Thesis Proposal

Glider Avionics Package

Ondrej Novak

Introduction:

The main goal of this thesis is to design a suitable device that will assist professional and amateur glider pilots in making decisions during their flights. This device will display all flight characteristics such as altitude, air speed, vertical speed, compass and position, and it will perform various calculations to ease the pilot's effort in decision-making process. The device should be able to calculate and display direction and strength of wind at current altitude, optimal speed to reach the destination at specified altitude and it should also give the pilot information about the average vertical speed during the day. Because flying gliders is closely related to physics, meteorology and navigation, my research is going to touch all those elements with focus on digital processing of natural variables. The final device should be helpful to pilots, easy to operate, and it should also make it easy for competition judges and referees to measure pilot's performance. My goal is to include features that are not available in current commercial devices specifically designed for gliders. It is also desirable to keep the cost low and reasonable.

Related Work:

The theory of airspeed measurements, Bernoulli's Principle and other related issues are described at the Virginia Tech University's website [1]. I'm going to use those resources to select appropriate pressure sensors for my application. Mat Redsell describes some useful hints for final airspeed calibration [2] and Kevin Horton's RV-8 Project might be handy in identifying possible static system errors [3].

Marko Rocznik describes a very useful experience on his web site [4]. He's built electronic altimeter and variometer and his article is going to be very beneficial for my project since he provides detailed description of his work including circuit schematics.

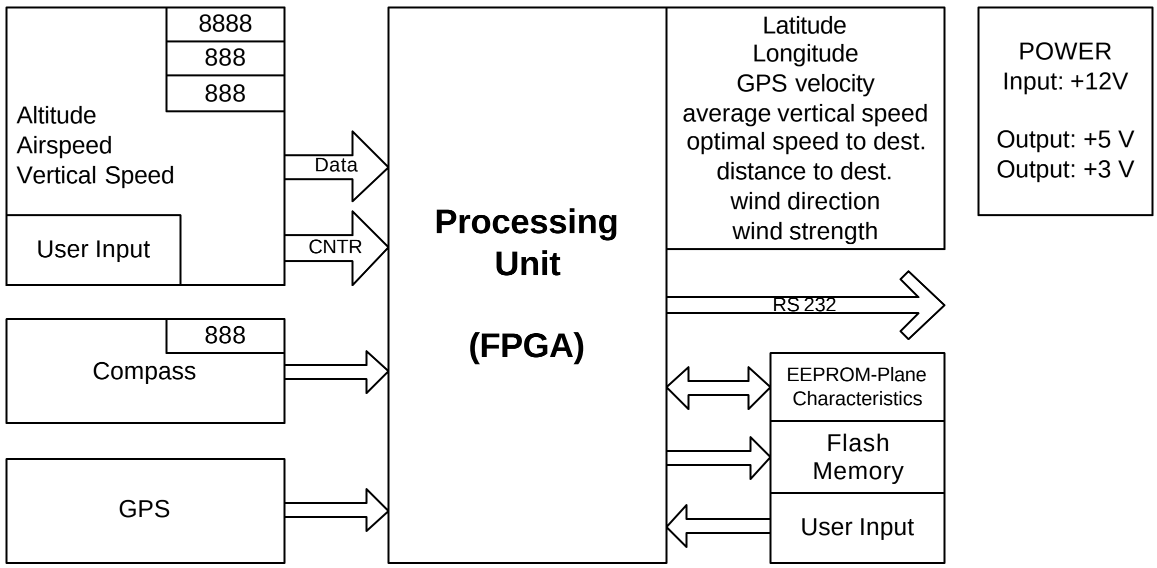
As the most difficult part of my project I consider building an accurate and inexpensive compass. All commercial products are very expensive so I'm going to design my own three-dimensional tilt-compensated compass. Michael Caruso wrote a very useful material describing development of such a device [5]. He discussed the issues of magnetic field measurements and offers solutions to overcome common obstacles. Considering requirements for my purpose, I've selected magnetic field sensors manufactured by Honeywell. I'm going to take advantage of the manufacturer's recommended application [6] with several modifications to fit my needs.

To be able to perform proposed calculations, I'll also need to realize a GPS receiver. After researching numerous possible resources, I've decided to use Motorola's M12+ Oncore chip for which the manufacturer provides detailed information and application descriptions [7].

