



University of Tennessee, Knoxville  
**TRACE: Tennessee Research and Creative  
Exchange**

---

[Masters Theses](#)

[Graduate School](#)

---

8-2009

## **Flight Test and Evaluation of a Low-Cost, Compact, and Reconfigurable Airborne Data Acquisition System Based on Commercial Off-The-Shelf Hardware**

Christopher George Ludwig  
*University of Tennessee - Knoxville*

Follow this and additional works at: [https://trace.tennessee.edu/utk\\_gradthes](https://trace.tennessee.edu/utk_gradthes)



Part of the [Aerospace Engineering Commons](#)

---

### **Recommended Citation**

Ludwig, Christopher George, "Flight Test and Evaluation of a Low-Cost, Compact, and Reconfigurable Airborne Data Acquisition System Based on Commercial Off-The-Shelf Hardware. " Master's Thesis, University of Tennessee, 2009.  
[https://trace.tennessee.edu/utk\\_gradthes/49](https://trace.tennessee.edu/utk_gradthes/49)

This Thesis is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Masters Theses by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact [trace@utk.edu](mailto:trace@utk.edu).

To the Graduate Council:

I am submitting herewith a thesis written by Christopher George Ludwig entitled "Flight Test and Evaluation of a Low-Cost, Compact, and Reconfigurable Airborne Data Acquisition System Based on Commercial Off-The-Shelf Hardware." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Aviation Systems.

Stephen Corda, Major Professor

We have read this thesis and recommend its acceptance:

John F. Muratore, Borja Martos

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Christopher George Ludwig entitled "Flight Test and Evaluation of a Low-Cost, Compact, and Reconfigurable Airborne Data Acquisition System Based on Commercial Off-The-Shelf Hardware". I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Aviation Systems.

---

Stephen Corda, Major Professor

We have read this thesis  
and recommend its acceptance:

---

John F. Muratore

---

Borja Martos

Accepted for the Council:

---

Carolyn R. Hodges, Vice Provost and  
Dean of the Graduate School

(Original signatures are on file with official student records.)

**FLIGHT TEST AND EVALUATION OF A LOW-COST, COMPACT,  
AND RECONFIGURABLE AIRBORNE DATA ACQUISITION  
SYSTEM BASED ON COMMERCIAL OFF-THE-SHELF  
HARDWARE**

A Thesis Presented for  
the Master of Science  
Degree  
The University of Tennessee, Knoxville

Christopher George Ludwig  
August 2009

## **Acknowledgements**

I wish to thank those whom helped to make this thesis possible. I would like to thank Carl Morgan for his work creating a software application to decode data files and for assisting with our early efforts. I would also like to thank our chief mechanic, Greg Heatherly, our instrumentation technician, Steve Amyot, and our electrical engineer, Mike Leigh for their help and guidance through the last two years. In addition, I would like to thank Professor John Muratore for teaching me the tools of data analysis, Mr. Borja Martos for his support and encouragement, and Dr. Stephan Corda for his guidance as my advisor.

Finally, I would like to thank Jeff DeFouw, Carlos Fernandez, and Greg Toman at Grand Rapids technology for their technical help and support. I would also like to thank them for allowing the reproduction of their company's data file formats and interface control documents in this thesis.

## **Abstract**

Digitization of physical parameters for the display and recording by computers is the essential aspect of any airborne data acquisition system. The objective of this thesis was to develop a data acquisition system for General Aviation research and certification flight testing based on a low-cost Commercial Off-The-Shelf (COTS) hardware, in particular, a common glass cockpit system for experimental aircraft. A kneeboard computer was used to monitor data communications between the various devices of the Grand Rapids Technology (GRT) Electronic Flight Information System (EFIS). The monitored data was then displayed for use in-flight, and recorded aboard the aircraft for post-flight data reduction. The developed system and software was tested in simulation on virtual and actual hardware, on an Extra 300 in ground testing, and in flight. An in flight air-data calibration and several common stability and control certification test points were flown to evaluate and demonstrate the usefulness of the system. Special consideration was paid to work flow prior to, during, and after the flight with the overall goal of reducing the time required for data reduction. The output of this research work includes software for decoding data files logged on one common low-cost EFIS, software for monitoring, displaying, and recording EFIS data on a kneeboard computer in-flight, and tools for managing and viewing data files after the flight. From this research work, it is concluded that commercially available EFIS systems do in fact provide a core data set which is useful in flight research and flight test certification programs. The 15 Hz sampling rate of the GRT system was more than sufficient for all the test points evaluated as a part of this research. The cost of the tested hardware was less than \$10,000 at current pricing (2009). The resultant system is compact, adds little weight to a test aircraft, has few interfaces to aircraft systems, and allows for future growth and the incorporation of new sensor types and interfaces. The addition of a flight test air-data boom with angle of attack and sideslip vanes and control position and force sensors would create a very complete data acquisition package without the expense of purpose designed hardware.

## Table of Contents

<b>I.</b>	<b>INTRODUCTION .....</b>	<b>1</b>
A.	PROBLEM STATEMENT .....	2
B.	DESIGN REQUIREMENTS .....	3
1.	<i>Primary Design Goals</i> .....	3
2.	<i>Minimum Data Requirements</i> .....	3
3.	<i>User Interface</i> .....	4
<b>II.</b>	<b>LITERATURE REVIEW .....</b>	<b>5</b>
A.	PURPOSE DESIGNED AND BUILT COMMERCIALY AVAILABLE FLIGHT TEST SYSTEMS.....	5
B.	STRAP-DOWN AND CARRY-ON SYSTEMS.....	5
C.	EXPERIMENTAL – HOMEBUILT DEVELOPMENTS .....	7
<b>III.</b>	<b>FLIGHT TEST DATA ACQUISITION SYSTEM DEVELOPMENT.....</b>	<b>8</b>
A.	ELECTRONIC FLIGHT INFORMATION SYSTEM DESCRIPTION .....	8
1.	<i>Engine Information System (EIS), “Model 4000”</i> .....	8
2.	<i>“Horizon” Air Data Computer (ADC) / Attitude, Heading and Reference System (AHRS)</i> .....	8
3.	<i>Magnetometer</i> .....	9
4.	<i>Global Positioning System (GPS)</i> .....	9
5.	<i>Rear Horizon 1 Electronic Flight Information System (EFIS)</i> .....	10
6.	<i>Front Horizon 1 Electronic Flight Information System (EFIS)</i> .....	10
B.	A INTRODUCTION TO NATIONAL INSTRUMENTS LABVIEW 8.5 .....	10
C.	GRT POST PROCESSOR DEVELOPMENT .....	12
D.	DATA VIEWER DEVELOPMENT .....	13
E.	REAL TIME DATA ACQUISITION DEVELOPMENT .....	13
1.	<i>Aircraft System Interfaces and Acquisition Hardware Selection</i> .....	14
2.	<i>GRT Real-Time Software</i> .....	15
F.	SOFTWARE DEVELOPMENT TESTING IN SIMULATION .....	19
1.	<i>Early Development Testing Using Serial Port Emulators</i> .....	19
2.	<i>Virtual Hardware Test Methods</i> .....	20
3.	<i>Hardware in the Loop Test Methods</i> .....	20
<b>IV.</b>	<b>TESTING APPROACH AND METHODOLOGY .....</b>	<b>21</b>
A.	THEORETICAL MAXIMUM DATA RATE TEST .....	21
B.	GROUND TESTING .....	21
1.	<i>Electro-Magnetic Interference (EMI) Checks</i> .....	21
2.	<i>AHRS Attitude and Heading Drift Check</i> .....	22
3.	<i>Power Sweep</i> .....	22
C.	FLIGHT TESTING AND EVALUATIONS .....	22
1.	<i>Evaluation of Timestamp Accuracy</i> .....	22
2.	<i>Power Sweep</i> .....	23
3.	<i>Static Pressure Position Error Correction</i> .....	23
4.	<i>Maneuver Evaluations</i> .....	23
<b>V.</b>	<b>RESULTS AND DISCUSSION .....</b>	<b>26</b>
A.	THEORETICAL MAXIMUM DATA RATE TEST .....	26
B.	GROUND TESTING .....	26
1.	<i>Electro-Magnetic Interference (EMI) Checks</i> .....	26
2.	<i>AHRS Attitude and Heading Drift Check</i> .....	27
3.	<i>Power Sweep</i> .....	27
C.	FLIGHT TESTING AND EVALUATIONS .....	27

1.	<i>Evaluation of Timestamp Accuracy</i> .....	28
2.	<i>Power Sweep</i> .....	28
3.	<i>Static Position Error Correction</i> .....	28
4.	<i>Maneuver Evaluations</i> .....	29
D.	PERFORMANCE OF THE GRT SYSTEM .....	31
E.	EASE OF GROWTH AND EXPANSION .....	32
F.	POST PROCESSING WORK FLOW .....	33
G.	USEFULNESS FOR FLIGHT TEST CERTIFICATION PROGRAMS .....	33
H.	USABILITY OF THE USER INTERFACE IN FLIGHT .....	34
I.	DEVELOPMENT TIME AND COST .....	34
J.	UNRESOLVED ISSUES .....	36
<b>VI.</b>	<b>CONCLUSIONS</b> .....	<b>37</b>
A.	RECOMMENDATIONS FOR FUTURE RESEARCH AND DEVELOPMENT .....	37
	<b>REFERENCES</b> .....	<b>39</b>
	<b>APPENDICES</b> .....	<b>42</b>



## **List of Tables**

Table 1	AHRS Published Specification Summary .....	9
Table 2	Hardware Weight and Cost Budget .....	35
Table 3	Demo Log File Format .....	68
Table 4	Demo Log Data Sources .....	68
Table 5	Model 4000/6000 EIS Data Format.....	69

## List of Figures

Figure 1	Extra 300 .....	43
Figure 2	Rear Instrument Panel .....	43
Figure 3	Video Equipment in Use on Red Bull Aircraft .....	44
Figure 4	GRT Engine Information System.....	44
Figure 5	GRT EIS Transducers .....	45
Figure 6	GRT Attitude, Heading, and Reference System.....	45
Figure 7	GRT Magnetometer .....	46
Figure 8	GRT Electronic Flight Information System .....	46
Figure 9	Post Processor User Interface.....	47
Figure 10	Reference Node Method for Calling SubVI's .....	48
Figure 11	Data Viewer .....	49
Figure 12	Installation of RS-232 to USB Optically Isolated Hub .....	50
Figure 13	Aircraft System Interfaces.....	51
Figure 14	Program Architecture .....	52
Figure 15	Invoke node Method for Calling SubVI's .....	53
Figure 16	Graphic User Interface Setup Tab .....	54
Figure 17	Graphic User Interface Data Tab .....	55
Figure 18	Graphic User Interface Events Tab .....	56
Figure 19	Events File Sample.....	57
Figure 20	Extra System Simulator (GRT Device Simulator) .....	58
Figure 21	Hardware in the Loop Software Testing .....	59
Figure 22	Histogram Comparison of Elapsed Times.....	60
Figure 23	Static Position Error Correction, Airspeed .....	61
Figure 24	Static Position Error Correction, Altitude .....	62
Figure 25	Dutch Roll Data .....	63
Figure 26	Spiral Data.....	64
Figure 27	Stall Data.....	65
Figure 28	Spin Data.....	66
Figure 29	Maximum Rate Aileron Roll Data .....	67

## Nomenclature

$HDG$	=	Heading, deg
$h_i$	=	Indicated altitude, ft
$p$	=	Roll rate, deg/s (positive right wing down)
$q$	=	Pitch rate, deg/s (positive nose up)
$r$	=	Yaw rate, deg/s (positive right turn)
$V_i$	=	Indicated Airspeed, kts
$V_{iw}$	=	Weight Corrected Airspeed, kts
$VSI$	=	Vertical speed, fpm
$X-Accel$	=	Acceleration, G (positive forward)
$Y-Accel$	=	Acceleration, G (positive right)
$Z-Accel$	=	Acceleration, G (positive up)
$\beta$	=	Sideslip angle, deg
$\Delta h_{pc}$	=	Altitude Position Error Correction, ft
$\Delta V_{pc}$	=	Airspeed Position Error Correction, kts
$\theta$	=	Pitch angle, deg (positive nose up)
$\phi$	=	Roll angle, deg (right wing down)
$\psi$	=	Heading, deg

## Abbreviations

$A/C$	=	Aircraft
$ADC$	=	Air Data Computer
$AHRS$	=	Attitude Heading and Reference System
$ASCII$	=	American Standard Code for Information Interchange
$CG$	=	Center of Gravity, aft of datum (wing leading edge)
$COTS$	=	Commercial Off The Shelf

<i>CPU</i>	=	Central Processing Unit
<i>EFIS</i>	=	Electronic Flight Information System
<i>EIS</i>	=	Engine Information System
<i>EMI</i>	=	Electro-Magnetic Interference
<i>FTE</i>	=	Flight Test Engineer
<i>GA</i>	=	General Aviation
<i>GAU</i>	=	Geospatial Awareness Unit
<i>GPS</i>	=	Global Position System
<i>GRT</i>	=	Grand Rapids Technology
<i>GRT RT</i>	=	Grand Rapids Technology Real-time (a UTSI developed software)
<i>GRT PP</i>	=	Grand Rapids Technology Post Processor (a UTSI developed software)
<i>GUI</i>	=	Graphic User Interface
<i>IP</i>	=	Internet Protocol
<i>MB</i>	=	Megabyte
<i>MHz</i>	=	Megahertz
<i>NACA</i>	=	National Advisory Committee for Aeronautics
<i>NMEA</i>	=	National Maritime Electronics Association
<i>NTPS</i>	=	National Test Pilot School
<i>OEM</i>	=	Original Equipment Manufacturer
<i>PDA</i>	=	Personal Digital Assistant
<i>PIC</i>	=	Programmable Intelligent Computer
<i>RS-232</i>	=	Recommended Standard 232, Electronic Industries Association, 1969
<i>RVSM</i>	=	Reduced Vertical Separation Minimum
<i>SBIR</i>	=	Small Business Innovation Research
<i>SubVI</i>	=	Sub-Virtual Instrument
<i>TCP</i>	=	Transmission Control Protocol
<i>TP</i>	=	Test Pilot
<i>UMPC</i>	=	Ultra Mobile Personal Computer

*USB* = Universal Serial Bus  
*UTSI* = University of Tennessee Space Institute  
*VHF* = Very High Frequency  
*VI* = Virtual Instrument  
*W&B* = Weight and Balance

# **I. Introduction**

The essential aspect of flight test instrumentation is the conversion of physical measurements into discrete numerical values for post flight analysis. Early flight test data collection consisted of an observer or teams of observers manually recording values read from mechanical instruments. The observer provided conversion of physical measures to numerical values in flight. This data collection technique is limited by the number of readings that one observer can record in a given length of time. As test programs grew in complication, the practice of photographing or filming the mechanical gauges came into practice [1]. While it was now possible to collect and record more physical measurements, each measurement still had to be manually read from the developed photos and reported as a numerical value. The time intensive manual work which was previously completed on the aircraft was moved to the ground and completed after the test.

As electronics grew smaller and lighter, they were incorporated into flight test instrumentation systems. Early electrical systems were completely analog and involved the creation of strip charts using an electrically driven pen on a reel of paper chart. Again, the data was converted to numerical values by hand after the test. As technology progressed, the data transmission and storage was taken over by analog electronic devices such as telemetry radios and tape recorders. The bulkier equipment required to digitize the analog data was located on the ground. The next revolution in flight test instrumentation came with the ability to efficiently digitize measurement values and transmit and store large volumes of information in a digital format. Digital data acquisition systems allowed for onboard data analysis and the creation of test specific data displays.

As the state of the art progressed, each innovation was incorporated into the larger and better funded flight test programs. Commercial Off-The-Shelf (COTS) hardware and software packages were developed which utilized standard formats for data storage, telemetry, and data bus monitors. These systems remain expensive and are primarily used by military projects and larger commercial aircraft. Such packages still remain out of reach of smaller general aviation programs. While some General Aviation (GA) certification programs make use of airborne instrumentation, many are still being conducted using hand recorded parameters [2].

Development of electronics for commercial and military aircraft has progressed steadily over the years while the development of electronics for GA fell behind. However, after years of stagnation and very slow development, there has been a recent explosion in the development of digital devices for the GA cockpit. Still, the cost and complexity of purpose designed and built digital airborne data acquisition has led the smaller test teams to continue to routinely use hand recorded data for certification programs.

Recent advances in computer technology have made glass cockpit systems accessible for most GA owners. These systems range in complexity and cost, especially between the systems designed for certified aircraft and the systems designed for experimental aircraft. While the certified systems are more expensive with built-in redundancy and a much improved software interface, the data available to the system is largely the same as that available to the less expensive experimental systems. Most systems offer airspeed and altitude data, attitude and heading data, and position data, usually based on the Global Positioning System (GPS). Most systems also offer engine monitoring capability and also allow for the addition of additional sensors for special applications. These complete systems range in prices, but uncertified systems can be purchased for as little as \$10,000. These systems can be used to perform much of the data acquisition needs of small aircraft test programs. Most new production GA aircraft made by the major manufacturers come standard with certified glass cockpit systems.

These glass cockpit systems already present many of the basic parameters needed for performance and stability and control flight testing in a digital format. A flight test team can extract the digital data from these systems and make use of the system as a data source thus greatly reducing the cost of a complete data acquisition system. In many cases, the required hardware to collect and digitize the bulk of the flight parameters is already installed in the aircraft. In other cases, the test team could choose to purchase a \$10,000 system and install it either temporarily, or permanently, as the heart of the data acquisition system.

## **A. Problem statement**

The University of Tennessee Space Institute (UTSI) department of Aviation Systems operates an Extra 300 for the purpose of introducing students to aerobatic flight and flight test methods for stalls and spins. Figure 1 is an image of the UTSI Extra 300. All figures are located in the Appendix. In addition to the teaching role, it was desired to begin using the aircraft for research. The aircraft was fitted with a typical complement of mechanical flight instruments for an aerobatic aircraft. A data acquisition system was

desired that would allow for digitally recording of flight data and also conform to the weight and space limitations of the aircraft and the budget limitations of the program. The availability of commercial glass cockpit systems has presented the flight test community with a selection of low-cost, commercial off-the-shelf sensor packages. The decision was made to install one such system in the Extra 300 in an effort to procure an acquisition system which would conform to the cost, size, and weight constraints. A low-cost Electronic Flight Information System (EFIS) built by Grand Rapids Technology (GRT) was purchased and installed as a means to collect flight test and research data. The upgraded rear cockpit is pictured in Figure 2. A second EFIS was fitted to the front cockpit for the student or flight test engineer to use.

The objective of this thesis was to develop a data acquisition system based on a low-cost Commercial Off-The-Shelf (COTS) hardware including a glass cockpit system, in particular, the system manufactured by Grand Rapids Technology. The final result of the work will take into consideration the post flight work flow including data processing, in flight access to test parameters, ease of growth and expansion of the system, data quality and usefulness for testing applications, and any other considerations to ensure successful use of the system for research and flight testing purposes. The development of a data acquisition system based on the GRT EFIS system would include the installation, ground testing, and flight testing to demonstrate the data quality and prove the usefulness of the system.

