

CU ECEN3/5730

***Workbook for Practical PCB Design and
Manufacture***

Fall 2025

Eric Bogatin

Updates by Tim Swettlen

Addie Rose Press

Sixth Edition

Copyright 2023 by Eric Bogatin

All rights reserved

ISBN xxxxxxxxxxxxxxxx

A Publication of Addie Rose Press



Table of Contents

Table of Contents

Chapter 1 About the ECEN 3730/5730 class and labs13
1.1 Name badges.....	.13
1.2 Daily Quizzes.....	.13
1.3 Assignments and your portfolio page.....	.14
1.1 Four types of assignments.....	.15
1.2 Principles of the labs19
1.3 Collaboration is Encouraged20
1.4 Most important rule for the lab: Safety comes first20
1.5 Your Lab kit20
1.6 Making Up Missed Labs.....	.20
Chapter 2 Board design projects.....	.21
2.1 Bare boards and assembled boards.....	.21
2.2 Board Design Milestones21
2.3 Circuit board build schedule22
2.4 A Special offer for all PCB Designers22
Chapter 3 Build Your Portfolio.....	.23
Chapter 4 Lab 0: Your name badge and your business card25
4.1 Your name badge.....	.25
4.2 Adding a QR code to your circuit board26
4.3 Directions to make our name badge circuit board28
4.4 Your name tag board assignment and rubric.....	.29
Chapter 5 BOARD 1: practice board.....	.29
5.1 Purpose of this board30
5.2 Adding a QR code to your circuit board34
5.3 Create a Business Card.....	.35
5.4 Part 1: submit your design files.....	.35
5.5 Part 2: Bring up and test38
5.6 Part 3: Final report for brd 1 and grading rubric:.....	.40
Chapter 6 Lab 1: Mastering the scope.....	.42
6.1 Signals and Noise: Why the scope will be the first instrument you will turn to42
6.2 Before you come to the lab.....	.43
6.3 Specific don'ts when using a scope43

6.4 Get familiar with the scope using BNC and minigrabber using the compensation signal	45
6.5 The 10x probe	45
6.6 Master the scope	46
Chapter 7 Lab 2: SBB-build 555 timer.....	47
7.1 What you will do in this lab	47
7.2 The solderless breadboard	47
7.3 Here are two timer options:.....	49
7.4 Purpose of this lab:.....	49
7.5 The circuit you will build	50
7.6 The rest of the circuit	52
7.7 Some hints	53
7.8 What you should pay attention to:.....	54
7.9 Check off with your TA.....	54
7.10 Your lab report	55
7.11 Grading rubric: 3 points total	55
Chapter 8 Brd 1: Schematic CDR.....	56
Chapter 9 Brd 1: Final CDR	56
Chapter 10 BOARD 2: Switching noise with good and bad layout	58
10.1 Before starting this board	58
10.2 Board 2 Design Process	58
10.3 Brd 2 POR	59
10.4 New parts in this board	60
10.5 Part 1: Submit design files	63
10.6 Part 2: bring up and test of brd 2	64
10.7 Rail compression- (Grad students only)	66
10.8 Rubric for your final report (10 points)	66
Chapter 11 Lab 3: Thevenin model of the waveform generator	69
11.1 Before you come to this lab	69
11.2 Cable connections for this lab	69
11.3 Exp 1: Setting up the scope and waveform generator	69
11.4 Exp 3: Measure the Thevenin model of the waveform generator	70
11.5 Exp 4: Optional Experiment, An alternative way of measuring the Thevenin resistance	72
11.6 Analysis and so what?.....	75
11.7 The lab report	77
11.8 Grading rubric:	77

Chapter 12 Lab 4: Using the Scope to measure loop to loop cross talk	78
12.1 Purpose	78
12.2 Before you start this lab.....	78
12.3 Exp 1: Set up the waveform generator as an aggressor loop	78
12.4 Exp 2: measure the cross talk between the aggressor loop and the 10x probe as the victim loop.....	79
12.5 Check off by your TA	80
12.6 The lab report:.....	80
12.7 Grading rubric:	80
Chapter 13 Lab 5: SBB PDN and slammer circuit.....	81
13.1 Important safety tip	81
13.2 Purpose of the lab	82
13.3 Prep before you start this lab	83
13.4 The big picture	83
13.5 Exp 1: Install the Arduino IDE.....	84
13.6 Exp 2: Build the slammer circuit with no decoupling capacitors and use the slow edge from the op amp.	85
13.7 Exp 3: Power consumption and dissipation and not blowing up a component	88
13.8 Exp 4: Set up the two driver signals to the transistor's base	89
13.9 Exp 5: Measure switching noise with a slow rise time and no decoupling capacitors	89
13.10 Exp 6: Measure switching noise with a fast rise time signal	92
13.11 Exp 7: Add the decoupling capacitor	92
13.12 Exp 8: Estimate the size of the decoupling capacitor you need to use	93
13.13 Exp 9: change the location of the decoupling capacitor	94
13.14 Check out by your TA	95
13.15 In your report, you should include	95
13.16 Grading rubric:	96
Chapter 14 Lab 6: Assembly practice board.....	97
14.1 Purpose of this lab	97
14.2 What you will need.....	97
14.3 What you will do.....	98
14.1 Exp 1: Clean your soldering iron tip	101
14.2 Exp 2: soldering 1206 0 ohm jumpers.....	102
14.3 Exp 3: solder smaller parts.....	103
14.4 Exp 4: removing solder bridges.....	103
14.5 Assemble a QFP part to the board	104

14.6 What you will turn in to complete this lab.....	105
Chapter 15 Lab 7 Assembly practice: build and characterize hex inverter board	107
15.1 Purpose of this lab	107
15.2 Before you begin this lab.....	107
15.3 The board	107
15.4 Assembling your board.....	108
15.5 Bring up and test your functional board.....	111
15.6 Build a ring oscillator	112
15.7 Does your board "work"?	113
Chapter 16 Lab 8: SBB version of the hex inverter circuit	114
16.1 Purpose of this lab	114
16.2 Exp 1: Build the circuit.....	114
16.3 Exp 2: Measure the switching signals	115
16.4 Exp 3: measure the quiet LOW and HIGH signals.....	115
16.5 Grading rubric: 3 points total	116
Chapter 17 Lab 9: measure cross talk between signal-return loops in a special test board	117
17.1 So What: Best probing practices for low cross talk	117
17.2 Purpose of this lab	117
17.3 What you will need.....	118
17.4 Prep before you start this lab	119
17.5 What you will do.....	120
17.6 Some background.....	120
17.7 Part 1: Cross talk with a continuous return plane	122
17.8 Part 2: Cross talk with no plane, but an adjacent return trace.....	125
17.9 Lab report and grading rubric.....	126
Chapter 18 Lab 27: Best measurement practices for high-speed signals to reduce artifacts	128
18.1 Exp 1: set up the Arduino as an aggressor source.....	128
18.2 Exp 3: measure the Arduino digital pin with the shortest tip loop inductance possible with the spring ground tip	130
18.3 Exp 4: why the 1x setting should not be used for signals with a rise time longer than a few microseconds.....	131
18.4 Exp 5: Impact of tip loop-inductance on probe-to-probe loop cross talk.....	132
18.5 Exp 6: probe to probe cross talk with ground spring tips.....	132
Chapter 19 Brd 1: bring up, test and evaluate	135
19.1 Assembly practice of your Brd 1 practice board.....	135
Chapter 20 Brd 2: Final CDR	136

Chapter 21 BOARD 3: A Golden Arduino PCB	138
21.1 Purpose of this lab	138
21.2 The POR	138
21.2.1 Special features and components to add.....	139
21.2.2 Graduate students are required to add the following features to their Arduino boards:	140
21.2.3 Develop and document your POR:	140
21.2.4 Do not just copy a reference design.....	141
21.3 BOM and non-commodity parts	142
21.3.1 The USB to UART chip is the CH340g.....	142
21.3.2 Crystal circuits	143
21.3.3 Debounce circuits and switches.....	146
21.3.4 The ATmega 328 microprocessor.....	148
21.3.5 The TVS chip	150
21.3.6 Using a ferrite bead to filter noise from the PDN to the AVCC pin of the 328	151
21.4 Part 1: Complete the schematic and layout for the Golden Arduino board	152
21.4.1 Use reference designs as a guide but take responsibility	152
21.4.2 Complete the schematic	152
21.4.3 Stackup and board outline	152
21.4.4 Start the layout.....	153
21.4.5 Add the header pins in the correct locations on your board	153
21.4.6 Layout tips: design for connectivity, signal integrity, assembly, test and bring up	155
21.4.7 Design for performance.....	156
21.4.8 Watch out for these common problems	157
21.4.9 Review your design yourself	157
21.4.10 Perform a CDR for brd 3	158
21.4.11 Export your Gerber and design files	158
21.4.12 Submit your board to JLCpcb to use their DFM	159
21.4.13 Part 1 grading rubric	160
21.5 Part 2: bring up, boot load and test of brd 3.....	160
21.5.1 Assembly and bring-up.....	160
21.5.2 Burn the bootloader	161
21.5.3 Bring up and test.....	165
21.5.4 Debugging.....	165
21.5.5 Here are some of the common errors you might encounter.....	166
21.6 Measuring the switching noise on your Arduino board and compare with the commercial board.....	168

21.6.1 Measure the inrush current from a USB power rail or a 5 V AC to DC converter	168
21.6.2 Near Field Emissions.....	169
21.6.3 Your report for bring up and test.....	169
21.7 Step 3: Final report- counts as a midterm.....	169
21.7.1 Grading rubric.....	169
Chapter 22 Lab 10: boot load a 328 uC.....	171
22.1 The 328 uC.....	172
22.2 Solder the header pins to the CH340g to UART interface board	174
22.3 Burn the bootloader.....	174
22.4 Step 3: USB Programming.....	183
22.5 Step 4: Confirmation of successful bootloading.....	187
22.6 Grading rubric	187
Chapter 23 Lab 11: measure trace resistance.....	188
23.1 Before you come to the lab	188
23.2 What you will do in this lab.....	188
23.1 Exp 1: Analyze the Test Board.....	188
23.1 Exp 2: Estimate the resistance of each trace	189
23.2 Exp 3: Measure the resistance of each trace using a 2-wire DMM.....	190
23.3 Exp 4: Measure the resistance of each trace using the 4-wire method.....	191
23.4 Grading rubric	192
Chapter 24 Lab 12: blow up traces.....	193
24.1 Exp 1: estimate the max current handling capacity of a trace.....	193
24.2 Exp 2: increase the current through a trace and feel it get warm	193
24.3 What you will turn in or complete for this lab.....	194
24.4 In your lab report.....	195
24.5 Grading rubric:	195
Chapter 25 Lab 13: Build a crystal oscillator in a solderless breadboard	196
25.1 Exp 1: A ring oscillator	196
25.2 Experiment 2: Adding a 1 Meg resistor in the feedback loop.....	196
25.3 How a crystal works.....	197
25.4 Experiment 3: Measure the impedance of the crystal or ceramic resonator	198
25.5 Experiment 4: Add a crystal in the feedback loop.....	202
25.6 Experiment 5: Add a large value resistor to the feedback loop	204
25.7 Experiment 6: add small capacitors to suppress higher order modes.....	205
25.8 Grading rubric:	206

Chapter 26 Lab 14: The TVS diode array to protect against ESD events.....	207
26.1 Grading rubric:.....	210
Chapter 27 Lab 15: my good-bad switching noise board.....	211
27.1 Purpose of this lab.....	211
27.2 The board you will measure	211
27.3 Beyond "working"	213
27.4 Circuit Design for brd 2.....	213
27.5 Quiet LOW and quiet HIGH pins.....	215
27.6 What you will measure on our board.....	216
27.7 My results.....	218
27.8 In your report, you should include.....	221
27.9 Special notes for your lab report.....	221
27.10 Grading rubric.....	222
Chapter 28 Your Brd 2 bring up, test, evaluate.....	223
28.1 Assemble your brd 2	223
28.2 Testing your brd 2.....	223
28.3 Rubric.....	223
Chapter 29 Brd 3: final CDR	223
Chapter 30 BOARD 4: A 4-layer Instrument Droid board	225
30.1 The purpose of the board.....	226
30.2 Principle of operation.....	226
30.3 The design goals	228
30.4 The POR for brd 4	229
30.5 New components.....	229
30.6 The schematic and layout	231
30.7 Submitting the design files and grading rubric	232
30.8 Part 2: assemble and test your board.....	232
30.9 Part 3: document your board and rubric	232
Chapter 31 Lab 16: diff or SE signaling and ground noise	233
31.1 Set up the SBB.....	233
31.2 Exp 1: setting up both Differential and single-ended measurements.....	242
31.3 Exp 2: Generating a voltage drop between the local grounds	247
31.4 Layout considerations for differential pairs	250
31.5 A "Noisy" Ground.....	251
31.6 Check off by your TA	252

31.7 The Lab Report.....	253
Chapter 32 Lab 17: I2C communications.....	254
32.1 The I2C bus	254
32.2 Impact from the capacitance on the I2C bus	255
32.3 The internal pull up resistor on the 328	256
32.4 Exp 1: measure the I2C signals	257
32.5 Exp 2: change the pull up resistance and see the change in the rise time of the signals	258
32.6 Adding test points and indicator LEDs to the SCL and SDA lines.....	258
32.7 Cross talk between the SCL and SCK lines	259
32.8 In your report	260
Chapter 33 Lab 18: Measure the in rush current and operation current of a board.....	261
33.1 The principle of a current sense resistor	261
33.2 Build or select a circuit in which to measure the current draw	262
33.3 A simple 555 timer circuit.....	263
33.4 Measuring a differential voltage with two single-ended probes.....	266
33.5 Measuring the inrush current with two single ended probes	269
33.6 In your lab report, you should:	271
33.7 Grading rubric	271
Chapter 34 Lab 19 4-layer via to via cross talk board	272
34.1 Exp 1: Measuring the cross talk when there are no return vias	273
34.2 Exp 2: Measuring the cross talk when there are return vias	273
34.3 In your report	274
Chapter 35 Brd 3 assemble, bring up, boot load, test	275
Chapter 36 Lab 20: Switching noise in commercial Arduino and golden board.....	276
36.1 What you will do in this lab.....	276
36.2 The switching noise shield	276
36.3 Exp 1: Write the microcode.....	277
36.4 Exp 2: Measuring the switching noise on the quiet LOW	278
36.5 Exp 3: Measure the noise on the power rail when the microcontroller itself is the aggressor	279
36.6 Exp 4: Measure the on-board power rail noise when the board is the aggressor	279
36.7 Exp 5: Measuring the near field emissions from bottom of the board.....	280
36.8 Exp 6: generic measurements to characterize any Arduino Board	280
36.9 Your Lab Report	281
Chapter 37 Lab 21 SBB version of brd 4	283
37.1 Power consumption considerations	283

37.2 Setting a max current, or max voltage drop.....	284
37.3 The circuit	284
37.4 Building the circuit and the code	286
37.5 In your report	Error! Bookmark not defined.
37.6 Grading rubric:	294
Chapter 38 CDR of Brd 4.....	295
Chapter 39 Lab 22: Applications of brd 4 using the SBB version.....	297
39.1 Exp 1: finalize the hardware and understand your code.....	297
39.2 Exp 2: Measure something for which you know the answer	297
39.3 Exp 3: Measure another VRM for which you know the answer	298
39.4 Exp 4: Measure some unknown VRMs.....	298
39.5 Grading rubric:	299
Chapter 40 Lab 23 SBB circuits: smart LEDs.....	300
40.1 The Smart LED Component in your board 4 project.....	300
40.2 The smart LED strip used in this lab	301
40.3 The smart digital LEDs.....	302
40.4 Reading datasheets and design tradeoffs	303
40.5 Installing the Library	306
40.6 The experiment.....	308
40.7 In your report	308
40.8 Grading rubric:	308
Chapter 41 Lab 24 SBB circuit with the Buzzer	309
41.1 The buzzer	309
41.2 Reverse engineering and experimenting.....	310
41.3 In your report	312
41.4 Grading rubric:	312
Chapter 42 Lab 25 SBB circuits: ferrites	313
42.1 The Ferrite Filter test board.....	313
42.2 Experiment 1: build up your slammer circuit and measure the rail noise.....	314
42.3 Exp 2: expectations.....	317
42.4 Exp 3: measure the noise through the filters	318
42.5 So what?.....	320
42.6 Grading rubric	321
Chapter 43 Lab 26 ESD measurement and mitigation	322
43.1 Purpose of this lab	322

43.2 Exp 1: Build an Arduino E-field meter	322
43.3 Write a digital filter to average over n power line cycles (PLC).....	326
43.4 My sketch: display measurements averaged over n power line cycles.....	326
43.5 What you will do for the lab.....	330
43.6 Grading rubric	330
43.7 How the ADC measures electric field	330
43.8 A simple static charge experiment with a cat	333
43.9 Static charge experiments	335
43.10 Reducing static charge build up.....	336
43.11 Electrostatic damage (ESD).....	336
43.12 An advanced note	337
Chapter 44 Brd 4: assemble, bring up, bootload, test	339
Chapter 45 Lab extra: SBB circuits: debounce circuit.....	340
45.1 Debounce circuits and switches	340
45.2 Ferrites and Filtering.....	343
45.3 In your report, you should include.....	345
Chapter 46 Lab xx Cross talk in ribbon cables and return connections	346
46.1 Limitations of the Solderless breadboard	346
46.2 In your report, you should include.....	349
Chapter xx Lab xx: The heartbeat sensor	350
58.1 The 16-bit ADC.....	352

Chapter 1 About the ECEN 3730/5730 class and labs

The textbook for this course is [Bogatin's Practical Guide to Prototype Design of Solderless Breadboards and Circuit Boards](#), published by Artech House. Each week, there will be reading assignments. The principles we will follow for this course are detailed in this textbook.

The specific board design projects and weekly, in class labs, are detailed in this lab manual for the Practical PCB course. It contains all the details about all the labs and the way this course will be run.

Not all the labs in this manual will be completed in the class. The specific labs selected for each lab period will be called out in the announcements page on Canvas for each week.

All the labs will be conducted in the ECEE 281 Circuits Lab.

There is a lot of content provided as part of this class: the textbook, the lab manual, YouTube videos, and through Canvas. There is also a lot of information online which you can find by googling. However, much of the information posted online is either wrong or misleading. Be very careful relying on information you find from other sources than what comes from me.

As part of this class, you will be developing **engineering judgment**. This is applying your engineering skills to make your own decisions, especially when you do not have all the information you need. Do not believe everything you read, hear or see online or from others. Apply your engineering judgement to challenge authority and only accept what you judge to be accurate or reasonable.

If you encounter a contradictory statement you see online or from another source, bring it to class for discussion.

Less googling, more thinking.

1.1 Name badges

The very first homework assignment is for you to create a circuit board with your name on it. This will be your name badge you can wear for the rest of the semester. Please wear it during class so we can all learn your name.

Have some fun and personalize your name badge with a unique solder mask color. Your TA will help you discover your choices from JLCPCB. Want to be even more unique? Feel free to change the name badge shape

1.2 Daily Quizzes

This class starts promptly on time. This is only 75 minutes for all aspects of the lab. There will usually be announcements, a brief intro to the lab and then time for the lab or in class design review activity.

Every minute is important. If you come late to class, you will not have as much time to finish the lab and you will miss the lab intro. This wastes the time of the TAs and your neighbors in having to get the directions repeated. It is important to come to class on time.

As an incentive and a reward, there will be a written quiz handed out each lab period precisely at the start and collected no later than 5 minutes after the official start time. It will be a short question or two. This question will relate to the reading assignments or the previous labs or the lecture portion of the class.

The quizzes count for 20% of your grade. There will be about 25 quizzes. This means each quiz is about 0.8% of your grade. If you miss 1 quiz, your grade will drop by about 0.7 points. There is no make-up of any quiz. Do not even ask to make up the quiz.

There are two opportunities to get extra credit for quizzes. This will either substitute for quizzes you miss or give you extra credit for your quiz scores.

Some quizzes may have 2 or more questions. When they do, one answer will count for the quiz and the others will be extra credit. These quizzes with multiple questions will not be announced ahead of time and will be offered arbitrarily throughout the semester. There will be around 2 extra credit questions offered.

The second way of getting 3 extra credit points for quizzes is to create a portfolio web site page and pass the link to Prof Swettlen and your TA. You can do this at any time up to the last week of class. When you submit your portfolio page link, you will receive extra credit quiz points. This could substitute up to missing 3 quizzes, for example, or just count as extra credit.

In principle, you could receive as much as 5 extra credit points from quizzes.

1.3 Assignments and your portfolio page

Every one of your lab reports and final reports will look great on your portfolio page. These labs are very unique in the industry. Very few practicing engineers have any idea these sorts of measurements or design principles exist.

What you demonstrate in your labs is highly valuable to any hiring manager. I hear over and over again from former students how the principles they learned and demonstrated in this class **and were able to describe to a hiring manager** were what landing them a job.

As you write up each lab report, write it for a hiring manager to show off how much you understand about board design principles and best measurement practices.

When it comes time for a personal interview, try to steer the conversation around to some of the labs you did in this class. And then you can point them to the portfolio of your lab reports. If you can teach a hiring manager an important design principle and demonstrate your ability to start and finish a project, you will be hired.

Include other lab reports or assignments you did in other classes in your portfolio.

When a job says you need 1-2 years of experience, the experience you gained in this class counts at least for 1 year of professional experience. If you took a capstone class, this is another 1 year. Do not hesitate applying for a job requiring 2 years professional experience. If any hiring manager questions your experience, point to your portfolio page and tell them Prof. Swettlen told you that you have the equivalent of 2 years of professional experience.

1.1 Four types of assignments

You will turn in an assignment every week for this course. There will be four types of assignments you will turn:

- *A lab check off by your TA, done in class*
- *A lab report turned in the Monday following the lab*
- *A design file turned in before Friday (usually 9 am), AFTER the DFM check*
- *A board final report turned in the Monday after the board was assembled and tested.*

The syllabus has all the dates and assignments. The announcements page on Canvas will re-iterate the lab and what should be turned in each week onto canvas. The assignments page for each week will detail what should be posted and provide a rubric to submit your assignment.

1.1.1 Check off by your TA

Generally, each lab will have a check off by your TA. The check off is like a mini oral exam, which will take no more than 2 minutes. Its purpose is to give you a chance to practice speaking technish with the TA, show off what you have learned and done in the lab, and for the TA to evaluate if you have a clue to understanding what you are doing.

This is a brief oral exam where specific questions are asked, and you will have to answer or demonstrate what you did in the lab. Each TA check off is worth 1 point.

This is a chance to demonstrate your competency of the lab or assignment. If you are not sure of an answer, it is better to say you are not sure, rather than make up some answer. Your TA knows the answers and if you try to fake your way through the oral exam, you will not get the credit. You will waste your TA's time.

If you are unable to complete the TA check off during lab time, make arrangements with your TA to complete the check off. You may also be checked off by another TA or by Prof Swettlen. If I do the check off, know that I am a walking BS detector.

Each lab has 1 point for a TA check off. For some labs, there is only a TA check off.

Each design board file assignment (there are 4 of them) requires a TA check off at the end of the in-class CDR and has 1 point awarded after check off.

1.1.2 Lab reports are due on the Monday after the week in which the lab was assigned

Not every lab listed in this lab manual will be completed for this course. And not every lab completed will be written up as a lab report. But, at the very least, every lab assigned will have 1 point awarded for a TA check off.

When a lab report is due, it should be submitted on canvas the following Monday after the week in which the lab was assigned.

Each lab should take about an hour to complete. You should be able to complete them during the assigned lab time.

The lab report should be 1-2 pages. Before completing the lab, you will get a sign off from your TA who will ask you a few questions about what you did.

A lab report is worth 3 points. Generally, the rubric is

- *1 point for sign off by your TA.*
- *1 point if you showed good measurement examples in your report with an explanation and analysis of the measurement.*
- *1 point if you have a coherent summary of the purpose of the lab and what you were able to demonstrate. This will answer the so what? about the lab.*

Lab reports are due the following Monday after completing the lab, posted to canvas.

- *1 point is deducted if it is turned in late.*

If you want to see examples of lab reports, check the portfolio pages of former students, which are linked on Professor Bogatin's page: <https://www.colorado.edu/faculty/bogatin/current-and-past-students>

Commented [TS1]: Update this link

1.1.3 Design Files are due on Canvas as announced

There are four board design projects, plus your name badge board. In each board design project, there will be three steps.

Step 1 is completing and turning in the design files

Step 2 after you receive your board, step 2 is completing all the assembly, bring up, validation testing and characterization testing

Step 3 is writing up the final report.

In **step 1** you will create the plan of record, select the non-commodity parts and build your bill of materials (BOM). You will create the schematic and the layout. There is generally nothing to be turned in for the POR. But this info should be included in your final report.

The rest of step 1 is completing the schematic and layout. Note that you will be assembling your boards manually in class. This means you should ONLY use 1206 parts and NOT 0402 parts. You will have a lab to learn how to assemble these parts. You will see how difficult it is to manually assemble 0402 parts.

At the end of step 1 there will be an in-class critical design review (CDR). You should come to class on this day with your layout complete so you can share your design with the class when we do peer and class design reviews. You should participate in the design reviews and look for soft and hard errors for the other students' designs. Before you come to the class, you should have your layout complete. If you do not have it even started, you will not be checked off by the end of class time.

At the end of the CDR lab session, your final or near final layout design will be checked off by your TA. You are still responsible for finding and fixing any errors. There is no way a TA can spend 2 minutes looking at your design and find all the problems. All they can perform is a superficial check and verify your layout looks

complete. Take advantage of the CDR time to seek help on your design. You are ultimately responsible for your design. Class time will be allocated for you to work with your neighbor to do a peer design review. Many times, you can stare at a problem and not see it. But a fresh pair of eyes sees it instantly. In a peer design review, your partner will help find problems in your design and you will help find errors in their design. Practice reviewing another design will help you look for your own errors.

Before you submit your design files, you should post them on the JLCpcb web site and have them do a DFM review. Only after you get a pass from JLCpcb should you submit your design files to canvas. Your design files will count as one of the lab assignments.

You can have them do a DFM review by going through the process of placing the order for the board, but before placing the order, select the option to complete a DFM. DO NOT PLACE THE ORDER. All boards will be purchased by the class.

The day after the layout CDR your design files are due.

The design files will be collected and then ordered from the vendor. If you miss the deadline, you will not get your board ordered. We will only order boards once, in a batch, to save on costs. If you miss this deadline, your board will not be ordered until the next week. Your board will be late and you may not finish the design assignment. You always have the option to order the board yourself, at your own cost and responsibility.

For each board assignment, you will submit four files: the gerber with NC drill file, the BOM, the pick and place, and a screen capture of the JLCpcb DFM successful reply. The order will be for 5 bare boards. You should get these back 2 weeks after the order is placed.

You will hand assemble 1 or 2 of your bare boards to practice manual assembly of SMT components.

The grading rubric is:

- 1 point for sign off by your TA in class.*
- 1 point if your design files were turned in on time*
- 1 point if your design was accepted by the vendor for production*

If your design file is rejected by JLC, you may only end up with the 1 point for TA check off for the design assignment. This will only happen if you fail to go through the online DFM check on the JLC web site.

If your board fails the DFM check when the TA places the order for your board, it will slow up the delivery of ALL the boards and may affect the schedule for the rest of the students. DO NOT LET THIS HAPPEN FOR YOUR BOARD.

If your board is not ordered with the other student boards, you are still responsible for getting your design completed.

Step 2 in the board design process is assembly, bring up and test of your board. The details of what tests you ran, how you debugged your board, what you learned in the process, will be included in your final report.

In your POR, you define what it means to "work". When asked how you will test your board, never ever say, "I will turn it on and see if it works," unless in your next sentence you say, "By works, I mean..." What it means to

work is defined by the specification you write in the POR. The specification of what it means to work is defined in terms of the measurements you will perform to verify it meets this specification.

For example: the board will receive 5 V power, the oscillator output will be a 0 to 5 V signal, with a 500 Hz frequency and a 60% duty cycle. The red LEDs will turn on and have a 20 mA peak current.

In this case, the test plan is to verify the voltage on the rail with a scope, measure the oscillator output and measure the voltage across a resistor to get the current.

However, in addition to the functional specification you must meet to work, there is also the noise characterization. In this step, you will make sure your board works, do whatever debug or troubleshooting is required to get your board to work, and then conduct any characterization measurements.

Even though your board “works” there may still be noise. If you do not measure the noise, you will have no idea how much there is. Starting with board 2, measuring the noise in your boards will be one of the most important measurements you perform.

You should record scope screen shots using best measurement practices to show off your measurement skills. These measurements should tell the story about your board’s performance. The details should be included in your final report.

1.1.4 Final Board Report

Step 3 in every board assignment is writing up your final report.

Generally, you should write your final report with the idea in mind that a hiring manager will read your report to help them decide if they should hire you.

Final reports should be no more than 5 pages.

The three mid-terms count for 10% of your final grade each, and the final counts for 20% of your final grade. There are a total of 10 points possible in the midterms and 20 points in the final. This means each point is 1% of your final grade.

Boards 1, 2 and 3 are midterms. The report for board 4 is your final.

Generally, there are three parts of your report for a total of 10 points:

Section 1, 2 points. The POR details (before you design the board)

Section 2, 3 points. The details of the board design features. This is where you are showing the progression of the board design including the sketch, the schematic, the layout, the bare board and the assembled board maybe with lights on. This will look great in your portfolio, showing each process steps. What does it mean for your board to “work?” What is your specification? Did your board work? What did you measure, what is your analysis of the measurements? What was the noise? How did you measure it?

Section 3, 5 points. The “So what”. What did you learn from this project? How do the measurements you did illustrate good or bad design practices? If your board did not work, why not? Did you have to do any debugging? How did you find the errors? How did you fix them? What did you do wrong in this design? What did you do right? What will you do in future board designs based on what you learned on this project? Did you

add any more "usual suspects" to your check list of problems to avoid or root causes to check when a design does not work?

Remember, the purpose of the report is to show off to a hiring manager what you know and can accomplish. Hiring managers love seeing examples of challenges you encountered and how you overcame them.

In your final, report for brd 4, there is an additional 10 points when you show off at least 2 applications of your board.

In your board reports, you will get 1 point off for each of the following:

- 1 point off if you do not have a ground plane
- 1 point off if you have long gaps in the ground plane
- 1 point off if your decoupling capacitors are not close to the IC they decouple
- 1 point off if you do not have any test points,
- 1 point off if you do not have any indicator LEDs
- 1 point off if you do not have any isolation switches
- 1 point off if you do not have your name on your board

1.2 Principles of the labs

Each of the lab exercises you will do in this course has five important purposes:

1. *They are an opportunity to explore in **hands on experiments**, the **engineering principles** covered in the textbook and that you will implement in the boards you design.*
2. *You will learn to take a **concept** for a circuit, **evaluate datasheets**, and make **practical design tradeoffs** balancing acceptable performance with the cost in dollars, risk and schedule to design and build functioning circuits. When they don't work, you will have to **debug** them as well.*
3. *You will gain experience **describing these labs to each other**, to your TA, in your reports and to potential employees during job interviews. This will give you a chance to **practice speaking technical**. We call this language of engineers and scientists, "technish". You should take every opportunity to practice speaking and writing technish so you can describe your engineering projects clearly and unambiguously.*
4. *You will exercise many important **best measurement**, **best design** and **best assembly practices**. These skills will be valuable in all of your projects and differentiate you above other candidates when you apply for a job. Very few engineers are as experienced as you will be in using a scope and other lab equipment. The more experienced you are in these skills the more valuable you will be to an employer.*
5. *On your **portfolio page**, you can use each of the labs and board design projects as examples to show off your engineering abilities. Each lab and each project can be used as a great example of the engineering principles you understand and incorporated in your project. Many of these skills are poorly understood or known by many practicing engineers. These examples will help you **shine in your job interviews**.*

1.3 Collaboration is Encouraged

For all the in-class assignments, I encourage you to work with others, ask questions of your neighbors, of your TA and anyone else. The more you communicate with others, the more practice you will get speaking technish. Do not hesitate to ask questions or help someone else in the class. If you have a question, call someone over.