The proposed device is going to employ two microcontrollers. Detailed hardware and software description with various practical examples can be found in Jonathan W. Valvano's publication [8] and Manufacturer's reference manual [9].

Effort details:

Device Block Diagram:



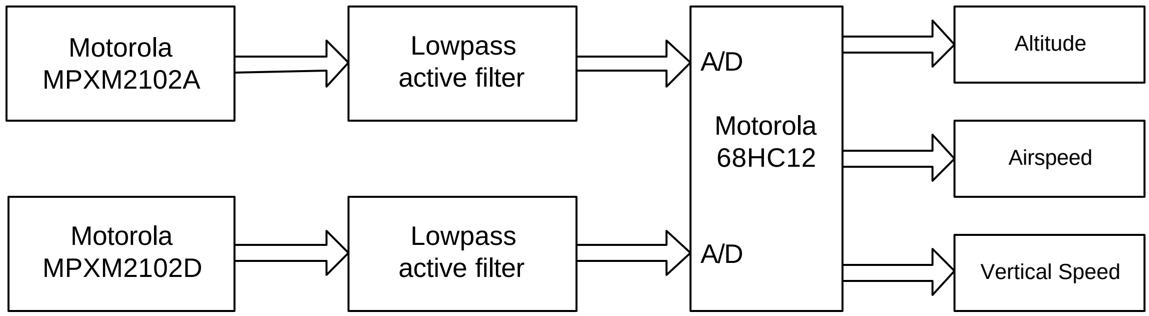
Measuring altitude, airspeed and vertical speed:

Sensor selection process:

To be able to indicate altitude, airspeed and vertical speed I have to be capable of measuring barometric (static) and dynamic pressures. Measuring static pressure is straight forward and many manufacturers offer sensors that will meet my needs. Dynamic pressure however, is the difference between the total and static pressure that are picked up by the pitot tube. There are two ways to obtain the dynamic pressure. First, I could have two pressure sensors, A/D converter and I could subtract those two quantities digitally in a microcontroller. This solution has a major glitch. The A/D converter would have to have at least 24-bit precision since the dynamic pressure is very small compared to the static pressure. Another solution is to use one differential sensor that will do the subtraction for me. In that case, I'll be able to use a 10-bit ADC, which is integrated in most microcontrollers.

There are two major sources of errors in digital pressure measurements. One comes from the fact that air density depends on temperature. This error however, is only about 4% per 20 degrees F and in all applications I've found, it was neglected. Another problem is a possible noise in the analog signal. To minimize both possible sources of errors, I selected temperature compensated pressure sensors made by Motorola. I'm also going to filter the analog signal from possible noises and I will calculate an average of 16 samples to make the output more stable and accurate.

Block diagram:



<u>Compass Design:</u>

Building an accurate electronic compass that will be used on-board of a plane is not as easy as it might look. At first, I assumed that measuring earth's magnetic field in two directions would be sufficient to accurately calculate the heading direction. As I found out, that was not a good assumption. Tilting the instruments during a flight brings an error large enough that I have to provide corrections. For this reason, it is necessary to sense the magnetic field in all three directions and mathematically correct the errors. As I found out, Honeywell offers a package that contains all sensors needed to build a compass with a tilt correction. The manufacturer also provides a lot of documentations and recommendations how to build an accurate magneto-resistive compass. I'll use the recommended schematics and I'll develop my own software to perform the calculations. To provide at least 1-degree accuracy, it is necessary to have a 3-channel 10-bit ADC. Since I'm already using one Motorola microcontroller and I'm familiar with programming this chip, I'm going to use another 68HC12 for the compass.

