DEVELOPMENT, IMPLEMENTATION AND FLIGHT TESTING OF PERIPHERAL VISION DISPLAYS FOR GENERAL AVIATION

A thesis presented to

the faculty of the

Fritz J. and Dolores H. Russ

College of Engineering and Technology of

Ohio University

In partial fulfillment

of the requirements for the degree

Master of Science

Jahnavi Chakrabarty

March 2005

This thesis entitled

DEVELOPMENT, IMPLEMENTATION AND FLIGHT TESTING OF PERIPHERAL VISION DISPLAYS FOR GENERAL AVIATION

BY

JAHNAVI CHAKRABARTY

has been approved

for the School of Electrical Engineering and Computer Science

and the Russ College of Engineering and Technology by

Michael S. Braasch
Thomas Professor of Engineering, EECS

Dennis Irwin,
Dean, Fritz J. and Dolores H. Russ
College of Engineering and Technology

CHAKRABARTY, JAHNAVI. M.S. MARCH 2005. Electrical Engineering

DEVELOPMENT, IMPLEMENTATION AND FLIGHT TESTING OF PERIPHERAL VISION DISPLAYS FOR GENERAL AVIATION (135pp)

Director of Thesis: Michael S. Braasch

This thesis is an investigation into peripheral vision displays for General Aviation. It documents the development and testing of two peripheral vision displays. The first is a right side display, used in conjunction with a forward display. Both the displays depict a synthesized view of the outside world. The right side display is to be used as a secondary instrument for indication of the aircraft attitude. The second type of display is a strip of LEDs placed in the pilots' peripheral field of vision. The focus of this thesis is the evaluation of this display system. The LED strips provide the pilot with an indication of the aircraft roll, informing him of any sudden changes in the airplane attitude. Flight testing of the system has shown that with a few modifications, this system can be used as a secondary instrument for attitude awareness and mitigate the effects of spatial disorientation.

Approved:

Michael S. Braasch

Thomas Professor of Engineering, EECS

Dedicated to Mamma, Rajat, Hrushi, Chintu and Sonu…………

ACKNOWLEDGEMENTS

The research presented in this thesis was funded by the NASA/FAA Joint University Program for Air Transportation Research. Mike Paglione is thanked for his contributions to the program and his helpful suggestions throughout the course of this research.

Sincere thanks to Jay Clark and Paul Nilles from the Ohio University Avionics Engineering Center for the long hours spent installing and testing the equipment. Special thanks to Jay for his invaluable support and dedication that made the flight testing possible. My heartfelt gratitude to Kadi Merbough for his technical support and encouragement through out the development phase of this research. His patience as well as his superior technical skills has solved many an impossible problem. A special thanks to Don Venable for his development of the microcontroller program.

I would also like to thank Dr. Hansman from MIT, Boston for his invaluable suggestion that allowed us to take this research in a new direction.

A very special thank you to Dr. James Rankin and Jamie Edwards for their patience and understanding. This research would not have progressed without their invaluable suggestions and unending support. I would also like to thank Mike Mowry and Eric Hutchison for the time and effort spent in testing and evaluating the system. A very sincere thanks to Bryan Branham who has been a part of the research from the time of inspection. His inputs and suggestions made it possible for this research to develop in the right direction.

A very warm and heartfelt thanks to the staff of the Avionics Engineering Center, especially Becky Tippie, Sharon Conner, Tracy Crabtree and Cathy Romanowski. Thank you Becky and Sharon for making the AEC feel like a home away from home and thank you Cathy and Tracy for caring for me when I was going through the toughest time of my life.

I would also like to thank my committee members for their continuous support and encouragement, every step of the way. Thank you Dr. Maarten, for your enthusiasm and constant support. I would also like to thank Dr. Diggle for making me smile at every opportunity possible. Your jest for life has me in awe.

I would like to specially thank Douglas Burch. Coming from a foreign land and starting new can be a scary experience. You accepted me as a friend and colleague and stood by me every step of the way. Your friendship, constant support and help at every turn in my life gave me the strength to believe in myself. Without your encouragement, I would not have come so far and for that I will always be in your debt.

A grateful thanks to Sai Kalyanaraman. You have taught me humility and excellence in the same breath.

To Dr. Braasch, I extend my lifelong gratitude and thanks. I feel truly honored and blessed to have met you and worked with you. The opportunities that you have provided me with have made my life fuller and more enriched that I thought possible. I applaud you for your dedication to your students and family. I will try to live my life by the example that you have set. For the courage and support that you offered to me when I

was at the lowest point in my life, I am eternally in your debt. In Indian scriptures, a teacher is akin to God and I offer you my sincere obeisance.

To my family, Mamma, Rajat and Hrushi and Sonu. Thank you Mamma, my best friend, for always being there. You are my guiding spirit. Rajat, it is your support and blessing that has got me so far. Sonu, I would have been lost without you. Thank you for taking care of me and making those terrible times bearable.

Finally I would like to thank the Lord Almighty and Chintu. You are both one and the same to me. Thank you for everything, every step of the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

General Aviation (GA), in its early inception was considered primarily for recreational flying. Over the years it has evolved into a very important segment of aviation providing personal, business as well as freight transportation. The services provided by the GA span over various fields such as law enforcement, forest-fire fighting, air ambulance, logging, fish and wildlife spotting, crop protection as well as other vital services [1].

GA proves to be cheaper and faster than flying commercial in many cases, making it very lucrative as well as attractive to the average traveler. Just like people made the transition from trains and buses to personal cars, many people choose to fly their own private planes to make short term travel needs faster and hassle free. Airplanes travel two to four times faster than cars; this is one of the primary reasons we fly [2].

GA provides us with a more convenient mode of transportation, but it also requires expertise and training on behalf of the pilot to make the flight safe and secure. Should an impact occur, crash physics work against us. Double the speed, quadruple the impact; triple the speed and the impact force goes up nine times. Add the vertical component introduced by altitude and one has to marvel that aircraft protect their occupants as well as they do. Motor vehicles have about ten times as many accidents per mile as do general aviation aircraft, but the aircraft have about seven times as many fatal accidents per mile as the motor vehicles. The 1999 Nall Report states “On average, only one aircraft accident in five results in fatalities”. Furthermore, the fleet flies almost 10 million miles before there is a fatal loss [2].

1.1 General Aviation vs. Commercial Airlines

Though GA is the choice for an increasing number of operations, little has changed in the GA cockpit in terms of safety enhancing features. Single pilot operation General Aviation (GA) accident rates are much higher than commercial airlines. Some major differences between GA and commercial flight, which also accentuate this fact, are as follows:

i. As opposed to commercial operations, GA pilots conduct a wider range of operations with reduced regulation and minimal infrastructure.

ii. There is a large variation in the training levels of GA pilots, with their qualifications ranging from single pilot to ATP (Airline Transport Professional).

iii. GA aircraft have fewer cockpit resources with the aircraft operations being predominantly single pilot.

iv. GA aircraft fly to more than 15,000 landing facilities, many of which lack infrastructural facilities such as precision landing equipment, long runways and good lighting as installed at commercial airports.

v. There are more take offs and landings per hour – the highest risk phases of any flight.

vi. The GA pilot is primarily responsible for safety of the flight. There is no dispatcher or meteorologist to share the responsibility.

vii. The aircraft is most likely not equipped to handle adverse weather conditions and is forced to fly through bad weather instead of over it.

viii. Higher incidence of spatial disorientation due to lack of experience.

1.2 Spatial Disorientation

The senses we use to maintain our balance and perceive our orientation to the world around us depend primarily on our visual cues. Without vision providing a primary frame of reference to the outside world, pilots become increasingly susceptible to spatial disorientation.

Spatial Disorientation as defined by the Air Force Manual (AFM 11-217) "is an incorrect perception of one's linear and angular position and motion relative to the plane of the earth's surface. It is an erroneous percept of any of the parameters displayed by aircraft control and performance flight instruments [3]."

There are three primary sensory sources of spatial orientation:

i. *Visual System:* relates the orientation of the body to an externally fixed frame of reference. When visibility is good, vision dominates spatial orientation.

ii. *Vestibular System:* Organs found in the inner ear (the otoliths), which assumes that the gravitational force is vertical, often being incorrect in a flight situation.

iii. Somatosensory System: Nerves in the skin, muscles and joints(propioceptors) which along with the hearing, sense position based on gravity, feeling and sound[4].

Visual dominance exists when the pilot receives through the eyes all the information used to maintain correct orientation. (AFM 11-217). When visibility is bad due to conditions such as fog, clouds, haze, darkness or sky backgrounds with indistinct contrast, the otoliths and proprioceptors take over spatial orientation which give rise to numerous illusions causing fatal misjudgements in flight [3,4].

A detailed analysis of accidents by the Nall commission over a ten-year period (1987-1996) with an emphasis on spatial disorientation recognized it as a very significant contributory factor to fatal accidents. During this period, there was an average of almost 37.6 accidents per year, of which 33.9% were fatal. At this rate, there is one fatal spatial disorientation accident every eleven days. Over 90 percent of all the accidents during this time in which spatial disorientation was a factor resulted in fatalities.

Accidents resulting from spatial disorientation are mostly suffered by non instrument-rated pilots attempting to complete visual meteorological condition (VMC) flights in instrument meteorological conditions (IMC) [2]. Though the percentage of such accidents has reduced to 1.6% in 2003, 89.4% of them were fatal [1]. In 2002, the Nall Report stated that only 49% of the GA pilots had an IFR Rating making an attempted VFR into IMC, a risky flight operation [5].

Due to the high likelihood of GA aircraft experiencing IFR conditions during flight, there is a need for a visual guidance system that could provide a synthesized view of the outside world. This is especially useful during the take-off and landing phase of the flight when accidents are most likely to occur. During take off and landing as well as during a bank, the pilot finds himself intuitively looking to his side for visual guidance and peripheral orientation.

The goal of this research is to focus on side displays that enhance the peripheral vision of the pilot. Two types of displays are evaluated. The first is a synthetic display that provides the orientation of the aircraft with respect to his surrounding terrain on a screen that is placed in his side view. The second is a strip of LEDs, placed in his peripheral field of vision, which light up according to the bank of the aircraft, indicating the aircrafts' attitude to the pilot.

Both the systems have been flight tested to evaluate the extent of orientation information that is provided to the pilot. Extensive flight testing and pilot evaluations of the LED system have been performed. This research tries to validate the usefulness of the LED system in providing peripheral guidance to the GA pilot.

2. Vision and its modes – Importance of peripheral information

Vision is without a doubt the most important sensory function on which flight is based. Without it, flight as we know it today would be impossible. The two most important functions of vision while flying an airplane are object recognition and spatial orientation. Understanding of these two functions gives us an insight into the visual illusions that a pilot faces when spatially disoriented [6].

2.1 The Two Modes of Vision

The human ability to move and orient oneself in three-dimensional space is primarily controlled by the visual system. Evidence from studies has shown that vision must be considered as two separate processes. The two processes are controlled by two different systems referred to as the “focal system” and the “ambient system” [6].

2.1.1 Focal Vision

This mode of vision is used in answering the question of “what”. It tells us about the form of the object, the pattern it contains. It is concentrated in the central visual fields and is used for perceiving fine details of objects in our personal visual space. Focal vision uses approximately 30 degrees of the central visual field. Its contribution to orientation is derived from its perception of distance and depth. It is affected by the factors such as adequate luminance and presence of refractive error [6].

2.1.2 Ambient Vision

The ambient mode of vision answers the question of "where" with regards to where the observer is in space, and whether the observer or the surrounding environment is moving. It subserves spatial orientation and localization. It is mediated with large stimuli of the peripheral visual field in coarse detail [6].

Ambient vision is involved with orienting the individual in the environment and is largely independent of focal vision. It does not necessarily need stimulus energy or detailed optical quality to provide orientation. This can be demonstrated with the help of the following examples.

This short experiment was performed at a presentation at the 2004 EAA Airventure, Oshkosh, Wisconsin at the National Consortium for Aviation Mobility. Similar experiments help by the Air Force Research Laboratory corroborate these finds [7].

A volunteer was asked to walk the length of a platform while reading a book. He was asked to look directly into the book while walking. As he reached the end of the platform, the volunteer stopped a little before the edge. Next he was asked to wear a foggle - a mask that restricts the limits of your vision to the central field, blocking out the peripheral view. When asked to walk again while reading a book the volunteer could not judge the nearing of the edge of the platform [7].

This second experiment was put forth by Dr. Richard Malcolm when trying to develop the 'Malcolm Horizon' that will be discussed shortly hereafter.

If one stands with the heel of one foot resting against the toe of the other foot, and then closes one eye, one immediately notices that it is a fairly difficult job to maintain steady balance. If one now takes a tube of paper, rolled up like a toy telescope, and places this in front of the open eye so that all of the peripheral vision is blocked, then one finds that it is very difficult to maintain one’s balance. However, if the converse of this experiment is performed, and a clenched fist is brought up to the open eye so as to obscure all the central visual field, leaving only the peripheral vision functioning, then we are surprised to find that maintaining ones balance becomes fairly easy again [8].

As seen from the above two experiments, ambient vision is reflexive. There is a potential advantage in displaying orientation information in an aircraft as opposed to traditional displays which require training and interpretation. Studies also show that processes requiring higher levels of information processing are more vulnerable to loss during stressful conditions than reflexive actions [9].

2.2. Need to Study Peripheral Information

The disorientation that is felt by the pilot during high workload or stressful flying conditions needs to be reduced by additional visual cues that can safely guide the pilot. In lieu of this, the study of peripheral vision for providing visual cues was undertaken.

There are several arguments that support the fact that cues from the peripheral vision help in maintaining spatial orientation.

Peripheral vision is also referred to as the ambient mode of vision, and receives stimuli from the vestibular, somatosensory and the auditory senses along with the visual senses to maintain spatial orientation, posture and gaze stability [9].

As this kind of vision is used for balancing posture, it is well suited for correct processing of the orientation information provided. The need to read and interpret information from the panel is reduced thus saving valuable time in severe disorientation scenarios such as high workload or psychological stress [10].

Peripheral vision still works well when the retinal image is blurred, as it often can be by severe turbulence and/or vibrations. Since disorientation is often the result of turbulence or vibrations, it is advisable to provide information to the ambient mode which functions better in reduced retinal clarity. During severe flight conditions, a condition called ‘pseudo-myopia’ occurs making it difficult for some pilots to read the instrument panel. This can lead to increased spatial disorientation and providing cues to the ambient vision can help reduce some of the adverse effects [10].

An ambient vision device is usually large and hence is easier to see and interpret if the pilot is experiencing disorientation [10].

When attitude information is provided to the pilot via an ambient vision device, he/she gets the information continuously no matter what his/her task at hand. Thus unperceived changes in attitude are easier to detect and the pilot can react to them quickly and more efficiently [10].

The above reasons led a Dr. Richard Malcolm to invent a crude peripheral vision system that would provide peripheral cues to the pilot when in flight. The display dubbed 'The Malcolm Horizon' in its earliest version was a light bulb surrounded by a highly polished cylindrical metal reflector. The cylinder and the bulb were mounted on gimbals and connected to tiny electric motors and the entire system was driven by a joystick. The system could provide the pitch and roll of the aircraft and was reflected onto the instrument panel of the aircraft. One of the earliest experiments with the Malcolm Horizon was in a moving base simulator of the Sea King helicopter belonging to the Canadian Armed Forces [8].

Investigation on the Malcolm horizon commenced in the early 70's and further testing and investigation led to the development of the Peripheral Vision Horizon Display (PVHD). The PVHD was developed in the early 80's with preliminary testing on the US Air Forces' SR-71 Black Bird reconnaissance aircraft.

2.3 The Peripheral Vision Horizon Display

The Peripheral Vision Horizon Display (PVHD) is a continuous expanded artificial horizon produced by sweeping a red laser at high frequency across the instrument panel in front of the pilot. The resulting line is about 0.6 cm wide and 60 cm long with its roll axis centered directly in front of the pilot [11].

The concept behind the development of the PVHD was to provide unconscious attitude cues to the pilot through his peripheral vision that would reduce the likelihood/severity of spatial disorientation and serve as secondary attitude indicator (AI).

This would allow the pilot more time to concentrate on other instruments. The end result would be improved performance during unusual attitude recoveries and precise instrument approaches in IMC. With the PVHD system, the cockpit of the aircraft or the instrument panel becomes the stable reference and the artificial horizon is regarded as the moving object.

Various tests were conducted with the PVHD in aircrafts such as the F-111, F-15, F-16, NASA T-37, Calspan NT-33 and single seat night attack (SSNA) A-10. The results of these tests were documented in a conference held by the NASA Ames Research Center in March 15-16, 1983 at the Dryden Research Center. The following conclusions were drawn based on these tests.

2.4 Advantages of using the PVHD

i. For the F-111, with plenty of instrument panel available, the PVHD appeared quite compatible. The roll axis if centered in front of the pilot provided accurate roll-pitch illusion [12].

ii. The perpetual reversal of roll information from the standard artificial horizon, that occurs occasionally even in the experienced pilots, was assumed to have less likelihood of occurrence with a peripheral vision device [10].

iii. The PVHD would serve as an orientation device and visual workload reduction device; i.e., it would alert the pilot of sudden attitude changes even when pilots' attention was centered away from the instrument panel [13].

iv. It seemed to detect the aircraft movements and bring the pilots' attention to the Attitude Indicator (AI). Less time was spent looking at the AI as the pilot was made aware of the changes in attitude [13].

v. Certain deviations in certain flight parameters (e.g., heading and rate of climb errors) were reduced with the PVHD [13].

Though the PVHD seems to provide the information necessary to alleviate the effects of spatial disorientation; the following drawbacks make the system currently impractical for GA use.

2.5 Disadvantages of PVHD

i. The quality of the horizon projection needs improvement; bright dots are substituted for the horizon at lower power settings, and when a line appears to connect the dots, it wavers continuously. It is felt that unless the display is corrected to appear sharp and distinct as expected of laser, subconscious adaptation to the PVHD may be prolonged or may never occur [12].

ii. Although the PVHD has ten discrete brightness levels, it is too dim for effective use in any form of daylight [13].

iii. For the F-15, there were occasional annoying reflections. Also the PVHD did not show up when projected onto multifunction display (MFD) surfaces, although this could be corrected with a different surface coating [12].

iv. For the F-16, surface on the upper panel was limited and that surface which was available was broken up by the HUD control panel. Thus the PVHD was useful only as a head down display when projected below the HUD control panel [12].

v. When aimed too low, the PVHD struck the pilot's knees which jut up above displays on the centre pedestal, due to the tilt-back seat [12].

vi. In the SSNA A-10, due to low display area, the nominal position of the horizon line was moved to the middle of the upper instrument panel. Since the actual range of the PVHD could not be changed to accommodate this change in position, the PVHD line could move only a limited distance in the pitch down direction before reaching its limits [14].

vii. Due to the low display area, substantial portions of the area were blocked from the pilot's vision by the stick and the pilot's arm. The pilot's peripheral vision was also reduced by his oxygen mask [14].

viii. When the upper display area was used, the PVHD caused major reflection in the pilots' eyes due to its installation geometry and location of right MFD (Multi Function Display) [14].

ix. For the RF-4C aircraft; although it had ten different brightness levels, the display was too dim for effective use in any form of daylight [15].

x. The pitch of the aircraft was barely discernible, regardless of the scale chosen; even with a scale of 3:1(one degree of PVHD movement corresponds to three degrees of aircraft pitch)[15].

xi. Also a “pendulum effect” was observed in the roll if the display was repositioned in pitch at other than its centre; i.e. the display rolled about a point other than the intersection of the horizon line and the sky pointer [15].

xii. The PVHD uses a red laser. Red laser signifies danger. Red light striking a red enunciator will give a momentary red flash to the pilot, which adds stress to the pilot [15].

xiii. Motion of the display unnecessarily drew the pilots’ attention. Thus the pilot noticed changes in attitude which he would not have noticed otherwise which proved to be distracting [13].

xiv. Awareness of peripheral information was less likely under a high workload [11].

xv. With a PVHD system, it is difficult to achieve a suitable installation in a fighter type aircraft cockpit like the SSNA [14].

Recently, interest in the Malcolm horizon has resurfaced with the goal to reduce instrument related spatial disorientation. Simulations were carried out where the size of the horizon was varied along with its brightness levels.

Preliminary studies conducted by the NASA Langley Research Center show that the Malcolm horizon does improve the spatial awareness though it was tested with only roll information and is inconclusive with regards to testing with pitch as well as roll being displayed [16].

Furthermore the PVHD is being researched by Munro and Associates along with the Michigan SATS Lab and their initial findings were presented at the 2004 EAA Airventure, Oshkosh, Wisconsin at the National Consortium for Aviation Mobility [7].

Thus, the PVHD as an aid to attitude awareness needs to be evaluated more thoroughly as the drawbacks currently outweigh the virtues of the system.

This leads us to the work done by Douglas Burch and Dr. Michael Braasch at the Avionics Engineering Center at Ohio University. The Synthetic Vision System had originally been developed to provide only foveal information by generating a photographic view of the outside world. This was later modified to provide peripheral view as well. The system will be described in detail in the following chapters.

3. General Aviation Synthetic Vision Display (GASVD)

We have seen the importance of peripheral vision in maintaining orientation in three-dimensional space. We have also seen the studies that have been conducted on the most prominent version of a peripheral vision system. Keeping that in mind we delve deeper into synthetic vision systems and similar research work done here at Ohio University. A synthetic vision system was developed by Douglas Burch under the guidance of Dr. Michael Braasch [17]. This system aims to serve as a secondary instrument for guiding the GA pilot. This chapter deals with the work done by Douglas Burch in the development and implementation of this system which forms the basis of this research.

3.1 Background Work Done By Douglas Burch

The display system initially developed has the following components: A GPS Receiver, an Attitude and Heading Reference System (AHRS), a Data Processing Computer, Synthetic Vision System and the Display unit.

3.1.1 NovAtel GPS Receiver

The NovAtel OEM-4 3151R Power Pack GPS Receiver provides independent position and velocity measurements at a rate of 20Hz. It calculates the velocity directly from the Doppler measurements made by the receiver. The position and velocity measurements are sent in ASCII format at a data rate of 115,200 baud. Due to the complexity of the binary string format, data was chosen to be transmitted to the Data Processing Computer in ASCII format. The position string contains latitude, longitude and MSL height of the antenna where as the velocity string contains vertical speed, horizontal speed and the ground track which is used to augment the measurements from the AHRS [17].

3.1.2 Attitude and Heading Reference System

The AHRS (Attitude and Heading Reference System) provides robust roll and pitch measurements at the rate of 60 Hz to the Data Collection Computer. The unit in use is the AHRS 400CC-100™ built by Crossbow® designed for high accuracy measurements under low flight dynamics with gyroscopic and accelerometer capabilities of ±100˚/sec and ± 2G respectively.

