Welcome to ECE2049!

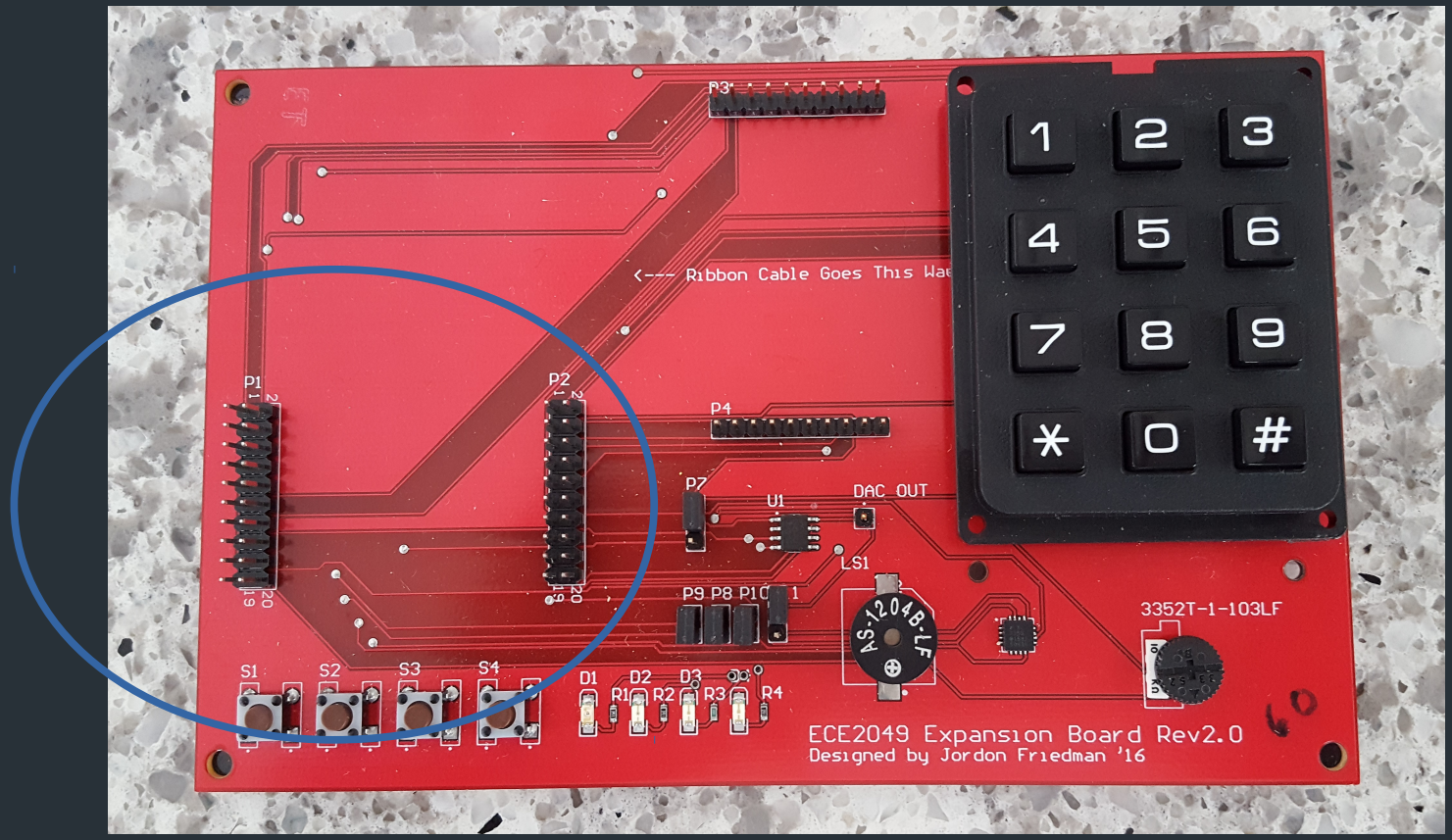
Lab 0: The Introduction

Welcome to Lab!

- Up first …. handing out NEW MSP430F5529 lab IO boards!
- You should have purchased your MSP430F5529 Launchpad
- You will be signing out a specific serialized IO board to use for the whole term.
- ***YOU MUST RETURN THE BOARD TO THE COURSE STAFF at the end of term to receive a grade. You break it, you buy it!***

Welcome to Lab!

