EE498
An Airspeed and Altitude Data Logger

By John Voth

Department of Computer and Electrical Engineering
University of Alberta
11 December 1996

Table of Contents

1. Introduction

1.1 Project Statement

The purpose of this project is to produce an airspeed and altitude data acquisition unit. The intended use of this project is to measure the flying performance of radio controlled model aircraft. The data logger rides inside the aircraft while the airspeed and altitude measurements are made using pressure transducers. The data is saved in battery-backed static RAM.

1.2 Scope of Project

The hardware central to this project will be the Motorola S68HC11EBU Design Board (which is based on the M68HC11E9 microprocessor unit and which contains all the on-board I/O circuitry required in this project), Motorola pressure transducers MPX2100AP, MPX5010DP and supporting hardware i.e.: memory chips, address decoding circuitry, etc. This Design Board provides excellent functionality at a small cost of weight and size with respect to the aircraft bearing the unit.

The pressure transducers are strain gauge style units which are capable of linearizing the exponential nature of barometric pressure. They are temperature compensated which prevents them from producing erroneous values due to extreme temperatures as experienced at high altitudes.

The project will have two special battery-backed 32Kx8 Static RAM chips that will provide semi-permanent data storage so that the unit can be shut down and then later on in time the data can be retrieved once the unit is turned back on. This allows for the user to make test flights without having to immediately doing computer analysis on the data. The data can be analyzed at a later time.

The software in this project is comprehensive. The first set of routines are for the functional testing of each of the hardware blocks in the system. The second set of routines work in concert with the hardware to measure, to store and to display the airspeed and the altitude being measured.

2. Theory of Operation

Before attempting to measure the airspeed and the altitude of an aircraft, the theory of how these measurements are taken must be described. A solid understanding of altitude and airspeed measuring will aid in the understanding of the overall project and in the circuitry used to make the measurements. To supplement what is described below, I have included in the index an article called “**Understanding Altimeters**” from Flight Training magazine, February 1996.

2.1 Altitude Sensing

The foundation of altitude sensing is the barometer. A barometer is simply a device that measures the local atmospheric pressure. The altimeter makes this same measurement and presents it in a form that is most useful to a pilot.

Atmospheric pressure is defined as a measure of the weight of the overlying air above the place where the measurement is being taken. This weight of overlying air decreases as the height above sea level increases. This change is approximately 1 inch of mercury for every 1000 feet change in altitude. This is how a barometer can be calibrated to measure altitude.

According to the aviation industry's specification, at the standard air conditions of 15'C, the weight of a column of air, one square inch in area, is 14.7 lbs. At sea level. This column exerts a pressure of 14.7 lbs/in^2. On a barometer, this pressure is measured as 29.92 inches of mercury and on an altimeter this pressure is measured as 0 (zero) feet. If the barometric pressure was measured at 10,000 feet altitude, the weight of one square inch column of air would have decreased to 20.58 inches of mercury. The decrease in pressure would be sensed by the altimeter and it would register as an increase of altitude to 10,000 feet.

2.1.1 Altitude Sensing Errors

It is true that atmospheric conditions vary from location to location and, as a result, an altimeter is subject to errors. These errors are induced by both pressure and temperature differences from standard air conditions.

2.1.1.1 Pressure Effects

Since barometric pressure changes from place to place, an altimeter should be adjustable so that the relative barometric pressure is factored into the altimeter measurements for the area that the equipment is being used in. On modern aircraft, the altimeter has a pilot operated calibration adjustment knob. This allows the pilot to adjust the altimeter for the local atmospheric conditions since the aircraft will most likely encounter differing weather conditions as the craft travels to its destination. This prevents the instruments from registering incorrect values.

On modern aircraft, the altimeter is adjustable in units of "inHg". Each 0.10 inHg added to the altimeter setting will increase the indicated altitude by approximately 100 feet. So, it can be seen that as an aircraft flies from an area of low pressure to an area of high pressure the altimeter will read high without correction.

Of special note is the concept of **pressure altitude**. It is the height above sea level corresponding to a given barometric pressure under standard conditions. This is essentially an altimeter without any adjustments made to the calibration. This concept arose during the testing phase of this project.

2.1.1.2 Temperature Effects

The atmosphere at any given point is rarely ever at the standard temperature of 15'C since there is constant heating and cooling. This induces an error in the indicated altitude. The amount of error depends on the degree to which the temperature differs from standard conditions. If the temperature is colder than standard air, the true altitude above sea level will be lower than indicated. Conversely, if the temperature is higher then the altitude measured will be higher.