It is a strap down inertial MEMS (Microelectromechanical System) subsystem with unit dimensions 3.0 in X 3.75 in X 4.1 in. It has a start up time of less than 1.0 seconds and is fully stabilized in less than 1 minute. It has a wide input voltage range ranging from 9 to 30 volts drawing only 275 milliamps making it ideal for GA applications [18].

3.1.3 Data Processing Computer

The Data Processing Computer utilizes the QNX 4.25 operating system. The computer interfaces with the AHRS unit as well the GPS Receiver via two RS-232 data links and computes attitude information using software written in C. This information is converted into a state vector and then sent to the synthetic vision computer [17].

3.1.4 Synthetic Vision System

The static terrain database is stored on a separate computer which runs on Microsoft Windows© operating system. A program written in Visual Basic© manipulates the Revolution – 3D graphics engine [19] to render the synthetic view of the outside world. The program utilizes the state vector data sent to it by the Data Processing Computer to manipulate the virtually generated camera. The touchdown point of runway 25 at UNI is the surveyed point which provides the basis for the static terrain database. This provides the pilot with the forward looking view of the outside world. This forward view is depicted on the display unit. Figure 3.1 shows the architecture of the system that was initially implemented.

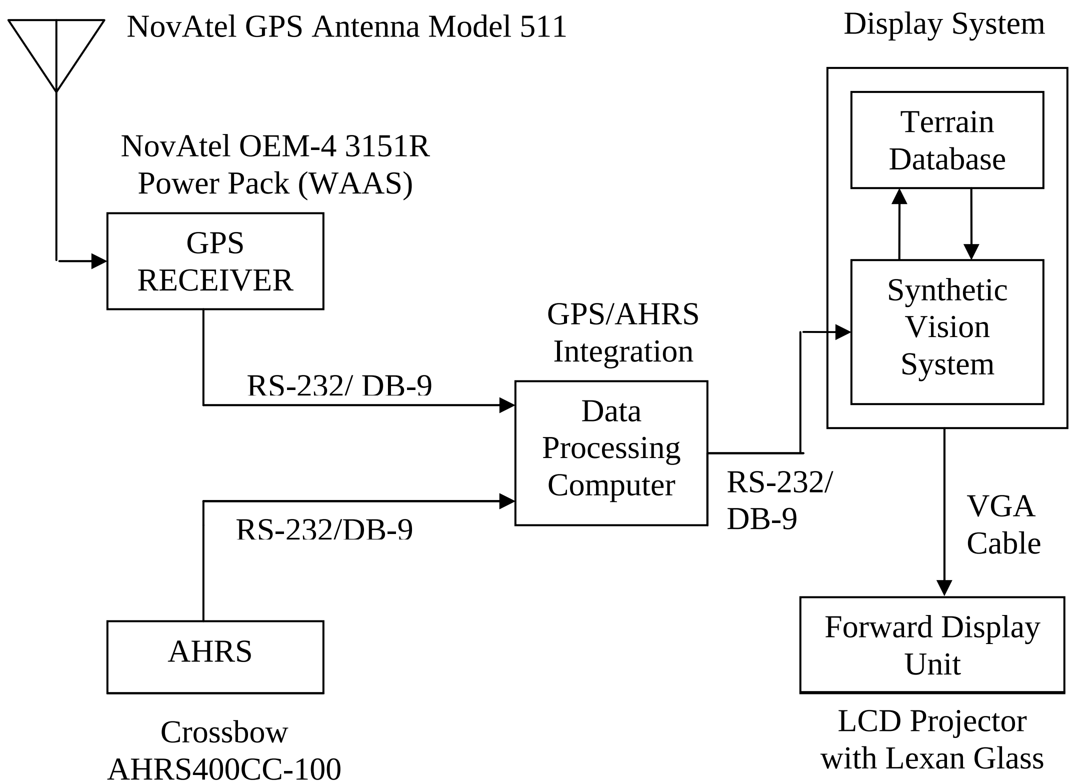

Figure 3.1: Architecture of the Head-Up Display System

The system so described was developed and tested by Douglas Burch and his results were presented in his Masters Thesis titled “Implementation and Evaluation of a General Aviation Synthetic Vision Display System”.

This system was then modified so that it would be able to provide peripheral cues as well. This was achieved by the incorporation of a right side view for the synthetic vision system by modifying the Visual Basic code such that the virtual camera would now point to the right of the pilot.

3.2 Multi-View Display Setup

Take-off and landing which are performed during every flight, continue to be the riskiest phases of any flight. In 2001, there were 634 take-off and landing accidents which account for nearly two accidents a day. The fatality rate for these accidents was 23.5% [5]. Thus, the need for additional guidance during these critical phases of flight cannot be over emphasized. During climb, the forward looking view would essentially show only the sky above. It is in this phase of flight that the pilot finds himself/herself intuitively looking out of the window for peripheral guidance. Thus, it was thought that a modification to the synthetic vision software be brought about such that it could give the pilot a side looking view as well. The Visual Basic code was modified and loaded on another computer so that it now showed a right side virtual view of the outside world. The modified software used the same data packets from the Data Processing Computer. The software could have been modified to provide a left side view as well but due to issues regarding the optimum placement of the display unit, this scheme has currently been abandoned. It is definitely a topic for further research and implementation. Figure 3.2 shows the architecture of the multi view synthetic vision display system.

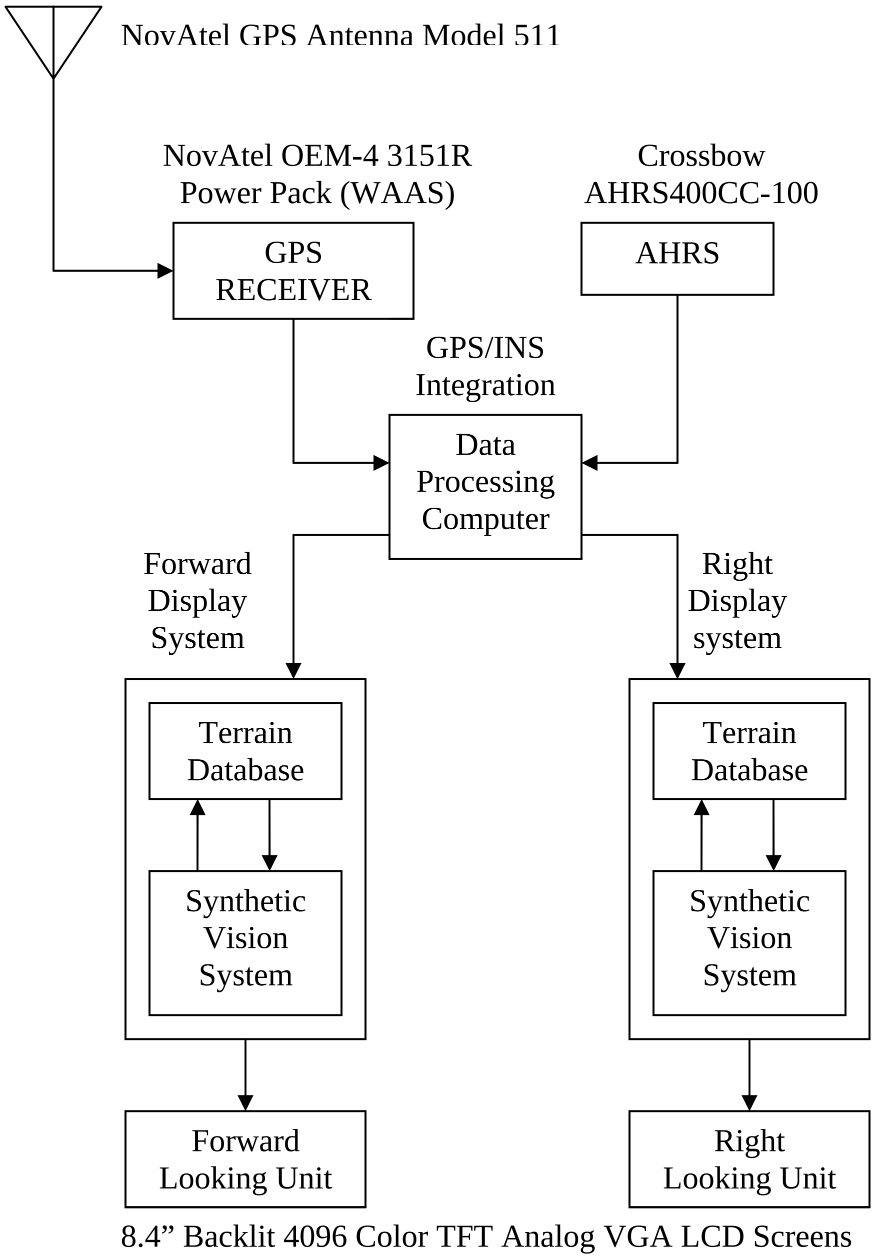

Figure 3.2: Architecture of Multi View Display System

3.2.1 Display Unit

For this upgraded configuration the display units chosen were 8.4" Backlit 4096 Color TFT Analog VGA LCD screens. The screen is a Thin Film Transistor Liquid Crystal Display with a separate AD2 controller board requiring a 5V DC supply. The software for the synthetic vision system was loaded on two different laptops. The choice of the TFT LCD screens allowed the direct routing of the laptops to the two screens via VGA cables. This eliminates the need for the projector and the Combiner Lexan Glass for rendition of the image. The forward looking display was run from a Dell® Inspiron 8000™ Laptop and the right looking display was run from the HP Pavilion XT125 1.6 GHz Celeron Laptop.

3.3 Placement of the Display Units

3.3.1 Forward Display Unit - Panel Mounted VS Head-Up Display

Earlier flight tests of the display system used a forward-looking head up configuration. The advantage provided by this approach was that the pilot had an “out-of-the-window” view of the real world. A head up display would provide the pilot with a view of the synthetic world superimposed on the real world. This was changed to a panel mounted configuration in the later stage as the following benefits ensued [20]:

Installation

The panel mounted display provides ease of installation and also a choice as to its placement in the panel with respect to the pilots’ as well as the engineers’ discretion.

The head-up configuration employed a projector for rendering the image onto the Lexan 9034 combiner glass screen. The projector can now be eliminated from the setup which reduces the installation work load and cost [20].

Usability

Panel mounting of the display makes it more accessible and flexible in terms of use. The forward screen is hinged to the glare shield of the instrument panel, thus it can be folded away when not needed [20].

Power consumption

The head up display requires an LCD projector for display purposes which increases power consumption. The panel mounted display can be connected directly to the laptop. This reduces the load of the power supply of the airplane and mitigates the need for the UPS (Uninterruptible Power Supply) [20].

Safety

As the display is not in the field of view of the pilot, it does not obstruct his view of the outside world. This reduces the risk factor during VMC flying conditions and also in a situation where the pilot might just be breaking out of the clouds. The synthetic vision system is intended to be used as a secondary instrument for additional guidance. A panel mounted configuration would encourage the pilot to use the system only when necessary [20].

Cost

The panel mounted display is less expensive in terms of installation cost as well as power consumption [20].

Figures 3.3 and 3.4 show the panel mounted display when in use and also when it is stowed away.

Figure 3.3: Forward Looking Display (When in use)

Figure 3.4: Forward Looking Display (Folded Up)

3.3.2 Right Display Unit

The decision to include the right display in the architecture was manifold. Most importantly, it was thought that this would provide the pilot with the peripheral cues that he/she would need during take-off, landing and during banking maneuvers. Also the right side view would help augment the information that was already provided by the forward view. As this display was in the peripheral vision of the pilot it would not provide any distractions yet at the same time, if the forward display was stowed away, it would inform the pilot of any sudden deviations in the pattern of flight.

3.3.3 Placement of the Right Display Unit

It was decided that the most suitable mount for the display unit would be the right side door of the airplane. This way it would not obstruct the pilots' view of the right side window. The screen for the right looking display was mounted on the arm rest of the door. L-shaped aluminum plates were welded together and then fixed to the door to secure the screen in place as seen in Figure 3.5.

The display unit was strategically placed such that the body of the safety pilot seated to the right of the pilot would not obstruct his view of the screen.

Figure 3.5: Right Looking Display

3.4 Equipment Installation

The equipment for the flight test was mounted in the airplane in a 19” rack made from 2024 T3 aluminum. The rack has slots on which the various pieces of equipment are mounted. The bottom shelf is used to mount two 14V-28V DC/DC converters and one 28V–115V DC/AC inverter. The Data Processing Computer is mounted on the second shelf. An aluminum bracket is used to secure the Data Collection computer onto the rack [20].

Figure 3.6: Data Processing Computer and Power Supply

The AD2 controller boards for the TFT LCD monitors are mounted on an aluminum plate fixed on the top shelf of the rack and covered due to their delicate nature. Nylon fasteners are used to fasten the covers to provide adequate insulation. Extension cords are used to connect the TFT LCD screens to the AD2 controller boards. The GPS receiver is also mounted on the aluminum plate. A mount was built for the AHRS unit near the center of gravity of the aircraft [20].

Figure 3.7: GPS Receiver Mounted on the Equipment Rack

Figure 3.8: AHRS Mounted Near the CG of the Aircraft

3.5 Power Requirements

The two converters mounted on the bottom shelf are needed to satisfy the current requirements of the Data Processing Computer. The power supply for the controller circuits of the LCD screens comes from a terminal strip which is powered by the airplane itself. The terminal strip is also used to power the GPS Receiver and the AHRS unit. The previous architecture consisted of a lead acid battery powering the GPS and AHRS unit. The Data Collection Computer was powered using a UPS (Uninterruptible Power

Supply). Due to the limited supply of power and issues concerning the weight of the UPS, its use was discontinued [20].

3.6 Flight Test Airplane

The Avionics Engineering Center's 1980 Piper Saratoga (N8238C) has been used to conduct the various tests as the system has developed through various stages. It is a single engine, low-wing aircraft, which can seat up to six occupants. It runs on a Lycoming IO-540-K1G5 (300 Hp) engine with a 3-blade constant speed propeller. It can carry up to 1384 lbs. of useful weight apart from its own empty weight of 2216 lbs. The Piper Saratoga has adequate space for the equipment to be installed in the passenger area. The rack that was built for the equipment was placed behind the seat of the safety pilot. This allowed adequate seating for the research engineer to sit at the back to monitor all the equipment. The Piper Saratoga was flown on every occasion during the development and testing phases of the various Peripheral vision displays.

Figure 3.9: The Piper Saratoga

4. Flight Testing the Multi-View General Aviation Synthetic Vision Display (GASVD)

The previous chapter describes the various components of the general aviation synthetic vision display or the GASVD. The two screens that were used for primary guidance were the forward display and the right display. These screens are so placed as to give maximum focal as well as ambient cues to the pilot in flight.

As the GASVD consists of a forward and a right looking display, it was decided that during the flight test, right traffic patterns would be performed, specifically, to test the functionality and usability of the right looking display.

4.1 The Hazard Action Matrix (HAM)

Before every flight test, a hazard analysis has to be performed to ensure that there is no possibility of loss of life and/or property [21]. In accordance with this procedure a Hazard Action Matrix (HAM) was prepared. The HAM identifies potential hazards then determines the hazard severity and probability. It addresses certain generic issues such as electrical rack power supply failure, failure of research equipment, external antenna and provides the precautions as well as remedies for the same. Alleviating actions include thorough pre and post-flight inspections. In case a hazard does occur, remedies must be assigned for each failure mode. The HAM also addresses issues specific to the current flight test such as partial blockage of the instrument panel due to the panel mounted forward display, blocking of the right looking display due to the safety pilot in the right seat. These situations were remedied before the flight test was conducted.

The forward display blocks a partial view of the altimeter, thus MSL height in feet is displayed on the screen. Also during take-off and landing the display is folded away so that it does not obstruct any of the panel instruments. The display screen on the right is mounted on the armrest of the right door to provide maximum viewing angle to the pilot during flight.

4.2 Flight Testing the GASVD

4.2.1 First Flight Test

Two flight tests were conducted to evaluate the performance of the GASVD. The first test was conducted at 12:15 hours on the 10th of September 2003 at UNI airport Albany. Weather conditions were VFR (Visual Flight Rules). Several right traffic patterns were flown on UNI 7. The data collected by the software from the AHRS and GPS Receiver at 60Hz and 20Hz was processed real time and sent to the synthetic vision computer in the form of a state vector. Due to the time of the day the display was unreadable. This was due to the fact that the forward display was mounted on the glare shield in the cockpit and was not afforded the same glare protection as the instruments. The flight test was concluded and the second flight test was scheduled for a near dusk flight.

4.2.2 Second Flight Test

The second flight test was conducted at 19:45 hours on the 17th of September 2003 at UNI airport, Albany. The weather was clear (VFR). Four right traffic patterns were made on UNI 7. The flight test was recorded on video to document the performance of the GASVD.

The display performed flawlessly from an engineering perspective. It provided the pilot with accurate attitude information and rendered situational awareness with respect to the runway. During take-off the forward display was folded out of the way and the pilot was told to fly visually with visual cues from outside the window. Once the pilot was adjusted he was asked to fly the right pattern using the displays for primary guidance. The forward as well as the right view, in conjunction, provided the pilot with the necessary situational awareness with respect to the runway. During turns in the right pattern, the banking as viewed from right display provided the necessary guidance the pilot needed in terms of his position with respect to the runway. Figure 4.1 shows the right traffic patterns flown over the airport.

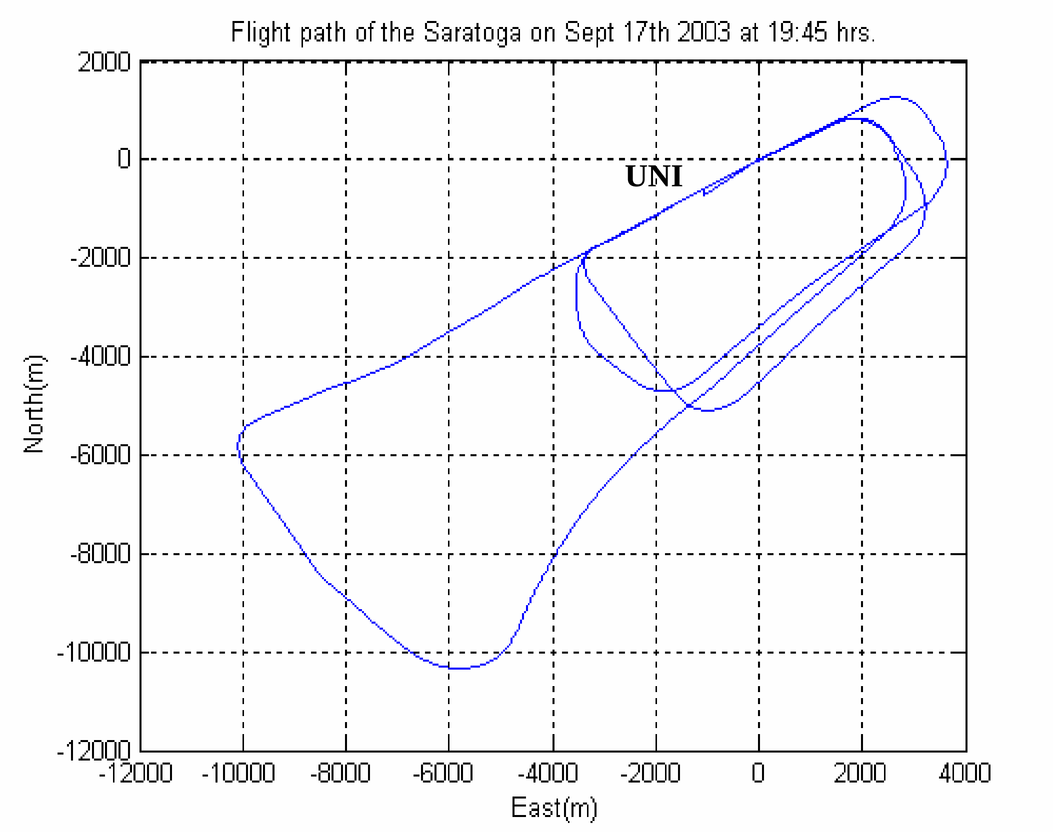

Figure 4.1: Right Traffic Patterns Flown by the Saratoga on Sept 17th, 2003.

Figure 4.2 shows the pitch and roll and ground track of the aircraft as measured by the AHRS unit during the flight test.

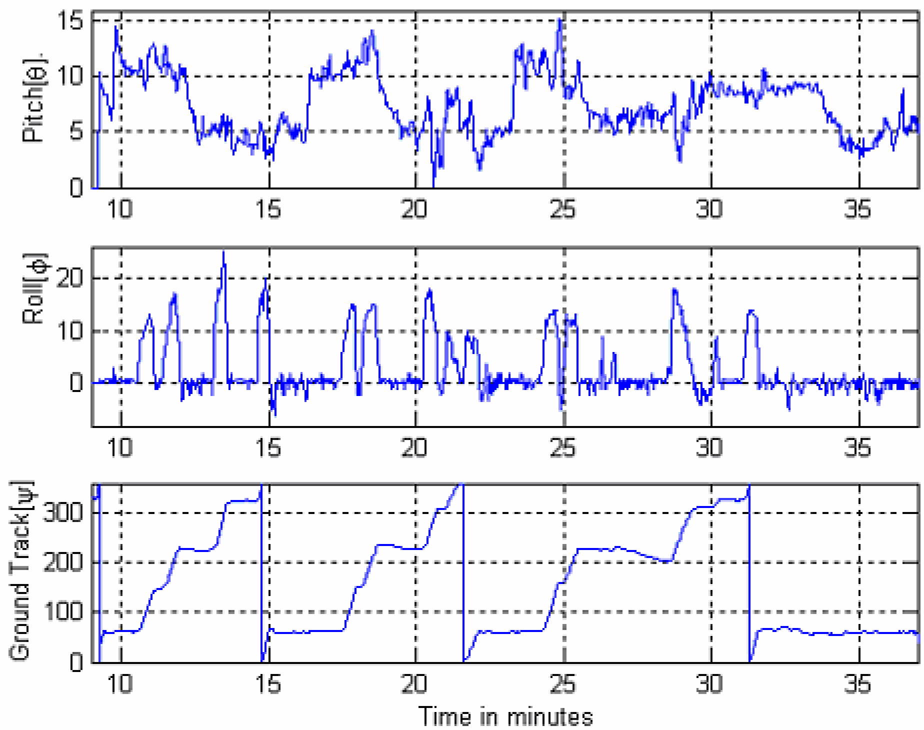

Figure 4.2: Pitch, Roll and Ground Track of the Flight Path.

