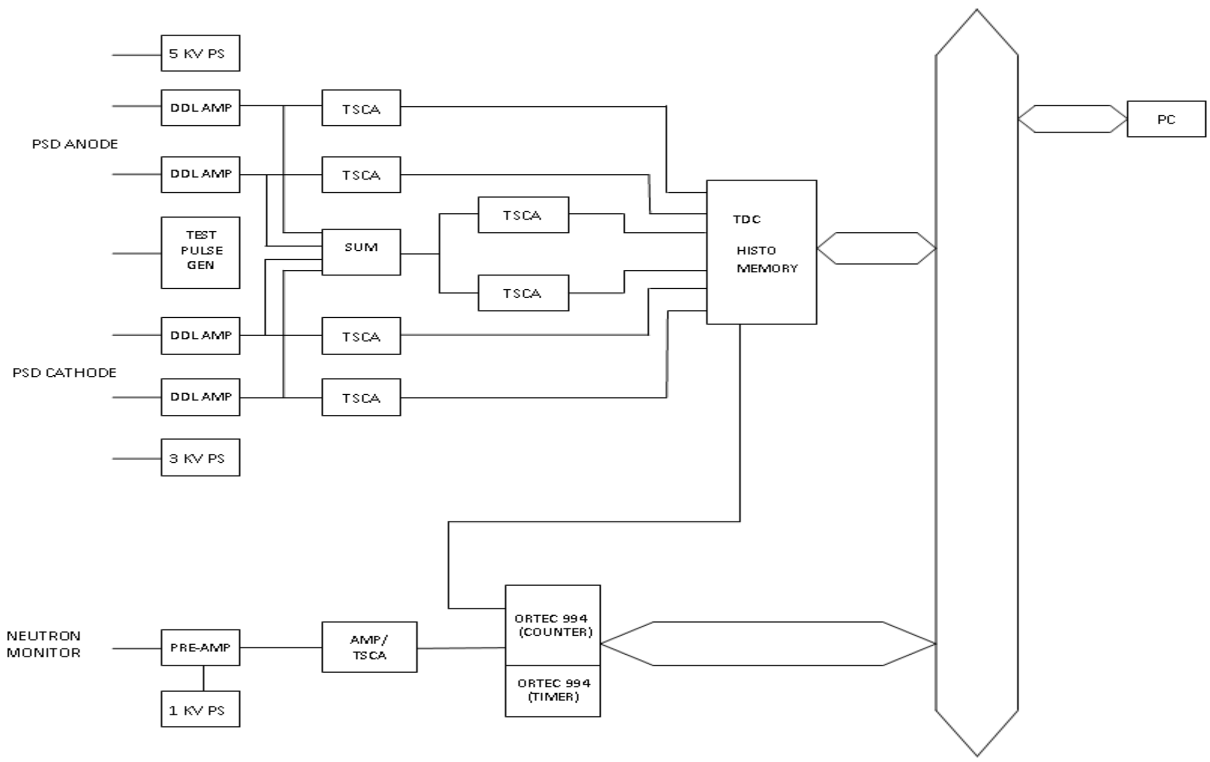


FIG. 2: A schematic layout of a data acquisition system for SANS instrument

In Fig. 2 shows a schematic layout of a data acquisition system for SANS instrument. The data collected by PSD anode and PSD cathode are accumulated in RISOE TDC histogram memory. This TDC histogram memory is connected to ORTEC 994, which acts as counter and also display the accumulated data from TDC histogram and neutron monitor detector. The other unit of ORTEC 994 functions as timer and both ORTEC 994 and TDC histogram are communicate with host computer through GPIB.

GPIB BUS

GPIB INTERFACING PLATFORM

In this research, PC-LabCard GPIB was replaced with National Instrument GPIB because PC-LabCard GPIB require ISA (Industry Standard Architecture) slot on the computer. Nowadays, most modern computers are not equipped with the ISA slot**.** Thus, the GPIB model NI PCI-GPIB (Fig. 3), National Instrument product is used. This GPIB card model is installed using PCI slot on the computer.

This NI PIC-GPIB used NI-488.2 driver/API (Application Program Interface) as data acquisition software in order for the DAQ hardware to work with a computer. This NI-488.2 driver will work with WaveMetrics Igor software after installing the XOP (External Operation) add-on. There are some differences between the National GPIB driver and the PC-Lab GPIB driver. The NI-488.2 driver used subroutine structure and PC-Lab card GPIB used character-I/O structure. The NI-488.2 driver structure includes special subroutines already programmed by a vendor. A subroutine-structured driver gets its name because these subroutines are called as routines or functions from the programming language in which the application program was written. The subroutine structure is faster, easily handles buffered DMA transfers, and uses a structured, hierarchical programming style familiar to users of modern programming languages (Jernigan, 1999).

