NEW CONTROL SOFTWARE FOR CERBERUS 3D NANOINDENTATION SYSTEM

by

Bhakt Vatsal Trivedi
(2010041)

SUPERVISORS:

Mr. Saket Sourav
Research Engineer
IIITDM - Jabalpur

Dr. Graham L. W. Cross
PI, CRANN Nanotechnology Institute
Trinity College Dublin

Computer Science and Engineering Discipline

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DESIGN AND MANUFACTURING, JABALPUR
(Mid Term Report - 2013)

1. Introduction

Evaluation of mechanical properties of thin films has become important with the extending application of functional thin films, such as the protection film in the magnetic recording media and heads, the interconnects in the integrated circuit, the surface modified layer in the ion implanted materials, etc. Nanoindentation is a powerful and simple means for evaluating the mechanical properties of thin films. The data of nanoindentation measurement is usually expressed by a curve of applied load fa versus displacement (penetration depth) h.

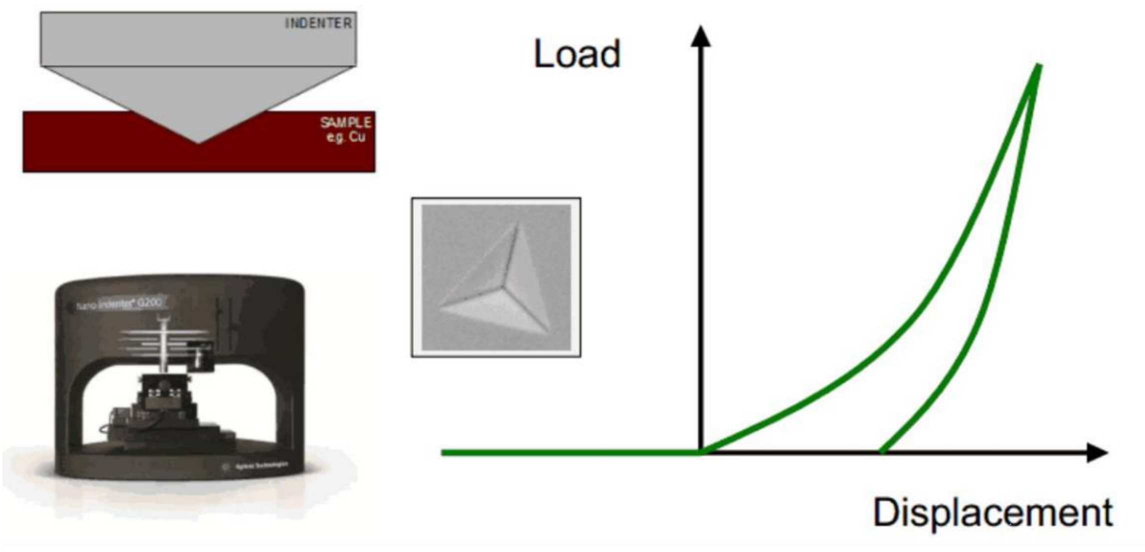


Figure 1: Indentation Technique and plot for the result.

1.1 Cerberus Nanoindentation System

Recently, several nanoindentation instruments were commercialized. Cerberus is one such instrument. Cerberus is controlled by an electronic hardware system that consists of an ISA-bus backplane which holds a number of ISA daughter-boards that control various functions of the indenter. These include interface cards to three external lock-in amplifier (LIA) boxes that are used to run the dynamic mode of the instrument in each of the x, y, and z axes. The entire system is managed by a Single Board Computer (SCB) ISA board, which is currently an old Intel 80286 based PC running Microsoft Windows 3.1 (i.e. DOS).

The **Cerberus controller software** is a large program written in the HT Basic interpreted language. It consists of low-level subroutines that communicate with the ISA boards and LIAs, intermediate level algorithms that make Cerberus perform nanoindentation tests by making calls to the low-level subroutines, and high level GUI elements that interact with the user.

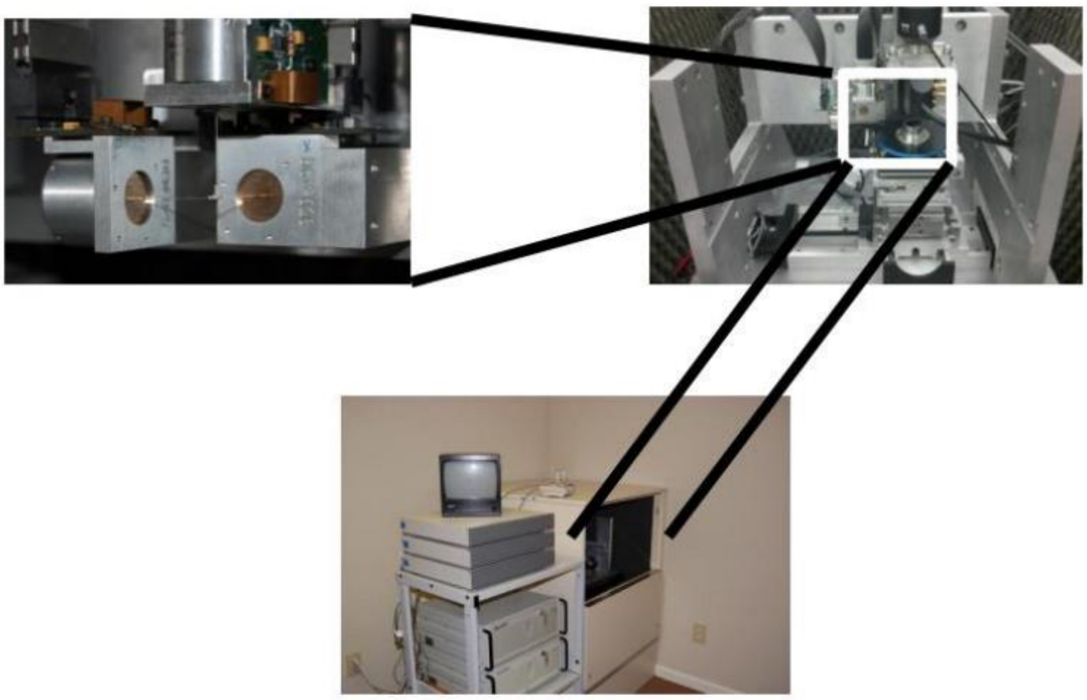


Figure 1.2 : The Cerberus System

1.2 Aim of the Project

The purpose of this project is to upgrade the existing control software system that runs the “Cerberus”. The goals of the project is as follows: **“To replace the HT Basic GUI elements with a modern GUI environment running under Windows 7 and written in Python suitable for scientific programming and real-time control, data storage and data display and review.”**