However, you should still perform all the measurements and all aspects of the design-projects on your own. The labs and design projects are not team projects. It is important for each student to gain the "muscle memory" of pushing the buttons and turning the knobs and finding the pull-down menu. Every student will do all aspects of the labs and design assignments. All the reports should be written on your own.

1.4 Most important rule for the lab: Safety comes first

I guarantee there will be a quiz question for which the answer is safety comes first.

Most important lab rule is safety for yourself, and your fellow lab attendees, comes first. Some of the labs will involve soldering or potentially blowing stuff up. Always pay attention to potential safety hazards and use precautions, such as wearing safety glasses and alerting your neighbors of possible smoke.

NEVER DO ANY SOLDERING WITHOUT WEARING SAFETY GLASSES. If there is a chance you might blow a component up, wear safety glasses.

No food is allowed in the lab. Only sealed liquid containers are allowed. This is so that if they are accidentally knocked over, they do not spill on the expensive lab instruments or your circuits.

If your fingers are sticky, please do not handle any instruments, your kits, or any other cables in the lab. Wash your hands first.

1.5 Your Lab kit

You will purchase two kits from the e-store. One is a core kit used for many of the labs. The second is a specialized kit just for this class containing parts used in the labs..

You will be given a few specialized boards throughout the semester. These will be returned after the lab is completed. Please return these specialized boards so that we can keep the cost of the lab as reasonable as possible.

Accidents happen, and it is ok if something breaks. If something breaks, or is missing, let your TA know and the parts will be replaced.

1.6 Making Up Missed Labs

You should be able to complete all the lab assignments during class time. If you need to come in at another time, or want to stay late, you are welcome to use the lab assuming no other class is in the lab.

If you miss a lab, check in with your TA or Prof Swettlen to receive permission to make up the lab. Life happens, but you should have a good excuse when you ask permission to make up a missed lab. Then you can arrange with your TA to make up the lab but be considerate. This is an inconvenience to your TA and is not a regularly scheduled time.

Chapter 2 Board design projects

There are four board design projects in this course, not counting your name badge. In order to complete them all in the short time we have in this semester, and provide at least a 2-week production time, the schedule is very tight and very rigid. There is little room for slack.

This means everyone must pay attention to the design schedule. This will always be the case in almost any board design project you work on.

An important note. When you google information, you often have no idea of the quality of the source. There are many designers online who design boards but do not use best practices. Be very careful when you look at a reference design online. It may teach BAD habits. This is the value of this class. You will learn BEST design practices.

For example, it is very common to see designs that use a copper ground flood on signal layers. This can result in MORE noise than just leaving it off. This is a BAD design practice yet is very common.

2.1 Bare boards and assembled boards

For each of the four board assignments, you will assemble the boards yourself and I will supply you with the parts. I only stock 1206 size parts. If you design with any other size parts, you will not be able to assemble them onto your board and your board will not be complete.

In the past we used to have JLC assemble 2 boards for you using your BOM. These costs were prohibitive to maintain this practice, but we still want to mimic the assembly process by generating the full set of design files. This means that when you submit a board assignment, you will be submitting three design files: the gerber files with the NC drill file, the BOM and the pick and place file.

The gerber files and NC drill file are ALWAYS needed. These define the layer features and the via features. The pick and place file is normally required for JLC to perform assembly. The Pick and place file will become optional later in the semester as we discuss PCB assembly in class

The BOM file is required 1) to mimic the typical process AND 2) to provide one last chance to ensure you only used 1206 sized parts. Each part should be on a separate row. DO NOT combine different reference designator parts on the same row. Always review your BOM before you submit it.

You will get your bare boards back 2 weeks after they are ordered.

2.2 Board Design Milestones

There are seven steps in designing a circuit board, as outlined in the textbook:

1. *Develop the plan of record*

2. Identify the non-commodity parts and build the BOM
3. Create the schematic
4. Turn the schematic into the layout
5. Get the board and parts back and perform any required assembly
6. Bring up, troubleshoot and test your board
7. Complete the documentation

In our class, we map these seven steps into seven specific milestones:

1. POR complete, including block diagram and rough schematic
2. schematic is complete
3. layout is complete
4. final CDR is complete
5. fabs ordered
6. bring up and test is complete
7. final report turned in

Each board design project you complete in this class will have specific dates assigned to these milestones. These dates are hard and fixed. They cannot be moved. Pay attention to the schedule for each board.

2.3 Circuit board build schedule

The schedule is critical. If you miss the deadline to submit your board, it will not be ordered on time. You may receive a 0 for that board project if your board is not ordered. Completing your design on schedule is important.

Check the current course syllabus for the final schedule of the board design projects.

2.4 A Special offer for all PCB Designers

I want to encourage all students to design circuit boards. I will order and pay for any standard boards that you wish to purchase. As long as your board uses standard features, is less than 3.9 inch x 3.9 inches and is 2 layer, and the cost is only \$2 for 5 boards, I will pay. This saves you \$30 in shipping (plus the board cost).

Be sure to have your name on your boards. Send the gerber files to your TA by email.

If you need a larger board or some special feature, ask before you send the order to your TA.

Chapter 3 Build Your Portfolio

My hidden agenda for all the lab reports and documentation is to give you content you can add to your portfolio. Every weekly lab report and mid-term report should be written with the idea of posting it on your portfolio page. This should be in a format that would make sense to an employer.

You should build a portfolio site.

Your next step after you graduate with your degree is to get a job. Many of you will get a job in industry, some of you will go to academia, some of you may start your own companies.

Regardless of your next step, you should have a showcase of your achievements in college to show off what you are capable of to your future employers.

The best way of doing this is by creating a simple on-line portfolio. There are many ways of doing this. The absolutely simplest way is using google sites. They offer a Student Portfolio template to get started.

Here is the step by step to create a google site: <https://support.google.com/sites/answer/6372878?hl=en>

This link will get you to your google sites pages: <https://sites.google.com/new?tgif=d>

By the end of week 1, you should create a google site which will be your portfolio. What you put up there is up to you. However, I strongly recommend you consider using selective labs you do in our class as examples to add to your portfolio.

For example, in wk 1, you are building a 555 timer and characterizing it. Your lab report is a 1-2 page description. Why not consider writing it up to put it on your portfolio page? What new thing did you learn? What skill would you want to show off to a potential employer?

Maybe you could include:

- *Your ability to translate a datasheet into a working circuit*
- *Your observation of the difference in performance between two different 555 timers*
- *How you designed for a specific freq and duty cycle and your ability to measure it and achieve it*
- *Special skills at best scope measurement practices.*

The details are up to you. The better organized your portfolio is, the better the message you project to anyone viewing it.

As you complete other projects in other classes, you should consider adding the description with pictures, on your portfolio page.

Your portfolio is your chance to show off what you have accomplished in college. Your ability to finish a project, demonstrating your skills in engineering principles, best design practices and best measurement practices are what will get you your first job. One of the most important skills a hiring manager is looking for is your ability to overcome challenges to complete a project. Be sure to emphasize some of the challenges you encountered and how you did not let them stop you from completing the project.

While listing the classes you took are important, showing how you applied what you learned to complete specific projects is even more important. This is what you can show off in your portfolio page.

Every lab report and board project report you complete for this class would look great on your portfolio page. When you write up your report, keep this in mind. When you get feedback from your report, edit your final report and then post it to your portfolio page.

Pictures and even links to YouTube videos are a real plus to describe your projects.

You should think about all of the projects you completed in your other classes as opportunities to show off your skills to a hiring manager and post them on your portfolio page.

Historically, many students having graduated from this Practical PCB class used the projects they completed to help them land either a summer internship or a full-time job. Talking about their projects and what they learned at their interviews contributed to them getting hired.

Chapter 4 Lab 0: Your name badge and your business card

This project will give you practice navigating Altium, becoming familiar with editing the overly layer (often called silk screen layer) and exporting the proper files for manufacturing. Since you will be editing an existing badge file, the project should take less than an hour – most finish the edits in 30 minutes. Be sure to wear your PCB name badge in class so everyone can learn your name.

Please refer to Canvas for two videos where we walk you through the complete process. We also will post the current name badge template on Canvas in the same folder.

Files → files/folder/Name Badge Information

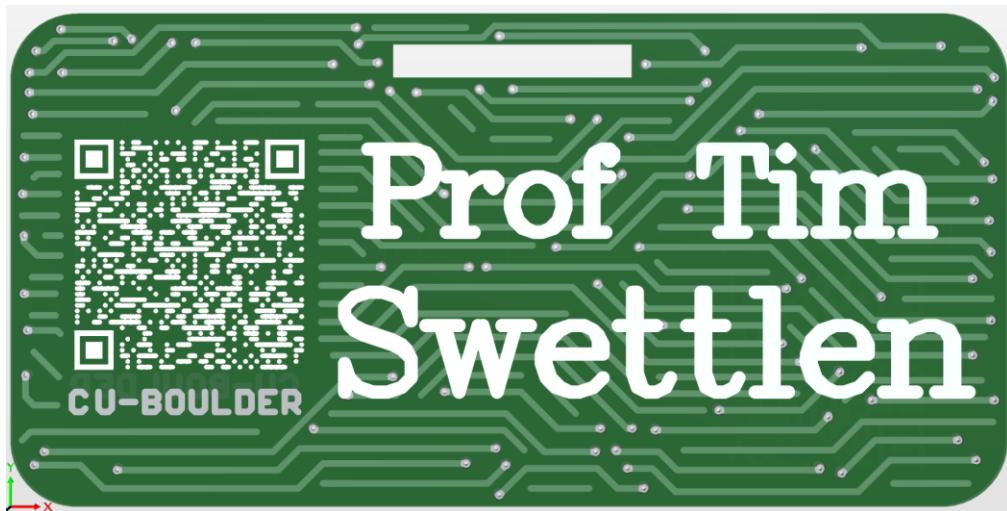
4.1 Your name badge

This first board will become your name badge which you will wear in class.

As your first board design project, you will just edit a simple board with an outline of roughly 3x1 inch and print your name in big letters using silk screen. You will be given a steel plate with adhesive on one side, and a small magnetic plate.

You attach the adhesive to the back of your PCB and then use the magnetic plate inside your shirt to stick the PCB name badge to your shirt.

When you get your board back, you will have an instant name badge to use in our class or wear anywhere else you want. My name badge includes a slot so extra boards can also be used as luggage tags:



4.2 Adding a QR code to your circuit board

Anyone can create a QR code which is encoded for a specific URL. Point the camera of your smart phone to the QR code and the camera will read the embedded url and your browser will point to this url which you can access with one click.

There are many web sites to create a custom QR code with an embedded url. In Prof. Bogatin's video, he used qr.io. The site has changed their policy and the QR codes are fully free. DO NOT USE THIS WEB SITE.

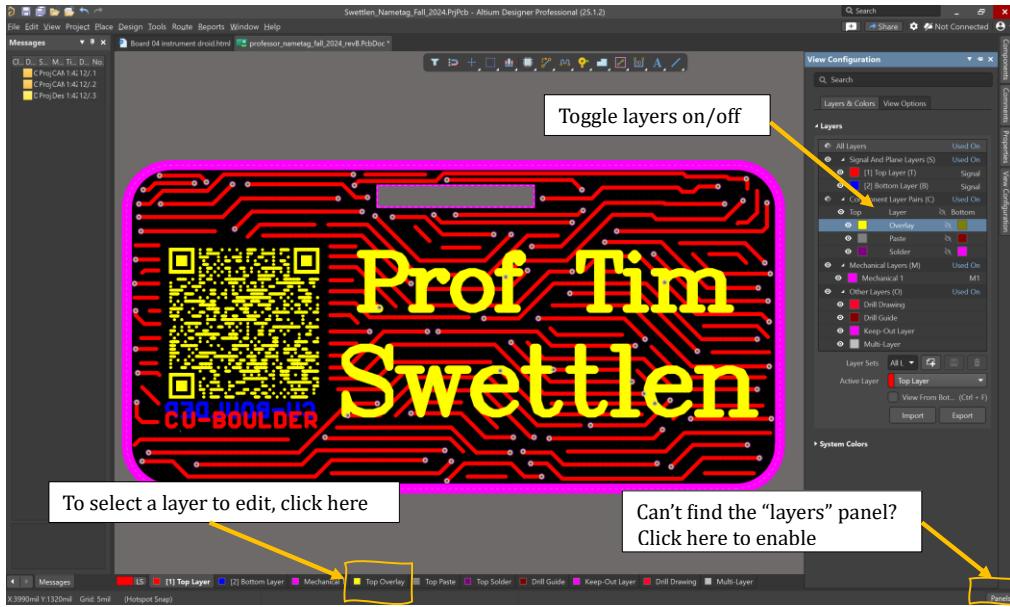
A better website to use is <https://genqr.com/>. Once on the GenQR Code site, create a **static** QR code. Static codes can never be changed, they contain the data directly encoded. Therefore it's impossible to put in ads afterwards.

For example, here are the QR code that redirects you to Professor Bogatin and Professor Swettlen's site:



QR codes can be added to any board to direct a user to any web site. To add the QR code image to your circuit board, you have many options. The cleanest way is to save the image you want to add to your board to a .jpg or .png file. I personally grab the image with a screen capture tool and save it as a *.png

Then, you can use the place/graphic command to highlight a region of your board where you want the image to go. This is shown in the figure below:



Be sure you have selected as the focus, the overlay layer so that the image is pasted in the silk screen layer. You click and drag the mouse to create a box in which the image will be pasted. This opens up a dialog box from which you select the jpg file. Then click paste. I used a resolution of 300 dpi. The image is pasted into the box. The result is shown in the figure below. You can click on any region of the image and drag it to position it as you want.

Be sure that you have adjusted the focus of the layout tool to the silk screen or overlay layer.

Toggle into the 3D view to see what your board will like. Toggle is as easy as hitting the number "3" for 3D. Type "2" to return to the layer view.

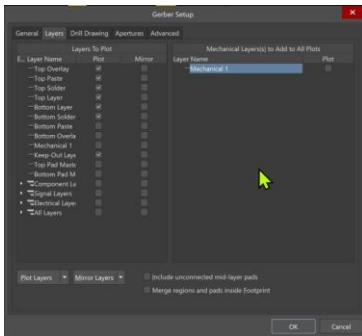


4.3 Directions to make our name badge circuit board

1. This video will walk you through the full process
<https://canvas.colorado.edu/courses/121241/files/folder/Name%20badge%20information>
2. Request an Altium student license
 - a. <https://www.altium.com/education/students>
 - b. [SBW 1-5 open Altium](#)
3. Install Altium- follow the directions on the web page for this course
4. Unzip the nametag.zip file, from the Canvas page .
5. Open up the PCB layout file
6. Change the name to your name in the top overlay layer
7. Remove the QR code currently there and add your own as outlined above
8. Watch skill building video SBW 1-24 [SBW 1-24: Export Gerbers and Drill Files](#)



12. Save gerber files. In this step, be sure to include the following files:



13. Save NC drill file. It will appear in the gerber output file.
14. Zip up
15. Create an account on JLCpcb.com
16. Pretend to order your board
17. Before you place order, select check the file
18. If you get a confirmation email back, then upload your gerber_name.zip file to canvas

4.4 Your name tag board assignment and rubric

This board is due BEFORE you come to the second class meeting session. All board design files are due to be posted on a Thursday morning by 9 am on canvas. If you miss this date, you will have to wait for the next Thursday to submit your board design file.

This is a board design project. This board design project is worth 3 points.

1 point of it is accepted by the JLC DFM before you submit to canvas

1 point if it is ordered and accepted by the Thurs 9 am deadline, of the first week of class

1 point if you get your board back and wear it to class.

If you miss the Thurs 9 am deadline, be sure to submit your design files by the next Thurs and receive partial credit.

Chapter 5 BOARD 1: practice board

Very important: before you submit your design files, review the checklist in this chapter and in the textbook. Get in the habit of reviewing a check list. You cannot remember all the features to pay attention to in your head. Use a check list and add to it each time you learn of another feature to add, or to not include.

There are many more ways of screwing up a board, than to getting it right.

This is your first functional circuit board. You will design it using best design practices, as outlined in the [Skill Building Video series, SBW-1](#).

I will order 5 bare boards and 2 assembled boards for you. However, you will do the assembly of this board yourself. You will have the assembled boards from JLC as practice to learn the process of ordering assembled boards.

You can compare the boards you assembled and the ones JLC assembled. If there are issues with the assembled boards, make a note of this in your final report, if you get the assembled board in time, or take this new information into account in your next board design project.

Remember, since you are assembling this board by hand, use 1206 parts. These are the ONLY size parts I have in stock and which you will get in your kit of parts. If you use 0402 footprints on your board, you will not have any parts to assemble, and your board will not work.

5.1 Purpose of this board

The purpose of this board project is to give you a chance to experience the entire board design flow and process with a simple, functional design.

In addition, you will use this board to practice manual SMT assembly. This means use 1206 size parts so that you make it easy for your assembly. For the LEDs, also use 1206 size parts.

When you are done, you will have a functional circuit board, in the size of a business card which you could use as a business card when you go for a job interview. This business card should have a QR code link to your portfolio page or github page.

You will supply power through an external power jack. There will be a 555 timer used in an astable vibrator circuit that will drive four different LEDs with different series resistors. These four LEDs should be the same color. This will show you how bright an LED will be with different series resistors. You will add isolation switches, inductor LEDs and test points.

There are five habits you should pay attention to in this board design project and every other future one:

1. *Read the datasheets for each component you are going to use. Will the part selected meet the performance you want? If there is a concern, maybe you should build an evaluation board either as a solderless breadboard or a qualification circuit board before you waste a 3-week delay in getting the board back before you discover you used the wrong part, or used it incorrectly.*
2. *Select specific parts based on a tradeoff of engineering and practical considerations. Choose parts that are available, meet at least the minimum performance required, have an acceptable footprint and that have a low risk of not working.*
3. *Once you start the design, it becomes yours. Take responsibility for your design. Just because a datasheet or a reference design or a design you found on the internet says do it a specific way, does not mean it is the right way or even if it will work. Use your engineering judgement. Never follow a direction blindly. Understand each and every design decision and make it yours.*
4. *If you are not sure about a design decision, research it. Ask another student. Ask the TA, ask another expert. Do a literature search. Do a google search. Evaluate the information you find online critically. Just because it is online does not mean it is correct or is a best design practice. Maybe you will have to build a prototype and evaluate the part yourself to reduce the risk. Sometimes you will have to make a decision*

without all the information you need. This is where you must use your engineering judgement. This grows with experience.

5. Many design decisions will be based on a balance between achieving the performance you want while keeping the costs acceptable. The costs are in terms of dollars, schedule and risk. These are all important cost factors.

Here is an example of a similar board to what you will be building, undergoing test and bring up:



5.1.1 POR for brd 1

Your circuit board will contain:

1. A power plug to use an external 5 V AC to DC charger to power your board
2. A 555 timer chip and circuitry designed for about 500 Hz and 60% duty cycle.
3. Using parts in the JLC integrated library we provide for you. If you wish to assemble the board, you should select parts you can assemble, like 1206 parts. If you have equivalent options, use basic parts from JLC which are lower cost. Why pay extra if you do not need to?
4. Add 4 LEDs of all the same color and series resistors of: 10k, 1k, 300, and 50 Ohms.
5. Use indicator lights, test points and isolation switches as appropriate.
6. Design to measure the 5 V input rail, the 555 output voltage and the current through the 50 Ohm LED.

7. Note: will the 555 IC you select support driving all the current through all the LEDs? How much current is this?

5.1.2 What you will need

Altium Designer and the latest integrated libraries provided for the class.

When your bare board comes back from the vendor, you will be given a kit with all the parts and you will do the manual assembly of the parts.

5.1.3 A 555 timer chip

The 555 is a powerful component from which we can build a variety of oscillators and frequency related functions. This is the part commonly used.

There are a number of versions of the 555 timer. We have two versions in our integrated library:

LCSC Part# C90760

LCSC Part# C7593

Which one will you use?

This is where design tradeoffs come in. Your criteria for which one to use might be based on:

1. Cost: is it an extended or basic part? Check the JCLpcb web site: <https://jlcpcb.com/parts> and search on the LCSC part number.
2. If the unit cost is < \$0.5, the part cost will be in the noise.
3. Is it in stock? Check the same web page as above.
4. Can it operate at 5 V rail?
5. Can it handle the current it needs to source or sink to drive the LEDs?
6. Can it switch fast enough for the application?
7. Are there any other features we need for the part for this application?
8. Does one part have more or less risk than the other part?

You decide which part you want to use, based on your analysis. There are no wrong decisions, if you can justify your decision. Sometimes it is a personal preference. That is an ok answer as well.

For me, I am a speed freak. I like faster parts and I am willing to pay a little extra for faster parts. I also like to push parts to their limits and see if I can break them, so don't always do what I say, unless you are wearing safety glasses.

However, if you use the faster part in this application, can it drive all the current needed for the LEDs? Is it more important to see a faster edge than to drive all the LEDs? You can decide this for your application.

This analysis and your part selection should be stated in your POR, which will be in your final report.

If you choose, you can also plan to assemble one board with the fast 555 and one board with the slow 555. I will give you parts for the two boards. You can compare the output signals and the current through the LEDs.

5.1.4 Indicator LED circuits

We will use an LED primarily as an indicator of a line being high. This can be for a data line or a power line. There are five different color LEDs in the LCSC integrated library. They are all in a 1206 part size. Each color will have a slightly different forward voltage drop.

The only parts we stock in inventory are 1206 parts.

For the other indicators, just be sure you can drive them with a 3.3 V DC signal and a current limiting resistor. What is the criteria for selecting the current limiting resistor for general indicator lights?

As an indicator, how much current do you want to use in the LED? One purpose of this practice board is to see that even 3 mA of current is plenty to see an LED on. The precise current is not important. Even as low as 1 mA of forward current is enough to make the LED easily visible.

You can decide in your project which colors will correspond to what message or indication.

Of course, it does not matter which order the LED or resistor are in the circuit. However, if we want to have the option of easily measuring the current through the LED one approach could be to measure the voltage across the resistor.

If the resistor is on the high side of the LED and the LED's cathode is connected to ground, then we would need a differential probe to measure the voltage across the resistor to get the current.

While this is possible, it is sometimes easier to just use a 10x probe which is ALWAYS referenced to ground. In this case, the low side of the resistor should be connected to ground. To make this possible, the preferred order is the LED then the resistor connected to ground.

Of course, you can add test points to anywhere else in the circuit you want.

5.1.5 Test points

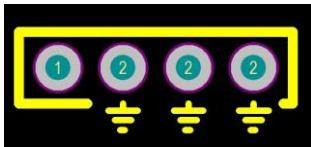
All the test points will be used for the 10x probe. This part is found in the integrated library.

The signal pin of the tip of the 10x probe is inserted into pin 1 of the test point. The small ground spring of the 10x probe will fit in the far-right hole. The other two ground holes are there because I have some other high bandwidth probes that use the adjacent ground hole. This makes this test point a "universal" test point.

There is no LCSC part number for the test point, since this is just some holes.

There are multiple test points in the library. For the 10x probe, make sure you are using the test point with 4 holes, one connected to the node for which you wish to measure the voltage and the other 3 connected to ground.

This is test point TP_10x_probe. The footprint is shown here:



5.1.6 Isolation Switches

To isolate one region of the circuit from another, use a 2-pin header pin as a switch. These can be found in the integrated library. You will just need to add the shorting flag to your board as needed.

You can use isolation switches anywhere in your circuit to isolate some parts of your circuit from others, except between the IC power pin and its decoupling capacitor. Why is this not a good idea? If you want to isolate the power from a specific IC, where do you place the isolation switch?

Feel free to add any isolation switches anywhere else in your circuit you may choose to help you debug your board when it comes back.

5.1.7 There are three parts and assignments for this board project (and all the other board projects)

Part 1: submitting your design files (3 points)

Part 2: bringing up your board (write up included in your final report)

Part 3: Mid-term which is your final report (brds 1, 2, 3, are 10% of your grade each, brd 4, is 20% of your grade.)

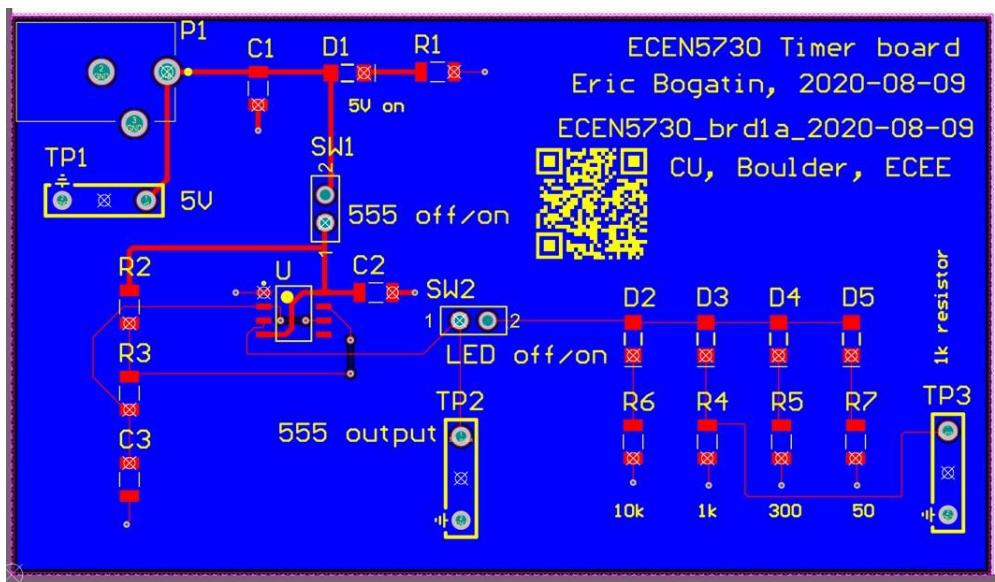
5.2 Adding a QR code to your circuit board

See the directions in creating your name badge for adding a QR code to your board. This is completely optional, but a good idea.

5.3 Create a Business Card

You can expand this idea to create a business card from a printed circuit board and even add a QR code to your board. This way, you can literally give out a circuit board as your business card with, or without components, with your contact information and a QR code which points to your portfolio page. What could be a better tool to advertise your expertise in circuit board design, not only a circuit you designed using the best design practices taught in this class, but also a direct link back to your portfolio page.

Here is an example of a PCB Business card.



5.4 Part 1: submit your design files

Before you come to the lab, you should

- *review the videos in SBW-1. This walks you through this board design.*
- *make a start at sketching the block diagram, the schematic, the schematic capture and the layout.*
- *create your portfolio page or have a personal web site you will reference on your board*

You will complete the schematic and have it reviewed in the CDR

You will complete the layout and have it reviewed in the CDR

You will have your layout design checked off by your TA before the end of class.

You will create the three design files for your board

You should go through the process of getting your complete design file accepted by JLCpcb before you submit your design files on canvas. This will assure you that you have an acceptable design.

5.4.1 Prep before you start this lab

All the details are in the skill building workshop SBW-1 on my [YouTube channel](#). Review each of these videos BEFORE YOU COME TO THE LAB. They will walk you through, from beginning to end, how to design this board.

5.4.2 Schedule for this board

This is an accelerated design cycle for this board. It is a simple board.

The specific schedule is posted in the syllabus. Here are the key milestones:

- *View the skill building workshops for getting started with Altium.*
- *Complete the POR for this board on your own.*
- *Complete the schematic for this board on your own*
- *Come to the lab with the schematic complete.*
- *In-class schematic CDR*
- *Complete the layout on your own*
- *In-class layout CDR and check off by TA. Your design files will not be accepted until you check off your design with your TA.*
- *Go through the ordering process on JLCpcb web site to verify your design files. DO NOT ORDER YOUR BOARDS.*
- *Turn in the three design files for your board fab by 9 am on the due day. The boards will be ordered starting at 9 am on this day.*

The in-class design review. You will learn as much when you review someone else's design as when your design is reviewed by others.

When you are reviewing a design, be sure to look for good features that you might want to include in your design, and features that are done well.

Look for hard errors that will cause the board to not work.

Look for soft errors that might not reduce the noise to as low a level as possible.

Look for cosmetic errors that might not be good habits and might result in more difficulty (read, higher risk) when assembling, testing or using the board.

Listen to the rationalization the designer used to make their personal preference decisions. Maybe you would want to follow those guidelines as well.

The design review is a discussion. While another pair of eyes looking at a design is always a good thing, sometimes both the designer and the reviewer will learn new insights in the discussion, when forced to articulate their thinking behind a decision.

1. Selected students will present their schematic for group review.
2. Students will pair off and do a peer design review with each other's design.
3. Students will revise their designs.
4. Selected students will present their layout for group review.
5. Students will pair off and do a peer design review with each other's layout.
6. Students will revise their layout.
7. Before the end of the lab, each student will have their schematic and layout checked off by their TA. Their TA will NOT correct their designs, but give a go, no go. If it is a no go, students will have to figure out why and make corrections.
8. Before you submit your design files, submit them on JLCpcb web site and verify they are acceptable.

5.4.3 Common mistakes to avoid

Very important: before you submit your design files, review the checklist in this chapter and in the textbook. Get in the habit of reviewing a check list. You cannot remember all the features to pay attention to in your head. Use a check list and add to it each time you learn of another feature to add, or to not include.

There are many more ways of screwing up a board, than to getting it right.

- 1 . Be sure to review the checklist in the textbook.
- 2 . Do not forget to add your name to the board
- 3 . Do not change the reference designators. You can make them smaller but do not change them
- 4 . Do not change the Design ID name of any part. This is tied to the database
- 5 . If you do not want a specific part to be placed by JLC, then in the specific part's parameter values in the schematic, you can delete the LCSC C part number. The LCSC part number will not appear in the BOM or pick and place files

- 6 . Add labels to the test points and LED indicators so that anyone looking at the board can tell what they mean. Don't use TP3. This means more work for you to look up what TP3 means.
- 7 . Do not exceed a board size of 3.9 in x 3.9 inches unless you have a strong compelling reason.
- 8 . Your board price should not be higher than \$4, otherwise, you are using features that are too small. Change them.
- 9 . Do not order your board. Just go through all the steps on the JLCpcb web site up to this point to have them do a DFM check of your design files.
- 10 . In your BOM, make sure each reference designator part is on a separate line. Do not group reference designators together in the BOM.
- 11 . Even though the bottom layer will be a ground plane, make sure you selected the bottom layer in the **stack up** as a "signal" layer, NOT a plane layer. You will add a polygon pour to the bottom layer as the ground plane. This way the bottom layer will be a positive layer like the signal layer.