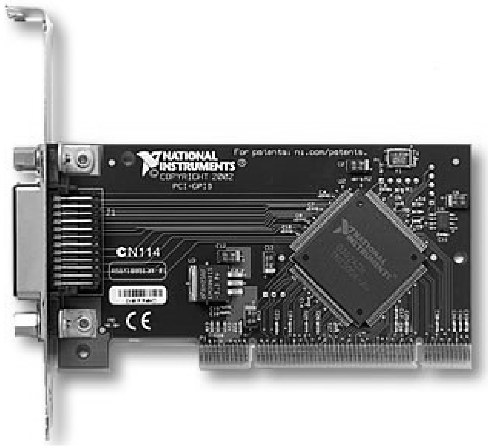
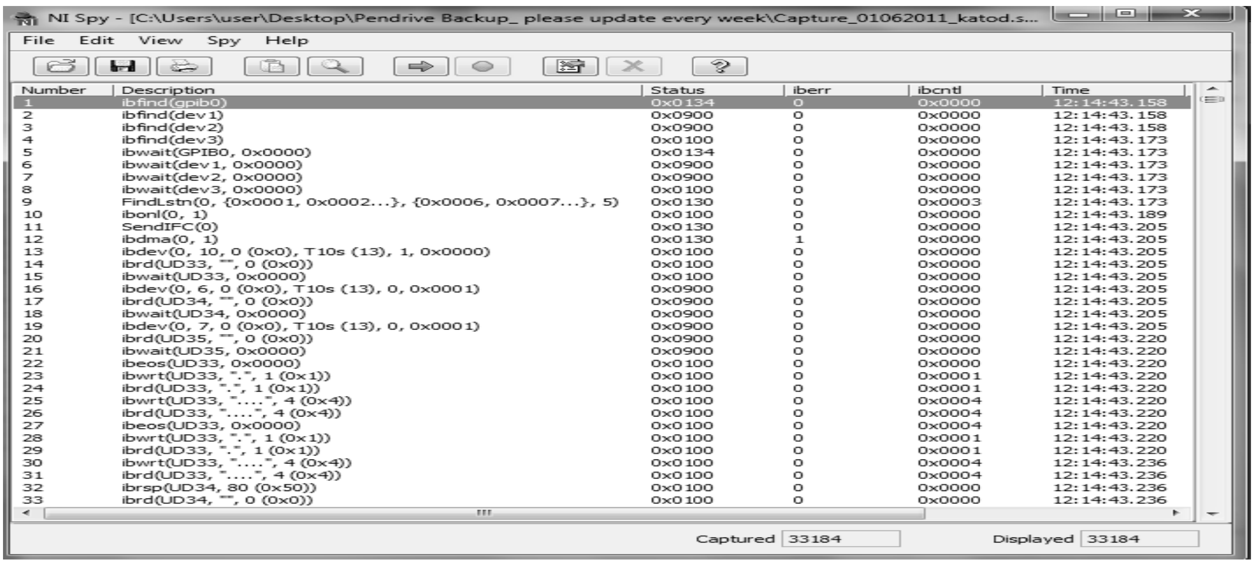


FIG. 3: National Instrument GPIB model NI PCI-GPIB

Table 1 shows some comparison of GPIB driver code functions between PC-LabCard GPIB (current SANS data acquisition program) and NI PCI-GPIB. The PC-LabCard GPIB uses PCL-848 function in QuickBasic Language and for NI PCI-GPIB we use NI-488.2 driver in WaveMetrics Igor based.

National instrument included a utility that can be used for debugging purpose. This utility is called as NI Spy (Fig. 4). This NI Spy can 'spy' on drivers calls. The NI Spy records all device calls and board level calls with a time stamp. We can easily detect errors and timing issue in our applications. We also can see the data buffer that being sent by a device using NI spy.



NI Spy - [C:\Users\user\Desktop\Pendrive Backup_ please update every week\Capture_01062011_katod.s...

File Edit View Spy Help

Number	Description	Status	iberr	ibcntl	Time
1	ibfind(gpib0)	0x0134	0	0x0000	12:14:43.158
2	ibfind(dev1)	0x0900	0	0x0000	12:14:43.158
3	ibfind(dev2)	0x0900	0	0x0000	12:14:43.158
4	ibfind(dev3)	0x0100	0	0x0000	12:14:43.173
5	ibwait(GPIB0, 0x0000)	0x0134	0	0x0000	12:14:43.173
6	ibwait(dev1, 0x0000)	0x0900	0	0x0000	12:14:43.173
7	ibwait(dev2, 0x0000)	0x0900	0	0x0000	12:14:43.173
8	ibwait(dev3, 0x0000)	0x0100	0	0x0000	12:14:43.173
9	FindLstn(0, {0x0001, 0x0002...}, {0x0006, 0x0007...}, 5)	0x0130	0	0x0003	12:14:43.173
10	ibonl(0, 1)	0x0100	0	0x0000	12:14:43.189
11	SendIFC(0)	0x0130	0	0x0000	12:14:43.205
12	ibdma(0, 1)	0x0130	1	0x0000	12:14:43.205
13	ibdev(0, 10, 0 (0x0), T10s (13), 1, 0x0000)	0x0100	0	0x0000	12:14:43.205
14	ibrd(UD33, "", 0 (0x0))	0x0100	0	0x0000	12:14:43.205
15	ibwait(UD33, 0x0000)	0x0100	0	0x0000	12:14:43.205
16	ibdev(0, 6, 0 (0x0), T10s (13), 0, 0x0001)	0x0900	0	0x0000	12:14:43.205
17	ibrd(UD34, "", 0 (0x0))	0x0900	0	0x0000	12:14:43.205
18	ibwait(UD34, 0x0000)	0x0900	0	0x0000	12:14:43.205
19	ibdev(0, 7, 0 (0x0), T10s (13), 0, 0x0001)	0x0900	0	0x0000	12:14:43.205
20	ibrd(UD35, "", 0 (0x0))	0x0900	0	0x0000	12:14:43.220
21	ibwait(UD35, 0x0000)	0x0900	0	0x0000	12:14:43.220
22	ibeos(UD33, 0x0000)	0x0100	0	0x0000	12:14:43.220
23	ibwrt(UD33, ".", 1 (0x1))	0x0100	0	0x0001	12:14:43.220
24	ibrd(UD33, ".", 1 (0x1))	0x0100	0	0x0001	12:14:43.220
25	ibwrt(UD33, "....", 4 (0x4))	0x0100	0	0x0004	12:14:43.220
26	ibrd(UD33, "....", 4 (0x4))	0x0100	0	0x0004	12:14:43.220
27	ibeos(UD33, 0x0000)	0x0100	0	0x0004	12:14:43.220
28	ibwrt(UD33, ".", 1 (0x1))	0x0100	0	0x0001	12:14:43.220
29	ibrd(UD33, ".", 1 (0x1))	0x0100	0	0x0001	12:14:43.220
30	ibwrt(UD33, "....", 4 (0x4))	0x0100	0	0x0004	12:14:43.236
31	ibrd(UD33, "....", 4 (0x4))	0x0100	0	0x0004	12:14:43.236
32	ibrsp(UD34, 80 (0x50))	0x0100	0	0x0000	12:14:43.236
33	ibrd(UD34, "", 0 (0x0))	0x0100	0	0x0000	12:14:43.236