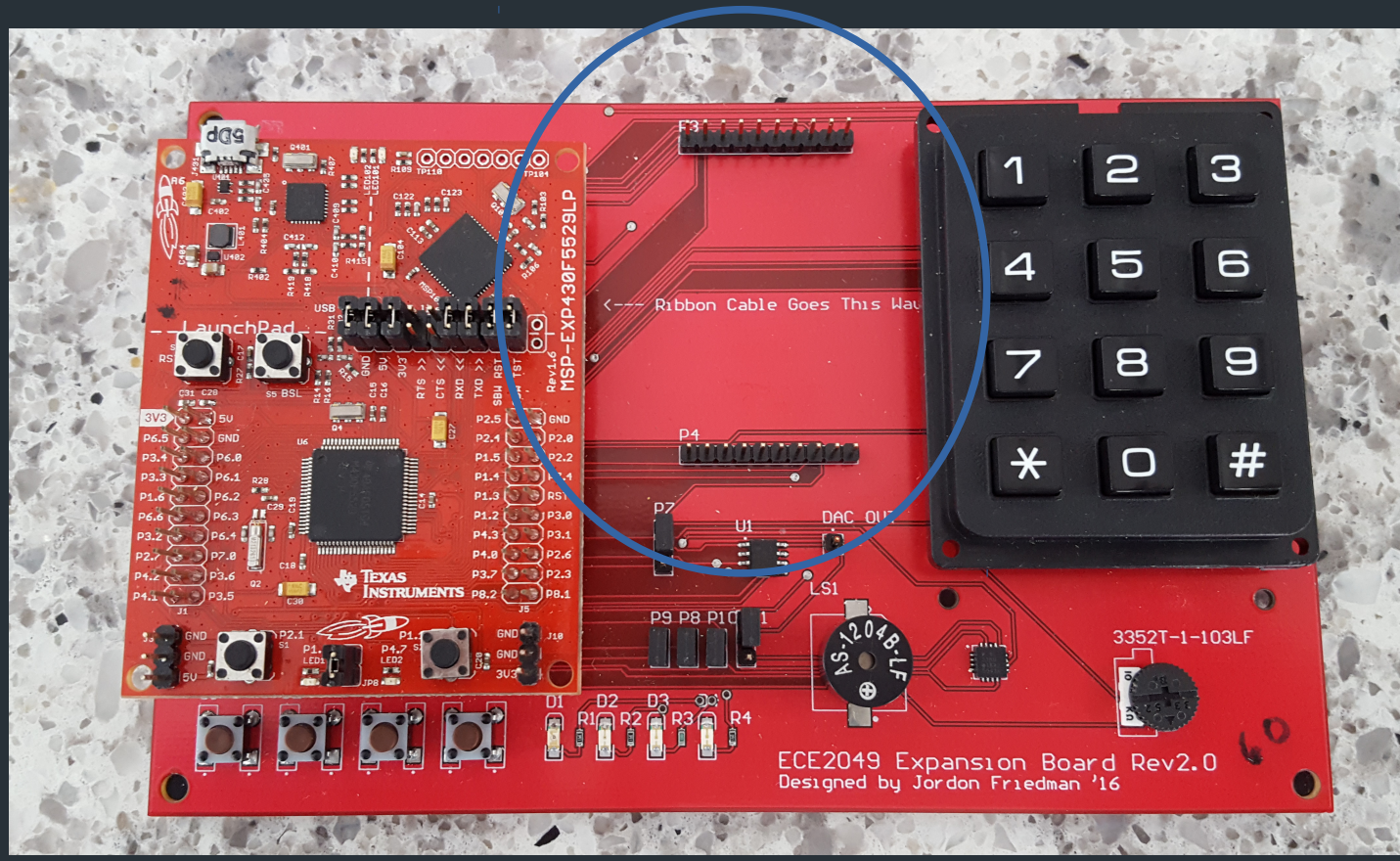
- Assembling the board is easy. Here is the IO board. The 40 pin jumper above the buttons is where the Launchpad goes



Welcome to Lab!

