



**DZERO SUPERCONDUCTING SOLENOID
QUENCH DATA AND SLOW DUMP
DATA AQUISITION**

**ENGINEERING NOTE
3823.111-EN-496**

**Dan Markley
<http://dmacs-nt-server.fnal.gov/>
Fermi Accelerator Lab
June 9, 1998**

Work Supported by the U.S. Department of Energy under
contract
No. DE-AC02-76CHO3000

Fermilab has paid for and is the sole owner of any programming code
written for this project. This programming code may not be duplicated or
distributed without proper consent.

Table of Contents

- I. ABSTRACT
- II. INTRODUCTION
- III. OVERVIEW OF PROGRAM SEQUENCING
- IV. FAST QUENCH DATA
 - A. Parameters
 - B. Ladder Logic
 - C. Special Function Logic
 - D. Visual Basic Module Code
 - E. Visual Basic Form Code
- V. SLOW DUMP DATA
 - A. Parameters
 - B. Ladder Logic
 - C. Special Function Logic
 - D. Visual Basic Module Code
 - E. Visual Basic Form Code
- VI. FIRST FAULT

I. ABSTRACT

This Dzero Engineering note describes the method for which the 2 Tesla Superconducting Solenoid Fast Dump and Slow Dump data are accumulated, tracked and stored.

II. INTRODUCTION

The 2 Tesla Solenoid has eleven data points that need to be tracked and then stored when a fast dump or a slow dump occur. The TI555 (Texas Instruments) PLC (Programmable Logic Controller) which controls the DC power circuit that powers the Solenoid, also has access to all the voltage taps and other equipment in the circuit. The TI555 constantly logs these eleven points in a rotating memory buffer. When either a fast dump (dump switch opens) or a slow dump (power supply turns off) occurs, the TI555 organizes the respective data and will download the data to a file on D0-CCRS2. This data in this file is moved over ethernet and is stored in a CSV (comma separated format) file which can easily be examined by Microsoft Excel or any other spreadsheet.

The 2 Tesla solenoid control system also locks in first fault information. The TI555 decodes the first fault and passes it along to the program collecting the data and storing it on D0-CCRS2. This first fault information is then part of the file.

These data files that are created are stored on D0-CCRS2 on "D" Hard Drive in D:\Quenchdata. The file name is constructed using the following format:

Sol_FDATA "Year" "Month" "Day" "Hour" "Min"

Example: Sol_FDATA 1998Y 3M 26D 12H 19M

Or

Sol_SDATA "Year" "Month" "Day" "Hour" "Min"

Example: Sol_SDATA 1998Y 5M 8D 10H 29M

Address	Description	Raw Low	Raw Hi	Correction	Low	Hi
DC_AI1	0-10V Analog In - Lead A V	0V	10	DC_AI Corrected (V-5)/50	-0.1V	0.1V
DC_AI4	0-10V Analog In - Lead B V	0V	10	DC_AI4 Corrected (V-5)/50	-0.1V	0.1V
DC_AI7	0-10V Analog In - Trans A V	0V	10	DC_AI7 Corrected (V-5)/50	-0.1V	0.1V
DC_AI10	0-10V Analog In - Trans B V	0V	10	DC_AI10C Corrected (V-5)/50	-0.1V	0.1V
DC_AI13	0-10V Analog In - Chimney A V	0V	10	DC_AI13 Corrected (V-5)/50	-0.1V	0.1V
DC_AI16	0-10V Analog In - Chimney B V	0V	10	DC_AI16 Corrected (V-5)/50	-0.1V	0.1V
DC_AI19	0-10V Analog In - Outer Coil V	0V	10	DC_AI19 Corrected (V-5)*2	-20V	20V
DC_AI21	0-10V Analog In - CT Imbal V	0V	10	DC_AI21 Corrected (V-5)/25	-0.2V	0.2V
DC_AI24	0-10V Analog In - QT Imbal V	0V	10	DC_AI24 Corrected (V-5)/12.5	-0.4V	0.4V
DC_AI41	0-10V Bipolar An In - Mag Curr-2	-10V	10V		-5,000A	5,000A

Signal description and Conditioning

III. OVERVIEW OF PROGRAM SEQUENCING

1. Data is recorded in circular fixed address buffer memory for each of the points in the PLC variable memory. All points use a common indexer. All points are stored in an integer scaled format.
2. Upon a trip the data continues to be recorded in the circular memory buffer for the designated time.
3. Once the designated recording time has expired the PLC terminates the rotating buffer recording.
4. The PLC then converts data points one, two, and three from integer to a real number and stores the blocks of data in a variable memory area designated as the transport area. The PLC then sets an internal coil that is in a database block of the Intellution FIX SCADA node, named D0-CCRS2.

5. D0-CCRS2 database is configured to launch one of two Visual Basic programs depending on whether the slow dump coil or the fast dump coil was set.
6. The Visual Basic program transfers the data from the PLC transport area over Ethernet, using the TCP/IP protocol, to an internal array on D0-CCRS2. Then sets a coil in the PLC signaling the end of the transfer.
7. The PLC then converts data points four, five, and six from integer to a real number and stores the blocks of data in a variable memory area designated as the transport area. The PLC then sets an internal coil that signals the program that the transport area is ready.
8. The Visual Basic program transfers the data from the PLC transport area over Ethernet, to an internal array on D0-CCRS2. Then sets a coil in the PLC signaling the end of the transfer.
9. The PLC then converts data points seven, eight, and nine from integer to a real number and stores the blocks of data in a variable memory area designated as the transport area. The PLC then sets an internal coil that signals the program that the transport area is ready.
10. The Visual Basic program transfers the data from the PLC transport area over Ethernet, to an internal array on D0-CCRS2. Then sets a coil in the PLC signaling the end of the transfer.
11. The PLC then converts data points ten and eleven from integer to a real number and stores the blocks of data in a variable memory area designated as the transport area. The PLC then sets an internal coil that signals the program that the transport area is ready.
12. The Visual Basic program transfers the data from the PLC transport area over Ethernet, to an internal array on D0-CCRS2. Then sets a coil in the PLC signaling the end of the transfer. It also transfers a coded number from the PLC to interpret the First Fault.
13. The Visual Basic Program stores the array to a disk file, then terminates.
14. The PLC resets it's indexers and continues to store the eleven data points in rotating memory buffers waiting for the next trip.

IV. FAST QUENCH DATA

A. FAST QUENCH DATA Parameters

All eleven Fast Data points are stored every 25mS or at a rate of 40Hz for a total of 3600 data values/point. These data points are stored continuously in the TI555 variable memory for a total time of 90 seconds. The PLC programming is configured such that after the dump switch opens the rotating buffer continues recording for 60 more seconds. The end result is after a dump all eleven points are recorded for 30 seconds before the event, up to 60 seconds after the event.

