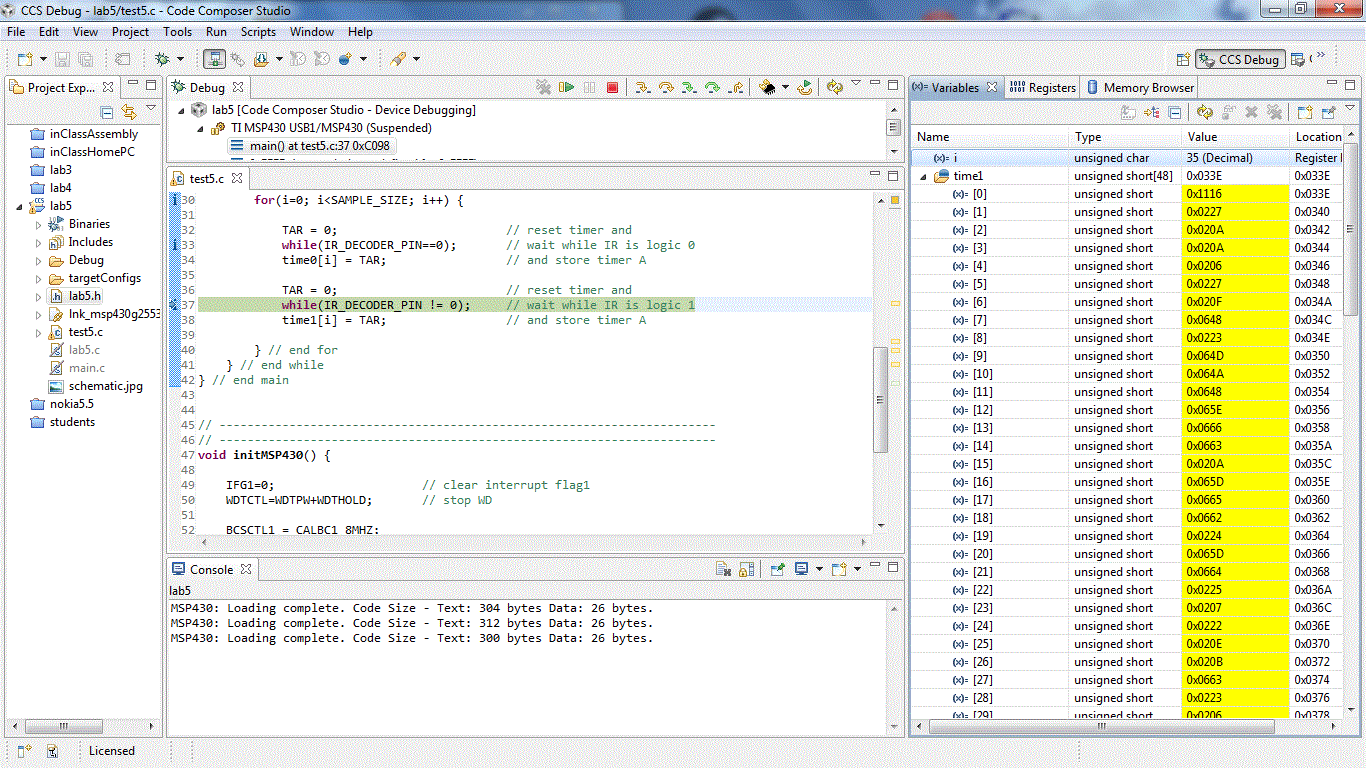
**Lab day 1**

Answers to these activities should be posted to GitHub.

**Timer Counts**

Build a project around test5.c and then download it onto your LaunchPad. Make sure to open the variables tab (View -> Variables). I also like to clear memory from the Memory Browser tab (View -> Memory Browser), Fill Memory from 0x200 to 0x400 with 0's. Run the program and then press a button on a remote. Then pause the program and look at the variables. You should see something like the following.   


