GLIDER AVIONICS PACKAGE

A Thesis

Presented to

the Faculty of the Department of Computer Science

University of Utah

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Engineering

By

Ondrej Novak

May, 2005

ABSTRACT

The purpose of this project is to design a complete package of instruments that will serve the needs of glider pilots. This package should include instruments essential for safe sailing such as indicators of airspeed, altitude, vertical speed and a compass. Additionally, it should include instruments for the pilot's convenience such as GPS and provide calculations to ease the pilot's decisions during competitive flying. Moreover, it should store crucial flight data for competition referees and for later analysis.

The device I've created is designed as an embedded system with a single controller, which provides simple user interface and high flexibility at a low price. The input data for flight status information is collected by two silicon pressure sensors attached to a standard pitot tube accessible in all gliders. The compass uses 3-axis magnetoresistive sensors and a 2-axis accelerometer for tilt compensation. GPS data is provided by a commercial M12+ receiver and processed along with flight data in a Freescale MC68HC912B controller. This affordable device utilizes the most competitive instruments on the market today, ensuring optimal performance when in flight.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank Al Davis, who was my advisor since the beginning of this project, for his strong encouragement and guidance. He has been very patient with me through the entire experience and been a source of great support and inspiration. This project has benefited greatly from the numerous meetings I have had with him. I have learned a lot from his style of research and the clarity with which he attacks problems. I would also like to thank Mike Kirby who helped me with algorithms and computational methods and Corey Hatch who provided me with free National Semiconductor parts and an FPGA board for testing and development.

CHAPTER 1

INTRODUCTION

Flying has always been extraordinarily attractive to humans. Historically, people were trying to imitate birds for hundred of years before the Wright Brothers successfully flew for the first time in 1903. Since that time, we have developed aircrafts for a variety of purposes and flying became an ordinary part of human life. Besides providing fast transportation for people and cargo, airplanes are used for carrying bombs and spying on the enemy. However, there are also a lot of people who like to fly for fun. Those people like to feel free like birds and enjoy the beauty of nature.

Probably the kind of flying that is closest to that of birds is flying gliders. The experience isn't disturbed by any noise from engines and the pilot can really enjoy being free. To be able to stay in the air, the pilot has to be familiar with natural processes such as convection, aerodynamic laws, weather prediction or navigation. As usual, with any kind of discipline, people like to compete and glider pilots are no exception. There are many events every year when the flying community gets together and competes for prizes and entertainment. The most common kind of competition is when pilots get assigned a route and they need to complete the route in the shortest time possible without landing. The purpose of this project was to design an electronic device that would provide the pilots with all the essential flight information and compute some quantities that would be useful to the pilot to make crucial decisions during the competitive flying.

For safety, there are several crucial quantities that need to be measured. Airspeed, for example, is extremely important. Every aircraft has its minimum and maximum airspeed. Flying below the minimal speed will cause the pilot to lose control of the plane. Flying over the maximum speed will most likely damage the aircraft and could cause loss of control as well. Another example of an important quantity is altitude. Altitude above the ground is important to the pilot to accurately guess how far he/she can fly. Also, it is important to know the altitude to avoid obstacles like mountains or man-made objects like signal transmitters.

As I already mentioned, knowledge of physical laws is important for glider pilots. Since there is no other source of energy than the surrounding air, the pilot has to be able to take advantage of those natural laws. Convection is the most important. The pilot has to be able to identify streams of up-going air that eventually ends up forming clouds. Flying in circles in those streams provides the necessary energy in the form of gained altitude. It is not easy to identify such air lifts and the strength of those lifts varies as well. For this purpose, there is an indicator of vertical speed that tells the pilot whether he or she is ascending or descending and at what rate.

A compass is another instrument essential to flying. Pilots need to know what direction to fly to get to the desired destination. Along with a compass, there are a few other quantities that are very helpful when navigating. Global Positioning System (GPS) is an easy solution to obtain all the information for easy navigation such as current position, altitude above sea level, heading, date, universal time (UTC) and ground speed.

During competitive flying pilots have to make decisions based on the current weather and current position. Most of the weather information is very subjective and it is impossible to develop a device that would make the decision for the pilot. It is possible however to collect some data that would make it easier to decide. For example, it is useful to know the average strength of the air lifts during the day. If, for example, the average rate is 3 m/s, then the pilot can decide not to waste time with a 1.5-m/s lift because it is almost certain that he/she will find a better lift down the road.

Also, the speed at which the plane is flying determines drag and consequently the distance the plane can fly from a given altitude. There are methods to compute the optimal speed to destination and such an indicator can be very useful at the last stage of a competitive flight.

In summary, the main goal of this project 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 the 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 these 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 a 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.

CHAPTER 2

PRINCIPLES

2.1 Indicator of Airspeed

Aircrafts usually display the Indicated Airspeed (IAS). IAS will differ from the true airspeed (TAS) at air densities other than some reference density. Air density is affected by temperature, atmospheric moisture content (humidity, and pressure altitude). Due to the lower air density at higher altitude, the airspeed indicator reads lower than true airspeed when we fly at higher altitudes. Such an error is negligible in glider application because most gliders don't have pressurized cabins to reach altitudes over 4000 meters.

The airspeed indicator is a calibrated pressure gauge that responds to the difference in two pressures. One pressure, the total pressure, is obtained by a device called the pitot tube (see Figure 1). The second pressure, called static pressure, is taken from a carefully positioned static port where the pressure does not vary with aircraft speed or attitude, but only with altitude. Such static port is usually realized as a set of holes on the side of the pitot tube.

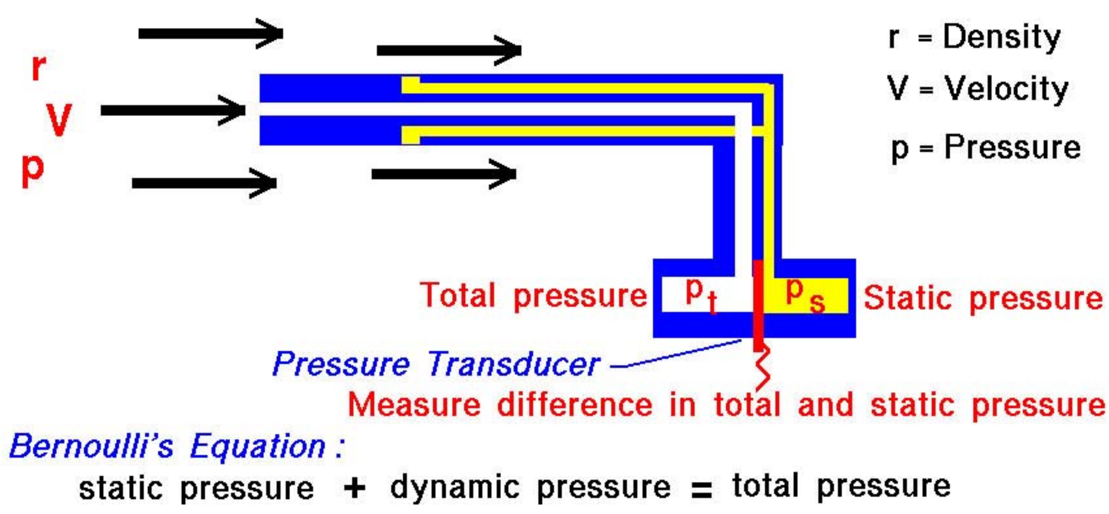

Figure 1 - Pitot tube principles

2.2 Indicator of Altitude (Altimeter)

The traditional altimeter found in most aircrafts works by measuring the air pressure from a static port in the airplane. Static pressure decreases with altitude – about 1 hectopascal per 27 feet close to sea level. The altimeter is calibrated to indicate altitude directly in accordance with the mathematical standard defined by the International Standard Atmosphere.

Generally, there are two ways to indicate altitude. The regional static pressure at mean sea level is called QNH and the static pressure which will calibrate the altimeter to show the height above ground at a given airfield is called QFE. The user should be allowed to select what mode he/she wants to use. Also, since the sea level air pressure varies with the weather, the user has to be able to adjust the reference pressure. An altimeter however, cannot be adjusted for variations in air temperature. Difference in temperature from the ISA model will therefore cause an error in indicated altitude.

2.3 Indicator of Vertical Speed

The indicator of vertical speed is also called a variometer. This instrument is used to inform the pilot of the rate of descent or ascent. In the simple mechanical form, the instrument consists of an air bottle connected to the external atmosphere by a small tube. When the airplane moves up or down in the atmosphere, the pressure inside the air bottle changes to match the external air pressure. This causes the air to move through the tube. The faster the aircraft is ascending or descending, the faster the air flows. The variometer simply measures and displays the direction and speed of the airflow in the tube. In its electronics form, all I need to do is to read the current altitude at time t1, wait some period of time and read the altitude again. The vertical speed then is:

$$vertical\ speed = \frac{A2 - A1}{t2 - t1}$$

Where A1 is altitude at time t1

A2 is altitude at time t2

2.4 Compass

In general, a compass evaluates the Earth's magnetic field. Figure 2 gives an illustration of the field's shape.