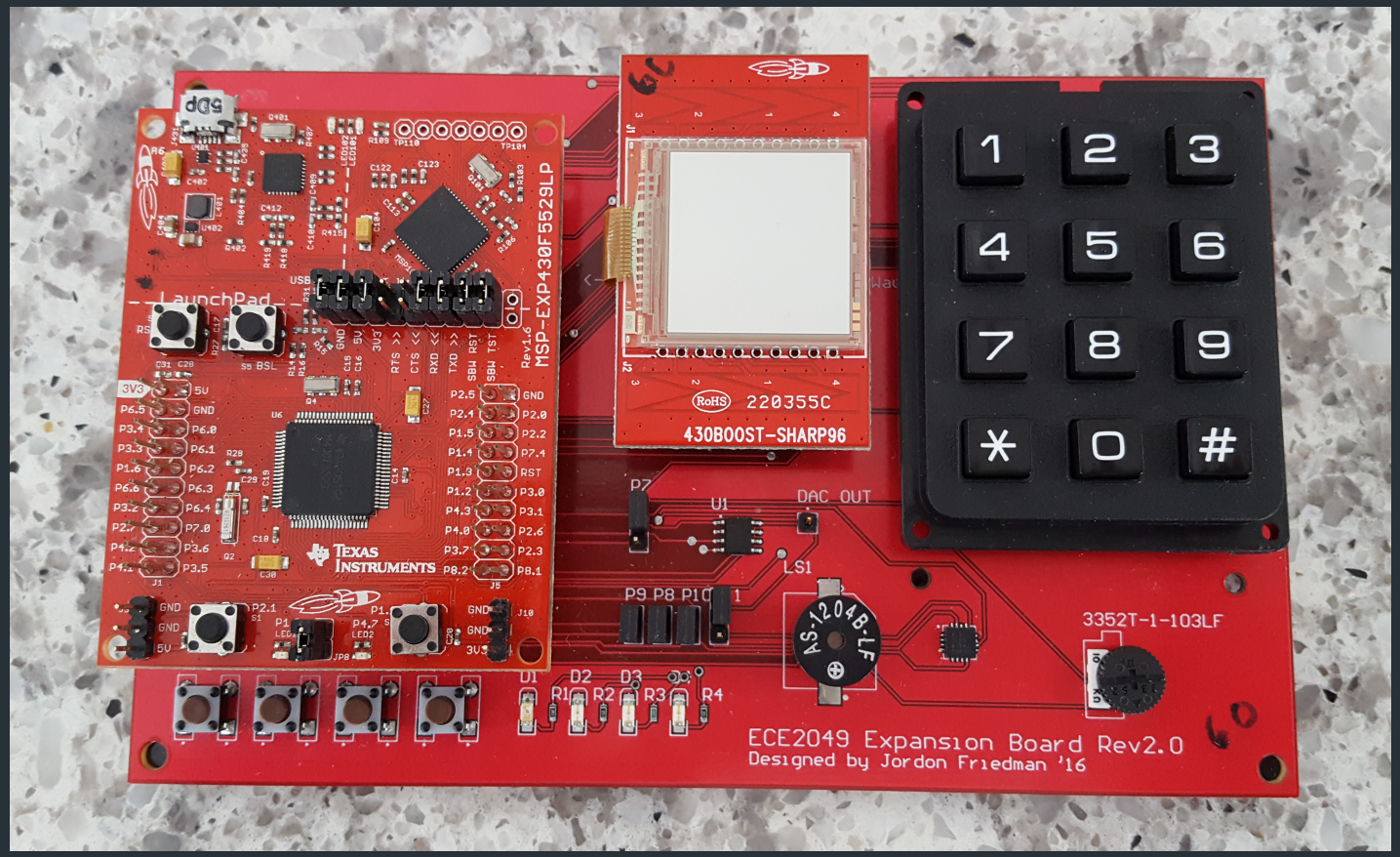
- Be SURE the Launchpad is correctly seated on 40-pin header!!!
- Next the LCD goes on the 20-pin header

- RIBBON CABLE goes to the LEFT!



Welcome to Lab!

- Your board should now look like this.
- You should also have been given a protective box.
  → USE IT! You will need this board all term



Welcome to Lab!

- Make sure your board has these five 2-pin Jumpers
- DOUBLE CHECK that everything is properly mounted and seated before powering the board up!!

Welcome to Lab!