During the flight test the pilot was asked to use the forward as well as the right display for primary guidance. The forward display was to provide an 'out of the window' view to the pilot. The right display was for orientation information. As the right display was visible in the peripheral view of the pilot, it was to provide visual cues indicating the attitude of the aircraft.

4.3 Performance of the Forward Looking Display

Figure 4.3 shows the forward display system during a right turn made in the crosswind leg of the right traffic pattern. It can be seen that the horizon of the real world matches that displayed on the screen, conveying that the pitch and roll information are accurate and real time. This makes the panel mounted display more viable in terms of image rendition.

Figure 4.3: Forward Display in a Right Turn.

Figure 4.4: Forward display during Low Approach.

Figure 4.4 shows the forward display during a low approach over runway 25 at the Ohio University airport. The runway placement is skewed. This is due to GPS positioning error. We use open loop GPS and hence get an error of about 3 meters in our calculations.

4.4 Performance of the Right Looking Display

The right looking display placed at the armrest of the right door was to be used mainly for orientation information. Figure 4.5 shows the image as seen in the display screen.

Figure 4.5: Right Display during Downwind Leg of Traffic Pattern.

The right display screen did provide the necessary attitude information that was needed, especially during banking. The pilot had visual cues to corroborate the information given by his instruments when the aircraft made a turn or changed its profile from straight and level flight. This helped the pilot to judge the attitude of his aircraft without looking at the instruments all the time.

There were a few difficulties that were faced during the flight test. The display was to the right of the pilot and as it was placed on the door, it was beyond the peripheral field of vision of the pilot. It had to be directly looked at for spatial information.

There was discrepancy in image size as can be seen in Figure 4.5. This can be remedied by reducing the amount of peripheral information that is displayed on the screen.

This led us to believe that though the idea of a side display did hold merit, we would need better placement and greater visual detail for it be provide accurate guidance to the pilot.

4.5 Pilot Perspective

During as well as post flight, several comments were made by the pilot regarding the performance of the GASVD, which are as follows:

The system could be very useful as a secondary instrument for increased situational awareness. When both the displays were used in tandem, it was found that the pilot thought it easier to navigate the turns in the traffic pattern. The drawback was the lack of altitude information. This can be remedied by having a digital readout of the altitude on the forward screen or by having a graphic altitude bar on the side of the screen.

For increased situational awareness and conformance to the ‘out of the window’ view, it was suggested to have parts of the airplane depicted in the screen; for e.g.; the nose in the forward display and the wing tips in the side display. This would give the pilot a visual reference as to how the aircraft is aligned with respect to the synthesized objects.

The pilots recommended the addition of a bird’s eye or top-down display to allow for better situational awareness similar to a moving map display. It was suggested that it be possible to switch between the side display and the birds eye view allowing the pilot to vector his flight route more accurately.

As the flight was at dusk time the brightness of the screen was a concern and provision should be made to adjust contrast as well as switch off the screen at will. This would allow the pilot to discontinue the use of the screen when not required.

4.6 Issues Regarding the Side Display

The flight tests as well as the feedback from the pilots validated that a side display was a helpful tool in increasing situational awareness and spatial orientation. The use of the side display is, however, thwarted by some issues.

The side displays would provide maximum information if located on the window of the airplane, in direct peripheral view of the pilot. This is not possible, as it would interfere with the viewing of air traffic outside the window, thus greatly reducing its efficacy.

For a pilot in the left seat, the side display on the left armrest renders a very poor viewing angle to the pilot with reduced image clarity due to the proximity to the screen. A side display to the right of the pilot on the armrest might be blocked by the body of the right seat passenger.

Installation of the side display is an arduous procedure, as routing cables to the screen have to be secured out of the way of the pilot and passenger. Also the placement of the display screen on the armrest makes it vulnerable to damage.

These and other difficulties with the side display led to the conceptualization of the LED- Forward View Symbiotic Display.

5. LED- Forward View Symbiotic Display: Concept and Architecture

Peripheral visual cues when provided to the pilot not only increase situational awareness but reduce dependence on the instrument panel. The advantage of providing spatial information to the pilot in his peripheral field of vision is that he can choose to ignore it when deemed unnecessary. It is a continuous source of information which does not get in the way of the other instruments and provides the constant feedback that is necessary in maintaining orientation.

The development and flight testing of the side displays led to the deeper understanding of the type of peripheral guidance system that is needed. It should provide the necessary information without causing any major installation issues and also be positioned such that it affords maximum benefit to the pilot.

The PVHD demonstrated the advantage of displaying a horizon line over a wide field of view. The concept was unique but was encumbered with issues such as operation, reflection off the instrument panel and the canopy etc and thus, could not provide the necessary peripheral guidance.

5.1 The LED – Forward View Symbiotic Display

At a JUP meeting held in October 2003 at MIT, Boston it was suggested by Prof. R. John Hansman of MIT to introduce vertical LED strips in the peripheral field. The LEDs on the two strips placed to the left and right of the pilot would light up to indicate the horizon. A third strip in the center would light up to indicate the pitch of the aircraft thus giving the pilot an indication of the attitude of the aircraft. The strips would light up half way to indicate straight and level flight and then adjust for non zero pitch and roll. This concept infused the enthusiasm to develop and implement the modified synthetic display system.

5.2 Installation of the LED strips

The test aircraft for this hybrid system was the Piper Saratoga. The figure below shows the location selected for the mounting of the LED strips.

Figure 5.1: Placement of the LED Strips

This type of mounting ensures that the LED strips provide peripheral information which is essentially ‘roll’. This offers us a number of advantages:

i. Ease of installation as the strips can be attached to the two side panels using Velcro tape, making it easy to install and remove.

ii. The strips can provide peripheral information without obstructing the outside view of the world.

iii. The strips do not require additional mounting space on the instrument panel and thus does not obstruct the viewing of any of the instruments.

iv. The position of the LED strips is such that they do not provide distraction to the pilot and provide information only when needed.

v. The strips will not cause any canopy reflections, thus reducing the distractions.

5.3 Color of the LEDs

One important aspect of the LED strips was the selection of the color of the LEDs. In the PVHD system, red was chosen for painting the horizon line across the instrument panel. As already seen, this caused a number of problems. Red inherently signifies danger. When used in the PVHD system it caused major reflection in the pilot’s eyes as well as canopy reflections.

The choice of color of the LEDs proved to be a tricky one. The human eyes provide a motion sensor system with nearly 180 degrees horizontal coverage. The eye's peripheral vision system only supports low resolution imaging but offers an excellent ability to detect movement through a wide range of illumination levels, due to the sensors called rods which cover the majority of the eye's internal retina layer. It is also responsible for the very wide angle of our peripheral vision but it provides very little color information. The eye's high resolution color vision system provided by the sensors called cones, has a much narrower angle of coverage and is localized to the back of the eye near the fovea. This system can flexibly adapt to widely varying illumination colors and levels [22].

Any significant motion detected in the peripheral field prompts our eyes to try and identify its cause by looking directly at it, even if that is for the fraction of a second. This prompts us to believe that though the color of the LEDs may not be immediately apparent in the peripheral field, if the movement caused by the lighting of the LEDs was significant, then the eyes would momentarily glance in its direction and identify the color that was being lit at that instant.

The color chosen for initial testing was green. The subject of color was left open to change with respect to the pilots’ response to the system during flight testing with candidate colors being blue and yellow.

5.4 Architecture of the LED – Forward View Symbiotic Display

The architecture remains the same as the GASVD with a few changes to incorporate the replacement of the side display with the LED system. The components that remain the same are the NovAtel GPS Receiver, Attitude and Heading Reference System, Data Processing Computer, Synthetic Vision System which have been described in Chapter 3.

The changes made to the architecture of the GASVD comprise of the Microcontroller Circuitry and the LED strips.

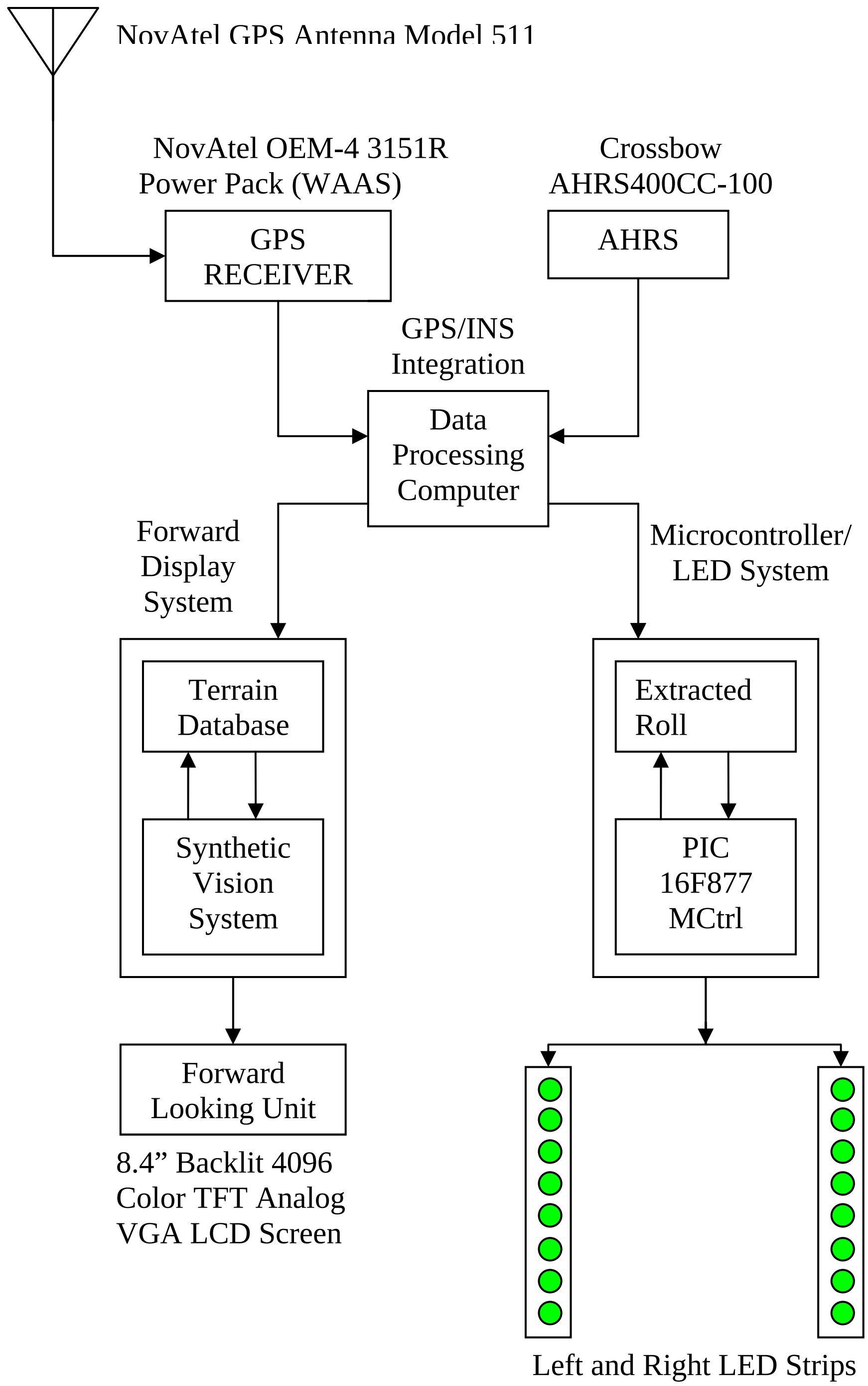

Figure 5.2: Architecture of LED – Forward View Symbiotic Display

5.5 LED Microcontroller System

Figure 5.3 shows the microcontroller circuit for the LED – Forward View Symbiotic Display.

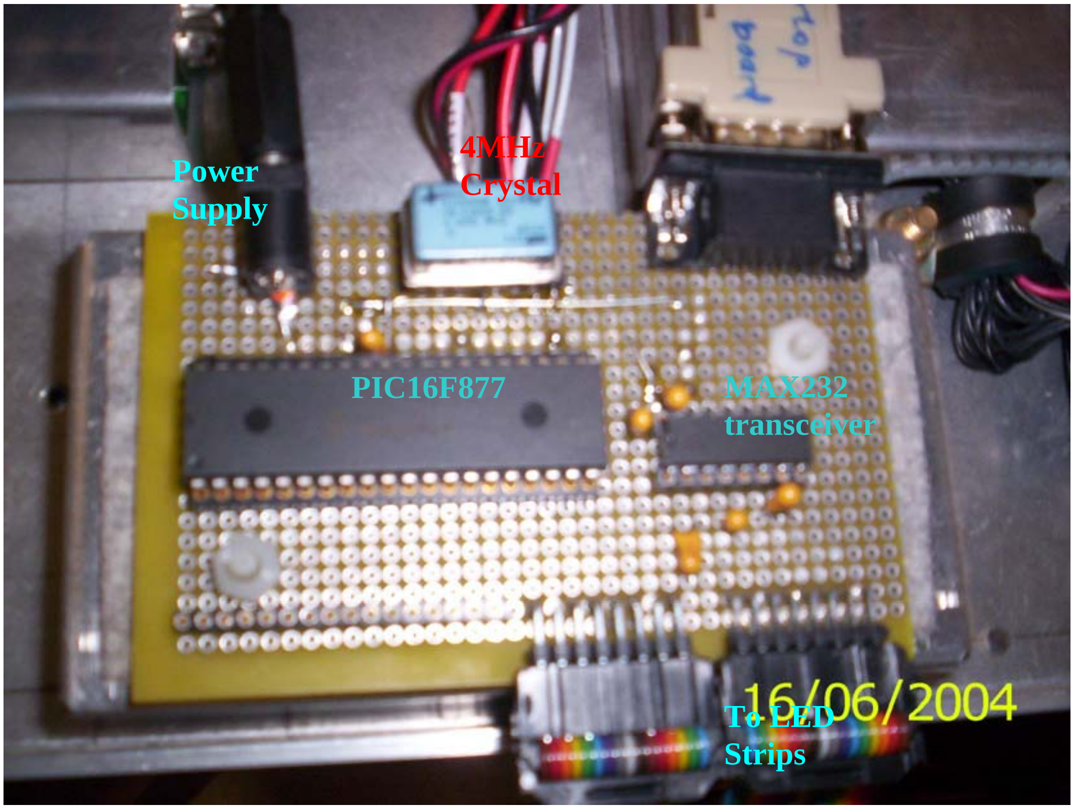

Figure 5.3: Microcontroller Circuitry

The components of the microcontroller circuitry are the PIC16F877, a 4MHz crystal oscillator, a MAX232 transceiver. Connectors are soldered onto the PCB for the RS232 cable the two LED strips and the power supply. Figure 5.4 shows the connection diagram for the various components.

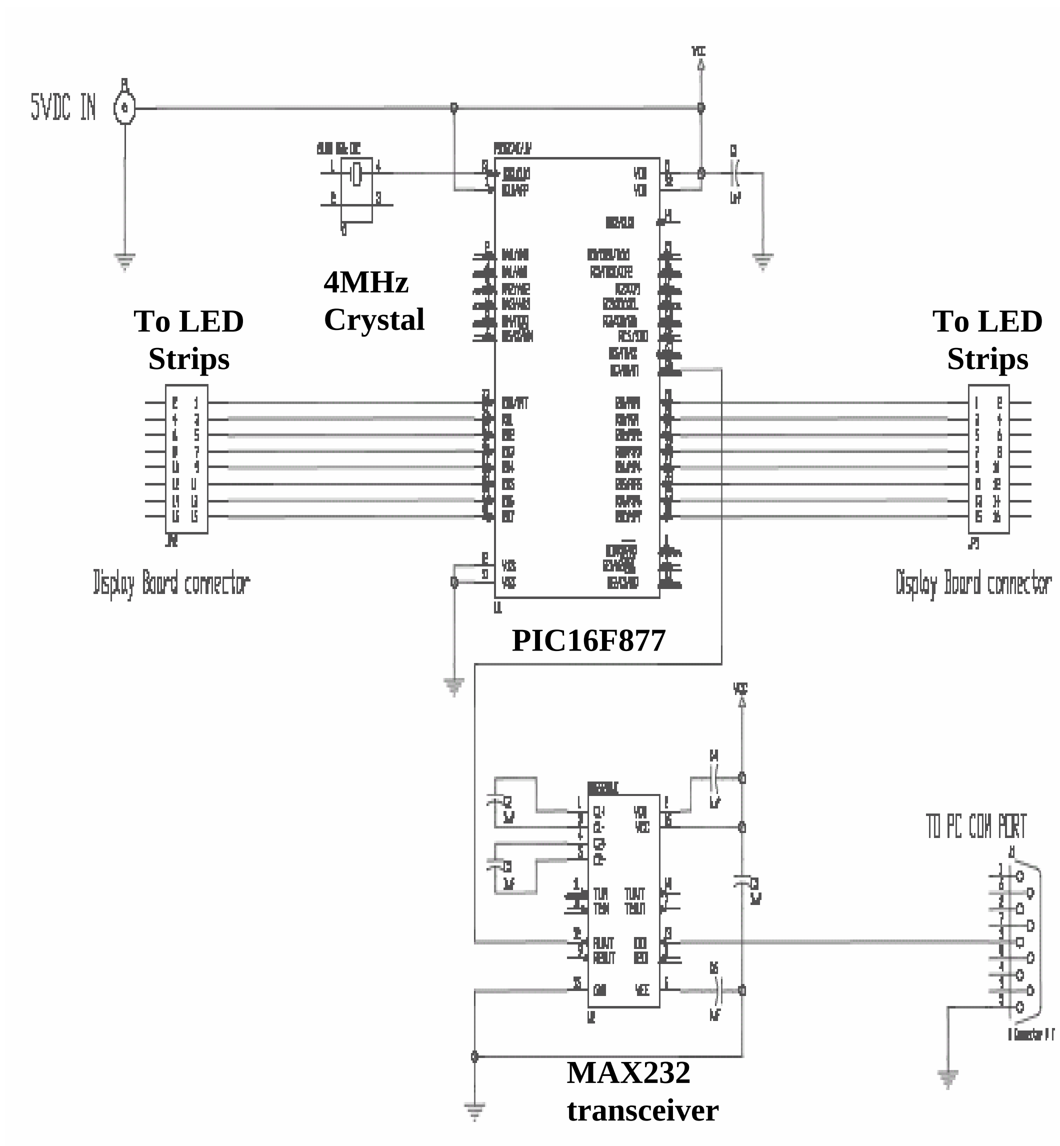

Figure 5.4: Connection Diagram for the Microcontroller Circuitry

The PIC16F877 is a 40 pin 8-bit CMOS FLASH Microcontroller. Due to its EEPROM capability, it is easy to test the program which is coded on the chip. The chip can be conveniently reprogrammed several times allowing changes to the original code with ease.

The 4MHz crystal oscillator is chosen to provide timing signals to the microcontroller. The MAX232 Transceiver is a single supply RS232 device that allows data flow between the data processing computer and the microcontroller. It also provides power to the RS232 cable and has high noise immunity. The power supply to the PCB is a 5V DC Supply at 25mA.

5.6 Types of LEDs

There are many LEDs in the market and they are classified according to color, type of mount, size, application, type of encapsulation, brightness etc.

The most standard LED type- the Super Bright T1 ¾ was used in our LED strips. It has a cross-sectional diameter of 5mm. It has a dome shaped encapsulation and two leads for connection; the shorter lead being the cathode. The LEDs with a diffused lens were chosen as these provide a greater viewing angle than the clear-lens LEDs. Diffused LEDs also reduce the incidence of stress to the eyes of the pilot as the light is not directed in just one particular direction as is the case for clear-lens LEDs. Figure5.5 [23] shows the structure of the LEDs.

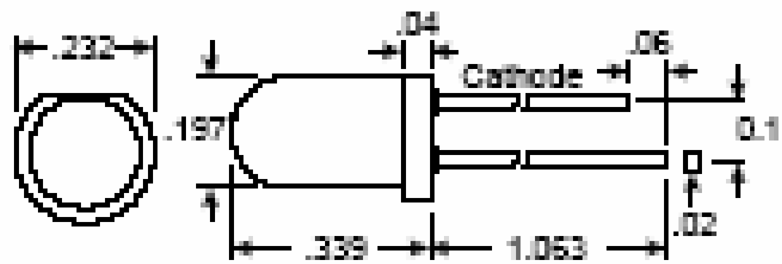

Figure 5.5: Diagram of the type of LEDs used (measurements in inches).

Initially when the strips were built, only green LEDs were chosen to be tested on the strips. After the first flight test, it was suggested that other colors also be used and hence yellow and red LEDs were also tested. The following table describes the specification of the LEDs that were used [23].

Forward Current = 20mA

COLOR		FORWARD VOLTAGE (Vr in volts)	LUMINOUS INTENSITY(mcd)		VIEWING ANGLE	MANUFACTURER
LED	LENS		Min	Typ		
Green	Diffused	2.2	40	60	60^o	Kingbright LEDs
Yellow	Diffused	1.7	180	300	44^o	Xicon LEDs
Red	Diffused	1.85	200	300	60^o	Kingbright LEDs

Table 5.1: Types of LEDs

Luminous intensity above 600 mcd tended to cause discomfort to the eye when coming from a point source as was tested by using LEDs with higher intensities. The specified LEDs were thus chosen as they did not cause any reflections as well as stress to the eyes.

5.7 LED Strips

Two LED strips were built to be placed as shown in Figure 5.1. The strips are built out of standard FR4 PCB of 6.2mm thickness with a green solder mask. Each strip is 40X3 cms long with single inline header sockets fixed on them. These sockets allow us to change the LEDs as we wish. This is a very useful feature as it allowed us to test different colored LEDs during flight testing without having to build new strips each time. The sockets were fitted such that 56 LEDs fitted on each strip. The length of the strip afforded the use of the standard 5mm LEDs as described earlier.

5.8 LED circuit

As can be seen from Figure 5.5, there are 56 LEDs on each of the strips. The strips are connected to jumpers which are then connected to the Microcontroller circuitry. The LEDs are connected in a series/parallel combination. As seen in the diagram the LEDs are connected row wise, starting with the first LED on the topmost left corner-going down the row and then connecting to the left most LED on the second row and so on. Thus the LEDs in a row are serially connected while the LEDs in one column are in a parallel connection. At the beginning of each row is a current limiting resistor of 70 ohms. The LEDs have a maximum forward current of 20mA and the resistor helps in keeping the current flow to the LEDs constant.