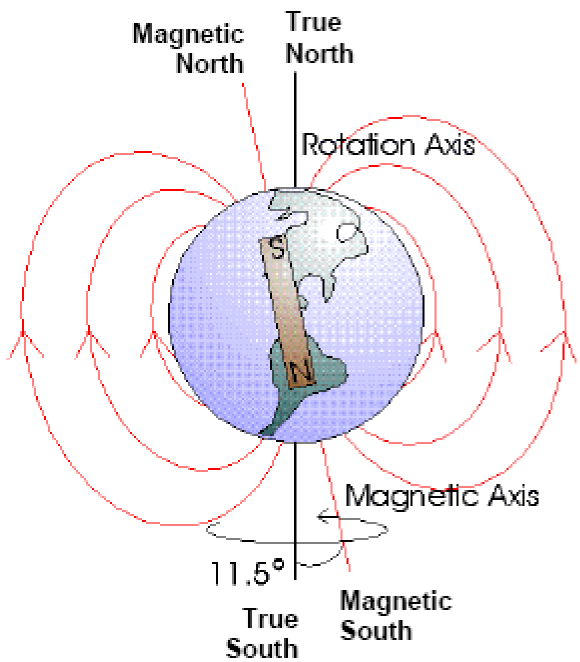

Figure 2 - Earth's magnetic field

Hypothetically, to compute the heading of an airplane, I would have to accurately measure the magnetic field in only two directions; x and y. If the two values are known then:

$$Azimuth\,(Heading) = \arctan \frac{Y}{X}$$

Where X and Y are magnetic field magnitudes
in x and y directions respectively

This simple computation is sufficient only in a very limited number of applications. For application onboard a plane, a more complicated model has to be considered. The reason why the simple model is not sufficient is its sensitivity to tilting. Such a compass would work fine only until it is tilted. The error increases with the tilt and such a device becomes useless even with small magnitudes of tilt. Figure 3 demonstrates the error.

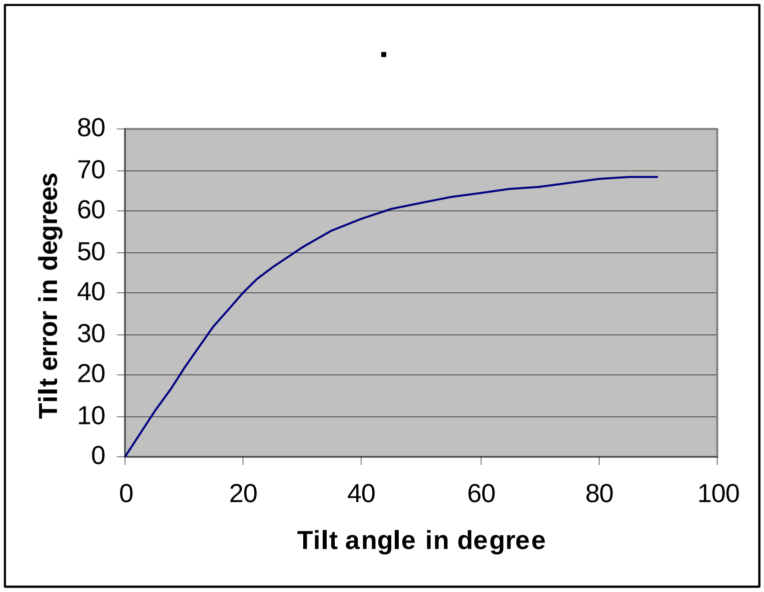

Figure 3 - Tilt error magnitudes for Hamburg. Inclination = 68 deg.

It is obvious from Figure 3 that a 2-dimensional compass is useless for avionics applications since airplanes operate under large angles most of the time.

It is possible to compensate for the error caused by tilting if the magnitude of the magnetic field is know in all three dimensions. It is also necessary to obtain information about the pitch and roll under which the compass is operating. The definition of pitch and roll is pictured in Figure 4:

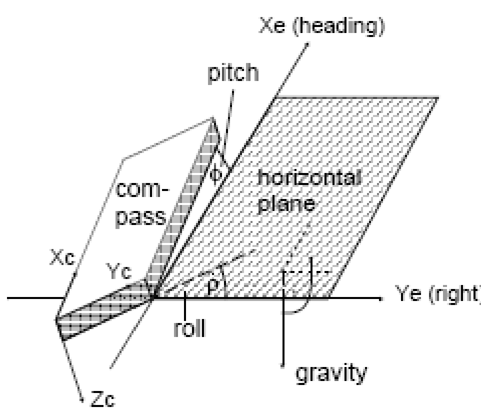

Figure 4 - Definition of pitch and roll

As soon as we have the required information, the correction can be computed in the following way. The equations:

$$X' = X * \cos(\phi) + Y * \sin(\theta) * \sin(\phi) - Z * \cos(\theta) * \sin(\phi)$$
$$Y' = Y * \cos(\phi) + Z * \sin(\theta)$$

create tilt compensated X and Y magnetic vectors (X’,Y’) from the raw X,Y, and Z magnetic sensor inputs plus the pitch (ϕ) and roll (θ) angles. Once X’ and Y’ are computed, the compass heading can be computed by the equation:

$$Azimuth\,(Heading) = \arctan(\frac{Y'}{X'})$$

2.5 GPS

The Global Positioning System (GPS) is widely used for all kinds of applications today. Its functionality is extremely complex and is based on computing the distance to at least four satellites from the difference between local time and the time the satellite signals were sent. Because of this massive deployment of this technology, the receivers have become affordable and fairly simple to use. For my project, I’m going to take advantage of the affordability and will employ a commercial receiver that will be connected to a CPU where the data will be evaluated and displayed. The data I’m interested in is the current latitude and longitude, universal time, date, elevation, heading and speed relative to the ground.

2.6 Indicator of Average Speed

It is not to the pilot’s advantage to keep track of the descending rate during the day as he/she is constantly searching for up-going air lifts to gain potential energy. It is important however to know the average rate at which he/she was able to gain altitude. During competitive flying, it is important to be fast. The information about the average rate will help to decide whether or not it is worth staying in the lift or if it is more efficient to continue the flight and hope for a stronger lift later on.

Since the proposed device will process signals in digital form, to implement the average vertical speed indicator, all that needs to be done is to keep track of the ascending rate in certain periods of time.

2.7 Indicator of Optimal Speed

This indicator becomes very useful at the very last part of a competitive flight. When the pilot is approaching the final destination, he or she has to decide what altitude is sufficient to reach the destination. Of course, the speed at which the destination is approached plays an important role in the decision making process. Since this is a competitive flight, higher speed is desirable. On the other hand, flying at higher speeds means more friction and higher descent rates. There is a method that pilots currently use to decide at what velocity they should approach the destination to take maximum advantage of their current altitude. Figure 5 shows the most common device to provide the computation.

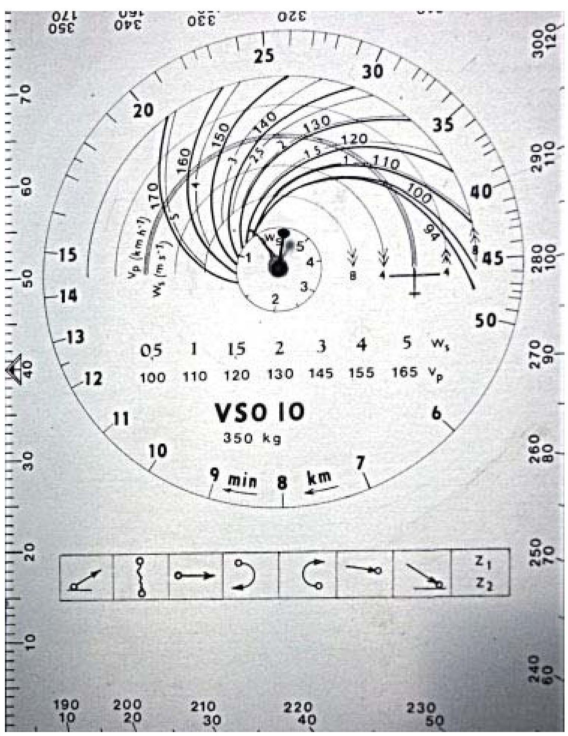

Figure 5 - Device used to compute optimal speed

This device is based on plane performance characteristics provided by the manufacturer and my goal was to include this device in electronics form in the proposed instrument package.

Provided the plane characteristics, the device should be able to evaluate data collected from altimeter, wind indicator and GPS. It should compute the distance to destination, and with the information about altitude and wind, it should find and display the optimal speed at which the pilot can safely reach the destination. The pilot then can decide whether or not he or she should spend some time to find a lift, gain some altitude and come home faster or just continue the flight at the given speed.

2.8 Indicator of Current Wind

From the previous paragraph, it is obvious that wind has a significant influence on the optimal speed computation. Information about wind then becomes crucial for competitive flying. Currently, pilots obtain such information by using radios and asking the ground personnel who have the ability to detect the wind's direction and strength. Such information can be misleading since wind varies significantly with distance and elevation. Ideally, there should be a device that could detect and compute the wind at the current position of the plane. It is very tricky, however, to do such a thing. I believe however, that with the set of instruments I proposed for this package, I should be able to provide the pilot with reasonably accurate values.

My idea was to use the GPS, compass, and geometry to compute the information about wind. Figure 6 provides a graphical form of the method I'm going to use.