- The most common failure point on these boards is the LCD screen
- The headers on the LCD are rather weak and break off easily. To prevent breakage
    - Do not move the screen!!
    - Seriously, be careful with the SCREEN

Welcome to Lab!

- OK, onto the fun stuff …
- Feel free to get started until we start talking!
- The lab document is located on the course website: http://users.wpi.edu/~ydoroz/ece2049/ece2049_labs.htm
- Please work in groups of two
  - You may work alone if there is space
- Let us know if you have any questions!

How labs work

- Attend YOUR lab section each week (all are full)
- Instructions and requirements for each lab are on the course website
- Complete the pre-lab and then work through the lab requirements (get pre-lab signed off!!)
- Ask us if you have questions or need sign-offs

We are your course staff— We're here to help you!

How embedded programming works

Or not …

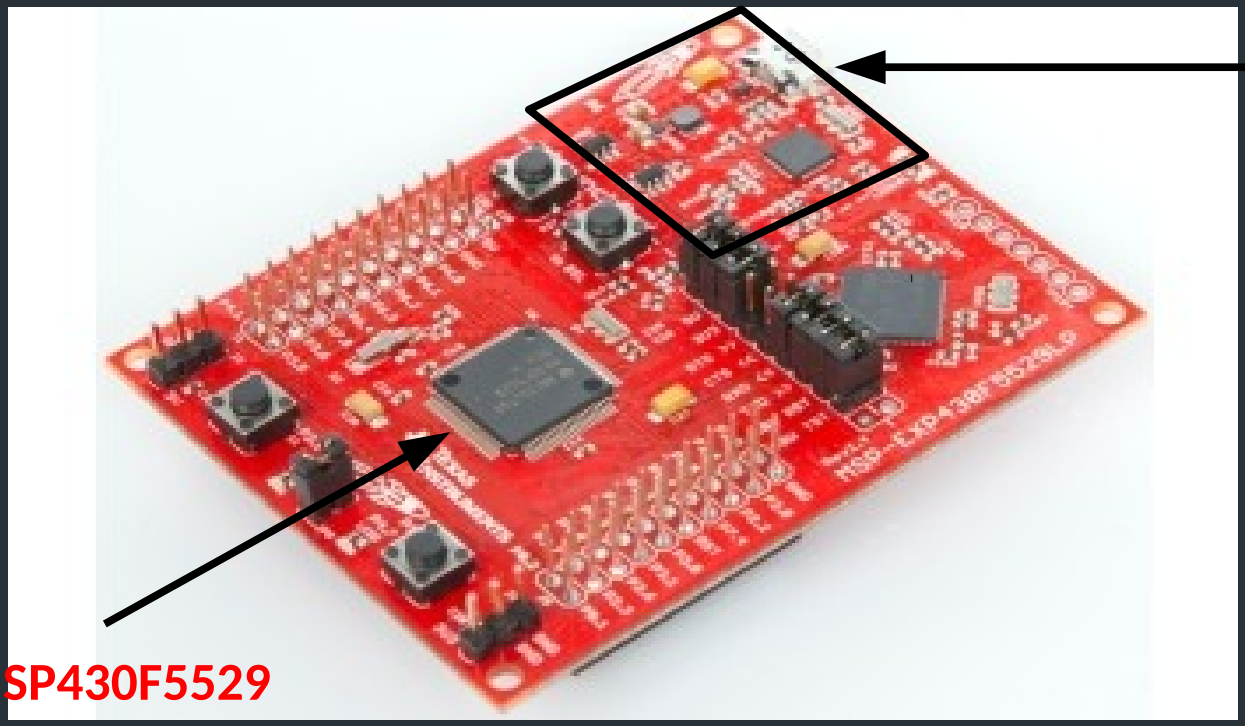
1. Write some code
2. Compile your code (fix it, test it, repeat, again …)
3. Load the compiled code onto the MSP430
4. Run your code on the MSP430 using the debugger

The last step is different than using an Arduino (RBE folks) or just running programs on your local machine:

You get to see how your code executes!