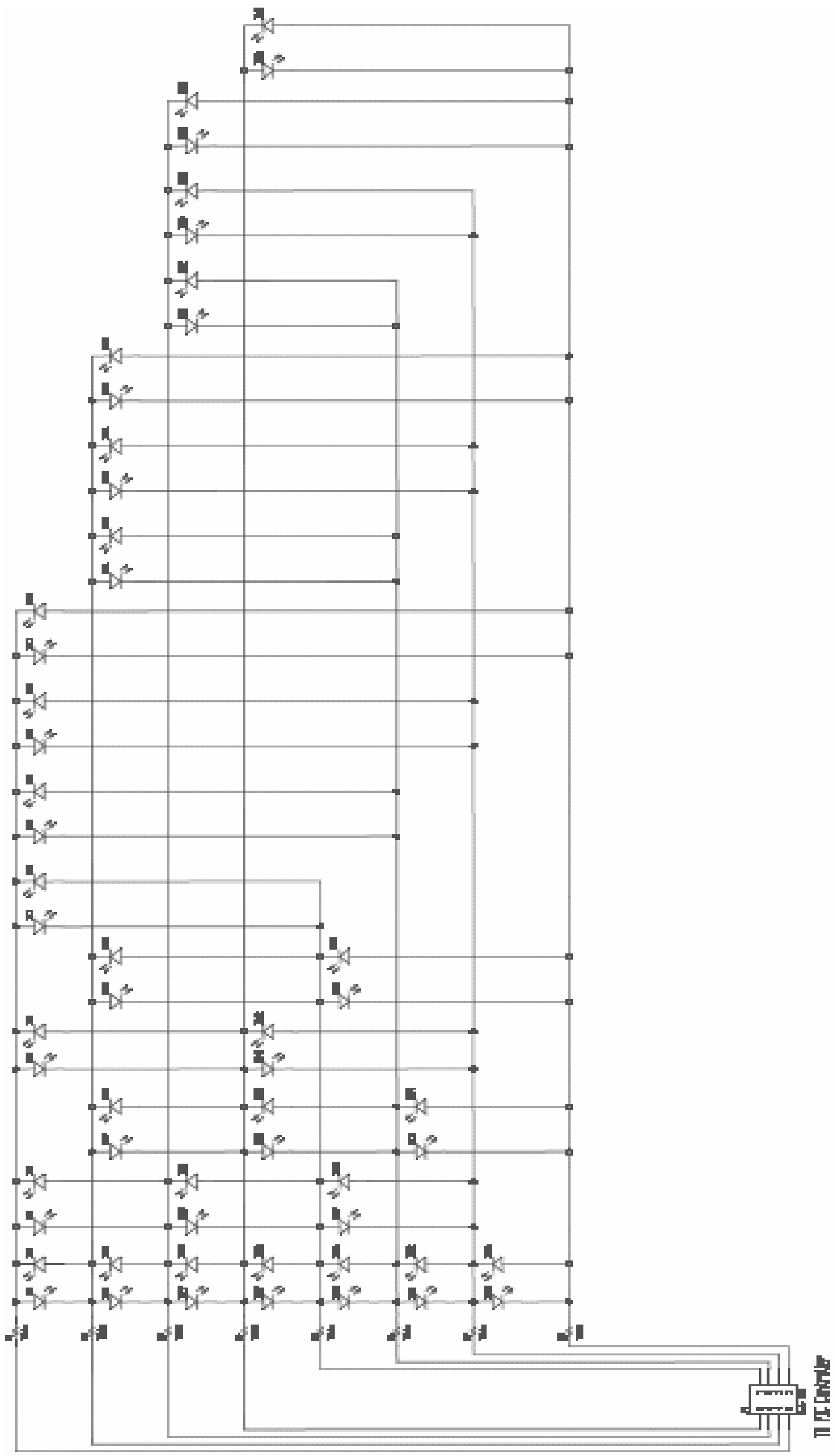

Figure 5.6: Circuit Diagram for the LEDs for Each Strip

5.9 Data collection and Parsing

The data collection for the forward display remains the same as it was described in the thesis presented by Douglas Burch. The data collection software is an interrupt driven C program that initializes the hardware and then performs the necessary collection and manipulation of the data strings sent from the GPS receiver and AHRS unit. The GPS receiver is set up to transmit its strings to the data collection computer at 115200 baud, no parity, 8 data bits, 1 stop bit, no handshaking, and echo off. The AHRS unit is set up with a data rate of 38400 baud, no parity, 8 data bits, 1 stop bit, no handshaking, and echo off. The GPS data and the AHRS data are collected on ports one and two of the data collection computer respectively. Communication ports three is configured to send the state vector to the synthetic vision computer while communication port four is configured to send the state vector to the Microcontroller computer [17].

Every time a new GPS position or velocity string is received the global variables in the C program are updated and the state-vector loaded accordingly. The state-vector is received by the synthetic vision computer via a DB-9/RS-232 link. The Microcontroller software is an object-oriented program written in Visual Basic© that processes the state-vector information. The program sets up the communication port for receiving the state-vector message from the data processing computer. It then initializes a set of global parameters necessary for the attitude and position of the aircraft.

It now monitors the communication port for an interrupt event. When an event is received the state-vector message is checked to ensure that all of the necessary data fields are present. If the message is intact, it is sent to the parsing algorithm, if not then it is ignored. Once the state-vector is parsed, a set of global variables is loaded with the state-vector information and the main routine of the program is notified that a valid state-vector has been received and processed. The main loop of the program contains a Boolean statement whose condition is set by the communication port interrupt. If the Boolean variable is true, then the main routine updates the global variables containing the attitude and position data, if it is false it assumes that the current state-vector information is valid [17].The state vector that is sent to the microcontroller computer has the format as shown in Table 5.2.

The visual basic program now parses out only the roll data and stores it as a separate local variable. Communication port 4 is now made available for communication with the microcontroller chip. The roll data is sent over the communication port to the microcontroller chip for further processing. The limit on the roll data is set to +/- 28 degrees. If the value of roll exceeds the limit then it is rounded off to +/- 28 degrees. The data is sent to the microcontroller at 20 Hz. The microcontroller is configured for high speed serial communication with a baud rate of 19.2 kilobaud.

Field	Parameter	Units
1	#FD	None
2	Time, GPS Seconds into the Week	Seconds
3	Local East with respect to ENU origin	Meters
4	Local North with respect to ENU origin	Meters
5	MSL Altitude	Meters
6	Ground Speed	Meters/second
7	Ground Track in degrees from North	Degrees
8	Pitch	Degrees
9	Roll	Degrees
10	*	None

Table 5.2 – State-Vector Fields, Parameters, and Units [17]

The Microcontroller program is set up such that three LEDs light up simultaneously. The main program is set to receive interrupts from the serial port. It sets up a register that takes in the value received over the serial port. This register is configured for one LED strip. Once the interrupt is received, the truth register is updated and the LED corresponding to the value in the register is lit up for that particular strip. The adjacent LEDs are lit up as well. Error checking is performed such that non-existent LEDs (LED 0 or LED 57) are not being lit up. A second truth register takes the complement of the value in the first register and correspondingly lights up the LEDs on the second strip. As long as no interrupt is received the current LEDs remain lit. When an interrupt is received, the truth registers are updated and the corresponding LEDs are lit again.

As an example, assume the first value that is sent serially to the microcontroller is 28 degrees. This is the input that is seen by the microcontroller and it sends this out to the LED strips. The LEDs that light up are 27, 28 and 29 on both the strips indicating straight and level position of the aircraft. This piece of code for the microcontroller was developed by Don Venable.

5.10 The Working of the LED Strips

The LED strips are so configured that only 3 LEDs light up at a time. The original architecture had the strips light up from the bottom to the point of roll on the aircraft. This was not possible due to the current limitations on the LEDs. For the LEDs to light up from the bottom the current would have exceeded the 20 mA current rating and lead to the LEDs burning out. Thus it was decided to test the strips with three LEDs lit up at one time. The LED strips were now built and ready for bench testing. The LED strips were tested on the bench at Stocker Center with data from previous flight tests. They were made to perform in conjunction with the forward display unit to see if the roll information as it appeared on the screen corroborated with the roll being displayed on the strips. This essentially means that the LEDs followed the horizon and emulated an artificial horizon inside the cockpit. The direction of the LEDs being lit up was exactly opposite to the direction of the wingtips of the aircraft. The system performed flawlessly on the bench test and was then installed on the Piper Saratoga.

6. Flight Testing the LED-Forward View Symbiotic Display

Having performed flawlessly in the bench tests, it was now important to see how the strips performed in actual flight. The important aspect to be tested was if it would provide the peripheral guidance that was necessary without causing any significant reflections or hindrance to the pilots. The reactions of the pilots would be a true measure of the usefulness of this system.

6.1 Installation of the LED Strips

The location of the earlier components for the system remained the same. The Microcontroller circuit was screwed onto the Aluminum board on which the GPS Receiver was placed. The LED strips were installed in the location chosen as can be seen in Figures 6.1 and 6.2. The installation was performed by Jay Clark and Paul Nilles at the Ohio University Airport.

Figure 6.1: Positioning of the LED Strips (Forward View)

Figure 6.2: Positioning of the LED Strips (Side View)

6.2 Subject Pilots for Flight Testing the LED – Forward View Symbiotic Display

The LED system was built keeping in mind that the average GA pilot flies the airplane as is needed and not necessarily on a regular basis. This would cover a wide range from student pilots to experienced pilots. Keeping this in mind the system was tested and flown with various pilots with different levels of flight time as well as ratings. There were six different pilots that were asked to test the system over six flight tests.

Each flight test was with a different pilot pair allowing for a comparison for the pilots reactions.

Subject pilot # 1 is a Certified Flight Instructor (CFI) with Aircraft Single-Engine, Multi-Engine, and Instrument Ratings with 900 hours of flight time.

Subject Pilot # 2 holds a Multi-Engine Commercial Pilot Certificate with Instrument Rating and is also a Certified Flight Instructor with 5800 hours of flight time.

Subject Pilot # 3 is a Certified Flight Instructor with 300 hours flight time.

Subject Pilot # 4 is an ATP (Airline Transport Pilot), CFI (Certified Flight Instructor), MEI (Multi-Engine Instructor), Ex-Military Pilot, and also has B200, DC3 and L29 Type Ratings, with 6000 hours flight time.

Subject Pilot # 5 has a VFR Single-Engine Land Rating with about 125 hours of flight time.

Subject Pilot # 6 is a Student Pilot with 53 hours of flight time.

6.3 Flight Tests

Each of the flight tests were conducted at near dusk time. This is due to the fact that the forward screen was not glare protected rendering it unreadable during day time. Near dusk flights allowed us to test the forward screens along with the LED system. Each flight test was video recorded to document the performance of the symbiotic system.

6.4 Flight Test One

The first flight test was conducted on June 16th, 2004 at 8.45 p.m. at the UNI Airport at Albany, Ohio. Subject Pilot # 1 was the primary pilot in the left seat and Subject Pilot # 2 was the safety pilot. The weather was clear (VFR). Several right-pattern approaches were made on UNI 7. This first test was conducted to test the system in actual flight. The primary goal of the flight test was to see if the LED strips did actually emulate the attitude of the aircraft especially when the aircraft was in a bank. All the LEDs that were mounted were green in color and LED numbers 27, 28 and 29 were lit up to indicate straight-and-level flight. The system performed flawlessly and the LED strips lit up according to the roll of the aircraft.

The LED strips were stuck on directly to the two side panes as shown in Figures 6.1 and 6.2. Due to this the LED strips were facing each other and not the pilots. This made is difficult the view the LEDs. It was also difficult to tell straight and level flight as all the LEDs were green in color and thus level flight was indistinguishable from a bank.

After twenty minutes into the flight test, the AHRS unit developed a bias and the test was terminated.

6.5 Modifications to the LED – Forward View Symbiotic Display

The AHRS unit was consequently bench tested and was found to have developed a random bias which did not repeat itself. After repeated testing wherein the AHRS unit performed flawlessly, it was reinstalled in the Saratoga.

Due to the fact that the strips were directly fastened onto the side panes on the aircraft, the viewing angle for the pilots was greatly reduced. Thus the LED strips were tilted such that they were now facing the pilots and not each other. This would hopefully lead to the LED strips to be more apparent in the peripheral vision of the pilot. The colors of the LEDs were also modified. The strips also needed an indication for straight and level flight. This was accomplished by having three amber LEDs at the center for straight-and-level and green LEDs for all other attitudes. With these modifications in place the second flight test was scheduled.

6.6 Flight Test Two

The second flight test was conducted on August 16th, 2004 at 20:10 hrs at UNI. The weather was clear (VFR). The pilots were Subject Pilot # 3 as the primary pilot in the left seat and Subject Pilot # 4 as the safety pilot. Several rectangular traffic patterns were flown on UNI 7. The flight path is as shown in Figure 6.3.

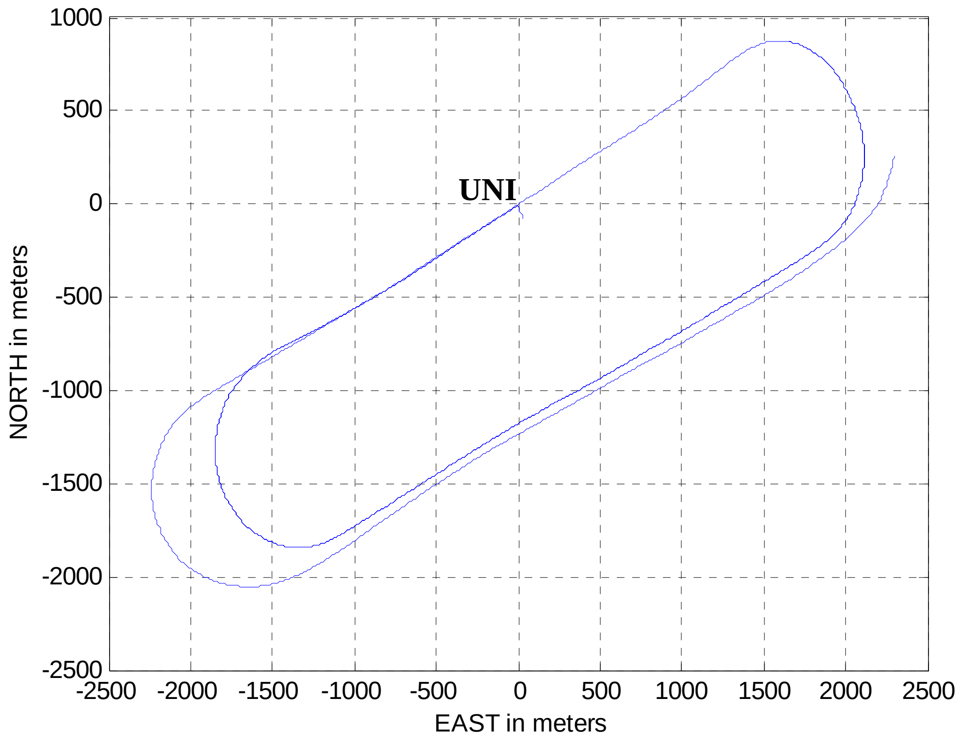

Figure 6.3: Flight Path of the Saratoga on August 16th, 2004 at 20:10 hrs

From an engineering perspective the LED display system performed flawlessly providing real-time attitude information and situational awareness with respect to the runway. Some of the observations that were made during the flight test were as follows:

The angled led strips led to better viewing by both the pilots. As the strips now faced the pilots they were shielded from direct glare as well and the LEDs appeared brighter rendering them more visible. Though the strips were now more visible it was still difficult to discern between the green and amber LEDs especially when the airplane transitioned from straight and level flight into a turn.

The strip on the right was not visible unless directly looked at by Subject Pilot # 3.

The forward screen had a few parameters such as the pitch, roll and height above mean sea level displayed on the top right corner. The parameters had a number of trailing decimal points displayed which led to them being unreadable and distracting.

The flight test proved that angled strips did lead to better viewing and thus increased peripheral guidance to the subject pilot.

6.7 Further Modifications to the LED – Forward View Symbiotic Display

The colors of the LEDs were further modified so that they would be more discernable during flight. The LEDs were changed to green for straight and level, amber for bank angles up to 25 degrees and red for maximum bank of 25 to 30 degrees.

The LED strips were maintained at the angled position as this provided maximum viewing to the pilots.

6.8 Flight Test Three

The third flight test was scheduled for August 23rd, 2004 at 20:00 hrs at UNI. The weather was clear (VFR). The pilots were Subject Pilot # 5 as the primary pilot in the left seat and Subject Pilot # 2 as the safety pilot. Due to unforeseen problems with the power supply the flight test had to be terminated and rescheduled for the next day.

6.9 Continued Testing

The system was flown three more times after the second successful flight test. The pilots that flew the system were asked to test the validity of the LED system as a primary source of peripheral guidance. Most of the test scenarios consisted of the safety pilot putting the airplane in an unusual attitude and the primary pilot using the LEDs to recover the plane. The Saratoga was flown from UNI to Athens on all three occasions. The unusual attitudes flown by the pilots can be seen in the meandering paths taken by the Saratoga in the following figures.

6.10 Flight Test Four

The fourth flight test was conducted on August 24th, 2004 at 20:00 hrs at UNI. The pilots were Subject Pilot # 5 as the primary pilot in the left seat and Subject Pilot # 2 as the safety pilot. The weather was clear (VFR). The flight path is as shown in Figure 6.4. It can be seen from the flight path that the pilot was asked to turn as often as possible as this would allow the evaluation of the LED strips as a guide to peripheral vision.

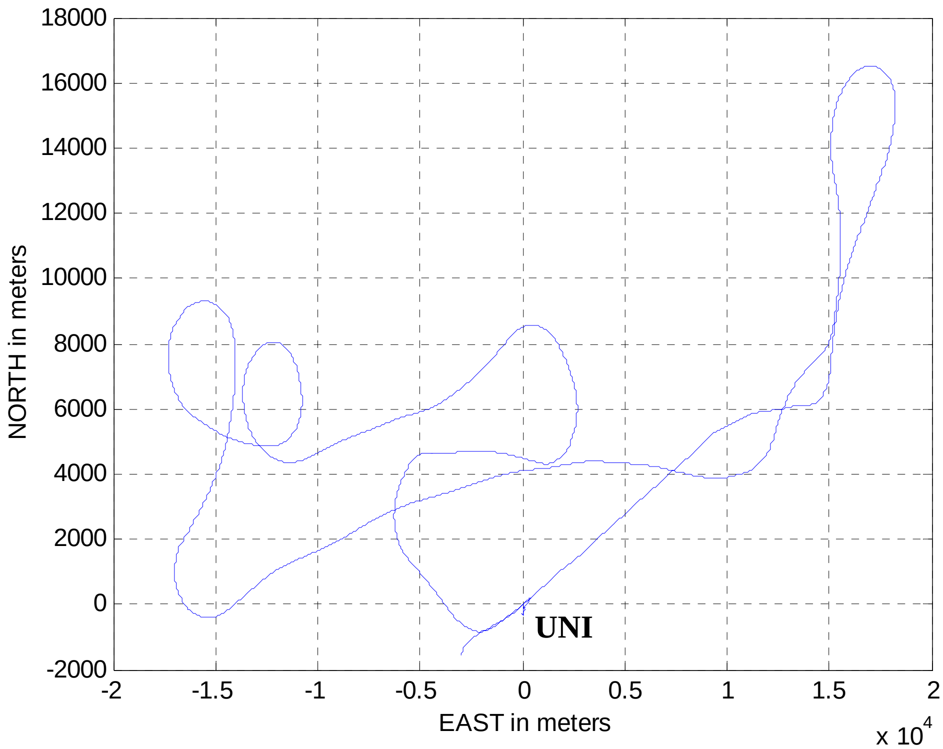

Figure 6.4: Flight Path of the Saratoga on August 24th, 2004 at 20:00 hrs

In this test, the pilot was asked to use the led strips for guidance when banking or turning. The pilot was asked to use the LEDs to determine the approximate extent of the turn. He was also asked to judge whether the LEDs did provide enough peripheral awareness so as to alert him of the attitude of the aircraft.

During the flight the LEDs provided accurate guidance to the pilot. The left strip was more intuitive as it fell directly in the peripheral field of vision. The behavior of the strips akin to the artificial horizon led to better peripheral perception of the aircrafts attitude.

The colors selected also provided a more intuitive understanding of the relative bank of the aircraft body.

A major disadvantage that was encountered during the test was that the right LED strip fell directly on the edge of the frame of the glasses worn by the pilot, rendering it useless. The test was terminated after a few successful turns were guided primarily by the LED strips.

6.11 Flight Test Five

The fifth flight test was conducted on August 25th, 2004 at 20:00 hrs at UNI. The pilots were Subject Pilot # 1 as the primary pilot in the left seat and Subject Pilot # 2 as the safety pilot. The weather was clear (VFR). The flight path is as shown in Figure 6.5. As it can be seen again from the flight path, the pilot made an increased number of turns to allow the evaluation of the LED strips as a guide to peripheral vision.

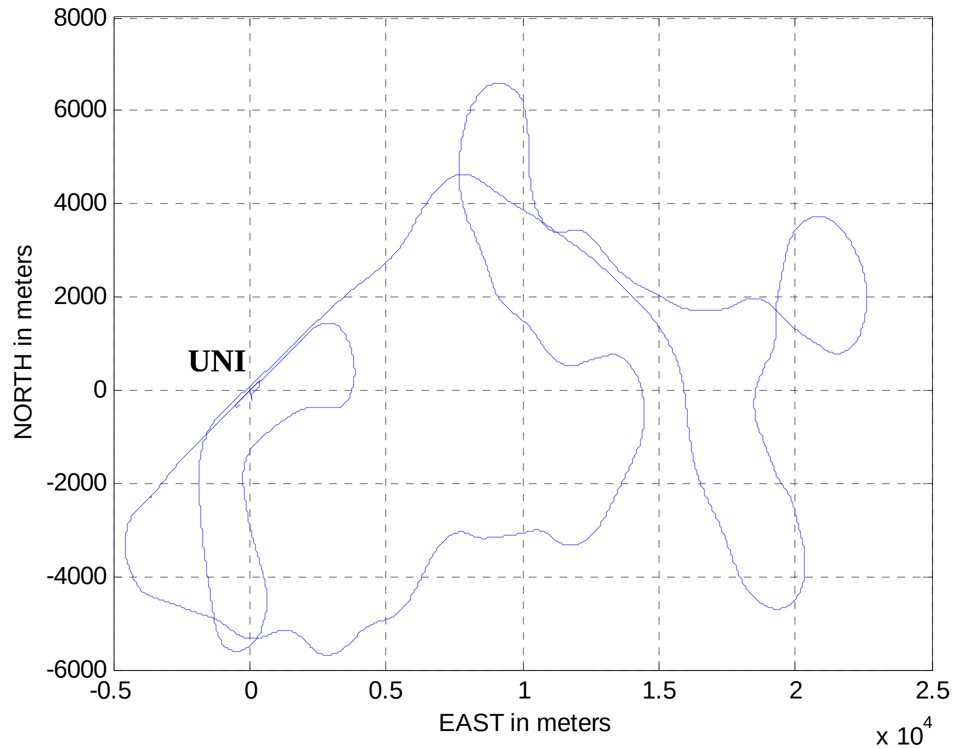

Figure 6.5: Flight Path of the Saratoga on August 25th, 2004 at 20:00 hrs

This flight test was conducted to test the performance of the system is unusual attitudes. The safety pilot would take over flying the Saratoga while the primary pilot looked down. Once the primary pilot was unaware of the attitude of the aircraft he was asked to take over and regain straight and level flight with the help of the LED strips. This was practiced a number of times to judge if consistent guidance could be provided by the LED strips to the pilot.

