ESE 381 Embedded Microprocessor Systems Design II

Spring 2019 - K. Short - revised April 14, 2019 10:15 am

Laboratory 10: Altitude Tracking Module (ATM)

To be performed the week starting April 14th. Task signatures without penalty the week of April 21st.

Prerequisite Reading
1. MPXHZ6130A Integrated Silicon Absolute Pressure Sensor Data Sheet.
2. REF19x Precision Micropower Low Dropout Voltage References Data Sheet.

Overview
Many embedded systems use a transducer to measure a physical measurand and directly display its value or provide other useful information computed using the value of the measurand. In this laboratory, you will design a system that does both.

When a pilot of an aircraft is assigned an altitude to fly by air traffic control (ATC), the pilot must not deviate significantly from the assigned altitude. This requirement is necessary to maintain vertical separation between aircraft to avoid collisions. A deviation of 300 feet or more from the assigned altitude can result in an ATC violation.

If a pilot in an aircraft without an autopilot is busy or becomes distracted and the aircraft is not well trimmed, it is easy for the aircraft to deviate from its assigned altitude. A low cost instrument that can get a pilot's attention when the aircraft starts to deviate from the assigned altitude would allow the pilot to quickly correct the deviation and avoid a violation (or worst a collision). An Altitude Tracking Module (ATM) is such an instrument.

The basis for the design of this Altitude Tracking Module (ATM) is that under normal conditions (standard atmosphere) atmospheric pressure decreases by 1" Hg (inch of mercury) for every 1,000 foot increase in altitude. Using this assumption, if you have a transducer that can measure atmospheric pressure, then changes in atmospheric pressure can be measured and translated to changes in altitude.

An ATM can be designed to be portable and packaged as a single module for use in aircraft with non-pressurized cabins. Most general aviation aircraft do not have pressurized cabins. Integrated solid state pressure transducers with on chip signal conditioning are available for use in altimeter or barometer applications. The NXP MPXHZ6130A is such a device. The MPXHZ6130A provides an output voltage that changes linearly from 0.2 to 4.8 volts for pressures from 15 to 130 kPa (2.2 to 18.85 psi).

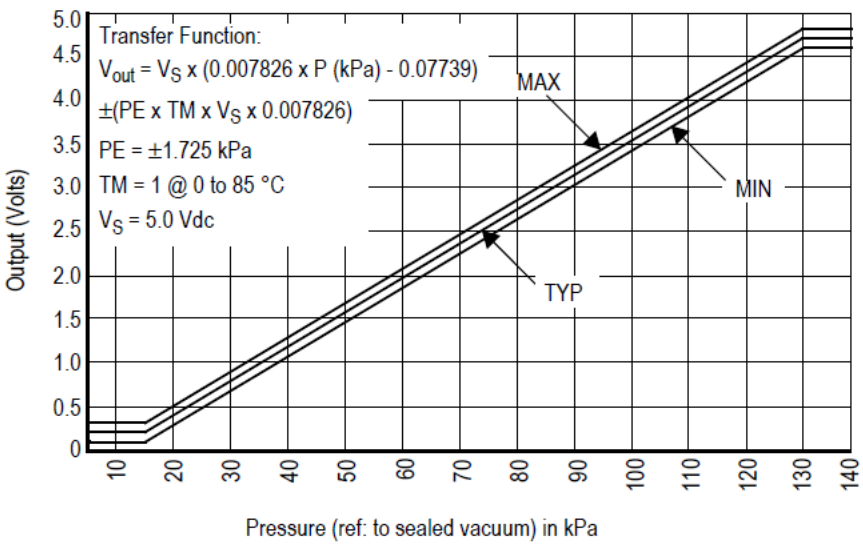
One kilo Pascal (kPa) equals 0.145 pounds per square inch (psi). And, since 29.92 “Hg, the pressure at sea level on a standard day, equals 14.7 psi (101.325 kPa) we can determine that a change of 1”Hg corresponds to a change of 3.388 kPa.

Using the MPXHZ6130A we are able to measure atmospheric pressure and to implement an Altitude Tracking Module (ATM).

Design Tasks

Design Task 1: Altitude Tracking Module

Ideally, we would want the MPXHZ6130A pressure transducer’s output to be connected to a channel of the ATSAML21J18B’s ADC. However, since the MPXHZ6130A operates at 5.0V supply voltage and the ATSAML21J18B is being operated at 3.3V, they are not directly compatible. For example, on a standard day with a pressure of 101.3 kPa, the output of the MPXHZ6130A would be about 3.5V.



The input of the MPXHZ6130A’s ADC is limited to VDDANA, which is 3.3V on the Xplained Pro board. In an actual system, we could scale the ATSAML21J18B’s output. For simplicity in the laboratory we will simulate the scaled output of the MPXHZ6130A using a potentiometer. We will assume that the MPXHZ6130A output is scaled by 1/2 and use the 2.5V output of the REF192 as the voltage reference for analog-to-digital conversions.

The ATM is to have a single SET pushbutton input and three LED outputs. Its front panel would be similar to the inexpensive commercial product shown below. Except that our prototype does not have the ON pushbutton at the top left.

When the system is powered ON, all three LEDs must be simultaneously turned ON for 1 second and then turned OFF. This demonstrates to the pilot that all three LEDs are operational. Use a software delay subroutine for this purpose.

This system must operate as follows. After establishing level cruise at the assigned altitude, the pilot presses and releases the SET pushbutton. This establishes the assigned altitude for the ATM. The green LED turns ON to indicate that the deviation between the assigned and actual altitude is less than 100 feet. If the aircraft climbs to more than 100 feet, but less than 200 feet, above the assigned altitude, the green LED stays ON and the yellow LED turns ON. This indicates a high, above assigned altitude, situation for the aircraft. If the aircraft exceeds the assigned altitude by more than 200 feet, the green LED goes OFF and the yellow LED stays ON.

If the aircraft descends to more than 100 feet below, but less than 200 feet, below the assigned altitude, the green LED stays ON and the red LED turns ON. This indicates a low, below assigned altitude, situation for the aircraft. If the aircraft is below the assigned altitude by 200 feet or more, the green LED goes OFF and the red LED stays ON.

If the aircraft returns to an altitude less than 200 feet above or below the assigned altitude, the green LED turns back ON. If the aircraft returns to an altitude less than 100 feet above or below the assigned altitude, only the green LED is ON.

The following table shows under what conditions each LED is ON or OFF in terms of altitude deviation.

Table 1: ASM LED Operation

Altitude	Red LED	Green LED	Yellow LED
+200 ft.	OFF	OFF	ON
+100 ft.	OFF	ON	ON
SET	OFF	ON	OFF
-100 ft.	ON	ON	OFF

Table 1: ASM LED Operation

Altitude	Red LED	Green LED	Yellow LED
-200	ON	OFF	OFF

Write a program, `atm`, that provides the desired system operation. Simulate your program completely before your laboratory session. One approach to writing this program is to determine what a 100 foot change is when expressed as a bit change in the output from the ADC. Knowing this, you can compute the boundaries for the altitude changes needed to control the LEDs. Try to use only integer arithmetic in your program.

Draw the system schematic, use a REF192 as the reference voltage.

Submit your schematic, program listing, and program verification strategy as part of your prelab.

Design Task 2: Altitude Tracking Module with Blinking LEDs

A drawback of annunciator lights in an aircraft is that the aircraft's instrument panel is full of them. Blinking an LED attracts a pilot's attention allowing a quicker awareness of an altitude deviation.

The design of Task 1 can be modified to blink the red or yellow LEDs when the aircraft is more that 100 ft. from its assigned altitude. The following table shows whether each LED is ON, OFF, or Blinking in terms of altitude deviation.

Table 2: ASM LED Operation

Altitude	Red LED	Green LED	Yellow LED
+200 ft.	OFF	OFF	Blinking
+100 ft.	OFF	ON	Blinking
SET	OFF	ON	OFF
-100 ft.	Blinking	ON	OFF
-200	Blinking	OFF	OFF

Write a program, `atm2`, that is a modification of `atm` that provides the desired system operation. When blinking an LED, your program must turn the LED ON for 0.25 seconds and OFF for 0.25 seconds. Simulate your program completely before your laboratory session.

Submit your program listing and program verification strategy as part of your prelab.

Design Task 3: Computing and Displaying the Altitude

The ATM does not need to compute actual altitude to operate properly. It functions by determining changes in altitude, beyond certain limits, as corresponding changes in pressure. However, it would be helpful to display the altitude being simulated to debug or demonstrate the operation of the ATM. To do this, you simply have to extend the ATM's functionality to make it a rudimentary pressure altimeter.

The standard atmosphere is an artificial model of the atmosphere that agrees well with the real atmosphere on a cool day in temperate regions. This model is called the International Standard Atmosphere (ISA).

A standard altimeter follows the ISA model, but includes a provision to shift the entire relationship up or down. That is, to change the value of the pressure at sea level from the value for the standard atmosphere represented by the model. This adjustment is referred to as the "altimeter setting."

Thus, a pressure altimeter must allow the altimeter setting to be entered. We will use units of "Hg (inches of mercury) to express our altimeter setting. The recorded extremes in sea-level pressure range from 25.69 (in a hurricane) to 32.00 (during an extreme cold winter). The sea-level pressure in the standard atmosphere is 29.92. Thus, for our electronic altimeter we will provide for the entry of altimeter setting values from 25.00 to 32.50 "Hg. To accomplish this using only integer arithmetic, these values can be represented in memory scaled by 100. Thus, the stored values would range between 2500 to 3250 and would represent pressure in hundreds of an inch of mercury. Computations would be done using these units and results could be unscaled for display. Unscaling for display as a decimal value could be accomplished by simply placing a decimal point in the correct position on the display.

The equation for the altitude versus pressure has the basic form:

$$\text{altitude} = -K * \ln(P_0/P)$$

where P_0 is the actual pressure at sea level (altimeter setting) and P is the pressure at the altitude being determined. Evaluation of the equation by a microcontroller requires the computation of the natural logarithm.

A much simplified, but less accurate equation, appropriate for lower altitudes, is:

```
altitude = 1000 * (alt_setting - measured_barometric_pressure)
```

which is the equation you must use to determine the altitude for this task. This equation assumes that the barometric pressures or in inches of mercury and altitude is in feet. If the pressures are in hundreds of an inch of mercury, then the multiplier is 10, not 1000.

Write a program, `atm3`, which is a modification of `atm2`, that displays the altitude in feet on the LCD. The altimeter setting will be a constant in your program, defined by a #define statement. The appropriate value to use in the laboratory for the altimeter setting will be given to you during your laboratory section. For program development and debugging outside of the laboratory, use the value 29.92.

Submit your program listing and program verification strategy as part of your prelab.

Laboratory Activities

Laboratory Task 1: Altitude Tracking Module atm, Simulated Using a Potentiometer

Create a project named `atm`. Add your program `atm` and compile and debug it. Power up your circuit. Download your program. Run and debug your program. If there are problems, identify their sources and correct them.

Note that when debugging you should monitor the ADC's operation by viewing its registers in the I/O View Toolbar of Studio.

Connect the trimpot output to ADC input. For this task you must use the trimpot's output to simulate the MPXHZ6130A with its output scaled by 1/2. Using the trimpot generate voltages from 0 to 2.5V, in 0.5 V steps.

Run your program full speed. Vary the analog input voltage and determine whether your ATM provides the proper LED indications to the pilot. Verify the operation of the ATM LEDs against the simulated scaled MPXHZ6130A output voltages.

Have a TA verify that your circuit performs as required. Also, have the TA verify your computed input voltage values. Get signed off by the TA.

Laboratory Task 2: Altitude Tracking Module atm2 with Blinking LEDs, Simulated Using a Potentiometer

Create a project named `atm2`. Add your program `atm2` and compile and debug it. Power up your circuit. Download your program. Run and debug your program. If there are problems, identify their sources and correct them.

Run your program full speed. Vary the analog input voltage and determine whether your ATM provides the proper blinking LED indications to the pilot. Verify the operation of the ATM LEDs against the simulated scaled MPXHZ6130A output voltages.

Have a TA verify that your circuit performs as required. Also, have the TA verify your computed input voltage values. Get signed off by the TA.

Laboratory Task 3: Altitude Tracking Module with Blinking LEDs

Create a project named `atm3`. Add your program `atm3` and compile and debug it. Power up your circuit. Download your program. Run and debug your program. If there are problems, identify their sources and correct them.

Run your program full speed. Vary the analog input voltage and determine whether your ATM plus pressure altimeter displays the correct altitudes on the LCD display and controls the LEDs appropriately.

Questions

1. What change in voltage from the scaled pressure transducer output corresponds to a 100 foot change in altitude? What is the output change, in bits, from the ADC corresponding to a 100 foot change in altitude?

2. What would be the output change, in bits, corresponding to a 100 foot change in altitude if a 16-bit external ADC were used? What would the change in bits be if an external 24-bit ADC is used

3. What is the smallest change in altitude that the hardware used in this design is capable of detecting?