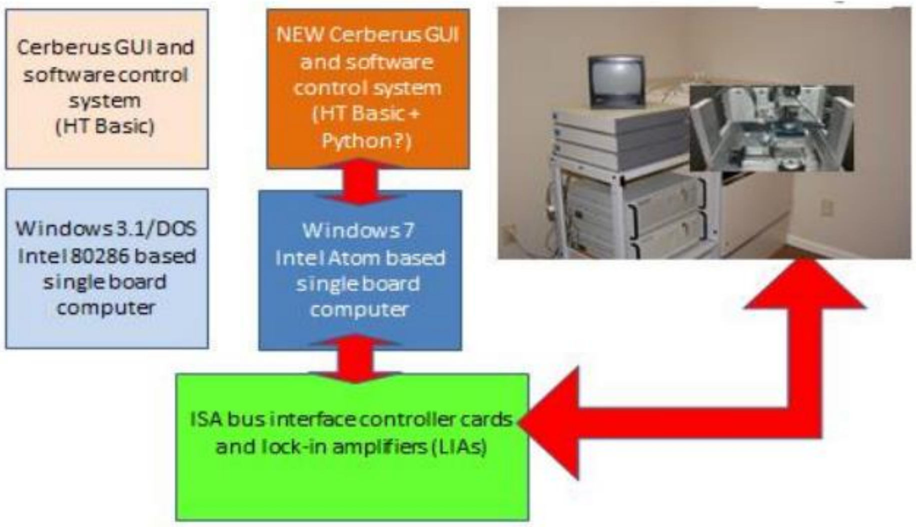
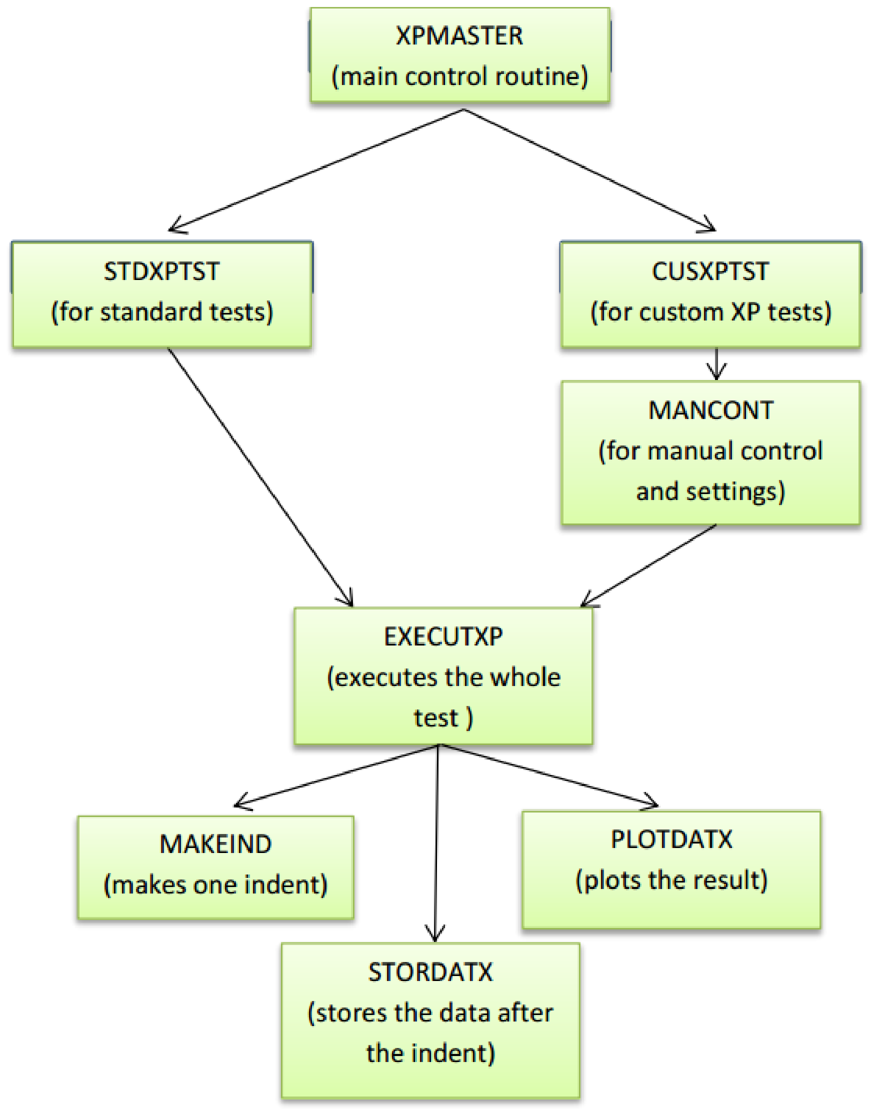

Cerberus GUI and software control system (HT Basic)
NEW Cerberus GUI and software control system (HT Basic + Python?)
Windows 3.1/DOS Intel 80286 based single board computer
Windows 7 Intel Atom based single board computer
ISA bus interface controller cards and lock-in amplifiers (LIAs)

Figure 2: The whole idea of the project.

2. Analysis of Existing code

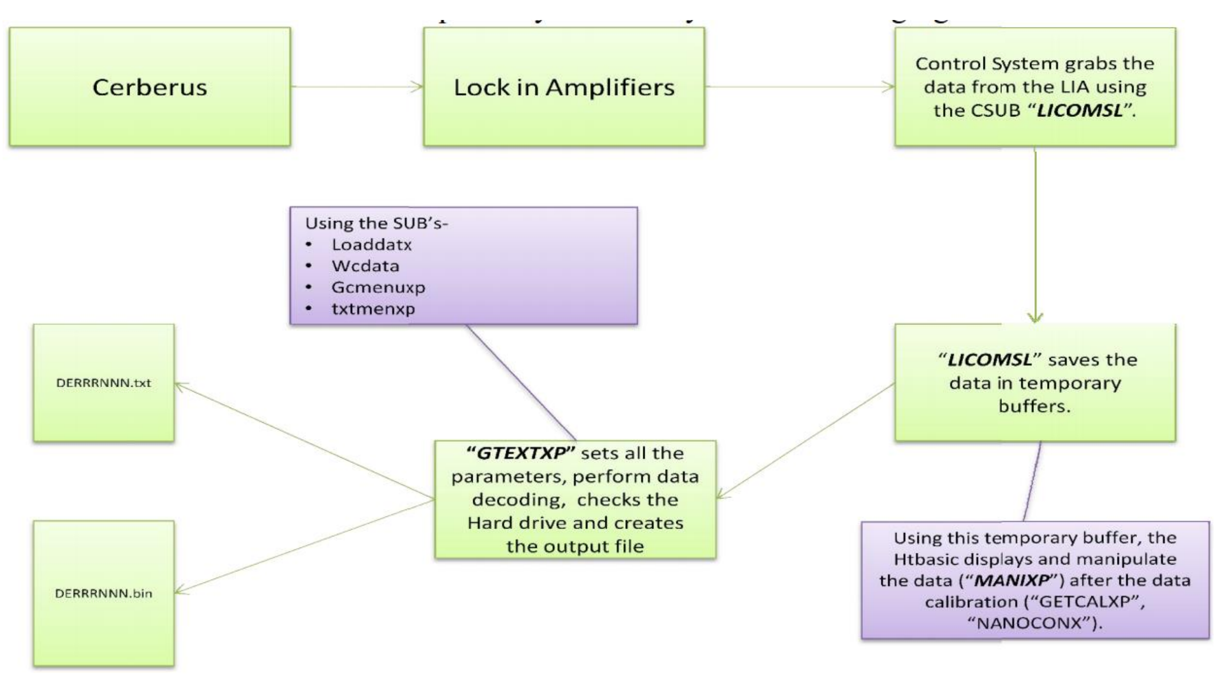
2.1. Control Flow

In order to understand the system we first need to understand the control flow of the control system. The following flowchart explains the flow of the control system for the standard test and custom test options:



2.2. Data Flow

To understand the system correctly we first need to understand the data flow of data in the system. The HtBasic Code is largely a sequential code with subroutines doing various tasks. The major task was to figure out the data flow from the Cerberus system and how this data is handled by the htbasic control software. This is depicted systematically in the following figure:


Cerberus
Lock in Amplifiers
Control System grabs the data from the LIA using the CSUB "LICOMSL".
"LICOMSL" saves the data in temporary buffers.
Using this temporary buffer, the Htbasic displays and manipulate the data ("MANIXP") after the data calibration ("GETCALXP", "NANOCONX").
"GTEXTXP" sets all the parameters, perform data decoding, checks the Hard drive and creates the output file
Using the SUB's-
Loaddatx
Wcdata
Gcmenuxp
txtmenxp
DERRRNNN.txt
DERRRNNN.bin

So the first thing that is done after getting the data from the LIA is the calibration. We now look into this deeply.

2.3 Calibration Constants

The data coming from the system is raw voltage value.The Cerberus system uses various constants to convert the raw voltages into meaningful data. After deep analysis of code and standard nanoindentaion references the following constants were figured out. The Calibration are shown in Appendix I

These values are the constants involved in the experiment. But Some Calibrations value depend on the data. These are referred as Calibration Values and are shown in the following table.

