

# **MLRS TCS Electronic Function Tester**

By

Larry Mattison

Submitted to  
the Faculty of the Information Engineering Technology Program  
in Partial Fulfillment of the Requirements for  
the Degree of Bachelor of Science  
in Information Engineering Technology

University of Cincinnati  
College of Applied Science

May 2004

# MLRS TCS Electronic Function Tester

by

Larry Mattison

Submitted to  
the Faculty of the Information Engineering Technology Program  
in Partial Fulfillment of the Requirements  
for  
the Degree of Bachelor of Science  
in Information Engineering Technology

© Copyright 2004 Larry Mattison

The information in this document is proprietary and may not be reproduced or distributed  
in whole or in part without the permission of the owner.

---

Larry Mattison

---

Date

---

Professor Annu Prabhakar, Faculty Advisor

---

Date

---

James F. Sullivan, Department Head

---

Date

## **Acknowledgements**

I would like to thank my boss for giving me the opportunity to do a project like this that allows me to utilize my computer skills gained at school on a project for work. I would also like to acknowledge all the tips received from the National Instruments Developer Zone website. Without the other programmers help, it would have taken me much longer to complete key components of my software.

# Table of Contents

## Section

Acknowledgements

Table of Contents

List of Figures

Abstract

1. Statement of the Problem	1
2. Description of the Solution	1
2.1 User Profile	2
2.1.1 Production	2
2.1.2 Engineering	3
2.1.3 Developer	3
2.1.4 Management	3
2.2 Design Protocols	3
2.2.1 Programming	3
2.2.2 Networking	4
2.2.3 Multimedia	4
3. Deliverables	4
4. Design and Development	5
4.1 GUI Design	5
4.1.1 Startup Screen	5
4.1.2 Login Screen	6
4.1.3 Main Test Screen	6
4.2 Data Storage	8
4.3 Print Out	9
4.4 Budget	10
4.5 Software and Hardware Requirements	10
4.6 Timeline	11
5. Proof of Design	12
5.1 Electronic MLRS TCS Function Tester.	12
5.2 User Interface Written in NI Labview.	13
5.3 Operator Can Test the TCS.	13
5.4 Tester Displays Real Time Test Data.	14
5.5 Data is Stored Locally and on the Network.	15
5.6 Software Tracks Oper, Date, Time, Assembly Level and Pass Fail Status.	15
5.7 The Software Prints a Record of Each Test.	16
5.8 Parameters Stored in a Text File.	17
5.9 Data Retrieval Program.	18
5.10 Additions.	19
6. Conclusion	19
References	20

## List of Figures

<b>Figure Number</b>	<b>Page</b>
Figure 1. Startup Screen	6
Figure 2. Main Test Screen	8
Figure 3. Print Out	9
Figure 4. Internal Wiring of the TCS Electronic Tester	13
Figure 5. TCS Setup Screen	15
Figure 6. Test Data Storage	16
Figure 7. Print Out	17
Figure 8. Histogram Analysis	18
Figure 9. Calibration Routine	19

## **Abstract**

*The MLRS TCS Electrical Function Tester* is a hardware and software system designed to test the functionality of the U.S. Army's MLRS TCS fuze. The Army contracted KDI Precision Products to manufacture and test the electronics that control when the MLRS missile detonates; this device is called a TCS fuze. The tester is designed to allow an operator to simulate missile in flight conditions on different assembly levels of the fuze. The fuze is connected to the function tester and the operator uses the test software to specify what tests they want to run.

The function tester also has the capability to allow users to access the stored test data on the hard drive and perform statistical analysis on it. The user can access specific files which are stored by shop order number and select different criteria such as pass, fail, date, test and get a graphical representation of the histogram data. They can upload a synopsis of this data to a drive on the LAN or print out summaries directly from the data retrieval software. With the combination of the testing capabilities and the data retrieval software, this tester is a valuable testing and troubleshooting tool for the MLRS TCS program.