'Description	PLC Integer Address	PLC Real Address
#0 Lead A Voltage(WX1)	V50000-V53599	V1000-V8199
#1 Lead B Voltage(WX4)	V53600-V57199	V8200-V15399
#2 Transition A Voltage(WX7)	V57200-V60799	V15400-V22599
#3 Transition B Voltage(WX10)	V60800-V64399	V1000-V8199
#4 Chimney A Voltage(WX13)	V64400-V67999	V8200-V15399
#5 Chimney B Voltage(WX16)	V68000-V71599	V15400-V22599
#6 Outer Coil Voltage(WX19)	V71600-V75199	V1000-V8199
#7 Inner Coil Voltage(WX20)	V75200-V78799	V8200-V15399
#8 CenterTap Imbalance Voltage(WX21)	V78800-V82399	V15400-V22599
#9 Quarter Tap imbalance Voltage(WX24)	V82400-V85999	V1000-V8199
#10 Magnet Current(WX41)	V86000-V89599	V8200-V15399


```

!FDATA          TMR99-----+
!AQUIRE        !           !
!FINISHED       !           !
!   C100         !           !
440 [---] [-----] [-----] (    )
!               P=   300.0  !
!               !           !
!               !           !
!               !           !
!               !           !
!FDATA          !           !
!AQUIRE        !           !
!FINISHED       !           !
!   C100         !           !
[---] [-----] [-----]
!               +-----+
!FDATA          TMR100-----+
!TRIGGER        !           !
!               !           !
!   C98          !           !
445 [---]/[-----] [-----] (    )
!               P=    60.0  !
!               !           !
!               !           !
!               !           !
!               !           !
!FDATA          !           !
!TRIGGER        !           !
!               !           !
!   C98          !           !
[---]/[-----] [-----]
!               +-----+
!FDATA          SFPGM1-----+
!RECORD OR      !           !
!DUMP STAT      !           !
!   C99         !           !
450 [---] [-----] [-----] (    )
!               !           !
!               !           !
!               !           !
!               !           !
!               !           !
!FDATA          SFPGM9-----+
!RECORD OR      !           !
!DUMP STAT      !           !
!   C99         !           !
453 [---] [-----] [-----] (    )
!               !           !
!               !           !
!               !           !
!               !           !
!               !           !

```


	MOVE13-----+	MOVE14-----+	MOVE34-----+	Dummy
	! WORD	! WORD	! WORD	
	! Trans B	! Chimney A	! Chimney B	
>	! Volts	! Volts	! Volts	C1
a-----				----()
>	!TS:WX10	!TS:WX13	!TS:WX16	
	!	!	!	
	!	!	!	
	!SI:	!SI:	!SI:	
	! FDATA	! FDATA	! FDATA	
	! WX10 STORE	! WX13 STORE	! WX16 STORE	
	! BEGINNING	! BEGINNING	! BEGINNING	
	!TD:V60800	!TD:V64400	!TD:V68000	
	!	!	!	
	!DI:V100	!DI:V100	!DI:V100	
	! N=+1	! N=+1	! N=+1	
	+-----+	+-----+	+-----+	

	MOVE15-----+	MOVE16-----+	MOVE17-----+	
! FDATA	! WORD	! WORD	! WORD	
!RECORD OR	! Outer Coil	! Inner Coil	! Center Tap	
!DUMP STAT	! Volts	! Volts	! Imbal Volts	>
! C99				
509 [---]/[-----]				a-----
!	!TS:WX19	!TS:WX20	!TS:WX21	>
!	!	!	!	
!	!	!	!	
!	!	!	!	
!	!SI:	!SI:	!SI:	
!	! FDATA	! FDATA	! FDATA	
!	! WX19 STORE	! WX20 STORE	! WX21 STORE	
!	! BEGINNING	! BEGINNING	! BEGINNING	
!	!TD:V71600	!TD:V75200	!TD:V78800	
!	!	!	!	
!	!DI:V100	!DI:V100	!DI:V100	
!	! N=+1	! N=+1	! N=+1	
!	+-----+	+-----+	+-----+	

	MOVE18-----+	MOVE19-----+	Dummy
	! WORD	! WORD	
	! Qtr Tap	! MagCurr-2	
>	! Imbal Volts	! Holec Out	C1
a-----			----()
>	!TS:WX24	!TS:WX41	
	!	!	
	!	!	
	!SI:	!SI:	
	! FDATA	! FDATA	
	! WX24 STORE	! WX41 STORE	
	! BEGINNING	! BEGINNING	
	!TD:V82400	!TD:V86000	
	!	!	
	!DI:V100	!DI:V100	
	! N=+1	! N=+1	
	+-----+	+-----+	


```

!FDATA      ADD10-----+
!RECORD OR  !
!DUMP STAT  !   FDATA   !
!   C99     !   LADDER  !
546 [---]/[---] ! INDEXER !-----+ C1
!           ! A:V100    !
!           !           !
!           !           !
!           !           !
!           ! B:+1     !
!           !           !
!           !   FDATA   !
!           !   LADDER  !
!           !   INDEXER !
!           ! C:V100    !
!           !           !
!           +-----+

```

```

!FDATA      FDATA      LDC25-----+
!RECORD OR  LADDER     !
!DUMP STAT  INDEXER    !   FDATA   !
!   C99     V100 +3600 !   LADDER  !
552 [---]/[-----]>= INT[---] ! INDEXER !-----+ C1
!           ! A:V100    !
!           !           !
!           ! N=0       !
!           !           !
!           !           !
!           +-----+

```

C. FAST QUENCH DATA Special Function Logic

TITLE: FDATA RL1

SF PROGRAM 1

```
-----
00001 *      ***THIS SFPGM SCALES AND MOVES FAST DATA TO REAL V MEMORY***
              V100 IS INDEXER
              V102 IS THE REAL NUMBER INDEXER

00002 *      THIS CONDITIONS THE FIRST 3 FAST DATA REAL TAGS
00003 IMATH   V101 := V100 + 1
00004 IMATH   V102 := 1
00005 LABEL   LABEL.....: 1
00006 IF      V101 >= 3601
00007 IMATH   V101 := 1
00008 ELSE
00009 ENDIF
00010 SCALE   BINARY INPUT... (V50000(V101)) SCALED RESULT.: V1000.(V102)
              LOW LIMIT.....: -0.1          HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO           BIPOLAR.....: NO
00011 SCALE   BINARY INPUT... (V53600(V101)) SCALED RESULT.: V8200.(V102)
              LOW LIMIT.....: -0.1          HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO           BIPOLAR.....: NO
00012 SCALE   BINARY INPUT... (V57200(V101)) SCALED RESULT.: V15400.(V102)
              LOW LIMIT.....: -0.1          HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO           BIPOLAR.....: NO
00013 IMATH   V101 := V101 + 1
00014 IMATH   V102 := V102 + 2
00015 IF      V102 < 7201
00016 GOTO     LABEL.....: 1
00017 ELSE
00018 ENDIF
**** END ****
```

