EE 318 Electronic Design Lab -I, Project Report, EE Dept., IIT Bombay, April 2007

ALTIMETER/VARIOMETER

Group No. D03

Hitendra Rohra(04D07020)<hitendra@ee.iitb.ac.in>

Jigar Vora(04D07021)<jigar@ee.iitb.ac.in>

Abhishek Patil(04D07022) <abhishekpatil@ee.iitb.ac.in>

Ravindra Meena(03D07015)<ravindram@ee.iitb.ac.in>

Supervisor: Prof. Dipankar

Course Instructor: Prof. H.Narayanan

Abstract

The objective of this project is to build an altimeter and variometer. Basically an altimeter is a device used to measure the altitude and variometer measures the rate of ascent and descent. We have attempted to design a robust and versatile circuit which meets the basic requirements of the paraglider pilots. The main features of our design are that it is small in size, low cost and weighs less than conventional designs. We have also attempted to reduce the power consumption of the device. The design approach followed has been explained in this report.

1. Introduction

Paragliding is an exciting activity and many people pursue it as a hobby. During daylight, the solar rays heat the earth's surface which in turn heats up the air close to it. As hot air has low density it starts to rise up. This is called a thermal. Para glider pilots use this rising thermal to gain height. The speed can be of the order of 1-5 m/s, however it may reach up to 15m/s. With the help of these thermals they can go up to the height of 10000-12000ft. When a para glider enters into a thermal he is lifted up. At times it is difficult to detect this lift and hence a pilot needs a **variometer**, a device for measuring altitude as well as rate of ascent and descent.

2. Design approach

The main motivation was to implement an accurate, hand held, easy to use altimeter / variometer at an affordable price. The device should have low power consumption and light weight so that it can be carried by the paraglider on long flights (4 – 5 hrs). The main purpose of a variometer is to help the pilot to detect thermals. Since the sizes of thermals are about 50 meters across, it has to be detected within a span of 1-2 seconds. The accuracy of the speed calculated should be of the order of 1 m/s. Other constraints include the output from the pressure sensor, which has a noise in frequency of 1/f type. Also as height increases, the rate of change of pressure with altitude decreases and small changes in altitude before increasingly difficult to detect.

The block diagram of the circuit done so far is as follows:

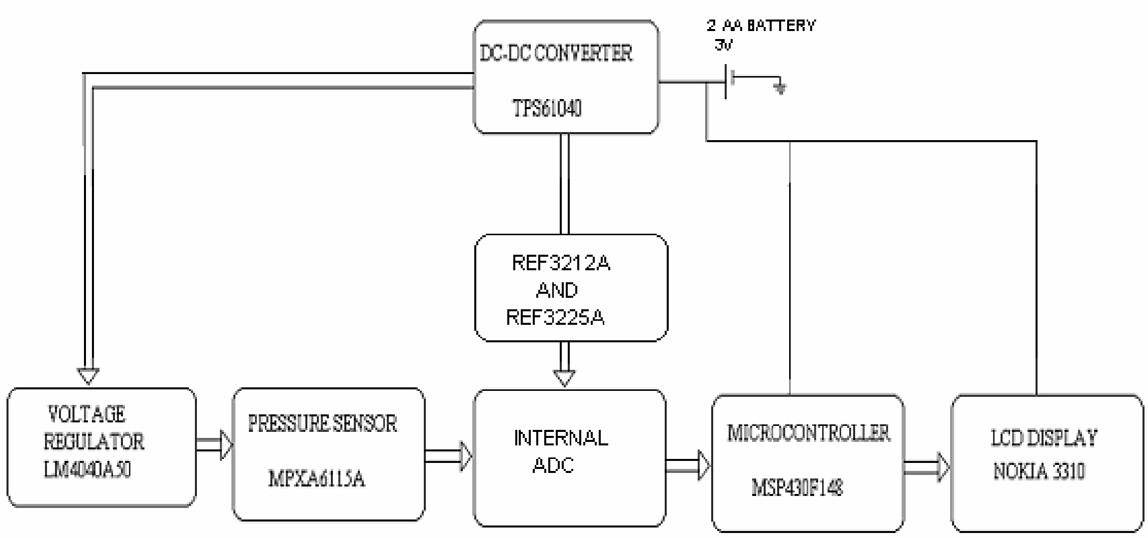

Figure 1: Block Diagram

The final system has been designed in various parts:

1. The sensors available to us required a power supply of 5V, which was not directly available in our system. Since devices have different voltage requirement we are using precision voltage regulators and DC-DC boost converters. This allows us to work with a single lithium ion battery.
2. The pressure sensor used is MPXAZ6115A. The output of pressure sensor is connected to a RC low pass filter, to remove a high frequency noise of 1/f type. The variation of output voltage of the pressure sensor with altitude, as measured by us, is shown in table 1.
3. The internal 12 bit ADC of the microcontroller MSP430F148 is used to convert the analog output form the pressure sensor to digital form. The maximum count of the ADC is 4096. If we make 1 LSB change in ADC count corresponds to 1m change in altitude we can have a maximum height of about 4000 m.

