Name ___________________________________

Name ___________________________________

ME430 Mechatronic Systems:

Lab 6: Preparing for the Line Following Robot

The lab team has demonstrated the following tasks:

______________ **Part (A) Controlling 7-Segment Displays with Pushbuttons**

______________ **Part (B) Controlling 7-Segment Displays with the PIC**

______________ **Part (C) Reading and Displaying an Analog Input with the PIC**

Introduction:

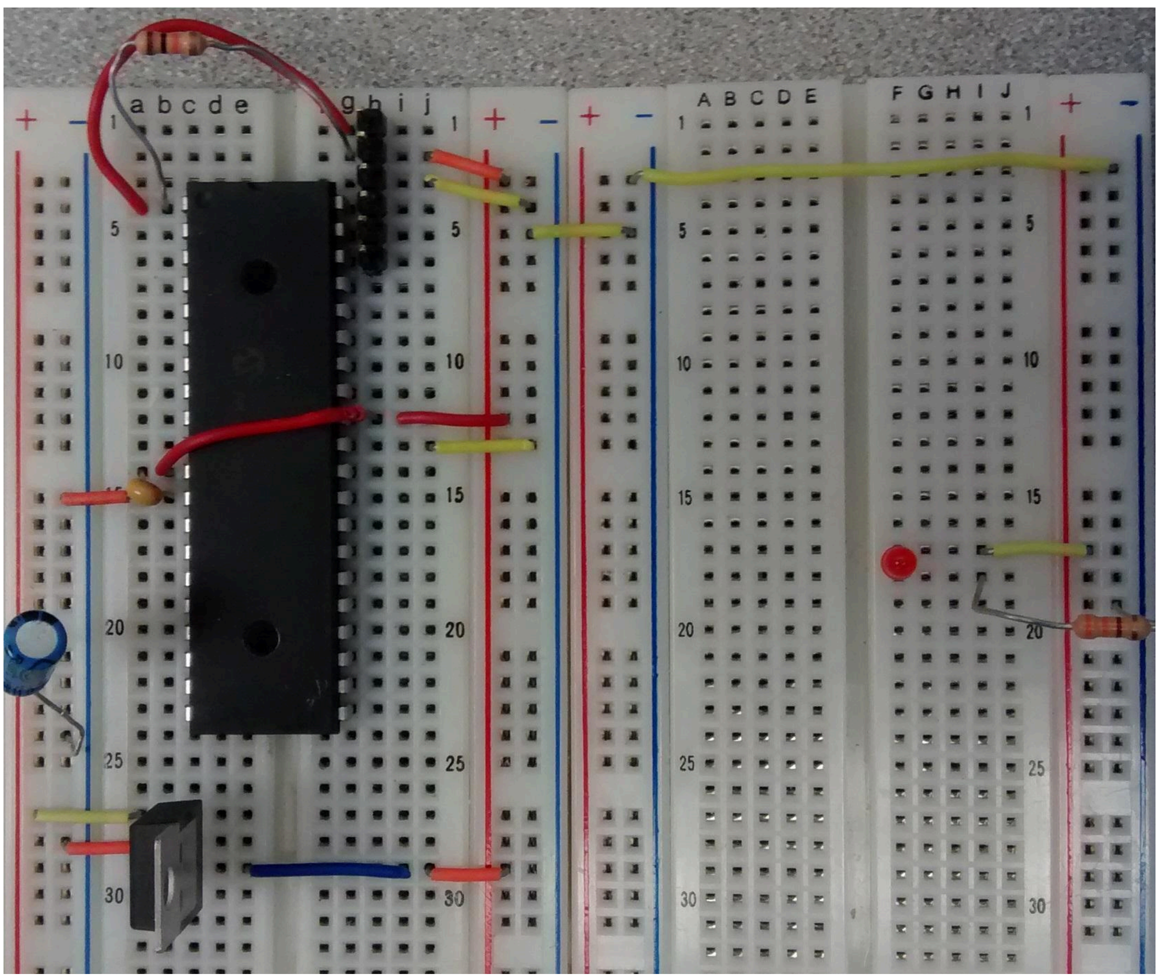
You may not realize it yet, but we are working towards creating a Line Following Robot out of our breadboarded PIC chip. It will run completely autonomously, and follow a line on the floor at the front of C111. Basically, we start the robot straddling the line, and the robot has two drive motors that drive straight along unless it senses that it has crossed the line on one side. Then it shuts off the motor on that side, which turns it back towards the line. It is a very simple control scheme which reads two sensors and drives two motors. Lab 5 was the start of the project, this lab (Lab 6) continues it, and Lab 7 will finish it! In this particular lab we are going to work on reading inputs and displaying outputs, so that we aren't trying to program the robot with no idea what values it is sensing.