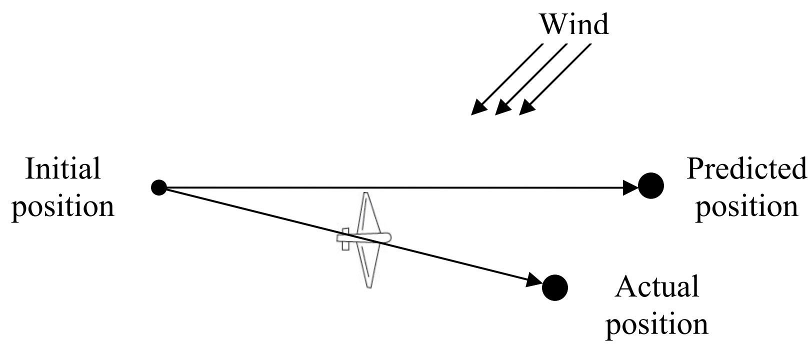

Figure 6 - Wind computation

The computational method is described in Chapter 5.7.

CHAPTER 3

PART SELECTION

3.1 Pressure Sensors

3.1.1 Airspeed Indicator

As I described in the previous chapter, for measuring airspeed and altitude, I need two pressure sensors. As the first task in their selection, I had to find out what range of pressures I was going to be working with. I provide a case study that gives the answers. Before I present the example I would like to recall that the maximum allowed speed for most gliders is about 250 km/h and the formula for dynamic pressure computation is:

$$dynamic\,pressure = \frac{rV^2}{2}$$

Where: r is air density and
V is velocity

Case study:

A plane cruising at 300 meters, airspeed 69 m/s (250 km/h), air density 1.212 kg/m³. The International Standard Atmosphere (ISA) defines atmospheric pressure at 300 meters to be about 1000 hPa.

$$\text{static pressure} = 1000\ \text{hPa} = 14.5\ \text{psi}$$
$$\text{dynamic pressure} = \tfrac{1}{2}rV^2 = \tfrac{1}{2} \times 1.212 \times 69 \times 69 = 2885\ \text{N/m}^2 = 28.85\ \text{hPa} = 0.42\ \text{psi}$$
$$\text{total pressure} = \text{static} + \text{dynamic} = 1000 + 28.8 = 1028.8\ \text{hPa} = 15\ \text{psi}$$

From this study I found out that for the speed indicator, I need a differential pressure sensor with a range between 0 and 0.42 psi. Since I was planning on digitizing the output signal with an analog-to-digital converter (ADC), it would be ideal if the output voltage was in the range between 0 and 5 volts. An ideal candidate seemed to be MPXV5004G series sensor, manufactured by Motorola. This is a differential silicon pressure sensor with the operation range between 0 and 0.65 psi and output voltage between 1 and 4.9 volts. When selecting this sensor, I was also considering its price.

3.1.2 Altimeter

From the previous case study I found out that the maximum atmospheric pressure will be around 14.5 psi. I had the same requirements for the output voltage and that it be in a reasonable price range. An ideal candidate seemed to be a MPX4101A series sensor also made by Motorola. This sensor is designed for measuring atmospheric pressure with an output voltage between 0.25 and 4.95 volts. The sensor is also calibrated and temperature compensated.

3.2 Magnetic Field Sensors

Since my application requires measuring the Earth's magnetic field in all three dimensions and I also need information about the tilt and roll, I was looking for a package that would meet all those needs. I was excited when I found out that Honeywell offers a set of sensors that are designed exactly for such a use. The catalog number is HMC1055 and this package includes the following:

- 2-axis magnetoresistive sensor for X-Y axis Earth's field detection
- 1-axis magnetoresistive sensor for Z-axis Earth's field detection
- 2-axis accelerometer for 60º tilt compensation
- 2.2 to 5.5 volt power supply range
- 3-axis compass reference design

The 2-axis accelerometer generates two outputs square-wave signals whose duty cycle corresponds to the actual tilt and roll.

3.3 GPS Receiver

My requirements for the GPS receiver were pretty simple. It had to be reasonably small, dependable, power efficient and affordable. It was also important that I should be able to easily connect this GPS to the controller where I could receive and process the desired data. After some research, I decided to use an OnCore M12+ receiver made by Motorola. This receiver provides all the information I was looking for, such as precise position, heading, ground speed, time and elevation. It requires a 3-volt power supply and an external antenna. The communication is done through a serial port using Motorola's binary protocol which is fairly easy to learn. For the external antenna requirement, I followed the manufacturer's recommendation and purchased an active HAWK antenna.

3.4 Controller

Looking at the selected parts and components, I created the following list of requirements that I had to meet when selecting the central processing unit (CPU).

- 6-channel ADC (at least 10-bit precision)
- 16-bit data path
- Input capture
- Output compare
- Serial interface

The 10-bit precision requirement on the ADC came from the fact that I would like to have the compass with a 1 degree precision. Also, studying some related work done by other designers, a minimum of 10-bit ADC was recommended for the altimeter as well.

An important factor was also availability of development tools. Because of my precious experience with programming Motorola 68HC11, I decided to stay with the same architecture. I decided to use a Motorola 68HC12 series controller which is just a 16-bit version of 68HC11. This controller meets all my requirements and for a reasonable price, I was able to buy an evaluation board which was easy to program.

3.5 User Interface

I decided to have two different ways of displaying values. The basic set of instruments that includes altimeter, air speed indicator, vertical speed indicator and compass should be easily visible to the user. Ideal for this purpose seemed to be using four LED displays. The other set of instruments that are mostly for the pilot's convenience, will be displayed on an LCD display. LCD displays generally require less power but it is sometimes hard to see the values under different light conditions.

For the compass, airspeed, and vertical speed indictor, I'll need 3-digit displays since the displayed values are never going to exceed 999 for either of them. The altimeter which I decided to implement in meters will need four digits since it is possible to reach altitudes in the magnitude of thousands of meters but always less than 10,000 meters. Even though I want the displays to be easily visible, I also need to take the energy consumption into account. I selected the smallest possible displays from Digikey's catalog.

For the LCD display I wanted to be able to display at lest 8 values. It seemed reasonable to fit two values per line on a 4-line display. Those displays are pretty affordable even with a backlight. I selected one from Digikey's catalog.

Besides displaying values, the user needs to be able to change some settings or modes. I used Digikey's catalog again to select a power on/off switch, reset button and a digital decoder for atmospheric pressure adjustments.

CHAPTER 4

DESIGN DETAILS

Figure 7 shows a high level block diagram of the proposed device.

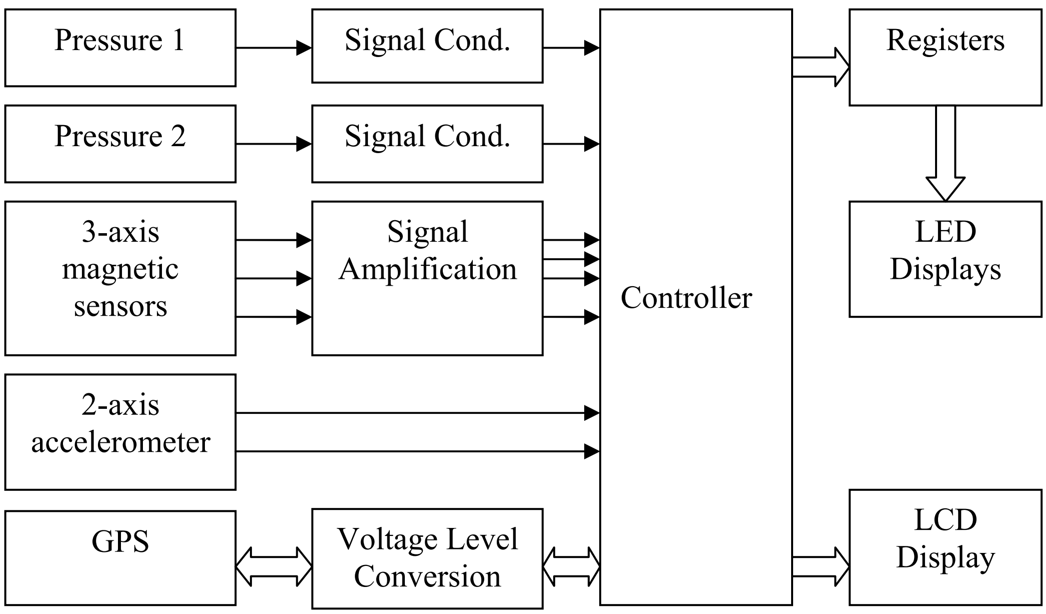

Figure 7 - Device block diagram

This block diagram doesn’t include a power supply which is needed for the device to operate. I’m going to describe such power supply first and then I’ll describe the design of each block.

4.1 Power Supply

In most gliders, there is a 12-volt lead-acid battery to power up the on-board radio. I was planning to use this battery for my device but I had to build a regulator to reduce the voltage to levels needed for my device. Most of the logic, including the CPU requires a 5-volt power supply to operate and the GPS receiver requires only 3 volts. The proposed regulator then had to provide at least those two voltages. Since this proposed device was going to be the first prototype, I also included a 12-volt regulator on the circuit board to make it convenient to program the CPU's EEPROM. Of course the 12-volt regulator wouldn't be included on the circuit board for the final version of the device. Also, for the first prototype I doubled the 5-volt regulator to provide higher current if needed. Again, this wouldn't be necessary for the final circuit board design.

For the 5-volt power supply I used a standard MC7805 regulator using recommended manufacturer's circuitry. See figure 8 for details.