# MLRS TCS Electronic Function Tester

## 1. Statement of the Problem

KDI Precision Products is one of only a few companies that design and manufacture electronic fuzes for the government. They were awarded a sub-contract to build the MLRS TCS Electronic Function Tester for Lockheed Martin. The challenge was to design and implement a functional tester that can communicate serially with the fuze and test the fuze at five different assembly levels. The tester must be able to do a variety of different measurements such as voltage, resistance, time and energy. This tester must also store data locally and to the network for statistical analysis of any failures that may occur.

## 2. Description of the Solution

The MLRS TCS Electronic Function Tester is an integrated system that includes a PC and GPIB measurement devices. The core of the system is the main test software designed with National Instruments Labview. Labview was chosen for the following reasons:

- Flexibility. When designing software that will interface with so many different devices, Labview gives the software developer the ability to rapidly change and implement new features that may be added by the test engineer.
- GUI Design. Labview has many built in controls and displays that allow the developer to minimize his or her time on GUI design while concentrating on the core code.
- Portability. Labview has the option of saving all the subroutines into one library file. This makes it easier for the developer to distribute and archive copies of the test software. Labview also has an auto installation feature which creates a

distribution application. This allows the application to automatically install in the proper directories for proper functionality.

- Compatibility. Many electronic measurement vendors support Labview and they release driver utilities for their devices. This allows the developer to integrate these routines into the software for full compatibility with the device they are using. Labview is also compatible with all Windows based operating systems.
- Debugging. Labview has excellent debugging features. Using these controls, the developer can visually watch where the data is flowing in the software. This is very useful when troubleshooting software problems.

## **2.1 User Profile**

There are four main users for the MLRS TCS Function Tester. The four levels are Production, Engineering, Developer and Management.

### **2.1.1 Production**

Production personnel are the primary users of the test equipment. Their job consists of starting the tester, inputting the manufacturing information and running the test equipment. They need to have a basic knowledge of computers with some previous experience. If a problem occurs, they need to be able to shut down the tester and the computer and restart the system. A high school degree or GED equivalent is helpful. They rely on instructions and their supervisors to perform the job.

### **2.1.2 Engineering**

Engineering personnel support and repair the hardware and the software of the tester. They must be able to diagnose any problems with the test equipment. Familiarity with the functionality of the fuse and of the software is also required. They are required to

give guidance to production on proper methods of testing and of the use of the test equipment. Typically reports to the program manager. An associate's degree is required.

### **2.1.3 Developer**

The developer's role is to add or change any of the test parameters. They must have thorough knowledge of the equipment such as the software and computer operating system. Their job is to edit and update the software any time management or engineering needs additional tests or if the current test requirements change. An associate's degree with computer software experience is required.

### **2.1.4 Management**

Management personnel only use the data they retrieve from the tester. They are interested in test throughput and failure analysis. They typically direct the engineer and production in regards to tester issues and problems such as scheduling conflicts. They must be familiar with the basic principles of the tester and the fuze. They also develop the plans and schedules for production. Generally reports to an executive.

## **2.2 Design Protocols**

I focused on two main areas within the Information Engineering Technology program for this project. The main focus was on programming with the rest dedicated to networking. There was also some multimedia included in the GUI design aspect of the project. Below are the different areas and their application in my project.

### **2.2.1 Programming**

I used Labview's graphical language 'G' to program the majority of the software. The program controls all the equipment connected to the computer. It also communicates

with the electronic fuze being tested. The program creates and stores the data file after the testing is complete.

I also use HP Basic to communicate with the measurement devices in the system. This is a modified version of BASIC that Hewlett Packard uses for all their test and measurement devices

### **2.2.2 Networking**

A key component of the electronic tester is its capability to upload data to the company LAN. The computer is running Windows 2000 with a network card. The software automatically uploads critical data and histogram information to a specific location on the LAN designated by management. This allows management to gain access to the real time data that the tester creates. Using this data, management can view histograms of different tests and see failure trends.

A secondary networking component is the setup and implementation of the GPIB network. This network allows the measurement devices to communicate with the main controller which is a PCI based card in the computer. These devices use the HP Basic language to communicate with the software and routines were written to initialize the instruments, measure the data and retrieve the data.

### **2.2.3 Multimedia**

When designing a tester GUI for our production environment, it was critical to make it as user friendly as possible. A simple screen with intuitive controls was essential. The GUI also displays needed information for engineering when required. The GUI was designed using National Instruments programming software Labview.

