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

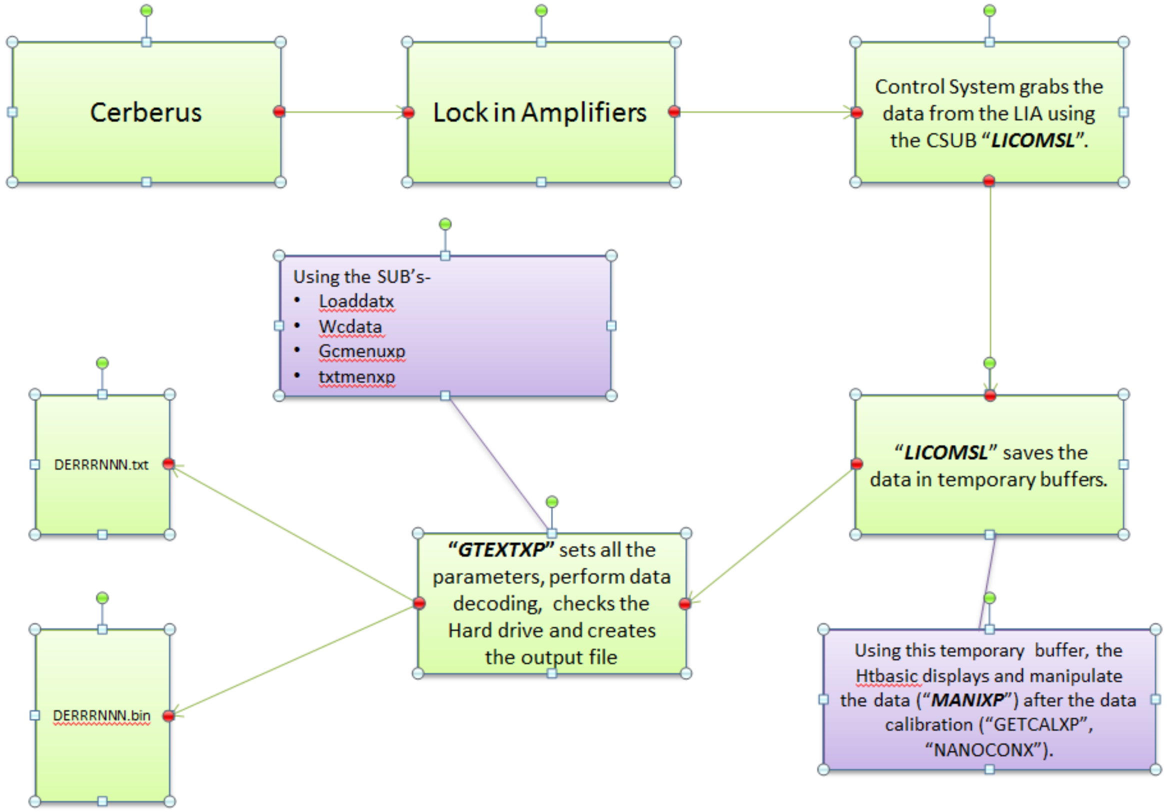
Computer Science and Engineering Discipline

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DESIGN AND MANUFACTURING JABALPUR
(6th June, 2013 – 20th June, 2013)

INTRODUCTION:-

The target of this phase was to dissect the htbasic code and find the data flow in the system. In this phase I also wrote a python script for the text to binary converswion of the data.

Data Flow in the system:



Cerberus

Lock in Amplifiers

Control System grabs the data from the LIA using the CSUB “***LICOMSL***”.

Using the SUB’s-
- Loaddatx
- Wcdata
- Gcmenuxp
- txtmenxp

DERRRNNN.txt

DERRRNNN.bin

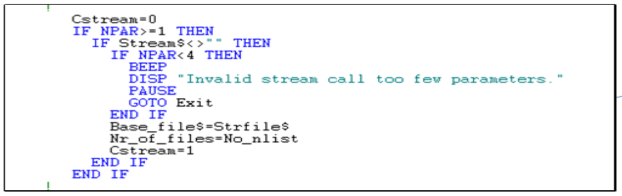
“***LICOMSL***” saves the data in temporary buffers.