Another special note is the concept of **density altitude**. It is the pressure altitude which is corrected for temperature. To determine the density altitude, the pressure altitude and the temperature must be known. This formula shows how density altitude is determined:

Density altitude = pressure altitude + [100 * (actual temperature-standard temperature)]

where

Standard temperature = 15'C

and

The temperature of standard air decreases 2'C per 1000 feet of altitude.

For every 1'C difference from standard air, there is a 100 foot difference in the pressure altitude (i.e.: +ve difference = increase in altitude).

For example, the density altitude at an airport at 2000 feet altitude at 25'C will be:

Density altitude = 2000 + [100 * (25' - (15' - (2*1'))] = 3400 feet.

Density altitude is an important concept and it was encountered during the testing phase of this project.

2.2 Airspeed Sensing

Airspeed is a measurement of the speed of an aircraft moving through the air. This is not the same as ground speed. Airspeed takes into account the movement of the atmosphere past the aircraft. Airspeed is measured by sensing the difference between the dynamic air pressure and the static air pressure. Dynamic air pressure is the pressure experienced by the aircraft as it moves forward through the atmosphere. Static air pressure is simply the barometric pressure at the present altitude of the aircraft. These concepts are synonymous with a ship traveling against the currents in the ocean.

2.2.1 Airspeed Sensing Errors

Airspeed is affected by several errors for which correction should be made: density, position, lags icing and water. We will only consider the first two sources of error since they were experienced during the testing phase.

2.2.1.1 Density

As discussed previously, the density of air depends on the atmospheric pressure and temperature. Since these values change frequently, a standard value for density has to be assumed in order to calibrate the airspeed indicator.

The standard for calibrating airspeed indicators is normal sea level pressure (29.92 inHg) at 15'C. A rough correction can be made to the indicated airspeed of 2% for every 1000 feet of pressure altitude.

For example, indicated airspeed at 10,000ft = 130 Knots.
The correction: 10 * 2% = 20%
so, 20% of 130 Knots = 26 Knots
therefore, the True Airspeed = 130 + 26 = 156 Knots.

2.2.1.2 Position errors

Position errors are mechanical in nature. The angle at which the Pitot tube meets the airflow has an effect on the measured airspeed. The Eddie currents, which are formed as air passes over the wings and struts of the aircraft, also affect the airspeed. These effects are reduced by placing the Pitot tube as far ahead of the leading edge of the wing as is practically possible.

3. Circuit Description

The project makes use of the Motorola 68HC11 EVBU Design Board because of its high scale of integration and small size. The EVBU provides a CPU, a communications port and a prototyping wire wrap area. The memory and sensors have been added to this prototyping area. The results are a small, lightweight, functional data logging unit capable of being installed in a radio controlled aircraft with minimal weight burden to the aircraft and a data logger capable of operating well on battery power.

3.1 System Block Diagram

Each of the system blocks will be broken down to explain the working relationships.

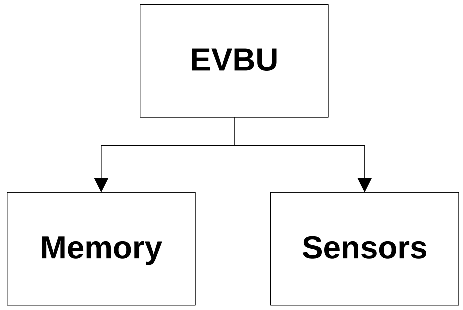


Figure 1: Functional Block Diagram

3.1.1 EVBU Block

As shown in figure 1, the data logger is centered on the Motorola EVBU Design Board. In addition to facilitating new circuitry on board, the Design board also handles all the communications with a host system, with the sensor circuitry and with the added memory.

The EVBU has a Motorola 68HC11E9 as its CPU. The 68HC11 is a highly integrated microprocessor which contains, among other things, a UART, an eight bit Analog to Digital convertor and multiplexed data and address lines. This project will make use of these functions extensively.

PORTE is an input port for the analog to digital convertor and it has eight channels that can be used to measure analog signals. This project makes use of only two of these channels.

PORTB and PORTC are the input and output ports which comprise the multiplexed data and address lines and they are used to connect the extra memory to the CPU.

The index contains three pages of schematics that show how the Motorola EVBU is electrically connected.

3.1.2 Memory Block

The memory block consists of six components: two 32Kx8 Static RAM's, two *SmartSockets* and two glue logic chips.