## **3. Deliverables**

1. An electronic MLRS TCS Function Tester.
2. The user interface is written in National Instruments Labview.
3. The operator is able to test the TCS electronic fuze.
4. The software displays real time test data and results.
5. The software uploads the data to the LAN and to the local drive.
6. The software tracks operator, date, time, assembly level and pass fail status.
7. The software prints a record of the test data after each run.
8. The software gets the test criteria from a separate ini file that can be manipulated with a text editor.
9. A separate application is included which allows managers to:
  - Download the data from the LAN.
  - Select individual tests and get histogram results.
  - Display the histogram data in a graphical format.
  - Sort the data by date, operator, pass or fail.

#### **4. Design and Development**

The MLRS TCS Function Tester is a flexible test setup that allows the operator to test five different assembly levels with different tests run at each level. The tester also displays pass fail status, shop order number, operator, date, time and various test data. Once the data is gathered, the tester then stores the data to the hard drive and the LAN. The critical areas of design are the GUI, data storage and print out.

##### **4.1 GUI Design**

###### **4.1.1 Startup Screen**

This screen starts up when the computer is powered up. It has a built in five minute delay that waits for all the test equipment to warm up. During this wait, the software

opens up a comma delimited initialization file that I created with Wordpad. Stored in this file are all the test parameters and tester information such as part number and software revision. The software puts these strings into global variables for use during the test. The software then performs a self test on all the equipment to verify they pass self test. After the self test, the screen prompts the user to verify the correct date and time and gives them the option to change it if necessary. See figure 1.



**Figure 1. Startup Screen**

#### **4.1.2 Login Screen**

After the operator verifies the correct date and time, a login screen appears. This window is a small pop up type screen that prompts the operator to enter their name, shop order, serial number and any comments. It also displays the date and time on the top of the window.

#### **4.1.3 Main Test Screen**

This is the main test screen that the operator sees. The top of the screen is the title of the tester, MLRS TCS Tester. This screen also displays the tester ID number, the tester

part number and the software revision. These fields are populated from information that the software retrieves from a comma delimited initialization file. This screen also has controls allowing the operator to select test temperature, what level of test to perform, printing options, a stop on failure option, a start button and a quit button.

In the center of the Main Test Screen are the pass fail indicators and the data fields. The indicators are split up into sixteen different tests which may include more than one measurement per test. As the test progresses the indicator turns black indicating the start of the test. After the measurement has been performed and the software calculates the pass fail status, the indicator lights up red or green and the data fields containing the measurement changes color accordingly. After all sixteen tests are performed a large indicator at the bottom of the screen lights up red or green to notify the operator the status of the fuze. If the operator chooses different test levels, the tests that aren't required at that level will be skipped. The indicators and the data fields that are being skipped are also dimmed out. See figure 2.