Our *NEW* MSP430 Boards

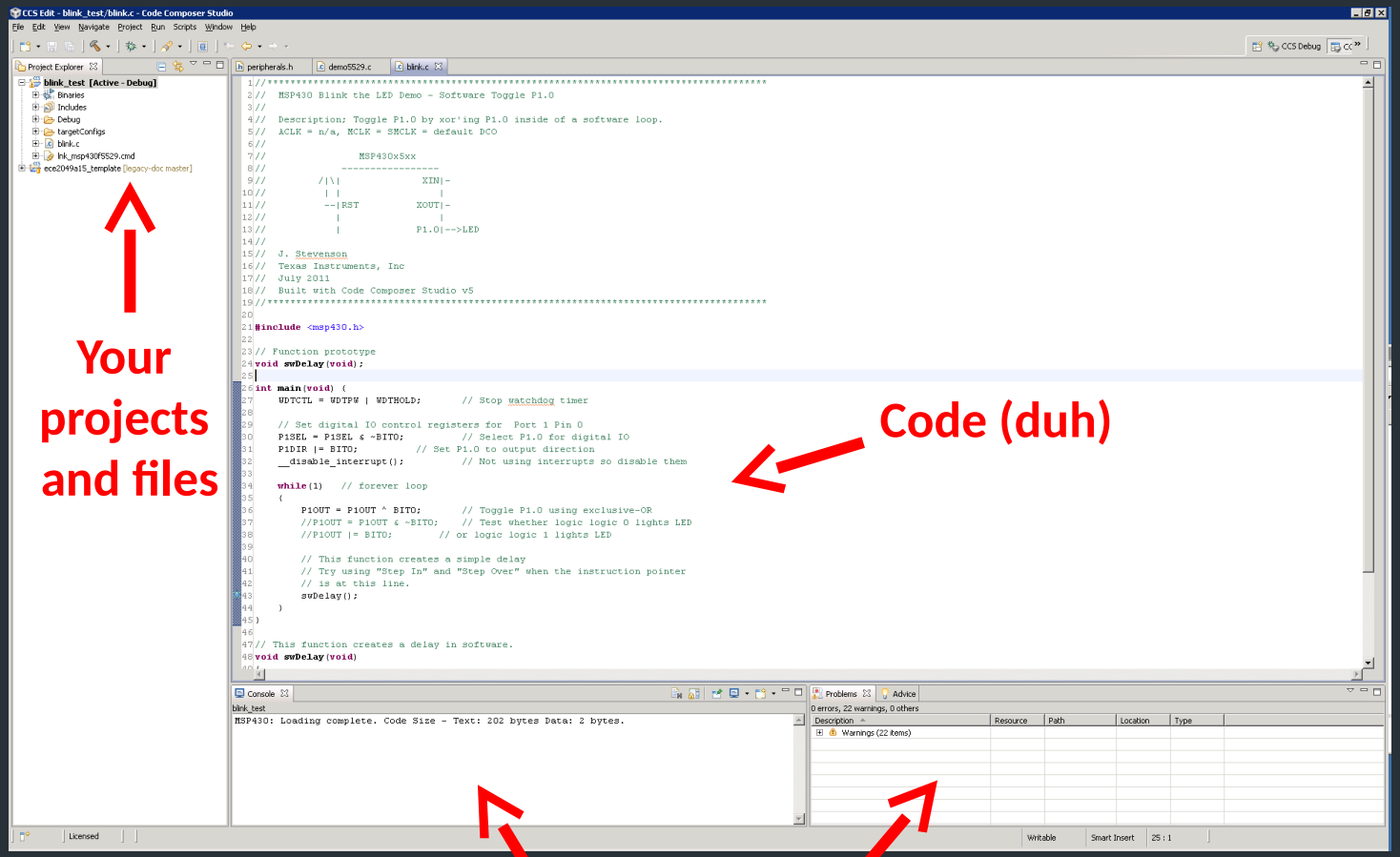
Debugger
(Allows us to program the target,
Actually uses another MSP430!)