5.4.4 Rubric for part 1 (3 points)

You will turn in your design files as the final assignment. Rubric:

- 1 point for sign-off of your layout (and schematic) by your TA
- 1 point if you turn in your design files on time
- 1 point if your design files are accepted by the fab vendor.

5.5 Part 2: Bring up and test

Review your board, take pictures, perform measurements and collect scope screen shots. Remember, in your report, you should include an analysis of your measurements. Just showing a measurement with no analysis is not enough. The details are included in your final report for the board.

5.5.1 What does it mean to work?

Normally, before you design your board, you have a specification of how your board should behave. This is part of the engineering requirements for your board. These define what it means for your board to "work." When your board is complete, you test your board to these engineering specs.

We jump-started the brd 1 design to get it done quickly. Having completed your board design, what would you list as the engineering requirements to which you would test your board?

Here are examples of some possible specifications:

1. Board is powered with a 5 VAC to DC regulator.

2. There should be an indicator LED turning on when the power is on
3. The 555 timer generates a square wave signal with a frequency and duty cycle of about 500 Hz and 70%, depending on what was specifically designed.
4. The output of the 555 timer should drive four LEDs of different brightnesses, controlled by resistors.
5. The current through one of the LEDs can be measured.
6. There are inductor LEDs and switches.

5.5.2 Measure key signals

You should measure at least the following signals on your board with a scope, NOT a DMM:

1. The output voltage of the 555 timer, verifying the frequency, duty cycle and rise time, with no LEDs connected.
2. The same measurement with all the LEDs connected. What do you estimate the total current draw to be?
3. What do you estimate the Thevenin output resistance of the 555 timer to be based on the open circuit voltage and the loaded voltage.
4. The current through one of the LEDs.
5. The switching noise voltage on the 5 V power rail, synchronous with the 555 switching signal
6. Estimate the current through each of the four different LEDs based on the output voltage, the forward drop across the LED and the value of the resistors.
7. Do you have any recommendations on the minimum current through an LED to make it visible as an indicator light? Remember, the duty cycle of the LEDs is only about 70%. A smaller current is better as it does not waste as much power, but you want a high enough current to be easily visible.

5.5.3 Grading rubric for part 2:

There is no report required or separate grade for the bring-up and test section. The report for your bring up and test is to be included in your final report. Remember, your final report should be written to post on your portfolio page. This means clearly written, a brief description of what you are showing in your measurement and clear measurement examples and analysis. Lastly, there should be a so what? Section. What did you learn from this exercise that will apply to other design projects?

NEVER, EVER show measured data without clearly identifying what it is and what it means. Provide an analysis of the measurement. Answer the so what? question about this measurement. What does it tell you?

There will always be 1 point taken off in any report if you just show a measurement without providing an analysis of what it is and what it means.

5.6 Part 3: Final report for brd 1 and grading rubric:

Your report counts as a mid-term, worth 10 points total.

In your report, which should be < 5 pages, you should include:

(2 points) The project overview:

- *Your POR and expectations for what it means to "work"*
- *Sketch of the schematic you started with*
- *The actual schematic capture used in Altium Designer*
- *The board layout you ended up with*
- *A picture of your board*
- *A picture of your assembled board (maybe with lights on)*

(3 points) What worked:

- *Your definition of what it means to work*
- *Your expectations for any performance features*
- *What you actually measured to verify your board "worked"*
- *Your analysis of the measurements*
- *Demonstration of best design practices and best measurement practices*
- *What did not work*

(5 points) Analysis of your project:

- *What worked and you did well and want to do in future designs*
- *What did not work, and you will want to do differently in future designs. There are no points deducted if your board has errors as long as you analyze why they occurred and what to do differently next time to avoid these errors. A mistake is valuable if you learn from it.*
- *Were there any hard errors- why did they go wrong*
- *Were there any soft errors that you would like to do differently next time?*

Points off:

1 point off if your board does not have your name

1 point off for each day your report is late

1 point off if you show data without an analysis of the so what?

Remember, your report would look great in your portfolio. This is an example of the design flow from concept to holding a working board in your hand. Show it off.

If you wish to see examples of other final reports or other lab reports, check this page showing a few reports from past students: https://drive.google.com/drive/folders/1HPslcq-u69Zmxo_IQq_pV9smFHrnU3wE?usp=sharing

Chapter 6 Lab 1: Mastering the scope

The oscilloscope, or “**scope**” is the most versatile and useful instrument you will use in any electronics lab. It will measure and display the voltage at its input as a function of time. We refer to the voltage being measured as a **signal**. Generally, we refer to the signal as the voltage component we care about that has the information we are looking for. All other aspects of the measured voltage are **noise**. We will always be measuring signals and noise.

Unfortunately, we cannot sense voltage with our five senses except in extreme cases, and then it is usually as pain. We must use instruments to open a window for us into the important world of voltages so we can open the lid on our circuits and peer at their inner workings. To know how your device under test (DUT) behaves, we have to measure voltages.

When it comes to designing circuit boards, we want to engineer interconnects to reduce the noise. When it comes to measuring, we want to reduce any noise that comes from our measurement process. We call these **measurement artifacts**. An artifact is noise that comes from the measurement process. They are usually avoidable using best measurement practices.

If the scope is set up incorrectly, or the connection from the device under test (DUT) to the scope is not optimized, the measurement may contain an artifact which is incorrectly interpreted as signal. This is why learning the best measurement practices is so important.

In this lab you will learn some of the best measurement practices for using the scope.

6.1 Signals and Noise: Why the scope will be the first instrument you will turn to

For historical reasons, we refer to signals which are constant as DC voltages and changing signals as AC, for alternating current. The term, AC, refers to all the signal components which are not constant. It does not have to be a 60 Hz sine wave. It can vary in any way. The AC component is anything other than the DC component of the signal.

Sometimes, measuring the noise coming from the DUT is as important as measuring the signal from the DUT. For example, in a power rail, the “signal” is the DC voltage carried by the rail. Anything else, the AC component, is noise. In the power rail, while the DC component is important, the noise on the power rail is often a source of problems and is as important to know about as the DC component.

It is tempting to use a digital multimeter (DMM) to measure what you think is a DC signal, such as a power rail. But, a DMM will tell you nothing about an AC signal component. This means, it will hide the noise and may give you a misleading confidence in the power rail.

To get a more accurate view of even a power rail, the first instrument you should use to measure it, or any signal of interest, should be a scope. If you see that the signal is constant, you may be able to measure its value directly from the scope. If you need higher accuracy of the DC voltage, then it is appropriate to use a DMM.

Unless you know for sure your voltage is only DC, **ALWAYS** use a scope as the first instrument to measure any voltage signal.

6.2 Before you come to the lab

Review the [SBW-7 about the 4024 scope](#). This covers the scope and function generator. View all the videos through SBW-7-11 the sync output of the function generator.

The most important videos which you should watch with the scope in front of you so you can play along are:

SBW 7-1 (7 min)

SBW 7-2 (6 min)

SBW 7-3 (19 min)

SBW 7-6 (12 min)

6.3 Specific don'ts when using a scope

Most scopes have a panic button. This is called auto set up. NEVER use this button. It assumes the scope is smarter than you are. If you don't know how to set up a scope and are drawn to use the auto set up button, learn how to use the scope. Some features of a scope turn the scope into a "smart" scope. This is a misnomer. Never assume the scope is smarter than you are.

If you cannot set up the scope yourself or measure a value with your mark 1 eyeball, how do you expect the scope to do it for you?

Before you perform a measurement, you should have an idea of what to expect. This is rule #9. This will give you a feel for the scale settings you would need on the scope to see the signal. Try to apply rule #9 to all of your measurements. Sometimes you will not have a clue what to expect. The more you learn about your DUT, the better you will be able to use rule #9. This means the better you understand your DUT, the better you will be able to apply rule #9.

Take every opportunity to practice rule #9 by always thinking about the signals you expect to measure.

When you measure what you anticipate, you will have higher confidence that you really understand your DUT. Rule #9 is a confidence builder. When you don't see what you expect, there is always a reason for it. Make sure you have eliminated measurement artifacts. Maybe you have more to learn about your DUT.

1. *Always try to apply rule #9 and anticipate what you expect to measure. This will help you set up the scope and understand your DUT.*
2. *Never press the auto set up button.*

Many scopes will store any special setup conditions after they are turned off. If your scope is also used by others, you may not know how it is set up when they turn it off and you are turning it on. In order to get it into a known state so you don't have to go through every combination of settings, press the DEFAULT set up button. This places the scope in the factory reset condition, a known state every time.

3. *When you come to your scope the first time of the day, always press the default set up button.*

More often than not, the measurement or cursor functions are a distraction. ALWAYS turn them off unless you have already used your mark 1 eyeball to measure what you need from the front screen, so you have an estimate. In most labs this initial estimate is good enough to use without turning on cursors or measurements.

4. *Always start with cursors and measurements off so you can estimate any values you need from the front screen.*
5. *Do not use the cursor unless you have first measured the value with your mark 1 eyeball*

One of the common artifacts using a cursor is that the numerical read out is from a different channel than the one you think you are on.

Just because a measurement function says rise time, does not mean it is measuring the rise time feature you think it does. If your signal edge has some structure to it, where is the rise time measurement function measuring? The measurement function will always give you a number. It is just not clear what this number means. If you can't measure the figure of merit with your eye from the screen, how do you expect the scope to do it?

6. *Do not use a measurement function unless you know exactly how it is calculating what you think it does.*

In order to make it easy to measure a signal off the front screen, adjust the scales so you can read the signal directly. This means make the zero voltage and time in the center of the screen. Press in the vertical or horizontal adjust buttons to auto center the zero positions.

Use the coarse scale adjust setting for the time and voltage and change their scales so the signal is expanded to most of the screen and the time base shows the features you care about. The scales should be whole numbers like 1 V/div or 0.2 V/div or 1 msec/div or 50 usec/div. This will make it easy to read trace features directly off the front screen. You have the entire screen height available for you. If you want to measure some features of your signal, adjust the vertical scale so it uses most of the screen, rather than keeping the trace within a fraction of a division.

7. *Center the voltage and time scales to the center of the screen*
8. *Adjust the vertical and horizontal scales to expand the displayed signal on scales and display scales with nice, whole number.*
9. *Adjust the scale values to use most of the scope screen to display your signal of interest.*

When displaying a repetitive signal, you should get it to be stationary on the screen. This is done with the trigger controls. If you are viewing what you think is a repetitive signal and it is not stationary, but smeared out, DO NOT use the run/stop button. This leaves too much information off the screen. You want to see how constant and stationary your signal is, not a snapshot in time.

If you have a repetitive signal, you can make it stationary with the trigger. If you have a transient signal, you can also use the trigger set up to capture and hold it on the screen.

10. *Do not use the stop feature for repetitive signals. Adjust the trigger instead.*
11. *Always start with the trigger on auto mode. Only use normal mode in special cases.*

12. Select the correct source channel, the slope and then the DC threshold level
13. Know the difference between the auto and the normal mode settings

Generally, the best connection between the DUT and the scope is a 10x probe. This will give you the highest bandwidth of the measurement, the highest impedance to the DUT and the easiest way to connect to the DUT.

14. Always compensate the 10x probe before using it and understand what is going on inside the 10x probe
15. Check to make sure the scope is displaying the signal with a 10:1 attenuation in the probe setting so the displayed voltage is the TIP voltage.
16. Never use a 10x probe except between the DUT and the scope. Never use a 10x probe to connect a signal source to any other instrument.
17. Never use the 10x probe on the 1x probe setting. This will dramatically decrease the measurement bandwidth.
18. Note that the 10x probe ALWAYS and ONLY measures single ended signals. NEVER connect the ground lead of the 10x probe to any node in your DUT which might be at other than earth ground potential.

6.4 Get familiar with the scope using BNC and minigrabber using the compensation signal

Connect the BNC cable with mini grabber to the channel 1 of the scope and the minigrabber tips to the compensation signal. You should see a 1 kHz square wave with about 1-2 V peak to peak value. Adjust the scales to see this.

Practice using the vertical and horizontal controls to manipulate what you see on the screen. Practice using the trigger controls. Understand the difference between auto and normal trigger modes.

Make note that the Keysight 4024 scope and many others like it ONLY measure signal ended signals. This means every signal voltage is referenced to the local ground of your circuit, AND this ground is connected to earth ground through the scope's probe return pin, which is connected to the chassis ground, which is connected to the earth ground.

One advantage of the Digilent AD2 scope is that it is able to measure a true differential signal, but its bandwidth is limited to only 35 MHz. Much of engineering is a trade off between cost and various performance metrics. All the signals we care about in this class have a bandwidth higher than 35 MHz, so we will not be able to use the AD2 scope for our course. We have to use the Keysight 4024 scope.

6.5 The 10x probe

Be sure to review the [SBW 7-6](#) on the 10x probe.

Check the compensation of your two 10x probes and adjust them if needed.

Change the color bands on your 10x probes so they correspond to the channel colors on your scope. The yellow band probe should connect to channel 1 and the green band probe should connect to channel 2, for example.

Note that the 10x probe ALWAYS and ONLY measures single ended signals. NEVER connect the ground lead of the 10x probe to any node in your DUT which might be at other than earth ground potential.

6.6 Master the scope

Follow the videos about using the 4024 scope with a scope and the cables in front of you. Learn to push each button from the video and use every opportunity to practice the best measurement practices so they become second nature to you and you are not hunting around and randomly trying every knob.

Remember, there are only 3 basic sections of the scope to master:

The vertical controls

The horizontal controls

The trigger controls

Never use the cursors or measurement functions unless you absolutely have to, AFTER you have measured the signal from the front screen with your mark I eyeballs.

6.7 Check off with your TA

When you have successfully captured the scope compensation waveform, call your TA over to share your results and answer one question on the waveform figure of merit. An example of a figure of merit would be:

Frequency: The rate at which the waveform repeats, measured in Hertz (Hz).

Amplitude: The peak voltage or current level of the wave.

Duty Cycle: The ratio of the time the signal is at its high level (on-time) to the total period of the waveform, typically expressed as a percentage. For a true square wave, the duty cycle is 50%.

Rise Time: The time it takes for the signal to transition from its low level to its high level. Fast rise times indicate a sharper transition.

Fall Time: The time it takes for the signal to transition from its high level to its low level. Fast fall times also indicate a sharper transition.

Overshoot/Uncertain: The momentary exceeding of the target high or low levels during the transition.

Chapter 7 Lab 2: SBB-build 555 timer

The first board project you will build is brd 1, the practice board design, using a solderless bread board (SBB). Before you commit the time (can be as long as 3 weeks) or the money (typically about \$10 per board for our class), you want to have high confidence the components connected together, as described in the schematic, will work.

One way we increase confidence is by going through design reviews. The preliminary design review (PDR) happens early in the design cycle to review the plan of action. The critical design review (CDR) happens before we commit major resource. In addition, to reduce risk, if it is practical, you can build a solderless breadboard version of the circuit to test it out, or, if you have the tools available and are skilled in the art, you can build a virtual prototype using simulation.

Before you begin this lab, [view this video where I walk you through some of the details](#).

7.1 What you will do in this lab

In this lab, you will build a solderless bread board version of the circuit of brd 1.

The power is from the 5 V rail on the Arduino board

Build a 555 timer oscillator circuit with a fast, then a slow 555 timer

Drive a red led with a 1 k resistor and then a 50 ohm resistor

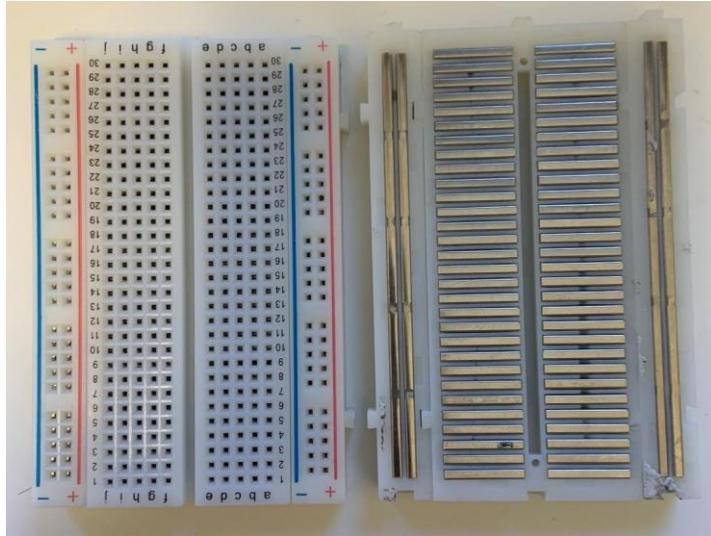
Estimate the currents in the LED in each case

Measure the signal features with the slow and fast 555 timer.

7.2 The solderless breadboard

You should read the chapter in the textbook on the solderless breadboard: chapter 11.

Here is the inside of the solderless bread board showing the actual connectivity between the rows and columns of holes.



You should be able to test this out using an ohmmeter and short wires you plug into the holes. How will you test this connectivity?

Often, you can build a solderless breadboard version of your project in hours and test out circuit alternatives before you finish the schematic capture and send the layout files to fab.

Before you start the design of brd 1, you will build most of the circuit in a solderless breadboard to get familiar with the circuit design and performance. Use this solderless breadboard circuit to gain confidence in your design.

You should have read chapter 11 in the textbook and be aware of some of the best practices using the SBB. For example,

use color coding for the wires.

use short, custom length wires for all the hook up.

DO NOT use long floppy jumper wires for hook up. This is what inexperienced students do, not professional engineers.

With a little practice, you will become proficient at cutting and stripping short lengths of wire to meet each custom requirement.

Follow-on labs will give you further opportunities to practice using the best practices in building SBB circuits and in performing scope measurements and the problems that arise not following these best practices.

7.3 Here are two timer options:

There are two 555 timers chips in 8-led DIP packages in your kits. If you do not have these two, ask for them. They are similar to the ones in the integrated library:

Fast: LMC555CMX or the TLC555IDR C6987 or the 7555

Slow: NE555DR C7593

7.4 Purpose of this lab:

Build a prototype version in a solderless breadboard of the brd 1 project you will design and build as a circuit board.

Try your hand at building a functional circuit in a solderless breadboard.

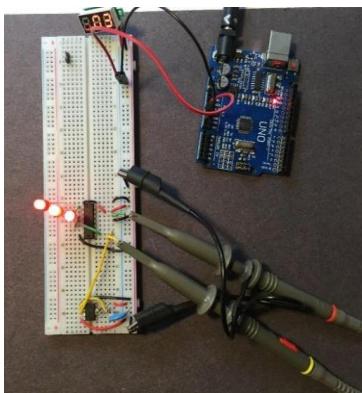
Get some hands-on experience with scope measurements.

Practice finding datasheets of components and reading them.

Find useful circuits online but make them your own. If you use a specific component in your schematic, know why! Do not follow a direction just because the datasheets says so.

Make some design decisions, balancing tradeoffs of parts and performance. Practice rule #9

An example of the simple solderless breadboard circuit you will build is shown below.



An example of the circuit you will build, debug and characterize. The top part of the circuit with the LEDs is with a hex inverter- not necessary for this lab.

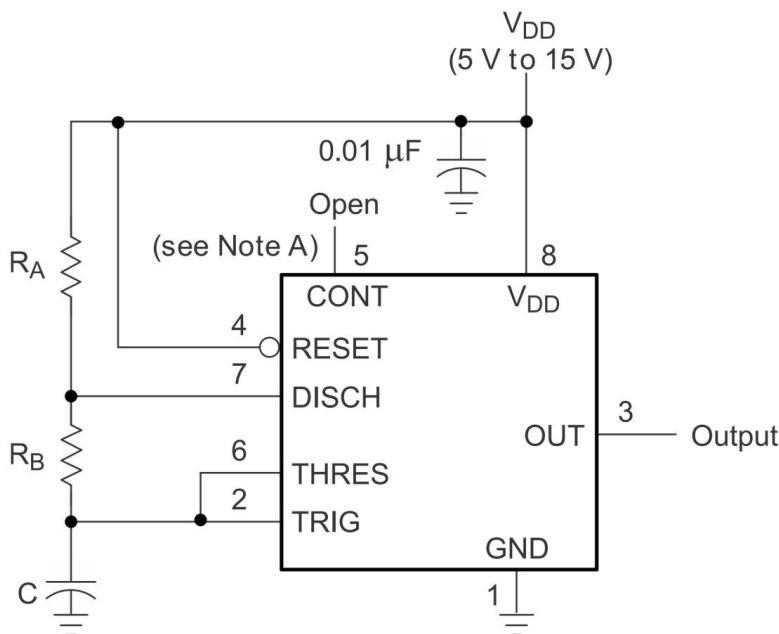
7.5 The circuit you will build

Goal: design a 555 astable vibrator circuit operating at about 500 Hz and about 50% duty cycle. Use the 555 timer chips and the resistors and capacitors in your kit. Pay attention to the circuit design in the SBB and how you will replicate it in your circuit board. These are not rigid requirements, but guidelines.

Much of engineering is about design tradeoffs. Always be aware of the flexibility you may have in a design. Is the requirement to within 1% of the specification or to within 50% of the specification? When in doubt, ask, or make up an answer and make sure you clearly state your assumptions.

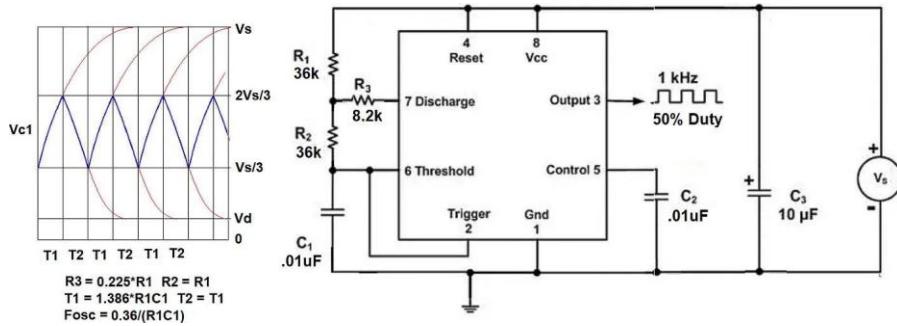
In this case, the exact operating frequency is not important, as long as it is between 300 Hz and 1 kHz. Likewise, the duty cycle is not critical as long as it is within 40% and 75%.

You can use the [datasheet for the 555 chip](#) to borrow the circuit for an astable vibrator. An example is shown in the figure below:



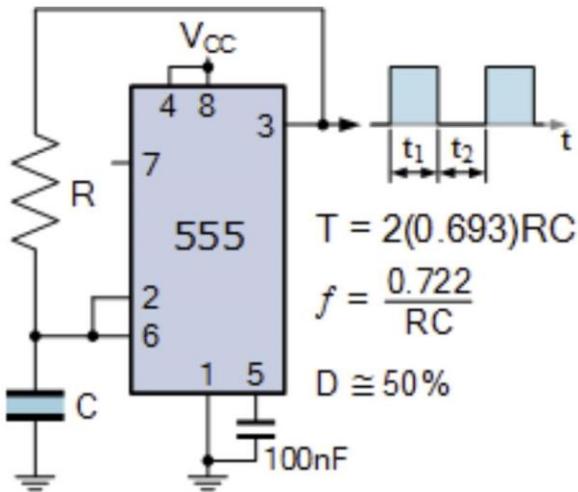
But, this is not the only design. Here are two others that offer a better approximation to a 50% duty cycle.

You can try this circuit from an [EDN article](#) that will give you a true 50% duty cycle square wave. This circuit is shown in the figure below.



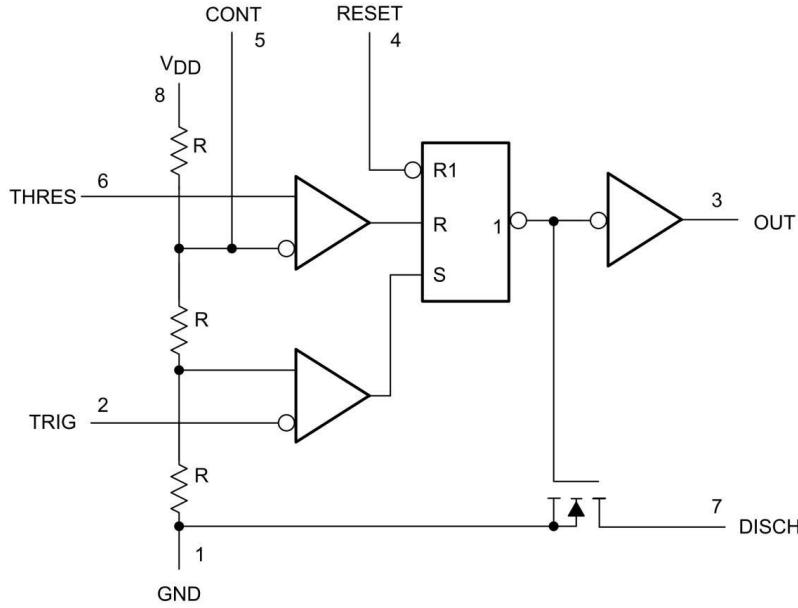
An alternative circuit to give a near 50% duty cycle with the 555 timer chip.

There is even a simpler circuit to generate a 50% duty cycle square wave mentioned [on this site](#), and shown below:



A simple circuit for a 50% duty cycle square wave generator.

With all of these choices, how will you decide which circuit to use for your design? Even though they are all on published web sites, do they all work? How will you know? One way of evaluating a reference design is to understand how each component works and convince yourself it is either a good design or a questionable one. For example, using the block diagram for the 555 chip, shown below, can you understand how the simple circuit above generates a 50% duty cycle square wave signal?



Block diagram of the 555 timer ship

Never blindly use a published circuit unless you are willing to take responsibility for it. If you are not confident the circuit will perform as you expect, you are keeping your fingers crossed and “hoping” it will work as you want.

Hope should never be part of the design process.

7.6 The rest of the circuit

You will use the 5 V rail from an Arduino board to power your circuit. Plug the Arduino into a 9 V AC to DC converter. Use jumper wires from the 5 V pins of the Arduino to your board. Did you also connect the return path?

You will build one circuit with the fast 555, and then replace it with the slow 555 and compare the rise and fall times of the outputs, the shape of the waveforms and the output voltage levels when driving an open and when driving a 50 ohm resistor and LED in series.

How much current is flowing in this circuit?

Never pass up an opportunity to measure something as an important consistency test

Once you have your circuit working, drive an LED with alternately, a 50 ohm series resistor, a 1k, and a 10k series resistor. Notice any differences in the output voltage? Why?

With the scope and 10x probe, measure the:

Rise and fall time of the 555 output for each device, with and without the 50 ohm resistor and LED.

The frequency and duty cycle- how does this compare with your predicted values.

Only use cursors or measurement functions on the scope AFTER you have estimated the figures of merit with your mark I eyeball.

7.7 Some hints

This first lab is an initial opportunity at practicing best design practices and best measurement practices to design, build, debug and measure a functioning circuit. When you measure it with a scope, use the 10x probe. Be sure to compensate the 10x probe before you use it. Also, add the color band to your scope probe so you can tell which probe is displayed on which trace on the screen.

You will break 4 common, bad habits:

1. *Do not use long floppy jumper wires. Use custom made, short color coded AWG 22 hook up wire.*
2. *Do not use long wires to connect the 10x probe to the nodes in your circuit to test. Use short wire leads.*
3. *Do not blindly follow a reference design. Understand it and make it your own.*
4. *Do not randomly turn knobs of the scope until you see something. Use best measurement practices when using the scope to measure the signals from your circuit. It is ok to play around, and try things as long as you use this chance to learn best practices.*

In following labs, the right way to engineer the interconnects and perform quality measurements will be the focus. It is more important to complete the circuit, get it working and end up with some measurements which seem reasonable.

You should at least practice Rule #9 in this lab. (Rule #9, of course, is never do a measurement or simulation without first anticipating what you expect to see.)

If you wanted to practice using best practices, then you would:

1. *Use the rolls of AWG22 wires and cut custom lengths of the right colors. Use the needle nose and diagonal cutters to strip a ¼ inch length of bare wire on each end. After practicing this, you will get good at it. It will take 1 second to create a custom, short wire of the right length and color.*
2. *Use red for power paths and black for ground*
3. *Use short leads from the 10x probe to the SBB test points.*
4. *Compensate your 10x probe before you use it for measurements.*

5. Adjust the scope controls with intent, rather than randomly turning knobs.

7.8 What you should pay attention to:

1. How the holes in the solderless breadboard are connected.
2. How to read a datasheet to understand the electrical properties of the devices
3. How to read a schematic and translate it into a physical layout
4. How to wire up the connections to build a circuit
5. How to probe a circuit with a scope
6. How to use a scope to see a clear voltage signal
7. How to use measurements on a scope to debug a circuit
8. How to use Rule #9 to help you debug a circuit and get quality measurements.
9. How to use an Arduino Uno as a power source for 9 V, 5 V and 3.3 V voltages.
10. How to build a circuit in stages, debugging each part of the circuit as you go.
11. How to trigger a scope on a switching signal on one channel
12. How to adjust the time base of the scope to zoom in the features of interest.
13. How to extract an important figure of merit from the front screen of the scope using your Mark I eyeball.
DO NOT USE cursors or the measurement function.

If you are unfamiliar with some of these principles, check the skill building workshops on my [YouTube Channel](#).

7.9 Check off with your TA

In order to get credit for this lab, you must go through a check out with your TA. If you do not get checked off during the lab, you will lose 1 point for this lab.

At some point when you are ready, call your TA over for them to review what you have done. You should be prepared to answer the following questions:

1. What is the connectivity of your solderless breadboard?
2. How are you routing power and signals on your solderless breadboard?
3. What did you predict for the frequency and duty cycle and what did you measure for each of the two 555 timers? Was there a difference?
4. How did you verify your 10x probe was compensated?

5. *What is the rise and fall time of the two different 555 timer chips and how did you measure them?*
6. *What is the difference in brightness of the LEDs with 1k and 50 ohm resistors for each of the two 555 timers?*
7. *Why is there a difference?*
8. *What was the output voltage of the different 555 timers with and without the LED and resistor?*
9. *Did you learn anything new from this lab?*
10. *Did your circuit work the first time or did you have to debug any of it?*

7.10 Your lab report

Remember, each lab report you do for our class will look great on your portfolio page. Write it up as though a perspective employer were going to read it. Use this as a chance to show off your engineering skills and expertise.

In your 1-2 page lab report summary, include a screen shot showing the scope trace of the output of your 555 timer showing at least 2 cycles on easily readable scales.

Include a screen shot zoomed in on the time base showing the rise or fall time on a scale allowing you to measure it from the front screen.

You select which 555 timer to use in your screen shots. Or, you can compare two of them.

Include a description of what these scope traces mean and your analysis of them. How well do they match what you expect based on the circuit you designed and the data sheets?

When you are done with your report, create a pdf and post it on the canvas page under assignments for wk 1.

7.11 Grading rubric: 3 points total

1 point if checked off by your TA

1 point if you have screen shots that are easy to read the information about the waveforms and you have an accurate analysis of the measurements.

1 point if you have a coherent summary of the project.

Remember, this report will look great on your portfolio page.

Chapter 8 Brd 1: Schematic CDR

You should come to class having completed the schematic for brd 1. There will be a brief review in class of the schematic. A few students will share their designs and the class will review the design.