2.4 Calibration Values:

Z-Axis	
Displacement Range	[(ZDisp[1]/10000)-1] * Rounded to zero
Displacement Offset	((ZDisp[1]-((Z-Displacement Range+1)*10000))-5)/10 * Rounded to zero
DC Displacement Calibration	Z-DC- Displacement Constant Term + Z-DC- Displacement Linear Term * [Z-Displacement Offset-128] + Z- DC- Displacement Squared Term * [Z-Displacement Offset-128]^2
AC Displacement Calibration	Z-AC- Displacement Constant Term + Z-AC- Displacement Linear Term * [Displacement Offset-128] + Z- AC- Displacement Squared Term * [Z-Displacement Offset-128]^2
Load Range	[Zload[1]/10]-1 * Rounded to zero
DC Load Calibration	Z-DC- Load Constant Term + Z-DC- Load Linear Term * [Z-Displacement Offset-128] + Z-DC- Load Squared Term * [Z-Displacement Offset-128]^2
AC Load Calibration	Z-AC- Load Constant Term + Z-AC- Load Linear Term * [Z-Displacement Offset-128] + Z-AC- Load Squared Term * [Z-Displacement Offset-128]^2
Modcal Factor	Z-Axis Modcal Factor
X-Axis	

Displacement Range	[(XDisp[1]/10000)-1] * Rounded to zero
Displacement Offset	((XDisp[1]-((X-Displacement Range+1)*10000))-5)/10 * Rounded to zero
DC Displacement Calibration	X-DC- Displacement Constant Term + X-DC- Displacement Linear Term * [X-Displacement Offset-128] + X- DC- Displacement Squared Term * [X-Displacement Offset-128]^2
AC Displacement Calibration	X-AC- Displacement Constant Term + X-AC- Displacement Linear Term * [Displacement Offset-128] +X- AC- Displacement Squared Term * [X-Displacement Offset-128]^2
Load Range	[Xload[1]/10]-1 * Rounded to zero
DC Load Calibration	X-DC- Load Constant Term + X-DC- Load Linear Term * [X-Displacement Offset-128] + X-DC- Load Squared Term * [X-Displacement Offset-128]^2
AC Load Calibration	X-AC- Load Constant Term + X-AC- Load Linear Term * [X-Displacement Offset-128] + X-AC- Load Squared Term * [X-Displacement Offset-128]^2
Modcal Factor	X-Axis Modcal Factor
Y-Axis	
Displacement Range	[(YDisp[1]/10000)-1] * Rounded to zero
Displacement Offset	((YDisp[1]-((Y-Displacement Range+1)*10000))-5)/10 * Rounded to zero
DC Displacement Calibration	Y-DC- Displacement Constant Term + Y-DC- Displacement Linear Term * [Y-Displacement Offset-128] + Y- DC- Displacement Squared Term * [Y-Displacement Offset-128]^2
AC Displacement Calibration	Y-AC- Displacement Constant Term + Y-AC- Displacement Linear Term * [Displacement Offset-128] +Y- AC- Displacement Squared Term * [Y-Displacement Offset-128]^2
Load Range	[Yload[1]/10]-1 * Rounded to zero
DC Load Calibration	Y-DC- Load Constant Term + Y-DC- Load Linear Term * [Y-Displacement Offset-128] + Y-DC- Load Squared Term * [Y-Displacement Offset-128]^2
AC Load Calibration	Y-AC- Load Constant Term + Y-AC- Load Linear Term * [Y-Displacement Offset-128] + Y-AC- Load Squared Term * [Y-Displacement Offset-128]^2
Modcal Factor	Y-Axis Modcal Factor

After the constants and values are figured out the next job is to manipulate the data to perform calculations

2.5 Data Calculations:

As mentioned earlier, the data in the files coming from the Cerberus are raw Voltages. So after the calibrations the final data values are calculated which is final objective. The values that are calculated are as following:

Quantity	Description
Time (s)[i]	Data.time[i] – data.time[2] i from 2 to max
Z-Disp R3 (V)[i]	Data.ZDisp[i] – 5 –Z2*10 – (Z1+1)*10000
Z-Disp R3 Zeroed (V)[i]	Z-Disp R3 [i] - Z-Disp R3 [1]
Z-Disp Total (nm)[i]	Z-Disp R3 Zeroed[i] * Z3 / 2^(Z1)
X-Disp R3 (V)[i]	Data.XDisp[i] – 5 – X2 *10 – (X1+1)*10000
X-Disp R3 Zeroed (V)[i]	X-Disp R3[i] - X-Disp R3[1]
X-Disp(nm)[i]	X-Disp R3 Zeroed[i]*X3/2^(X1)
Y-Disp R3 (V)[i]	Data.YDisp[i] – 5 –Y2*10 – (Y1+1)*10000
Y-Disp R3 Zeroed (V)[i]	Y-Disp R3[i] - Y-Disp R3[1]
Y-Disp(nm)[i]	Y-Disp R3 Zeroed[i]*Y3/2^(Y1)
Z-Disp Zeroed(nm)[i]	Z-Disp Total[i] – zero_disp_z

x-Disp Zeroed (nm) [i]	x-Disp Total[i] – zero_disp_x
y-Disp Zeroed (nm) [i]	y-Disp Total[i] – zero_disp_y
Time Zeroed (s) [i]	Time[i] – zero_time
Z-Amplitude[i]	Data.ZAmp[i]*Z4/2^(Z1)
Z-Force (µN)[i]	[1] -> data.ZExcite[1]*C23*C26 [i] -> data.ZExcite[i]*Z8*Z7
Z-Load (V)[i]	Data.Zload[i]- (Z5+1)*10 -5
Z-Load Zeroed (V)[i]	-(ZLoad[i]-zero_load)
Z-Load (mN)[i]	Z-Load Zeroed [i]*Z6/1000
Z-Load Corrected for Kls (mN)[i]	Z-Load (mN)[i] – SLOPE(load_for_kls, disp_for_kls)* Z-Disp Zeroed (nm)[i]
Raw Z Phase (Deg)[i]	Data.ZPha[i]*1
Corrected Z Phase (Deg)[i]	Raw Z Phase (Deg)[i]-(-0.0844*125-0.0596)
COS Z Phase (_)[i]	COS(Corrected Z Phase (Deg)[i]*pi/180)
F/X CosPHI (N/m) [i]	Z-Force (µN)[i]/ Z-Amplitude[i]*COS(Corrected Z Phase (Deg)[i]*PI()/180)*1000
F/X CosPHI-{k-mw^2} (N/m) [i]	F/X CosPHI (N/m) [i]-C7Z-C27Z*2*PI()*125*2*PI()*125
Z-Stiffness (N/m)[i]	1/(1/ F/X CosPHI-{k-mw^2} (N/m) [i]-1/C33Z)
Z-P/S^2 (m^2/N)[i]	Z-Load Corrected for Kls (mN)[i]/ Z-Stiffness (N/m)[i]/ Z-Stiffness (N/m)[i]/1000
X-Amplitude (nm)[i]	Data. X-Amp (V)[i]*X4/2^X1
Y-Amplitude (nm)[i]	Data. X-Amp (V)[i]*Y4/2^Y1
X-Force (µN)[i]	XExc(v)[i]*C23X*C26X
Y-Force (µN)[i]	YExc(v)[i]*C23Y*C26Y
Z-F/X (N/m)[i]	Z-Force (µN)[i]/ Z-Amplitude[i]*1000
X-F/X (N/m)[i]	X-Force (µN)[i]/ X-Amplitude (nm)[i]*1000
Y-F/X (N/m)[i]	Y-Force (µN)[i]/Y-Amplitude (nm)[i]*1000
X-Harmonic F for Const X/Quasi-static Z-Force (_)[i]	X-Force (µN)[i]/ Z-Load Corrected for Kls (mN)[i]/1000
Y-Harmonic F for Const X/Quasi-static Z-Force (_)[i]	Y-Force (µN)[i]/ Z-Load Corrected for Kls (mN)[i]/1000
Raw X Phase (Deg)[i]	Data. X-Pha (Dg)[i]*(1)
Corrected X Phase (Deg)[i]	Raw X Phase (Deg)[i]-(-0.0844*125-0.0596)
COS X Phase (_)[i]	COS(Corrected X Phase (Deg)[i]*PI()/180)