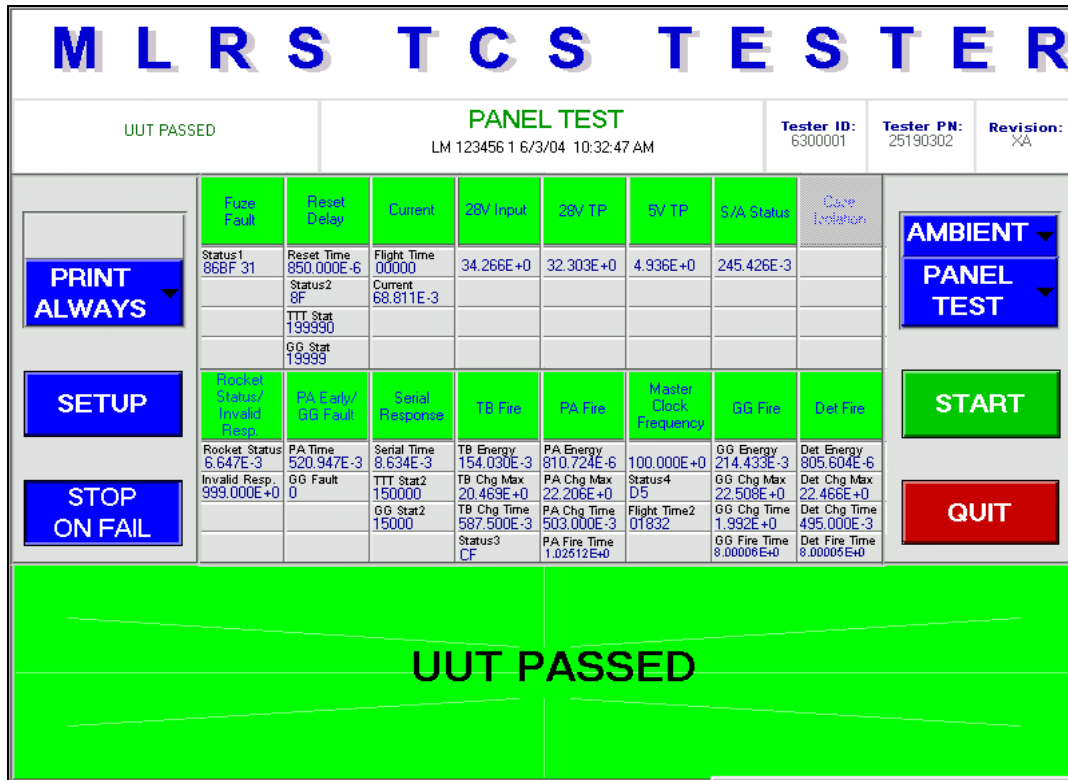


Figure 2. Main Test Screen

#### 4.2 Data Storage

At the end of the test, the computer creates a comma delimited text file. The name of the file will be the shop order number and the file is stored in a test level subdirectory under a data subdirectory. The program takes all the global variable test data, converts the data from numerical to string and writes the data to the text file. The file will also contain the operator, data time and a pass fail string consisting of 1's and 0's to indicate the status of each test.

The software also connects to the LAN and writes an identical file to a directory assigned to the tester. If no LAN connection is present, the software detects this and skips the storage to the network without interrupting the production operator using the equipment. A separate utility resides on the engineer's or manager's machine which

allows access to the data and gathers histogram information. The manager also has the ability to import the comma delimited file into Excel for spreadsheet manipulation.

### 4.3 Print Out

The computer is connected to a laser printer that prints out the test results of each run. The print out contains the serial number, shop order, revision, date and time, tester ID, operator, temperature, assembly and file location of the data. The rest of the print out is the test descriptions followed by the measured data and the pass fail status. At the bottom of the print out is a final pass fail status indicating if the fuze passed all the tests or not. See figure 3.

MLRS TCS PANEL TEST DATA SHEET				Date&Time: 6/3/2004 10:18:29 AM
Serial #: 1	Shop Order: 123456	Rev: XA	File Location: C:\testdata.dat	Tester ID: 6300001 P.N. 25190302
Operator: jm	Temp: Ambient	Assembly: 25190003		
Fuze Fault (Hex) 86BF1 p	S/A Status (Ohm) 242.41600E-3 p	PA Energy (Joules) 586.50000E-3 p	GG Fire Time (sec) 22.50800E+0 p	
Reset Time (sec) 1.30000E-3 p	Rocket Status (Ohm) 7.39900E-3 p	PA Chg Max (Volt) 1CF p	Det Energy (Joules) 1.97900E+0 p	
Status 2 (Hex) 8F p	Invalid Response Time (sec) 999.00000E+0 p	PA Chg Time (sec) 810.19380E-6 p	Det Chg Max (Volt) 8.00005E+0 p	
TTT Stat (sec) 199990 p	Serial Response Time (sec) 520.88229E-3 p	PA Fire Time (sec) 22.21700E+0 p	Det Chg Time (sec) 805.52240E-6 p	
GG Stat (sec) 19999 p	TTT Stat 2 (sec) 0.00000E+0 p	Master Clock Frequency (Hz) 501.00000E-3 p	Det Fire Time (sec) 22.46600E+0 p	
Flight Time (sec) 00000 p	GG Stat 2 (sec) 8.59375E-3 p	Status 4 (Hex) 1.02511E+0 p		
Current (Amp) 68.82131E-3 p	TB Energy (Joules) 150000 p	Flight Time 2 (sec) 100.00030E+0 p	Comments:	
28V input (Volt) 34.26833E+0 p	TB Chg Max (Volt) 15000 p	GG Energy (Joules) 1D5 p		
28V TP (Volt) 32.30520E+0 p	TB Chg Time (sec) 153.89534E-3 p	GG Chg Max (Volt) 01833 p		
5V TP (Volt) 4.93657E+0 p	Status 3 (Hex) 20.62500E+0 p	GG Chg Time (sec) 214.53519E-3 p		
<b>UUT PASSED</b>				