The memory has been mapped into the memory map of the CPU. In fact, it consumes the entire 64 Kbyte memory space. This memory map was chosen for a couple of reasons.

First, the program could reside in memory and appear to function like it was in a ROM thus becoming a restart-able, semi-permanent program.

Second, a large portion of the RAM could be used for power-off data storage.
But the most important reason for this memory map is because it allows the project to have recording times above one hour in duration. The schematic of the memory block is found in the index showing how the memory is attached to the system.

A most special feature of the memory block is the fact that it is battery-backed. This was designed into the system so that the user could be confident that the recorded data would be retained until such time as the data could be downloaded to a host computer and subsequently analyzed.

The battery-backup function is performed with Dallas Semiconductor *SmartSockets. SmartSockets* are an intelligent RAM socket that actively senses the Vcc line at all times. When Vcc begins to drop below a certain threshold the active circuitry on the *SmartSocket* comes alive and begins to hold the Vcc line on the RAM chip at a level which ensures that the contents of RAM are maintained. They were chosen because they are relatively inexpensive and provide at least ten years of service.

3.1.3 Sensor Block

The sensor block consists of two Motorola Pressure transducers and associated instrumentation amplifiers. They are strain gauge style of pressure sensors. Motorola pressure transducers have been utilized because of their ability to linearize the exponential nature of atmospheric pressure. These two sensors are the heart of the sensor block of the project.

3.1.3.1 Altitude Sensing

The altitude is sensed using a pressure transducer. It converts the applied pressure into a proportional, differential voltage signal. In this project, the applied pressure approached a maximum level of 100 kPa or 30.5 inHg as dictated by the pressure at sea level so, the sensor output had to have a linear output over this pressure range. In addition, the measured pressure had to be referenced to a known value.

Table 1 shows a list of altitudes and their corresponding pressure values as referenced to standard air conditions.

Table 1: Air Pressure vs. Elevation

Altitude (Feet)	Equivalent Pressure (inHg)
-1000	31.0185
-900	30.9073
0	29.9213
500	29.3846
1000	28.8557
1500	28.3345
2000	27.8210
3000	26.8167
4000	25.8418
6000	23.9782
8000	22.2250
10000	20.5770
12000	19.0294

The MPX2100AP pressure sensor was chosen because it is capable of measuring from 0 to 100 kPa. It is also temperature compensated and its output is linear over its sensor range. Its output is also referenced to an internal cavity which is evacuated to zero pressure.

This sensor required the use of some signal conditioning circuitry because its output is to be measured by the analog to digital conversion system. The A/D system requires the signal to be between 0.5 V and 4.5 V so that it can be measured. But the pressure transducer produces its output over a range of 0 mV to 40 mV and it produces 0 mV when it's reading the maximum. So, an instrumentation amplifier was created to perform the amplification the inversion, and the level shifting. The sensor schematic in the index shows how this was done.

To determine component values, calculation of the full scale output of the sensor, the gain and the offset of the amplifier have to be determined.

The spec sheets on the sensor state that the full scale output is 0 to 40 mV when the supply voltage is at 10 V. Since this system uses 8 V, the resultant full scale output of the sensor will be (8/10 * 40 mV) = 32 mV. Therefore, the full scale sensor output will be 32 mV.

A minimum range was chosen to be 15 inHg and the sensor output voltage at that value is approximately 16 mV. So, the output range of the sensor will be 16 mV to 32 mV. This is the range that must be inverted, amplified and level shifted by the amplifier.

It is necessary to calculate the number of steps used by the A/D convertor. A resolution of 0.1 inHg has been chosen and from this detail, the amplifier gain and range of the offset voltage can be found.

(30.5 - 15)inHg * 10 steps/Hg = 155 steps

The voltage span can now be found. The resolution of the eight bit A/D convertor with the reference voltage set to 5 V is

(5.0 - 0.0)Vref / 255 steps = 19.61 mV

So, for each step, the A/D will detect approximately 19.6 mV of change.

The voltage span of the sensor output is (32 -16) mV = 16 mV.

Now the gain of the amplifier can be found.

Gain = (155 steps * 19.5 mV/step) / (32 - 16) mV = 187

The gain can now be used to determine appropriate resistor values and the offset voltage for the amplifier circuit defined by this transfer function.