Captured 33184 Displayed 33184

FIG. 4: NI Spy Utility Window

TABLE 1: Comparison of GPIB command

Purpose	**PC-LabCard GPIB (QuickBasic Language)**	**NI PCI-GPIB (WaveMetrics Igor Pro Version 6)**
• Open and initialize a device • Set device time out • Set device terminator	• CALL ABSOLUTE (IOPORT%, MYADDR%, SETTING%, INIT%) • CALL ABSOLUTE (TMO%, TIMEOUT%) • CALL ABSOLUTE (ADDR%, OUTEOL%, OUTEOL$,INEOL%,INBYTE%,EOL %)	• NI4882 ibdev={boardIndex, pad, sad, tmo, eot, eos}
• Clear a specific device	• CALL ABSOLUTE (ADDR%, DEVCLR%)	• NI4882 ibclr={ud}
• Read data from a device into a user buffer	• CALL ABSOLUTE (ADDR%, D$, ENTER%)	• NI4882 ibrd={ud, cnt}
• Write data to a device from a user buffer	• CALL ABSOLUTE (ADDR%, D$, OUTPUT%)	• NI4882 ibwrt={ud, str, cnt}

THE IGOR SANS DATA ACQUISITION PROGRAM

WaveMetrics Igor Pro is software that rich and powerful for data analysis and visualization program for Windows and Mac, offering quality graphics publication, a wave model for handling of variables and particular strengths in signal processing (*Grant*, 2008). WaveMetrics Igor Pro can communicate with instruments by using XOP (external operation) called VDT2 (very dumb terminal) for serial port (RS232) and NIGPIB2 for general purpose interface bus (GPIB) (WaveMetrics, 2008).

The program structure of this system is developed based on the current SANS data acquisition program (QuickBasic Fig. 5). The graphical user interface (GUI) for the new SANS DAQ was developed using WaveMetrics Igor software and all of the device interface program were written in Igor procedure. For performing reduction and analysis of SANS data, a program is being developed using Igor software (Kline, 2006). This SANS reduction program will be combined with the new SANS DAQ program into one Igor software package

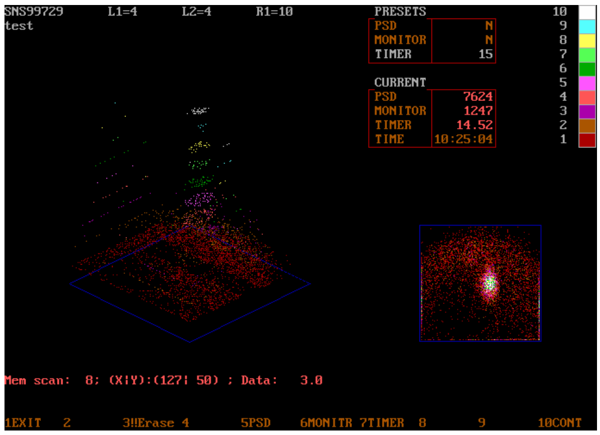


FIG. 5: SANS Data Acquisition in QuickBasic. This data acquisition program is run in DOS operating system

In the new SANS DAQ program (Fig. 6), it has a 2D data display to present the scattering pattern in 2D colour image. It also has a pulser data display which is used for electronic alignment.

In the new SANS DAQ, as the program starts, it will set the board number for GPIB in case if the computer is installed with more than one GPIB board. After setting the GPIB board number, the program will detect the TDC and ORTEC instrument and initiate TDC, ORTEC counter and ORTEC timer. To initialize each device we use *ibdev* function. This is an example how to set *ibdev* in IGOR:

- NI4882 ibdev={ 0,7,0,10,0,1}

  (1 2 3 4 5 6)

1- The GPIB Board Index. The board index for this program is 0. It depends how many GPIB board the user have.
2- Device Primary address. Device primary address is an address that being given by manufacturer. The primary address for Timer (ORTEC 994) is 7, Counter (ORTEC 994) is 8 and TDC is 10.
3- Device secondary address. There is no secondary address for each devices is this program

4- Timeout. The timeout value for this data acquisition program is 10 (300ms)
5- EOT-asserting EOI on last byte of data sourced. This terminator setting only being used by TDC in this DAQ program.
6- EOS- End-Of-String terminator. This terminator is used by Counter (ORTEC 994), Timer (ORTEC 994) and not being used by TDC.