Debug
USB Port

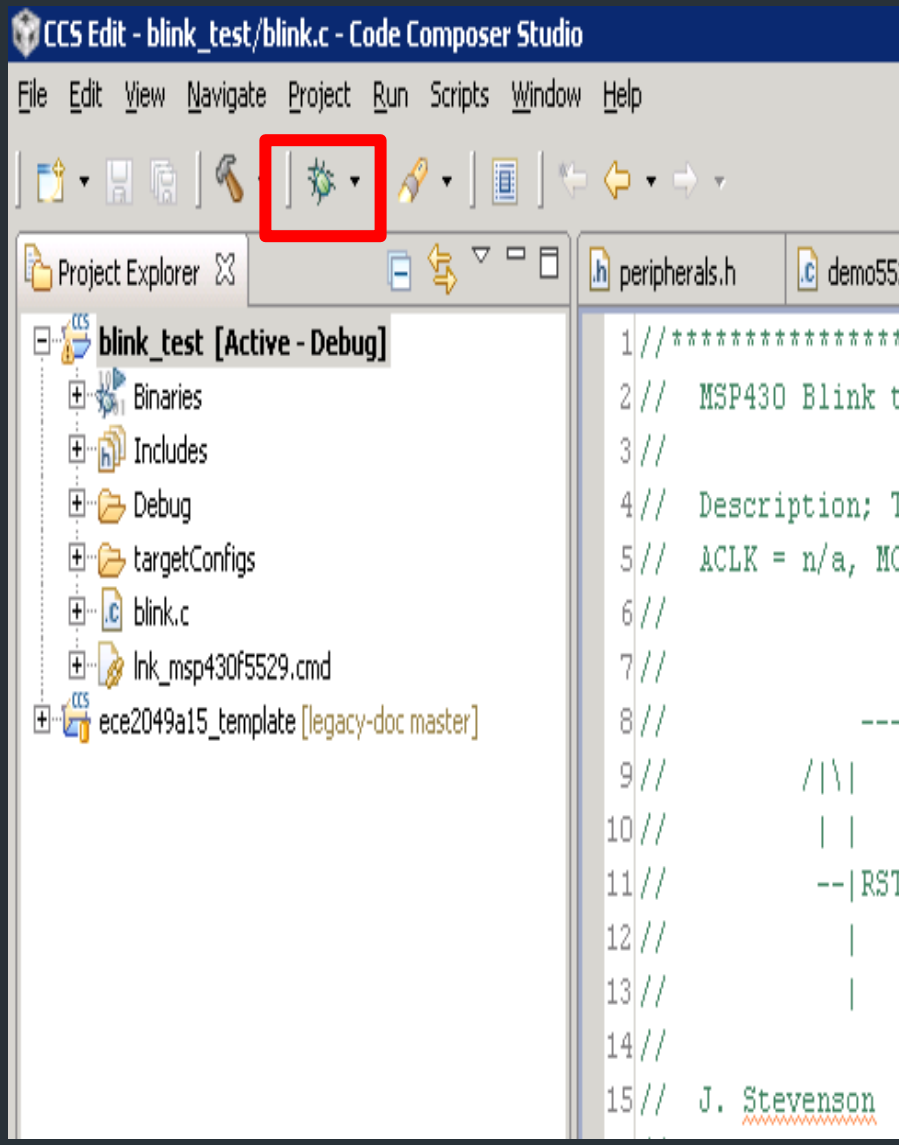
The MSP430F5529
we program in lab
(Called the "Target")

Code Composer Studio (CCS)