All students will be looking for:

1. *Any possible hard errors which will prevent the correction operation of the board*
2. *Any soft errors which will increase the noise or the risk of the board not meeting spec*
3. *Any good design features which might be good features to include in everyone else's designs.*

Use the check list in the lab manual and in the textbook to go through the list of possible, or commonly occurring errors to verify none of these are present in the design.

After a few designs have been reviewed by the class, there will be time in class for all students to complete their schematics.

If anyone needs help, ask your neighbor, ask a TA or ask Prof Bogatin.

There is no TA sign off for the schematic. There is nothing due at the end of this class except completion of the schematic.

If you complete the schematic during class, move on to the layout.

Chapter 9 Brd 1: Final CDR

Before coming to class, you should have completed the layout.

In class, a few volunteers will be selected to present their layout to the class. While they present their layout, the rest of the class will critically listen and offer recommendations to the designer.

Be on the lookout for:

1. *Any possible hard errors which will prevent the correction operation of the board*
2. *Any soft errors which will increase the noise or the risk of the board not meeting spec*
3. *Any good design features which might be good features to include in everyone else's designs.*

Use the check list in the lab manual and in the textbook to go through the list of possible, or commonly occurring errors to verify none of these are present in the design.

After a few layouts have been reviewed, you will have time to complete your layout in class.

Before you leave, you should review your layout with your TA to have them check off your design.

It is not necessary your design be completed by the end of class time, but it should be far enough along so that any major issues can be identified.

Before you post your three design files on canvas, you must:

1. *Review your schematic and layout against the checklists provided in the lab manual and textbook*
2. *Have another student review your schematic and layout*
3. *Have your TA check off your layout*
4. *Post your design files to the JLC website and complete the DFM and get an email acknowledgement back from them that your design is accepted and ready for ordering. This may take as much as 2 hours to get an email back.*
5. *If you received any error messages from the DFM check, you have corrected them, and resubmitted your design for DFM check.*

Design files are due by 9 am on Thursday morning. This is when they will be ordered. If you miss this deadline, your board will not be ordered. You will have to wait for the next week to have your board ordered.

Turning in your design files counts as 3 points. The grading rubric is:

- 1 point for sign off by your TA in class.*
- 1 point if your design files were turned in on time*
- 1 point if your design was accepted by the vendor for production*

If your design file is rejected by JLC, you may only end up with the 1 point for TA check off for the design assignment. This will only happen if you fail to go through the online DFM check on the JLC web site.

If your board fails the DFM check when the TA places the order for your board, it will slow up the delivery of ALL the boards and may affect the schedule for the rest of the students. DO NOT LET THIS HAPPEN FOR YOUR BOARD.

If your board is not ordered with the other student boards, you are still responsible for getting your design completed. Submit it by the next week.

Chapter 10 BOARD 2: Switching noise with good and bad layout

The purpose of this board is for you to show off how layout decisions influence the amount of switching noise and the best measurement practices for measuring switching noise in a PCB.

10.1 Before starting this board

This board design brings together and demonstrates how to implement interconnect design features to reduce the two important switching noise sources: on the PDN and between signal-return paths.

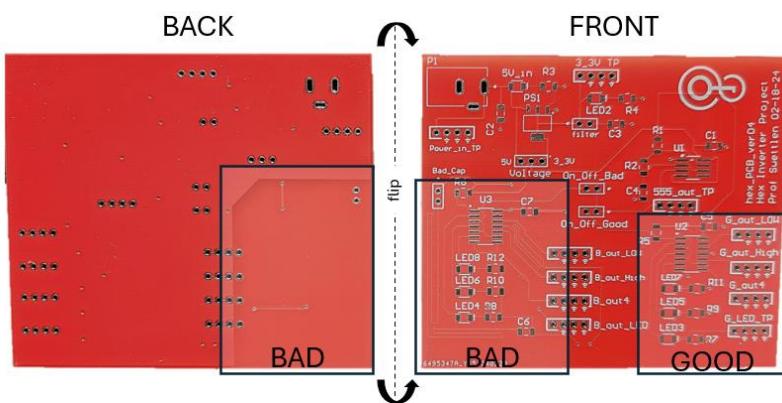
Before you complete this board design, you should have read chapters 2, 3, 4, 12, 13 and completed labs 4, 5, 7, 8, 9.

10.2 Board 2 Design Process

The key feature of this board is that you will build two identical hex inverter circuits. For one circuit, you will do the layout using best design practices: continuous return plane and decoupling capacitor close to the IC's power pin.

But in the second circuit, you will do everything wrong. There will be no ground plane under the IC and its routing. Instead, you will use just a trace to route the ground connections. And the decoupling capacitor will be far away from the IC pin it is decoupling.

Be sure to use the match placement of the parts and test points in both regions of the board. Just the routing will be different. Here is a rough idea of one example of this final board. Note, the back side of the board lacks a reference plane under the bad region of the board. For details of how to make an irregular plane shape, refer to SBW 1-26: Shaping GND on YouTube ([link](#)).



10.3 Brd 2 POR

Spending a few hours upfront, before you begin your project to think through the details will be a good investment to help you identify potential risks and how to mitigate or avoid them as early in the design cycle as possible.

In this board design project, you will get started practicing the best design principles in a simple 2-layer board. The goal is to design and build a board which will:

1. Convert 5 V in to 3.3 V; you learned this in lab #7.
2. Create a clock signal of about 500 Hz and about 50% duty cycle.
3. The hex inverters that we provide can operate at a 5 V or 3.3 V rail. Grad students must add a switch to control the hex inverter chips with either 5 or 3.3 V. This is optional for undergrads.
4. Add a switch to selectively connect the 555 output to either the good or bad layout hex inverters.
 - a. When not connected to the timer output, all the switching inputs should be tied HIGH so they do not float and switch unpredictably. How will you implement this? (Hint, you learned this in lab #8).
5. Each hex inverter has six inputs. Drive four of the inputs to demonstrate good layout and bad layout.
6. On three of four outputs of each hex inverter use LEDs and 50 ohm resistors as the load. Connect the output of the fourth inverter to a test point to act as a trigger for the scope.
7. Estimate the current you expect to draw from the inverter for the LED and the 50 ohm resistor load.
8. Plan to extract the Thevenin resistance of the output pin of one of the I/O. (Hint: measure the voltage on the output of an inverter with the LED and resistor compared to the inverter with no load using a 10x probe and the scope. From the voltage drop on the I/O and the current draw, you can calculate the Thevenin resistance of the I/O pin.)
9. Of the two remaining inputs, set up one hex inverter as a quiet HIGH and inverter as a quiet LOW.
10. Use good debug techniques and add inductor LEDs, test points and circuit isolation switches as appropriate.
11. Engineer the layout on one side of the board with best design practices and the other side of the board with bad layout practices. In the bad layout, move the decoupling capacitors far away from the Vcc pin.
12. Keep the part placement and routing identical for the two regions of the board, except for the location of the decoupling capacitor.
13. A summary of test points for your board (I count 11):
 - a. The scope trigger output
 - e. The 555 output signal
 - f. The 3.3. V rail on the board
 - g. The 5 V rail on the board
 - h. On both the good and bad
 - i. Voltage across one of the 50 ohm resistors

- ii The quiet high*
- iii The quiet low*
- iv The fourth inverter for a trigger and for the Thevenin Voltage measurement*

14. Use best scope measurement practices when you test your board by using the 10X test points.

One of the hex inverter circuits will not have a ground plane underneath it. It will just use ground wires to route the ground connections around.

You will use this board example to step through the entire board design process.

In your POR, which should be included in your final report, be sure to include:

1. *Rough schematic or block diagram*
2. *List of the significant parts and links to key datasheets*
3. *Definition of what it means to "work"*
4. *The schedule.*
 - a *While this class sets main milestones like CDR, assembly, you should learn to set a timeline and commit to it. The best engineers are able to outline a complex project months ahead of time and stick to their schedule.*
5. *The test, characterization plan.*
 - a *Don't assemble your board and just start collecting plots. Think ahead of time what you plan to measure and try to execute to this plan, it's hard but it keeps you focused. Also think about how to use the save function and stack meaningful data on the same plot. It's very powerful to have the "good" and "bad" results on one shared plot. Keeping consistent colors (bad always yellow, good always green) is also a very professional skill.*
6. *The power budget and how you will supply power*
7. *Potential risk sites and how you might avoid them.*
 - a *In industry Failure Mode and Effects Analysis (FMEA) is a tool used to rank risks and discuss ways to remove or reduce their chance of impacting the design. This is a big tool for our simpler boards, but you still should be able to think through a few key common issues. (Ex., using wrong sized parts like an 0402. To avoid this I will use the design checklist and have a fellow student review my BOM list.)*

10.4 New parts in this board

If you were designing this circuit independently, you would need to consider which 555 timer will you use for this application. Unlike in board #1 where the 555 timer drove three LED/resistor loads, this 555 circuit only drives the high impedance hex inverters. We will provide the faster 555 chip for this design when you

assemble it. To aid in assembly and bring up, you should add an isolation switch to disconnect the 555 from the hex inverter inputs. This allows the ability to debug these sections separately.

When the input to the hex inverters is not connected to the 555 output, if the inputs float, the hex will randomly trigger from ESD events. How will you connect the inputs to keep the outputs low when the 555 is not connected? (Hint: We saw this demonstrated in lab #8.)

10.4.1 The LDO

We usually supply power to a board from some external DC voltage. This can be a battery or an AC to DC converter. On the board, there will often be other Voltage Regulator Modules (VRM) to create other voltages or provide more stable voltage supplies.

An alternative DC voltage can be provided by a linear regulator. These devices typically need to have supplied a DC voltage higher than 2 V above the required output voltage.

When a low current and very stable voltage is required, and the input voltage is less than 2 V above the required output voltage, it is common to use a low drop out (LDO) voltage converter. The voltage accuracy of its output is typically 1% absolute accuracy.

For example, if the input voltage is 5 V and the output voltage required is 3.3 V, then a linear regulator can't be used because there is only a 1.7 V difference. Most linear regulators require > 2 V difference. Only an LDO would work.

An LDO is generally very easy to use. There is often an enable pin. This should be tied high to the input voltage to enable or turn on the output voltage. (In more complex circuits, this enable pin can be controlled to turn off large blocks of the circuit to save battery. We will simplify our design and tie the enable pin high.)

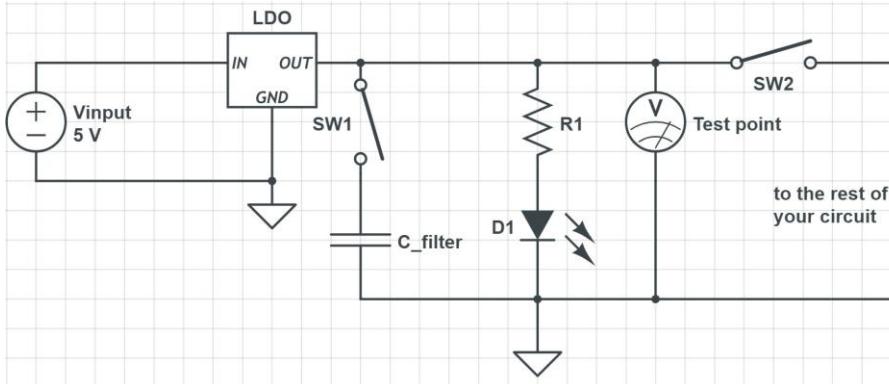
There is a feedback circuit inside the LDO which monitors the output voltage and adjusts the internal resistance of a series MOSFET to keep the output voltage within a specific range, compared to an internal reference voltage.

There is some response time to this feedback circuit. Sometimes, if there is a noise fluctuation that causes the output to vary, the feedback loop can go into oscillation. To prevent this, the output voltage fluctuations must be slowed down. This is done by adding a **filter** capacitor to the output of the LDO. Its value is not critical. **Any value in the 1 uF to 22 uF range will filter the high frequency noise to prevent oscillation.** DO NOT confuse this with a decoupling capacitor.

This filter capacitor should be placed in close proximity to the output of the LDO, to provide a stable voltage at the output, so that noise fluctuations do not cause the LDO to go into an unstable state. This is NOT a decoupling capacitor. It is a filter capacitor associated with the LDO.

Add an isolation switch in series with the output of the LDO and the 1-22 uF filter capacitor. This way you can look at the output voltage on the LDO to see any oscillation and then switch the filter capacitor onto the LDO to suppress the oscillation. Add a test point on the output of the LDO so you can see the oscillations.

The oscillations will be a large voltage fluctuation at about 1 MHz. The circuit you can use is like this one:



The LDO we have is the AMS1117: LCSC Part# C6186. You can do a google search and find the datasheet for this part. Note, the large tab in the middle (pin 4) is electrically connected to the 3.3 V output. It should be connected to the 3.3 V net in your schematic.

10.4.2 Hex inverter

The Hex inverter, 74AHC14, is a very simple chip. It will operate at either 5 V or 3.3 V. Grad students will add a 3 way switch between the 5 V and 3.3 V rails and the Vcc input to the Hex chip to allow switching between the two voltages. Remember, place the decoupling capacitor on the Hex side of the switch. This way you can compare the 3.3 V and the 5 V operation.

This chip has 6 inverters available, each with an input and an output. An inverter circuit component will take an input digital level and output the opposite. If a 3.3 V signal is at its input, its output will be 0 V. A 0 V signal at the input will result in a 3.3 V signal at its output.

NEVER use an inverter with an input pin that floats. The input impedance of an inverter is generally very high. This means stray electric fields as from static charges, AC pickup, or cross talk from a nearby trace, may change the voltage at the input and cause the output to switch all by itself.

You can do a cool magic trick with an inverter with inputs that float. Connect the outputs of a few inverters to LEDs and current limiting resistors. Rub your hair or your clothes a little or sit in a cloth chair to build up some static charge. When you waive your hands over the board, in the vicinity of the input pins of the hex inverter, the LEDs will flash on or off. Your static fields are inducing a HIGH or LOW signal at the gate inputs of the MOSFETs of the inverters.

Tie all connections that might be open to a pull up resistor to the power rail. This way the outputs will be 0 V. What value of pull up resistor should you use? I recommend a pull up resistor of about 10k. What influences this value?

10.4.3 LEDs driven by the HEX inverters

For the LEDs that the HEX inverters drive, we are using them to drive a lot of current and as indicators so we can see the inverters are being driven. For these applications, use a red LED so that we have the lowest forward voltage drop and can get about 30 mA of current switching. Very important to use the same LEDs for the good and bad layout circuit. After all, we want the only differences between these two circuits to be the layout. We will only provide red LEDs for your hand assembly.

10.4.4 Indicator LEDs

Depending on our stock, you can choose a variety of colors depending on what you want to indicate. A red LED will have the lower forward voltage drop and the highest current in our circuit. A blue LED will have the highest forward voltage drop and the lowest current in our circuit, for the same series resistor. Why is this?

What color would you like to use for your indicator LEDs? You choose. Based on your board 1, what series resistor do you want to use for your indicator LEDs?

10.5 Part 1: Submit design files

Before you come to the lab, you should make a start at sketching the POR, the block diagram, the schematic. You should have a good idea of what the purpose of this board is.

Complete your schematic: Only use 1206 sized capacitors, LEDs, and resistors.

Use the 10x probe test points. Add indicator LEDs. Add isolation switches. Grads add a switch to select 3.3 V or 5 V to power the hex inverters.

You should complete your schematic on your own.

Complete the layout: You will complete the layout before coming to class. We will do a CDR of the layout in class and your design will be signed off by the TA before the end of class.

You should go through the process of getting your complete design file accepted by JLCPCB before you submit your design files on canvas. This will assure you that you have an acceptable design.

Before the deadline, you will post your three design files on canvas.

10.5.1 Common mistakes to avoid:

- 1 Be sure to review the checklist in the textbook.
- 2 Do not forget to add your name to the board and to the zip file so that your TA can easily id your board.
- 3 Do not change the reference designators. You can make them smaller but do not change them.
- 4 Do not change the Design ID name of any part.

- 5 Add labels to the test points and LED indicators so that anyone looking at the board can tell what they mean.
- 6 Do not exceed a board size of 3.9 in x 3.9 inches unless you have a strong compelling reason.
- 7 Your board price should not be higher than \$2, otherwise, you are using features that are too small. Change them.
- 8 Do not order your board. Just go through all the steps on the JLCpcb web site up to this point to have them do a DFM check of your design files. If your files fail when the TA places the order, you will be penalized one point.
- 9 Make sure the good decoupling capacitors are close to your Vcc pin.
- 10 Minimize the number of cross unders
- 11 Keep all cross unders short. Said differently, do not route long signal traces in the ground plane.
- 12 Make sure you only 1206 sized capacitors, LEDs, and resistors

10.5.2 Rubric for part 1 (3 points)

You will turn in your design files as the final lab assignment.

Rubric:

- 1 point for sign-off of your layout (and schematic) by your TA
- 1 point if you turn in your design files on time
- 1 point if your design files are accepted by the fab vendor at the time of order

10.6 Part 2: bring up and test of brd 2

As you go through the bring up and test phase, be sure to think about how you want to document your final report. Take pictures along the way.

Remember, your final report on brd 2 will look great in your portfolio to show off your skills in designing a board for lower noise.

10.6.1 Inspect your bare board

Before you assemble your bare board, inspect it. Look for any obvious defects- short or opens. Maybe measure the resistance between the power and ground conductors. What do you expect it to be? What do you measure?

Take a picture of your bare board. The combination of your sketch of the schematic, the final schematic, the layout, the bare board and then your assembled board will make a great story board for the process of creating a circuit board.

10.6.2 Assemble your board

You will be given a kit of parts all using 1206 parts. If you used 0603 or 0402 parts in your design, the technical term for this is, "you're screwed." We only have 1206 parts in stock.

Verify you have all the parts you need. If you did not include the reference designator labels on your board, how will you know what parts go where? If you did not include descriptive labels for your test points, you will have made your life harder by having to continually referring to your design file on your computer to recall where each test point connects.

If you assemble your whole board and then turn it on to test it, it may be difficult to debug. You might consider assembling sections at a time and testing these sections. Maybe do the power first, then the 555 timer, then one of the hex circuits, then the other one. This way you can test and verify as you go.

10.6.3 What does it mean to work?

When you power on your board, what is your criteria for the board "working?" This should be defined in your POR as your performance spec. What are the functions you want your board to demonstrate?

Just saying, "I'm going to test if my board works" is a meaningless statement. What are the specific tests you are going to do and what is the criteria for passing or working?

For example, you should expect to measure about 5 V on the power rail and about 3.3 V on the LDO output. You should see a frequency, output voltage, and duty cycle from your 555 based on what you designed it for. You should see an inverted signal on one of the hex inverter outputs. If you measure the signals you expect to see, you can then say your board works, based on the specific criteria.

Your quiet low line should be close to gnd. Your quiet high line should be close to the Vcc.

10.6.4 Demonstrating debug features

One of the purposes of this board and all the boards in our class, is to exercise good design habits, such as design for debug. You should have indicator LEDs, test points and isolation switches. You should make note of these features and how well they work in your final report.

10.6.5 Switching noise and layout

The main purpose of your board is to show the impact of good and bad layout techniques in reducing switching noise.

The amount of noise you measure will depend on the rise time (and fall time) of the signals and the layout. What is the rise time you measure for the inverter you use to trigger the scope? Be sure to show this in your report.

You should have quiet hi and quiet low outputs from which to measure power rail noise and ground bounce.

When the scope is triggered on the rising or falling edge of the output trigger, what do you see as the difference in the quiet hi and quiet low noise for the different layout designs?

How does the noise you measure on the board level 5 V or 3.3 V rail compare with the quiet hi signal? Isn't this quiet hi signal also connected to the power rail? Why is there a difference between the board level power rail noise and the quiet hi voltage?

Be sure to measure and compare the voltage noise on the quiet HI pins for the good and bad layout examples.
Be sure to use best measurement practices.

What do you conclude about how layout decisions influence the noise?

Based on what you observe, what are your recommendations for doing layout to reduce switching noise.

10.7 Rail compression- (Grad students only)

You are measuring the quiet hi and the quiet low outputs. These are literally the voltages on the hex inverter die itself for the Vcc and Vss rails, relative to the local ground on your board where your probes are connected. The difference between these measured voltages is the rail compression on the die. This is the actual voltage between the 3.3 V and gnd on the die that the outputs see.

If you use three scope probes, one to measure the inverter output to trigger the scope, one to measure the quiet hi and one to measure the quiet low, you can use a math function on the scope to subtract the hi and low signals to see the voltage rail on the die.

When this voltage between the high and low sides changes, we call this the rail compression or rail collapse.

How does the rail compression on the die compare to the quiet low, quiet hi and on-board voltage rails?

The bring up and test report should be included in your final report. There is a great article on Signal Integrity Journal by Professor Bogatin that discusses this topic (on the same circuit) for you to reference. Savvy students will notice that measuring at various points actually highlights the inductance in the power rail just like we learned from lab #5. The link to the article is:

<https://www.signalintegrityjournal.com/articles/2790-measuring-only-board-level-power-rail-noise-may-be-misleading>

10.8 Rubric for your final report (10 points)

Your final report for brd 2 is a midterm. It is worth 10 points total. Remember, your report would look great in your portfolio. This is an example of the design flow from concept to holding a working board in your hand. Show it off.

Here is the scoring:

(2 points) The project overview:

- *Your POR and expectations for what it means to "work"*
- *Sketch of the schematic you started with*
- *The actual schematic capture used in Altium Designer*
- *The board layout you ended up with*
- *A picture of your board*
- *A picture of your assembled board (maybe with lights on)*

(3 points) What worked:

- *Your definition of what it means to work*
- *Your expectations for any performance features*
- *What you actually measured to verify your board "worked"*
- *Your analysis of the measurements*
- *Demonstration of best design practices and best measurement practices*
- *What did not work*

(5 points) Analysis of your project:

- *What worked and you did well and want to do in future designs*
- *What did not work and you will want to do differently in future designs.*
- *Were there any hard errors- why did they go wrong*
- *Were there any soft errors that you would like to do differently next time?*

3 points extra credit possible:

+1 point if you can demonstrate and measure the Thevenin output resistance of the Hex inverter I/O

+1 point if you configure your board and show the LDO oscillations without the output filter capacitor.

+1 point if you can demonstrate any difference in performance, like rise and fall time and amount of PDN and switching noise, with a 3.3 V or 5 V rail on the hex inverters.

Points off:

1 point off if your board does not have your name

1 point off for each day your report is late

1 point off if you do not show the rise time of a signal and the good and bad switching noise scope traces

1 point off if you do not have a coherent explanation for why the noise is different in the two parts of your circuit and how layout affects switching noise

1 point off if you do not include a section on what you did right and what you did wrong on this design with recommendations for how to do better on your next design.

Always remember, regardless of the grade, your final report will look great on your portfolio page and if done well will impress any hiring manager.

Chapter 11 Lab 3: Thevenin model of the waveform generator

In this lab, you will get a brief intro to using the waveform generator and understanding its output impedance.

11.1 Before you come to this lab

Be sure to view SBW 7-9 (12 min) and SBW 7-10 (26 min) in this series: [SBW-7 about the 4024 scope](#) and read sections 6.5 and 6.6 in the textbook on Thevenin models. Please note that for this lab we will use the internal waveform generator of the DSO scope in place of the external function generator.

11.2 Cable connections for this lab

How you connect the waveform generator and scope is very important.

For the initial measurement to study the waveform generator output, you should connect the output of the waveform generator to the scope with a BNC-to-BNC cable.

For some of the experiments in this lab, you will need two connections between the scope and the waveform generator: a sync signal and the waveform generator output into the scope. In these cases, use the BNC-to-BNC cable between the sync and scope and the BNC to mini grabber from the output of the waveform generator to the 10x probe and the 10x probe into the scope.

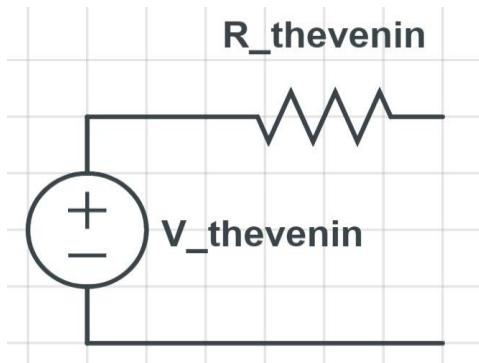
Remember, never use the 10x probe with its BNC connected to the waveform generator. You should understand why this is a bad idea based on how the 10x probe works.

11.3 Exp 1: Setting up the scope and waveform generator

1. *Use the BNC-to-BNC cable between the channel 1 of the scope and the waveform output of the scope.*
2. *What attenuation setting should you use for the scope?*
3. *Set up the internal waveform generator for a 1 V p-p square wave of 10 kHz. Make sure the waveform generator is on High Z output.*
4. *What do you measure on channel 1 of the scope? Adjust the scope vertical, horizontal scales and the trigger.*
5. *Change the waveforms and the frequency and the amplitude of the waveform generator. What do you expect to see on the scope? What do you actually measure?*

11.4 Exp 3: Measure the Thevenin model of the waveform generator

Think of every voltage source as a Thevenin equivalent model on the inside. This model is an ideal voltage source and an ideal output resistance as shown below. This is what is going on inside the waveform generator. You only see the output pins:



To characterize the waveform generator, you will measure its Thevenin voltage and its Thevenin resistance. In particular, you will compare these two figures of merit as you change some of the settings on the waveform generator.

Do not confuse what the setting on the waveform generator is with what you extract as the Thevenin voltage and resistance from your measurement. They may not be the same. When we use our own measurements to understand how an instrument or component works, we call this reverse engineering. We are going to reverse engineer the waveform generator.

What we will find is there is a difference between what the setting on the waveform generator reads and what we measure. But there is a reason for it. Could it be any more confusing?

11.4.1 How do you measure the Thevenin voltage of the voltage source?

Set the output of the waveform generator to high impedance. Use the mini grabber from the output of the waveform generator to the 10x probe tip. Set the waveform generator to a square wave, 1 V peak to peak and 10 kHz, or about these values. It doesn't matter about the exact values, only that you know what they are.

Measure the Thevenin voltage of the source. Change the output to 50 ohms load. Make sure the output voltage setting is set for 1 V peak to peak. What is the new measured Thevenin voltage? Change the output impedance to any other value. What is the Thevenin voltage you measure?

For any setting of the waveform generator, you will need to know what the actual Thevenin voltage of the source is. Figure out how the voltage amplitude you set on the waveform generator relates to its internal Thevenin voltage. This way, you will always know the Thevenin voltage of the waveform generator based on how you set it up. Hint: you need to know the output load setting and the amplitude setting.

11.4.2 How do you measure the Thevenin resistance of the source?

Most textbooks say to measure the Thevenin resistance by shorting the output of the circuit and measuring the short circuit current. In principle this will give you the Thevenin resistance.

However, generally, this is a bad idea. Voltage sources don't always like being shorted. They can sometimes blow up. And, this may put the voltage source into a state it is not designed for or outside the working, specified range of operation.

If you can't short the outputs, how do you measure the Thevenin resistance?

You will connect a resistor across the output of the waveform generator, so it loads the output. This limits the current from the voltage source. But, as current flows through the source, there will be a voltage drop across the Thevenin resistor. This means the measured output voltage across the load will not be the Thevenin voltage, which is unaffected by the load resistor.

The voltage difference you measure between the Thevenin voltage, and the loaded voltage, is a measure of the Thevenin resistance of the source.

What is the internal Thevenin resistance based on the load resistance, the Thevenin voltage of the loaded output voltage? You should be able to derive this relationship. This is described in the textbook and the skill building workshop video about the waveform generator which you should have viewed before you started this lab.

You will add an external resistor across the mini grabber leads and measure the voltage across it using the 10x probe. What value resistor should you add? Too high a resistor may not give you enough of a voltage change to measure with the scope. Too low a voltage level may generate too much power consumption and cause the resistor to blow up.

Or, too low a resistor may draw too much current from source and bring the operation of the source into its current limited mode where it does not behave like a Thevenin model anymore.

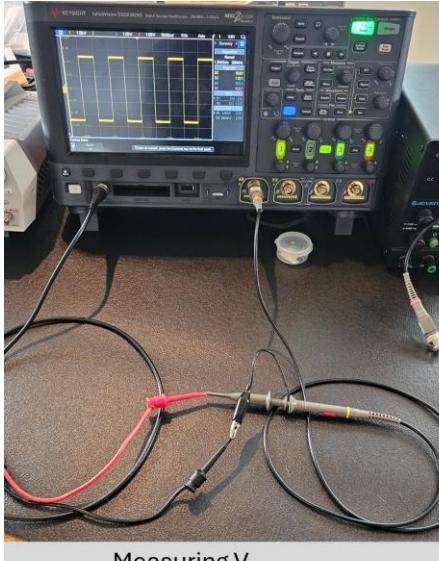
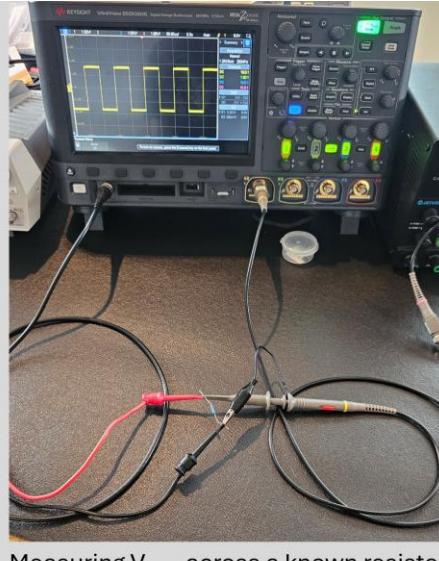
How do you pick the resistor value? First, determine how low is too low. Keep the power consumption below 0.25 watts to be safe. If the output DC voltage were 1 V, the resistance for a power consumption of 0.25 watts is 4 ohms. This means, keep the resistor you use larger than 4 ohms, IF THE OUTPUT VOLTAGE IS 1 V. If you use a higher voltage, raise the min resistor value.

Generally, start with a resistor value that is much higher than this minimum. A reasonable starting value may be 100 ohms. Select a 100 resistor from your kit. Measure the voltage drop on the output of the signal when you connect a 100 ohm resistive load across the output of the waveform generator. If the voltage drop is too small to measure, then make the load resistor value smaller.

Use the following equations to calculate the Thevenin resistance:

$$R_{Thevenin} = \frac{(V_{Thevenin} - V_{Load})}{I} ; \text{ where } I = \frac{V_{Load}}{R_{Load}}$$

$$R_{Thevenin} = R_{Load} \times \frac{(V_{Thevenin} - V_{Load})}{V_{Load}}$$

Measuring V_{Thevenin} Measuring V_{Load} across a known resistor

While the output load impedance of the waveform generator is set to high Z, measure the Thevenin voltage of the waveform generator and its Thevenin resistance. How does the Thevenin voltage compare to the voltage you set the waveform generator for?

Set the output load of the waveform generator on 50 ohm load. Repeat the same measurements. How does the Thevenin voltage you measure compare with the voltage value you set? Did the Thevenin resistance change?

Hint. Changing the output impedance will not change the display but go back and look at the “new” amplitude setting. The waveform generator amplitude will now be different by a factor of 2.

Can it get any more confusing?