“***GTEXTXP***” sets all the parameters, perform data decoding, checks the Hard drive and creates the output file

Using this temporary buffer, the Htbasic displays and manipulate the data (“***MANIXP***”) after the data calibration (“GETCALXP”, “NANOCONX”).

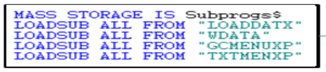
Analysis of HTBASIC code:

Description of Routine “GTEXTXP”



```
!
   Cstream=0
   IF NPAR>=1 THEN
     IF Stream$<>"" THEN
       IF NPAR<4 THEN
         BEEP
         DISP "Invalid stream call too few parameters."
         PAUSE
         GOTO Exit
       END IF
       Base_file$=Strfile$
       Nr_of_files=No_nlist
       Cstream=1
     END IF
   END IF
!
```

Checks for no. of parameter
If between 1 to 4 then set given
as base file otherwise take the
default .

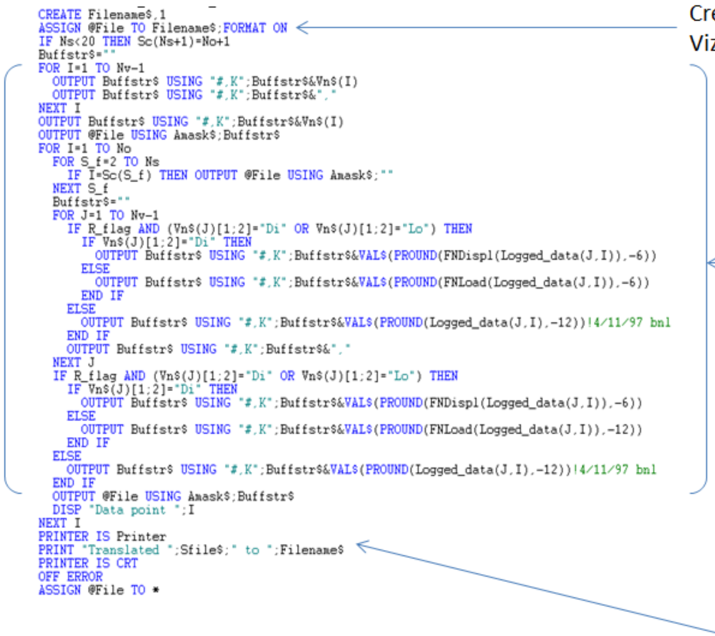


```
MASS STORAGE IS Subprogs$
LOADSUB ALL FROM "LOADDATX"
LOADSUB ALL FROM "WDATA"
LOADSUB ALL FROM "GCMENUXP"
LOADSUB ALL FROM "TXTMENXP"
```

Loads the CSUB’s necessary:
- Loaddatx: file load routine , loads the default buffer or the specified buffer.
- Wdata: This subprogram loads up logged_data with the data from only those subfiles and variables specified by Sf(*) and Var(*).
- TXTMENXP: This is the parameter menu for the generic text file generator GTEXTXP.

  foutflag 0/1 no/yes copy text files to floppy
  Mp$ output format flag 'M' Mac or 'P' PC text format
  subf_flag 0/1 no/yes include subfile number in file?
  dp_flag 0/1 no/yes include original observation # in file?

Description of Routine “GTEXTXP” (contd..)