## **B. Design Requirements**

The basic design requirements were basically maintained throughout the project. However, the originally conceived method for achieving the requirements changed significantly as the work progressed due to the learning gained throughout the project. The following is a list of design goals and requirements:

### ***1. Primary Design Goals***

- 1) System Weight must be kept to the minimum.
- 2) Minimal Power Consumption
- 3) Scalable hardware and software capable of supporting future research needs.

### ***2. Minimum Data Requirements***

- 1) Roll, Pitch, Yaw, rotational rates about the X, Y, and Z axes, and linear accelerations along the X, Y, and Z axes.
- 2) Airspeed and altitude measurements derived from pitot-static measurements.



- 3) Global Positioning System (GPS) supplied Latitude, Longitude, Groundspeed, and Ground Track.
- 4) Engine parameters including engine RPM, Manifold Pressure, Fuel Flow Rate, Outside Air Temperature, and Fuel Quantity.

### **3. *User Interface***

- 1) Ability to view time history of test parameters.
- 2) Ability to view a measure of the variation of values in the time histories.
- 3) Ability to select and display primary test values prominently on screen in an easy to read format.
- 4) Incorporate a simple and easy method to interface to other programs to share data.

## **II. Literature Review**

While there are several publications that detail flight test instrumentation systems for military programs [1] and [3], there is very little published data covering flight test instrumentation for light aircraft. A literature search was conducted to find any related works on the subject. The search focused on publications by the Society of Flight Test Engineers, the American Institute of Aeronautics and Astronautics, and the Advisory Group for Aerospace Research and Development, but also included internet and library searches which included Theses and Dissertations. The results of the search can be categorized as follows:

### **A. Purpose Designed and Built Commercially Available Flight Test Systems**

Commercially available systems include basic avionics bus monitors, complete flight data recorders, and pre-packaged flight test instrumentation systems. These commercial systems offer a complete capability of recording digital data sources as well as analog sources.

Simtec Buergel AG of Switzerland offers a system which includes a multi-port flush air data boom and supports acquisition of GPS and a gyro package [4]. Reported pricing is \$38,000 [5]. Flush air-data probes offer accurate readings over only a limited range of angle of attack and angle of sideslip. Since future research of the Extra would likely include high angle of attack and departed flight research, a flush air data system was not preferred. Also, the Icasim system cannot acquire additional sensors, and would therefore need to be augmented with an additional acquisition system.

### **B. Strap-Down and Carry-On Systems**

Several flight test organizations have pursued the development of a strap-down or carry-on sensor package which includes a user interface to allow testing of multiple aircraft without installed data acquisition and instrumentation hardware. A 2007 paper by Western Michigan University in Kalamazoo detailed two very different approaches to collecting data from a GA aircraft for flight test education. The first system was a heavily modified Cessna 182 with a large instrumentation system that was completely independent of the aircraft systems [6]. The design included an AC generator powered by an internal combustion engine sealed in a metal box. While achieving the goals of the project, the design of the system

was deemed unsuitable for the Extra due to the excessive weight and space required for the package. The second system detailed in the same paper consisted of an unmodified Cirrus SR20 and an Appareo Geospatial Awareness Unit (GAU) 1000. The GAU 1000 is a small unit which contains a MEMS gyro package, GPS receiver, memory card, and a battery. Limited engine data is downloaded from the Cirrus aircraft itself. The combination of the Appareo system and Cirrus engine data does offer some important performance data; however, it does not offer airspeed data, does not provide an interface with the data in flight, and does not have the ability to be scaled or expanded to record additional parameters. For these reasons, this option was not selected for the Extra 300 data acquisition system.

A 2003 project by Purdue University developed a strap down sensor pallet which could be installed in the baggage space of a light aircraft [7]. The system was developed to allow flight test engineering education using Part 23 certified aircraft. The portable data acquisition system included a MEMS gyro package, GPS receiver, and static and dynamic pressure sensors. A laptop acquired the data and acted as a data server sending data to a remote ultra mobile sized laptop running a user interface for the test engineer. The entire system was powered by a battery pack and utilized LabVIEW [8] software for data acquisition and the user interface.

The Purdue system lacked engine instrumentation or analog data inputs, but the basic architecture of Laptop running LabVIEW allows for significant expansion and growth. A similar strap down system could be made to function in the Extra if desired. Aircraft flying in the Red Bull Air Races routinely fly with video transmission boxes on their baggage shelf at load factors well exceeding 10G, purportedly reaching +14G in races for brief instances. One such system installation is shown in Figure 3. However, mounting hardware in the baggage area would cause additional weight and Center of Gravity (CG) problems. It was desired to find another solution.

The Center of Aeronautical Studies (CEA) of the Federal University of Minas Gerais conducted a project to develop a light aircraft flight test equipment package consisting of analog to digital converters, with support for a GPS and solid state gyro package [5]. The complete system supported the acquisition of airspeed, altitude, angle of attack, and angle of sideslip from an airdata boom. No details were given on the final cost or weight of the system. Data was packaged using Programmable Intelligent Computer (PIC) Microcontrollers and sent to a Personal Digital Assistant (PDA) for data logging purposes. This particular

system was designed well and serves the intended purpose well, but it lacks the ability to easily grow to accommodate new instruments as research requirements grow. Also, there are no details regarding if it is possible to interact with acquired data in flight through a graphical user interface on the PDA. However, a user interface could certainly be added to the system with very little effort.

A 2006 report titled “Carry-on and Quick-Install Instrumentation for Flight Test” detailed the design of a portable carry-on system in response to an Air Force Small Business Innovative Research (SBIR) program solicitation [9]. The final system consisted of a “map case” containing solid state gyros, accelerometers, 3-axis magnetometer, and a GPS receiver. Data from the map case was sent to a PDA device for use as a data logger and user interface. The greatest innovation of this system was the level of programming in the PDA. The user interface accepted both manual and voice commands, allowed for voice recording, and provided data reduction of test points in flight. The limitations of the system were the lack of avionics bus data and the lack of access to pitot-static data.

The primary goal of the Strap-Down and Carry-On systems was to facilitate flight test data acquisition without affecting the aircraft certification. The Extra 300 was designed as an Experimental aircraft and is currently operated as a Public aircraft, thus eliminating this driving requirement. Instead, a more integrated approach was desired.

### **C. Experimental – Homebuilt developments**

Experimental-Homebuilt aircraft include high-speed, long-range cross-country aircraft equipped with full glass cockpits. These aircraft include the latest in technology including engine monitors, GPS, AHRS, in-flight weather, and synthetic vision displays of terrain. Most of these systems offer the capability to record a log file of the flight data. This log file is useful in the required initial flight testing of all homebuilt aircraft. Many homebuilders have created software to decode or display the logged data and have made these programs freely available on internet through builder’s forums, although most have never been officially published.

### **III. Flight Test Data Acquisition System Development**

#### **A. Electronic Flight Information System Description**

The Grand Rapids Technology (GRT) produces glass cockpit systems for the experimental aircraft market. The GRT Dual EFIS system was installed in the Extra to provide convenient access to commonly needed flight parameters. The GRT system consists of several devices linked together by RS-232 communications links [10]. Details of each device are as follows:

##### **1. *Engine Information System (EIS), “Model 4000”***

The EIS is located in the rear instrument panel and is powered by the Master bus. Though it is installed on an independent circuit breaker, there is no separate switch for the EIS. The EIS performs the duty of analog to digital conversion of all engine parameters and offers the pilot a simple numerical display of engine parameters. The EIS is pictured in Figure 4, and it is also visible in Figure 2 in the lower left portion of the instrument panel. The EIS acquires cylinder head temperature and exhaust gas temperature for every cylinder and also oil temperature, outside air temperature, oil pressure, manifold pressure, and fuel pressure and flow. Many of these sensors are visible in Figure 5. In addition to supporting various engine sensors and providing a pilot display, the EIS is programmed with critical engine alarms and a fuel totalizer. The EIS transmits data in a single data sentence to the EFIS at a rate of 10 Hz using RS-232 protocol.

In addition to the current sensors on the aircraft, there are four unused auxiliary analog inputs which could be used to acquire additional data. The EIS also provides a filtered 5V excitation power supply for sensors. Unfortunately, the quantity and sample rate of the acquired analog channels is rather low by data acquisition standards due to the intended use of the system. Even so, the GRT system offer convenience over hand recorded data.

##### **2. *“Horizon” Air Data Computer (ADC) / Attitude, Heading and Reference System (AHRS)***

The AHRS is located on top of the top longerons just in front of the spar on the aircraft centerline as shown in Figure 6. The AHRS is powered via the avionics bus through the avionics master switch and is circuit breaker protected. The AHRS contains accelerometers, MEMS gyros, and total and differential air pressure sensors.

The AHRS combines the magnetometer data with the sensors on the AHRS to compute a filtered attitude and heading solution. The published specifications [11] of the AHRS are listed in Table 1. Data from the AHRS is sent in one of 4 data sentences to the rear EFIS. The primary data sentence is transmitted at 15 Hz and contains airspeed, altitude, attitude, heading, normal acceleration, and slip/skid. When enabled, the maintenance message is transmitted after the primary message and contains raw accelerations and rotational rates. There is also a low rate status message and a calibration message, but these do not contain data of interest. The AHRS has two serial out connections, but only one is in use as installed. This is connected to the rear EFIS.

### 3. *Magnetometer*

The magnetometer is located on top of the top longerons on the aircraft centerline in the tail section of the aircraft as shown in Figure 7. The Magnetometer is powered by the AHRS and communicates only with the AHRS. The calibration of the Magnetometer is performed from within the maintenance menus on the rear EFIS.

### 4. *Global Positioning System (GPS)*

There is an Original Equipment Manufacturer (OEM) GPS attached to case on the rear EFIS. This GPS sends data directly into the EFIS via the EFIS serial input 1 and also receives power directly from the EFIS. The OEM GPS communicates using National Maritime Electronics Association, NMEA [12] formatted

**Table 1 AHRS Published Specification Summary**

Parameter	Specification
Maximum Angular Rate <sup>1</sup>	200 deg/s
Internal Update Rate <sup>1</sup>	200 Hz
Data Transmission Rate	15 Hz
Heading Accuracy <sup>1</sup>	+/- 2 deg
Static Attitude Accuracy <sup>1</sup>	+/- 0.5 deg
Dynamic Attitude Accuracy <sup>1</sup>	+/- 2.0 deg
Altitude Range <sup>1</sup>	-1000 to +32,000 ft
Airspeed Range <sup>1</sup>	35 to 285 kts
Operating Temperature Range <sup>1</sup>	-25 to +70 degC
Normal Acceleration Range <sup>1</sup>	+/- 10 g
Input Voltage <sup>1</sup>	9 to 32 Vdc
Power Required <sup>1</sup>	250 mA
Interface <sup>1</sup>	2 RS-232 outputs per AHRS

<sup>1</sup> Data collected from GRT Product Specification Sheet [11].

messages. The EFIS mirrors the GPS data from serial input 1 onto serial output 1. The NMEA messages transmitted by the GPS include GPRMC, GPGGA, GPGSA, GPGSV, PGRME, PGRMB, and PGRMM. Only GPRMC and GPGGA are of interest for flight testing purposes.

#### **5. *Rear Horizon 1 Electronic Flight Information System (EFIS)***

The rear EFIS, pictured uninstalled in Figure 8 and installed in the instrument panel in Figure 2, receives all the data on the aircraft. Additionally, there is the option of acquiring a limited number of analog channels on the EFIS itself. The analog parameters would be logged to the demo log file, but would not be available outside the EFIS in flight. The rear EFIS is used to update AHRS firmware and to calibrate the airspeed and altitude sensors on the AHRS when required. This is performed using a simple onscreen user interface. The rear EFIS combines all the data arriving from various sources and relays flight pertinent data to the forward EFIS. It does not, however, relay unnecessary maintenance information such as AHRS maintenance messages.

#### **6. *Front Horizon 1 Electronic Flight Information System (EFIS)***

The front EFIS receives all data from the rear EFIS. The inter-display link does not necessarily carry the full data rate and will automatically reduce the data rate depending on the number of parameters being sent. Aside from the lack of access to full maintenance messages in the menus, the front EFIS is identical to the rear EFIS.

### **B. A Introduction to National Instruments LabVIEW 8.5**

LabVIEW 8.5 [8] was used for most of the programming throughout this thesis. As it is not a universally recognized program, a brief introduction is presented here to familiarize the reader with the terminology and structure of the unique software tool. LabVIEW combines a proprietary graphical programming language with powerful built in support for common communications protocols, instrumentation hardware, and common data processing tasks. Unlike other programming languages which require the user to know correct syntax and textual commands, LabVIEW provides the user with a drag and drop method of building a program and supplies the user with the correct syntax and examples of the use of most functions. In addition to the installed help resources, there is a strong online support community sharing example programs and collaborating on user forums to find solutions to user's questions.

Programs in LabVIEW are called Virtual Instruments (VI's), and are universally referred to as VI's. A program is identifiable on a computer by the unique .vi extension. Every VI consists of two screens on the computer, a front panel, and a block diagram. The front panel contains the user interface, while the block diagram contains the code of the VI and is not normally seen except when editing a VI. When an object, be it a control or an indicator, is placed on the front panel, a terminal for that object is placed on the block diagram. The user then wires the data into or out of the terminals for the object on the block diagram. Therefore, the block diagram contains a collection of blocks representing programming functions or packaged tasks, and wires connecting them together to indicate the flow of data. Additional structures perform tasks similar to traditional if-then-else statements and programming loops.

The ease of use comes from the fact that all available functions and data indicators and controls are available in palettes and can be dragged onto the screen and dropped into place. A user does not need to know how to program a particular function in LabVIEW, but merely that they want a particular function. When the function is found in the palette and placed on the block diagram, a context help menu informs the user what is required to make the function work and offers a help file explanation, example VI's using the function, and any additional resources available.

Subroutines in LabVIEW are referred to as subVI's. When opened, they look identical to any other VI and have a front panel and a block diagram. The user can specify whether to show the front panel or whether to hide it depending on whether the user needs to interact with the user interface of that subVI.

Any LabVIEW VI can be packaged into an executable or a stand alone installation file for use on other computers which do not have a full LabVIEW installation. The one caveat is that executable files require a LabVIEW runtime engine to work, but that is available from National Instruments as a free download. LabVIEW VI's are small, with all but the largest VI's well smaller than 1 MB each, however, the runtime engine is about 60 MB in size.

Perhaps the greatest strength of LabVIEW is the built in support for acquisition tasks. Not only does LabVIEW support National Instruments and other standard industry acquisition hardware, it also offers easy to use functions for common data communications protocols, file I/O, and running formulas and scripts in nodes, including running MATLAB [13] scripts. LabVIEW also offers easy access to advanced



programming functions such as data queues, timing and synchronization tools, and conversion tools for changing data from one format to other formats.

### **C. GRT Post Processor Development**

The GRT EFIS has the capability to log data files referred to as “Demo Log” files. These files were originally intended to be used to replay flight data on the ground for demonstration purpose, hence the name. A Demo file contains a list of every data sentence received by the EFIS in addition to user keystroke data. Therefore, all data acquired by the system can be accessed and used by the flight test team after the flight, with some limitations. The onboard EFIS memory will only hold about 9 MB of data. Also, the memory of the EFIS is wiped at start up. Any data not downloaded before shutting down is lost. Data is also lost in the case of an inadvertent power reset in flight. The 9 MB of space was found to be sufficient for approximately 50 minutes of flight. For longer flights, the recording could be stopped and restarted in flight, thus dumping the recorded data to a memory stick to allow recording flight data for longer flights.

The format of the Demo Log file is quite simple and easy to decode. The file has no header and is a byte by byte log of every data packet received by the EFIS along with the source of the data packet, the time it was received, and the length of the following data, in bytes. Grand Rapids Technology has granted permission to reproduce the EFIS file format description and AHRS Interface Control Document in this thesis. This information can be found in Appendix 2 and Appendix 3.

The first software developed for the EFIS was a MATLAB script which decoded EFIS Demo Log files and provided a limited user interface for the purpose of selecting data to be saved in ASCII file formats. The development of this program was a collaboration within the Aviation Systems department. The MATLAB script required as much as 18 minutes to process a single Demo file. A second attempt was made to create a program to decode Demo files using LabVIEW 8.5. The resulting program was able to decode Demo files in as little as one minute or less. The LabVIEW program was then improved with a user interface and eventually evolved into the program suite referred to as the “GRT Post Processor”. The User Interface of the Post Processor program is presented in Figure 9. The top level VI of the Post Processor is a launcher for various subprograms which perform specific functions on the data. Each subprogram runs and performs its function, then closes and releases its memory allocation after returning the output or modified

job data array. Continued use of the program does not erode system memory thus resulting in a more stable program. Figure 10 details the Reference Node method for calling subVI's.

#### **D. Data Viewer Development**

The process of turning an EFIS Demo log file into an ASCII tab delimited data file created text files of up to 200,000 lines of data and took up as much as 80 MB when saved. Working with such files became a real challenge. These files could not be imported into Microsoft Excel 2003 whole. When imported to Microsoft Excel 2007, the file size made locating the data of interest a difficult challenge. From this problem came the LabVIEW based program referred to as the "Data Viewer". The Data Viewer application is used in both the GRT Post Processor Suite and was also developed as a standalone application for viewing ASCII data files from virtually any source.

Whether embedded in the Post Processor Suite or used as a standalone application, the Data Viewer offers the user the ability to scroll through data files and visualize data in a strip chart format. The program has simple controls for zooming in on data and scrolling through time. Once data of interest has been found, there are several options available to the user to save the desired data for future processing. The user has the option to output all parameters from an instance in time, to output the average values along with the minimum, maximum, and standard deviation of each parameter for a selected time interval, and to output the original data from a given time interval. This output data is then used for reporting purposes and further analysis. To aid the user in locating desired data, the application allows the user to jump to particular time intervals or events in the data. To date, the application has been tested on file sizes up to 250 MB without any difficulty or delayed response to user actions. The user interface of the Data Viewer is presented in Figure 11 of the Appendix.

#### **E. Real Time Data Acquisition Development**

While the Demo log file allowed post-flight access to data, there were several limitations. The GRT EFIS displays flight data in formats pertinent to a pilot. While a moving map and an artificial horizon provides good situational awareness, it does not provide data in formats preferred by flight test engineers. Optimal displays for flight test engineers would include a time history of data and visual representations such as strip charts and cross plots. Additionally, it was originally intended to install additional sensors in

the aircraft and use the EFIS to acquire the data from those sensors. The quantity of additional sensors that could be acquired by the EFIS was limited, and the quality of the resulting data was not well quantified.

This led to the decision to build upon the current capability of the EFIS by adding a small computer running LabVIEW to provide continuous acquisition and logging in the aircraft. A small kneeboard computer, referred to in this paper as an Ultra Mobile Personal Computer (UMPC), was chosen for use in flight. The UMPC monitored communications between the various GRT devices, decoded the raw data, logged the data to disk, and displayed the data for the flight test engineer in flight. All of this was performed with minimal modification to the aircraft. The same algorithm used to decode Demo Log files was used in flight to decode the GRT data.