TITLE: FDATA RL2

SF PROGRAM 2

```
-----
00001 *      THIS CONDITIONS THE NEXT 3 FAST DATA REAL TAGS
00002 IMATH   V101 := V100 + 1
00003 IMATH   V102 := 1
00004 LABEL   LABEL.....: 1
00005 IF      V101 >= 3601
00006 IMATH   V101 := 1
00007 ELSE
00008 ENDIF
00009 SCALE   BINARY INPUT... (V60800(V101)) SCALED RESULT.: V1000.(V102)
              LOW LIMIT.....: -0.1          HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO           BIPOLAR.....: NO
00010 SCALE   BINARY INPUT... (V64400(V101)) SCALED RESULT.: V8200.(V102)
              LOW LIMIT.....: -0.1          HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO           BIPOLAR.....: NO
00011 SCALE   BINARY INPUT... (V68800(V101)) SCALED RESULT.: V15400.(V102)
              LOW LIMIT.....: -0.1          HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO           BIPOLAR.....: NO
00012 IMATH   V101 := V101 + 1
00013 IMATH   V102 := V102 + 2
00014 IF      V102 < 7201
00015 GOTO     LABEL.....: 1
00016 ELSE
00017 ENDIF
**** END ****
```

TITLE: FDATARL3

SF PROGRAM 3

```
-----
00001 *          THIS CONDITIONS THE NEXT 3 FAST DATA REAL TAGS
00002 IMATH      V101 := V100 + 1
00003 IMATH      V102 := 1
00004 LABEL      LABEL.....: 3
00005 IF         V101 >= 3601
00006 IMATH      V101 := 1
00007 ELSE
00008 ENDIF
00009 SCALE BINARY INPUT..: (V71600(V101)) SCALED RESULT.: V1000.(V102)
          LOW LIMIT.....: -10.0          HIGH LIMIT.....: 10.0
          20% OFFSET.....: NO            BIPOLAR.....: NO
00010 SCALE BINARY INPUT..: (V75200(V101)) SCALED RESULT.: V8200.(V102)
          LOW LIMIT.....: -10.0          HIGH LIMIT.....: 10.0
          20% OFFSET.....: NO            BIPOLAR.....: NO
00011 SCALE BINARY INPUT..: (V78800(V101)) SCALED RESULT.: V15400.(V102)
          LOW LIMIT.....: -0.2           HIGH LIMIT.....: 0.2
          20% OFFSET.....: NO            BIPOLAR.....: NO
00012 IMATH      V101 := V101 + 1
00013 IMATH      V102 := V102 + 2
00014 IF         V102 < 7201
00015 GOTO       LABEL.....: 3
00016 ELSE
00017 ENDIF
**** END ****
```

TITLE: FDATARL4

SF PROGRAM 4

```
-----
00001 *          THIS CONDITIONS THE LAST 2 FAST DATA REAL TAG
00002 IMATH      V101 := V100 + 1
00003 IMATH      V102 := 1
00004 LABEL      LABEL.....: 1
00005 IF         V101 >= 3601
00006 IMATH      V101 := 1
00007 ELSE
00008 ENDIF
00009 SCALE BINARY INPUT..: (V82400(V101)) SCALED RESULT.: V1000.(V102)
          LOW LIMIT.....: -0.4           HIGH LIMIT.....: 0.4
          20% OFFSET.....: NO            BIPOLAR.....: NO
00010 SCALE BINARY INPUT..: (V86000(V101)) SCALED RESULT.: V8200.(V102)
          LOW LIMIT.....: -5000.0        HIGH LIMIT.....: 5000.0
          20% OFFSET.....: NO            BIPOLAR.....: YES
00011 IMATH      V101 := V101 + 1
00012 IMATH      V102 := V102 + 2
00013 IF         V102 < 7201
00014 GOTO       LABEL.....: 1
00015 ELSE
00016 ENDIF
**** END ****
```

D. FAST QUENCH DATA Visual Basic Module Code

```
'This Visual Basic 5.0 Program was written by Dan Markley
' and is the Property of FermiLab. January of 1998.
'This program retrieves fast quench data that was stored in
' Dzero Superconducting Solenoid Control System TI555
' Programmable Logic Controller. It stores all the data in
' a disk file with todays date and time as a file name.
'This data is stored as a real number(in PLC 2 16 bit Vmems)
'There are 11 parameters with 3600 (7200 Vmems)values per 90 seconds
each.
```

Description	PLC Integer Address	PLC Real Address
#0 Lead A Voltage(WX1)	V50000-V53599	V1000-V8199
#1 Lead B Voltage(WX4)	V53600-V57199	V8200-V15399
#2 Transition A Voltage(WX7)	V57200-V60799	V15400-V22599
#3 Transition B Voltage(WX10)	V60800-V64399	V1000-V8199
#4 Chimney A Voltage(WX13)	V64400-V67999	V8200-V15399
#5 Chimney B Voltage(WX16)	V68000-V71599	V15400-V22599
#6 Outer Coil Voltage(WX19)	V71600-V75199	V1000-V8199
#7 Inner Coil Voltage(WX20)	V75200-V78799	V8200-V15399
#8 CenterTap Imbalance Voltage(WX21)	V78800-V82399	V15400-V22599
#9 Quarter Tap imbalance Voltage(WX24)	V82400-V85999	V1000-V8199
#10 Magnet Current(WX41)	V86000-V89599	V8200-V15399

```
Public connect As Integer
Public coilwrite As Integer
Public coilread As Integer
Public qdata(3600, 11) As Single
Public colnumber As Long
Public currstartadd As Long
Public rowcount As Long
Public comerror As String
Public fault As Integer
Public Ffault As String
```

```
Sub Main()
frmDocument.Text1 = "Initailizing Connection to Solenoid PLC"
frmDocument.Text4 = "None"
connect = 0
Call busy
'Opening a Connection over Ethernet to PLC
frmMain.E25721.Function = 0
frmMain.E25721.Refresh

Call busy
frmDocument.Text1 = "Retrieving Data from Solenoid PLC"
connect = 4
rowcount = 0
'11 colnumber Loops for 11 parameters
For colnumber = 0 To 10 Step 1
If colnumber = 0 Then startadd = 1000 Else
If colnumber = 1 Then startadd = 8200 Else
If colnumber = 2 Then startadd = 15400 Else
If colnumber = 3 Then startadd = 1000 Else
```

```

If colnumber = 4 Then startadd = 8200 Else
If colnumber = 5 Then startadd = 15400 Else
If colnumber = 6 Then startadd = 1000 Else
If colnumber = 7 Then startadd = 8200 Else
If colnumber = 8 Then startadd = 15400 Else
If colnumber = 9 Then startadd = 1000 Else
If colnumber = 10 Then startadd = 8200 Else
    For currstartadd = startadd To startadd + 7000 Step 200
        Call busy
        Call loading
        connect = 1
        If rowcount = 3600 Then rowcount = 0 Else
        'Pulling 200 VMEMS from PLC
        frmMain.E25721.Function = 8
        frmMain.E25721.DataWords = 200
        frmMain.E25721.StartAddr = currstartadd
        frmMain.E25721.MemType = 0
        frmMain.E25721.Refresh
        frmDocument.Text6 = currstartadd
    Next currstartadd
    If colnumber = 2 Then Call nextsfpgm Else
    If colnumber = 5 Then Call nextsfpgm Else
    If colnumber = 8 Then Call nextsfpgm Else
    Call busy
    Call loading
Next colnumber
    frmDocument.Text2 = "data"

Call busy
frmDocument.Text1 = "Finished"
frmMain.E25721.Function = 5
frmMain.E25721.MemType = 12
frmMain.E25721.DataWords = 1
frmMain.E25721.InfoIndex = 0
frmMain.E25721.RandomAddr = 97
frmMain.E25721.Info = 3
frmMain.E25721.Refresh

'This read gets the first fault word
Call busy
frmMain.E25721.Function = 4
frmMain.E25721.MemType = 0
frmMain.E25721.DataWords = 1
frmMain.E25721.InfoIndex = 0
frmMain.E25721.RandomAddr = 110
frmMain.E25721.Refresh
Call busy
frmMain.E25721.InfoIndex = 0
fault = frmMain.E25721.Info
Call DECODE

Call busy
frmDocument.Text1 = "Closing Connection to Solenoid PLC"
connect = 3
frmMain.E25721.Function = 1
frmMain.E25721.DataWords = 50
frmMain.E25721.StartAddr = 10
frmMain.E25721.MemType = 0
frmMain.E25721.Refresh