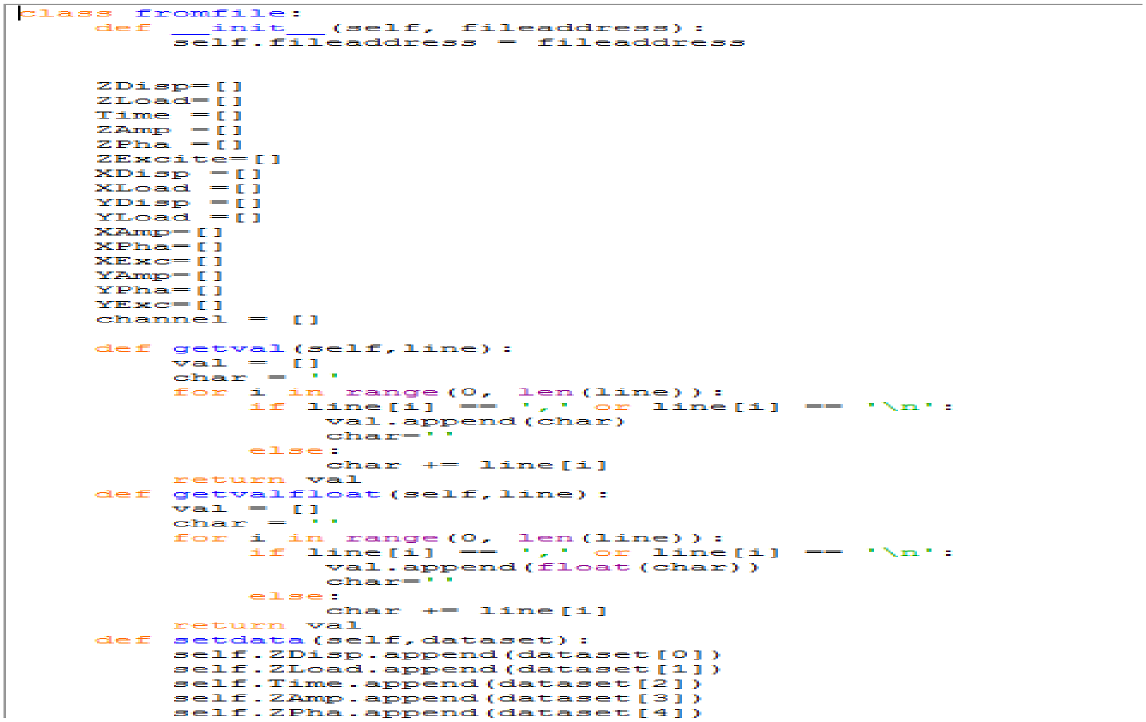
```
CREATE Filename$,1
ASSIGN @File TO Filename$;FORMAT ON
IF Ns<20 THEN Sc(Ns+1)=No+1
Buffstr$=""
FOR I=1 TO Nv-1
  OUTPUT Buffstr$ USING "#,K";Buffstr$&Vn$(I)
  OUTPUT Buffstr$ USING "#,K";Buffstr$&","
NEXT I
OUTPUT Buffstr$ USING "#,K";Buffstr$&Vn$(I)
OUTPUT @File USING Amask$;Buffstr$
FOR I=1 TO No
  FOR S_f=2 TO Ns
    IF I=Sc(S_f) THEN OUTPUT @File USING Amask$;""
  NEXT S_f
  Buffstr$=""
  FOR J=1 TO Nv-1
    IF R_flag AND (Vn$(J)[1;2]="Di" OR Vn$(J)[1;2]="Lo") THEN
      IF Vn$(J)[1;2]="Di" THEN
        OUTPUT Buffstr$ USING "#,K";Buffstr$&VAL$(PROUND(FNDispl(Logged_data(J,I)),-6))
      ELSE
        OUTPUT Buffstr$ USING "#,K";Buffstr$&VAL$(PROUND(FNLoad(Logged_data(J,I)),-6))
      END IF
    ELSE
      OUTPUT Buffstr$ USING "#,K";Buffstr$&VAL$(PROUND(Logged_data(J,I),-12))!4/11/97 bnl
    END IF
    OUTPUT Buffstr$ USING "#,K";Buffstr$&","
  NEXT J
  IF R_flag AND (Vn$(J)[1;2]="Di" OR Vn$(J)[1;2]="Lo") THEN
    IF Vn$(J)[1;2]="Di" THEN
      OUTPUT Buffstr$ USING "#,K";Buffstr$&VAL$(PROUND(FNDispl(Logged_data(J,I)),-6))
    ELSE
      OUTPUT Buffstr$ USING "#,K";Buffstr$&VAL$(PROUND(FNLoad(Logged_data(J,I)),-12))
    END IF
  ELSE
    OUTPUT Buffstr$ USING "#,K";Buffstr$&VAL$(PROUND(Logged_data(J,I),-12))!4/11/97 bnl
  END IF
  OUTPUT @File USING Amask$;Buffstr$
  DISP "Data point ";I
NEXT I
PRINTER IS Printer
PRINT "Translated ";Sfile$;" to ";Filename$
PRINTER IS CRT
OFF ERROR
ASSIGN @File TO *
```

Creates an i/o link for filename$
Viz. output file

This is the part where data is written to the text files.