Raw Y Phase (Deg)[i]	Data. Y-Pha (Dg)[i]*(1)
Corrected Y Phase (Deg)[i]	Raw Y Phase (Deg)[i]-(-0.0844*125-0.0596)
COS Y Phase (_)[i]	COS(Corrected Y Phase (Deg)[i]*PI()/180)
X-F/X CosPHI (N/m)[i]	X-F/X (N/m)[i]* COS X Phase (_)[i]
X-F/X CosPHI-{k-mw^2} (N/m)[i]	X-F/X CosPHI (N/m)[i]-(C7X-C27X*2*PI()*125*2*PI()*125)
X Stiffness (N/m)[i]	1/(1/ X-F/X CosPHI-{k-mw^2} (N/m)[i]-1/C33X)
Y-F/x CosPHI (N/m)[i]	Y-F/X (N/m)[i]* COS Y Phase (_)[i]
Y-F/x CosPHI-{k-mw^2} (N/m)[i]	Y-F/x CosPHI (N/m)[i]-(C7Y-C27Y*2*PI()*125*2*PI()*125)
Y Stiffness (N/m)[i]	1/(1/ Y-F/x CosPHI-{k-mw^2} (N/m)[i]-1/C33Y)
X P/S^2 (m^2/N)[i]	Z-Load Corrected for Kls (mN)[i]/ X Stiffness (N/m)[i]/ X Stiffness (N/m)[i]/1000
Y P/S^2 (m^2/N)[i]	Z-Load Corrected for Kls (mN)[i]/ Y Stiffness (N/m)[i]/ Y Stiffness (N/m)[i]/1000
X-SinPHI (_)[i]	SIN(Corrected X Phase (Deg)[i]*PI()/180)
X-F/X SinPHI (N/m)[i]	X-F/X (N/m)[i]* X-SinPHI (_)[i]
X-F/X SinPHI-ciw (N/m)[i]	X-F/X SinPHI (N/m)[i]-C28X*125*2*PI()
Cw/S X-Axis (_)[i]	X-F/X SinPHI-ciw (N/m)[i]/ X Stiffness (N/m)[i]
Y-SinPHI (_)[i]	SIN(Corrected Y Phase (Deg)[i]*PI()/180)
Y-F/X SinPHI (N/m)[i]	Y-F/X (N/m)[i]* Y-SinPHI (_)[i]
Y-F/X SinPHI-ciw (N/m)[i]	Y-F/X SinPHI (N/m)[i]-C28Y*125*2*PI()
Cw/S Y-Axis (_)[i]	Y-F/X SinPHI-ciw (N/m)[i]/ Y Stiffness (N/m)[i]

So at this point we have figured out how to do the final calculation from the text file but to make this system real time we need to grab the data from the binary files.

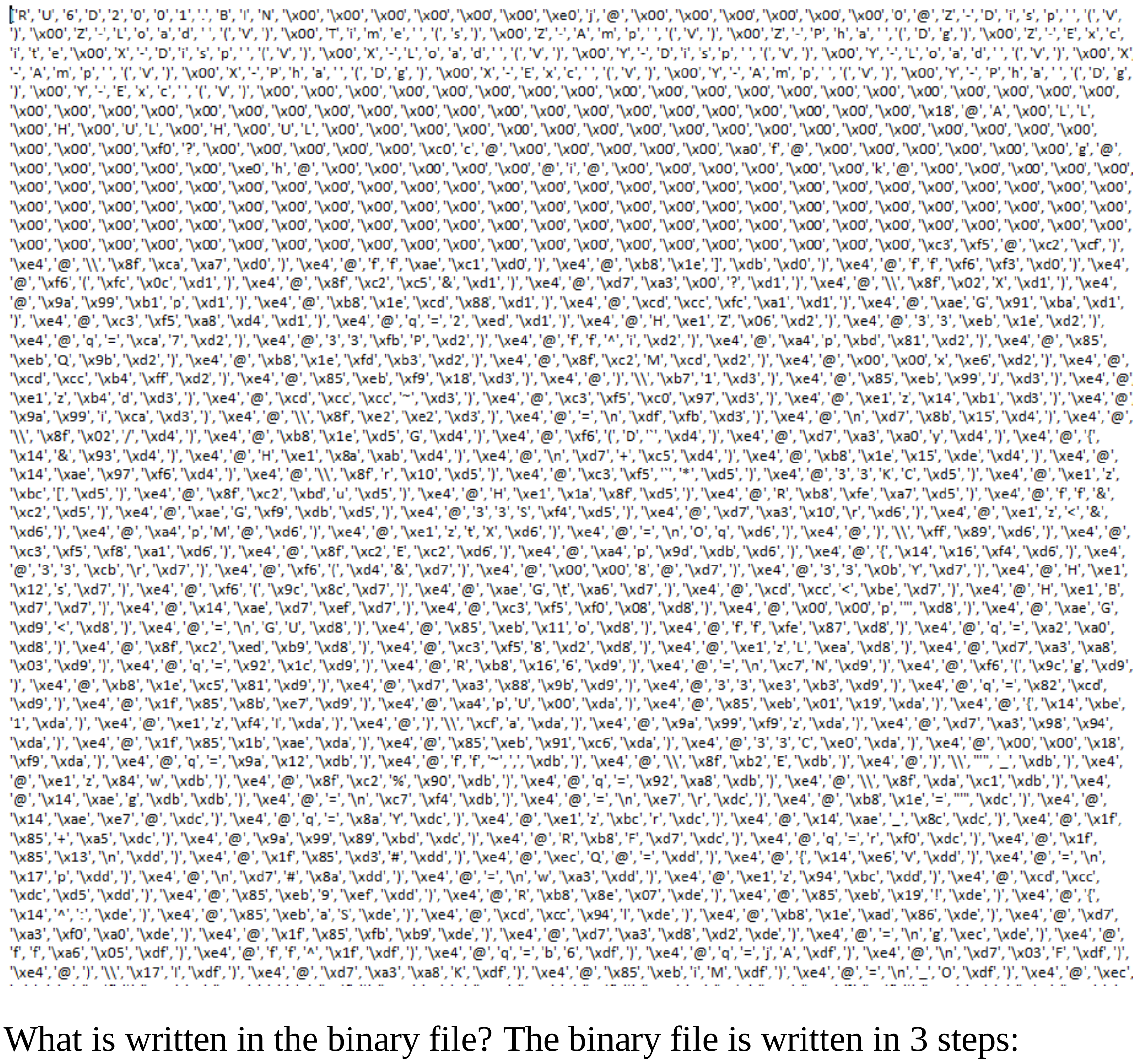
3. Binary Data:

3.1 Contents of the binary file.

In order to get the binary data, I wrote a python script. After observing the existing code, it became clear that data was written in a sequential manner.

1. The file name (RU6D2001.bin , in this case).
2. Channels in the data stream.
3. The actual readings.(logged_data(*))
4. Command sequence.