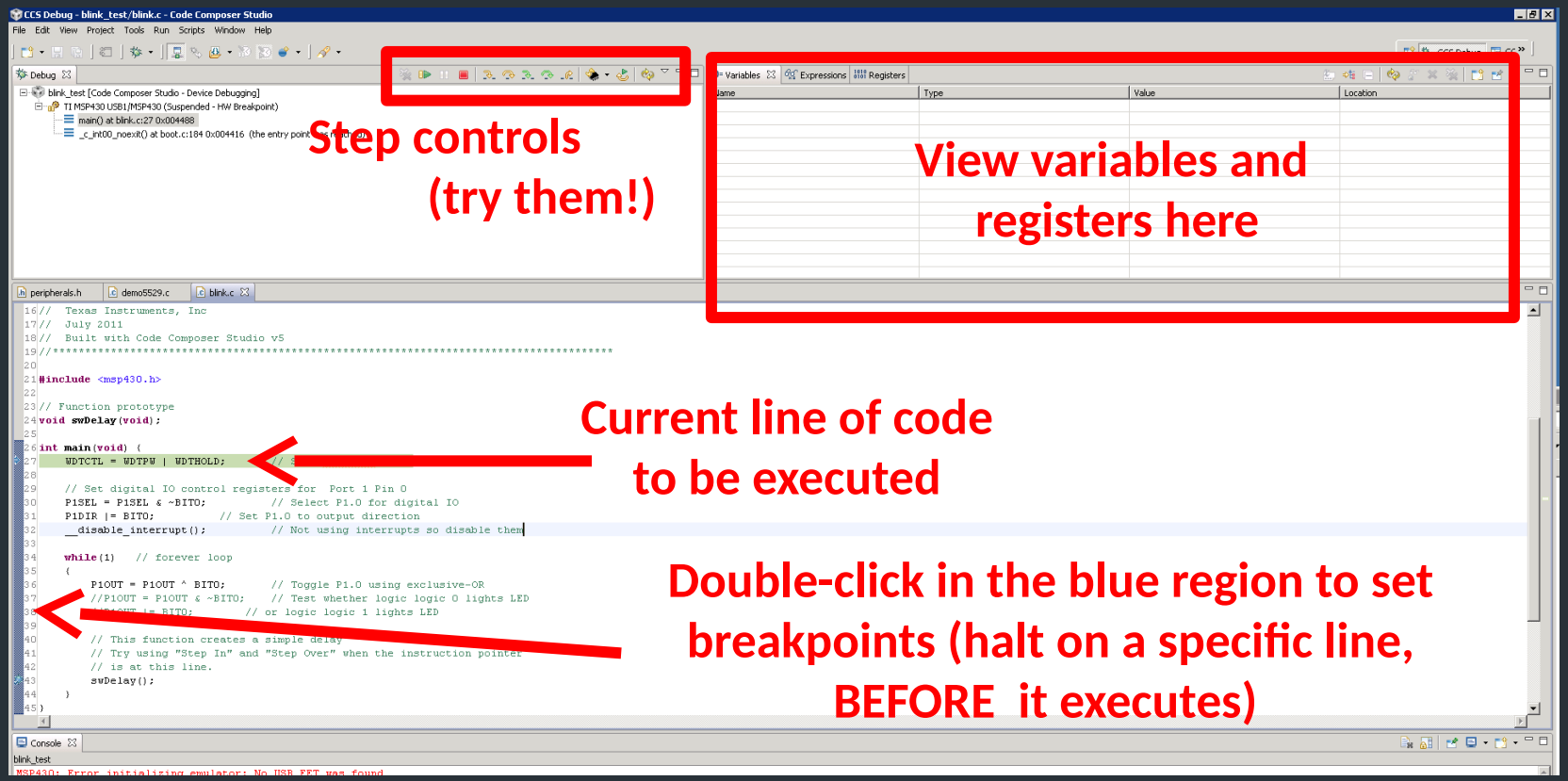


Debugging

- Click “Debug” to compile and load your code
- Errors and warnings appear in the console
  - If errors exist, you must fix them before running
- Every time you change the code, you must build and debug it again!
  - New code must be loaded onto the chip!



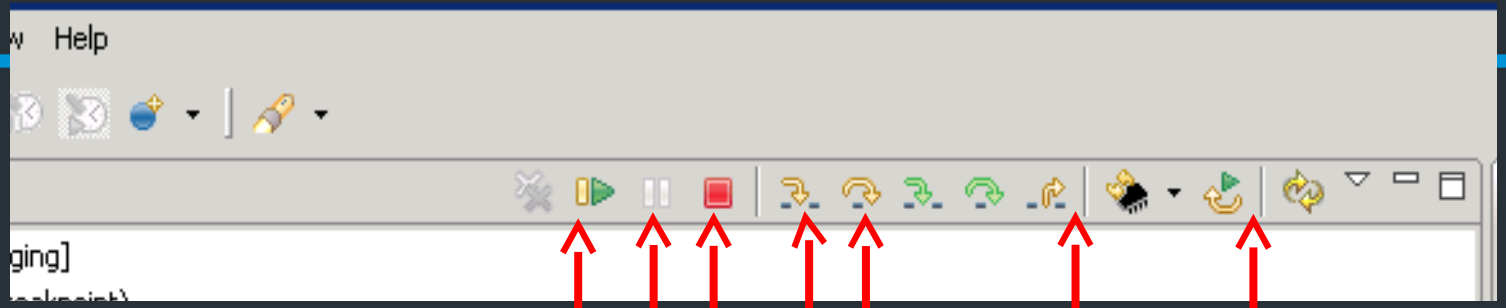
Debugging Perspective



Try to play with these features to see how they work.
They will help you **a lot** in future labs.