### ***1. Aircraft System Interfaces and Acquisition Hardware Selection***

While there are many options for hardware for the in-flight data acquisition including embedded computers, the 600 MHz Samsung Q1 UMPC proved appropriate for the purpose. The available payload on the Extra 300 is extremely limited. In addition, the power available for onboard electronics is limited. Given these considerations, the use of a self-powered computer which was not permanently mounted to the airframe was preferred. While lower powered computers were available such as PDA's and similar devices, the use of Windows XP as the operating system on the Samsung Q1 UMPC was a significant advantage. Since the Samsung UMPC's used the same operating system as the department's lab computers, it was easy to compile LabVIEW programs on lab computers and deploy them to the UMPC's for use in flight. This reduced the system resource requirements to run the VI's. In addition, Samsung Q1 UMPC's were available for use as they were also in use aboard other department aircraft as personal data displays for students.

The communications protocol between the various devices on the aircraft utilizes RS-232 protocol with a simple to decode data format. This led the decision to interface to the aircraft systems using an RS-232 to USB hub. The acquisition system ties into the original onboard EFIS system in only one location. A short "T" harness has been added between the main connector of the rear EFIS and the aircraft wiring harness. This "T" harness allows the EFIS to share data with an RS-232 hub. The RS-232 hub, "T" harness, mounting hardware and final installation is visible in Figure 12. The RS-232 hub is connected to the transmit and ground wires coming from the AHRS and EIS. It is also connected to the serial 1 output on

the EFIS to pick up GPS data. The “T” harness was constructed using the standard orange color coding to identify it as non-permanent wiring on the aircraft. In order to minimize the risk of interfering with the communications within the GRT system, only the receive pins on the RS-232 are connected. The transmit pins have been left unattached to minimize the risk of blocking communications between the GRT devices. Since the aircraft is operated as a Public aircraft, certification requirements were greatly simplified compared to Part 23 certified aircraft. The modification and approval process was carried out following established departmental policies.

The RS-232 hub has optically isolated inputs and is powered via the USB connection with the Samsung UMPC. When the USB cable is disconnected, the RS-232 hub is effectively an inert device. It’s presence in the system does slightly reduce the power of the signal received by the EFIS. However, due to the relatively short transmission distances, this is not an issue. As such, the hub is mounted in the aircraft at all times. The Samsung UMPC may be optionally used if desired on a test by test basis. A detail of the interface to aircraft systems is available in Figure 13.

## ***2. GRT Real-Time Software***

The software which was developed to read data from the devices of the GRT system is known as GRT Real-time or GRT RT within UTSL. This name is something of a misnomer, as neither the software, nor the operating system, are deterministic real time programs. They are, at best, near real time. The name was chosen as a means of differentiating the in-flight software from the post-flight Demo Log decode software. The capability exists within LabVIEW to deploy applications to real-time embedded systems; however, it was decided to use a more readily available Microsoft Windows XP operating system instead.

The GRT RT software used with the Grand Rapids Tech EFIS system was designed to achieve three basic goals. The first goal is to allow the user to view data in flight in a format that is useful to an engineer. The basis for this goal is that the EFIS already displays flight data in a display that is useful to a pilot. To satisfy this goal, the software has been designed with simple to use drop down menus to select up to 12 parameters for display. Also, up to 8 parameters can be kept on running strip chart styled graphs which provide time histories. The second goal was to make the software and system robust and easy to use and troubleshoot. The program has been designed in a modular fashion where the user interface provides supervisory control over the acquisition and logging programs. As such, the user can kill the acquisition

and logging programs and resurrect them without the need to restart the entire program, thus preserving the job settings. In addition, the program has simple troubleshooting functions such as alarms to identify program faults and to inform the user of inoperative acquisition devices. The third and final goal of the program was to create a process which would minimize the time spent identifying the data needed for input into other analysis programs after the flight. To achieve this goal, an events system has been set up which allows the user to easily record a human readable events file in flight and then import the events files into the Data Viewer program after the flight.

### **Program Architecture**

The GRT Real-time program has been designed to be both user friendly and error tolerant. It will recover from most faults. In order to achieve this, the program has been designed in three separate parts. The Graphical User Interface (GUI) is launched first and collects all the information from the user necessary to run the job. To simplify the in flight tasks, the GUI is the only program that the user interacts with. A diagram of the program architecture is presented in Figure 14.

The user must click a button to start acquisition and also to start logging. Since these functions do not start automatically, the user can configure the GUI and display on the ground before going to the aircraft. Once plugged in to the aircraft with the systems online, the user then starts the acquisition process. Should an error develop, the user can stop the acquisition process. This aborts the acquisition program, even if the program is not responding.

The logging subprogram was set up in a similar fashion. This allows these programs to error out and be resurrected quickly and easily in flight without loss of time. This is the rationale behind the modular software design. Figure 15 shows the method of calling the acquisition and logging programs while retaining the ability to abort and restart the programs.

### **Execution Systems, Priorities, and Ensuring Accurate Timestamps**

The choice of a compiled LabVIEW program running on Windows XP on a consumer grade device was driven by the desire to keep the learning curve associated with the software development and hardware costs as low as possible. However, Windows XP is not a real time operating system. There is no way to guarantee that the processor will be ready and waiting to perform the acquisition task when data is sent by a

device. However, the use of subVI prioritization and careful selection of execution systems can ensure that the wait time before a task is carried out is as short as possible.

LabVIEW is designed to take advantage of multithreaded execution systems. The user may select which subVI's to group together on a particular thread and which to keep on separate threads. The user may also specify the execution priority of each subVI. SubVI's with higher execution priorities that do not share an execution system have the highest priority for system resources.

To ensure accurate timestamps of incoming data, it is necessary to ensure the subVI's which receive data from the devices are ready and waiting for the data. Also, it is necessary to apply timestamps as soon as the data is pulled from the RS-232 buffer. The AHRS transmits in excess of 30 data sentences every second, 15 primary messages, 15 maintenance messages, and also at least 1 low rate message. This is the fastest expected data rate. The AHRS therefore has the highest specified execution priority and is executed in a designated execution system. The EIS transmits 10 data sentences every second and the GPS only updates once a second, but transmits that data in seven different data sentences, two of which are decoded by the program. The GPS and EIS subVI's share an execution system and have a lower priority than the AHRS. The use of careful prioritization and ample CPU speed ensures that the software is always waiting for new data from the hardware.

Each of the acquisition SubVI's remove bytes from the RS-232 buffer and apply a timestamp as soon as a sync word is identified. In the manner, the most accurate timestamps possible are achieved. Also, this ensures the shortest possible delay between the time the data is transmitted on the RS-232 line and the time a timestamp is applied.

Since the decoding of the data sentences can take varying lengths of time to accomplish, it is possible that the timestamps would be logged out of order. To prevent this, a buffer of recent data is maintained prior to logging. The oldest timestamps in the buffer are identified and sent to the logger first. About 1 to 2 seconds of data is held in the buffer. This does not affect the data display, however. Data for the display bypasses the buffer to ensure the user has up-to-date data in flight.

## **Graphic User Interface**

The Graphic User Interface (GUI) is the visible portion of the program. Although the software is really a combination of three independent programs, the user only interacts with one, the GUI. This makes it

easier for the user to navigate the application in flight. Once the program is running, there is no need to change focus to another window in flight. The key features of the GUI are a set of tabbed displays, a Master Alarm, a clock, event buttons, and an event counter. Additionally, there are three tabs which contain various controls as shown in. Descriptions of each of the three tabs are as follows:

#### *The Setup Tab*

The setup tab, presented in Figure 16, contains the controls to start and stop the acquisition and logging programs, an Alarm display, a complete parameter list, and the stop button to end the program. The parameter list shows every available acquired measure and is useful for quickly checking for missing or bad data.

Every effort has been made to provide the user with useful information in the case of a problem with the acquisition system. The “Master Alarm” indicator on the top of the screen alerts the user to a problem with the acquisition hardware or software. The alarms display on the “Setup” tab is then used to direct the user to the cause of the problem and to provide a clue as to the troubleshooting steps. Individual alarms identify a loss of data from a particular device, loss of GPS signal lock, rejected checksums, failure to log data, and also monitors various buffers and queues to make sure the software is keeping pace with the acquisition devices. In the case of rejected checksums, data packet with the rejected checksum is logged to a diagnostic file. Should there be an excess in rejected checksums, this file could be used to partially reconstruct the lost data. Also, it may be useful in determining why the checksums were rejected.

#### *The Display Tab*

The display tab, presented in Figure 17, contains a data block of 12 displayed parameters, two tabs of four strip charts each, and controls for the display and charts. Parameters in the display can be quickly selected using the drop down menu above each data box. The data in the four charts are selected using dropdown menus at the top of the group of four charts. To the left of each chart is a group of three data fields. The top field shows the current value, the middle tab shows the average of the time period visible on the graph, and the bottom field shows a measure of the variation of the data. The user selects which measure of variation they want on the setup tab. The options include Standard Deviation, Range, Mean Absolute Deviation, and Inter-quartile Range. The graph controls in the middle of the tab allow the user some additional options. The user may select the duration of time that data is held in the buffer in seconds.

The user may also clear all data on the screen and may select to stream data or to accumulate data, effectively setting the buffer size to infinity. There is also a button which allows the user to freeze the displays on the screen to write down a snapshot of values or show the charts to another person.

#### *The Events Tab*

The Events tab, presented in Figure 18, affords the user a few advanced program features. The use of the Events tab is entirely optional. The top portion of the tab offers a checklist which may be used to prompt the user to the next test point. In addition to prompting the user to the next test step, the current checklist item and the time is recorded in an events file when user completes a checklist item or moves on to another checklist item. The middle portion of the tab allows the user to select an event name. When the user selects the “Begin Event” or “End Event” buttons at the top of the screen, the software logs the elapsed time, date and local time, event number and the event name that the user selected on the Events tab. The checklist and list of available event names are defined prior to the flight in a text file. Should the user want an event name that was not prepared in advance, the user may enter additional names under “User Defined Events”.

By using the checklist, event names, and the Begin Event and End Event buttons in flight, the software constructs a human-readable text file recording the steps of the flight test. This events file is useful for annotating data and for archiving with the flight data. The events file may also be imported into the Data Viewer program to automatically annotate the data and direct the use to data of interest. An example of an events file output is presented in Figure 19. The first column in the file is elapsed time in seconds since midnight in local time.

## **F. Software Development Testing in Simulation**

There were several methods that were developed to test the GRT Real-time software on the computer throughout the project. The ability to test the software and interfaces to the aircraft in simulation greatly streamlined the development of the software.

### ***1. Early Development Testing Using Serial Port Emulators***

Initially, the aircraft systems were powered up on the ground. The “T” harness was installed and a temporary RS-232 to USB hub and laptop was connected. The Windows XP utility called HyperTerminal was used to record log files of the transmission of each of the GRT devices. These separate log files were



then replayed on a laptop using a commercially available serial port data emulator program. In addition to the serial port data emulator, virtual serial port pairs were installed on the test laptop such that Extra data transmitted on three serial ports were mirrored on virtual hardware and made available on the same laptop without the need for any physical hardware. These virtual serial port pairs are referred to as “Virtual Null Modem Cables”. This allowed early builds of the acquisition subVI’s of the GRT RT to be tested as they were designed and built.

## ***2. Virtual Hardware Test Methods***

As development of the software progressed, it was necessary to be able to simulate flight data and not just ground data. For this purpose, a program was developed which is referred to as the “Extra System Simulator”. The user interface for this program is presented in Figure 20 of the Appendix. This program simulates the GRT devices of interest installed on the Extra 300. The program reads in a Demo Log file and then broadcasts the data sentences from each device on separate serial ports at the same time intervals as the original data was recorded by the EFIS.

Again, virtual null modem cables were used. In this manner, the Extra System Simulator and the GRT RT software could be run on the same laptop. While this worked well for software development, it did have one significant limitation. The Virtual Null Modem method had a reduced data throughput and slowed the data transmission.

## ***3. Hardware in the Loop Test Methods***

The final test method used was the combination of the Extra System Simulator with actual flight hardware. In this scenario, the GRT RT software was installed and run on the Samsung Q1 UMPC and the Extra System Simulator was run on a laptop. The two computers were then connected using two RS-232 to USB hubs and actual null modem adapters as shown in Figure 21. As such, the UMPC running the GRT RT software responds exactly as if it were installed in the real aircraft in flight. This test method was used to validate the software before the first ground and flight tests and is still used to validate new software builds before being rolled out on evaluation flights. The ability to evaluate a software build in the lab has greatly streamlined the process of validating software after updating and adding features.

## **IV. Testing Approach and Methodology**

The testing of the EFIS based data acquisition system began with developmental testing of the GRT RT program using the Extra System Simulator and calibrations of the various sensors on the GRT EFIS system. This was followed by on aircraft testing on ground power, ground testing with all systems turned on and the engine running, and finally, flight evaluations of recording on both the EFIS Demo Log and the GRT RT in-flight data acquisition system.

### **A. Theoretical Maximum Data Rate Test**

Further testing was conducted using the Extra System Simulator to determine the maximum theoretical acquisition rate of the UMPC and GRT RT software. The data output rate of the AHRS is fixed at 15 Hz. This test was intended to see how the software would perform on a theoretical high rate AHRS system. The Extra System Simulator was modified to output AHRS data at a user specified rate while outputting GPS and EIS data at the normal rates.

### **B. Ground Testing**

Prior to ground testing, the airspeed and altitude sensors of the Extra were calibrated using an (RVSM) compliant calibration set following instructions in the GRT User's Guide [14]. Following the calibrations, initial ground tests were performed with the aircraft in the hanger on a ground power unit and were conducted to ensure the software performance on the aircraft was the same as on the Extra System Simulator. To safely conduct ground testing, the original intermittent duty relay for the ground power supply on the Extra was replaced with a continuous duty relay at the time of the EFIS installation. This allowed the systems to be operated for extended periods on ground power. After initial testing on ground power, a full up ground test was conducted with all systems powered up and the engine running. The ground run test served as the final evaluation prior to receiving approval for limited flight evaluations and included the following tests:

#### **1. *Electro-Magnetic Interference (EMI) Checks***

The EMI test plan involved activating the aircraft systems one at a time and monitoring for interference. The systems were first turned on one at a time and tested without the data acquisition system online. Then

all systems, except the engine, were shut down. The systems were then restarted in sequence with the data acquisition system online and logging. Data recorded on the EFIS and the UMPC was inspected for unexpected data losses. Ancillary systems were also tested and checked for interference. One such test of interest was the affect of radio transmissions on the magnetometer and heading solution. Due to the relatively small size of the aircraft, there are limited options for installing devices. The VHF radio antenna is located in leading edge of the tail while the magnetometer is located in the tail cone. A test was conducted by first observing the heading solution without transmitting. The radio was then keyed for typical time durations as would be expected in normal use and the affect on the magnetometer raw values and AHRS heading solution was observed.

### ***2. AHRS Attitude and Heading Drift Check***

In addition the EMI check of the heading, the attitude and heading solutions were checked for stability and drift during the ground run. This was carried out by observing the pitch, roll, and heading values while the aircraft was secured in the chocks with the engine running.

### ***3. Power Sweep***

A power sweep was conducted from idle power to the highest safe power setting manageable on the ground. The resultant data was then inspected for increases in electrical noise and changes in vibration levels in the accelerometers and MEMS gyros at the various RPM's. Since full power cannot be safely obtained on the ground, a second power sweep through the upper RPM settings will be performed in flight to observe the levels of electrical noise and vibrations at the higher power settings.

## **C. Flight Testing and Evaluations**

Following the completion of ground testing, the flight testing and evaluation was planned for the aircraft. The flight tests and evaluations were intended to check the quality of the recorded data and demonstrate the usefulness of the system by performing routine flight test maneuvers.

### ***1. Evaluation of Timestamp Accuracy***

Since the GRT RT program is not run on a hard real-time operating system, it was necessary to evaluate the quality of the data timestamps. The timestamp accuracy of the GRT Real-time software can be compared to the timestamp accuracy of the EFIS itself by analyzing an interval of data from a flight where there is both EFIS recorded data and UMPC recorded data and comparing the scatter of elapsed times

between the timestamps. This comparison is made by assembling a histogram of the elapsed time recorded by each device.

## **2. *Power Sweep***

A power sweep was conducted in flight at higher power settings than those tested on the ground. The procedure for the power sweep is to select RPM settings between 2000 and 2500 RPM. The manifold pressure is adjusted as required to maintain the target engine RPM.

## **3. *Static Pressure Position Error Correction***

Static Pressure Position Error is the difference between the static pressure measured by the aircraft static ports and the true ambient static pressure outside of the air which is disturbed by the passing aircraft. The Position Error Correction accounts for this error with a correction factor. A four-sided wind-box is flown maintaining constant heading, altitude, and airspeed on each side of the box. GPS supplied course and ground speed data from any three sides of the box can be used to calculate a wind vector and the true airspeed of the aircraft. The difference between the GPS supplied True Airspeed and the Equivalent Airspeed measured by the aircraft Pitot-Static system is the Position Error Correction for airspeed [15]. From this value, the correction for altitude can be calculated.

This test must be flown under smooth, turbulence free conditions. Each side of the wind-box must be flown at the same airspeed and altitude. The desired headings for the box are 315, 225, 135, and 45 degrees; however the headings do not need to be flown on the exact headings as long as the heading is held constant during each data point. The EFIS records all required data for data reduction.

## **4. *Maneuver Evaluations***

Several Stability and Control test points were flown to demonstrate the capabilities of the data acquisition system; however, this should not be considered a Stability and Control evaluation of the aircraft.

### **Long Period Longitudinal Response**

The long period response is a longitudinal oscillation of an aircraft about the pitch axis and is a measure of the longitudinal dynamic stability. The long period response of the Extra was investigated to demonstrate the usefulness of the data acquisition system for stability and control evaluation tasks. Investigation of the long period longitudinal response of the Extra begins with a trim shot at the desired test

air speed and altitude. Then, without changing the power setting, the aircraft is pitched up slowly and gently to slow 5 to 10 kts below the trim speed. The back pressure on the control stick is then relaxed slowly until the stick returns to the trim position. The pilot then holds the controls fixed or free as desired. The typical aircraft response is a slow oscillation in pitch with the aircraft trading energy between altitude and airspeed. The EFIS records the variation in Airspeed, Altitude, and pitch throughout the maneuver. Angle of Attack is normally recorded also, but it is not currently available on the Extra.

### **Short Period Longitudinal Response**

The short period longitudinal response of the aircraft was also evaluated to demonstrate an additional stability and control certification task using the data acquisition system. The first step to exciting the aircraft short period response is to trim the aircraft at the desired altitude and airspeed. Next an elevator doublet is applied and the controls are returned to the trim position and either held fixed, or left to float free. The response of interest is a short period oscillation in the pitch axis with changes in angle of attack. The EFIS records the variation in airspeed, altitude, and pitch throughout the maneuver. Angle of attack is normally recorded also, but it is not currently available on the Extra.

### **Dutch Roll**

Dutch roll is a lateral oscillation with a coupling between the yaw axis and the roll axis. The typical measure of the Dutch Roll mode is the number of cycles to half amplitude, and the  $\phi/\beta$  ratio where  $\phi$  refers to the roll angle and  $\beta$  refers to the angle of sideslip [16]. The Extra does not have the ability to measure and record the angle of sideslip, however, the heading of the aircraft is recorded. Since the Dutch Roll produces an oscillation about the yaw axis, the aircraft heading can be substituted for the sideslip angle. While this does not produce an exact  $\phi/\beta$  ratio, it does allow the determination of the cycles to half amplitude.