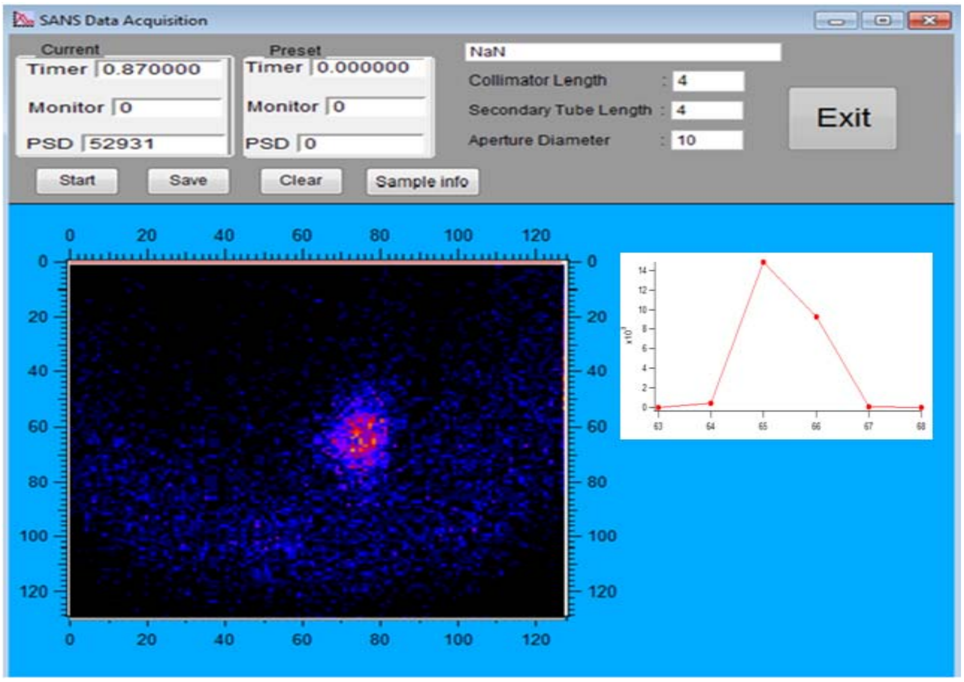


FIG. 6: New SANS Data Acquisition in WaveMetrics Igor in Windows XP operating system

During the *ibdev*, it will return a V_flag value for the Unit descriptor value and each device has different Unit descriptor value. The Unit descriptor (ud) was used as a unique identifier for the board or device descriptor returned by the *ibfind* function or the *ibdev* function. The timeout value is the approximate minimum length of time that I/O functions such as *ibrd, ibwrt*, and *ibcmd* can take before a timeout occurs. If it take after the timeout occurs, the program will receive a timeout error.

The TDC histogram and 2 units of ORTEC 994 need to be initialized by DAQ program before start counting and followed by data transfer from TDC histogram memory to the host computer. The data consist of 65536 memories, which were occupied by 128 x128 pixel data from the 2D PSD based on the calculation (128x128 pixel) x4 parameter (register byte D,E,B,C). At the end, the program will issue ORTEC stop commands to stop the devices by clicking ‘Stop’ button or reaching the preset value. It will take last readings of all devices after issuing stop commands. In Fig. 7, it shows the program flowchart for the new SANS data acquisition program.

ORTEC 994 DUAL COUNTER AND TIMER

The ORTEC 994 dual counter and timer (Fig. 8) can be used as two counters recording separate events (Counter A and B). In this SANS data acquisition system 2 units of ORTEC 994 is used as a timer and other one as a counter to display Neutron Monitor event and accumulated PSD event.

By adding the IEEE-488 option, the full power of CCNIM (Computer- Controlled NIM) can be obtained. These plug-in boards allow computer control of all functions normally selectable from the front panel, including start and stop count, readout, reset, setting the preset value, selecting the displayed counter, and selecting the desired time base. Computer readout with either of the two CCNIM options includes A and B counts, the preset value, and which counter is being displayed. The computer can disable all front-panel controls in the Remote mode. For this SANS data acquisition system, we chose the counter B (PSD event) to be displayed on the ORTEC 994. The IEEE- 488 option also reads the overflow status for both counters.