```

'The next section creates a disk file with todays date and time.

```

frmDocument.Text1 = "Saving Data to Disk File"
Dim one, two As Variant
Dim yrtemp, motemp, dytemp, hrtemp, mintemp As Integer
Dim yr, mo, dy, today, hr, min As String
one = Date
two = Time
yrtemp = Year(one)
motemp = Month(one)
dytemp = Day(one)
hrtemp = Hour(two)
mintemp = Minute(two)
yr = Str(yrtemp)
mo = "Y" & Str(motemp)
dy = "M" & Str(dytemp)
hr = "D" & Str(hrtemp)
min = "H" & Str(mintemp) & "M"
today = "D:\QUENCHDATA\" & "Sol_FDATA" & yr & mo & dy & hr & min &
".csv"
Open today For Output As #1
Print #1, comerror
Print #1, "first fault was:", Ffault
If comerror <> "Successful Communication" Then
    Close #1
End
End If
Print #1, "Quench time,"; "Lead A Voltage,"; "Lead B Voltage,"; _
"Transition A Voltage,"; "Transition B Voltage,"; "Chimney A
Voltage,"; _
"Chimney B Voltage,"; "Outer Coil Voltage,"; "Inner Coil Voltage,";
_
"Center Tap Imbalance Voltage,"; "Quarter Tap Imbalance Voltage,";
"Magnet Current"
Dim qtime As Single
qtime = -30#
For i% = 0 To 3599 Step 1
    Print #1, qtime; ","; qdata(i%, 0); ","; qdata(i%, 1); ",";
qdata(i%, 2); _
","; qdata(i%, 3); ","; qdata(i%, 4); ","; qdata(i%, 5); ",";
qdata(i%, 6); _
","; qdata(i%, 7); ","; qdata(i%, 8); ","; qdata(i%, 9); ",";
qdata(i%, 10)
    qtime = qtime + 0.025
Next i%
Close #1
frmDocument.Text1 = "Closing Program"

End

End Sub

Sub nextsfpgm()
    Call busy
    Call loading

    If colnumber = 2 Then
        coilwrite = 116
        coilread = 117
    Else
        End If
    If colnumber = 5 Then

```

```

        coilwrite = 118
        coilread = 119
    Else
        End If
    If colnumber = 8 Then
        coilwrite = 120
        coilread = 100
    Else
        End If
    Call busy
    frmMain.E25721.Function = 5
    frmMain.E25721.MemType = 12
    frmMain.E25721.DataWords = 1
    frmMain.E25721.InfoIndex = 0
    frmMain.E25721.RandomAddr = coilwrite
    frmMain.E25721.Info = 3
    frmMain.E25721.Refresh
    Call busy
60    frmMain.E25721.Function = 4
    frmMain.E25721.MemType = 12
    frmMain.E25721.DataWords = 1
    frmMain.E25721.InfoIndex = 0
    frmMain.E25721.RandomAddr = coilread
    frmMain.E25721.Refresh
    Call busy
    frmMain.E25721.InfoIndex = 0
    If frmMain.E25721.Info < 1 Then
        GoTo 60
    Else
        End If
End Sub

Sub busy()
10  If frmMain.E25721.busy = True Then
        DoEvents
        GoTo 10
    Else
        End If
End Sub

Sub loading()
20  If connect = 1 Then
        DoEvents
        GoTo 20
    Else:
        End If
End Sub

Sub DECODE()
'Module A
If fault = 1000 Then Ffault = "Lead A Permit" Else
If fault = 1001 Then Ffault = "Lead Transition A Permit" Else
If fault = 1002 Then Ffault = "Chimney Lead A Permit" Else
If fault = 1003 Then Ffault = "Chimney Lead B Permit" Else
If fault = 1004 Then Ffault = "Lead Transition B Permit" Else
If fault = 1005 Then Ffault = "Lead B Permit" Else
If fault = 1006 Then Ffault = "Center Tap Imbalance Permit" Else
If fault = 1007 Then Ffault = "Quarter Tap Imbalance Permit" Else

'Module B
If fault = 2000 Then Ffault = "Control System Permit" Else
If fault = 2001 Then Ffault = "Accelerator Permit" Else

```

```

If fault = 2002 Then Ffault = "LCW Permit" Else
If fault = 2003 Then Ffault = "Crash Button Permit" Else
If fault = 2004 Then Ffault = "Steel Limit Switch's Permit" Else
If fault = 2005 Then Ffault = "Cooling Water Flow Permit" Else
If fault = 2006 Then Ffault = "Temperature Permit" Else
If fault = 2007 Then Ffault = "DC Filter Permit" Else

'Module C
If fault = 3000 Then Ffault = "Dump Switch Permit" Else
If fault = 3001 Then Ffault = "Ground Fault Permit" Else
If fault = 3002 Then Ffault = "Reversing Switch Permit" Else
If fault = 3003 Then Ffault = "Current Transducer Permit" Else
If fault = 3004 Then Ffault = "DC Solenoid Current Limit Permit" Else
If fault = 3005 Then Ffault = "Rack Smoke Detector Permit" Else
If fault = 3006 Then Ffault = "DC P.S. Security Gate Permit" Else
If fault = 3007 Then Ffault = "Cryogenic System Permit" Else
If fault = 0 Then Ffault = "No First Fault Detected" Else

End Sub

```

E. FAST QUENCH DATA Visual Basic Form Code

```

Private Sub E25721_OutCome(ByVal Result As Integer)

'This section of code is executed after every
'E2572 activeX call is finished.
'This checks the error status of the E2572 call.

If E25721.Result <> 0 Then
    If E25721.Result = -1 Then
        comerror = "Error response from the PLC"
    ElseIf E25721.Result = -2 Then
        comerror = "Error response from the CTI 2572 module"
    ElseIf E25721.Result = 32001 Then
        comerror = "Invalid Function code received"
    ElseIf E25721.Result = 32002 Then
        comerror = "ECC mismatch for received response"
    ElseIf E25721.Result = 32003 Then
        comerror = "Communication timeout occurred"
    ElseIf E25721.Result = -2 Then
        comerror = "Error response from the CTI 2572 module"
    ElseIf E25721.Result = 32004 Then
        comerror = "Error occurred in Socket operations"
    ElseIf E25721.Result = 32005 Then
        comerror = "WinSock DLL can not be initialized"
    ElseIf E25721.Result = 32006 Then
        comerror = "PortNumber supplied is below 1024"
    ElseIf E25721.Result = 32007 Then
        comerror = "Error occurred while connecting to specified IPAddr
and PortNumber"
    ElseIf E25721.Result = 32008 Then
        comerror = "Message length exceeded the limit"
    ElseIf E25721.Result = 32009 Then
        comerror = "Can not execute function without connection to CTI
2572 module"
    End If
Else
    comerror = "Successful Communication"
End If
frmDocument.Text4.Text = comerror