### **Spiral Roll**

The spiral mode is a measure of the behavior of an aircraft after the wings have been disturbed from a level attitude [16]. The test point is performed by first trimming the aircraft at a desired speed and heading. The pilot then rolls the aircraft to a small bank angle using the rudder only. Once the bank has been established, the rudder is slowly brought back to the neutral position and the change in bank angle is

observed. The spiral roll is normally investigated to the right and left, and with the control free floating and held fixed.

### **Stalls**

Approach to landing stalls are typically performed in the landing configuration with either idle power or the power setting which produces zero thrust [16]. The propeller pitch is set to full fine, and the aircraft is trimmed at an airspeed which is 1.5 times the expected stall speed. The aircraft is then flown in the stabilized descent while slowing at a deceleration rate of 1 kt/s. The stall speed is the speed in which the aircraft exhibits an uncontrolled change in pitch, roll, or yaw, or when the stick reaches the aft stop. The recorded stall speed is then corrected to gross weight according to equations presented in Kimberlin's text [16].

### **Normal Upright Spins**

The normal upright spin is entered in the same manner as the Stall, except with power reduced to idle, and with special attention paid to ensuring the slip/skid ball is centered in the lead up to the spin. As soon as there is an indication of stall, the stick is brought fully aft and held against the stops, and rudder is pushed fully in the direction of desired spin. The recovery for the spin follows the NACA method: Power to Idle, Ailerons to Neutral, Rudder Fully Opposite the Spin, and Elevator Forward through Neutral.

### **Maximum Rate Aileron Roll**

An aileron roll is performed by pitching up slightly, checking forward on the stick to neutral elevator, and applying full aileron deflection for a rotation through 360 degrees. This tests the AHRS response to high rotational rate maneuvers. It is expected that the AHRS will enter an in-flight reset during this maneuver.

## **V. Results and Discussion**

The complete set of tests was conducted as planned. Learning from each test led to improvements in the program and retesting when necessary before beginning the next planned test. The final results of each test are presented here.

### **A. Theoretical Maximum Data Rate Test**

The maximum data rate achieved at the current AHRS baud rate was 50 data sentences per second. Using higher baud rates allowed up to 100 sentences per second to be received, decoded, and logged. Even at these rates, the load on the 600 MHz UMPC CPU is not noticeably increased. The practical maximum sampling rate would be defined by the capability of the device sending the data and quality of the received RS-232 data signal.

### **B. Ground Testing**

The testing on ground power was straight forward and illuminated few problems. The software performed in a similar manner as it had on the Extra System Simulator. Use of actual aircraft hardware provided no changes to the software or hardware configuration. The aircraft was then cleared for ground testing with the engine running.

#### ***1. Electro-Magnetic Interference (EMI) Checks***

The EMI tests were performed as planned. The addition of the isolated RS-232 hub and UMPC did not produce any observable interference during ground testing. Tests did however show that the raw magnetometer output was affected by various systems on board the aircraft. The raw magnetometer solution was shifted by a constant bias of 4 degrees when the alternator was switched on and off. Over a long period of time, this would create a 4 degree error in the AHRS heading solution. Given the limited choice of locations available for the magnetometer, no improvements have been made.

The VHF radio also affected the accuracy of the raw magnetometer values. The raw solution swung by as much as 10 to 15 degrees during long periods of radio transmission (20-30 seconds). The affect of the above issues on the AHRS yaw solution is somewhat reduced due to the use of Kalman filters to blend sensor inputs. During the entire duration of the ground run of approximately 20 minutes, the aircraft was

secured in chocks, and the AHRS yaw solution varied by 5 degrees. The largest variations occurred during radio transmissions, the typical variation in the AHRS yaw solution for the remainder of the time was 2 to 3 degrees.

Two software bugs were discovered and fixed during the ground run. The first corrected missing GPS altitude data and the second resulted in the placement of a small buffer and sorting routine between the acquiring of the data and the writing to log to ensure sequential timestamps in the log file.

## **2. *AHRS Attitude and Heading Drift Check***

AHRS pitch and roll solutions were stable throughout the entire ground run with the total variation in the solution being less than 0.4 degrees in each axis. Predicted variation of the roll and pitch solution in a static condition is  $\pm 0.5$  deg, according to the AHRS specification sheet [11] published by GRT. Accuracy of the roll and pitch solution in flight was not verified.

## **3. *Power Sweep***

Variation of the X, Y and Z accelerometers with the engine running at 1000 rpm was confined to 0.05G, 0.02G, and 0.07G, respectively. The period of the variation in accelerometers was on the order of 1 to 4 seconds and is likely due to prop blast buffeting the tail. There were no higher frequency vibrations evident in the data. The power sweep was applied from idle at 800 RPM to the run up power setting at 1800 PRM. The typical variation in the X, Y, and Z axes was limited to 0.04G, 0.04G, and 0.4G, respectively. Again, the period of the variation in acceleration for the three axes was typically between 1 and 4 seconds suggesting the cause was buffeting of the tail at higher power settings with the Yaw axis receiving the worst of the buffeting. There was no evidence of any resonance with higher frequency vibrations during the power sweep. There was also no evidence of increased electrical noise or interference during the power sweep.

## **C. Flight Testing and Evaluations**

Ground testing identified only two minor software problems, both of which were fixed and newest build of the software was testing using the Extra System Simulator. The aircraft was cleared for the in flight evaluations.



### ***1. Evaluation of Timestamp Accuracy***

A 10 minute time interval was taken from both the decoded EFIS Demo Log file and the UMPC recorded GRT Real-time ASCII data file. The two data sets spanned the same time interval during actual flight. The elapsed time between every successive data point in each file was plotted in a histogram. Only messages of interest from the AHRS, EIS, and GPS in the Demo Log file was graphed, maintenance messages were ignored.

Data from the two sources were plotted side by side in the same histogram plot to determine the relative quality of time stamp accuracy between the two logging devices. Figure 22 presents the elapsed times for both devices. There is considerable scatter in the recorded elapsed times from each recording device for each of the three data sources. The elapsed times recorded by the EFIS are generally more diffuse for the AHRS and EIS while the GRT RT recorded elapsed time are more concentrated in several groups. In general however, there does not appear to be a severe difference in performance between the EFIS and the GRT RT software. As for the GPS data, the GPS onboard the Extra calculates one solution per second and then transmits this data at a low data rate through seven different NMEA messages, of which two are of interest and are decoded by the two programs. The EFIS software records the two separate messages and applies two separate time stamps, even though the both data sentences originate from the same GPS solution at the same time. The GRT RT software collects both NMEA messages, combines the data, and assigns one timestamp from when the first message arrived. As such, the two elapsed time spikes recorded by the EFIS Demo log add up to a data rate of 1 Hz, and the single elapsed time stamp recorded by the GRT RT program is also at 1 Hz. The timestamp of the GPS recorded by the GRT RT software is more accurate even though the elapsed time is slightly more scattered.

### ***2. Power Sweep***

The in-flight power sweep was performed between 2000 and 2500 RPM; manifold pressure was adjusted accordingly for each RPM setting. There was no evidence of excessive vibration or increased electrical noise or interference at any of the power settings during the power sweep.

### ***3. Static Position Error Correction***

Several air-data calibration flights were flown and the data was reduced using two separate GPS data reduction methods, Kimberlin's GPS reciprocal heading method [16], and the wind-box method [15] as

detailed by the National Test Pilot School (NTPS). Initial results varied significantly from the published values. The results indicated a very large error at low speeds and were in opposite magnitude to the manufacturer's values. The flight test department at Extra Aircraft LLC was consulted. From these discussions, it was discovered that the pitot-static tube installed on N714X had been modified since the aircraft was originally manufactured and no-longer conformed to the manufacturer's specifications. A new pitot-static tube was procured and installed. The new tube was also slightly out of specification, but was much closer to the recommended specifications than the old tube. Air-data calibration flights were repeated, this time using only the NTPS GPS wind box method as it was found to be easier and faster to fly accurately. The results are presented in Figure 23 and Figure 24 along with the manufacturer's published data for the airspeed calibration. The new data set still differs from the manufacturer's data; however, it is much closer and the current pitot-static tube provides reasonably accurate airspeeds near stall.

#### ***4. Maneuver Evaluations***

##### **Long Period**

The Long Period test points were flown; however, the dynamic response of the Extra was heavily damped and the resultant data traces are not of value for demonstrating the value of the system. Two particular characteristics of the Extra which may lend itself to the sedate response to the Long Period excitation are the inordinately large tail volume and large wing with a thick airfoil. The relative drag to mass ratio of the aircraft would be higher than that found on most aircraft and may have contributed to the strong damping of the oscillation.

##### **Short Period**

The Short Period test points were also flown; however, the dynamic response of the Extra was heavily damped and the resultant data traces are not of value for demonstrating the use of the system. The Extra 300 is an aerobatic aircraft with near neutral static stability. Since the aircraft is not overly stable, there is not a strong restoring tendency following a deviation from trim. Hence, the initial energy in the short period oscillation is low and damps very quickly, producing no discernable overshoot.

## **Dutch Roll**

Figure 25 presents the Dutch Roll data traces for roll angle and heading versus elapsed time. The data trace indicates that the Dutch Roll was completely damped after two to three cycles. While the Dutch Roll damps out rather quickly, it is extremely lively in the yaw axis. This is likely caused by the higher than usual rudder authority and the attention paid to minimizing the weathercock stability of aerobatic aircraft. The cross plot also shows considerable more movement about the yaw axis than about the roll axis. The data density of all plots is more than sufficient to accurately describe the motion of the Dutch Roll mode.

## **Spiral Roll**

The Extra 300 Spiral Mode response was investigated for disturbances to the left and to the right, for the controls fixed and the controls free cases. Figure 26 presents the results from all four cases. Both cases to the right were divergent, while both cases to the left returned towards wings level flight. It is likely that the difference in response between spirals to the left and right is caused by propeller affects. The controls fixed case to the left returned to wings level and then continued to diverge to the right. It is possible that the rudder position held in this case was not exactly neutral.

## **Stalls**

The approach to landing stall data recorded using the EFIS Demo Log function illustrates the value and quality of the data available from the low-cost EFIS systems. Figure 27 presents data traces for the stall. The stall was approached at a constant 1 kt/s rate of deceleration. The stall break is well defined with an un-commanded drop of the right wing with a slight yawing to the right, and a drop of the nose at 17 seconds into the trace. The weight corrected stall speed [16] was found to be 66.1 kts as measured from the original production pitot-static tube. The stall occurred at higher than expected airspeeds. Later pitot-static calibration flights pointed to the non-conforming pitot-static tube as the source of the discrepancy. The installation of a swivel head boom would likely improve the data accuracy even further for the stall speed determination. The data recording rate of 15 Hz for the AHRS was more than adequate to provide accurate results for stall testing.

### **Normal Upright Spin**

Spin data from a decoded Demo Log File is another example of the value of data recorded by a low-cost EFIS system. Figure 28 presents time histories for the spin. The entry altitude for the spin was 4664 ft and the recovery altitude was 3612 ft. A total altitude of 1052 ft was lost during the 1 turn spin and subsequent recovery. The descent rate just before the initiation of the recovery was 2200 fpm and still steadily increasing. The largest descent rate was reached in the recovery and was 7086 fpm.

Pitch initially dropped to 60 degrees nose low, and then pitched up to 30 degrees nose low before pitching back down again. The pitch reached a minimum of 83 degrees nose low during the recovery.

Rotational rates about the yaw and roll axis continued to vary throughout the spin. The maximum yaw rate achieved was 54 deg/s to the left. The maximum roll rate was 131 deg/s to the left. The 15 Hz data acquisition rate provides good data density even in highly dynamic maneuvers such as a spin. The addition of angle of attack, sideslip and control position and forces would provide a very complete picture of what is happening in a spin.

### **Maximum Rate Aileron Roll**

Maximum rate aileron rolls were performed during multiple flights. In each case, the AHRS entered an in-flight realignment which blanks out the artificial horizon for the pilot. Figure 29 presents a data trace from one such aileron roll. The reported roll rate in the data trace maxes out at 240 deg/s and is very much constant at this value suggesting that the aircraft rolled at a higher rate, but the AHRS does not output higher rates. Although an in-flight realignment of the AHRS blanked out the pilot's artificial horizon for up to two minutes, the attitude solution is available in the recorded AHRS data message and appears reasonable and accurate.

## **D. Performance of the GRT System**

The GRT GPS, AHRS, and EIS were found to provide a good low-cost source of digitized data for flight test purposes. The quality of the EIS data is more than sufficient for most applications. The GPS included with the EFIS was also sufficient for many flight testing applications, but the addition of a differential GPS would likely be a benefit for takeoff and landing certification testing. The AHRS data

seemed to be accurate and reliable, but special attention should be made to the location of the magnetometer, especially and a small airframe such as the Extra 300.

The only hardware limitation encountered in flight was the angular rate limit of 240 deg/s for the AHRS. This rate limit was only encountered when using the full roll capability of the aerobatic aircraft. Even though the AHRS entered an in-flight realignment, the AHRS continued to transmit a full set of data. Only the roll rate was capped at 240 deg/s. When this limit was discussed with Grand Rapids Technology, they offered solutions for extending the rotational rate limits to 400 deg/s and even over 500 deg/s if required.

The various acquisition devices are not time synchronized, and the time stamp must be applied in the recording device. While this always creates some uncertainty in the true time of the data, it does ensure that data from all sources is always synchronized. Variation in the recorded timestamp from the devices was measured and quantified. This variation was small and acceptable for data record by the EFIS and the UMPC running XP and GRT RT.

The basic GRT system allowed for the acquisition of some additional analog parameters, but the real-world usefulness of these auxiliary analog inputs was limited by the accuracy of the analog inputs to the EIS and EFIS, and the limited data rates of these two devices.

The Demo Log function of the GRT EFIS allowed for the use of the system as a Flight Test tool without any modification, however, the system would have been considerably more useful if the onboard memory was increased to hold data from a full flight and store the data in non-volatile memory.

In summary, the basic GRT EFIS system provided a good low-cost source of flight data, but without the addition of in-flight data acquisition, was found to be lacking the full capability needed for many flight testing applications. The use of LabVIEW and a UMPC in this function takes full advantage of the data provided by the GRT EFIS and provides the user with growth and expansion capability, the possibility of custom made user interfaces, and support for additional devices and software in-flight.

## **E. Ease of Growth and Expansion**

LabVIEW offers recording of Audio and Video streams, direct support for serial, Bluetooth, and TCP/IP communications, and is very effective at decoding binary and ASCII data. LabVIEW also supports data logging in common ASCII formats, binary formats, and common database formats. LabVIEW alone

can interface with most instruments using a digital interface and display and log the data. In addition to the abilities of LabVIEW, National Instruments manufactures devices for analog and digital data acquisition and control in most common industry form factors. This includes devices of special interest to flight test teams such as avionics bus monitors and IRIG synchronization and timing. For applications requiring deterministic real-time operating systems, more advanced LabVIEW licenses offer such deployments for hardware ranging from ordinary personal computers to industrial acquisition form factors.

The use of the Samsung UMPC combined with an inexpensive addition of a National Instruments USB-based data acquisition device to acquire additional analog parameters would allow for a significant level of low-cost capability for a system based on a COTS glass cockpit. Additional devices could easily be added and integrated as needs of the test team grows and matures.

Additionally, the UMPC and the “GRT Real-time” software were tested to see if it would perform well with higher rate devices. This test showed that the software and UMPC could easily handle data rates up to 100 Hz with the limiting factor being the RS-232 device I/O, and not the CPU. If necessary, devices with higher data rates than the standard GRT hardware could be used as a data source. The data rate limits for RS-232 communications would not apply to USB acquisition devices which are often rated for as much as 250,000 samples per second, per USB acquisition device.

## **F. Post Processing Work Flow**

The custom made Data Viewer allowed users to quickly navigate the data from an entire flight, annotate data of interest, and export the required data for processing and reporting. This program has already been introduced to a class of students for post flight use with data from the Extra and another instrumented aircraft in use at the school. The student and faculty feedback has already been incorporated into improved versions of the program. This author believes there is a significant commercial market for a software application with similar capabilities.

## **G. Usefulness for Flight Test Certification Programs**

While there is no requirement for airborne data acquisition for Light Sport or Part 23 aircraft certification, the use of low-cost airborne data acquisition offers the test team an efficient means of recording the necessary flight data for certification and an easier means to show compliance to the

regulatory authorities. The desired data for a full certification program would include that available from the GRT EFIS, control positions and forces, and an air-data boom complete with angle of attack and sideslip. The total pressure and static ports of test boom could be piped directly into the GRT AHRS without modification. The remainder of the parameters could be acquired with a low cost National Instruments USB data acquisition device.

## **H. Usability of the User Interface in Flight**

The Samsung UMPC has been used in flight on a Piper Navajo and Extra 300 for data display and logging. The screen has proved to be perfectly readable in the Piper Navajo and has sufficient brightness for use in the Extra 300 in full sunlight. However, the user interface has to be kept clean and simple to allow for easy use. In addition, all user buttons on the interface have to be kept large to make use of the touch screen easy in a moving aircraft. The Samsung Q1 has both a small split physical keyboard and an onscreen keyboard. Both are very difficult to use in the Extra, making a well designed user interface even more important.

The Setup Tab with intuitive and informative alarms proved very useful. In addition, it is necessary to have clear instructions for the FTE to take when an alarm is encountered in flight. The Data Tab proved to be well laid out, but the option to display data in a cross-plot format would have made visualization of certain test data such as the Dutch Roll much easier. Also, the incorporation of a second block of digital parameters and possibly extra graphed parameters would allow the user to prepare data displays for each anticipated test point before the flight. This would allow the test engineer or test pilot to perform their duties with less heads down time between test points. The Events Tab proved useful, but the advanced features were not entirely necessary. The writing pad proved to be difficult to use in flight.

## **I. Development Time and Cost**

The total hardware cost of the tested data acquisition system was in the range of \$11,000 including the purchase cost of the GRT EFIS, Samsung Q1 UMPC, and optically isolated RS-232 to USB hub. Table 2 contains the list price of each device at current prices along with the installed weight of the major devices on the aircraft. The weight of wiring harness was not included as each installation would be unique. Additional development costs would include the labor to install and calibrate the system, the cost of

**Table 2 Hardware Weight and Cost Budget**

Device	Weight <sup>1</sup>	Cost <sup>2</sup>
GRT EFIS 1 with GPS	2.3 lb	\$2000 + \$400
GRT EFIS 2 <sup>3</sup>	2.25 lb	\$2000
GRT AHRS	2.1 lb	\$3000
Magnetometer	0.6 lb	Included with AHRS
GRT EIS	0.9 lb	\$1550
Samsung Q1	1.5 lb	\$950 <sup>4</sup>
RS-232 to USB Hub	1.35 lb	\$180

<sup>1</sup> As weighed prior to installation with mounting hardware. Not including wiring harnesses.

<sup>2</sup> Cost of GRT devices as of March 2009. Other pricing schemes available for packages.

<sup>3</sup> Installation of a second EFIS is optional.

<sup>4</sup> Typical pricing as of March, 2009.

LabVIEW software licensing, and the programming time to develop custom applications in LabVIEW. Although the cost of LabVIEW licenses may be seen by some as excessive, the cost of the software license is more than offset by the ease of which LabVIEW can be implemented. Program development can be carried out within the test team, or by a contractor with program maintenance and updating carried out by members of the test team.