Shows the message

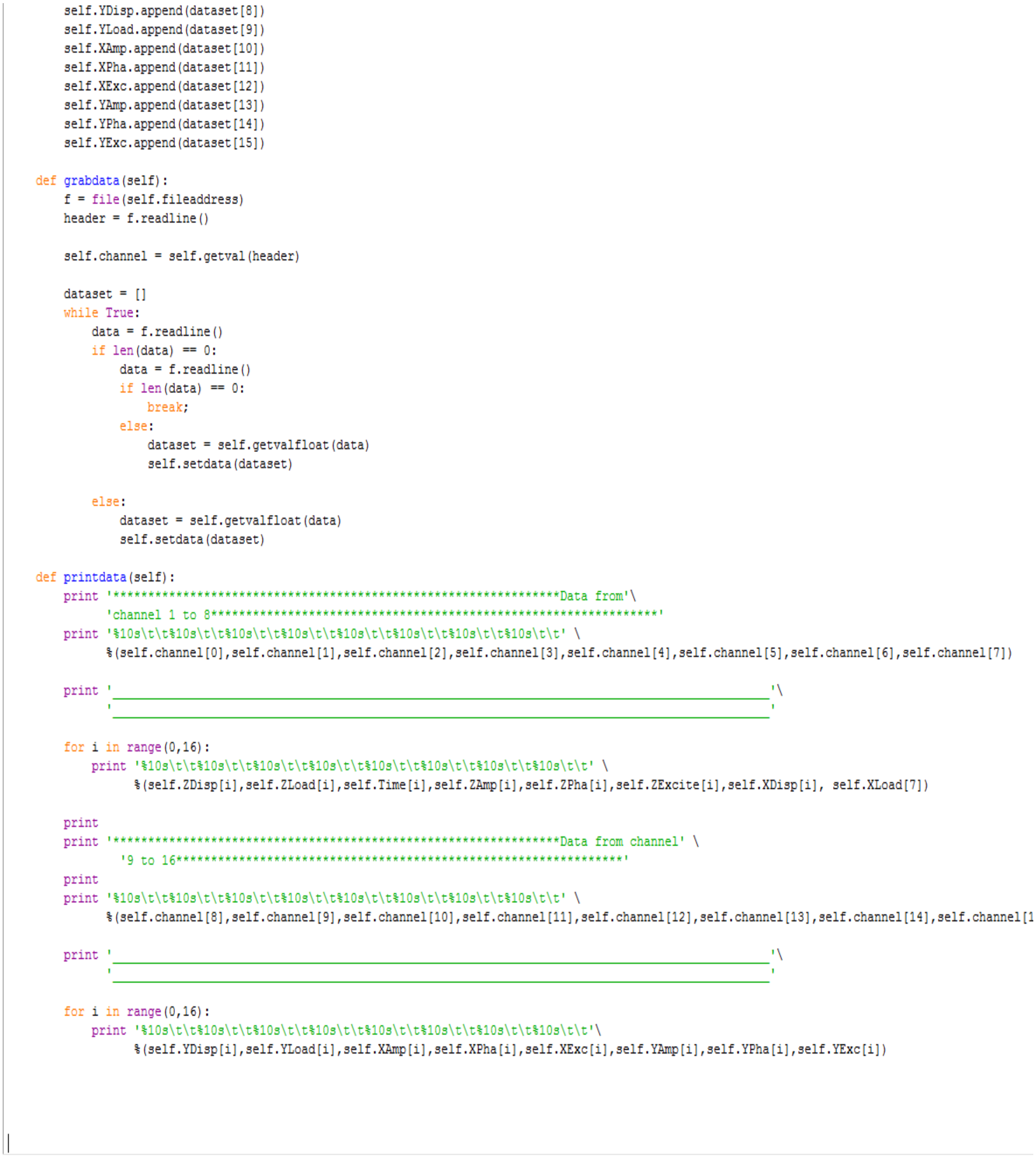
New python code:



```
class fromfile:
    def __init__(self, fileaddress):
        self.fileaddress = fileaddress


    ZDisp=[]
    ZLoad=[]
    Time  =[]
    ZAmp  =[]
    ZPha  =[]
    ZExcite=[]
    XDisp  =[]
    XLoad  =[]
    YDisp  =[]
    YLoad  =[]
    XAmp=[]
    XPha=[]
    XExc=[]
    YAmp=[]
    YPha=[]
    YExc=[]
    channel = []

    def getval(self,line):
        val = []
        char = ''
        for i in range(0, len(line)):
            if line[i] == ',' or line[i] == '\n':
                val.append(char)
                char=''
            else:
                char += line[i]
        return val
    def getvalfloat(self,line):
        val = []
        char = ''
        for i in range(0, len(line)):
            if line[i] == ',' or line[i] == '\n':
                val.append(float(char))
                char=''
            else:
                char += line[i]
        return val
    def setdata(self,dataset):
        self.ZDisp.append(dataset[0])
        self.ZLoad.append(dataset[1])
        self.Time.append(dataset[2])
        self.ZAmp.append(dataset[3])
        self.ZPha.append(dataset[4])
```



```
        self.YDisp.append(dataset[8])
        self.YLoad.append(dataset[9])
        self.XAmp.append(dataset[10])
        self.XPha.append(dataset[11])
        self.XExc.append(dataset[12])
        self.YAmp.append(dataset[13])
        self.YPha.append(dataset[14])
        self.YExc.append(dataset[15])

    def grabdata(self):
        f = file(self.fileaddress)
        header = f.readline()

        self.channel = self.getval(header)

        dataset = []
        while True:
            data = f.readline()
            if len(data) == 0:
                data = f.readline()
                if len(data) == 0:
                    break;
                else:
                    dataset = self.getvalfloat(data)
                    self.setdata(dataset)

            else:
                dataset = self.getvalfloat(data)
                self.setdata(dataset)

    def printdata(self):
        print '****************************************************************Data from'\
              'channel 1 to 8****************************************************************'
        print '%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t' \