Python interpreter has methods to directly access the binary file and using those methods file is explored. The following figure shows the actual content of the binary file:



```
['R', 'U', '6', 'D', '2', '0', '0', '1', '.', 'B', 'I', 'N', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\xe0', 'j', '@', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '0', '@', 'Z', '-', 'D', 'i', 's', 'p', ' ', '(', 'V',
')', '\x00', 'Z', '-', 'L', 'o', 'a', 'd', ' ', '(', 'V', ')', '\x00', 'T', 'i', 'm', 'e', ' ', '(', 's', ')', '\x00', 'Z', '-', 'A', 'm', 'p', ' ', '(', 'V', ')', '\x00', 'Z', '-', 'P', 'h', 'a', ' ', '(', 'D', 'g', ')', '\x00', 'Z', '-', 'E', 'x', 'c',
'i', 't', 'e', '\x00', 'X', '-', 'D', 'i', 's', 'p', ' ', '(', 'V', ')', '\x00', 'X', '-', 'L', 'o', 'a', 'd', ' ', '(', 'V', ')', '\x00', 'Y', '-', 'D', 'i', 's', 'p', ' ', '(', 'V', ')', '\x00', 'Y', '-', 'L', 'o', 'a', 'd', ' ', '(', 'V', ')', '\x00', 'X
'-', 'A', 'm', 'p', ' ', '(', 'V', ')', '\x00', 'X', '-', 'P', 'h', 'a', ' ', '(', 'D', 'g', ')', '\x00', 'X', '-', 'E', 'x', 'c', ' ', '(', 'V', ')', '\x00', 'Y', '-', 'A', 'm', 'p', ' ', '(', 'V', ')', '\x00', 'Y', '-', 'P', 'h', 'a', ' ', '(', 'D', 'g',
')', '\x00', 'Y', '-', 'E', 'x', 'c', ' ', '(', 'V', ')', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00',
'\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x18', '@', 'A', '\x00', 'L', 'L',
'\x00', 'H', '\x00', 'U', 'L', '\x00', 'H', '\x00', 'U', 'L', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00',
'\x00', '\x00', '\x00', '\xf0', '?', '\x00', '\x00', '\x00', '\x00', '\x00', '\xc0', 'c', '@', '\x00', '\x00', '\x00', '\x00', '\x00', '\xa0', 'f', '@', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', 'g', '@',
'\x00', '\x00', '\x00', '\x00', '\x00', '\xe0', 'h', '@', '\x00', '\x00', '\x00', '\x00', '\x00', '@', 'i', '@', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', 'k', '@', '\x00', '\x00', '\x00', '\x00', '\x00',
'\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00',
'\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00',
'\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00',
'\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\xc3', '\xf5', '@', '\xc2', '\xcf', ')',
'\xe4', '@', '\\', '\x8f', '\xca', '\xa7', '\xd0', ')', '\xe4', '@', 'f', 'f', '\xae', '\xc1', '\xd0', ')', '\xe4', '@', '\xb8', '\x1e', ']', '\xdb', '\xd0', ')', '\xe4', '@', 'f', 'f', '\xf6', '\xf3', '\xd0', ')', '\xe4',
'@', '\xf6', '(', '\xfc', '\x0c', '\xd1', ')', '\xe4', '@', '\x8f', '\xc2', '\xc5', '&', '\xd1', ')', '\xe4', '@', '\xd7', '\xa3', '\x00', '?', '\xd1', ')', '\xe4', '@', '\\', '\x8f', '\x02', 'X', '\xd1', ')', '\xe4',
'@', '\x9a', '\x99', '\xb1', 'p', '\xd1', ')', '\xe4', '@', '\xb8', '\x1e', '\xcd', '\x88', '\xd1', ')', '\xe4', '@', '\xcd', '\xcc', '\xfc', '\xa1', '\xd1', ')', '\xe4', '@', '\xae', 'G', '\x91', '\xba', '\xd1',
')', '\xe4', '@', '\xc3', '\xf5', '\xa8', '\xd4', '\xd1', ')', '\xe4', '@', 'q', '=', '2', '\xed', '\xd1', ')', '\xe4', '@', 'H', '\xe1', 'Z', '\x06', '\xd2', ')', '\xe4', '@', '3', '3', '\xeb', '\x1e', '\xd2', ')',
'\xe4', '@', 'q', '=', '\xca', '7', '\xd2', ')', '\xe4', '@', '3', '3', '\xfb', 'P', '\xd2', ')', '\xe4', '@', 'f', 'f', '^', 'i', '\xd2', ')', '\xe4', '@', '\xa4', 'p', '\xbd', '\x81', '\xd2', ')', '\xe4', '@', '\x85',
'\xeb', 'Q', '\x9b', '\xd2', ')', '\xe4', '@', '\xb8', '\x1e', '\xfd', '\xb3', '\xd2', ')', '\xe4', '@', '\x8f', '\xc2', 'M', '\xcd', '\xd2', ')', '\xe4', '@', '\x00', '\x00', 'x', '\xe6', '\xd2', ')', '\xe4', '@',
'\xcd', '\xcc', '\xb4', '\xff', '\xd2', ')', '\xe4', '@', '\x85', '\xeb', '\xf9', '\x18', '\xd3', ')', '\xe4', '@', ')', '\\', '\xb7', '1', '\xd3', ')', '\xe4', '@', '\x85', '\xeb', '\x99', 'J', '\xd3', ')', '\xe4', '@
'\xe1', 'z', '\xb4', 'd', '\xd3', ')', '\xe4', '@', '\xcd', '\xcc', '\xcc', '~', '\xd3', ')', '\xe4', '@', '\xc3', '\xf5', '\xc0', '\x97', '\xd3', ')', '\xe4', '@', '\xe1', 'z', '\x14', '\xb1', '\xd3', ')', '\xe4', '@
'\x9a', '\x99', 'i', '\xca', '\xd3', ')', '\xe4', '@', '\\', '\x8f', '\xe2', '\xe2', '\xd3', ')', '\xe4', '@', '=', '\n', '\xdf', '\xfb', '\xd3', ')', '\xe4', '@', '\n', '\xd7', '\x8b', '\x15', '\xd4', ')', '\xe4', '@',
'\\', '\x8f', '\x02', '/', '\xd4', ')', '\xe4', '@', '\xb8', '\x1e', '\xd5', 'G', '\xd4', ')', '\xe4', '@', '\xf6', '(', 'D', '`', '\xd4', ')', '\xe4', '@', '\xd7', '\xa3', '\xa0', 'y', '\xd4', ')', '\xe4', '@', '{',
'\x14', '&', '\x93', '\xd4', ')', '\xe4', '@', 'H', '\xe1', '\x8a', '\xab', '\xd4', ')', '\xe4', '@', '\n', '\xd7', '+', '\xc5', '\xd4', ')', '\xe4', '@', '\xb8', '\x1e', '\x15', '\xde', '\xd4', ')', '\xe4', '@',
'\x14', '\xae', '\x97', '\xf6', '\xd4', ')', '\xe4', '@', '\\', '\x8f', 'r', '\x10', '\xd5', ')', '\xe4', '@', '\xc3', '\xf5', '`', '*', '\xd5', ')', '\xe4', '@', '3', '3', 'K', 'C', '\xd5', ')', '\xe4', '@', '\xe1', 'z',
'\xbc', '[', '\xd5', ')', '\xe4', '@', '\x8f', '\xc2', '\xbd', 'u', '\xd5', ')', '\xe4', '@', 'H', '\xe1', '\x1a', '\x8f', '\xd5', ')', '\xe4', '@', 'R', '\xb8', '\xfe', '\xa7', '\xd5', ')', '\xe4', '@', 'f', 'f', '&',
'\xc2', '\xd5', ')', '\xe4', '@', '\xae', 'G', '\xf9', '\xdb', '\xd5', ')', '\xe4', '@', '3', '3', 'S', '\xf4', '\xd5', ')', '\xe4', '@', '\xd7', '\xa3', '\x10', '\r', '\xd6', ')', '\xe4', '@', '\xe1', 'z', '<', '&',