The most likely application for this type of system is for a Part 23 certification program with an aircraft that is already equipped with a glass cockpit. In these cases, added cost of the data acquisition hardware may be as little as \$10,000 per airframe depending on exactly what additional sensors and signal conditioning is required. While each program developed was continuously improved with added capability, the initial versions of the GRT Post Processor and the Data Viewer programs took less than 80 man-hours each to complete. The first useful version of the GRT Real-time software likely took 160+ man-hours to develop and test. In all, the total project duration to turn an installed GRT EFIS into a data acquisition system with all the afore-mentioned software tools and testing required 8 months for one person working part-time. The time required to implement this system for an actual flight test program would be considerably shorter if an employee could be dedicated to the task fulltime. The only prior knowledge the author had of LabVIEW at the onset of the project was one course on the subject taken at UTSL.



## **J. Unresolved Issues**

Three flights were flown using the UMPC and GRT RT software, and one ground test was made. Two unresolved errors were encountered during these flights. During the first flight, 598 AHRS messages were lost during the first 13.5 minutes of flight, most likely due to electrical noise or grounding issues. This equated to about 10% of the AHRS data messages being filtered out by failed checksums. The resultant AHRS data during this time contained many data spikes where the checksum passed, yet there was in fact an error in the data sentence. Most of this bad data could be filtered out if a purpose designed algorithm was developed.

On the third flight, similar errors occurred in data originating from the GPS. Multiple messages failed the checksums, and multiple messages indicated that the GPS lock was lost. In addition, the logged data from all sources contained considerable spikes. The data recorded by the EFIS at this time was clean and reported a good GPS lock throughout.

## **VI. Conclusions**

The stated objective of this research work was to develop an acquisition system based on low cost commercial off the shelf hardware including a glass cockpit system. The research work was to give consideration to post processing work flow, in flight access to test parameters, growth and expansion capabilities, and data quality and usefulness as desired for flight test applications.

The primary design goals were met and the minimum required parameters were made available. The system requires no power from the aircraft, has a minimal weight and space impact, and will easily accommodate future growth and needs. The system provides a basic set of data to the user in flight in the current configuration, and additional programming would greatly enhance in-flight data validation. Methods, software tools, and a work flow have been developed not only for processing of data after the flight, and also identifying test data during the flight. Additionally, the choice of hardware and software for in flight data acquisition offers the user the ease of system growth and expansion in capability that is required for most flight testing or research applications.

A method was developed and refined for quickly identifying and selecting pertinent data post flight for further analysis. These post flight tools have been tried out and put to use by the first class of students with total success.

The use of the Demo Log file provides an easy method to record data and serves as a good backup flight record even when the GRT-Real-time program is used. The use of GRT Real-time in-flight offers more interaction with the flight data and better visualization of parameters, but can also be a distraction for single pilot operations.

Though a nagging problem with failing checksums and garbled data passing the checksum remained at the completion of the testing phase, it is believed that these problems would have been overcome with further testing.

### **A. Recommendations for Future Research and Development**

- 1) Configure interfaces for common certified glass cockpit systems to allow low cost flight testing on aircraft with production systems installed.

- 2) Future development work should attempt to incorporate in flight calculations, parameter limit alarms, and maneuver-specific algorithms to streamline test point validation in-flight and reduce post-flight data reduction.
- 3) A low cost solution for data telemetry should be developed using low-cost commercial hardware. To be effective, the solution would need to work without the need for expensive tracking antennas. Possible transmission devices may include long range radio modems, wireless internet, or use of consumer cellular networks for data transmission.
- 4) The ability to replay flight data and view the flight in a virtual world would allow for the presentation of flight data in a more understandable fashion. Such visualization could be conducted in real time in a telemetry room, or after the flight from a saved data file. Some consumer-grade flight simulation programs have the ability to direct the aircraft by externally supplied attitude and position data. A relatively simple interface could be developed to provide flight test teams with 3D flight visualization.

## References

## List of References

- [1] Klijn, J. M., "Development of Flight Test Instrumentation: an Evolutionary Approach," National Aerospace Laboratory NRL, NRL-TP-2006-407, 2006, pp. 6, 1-18.
- [2] Cusimano, G. and Roberts, N., "A Century of Flight Test – The More Things Change, the More They Stay the Same," *Society of Flight Test Engineers 34<sup>th</sup> Annual International Symposium*, 2004-34-11, Society of Flight Test Engineers, 2004, pp. 8-9
- [3] Mace, W. D. and Pool, A., "AGARD Flight Test Instrumentation Series," North Atlantic Treaty Organization Advisory Group for Aerospace Research and Development, AGARD-AG-160, Vol. 1-19, 1973.
- [4] Buergel, V., "Icasim Smart Air Data Boom," [online product specification sheet], 3<sup>rd</sup> Edition, Simtec Buergel Ltd., URL: <http://www.simtec.ch/images/stories/Pdf/Flyer-Icasim-Smart-Air-Data-Boom.pdf> [cited March 2007].
- [5] Iscold, P. H., Mol, F., Brasil, N. S., and Maciel, B., "Development of Light Aircraft Flight Test Equipment," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA 2006-6004, AIAA, Washington, DC, 2006, pp. 6
- [6] Liu, T. and Schulte, M., "Flight Testing Education at Western Michigan University," *AIAA Aerospace Sciences Meeting and Exhibit, 45<sup>th</sup>, Reno*, AIAA 2007-0700, AIAA, Washington, DC, 2007, pp. 1-21
- [7] Garman, K. E. and Andrisani D. A., "A Portable Data Acquisition System for Flight Testing Light Aircraft," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA 2003-5618, AIAA, Washington, DC, 2003, pp. 1-9
- [8] LabVIEW, LABORatory Virtual Instrument Engineering Workbench, Software Package, Ver. 8.5, National Instruments, Austin, TX, 2007.
- [9] Duke, E. L., Duke, W. C., Vanderpool, C. C., and Mork, B., "Carry-On and Quick-Install Instrumentation for Flight Test," *Society of Flight Test Engineers 37<sup>th</sup> Annual International Symposium*, 2006-37-11, Society of Flight Test Engineers, 2006, pp. 1-12
- [10] "GRT Horizon Installation Manual," [online product manual], Revision A, Grand Rapids Technology, URL: [http://www.grtavionics.com/File/Install Manuals/Horizon Install Rev A February 2009.pdf](http://www.grtavionics.com/File/Install%20Manuals/Horizon%20Install%20Rev%20A%20February%202009.pdf) [cited Feb 2009].
- [11] "Horizon AD/AHRS," [online product specification sheet], Grand Rapids Technology, URL: [http://www.grtavionics.com/File/AHRS Flyer.pdf](http://www.grtavionics.com/File/AHRS%20Flyer.pdf) [cited March 2009].
- [12] NMEA-0183, National Maritime Electronics Association, Electronic Interface Standard, Ver. 3.01, National Maritime Electronics Association, Severna Park, MD, 2002.
- [13] MATLAB, MATrix LABoratory, Software Package, Ver. 2007a, The Mathworks, Inc., Natick, MA, 2007.
- [14] "GRT Horizon User's Guide and Reference," [online product manual], Revision A, Grand Rapids Technology, URL: [http://www.grtavionics.com/File/User Manuals/Horizon User Rev C Feb 2009.pdf](http://www.grtavionics.com/File/User%20Manuals/Horizon%20User%20Rev%20C%20Feb%202009.pdf) [cited Feb 2009].

- [15] Lewis, G., "Using GPS to Determine Pitot-Static Errors," [National Test Pilot School online unpublished report], 14 Aug 2003 Edition, URL: [http://www.ntps.edu/Files/GPS\\_PEC\\_Method.doc](http://www.ntps.edu/Files/GPS_PEC_Method.doc) and [http://www.ntps.edu/Files/GPS\\_PEC.XLS](http://www.ntps.edu/Files/GPS_PEC.XLS) [cited March 2008].
- [16] Kimberlin, R. D., *Flight Testing of Fixed-Wing Aircraft*, AIAA Education Series, AIAA, Reston, 2003, pp. 37, 314-322

## **Appendices**

## Appendix 1



Figure 1 Extra 300

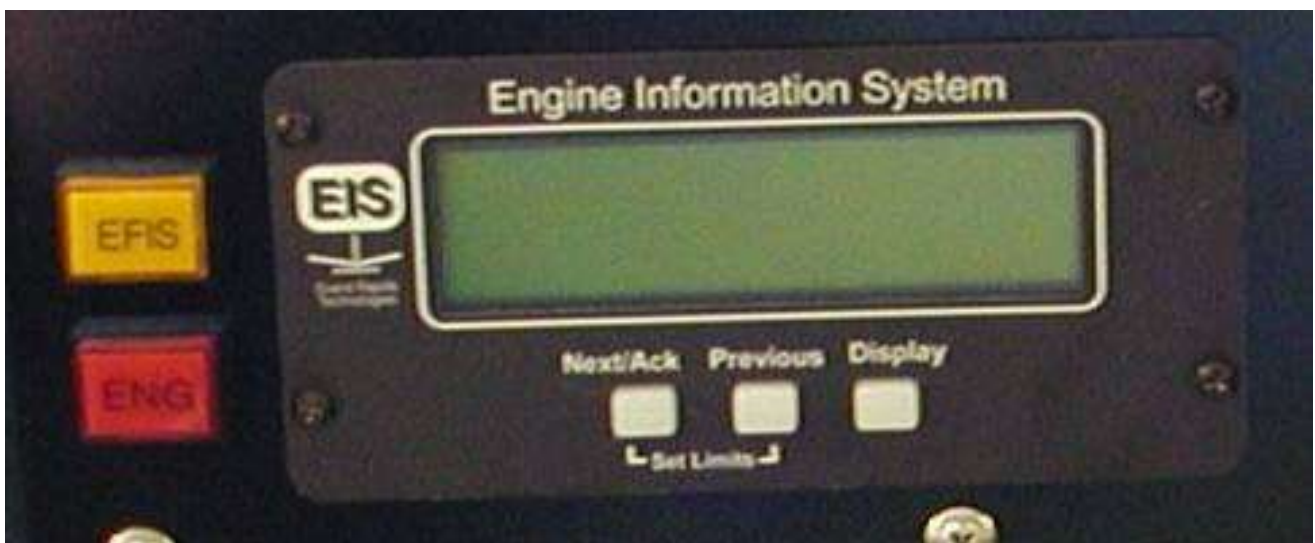


Figure 2 Rear Instrument Panel





**Figure 3 Video Equipment in Use on Red Bull Aircraft**



**Figure 4 GRT Engine Information System**



**Figure 5 GRT EIS Transducers**



**Figure 6 GRT Attitude, Heading, and Reference System**



Figure 7 GRT Magnetometer



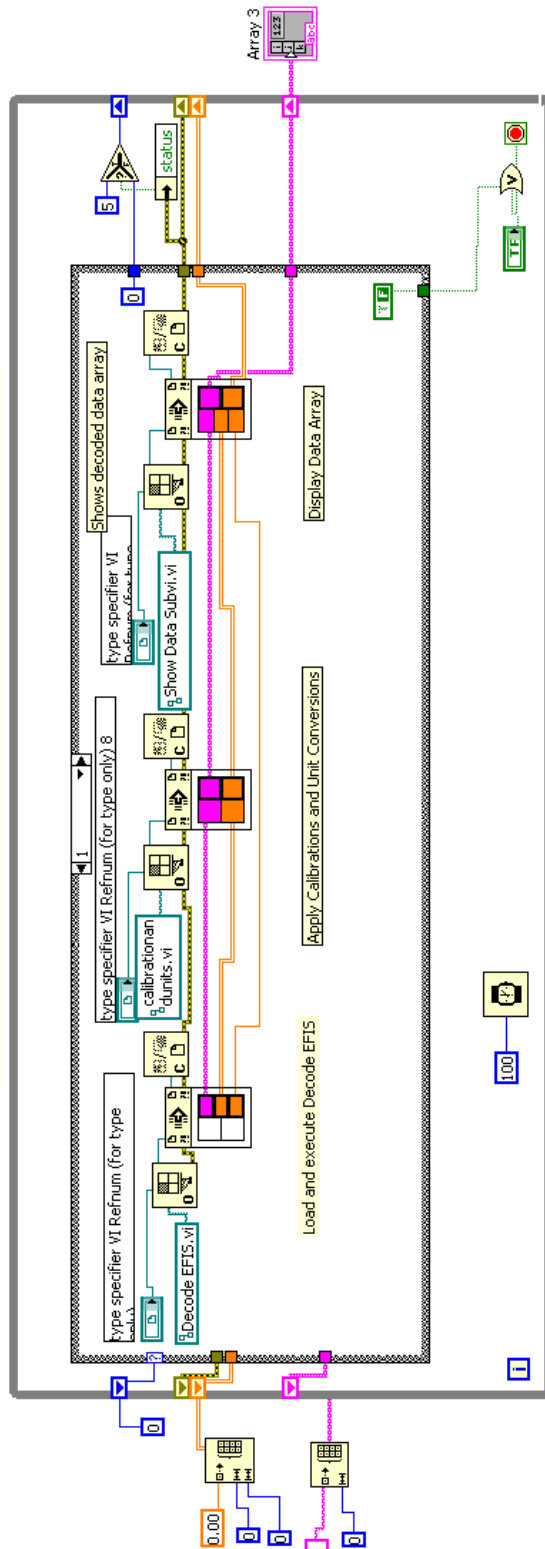
Figure 8 GRT Electronic Flight Information System





Figure 9 Post Processor User Interface

Each case loads and calls a VI dynamically. Each VI is loaded and run exclusively of the others. Thus, we can remove the previous VI from memory and send only the data array and header to the next application. This allows memory to be conserved.



### Figure 10 Reference Node Method for Calling SubVI's

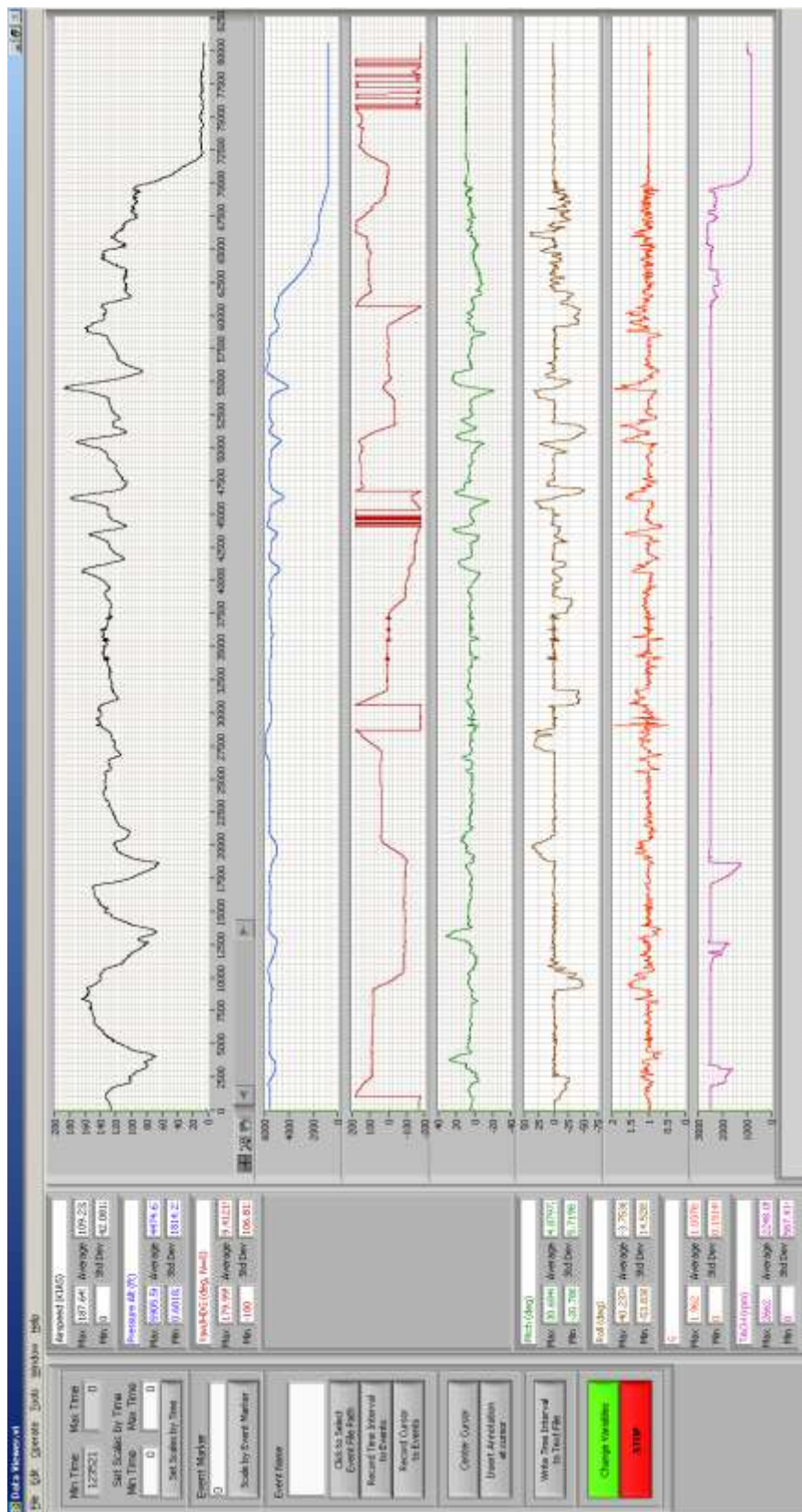
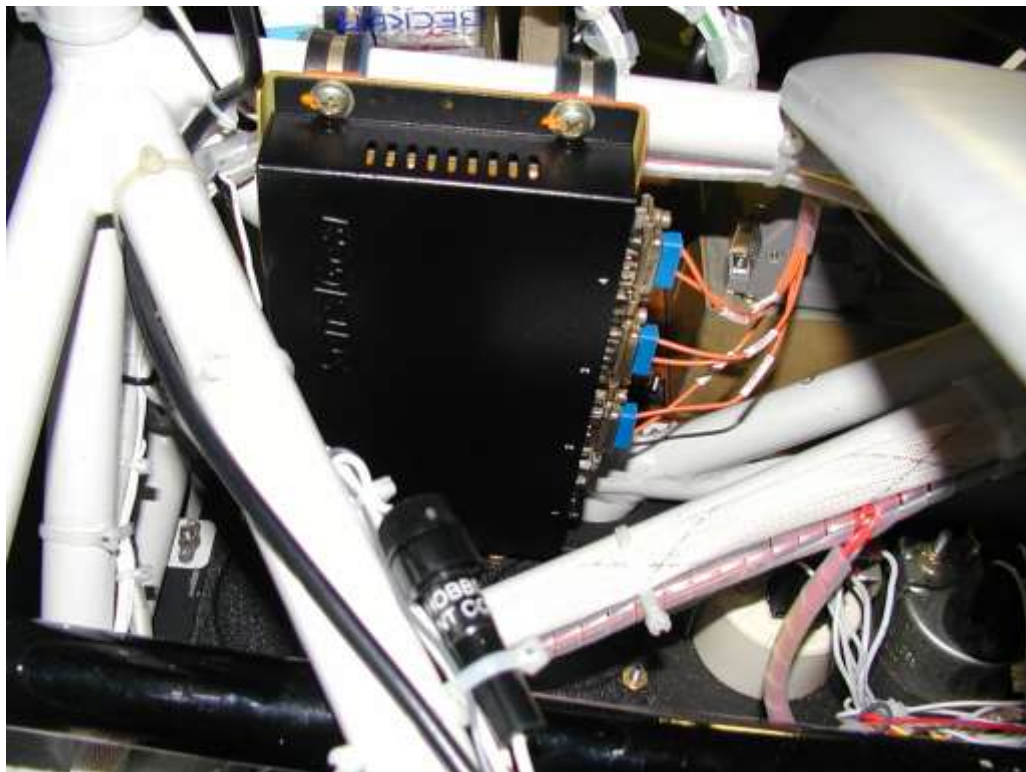