11.5 Exp 4: Optional Experiment, An alternative way of measuring the Thevenin resistance

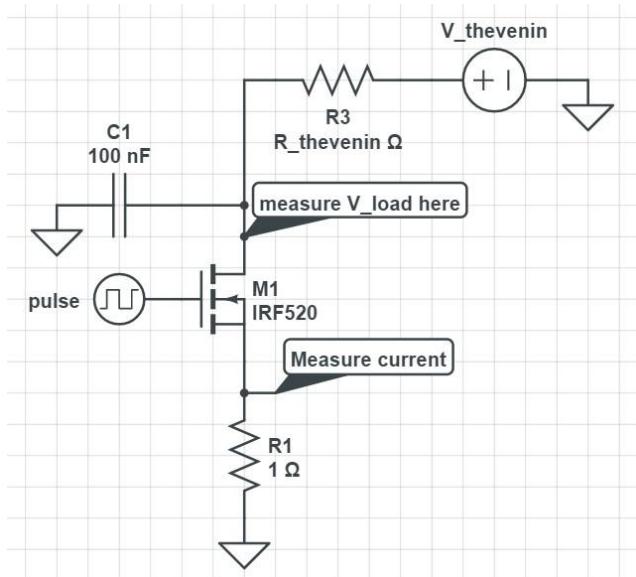
So far, you used a manual method of adding a resistor to the output and measuring the voltage drop between the unloaded Thevenin voltage and the loaded output voltage. This was manually intensive and took many seconds to do the measurement. This was ok for this specific application, but not suitable when there might be higher current draws and higher power dissipation.

For example, if you want to draw 1 A from the power source and you used a 1 ohm load resistor, this would be 1 watt of DC power consumption! You could easily burn up resistors or other components. This is why it is always good practice to estimate the amount of power consumption you expect to see BEFORE you do the measurement.

Another approach to adding a current load and measuring a voltage drop is to use a MOSFET to load the line and measure the current through a sense resistor. The MOSFET is turned on for only a short period of time, like a few msec, and the on-time duty cycle can be kept low enough so that the average power consumption is kept well below 0.1 watts, usually a safe value any component can dissipate.

When the MOSFET is used to draw some current from a power rail, we call this circuit a slammer circuit. It is an incredibly powerful circuit which we will use many times to characterize power rails. This application to measure the output Thevenin resistance is a very simple application of the circuit.

Here is the circuit you will build:



The purpose of the 100 nF capacitor is to prevent sometimes oscillations in the MOSFET when its current is turned on quickly. There can be feedback from the drain voltage to the gate which turns off the gate, which gives a dI/dt which creates more drain voltage in the opposite direction, which turns on the gate, which then drives more current, which turns off the gate, ... and you get the idea. The 100 nF will slow down the drain voltage changes. Try this circuit without the capacitor. You may not see any oscillation.

If you do, add the 100 nF capacitor as close to the drain as practical.

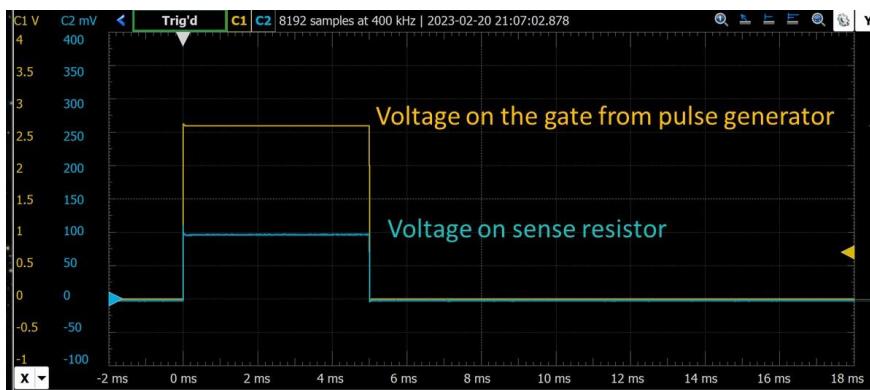
You will drive this circuit with a waveform generator using pulses. This way you can adjust the pulse width to be about 1 msec on time and change the output voltage to adjust the current through the MOSFET. Depending on the MOSFET you use, the threshold voltage is about 2 V. This means you will not get any current through the MOSFET until you apply at least about 2 V.

You will measure the current through the sense resistor. In this circuit, it shows a 1 ohm resistor. The value of sense resistor you use depends on the current range you want to look at. Pick a value of resistance so that for the maximum current draw you want to use, you will get about 1 V across the resistor.

For example, if you want to drive 100 mA, then use a 10 ohm resistor.

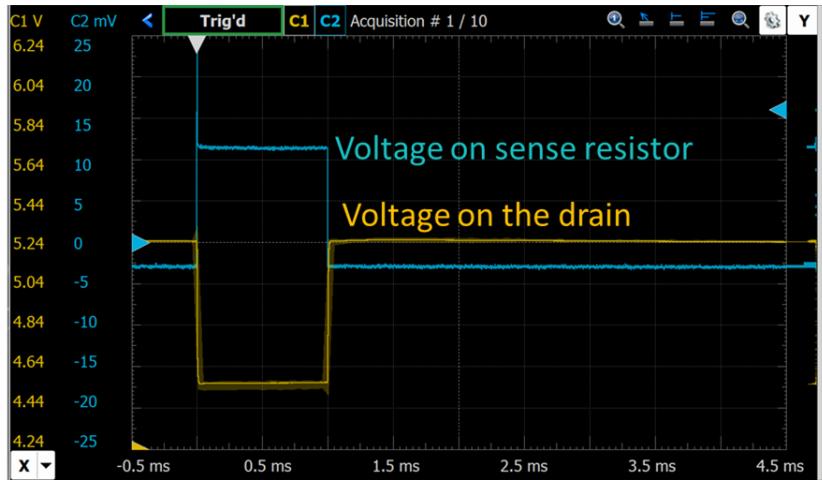
First, to set up the MOSFET, measure the input voltage on the gate and the voltage across the sense resistor as you slowly ramp up the pulse height voltage. Remember to apply rule #9: what do you expect to see? When the MOSFET turns on and you get some voltage across the sense resistor, you will see the gate voltage higher than the sense resistor voltage. This difference is the turn on threshold for the MOSFET. Until you evaluate the rest of the circuit, keep the current through the sense resistor below about 10 mA.

Below is an example of my measurement:



You will increase the voltage of the 10% duty cycle pulse generator into the MOSFET gate and measure the voltage on the gate and across the sense resistor. Once you have established you can control the sense resistor current with the pulse generator, take a look at the drain voltage.

With no current through the sense resistor, the voltage supply, the DC output of ANOTHER waveform generator, is the unloaded, or Thevenin voltage. When there is current through the sense resistor, the voltage on the drain is the loaded voltage. Here is an example of these measurements:



The Thevenin resistance can be calculated very easily as:

$$R_{\text{thevenin}} = \left(\frac{V_{\text{thevenin}} - V_{\text{load}}}{I_{\text{load}}} \right) = \frac{0.7V}{0.015A} = 47 \Omega$$

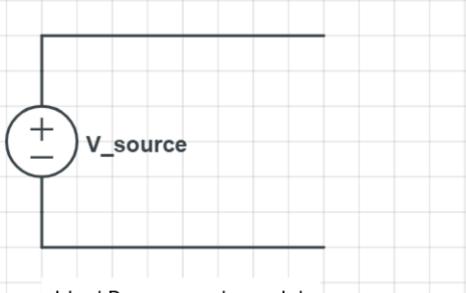
This is the output resistance of the voltage source. What do you measure for the waveform generator?

Be sure to limit the voltage drop of the Voltage source you are measuring to no more than a 25% drop from the Thevenin voltage. Usually, voltage sources turn on some nonlinear behaviors when the voltage drop is too large.

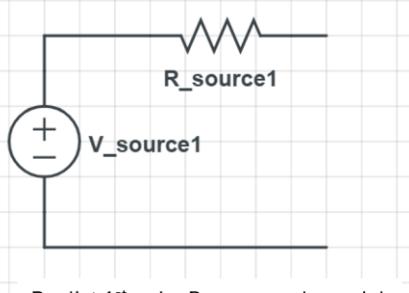
This is a generic method to measure the Thevenin voltage and Thevenin resistance of any voltage source. This is the method you will use when you design and build your board 4. You will build an instrument droid, an automated, microcontroller based board which will measure these features of any voltage source you attach to it, using this method.

11.6 Analysis and so what?

This experiment used a waveform generator as a power supply for you to characterize and learn an important lesson about power supplies. No power supply is ideal in that the output voltage is not always the set voltage. As you apply a lower resistance (larger load), the actual output voltage will be less than the set voltage. Understanding that the output voltage is a function of the load is a very, very important concept.



Ideal Power supply model

Realistic 1st order Power supply model

The waveform generator's internal resistance is large (~50 ohm) compared to dedicated power supplies (1-4 ohms). As we progress in this class, we will build more advanced ways of measuring these lower internal resistances with what we will call a slammer circuit.

Based on your measurements you should understand the following four principles of using the waveform generator:

1. *The load resistance setting on the waveform generator has nothing to do with the output impedance of the waveform generator. The output impedance of the waveform generator never changes. The load resistance term is a term you type into the waveform generator. It is supposed to be the value of the input impedance of the load you are placing across the waveform generator. This is a terrible way of designing the user interface of a waveform generator. What if you don't know the input load impedance of the device you are connecting to the waveform generator?*
2. *In order to have confidence in interpreting the settings on the waveform generator and the resulting output, always set the load impedance of the waveform generator to high Z. Then the internal Thevenin voltage is the same as the set voltage.*
3. *If you know what the set voltage is, it is up to you to know what the voltage across the DUT is, which will change as the load impedance of the DUT changes. This is what it means to say the waveform generator cannot drive a low impedance. It can, but the voltage across the load will NOT be the set voltage.*
4. *Independent of the setting for the output impedance, the Thevenin resistance of the waveform generator is always 50 ohms. You cannot change this.*

The parameter you were setting when you changed the load impedance of the waveform generator was not about the output impedance of the waveform generator. You were telling the waveform generator what load your DUT had that was connected to the waveform generator.

How do you even know what the input load impedance of your DUT is? Rarely will you know this. Why does the waveform generator need to know this?

As a best practice, always set the output load impedance of the waveform generator as high Z. This way the Thevenin voltage inside the scope is the value of the voltage you set. Its output Thevenin resistance is always 50 ohms. Given these two values, you, as an expert user, can estimate the voltage across the load if you know the load resistance of your DUT.

11.7 The lab report

While there is no lab report for Lab 3.

11.8 Grading rubric:

1 point if checked off by your TA

Chapter 12 Lab 4: Using the Scope to measure loop to loop cross talk

By this time, you should know the best measurement practices for the scope and the waveform generator. We will use them to explore the cross talk between signals due to inductive coupling.

12.1 Purpose

Purpose of this lab is to measure the loop-to-loop cross talk between two loops and exercise your understanding of the principles of mutual inductance and what design features will reduce this common source of noise.

These directions are WHAT the instruments should be set up for. You should be able to figure out HOW to set up the instruments to meet these conditions and WHY these are important settings.

12.2 Before you start this lab

To understand mutual inductance and inductive cross talk, you should have read Chapters 3 and 4 in the textbook.

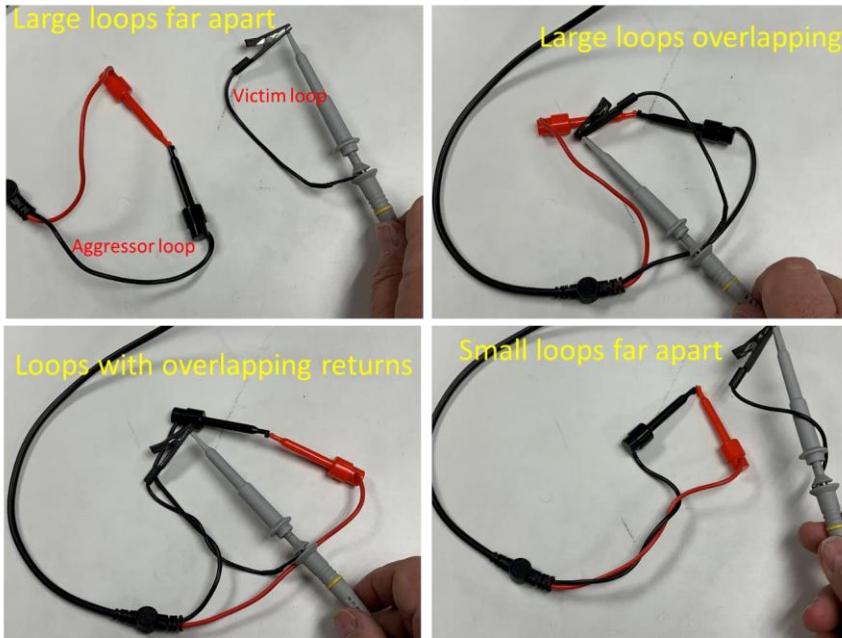
You should have mastered the labs 1 and 3, about the scope and the waveform generator. In order to start this lab you must first be an expert in using the scope and waveform generator. If you understand how to use the scope and waveform generator, you should be able to follow these simple directions and set up the scope and waveform generator correctly.

12.3 Exp 1: Set up the waveform generator as an aggressor loop

1. Use the sync signal to trigger the scope and the waveform generator on the square wave setting.
2. Use a BNC to BNC between the sync signal on the waveform generator and the scope channel 2
3. Use a BNC to mini grabber from the waveform generator output.
4. Short the ends of the waveform generator minigrabber cable together. Knowing the internal Thevenin voltage and the Thevenin resistance of the waveform generator, what is the current through the minigrabber loop when it is shorted?
5. From your previous measurement of the rise time of the waveform generator signal, when it was driving an open, what is the dI/dt in the minigrabber loop?
6. Trigger the scope on the sync waveform signal. How do you trigger the scope so that the rising edge of the signal output is also at $t = 0$? How can you verify this?
7. When the waveform generator output is shorted with the mini grabber leads, what is the direction of circulation of the current in the mini grabber?

12.4 Exp 2: measure the cross talk between the aggressor loop and the 10x probe as the victim loop

1. Use the 10x probe as the victim loop, with the signal and return leads shorted together. The 10x probe will be connected to channel 1 of the scope. The scope is triggered by the sync waveform from the waveform generator so the rising edge of the square wave of the waveform generator output is at $t = 0$.
2. Short the leads of the 10x probe leads together. This is the victim loop. We should not see any voltage displayed on the scope- the input to the 10x probe is shorted together. How do we get a voltage across a short, right? When the 10x probe and the minigrabber leads are far apart, what is the voltage displayed by the scope? Apply rule #9.
3. Measure the voltage noise induced in the 10x probe synchronous with the waveform generator edge. How will you trigger the scope to measure the synchronous noise in the 10x probe? Hint: DO NOT trigger the scope on the 10x probe signal.
4. When the 10x probe victim loop is placed on top of the aggressor loop, what is the signature of the noise voltage? Why does it have the shape it has? From the magnitude, estimate the mutual inductance. Remember, there is not a DC connection between the 10x probe victim loop and the waveform generator aggressor loop. Yet, there is coupling between them. Why is this?
5. Flip the orientation of the loops. What happens to the signature of the noise? Why?
6. Explore the geometry for the highest noise coupling and the lowest noise coupling.
7. Can you generalize from this experiment, design guidelines to reduce the amount of inductive cross talk between loops? In the figure below are four examples of loop-to-loop geometries you can explore.



12.5 Check off by your TA

Before you complete the lab and to get credit for the lab, your TA will come around and ask you questions about what you are doing and will ask you to demonstrate some features of your measurements. You must get an OK to get credit for the lab.

You may be asked any of the questions above and to demonstrate any of the measurements above or to explain any of your measurements.

12.6 The lab report:

While there is no lab report for Labs 3 or 4, writing them up as a lab report would look great on your portfolio page. You can demonstrate an important electrical effect of inductively coupled noise between an aggressor and a victim loop to any hiring manager. They eat this stuff up!

12.7 Grading rubric:

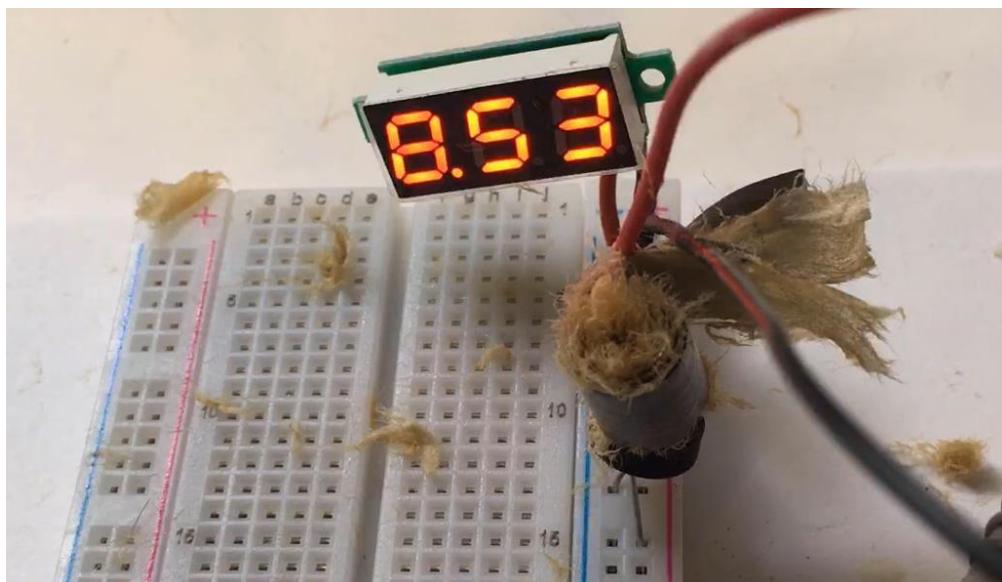
1 point if checked off by your TA

Chapter 13 Lab 5: SBB PDN and slammer circuit

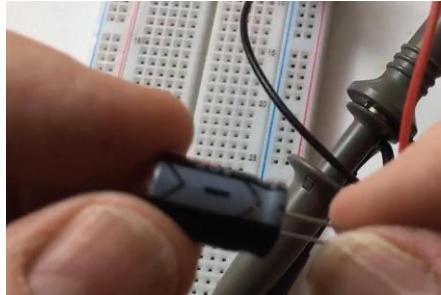
In this lab, you will build a simple slammer circuit which will draw a fast transient current from the power rail. This is exactly what happens when an IC suddenly switches current as when it drives I/O signals, or the core logic consumes a lot of current to perform some computational operation, which happens on each clock edge.

13.1 Important safety tip

Before you start this lab, be aware there is a potential danger in this lab. You will be using an electrolytic capacitor on a 9 V rail. If you connect the capacitor incorrectly, YOU WILL HAVE AN EXPLOSION. Be very careful how you connect the capacitor. Remember the most important rule in our class is safety always comes first. Here is the problem we want to avoid:



The electrolytic capacitor is labeled with a negative label on one side, shown below.



This pin ALWAYS plugs into the low voltage side of the power rail. This is the safe connection. If you connect the negative lead into the 9 V rail and the other lead into ground, the capacitor will blow up. DO NOT DO THIS.

Before you begin this lab, [view this video where](#), at great personal risk, I show you the consequence of connecting up the capacitor incorrectly. DO NOT DO THIS!

13.2 Purpose of the lab

When there is a sudden current draw on the power rail, current flows through the inductance of the power rail. The dI/dt through this inductance causes a voltage drop. When the current turns off, there is also a large change in the current and a spike up in voltage on the power rail. When the current turns off, we refer to this as "the release." This is a serious type of switching noise which can be dramatically reduced with a decoupling capacitor.

You will measure the signature and value of the voltage drop or droop on the power rail and the spike up during the release and see how much you can reduce it using the local charge storage of a decoupling capacitor.

The purpose of this lab is:

1. *To build a circuit demonstrating the origin of switching noise in the power path and the role of loop inductance.*
2. *To measure the switching noise on the power rail when there is a large current transient.*
3. *To explore the role of loop inductance between the IC and the decoupling capacitors.*
4. *To see the difference in switching noise for different dI/dt values of current transient.*
5. *To estimate how much capacitance is needed to provide adequate local charge storage.*

Through this exercise, you will explore the seven most important design principles to reduce noise in the power delivery path:

1. *Reduce the loop inductance between the IC that switches (the aggressor) and the nearest decoupling capacitor.*
2. *Keep the decoupling capacitor as physically close as possible to the IC it is decoupling.*
3. *Use at least 1 uF of decoupling capacitance, and then as large a capacitance as practical, depending on the circuit conditions.*
4. *Where you measure the noise on the power path influences how much noise you measure.*
5. *Short rise time current transients have a larger dI/dt and show more power rail noise than long rise time current transients.*
6. *When there is a step change in the current on the power rail there is more noise generated than just from the dI/dt . There is also an IR drop from the Thevenin resistance of the VRM and power rail.*
7. *The voltage on the power rail spikes up when the current in the power rail turns off.*

13.3 Prep before you start this lab

Read the section in the textbook about power rail switching noise, Chapter 13, and best practices using the solderless breadboard, Chapter 11.

[To get an overview of this lab, view this video.](#)

13.4 The big picture

You will build your circuits using a solderless breadboard, using best design practices as described in the textbook, in Chapter 11.

The slammer circuit is a MOSFET that turns on and draws a fixed current from the power rail. It will be triggered by either a digital signal from an Arduino digital I/O which has a 5 nsec rise time or by an opAmp output with a rise time of about 1 usec. You could just as easily use an NPN transistor. You will get a little larger current with the NPN and it will have a slightly shorter rise time, so the rail collapse noise will be larger with the transistor than the MOSFET, but the conclusions and the effects will be identical. Your choice.

The different rise times for the current from the power rail to turn on mean very different dI/dt . You will measure the switching noise on the drain pin (or collector pin if an NPN) for these different rise times with and without decoupling capacitors.

You will use an Arduino as the trigger source to turn on the MOSFET or NPN transistor. Its output rise time is very short, so you will also use an opAmp with a much longer rise time, to trigger the MOSFET so you can compare the power rail noise with two different rise times of current.

13.5 Exp 1: Install the Arduino IDE

1. Download the Arduino IDE from [Arduino.cc](https://www.arduino.cc). Then install it. I recommend using the older version of the IDE, rev 1.8.19. Do not install 2.x
2. Launch the Arduino IDE for the first time. A blank sketch will open up.
3. Connect your Arduino board to a USB port using the appropriate USB cable.
4. Under Tools/boards, select the Uno board. Under Tools/port, select the COM port to which your Arduino is connected.
5. In the blank sketch that opens automatically, press the upload button and lights should momentarily flash on the Arduino board and you will see "Upload done" at the bottom of the sketch.
6. Congratulations! your computer can now successfully communicate with your Arduino.

The Arduino will provide the input signal to turn on the MOSFET's gate, the slammer circuit.

Pin 13 of the Arduino will be used to generate the signal that triggers the slammer circuit. Modify the blink code to use an on-time of 1 msec and an off-time of 20 msec.

The code is simply:

```
void setup() {  
pinMode(13, OUTPUT);  
} void loop() {  
digitalWrite(13, HIGH);  
delay(1);  
digitalWrite(13, LOW);  
delay(20);  
}
```

You can literally copy this sketch from this soft copy and paste it in a blank sketch, upload it and it will run.

Pin 13 (and the ground connection) will be the signal source.

Be sure to use a duty cycle less than 10% or the MOSFET may get too hot.

If your computer does not see the port for the Arduino, you may have to download the driver for the USB to UART interface. It will either be the FTDI chip or the CH340g chip driver. You can get both of these from the Sparkfun web site.

For the driver for the FTDI chip visit the [Sparkfun site](#) for details to download and install the driver. On a windows computer, [download this driver and install](#).

If your Arduino Uno board has the CH340g driver chip, visit this [Sparkfun page](#), or [download this driver](#) for a windows computer.

If you are not sure which chip is on your Arduino, look carefully at your board. If you are still not sure, download and install both drivers. You cannot go wrong.

If you are not sure to which port your Arduino is connected, make note of the ports identified when you select ports under the Tools menu of your Arduino IDE. Then disconnect your Arduino. If the ports identified have not changed, your computer is not seeing your Arduino. Download the drivers, close the IDE and open it again.

If a port is missing when you disconnect your Arduino, the missing port is the one connected to your Arduino. Select this one.

When your computer can see the Arduino port and you can upload the code, and you can see the digital signal on pin 13 with the scope, you are all set to go.

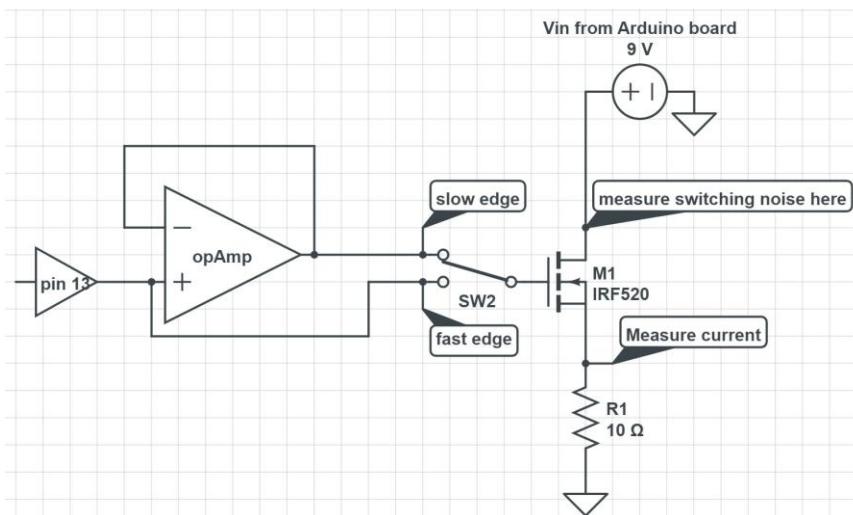
13.6 Exp 2: Build the slammer circuit with no decoupling capacitors and use the slow edge from the op amp.

We will build the slammer circuit in stages. Your Arduino will connect through USB to your computer. In addition, plug the 9 V AC to DC power supply into the power jack of your Arduino. This way, you will have a 9 V power rail available on the Vin pin of your Arduino board.

Don't take my word for it. Measure it with a scope.

Use this Vin connection to power the drain of the MOSFET.

First, we start with just the MOSFET and NO decoupling capacitors. The slammer circuit is shown in the figure below.



Do not connect the 9 V power to the transistor until you have estimated the power consumption and know what to expect.

The MOSFET circuit is often called a slammer circuit in that it will slam or sink a high current from the power rail when the MOSFET turns on.

The MOSFET (or NPN) will turn on with a low resistance and draw current from the power rail which flows through the 10 ohm resistor. The current will be limited by the voltage drop of the threshold voltage and the current drop across the current sense resistor adding up to the input voltage between the gate and local circuit ground.

You will measure the current waveform directly with the scope in channel 1. We can estimate it ahead of time, rule #9. If the turn on threshold of the MOSFET is about 1.5 V, then the current flow will be about

$$I = \frac{(5V - 1.5V)}{10\Omega} = 350mA$$

When the signal to the gate turns on, there will be about 350 mA current draw from the power rail. This current will turn on with the rise time of the source filtered by the response of the MOSFET.

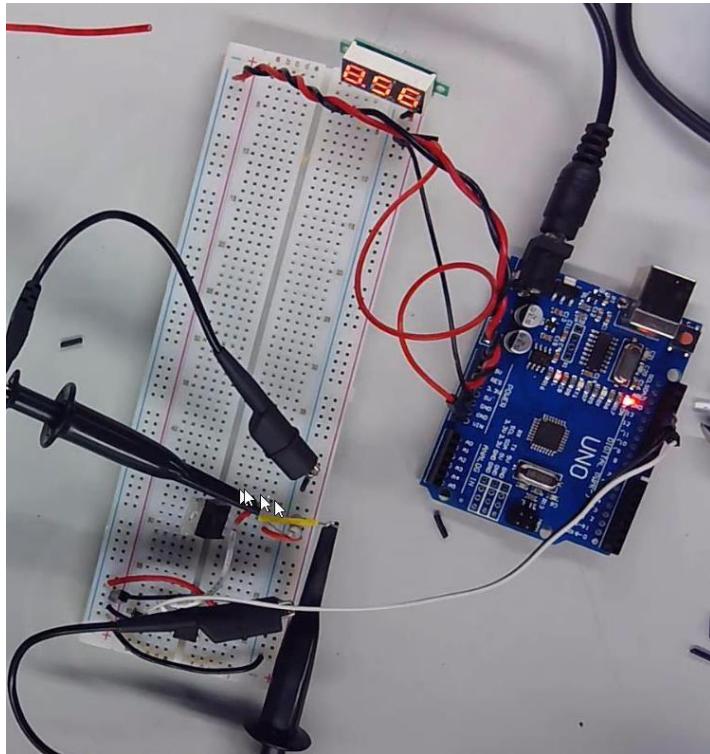
13.6.1 Use best design practices to assemble the SBB circuit

Follow the guidelines in Chapter 11 for using a SBB. These include short leads, color coded with power and ground using the vertical rail columns. Pick a strategy of ground on the inside, or the outside and be consistent.

13.6.2 Layout on the solderless breadboard

In this experiment, we want as long a distance between the slammer circuit and where we connect power to the SBB power rails. This means you should place the MOSFET at the bottom of the SBB and connect power to the top of the SBB. This will allow you to move a decoupling capacitor at various distances from the slammer circuit, between the slammer circuit and where the power supply connects to the SBB.

On the solderless breadboard, keep the MOSFET and the opAmp as close to the bottom of the solderless breadboard as convenient. This is illustrated below.



Add the op-amp follower at the bottom end of the solderless breadboard. Check the data sheet for the opAmp to see how to connect it up. Use the 5 V from the Arduino 5 V rail to power the opAmp. It does not matter which opAmp you use for this lab. Just check its datasheet to make sure you are connecting it up correctly.

Set up the op-amp in your board as a follower. Connect the output of pin 13 to the input of the follower circuit.

In the first series of measurements, use the slow edge, from the buffered opAmp output signal, as the input to the base of the slammer transistor. You can measure the voltage out of the op amp and see its long rise time.

Use best practices when measuring these signals with the scope. This generally means a 10x probe with short connections between the probe tip and the DUT. Use the correct scale settings on the scope.

1. *Measure the Arduino pin 13 signal with the scope when it is running and verify the duty cycle is less than 10%.*
2. *Be sure to use color coding for the solderless breadboard and the power and ground right hand side columns.*

3. Did you verify that the top half of the columns are connected to the bottom half of columns on your solderless breadboard?
4. You will be using two power rails on your solderless breadboard. Use the left columns for 5 V and the right column for 9 V, the power rail on which you will measure the switching noise. Do not plug the 9 V into the solderless breadboard yet until you have estimated the power consumption and the low duty cycle signals as described in the next experiment.
5. Power the opAmp with 5 V from the Arduino using a black and red wire from the Arduino to the solderless breadboard.
6. Did you use the correct color code?

13.7 Exp 3: Power consumption and dissipation and not blowing up a component

At steady state, the worst-case power consumption in the MOSFET is about $9\text{ V} \times 0.4\text{ A} = 3.6\text{ watts}$. This will heat the transistor, a lot! In practice, it would be less than since the voltage drop across the MOSFET would only be $9\text{ V} - 5\text{ V} = 4\text{ V}$. But, this is a worst case estimate.