Figure 3. Print Out

#### 4.4 Budget

	240 Man Hours (Engineer 1)	\$ 8000.00
	Computer (2.4G Pentium)	555.00
Hardware	LS2208 Bar Code Reader	190.00
	GPIB PCI Card	795.00
	HP 6612C Power Supply	1050.00
	HP 53131A Counter	2051.00
	HP 34907 DAQ/DMM unit	1477.00
	HP 54645 Scope	3000.00
Software	NI Labview	995.00
	NI Labview App Builder	995.00
	Microsoft Windows 2000	260.00
	Norton Antivirus Corporate	41.97
	<b>Project Total</b>	<b>\$19,409.97</b>

#### 4.5 Software and Hardware Requirements

##### Software Requirements

- National Instruments Labview
- National Instruments Application Builder
- Microsoft Windows 2000
- Norton Antivirus Corporate

## Hardware Requirements

- Computer
- Bar Code Reader
- GPIB PCI Card
- Power Supply
- Counter
- DMM
- Oscilloscope

## 4.6 Timeline

### **Senior Design 1 Fall 2003**

#### Weeks 1-5

- Order test equipment.
- Read test requirement document.
- Install Windows 2000 and Labview.

#### Weeks 6-10

- Begin design of GUI.
- Begin writing storage routines.
- Begin writing fuze communication routine.

### **Senior Design II Winter 2004**

#### Weeks 1-5

- Finalize storage routines.
- Finalize fuze communication routine.
- Install equipment into table.
- Begin wiring harness for equipment.

Weeks 6-10

- Incorporate harness into DMM/Switch unit.
- Write self test routines for each device.
- Write communication routines for each device.
- Write routines to gather data from scope, counter and DMM.
- Demonstrate working prototype.

### **Senior Design III Spring 2004**

Weeks 1-5

- Complete testing and troubleshooting.
- Complete operating instructions.
- Complete calibration instructions.