```


'This section loads the Array Qdata with the current
'data in the E2572 block of 200 vmems.

```
If connect = 1 Then
    Q% = 0
    For m% = 0 To 198 Step 2
        frmMain.E25721.InfoIndex = m%
        qdata(rowcount + Q%, colnumber) = frmMain.E25721.InfoFloat
        frmDocument.Text7 = qdata(rowcount, colnumber)
        Q% = Q% + 1
    Next m%
    rowcount = rowcount + 100
    frmDocument.Text2 = colnumber
    frmDocument.Text3 = rowcount
    frmDocument.Text5 = frmMain.E25721.TransactionTime
Else
End If
connect = 4

End Sub

Private Sub MDIForm_Load()
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
    LoadNewDoc
End Sub

Private Sub LoadNewDoc()
    Static lDocumentCount As Long
    Dim frmD As frmDocument

    lDocumentCount = lDocumentCount + 1
    Set frmD = New frmDocument
    frmD.Caption = "Document " & lDocumentCount
    frmD.Show
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As ComctlLib.Button)

    Select Case Button.Key

        Case "New"
            LoadNewDoc
```

```

Case "Open"
    'To Do
    MsgBox "Open Code goes here!"
Case "Save"
    'To Do
    MsgBox "Save Code goes here!"
Case "Print"
    'To Do
    MsgBox "Print Code goes here!"
Case "Cut"
    'To Do
    MsgBox "Cut Code goes here!"
Case "Copy"
    'To Do
    MsgBox "Copy Code goes here!"
Case "Paste"
    'To Do
    MsgBox "Paste Code goes here!"
Case "Bold"
    'To Do
    MsgBox "Bold Code goes here!"
Case "Italic"
    'To Do
    MsgBox "Italic Code goes here!"
Case "Underline"
    'To Do
    MsgBox "Underline Code goes here!"
Case "Left"
    'To Do
    MsgBox "Left Code goes here!"
Case "Center"
    'To Do
    MsgBox "Center Code goes here!"
Case "Right"
    'To Do
    MsgBox "Right Code goes here!"
End Select
End Sub

```

V. SLOW DUMP DATA

A. Slow Dump Data Parameters

All eleven Slow Dump Data points are stored every 100mS or at a rate of 10Hz for a total of 3000 data values/point. These data points are stored continuously in the TI555 variable memory for a total time of 300 seconds. The PLC programming is configured such that after the power supply turns off, the rotating buffer continues recording for 270 more seconds. The end result is after a dump all eleven points are recorded for 30 seconds before the event, up to 270 seconds after the event.

B. Slow Dump Data Ladder Logic

```

!                                     TEXT3-----+
!                                     !
!                                     ! THIS SECTION OF LADDER LOGIC CONTROLS: !
!                                     !
562 [-----+ ! 1. SLOW DUMP DATA ROTATING MEMORY !
!                                     !
!                                     ! 2. SDATA SFPGM'S (INTEGER TO REAL CONV) !
!                                     !
!                                     ! 3. POST DUMP SDATA ROTATING MEMORY RESET !
!                                     !
!                                     +-----+
!
! PS On/Off      DS Stat      SDATA
! Input          Input          TRIGGER
! 1=On           1=Closed
! X164           2      X174      C106
641 [---]/[-----:O:-----] [----- (RST )
!
! SDATA          SDATA
! RESET          TRIGGER
!
! C107           C106
647 [-] [-----*-- (SET )
!
! SDATA
! SFPGM #6
! TRIGGER
! C122
! -- (RST )
!
! SDATA
! SFPGM #7
! TRIGGER
! C124
! -- (RST )
!
! SDATA
! SFPGM #8
! TRIGGER
! C126
! -- (RST )
!
! SDATA
! RESET
!
! C107
+-- (RST )

```

```

!SDATA          TMR101-----+
!AQUIRE        !          !
!FINISHED       !          !
!  C109         !          !
663  [---] [-----] !          !----- C108
!          ! P= 300.0 !          !
!          !          !          !
!          !          !          !
!          !          !          !
!          !          !          !
!          !          !          !
!SDATA          !          !
!AQUIRE        !          !
!FINISHED       !          !
!  C109         !          !
[---] [-----] !          !
!          !          !

```

```

!SDATA          TMR102-----+
!TRIGGER        !          !
!  C106         !          !
668  [--]/[-----] !          !----- C110
!          ! P= 270.0 !          !
!          !          !          !
!          !          !          !
!          !          !          !
!          !          !          !
!          !          !          !
!SDATA          !          !
!TRIGGER        !          !
!  C106         !          !
[--]/[-----] !          !
!          !          !

```

```

!Dummy  ADD11-----+
!          !          !
!          ! SDATA    !
!  C1      ! LADDER SCAN !
673  [-] [-*-- COUNTER !----- C1
!          ! A:V105    !
!Dummy!    !          !
!          !          !
!          !          !
!  C1      !          !
[-]/[-+    ! B:+1      !
!          !          !
!          ! SDATA    !
!          ! LADDER SCAN !
!          ! COUNTER   !
!          ! C:V105    !
!          !          !
!          !          !

```


		MOVE28-----+	MOVE29-----+	Dummy
		! WORD	! WORD	
		! Qtr Tap	! MagCurr-2	
>		! Imbal Volts	! Holec Out	C1
a-----		!-----	!-----	()
>		!TS:WX24	!TS:WX41	
		!	!	
		!	!	
		!	!	
		!SI:	!SI:	
		! SDATA	! SDATA	
		! WX24 STORE	! WX41 STORE	
		! BEGINNING	! BEGINNING	
		!TD:V117000	!TD:V120000	
		!	!	
		!DI:V106	!DI:V106	
		! N=+1	! N=+1	
		+-----+	+-----+	
	!SDATA	SFPGM5-----+		SDATA
	!RECORD OR	!		SFPGM #5
	!DUMP STAT	!		FINISHED
	! C110	!		C121
794	[---] [-----]	!-----		()
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	+		
	!SDATA	SFPGM9-----+		Dummy
	!RECORD OR	!		
	!DUMP STAT	!		
	! C110	!		C1
797	[---] [-----]	!-----		()
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	+		
	!SDATA	SFPGM6-----+		SDATA
	!SFPGM #6	!		SFPGM #6
	!TRIGGER	!		FINISHED
	! C122	!		C123
800	[---] [-----]	!-----		()
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	!		
	!	+		

```

803      !SDATA          SFPGM7-----+          SDATA
      !SFPGM #7      !          SFPGM #7
      !TRIGGER      !          FINISHED
      !  C124      !          C125
      [---] [---] !          -----(      )
      !          !
      !          !
      !          !
      !          !
      !          !
      !          +-----+
      !SDATA          SFPGM8-----+          SDATA
      !SFPGM #8      !          SFPGM #8
      !TRIGGER      !          FINISHED
      !  C126      !          C127
806      [---] [---] !          -----(      )
      !          !
      !          !
      !          !
      !          !
      !          !
      !          +-----+
      !          !
      !          !