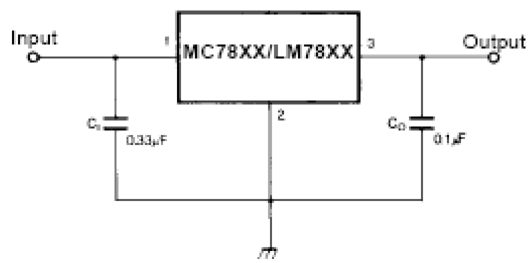

Figure 8 - 5-volt power supply

For the 12 and 3-volt power supplies, I used LM317 (LM117) based adjustable regulators hooked up as recommended by the manufacturer. See Figure 9 for details.

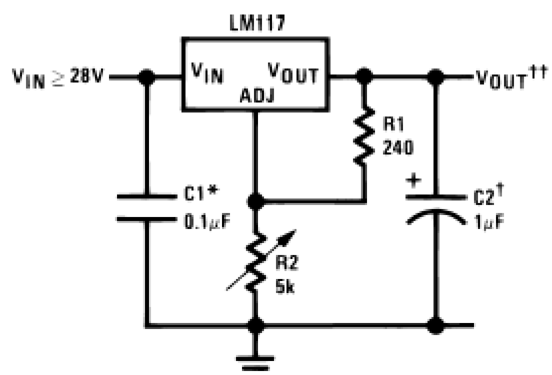

Figure 9 - 1.2 - 12-volt adjustable power supply

The regulators are represented by part numbers U31, U32, U33, U34, and U35 on the schematics in Chapter 6. Parts number U38, U39, U40, U41, U42, and U43 are connectors for easy connection of other modules. The voltage levels were adjusted to the required values by using potentiometers R11, R12, and R13.

4.2 Pressure Sensors

The sensor selection was described in Chapter 3. In the block diagram in Figure 7, they are symbolized as Pressure 1 and Pressure 2 and are followed by two low-pass filters. These filters are included for signal conditioning and should reject high frequency signals such as noises that could be introduced by various sources. For simplicity, I decided to use a Butterworth low-pass filter of order 2 with gain equal to 1. The following figure shows the schematics:

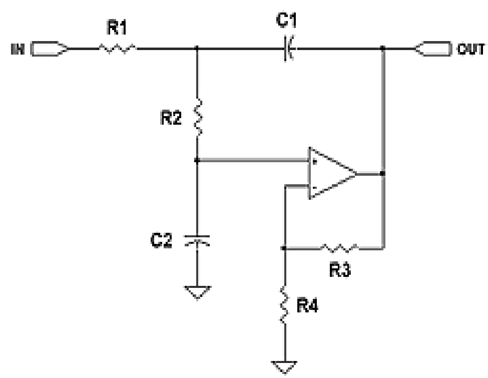

Figure 10 - Butterworth filter schematics

To find out the values of resistors and capacitors, I used an interactive design tool provided by Analog Devices on their official web page. For the operational amplifier, I chose to use an LM324 for its availability and low price. In the schematics in Appendix A, the pressure sensors have symbols U25 and U26 and the filters are built of one LM324 package – U44.

The conditioned analog signals are connected directly to the ADC inputs on the Motorola controller.

4.3 Compass

The compass is built of three blocks in the block diagram. One block consists of three magnetic field sensors, followed by signal amplifiers. These amplifiers are needed because the Earth's magnetic field is very weak and the expected output from the sensors is in the range of ±5 mV. I decided to use a reference schematics that was shipped with the sensor set I purchased. The reference schematic is shown in Figure 11.

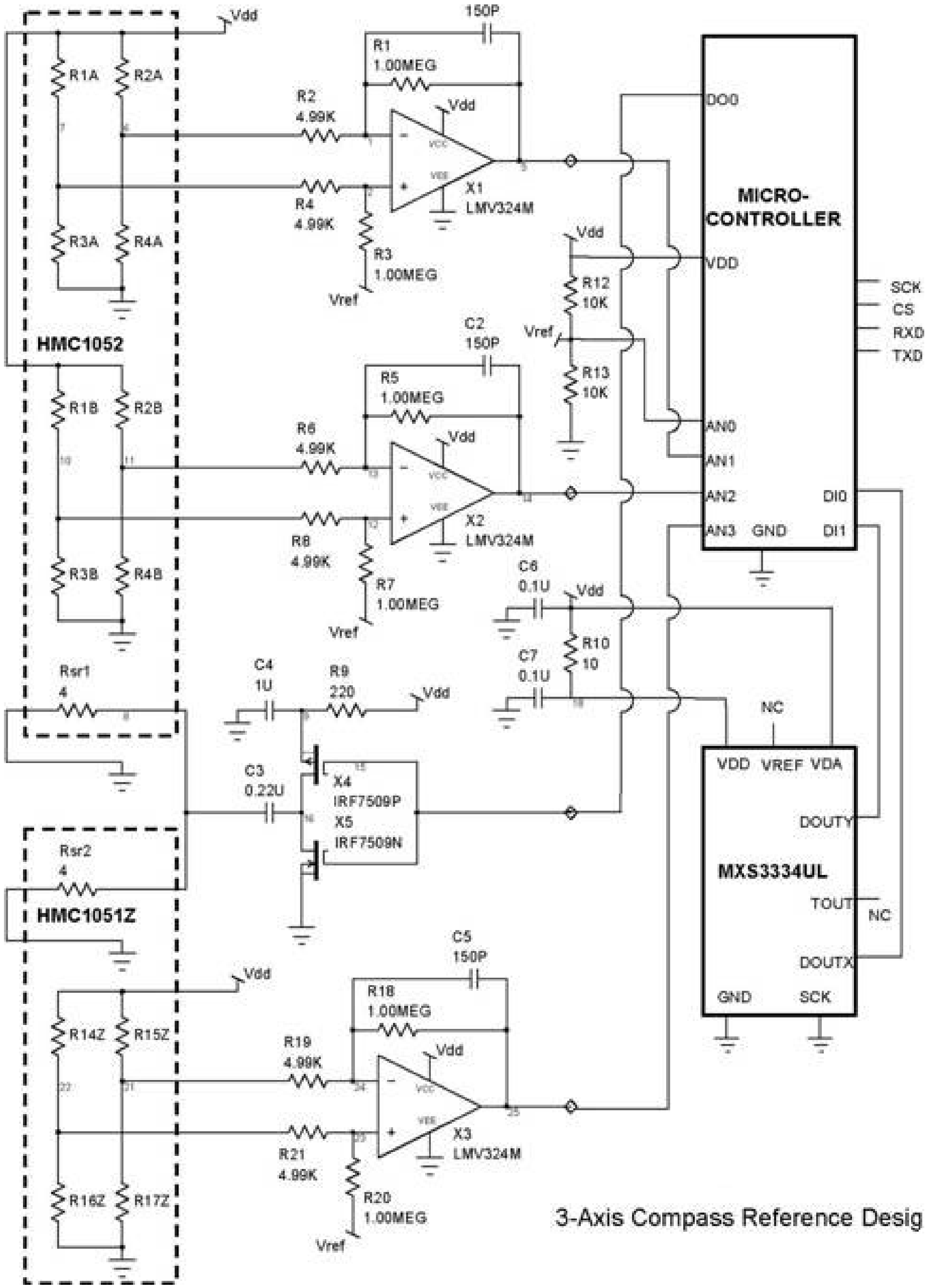

Figure 11 - Compass reference design

The output signal in each direction X, Y, and Z is amplified by an operational amplifier with gain of about 200. If, for example, the output in X direction is 2.8 mV, it is then amplified by the factor of 200 and referenced to a voltage of 2.5 V. The resulting signal has a value of $2.8e-3*200+2.5=\underline{\underline{3.06V}}$, which is then directly connected to an ADC channel on the Motorola controller. The reason why the signal is referenced to 2.5 volts (half of the power supply) is that each of the magnetic sensors is basically a Wheatstone bridge with an output voltage of ±5 mV. Since I'm using an ADC that is not capable of digitizing negative voltages, half of the power supply is added to make all the values positive.

The reference design in Figure 11 also includes the third block in my block diagram that completes the compass. This block includes a 2-axis accelerometer (MXS3334UL) whose outputs are directly connected to two input-compare inputs on the Motorola controller. The two accelerometer outputs generate square wave signals of 100 Hz with 50 percent duty cycle in leveled position. However, if the sensor is tilted, the duty cycle will change reflecting the tilt.

4.4 GPS

The GPS receiver in the block diagram is followed by a voltage level converter. The reason why is that the receiver communicates on 3-volt level logic but the Motorola controller is on 5-volt levels. I had to convert the signals in order to make communication possible. I took advantage of Motorola's evaluation board because it already has a converter to RS232 levels to communicate with a PC. All I had to do was to use one RS232 voltage converter made by Maxim which took care of the conversion. The manufacturer's recommended schematic is shown in Figure 12.