Final Week

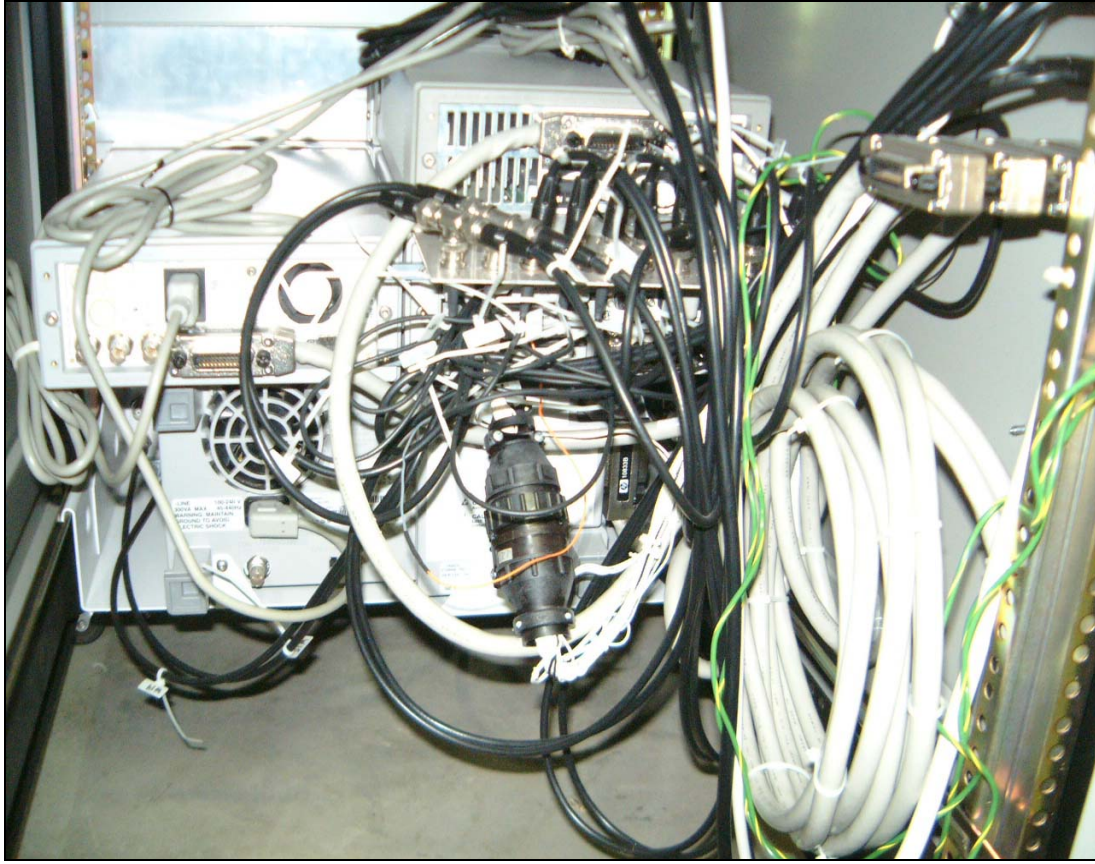
- Submit final documentation and perform presentation.

## **5. Proof of design**

This section will describe how deliverables of the project were fulfilled.

### **5.1 Electronic MLRS TCS Function Tester.**

The MLRS TCS Function tester is a combination of software and electronics. It consists of a computer, a printer, a scanner, a work desk, and various electronic components. The computer communicates with the electronics by a GPIB controller card installed in a PCI slot. The computer also communicates with the TCS fuze by a RS232 serial interface. The tester had to be built from the ground up with all the mounting hardware and wiring added. See Figure 4. After assembly, the tester had to be powered up and every piece of equipment had to run through a self check to verify it was in working order.



**Figure 4. Internal Wiring MLRS TCS Electronic Function Tester**

### **5.2 User Interface Written in NI Labview.**

The User Interface was written in National Instruments Labview primarily because our company is trying to standardize the software we use for our test equipment and Labview was chosen because of its ability to decrease the turnaround time it takes when creating and modifying the tester software. Revision changes and test additions can be accomplished in a matter of hours so the downtime to production is kept at a minimum.

### **5.3 Operator Can Test the TCS.**

The ability for the operator to test the TCS fuze may sound obvious, but it must be noted that most of KDI's production staff has a bare minimum of computer experience. Therefore, the tester had to be designed with usability in mind. The desk that was

ordered was specifically purchased with the height of the added wheels in mind so that an operator sitting at the desk for eight hours wouldn't become fatigued. The barcode reader was ordered so the operator didn't have to input the serial number of each unit manually; they could swipe the parts under the scanner. The laser printer was purchased so that the down time of changing out print cartridges would be minimized. In theory, the toner cartridge, while more expensive, should outlast a standard ink cartridge. The GUI was written so that all the control buttons on the screen were the same color (blue) and relatively the same size. The only two exceptions to the color scheme were the start and stop which were green and red respectively. The background of the GUI is a neutral grey color and the tester information such as operator, revision, etc., is highlighted by a white background. All these factors contribute to the functionality of the test setup which lessens the time it takes for the operators to learn how to operate the tester.

#### **5.4 Tester Displays Real Time Test Data.**

After the operator starts the test, the indicator which shows which portion of the test the software is on lights black. The software will then communicate with the TCS fuse and put it in the required mode for that part of the test. Then the software initiates the test by applying any signals or loads that the TCS requires and waits until it sees that the electronics performing that particular measurement is done. The software then asks the electronics what that measurement was and performs a limit check on it. If the measurement is a huge number (which occurs often if the TCS is not connected), the software substitutes a 999 into that field. After each measurement of that portion of the test is complete, the software writes that to the main test screen. The indicator of that portion of the test is then colored red or green depending of if all parts of that test passed or failed.