Vcc	Floor	Vout on GG building	Vout on 16 floor building.
4.35	0	3.54	3.53
4.35	3	3.53	3.52
4.35	6	3.52	3.52
4.35	10	__	3.51
4.35	13	__	3.50
4.35	16	__	3.50

Table 1

Table 1 shows the variation of output voltage of the pressure sensor with altitude. The readings were taken at the GG building as well as the 16 storey building in the campus.

4. The microcontroller module consists of MSP430F148, an LCD display (Nokia 3310), a buzzer and keypad. The count from the ADC is converted

to corresponding pressure and altitude values. The temperature measurement required for these calculations are taken from the temperature sensor present inside the microcontroller itself and converted into digital form using the microcontroller’s in-built ADC. The vertical velocity is also calculated and all the readings are then displayed on the LCD. A buzzer is provided which beeps with a frequency proportional to the velocity. A keypad is also provided to make settings.

The components used are explained in more detail as below:

Power Supply:

The device is supposed to run two AA batteries which have a combined series voltage of 3V. The battery voltage is sufficient to run the microcontroller and the LCD display. The pressure sensor however requires a 5V supply. For this we use a boost converter TPS61040. The boost converter can take an input of 3V and convert it to an output of around 5V. It can give an output current of a maximum 250mA. Since our circuit does not require more than a few mAs it is more than sufficient for our circuit requirements. The circuit for TPS61040 used is as follows:

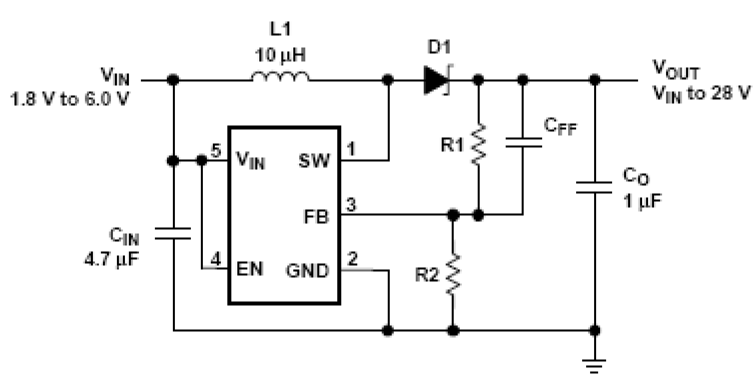

Figure 2: TPS 61040

The values of R1 and R2 chosen are 560 kΩ and 180 kΩ, since they give an output voltage of just higher than 5V. The value of C_{FF} is 0.2 nF.

To obtain an accurate value of 5V (needed for the pressure sensor) a precision voltage regulator LM 4040A50 is used. LM 4040 is a shunt regulator and is shown as below in figure 3.

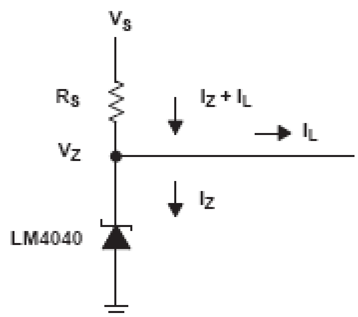

Figure 3: LM 4040

It can source upto 15mA of current and can be used for our purpose. The Vs in the figure 3 is about 5.3 V and the resistance Rs is 30Ω. Accurate voltage references of 2.5V and 1.25V are also required for setting the external references of (V_{eRef+} and V_{eRef-}) of the 12 bit ADC of the microcontroller. Since we want to determine the output voltage of the sensor with an accuracy of 1 mV, these references should also have an accuracy of the same order. Hence for this purpose accurate voltage references, REF 3212A (for 1.25V reference) and REF 3225A (for 2.5V reference) have been chosen. These references can source up to 10 mA of current and have a voltage accuracy of 1 mV. The circuit connection for REF 3225 is shown in figure 4.

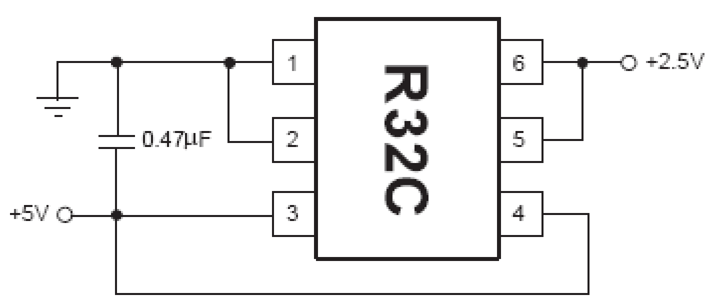

Figure 4: REF 3225

Pressure Sensor:

The pressure sensor used is MPXAZ6115A. It is a temperature compensated pressure sensor with a pressure range of 15kPa to 115kPa. It runs on a single 5V supply and gives an analog output which is proportional to the pressure. The 5V supply is received from the precision voltage regulator LM 4040A. The output of the sensor is connected to the input of the ADC using a simple low pass RC filter. The sensitivity of the pressure sensor is 45.9 mV/ kPa. The output of the pressure sensor as given by the datasheet is shown in figure 5.

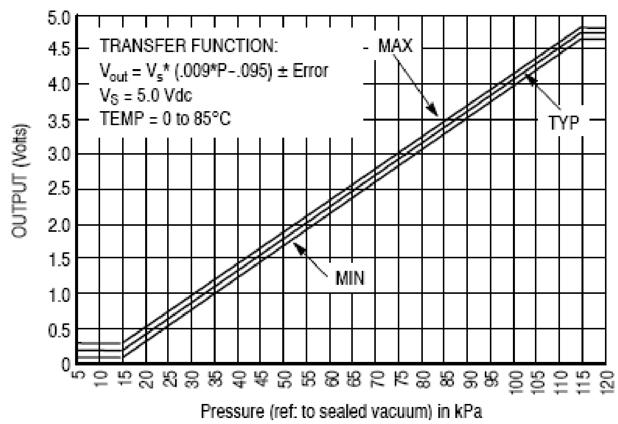

Figure 5: Output voltage (volts) versus pressure (kPa)for MPXAZ6115A

Analog to Digital converter:

The ADC used is the 12 bit internal ADC of the microcontroller. The 12 bit ADC has a count of up to 4096 and can be used for finding an altitude of up to 4000 m with an accuracy of 1 m. The analog reference voltages of the ADC are 1.25 V (V_{R-}) and 2.5 V (V_{R+}). They are set using accurate voltage references (REF 3212A and REF 3225A) as explained above. The output of the pressure sensor is about 4 V at ground, and is therefore scaled down to 2.4 V using a voltage divider (with resistance values 22 kΩ and 33 kΩ). The output value (N_{ADC}) the ADC varies linearly with the input voltage Vin as follows.

$$N_{ADC} = 4095 \times \frac{Vin - V_{R-}}{V_{R+} - V_{R-}}$$

Microcontroller:

The microcontroller used is MSP430F148, which is a low power microcontroller which can operate on a supply of 3V. The microcontroller performs 4 major functions:

(1) Calculating altitude and velocity from voltage corresponding to pressure sensor and calculating temperature from the internal temperature sensor.

(2) To display output on LCD and produce sound from the buzzer.

(3) To measure time of flight.

(4) To interface with the keypad.

The voltage reading from the ADC is taken and converted to the corresponding pressure assuming linear relationship. The altitude is calculated using the formula:

$$H = ((1-(P_H / P_{Ref})^{0.19026} \times T_{Ref}) / 0.00198122$$

When the altimeter is first switched on the pressure and temperature calculated are taken as the pressure and temperature at ground. The altitude is calculated in the code by using a look up table and taking interpolations in between the table values (this is done due to the difficulty in calculating powers). The altitude value thus found is sampled 64 times and an average of these samples is taken to obtain the final value of altitude (averaging is done to reduce error due to noise). The temperature is calculated using the inbuilt temperature sensor and then converted to digital form using the internal 12 bit ADC. The transfer function of the temperature sensor as per the data sheet is shown in figure 6. The output is displayed on the Nokia 3310 LCD. A varying frequency for the buzzer is provided by setting different counts to one of the timers of the microcontroller. The time of the flight can be calculated using one of the timers of the ADC and hence a record can be kept. Interfacing with the keypad is provided in order to set the reference points for zero altitude, to switch on or off the timer or the buzzer.

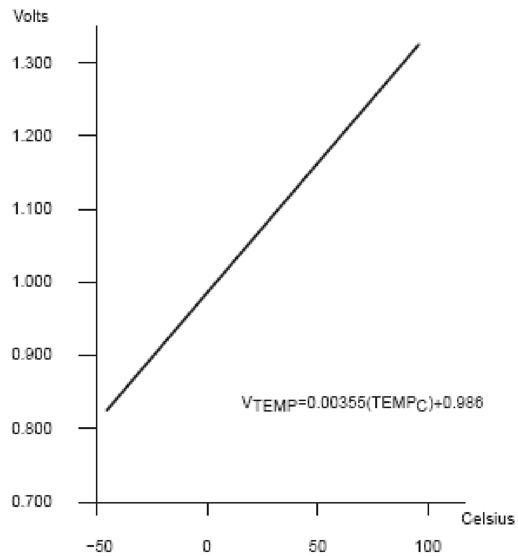

Figure 6: Typical Temperature sensor transfer function

3. Work done so far:

A PCB for the altimeter / variometer has been made with the microcontroller (MSP430F148), the boost converter (TPS 61040), the voltage references (REF3212A and REF3225A) and the pressure sensor present on the board itself. Pins have been provided to make suitable connections with the JTAG (for programming the MSP) and the Nokia 3310 LCD display. The LCD displays the real time altitude, speed, temperature and time of flight. In addition the unused port pins have been drawn out so that they can be used for connecting the buzzer and keypad.

The board file of the PCB made is shown in figure 7.

We have encountered a problem due to the boost converter. As our tests have shown the boost converter introduces noise in value of the ADC thus hampering results.

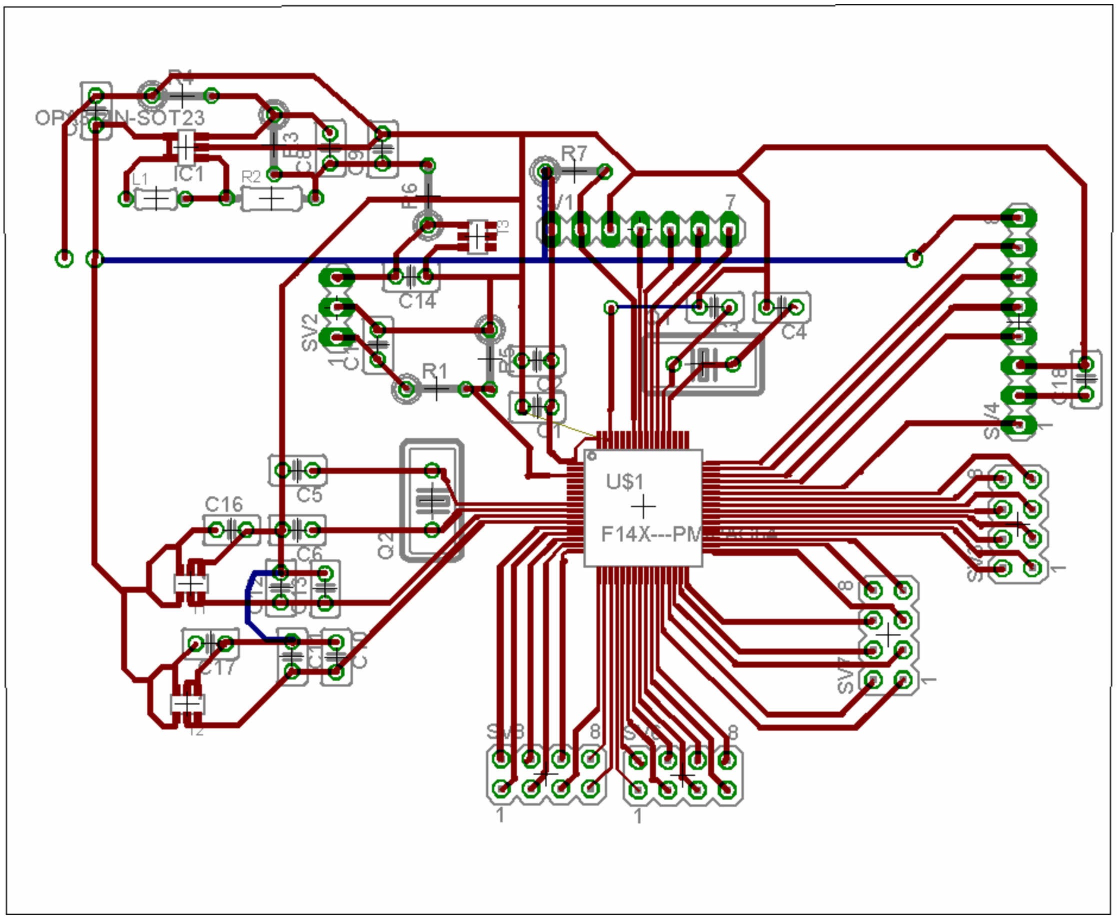

Figure 7: PCB layout

References:

1) www.ti.com
2) www.iar.com
3) www.freescale.com
4) www.olimex.com