The 10 ohm resistor acts as both a sense resistor to measure the current and a current limiting resistor to set the current. If the current through the resistor is 400 mA, the instantaneous power consumption through the resistor would be:

$$P = I^2 * R = 0.4^2 * 10 = 1.6\text{ watts}$$

The resistor is rated at only $\frac{1}{4}$ watt. This power consumption would be way more than the resistor can handle without heating to a high temperature. This power consumption is almost 10x the rated power dissipation of the resistor. If we run this current at DC, the resistor will get so hot that it will probably smoke, or worse.

This means, to send this much current through the resistor, we can't keep it on very long. To keep the average power consumption reduced by 10x, we use a pulse width modulated signal, with a duty cycle of less than 10%.

If we want the on-time to be 1 msec, the off-time should be at least 10 msec, and preferably 20 msec. This would be a frequency of 1/21 msec or about 50 Hz.

Now it should be obvious why it is so important to use a signal with a duty cycle less than 10%. Otherwise, the MOSFET or the 10 ohm resistor will smoke!

Verify the Arduino pin 13 has a < 10% duty cycle.

13.8 Exp 4: Set up the two driver signals to the transistor's base

Now that you see why you need a < 10% duty cycle signal, you can connect 9 V to the MOSFET.

Connect up the 9 V rail power to the MOSFET using the Arduino with the 9 V AC to DC converter plugged in.

In all the measurements in this lab we will use the 10x probe.

Remember, it is easy to make a measurement. It is hard to make a measurement without introducing artifacts.

When the pin 13 signal (and its return ground pin connection), is connected to the input of the opAmp buffer, there will be two signals available to drive the rest of the circuit: the raw input from the Arduino and the buffered output from the opAmp follower.

You can connect them into the gate of the MOSFET by how you connect the wires.

What is the rise and fall times of these two signals?

Be sure to measure both the rise and fall time for both of these signals using a 10x probe. It is very important to verify that the on-time is about 1 msec and the off-time is at least 10 msec or longer. If the duty cycle is larger than 10%, the circuit may heat up too much.

13.9 Exp 5: Measure switching noise with a slow rise time and no decoupling capacitors

1. *Initially, connect the slow rise time signal from the buffered output of the opAmp to the gate of the MOSFET. It should be a 5 V signal, 5% duty cycle.*
2. *Now you can connect the 9 V to the MOSFET power rail. Add the wires from the Arduino's 9 V supply and gnd at the top end of the SBB, far away from the slammer circuit.*
3. *Be sure to have no decoupling capacitors.*

Practice Rule #9 for all measurements.

4. *Measure the voltage on the gate. What did you expect to see?*
5. *Measure the voltage on the sense resistor. What is the current through the drain?*
6. *What is the rise time of the current and the dI/dt?*
7. *Use the current through the emitter as the trigger for the scope on chan1*
8. *On chan2, measure the switching noise on the power rail of the die- this is the collector pin.*
9. *What is the switching noise on the collector? Why does it have the shape it does? At short rise time and long rise time?*

10. How much of the voltage on your probe is from inductive coupling in the loops of the probe? How can you minimize the loop area?
11. From the steady state voltage droop on the power rail for a 400 mA current load, after the initial inductive spike, what do you estimate the Thevenin source resistance of the VRM to be? Remember, every voltage source can be modeled as a Thevenin source.
12. Does the noise change as you move down the power rail on the solderless breadboard?
13. Be sure to measure the switching noise on both the rising and falling edge of the current.

If the actual circuit is only the circuit in the figure above, there should be no voltage noise on the 9 V rail. Afterall, the 9 V source is an ideal voltage source in the schematic, right?

But, this is not the complete circuit. It is missing two important elements, the Thevenin model of the VRM and the loop inductance of the power conductors from the VRM on the Arduino to the collector of the transistor.

Draw the equivalent circuit model that includes these elements. Since the Thevenin resistance of the VRM and the interconnect inductance of the conductors from the VRM to the collector of the transistor are not actual discrete components you can see on the board, you must learn to see them with your engineer's mind's eye.

Where would you add them into the equivalent circuit model?

Given this model, why do you see the signature of the voltage noise you see?

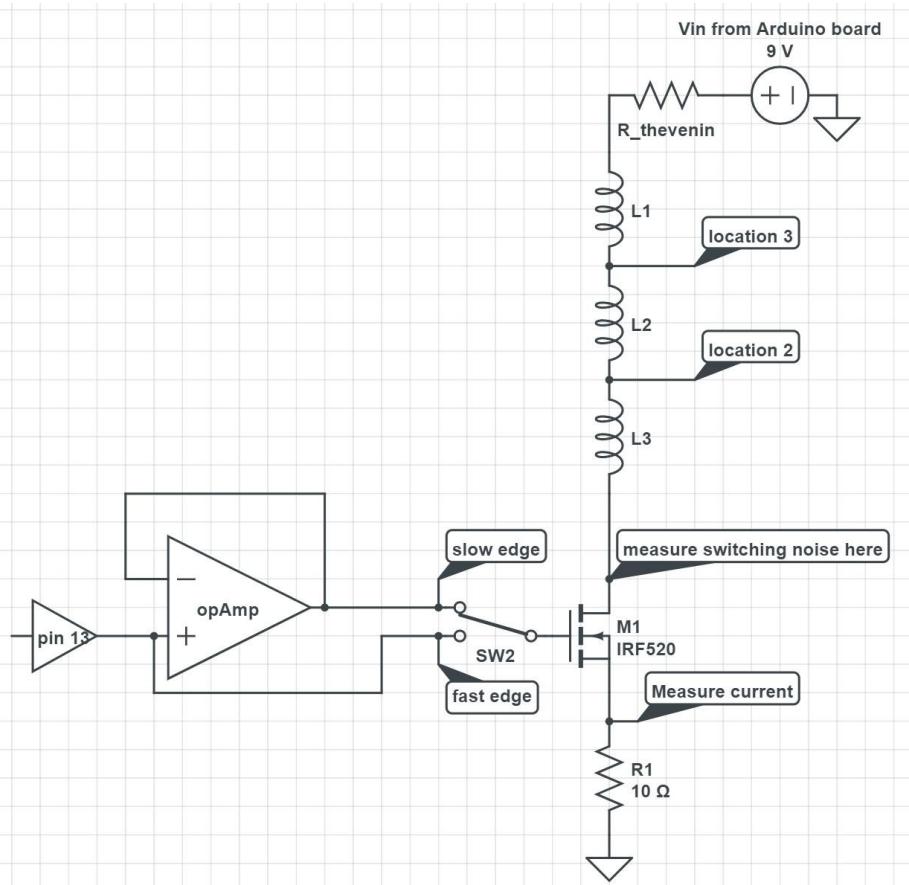
This is the switching noise with a slow edge. The biggest impact is from the IR drop from the source resistance of the VRM. It is a small amount of noise.

Given the voltage you measure, what do you estimate the Thevenin voltage and resistance of the Arduino VRM?

If you wanted to reduce this IR drop noise, what could you do to reduce the DC IR drop?

Is it any wonder that when your signal rise time is long, switching noise is not important?

Here is what I think the equivalent circuit model is:



Note, these inductors in the power rail are not discrete inductors you are adding. When you look with your eyes at the power rail of the solderless breadboard, you will not see any discrete inductor. You will just see the wires inside the solderless breadboard. You have to see these inductors with your engineer's mind's eye. This is a very important skill for all engineers. You see the interconnects with your eyes but see their equivalent inductance with your engineer's eye.

Each inductor models a piece of the power rail interconnects on the SBB. They will all have the same current and the same dI/dt through them. The closer to the slammer circuit, at location 2, for example, the more voltage droop noise you will measure compared to location 1.

But with a long rise time, dI/dt is small and the switching noise will be small.

13.10 Exp 6: Measure switching noise with a fast rise time signal

Now we will use the Arduino pin signal as the input to the base.

1. *Repeat these measurements using the fast rise time signal into the gate, instead of the slow rise time signal.*
2. *What is the rise time of the current turn on and turn off?*
3. *Measure the switching noise on the drain pin for the rising and falling edges.*
4. *How does the noise vary as you measure closer or farther from the slammer circuit?*
5. *Look at the signal on a very fast time scale and a slow time scale.*
6. *From the equivalent circuit including the interconnect loop inductance in the power rail, can you explain the features in the switching noise from this circuit model?*
7. *Why is the switching noise so different between the slow edge and the fast edge?*
8. *What do you think the small ripple voltage noise on the power rail is from? Hint its frequency is about 16 MHz, the same as the clock of the Arduino. And it is only present when the Arduino pin drives the current load on the power rail.*

Remember, the voltage drop across an inductor is:

$$\Delta V = L \frac{dI}{dt}$$

If we see a voltage drop of 8 V across the inductor, and the $dI = 400 \text{ mA}$ and the $dt = 50 \text{ nsec}$, then the loop inductance is $8 \text{ V} / 0.4 \text{ A} \times 50 \text{ nsec} = 1000 \text{ nH}$.

Why did we not see any switching noise with the opAmp signal having a rise time of 2 usec?

The expected switching noise was $1000 \text{ nH} \times 0.4 \text{ A} / 2 \text{ usec} = 0.2 \text{ V}$, too small a value to see compared to the pattern of the changing current.

13.11 Exp 7: Add the decoupling capacitor

In this experiment we will explore the impact on the switching noise we measure at the collector from the location of the decoupling capacitor and its value.

The voltage noise on the collector is due to the inductance and the dI/dt in the power rail. First do not add the decoupling capacitor.

As you move the measurement point closer to the VRM and measure the switching noise, you will see less switching noise because there is less loop inductance to the VRM.

1. *Measure the switching noise on the drain pin with the fast edge. This is the switching noise that would appear on the die itself.*

2. Repeat these measurements using the slow rise time signal. Is there any difference?

Be sure to apply Rule #9. What do you expect to see happen to the switching noise signature?

3. Measure the switching noise with the fast edge.
4. Use a large value capacitor as the decoupling capacitor. Before you insert it in the power rail between the 9 V and the ground, check its polarity.
5. If you reverse the polarity and insert the + side to the gnd and the - side to the 9 V, it will literally explode. DO NOT DO THIS. ([see the demo of the exploding capacitor](#))
6. Place the large capacitor far away from the drain pin and measure the switching noise on the drain pin.
7. What do you think will happen to the switching noise if we move the capacitor closer to the MOSFET drain pin?
8. Why does the switching noise decrease as you move the capacitor closer to the drain pin?
9. Look at the switching noise on different time scales and with the current switching off and switching on.
10. What is the impact on the switching noise from using a short rise time with pin 13 and long rise time, using the output of the opAmp?
11. When the decoupling capacitor is close to the collector pin, what is the switching noise on the rest of the power rail?
12. From this behavior, what is the decoupling capacitor actually decoupling?
13. Replace the 1000 uF capacitor with a 1 uF capacitor. Repeat these measurements. Is there any impact? The capacitor is much smaller, but still large enough. Above a certain value capacitance, it is not about the amount of capacitance it is about the loop inductance to the capacitor.
14. Does the value of the decoupling capacitor, either the 1000 uF or the 1 uF capacitor, effect the amount of switching noise?

13.12 Exp 8: Estimate the size of the decoupling capacitor you need to use

When the 350 mA of current flows through the drain on the power rail, there are three features in the voltage response on the power rail of the MOSFET drain:

1. The initial voltage droop due to the loop inductance in the power delivery path
2. The slow drop in voltage due to the charge depletion in the capacitor
3. The steady state DC voltage drop in the VRM due to its Thevenin source resistance

From the steady state voltage drop on the 5 V rail, calculate the Thevenin resistance of the 9 V VRM you are using to power the rail.

If the capacitor provides some charge storage, how slowly will the voltage drop with the 350 mA of current draw? How does this slope compare with the slope you measure for the voltage drop? You should be able to derive that:

$$\frac{dV}{dt} = \frac{1}{C} * I$$

If the current is 0.4 A and the C = 1000 uF, the dV/dt = 400 V/sec or 0.4 V/msec. How does your estimate of the slope of the voltage droop match what you actually measure?

During the time of the rising or falling edge, dt, when there will be inductive switching noise, we want all of the I to come from the capacitor, so none of it has to flow through the rest of the inductance of the power rail. But, this current only has to flow during the rising or falling edge. If we want to limit the voltage drop or droop, during the dt time, to 0.4 V, for example, how much capacitance do we need? Assuming the rise time is 1 usec, the capacitance we need to limit the voltage droop to 0.4 V is:

$$C = \frac{I * dt}{dV} = \frac{0.4 * 1\mu S}{0.4V} = 1\mu F$$

This says, anything more than about 1 uF will keep the voltage droop below 0.4 V. If you use 1 uF or 1000 uF, the voltage droop will be less than 0.4 V. If the rise time were shorter than the 1 usec of the op amp, what would the minimum size of the capacitor have to be to prevent a droop of more than 0.4 V?

If you want to reduce the droop voltage, do you want a bigger or smaller capacitor value? How will the sharp dip from the L di/dt noise be affected by using the 1 uF or 1000 uF capacitor? How much charge depletion is there in the capacitor during the dt time?

Keep in mind that while this is the voltage droop on the capacitor, there is still some loop inductance in the power/ground rail between the MOSFET and the decoupling capacitor. The L di/dt noise will also be present on the MOSFET drain pin.

But, the switching noise will only be from the inductance from the drain pin to the decoupling capacitor. The capacitor will "decouple" the rest of the inductance from the capacitor to the VRM itself.

13.13 Exp 9: change the location of the decoupling capacitor

Using the 1 uF capacitor, measure the voltage droop switching noise from the inductance of the power rail interconnects, with the scope probe on the collector. Use the Arduino pin itself with a 5 nsec rise time as the driver for the slammer circuit.

What is the rise time of the current through the sense resistor? What is the total pulse width of the voltage droop? It is during this time all the current should come from the decoupling capacitor. Given this time and the current, what do you estimate the voltage drop to be for the 1 uF and 1000 uF capacitors? What do you measure?

Why does this voltage droop increase as you move the decoupling capacitor location farther and farther from the switching slammer circuit?

Measure the switching noise on the drain with the 1 uF capacitor close to the drain pin and then farther from the drain pin. Where should you locate the decoupling capacitor to reduce this noise the most? Does the switching noise depend on the size of the capacitor?

13.14 Check out by your TA

Before you leave the lab, call your TA over for a check out of your experiments. Be prepared to answer any of the above or following questions:

1. *What is the equivalent circuit model for your circuit, including the VRM and the power rail inductance?*
2. *What is the transient current through the MOSFET? What is the rise and fall time?*
3. *Where does the power rail switching noise come from?*
4. *Why is it called switching noise?*
5. *How are you triggering the scope?*
6. *How much inductive pick-up noise is there between your probes?*
7. *When decoupling the power rail, what will be the impact of using a larger value decoupling capacitor?*
8. *What is the most important quality of a decoupling capacitor to decouple the power rail noise?*
9. *What did you observe for the switching noise with the slow and fast edges and the rising and falling current edges and no decoupling capacitors? How do you interpret the results?*
10. *Why is switching noise a bigger concern for shorter rise time signals?*
11. *What did you observe after you added the decoupling capacitor? To reduce the switching noise, where do you want to place the decoupling capacitor and why?*
12. *What do you conclude about the size of the decoupling capacitor to use to reduce switching noise and its location? What is the most important quality of the decoupling capacitor?*
13. *What feature of the interconnect are you reducing by moving the decoupling capacitor as close to the slammer circuit as practical?*
14. *Based on these observations, what do you conclude about the best design practices for designing the power delivery path and the use of decoupling capacitors?*

13.15 In your report, you should include

1. *Draw the equivalent circuit of the slammer circuit including the Thevenin model of the 9 V VRM and the loop inductance of the path from the VRM to the MOSFET drain.*
2. *Show a photo of your circuit.*

3. With a scope screen capture, illustrate the impact on the switching noise with and without the decoupling capacitor
4. With a scope screen capture, illustrate the impact on the switching noise of the 1 uF and 1000 uF capacitor.
5. Based on your measurements:
 - a. What is the Thevenin voltage and resistance of the VRM?
 - b. What is the loop inductance of the power path from the collector to the VRM?
 - c. Show your measured values and how you made these estimates.
6. Be sure to add your analysis and interpretation of each scope trace you include. Just including a scope screen with no explanation is worthless.
7. You may need 2 pages to fit the scope plots, your analysis and your calculations.
8. Remember, a good explanation and associated scope traces to illustrate your explanation will look great on your portfolio. **Hiring managers will eat this up.**

13.16 Grading rubric:

1 point if you are checked off by your TA

1 point if your scope traces clearly show the features you want to illustrate the cause of switching noise and your explanations are clear.

1 point if you offer a clear explanation of this lab and what you learned about design principles from these experiments

Chapter 14 Lab 6: Assembly practice board

14.1 Purpose of this lab

The purpose of this lab is to become an expert at soldering 1206 parts to a board and to experience the difficulty of assembling other size parts.

Before you come to this lab, be sure to view the [Skill Building Workshops-3 on Secrets to Great Soldering](#).

As you run the lab, you can review the specific videos and then follow their directions.

Read the sections in the textbook about thermal reliefs pads, section: 20.7

Remember, a good solder joint is about three important conditions:

- *The right tip temperature, about 700 deg F*
- *A clean tip- clean with solder flux or rosin core solder and the brass sponge*
- *Plenty of solder flux*

If you do not use solder flux, you will make terrible solder joints.

You should ALWAYS ask for additional solder flux. Use the solder pens supplied as a **paint brush** to apply the solder flux from a pool of solder flux on your blue silicone rubber mats.

You can never have too much solder flux.

14.2 What you will need

For this lab you will need:

- *the practice assembly board*
- *a kit of 0 ohm jumpers of sizes 1206, 0805, 0603 and 0402*
- *a soldering kit*
- *safety glasses*

Check out a soldering kit from your TA. If you are missing any elements like solder, solder flux, tweezer, or fan, let your TA know and they will replace the parts.

Life happens. If you break something or use up something, let your TA know and they will replace it.

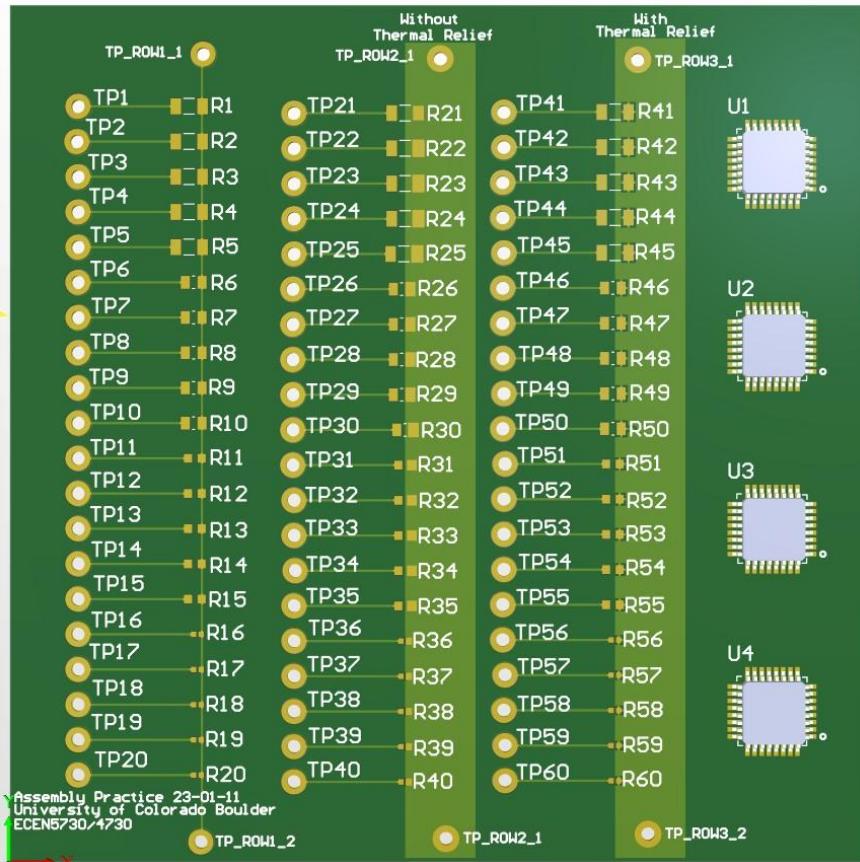
When you turn in your soldering kit, it should be complete.

These kits are used by many students who are not as careful as you are. You are helping the lab by keeping the kit up to date.

Remember. ALWAYS wear safety glasses when you are soldering. If you see someone soldering without safety glasses, gently remind them they need to wear safety glasses.

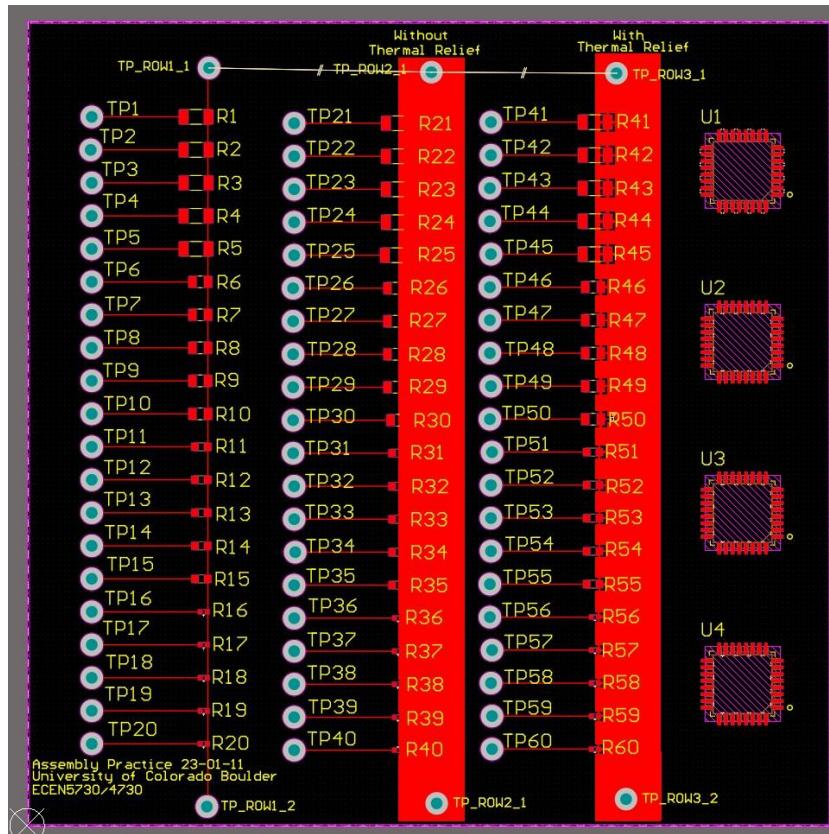
14.3 What you will do

There are three columns of pads in the test board, shown in the figure below.



The assembly test board with a few parts already assembled on it. Note the three columns each with a different type of pad on the right side.

Here is the artwork:



The top layer of the test board, showing the copper on the top layer, the solder mask pads and the silk screen.

This board has a HASL finish with leaded solder. There is already a small amount of leaded solder on each pad.

Look over this board carefully. Note that there are groups of 5 identical assembly test pads.

The top-most group of pads are for 1206 parts. The next ones down are for 0805 pads, and the next, 0603 pads and then 0402 pads.

You will practice assembly of 0 Ohm jumper parts on the pads. The test of a good solder joint will be a visual inspection and using a DMM to measure the resistance from each test pad to the bussed test pad, labeled TP_ROW1, TP_ROW2, TP_ROW3

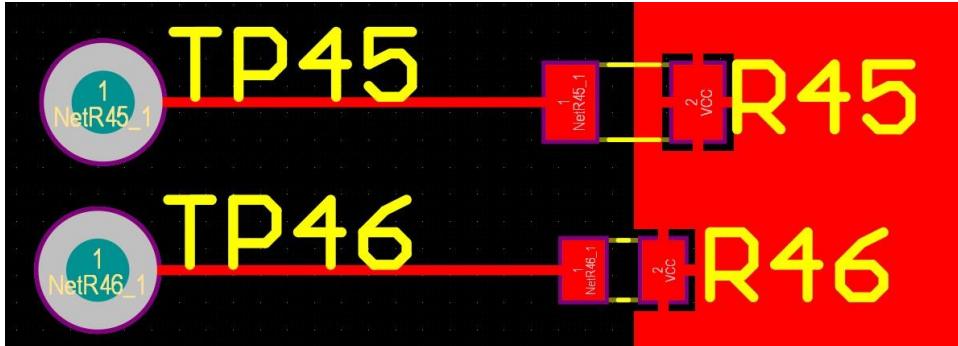
The middle column of pads, starting with TP21, have their right pads embedded in part of a plane. This copper plane will suck heat from the soldering iron and make it more difficult to solder the part.

You may need a higher temperature or apply the heat for a longer time to melt the solder. This is a lesson in being careful placing pads in the middle of planes and just using solder mask to define them. Such pads are difficult to solder to. Same with through holes.

If you design a via hole into which you plan to insert and solder a pin, that connects to a plane or large piece of copper, make sure you use a thermal relief via.

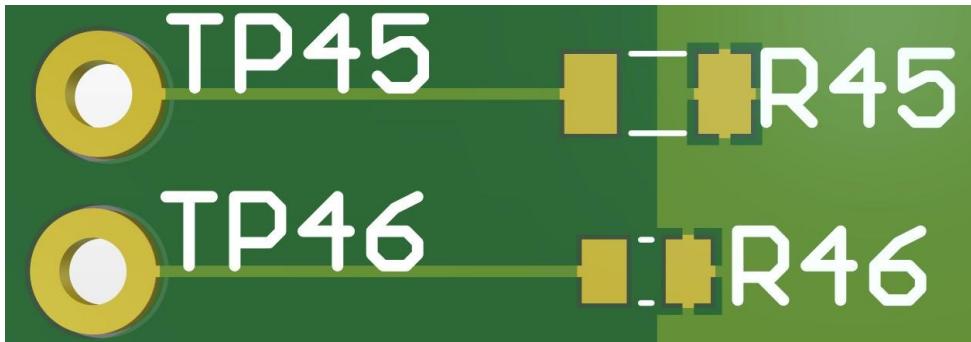
This is the ONLY purpose of a thermal relief via. There is no need for a thermal relief via if you are not planning to solder to it, or near it.

The third column from the left, starting with TP41, has the same pads in the middle of a plane, but using a special connection to the pads called a thermal relief. The figure below shows a closeup of pads with a thermal relief.



The pads in the far right column are in the middle of a plane, but separated from the plane using thermal reliefs.

Here is the board showing the solder mask opening:



The green colored square on top is the solder mask. This is the opening in the solder mask where solder is exposed. Note that only the pad is exposed for soldering.

Connecting the pad and the plane are a few very narrow, short traces which will increase the thermal resistance from the pad to the plane and keep heat from flowing quite as fast as without them.

Generally, the width of the legs should NEVER be narrower than 6 mils, otherwise the fab house will charge more.

Your first question will be, what about the resistance of the thermal relief legs. Won't these narrow segments increase the resistance from the pad to the trace?

You should be able to quickly estimate this resistance just by looking at the pad. Take a moment to perform your own estimate.

Here is my estimate: Each leg is about 1 square in size. Its resistance will be 0.5 mOhms. There are four in parallel. The equivalent parallel resistance will be $0.5 \text{ mOhm}/4 = 0.12 \text{ mOhms}$. This is so tiny as to be insignificant and not a concern.

Thermal relief pads, and the same pattern for vias, are very useful when we are soldering to the pads or have a through hole component going through the via. Their sole purpose is to facilitate soldering to the pad.

The third column is pads with thermal reliefs for each pad. They should be easy to solder.

14.1 Exp 1: Clean your soldering iron tip

Always start with a soldering iron with a clean tip. This means no tin oxide on the tip. Solder should melt on the surface and leave the tip shiny. If the tip is a dark grey or solder beads up on it, the tip is covered with tin oxide. Cleaning a tip means cleaning off the tin oxide. This is also called "tinning" the tip. Practice tinning your soldering iron. This will be a trick you use over and over again. There are many ways of doing this.

The quick and easy way is using the tinning cup that is in your soldering kit. If you do not see the tinning cup, ask your TA for one. It is basically solder paste with a small amount of solid flux.

Make sure your soldering iron is at about 700 degF. Dip the hot soldering iron tip into the dish of solid solder and paste. It will melt the solder and the solder flux will dissolve any oxide from the surface of the tip. Roll the tip around until the tip is evenly coated with solder. It takes less than 5 seconds. Repeatedly wipe the tip of the soldering iron in the copper sponge to clean off excess solder.

Alternatively, the solder wire has flux in its core. When you melt some of the solder wire, you will bring flux in contact with the tip. If there is not too thick a layer of tin oxide on the tip, you can dissolve it by just melting a few inches of solder wire with the tip.

You should melt solder wire that contains a resin core on the tip until the solder flows over the surface of the tip, while also every few seconds, rubbing the tip in the brass sponge and applying new solder.

Note, when the hot soldering iron encounters the solder flux, the flux will vaporize. Do not breathe in the solder flux vapor. There should be an orange fan in your solder kit which will suck up the solder flux vapors. Use it.

If the tip is kept at 700 degF, you will have to re-tin the tip and clean off the tin oxide every 5 to 10 minutes. If the tip has been left at 800 degF, the tin oxide layer will be thicker and harder to dissolve. NEVER set the temperature to 800 degF and leave it. If you turn on your soldering iron and it is set for 800 degF, you know the last person who used it did not know anything about soldering.

Never leave the tip hot for longer than a few minutes if you are not using the soldering iron. The tin oxide layer will grow on the surface, and it will make it more difficult to clean the next time.

14.2 Exp 2: soldering 1206 0 ohm jumpers

Try soldering the 1206 parts to the pads in each column. Use the approach in the video:

1. *Add flux to the pads*
2. *Place the part*
3. *Hold it down with tweezers*
4. *Add a little solder to the tip of the iron*
5. *Touch it to one of the terminals and pads to reflow some solder*
6. *When one pad is tacked down, repeat to the other pad, while touching the solder wire to the heated pad*
7. *Then go back and add solder to the first pad*
8. *Look at the quality of the solder joint under the microscope.*
9. *Try soldering a pad with no solder flux. Make a terrible joint and look at it under the microscope*
10. *Measure the resistance of the solder join with a DMM*
11. *Repeat for each of the pads of the 1206 and each of the three columns.*

So you can understand the significance of solder flux, try soldering a few 1206 parts with NO solder flux. Can you achieve a good solder joint? Do the best you can with no flux. Of course, the solder wire has some solder flux, so it is hard to avoid no solder flux, but there is still a noticeable difference using extra flux and no extra flux. What is the purpose of the flux?

Look at the quality of the solder joint under a microscope. Try taking a picture using the smart phone camera attachment.

To clean up the poor solder joint, add some solder flux and reflow the joint. Even a bad joint can be dramatically improved by adding solder flux.

When you practice soldering parts, the goal is not to master putting a few 0402 parts down, but to master soldering dozens of 1206 parts. When you are assembling a board by hand, which you will be doing, you may have 50 parts on a board. You want 100% perfect parts. It is not about a heroic effort for one 0402 part, it is about routinely soldering 1206 parts.

There are many right ways of manually soldering small parts, some easier than others. You should find a process that works for you using lead free solder.