              %(self.channel[0],self.channel[1],self.channel[2],self.channel[3],self.channel[4],self.channel[5],self.channel[6],self.channel[7])

        print '_______________________________________________________________________________________________'\
              '_______________________________________________________________________________________________'

        for i in range(0,16):
            print '%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t' \
                  %(self.ZDisp[i],self.ZLoad[i],self.Time[i],self.ZAmp[i],self.ZPha[i],self.ZExcite[i],self.XDisp[i], self.XLoad[7])

        print
        print '****************************************************************Data from channel' \
                '9 to 16****************************************************************'
        print
        print '%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t' \
              %(self.channel[8],self.channel[9],self.channel[10],self.channel[11],self.channel[12],self.channel[13],self.channel[14],self.channel[1

        print '_______________________________________________________________________________________________'\
              '_______________________________________________________________________________________________'

        for i in range(0,16):
            print '%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t%10s\t\t'\
                  %(self.YDisp[i],self.YLoad[i],self.XAmp[i],self.XPha[i],self.XExc[i],self.YAmp[i],self.YPha[i],self.YExc[i])
```

RESULT & DISCUSSION:-

1. After deep evaluation of the available options due to its slight advantage over Matlab and increasing use in the scientific community, Pythonxy is selected as the final environment.
2. Following points about the "XPMASTER" code were identified:
   a. We must check whether it is "rerun" of the "XPMASTER" to do calibration.
   b. Identify the printer status initially, and make it available throughout the program.
   c. Select the appropriate drivers for the printer; in this case it is "HP-PC".
   d. Perform the appropriate test to ensure that appropriate displacement card, load card and motor card are present.

NEXT OBJECTIVE:-

The objectives of the next phase in the project are as follows-

- ✓ To fully explore the htbasic code and create the interaction model between different subroutines.
- ✓ To study the raw binary data from the Cerberus system and write a python code to convert it into the desired txt format.