Vout = -[R2/R1 + 1] * V + Voff

where V is the differential output of the sensor. (Note: this transfer function has been gleaned from the Motorola Pressure transducer data book)

The gain of 187 can be realized with many combinations of resistors. This combination was chosen:

R1 = R3 = 121
R2 = R4 = 22K

The offset voltage of 0.5 V is assumed to be the output of the sensor at the maximum pressure of 30.5 inHg. Variable resistors were used so that trimming of the offset voltage could be done. This allows to circuit to be calibrated for local weather fluctuations.

The resultant circuit is laid out in the Sensor Block schematic found in the appendix.

3.1.3.2 Airspeed Sensing

The airspeed is also sensed using a pressure transducer. It converts the applied pressure into a voltage capable of being read by the A/D convertor.

Table 2 shows a list of airspeeds and their corresponding pressures.

Table 2: Airspeed vs. Pressure

Airspeed	
Knots	**Inches of Mercury (inHg)**
60	0.1727
80	0.3075
100	0.4814
110	0.5832
120	0.6950
130	0.8168
140	0.9488
150	1.0910
175	1.4918
200	1.9589
225	2.4943
250	3.1002
275	3.7792
300	4.5343
Note: 1 knot ≅ 1.15 statute miles	

In this project, the applied pressure approached a maximum level of 1.09 inHg. This was due to the fact that the test bed for the data logger was a Cessna 172RG which flew at a maximum speed of 145 Knots.

A MPX5010DP pressure transducer was used to make these measurements because of its special characteristics. Its most exciting characteristic is its built-in microprocessor interface circuitry. The sensor has an on-board amplifier which scales the output between 0 V and 5 V. This simplified the airspeed circuitry phenomenally. Support circuitry is limited to a resistor-capacitor signal conditioning circuit for the output signal line. The schematic for this circuit appears on the Sensor Block schematic in the appendix.

4. Software Description

4.1 Introduction

The bulk of the software written for the project focuses on storing of the sensor data in memory. The remaining portion of the software focuses on the user interaction functions. The major blocks of the software will be broken down to reveal their supporting role.

4.2 Program Structure

The structure of the program is quite simple. Dividing up the program into functional blocks reveals a simple, effective and orderly manner of code execution. Figure 2 below reveals the basic program structure. Figure 3 shows the basic functionality of the Interrupt service routine.

4.2.1 Main Program

The main routine is responsible for the majority of the work. It initializes the on-board peripherals of the 68HC11 namely the A/D convertor, the Real Time Clock interval, the Real Time Clock interrupts. It also is responsible for accepting commands from the user.

The user is able to enter in these commands:

A: Activate Analog recording
C: Clear Saved Data
D: Download Saved Data
H: Display Help Menu
S: Set Sampling Rate
T: Display Sampling Rate
Esc: Stop Recording Data

C code for each of these functions can be found in the appendix.

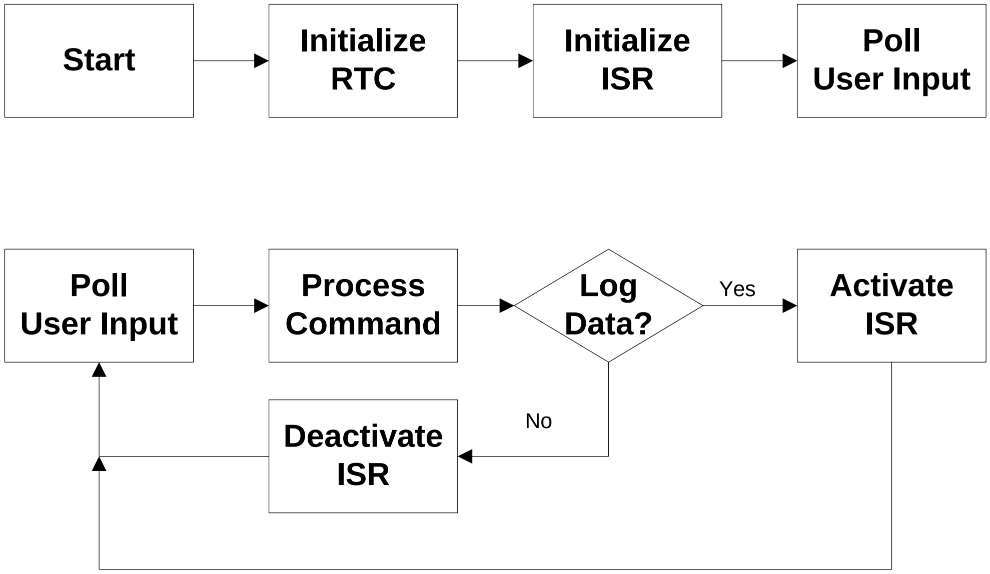


Figure 2: Main Program Structure

4.2.2 Interrupt Service Routine

The interrupt service routine is responsible for the critical functions of sampling the sensors, storing the data in memory and then displaying the data to the user. Figure 3 illustrates this. In order to do this, specific interval timing has to be maintained at all times. The 68HC11 has an on-board Real Time Clock peripheral that can be set to generate RTC interrupts at regular intervals.