This pair of pilots was the same that had tested the system on the first flight test. It was immediately apparent to them that the angled placement of the strips provided a greatly improved viewing angle and also reduced the glare that was caused by the LED light bouncing off the window.

The LED strips performed flawlessly and provided accurate roll information every time. Midway during the flight, Subject Pilot # 1 suggested that the inputs to the LEDs be changed such that they now emulated the wings of the aircraft. Once this was done it seemed to him that this provided better guidance to him as it told him immediately what the aircraft was doing. It was easier to comprehend as it followed the attitude indicator on board. It was also easier to see the transition between the colors of the LEDs from green to amber to red as the plane increasingly banked.

6.12 Flight Test Six

The sixth flight test was conducted on August 30th, 2004 at 20:00 hrs at UNI. The pilots were Subject Pilot # 6 in the left seat and Subject Pilot # 4 as the safety pilot. The weather was clear (VFR). Due to certain problems encountered during the data collection it was not possible to graph the flight path of the test. In this test as well, the pilot was asked to test the system in unusual attitudes as well as turns.

As before the primary pilot was asked to look down as the safety pilot put the aircraft in an unusual attitude. This time the pilot wore ‘foggles’ so that he could see nothing but the instruments in front of him and the LED strips. He was then asked to get the aircraft back to straight and level using the LEDs. The pilot found it relatively easy to do so as he looked for the green LEDs to light up telling him that the aircraft was straight and level. The pilot preferred the LED strips to emulate the artificial horizon rather than the body (wings) of the aircraft. With the foggles on, the strips were not clearly visible in the peripheral field of vision but without them both the strips were clearly visible to the pilot.

During the turns, the amber LEDs would light up giving him an indication of the approximate bank angle. If the bank angle exceeded 30 degrees the red LEDS would light up, indicating that the aircraft had exceeded the maximum limit. This would help him reduce the angle to come back to the amber zone.

6.13 Pilot Briefings

After the flight tests were concluded, a briefing with the pilots was held. This allowed the pilots to discuss their reactions as well as view their suggestions about the system. Some of the initial observations that were made by the pilots are as follows:

To begin with the pilots felt that the LED strips succeeded in providing adequate peripheral guidance while flying. The positioning of the strips however, was a problem as the right strip was not in the peripheral field and had to be directly looked at for guidance.

This was not the case when subject pilot # 6 flew the system leading us to believe that the height and seating of the pilot made a difference to the viewing angle of the strips.

The pilots with less than 150 hours of flight felt that the LED strips when acting as an artificial horizon was more intuitive to follow. A comment to this effect was made wherein the pilot said that with the increased use of flight simulators and video games it was easier to comprehend and thus follow an artificial horizon. This was refuted by the pilots with greater experience in flight. They believed that the LED strips when following the wing tips was easier to comprehend as it emulated the attitude indicator on board.

The transition from green to amber when the plane was in a bank was easily discernable. Due to the low luminous intensity LEDs there was no bright light shining into the eyes of the pilots. The LEDs did not flash or cause any reflections inside the canopy granting it a greater edge over the PVHD and the artificial horizon. A few suggestions were also made that might be useful in enhancing the system:

It was suggested to have two pairs of LED strips for each of the pilots. This would bring the right strips closer in the peripheral filed of vision and provide the same viewing angle for both the pilots regardless of their seating. The current placement of the right strip made it difficult for the pilot in the left seat to view it and vice versa for the pilot in the right seat for the left strip.

Cabin lights are red. It was suggested to change the LED color for the maximum bank to something other than red as they would be washed out when the cabin lights are turned on.

The scale on the LEDs is currently at 1 LED to 1 degree of roll. Changing the scale with such that it is more sensitive to greater bank angles would prove the system more useful in highly unusual attitudes.

The LED strips are built such that the middle three LEDs (LED number 27, 28 and 29) are green followed by amber LEDs on both sides and then the last five LEDs at the top and the bottom are red. The amber LEDs when lit did not always tell the pilot the direction of the bank. Due to the placement of the strips it was not possible for the pilot to visualize a straight line between the strips. Hence when the amber LEDs were lit and if the pilot looked at them momentarily, it was not evident that the aircraft was banked left or right. The pilot could not tell if the top or the bottom row of amber LEDs on the left strip were lit thus not giving a clear indication of the direction of bank. It was thus suggested to have different colors on the two strips. For example, amber for left bank and blue for right bank. This would immediately tell the pilot the direction of the bank and if a correction was necessary.

It was also thought that the LED strips as either an artificial horizon or following the wing tips would provide the necessary guidance with proper training.

In conclusion each of the pilots felt that the concept of the LED-Forward view symbiotic system held great merit. In the testing phase itself it proved as a useful secondary instrument and provided the necessary visual cues that were necessary to maintain orientation. It was also useful in regaining orientation when recovering from an unusual attitude situation.

7. Conclusions and Future Work

As the dependence on GA increases for various applications, the need to make the GA cockpit safer grows with it. Instrumentation in the GA cockpit has undergone little change over the last 50 years as compared to commercial airlines having access to the latest in technology for increased safety. There has to be a gradual but sure change wherein the GA cockpit evolves to incorporate the latest in technology in a cost effective manner.

Synthetic vision as a safety feature is in itself not a new concept but its application to the GA world is still in its infancy. Though still rudimentary in application, the study of various types of displays for the GA cockpit is the first step in making the skies safer for GA pilots.

For any pilot, take offs and landings are the most crucial phases of flight. It is in this phase that the pilot intuitively finds himself looking out of the side window for a visual reference rather than out of the front. This natural reaction led us to investigate further into the realm of peripheral displays.

Studies have shown that peripheral vision is essential for a human being to maintain spatial orientation. Experiments have shown that a person can no longer navigate his movements once his peripheral vision is blocked. This proves that as a secondary instrument, a peripheral display can be very useful if used with proper training and guidance.

This research has tried to document the early efforts made to develop a working model of a peripheral display, the PVHD. Though unique in concept it was plagued with technical difficulties such as reflections from the canopy, misalignment with the instrument panel, dimness of the light source rendering it useless in daylight etcetera. Though the PVHD had its share of issues, the concept was not without merit and led us to investigate other types of peripheral devices.

The first device developed and tested in this research was the general aviation synthetic vision display. It consisted of a forward display that was previously tested and also a side display that provided the pilot with a virtual side-view of the outside world. The display worked flawlessly in conjunction with the forward display and provided the pilot with the necessary orientation information that was needed to maintain straight and level flight. Though meritorious in its concept, the side display had certain installation issues due to the space constraints inside the aircraft. The side display would prove to be useful in an aircraft where there would be more flexibility in its placement within the cockpit.

The central theme of this research was the development and testing of the LED – Forward view symbiotic display. The concept of using LED strips proved relatively simple in development as well as in installation in the aircraft.

The flight testing that ensued with the symbiotic system proved that the use of the LED strips was definitely a step in the right direction. The strips were used by various pilots of varying degrees of experience. It was generally agreed upon by all the pilots that the system in its initial phase did hold merit in its concept. Various suggestions were made for its improved implementation that constitutes the future work of this research.

The LED strips are currently in a testing phase. They are meant to be used as a secondary instrument for peripheral visual reference. The system is not meant to replace any of the instruments that already exist in the cockpit. It is not meant to be used as a primary means of visual reference. The guidance provided by the system is meant to augment the information that the pilot is already getting from his instrument panel. The symbiotic system is meant to provide a visual warning signal to the pilot in case his senses are telling him otherwise. The favorable feedbacks from the pilots that have flown the system prompt further investigation into the system.

Flight testing of the LED- Forward view symbiotic system has led us to believe that the system needs to be tested more such that it can be evaluated for its validity in the GA market. Future work includes the use of the smaller LEDs – T1 with 3mm diameter. These LEDs would allow the use of smaller strips. The installation of the smaller strips can be made closer in the peripheral field of view without any interference with the instruments that are already in the instrument panel.

The side displays also are in need of further investigation. With simple modifications the system can have three displays – left, forward and right. With proper placement the system could provide a panoramic virtual view of the outside world.

Another aspect of this research is the study of helmet mounted displays. Though used only in military aircraft, with increasing development the displays studied here could in the future, be incorporated into a helmet for more realistic viewing.

This research provides an insight into the need for peripheral displays and also explores the various concepts that can be used for its implementation. The goal of this research is not to replace the existing instruments. It is to develop a warning system that can alert the pilots to the dangers of spatial disorientation and in effect ameliorate its influence on the safety of GA in general.

References

[1] 2003 Nall Report, AOPA Air Safety Foundation. “General Aviation Accident Trends and Factors for 2002”. February 2004.
http://www.aopa.org/asf/publications/03nall.pdf

[2] 1999 Nall Report, AOPA Air Safety Foundation. “General Aviation Accident Trends and Factors for 1998”. August 2001.
http://www.aopa.org/asf/publications/99nall.pdf

[3] Snow, M.P., “Head Up Synthetic Vision Displays For Prevention Of Spatial Disorientation”, Air Force Research Laboratory. November 2002.
http://www.spatiald.wpafb.af.mil/RecentTrends.aspx

[4] Spatial Disorientation Safety Advisor, AOPA Air Safety Foundation.
http://www.aopa.org/asf/publications/sa17.pdf

[5] 2002 Nall Report, AOPA Air Safety Foundation. “General Aviation Accident Trends and Factors for 2001”.October 2002.
http://www.aopa.org/asf/publications/02nall.pdf

[6] Spatial Disorientation Countermeasures, Air Force Research Lab Site.
http://www.spatiald.wpafb.af.mil/overview.aspx

[7] McKinley, John, “Making Flight Performance Safer Through Automated Intelligence”, Presentation at the 2004 EAA Airventure, Oshkosh, WI, July 30th, 2004.

[8] Malcolm, R, *The Malcolm Horizon – History and Future*. Proceedings of NASA Conference on Peripheral Vision Horizon Display (PVHD) – Corrected Copy (NASA CP -2306) (pp. 11-40) Edwards AFB, CA: Dryden Research Facility.

[9] Leibowitz, H.W., Shupert, C.L., Post, R.B., *The Two Modes Of Visual Processing: Implications For Spatial Orientation.* Proceedings of NASA Conference on Peripheral Vision Horizon Display (PVHD) – Corrected Copy (NASA CP -2306) (pp. 41-44) Edwards AFB, CA: Dryden Research Facility.

[10] Money, K.E., *Theory Underlying The Peripheral Vision Horizon Device.* Proceedings of NASA Conference on Peripheral Vision Horizon Display (PVHD) – Corrected Copy (NASA CP -2306) (pp. 45-55) Edwards AFB, CA: Dryden Research Facility.

[11] Hameluck, D., and Stager, P. (1987b). *The Peripheral Vision Horizon Display: A Review.* In R.S. Jensen (Ed.), Proceedings of the Fourth International Symposium on

Aviation Psychology, (pp 51-57). Columbus, OH: The Ohio State University, Department of Aviation.

[12] McNaughton, G.B. (1983*). Personal experience with the PVHD and opinion of situations in which a wide filed of view might be helpful*. Proceedings of NASA Conference on Peripheral Vision Horizon Display (PVHD) – Corrected Copy (NASA CP -2306) (pp. 1-9) Edwards AFB, CA: Dryden Research Facility.

[13] Hameluck, D., and Stager, P. (1987a*). Instrument Scanning and Subjective Workload with the Peripheral Vision Horizon Display*. Report prepared for the Defense and Civil Institute of Environmental Medicine, Toronto, under Contract no. W7711-6-9116/01-SE from the Department of Supply and Services.

[14] Nims, D.F. (1983). *Peripheral Vision Horizon Display on the Single seat Night Attack A-10.* Proceedings of NASA Conference on Peripheral Vision Horizon Display (PVHD) – Corrected Copy (NASA CP -2306) (pp. 89-96) Edwards AFB, CA: Dryden Research Facility.

[15] Hammond, L.B., Jr. (1983). *Peripheral Vision Horizon Display testing in an RF-4C aircraft.* Proceedings of NASA Conference on Peripheral Vision Horizon Display (PVHD) – Corrected Copy (NASA CP -2306) (pp. 97-102) Edwards AFB, CA: Dryden Research Facility

[16] Comstock. Jr, J.R., Jones, L.C., and Pope, A.T., "The Effectiveness of Various Indicator Display Sizes and Extended Horizon Lines on Attitude Maintenance in a part task simulation". NASA Langley Research Center, Hampton, VA and Virginia Tech, Blacksburg, VA.
http://techreports.larc.nasa.gov/ltrs/PDF/2003/mtg/NASA-2003-47hfes-jrc.pdf

[17] Burch, D., "Implementation and Evaluation of a General Aviation Synthetic Vision Display System", Masters Thesis, Fritz J. and Dolores H. Russ
College of Engineering and Technology, Ohio University, March 2004.

[18] Crossbow Technology, Inc. AHRS400 Series User's Manual, 41 E. Daggett Dr., San Jose, CA 95134, http://www.xbow.com

[19] Revolution 3-D Home Page, http://www.revolution3d.net, September 2002

[20] Chakrabarty, J., Burch, D., Kalyanaraman, S., Braasch M., *Multi – View Synthetic Vision Display System for General Aviation*, Proceedings of the 2004 Institute of Electrical and Electronics Engineers (IEEE) Aerospace Conference, Big Sky, MT, March 2004.

[21] Policy and Procedures for Flight Operations, Avionics Engineering Centre, Ohio University, May, 2003.

[22] Color Vision http://www.photo.net/photo/edscott/vis00010.htm

[23] Mouser Electronics
www.mouser.com
http://www.kingbright-led.com/data/spec/L53SRD-D.pdf
http://www.mouser.com/catalog/specsheets/0935131.pdf
http://www.mouser.com/catalog/620/49.pdf

Appendix A: Hazard Action Matrix

The Hazard Action Matrix (HAM) identifies potential hazards then determines the hazard severity and probability. The hazard severity had four categories.

Category 1 is loss of life/crew,

Category 2 is loss of vehicle,

Category 3 is system degraded/loss of mission, and

Category 4 is safe operation.

The hazard probability is categorized into:

A – likely to happen frequently,

B – likely to happen occasionally,

C – likely to happen once,

D – unlikely,

E – extremely improbable.

Based on these categories, Loss of life during a data collection flight must be either unlikely to happen (D) or extremely improbable (E). Categories A, B, and C are not allowed. Loss of vehicle during a data collection flight must be either likely once (C), unlikely (D), or extremely improbable. Categories A and B are not allowed. A hazard's severity or probability may be reduced through Alleviating Actions which mitigate the risk. For example, the probability of a pedestrian being hit by a prop strike can be reduced by having an observer on the ground [4].

A sample HAM is shown.

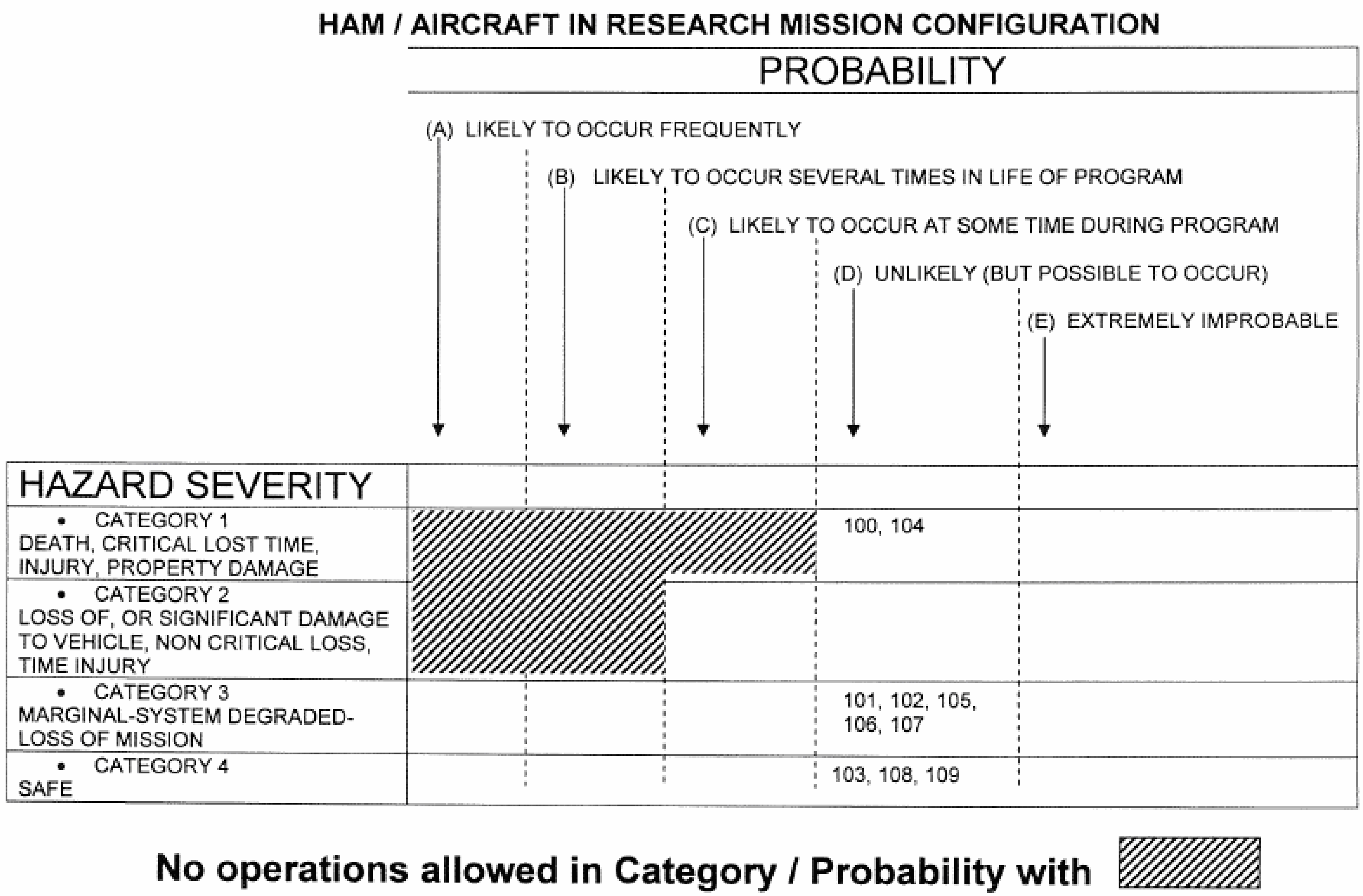

Figure A.1: HAM for Research Mission Aircraft

FEMA / ALL AEC RESEARCH MISSION AIRCRAFT - EXAMPLE

Hazard #	Severity	Likely-hood	System	Failure Mode	Indication	Action when Hazard Occurs	Alleviating Action
100	1	D	Research rack power supply	Electrical fire	Fumes/ smoke/ fire in lower rack shelf area and aircraft cabin	Control aircraft, pull rack power supply circuit breaker, don O2 masks as needed	Thorough scheduled, pre and post-flight inspections
101	3	D	Research equipment	Research rack equipment fails	Loss of data, loss of computer display, erroneous data collected	Research equipment operator informs pilot and commences troubleshooting, and/ or return to base	Thorough scheduled, pre and post-flight inspections of research equipment, diligent use of design checklist
102	3	D	Research equipment external antenna	Antenna power failure	Loss/ lack of data collection	Research equipment operator informs pilot and commences troubleshooting, and/ or return to base	Thorough scheduled, pre and post-flight inspections of research equipment, diligent use of design checklist
103							
104							
105							
106							
107							
108							

Figure A.2: Example of AEC Research Mission HAM

Appendix B: Synthetic Vision Code

The Synthetic Vision code communicates with the data collection computer and sends an interrupt to the microcontroller as soon as it receives a new state vector.

There are two versions of the main program. The first is the modified forward view display program such that it now depicts the right side view of the outside world. The second is the code used to isolate the roll information and send it to the microcontroller for it to be displayed on the LED strips. The comments in the program allow the reader to understand the flow of logic. Except for the main program, the remaining synthetic vision code remains the same for both the types of displays.

The original code was written by Douglas Burch.