Part (A) Controlling 7-Segment Displays with Pushbuttons

In this part of the lab we will learn about controlling 7-segment (LED) displays so that they can output numbers.

<u>Clean up the Breadboard:</u>

First, we want to clean up from the last lab. Remove any extraneous switches, LEDs, and chips so that we are back to just a programmable PIC and its associated circuitry. Leave in place only the PIC wiring to the PICKit, PIC power/ground, decoupling capacitors, the voltage regulator wiring, and the connections between the rails for power and ground (typically the inside two rails are 5v reg and the outside two rails are for unregulated power). Here is an example image of what your board ***might*** look like.

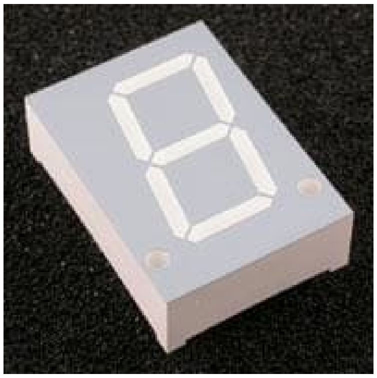


Check that you can still download some program (it doesn't matter which one) so that you know you haven't removed anything important.

_______It all still works (self check-off)

Basic 7-Segment Display Wiring:

Next, locate the 7-Segment Displays.



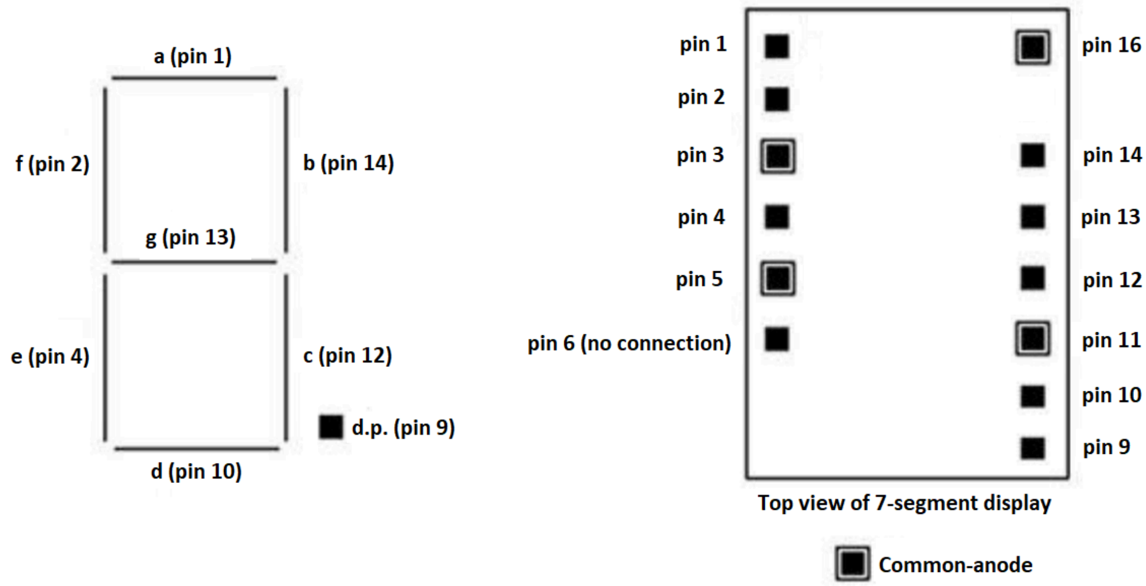
You should each have received one of these in your kit, for a total of (at least) two per team. Each of the 7 segments is a separate LED, and there are also 2 decimal points that are LEDs. As we know, LEDs need resistors so find sixteen 330 ohm resistors in your kits to go with the display.