For this project, the RTC interrupts were selected to fire every 32.77 msec. With this interval, specific predictions can be made concerning the sampling rate. With this information, the program has been constructed to allow the user to change the sampling rate to increase or decrease the resolution of the data collected.

Here are some calculations that illustrate the effect of the RTC interrupt rate on the amount of interrupts per second and the bytes per second that the data logger would be collecting at that specified rate. The user has the ability to select the sampling rates shown in the left column.

Table 3: Sampling Rates vs. Interrupt Speed

Rate = 0.032768 seconds		
Seconds	**Interrupts/Second**	**Bytes/Second**
0.1	3.05	30
0.2	6.10	15
0.5	15.26	6
1.0	30.51	3
2.0	61.04	1.5
5.0	152.6	0.6

10.0	305.2	0.3
20.0	610.4	0.15

Again, pseudo code and c code for the interrupt service routine is found in the appendix.

The above table leads to an important detail which needs to be addressed. How long will it take to fill all available memory? Using the amount of data being transferred each sample; this table has been constructed to illustrate the amount of time it would take to fill the system memory (approximately 20 Kbytes).

Table 4: Time to Fill Storage Memory

Rate	Seconds	Minutes
0.1	682.7	11.38
0.2	1365.3	22.76
0.5	3413.3	56.89
1.0	6826.7	113.8
2.0	13653.3	227.56
5.0	34133.3	568.89
10.0	6826.7	1137.78
20.0	13653.3	2275.56

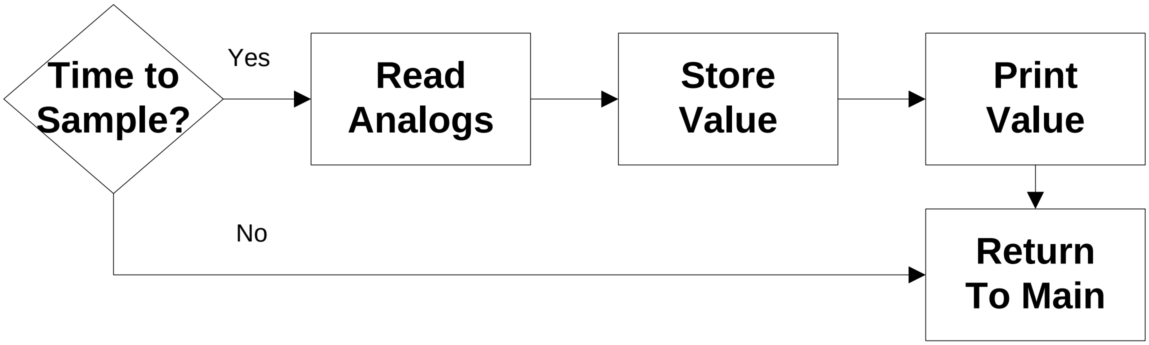
This table reveals the reason why this project requires a substantial memory complement for storing data. During the testing phase, a rate of one second was no burden for the system to keep up to.

5. Hardware Testing

5.1 Hardware

Hardware testing consisted of static and dynamic tests.

Figure 3: Interrupt Service Routine Structure



Static tests consisted of simple electrical checks to ensure that all wire wrapping was completed correctly.

Dynamic tests consisted of checking all signal lines to ensure that they were acting appropriately i.e.: no signals were shorted out to Vcc, ground or any other signal line.

5.2 Software

Software testing began by using the BUFALLO monitor program to check the operation of the peripheral devices. Memory was written to and read back to make sure that each location was usable. Memory retention was tested simply by writing a specific pattern to a known location, cycling the power and then reading that same location again. The pressure sensors were measured by reading the individual A/D convertor channels.

5.3 Calibration

Calibration of the pressure transducers are of great importance. Special attention was given to this.