'\xd6', ')', '\xe4', '@', '\xa4', 'p', 'M', '@', '\xd6', ')', '\xe4', '@', '\xe1', 'z', 't', 'X', '\xd6', ')', '\xe4', '@', '=', '\n', 'O', 'q', '\xd6', ')', '\xe4', '@', ')', '\\', '\xff', '\x89', '\xd6', ')', '\xe4', '@',
'\xc3', '\xf5', '\xf8', '\xa1', '\xd6', ')', '\xe4', '@', '\x8f', '\xc2', 'E', '\xc2', '\xd6', ')', '\xe4', '@', '\xa4', 'p', '\x9d', '\xdb', '\xd6', ')', '\xe4', '@', '{', '\x14', '\x16', '\xf4', '\xd6', ')', '\xe4',
'@', '3', '3', '\xcb', '\r', '\xd7', ')', '\xe4', '@', '\xf6', '(', '\xd4', '&', '\xd7', ')', '\xe4', '@', '\x00', '\x00', '8', '@', '\xd7', ')', '\xe4', '@', '3', '3', '\x0b', 'Y', '\xd7', ')', '\xe4', '@', 'H', '\xe1',
'\x12', 's', '\xd7', ')', '\xe4', '@', '\xf6', '(', '\x9c', '\x8c', '\xd7', ')', '\xe4', '@', '\xae', 'G', '\t', '\xa6', '\xd7', ')', '\xe4', '@', '\xcd', '\xcc', '<', '\xbe', '\xd7', ')', '\xe4', '@', 'H', '\xe1', 'B',
'\xd7', '\xd7', ')', '\xe4', '@', '\x14', '\xae', '\xd7', '\xef', '\xd7', ')', '\xe4', '@', '\xc3', '\xf5', '\xf0', '\x08', '\xd8', ')', '\xe4', '@', '\x00', '\x00', 'p', '"', '\xd8', ')', '\xe4', '@', '\xae', 'G',
'\xd9', '<', '\xd8', ')', '\xe4', '@', '=', '\n', 'G', 'U', '\xd8', ')', '\xe4', '@', '\x85', '\xeb', '\x11', 'o', '\xd8', ')', '\xe4', '@', 'f', 'f', '\xfe', '\x87', '\xd8', ')', '\xe4', '@', 'q', '=', '\xa2', '\xa0',
'\xd8', ')', '\xe4', '@', '\x8f', '\xc2', '\xed', '\xb9', '\xd8', ')', '\xe4', '@', '\xc3', '\xf5', '8', '\xd2', '\xd8', ')', '\xe4', '@', '\xe1', 'z', 'L', '\xea', '\xd8', ')', '\xe4', '@', '\xd7', '\xa3', '\xa8',
'\x03', '\xd9', ')', '\xe4', '@', 'q', '=', '\x92', '\x1c', '\xd9', ')', '\xe4', '@', 'R', '\xb8', '\x16', '6', '\xd9', ')', '\xe4', '@', '=', '\n', '\xc7', 'N', '\xd9', ')', '\xe4', '@', '\xf6', '(', '\x9c', 'g', '\xd9',
')', '\xe4', '@', '\xb8', '\x1e', '\xc5', '\x81', '\xd9', ')', '\xe4', '@', '\xd7', '\xa3', '\x88', '\x9b', '\xd9', ')', '\xe4', '@', '3', '3', '\xe3', '\xb3', '\xd9', ')', '\xe4', '@', 'q', '=', '\x82', '\xcd',
'\xd9', ')', '\xe4', '@', '\x1f', '\x85', '\x8b', '\xe7', '\xd9', ')', '\xe4', '@', '\xa4', 'p', 'U', '\x00', '\xda', ')', '\xe4', '@', '\x85', '\xeb', '\x01', '\x19', '\xda', ')', '\xe4', '@', '{', '\x14', '\xbe',
'1', '\xda', ')', '\xe4', '@', '\xe1', 'z', '\xf4', 'I', '\xda', ')', '\xe4', '@', ')', '\\', '\xcf', 'a', '\xda', ')', '\xe4', '@', '\x9a', '\x99', '\xf9', 'z', '\xda', ')', '\xe4', '@', '\xd7', '\xa3', '\x98', '\x94',
'\xda', ')', '\xe4', '@', '\x1f', '\x85', '\x1b', '\xae', '\xda', ')', '\xe4', '@', '\x85', '\xeb', '\x91', '\xc6', '\xda', ')', '\xe4', '@', '3', '3', 'C', '\xe0', '\xda', ')', '\xe4', '@', '\x00', '\x00', '\x18',
'\xf9', '\xda', ')', '\xe4', '@', 'q', '=', '\x9a', '\x12', '\xdb', ')', '\xe4', '@', 'f', 'f', '~', ',', '\xdb', ')', '\xe4', '@', '\\', '\x8f', '\xb2', 'E', '\xdb', ')', '\xe4', '@', ')', '\\', "'", '_', '\xdb', ')', '\xe4',
'@', '\xe1', 'z', '\x84', 'w', '\xdb', ')', '\xe4', '@', '\x8f', '\xc2', '%', '\x90', '\xdb', ')', '\xe4', '@', 'q', '=', '\x92', '\xa8', '\xdb', ')', '\xe4', '@', '\\', '\x8f', '\xda', '\xc1', '\xdb', ')', '\xe4',
'@', '\x14', '\xae', 'g', '\xdb', '\xdb', ')', '\xe4', '@', '=', '\n', '\xc7', '\xf4', '\xdb', ')', '\xe4', '@', '=', '\n', '\xe7', '\r', '\xdc', ')', '\xe4', '@', '\xb8', '\x1e', '=', "'", '\xdc', ')', '\xe4', '@',
'\x14', '\xae', '\xe7', '@', '\xdc', ')', '\xe4', '@', 'q', '=', '\x8a', 'Y', '\xdc', ')', '\xe4', '@', '\xe1', 'z', '\xbc', 'r', '\xdc', ')', '\xe4', '@', '\x14', '\xae', '_', '\x8c', '\xdc', ')', '\xe4', '@', '\x1f',
'\x85', '+', '\xa5', '\xdc', ')', '\xe4', '@', '\x9a', '\x99', '\x89', '\xbd', '\xdc', ')', '\xe4', '@', 'R', '\xb8', 'F', '\xd7', '\xdc', ')', '\xe4', '@', 'q', '=', 'r', '\xf0', '\xdc', ')', '\xe4', '@', '\x1f',
'\x85', '\x13', '\n', '\xdd', ')', '\xe4', '@', '\x1f', '\x85', '\xd3', '#', '\xdd', ')', '\xe4', '@', '\xec', 'Q', '@', '=', '\xdd', ')', '\xe4', '@', '{', '\x14', '\xe6', 'V', '\xdd', ')', '\xe4', '@', '=', '\n',
'\x17', 'p', '\xdd', ')', '\xe4', '@', '\n', '\xd7', '#', '\x8a', '\xdd', ')', '\xe4', '@', '=', '\n', 'w', '\xa3', '\xdd', ')', '\xe4', '@', '\xe1', 'z', '\x94', '\xbc', '\xdd', ')', '\xe4', '@', '\xcd', '\xcc',
'\xdc', '\xd5', '\xdd', ')', '\xe4', '@', '\x85', '\xeb', '9', '\xef', '\xdd', ')', '\xe4', '@', 'R', '\xb8', '\x8e', '\x07', '\xde', ')', '\xe4', '@', '\x85', '\xeb', '\x19', '!', '\xde', ')', '\xe4', '@', '{',
'\x14', '^', ':', '\xde', ')', '\xe4', '@', '\x85', '\xeb', 'a', 'S', '\xde', ')', '\xe4', '@', '\xcd', '\xcc', '\x94', 'I', '\xde', ')', '\xe4', '@', '\xb8', '\x1e', '\xad', '\x86', '\xde', ')', '\xe4', '@', '\xd7',
'\xa3', '\xf0', '\xa0', '\xde', ')', '\xe4', '@', '\x1f', '\x85', '\xfb', '\xb9', '\xde', ')', '\xe4', '@', '\xd7', '\xa3', '\xd8', '\xd2', '\xde', ')', '\xe4', '@', '=', '\n', 'g', '\xec', '\xde', ')', '\xe4', '@',
'f', 'f', '\xa6', '\x05', '\xdf', ')', '\xe4', '@', 'f', 'f', '^', '\x1f', '\xdf', ')', '\xe4', '@', 'q', '=', 'b', '6', '\xdf', ')', '\xe4', '@', 'q', '=', 'j', 'A', '\xdf', ')', '\xe4', '@', '\n', '\xd7', '\x03', 'F', '\xdf', ')',
'\xe4', '@', ')', '\\', '\x17', 'I', '\xdf', ')', '\xe4', '@', '\xd7', '\xa3', '\xa8', 'K', '\xdf', ')', '\xe4', '@', '\x85', '\xeb', 'i', 'M', '\xdf', ')', '\xe4', '@', '=', '\n', '_', 'O', '\xdf', ')', '\xe4', '@', '\xec',
```