B.1 Main Program for the Right Side Display

```
'* Author:  Douglas Burch
'* Date:    15 Jan 2002
'* MOdified by Jahnavi Chakrabarty
'* Date:    June 2003
'* Purpose:
'*  Initalizes COMM1 port (RS-232) to 9600 bps then sets the Novatel OEM-4 GPS
'*  receiver's baud rate to 115200 bps.
'*      115200 bps
'*      No Parity
'*      8 Data Bits
'*      1 Stop Bits
'*      No Handshaking
'*      Echo Off
'*  Comm1 is opened and all previous commands for the NovAtel OEM-4 RCVR PWRPAK-4
'*  3151R are unlogged.
'*  The BESTPOSA and BESTVELA strings are requested, each at 20 Hz.
'*  A Comm Event is used so that the input buffer is only read when a comm event
'*  occurs indicating a precise amount of data. Once the buffer is read the
'*  string is sorted according to type.
'*  There are two different string headers, BESTPOSA and BESTVELA. This program
'*  looks for the string header and then processes the string accordingly. The
'*  Latitude, Longitude, and Height (LLH) are parsed from the BESTPOSA string and
'*  the Horizontal Speed, Vertical Speed, and Ground Track are parsed from the
'*  BESTVELA string.
'*  The time stamp is also parsed from each string and used to correlate the velocity
```

```
'*  to a given position. Once the strings have been matched the velocity vector is
'*  calculated and then the acceleration and its components.
'*  The acceleration is filtered and the gravitational components are calcualted.
'*  Once this is complete a reference vector is calcualted. Once all the needed
'*  information is obtained the Pseudo-Roll (PSR) and Flight Path Angle (FPA) are
'*  Calculated and dumped to the file "fltdata".
'*  There are a total of eight flight parameters that are generated by this program:
'*      Time Stamp
'*      Latitude
'*      Longitude
'*      Height
'*      Velcity
'*      Ground Track
'*      Flight Path Angle
'*      Pseudo-Roll
Option Explicit 'Avoid Undifined Variables.
'*****************************************************************
'* API Call Declarations.
'*****************************************************************
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
'//declare refenrences to Revolution3D
Public Engine       As New R3D_Engine
Public Pipeline     As New R3D_Pipeline
Public Control      As New R3D_Control
Public Interface2D  As New R3D_Interface2D
Public MeshBuilder  As New R3D_MeshBuilder
Public MaterialLib  As New R3D_MaterialLib
Public TextureLib   As New R3D_TextureLib
Public PowerMonitor As New R3D_PowerMonitor
Public Tools        As New R3D_Tools
Public Camera       As New R3D_Camera
Public SkyDome      As New R3D_SkyDome
Public PolyVox2     As New R3D_PolyVox2
Public Helper       As New R3D_Helper
Public BSPSystem    As New R3D_BSPSystem
Private Sub cmd_Exit_Click()
   Unload Me
End Sub
Private Sub Form_Load()
   '*** SET UP IMAGE TO BE DISPLAYED. ***
   Call SetUpGameEngine
   Call SetUpPipeLine
   Call CreateSky
   Call CreateTerrain
   Call CreateRunway
   '*** SET THE STARTING POSITION FOR THE CAMERA ***
   Call SetPosition
   '*** Create the PAPI and set it beside the runway ***
   Call CreatePAPI_Light_1
   Call CreatePAPI_Light_2
   Call CreatePAPI_Light_3
   Call CreatePAPI_Light_4
      '*** SET THE PIXEL TO METER RATIO ***
```

```
    snglPixel_Per_Meter = 0.1
    '*** MAIN LOOP ***
    '// While condition set to true.
    blnRUN = True
      '// Timer
    snglStartTime = Timer
    snglDelayTime = dt '0.05 = 20Hz
    '*** INITIALIZE THE GPS RECEIVER ***
    Call Initialize_NovAtel
     While (Not blnRUN = False)

       '//receive the pressed keys
       Control.Keyboard_ReceiveKeys
           '//check if the user has pressed and exit if true
       If (Control.Keyboard_GetKeyState(R3DKEY_END) = True) Then blnRUN = False
          '***** SET POSITION, ALTITUDE, AND ATTITUDE *****
       If (blnAttitudeCalculated) Then
           blnAttitudeCalculated = False
           '***** Convert Doubles to type Single. *****
          Call ConvertFlightDataToSingle
          '***** SET CAMERA ATTITUDE. *****
                'If (snglSpeed >= 10) Then
               '*** Pseudo-Roll ***
             snglCurrentPseudoRoll = snglPseudoRoll - snglPreviousPseudoRoll
             snglPreviousPseudoRoll = snglPseudoRoll
             'Camera.RotateZ (-snglCurrentPseudoRoll) 'Camera Roll is in different coordinate frame.
             Camera.RotateZ (-snglCurrentFlightPathAngle) 'Camera Roll is in different coordinate frame.
              '*** Flight Path Angle ***
             snglCurrentFlightPathAngle = snglFlightPathAngle - snglPreviousFlightPathAngle
             snglPreviousFlightPathAngle = snglFlightPathAngle
             'Camera.RotateX (-snglCurrentFlightPathAngle) 'Camera Pitch is in different coordinate frame.
             Camera.RotateX (snglCurrentPseudoRoll) 'Camera Pitch is in different coordinate frame.
           'Else
             ' Camera.RotateZ (0) 'Camera Roll is in different coordinate frame.
             'Camera.RotateX (0) 'Camera Pitch is in different coordinate frame.
          'End If
           '***** Altitude and Position Setting. *****
          '*** Calcualte Altitude Correction of Camera. ***
           '//Get the height (y) on the polyvox terrain at the position the camera currently is...
          Camera.GetPosition snglPosition_X, snglPosition_Y, snglPosition_Z
          snglTerrainHeight = PolyVox2.Scape_GetAltitude(Helper.R3DPoint2D(snglPosition_Z,
snglPosition_Z))
          Camera.SetPosition snglAircraft_X, snglAircraft_Y, snglAircraft_Z
          '***** Set the Heading. *****
          snglGroundTrack = snglGroundTrack + 90
          snglCurrentGroundTrack = snglGroundTrack - snglPreviousGroundTrack
          snglPreviousGroundTrack = snglGroundTrack
           Camera.RotateY (snglCurrentGroundTrack)
      End If
        DoEvents
        '***** GET ATTITUDE AND ALTITUDE INFORMATION FROM THE CAMERA. *****
        '// Get Roll, Pitch, and Heading.
       Camera.GetRotation snglPitch, snglHeading, snglRoll
```

```
        '//get the height (y) on the polyvox terrain at the position the camera currently is...
        Camera.GetPosition snglPosition_X, snglPosition_Y, snglPosition_Z
        snglTerrainHeight = PolyVox2.Scape_GetAltitude(Helper.R3DPoint2D(snglPosition_Z,
snglPosition_Z))
        '/*** CALCULATE THE GLIDE SLOPE. ***/
        dblGlideSlope = CalculateGlideSlope(snglPosition_X, snglPosition_Y, snglPosition_Z)
        '/**** PAPI CONTROL ****/
        Call SetPAPI(dblGlideSlope)
        '***** SCREEN CLEAN UP *****
        '// Clear the screen of previous data.
        Pipeline.Renderer_Clear

        '// Render the scene, Display the terrain.
        Pipeline.Renderer_Render
        ' Sent Text to the Screen
        Interface2D.Primitive_SetDrawColor 0, 0, 0
        Interface2D.Primitive_DrawText 1, 1, "Roll: " + CStr(snglPseudoRoll)
        Interface2D.Primitive_DrawText 1, 15, "Pitch: " + CStr(snglFlightPathAngle)
        Interface2D.Primitive_DrawText 1, 30, "GTrack: " + CStr(snglGroundTrack)
        Interface2D.Primitive_DrawText 1, 45, "GPS Time: " + CStr(dblTimeStamp)
        Interface2D.Primitive_DrawText 1, 60, "MSL Altitude(ft): " + CStr((snglUp + 231) * 3.2808)
        '// Display Text and Camera Position.
        Pipeline.Renderer_Display
    Wend
    '//terminate the engine and call vb's end
    Engine.TerminateMe
    End
End Sub
Public Function CalculateGlideSlope(X As Single, Y As Single, Z As Single) As Double
'* X is the Local X
'* Y is the Local Height
'* Z is the Local Z
Dim PI As Double

    PI = 4 * Atn(1)

Dim dblILS_X As Double      'ILS Local X Position
Dim dblILS_Z As Double      'ILS Local Y Position
Dim dblAircraft_X As Double 'Aircraft Local X Position
'Dim dblAircraft_Y As Double 'Aircraft Local Altitude
Dim dblAircraft_Z As Double 'Aircraft Local Y Position
Dim dblHeight As Double
Dim dblDelta_X As Double
Dim dblDelta_Z As Double
Dim Range_to_ILS As Double

    '*** THE GLIDE SLOPE IS CALCULATED BASED ON THE RANGE FROM THE AIRCRAFT TO
    '    THE CENTER OF THE THE RUNWAY, 1000 FEET FROM THE THRESHOLD. THIS IS
    '    PEFORMED IN THE LOCAL COORDINATE FRAME BASED ON PIXELS. IT WOULD BE
    '    LOGICAL TO THINK THAT SOME OF THE RESOLUTION IN GLIDESLOPE WOULD BE
LOST
    '    DUE TO THE FACT THAT THE CALCULATION IS BASED ON PIXEL COUNT AND NOT
    '    ACTUAL X,Y,Z RELATIVE TO THE ILS POINT. THIS IS NOT THE CASE DUE TO THE
```

```
    '   FACT THAT THE AIRCRAFT CAMERA IS MOVED IN PARTIAL PIXEL INCREMENTS.

    dblAircraft_X = X
    dblHeight = Y
    dblAircraft_Z = Z

    'dblILS_X = 486.024     'OLD
    'dblILS_Z = 497.003     'OLD

    dblILS_X = 503.669
    dblILS_Z = 507.19

    dblDelta_X = dblILS_X - dblAircraft_X
    dblDelta_Z = dblILS_Z - dblAircraft_Z

    Range_to_ILS = Sqr(dblDelta_X * dblDelta_X + dblDelta_Z * dblDelta_Z)

    dblGlideSlope = Atn(dblHeight / Range_to_ILS) * (180 / PI)

    CalculateGlideSlope = dblGlideSlope

End Function
Public Sub SetPAPI(dblGlideSlope As Double)

    '// TRUE = Red
    '// FALSE = White
    '*** VERY HIGH (W)(W)(W)(W) ***
    If (dblGlideSlope > 3.5) Then

        Call SetPAPI_1(False)   'WHITE
        Call SetPAPI_2(False)   'WHITE
        Call SetPAPI_3(False)   'WHITE
        Call SetPAPI_4(False)   'WHITE

    '*** HIGH (W)(W)(W)(R) ***
    ElseIf (dblGlideSlope > 3.25 And dblGlideSlope <= 3.5) Then

        Call SetPAPI_1(False)   'WHITE
        Call SetPAPI_2(False)   'WHITE
        Call SetPAPI_3(False)   'WHITE
        Call SetPAPI_4(True)    'RED

    '*** ON GLIDE SLOPE (W)(W)(R)(R)
    ElseIf (dblGlideSlope >= 2.75 And dblGlideSlope <= 3.25) Then

        Call SetPAPI_1(False)   'WHITE
        Call SetPAPI_2(False)   'WHITE
        Call SetPAPI_3(True)    'RED
        Call SetPAPI_4(True)    'RED

    '*** LOW (W)(R)(R)(R) ***
    ElseIf (dblGlideSlope >= 2.5 And dblGlideSlope < 2.75) Then
```

```
      Call SetPAPI_1(False)   'WHITE
      Call SetPAPI_2(True)    'RED
      Call SetPAPI_3(True)    'RED
      Call SetPAPI_4(True)    'RED

   '*** VERY LOW (R)(R)(R)(R) ***
   ElseIf (dblGlideSlope < 2.5) Then

      Call SetPAPI_1(True)   'RED
      Call SetPAPI_2(True)    'RED
      Call SetPAPI_3(True)    'RED
      Call SetPAPI_4(True)    'RED

   End If
End Sub

Private Sub Form_Unload(Cancel As Integer)

   If (frmNovAtel.SerialComm1.PortOpen = True) Then
      frmNovAtel.SerialComm1.PortOpen = False
   End If
   End Sub

Property Get AttitudeCalculated() As Boolean

   AttitudeCalculated = blnAttitudeCalculated

End Property

Property Let AttitudeCalculated(blnAttitudeCalculatedIn As Boolean)

   blnAttitudeCalculated = blnAttitudeCalculatedIn

End Property

Public Sub FlightDataToFile()

   Write #intFLTDATAHandler, dblPOS_TimeStamp, _
                     dblEast, _
                     dblNorth, _
                     dblUp, _
                     dblVelocityMagnitude, _
                     dblGroundTrack, _
                     dblFlightPathAngleDeg, _
                     dblPseudoRollDeg

End Sub

Public Sub SetUpGameEngine()

   '// The first thing that needs to be done is to set up the parameters
   '// of the game engine. For the flight display we want a full screen
   '// view. The field of veiw is set to 50 degrees and the view distance
```

```
    '// is set to 350.
    '// By setting the .InitializeMe property to True the start up menu is
    '// ignored and the program goes directly to execution. Had the property
    '// been set to False then a start up prompt would have appeared.

    With Engine
        .Inf_SetFieldOfView 45
        .Inf_SetViewDistance 5000
        .Inf_SetNearClippingPlane 0
        '.Inf_SetRenderTarget frmEHUD.hWnd, R3DRENDERTARGET_FULLSCREEN
        .Inf_SetRenderTarget frmEHUD.hWnd, R3DRENDERTARGET_WINDOW
        '.Inf_SetProjectionType R3DPROJECTIONTYPE_PERSPECTIVE
        .Inf_ForceResolution 1024, 1024, 10
        If .InitializeMe(True) = -1 Then End
    End With

    'Position the Window and set the size.
    Me.Width = (Screen.Width - (Screen.Width / 4))
    Me.Height = (Screen.Height - (Screen.Height / 3))

    'Place window on screen.
    Me.Left = 0
    Me.Top = 0

End Sub

Public Sub SetUpPipeLine()

    '// The pipeline describes how the image will be rendered or executed.
    With Pipeline
        .SetAmbientLight 0, 0, 0
        .SetBackColor 30, 30, 140
        .SetDithering False
        .SetSpecular False
        .SetFillMode R3DFILLMODE_SOLID
        .SetShadeMode R3DSHADEMODE_GOURAUD
        .SetTextureFilter R3DTEXTUREFILTER_LINEARFILTER
        .SetColorKeying True
        '.SetFog 250, 150, 30, 0, 500, R3DFOGTYPE_LINEAR
    End With

    '// Now that we know how the display will be executed we have to create
    '// the image to be rendered through loading texture files into strings.

End Sub

Public Sub CreateSky()

    '// Load the texture for the sky.
    TextureLib.Texture_Load "sky_panorama", "skyground_panorama.bmp"

    '//add the sky sphere with a radius of 2500.
    SkyDome.Sphere_Create "sky_panorama", "", 2500
```

```
End Sub

Public Sub CreateTerrain()

    '// The Terrain texture or "look" is created by loading in a file that contains grass,
    '// rocks or something of that nature. In this case a rocking terrain is loaded in. The
    '// path to the file is given and then it is bound to a string as a sort of nick name.

    TextureLib.Texture_Load "rockyground", "checkerdetail.bmp"

    '// The terrain is created using two files. One file "height.bmp" is a bitmap
    '// file which contains a grayscale bitmap. The lite color values are high
    '// elevation terrain features and the dark color values are the low elevation
    '// terrain features. The second file contains the detail that is laid ontop of
    '// the elevated terrain, "detail03.jpg".

    '// To generate the terrain we use the PolyVox2 object.

    PolyVox2.Scape_Create "terrain", "largeflatterrain2.bmp", 1500, True,
POLYVOXDETAIL_AVERANGE
    PolyVox2.Scape_SetTexture 1, "rockyground", R3DTEXTURELAYERCOMMAND_COLORBLEND

    '// The texture scale sets the resolution on the loaded texture. The bitmap is 512x512 for
    '// both the texture and the terrain. The texture(rockyground) is compressed so that 50x50
    '// images exist in the 512x512 bitmap expressing the terrain features.
    PolyVox2.Scape_SetTextureScale 1, 50, 50

    '// The texture stages are added and given priority based on a reaction percentage.
    PolyVox2.Blender_AddTexture "new_2.bmp", 50
    PolyVox2.Blender_AddTexture "new_4.bmp", 20

    '// The dynamic textures are generated and then blended together. The _setblendpower() sets
    '// the percentage of blend between the two dyanmic textures added above. This is compiled
    '// and stored as a temporary file in the TerrainData directory.
    PolyVox2.Blender_SetBlendPower 40
    PolyVox2.Blender_Compile "temp.bmp", R3DTEXTURERESOLUTION_32X32

End Sub

Public Sub CreateRunway()

    '// Load the texture for the runway.
    TextureLib.Texture_Load "runway", "runway.bmp"

    '// Load the numbers for the runway.
    TextureLib.Texture_Load "runwaynumber25", "runwaynumber25.bmp"
    TextureLib.Texture_Load "runwaynumber7", "runwaynumber7.bmp"

    '// Load the texture for the main taxi and minor taxi ways.
    TextureLib.Texture_Load "maintaxi", "maintaxi.bmp"
    TextureLib.Texture_Load "minortaxi", "minortaxi.bmp"
```

```
    '// Load the Material for the Hanger.
    TextureLib.Texture_Load "hangersiding", "metalsiding.bmp"
    '// The runway was created using a flattened box. This gives the illusion of the runway
    '// having some depth.
    MeshBuilder.Mesh_Create "runway"
    'MeshBuilder.Mesh_AddBox "runway", "matsundown", 3, 0.01, 64.5, R3DBLENDMODE_NONE,
R3DCULLMODE_DOUBLESIDED
    'MeshBuilder.Mesh_SetPosition 456.562, 0.005, 479.993
    MeshBuilder.Mesh_AddBox "runway", "matsundown", 3, 0.01, 85.36, R3DBLENDMODE_NONE,
R3DCULLMODE_DOUBLESIDED
    MeshBuilder.Mesh_SetPosition 475.052, 0.005, 490.668
    Call MeshBuilder.Mesh_SetRotation(0, 240, 0)
    MeshBuilder.Mesh_Finalize

    '// Add the Runway Numbers to the ends of the runway.
    '// 25
    MeshBuilder.Mesh_Create "runwaynumber25"
    MeshBuilder.Mesh_AddBox "runwaynumber25", "matsundown", 3, 0.007, 3,
R3DBLENDMODE_NONE, R3DCULLMODE_DOUBLESIDED
    'MeshBuilder.Mesh_SetPosition 509.40192, 0.009, 510.5
    MeshBuilder.Mesh_SetPosition 546.357, 0.009, 531.836
    Call MeshBuilder.Mesh_SetRotation(0, 60, 0)
    MeshBuilder.Mesh_Finalize
    '// 7
    MeshBuilder.Mesh_Create "runwaynumber7"
    MeshBuilder.Mesh_AddBox "runwaynumber7", "matsundown", 3, 0.007, 3,
R3DBLENDMODE_NONE, R3DCULLMODE_DOUBLESIDED
    MeshBuilder.Mesh_SetPosition 403, 0.009, 449
    Call MeshBuilder.Mesh_SetRotation(0, 240, 0)
    MeshBuilder.Mesh_Finalize

End Sub

Public Sub SetPosition()

    '// Set the start position for the camera. The camera is set to the center
    '// of the scene on a heading of 0 degrees.
    snglPosition_X = 567 '556
    snglPosition_Y = 0
    snglPosition_Z = 534 '556
    snglHeading = 0
    PI = 3.14159265358979

    snglCurrentFlightPathAngle = 0
    snglCurrentPseudoRoll = 0
    snglPreviousFlightPathAngle = 0
    snglPreviousPseudoRoll = 0
    dt = 0.05 'Seems a little slow, dropping to 0.03 was 0.05.
    snglPixel_Per_Meter = 0.1

    snglPosition_Y = 12 * snglPixel_Per_Meter
    Camera.SetPosition snglPosition_X, snglPosition_Y, snglPosition_Z
```

```
    '// Point nose of aircraft towards on a heading of zero degrees.
    Camera.SetRotation 0, 0, 0

End Sub

Public Sub CreatePAPI_Light_1()

    '// FURTHEST FROM RUNWAY.

   '// Load the texture for the PAPI's.
    TextureLib.Texture_Load "PAPIred", "PAPIred.bmp"
    TextureLib.Texture_Load "PAPIwhite", "PAPIwhite.bmp"

    '// Create the Light for the 1st or far left PAPI and default it to white.
    MeshBuilder.Mesh_Create "PAPI_1"
    MeshBuilder.Mesh_AddBox "PAPIred", "matsundown", 0.5, 0.5, 0.01, R3DBLENDMODE_ADD,
R3DCULLMODE_DOUBLESIDED
    'MeshBuilder.Mesh_SetPosition 496, 1, 489
    MeshBuilder.Mesh_SetPosition 513.639, 1, 499.187
    Call MeshBuilder.Mesh_SetRotation(0, 240, 0)
    MeshBuilder.Mesh_Finalize

End Sub

Public Sub CreatePAPI_Light_2()

    '// Load the texture for the PAPI's.
    TextureLib.Texture_Load "PAPIred", "PAPIred.bmp"
    TextureLib.Texture_Load "PAPIwhite", "PAPIwhite.bmp"

    '// Create the Light for the 2nd or middle left PAPI and default it to white.
    MeshBuilder.Mesh_Create "PAPI_2"
    MeshBuilder.Mesh_AddBox "PAPIred", "matsundown", 0.5, 0.5, 0.01, R3DBLENDMODE_ADD,
R3DCULLMODE_DOUBLESIDED
    'MeshBuilder.Mesh_SetPosition 494, 1, 491
    MeshBuilder.Mesh_SetPosition 511.639, 1, 501.187
    Call MeshBuilder.Mesh_SetRotation(0, 240, 0)
    MeshBuilder.Mesh_Finalize

End Sub

Public Sub CreatePAPI_Light_3()

    '// Load the texture for the PAPI's.
    TextureLib.Texture_Load "PAPIred", "PAPIred.bmp"
    TextureLib.Texture_Load "PAPIred", "PAPIwhite.bmp"
        '// Create the Light for the 3rd or middle right PAPI and default it to white.
    MeshBuilder.Mesh_Create "PAPI_3"
    MeshBuilder.Mesh_AddBox "PAPIred", "matsundown", 0.5, 0.5, 0.01, R3DBLENDMODE_ADD,
R3DCULLMODE_DOUBLESIDED
    'MeshBuilder.Mesh_SetPosition 492, 1, 493
    MeshBuilder.Mesh_SetPosition 509.639, 1, 503.187
    Call MeshBuilder.Mesh_SetRotation(0, 240, 0)
```

```
    MeshBuilder.Mesh_Finalize

End Sub

Public Sub CreatePAPI_Light_4()

    '//CLOSEST TO RUNWAY

    '// Load the texture for the PAPI's.
    TextureLib.Texture_Load "PAPIred", "PAPIred.bmp"
    TextureLib.Texture_Load "PAPIwhite", "PAPIwhite.bmp"
    '// Create the Light for the 4th or far right PAPI and default it to white.
    MeshBuilder.Mesh_Create "PAPI_4"
    MeshBuilder.Mesh_AddBox "PAPIred", "matsundown", 0.5, 0.5, 0.01, R3DBLENDMODE_ADD,
R3DCULLMODE_DOUBLESIDED
    'MeshBuilder.Mesh_SetPosition 490, 1, 495
    MeshBuilder.Mesh_SetPosition 507.639, 1, 505.187
    Call MeshBuilder.Mesh_SetRotation(0, 240, 0)
    MeshBuilder.Mesh_Finalize

End Sub

Public Sub SetPAPI_1(blnState As Boolean)
    ' TRUE = RED
    ' FALSE = WHITE
    With MeshBuilder

        If (blnState) Then
            .Mesh_SetPointer "PAPI_1"
            .Mesh_SetTexture "PAPIred", 0, R3DTEXTURELAYERCOMMAND_BUMP
        Else
            .Mesh_SetPointer "PAPI_1"
            .Mesh_SetTexture "PAPIwhite", 0, R3DTEXTURELAYERCOMMAND_BUMP
        End If

    End With

End Sub

Public Sub SetPAPI_2(blnState As Boolean)

    ' TRUE = RED
    ' FALSE = WHITE

    With MeshBuilder

        If (blnState) Then
            .Mesh_SetPointer "PAPI_2"
            .Mesh_SetTexture "PAPIred", 0, R3DTEXTURELAYERCOMMAND_BUMP
        Else
            .Mesh_SetPointer "PAPI_2"
            .Mesh_SetTexture "PAPIwhite", 0, R3DTEXTURELAYERCOMMAND_BUMP
        End If
```

```
    End With

End Sub

Public Sub SetPAPI_3(blnState As Boolean)

    ' TRUE = RED
    ' FALSE = WHITE

    With MeshBuilder

        If (blnState) Then
            .Mesh_SetPointer "PAPI_3"
            .Mesh_SetTexture "PAPIred", 0, R3DTEXTURELAYERCOMMAND_BUMP
        Else
            .Mesh_SetPointer "PAPI_3"
            .Mesh_SetTexture "PAPIwhite", 0, R3DTEXTURELAYERCOMMAND_BUMP
        End If

    End With

End Sub

Public Sub SetPAPI_4(blnState As Boolean)

    ' TRUE = RED
    ' FALSE = WHITE

    With MeshBuilder

        If (blnState) Then
            .Mesh_SetPointer "PAPI_4"
            .Mesh_SetTexture "PAPIred", 0, R3DTEXTURELAYERCOMMAND_BUMP
        Else
            .Mesh_SetPointer "PAPI_4"
            .Mesh_SetTexture "PAPIwhite", 0, R3DTEXTURELAYERCOMMAND_BUMP
        End If

    End With

End Sub

Public Sub Development_DisplayParameters()

    '// Development Display Parameters.
    Interface2D.Primitive_DrawText 1, 1, "Terrain Height: " + CStr(snglTerrainHeight * 10)
    Interface2D.Primitive_DrawText 1, 15, "Camera Altitude : " + CStr(snglPosition_Y * 1)
    Interface2D.Primitive_DrawText 1, 30, "fltdata Alt   : " + CStr(snglUp * 3.28)
    Interface2D.Primitive_DrawText 1, 45, "Camera Roll   : " + CStr(snglRoll)
    Interface2D.Primitive_DrawText 1, 60, "fltdata Roll  : " + CStr(snglPseudoRoll)
    Interface2D.Primitive_DrawText 1, 75, "Camera Pitch  : " + CStr(snglPitch)
    Interface2D.Primitive_DrawText 1, 90, "fltdata Pitch : " + CStr(snglFlightPathAngle)
```

```
    Interface2D.Primitive_DrawText 1, 105, "Camera Heading : " + CStr(snglHeading)
    Interface2D.Primitive_DrawText 1, 120, "fltdata GroundTrack : " + CStr(snglGroundTrack)
    Interface2D.Primitive_DrawText 1, 135, "Ground Speed : " + CStr(snglSpeed * 1.9438)

    Interface2D.Primitive_DrawText 700, 1, " Local X : " + CStr(snglPosition_X)
    Interface2D.Primitive_DrawText 700, 15, " Local Y : " + CStr(snglPosition_Z)

End Sub

Public Sub DisplayFramesPerSecond()

    '// Current fps (frames per second)
    Interface2D.Primitive_SetDrawColor 0, 0, 0
    Interface2D.Primitive_DrawText 380, 1, "fps:" + CStr(PowerMonitor.GetFrameTime)

End Sub

Public Sub ConvertFlightDataToSingle()

    ' New parameters for processing "HUB" packet data. Should not need those
    ' above once this is working.

    varSpeed = dblVelocity
    snglSpeed = varSpeed
    varGroundTrack = dblGroundTrack
    snglGroundTrack = varGroundTrack
    varFlightPathAngle = dblPitch
    snglFlightPathAngle = varFlightPathAngle

    varPseudoRoll = dblRoll
    snglPseudoRoll = varPseudoRoll
    '// Convert East in Meters to East in Pixels.
    varEast = dblEast
    snglEast = varEast
    snglAircraft_X = snglEast * snglPixel_Per_Meter + 512
    '// The convention is X cross Z = Y, so Z is North.
    varNorth = dblNorth
    snglNorth = varNorth
    snglAircraft_Z = snglNorth * snglPixel_Per_Meter + 512

    varUp = dblUp
    snglUp = varUp
    snglAircraft_Y = snglUp * snglPixel_Per_Meter + 0.8

End Sub

Public Sub Initialize_NovAtel()

    '*** SET COMM PORT 1 ACTIVE. ***
    frmNovAtel.SerialComm1.CommPort = 1

    '*** CHOOSE TO READ ONE CHARACTER AT A TIME. ***
    frmNovAtel.SerialComm1.RThreshold = 1
```

```
'*** READ ONE CHARACTER AT A TIME FROM THE BUFFER ***
frmNovAtel.SerialComm1.InputLen = 1

'*** READ ASCII TEXT DATA FROM BUFFER VERSUS BINARY ***
frmNovAtel.SerialComm1.InputMode = comInputModeText

'*** SET THE COMM PORT RATE TO 9600 BPS ***
frmNovAtel.SerialComm1.Settings = "9600,N,8,1"

'*** SEND DATA TO THE NOVATEL RECEVIER. ***

'The NovAtel receiver may be set at 9600 bps if just powered on or
'it may already be on and set at 115200 from previous use by this
'program. A command to set the rate to 9600 bps must be sent at both
'rates to insure the Novatel receiver's communication rate is known.

'First check to see if port is open, if it is close it and reopen it.
'This seems strange but it works well to insure that there is not a
'conflict between port allocation.

If (frmNovAtel.SerialComm1.PortOpen = True) Then
   frmNovAtel.SerialComm1.PortOpen = False
   frmNovAtel.SerialComm1.PortOpen = True
Else
   frmNovAtel.SerialComm1.PortOpen = True
End If

'Port is now open and set to transmit at 9600 bps. Send command to RCVR to
'request that it receive and transmit at 9600 bps.
frmNovAtel.SerialComm1.Output = "com com1 9600 N 8 1 N off on" & vbCr & vbLf

'Every time a command is sent there should be a brief moment for the
'receiver to accept the command and take action on it.
Sleep (300)

'If the RCVR is set at 115200 then the same request must be sent at that
'rate to get the receiver back to 9600 bps.

'First close the commport in order to change the rate.
frmNovAtel.SerialComm1.PortOpen = False

'Change the comm rate to 115200.
frmNovAtel.SerialComm1.Settings = "115200,N,8,1"

'Open comm port to send new request.
frmNovAtel.SerialComm1.PortOpen = True

'Send the new request at 115200 for the RCVR to transmit at 9600.
frmNovAtel.SerialComm1.Output = "com com1 9600 N 8 1 N off" & vbCr & vbLf
Sleep (300)

'Set the commport transmission rate back to 9600 bps.
```

```
    'First close the port.
    frmNovAtel.SerialComm1.PortOpen = False
    frmNovAtel.SerialComm1.Settings = "9600,N,8,1"

    '*** AT THIS POINT IN TIME THE RCVR AND THE COMMPORT SHOULD BOTH BE ***
    '*** TRANSMITTING AT 9600 BPS. ***
        'Open comm port.
    frmNovAtel.SerialComm1.PortOpen = True

    '*** UNLOG ALL PREVIOUS REQUEST FROM THE RECIEVER. ***
    frmNovAtel.SerialComm1.Output = "unlogall" & vbCr & vbLf
    Sleep (300)

    '*** SET THE RECEIVER RATE BACK TO 115200 BPS ***
    frmNovAtel.SerialComm1.Output = "com com1 115200 N 8 1 N off on" & vbCr & vbLf
    Sleep (300)

    '*** SET THE COMM PORT RATE TO 115200 BPS ***
    frmNovAtel.SerialComm1.PortOpen = False
    frmNovAtel.SerialComm1.Settings = "115200,N,8,1"

    '*** OPEN COMMPORT AND SEND REQUEST FOR BESTPOSA AND BESTVELA. ***
    frmNovAtel.SerialComm1.PortOpen = True
    frmNovAtel.SerialComm1.Output = "log com1 bestposa ontime .1" & vbCr & vbLf
    Sleep (300)
    frmNovAtel.SerialComm1.Output = "log com1 bestvela ontime .1" & vbCr & vbLf
    Sleep (300)
End Sub
```