Now, attach BOTH of your 7 segment displays to the very bottom of your breadboards, with the decimal points at the bottom. (See photo on next page for the definition of "top").

We want to make sure that we can turn on a single segment of these displays. Use a single 330 ohm resistor to connect the pin on the top left hand side (pin 1) of the left hand display to ground. Next, connect the pin on the top right (pin 16) of the left hand display to the unregulated power line. Connect the power. The top LED (segment a) on the left hand display should light up. Pretty nice!

Now, we need to learn a bit more about these 7-segment displays. Your 7-segment displays have an extra decimal point to the left of the decimal, which we will ignore. A 7-segment display has 8 LEDs in total (seven for the decimal and one for the decimal point) but pull one of them out of the board and count the pins. We might have expected to find 16 pins (two for each LED) but there aren't that many. In a way, this makes sense—they all need to get power and all need to go to ground (through a current-limiting resistor). 7-segment LED displays come in two types: common-anode and common-cathode. When you previously hooked up LEDs often one of the legs went into ground. Then, all of your LEDs were common-cathode since all of the cathode legs were held together. These 7-segment LEDs are common-anode so all 8 anodes need to be held high. In order to make a light come on you need a 330 ohm resistor going to ground. You only need 9 wires: one common-anode and 8 individual cathodes. For this particular 7-segment display there are 4 pins which are all the same common-anode and one pin that should not be connected to anything, so you should have 13 legs on your 7 segment display. (We don't know why they chose to have 4 common-anode pins instead of just one or why there is a pin that should not be connected to anything.)

Here is the pin layout for our 7-segment display:



Now we want you to display the number seven on one of the 7-segment displays. This will require you to turn on 3 of the segments, so be sure that you use 3 resistors. (Remember that each LED needs its own resistor!)

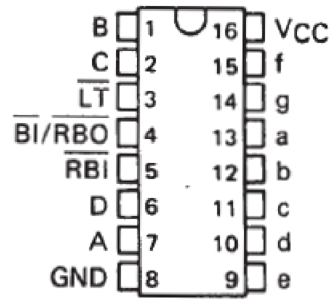
__________ We made the number seven! (Self check-off).

Controlling 7-Segment Displays with the 7447 chips:

Clearly we could program 7 pins on the PIC to control the 7 LEDs. However, that is going to occupy a lot of PIC output pins, and then we will still need two Darlington chips to provide adequate power to run the LEDs. It is very common to want to take a binary number and show it on the 7-Segment Display, so there is a special chip just for this purpose.

Our goal in this part of the lab is to use the 74LS47 ("BCD to 7 segment driver") chip and control a 7-segment display. (Be careful if you use this on your projects: the 7447 chip works with common-anode displays, like we have, but you need a 7448 chip if you buy common-cathode displays.)

The datasheet for the 7447 has this pinout:



The connections for this chip are shown below. (If you find this table difficult to read, or if you want to see more of the datasheet, you can Google "TI SN74LS47N datasheet". That's how we generally find the datasheets we use anyway.)

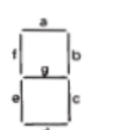
'46A, '47A, 'LS47 FUNCTION TABLE (T1)

DECIMAL OR FUNCTION	INPUTS						$\overline{BI}/\overline{RBO}$†	OUTPUTS							NOTE
	$\overline{LT}$	$\overline{RBI}$	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

H = high level, L = low level, X = irrelevant

NOTES:
1. The blanking input ($\overline{BI}$) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input ($\overline{RBI}$) must be open or high if blanking of a decimal zero is not desired.
2. When a low logic level is applied directly to the blanking input ($\overline{BI}$), all segment outputs are off regardless of the level of any other input.
3. When ripple-blanking input ($\overline{RBI}$) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go off and the ripple-blanking output ($\overline{RBO}$) goes to a low level (response condition).
4. When the blanking input/ripple blanking output ($\overline{BI}/\overline{RBO}$) is open or held high and a low is applied to the lamp-test input, all segment outputs are on.