```


C. Slow Dump Data Special Function Logic

TITLE: SDATARL1

SF PROGRAM 5

```
-----
00001 *      ***THIS SFPGM SCALES AND MOVES FAST DATA TO REAL V MEMORY***
              SAMPLE 1 USES INTEGER VMEM BETWEEN V103000 AND V105999
              SAMPLE 2 USES INTEGER VMEM BETWEEN V106000 AND V108999
              V106 IS INDEXER
              V108 IS THE REAL NUMBER INDEXER
              SAMPLE 1 WILL USE REAL VMEM BETWEEN V73000 AND V78999
              SAMPLE 2 WILL USE REAL VMEM BETWEEN V79000 AND V84999
00002 *      THIS CONDITIONS THE FIRST 5 FAST DATA REAL TAGS
00003 IMATH   V107 := V106 + 1
00004 IMATH   V108 := 1
00005 LABEL   LABEL.....: 1
00006 IF      V107 >= 3001
00007 IMATH   V107 := 1
00008 ELSE
00009 ENDIF
00010 SCALE BINARY INPUT..: (V90000(V107)) SCALED RESULT.: V22600.(V108)
              LOW LIMIT.....: -0.1                HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO                  BIPOLAR.....: NO
00011 SCALE BINARY INPUT..: (V93000(V107)) SCALED RESULT.: V28600.(V108)
              LOW LIMIT.....: -0.1                HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO                  BIPOLAR.....: NO
00012 SCALE BINARY INPUT..: (V96000(V107)) SCALED RESULT.: V34600.(V108)
              LOW LIMIT.....: -0.1                HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO                  BIPOLAR.....: NO
00013 IMATH   V107 := V107 + 1
00014 IMATH   V108 := V108 + 2
00015 IF      V108 < 6001
00016 GOTO     LABEL.....: 1
00017 ELSE
00018 ENDIF
***** END *****
```

TITLE: SDATARL2

SF PROGRAM 6

```
-----
00001 *      THIS CONDITIONS THE NEXT 1 FAST DATA REAL TAGS
00002 IMATH   V107 := V106 + 1
00003 IMATH   V108 := 1
00004 LABEL   LABEL.....: 1
00005 IF      V107 >= 3001
00006 IMATH   V107 := 0
00007 ELSE
00008 ENDIF
00009 SCALE BINARY INPUT..: (V99000(V107)) SCALED RESULT.: V22600.(V108)
              LOW LIMIT.....: -0.1                HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO                  BIPOLAR.....: NO
00010 SCALE BINARY INPUT..: (V102000(V107)) SCALED RESULT.: V28600.(V108)
              LOW LIMIT.....: -0.1                HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO                  BIPOLAR.....: NO
00011 SCALE BINARY INPUT..: (V105000(V107)) SCALED RESULT.: V34600.(V108)
              LOW LIMIT.....: -0.1                HIGH LIMIT.....: 0.1
              20% OFFSET.....: NO                  BIPOLAR.....: NO
00012 IMATH   V107 := V107 + 1
00013 IMATH   V108 := V108 + 2
00014 IF      V108 < 6001
00015 GOTO     LABEL.....: 1
00016 ELSE
00017 ENDIF
***** END *****
```

```
00001 *          THIS CONDITIONS THE NEXT 2 FAST DATA REAL TAGS
00002 IMATH      V107 := V106 + 1
00003 IMATH      V108 := 1
00004 LABEL      LABEL.....: 2
00005 IF         V107 >= 3001
00006 IMATH      V107 := 0
00007 ELSE
00008 ENDIF
00009 SCALE BINARY INPUT...: (V108000(V107))SCALED RESULT.: V22600.(V108)
          LOW LIMIT.....: -10.0          HIGH LIMIT.....: 10.0
          20% OFFSET.....: NO             BIPOLAR.....: NO
00010 SCALE BINARY INPUT...: (V111000(V107)) SCALED RESULT.: V28600.(V108)
          LOW LIMIT.....: -10.0          HIGH LIMIT.....: 10.0
          20% OFFSET.....: NO             BIPOLAR.....: NO
00011 SCALE BINARY INPUT...: (V114000(V107))SCALED RESULT.: V34600.(V108)
          LOW LIMIT.....: -0.2           HIGH LIMIT.....: 0.2
          20% OFFSET.....: NO             BIPOLAR.....: NO
00012 IMATH      V107 := V107 + 1
00013 IMATH      V108 := V108 + 2
00014 IF         V108 < 6001
00015 GOTO        LABEL.....: 2
00016 ELSE
00017 ENDIF
**** END ****
```

```
00001 *          THIS CONDITIONS THE NEXT 1 FAST DATA REAL TAGS
00002 IMATH      V107 := V106 + 1
00003 IMATH      V108 := 1
00004 LABEL      LABEL.....: 3
00005 IF         V107 >= 3001
00006 IMATH      V107 := 0
00007 ELSE
00008 ENDIF
00009 SCALE BINARY INPUT...: (V117000(V107))SCALED RESULT.: V22600.(V108)
          LOW LIMIT.....: -0.4           HIGH LIMIT.....: 0.4
          20% OFFSET.....: NO             BIPOLAR.....: NO
00010 SCALE BINARY INPUT...: (V120000(V107))SCALED RESULT.: V28600.(V108)
          LOW LIMIT.....: -5000.0        HIGH LIMIT.....: 5000.0
          20% OFFSET.....: NO             BIPOLAR.....: YES
00011 IMATH      V107 := V107 + 1
00012 IMATH      V108 := V108 + 2
00013 IF         V108 < 6001
00014 GOTO        LABEL.....: 3
00015 ELSE
00016 ENDIF
**** END ****
```

D. Slow Dump Data Visual Basic Module Code

```
'
'This Visual Basic 5.0 Program was written by Dan Markley
' and is the Property of FermiLab. January of 1998.
'This program retrieves Slow Dump data that was stored in
' Dzero Superconducting Solenoid Control System TI555
' Programmable Logic Controller. It stores all the data in
' a disk file with todays date and time as a file name.
'This data is stored as a real number(in PLC 2 16 bit Vmems)
'There are 11 parameters with 3000 (6000 Vmems) values per 300 seconds
each.
```