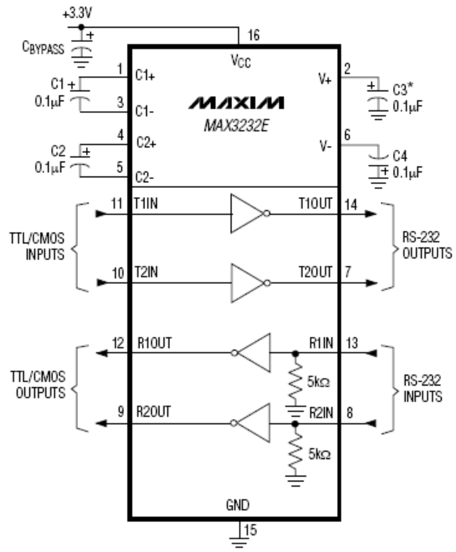

Figure 12 - Voltage level converter

4.5 User Interface

The block diagram shows the LCD display connected directly to the CPU. This is possible because the LCD module I purchased has its own controller and the communication protocol doesn't require any further circuitry. On the other hand, the LED displays will require some supporting hardware since the number of I/O ports available on the controller is limited. If I were going to use a dedicated I/O pin for each LED segment, with three 3-digit and one 4-digit displays, I would need 91 pins. Such a design would be ineffective, so I decided to use a register for each display and update the value at a sufficient rate so human eyes wouldn't notice flickering. In this design, I needed only 15 I/Os to serve all four displays. 4 bits were needed to select one of the 4 display registers, 4 bits were needed to select one digit within the display and 7 bits were used for the data. Figure 13 shows the design for one display:

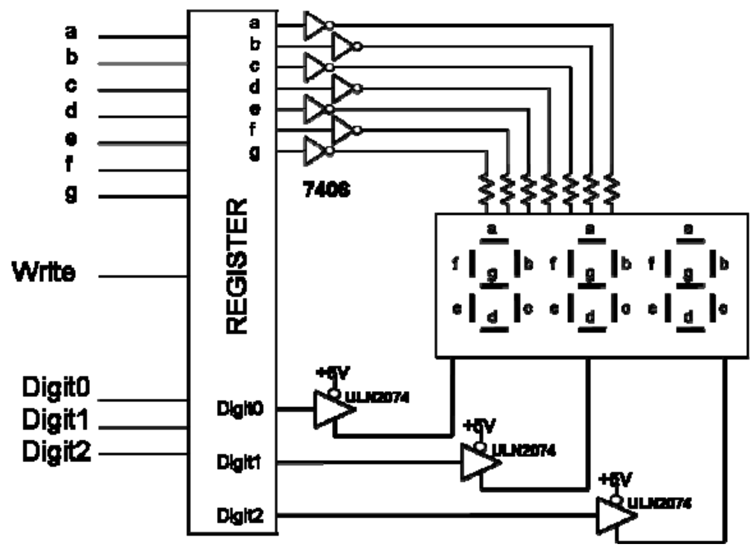

Figure 13 - Scanned LED display design

The registers are built of D flip flops and each digit is refreshed every 5 milliseconds. This means that the refresh rate is about 66 Hz, which is fast enough that human eyes don't see flickering.

The value of the resistors is 40Ω. From the manufacturer's datasheet, I found out that the LED segments operating point is 10 mA, 2V. Since the display was going to be scanned, I set the operation point to 30 mA for 33% duty cycle. Figure 14 shows the current through each segment.

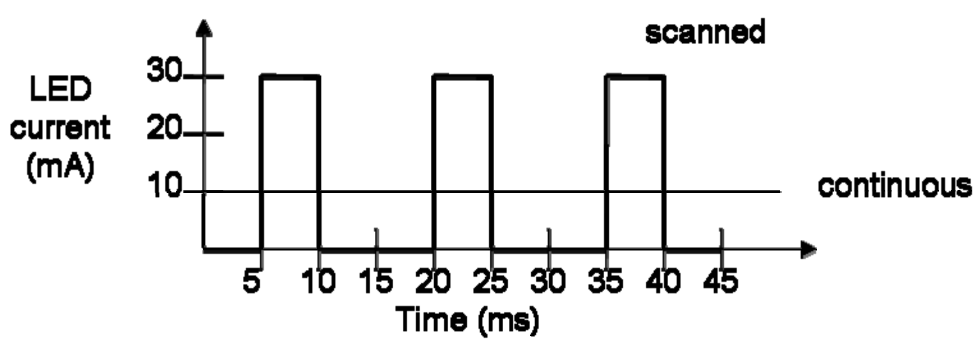

Figure 14 - Timing used to scan LEDs

$$R = \frac{5V - V_{OL} - 2V - V_{CE}}{30mA} = \frac{5 - 0.7 - 2 - 1.1}{30mA} = 40\Omega$$

Where V_{OL} is the voltage drop on a 7406 gate and

V_{CE} is the voltage drop on ULN2074

CHAPTER 5

ALGORITHMS

5.1 Altimeter

Finding out the current altitude is a straight forward process. After signal conditioning, the output voltage from the static pressure sensor is connected to one channel on the analog-digital converter (ADC). This analog signal is periodically digitized. To avoid small inaccuracies in readings, the average of 16 samples is computed. The result is then multiplied by 10 which gives the absolute altitude (QNH) in meters according to the international standard atmosphere (ISA). I figured out the constant 10 from the pressure sensor's datasheet and values defined by the ISA:

Table 1 - Altimeter pressure range

Altitude [m]	Pressure [kPa]	Output voltage [V]
0	101.32	4.7
5000	54.02	2.3

With a 10-bit ADC, the resolution is 5V / 1024 = 4.88 mV. From the table above, a change of 4.88 mV equals a change of 10 meters.

Most pilots however, prefer to see the altitude related to the airfield they took off from. To provide such information I store the elevation of the airfield in memory and subtract it from the QNH. The elevation of the airport is automatically measured when the device is turned on. It is also possible to make corrections by using a knob that is a part of the user interface. This knob changes the elevation stored in memory from the initial reading and so allows the user to adjust the altimeter in a way he/she desires.

5.2 Airspeed Indicator

Measuring the airspeed is even more straight-forward than altitude since there is no initial reading and no adjustments by the user are needed. The outputting signal from the differential pressure sensor is filtered and directly connected to an ADC channel. The signal is then periodically digitalized and the average of 16 samples is computed.

Table 2 - Airspeed pressure range, altitude = 300m

Speed [km/h]	Dynamic Pressure [kPa]	Output voltage [V]
0	0	1.25
250	2.8	3.80

From the table above, it is evident that a correction for the initial voltage of 1.25 volts has to be made. It is done by subtracting 256 from the measured 10-bit value. Also, from the above table, a change of 4.88 mV (change in the least significant bit) equals a change of 0.5 km/h. This means that the measured value, less 256 is then divided by 2 and displayed.

5.3 Vertical Speed Indicator

For this feature, I'm taking advantage of the fact that the GPS receiver reports the current position once every 1 second. The timing of the receiver is very precise so every time I receive a GPS message, I read the altitude indicator and subtract the reading I stored 1 second ago. This gives me the vertical speed in meters per second which is then displayed.

5.4 Compass

The compass is the most computationally intensive instrument in this package. I used an algorithm that was recommended by the magnetic sensor's manufacturer. There are 4 ADC channels used by the compass. Three of them are to measure the magnetic field in X, Y, and Z directions. One more is used to measure the reference voltage which is half of the power supply voltage. As explained in Chapter 4.3, this reference voltage is subtracted form the X, Y and Z readings to obtain positive and negative swings depending on the magnetic field's direction.

Besides obtaining information about the magnetic field, I also need to know the magnitude and direction of the tilt and provide error corrections. The outputs of the 2-directional accelerometer are directly connected to the controller's timer module. Using the input-capture feature, I accrue the total microcontroller clock cycles between two rising edges. Then I measure the number of cycles between a rising and falling edge and compute the duty cycle. 50% duty cycle equals zero pitch or roll. Figure 15 demonstrates the computation.

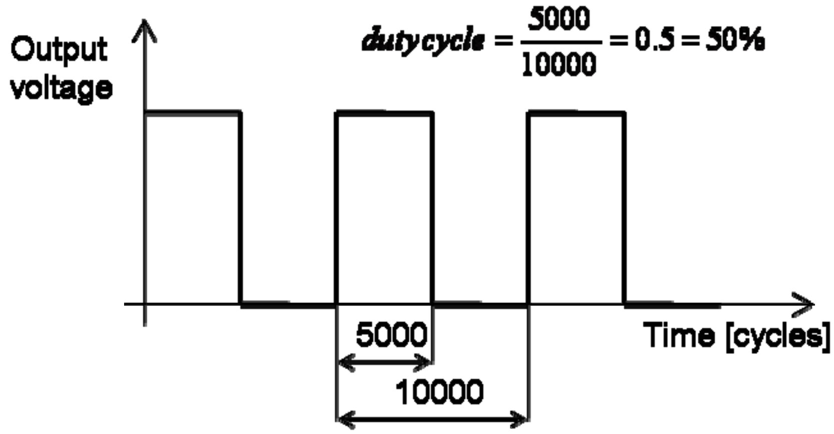

Figure 15 -Duty cycle computation

Once the duty cycle is measured for each axis output, these values can be compared to a memory mapped table, if the user desires the true pitch and roll angles. For example, if the pitch and roll data was to be known in one degree increments, a 90-point map would be created to match up gravitational values (sign independent) with corresponding degree indications. Because tilt-compensated compassing requires sine and cosine of the pitch and roll angles, the gravitational data is already formatted between zero and one and does not require further memory maps of trigonometric functions. The gravity angles for pitch and roll already fit the sine of the angles, and the cosines are just one minus the sine values (cosine = 1 – sine).