Remember three important guidelines for manual soldering:

1. *Always keep the soldering iron tinned and free of tin-oxide.*
 2. *Cold solder has an oxide layer on it. Always apply solder flux to dissolve the oxide before you try to melt and reflow solder.*
 3. *Use a temperature hot enough to melt the solder, but low enough to prevent the solder flux from vaporizing too fast. This is usually:*
- *Lead free solder has a melt temperature of about 240 degC or 470 degF. Use a solder iron around 370 degC or 700 degF. Hot air gun could be lower.*
 - *If soldering a part that has a larger thermal mass, consider using a higher temperature soldering iron.*

14.3 Exp 3: solder smaller parts

The other pads on this board are designed for 0805, 0603 and 0402 parts. Try soldering each of these down on your board. What is the smallest size you are comfortable soldering?

1. *Try the 0805 and at least one of the 0603 parts.*
2. *If you are feeling ambitious, try an 0402 part.*
3. *On the 32 lead QFN footprint, practice removing a solder short.*
4. *Add enough flux and solder to the pads so they are shorted together.*
5. *Using the copper wick and solder flux and an extra hot iron, suck up the extra solder to clean the pads.*
6. *Feel free to try any of these exercises without solder flux to see how difficult it is, as long as you repeat the exercise with plenty of solder flux.*

14.4 Exp 4: removing solder bridges

In addition to adding SMD parts to the board, you will also practice removing shorts from a board. On the right side of the board are four patterns for 32 lead QFP parts. First, practice removing solder shorts.

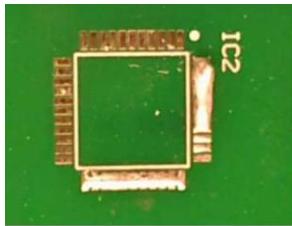
To begin, create some shorts. Add enough solder to the pads to short out a few adjacent pads.

Using the copper solder wick braid, remove the excess solder from the pads to remove the short. Try any method you want to practice this method.

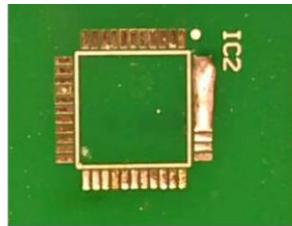
After you have tried your own methods, try this method:

1. Clip off any leading section of the copper braid that has solder on it. Discard this section.
2. Spread the braid to make it as flat as practical.
3. Soak it in solder flux. You cannot have too much solder flux.
4. Use a tip temperature of 800 deg F (a rare case where higher temperature is important due to the large thermal mass to heat up)
5. Add solder flux to the shorted pads.
6. Place the copper braid wick on top of the pads
7. Touch the iron to the top of the braid to reflow and melt the solder. The wick will soak up the solder.
8. Move the wick to a fresh region and repeat, especially adding more solder flux.
9. To make better thermal contact, use the side of the solder iron tip in contact with the copper braid.
10. This method will remove all excess solder from shorted pads. Do not be afraid of making shorts when you assemble parts. You can easily clean them up when you need to.
11. Be very careful: do not pull the braid off the leads or pads unless the solder is being melted by the iron. If you remove the iron before lifting the wick, you will leave the wick soldered to your pads- just add more flux, remelt and lift while melted.
12. Practice

The figure below shows examples of massive solder bridges on some pads and then after they are cleaned up using the method described above.



Before: with solder bridges



After removing solder bridges on bottom pads

14.5 Assemble a QFP part to the board

You will be given a 32 lead QFP (quad, flat package) from your TA. This is a functional micro controller, but costs about \$0.60. You will use this to practice assembling a part to your board.

There are many ways of doing this. This simplest is the following:

1. *Apply a liberal amount of solder flux to the pads*
2. *Place the part on the pads and move it around with tweezers to align it. If you have problems seeing it, use a microscope to align it. Make sure pin 1 is aligned to the reference mark on the board.*
3. *Hold the part down with the tweezers pressing on the top.*
4. *Using a clean soldering iron press down on one of the outer leads to "tack" the lead to the board. This will reflow the solder on the board and the small amount of solder on the iron.*
5. *Repeat this for another lead, on the other side of the package. These two leads will anchor the part in place.*
6. *Using a little solder on the tip of the iron, touch each lead and walk down the leads on one side and then the other. Use solder on the tip, not on the lead. If there is sufficient solder flux on the part, it will help to reflow the solder onto the lead and the pad.*
7. *You cannot have too much solder flux on the leads.*
8. *Repeat the process as you move down each side.*
9. *If you have a short, ignore it for now. Keep applying more solder flux.*
10. *When you are done, if there are any shorts, use the method above to remove them. Apply solder flux, using a copper wick, soaked in solder flux. Apply to the top of the leads, touch with a hot tip to reflow the solder. Remove the copper braid while the soldering iron is touching it.*
11. *If the solder wick is soldered to the leads, do not pull. Reheat and melt the solder and lift the braid while the solder is molten.*
12. *Look at all the leads under a microscope to see if you have any opens, lifted leads or shorts.*
13. *The most common problem when you solder leaded parts to a board is having leads that are not connected. ALWAYS be on the lookout. This means rotate your part under the microscope to check for leads that are not soldered to their pads.*
14. *The most important debug tool is a microscope so you can see lifted leads not touching pads.*

14.6 What you will turn in to complete this lab

You should solder all the 1206 and 0805 parts to your board. You should assemble at least one 0603 part. Try your hand at an 0402 part.

You should have created solder bridges on the 32 pin footprint and cleaned them up.

You should have soldered one QFP to your board with all good solder joints.

You do not have to turn in a lab report for this lab. But, in your optional lab report, include pictures of your assembled board. Try taking close ups with the microscope and your smart phone.

Add a comment about the ease or difficulty of soldering the 1206 parts on each of the three columns and any tricks you used to successfully solder the parts.

Add any comments about removing the solder shorts.

Add any comments about what you discovered about the importance of using solder flux. Your report and pictures will look great in your portfolio.

Be sure to call your TA over for a check off before you finish the lab.

Once you have mastered these technique, you will use them over and over again for every board project.

14.6.1 Grading rubric

There is no lab report for this lab, but remember, a report on what you did and learned will look great in your portfolio.

1 point if checked off by your TA

Chapter 15 Lab 7 Assembly practice: build and characterize hex inverter board

In this lab, you will practice assembling header pins into a circuit board and build your first functional circuit board.

15.1 Purpose of this lab

The purpose of this board is:

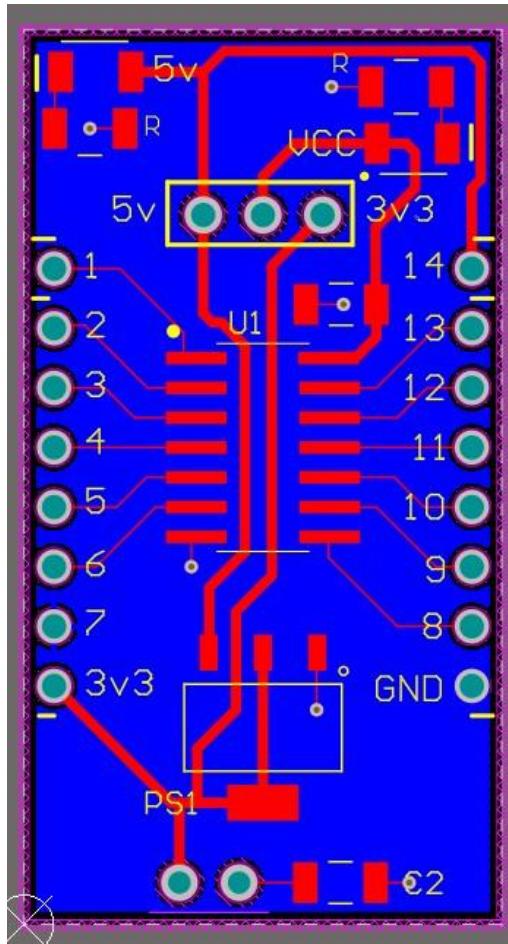
- *To practice assembly of through hole pins*
- *To practice assembly of SMT components*
- *To build and test a functional board.*

15.2 Before you begin this lab

Before you begin this lab, you should have viewed the videos about solder pins to a board, especially [SBW 3-4](#) and [SBW 3-14](#)

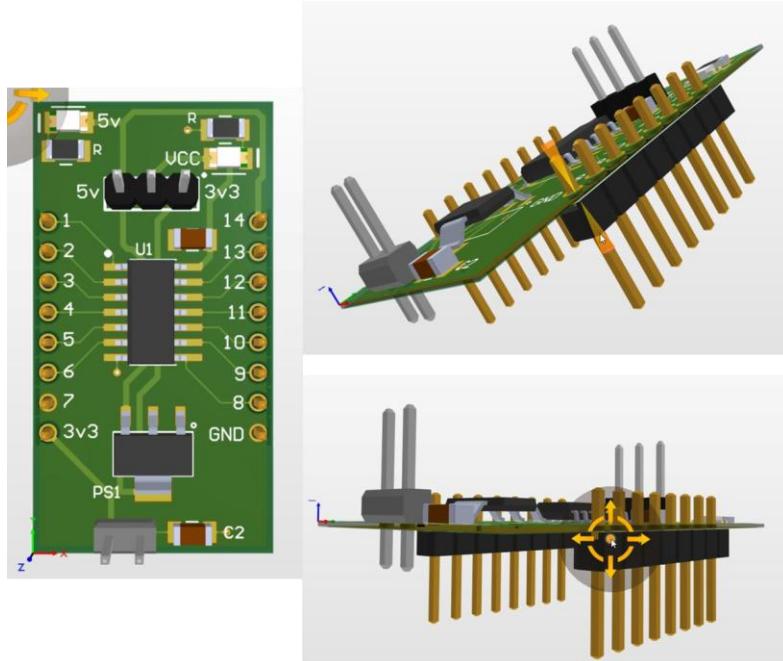
15.3 The board

You will get a hex inverter breakout board and a kit of components. Here is the board layout. It is basically a hex inverter with its pins brought to the edge of the board so the board can be inserted into a solderless breadboard. On the board is a 3.3 V LDO and some capacitors, resistors, switches and LEDs.:



15.4 Assembling your board

After your board is assembled, it should look like this:



Note the different orientations of the sets of header pins. The ones that plug into the solderless breadboard are mounted to the bottom of the board with their long sides pointing down. The two sets of pins used as jumper switches are mounted to the top of the board with their long sides pointing up.

I recommend soldering the headers pins that connect into the solderless breadboard first, so that you can then mount your board in a solderless breadboard which acts as a very stable base.

Special NOTE: There are two gnd pins on this board into which you will insert a header pin. Pin 7 connects to the bottom ground plane and has a thermal relief. However, the lower right pin, labeled as ground, also connects to the bottom ground plane, but DOES NOT have a thermal relief.

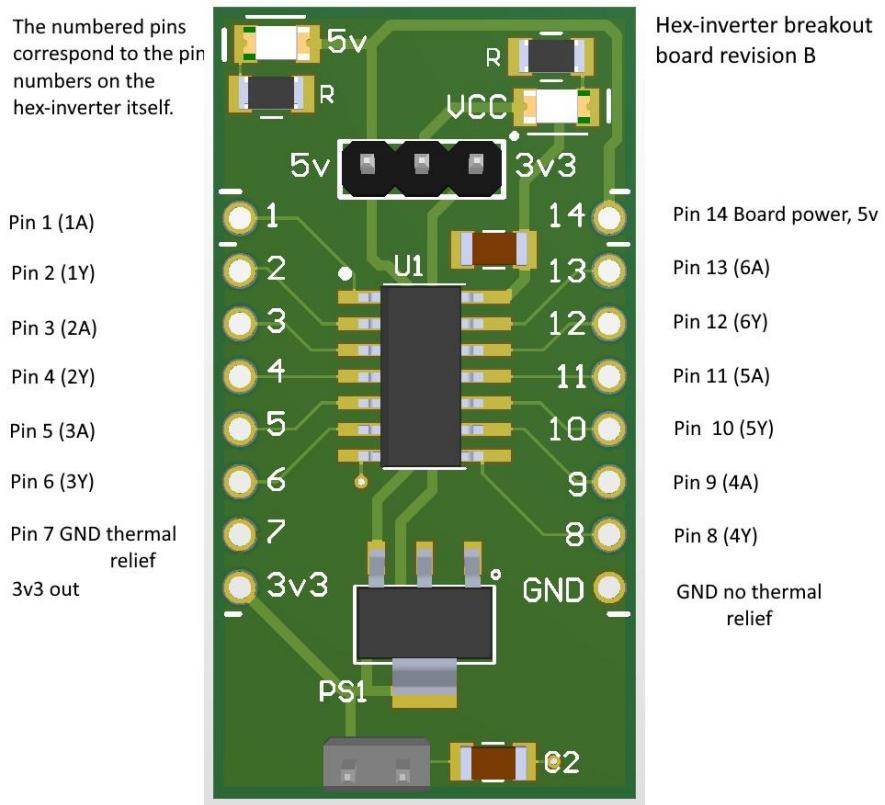
Pay attention to how hard or difficult it is to solder these pins into these two locations.

Here is a simple process to assembly your board:

1. Use your solderless breadboard as a fixture to hold the array of pins aligned, then place your board on top of the pins.
2. You will cut the strip of header pins into rows of 8 pins. Note your hex has 14 pins, which is 7 pins per row. However, the bottom two header pins are extra: the bottom left pin is to bring out 3.3 V from the onboard LDO to the solderless breadboard to test, and the bottom right header pin is an extra ground pin.
3. Apply solder flux to all the pins.

4. Tack one of the pins on each row to hold the pins and board in place.
5. Walk up and down the rows to solder all the pins. Remember, the solder flux is doing all the work.
6. Once the pins are soldered, you can leave your board in the solderless breadboard to assemble your other parts.

On your board, all the parts are not well labeled, so you will have to use your judgement for what goes where. Here is a diagram of what you will end up with:



In the upper part of the board the R components are 1k resistors. Next to them are LEDs. Note the polarity. It is marked on the board. The line is the cathode of the LED (low voltage side). These are used as power indicator LEDs. You can see the resistors are connected to ground. Identify the ground vias to the bottom layer. Now you know the direction of the current and the orientation you need for the LEDs.

If you are ever in doubt of the LED component polarity, look on its bottom and you will see a green bar on the cathode. Or, use a DMM to test for the polarity when it turns on. A DMM on resistance, will turn on most LEDs. Before you assemble the LED on the board, make sure it is oriented in the correct polarity.

Assemble the LDO, Hex inverter chip, decoupling capacitor on the Vcc pin of the hex inverter chip and the filter capacitor on the LDO.

You will be given an AHC hex inverter. If you want to also try one of the fast ones, or one of the slow ones, ask for it and we will get it for your second board.

After all the SMT components are added, then assemble the two sets of header pins that will be used as switches.

VERY IMPORTANT: When you assemble the header pins as switches, their long sides should be pointing up. You will attach shorting flags to these pins. Long side faces up!!

15.5 Bring up and test your functional board

If you think your board is ready to test, do a visual inspection of all the leads and all the components. Touch up any you think need a better solder joint.

Plug your board in a solderless breadboard.

You will supply 5 V using the onboard regulator from your Arduino. Of course, to get 5 V out from your Arduino board you should use the 9 V power supply plugged into your Arduino.

Then, plug the 5 V rail from the header pins in your Arduino board into pin 14 of your breakout board. Be sure to connect ground from your Arduino board to your hex board as well.

With the power select flag- at the top, removed, the LED for the 5 V should light up. With the flag removed, no power will be delivered to your hex chip.

Before you power up your hex inverters, you should connect each of your inputs to 5 V so that their outputs are low.

Then connect the flag to supply 5 V to your hex.

You should measure a 0 V on the output of your inverters. Always use the scope for these measurements.

Measure the 5 V rail, and the outputs. Measure the 3.3 V rail from the LDO that comes out on the bottom left pin labeled 3v3.

Move the select pin to the 3.3 V power rail. This will supply the hex chip with 3.3V.

The select pin switch on the lower end of the board connects the filter capacitor to the output of the LDO. With this switch open, you may see oscillation and noise on the 3.3 V rail. When you add the capacitor, you should see this noise on the 3.3 V rail decrease.

Of course, as soon as you connect the 3.3 V rail to the hex, you connect the decoupling capacitor on pin 14 to the LDO, so this might suppress any oscillations. You can look at the 3.3 V rail when it is not connected to the hex and see if it is oscillating.

15.6 Build a ring oscillator

There are many circuits you can build with your hex inverter. You will use this same chip in the brd 2 design and will see how to connect it up to measure a quiet low and a quiet HIGH. You can send a ramp signal into the hex inverter and measure the input switching threshold to transition from a high to a low or low to high signal. You should be able to figure out how to set this up.

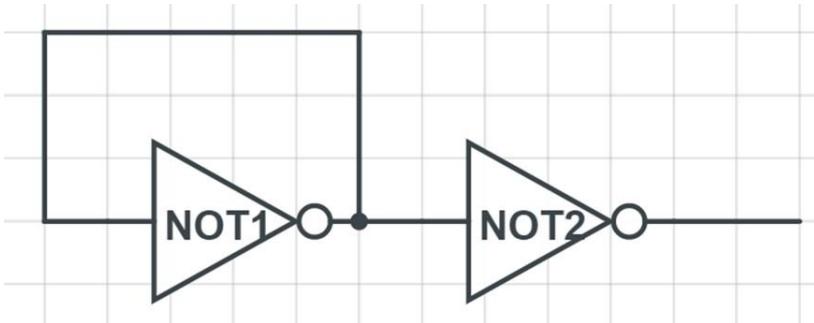
You can use a function generator or a digital pin from an Arduino to switch the input pin of an inverter and measure the rise and fall time of the output signal.

At the very least, you should build a ring oscillator with your hex inverter chip. This is described in detail in Lab 13.

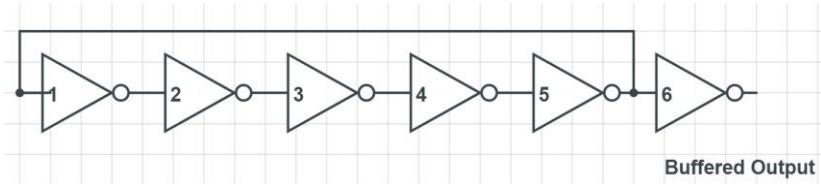
To create a ring oscillator, you will connect the input of inverter 1 to the output of inverter 1. This will cause the inverter to self-oscillate. The frequency is related to the delays between an input voltage turning on and the resulting output voltage switching. This is called the propagation delay.

If the input changes from a high to low, the output will change from a low to a high, delayed by the propagation delay. This change in state in the output will cause the input to change, and the output will then flip, another propagation delay later.

The time to make one cycle is 2 propagation delays. The frequency of oscillation is about $1/(2 \times \text{prop delay})$. If the propagation delay is 5 nsec, the oscillation frequency is about 100 MHz. In practice it is also better to buffer the output of the oscillating inverter with another inverter so that what you measure is the buffered output. This prevents your connection to the output with your scope probe from loading the oscillator and changing its frequency. The circuit to implement this is:



Depending on your IC, the propagation delay may be as short as 1 or 2 nsec, this means you could see a ring oscillator frequency as high as 500 MHz, which is very hard to measure with our 200 MHz bandwidth scopes. To slow it down a little bit, it may be better to connect 5 inverters in series as shown here:



You want to use an odd number of inverters so it self-oscillates. You also want to buffer the output with the last inverter. If you find it difficult to measure the frequency with 1 inverter in a ring oscillator because it is so high a frequency, connect 5 in series. You should measure a frequency of about 20-50 MHz, depending on specific IC you have on your board.

It will also vary if you use a faster or slower part. The difference will be the propagation delay and the rise time of that part.

15.7 Does your board “work”?

The definition of work is:

Do the power lights turn on?

Are you able to select the power supplied to hex using the switch?

Are you able to get the ring oscillator to oscillate.

Does the frequency depend on the power rail voltage?

Are the other outputs at 0 V?

There are other features your board can demonstrate. Afterall, it is a breakout board for a hex inverter. You can use this board to evaluate the properties of a hex inverter. For example, send a square wave into one input pin. What do you see on the output?

You should test your board for functionality based on what you expect to see. If it does not “work” you should investigate why.

You can also use this board for a future lab where you will use a hex inverter in a circuit.

Chapter 16 Lab 8: SBB version of the hex inverter circuit

16.1 Purpose of this lab

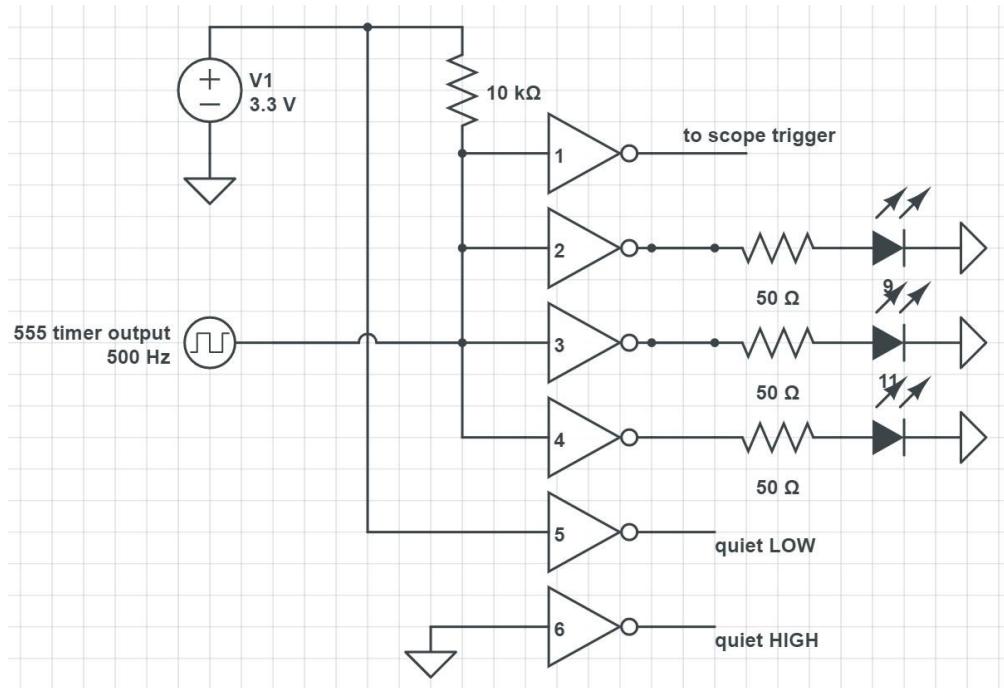
The purpose of this lab is to build another circuit and get familiar with the hex inverter. You will use the 555 timer to create a 500 Hz signal. This will drive the hex inverter. You will measure switching noise on the hex inverter IC.

This is exactly the circuit you will build in your brd 2. This is why you are building it as a solderless breadboard circuit, to get familiar with the circuit and its performance. In board 2, you will build two identical circuits on the same board. One will use good design practices, while the other will use what engineers who don't know about best practices would do. Many boards you will see use poor design practices.

16.2 Exp 1: Build the circuit

Start with the 555 timer chip configured as an astable oscillator with about a 50% duty cycle and 500 Hz frequency. This was the circuit in lab 2. Its purpose is to provide a clock signal which will drive the hex inverter inputs.

The hex inverter you will use is the SN74AHC14 hex inverter. An equivalent part data sheet [can be found here](#). Build the following circuit:



In this case, you can power the hex with 5V. Do not add any decoupling capacitors.

16.3 Exp 2: Measure the switching signals

Using a 10x scope probe, measure the output of the 555 chip in one channel. Trigger the scope on the 555 and measure the output of one inverter with channel 2.

What can you tell about the two signals? What is different? What is the rise and fall time of the inverter?

16.4 Exp 3: measure the quiet LOW and HIGH signals

When the output of an inverter is a low, this means its internal CMOS driver is connecting the on-die ground rail to the output. Measuring a low is like having a connection directly to the ground on the die. Likewise, when the output of an inverter is high, this means the CMOS output drive is connecting the output to the Vcc rail directly. By looking at a HIGH output, you are literally looking at the power rail on the die.

This method of looking at a HIGH or LOW signal is a powerful trick to “sniff” the voltage on the die. This is a technique we will use over and over again. It is described in this article:

<https://www.signalintegrityjournal.com/articles/2790-measuring-only-board-level-power-rail-noise-maybe-misleading>

We will use this in many of the upcoming labs.

Trigger the scope on the output signal of the hex. Measure the noise on the quiet HIGH and LOW pins.

Apply rule #9: what do you expect to see?

How does the noise on the quiet LOW signal depend on where your ground connection is on the scope probe?

How does the noise on the quiet HIGH pin change if you unplug the LEDs?

Add a 1 uF capacitor to the power rail in proximity to the 7414. Is there any change in the quiet HIGH noise?

16.5 Grading rubric: 3 points total

1 point if checked off by your TA

1 point if you have screen shots that are easy to read the information about the waveforms and you have an accurate analysis of the measurements.

1 point if you have a coherent summary of the project.

Remember, this report will look great on your portfolio page.

Chapter 17 Lab 9: measure cross talk between signal-return loops in a special test board

In this lab we are going to explore two different interconnect approaches to see how their radically different geometries will affect the amount of cross talk between an aggressor and victim signal-return path pair.

17.1 So What: Best probing practices for low cross talk

In a previous lab, you saw how the rise time of the aggressor signal affected the amount of switching noise you measured. When the switching noise, either on the power rail or the signal lines, is driven by an $L \frac{dI}{dt}$, a longer rise time signal means a smaller dI/dt and less switching noise.

When the rise time was 1 usec, the switching noise was so small it could hardly be measured. But when the rise time was 5 nsec, the switching noise was huge. This means that generally for signals with 1 usec rise time, switching noise is almost a “who cares.” Interconnects are transparent for these signals.

When you were designing circuits with typical op amps or slow 555 timer chips or audio signal applications, how you design the interconnect and how you probed the circuits did not matter. You could use long floppy interconnect wires and long scope probes. The interconnects had no impact.

But for any rise times that are shorter than about 100 nsec, switching noise may be a big concern. To reduce the cross talk in your interconnects, use short signal-return path leads and short leads in your scope probes.

In this lab we will see that when the rise times are shorter than 100 nsec, how we probe signals will influence the artifacts in measuring the fast edge signals and in how much cross talk we measure. If we are not careful how we set up the scope probes, we run the risk of measuring artifacts due to how we probe.

17.2 Purpose of this lab

You will explore two different geometries to measure the cross talk between one or more aggressor signals simultaneously switching and the noise induced in an adjacent victim signal return path.

In each case, you will also want to follow the interconnect path of the signal and return conductors of the aggressor and the victim loops.

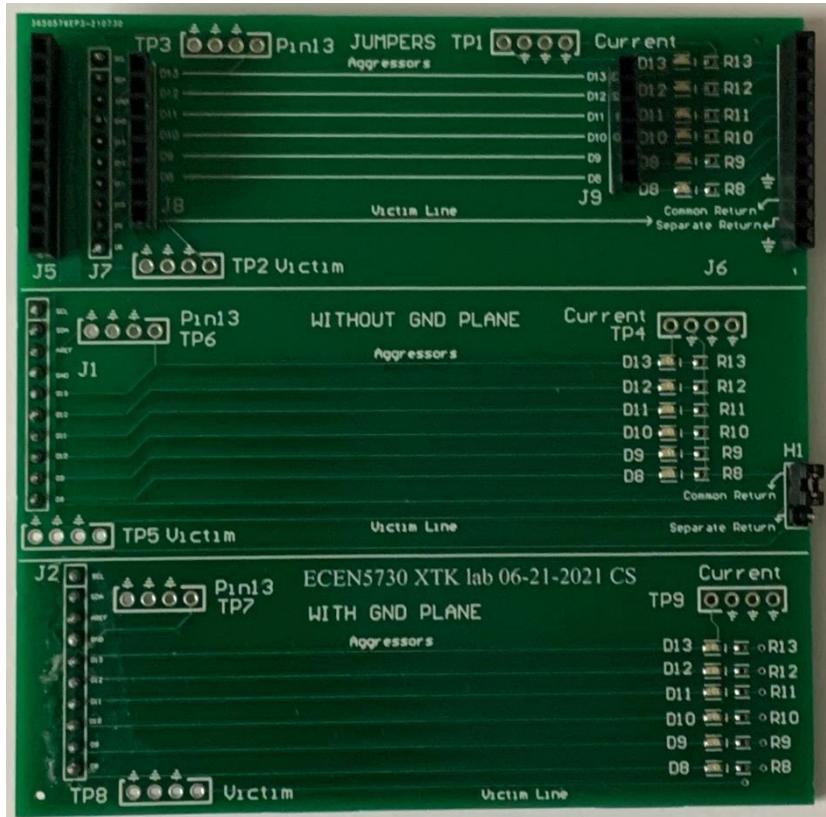
The purpose of this lab is:

1. *To gain practice with the best measurement practices to measure switching noise cross talk by triggering the scope on the aggressor signal.*
2. *To look at different wiring options for signal paths and return paths.*
3. *To learn to distinguish and identify the signal path and the return path of any interconnect*
4. *To compare the timing using a digitalWrite command and a PORTB command*

5. To write the microcode for a pulse train for 1 to 6 I/O that switch simultaneously with different patterns.
6. To measure the victim noise signature.
7. To evaluate how the switching noise scales with the number of simultaneous switching signals
8. To evaluate how the cross talk to the victim line varies as the physical wiring of the aggressor and the victim changes.
9. To see the routing geometry that creates the lowest cross talk.

17.3 What you will need

You will need the scope, two 10x probes with spring ground tips, an Arduino board and the special cross talk board you will get from your TA. Here is the board:



17.4 Prep before you start this lab

You should have read the sections in the textbook about cross talk:

Chapter 4: Electrical properties of interconnects

Chapter 11: Solderless breadboards

Chapter 12: Switching noise and return path routing

17.5 What you will do

There are 2 parts to this lab.

In part 1 you will measure the best-case cross talk, when there is a continuous return path.

In part 2, you will measure the cross talk between two loops built as a PCB traces and see the higher cross talk when the return is routed as another trace and not as a plane. You will also see the impact of using a shared return compared to a separate return path.

The signal source will be an Arduino generating from 1 to 6 output pins switching simultaneously in various patterns.

17.6 Some background

There are two important noise sources that arise because interconnects are not transparent: switching noise in the power and ground distribution path and switching noise as cross talk between aggressor signals and victim signals.

This lab focuses on the switching noise as cross talk between multiple aggressor signals and one victim line.

An aggressor signal-return path is any path that carries a signal that will couple to another path. A victim path is the signal-return path on which we measure the cross talk. If there is also a signal voltage on the victim path, then it is sometimes hard to distinguish what is the signal voltage and what is the cross-talk noise. To make our job of measuring the cross-talk noise easier, we will turn off the signal on the victim path so that any voltage we measure will only be cross talk noise. When the victim line does not have an intentional signal on it, we call this victim line a quiet line.

It's convenient to measure noise on a quiet victim line because nominally, there should be no signal. Any voltage you measure will be noise.

As we will see, the self-aggression noise (power rail noise) and the mutual-aggression noise (cross talk) depend just as much on the return paths as the signal paths.

Get used to thinking NOT signal paths, BUT signal-return paths.