B.2 Main Program for the Microcontroller

```
'*******************************************************************************
' Author:   Jahnavi Chakrabarty
' Date:     11 May 2004
' Purpose:  Display Real Time Calculated Pseudo-Roll from the Data Processing Computer.
'*******************************************************************************

Option Explicit 'Avoid Undifined Variables.

    '-- The Kernel32 API contains a function that pauses application execution
    '-- for a given amount of time, specified in milliseconds.
    '-- To use the function, it must first be declared in the
    '-- General Declarations section of the module in which it will be used:
    Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)    '

Private Sub Form_Load()

   With MSComm4

       .CommPort = 4
       .Settings = "19200,N,8,1"
       .PortOpen = True

   End With
 '*** Open the Flight Data File ***
    intFile = FreeFile
    Open "StateVectorData.txt" For Output As #intFile
    strFileHeader = "Header Time(s) East(m) North(m) Up(m) GrndSpd(m/s) GrndTrck(deg) Pitch(deg)
Roll(deg)"
    Print #intFile, strFileHeader

     '*** MAIN LOOP ***

    '// While condition set to true.
    blnRUN = True
    '*** INITIALIZE THE GPS RECEIVER ***
    Call Initialize_NovAtel
    While (Not blnRUN = False)
      '***** SET POSITION, ALTITUDE, AND ATTITUDE *****
       If (blnAttitudeCalculated) Then
              blnAttitudeCalculated = False

          '***** Convert Doubles to type Single. *****
          Call ConvertFlightDataToSingle

          Call Output_LEDStrips(snglPseudoRoll)
          Sleep (50)

       End If
       DoEvents
    Wend
End Sub
```

```
Public Sub Output_LEDStrips(snglPseudoRoll As Single)

Dim lit_LEDS As Long

    If snglPseudoRoll > 28 Then

        snglPseudoRoll = 28

    ElseIf snglPseudoRoll < -28 Then

        snglPseudoRoll = -28

    Else: snglPseudoRoll = Round(snglPseudoRoll)

   lit_LEDS = snglPseudoRoll + 27

    End If

    MSComm4.Output = Chr$(lit_LEDS)
    'Debug.Print " snglPseudoRoll "; snglPseudoRoll

End Sub

Property Get AttitudeCalculated() As Boolean

    AttitudeCalculated = blnAttitudeCalculated

End Property

Property Let AttitudeCalculated(blnAttitudeCalculatedIn As Boolean)
    blnAttitudeCalculated = blnAttitudeCalculatedIn
End Property

Public Sub ConvertFlightDataToSingle()

    varSpeed = dblVelocity
    snglSpeed = varSpeed

    varGroundTrack = dblGroundTrack
    snglGroundTrack = varGroundTrack

    varFlightPathAngle = dblPitch
    snglFlightPathAngle = varFlightPathAngle

    varPseudoRoll = dblRoll
    snglPseudoRoll = varPseudoRoll
    Debug.Print "snglPseudoRoll "; snglPseudoRoll

    '// Convert East in Meters to East in Pixels.
    varEast = dblEast
    snglEast = varEast
    'snglAircraft_X = snglEast * snglPixel_Per_Meter + 512
```

```
    '// The convention is X cross Z = Y, so Z is North.
    varNorth = dblNorth
    snglNorth = varNorth
    'snglAircraft_Z = snglNorth * snglPixel_Per_Meter + 512
    varUp = dblUp
    snglUp = varUp
    'snglAircraft_Y = snglUp * snglPixel_Per_Meter + 0.8

End Sub

Public Sub Initialize_NovAtel()

    '*** SET COMM PORT 1 ACTIVE. ***
    frmNovAtel.SerialComm1.CommPort = 1

    '*** CHOOSE TO READ ONE CHARACTER AT A TIME. ***
    frmNovAtel.SerialComm1.RThreshold = 1

    '*** READ ONE CHARACTER AT A TIME FROM THE BUFFER ***
    frmNovAtel.SerialComm1.InputLen = 1

    '*** READ ASCII TEXT DATA FROM BUFFER VERSUS BINARY ***
    frmNovAtel.SerialComm1.InputMode = comInputModeText

    '*** SET THE COMM PORT RATE TO 9600 BPS ***
    frmNovAtel.SerialComm1.Settings = "9600,N,8,1"

    '*** SEND DATA TO THE NOVATEL RECEVIER. ***

    'The NovAtel receiver may be set at 9600 bps if just powered on or
    'it may already be on and set at 115200 from previous use by this
    'program. A command to set the rate to 9600 bps must be sent at both
    'rates to insure the Novatel receiver's communication rate is known.
    'First check to see if port is open, if it is close it and reopen it.
    'This seems strange but it works well to insure that there is not a
    'conflict between port allocation.

    If (frmNovAtel.SerialComm1.PortOpen = True) Then
        frmNovAtel.SerialComm1.PortOpen = False
        frmNovAtel.SerialComm1.PortOpen = True
    Else
        frmNovAtel.SerialComm1.PortOpen = True
    End If

    'Port is now open and set to transmit at 9600 bps. Send command to RCVR to
    'request that it receive and transmit at 9600 bps.
    frmNovAtel.SerialComm1.Output = "com com1 9600 N 8 1 N off on" & vbCr & vbLf
    'Every time a command is sent there should be a brief moment for the
    'receiver to accept the command and take action on it.
    Sleep (300)

    'If the RCVR is set at 115200 then the same request must be sent at that
```

```
    'rate to get the receiver back to 9600 bps.

    'First close the commport in order to change the rate.
    frmNovAtel.SerialComm1.PortOpen = False
    'Change the comm rate to 115200.
    frmNovAtel.SerialComm1.Settings = "115200,N,8,1"
    'Open comm port to send new request.
    frmNovAtel.SerialComm1.PortOpen = True
    'Send the new request at 115200 for the RCVR to transmit at 9600.
    frmNovAtel.SerialComm1.Output = "com com1 9600 N 8 1 N off" & vbCr & vbLf
    Sleep (300)

    'Set the commport transmission rate back to 9600 bps.
    'First close the port.
    frmNovAtel.SerialComm1.PortOpen = False
    frmNovAtel.SerialComm1.Settings = "9600,N,8,1"

    '*** AT THIS POINT IN TIME THE RCVR AND THE COMMPORT SHOULD BOTH BE ***
    '*** TRANSMITTING AT 9600 BPS. ***
     'Open comm port.
    frmNovAtel.SerialComm1.PortOpen = True

    '*** UNLOG ALL PREVIOUS REQUEST FROM THE RECIEVER. ***
    frmNovAtel.SerialComm1.Output = "unlogall" & vbCr & vbLf
    Sleep (300)

    '*** SET THE RECEIVER RATE BACK TO 115200 BPS ***
    frmNovAtel.SerialComm1.Output = "com com1 115200 N 8 1 N off on" & vbCr & vbLf
    Sleep (300)

    '*** SET THE COMM PORT RATE TO 115200 BPS ***
    frmNovAtel.SerialComm1.PortOpen = False
    frmNovAtel.SerialComm1.Settings = "115200,N,8,1"
    '*** OPEN COMMPORT AND SEND REQUEST FOR BESTPOSA AND BESTVELA. ***
    frmNovAtel.SerialComm1.PortOpen = True
    frmNovAtel.SerialComm1.Output = "log com1 bestposa ontime .1" & vbCr & vbLf
    Sleep (300)
    frmNovAtel.SerialComm1.Output = "log com1 bestvela ontime .1" & vbCr & vbLf
    Sleep (300)

End Sub
```

B.3 Interrupt Service Routine

```
Public strGPS_String As String
Public strPOS_String As String
Public strVEL_String As String
Public blnProcess_String_FLAG As Boolean

Private Sub SerialComm1_OnComm()
 Dim varBuffer As Variant
 Dim strCharIn As String
 Dim strString As String
 Dim strPOSHeader As String
 Dim strVELHeader As String
 Dim strProcess_String As String
 Dim intStringLength As Integer
 Dim intPOSLength As Integer
 Dim intVELLength As Integer
 Dim intStringCount As Integer
 Dim strCRCchar As String
 Dim intCRCcheck As Integer
 Dim intStart As Integer
 Dim intStop As Integer
    strCRCchar = ","

    '*** Read Serial Buffer ***
    If (SerialComm1.CommEvent = comEvReceive) Then

        SerialComm1.RThreshold = 0
        varBuffer = SerialComm1.Input
        strCharIn = varBuffer
        DoEvents

        '*** Test for the start of the next string. ***
        'If found then copy string to processing string and continue collecting.
        If ("#" = strCharIn) Then
            blnPoundFound = True
        End If

        If (blnPoundFound = True) Then
            If Not ("*" = strCharIn) Then
                strGPS_String = strGPS_String & strCharIn
            End If
        End If

        If ("*" = strCharIn) Then
            blnPoundFound = False
            strProcess_String = strGPS_String & "*"
            strGPS_String = ""              'Clear GPS string for incoming string.
            blnProcess_String_FLAG = True   'Inform processing algorithm to start.
        End If

        SerialComm1.RThreshold = 1
```

```
    End If

    If (blnProcess_String_FLAG) Then

        intCRCcheck = 0
        intStart = 1
        intStop = Len(strProcess_String)
        While (intStart < intStop)
            intStart = InStr(intStart, strProcess_String, strCRCchar, 1)
            intStart = intStart + 1
            intCRCcheck = intCRCcheck + 1
        Wend

        If (intCRCcheck = 9) Then
            Call ProcessPacket(strProcess_String)
            Print #intFile, strProcess_String
            strProcess_String = ""
            blnProcess_String_FLAG = False
        Else
            strProcess_String = ""
            blnProcess_String_FLAG = False
        End If

    End If

End Sub
```

B.4 Data Extraction Routine

```
Option Explicit
    '-- The Kernel32 API contains a function that pauses application execution
    '-- for a given amount of time, specified in milliseconds.
    '-- To use the function, it must first be declared in the
    '-- General Declarations section of the module in which it will be used:
    Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

'***** THIS IS USED TO PROCESS DATA SENT FROM THE "HUB" COMPUTER. *****
Public Sub ProcessPacket(ByVal strDataString As String)

    'Extract the data varibles using PacketDataExtract.
    Call PacketDataExtract(strDataString)

    'Convert strings to floating point numbers.
    'Call StringDataToFloat

    dblTimeStamp = Val(strTimeStamp)
    dblEast = Val(strEast)
    dblNorth = Val(strNorth)
    dblUp = Val(strUp)
    dblVelocity = Val(strVelocity)
    dblGroundTrack = Val(strGroundTrack)
    dblPitch = Val(strPitch)
    dblRoll = Val(strRoll)
    'Debug.Print "dblRoll "; dblRoll

    ' Tell main form that an attitude and position is available.
    frmPseudoRolljanu.AttitudeCalculated = True

End Sub

Public Sub PacketDataExtract(ByVal strDataString As String)
 Dim CommaPlace1 As Integer
 Dim CommaPlace2 As Integer
 Dim PlaceCount As Integer
 Dim intFieldCount As Integer
 Dim strSearchChar As String * 1

    'The data packet is formated as follows:
    ' #FLTDATA,GPSTIME,EAST,NORTH,UP,VELOCITY,GROUNDTRACK,PICTH,ROLL*
    ' The GPSTIME is in field number 2.

    PlaceCount = 1
    intFieldCount = 1
    strSearchChar = ","

    While ((PlaceCount < Len(strDataString)) And (intFieldCount <= 8))
        CommaPlace1 = InStr(PlaceCount, strDataString, strSearchChar, vbTextCompare)
        CommaPlace2 = InStr(CommaPlace1 + 1, strDataString, strSearchChar, vbTextCompare)
        PlaceCount = CommaPlace1 + 1
```

```
      Select Case intFieldCount

        Case 1
          strTimeStamp = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "Time Stamp: " & strTimeStamp

        Case 2
          strEast = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "East: " & strEast

        Case 3
          strNorth = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "North: " & strNorth

        Case 4
          strUp = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "Up: " & strUp

        Case 5
          strVelocity = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "Velocity: " & strVelocity

        Case 6
          strGroundTrack = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "GroundTrack: " & strGroundTrack

        Case 7
          strPitch = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "Pitch: " & strPitch

        Case 8
          strRoll = Mid(strDataString, CommaPlace1 + 1, (CommaPlace2 - CommaPlace1))
          'Debug.Print "Roll: " & strRoll

      End Select

      intFieldCount = intFieldCount + 1

    Wend
End Sub

Public Sub POS_FileDataExtract(ByVal strGPS_String As String)

    '*** EXTRACT TIME STAMP ***
    strPOS_TimeStamp = DataExtract(strGPS_String, 6)

     '*** EXTRACT LATITUDE ***
    strLatitude = DataExtract(strGPS_String, 12)

     '*** EXTRACT LONGITUDE ***
    strLongitude = DataExtract(strGPS_String, 13)

     '*** EXTRACT Height***
```

```
    strHeight = DataExtract(strGPS_String, 14)

End Sub

Public Sub VEL_FileDataExtract(strGPS_String As String)

    '*** EXTRACT TIME STAMP ***
    strVEL_TimeStamp = DataExtract(strGPS_String, 6)

     '*** EXTRACT HORIZONTAL SPEED ***
    strHorizontalSpeed = DataExtract(strGPS_String, 13)

     '*** EXTRACT GROUND TRACK ***
    strGroundTrack = DataExtract(strGPS_String, 14)

     '*** EXTRACT VERTICAL SPEED***
    strVerticalSpeed = DataExtract(strGPS_String, 15)
End Sub

Public Function DataExtract(ByVal strStringIn As String, ByVal intFieldNumber As Integer) As String
 Dim intFieldCount As Integer
 Dim intFieldLength As Integer
 Dim intCharCount As Integer
 Dim strFieldMark As String * 1
 Dim strTestString As String * 1
    strFieldMark = ","
    intFieldCount = 0   'Must start at zero.
    intCharCount = 1    'Must Start at One.

    Debug.Print strStringIn & " Field Number : " & CStr(intFieldNumber)

Public Function ReturnFieldLength(ByVal strStringIn As String, ByVal intFieldStart As Integer) As
Integer
 Dim intFieldCharCount As Integer
 Dim strTestString As String * 1
 Dim strFieldMark As String * 1
 Dim blnFLAG As Boolean
    intFieldCharCount = 0
    strFieldMark = ","

    blnFLAG = True

    While (blnFLAG)

        strTestString = Mid(strStringIn, intFieldStart, 1)

        If (strTestString = strFieldMark) Then

            blnFLAG = False

        Else

            intFieldStart = intFieldStart + 1
```

```
          intFieldCharCount = intFieldCharCount + 1

       End If
    Wend
    ReturnFieldLength = intFieldCharCount
End Function

Public Function TimeStampCheck() As Boolean
 Dim intFLAG As Boolean
    intFLAG = False
    If (StrComp(strPOS_TimeStamp, strVEL_TimeStamp, vbTextCompare) = 1) Then
       intFLAG = True
       End If
    TimeStampCheck = intFLAG
End Function

Public Sub StringDataToFloat()

    dblPOS_TimeStamp = Val(strPOS_TimeStamp)
    dblLatitude = Val(strLatitude)
    dblLongitude = Val(strLongitude)
    dblHeight = Val(strHeight)

    dblVEL_TimeStamp = Val(strVEL_TimeStamp)
    dblHorizontalSpeed = Val(strHorizontalSpeed)
    dblGroundTrack = Val(strGroundTrack)
    dblVerticalSpeed = Val(strVerticalSpeed)
    '* New variables for packet data. Should not need those listed above.
    dblTimeStamp = Val(strTimeStamp)
    dblEast = Val(strEast)
    dblNorth = Val(strNorth)
    dblUp = Val(strUp)
    dblVelocity = Val(strVelocity)
    dblGroundTrack = Val(strGroundTrack)
    dblPitch = Val(strPitch)
    dblRoll = Val(strRoll)

End Sub

Public Sub ClearStrings()
    strPOS_TimeStamp = ""
    strLatitude = ""
    strLongitude = ""
    strHeight = ""

    strVEL_TimeStamp = ""
    strHorizontalSpeed = ""
    strGroundTrack = ""
    strVerticalSpeed = ""
End Sub
```