The equations:

$$X' = X * \cos(\phi) + \sin(\theta) * \sin(\phi) - Z * \cos(\theta) * \sin(\phi)$$
$$Y' = Y * \cos(\theta) + Z * \sin(\theta)$$

create tilt compensated X and Y magnetic vectors (X', Y') from the raw X, Y, and Z magnetic sensor inputs plus the pitch (ϕ) and roll (θ) angles. Once X' and Y' are computed, the compass heading can be computed by equation:

$$Azimuth(Heading) = \arctan\frac{Y'}{X'}$$

To compute the arc-tangent trigonometric function I used Taylor series:

$$\arctan(x) = x - \left(\frac{1}{3}\right)x^3 + \left(\frac{1}{5}\right)x^5 - \left(\frac{1}{7}\right)x^7 + ... = \sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}x^{2n+1}$$

The Motorola controller provides fractional division so no further computation was required. The computed value of azimuth is then displayed.

5.5 Average Vertical Speed

The average vertical speed is useful to the pilot only when the climb rate is included in the computation during the flight. The descent rate is not important as pilots need to gain altitude and not lose. Again, I will take advantage of the GPS timing and will create a counter that counts to 10. When the 10th GPS message is received (10 seconds) the vertical speed indicator is checked. If the value is positive, it is added to a register and kept. Also a register that keeps the number of samples is incremented by 1. Then the average is computed by dividing those two registers and the resulting value is displayed.

5.6 Optimal Speed Indicator

The optimal speed will be computed using information about the current wind, altitude and distance to destination. Chapter 5.7 describes the current wind computation and Chapter 5.1 describes altimeter.

The distance to destination will be computed using GPS. The GPS reports current position once every second. The current position is reported as latitude and longitude in milliarcseconds (mas). The destination's longitude and latitude is stored in the memory and is entered by the user at the beginning of the flight. To calculate the distance to destination is simple geometry.

$$
\begin{aligned}
&ds = \sqrt{dx^2 + dy^2} \\
&where \\
&dx = dest\_latitude - current\_latitude \\
&dy = dest\_longitude - current\_longitude
\end{aligned}
$$

Once I obtain the distance, I'll use a look-up table to find out the value of the optimal speed. Figure 5 shows a mechanical device used for finding out the optimal speed. The Motorola controller has plenty of EEPROM memory to store even large tables. I'll create a look-up table with 30 rows (15 to 45 km distance) and 16 columns (wind strength -8 to +8 m/s). This table will be referenced to the altitude of 1000 meters the same as the device in Figure 5. For this reason I'll have to divide the current distance by the altitude in kilometers.

$$lookup\_dist = \frac{current\_dist}{\frac{altitude}{1000}}$$

The values in the table will be in km/h and will then be displayed.

5.7 Current Wind Indicator

Detecting wind in the air is tricky since the airplane is constantly moving and changing its velocity. I would like to try to compute the wind using information provided by various instruments in the proposed package. My theory is based on simple geometry but it is possible that with errors introduced by the instruments and the fact that glider pilots usually fly only during slow or moderate winds, the results might be inaccurate. Figure 6 shows the basic idea. The current wind indicator is going to operate in 5-minute intervals. At the beginning of the interval, the current position, reported by the GPS, will be recorded. During the 5-minute time period, the average speed and heading reported by the compass will be computed. Similarly as in the average vertical speed computation, every 10 seconds the value of heading and speed will be added to a register and a counter will be incremented by 1. At the end of the 5-minute period, the average will be computed and a hypothetical position will be computed using the stored position at the beginning of the cycle. Then the hypothetical position will be compared with the current position reported by the GPS. In case of wind-free weather conditions, those two values should be very close. If however, there is wind involved, the difference in those positions will give me information about the wind – Wind vector. Figure 16 shows the idea.

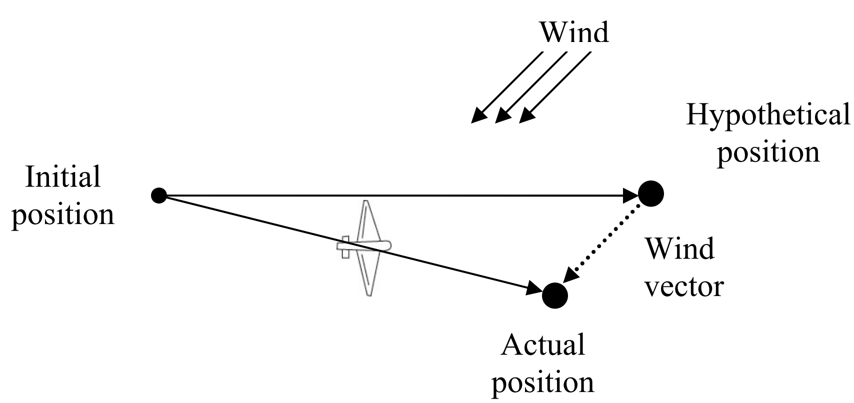

Figure 16 - Wind vector computation

The wind vector computation algorithm is similar to the distance to destination computation. I'm looking for the direction and magnitude of this vector.

$$
\begin{aligned}
&magnitude = \sqrt{dx^2 + dy^2} \\
&where \\
&dx = actual\_pos\_longitude - hypothetical\_pos\_longitude \\
&dy = actual\_pos\_latitude - hypothetical\_pos\_latitude \\
&direction = \arctan\left(\frac{dy}{dx}\right)
\end{aligned}
$$

The arctan() will be computed using Taylor series as described in Chapter 5.4.

CHAPTER 6

SCHEMATICS AND CIRCUIT BOARDS

For drawing schematics and designing circuit boards, I decided to use tools provided by PCB123 who also fabricated the final boards.