## 5.5 Data is Stored Locally and on the Network.

There is a setup button on the main test screen that allows the engineer to change where the data is stored. This information is gathered from a text ini file that the software opens when it is first started. The software reads each parameter then writes those values to global variables. When the software finishes running the test, it checks the global variable for the storage location and writes the data to that directory. See figure 5.

The screenshot displays the 'MLRS TCS SETUP SCREEN' with various configuration options. At the top, there are buttons for 'RETURN TO DEFAULT VALUES' (yellow), 'EXIT WITHOUT UPDATING' (white), and 'EXIT & UPDATE' (red). To the right, there is a 'Set Date Time' section with a date field set to '6/3/04' and a time field set to '10:44:07 AM', along with a 'Print From Data File' button. Below these are sections for 'Display Options' (with 'Stop On Failure' set to 'NO'), 'Print Options' (with 'CHANGE' and 'always'), and 'Storage Options' (with 'CHANGE' and 'local'). There are also 'Change Local Location' and 'Change Network Location' buttons, with corresponding text boxes showing 'C:\TCS Tester\Data' and 'p:\TCS\data' respectively. At the bottom, a 'CONFIG FILE' section shows a list of parameters: [TESTER SPECIFICS] (tester=MLRS TCS Tester, part number=25190302, rev.=XA), [PRINT SETUP] (print=always), and [STORAGE] (storage=local, local location=C:\TCS Tester\Data, network location=p:\TCS\data).

**Figure 5. TCS Setup Screen**

## 5.6 Software Tracks Operator, Date, Time, Assembly Level and Pass Fail Status.

The software displays the operator, date, time, assembly level, pass fail status, revision and tester id on the main screen. After each test, this data along with the actual test data is written to the hard drive and the network. The file is a delimited text file that can be imported easily into Excel. See Figure 6.



MLRS TCS PANEL TEST DATA SHEET				Date&Time:	6/3/2004 10:18:29 AM		
Serial #:	1	Shop Order:	123456	Rev:	XA		
Operator:	jm	Temp:	Ambient	Assembly:	25190003		
				File Location:	C:\testdata.dat		
				Tester ID:	6300001		
				P.N.:	25190302		
Fuze Fault (Hex)	86BF1	S/A Status (Ohm)	242.41600E-3	PA Energy (Joules)	586.50000E-3	GG Fire Time (sec)	22.50800E+0
Reset Time (sec)	1.30000E-3	Rocket Status (Ohm)	7.39900E-3	PA Chg Max (Volt)	1CF	Det Energy (Joules)	1.97900E+0
Status 2 (Hex)	8F	Invalid Response Time (sec)	999.00000E+0	PA Chg Time (sec)	810.19380E-6	Det Chg Max (Volt)	8.00005E+0
TTT Stat (sec)	199990	Serial Response Time (sec)	520.88229E-3	PA Fire Time (sec)	22.21700E+0	Det Chg Time (sec)	805.52240E-6
GG Stat (sec)	19999	TTT Stat 2 (sec)	0.00000E+0	Master Clock Frequency (Hz)	501.00000E-3	Det Fire Time (sec)	22.46600E+0
Flight Time (sec)	00000	GG Stat 2 (sec)	8.59375E-3	Status 4 (Hex)	1.02511E+0		
Current (Amp)	68.82131E-3	TB Energy (Joules)	150000	Flight Time 2 (sec)	100.00030E+0	Comments:	
28V input (Volt)	34.26833E+0	TB Chg Max (Volt)	15000	GG Energy (Joules)	1D5		
28V TP (Volt)	32.30520E+0	TB Chg Time (sec)	153.89534E-3	GG Chg Max (Volt)	01833		
5V TP (Volt)	4.93657E+0	Status 3 (Hex)	20.62500E+0	GG Chg Time (sec)	214.53519E-3		

**UUT PASSED**

Figure 7. Print Out

### 5.8 Parameters Stored in a Text File.

All the limit parameters were stored in a separate text file. This allows changes to be made to the limits without having to release a new revision of the software. The initialization parameters are also stored in a separate configuration file. These parameters include the GPIB address for each instrument and the default options for the test software (password, default print, default storage, etc.). The test software opens these files when it starts and puts all the data into global variables to be used later when testing.