B.5 Global Variables

```
'*** Constants ***
 Public PI As Double
 Public dblTimeRateOfChange As Double
 Public i As Integer
 Public j As Integer
 Public k As Integer

'* East, North, Up.
 Public dblEast As Double
 Public dblNorth As Double
 Public dblUp As Double

'* File Manipulators.
 Public intGPSFileHandler As Integer
 Public intFLTDATAHandler As Integer
 Public blnAttitudeCalculated As Boolean

'* State Vector Information taken from GPS Strings.
 Public strPOS_TimeStamp As String
 Public strLatitude As String
 Public strLongitude As String
 Public strHeight As String

 Public strVEL_TimeStamp As String
 Public strHorizontalSpeed As String
 Public strVerticalSpeed As String
 Public strGroundTrack As String

'* Coverted State Vector Information from strings to floating points.
 Public dblPOS_TimeStamp As Double
 Public dblLatitude As Double
 Public dblLongitude As Double
 Public dblHeight As Double

 Public dblVEL_TimeStamp As Double
 Public dblHorizontalSpeed As Double
 Public dblVerticalSpeed As Double
 Public dblGroundTrack As Double

'* i,j,k Components of the Velocity Vector.
 Public dblVelocityVector(3) As Double
 Public dblPreviousVelocityVector(3) As Double
 Public dblVelocityNorth As Double
 Public dblVelocityEast As Double
 Public dblVelocityDown As Double
 Public dblVelocityMagnitude As Double
 Public dblUnitVelocityVector(3) As Double

'* i,j,k Components of the Acceleration Vector.
 Public dblAcceleration(3) As Double
```

```
 Public dblNormalAcceleration(3) As Double
 Public dblTangentialAcceleration(3) As Double
 Public dblAccelerationMagnitude As Double

'* i,j,k Components of the Gravitational Acceleration Vector.
 Public dblGravity(3) As Double
 Public dblNormalGravity(3) As Double
 Public dblTangentialGravity(3) As Double
 Public dblGravityMag As Double

'* Flight Path Angle Data.
 Public dblFlightPathAngleDeg As Double
 Public dblFlightPathAngleRAD As Double

'* Pseudo-Roll Data
 Public dblPseudoRollRad As Double
 Public dblPseudoRollDeg As Double

'* Lift and Reference Vectors.
 Public dblLiftVector(3) As Double
 Public dblLiftMag As Double
 Public dblReferenceVector(3) As Double
 Public dblReferenceMag As Double

'* Filters for Acceleration, Pseudo-Roll, and Flight Path Angle.
 'Public dblAcceleration_FILTER(3)(3) as Double, NOT used at this time.
 Public dblPSR_FILTER(3) As Double
 Public dblFPA_FILTER(3) As Double

'*** The following variables are used to control the attitude of the display. ***
 Public snglRoll As Single
 Public snglPitch As Single
 Public snglHeading As Single

 '***** Camera XYZ Position *****
 Public snglPosition_X As Single   'Local X
 Public snglPosition_Y As Single   'Local Height
 Public snglPosition_Z As Single   'Local Z

 '***** Aircraft ENU position as single *****
 Public snglEast As Single
 Public snglNorth As Single
 Public snglUp As Single

 Public varEast As Single
 Public varNorth As Single
 Public varUp As Single

 'Convention for Y and Z are switched. X cross Z = Y.
 Public snglAircraft_X As Single    'Local Aircraft X (East)
 Public snglAircraft_Y As Single    'Local Aircraft Y (Height)
 Public snglAircraft_Z As Single    'Local Aricraft Z,(North)
```

```
'***** Camera Positioning on Terrain *****
Public snglTerrainHeight As Single
Public snglTerrainOffset As Single
Public snglCameraOffset As Single

'***** Flight Data Information *****
Public snglAltitude As Single
Public snglSpeed As Single
Public snglGroundTrack As Single
Public snglFlightPathAngle As Single
Public snglPseudoRoll As Single

'***** Variant Flight Data Information *****
Public varAltitude As Variant
Public varSpeed As Variant
Public varGroundTrack As Variant
Public varFlightPathAngle As Variant
Public varPseudoRoll As Variant

'***** Temporary Flight Data Storage ****
Public snglCurrentFlightPathAngle As Single
Public snglCurrentPseudoRoll As Single
Public snglCurrentAltitude As Single
Public snglTempAltitude As Single
Public snglCurrentGroundTrack As Single

Public snglPreviousFlightPathAngle As Single
Public snglPreviousPseudoRoll As Single
Public snglPreviousAltitude As Single
Public snglPreviousGroundTrack As Single

'***** Trig functions expect a double as input, must convert sngl to double through variant. *****
Public dblFPArad As Double
Public dblFPAdeg As Variant
Public varFPAdeg As Variant

'***** Movement of Camera through BitMap *****
Public dt As Single
Public snglPixel_Per_Meter As Single
Public snglAltitudeCorrection As Single
'Public gColResult As R3DCOLLISIONRESULT_TYPE  'Collision Detection.

Public blnRUN As Boolean

Public snglStartTime As Single
Public snglStopTime As Single
Public snglDelayTime As Single

Public intFile As Integer
Public strFileHeader As String
Public strFilePath As String

'***** New Variables for Processing Packet Data *****
```

```
Public strTimeStamp As String
Public strEast As String
Public strNorth As String
Public strUp As String
Public strVelocity As String
'Public strGroundTrack As String
Public strPitch As String
Public strRoll As String

Public dblTimeStamp As Double
'Public dblEast As Double
'Public dblNorth As Double
'Public dblUp As Double
Public dblVelocity As Double
'Public dblGroundTrack As Double
Public dblPitch As Double
Public dblRoll As Double

Public strDataPacket As String

Public blnPoundFound As Boolean

 '***** Glide Slope *****
Public dblGlideSlope As Double
 ' Public dbltempRoll As Double
```

Appendix C: Microcontroller Code

This part of the code was developed by Don Venable. The flow of logic is as follows:

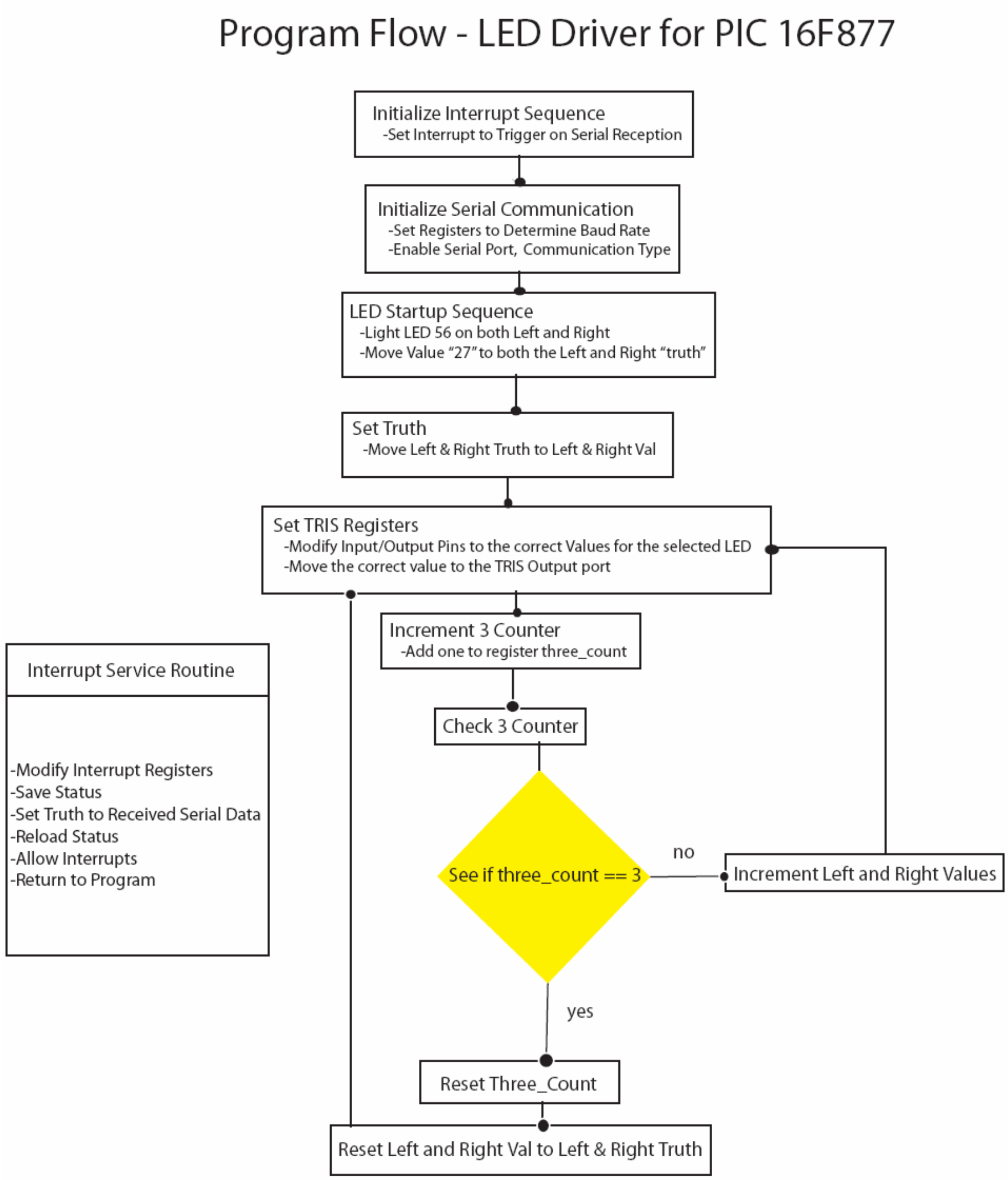

Figure C.1: Flow Chart for the Microcontroller Code

C.1 Microcontroller Routine

```
;Don Venable
;May 10, 2004
;LED Driver Code - Augmented Synthetic Display System
;Joint University Program
;Assembly Code for PIC16F877 Microcontroller

PCL equ 0x02
trisb equ 0x86
portb equ 0x06
trisc equ 0x07
trisd equ 0x08
portd equ 0x08
status equ 0x03                 ;This is the Status Register
intcon equ 0x0B                 ;Interrupt Control Register
pir_1 equ 0x0C                  ;This Register is the Perhip. Interupt FLAG register
pie_1 equ 0x8C                  ;This Register configures Perhip. Interrupts
rcreg equ 0x1A                  ;The Register which contains Received serial data
rcsta equ 0x18                  ;Receive Status and Control Register
spbrg equ 0x99                  ;Serial Port Baud Rate Generator, sets the baud used to communicate.
txsta equ 0x98                  ;Transmit Status and Control Register
rcstore equ 0x21          ;Location where I will move the contents from rcreg into, so that I can clear
rcreg
;The reason for this is that the RCIF flag will not clear unless RCREG has been read and cleared.
currentval equ 0x22             ;The current VALUE of which LED I am currently operating on.
three_count equ 0x23      ;I need a register to store the VALUE fifty_six, so I can subtract from it
status_store equ 0x24
duty_1 equ 0x25                 ;Storage for the duty timer
duty_2 equ 0x26                 ;Storage for the off delay timer
zero_check equ 0x27             ;Storage for ZERO CHECK
w_store equ 0x28;This will store the work reg when we enter the ISR so that nothing will be disturbed.
left_val equ 0x29
right_val equ 0x2A
duty_3 equ 0x2B
three_check equ 0x2C
right_truth equ 0x2D
left_truth equ 0x2E
fifty_six equ 0x2F

;Program Body

  org 0x00                          ;Start Loading Program Data @ address 0x00
  nop
  nop
  nop
  goto 0x05                         ;0x03 - Skip to 0x05, because 0x04 will get called anytime
there is an INTERRUPT
  goto ISRoutine  ;Calls the Interrupt Service Routine ( Anytthing that will happen during an interrupt )
;Now we can actually start the main program, now that the ISR formality is over.
;Let's set up the Serial Transmission to Interrupt when data is received
;Also let's make it a high speed transmission ( High Speed w/ baud @ 19.2 kilobaud )
```

```
;Setting up the Interrupt side
   bcf status, 6
   bsf status, 5                          ;This will goto bank 1, so we can acces PIE
   movlw b'11000000'                      ;This will enable Interrupts ( bit 7, GIE ), and Activate the Perhip.
Interrupts
   movwf intcon
   movlw b'00100000'                      ;This will enable the receive data interrupt (when data is received,
throw the interrupt )
   movwf pie_1                                        ;Modifying the Perhip. Interrupt Flag Register
;Set up the TRIS in/outputs
   movlw b'11000000'                      ;Pin 7 of PORTC is now setup as an INPUT so we can use it to receive
serial data
   movwf trisc
   movlw 0xFF
   movwf 0x81
;Setting up the Serial Port Configuration
   movlw b'00000100';Even though this is the TRANSFER control register, it still contains the bits that set
serial mode
   movwf txsta                  ;We set the port to asynch. mode, and to use a HIGH speed Baud rate
                                ;Now we are going to configure the baud rate
   movlw d'12';From page 98 of the PIC16f877 data sheet, this will set the baud rate generator to the
following:
   ;with a 4mhz crystal:    19.2 Kilobaud, with a 0.16% clock error

   movwf spbrg                  ;This register controls the baud rate generator, so we move our settings into it.
   bcf status, 5                ;Finally we will enable receiving on over the serial port ( Pins RC7 )
   movlw b'10010000'            ;This will set SPEN (Enable Serial Port) and CREN ( Continous Receive Enable
Bit )
   movwf rcsta                  ;Moves the settings into the control register
debug

   movlw d'56'
   movwf fifty_six
   movlw d'03'
   movwf three_check
   clrf three_count
   bsf status, 5
   movlw d'01'
   call table_tris
   movwf trisb
   bcf status, 5
   movlw d'01'
   call table_io
   movwf portb
   call delay
   call delay
   call delay
   bsf status, 5
   movlw d'56'
   call table_tris
   movwf trisb
   bcf status, 5
   movlw d'56'
```

```
   call table_io
   movwf portb
   movlw d'27'
   movwf right_val
   movwf left_val
   movwf left_truth
   movwf right_truth
     call delay
loopover
   clrwdt
   bcf status, 5
   movf left_val, w
   bsf status, 5
   call table_tris
   movwf trisb
   bcf status, 5
   movf right_val, w
   bsf status, 5
   call table_tris
   movwf trisd
   bcf status, 5
   movf left_val, w
   call table_io
   movwf portb
   bcf status, 5
   movf right_val, w
   call table_io
   movwf portd
   ;call delay
   bcf status, 2
   incf three_count, f
   movf three_count, w
   subwf three_check, w
   btfss status, 2
   goto not_zero
   clrf three_count
   movf left_truth, w
   movwf left_val
   movf right_truth, w
   movwf right_val
   goto loopover
not_zero
   incf right_val, f
   incf left_val, f
    goto loopover
delay                           ;3993 cycles
            movlw   0x1E
            movwf   duty_1
            movlw   0x04
            movwf   duty_2
delay_0
    clrwdt
            decfsz  duty_1, f
```

```
            goto     $+2
            decfsz   duty_2, f
            goto     delay_0                     ;3 cycles
            goto     $+1
            nop                           ;4 cycles (including call)
            return
;This is the interrupt routine. It should only be triggered when some data is received from the USART
Receiver.
ISRoutine
  bcf intcon, 7              ;This clears the GIE bit, which disallows any other interrupt from occuring
  bcf status, 5
  movwf w_store
  movf status, w
  movwf status_store
  clrwdt
  movf rcreg, w              ;This will put the contents of what was just received into my storage register
  movwf rcstore
  clrf rcreg;  The RCIF flag is hardware cleared, and it will only be cleared once RCREG is read and
cleared;
;It was already read above... So I am just clearing RCREG, which hopefully will clear the RCIF flag
  movwf left_truth
  subwf fifty_six, w
  movwf right_truth
  decf left_truth, f
  decf right_truth, f
  bsf intcon, 6
  bsf intcon, 7                         ;Resetting the GIE bit to allow interrupts, and ensuring that Intcon6
(Prehip. Interrupt) Is still set
   movf status_store, w
  movwf status
  movf w_store, w
 Retfie
table_io
 addwf PCL,f
 retlw b'00000000' ; NO LEDS TO BE LIT
 retlw b'00000001' ; Led 1
 retlw b'00000010' ; Led 2
 retlw b'00000001' ; Led 3
 retlw b'00000100' ; Led 4
 retlw b'00000001' ; Led 5
 retlw b'00001000' ; Led 6
 retlw b'00000001' ; Led 7
 retlw b'00010000' ; Led 8
 retlw b'00000001' ; Led 9
 retlw b'00100000' ; Led 10
 retlw b'00000001' ; Led 11
 retlw b'01000000' ; Led 12
 retlw b'00000001' ; Led 13
 retlw b'10000000' ; Led 14
 retlw b'00000010' ; Led 15
 retlw b'00000100' ; Led 16
 retlw b'00000010' ; Led 17
 retlw b'00001000' ; Led 18
```

```
 retlw b'00000010' ; Led 19
 retlw b'00010000' ; Led 20
 retlw b'00000010' ; Led 21
 retlw b'00100000' ; Led 22
 retlw b'00000010' ; Led 23
 retlw b'01000000' ; Led 24
 retlw b'00000010' ; Led 25
 retlw b'10000000' ; Led 26
 retlw b'00000100' ; Led 27
 retlw b'00001000' ; Led 28
 retlw b'00000100' ; Led 29
 retlw b'00010000' ; Led 30
 retlw b'00000100' ; Led 31
 retlw b'00100000' ; Led 32
 retlw b'00000100' ; Led 33
 retlw b'01000000' ; Led 34
 retlw b'00000100' ; Led 35
 retlw b'10000000' ; Led 36
 retlw b'00001000' ; Led 37
 retlw b'00010000' ; Led 38
 retlw b'00001000' ; Led 39
 retlw b'00100000' ; Led 40
 retlw b'00001000' ; Led 41
 retlw b'01000000' ; Led 42
 retlw b'00001000' ; Led 43
 retlw b'10000000' ; Led 44
 retlw b'00010000' ; Led 45
 retlw b'00100000' ; Led 46
 retlw b'00010000' ; Led 47
 retlw b'01000000' ; Led 48
 retlw b'00010000' ; Led 49
 retlw b'10000000' ; Led 50
 retlw b'00100000' ; Led 51
 retlw b'01000000' ; Led 52
 retlw b'00100000' ; Led 53
 retlw b'10000000' ; Led 54
 retlw b'01000000' ; Led 55
 retlw b'10000000' ; Led 56
   return
table_tris
 addwf PCL,f
 retlw b'00000000' ; NO LEDS TO BE LIT
 retlw b'11111100' ; Led 1
 retlw b'11111100' ; Led 2
 retlw b'11111010' ; Led 3
 retlw b'11111010' ; Led 4
 retlw b'11110110' ; Led 5
 retlw b'11110110' ; Led 6
 retlw b'11101110' ; Led 7
 retlw b'11101110' ; Led 8
 retlw b'11011110' ; Led 9
 retlw b'11011110' ; Led 10
 retlw b'10111110' ; Led 11
```

```
retlw b'10111110' ; Led 12
retlw b'01111110' ; Led 13
retlw b'01111110' ; Led 14
retlw b'11111001' ; Led 15
retlw b'11111001' ; Led 16
retlw b'11110101' ; Led 17
retlw b'11110101' ; Led 18
retlw b'11101101' ; Led 19
retlw b'11101101' ; Led 20
retlw b'11011101' ; Led 21
retlw b'11011101' ; Led 22
retlw b'10111101' ; Led 23
retlw b'10111101' ; Led 24
retlw b'01111101' ; Led 25
retlw b'01111101' ; Led 26
retlw b'11110011' ; Led 27
retlw b'11110011' ; Led 28
retlw b'11101011' ; Led 29
retlw b'11101011' ; Led 30
retlw b'11011011' ; Led 31
retlw b'11011011' ; Led 32
retlw b'10111011' ; Led 33
retlw b'10111011' ; Led 34
retlw b'01111011' ; Led 35
retlw b'01111011' ; Led 36
retlw b'11100111' ; Led 37
retlw b'11100111' ; Led 38
retlw b'11010111' ; Led 39
retlw b'11010111' ; Led 40
retlw b'10110111' ; Led 41
retlw b'10110111' ; Led 42
retlw b'01110111' ; Led 43
retlw b'01110111' ; Led 44
retlw b'11001111' ; Led 45
retlw b'11001111' ; Led 46
retlw b'10101111' ; Led 47
retlw b'10101111' ; Led 48
retlw b'01101111' ; Led 49
retlw b'01101111' ; Led 50
retlw b'10011111' ; Led 51
retlw b'10011111' ; Led 52
retlw b'01011111' ; Led 53
retlw b'01011111' ; Led 54
retlw b'00111111' ; Led 55
retlw b'00111111' ; Led 56

 return

END
```