Figure 11 Data Viewer



**Figure 12 Installation of RS-232 to USB Optically Isolated Hub**

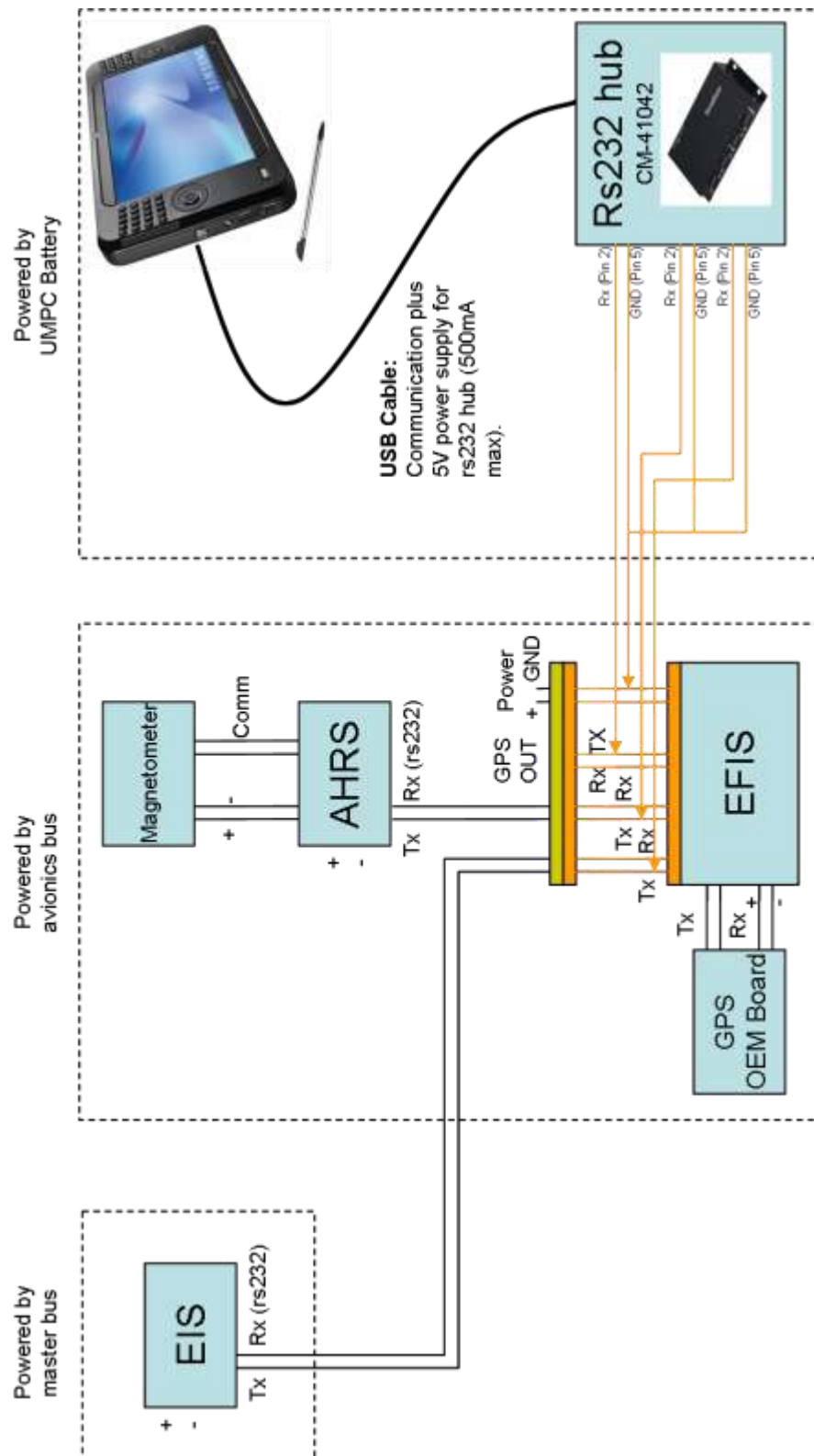
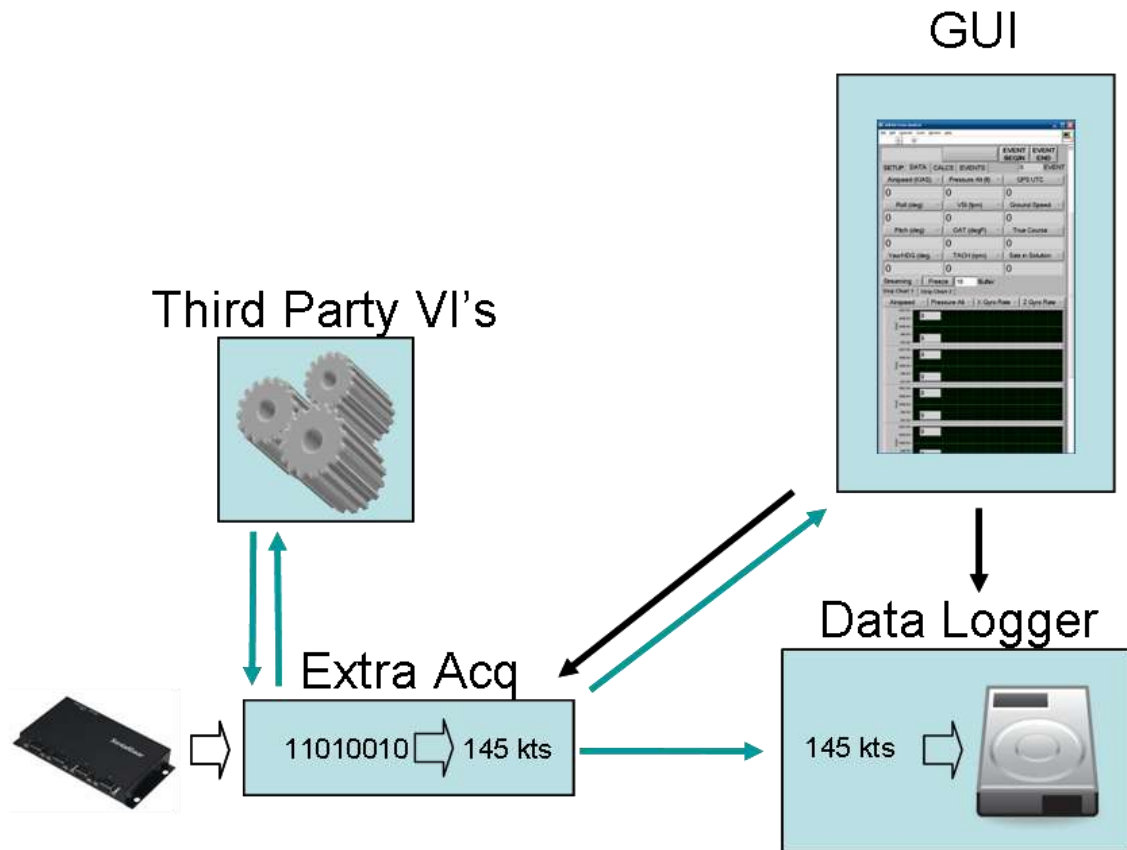


Figure 13 Aircraft System Interfaces





**Figure 14 Program Architecture**

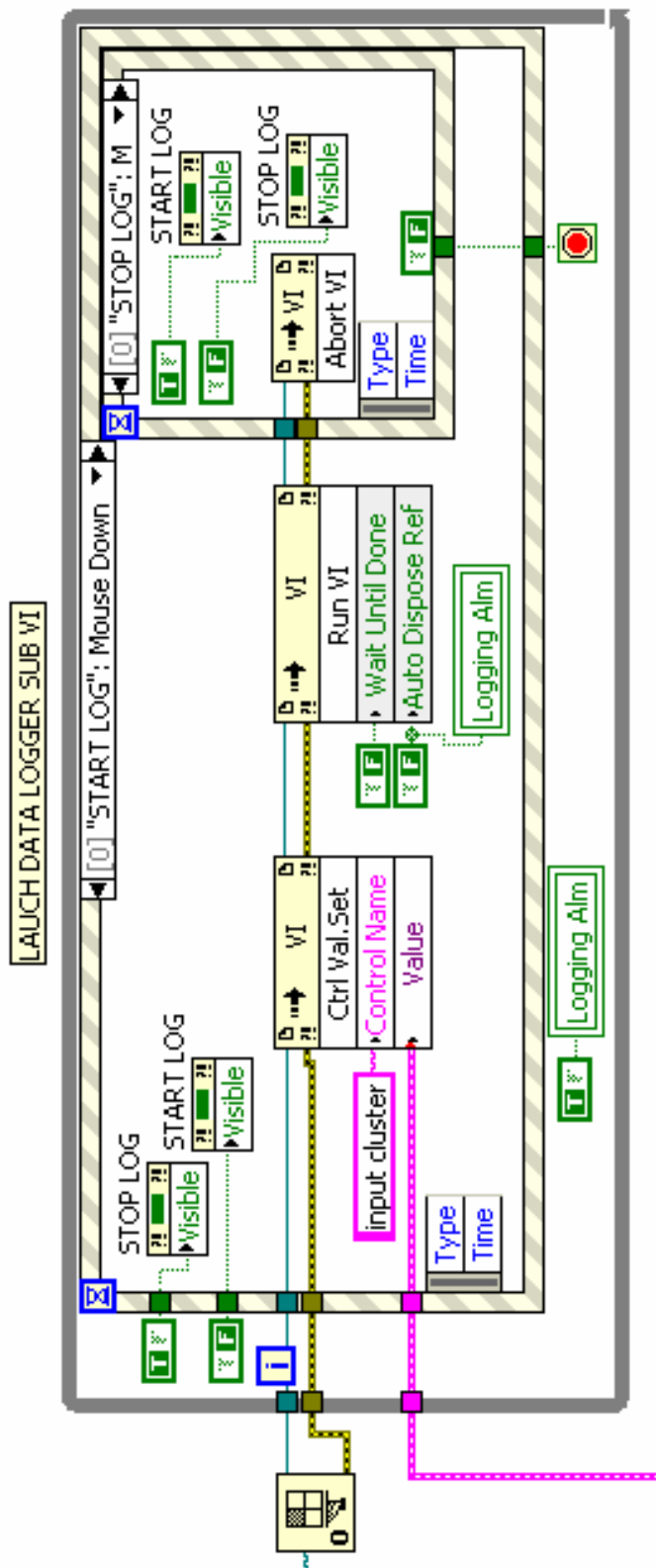


Figure 15 Invoke node Method for Calling SubVI's

11:06:45 PM

EVENT BEGIN

EVENT END

SETUP

DATA

EVENTS

0

EVENT

STOP VI

KILL VI

STOP AQ

Choice of Spread Calc

standard deviation

STOP LOGGING

AHRS Aquisition Fail

AHRS BUFFER

GPS Aquisition Fail

GPS Lock Lost

EIS Aquisition Fail

EIS BUFFER

0 AHRS Packets Lost

0 EIS Packets Lost

LOGGING

Engine Time	23.5
Fuel Quantity (gal)	18.5
Flight Timer (Hrs)	0
Flight Timer (Min)	28
Flight Timer (Sec)	19
Fuel Reserve (Min)	54
Baroset (Hg)	0
Event Counter	0

Figure 16 Graphic User Interface Setup Tab

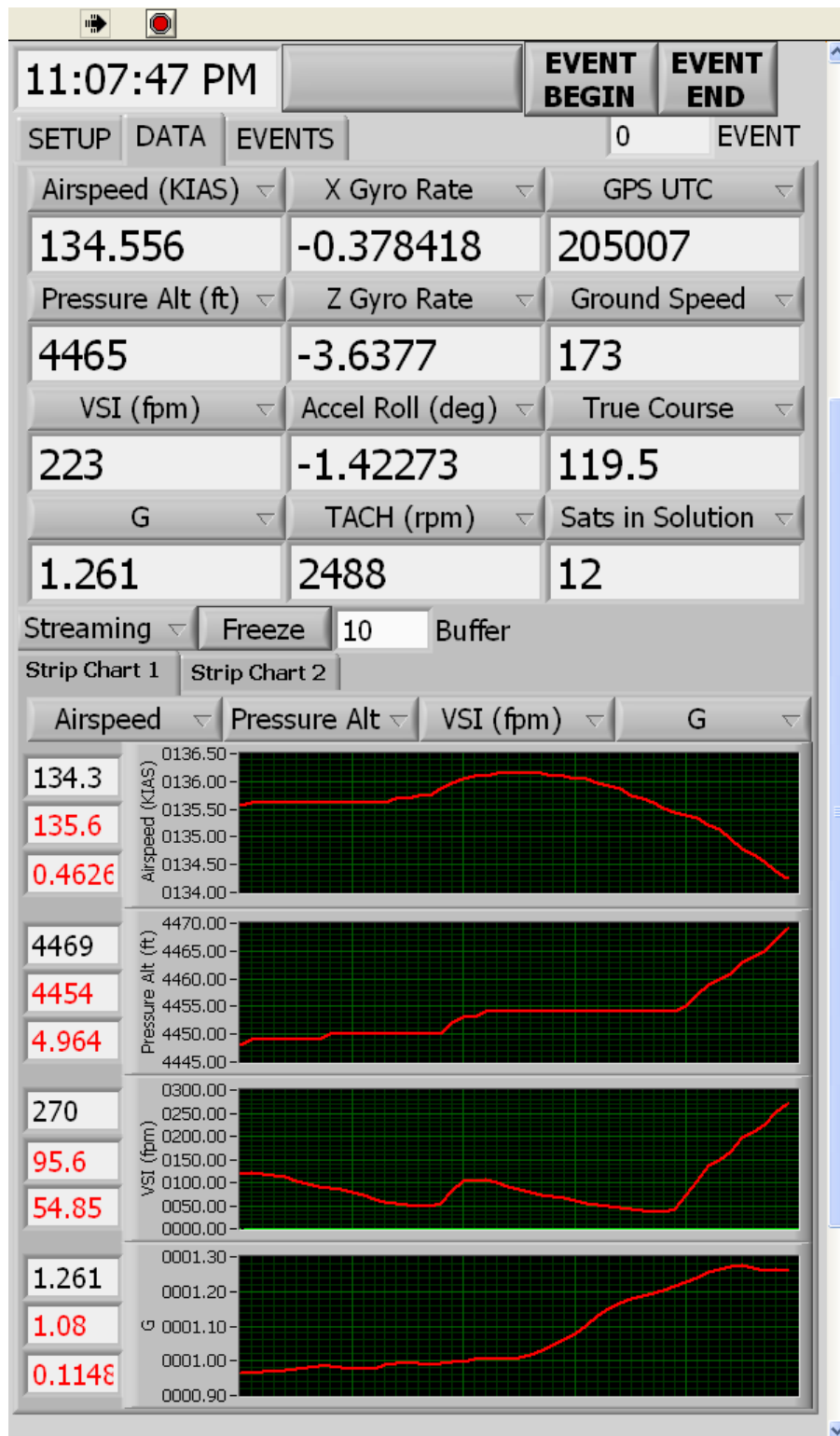


Figure 17 Graphic User Interface Data Tab

11:08:07 PM

EVENT BEGIN

EVENT END

SETUP

DATA

EVENTS

0

EVENT

1. Groundblock

-

Description

Perform normal start up as per checklist.

BACK

COMPLETE

REPEAT

NEXT

Event Title

Run up

Load Picture

Save / Clear

User Defined Events

**Figure 18 Graphic User Interface Events Tab**

58206.531 3. Start Acq Complete.  
 58211.125 6. Begin EFIS Logging Complete.  
 58369.359 7. Level Accel  
 58369.359 3/20/2009 4:12:49 PM Event 1 begin Level Accel  
 58381.281 3/20/2009 4:13:01 PM Event 1 end Level Accel  
 58496.609 3/20/2009 4:14:56 PM Event 2 begin Level Accel  
 58565.297 3/20/2009 4:16:05 PM Event 2 end Level Accel  
 58633.453 7. Level Accel Complete.  
 58768.312 9. Long Period  
 58768.312 3/20/2009 4:19:28 PM Event 4 begin Phugoid Controls Fixed  
 58789.203 3/20/2009 4:19:49 PM Event 4 end Phugoid Controls Fixed  
 58795.469 3/20/2009 4:19:55 PM Event 5 begin Phugoid Controls Free  
 58808.078 3/20/2009 4:20:08 PM Event 5 end Phugoid Controls Free  
 58824.359 9. Long Period Complete.  
 58837.312 10. Short Period  
 58837.312 3/20/2009 4:20:37 PM Event 6 begin Short Period Fixed  
 58868.094 3/20/2009 4:21:08 PM Event 6 end Short Period Fixed  
 58881.375 10. Short Period Complete.  
 58948.594 11. Dutch Roll  
 58948.594 3/20/2009 4:22:28 PM Event 7 begin Dutch Roll Fixed  
 58985.781 3/20/2009 4:23:05 PM Event 7 end Dutch Roll Fixed  
 58988.094 3/20/2009 4:23:08 PM Event 8 begin Dutch Roll Fixed  
 59023.094 3/20/2009 4:23:43 PM Event 9 end Dutch Roll Fixed  
 59067.266 11. Dutch Roll Complete.  
 59516.984 9. Long Period (REPEAT)  
 59516.984 3/20/2009 4:31:56 PM Event 15 begin Phugoid Controls Fixed  
 59536.766 3/20/2009 4:32:16 PM Event 15 end Phugoid Controls Fixed  
 59622.656 9. Long Period (REPEAT) Complete.  
 59653.078 19. Return to land