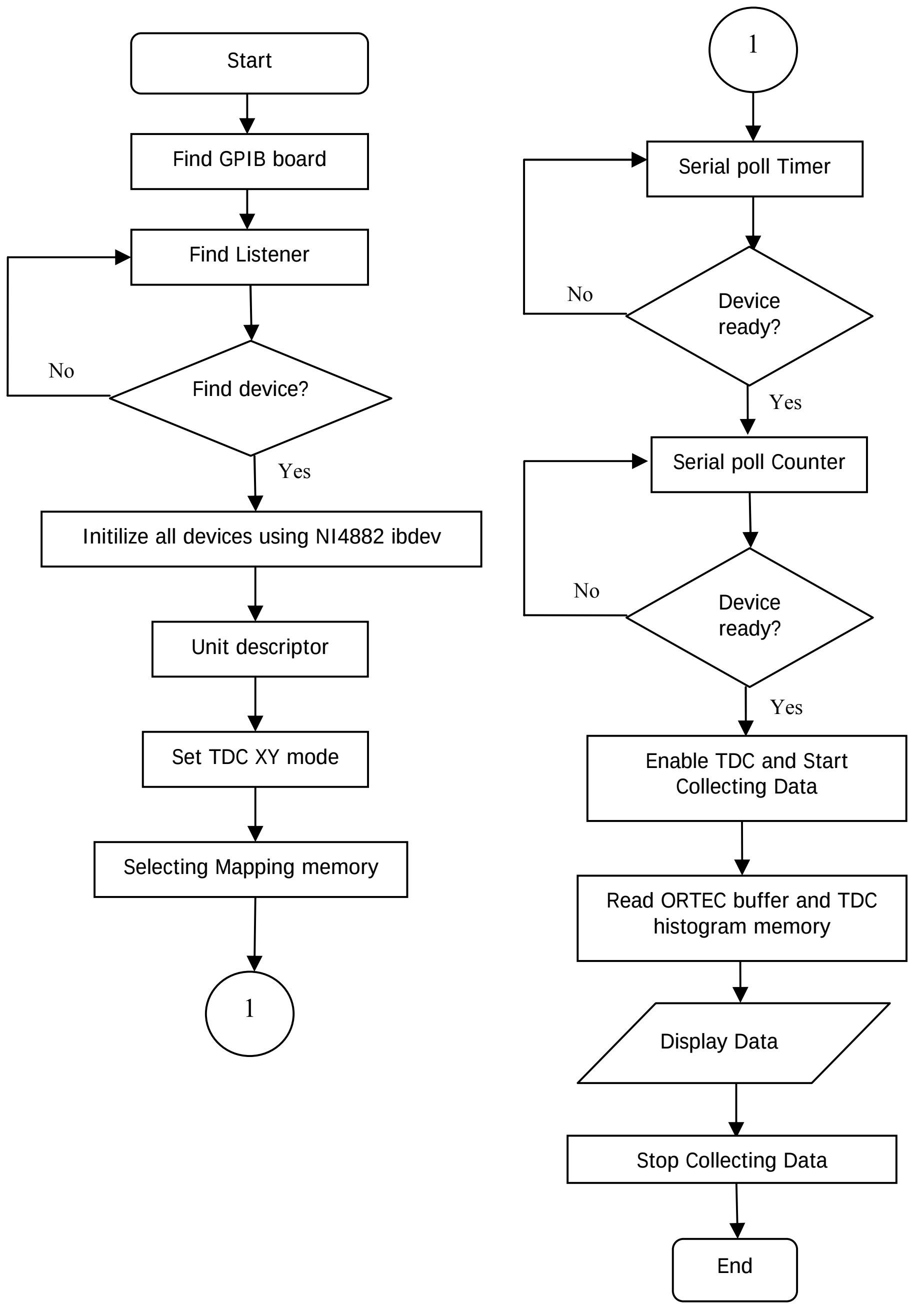


FIG. 7: New SANS data acquisition program flowchart

TABLE 2: EOS (End-of-String) configuration

Bit	**Configuration**	**High Byte**	**Low Byte**
A	Terminate read when EOS is detected.	0000:0100	EOS character
B	Set EOI with EOS on write function.	0000:1000	EOS character
C	Compare all 8 bits of EOS byte rather than low 7 bits (all read and write functions).	0001:0000	EOS character

Implementation of the IEEE-488 interface in the Model 994 is compatible with the Standard NIM (Nuclear Instrument Modules) Digital Bus. This both ORTEC 994 dual counter and timer is plug into the NIM 'bin' and received their power from a standard power supply attached to the rear of the bin (crate).

The ORTEC instrument need to serial polling each time before the ORTEC command is send to the instrument which is to make sure the instrument in ready condition because the controller (GPIB) must determine which device asserted the SRQ line and respond accordingly. The serial polling process is being put inside the ORTEC protocol (Fig. 9) and ORTECRead (Fig. 12). In response to the serial polling interface message, the NIM/488 Module shall return a status byte (U.S. NIM Committee, 1990). In this DAQ program the instrument will do serial polling until it receive serial buffer signal that equal to decimal 16 (hexadecimal 10) from the ORTEC instruments. If the system receives decimal 16 from the instruments during serial polling process, it shows the instrument is ready to received user data (ORTEC command) through *ibwrt* function.

FIG. 8: ORTEC 994 dual counter and time

In this DAQ program we need to set the ibeos before we send ORTEC command through ibwrt. The general format for the ibeos function in Igor is:

- NI4882 ibeos={ud, value}

Where the unit Descriptor (ud) is the unique identifier for the board or device (returned from the ibfind or ibdev function) and value is the EOS mode and character information. The value parameter is a hexadecimal value that consists of two bytes; a high byte and a low byte. The high byte is determined by what type of EOS configuration you want to use and the low byte is determined by which EOS character you wish to use. The EOS (End-of-string) configurations are based on Table 2.

The high byte can have the value of any combination of the three configurations mentioned in the table or it can have a value of zero for no EOS mode. The low byte has the ASCII value of the EOS character you plan to use, or a value of zero for no EOS character. These 2 bytes must then be packed into a hex value ibeos (ud, 0xAABB). AA corresponds to the high byte (configuration) in hex and BB corresponds to the low byte (EOS character) in hex. In our DAQ program, the AA corresponds is hex 04 because the High Byte is binary 100 (bin100=hex 04) and the BB corresponds is hex 0A (hex 0A=ASCII \n) because Low Byte is based on which EOS character we wish to use.

Because we set the low byte for ibeos is ASCII "\n" (new line feed), we need to include the ASCII character "\n" at the end of the ORTEC command when we send the command through ibwrt. For example: "CLEAR_ALL\n".

The process to start and to stop (Fig. 10) the ORTEC instruments was similar were each command that being issued to the instruments need to go through the ORTEC protocol. To start ORTEC 994, the following ORTEC command sequence is issued to the instrument (Fig.10) based on the list of ORTEC command in Table 3. In Fig. 11, the flowcart show the ORTECRead process which the ORTECRead process was used to read ORTEC 994 data from the DAQ counter and timer. The buffer data from the *ibrd* was convert from string to integer value.