The 100 kPa sensor used for sensing altitude was subjected to high vacuum calibration at a local lab facility. The testing equipment produced vacuum in increments of 1 inHg over a range of 2 inHg to 30 inHg. Here are the results of the calibration testing.

Table 5: Vacuum Calibration Testing

inHg	Counts
2	36
3	43
4	51
5	63
6	71
7	81
8	89
9	101
10	110
11	119
12	125
13	135
14	146
15	156
16	166
17	176
18	185
19	192
20	202
21	211

22	221
23	232
24	241

Calibration of the 10 kPa sensor used for measuring the airspeed was done during the field testing. This sensor is so sensitive to pressure changes that the lab equipment overwhelmed it immediately. The field calibration proved to be the most accurate method since the Cessna 172RG has equipment that is calibrated and has to be accurate by Canadian law. The results are shown in the field data interpretations.

5.4 Field Tests

Two field tests were conducted. The first was done using a Cessna 172RG light aircraft as the test platform. The second was done using my 1985 Volkswagen Golf as the test platform.

The Cessna has on-board instrumentation and it was used to verify calibration of both the altitude and airspeed sensors. For this test, a makeshift Pitot tube was attached to the Cessna's wing strut. The Pitot tube was attached so that its opening was perpendicular to the forward airflow so as to prevent mechanically induced errors.

The test flight consisted of a takeoff and a landing at the Municipal airport. The activities during the flight included a climb which leveled off to 4000 feet for about 10 minutes. The airspeed sensor was calibrated during this time. Then another climb which leveled off at 6000 feet was performed where the altitude sensor was calibrated. Subsequent maneuvers which tested the airspeed sensor were performed i.e.: both low and high speed flight. The landing commenced after this time. Plots of the data are located in the appendix.

The second field test consisted of a car ride to Leduc on the highway. During this test, the altitude data was ignored since it never changed due to low sensitivity.

The Pitot tube was fastened to the front of the car and perpendicular to the air flow. The trip began and continued quite uneventfully until I began speeding to 155 kph on the highway. Of special note was the sensors ability to detect the buffeting of passing traffic such as large highway trucks. You can see the many stops at traffic lights and also the stop right after the 155 kph run.

Plots of this data are found in the appendix.

6. Conclusions

6.1 Degree of Completion

At the time of writing this document, the project is fully functional. The software describe in the software section is fully written and fully debugged. The project accurately records samples from the pressure sensors and successfully stores the data until power can be restored to the unit.

6.2 Areas of Difficulty

Retrieving stored data from memory, after power was restored, was a challenge. For some time, the program zeroed out all the variables contained within the program. This was resolved through communication with the ImageCraft C support personnel (creators of the C compiler I used).

Calibration of the pressure sensors was a little tricky. Accurate sources of vacuum or pressure are difficult to locate. The lab equipment used to calibrate the altitude sensor was subject to error since the gauge it used was a mechanical device which experienced needle "stick-tion". This meant that it had to be continually tapped while measuring high vacuum.

Another source of error in the lab equipment was found in a rubber stopper. It did not seal completely at low vacuum. It required lots of vacuum to seal properly. This meant that the rubber stopper had to be held tight until the vacuum increased to acceptable levels.

These errors were challenged by the field testing. The altitude measurement was calibrated against the aircrafts altimeter. It was noticed that the pressure sensor accurately measured **pressure altitude** while it was off for indicated altitude because of local atmospheric conditions such as temperature and density.

Calibration of the airspeed was a challenge. The only method available to me was comparing the sensors output against the aircrafts airspeed indicator. The most difficult choice to make was in choosing to calibrate the sensor to **indicated airspeed** or to **calibrated airspeed**. The choice was made to use **calibrated airspeed** and afterwards it was noticed that this choice didn't matter all that much. It provided an accurate starting point for the airspeed sensor to begin measuring since the relative airspeed remained consistent regardless of this reference point.

For the Leduc field test, the airspeed measured low. This was due to the temperature being -17'C during the test. The low temperature had a noticeable effect since the plot illustrates the error - low readings. The ambient wind speed was negligible throughout the test so it shouldn't have made much of a difference.