**Figure 19 Events File Sample**

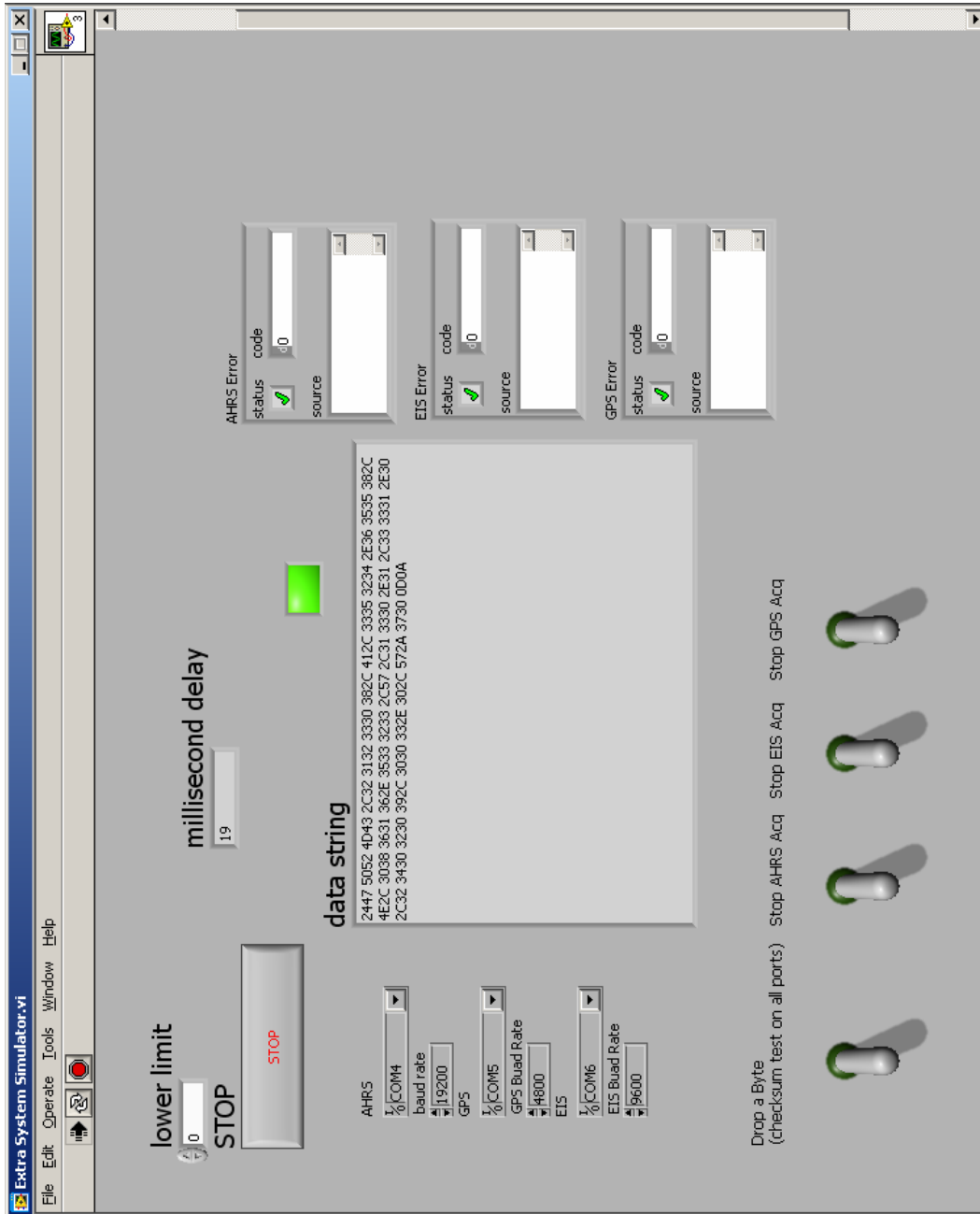


Figure 20 Extra System Simulator (GRT Device Simulator)



**Figure 21 Hardware in the Loop Software Testing**



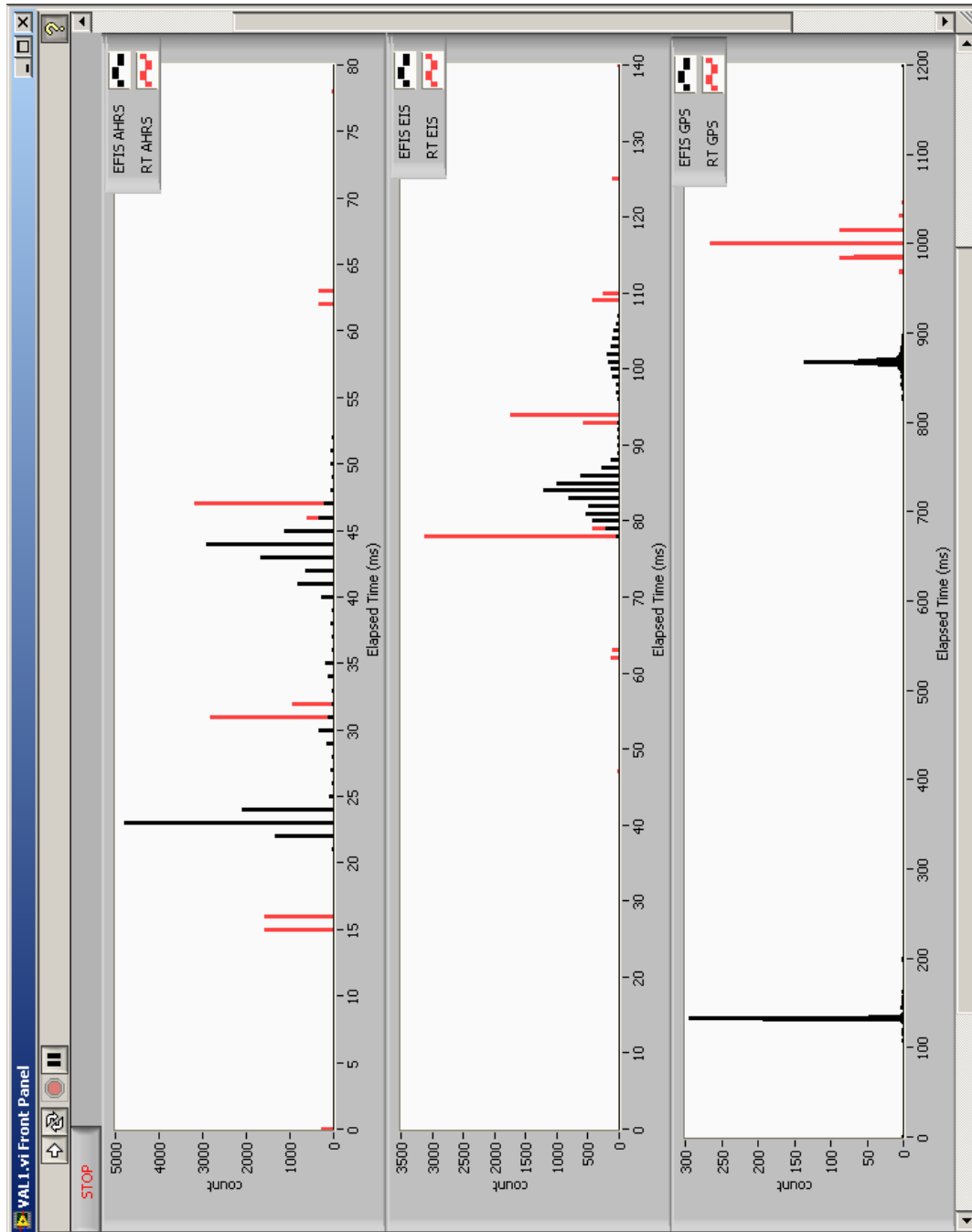
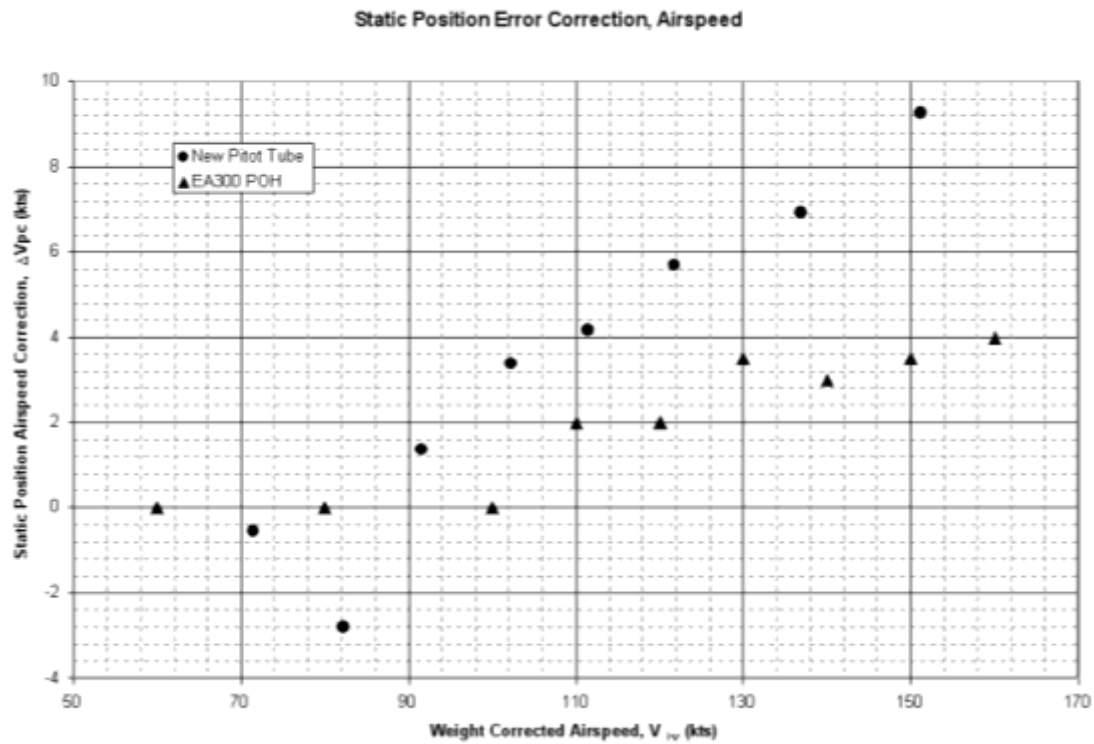


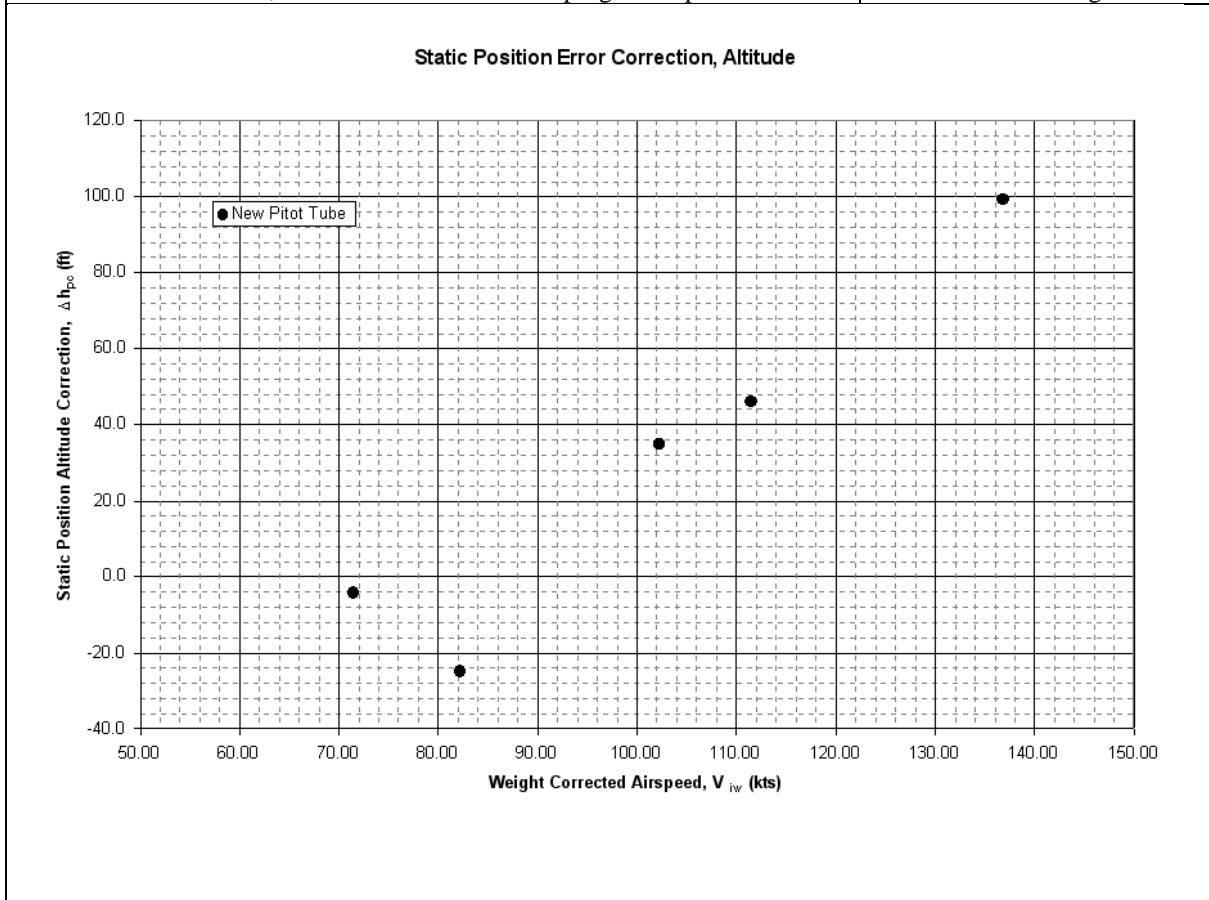
Figure 22 Histogram Comparison of Elapsed Times

Test A/C: Extra 714X	RPM: Various	Test Weight: Various	Reduced to Std Day
Test Date: Feb-09	Manifold Pressure: Various	Test CG: Various	Conditions
Method: GPS Wind-box, data from GRT Real-time program is presented.			Crew: Corda / Ludwig



**Figure 23 Static Position Error Correction, Airspeed**

Test A/C: Extra 714X	RPM: Various	Test Weight: Various	Reduced to Std Day
Test Date: Feb-09	Manifold Pressure: Various	Test CG: Various	Conditions
Method: GPS Wind-box, data from GRT Real-time program is presented.			Crew: Corda / Ludwig



**Figure 24 Static Position Error Correction, Altitude**

Test A/C: Extra 714X	RPM: 2513 indicated	Test Weight: 2035 lb	Reduced to Std Day
Test Date: 20-Mar-09	Manifold Pressure: N/A	Test CG: 33.9"	Conditions
Method: Rudder Doublet, data from GRT Real-time program is presented.			Crew: Corda / Ludwig

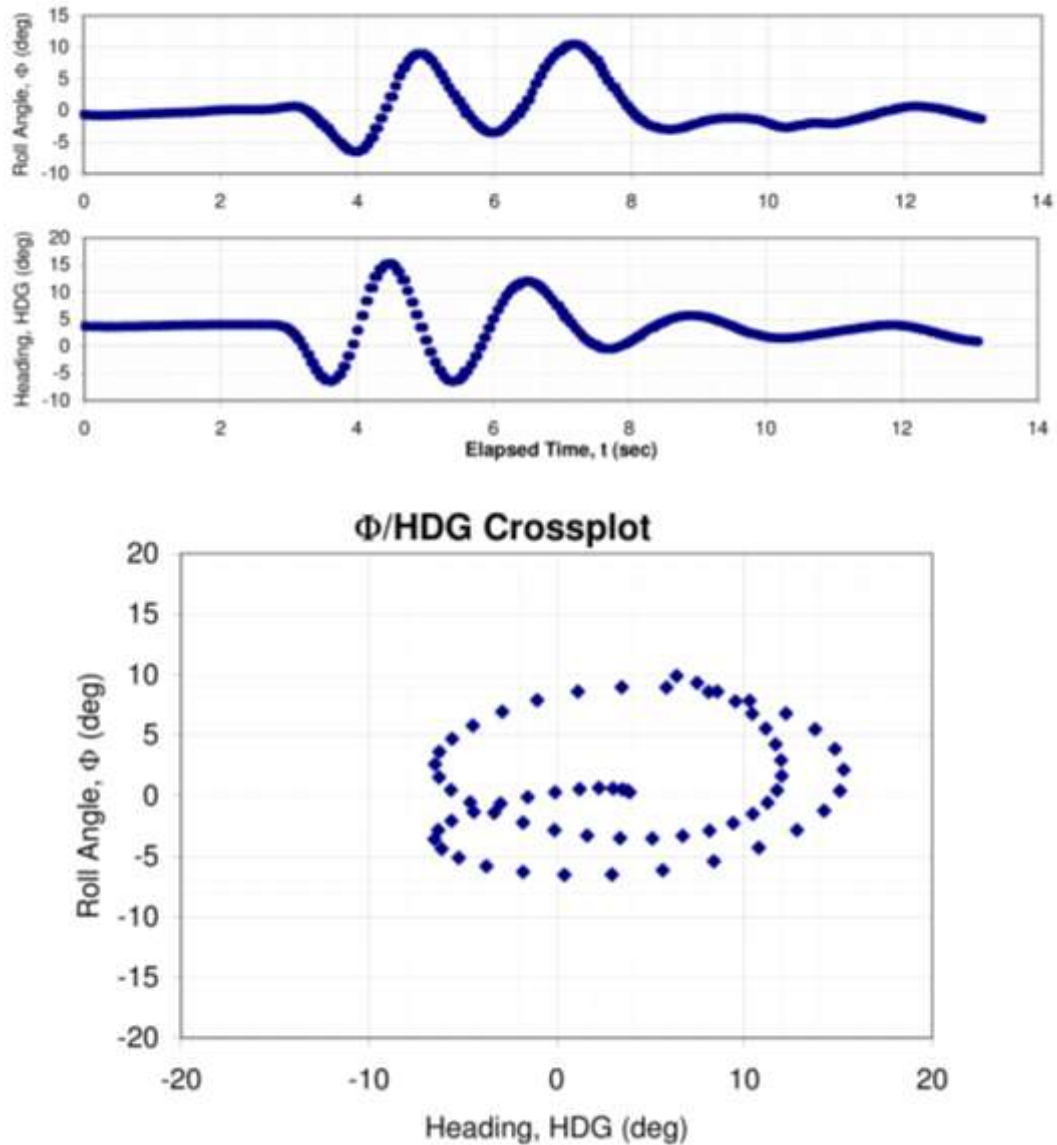


Figure 25 Dutch Roll Data

Test A/C: Extra 714X	RPM: 2519 indicated	Test Weight: 2026 lb	Reduced to Std Day
Test Date: 20-Mar-08	Manifold Pressure: N/A	Test CG: 33.9"	Conditions
Method: Spiral Mode, controls Fixed and Controls Free, data from GRT Real-time program is presented.			Crew: Corda / Ludwig

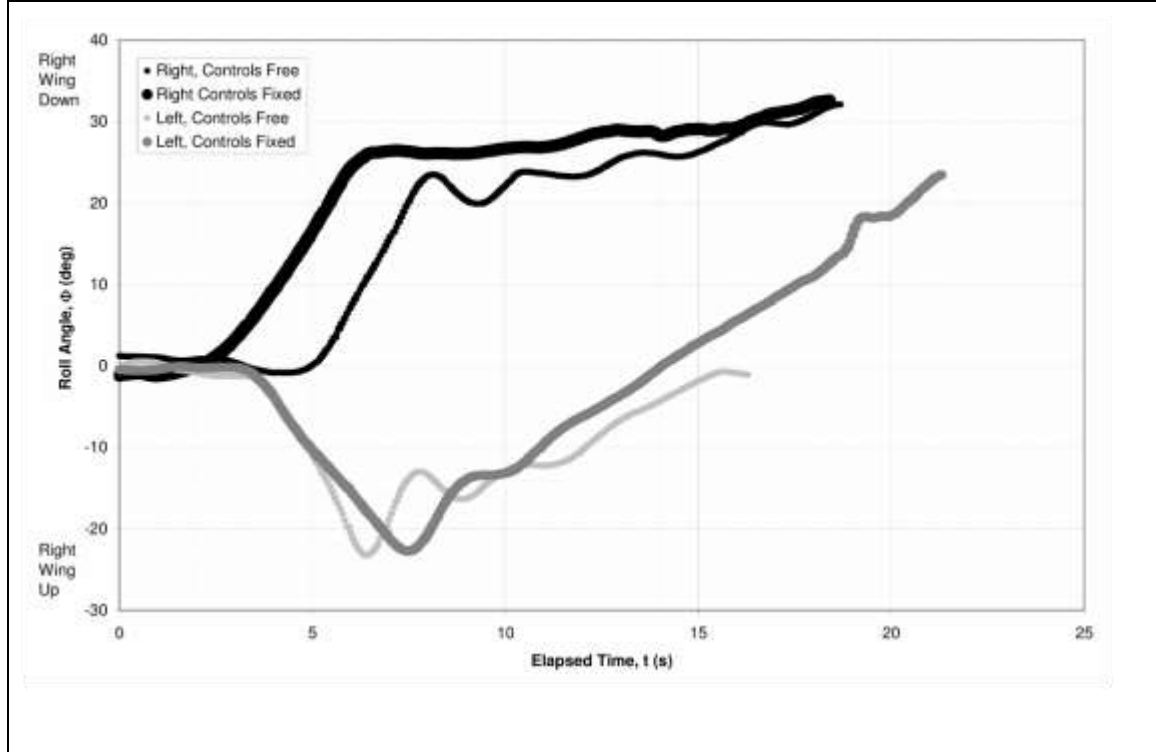


Figure 26 Spiral Data

Test A/C: Extra 714X	RPM: 1280 indicated	Test Weight: 1926 lb	Reduced to Std Day Conditions
Test Date: 19-Nov-08	Manifold Pressure: Idle	Test CG: 33.9"	
Method: Approach to landing Stall, data from EFIS Demo Log is presented.			Crew: Corda / Ludwig