Errors Caused by Tilting:

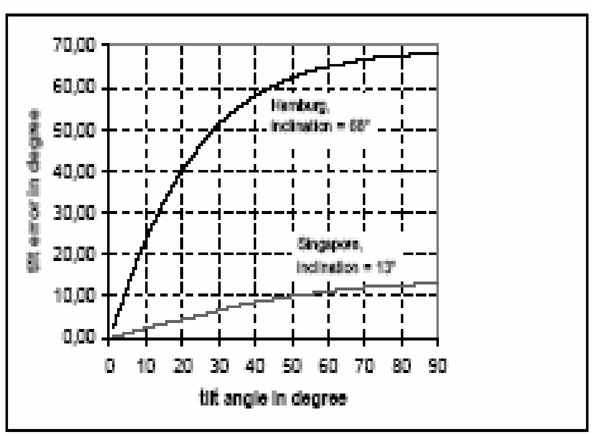


Figure 23 Tilt error magnitudes

Recommended Honeywell Schematics:

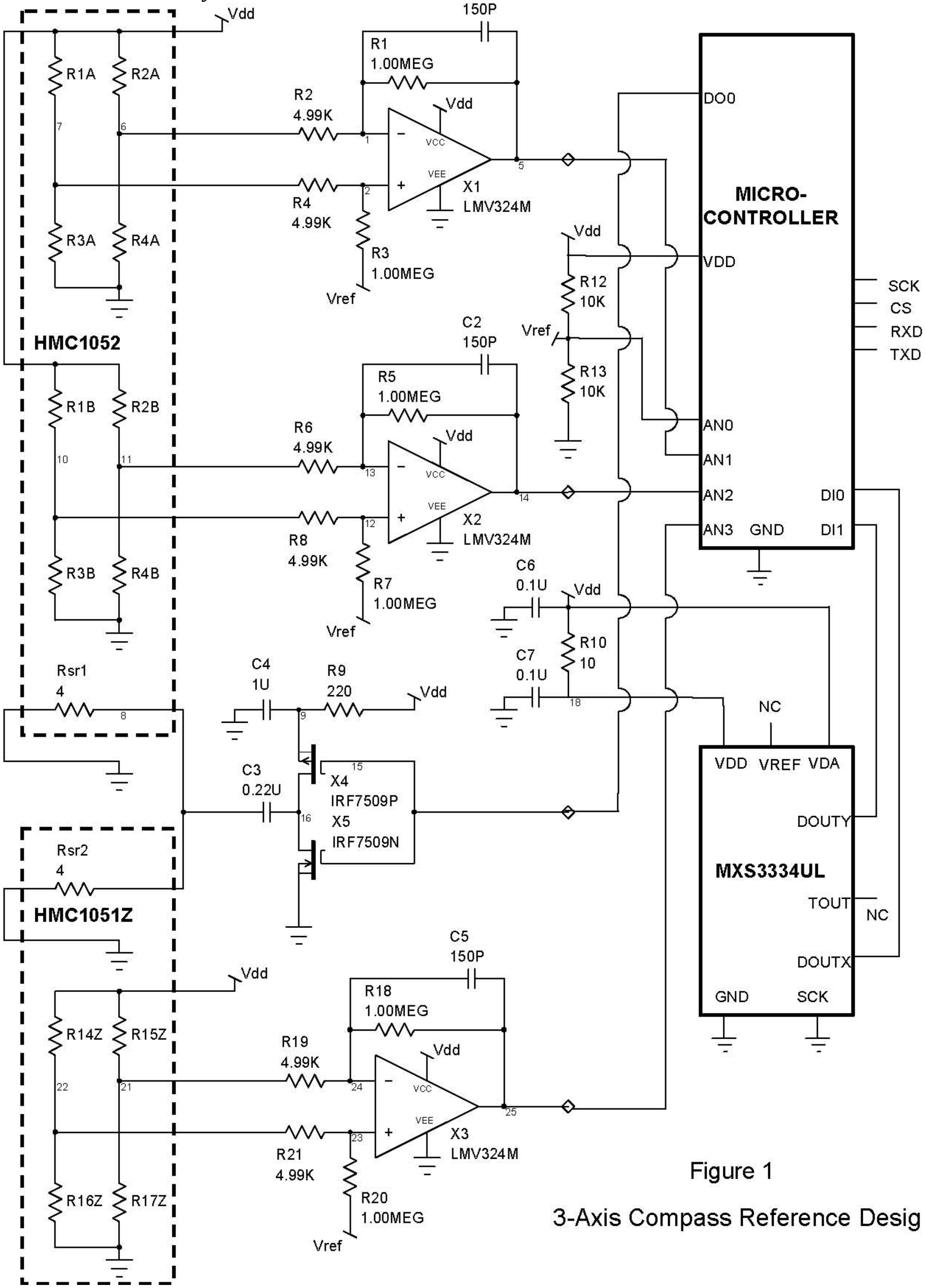


Figure 1

3-Axis Compass Reference Desig

GPS Receiver

As an optimal solution for my GPS needs seems to be the M12+ Oncore receiver manufactured by Motorola. This module communicates at 9600 baud using Motorola binary protocol. The outputting message includes: Latitude, longitude, height, velocity, heading and time.