Description	PLC Integer Address	PLC Real Address
#0 Lead A Voltage(WX1)	V90000-V92999	V22600-V28599
#1 Lead B Voltage(WX4)	V93000-V95999	V28600-V34599
#2 Transition A Voltage(WX7)	V96000-V98999	V34600-V40599
#3 Transition B Voltage(WX10)	V99000-V101999	V22600-V28599
#4 Chimney A Voltage(WX13)	V102000-V104999	V28600-V34599
#5 Chimney B Voltage(WX16)	V105000-V107999	V34600-V40599
#6 Outer Coil Voltage(WX19)	V108000-V110999	V22600-V28599
#7 Inner Coil Voltage(WX20)	V111000-V113999	V28600-V34599
#8 CenterTap Imbalance Voltage(WX21)	V114000-V116999	V34600-V40599
#9 Quarter Tap imbalance Voltage(WX24)	V117000-V119999	V22600-V28599
#10 Magnet Current(WX41)	V120000-V122999	V28600-V34599

```
Public connect As Integer
Public coilwrite As Integer
Public coilread As Integer
Public qdata(3000, 11) As Single
Public colnumber As Long
Public currstartadd As Long
Public rowcount As Long
Public comerror As String
Public fault As Integer
Public Ffault As String
```

```
Sub Main()
    frmDocument.Text1 = "Initailizing Connection to Solenoid PLC"
    frmDocument.Text4 = "None"
    connect = 0
    Call busy
    'Opening a Connection over Ethernet to PLC
    frmMain.E25721.Function = 0
    frmMain.E25721.Refresh

    Call busy
    frmDocument.Text1 = "Retrieving Data from Solenoid PLC"
    connect = 4
    rowcount = 0
    '11 colnumber Loops for 11 parameters
    For colnumber = 0 To 10 Step 1
        If colnumber = 0 Then startadd = 22600 Else
        If colnumber = 1 Then startadd = 28600 Else
```

```

If colnumber = 2 Then startadd = 34600 Else
If colnumber = 3 Then startadd = 22600 Else
If colnumber = 4 Then startadd = 28600 Else
If colnumber = 5 Then startadd = 34600 Else
If colnumber = 6 Then startadd = 22600 Else
If colnumber = 7 Then startadd = 28600 Else
If colnumber = 8 Then startadd = 34600 Else
If colnumber = 9 Then startadd = 22600 Else
If colnumber = 10 Then startadd = 28600 Else
    For currstartadd = startadd To startadd + 5800 Step 200
        Call busy
        Call loading
        connect = 1
        If rowcount = 3000 Then rowcount = 0 Else
        'Pulling 200 VMEMS from PLC
        frmMain.E25721.Function = 8
        frmMain.E25721.DataWords = 200
        frmMain.E25721.StartAddr = currstartadd
        frmMain.E25721.MemType = 0
        frmMain.E25721.Refresh
        frmDocument.Text6 = currstartadd
    Next currstartadd
    If colnumber = 2 Then Call nextsfpgm Else
    If colnumber = 5 Then Call nextsfpgm Else
    If colnumber = 8 Then Call nextsfpgm Else
    Call busy
    Call loading
Next colnumber
    frmDocument.Text2 = "data"

```

```

Call busy
frmDocument.Text1 = "Finished"
frmMain.E25721.Function = 5
frmMain.E25721.MemType = 12
frmMain.E25721.DataWords = 1
frmMain.E25721.InfoIndex = 0
frmMain.E25721.RandomAddr = 107
frmMain.E25721.Info = 3
frmMain.E25721.Refresh

```

'This read gets the first fault word

```

Call busy
frmMain.E25721.Function = 4
frmMain.E25721.MemType = 0
frmMain.E25721.DataWords = 1
frmMain.E25721.InfoIndex = 0
frmMain.E25721.RandomAddr = 110
frmMain.E25721.Refresh
Call busy
frmMain.E25721.InfoIndex = 0
fault = frmMain.E25721.Info
Call DECODE

```

```

Call busy
frmDocument.Text1 = "Closing Connection to Solenoid PLC"
connect = 3
frmMain.E25721.Function = 1
frmMain.E25721.DataWords = 50
frmMain.E25721.StartAddr = 10
frmMain.E25721.MemType = 0
frmMain.E25721.Refresh

```

'The next section creates a disk file with todays date and time.

```
frmDocument.Text1 = "Saving Data to Disk File"
Dim one, two As Variant
Dim yrtemp, motemp, dytemp, hrtemp, mintemp As Integer
Dim yr, mo, dy, today, hr, min As String
one = Date
two = Time
yrtemp = Year(one)
motemp = Month(one)
dytemp = Day(one)
hrtemp = Hour(two)
mintemp = Minute(two)
yr = Str(yrtemp)
mo = "Y" & Str(motemp)
dy = "M" & Str(dytemp)
hr = "D" & Str(hrtemp)
min = "H" & Str(mintemp) & "M"
today = "D:\QUENCHDATA\" & "Sol_SDATA" & yr & mo & dy & hr & min &
".csv"
Open today For Output As #1
Print #1, comerror
Print #1, "first fault was:", Ffault
If comerror <> "Successful Communication" Then
    Close #1
End
End If
Print #1, "Quench time,"; "Lead A Voltage,"; "Lead B Voltage,"; _
    "Transition A Voltage,"; "Transition B Voltage,"; "Chimney A
Voltage,"; _
    "Chimney B Voltage,"; "Outer Coil Voltage,"; "Inner Coil Voltage,";
-
    "Center Tap Imbalance Voltage,"; "Quarter Tap Imbalance Voltage,";
"Magnet Current"
Dim qtime As Single
qtime = -30#
For i% = 0 To 2999 Step 1
    Print #1, qtime; ", "; qdata(i%, 0); ", "; qdata(i%, 1); ", ";
qdata(i%, 2); _
    ", "; qdata(i%, 3); ", "; qdata(i%, 4); ", "; qdata(i%, 5); ", ";
qdata(i%, 6); _
    ", "; qdata(i%, 7); ", "; qdata(i%, 8); ", "; qdata(i%, 9); ", ";
qdata(i%, 10)
    qtime = qtime + 0.1
Next i%
Close #1
frmDocument.Text1 = "Closing Program"

End

End Sub

Sub nextsfpgm()
    Call busy
    Call loading

    If colnumber = 2 Then
        coilwrite = 122
        coilread = 123
    Else
```

```

        End If
    If colnumber = 5 Then
        coilwrite = 124
        coilread = 125
    Else
        End If
    If colnumber = 8 Then
        coilwrite = 126
        coilread = 127
    Else
        End If
    Call busy
    frmMain.E25721.Function = 5
    frmMain.E25721.MemType = 12
    frmMain.E25721.DataWords = 1
    frmMain.E25721.InfoIndex = 0
    frmMain.E25721.RandomAddr = coilwrite
    frmMain.E25721.Info = 3
    frmMain.E25721.Refresh
    Call busy
60   frmMain.E25721.Function = 4
    frmMain.E25721.MemType = 12
    frmMain.E25721.DataWords = 1
    frmMain.E25721.InfoIndex = 0
    frmMain.E25721.RandomAddr = coilread
    frmMain.E25721.Refresh
    Call busy
    frmMain.E25721.InfoIndex = 0
    If frmMain.E25721.Info < 1 Then
        GoTo 60
    Else
        End If
End Sub

Sub busy()
10  If frmMain.E25721.busy = True Then
        DoEvents
        GoTo 10
    Else
        End If
End Sub

Sub loading()
20  If connect = 1 Then
        DoEvents
        GoTo 20
    Else:
        End If
End Sub

Sub DECODE()
'Module A
If fault = 1000 Then Ffault = "Lead A Permit" Else
If fault = 1001 Then Ffault = "Lead Transition A Permit" Else
If fault = 1002 Then Ffault = "Chimney Lead A Permit" Else
If fault = 1003 Then Ffault = "Chimney Lead B Permit" Else
If fault = 1004 Then Ffault = "Lead Transition B Permit" Else
If fault = 1005 Then Ffault = "Lead B Permit" Else
If fault = 1006 Then Ffault = "Center Tap Imbalance Permit" Else
If fault = 1007 Then Ffault = "Quarter Tap Imbalance Permit" Else