† $\overline{BI}/\overline{RBO}$ is wire AND logic serving as blanking input ($\overline{BI}$) and/or ripple-blanking output ($\overline{RBO}$).



SEGMENT IDENTIFICATION

NUMERICAL DESIGNATIONS AND RESULTANT DISPLAYS

Now let's hook up the chips:

- Put the 7447 chips in the circuit boards above the 7-segment displays.
- Connect one of the common anode pins of the 7-segment displays to *unregulated* power.
- Connect the appropriate pins of the 7447 chip to *regulated* power and to ground.
- Figure out what to do with pins 3, 4, and 5, and do it.
- Connect the pins of the 7-segment display to the pins of the 7447 through 330 ohm resistors. (Don't worry about the decimal point right now—we'll connect that up separately later.)
- If you have this hooked up correctly, you should be able to hard-wire any binary number you want into pins A, B, C, and D of the 7447 chip and see the correct number on the seven segment displays. (Just run a wire to ground for binary 0 and run a wire to *regulated* power for binary 1.) Notice that numbers between 10 and 16 do give results on the 7-segment displays, although they are a bit strange.
- Hard-wire the 7447 chips so that the 7-segment displays show "42".

  _________ We made it display 42! (Self check-off).

- Now wire four basic switch circuits and connect them to the left 7-segment display's driver to pins A, B, C, and D. Figure out how to use the pushbuttons to count from 0 to 9.
- It should be VERY clear what D, C, B, and A represent. 0 will be all low, 1 will be only A high, 2 will be only B high, 3 will be BA high, … Put your switches in D, C, B, A order and it should be easy to figure out how to send any digit.
- When this is working call your instructor over to sign this part off on the first page.

Part (B) Controlling 7-Segment Displays with the PIC

In Part (A), we controlled the 7-segment displays “manually”, and in this part we will hook them up to the PIC to make the 7-segment displays count. Ultimately, we are going to put some more important outputs on the displays, but for now we just want to make sure that the PIC is connected properly to the displays and counting is a good check.

Although you have many output pins to choose from on the PIC, we would like everyone to make the same connections or we will never get all of them debugged! Connect the wires from the PIC to the 7447 displays so that, when you put a number on *PORTD*, that number shows up on the *left-hand 7-segment display*. Then connect the wires so that numbers written to *PORTB* appear on the *right-hand 7-segment display*.

Write a small piece of code that simply puts out a few numbers to the PORTs, and see if you can get this part to work.

______ We can use the PIC to write single digits to each of the two 7-segment displays (self check-off)

Now, write a program that “counts” and displays the numbers on the 7-segment displays. Start at 00 and increase. You don’t need to worry about leading blanks or what happens after you get to 99. When you have this working, call your instructor over to check you off on the front page.

Part (C) Reading and Displaying an Analog Input with the PIC

Now we have the 7-segment displays working well enough that we can use them to "print out" information from our breadboarded PIC. In this part of the lab, we will use the PIC to read in analog values from some potentiometers, much as we did with the PICDEM2 board in Lab 5. Then we will output a short form of the values to the 7-segment displays.

First, find two potentiometers. Each of your lab kits came with one, although they may not both look the same. Stick them in the breadboard somewhere not too far from the PIC. You will want one leg of each potentiometer to go to ground, and one leg to go to *regulated* power. We will use the third (middle, or wiper) leg of each potentiometer, which will vary from 0 to 5 Volts, as the input to the PIC. (Review the information on potentiometers in Lab 5 if this is confusing.)

Connect the wiper leg of one of the potentiometers to pin RA0 on the PIC. Write a short piece of code that uses the ADC to read the value from RA0, much as we did in Lab 5. Make sure that this is working properly by halting the program and using the watch window to examine the result.

Now, modify your code to display the hundreds digit of the analog value on the *left-hand* 7-segment display. Since the analog value goes from 0 to 1023, we should see numbers from 0 to 9, and then "c", displayed as we rotate the potentiometer. We might suggest a piece of code such as

```
sevenSegDisp=result/100;
PORTD=sevenSegDisp;
```

Make sure that RA0 is hooked up to the left-hand side display!

When the first display works properly, connect the other potentiometer to RA1 and display its hundreds digit on the right-hand side display. Call your instructor over to check off this part on the front page.