TABLE 3: List of ORTEC commands for SANS DAQ program

ORTEC command	Purpose
INITIALIZE	Cause the 994 to restart or initialize
CLEAR_ALL	Clear counters, count preset, and event counter
SET_MODE_MINUTES	Set minutes time base as the input to the preset counter
SET_MODE_EXTERNAL	Selects the external input to the preset counter in the 994
SET_DISPLAY <VALUE>	Selects the counter whose contents will be displayed on the 994 front panel
ENABLE_REMOTE	Places the 994 under the control of a host computer
START	Causes the 994 to start a counting cycle
SHOW_COUNTS	Shows the contents of all counters of the 994
STOP	Stops the 994 from counting
ENABLE_LOCAL	Places the 994 under local control

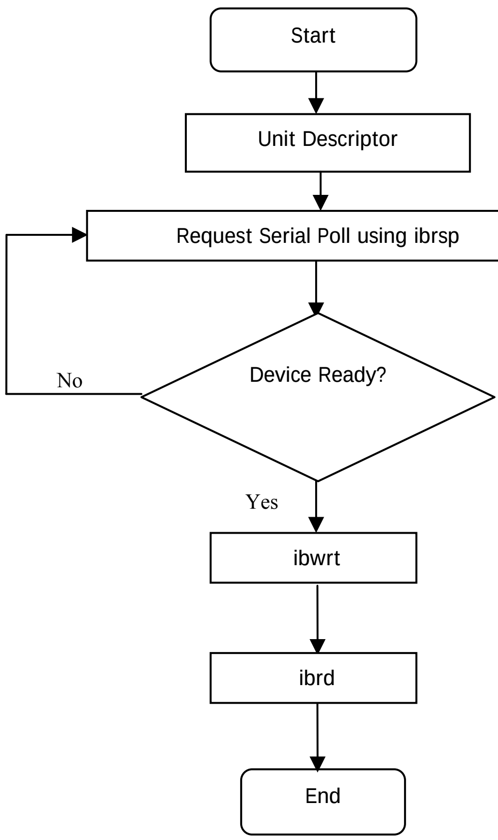


FIG. 9 ORTEC protocol flowchart

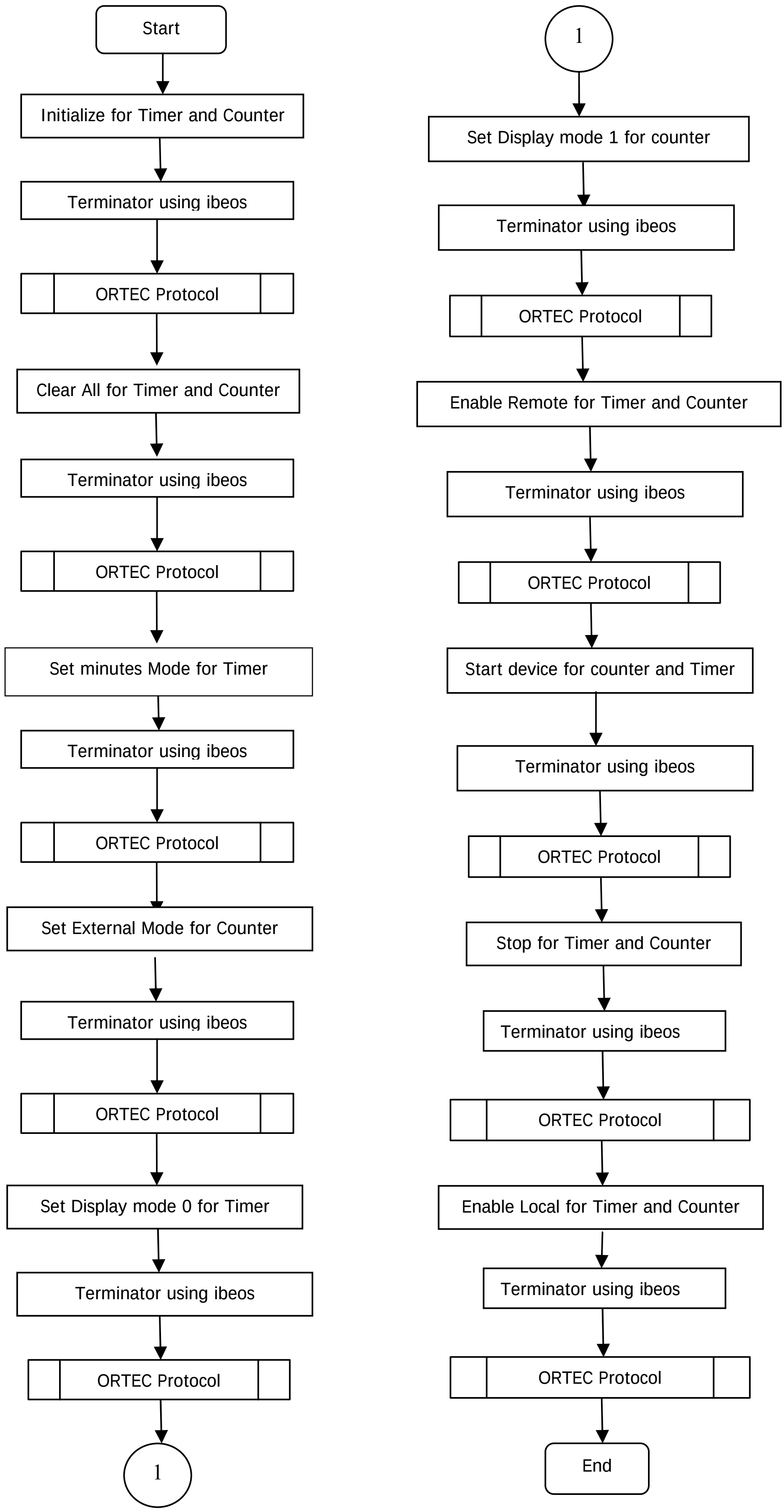


FIG. 10: ORTEC 994 command sequence to start and stop collecting data

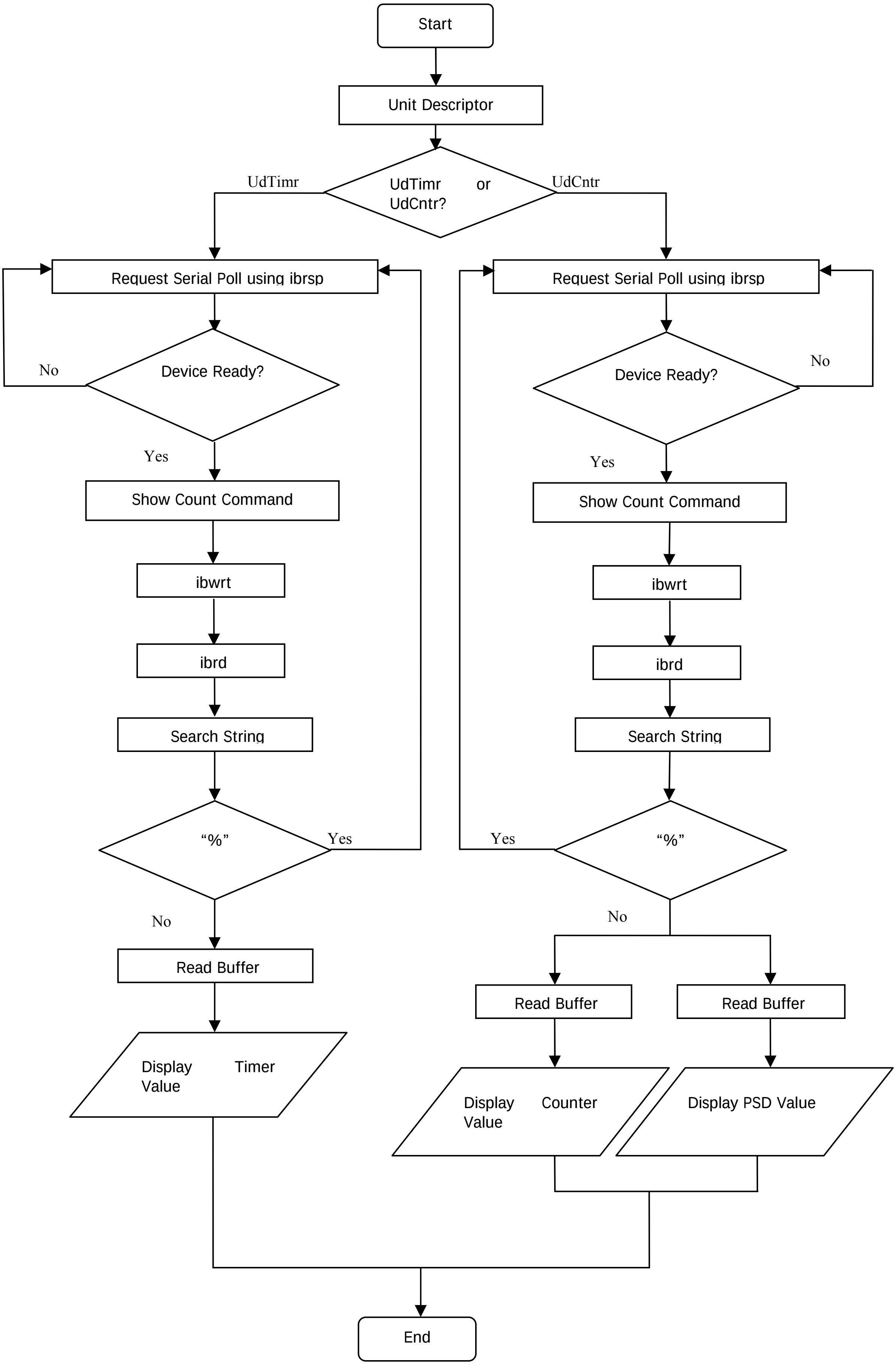


FIG. 11: ORTECRead process flowchart which being used for read the data buffer inside ORTEC 994 for data acquisition timer and counter instruments.

RISOE TIME-TO-DIGITAL CONVERTER (TDC) HISTOGRAM

The Risoe Time-to-Digital Converter (TDC) P2012a processes events from the Risoe 2D multiwire small angle neutron scattering detector. The TDC histogram has two LED indicator lamps. It is used as an indicator for:

- TDC Enable: Lights red when enable
- Accept Indicator: Flashes red for each event recorded

The TDC Histogram can communicate with the host computer by using GPIB bus. The primary address for TDC histogram is 10. This can be set by change the DIP switch setting. To do measurement by using TDC histogram, the following subroutine sequences (Fig. 12) need to be follow.