Feel free to ask us if you have questions!

Debugger Controls



Run to next breakpoint

Pause execution

End debug session

Step In

Step Over

Step to function return

Reset to beginning of program

These are the most important controls and they're covered in the video. Give them a try to figure out exactly how they work!

A VERY important warning

- When the debugger is running (stop button is Red),

DO NOT UNPLUG THE MSP430 BOARD!!!!

  - Doing so can PERMANENTLY damage the board. You will be sad and have to buy a new one. :-(
    - PLEASE DO NOT “BRICK” THE LAUNCHPAD!!

  - Instead, end your debug session by pressing the Stop (or “Terminate”) button or by clicking Debug again

Good luck and have fun!

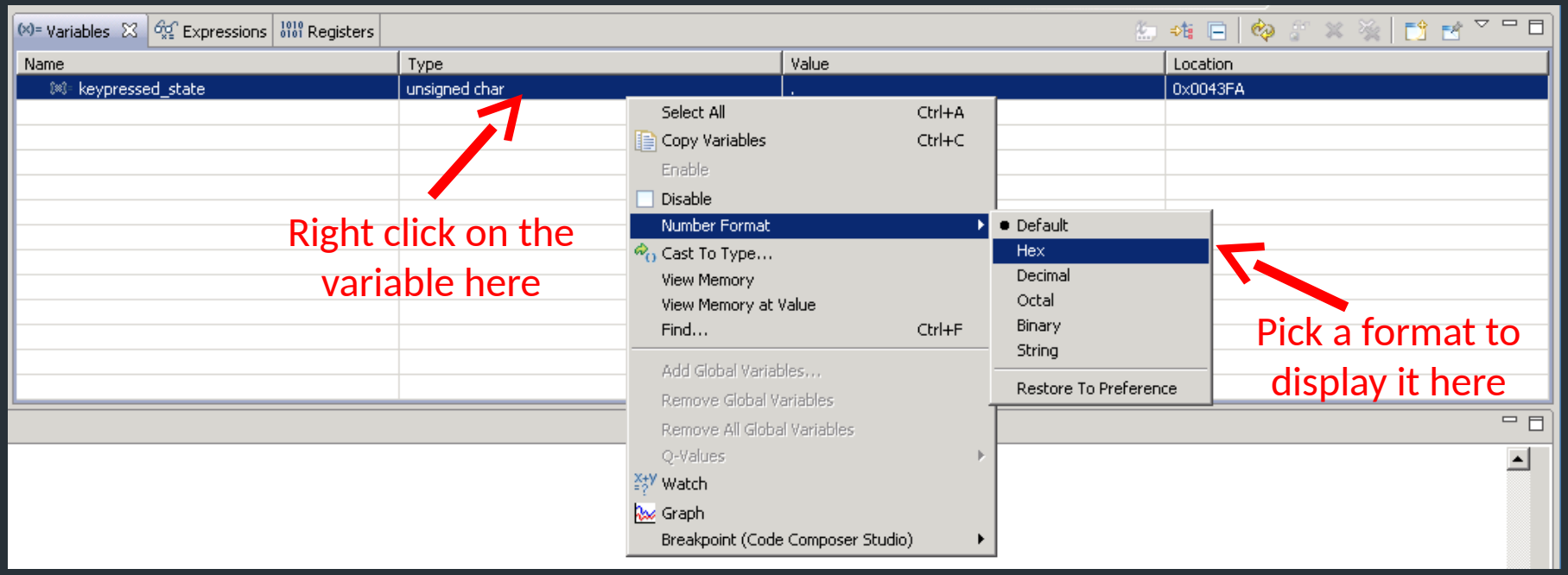
- Feel free to ask us if you have questions—we're here to help!
- These slides are available for you to view later
  - Additional notes and Frequently Asked Questions are at the end—take a look at these!

Thank you for listening!

Frequently Asked Questions (1/?)

- “Help! I lost a window in CCS!”
- Reset all of your windows to defaults, go to :
  - View > Reset Perspective

- See the next slide for more info…

Frequently Asked Questions

- "Why does this variable have a value of '.'?"
  - This is because CCS is trying to show it as an ASCII character, but it has a value that isn't a character. You can change the format like this: