# MODULE 08

## HYBRID CIRCUITS

PREREQUISITES: MODULE 11: MICROCONTROLLERS III.

OUTLINE OF MODULE 08:

What you will learn about in this Module:

Digital to Analog Conversion (current and voltage modes)
Analog to Digital Conversion
Voltage to Frequency and Frequency to Voltage Converters
Analog Multiplexers (solid-state analog switches)

What you will build in the lab:

If you are registered for this course, the PC board that you need is already built for you. You will need to check out the Module 11 test circuit board from one of the instructors. The schematic for this board is included, so that you can build this circuit if you need to, or if you would like to modify the circuit for other purposes. You will use the test circuit board to read an analog voltage using an analog-to-digital converter (ADC) and display it on the LCD screen (just like a digital volt meter).

INTRODUCTION:

Hybrid circuits are circuits that contain two or more types of signals, for example, a circuit that has both digital and analog signals. This is also sometimes called "mixed signal". Usually in circuits of this type there is some sort of signal conversion that takes place. Very common signal conversions include changing an analog signal into a digital signal, or the reverse, converting a digital value into an analog voltage. These and other types of signal conversion are so common that you can easily find integrated circuits to perform the conversion…mostly you just need to supply the chip with power and the signal, and then read off the converted signal. Mostly you will read about these different kinds of signal conversion chips, just so that you have some Idea of what is available for your designs. Then, you will program a microprocessor to convert an analog voltage into a digital signal and display it on the LCD display.

READINGS FROM HOROWITZ AND HILL (H&H): *ART OF ELECTRONICS*

9.15-9.31

ADDITIONAL READINGS & INTERNET RESEARCH:

On the Internet, locate and read the following application notes:
Texas Instruments: Understanding Data Converters (SLAA013)
Microchip: AN10: voltage-to-frequency and frequency-to-voltage converter
Maxim-IC: Selecting the right CMOS analog switch

# MODULE 08

## SELF QUIZ

1: What is the maximum sampling rate of the TLC4545 16-bit ADC?

2: If the full range of the signal that you are sampling with the TLC4545 is 0 to +5 volts, what is the smallest change in voltage that you can resolve? (Hint: this is a 16-bit ADC)

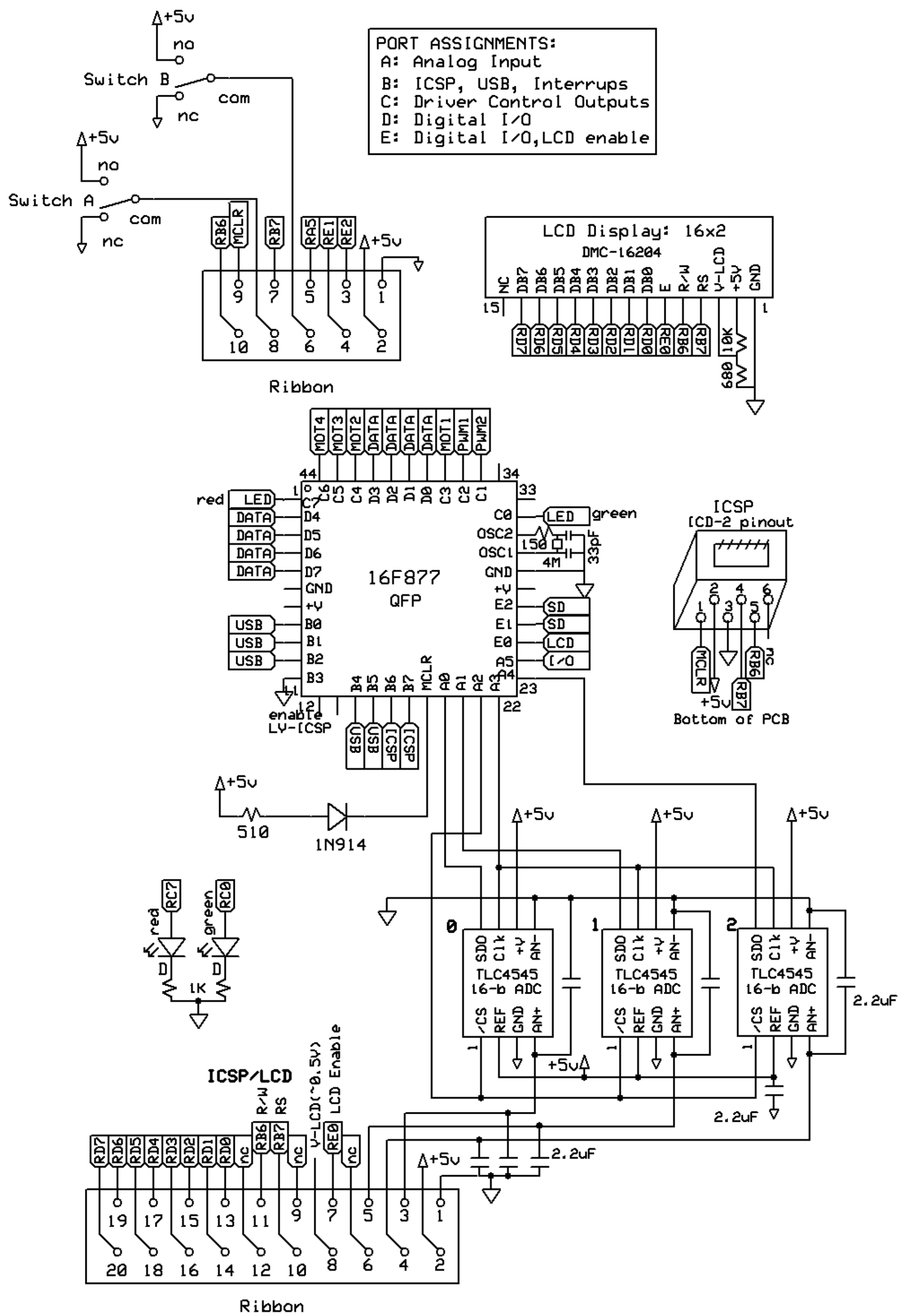
PLEASE ANSWER THE ABOVE QUESTIONS AND E-MAIL TO THE INSTRUCTOR
"I have neither given nor received aid on this examination, nor have I concealed any violation of the Honor Code"

X_____________________________________________________

# MODULE 08

## LABORATORY PROJECTS

1- You will need to check out the test circuit board from one of the instructors. It is the same as the test circuit board that you used for Module 11. The schematic is shown below. We have a limited number of these, so please only check them out for the day you intend to use them.



2- You will need to locate and carefully read the data sheet for the analog-to-digital converters. These are the TLC4545 chips on the test circuit board, and they are manufactured by Texas Instruments. You should carefully review this document, since you will use these ADCs in several subsequent modules. You want to become familiar with the performance of the ADC, as well as how to operate it. It has a “serial interface”, meaning that only a few wires are used to send commands and receive data. You should look at the timing diagrams on the TLC4545 data sheet. These will tell you something about how the digital commands and data are transferred to and from the ADC. This is a very common scheme for modern peripheral devices.
3- Not shown on this schematic are two 20 KΩ (pots). These are the two small rotary potentiometers on the test circuit, near the phone plug for the serial programmer. Each one has a 1/8” diameter white plastic shaft with a screw driver slot in the end to facilitate turning. The pot labeled “0” is connected to ADC #0, and the pot labeled “1” is connected to ADC #1. The potentiometers are connected to the +AN (positive analog input) pins on each ADC (pin #4). The potentiometers are being used as voltage dividers, to provide a voltage signal of 0.0 to 5.0 volts to each of the ADCs, just so you can see how they work. Sketch out how you would use a potentiometer, connected to +5V and ground, as a voltage divider to provide a voltage anywhere between +5V and ground. Now check the physical test circuit to see how they are actually connected (visually inspect the board, follow the wires, use the schematic…)
4- Everything that you will need for this module has already been assembled on the test circuit board, so all you need to do is program the PIC16F877 microcontroller. You can begin by downloading the software from the web page: module8.txt. Paste this text file into the PIC-C compiler, and compile it to generate hex code (a file named module8.hex), the same way that you did in Module 11, using ICSP.
5- When you are finished programming the test circuit, unplug the data cable that connects to the ICD-2.
6- Plug the test circuit into the wall, and you should see the LCD display come to life. The first message is: “Initializing Hardware”
7- After a brief pause, the test circuit will begin to display the voltage on each of the two ADCs. You can turn the pots with your fingers, and watch the voltage on the display change. Can you get the full 0 to 5 volt range from each pot?
8- Look carefully at the program source code that you just burned onto the PIC16F877 microcontroller, and draw a logical flow chart that shows what the software does. Make sure that each step makes sense to you.
9- You will note that there are quite a lot of additional things going on: conversion of 16-bit unsigned binary numbers to decimal values, the use of floating point math (can you find this in the code?), and the use of oversampling and median filtering to achieve a very stable voltage reading. This is a strategy similar to that used in digital volt meters, which only update their readings 2 or 3 times per second. Make sure you at least conceptually understand the source code before you go forward to the remainder of this module.

10-Experiment with how the voltage sampling is displayed. To begin with, instead of displaying decimal values, you can display the actual 16-bit values, one on each line of the display, as a string of 1's and 0's. Modify the microprocessor code to do this, but rename the file first, save the file, close the compiler, then re-open the compiler with the new file so that you do not lose the original source code file. This is one particularly annoying and quirky feature of the PIC-C compiler…even if you save a file that you are working on under a new name, it continues to compile the source code from the original file….well sometimes software is just that way. Best to rename, save, and restart. It only takes a few seconds.

11-Now, go back to the original source code. You will note that the code includes a "median filter". It is a median filter of rank 5. This means that it takes 5 samples before the current time point, 5 samples after, plus the "current" time point, for a total of 9 samples. It then puts these 9 samples in rank order (largest to smallest or vice versa), then pulls out the middle value, i.e., the median. In my experience, median filters are just about the most simple and useful filters for digital data. In some signal processing courses they are ignored because they are a non-linear filter, which means that a proof of how they work does not look pretty. But they work exceptionally well at totally removing a common type of noise called "salt-and-pepper noise", often found in digital data. This noise arises from many sources, but in practice it usually takes the form of spikes in the data. Most filters would allow these spikes to be counted into the data in some way. For example, a simple averaging filter (which calculates the numerical mean) would average in these spuriously high or low spike values. Before you use a median filter you need to be absolutely sure that the signal you are looking for is NOT composed of these "noise" spikes. Generally, a medial filter will eliminate spikes and other events that are equal to or less than the rank of the filter in width. In the case of the code in this module, a spike 5 samples wide would simply be ignored by the median filter. Median filters have many distinct advantages. They are very easy to implement in software, they do not distort edges, for example, square waves stay square unlike what happens in a low-pass filter, and they are very easy to tune…just adjust the rank up or down for more or less filtering. Enough about filters. Your assignment: modify the source code to eliminate the median filter, but leave everything else unchanged, and display the two pot voltages as decimal numbers, as was done with the original source code. Only now, every time you display a voltage it will only have been sampled once and not filtered.

12-Without a median filter, is the voltage value as stable? How much does it jump around?

13-Now modify the code to see how fast you can sample data. You will need to remove all superfluous delays. You'll need to reference the TLC4545 data sheet to determine what the minimum sampling and clock delays are. How can you determine how quickly you are actually sampling data?

# MODULE 08

## FEEDBACK

Was this Module useful and informative?

______________________________________________________________

Is there a topic that should get more or better coverage?

______________________________________________________________

In what way can this Module be improved:

Content: ______________________________________________

Depth of Coverage: ______________________________________

Style: ________________________________________________

Any additional comments that will help us to improve this course:

_______________________________________________________

_______________________________________________________

_______________________________________________________

If you prefer, you may e-mail comments directly to Bob Dennis: yoda@umich.edu