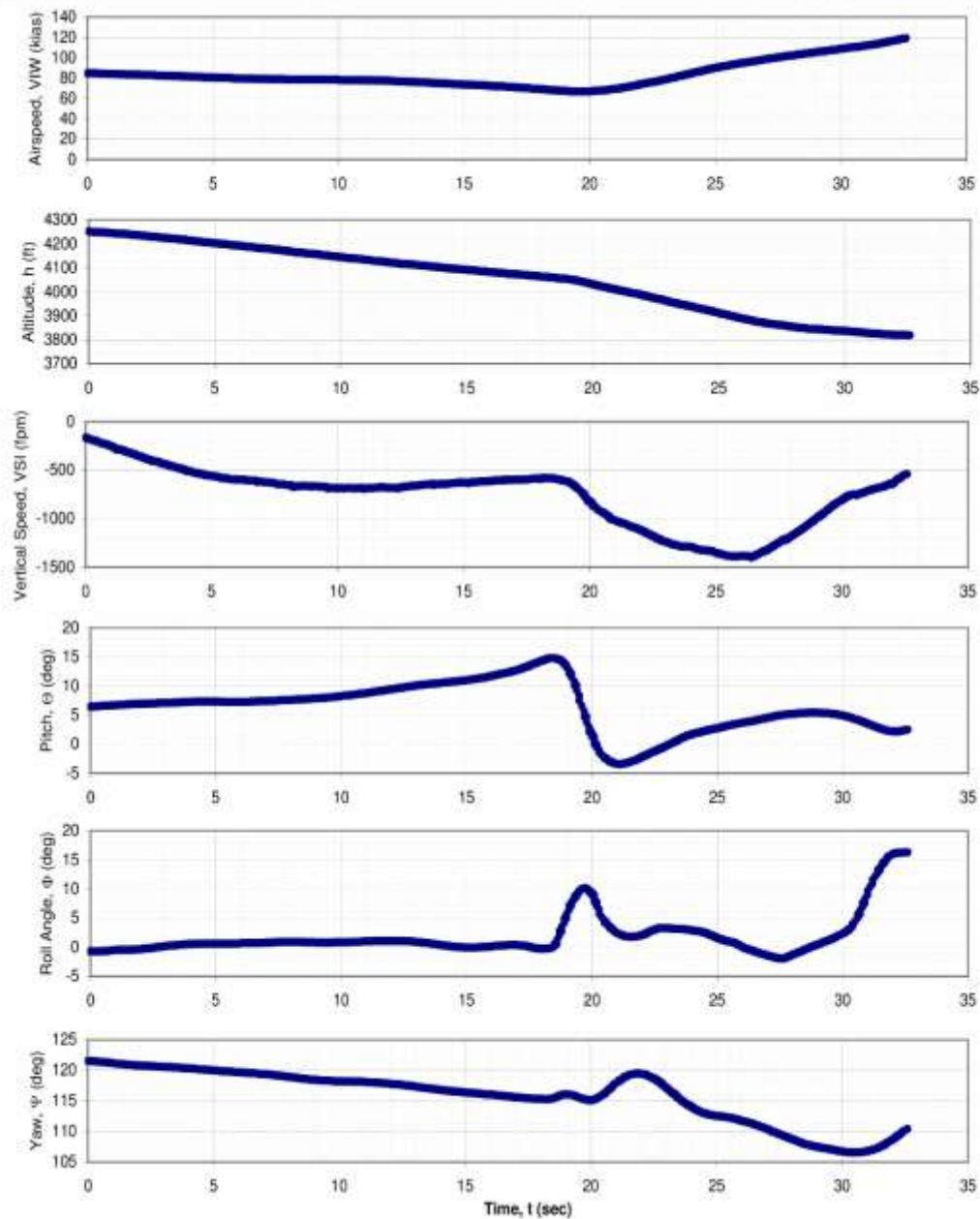


Figure 27 Stall Data

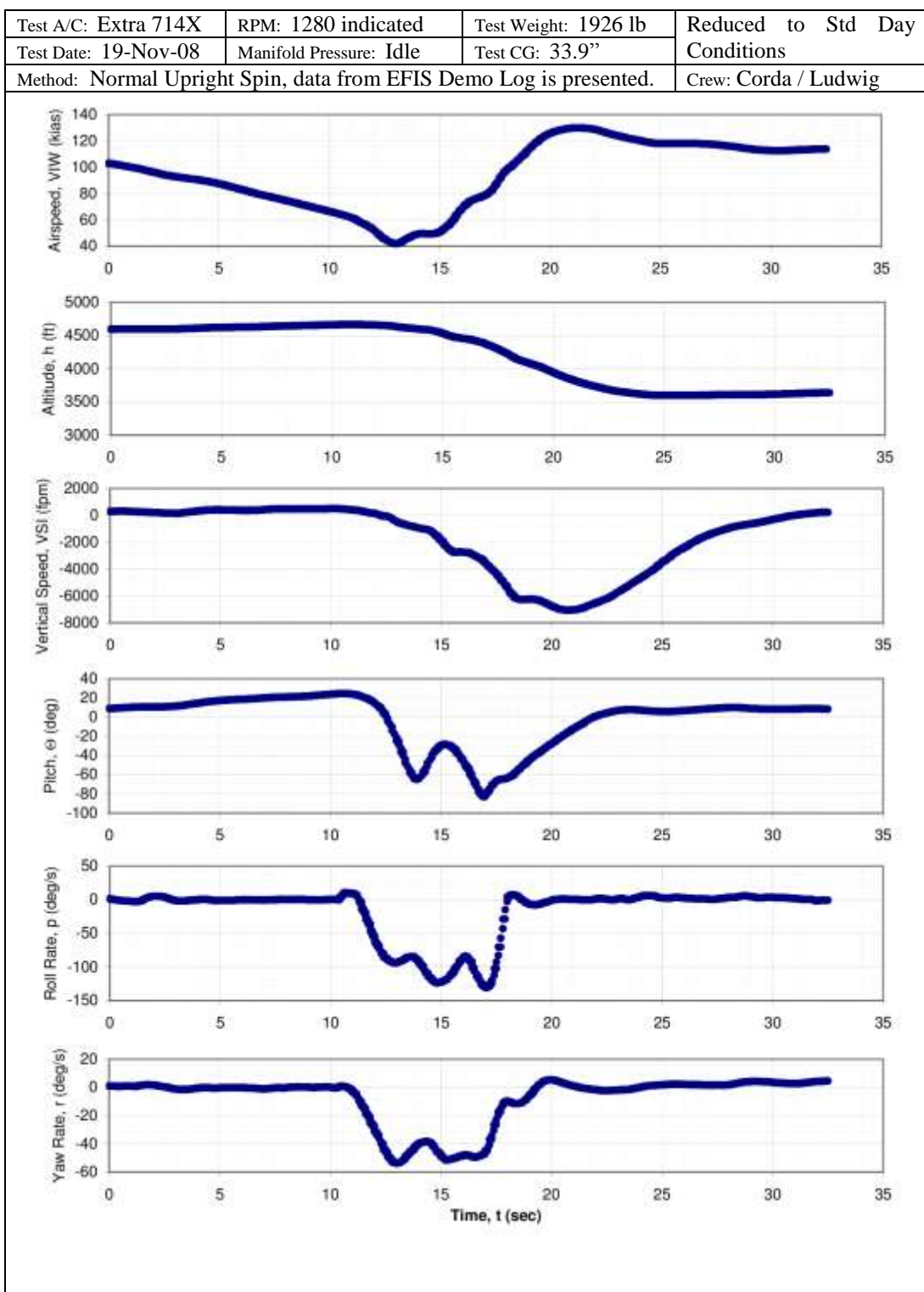


Figure 28 Spin Data

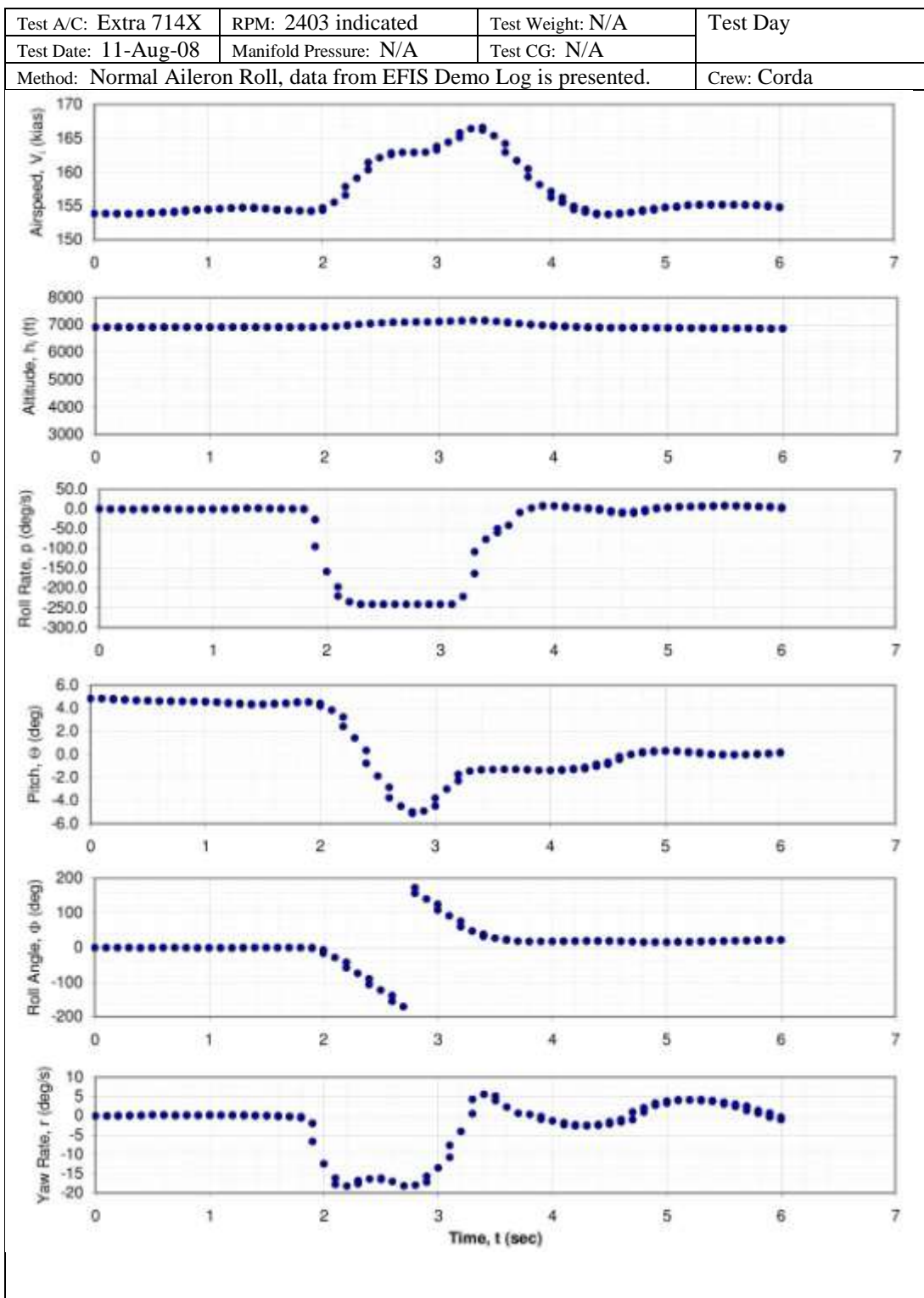


Figure 29 Maximum Rate Aileron Roll Data



## Appendix 2

The GRT Demo file format has been reproduced here with permission of Grand Rapids Technology. The file has no header consists of a byte by byte recoding with bytes to identify the source, timestamp, and length of following data.

**Table 3 Demo Log File Format**

Size (Bytes)	Field
1	Source
4	Timestamp
4	Length of data, XX bytes
XX	Data

The data source byte is an unsigned 8 bit integer as described in Table 4. The timestamp is an unsigned 8 bit integer with the most significant bit first. The timestamp is the time since boot of the EFIS in milliseconds. Length is also a 32 bit unsigned integer and specifies the number of bytes of data to follow. The first byte after the last byte of data will be another source byte for the next data packet.

**Table 4 Demo Log Data Sources**

Data type	Source
1	AHRS
2	EIS
3	GPS1
4	Analog EFIS Inputs
5	SL30-1
6	SL30-2
7	Settings (Set Menu)
8	State (button choices)
9	ARINC
10	Weather
11	GPS2
12	SL40

The GPS data format is ASCII text, transmitted at 4800 baud and contains the following NMEA formatted data sentences: GPRMC, GPGGA, GPGSA, GPGSV, PGRME, PGRMB, and PGRMM. The AHRS Data format is explained in the attached GRT AHRS Interface Control Document attached in Appendix 3. The EIS data format is presented in the following paragraphs.

**EIS Data Format:**

The EIS data is transmitted at 9600 baud, no parity, one stop bit, and 10 data sentences per second. Each data sentence begins with three sync bytes, and is ended with a checksum byte. The parameters are encoded in the data sentence as either an 8-bit or a 16-bit unsigned integer. The 8-bit integers are transmitted as one byte, whereas the 16-bit integers are split and transmitted as two bytes, a high byte and a low byte. The 16-bit integer can be rejoined by multiplying the value of the high byte by 256 and adding it to the low byte. The transmitted integers are then multiplied by a scale factor and added to a bias as required to calculate the final physical value. Some values only have a scale factor, some only have a bias, some have neither and are simply an integer, engine RPM is an example of this.

**Table 5 Model 4000/6000 EIS Data Format**

Byte	Data	Scale Factor	Bias	Unit	Data Description
0	Hex FE				1 <sup>st</sup> header byte
1	Hex FF				2 <sup>nd</sup> header byte
2	Hex FE				3 <sup>rd</sup> header byte
3	Tach High byte	1	0	RPM	Resolution of 1 RPM
4	Tach Low byte				
5	CHT1 High byte	1	0	DegF	Resolution of 1 Deg
6	CHT1 Low byte				
7	CHT2 High byte	1	0	DegF	Resolution of 1 Deg
8	CHT2 Low byte				
9	CHT3 High byte	1	0	DegF	Resolution of 1 Deg
10	CHT3 Low byte				
11	CHT4 High byte	1	0	DegF	Resolution of 1 Deg
12	CHT4 Low byte				
13	CHT5 High byte	1	0	DegF	Resolution of 1 Deg
14	CHT5 Low byte				
15	CHT6 High byte	1	0	DegF	Resolution of 1 Deg
16	CHT6 Low byte				
17	EGT1 High byte	1	0	DegF	Resolution of 1 Deg
18	EGT1 Low byte				
19	EGT2 High byte	1	0	DegF	Resolution of 1 Deg
20	EGT2 Low byte				
21	EGT3 High byte	1	0	DegF	Resolution of 1 Deg
22	EGT3 Low byte				
23	EGT4 High byte	1	0	DegF	Resolution of 1 Deg
24	EGT4 Low byte				
25	EGT5 High byte	1	0	DegF	Resolution of 1 Deg
26	EGT5 Low byte				
27	EGT6 High byte	1	0	DegF	Resolution of 1 Deg
28	EGT6 Low byte				
29	AUX5 High byte				
30	AUX5 Low byte				

Table 5 Continued

31	AUX6 High byte				
32	AUX6 Low byte				
33	ASPD High byte				Airspeed, units same as in EIS (unused on Extra 300)
34	ASPD Low byte				
35	ALT High byte	1	0	ft	Altitude in feet, not used on Extra
36	ALT Low byte				
37	VOLT High byte	0.1	0	V	Resolution of 0.1 Volts
38	VOLT Low byte				
39	FUELF High byte	0.1	0	Gal/hr	Resolution of 0.1 gal/hr
40	FUELF Low byte				
41	UNIT	1	0		Internal Unit Temp
42	CARB	1	0		2's compliment
43	ROCSGN	100	0	fpm	Vertical speed, 100 fpm resolution, 2's compliment (unused on Extra)
44	OAT	1	-50	degF	2's compliment, subtract 50.
45	OILT High byte	1	0	DegF	Oil Temperature
46	OILT Low byte				
47	OILP	1	0	psi	
48	AUX1 High byte				
49	AUX1 Low byte				
50	AUX2 High byte				
51	AUX2 Low byte				
52	AUX3 High byte				
53	AUX3 Low byte				
54	AUX4 High byte				
55	AUX4 Low byte				
56	COOL High byte	1	0	degF	Coolant Temperature
57	COOL Low byte				
58	ETI High byte	0.1	0	hrs	Hour meter, 0.1 hrs resolution
59	ETI Low byte				
60	QTY High byte	0.1	0	gal	Fuel Qty, Resolution = 0.1 gal
61	QTY Low byte				
62	HRS	1	0	Hrs	Flight timer in hours
63	MIN	1	0	Min	Flight timer in minutes
64	SEC	1	0	Sec	Flight timer in seconds
65	ENDHRS	1	0	Hrs	Fuel flow time until empty, hours
66	ENDMIN	1	0	Min	Fuel flow time until empty, minutes
67	BARO High byte	0.1	0	HG	Baroset in HG, 0.01 resolution (Baroset unused in Extra)
68	BARO Low byte				
69	TACH2 High byte	1	0	RPM	Second Engine Tach – 1 RPM resolution
70	TACH2 Low byte				
71	SPARE				Unused
72	Checksum	Unsigned 8-bit integer, sum of all bytes except first 3 header bytes and result complimented.			

## **Appendix 3**

The following document was provided by Grand Rapids Technology for inclusion into this thesis. The document details the data formats of the GRT AHRS.

## **Vita**

Christopher G Ludwig was born in Dubuque, Iowa. He studied Aerospace Engineering from Iowa State University and was awarded a Bachelor's Degree in May, 2001. He then worked in industry as a Test Engineer for six years before returning to school for an advanced degree.