This sequence can be done by sending an ASCII character to the TDC using *ibwrt* function (Table 4). For example, if we want to Start/Enable the TDC, we need to send *ƒ* ASCII value to the instrument by using *ibwrt* function. Those following ASCII characters for those subroutines are given by manufacture. In Wavemetrics Igor, we cannot send an ASCII directly to the instrument. Thus, we need to send a hexadecimal value that equivalent with the ASCII value. That hexadecimal value will be converted to ASCII value/character using Igor syntax char2num and then send the converted ASCII to the TDC instrument.

If a valid code (ASCII value) is received by the histogram during the following TDC subroutine, an "Ack" (Acknowledge) byte is return to the host computer via *ibrd* function. The Ack byte is hex 06.

After sending receiving the Ack byte, we need to send 4 bytes from the host computer into the registers byte D, E, B and C by convert hexadecimal values into ASCII characters. Those 4 bytes value is set by the manufacture. After sending the 4 bytes, the host computer will receive back some ASCII value from the register byte (D, E, B and C) via the *ibrd* function by. For reading histogram memory process, a hexadecimal 05 was send to the instrument to do the 'Read Histogram Memory subroutine. Then, the 4 bytes that being sent to the register byte (D, E, B and C) is:

D=0xFF

E=0xFF

B=0

C=0

The D and E value based on 65535 (FFFF) memories which the total histogram memory will being read during reading memory process. To get any data during reading histogram memory process, we need to do the s*scanf* process to search an ASCII character in the *ibrd* buffer. It is because during reading the histogram memory, the *ibrd* give 13 lines of *ibrd* buffer in one *ibrd* process. So, we need to search which lines have an ASCII character. If we do not do the *sscanf*, Igor will read only the first line of *ibrd* which the first line only have 0 value.

To obtain the actual signal data in integer form, the ASCII character received from the register byte is converted to number that represents the ASCII character. Fig. 13 shows profile of the signal generated by the line position pulser generator. This profile indicates the positioning of the incoming a line of PSD data read by the TDC - it is an electronic data alignment. The higher data in the graph is at point 64 at which the beam center should be located providing all the modules in the detection system are correctly set.

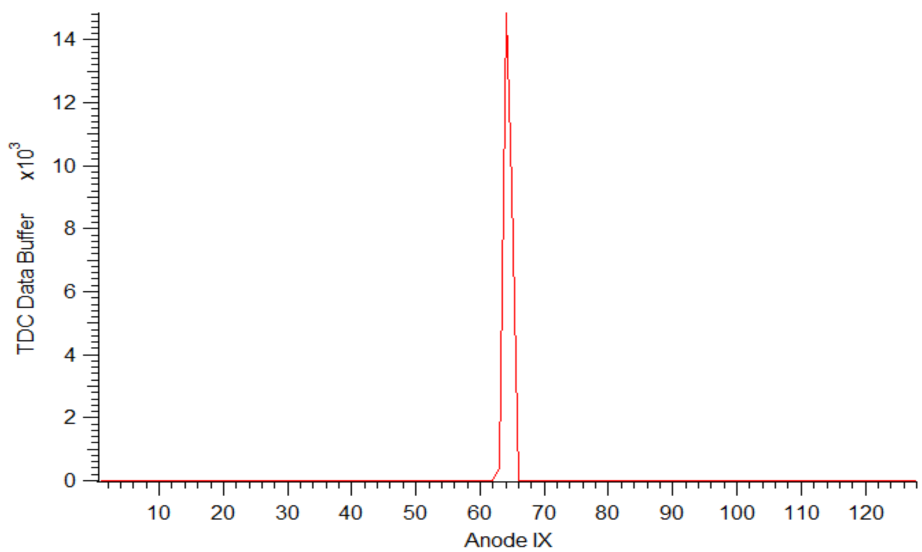


FIG. 13: Pulser data for beam alignment purpose.

TABLE 4: ASCII and hexadecimal value for each TDC subroutine

Subroutine	Hexadecimal Value	ASCII value
Select XY mode	0x80	€
Selecting Mapping Memory	0x86	†
Clear Histogram Memory	0x85	…
Start/Enable	0x83	ƒ
Read Histogram memory	0x05	|

SUMMARY

The new SANS data acquisition program by using WaveMetrics Igor Pro is convenient because it run in Windows OS. The development GPIB interface for this DAQ is based on standard interface such as the IEEE-488. Several upgrades on this program will be made in the future such as 3D display and combined this SANSDAQ and SANS data reduction into one Igor package program.

ACKNOWLEDGEMENTS

I would like to acknowledge Malaysian Nuclear Society for the financial support. Also like to acknowledge MR Hafizal Yazid and Mr Wilfred Sylvester Paulus from Malaysian Nuclear Agency and Bharoto from Badan Tenaga Atom Nasional (BATAN) for the technical support.

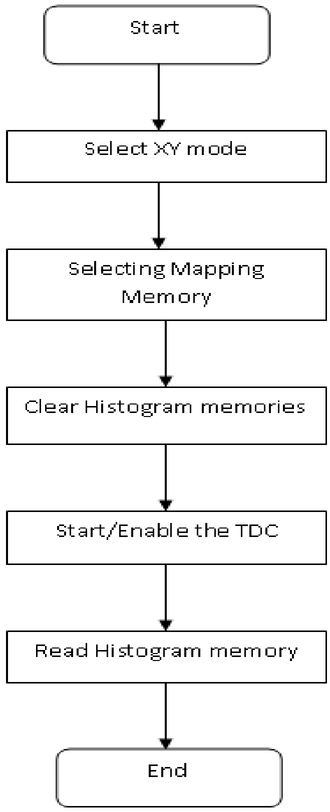


FIG. 12: TDC measurement sequence flowchart. Host computer will receive “Ack” byte for each subroutine process.

REFERENCES

Boualem Hammouda, (2009), Probing Nanoscale Structures- The SANS Toolbox, National Institute of Standards and Technology (NIST).

D. Jernigan and T. Dehne (1999), Programming with NI-488.2™ Software, National Instrument Application Note 002.

Eszter Rétfalvi, (2003), Small angle neutron scattering study of radiation damage in steel, PhD Thesis, Physic at Technical University of Budapest, Faculty of Natural Sciences.

Felix Grant, (2008), Igor Pro 6 Review, Scientific Computing World, http://www.scientific-computing.com/products/review_details.php?review_id=35.

M. A. M. Sufi. Y. Abdullah, J. Hamid R. Kassim, S. Radiman, M. Deraman and A.G. Ramli, (1996), Small-Angle Neutron Scattering Instrument at MINT, Asian Conference on X-Rays and Related Techniques in Research and Industry.

Steven R. Kline (2006), Reduction and analysis of SANS and USANS data using IGOR Pro, Journal of Applied Crystallography.

Tan Chier Fang, (2003), Determination of Silica Aerogels Nanostructure Characteristic By Using Small Angele Neutron Scattering Technique, Master Thesis, Faculty of Science Universiti Teknologi Malaysia.

U.S. NIM Committee (1990), Standard NIM instrumentation system, U.S. Department of Energy, Page B-7.

WaveMetrics (2008), Igor Pro Manual Version 6, Chapter IV-10 Page IV-208.