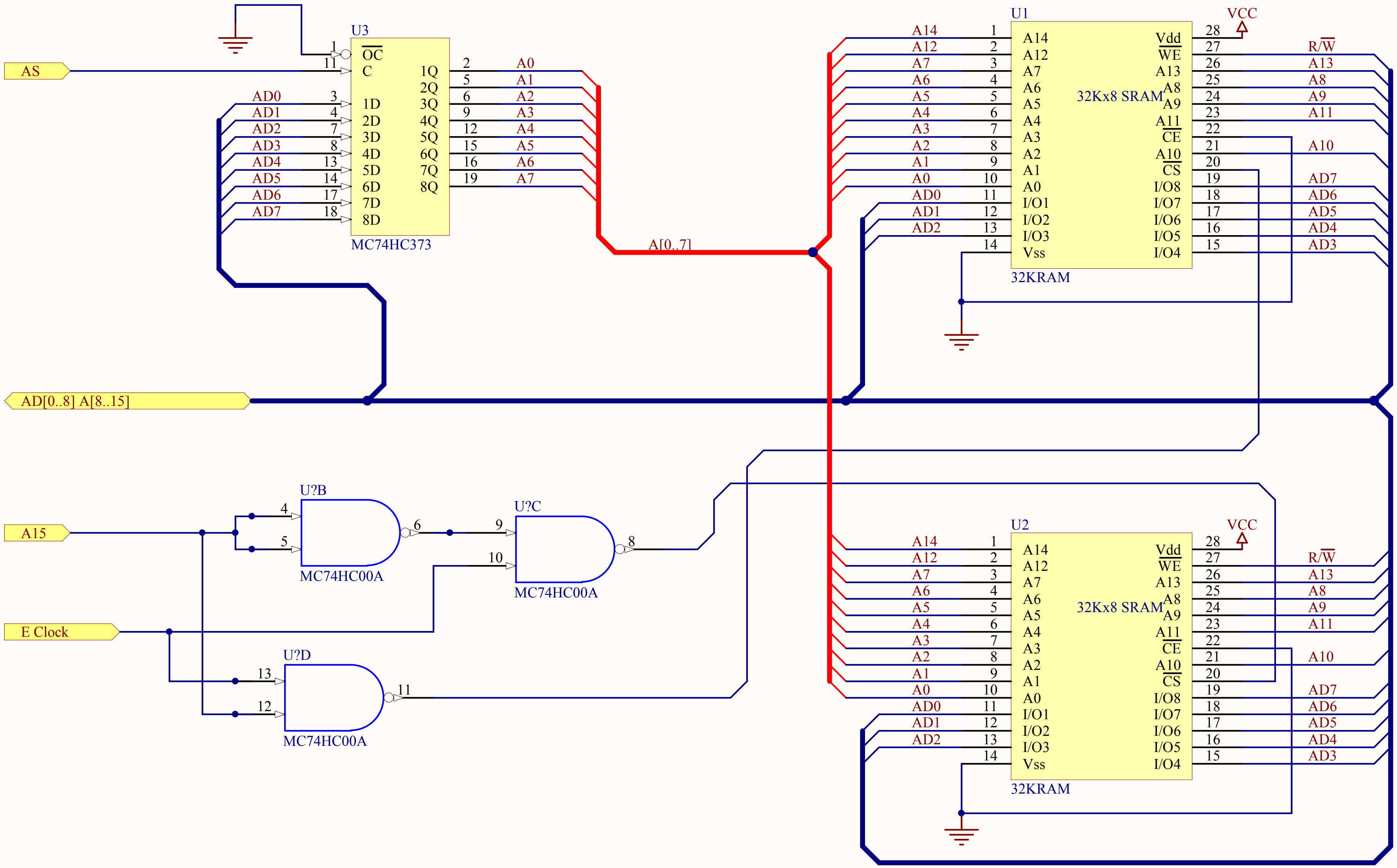
6.3 Recommendations

At this time, the project depends upon a laptop computer for user interaction with the data logger. It would be nice to have added on a small keypad and LCD for entering commands and monitoring data and program state. This wouldn't require too much additional input. It would require adding some more hardware to the project and the software can easily be edited to add the functionality.

At this time, the software does no post-sampling processing to the data. It might be useful to have to program do some scaling and level shifting of the data so that it could be readily useful. Counts from an A/D convertor just don't make sense sometimes. This is where Matlab stepped in and helped out.

Other than these things, this project was a lot of fun as well as being quite a handful to complete. I had to learn how to program a 68HC11 for the first time. The ImageCraft C compiler made that task a little easier. I had to read a lot of books to supplement my learning.