Processing Unit:

I'm going to use an FPGA board for the processing unit. Here, I'll analyze the incoming data, store needed values in the memory and output values on a LCD display. I've selected Digilab 2E board manufactured by Digilent, Inc. This board is available at a reasonable price, features a 200K-gate Xilinx Spartan 2E XC2S200E FPGA in a PQ208 package that provides 143 user I/Os. Xilinx provides full development software free of charge and the board ships with a serial cable and power supply. I'm going to use Verilog to program the FPGA.

There are eight values I want to display for the pilot:

- Latitude
- Longitude
- GPS velocity
- Average vertical speed
- Optimal speed to destination
- Distance to destination
- Direction of wind
- Strength of wind

Eight LED displays would raise a power consumption issue so I've decided to use a LCD display instead. Reasonable seems to be LCM-S02004DSF manufactured by Lumex. This is a 4-line display with 20 characters on each line.

Plane characteristics will be downloadable through RS232 port provided by the FPGA board and will be stored in a EEPROM memory.

Power Supply:

Since most gliders are equipped with a 12-volt battery for their radios, I'm going to include a power adapter that will convert 12 volts into 5 and 3 volts. The adapter is going to use standard 7805 and LM317 power regulators.

Debugging Process:

Each of the proposed modules will be debugged in the lab environment on its own and later on, the complete system will be tested and corrected. Programs running on the Motorola 68HC12 microcontrollers will be simulated in a simulator, which is downloadable, free of charge on the Internet. I became familiar with this simulator in ECE3720 class. When the system is complete, I'll install the sensors in my car and drive in the mountains to make up altitude changes and vertical speed. Even some of the calculations might be tested in a car, such as optimal speed to destination or distance to destination. Those calculated values could be easily verified with a mechanical ruler that pilots use.

Risks:

In any kind of project, there is always the risk of not getting parts on time or not being able to get them in small quantities at all. To eliminate this risk, I'm going to order all crucial parts as soon as possible and I've also researched multiple distributors in case there is a problem with one.

Another problem I could encounter is with implementing the electronic compass. Even though, I've spent many hours searching for some practical experiences with building an accurate magneto-resistive compass, I wasn't able to find any resources that would ensure me that the recommended manufacturer's design will work. To minimize this problem, I'm going to start working on the compass during summer and I would like to get it completely done and tested by the end of July.

Also, because of the nature of this project, some functions will have to be tested during an actual flight. I'll have to find a person willing to accept responsibility for such test flights. I'm going to do a lot of testing in the lab and in my car so the amount of testing and possible corrections during those flights is minimum.

Interface Specifications and BOM:

User Interface:

There are going to be several buttons for the pilot to control the device.

- Power ON/OFF
- Altimeter mode selector
- Altimeter reset
- Next destination selection

➢ Plane characteristics will be downloadable through RS232 interface and will be done before the device is installed in a plane.

➢ List of destinations will be stored in a flash memory chip before the plane takes off. This will eliminate the need for a keypad and will give the pilot the comfort of entering the data on a PC.

➢ Flight data for referees will be stored on the same memory chip during the flight.

Bill of Materials

		part number	**lead time**	**unit cost**	**quantity**	**total cost**
Part:	**Motorola MPXM2102A**					
Prim. Dist.	Arrow	MPXM2102A	in stock	$5.76	$1.00	$5.76
Sec. Dist.	Digi-key	MPXM2102A-ND	in stock	$7.90	$1.00	$7.90
Part:	**Motorola MPXM2102D**					
Prim. Dist.	Digi-key	MPXM2102D-ND	in stock	$7.90	$1.00	$7.90
Sec. Dist.	Arrow	MPXM2102D	12 weeks	$5.76	$1.00	$5.76