What is written in the binary file? The binary file is written in 3 steps:

1. Filename, number of readings, number of channels, name of the channels….
2. Logged_data(*) → contains all the readings.

3. Ctrl_seq$(*) → control sequence of the experiment.
   HTBasic uses a different kind of protocol for file writing:

- It writes the string variable as it is.
- It writes the numerals in **standard numeric representation** format.
- It writes the arrays in the row major order.

3.2 Getting data from binary file

3.2.1 Approaches

We had two ways to get the data from binary file:

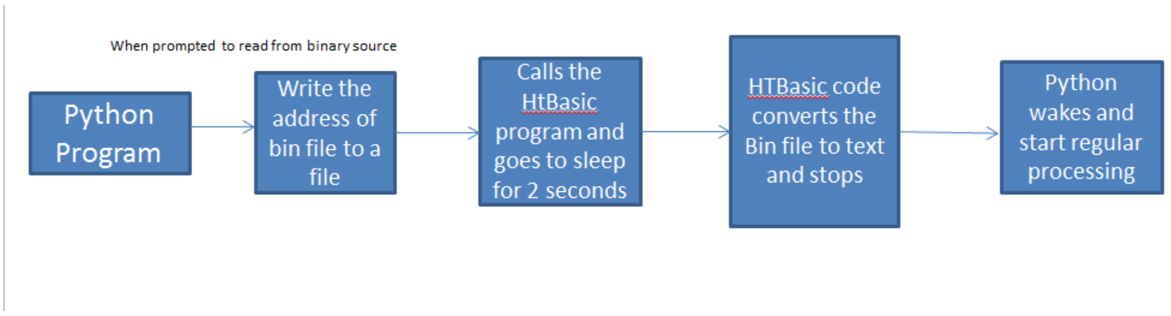
1. To write a python script to read the file and the standard representation, as python does not have a module to read these files directly, but NumPy module has a function NumPyfromstring('string') to read the number one at a time.

2. The second way is to write a HTBasic routine to read the files and call that routine from the python program.

Since the total number of reading is around 3500, so it's not efficient to convert every reading from the standard numeric format to a float. The second way is better considering the time and space complexity.

3.2.2 Final algorithm

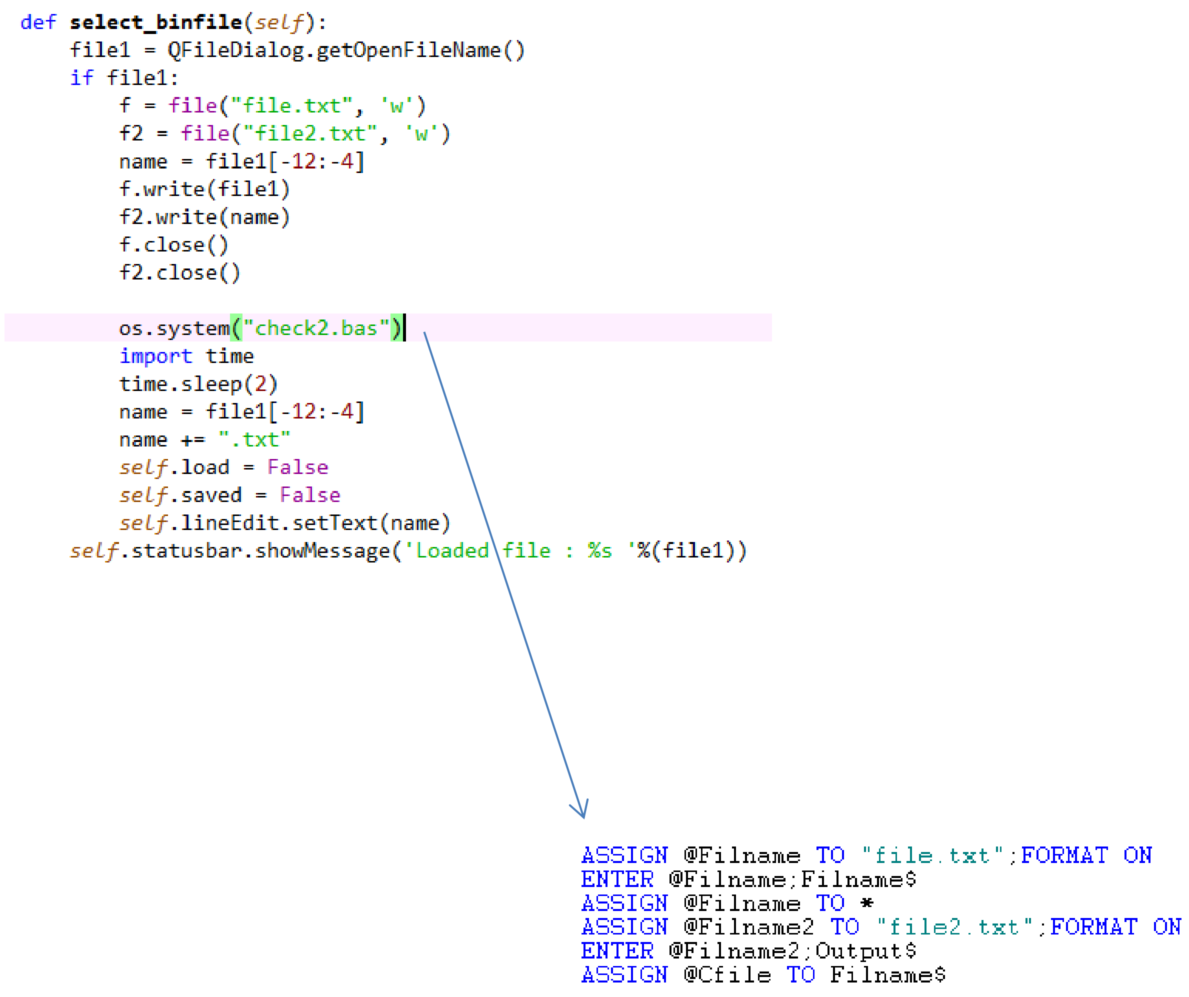
The final algorithm is depicted in the following figure:


When prompted to read from binary source
Python Program
Write the address of bin file to a file
Calls the HtBasic program and goes to sleep for 2 seconds
HTBasic code converts the Bin file to text and stops
Python wakes and start regular processing

When to python program, Cerberus Data Analyser(CDA), is prompted to receive binary data from the disk, it reads the address of the binary file and writes it to a file. After writing the address the python program calls a HTBasic routine and goes to sleep for 2 seconds. The HTBasic code reads the address of the bin file from the file and converts the binary file to a text file as produced by the original control system. Then CDA wakes up and process this text file and produce the result.

3.2.3 Interplay between Python and HTBasic

The following figure shows the interplay between CDA and HTBasic codes. On the left side is the python code and right one on the HTBasic code.



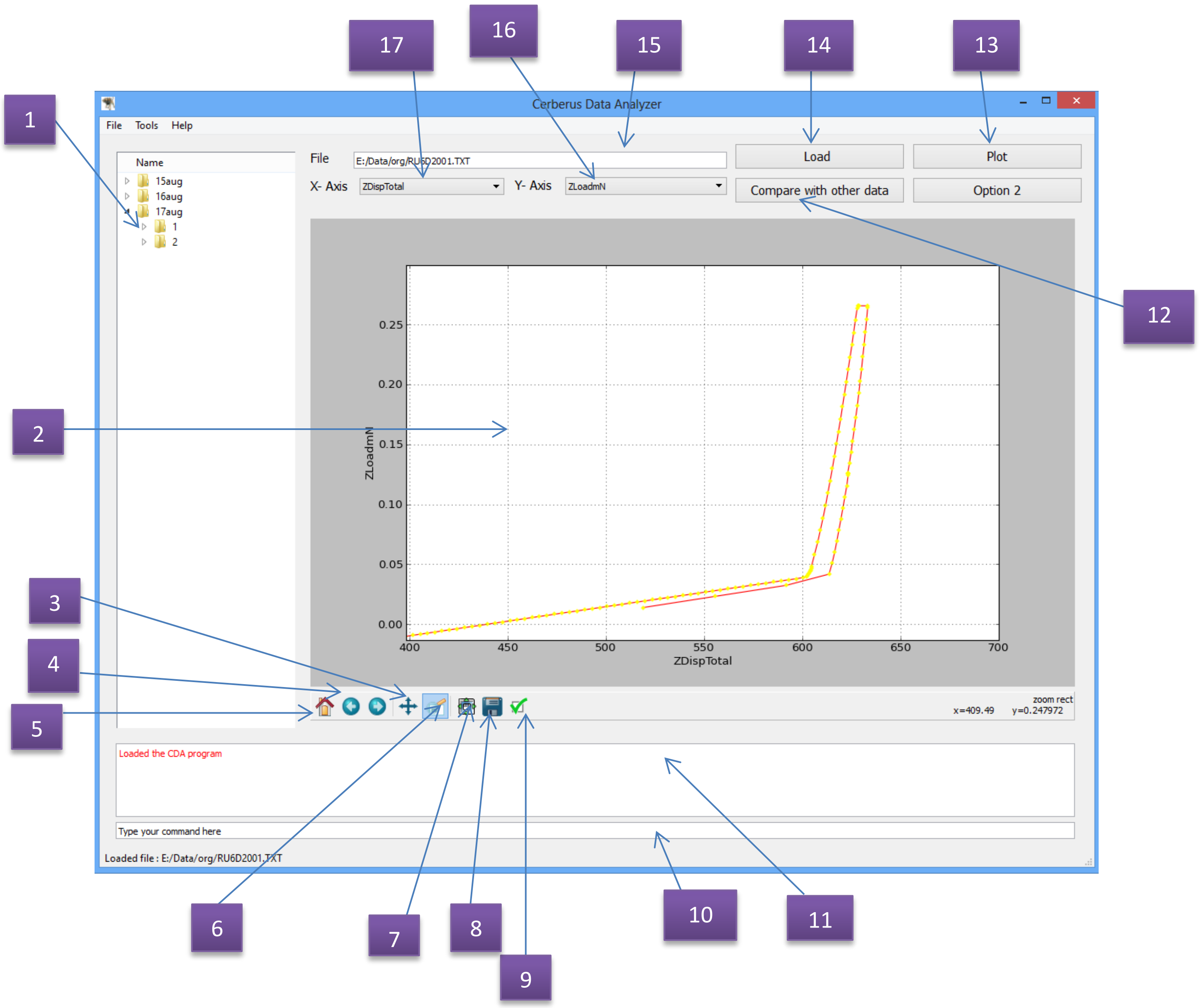
```python
def select_binfile(self):
    file1 = QFileDialog.getOpenFileName()
    if file1:
        f = file("file.txt", 'w')
        f2 = file("file2.txt", 'w')
        name = file1[-12:-4]
        f.write(file1)
        f2.write(name)
        f.close()
        f2.close()

        os.system("check2.bas")
        import time
        time.sleep(2)
        name = file1[-12:-4]
        name += ".txt"
        self.load = False
        self.saved = False
        self.lineEdit.setText(name)
    self.statusbar.showMessage('Loaded file : %s '%(file1))
```

```
ASSIGN @Filname TO "file.txt";FORMAT ON
ENTER @Filname;Filname$
ASSIGN @Filname TO *
ASSIGN @Filname2 TO "file2.txt";FORMAT ON
ENTER @Filname2;Output$
ASSIGN @Cfile TO Filname$
```