```

```

'Module B
If fault = 2000 Then Ffault = "Control System Permit" Else
If fault = 2001 Then Ffault = "Accelerator Permit" Else
If fault = 2002 Then Ffault = "LCW Permit" Else
If fault = 2003 Then Ffault = "Crash Button Permit" Else
If fault = 2004 Then Ffault = "Steel Limit Switch's Permit" Else
If fault = 2005 Then Ffault = "Cooling Water Flow Permit" Else
If fault = 2006 Then Ffault = "Temperature Permit" Else
If fault = 2007 Then Ffault = "DC Filter Permit" Else

'Module C
If fault = 3000 Then Ffault = "Dump Switch Permit" Else
If fault = 3001 Then Ffault = "Ground Fault Permit" Else
If fault = 3002 Then Ffault = "Reversing Switch Permit" Else
If fault = 3003 Then Ffault = "Current Transducer Permit" Else
If fault = 3004 Then Ffault = "DC Solenoid Current Limit Permit" Else
If fault = 3005 Then Ffault = "Rack Smoke Detector Permit" Else
If fault = 3006 Then Ffault = "DC P.S. Security Gate Permit" Else
If fault = 3007 Then Ffault = "Cryogenic System Permit" Else
If fault = 0 Then Ffault = "No First Fault Detected" Else

End Sub

```

E. Visual Basic Form Code

```

Private Sub E25721_OutCome(ByVal Result As Integer)

'This section of code is executed after every
'E2572 activeX call is finished.
'This checks the error status of the E2572 call.

If E25721.Result <> 0 Then
    If E25721.Result = -1 Then
        comerror = "Error response from the PLC"
    ElseIf E25721.Result = -2 Then
        comerror = "Error response from the CTI 2572 module"
    ElseIf E25721.Result = 32001 Then
        comerror = "Invalid Function code received"
    ElseIf E25721.Result = 32002 Then
        comerror = "ECC mismatch for received response"
    ElseIf E25721.Result = 32003 Then
        comerror = "Communication timeout occurred"
    ElseIf E25721.Result = -2 Then
        comerror = "Error response from the CTI 2572 module"
    ElseIf E25721.Result = 32004 Then
        comerror = "Error occurred in Socket operations"
    ElseIf E25721.Result = 32005 Then
        comerror = "WinSock DLL can not be initialized"
    ElseIf E25721.Result = 32006 Then
        comerror = "PortNumber supplied is below 1024"
    ElseIf E25721.Result = 32007 Then
        comerror = "Error occurred while connecting to specified IPAddr
and PortNumber"
    ElseIf E25721.Result = 32008 Then
        comerror = "Message length exceeded the limit"
    ElseIf E25721.Result = 32009 Then
        comerror = "Can not execute function without connection to CTI
2572 module"
    End If
Else
comerror = "Successful Communication"

```

```

End If
frmDocument.Text4.Text = comerror

'This section loads the Array Qdata with the current
'data in the E2572 block of 200 vmems.

If connect = 1 Then
    Q% = 0
    For m% = 0 To 198 Step 2
        frmMain.E25721.InfoIndex = m%
        qdata(rowcount + Q%, colnumber) = frmMain.E25721.InfoFloat
        frmDocument.Text7 = qdata(rowcount, colnumber)
        Q% = Q% + 1
    Next m%
    rowcount = rowcount + 100
    frmDocument.Text2 = colnumber
    frmDocument.Text3 = rowcount
    frmDocument.Text5 = frmMain.E25721.TransactionTime
Else
End If
connect = 4

End Sub

Private Sub MDIForm_Load()
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
    LoadNewDoc
End Sub

Private Sub LoadNewDoc()
    Static lDocumentCount As Long
    Dim frmD As frmDocument

    lDocumentCount = lDocumentCount + 1
    Set frmD = New frmDocument
    frmD.Caption = "Document " & lDocumentCount
    frmD.Show
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As ComctlLib.Button)

    Select Case Button.Key

```



```

Case "New"
    LoadNewDoc
Case "Open"
    'To Do
    MsgBox "Open Code goes here!"
Case "Save"
    'To Do
    MsgBox "Save Code goes here!"
Case "Print"
    'To Do
    MsgBox "Print Code goes here!"
Case "Cut"
    'To Do
    MsgBox "Cut Code goes here!"
Case "Copy"
    'To Do
    MsgBox "Copy Code goes here!"
Case "Paste"
    'To Do
    MsgBox "Paste Code goes here!"
Case "Bold"
    'To Do
    MsgBox "Bold Code goes here!"
Case "Italic"
    'To Do
    MsgBox "Italic Code goes here!"
Case "Underline"
    'To Do
    MsgBox "Underline Code goes here!"
Case "Left"
    'To Do
    MsgBox "Left Code goes here!"
Case "Center"
    'To Do
    MsgBox "Center Code goes here!"
Case "Right"
    'To Do
    MsgBox "Right Code goes here!"
End Select
End Sub

```

VI. FIRST FAULT

```
TISOFT
DATE: 03-27-98      DZERO Solenoid Control PLC      555
DOSOL      RN

TITLE: FFault      SF PROGRAM      9

      CONTINUE ON ERROR (Y,N): YES
      ERROR STATUS ADDR (Y,C,WY,V):
      PROGRAM TYPE (N,P,C,R): NORMAL
      CYCLE TIME (SEC): 0.0
-----
00001 *      THIS SPECIAL FUNCTION BUILDS A NUMBER FOR DECODING THE
              FIRST FAULT DETECTION.
              THIS NUMBER IS THEN PULLED OUT BY THE DATA AQUISITION
              PROGRAM AND DECODED INTO TEXT AND STORED IN THE DATA FILE.
00002 IMATH   V110 := 0
00003 IF      X129 = 0
00004 IMATH   V110 := V110 + 4
00005 ENDIF
00006 IF      X130 = 0
00007 IMATH   V110 := V110 + 2
00008 ENDIF
00009 IF      X131 = 0
00010 IMATH   V110 := V110 + 1
00011 ENDIF
00012 IF      X132 = 1
00013 IMATH   V110 := V110 + 1000
00014 ENDIF
00015 IF      X133 = 1
00016 IMATH   V110 := V110 + 2000
00017 ENDIF
00018 IF      X134 = 1
00019 IMATH   V110 := V110 + 3000
00020 ENDIF
****      END      ****
```