## 5.10 Additions

I found that there were additions needed that weren't included in my original list of deliverables. These included a calibration routine that allows the calibration technician to verify the functionality of each piece of equipment annually. This software will communicate with each piece of electronics and simulate the same environment that the TCS fuze sees during testing. This will allow the calibration technician to verify with separate instrumentation that the tester performs to specification. See figure 9.

The screenshot shows the 'MLRS TCS CALIBRATION ROUTINE' software interface. It features a grid layout with various input fields and buttons. The interface is divided into several sections:

- Load Resistance:** Includes fields for '12 Ohm (Ohms)', '1 MOhm (MOhms)', and a 'GO' button.
- PWB Resistance:** Includes fields for '14 to 15 (Ohms)', 'A to B (Ohms)', '13 to 25 (Ohms)', 'Det (Ohms)', and 'PA (Ohms)', each with a 'GO' button.
- HOUSING Resistance:** Includes fields for '14 to 15 (Ohms)', 'A to B (Ohms)', '13 to 25 (Ohms)', 'Det (Ohms)', and 'PA (Ohms)', each with a 'GO' button.
- Voltages & Currents:** Includes fields for '24V (Volts)', '24V current (mAmps)', '36V (Volts)', '36V current (mAmps)', and '5V (Volts)', each with a 'GO' button.
- Counter & Generator:** Includes fields for 'Internal Time (Hz)', 'Frequency (Hz)', and 'Frequency Meas.', each with a 'GO' button.
- UD Resistance:** Includes fields for 'UD Closed (Ohms)', 'UD Open (MOhms)', and 'Enable Resistance', each with a 'GO' button.
- Enable Resistance:** Includes fields for 'Enable Closed (Ohms)' and 'Enable Open (MOhms)', each with a 'GO' button.
- 20 uS Pulse:** Includes fields for 'Pulse Time (uS)' and 'Amplitude (Volts)', each with a 'GO' button.
- 800 uS Pulse:** Includes fields for 'Pulse Time (uS)' and 'Amplitude (Volts)', each with a 'GO' button.
- Switches & RS232:** Includes fields for 'Safety Switch', 'Loaded Key', 'Unloaded Key', 'RS232 Vrms', and 'RS232 Bit Rate', each with a 'GO' button.
- Operator Field:** A text box at the top right for entering the operator's initials.
- Action Buttons:** 'OK' (yellow), 'PRINT' (green), and 'DONE' (red) buttons are located on the right side.
- Instructions:** A text box at the bottom right says 'Please enter in the operator's initials in the "Operator" field.'

Figure 9. Calibration Routine

## 6. Conclusion

In closing I found this project extremely challenging but rewarding also. The mechanical part of assembling all the different components was difficult but the software posed unique challenges also. The communication routine that had to be written to talk

to the fuze took a bit longer than I anticipated and getting the DMM/Switch unit to setup and measure when needed was also a bit harder than I thought it would be. One of the nicer challenges I faced was downloading a million bits from the digital scope and writing a routine to sort through that data and calculate fire pulse energy.

It helped that my company sent me to a Labview seminar for a few days and I feel that this experience will help my professional development at my job. I hope that this will give me more opportunities to use my skills that I acquired in other test setups at work.

## References

1. National Instruments Developer Zone. <http://zone.ni.com/zone/jsp/zone.jsp>. 2004.
2. Agilent. <http://we.home.agilent.com/USeng/nav/-11145.0/ia.html>. 2004.
3. SXLIST (PCL Language). <http://www.sxlist.com/techref/language/pcls.htm>. 2004.
4. Agilent. *Agilent 34907 Users Guide*. 2002.
5. Agilent. *Agilent 54645 Users Guide*. 2003.
6. Agilent. *Agilent 6612 Users Guide*. 2001.
7. Agilent. *Agilent 53131A Programmers Guide*. 2003.