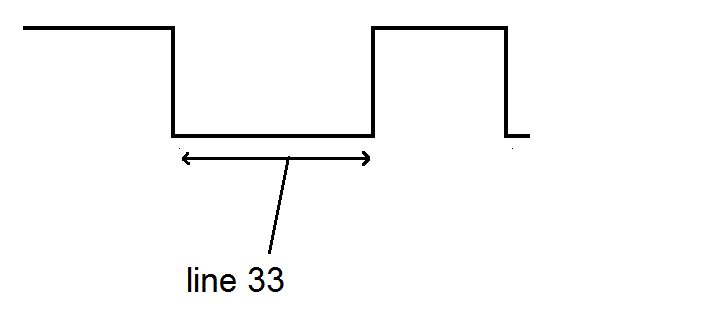
A few questions about what goes on in this program are in order before we proceed. 1) How long will it take the timer to roll over? 2) How long does each timer count last?

Answers:

16ms for the timer roll over.

I originally though (1/8E6)seconds; however, when double checking with the logic analyzer that was consistently getting the same values for the button presses, it showed that each timer count is (1/1E6)seconds.

The while(1) loop in main reads in the ir pules in the for loop.

Annotate the picture below to indicate which line of the for loop in the program is executed at which part of the pulse. You should show a total of 6 lines of code (lines 32-34 and lines 36-38).   


**IR data packets**

Before you start on this portion of the assignment, watch [Dave Jones' Trigger Hold-off Tutorial](http://www.youtube.com/watch?v=ta096oBzSac). You are going to need to use the O'scopes to examine the IR waveforms generated by a remote control of your choice.

Setup your LaunchPad like the picture below. Make sure to connect the power and ground in the correct order! Connect the the O'scope on the Vout pin of the Vishay Remote Control Decoder.   
![LaunchPad setup]

Set the trigger threshold to mid voltage and the trigger hold-off to accommodate an IR packet. On my remote control, this was about 80ms. Please note that remote control data packets are not standardized by any means, so the remote that you use to perform these experiment will almost certainly generate different results than those that your neighbor's will generate.

List the lengths of the pulses generated by the remote control in absolute time using the O'scope (3 significant figures) and in timer A counts. Note: "start logic 0 half-pulse" refers to the logic LOW portion of the start pulse, and "data 0 logic 1 half pulse" refers to the second half (which is a logic HIGH) of the pulse representing a zero bit.

| **Pulse** | **Duration (ms)** | **Timer A counts** |
| --- | --- | --- |
| Start logic 0 half-pulse | 8.4275ms | 8428 |
| Start logic 1 half-pulse | 4.140625ms | 4141 |
| Data 1 logic 0 half-pulse | 625us | 565 |
| Data 1 logic 1 half-pulse | 1.4843ms | 1513 |
| Data 0 logic 0 half-pulse | 625us | 560 |
| Data 0 logic 1 half-pulse | 421.9us | 418 |
| Stop logic 0 half-pulse | 571.875us | 565 |
| Stop logic 1 half-pulse | 481.25us | 470 |

Collect and tabulate in Excel 8 samples of timer A counts for each of the following pulse types (in decimal). Compute the average and standard deviation of each pulse type. I would suggest just grabbing it from the CCS variables tab.  
- Data 1, logic 1 half-pulse (high pulse with data 1) Standard Deviation = 18.629 and Average = 1503 counts

- Data 0, logic 0 half-pulse (low pulse) Standard Deviation = 2.8879 and Average = 560 counts

- Data 0, logic 1 half-pulse (high pulse with data 0) Standard Deviation = 20.597 and Average = 464 counts  
Ensure you label the rows and columns of your table so that I will know what the information in each cell means. For each pulse type list the range of timer A counts that would correctly classify 99.9999426697% of the pulses. This number has something to do with the [standard deviation](http://en.wikipedia.org/wiki/Standard_deviation#Rules_for_normally_distributed_data) (hint: look at the table in this section). Two Standard Deviations on either side of the Average Value counts

- Data 1, logic 1 half-pulse (high pulse with data 1) Range: (1465, 1541)

- Data 0, logic 0 half-pulse (low pulse) Range: (554, 566)

- Data 0, logic 1 half-pulse (high pulse with data 0) Range: (422, 506)

Write the codes (in hex) for several remote control buttons.

| **Button** | **code (not including start and stop bits)** |
| --- | --- |
| 0 | xC2CC |
| 1 | xC284 |
| 2 | xC244 |
| 3 | xC2C4 |
| Power | xC2D0 |
| VOL + | xC228 |
| VOL - | xC2A8 |
| CH + | xC298 |
| CH - | xC218 |