4. Results and discussions

The final GUI is shown in figure. It is developed in PyQt4 and graph is plotted using Matplotlib.



Loaded the CDA program

Type your command here

Loaded file : E:/Data/org/RU6D2001.TXT

1. Project Explorer: Displays the list of all the projects and gives the feature of opening the project from the list.
2. Plot: Displays the final plot. It is created using the promoted widget feature of Qt4 and matplotlib.
3. Pan: Pan through the graph.
4. Back and forward: Undo and redo the changes made in the graph.
5. Home: Undo all the changes.
6. Zoom Rect: Zoom the specified rectangle.
7. Adjust the axes

8. Save: Save the figure in the various formats available.
9. Settings: Change various settings like type of curve, markers etc.
10. Command line: Type any Python command here.
11. Result area: Shows the result of the command on the command line.
12. Compare the data with other tests. Both the files will be plotted on the plot area.
13. Plots the loaded file.
14. Load the data from the file and store it in the program memory.
15. Line area to display the address of the file. You can also enter the address the here.
16. Select Y-axis variable.
17. Select X-axis variable.

Each projects consists of many experiment and each experiment consists of many tests. Each test is actually a file from one Cerberus indent.

5. Summary and Conclusion:

The first job was to develop a GUI system, which is complete now. To achieve I analysed the old Cerberus system and figure the how the data is manipulated to achieve the final plot. After thorough analysis all the calibrations constants and values were figured. Then the formulae for the data manipulation were figured out. At this stage I implemented these in Python to achieve the final plot.

The next job was to read the data from the binary file. To do first job is to figure out how and what is written in the binary file. It was found out that every data is written in a specific format inherent to HTBasic. So to read that I wrote a HtBasic routine to read the data from the binary file and the convert this data to the text file as produced by the binary file. So in this way the conversion between the bin file and text file was done.

Now the last task was to develop the GUI. PyQt4 was used to develop the GUI, and with help of Matplotlib final grapgh was produced.

After the GUI is complete, the next target is to connect the GUI to the Cerberus system. The initial thought on this in to connect the old Cerberus system to the modern system running Windows 7. To achieve this we need to place a NIC(Network Interface Controller) in the old Cerberus system.

Appendix I

Calibration Constants

1. DC Calibrations

		Z-Axis	X-Axis	Y-Axis
DC Calibrations				
	Displacement Constant Term	9944.3	18506	11091
	Displacement Linear Term	2.0903	4.8506	1.6121
	Displacement Squared Term	-0.3789	-0.5316	-0.26013
	Load Constant Term	47624	43313	58032
	Load Linear Term	8.0558	21.588	3.5987
	Load Squared Term	0.003	0.0067645	0.00064854
	Spring Constant Term	230	207	204
	Spring Constant Linear Term	0.0055042		
	Spring Constant Squared Term	-0.010195		
	Spring Constant Cubed Term	1.77E-05		
	Range 0 Gain	1.00E+00	1.00E+00	1.00E+00
	Range 1 Gain	2.00E+00	2.00E+00	2.00E+00
	Range 2 Gain	4.00E+00	4.00E+00	4.00E+00
	Range 3 Gain	8.00E+00	8.00E+00	8.00E+00
	Range 4 Gain	1.60E+01	1.60E+01	1.60E+01
	Range 5 Gain	3.20E+01	3.20E+01	3.20E+01
	Range 6 Gain	6.40E+01	6.40E+01	6.40E+01
	Range 7 Gain	1.28E+02	1.28E+02	1.28E+02
	DC Load Frame Stiffness	2.20E+05	2.20E+05	2.20E+05

2. AC Calibrations

AC Calibrations	Z-Axis	X-Axis	Y-Axis
Displacement Constant Term	9935.9	18509	11088
Displacement Linear Term	2.0639	4.8118	1.6513
Displacement Squared Term	-0.37801	-0.53142	-0.26013
Load Constant Term	14023.5	12767	17098
Load Linear Term	2.36	6.4062	1.1621
Load Squared Term	-0.0002	0.0019054	-0.0013593
Modcal Factor	1.15E-08	1.15E-08	1.15E-08
Mass of Indenter	1.65E-04	1.45E-04	1.40E-04
Damping Coefficient Constant Term	0.019	0.0066396	0.017396
Damping Coefficient Linear Term	2.61E-06	0.00E+00	2.61E-06
Damping Coefficient Squared Term	7.94E-06	0.00E+00	7.94E-06
Damping Coefficient Cubed Term	-3.46E-09	0.00E+00	-3.46E-09
Damping Coefficient Quad Term	3.10E-09	0.00E+00	3.10E-09
AC Load Frame Stiffness	1.30E+05	2.12E+04	2.58E+04

	1.86E+05	2.14E+04	2.81E+04
			2.69E+04

3. Individual Masses

Individual Masses			
	Z-axis	X-axis	Y-axis
Mass from Uncoupled Dyncal	1.04E-04	9.96E-05	9.99E-05
Mass of Coupler	2.10E-05	2.10E-05	2.10E-05
Mass of Indenter (Ti rod)	1.47E-06	1.47E-06	1.47E-06
Mass of Quartz Rod	6.32E-06	6.32E-06	6.32E-06
	1.33E-04	1.28E-04	1.29E-04
Fitted Mass	1.65E-04	1.45E-04	1.40E-04

4. Area Function

Area Function		**Z-axis**	**X-Axis**	
	Lead Term	24.74350507	24.56	
	Linear Term	738.3149628	272.89	
	0.5 Term	-6111.09958	23376	
	0.25 Term	7260.883126	-288280	
	0.125 Term	4487.613794	754610	
	0.06125 Term	831.9219679	-492830	
	0.030625 Term	-1319.00673		
	0.0153125 Term	-2428.99083		
	0.00765625 Term	-2985.8581		

Acknowledgement

I want to thank my supervisors Dr. Graham Cross, FTCD, CRANN for the great opportunity to work in this project . I also thank Mr. Saket Sourav for the valuable guidance and support. I also want to thank Dr. Johann De Silva and Dr. Mithun Chowdhury for the precious guidance.