Part:	**Motorola 68HC912**					
Prim. Dist.	Arrow	MC68HC912B32CFU8	in-stock	$15.99	$2.00	$31.98
Sec. Dist.	Digi-key	MC68HC912B32CFU8-ND	in stock	$19.26	$2.00	$38.52
Part:	**Honeywell HMC1055**					
Prim. Dist.	Digi-key	342-1036-ND	in stock	$60.00	$1.00	$60.00
Sec. Dist.	Honeywell	MHC1055	in-stock	$33.40	$3.00	$100.20
Part:	**Motorola M12+ Oncore**					
Prim. Dist.	synergy-gps	P283T12N1x	in stock	$70.00	$1.00	$70.00
Part:	**Active GPS antenna**					
Prim. Dist.	synergy-gps	GC3LP279CA	4-6 weeks	$30.00	$1.00	$30.00
Part:	**FPGA D2E**					
Prim. Dist.	Digilent, Inc	D2E	in-stock	$109.00	$1.00	$109.00
Sec. Dist.	DSL lab		in stock	$0.00	$1.00	$0.00
Part:	**Lumex LCM-S02004DSF**					
Prim. Dist.	Mouser	696-LCM-S02004DSF	in-stock	$31.50	$1.00	$31.50
Sec. Dist.	Digi-key	67-1763-ND	in stock	$31.50	$1.00	$31.50
					Total:	$346.14

Distributor List:

Arrow Electronics	http://www.arrow.com/
Digi-key	http://www.digikey.com/
Mouser Electronics	http://www.mouser.com/
Digilent	http://www.digilent.us/
Honeywell	http://www.honeywell.us/

Initial Schedule Flow:

Summer 2004:

- Final research and order essential parts (pressure sensors, mag. field sensor, pitot tube, GPS module, GPS antenna, FPGA board, processors, displays, memory chips)
- Getting familiar with FPGA programming environment, GPS communication protocol and LCD display control.

Fall 2004

September 2004: Creating schematics and PCBs for electronic compass, power supply and altitude/speed modules. Getting SMT parts soldered on circuit boards, building and setting up power supply. Designing algorithms for Motorola 68HC12 and familiarizing with EEPROM programming using S-records. Measuring output signals from sensors to verify functionality.

Milestone: PCBs fabricated, power supply completed and tested. SMT parts soldered on.

October 2004: Finalizing algorithms, assembly coding and debugging algorithms for altitude/speed module, testing the altitude/speed module.

Milestone: Altitude/speed module debugged and tested.

November 2004: Finalizing user interface, output interface and creating algorithms for flight data calculations. Defining file formats for plane and flight characteristics.

Milestone: Being able to download and store plane characteristics in EEPROM.

December 2004: Verilog coding – LCD display control, registers for receiving data from altitude/speed and compass modules

Milestone: Being able to display values on LCD display.

Spring 2005

January 2005: Verilog coding - Communication with GPS, storing flight characteristics in flash memory, reading destinations from flash memory.

Milestone: Being able to receive data from GPS, store them in flash in specified time intervals and user should be able to select destination points.

February 2005: Verilog coding/debugging algorithms for calculating average vertical speed, distance to destination, direction/strength of wind and optimal speed to destination.

Milestone: All calculations tested with values generated in the lab.

March 2005: Assembling the whole system and testing

Milestone: Complete system working

April 2005: Solving problems

May 2005: Testing and demonstration

References:

[1] http://www.aoe.vt.edu/~lutze/AOE3104/airspeed.pdf

[2] http://www.continuo.com/marske/articles/airspeed%20calibration/airspeed%20calibration%20article%20jan%202003.htm

[3] http://go.phpwebhosting.com/~khorton/rv8/phplinks/index.php?&PID=48

[4] http://www.rocznik.de/Marko/electronic/project/vario/vario_engll.htm

[5] http://www.ssec.honeywell.com/magnetic/datasheets/lowcost.pdf

[6] http://www.ssec.honeywell.com/magnetic/datasheets/hmc1055.pdf

[7] http://www.synergy-gps.com/Mot_Manuals.html

[8] Embedded Microcomputer Systems: Real Time Interfacing

[9] http://e-www.motorola.com/files/microcontrollers/doc/ref_manual/CPU12RM.pdf