Appendix

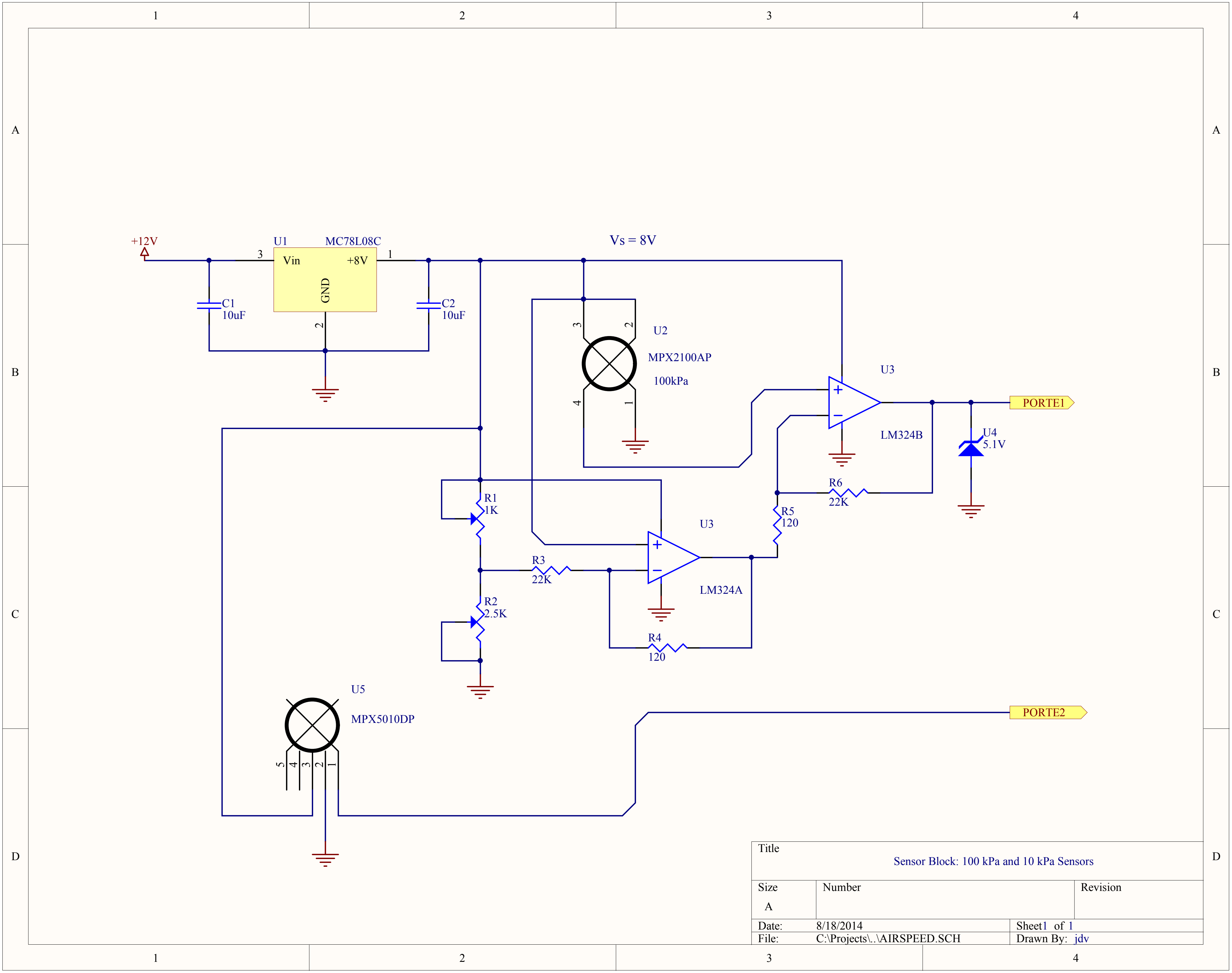


Title: Monitor Memory

Size	Number	Revision
A		

Date: 8/18/2014 Sheet 1of 1

File: C:\Projects\..\mon_mem.sch Drawn By: jdv



+12V
U1
MC78L08C
Vin
+8V
GND
C1
10uF
C2
10uF
Vs = 8V
U2
MPX2100AP
100kPa
U3
LM324B
U4
5.1V
PORTE1
R6
22K
R5
120
R1
1K
U3
LM324A
R3
22K
R2
2.5K
R4
120
U5
MPX5010DP
PORTE2
Title
Sensor Block: 100 kPa and 10 kPa Sensors
Size
A
Number
Revision
Date: 8/18/2014
File: C:\Projects\..\AIRSPEED.SCH
Sheet1 of 1
Drawn By: jdv