6.1 Schematics

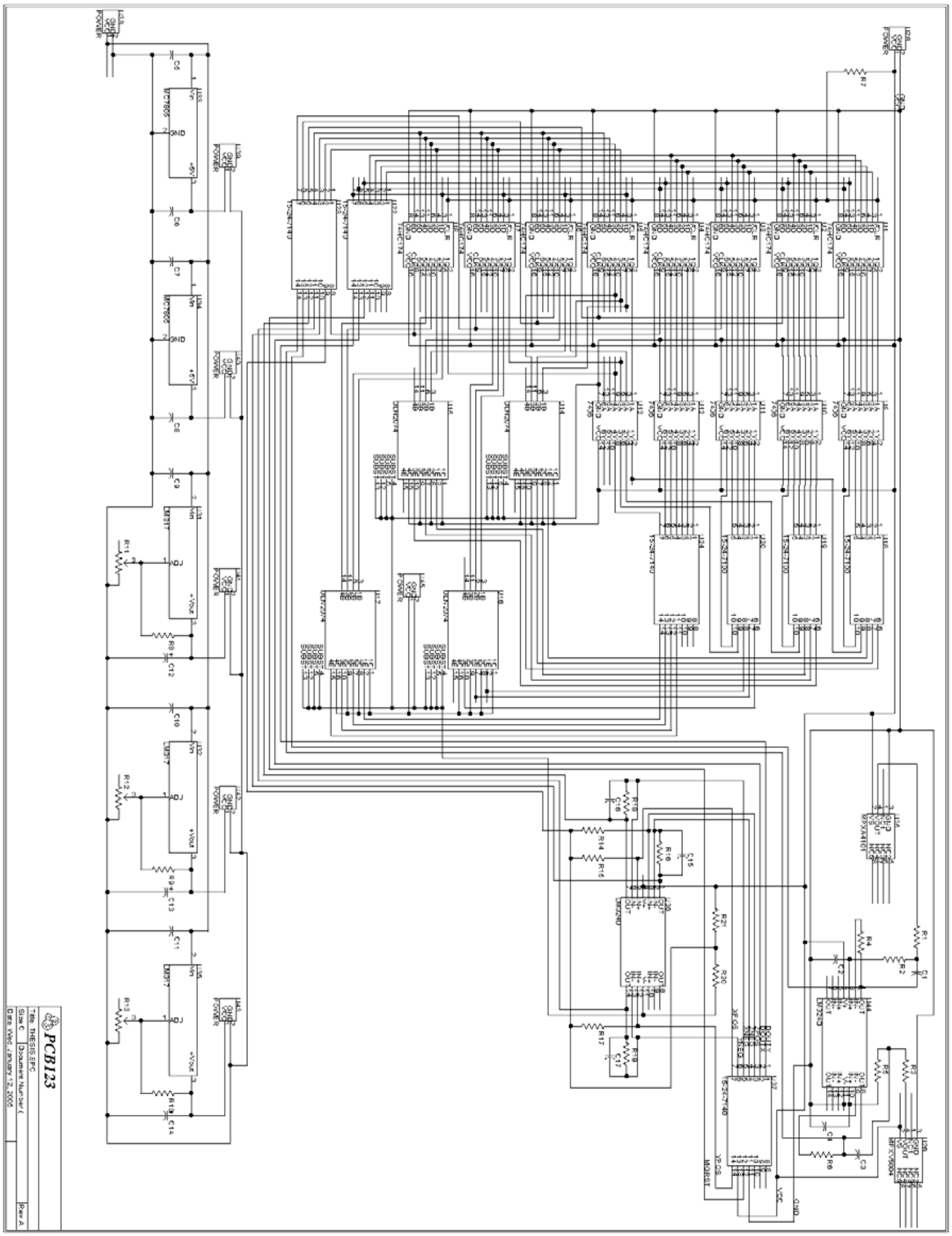

Figure 17 - Device schematic

6.2 Circuit Boards

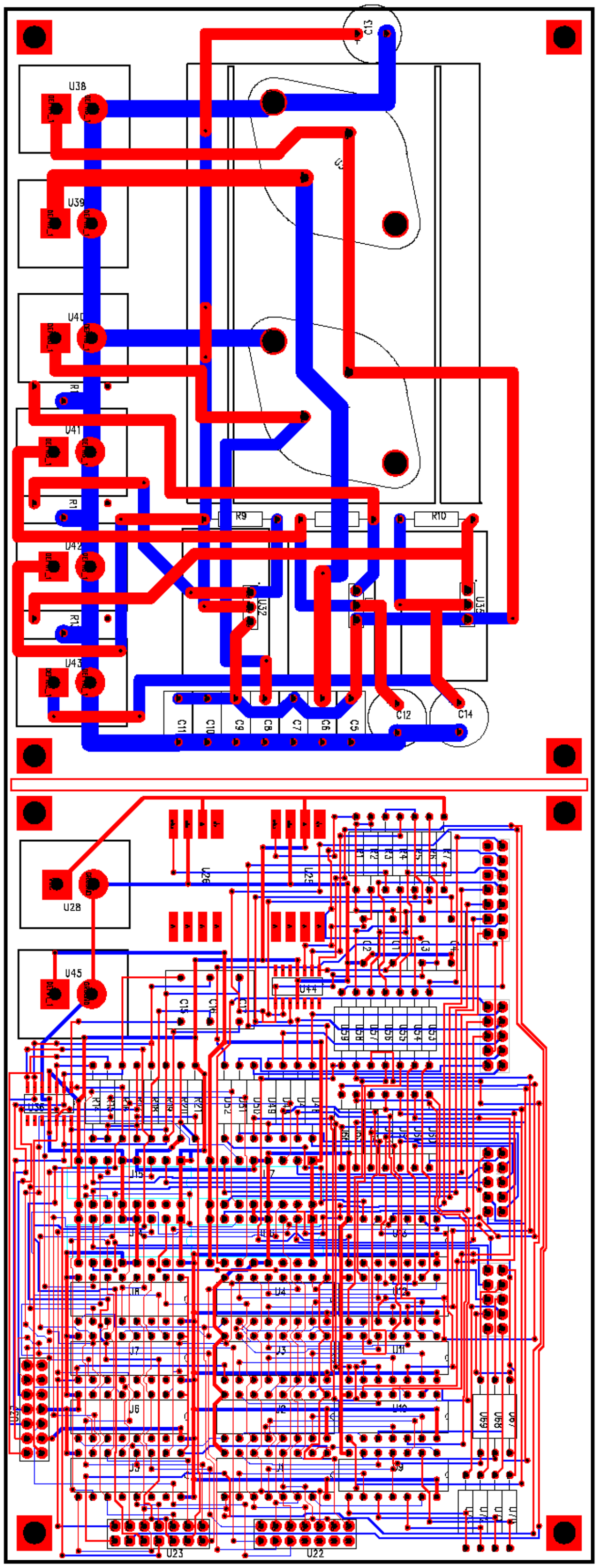

Figure 18 - Power supply (top), filters and registers (bottom)

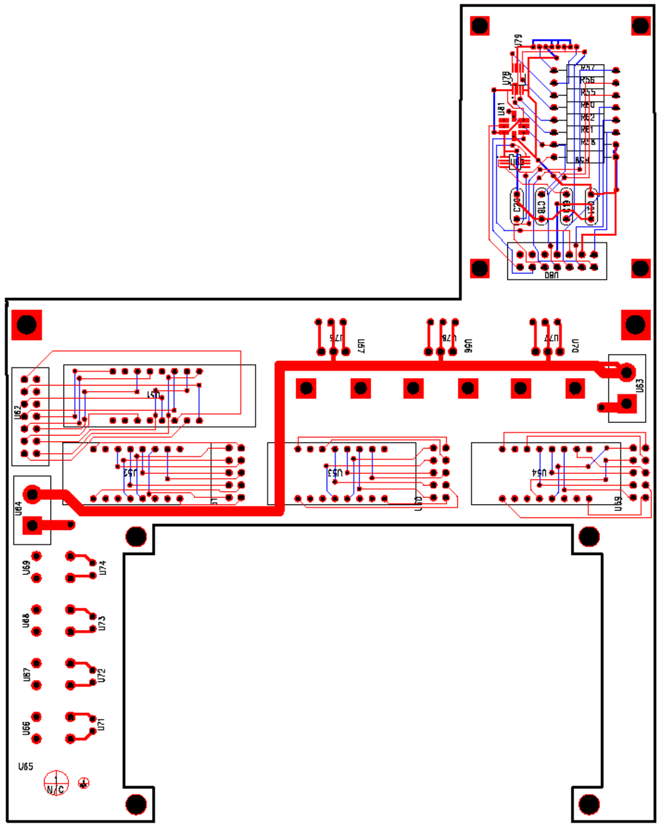

Figure 19 - Compass sensors (top), user interface (bottom)

The different boards are interconnected by flat ribbon cables. The pin-outs of used headers are described in tables bellow.

Table 3 - U22 header pin-out

pin	name	function
1	LED data	Data for LED registers – bit 0
2	LED data	Data for LED registers – bit 1
3	LED data	Data for LED registers – bit 2
4	LED data	Data for LED registers – bit 3
5	LED data	Data for LED registers – bit 4
6	LED data	Data for LED registers – bit 5
7	LED data	Data for LED registers – bit 6
8	Not used	
9	Not used	
10	Not used	
11	Not used	
12	RST_MAG	Reset for magnetic sensors
13	DOUT_X	Accelerometer output X
14	DOUT_Y	Accelerometer output Y

Table 4 - U23 header pinout

pin	name	function
1	LED Digit 0	Digit selection within one LED
2	LED Digit 1	Digit selection within one LED
3	LED Digit 2	Digit selection within one LED
4	LED Digit 3	Digit selection within one LED
5	Display 0	Display selection
6	Display 1	Display selection
7	Display 2	Display selection
8	Display 3	Display selection
9	V_ref	Reference voltage for compass
10	Mag_Z	Magnetic field in Z direction
11	Mag_Y	Magnetic field in Y direction
12	Mag_X	Magnetic field in X direction
13	Static_pr	Static pressure for altimeter
14	Dynamic_pr	Dynamic pressure for airspeed

Table 5 - U24 header pinout (4-digit LED)

pin	name	function
1	LED_a	LED segment a
2	LED_b	LED segment b
3	LED_c	LED segment c
4	LED_d	LED segment d
5	LED_e	LED segment e
6	LED_f	LED segment f
7	LED_g	LED segment g
8	Digit 0	Digit being refreshed
9	Digit 1	Digit being refreshed
10	Digit 2	Digit being refreshed
11	Digit 3	Digit being refreshed
12	Not used	
13	Not used	
14	Not used	

Table 6 - U18, U19, U20 header pinout (3-digit LEDs)

pin	name	function
1	LED_a	LED segment a
2	LED_b	LED segment b
3	LED_c	LED segment c
4	LED_d	LED segment d
5	LED_e	LED segment e
6	Digit 0	Digit being refreshed
7	Digit 1	Digit being refreshed
8	Digit 2	Digit being refreshed
9	LED_f	LED segment f
10	LED_g	LED segment g

Table 7 - Power supply connectors

component	function
U38	Input 12 – 15V
U39	Output 5V
U40	Output 5V
U41	Output 5V - adjustable
U42	Output 3V - adjustable
U43	Output 12V - adjustable

Table 8 - U37 header pinout

pin	name	function
1	DOUTY	Accelerometer output Y
2	DOUTX	Accelerometer output X
3	V+_Z	Mag. sensor output Z+
4	V-_Z	Mag. sensor output Z-
5	V-_X	Mag. sensor output X-
6	V-_Y	Mag. sensor output Y-
7	V+_X	Mag. sensor output X+
8	Not used	
9	Not used	
10	Not used	
11	GND	GND
12	RST_MAG	Reset for magnetic sensors
13	+5V	Power supply for mag. sensors
14	V+_Y	Mag. sensor output Y+

CHAPTER 7

RESULTS

The main focus of this project was to design and build a functional device that would be useful to glider pilots. Figures 20 and 21 show components and the assembled prototype of the device.

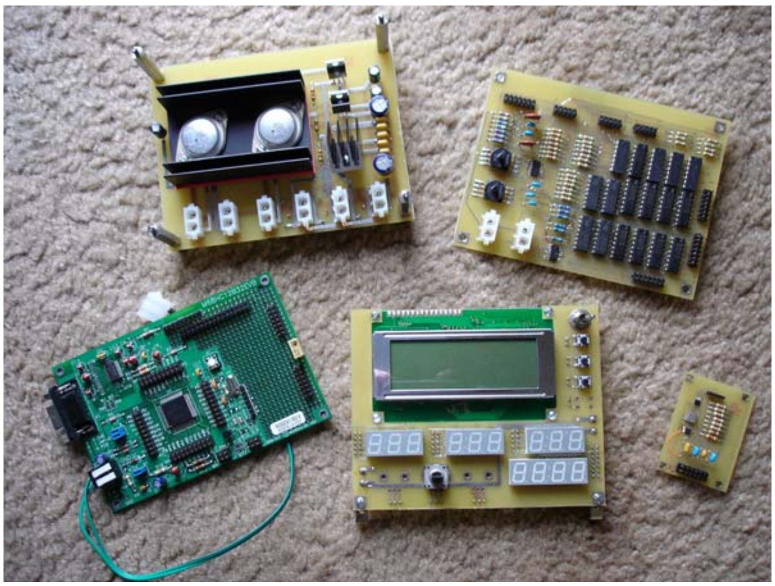

Figure 20 - Device modules

Figure 21 - Device prototype

This device is functional with several limitations. Some of the limitations are light and don't affect the overall usefulness but some of them are severe.

7.1 Pressure Sensors Range of Operation and Output Stability Issues

After building and testing the device I found out that the sensors I selected don't perform as well as I was expecting. Some discrepancies in my expectations came from not considering some aspects of this application and some I was unable to know about until the actual experience. The biggest surprise for me was that the output voltage wasn't as stable as I needed. This instability causes the controller's ADC to read slightly different values even if the pressure doesn't change. This causes the altitude indicator, for example, to show different values of altitude even though the device isn't moving or showing some random altitude while standing on the ground. This negative effect is even worsened by the fact that even a change on the least significant bit (4.8mV) means a difference in altitude of 10 meters which is not negligible to the pilot. I was trying to minimize this error by taking multiple readings and computing the average which provided some improvement in stability but the error was still large (in the magnitude of about 50 meters of altitude).

There was a similar problem with the dynamic pressure sensor but the error wasn't as significant for the airspeed indicator. The sensor performed much better and the range of operation was better chosen as well. A change at the least significant bit would cause an error of 0.5 km/h which is negligible to the pilot. The only problem there is the fact that it is quite distracting if the indicator isn't stable while standing on the ground. This instability issue could be fixed in software by doing some advanced mathematical analysis.

7.2 Compass Precision Issues

Testing showed that the reference design I used wasn't a good choice for this application and there are several reasons why. The first problem was that the magnetic field sensors didn't perform as specified by the manufacturer. Instead of the output voltage being in the range of ±5mV I was able to get only ±1.8mV which reduced the operation range for the ADC to only about 36% of its expected range. I tried to modify the design and adjust the operational amplifiers' gain to higher values so the range would improve but different abnormalities caused degradation of the range anyway. In my opinion, the problem was that in order to increase the gain and keep the input impedance of the amplifiers the same, I had to increase some resistor values to very high levels. Those large resistors then caused larger voltage drops and the reference voltage was degraded. The degradation was too large to obtain outputs suitable for processing with only 10-bit ADC with a resolution of only 4.8mV. When I measured the output voltages with a digital voltmeter, however, I was able to go through the compass algorithm "by hand" and obtained results that were fairly precise (about 6º error for the azimuth computation). However using the 10-bit ADC integrated on the Motorola controller made the device basically useless.

7.3 GPS Navigation

There were basically no issues with the GPS receiver and its performance. Everything worked as described by the manufacturer and the integration into my device was smooth.

7.4 Motorola 68HC912B32 Evaluation Board

Using Motorola's evaluation board seems to be a good choice as well. Although one board stopped working after several weeks of use but it was immediately replaced by the distributor and the new one worked perfectly fine. The only unpleasant surprise was that in this application, the integrated ADC would read different values when the input voltage was sampled multiple times. I didn't find any pattern in this error and it was completely random. My only hypothesis is that a lot of my code was interrupt driven and every interrupt caused changes in a large number of outputs. If those frequent changes happened during the sampling sequence and caused spikes in the power supply levels inside the chip, it could cause errors in the ADC readings.

CHAPTER 8

CONCLUSION

Overall, this project was a great experience. Even though the device didn't perform as I would have wished, it greatly extended the depth of my practical and theoretical knowledge. Because I worked on this project alone, I was involved in all the aspects of designing, testing and debugging. I gained a great deal of understanding of embedded system design, real-time signal processing, physical quantity sensing and GPS navigation.

From this experience, I can now see several errors I initially made in making decisions and I also understand that designing a device similar to this requires a lot more research, testing, and practical knowledge of the components used.

It is possible to make this device sophisticated in the future. If I find enough time, I would like to make it useful to my friends as well as myself. I would add more flexibility to the device by being able to upload plane characteristics and a flight plan from a computer or by using a flash card. I would also like to add the option of storing and downloading flight data during each flight. This feature would be particularly useful to competition referees and for analyzing pilot's errors.

REFERENCES

Valvano, Jonathan, *Embedded Microcomputer Systems: Real Time Interfacing*, Brooks/Cole, Pacific Grove, CA, 2000

Sedra, Adel and Kenneth Smith, *Microelectronic Circuits*, Oxford University Press Inc., New York, NY, 1998

Virginia Tech. “Airspeed Measurement” Online. Internet. [5 May, 2005]
Available WWW: http://www.aoe.vt.edu/~lutze/AOE3104/airspeed.pdf

Flying Wings. “Experiments In Flight Testing” Online. Internet [5 May, 2005]
Available WWW:
http://www.continuo.com/marske/articles/airspeed%20calibration/airspeed%20calibration%20article%20jan%202003.htm

Marko Rocznik. “Electronic Variometer” Online. Internet [5 May, 2005]
Available WWW: http://www.rocznik.de/Marko/electronic/project/vario/vario_engll.htm

Honeywell. “Applications of Magnetic Sensors for Low Cost Compass Systems” Online. Internet [5 May, 2005]
Available WWW: http://www.ssec.honeywell.com/magnetic/datasheets/lowcost.pdf

Honeywell. “3-axis Compass Sensor Set” Online. Internet [5 May, 2005]
Available WWW: http://www.ssec.honeywell.com/magnetic/datasheets/hmc1055.pdf

Synergy-Gps. “M12+ GPS receiver User’s Guide” Online. Internet [5 May, 2005]
Available WWW: http://www.synergy-gps.com/Mot_Manuals.html

Freescale. “M68HC12 & HCS12 Microcontrollers” Online. Internet [5 May, 2005]
Available WWW:
http://e-www.motorola.com/files/microcontrollers/doc/ref_manual/CPU12RM.pdf

APPENDIX

BILL OF MATERIAL

Bill of material for power supply, registers and filters (Figure 18)

Part	Value
C1	100n
C2	50n
C3	100n
C4	50n
C5	0.1u
C6	0.1u
C7	0.1u
C8	0.1u
C9	0.1u
C10	0.1u
C11	0.1u
C12	470u
C13	470u
C14	470u
C15	150p
C16	150p
C17	150p
R1	22.5k
R2	22.5k
R3	22.5k
R4	25k
R5	22.5k
R6	25k
R7	4.7k
R8	240
R9	240
R10	240
R11	2k
R12	2k
R13	2k

Part	Value
R20	10k
R21	10k
U1	74HC174
U2	74HC174
U3	74HC174
U4	74HC174
U5	74HC174
U6	74HC174
U7	74HC174
U8	74HC174
U9	7406
U10	7406
U11	7406
U12	7406
U13	7406
U14	ULN2074
U15	ULN2074
U16	ULN2074
U17	ULN2074
U18	10-pin header
U19	10-pin header
U20	10-pin header
U22	14-pin header
U23	14-pin header
U24	14-pin header
U25	MPXA4101
U26	MPXV5004
U28	pwr conn
U31	LM317
U32	LM317

Part	Value
U39	pwr conn
U40	pwr conn
U41	pwr conn
U42	pwr conn
U43	pwr conn
U44	LM324D
U45	pwr conn
U46	40
U47	40
U48	40
U49	40
U50	40
U51	40
U52	40
U53	30
U54	30
U55	30
U56	30
U57	30
U58	30
U59	30
U60	40
U61	40
U62	40
U63	40
U64	40
U65	40
U66	40
U67	40
U68	40

Part	Value
R14	1k
R15	1k
R16	1k
R17	1k
R18	1k
R19	1k

Part	Value
U33	MC7805
U34	MC7805
U35	LM317
U36	LM324D
U37	14-pin header
U38	pwr conn

Part	Value
U69	40
U70	40
U71	40
U72	40
U73	40

Bill of material for user interface and compass (Figure 19)

Part	Value
C14	1u
C15	0.22u
C16	0.1u
C17	0.1u
R41	4.9k
R42	4.9k
R45	4.9k
R46	4.9k
R47	4.9k
R48	4.9k
R53	220
R54	10
U7	IRF7509
U10	HMC1051
U17	HMC1052
U18	MXS3334UL
U28	14-pin header
U51	LTC4627JS
U52	LTC4624JS
U53	LTC4624JS
U54	LTC4624JS

Part	Value
U56	rot. Encoder
U57	rot. Encoder
U59	10-pin header
U60	10-pin header
U61	10-pin header
U62	14-pin header
U63	pwr conn
U64	pwr conn
U65	pwr switch
U66	button
U67	button
U68	button
U69	button
U70	rot. Encoder
U71	2-pin header
U72	2-pin header
U73	2-pin header
U74	3-pin header
U75	3-pin header
U76	3-pin header
U77	3-pin header

Miscellaneous parts:

Motorola Evaluation Board	M68HC12B32EVB
LCD display 20x4	LCM-S02004DSF