And, most importantly, forget the word "ground". While the return conductor may be connected to the ground net in your circuit, the ground net does NOT act like an infinite sink of current with all ground conductors connected together, distributing the return currents equally. Start getting used to referring to the ground conductor as the return conductor.

We will engineer the signal and return path of multiple aggressors and the signal-return path of the victim. Then we will drive the aggressor traces with a few different signals. The current in the aggressor signal return path loops will create magnetic fields around their loops. These field lines will extend around the victim loop as well.

When the magnetic field lines that pass through the victim loop are constant, there is no induced voltage on the victim loop. However, when the magnetic field lines change, they induce a voltage on the victim loop.

During what part of the signal is there a changing magnetic field? Of course, it is only when the signal is switching on or off. This is when the noise appears on the victim line.

To see the noise on the victim line and have confidence it is from the switching noise of the aggressor, **always** trigger the scope on the edge of the aggressor signal. The voltage on the victim that is synchronous with the edge on the aggressor is switching noise.

The switching noise only lasts for the rise or fall time of the aggressor signal. If multiple aggressor signals switch simultaneously, each of their switching noise on the victim line will add and the noise on the victim line from multiple, simultaneously switching signals will increase.

But, if the multiple aggressors are not switching with their edges exactly simultaneous, but shifted in time, while they will each induce switching noise, it will be time shifted on the victim line and the peak switching noise will be less.

The voltage noise induced on the victim loop is related to:

$$V_{victim} = M * n * \frac{dI_{aggressor}}{dt}$$

Where

M = the loop mutual inductance between the aggressor loop and the victim loop. This is only about the geometry of the loops- what we can engineer when routing traces and return paths.

n = the number of simultaneously switching aggressor signals

dI_{aggressor} = the current change in each aggressor signal dt =

the rise or fall time of the aggressor signal.

The crosstalk noise is driven by the dI/dt in the aggressor loop. The large current changes occur at the voltage edges when the signal switches voltage levels. This is why we call inductively generated noise, **switching noise**.

The larger the loop mutual inductance between the aggressor loop and the victim loop, the larger the inductive cross talk. If we want to reduce the crosstalk, we have to reduce the loop mutual inductance between the aggressor and victim loop. This is where interconnect design comes in.

There are five different physical design features in the interconnect we implement which will reduce the loop mutual inductance between aggressors and victim loops. In this lab, we will use discrete traces on a board and a plane to route return paths to create different geometry configurations.

An Arduino Uno will act as the signal source. We will drive multiple digital I/O as aggressors. Pay attention to the following best design practices to reduce switching noise. Look to see how these design principles are illustrated in each measurement you do. In all of your board designs, even if the switching noise is not large enough to cause problems, it is still a good habit to engineer interconnects to reduce this problem. It is absolutely guaranteed that one of the next boards you design will be sensitive to switching noise and these design principles will be critical to the success of a future board.

1. *Do not share return paths between the signal-return loops of the aggressor and victim. Use a separate return conductor.*
2. *Reduce the number of signals switching simultaneously which have mutual inductance to the victim loop.*
3. *Reduce the self-loop inductance of the signal-return loop of a path by bringing the signal and return path conductors as close together as practical.*
4. *Reduce the loop-mutual inductance between the aggressor signal-return paths and victim signal-return paths by keeping the two loops far apart.*
5. *On a PCB, use a continuous plane under the signal path to route the return currents to reduce the loop self-inductance of the signal-return paths and to reduce the loop mutual-inductance between aggressor and victim loops.*

17.7 Part 1: Cross talk with a continuous return plane

In Part 1 of the lab, use the bottom third section of the board. Follow the signal paths from the pin connections to the LEDs and the series resistors. The return connections are through the via at the end of the trace to the bottom plane. The far end of the victim line is shorted to the bottom plane as well.

You will drive the aggressor traces with different patterns of simultaneous switching signals. Measure the voltage across the 47 ohm resistor at the end of pin 13's trace. From this voltage, you can calculate the current that switches. Notice that the rising edge of the current turning on is much slower than the falling edge. The output transistor of the Arduino turns on with a longer rise time than it shuts off. The p-channel MOSFET on the output that turns the output on is slower than the n-channel transistor that turns the output low.

This means the dI/dt will be larger on the falling edge and the inductively generated cross talk noise will be larger on the falling edge.

17.7.1 Exp 1: Writing simultaneous digital outputs from an Arduino

In this section we want to generate simultaneously switching outputs from the Arduino. If we use the `digitalWrite` command, the outputs do not switch simultaneously. Do that experiment. Write the code to switch pin 13 and then pin 11 using a `digitalWrite` command. The outputs do not switch at the same time.

This is the value of using a powerful measurement instrument like the 200 MHz bandwidth scope. We can measure the small delay between channels switching sequentially rather than simultaneously. We can't use the `digitalWrite` command for this crosstalk experiment if we want the edges of multiple I/O to switch simultaneously. The `digitalWrite` command is NOT simultaneous.

Instead, we will use the [PORTB command](#).

My sketch to trigger pin 13 and pin 12 simultaneously is:

```
void setup() {  
pinMode(13, OUTPUT);
```

```

pinMode(12, OUTPUT);
pinMode(11, OUTPUT);
pinMode(10, OUTPUT);
pinMode(9, OUTPUT);
pinMode(8, OUTPUT);

} void loop() {
PORTB=B00111111;
PORTB=B00000000;

}

```

Using the PORTB command, the byte value identifies the state written to pins, 15, 14, 13, 12, 11, 10, 9, 8. In my sketch, I am turning on pins 13, 12, 11, 10, 9, 8, then turning them off. You should measure the voltage on pin 13 and pin 11, for example to convince yourself of this.

As written, the pins will switch on for 1 clock cycle then switch off. This is only 63 nsec. If you want the pins to be on longer, so you can see the noise drop to zero after the edge, just add multiple PORTB commands. Each will take 1 clock cycle to execute.

We will change the code to selectively switch any or all pins simultaneously. You should create 3 patterns:

1. *All pins switching simultaneously.*
2. *An increasing number of pins switching simultaneously, starting either with pin 13 or with pin 8.*
3. *One specific pin at a time switching, then turning off and another pin turning on, then switching off.*

For example, keeping the pins on for 4 clock cycles would be:

```

PORTB=B00111111;
PORTB=B00111111;
PORTB=B00111111;
PORTB=B00111111;

PORTB=B00000000;

```

17.7.2 Exp 2: measuring the voltage noise on the victim trace with a continuous return plane on the bottom layer

In this experiment, you will still be using the bottom quadrant of the special test board. The return path is a continuous plane. The victim line is shorted to the return plane at the end of the line.

Each of the signal pins pass through an LED and a 47 ohm resistor which is shorted to the plane on the bottom layer.

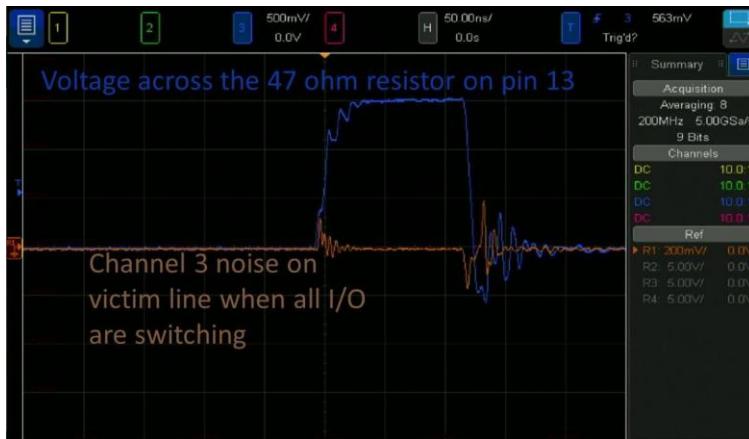
Use the voltage across the 47 ohm resistor at the end of pin 13 to trigger the scope. Measure the voltage noise on the victim trace. All measurements should be done using the spring ground tips. (why?) Save this noise on the victim line as your reference. You will compare this noise with the other noise you measure in other configurations. It is not zero. Even with the wide, continuous return plane, there is still cross talk between the aggressor signal traces and the victim trace due to the loop mutual inductances. Some of this noise may also be from the connector to the Arduino board and the fact that there is only 1 ground pin on the digital side of the Arduino pins.

Look at the noise with all the I/O switching. If you have time, you can evaluate which signal line induces the most noise on the victim. Which line do you expect?

The figure below shows the measured current through the resistor and the noise I measured on my victim line in the lower quadrant of the board.

When you use a reference channel on the scope you can see the scale on the lower right region of the screen under Ref. You can adjust this scale using the knobs on the right central region of the scope face above the Serial button. When you are comparing this base line reference noise to other examples, it is useful to adjust the scale of the reference signal to match the scale of the real time measurement.

You should measure the noise on the victim line with different patterns of aggressor signals. Below is the measured cross talk on channel 1, the victim pin, when I turned off and on just pin 13, then turned off and on just pin 12, all the way to pin 8. It is clear that all the cross talk comes from pin 8, though there is some cross talk from the other pins as well.



Why is there any cross talk from the trace on pin 13 to the victim line? They are very far away. The answer is the cross talk is probably coming from other regions of the board, like the connector and the 328 uC itself. It is probably not coming from just the proximity of the traces on this test board.

The noise on the victim trace when only pin 8 switches is larger than when all the pins switch. Why is this? It is probably because the I/O has more current, switching faster when only one pin switches than when all 6 I/O pins switch.

Save this cross talk as the reference cross talk to which the other cross talk will be measured.

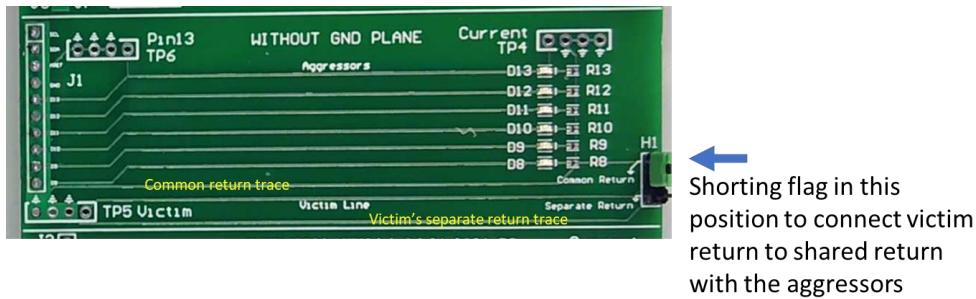
17.8 Part 2: Cross talk with no plane, but an adjacent return trace

Part 2 of this lab uses the middle section of the board. Again, all the signal pins are connected to traces that span the length of the board and connect to LEDs and resistors which are all bussed together through a common return trace. There is no plane on the bottom of the board.

Trace the signal paths and their return on this board. Notice that all the signal traces, have a common return. It is labeled in this figure below. Trace the path of the victim line. Using the shorting flag on the bottom right section of this board, you can select two different paths for the victim line's return path.

You can select the return path of the victim line to be the same line as the aggressors or its own separate return path, so it does not share the return path with the other switching signals.

You can drive these aggressor traces with the same signals as in the last section. This section of the board is shown here:



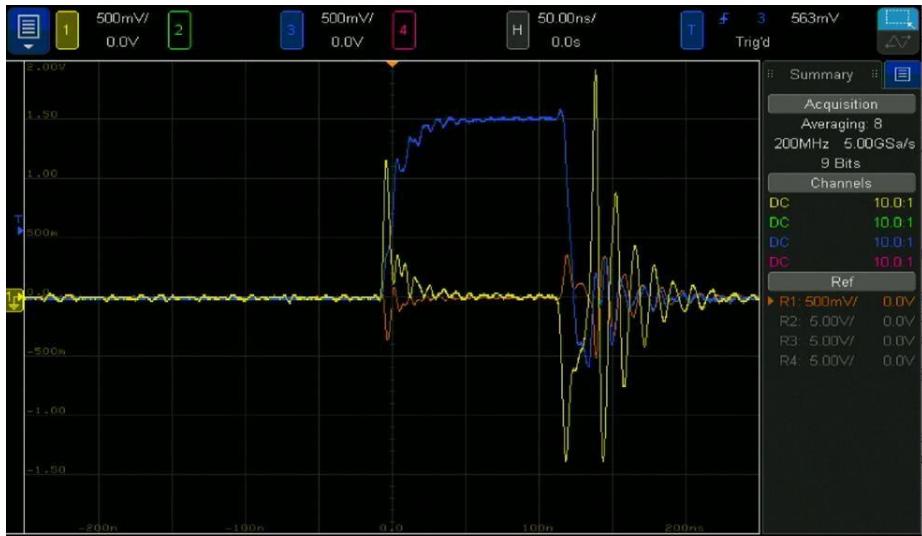
17.8.1 Exp 3: return of victim is common or separate with aggressor return

The victim trace has two options for its return path. It can connect to the same, common return as the aggressor signals. This means the ground bounce noise on the common return trace from the aggressors switching is also part of the signal-return path of the victim trace. This will give a very high cross talk.

To select the return path of the victim trace, use the position of the shorting flag. When the flag is in the upper position, the victim's return path is the shared common path. When the shoring flag position is the bottom position, the victim traces uses its own return trace. The aggressors are still using their shared conductor. It does not change. Be sure to trace these connections.

When the return path for the victim is selected as a separate return, there is still cross talk on the victim trace, but it is dramatically reduced.

The difference between the noise on the victim line with its own return or shared return is shown in the figure below. The yellow trace is the noise on the victim line with a common return and all I/O switching simultaneously. The orange trace is with a separate return.



Measure the cross talk on the victim trace with all the I/O switching and with the common return or the separate return. From which trace is the cross talk coming from? Is it coming from all of them equally or is one trace dominating the cross talk? How can you use the pattern of switching traces to determine which traces are generating most of the switching noise?

What do you conclude about routing signals and return paths for lowest cross talk?

17.9 Lab report and grading rubric

This is one of the most important labs in this course because it demonstrates why you should use a plane to reduce the cross talk and why you should never share a return path between an aggressor line and a victim line. Separate returns, like in a plane, will always result in lowest noise.

Remember, you should write your lab report for a hiring manager to read. This lab explaining how to measure switching noise cross talk and how it is affected by the number of I/O switching simultaneously and how the routing geometry affects it, is gold to hiring managers.

Your lab report should include a brief explanation of the two interconnect patterns on the board, with a continuous return plane and a separate return trace.

You should show the difference in cross talk between the three cases:

1. *Continuous return plane*
2. *Separate return trace that is not shared*
3. *Separate return trace is a shared return trace.*

You should include at least when all the signals are switching simultaneously at a minimum. You can also show what happens to the noise when the switching pattern changes, and your interpretation of the results.

Use appropriate pictures and scope traces to compare the noise. Make sure the scope's scale is appropriate to present the message you want to show.

Your report is worth 3 points:

1 point for check off by your TA

1 point if you show measurements using best measurement practices and appropriate scales

1 point if you have a clear interpretation of the results and can answer the so what? question.

Chapter 18 Lab 27: Best measurement practices for high-speed signals to reduce artifacts

We typically refer to any signal with a rise time less than about 100 nsec as a high-speed signal because the interconnects may not be transparent, and their physical design may influence noise in the circuit. The most common types of noise are all driven by a dI/dt through an inductance. When the rise time is longer than 100 nsec, the noise generated may be so small as to be difficult to measure.

When your circuit has high speed signals, take special precaution when measuring the signals to avoid artifacts from the probe you use to do the measurement.

In this special lab you will explore how long leads in the scope probes will screw up signals and contribute to much higher probe to probe cross talk. This is an important artifact to watch out for in all your measurements. It is also why all the test points are designed to us with the 10x scope probe and the small ground spring tip.

18.1 Exp 1: set up the Arduino as an aggressor source

In the first part of this lab, you will explore how the way you probe influences the measured signal. We will use an Arduino as the signal source. The rise time from an Arduino digital I/O has a rise time of about 3-5 nsec, depending on the generation of the microcontroller.

Write the code to switch pin 13 with the digitalWrite command with a simple 50% duty cycle, 500 Hz signal. Set pins 12 thru 8 as victim lines, with their outputs set LOW.

You should write your own code. Here is my sketch:

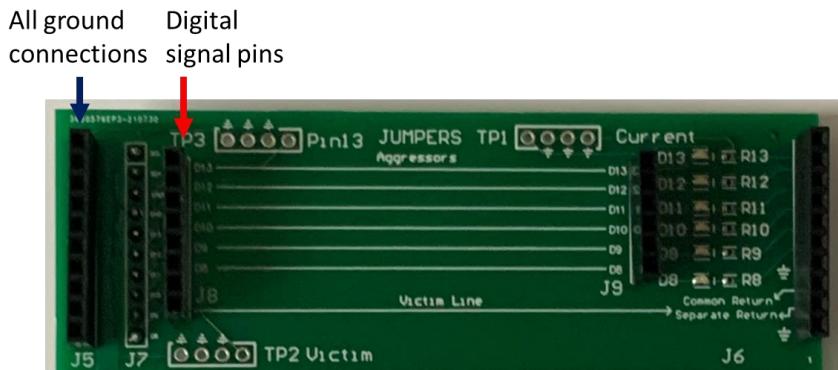
```
void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(11, LOW);
  digitalWrite(10, LOW);
  digitalWrite(9, LOW);
  digitalWrite(8, LOW);
  delay (1);
  digitalWrite(13, LOW);
  delay (1);
}
```

18.1.1 Exp 2: measuring the Arduino digital signal with long leads on the 10x probe

We will probe the signal on pin 13 and on pin 11 with a 10x scope probe. Unfortunately, there is only one ground pin, next to pin 13, so it is difficult to connect multiple 10x probes to this ground pin. Here is where the breakout cross talk board you receive from your TA comes in.

On the bottom of your board are pins that are inserted into the 10 holes of the Arduino's digital pins. For this first exercise, use the top section of the board. Plug it into the upper digital pin header on the upper right side of Arduino board.

On the top of the board, there is now a header socket that breaks out the one ground to multiple ground connections and there are holes for each digital pin. Each digital pin has an adjacent return connection. This is shown in the figure below:



Pin 13 is switching. The other pins are low and should not have any voltage on them. Measure the voltage on pin 13 with a 10x probe. Remember to follow the best practices using a 10x probe:

Make sure the probe is set for 10x

Make sure the scope is set for 10x attenuation

Use a color code band on the 10x probe to match the color of the trace on the scope Check the compensation of the 10x probe

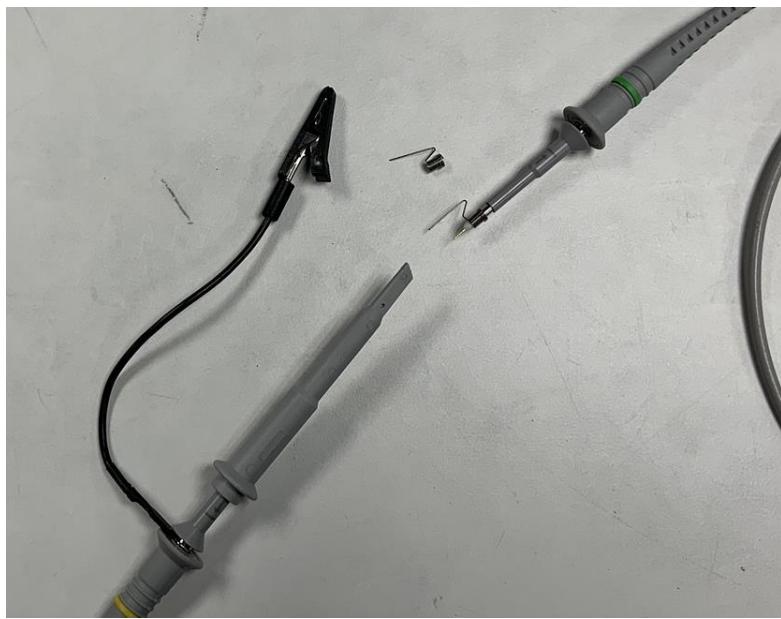
Use two long jumper wires to connect the 10x probe to the pin 13 and a ground connection. This will be the worst-case configuration. Measure the rising edge of the pin 13 signal. What is the rise time, what is the shape of the edge?

How much of this signal is the actual, real signal on pin 13, and how much is artifact due to the measurement system? Remember, you have a large loop inductance at the tip and you have a 10 pF capacitor at the probe tip.

Just from one measurement, it is hard to know what is real and what is artifact. If you did not know about the distortion from the tip inductance, you would think the signal from the Arduino pin has some ringing noise on it.

18.2 Exp 3: measure the Arduino digital pin with the shortest tip loop inductance possible with the spring ground tip

Next, we will make the inductance of the probe tip as small as possible. This is where the small spiral spring ground tips in the probe bag come in. Pull off the 10x probe cap of the 10x probe tip, sometimes called the "hat" exposing the sharp tip of the probe. Screw the spring probe tip onto the scope probe, as shown in the figure below.



Now you have the shortest practical signal-return loop on the probe tip.

With a little effort, you can insert the signal tip into the pin 13 hole and the ground tip into one of the ground holes.

Compare the measured signal on 13 with the long floppy jumper wires and the short loop with the spring ground clip. When the tip loop inductance is reduced, we reduce the measurement artifacts.

What is the rise and fall time of the Arduino signal now? How much ringing is there on the pin 13 signal? Did the signal from the pin 13 really change when you changed the probe tip? You didn't do anything to pin 13. How you probed the signal affected the signal. This is the artifact.

Here is my measurement in the figure below. The yellow trace is with the spring tip, the orange trace is with the long floppy wires.



18.3 Exp 4: why the 1x setting should not be used for signals with a rise time longer than a few microseconds.

Now that you know how the signal from pin 13 looks in the best case with a small inductance in the probe tip, set the 10x probe to the 1x setting. Be sure to also change the attenuation setting on the scope screen to 1x.

What is the rise time of the signal from pin 13 you measured? Did you change the signal from pin 13? Did you change the inductance of the probe tip? What is the actual rise time from pin 13? The longer rise time with the 1x setting is another artifact which has nothing to do with tip inductance.

This is an important observation. Using the 1x probe setting, you can never measure a signal with a rise time shorter than 100 nsec. If your signals are all 1 usec rise time or longer, using the 1x setting does not introduce an artifact. But, if you are measuring faster edges, never use the 1x setting.

Why does the 1x probe setting increase the rise time of the measured signal? Hint, this is reviewed in the skill building lecture about scopes on Never use the 1x setting. Watch this video before you ask me or your TA to explain it.

18.4 Exp 5: Impact of tip loop-inductance on probe-to-probe loop cross talk

Next, we look at the impact on probe-to-probe cross talk from the geometry of the tip. Put the probe hat back on and make sure it is set for 10x. Connect a second probe to the scope.

Using long, floppy jumper wires from the tip of each 10x probe, connect channel 1 to the pin 13 output and channel 2 of the scope to the other 10x probe. You should see the distorted signal on pin 13 in channel 1. It is distorted due to the large tip loop inductance ringing with the 10 pF of input capacitance built inside the 10x probe tip. Move the floppy wires connecting to pin 13 around and you will see the ringing noise of the pin 13 measurement change. Is the signal from pin 13 really changing?

Before you connect the second probe into pin 11, which is a LOW signal, connect both the signal and the return of the second probe into the ground pins, so they are shorted together to ground.

Since the probe tip is shorted to ground, you would expect to see no voltage on the second probe. What do you actually see that is synchronous with the pin 13 signal switching? How will you trigger the scope in order to see the noise on channel 2 synchronous with the switching on pin 13?

Where does this signal come from? Since it should be 0 V, all the signal on channel 2 you are measuring is an artifact from your measuring method.

Next, connect the second probe's signal wire into the pin 11 output. What do you expect to see on pin 11? If you write your code correctly, there should be a low signal on pin 11, which should be 0 V, just like you expected to see when the tips were grounded.

While the signal on pin 11 is mostly a low, it has a lot of synchronous noise on it. You see a lot of switching noise on pin 11. Some of this is real, but most of it is due to the mutual inductance of the loops in your 10x probe. Move the loops around and you will see how the loop geometry of your probes influences the crosstalk noise. This is a measurement artifact.

18.5 Exp 6: probe to probe cross talk with ground spring tips

Remove the caps from each probe and add the spring ground tip to your probes. Carefully, insert the probe 1 into pin 13 and its ground into the ground hole. Now you have the lowest loop inductance in each tip and the lowest loop mutual inductance between probe tips.

You are going to repeat the earlier measurement and look at the noise on probe 2 when the probe's signal and return tips are literally shorted together into the same ground pins. Insert the probe 2 into adjacent ground holes in the board. You may have to hold the probe to make a good connection between the center pin and the ground spring wire.

How large is the noise you see on probe 2 with short leads compared to when you used long floppy leads?

Using the second probe, insert it into the nearest signal pin to which it will fit and the adjacent ground pin. This is probably pin 11.

What is the signal quality on the pin 13 signal and what is the switching noise on the other adjacent victim line?

This is the residual switching noise of the board and the probe fixture. It is the best we can do given the design constraints. Most of this noise you see is probably real. It arises from mutual inductance in the uC chip,

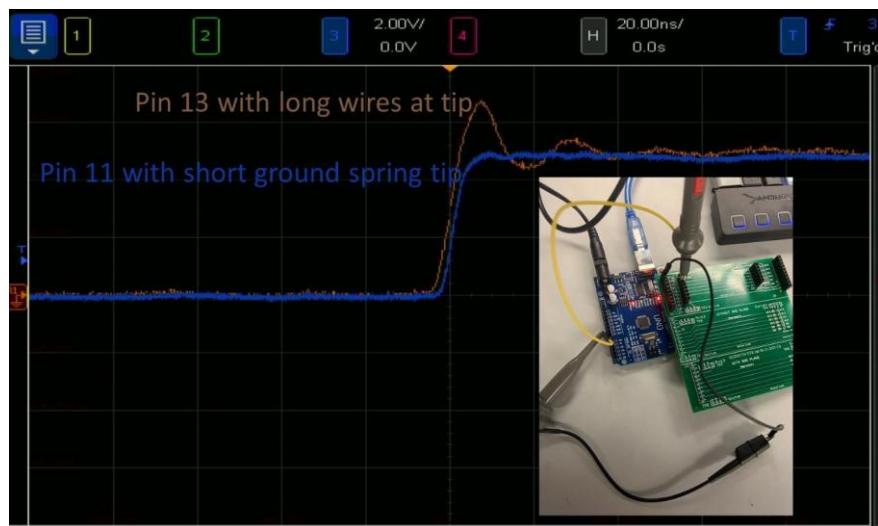
From this exercise, what do you conclude about the best measurement practices to use when measuring highspeed signals?

If your signals had a 100 nsec rise time, instead of their roughly 5 nsec rise time, this measurement artifact would be dramatically reduced. But if your signals have a few nsec rise times, this artifact swamps the signals.

Going forward, be aware of what the rise time of your signals are and the best measurement practices you should use to reduce this sort of artifact.

It should be clear now why the test points on your board are designed the way they are, so you can insert the 10x probe with the spring tips. This will give the best quality measurements given the probe design constraints when rise times are short.

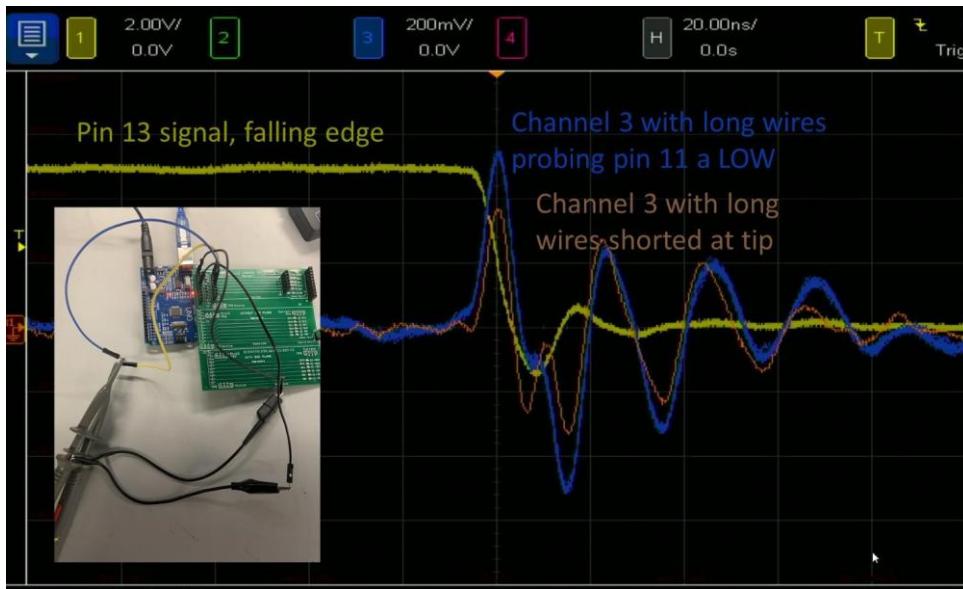
Here are my examples when I did this measurement. Below is the signal from pin 13 with long floppy wires on the tip and the signal from pin 11 with the small spring ground tip when pin 11 was also switching. All the ringing and overshoot in the pin 13 signal is artifact.



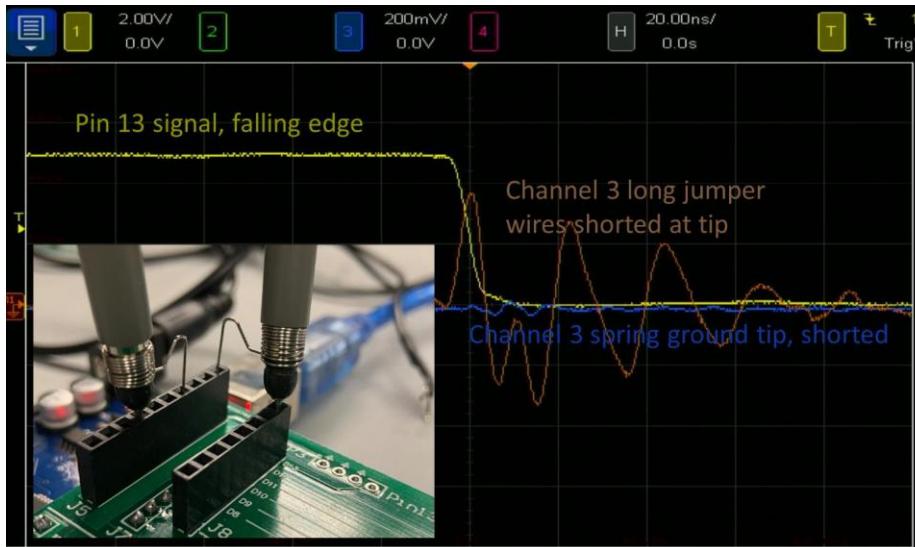
This measurement suggests, when probing high speed signals, use short wires at the tip, preferably the pin and spring ground tip.

When only pin 13 is switching, with the scope triggered on pin 13, the cross-talk noise in the other probe, looking at a quiet low pin, is almost 1 V peak to peak. When the quiet line is connected to the ground pin, so we have a short at the tip of the probes for channel 3, the cross-talk noise is reduced to only 700 mV peak to

peak. This is just cross talk in the probe tips due to the large loop inductances of the tip. This is shown in the figure below.



If you use short loop spring ground tips, the noise picked up between the probes can be dramatically reduced compared to using long wires to make the connection to the pin under test. In the figure below, the connection to pin 13 is with a spring ground tip. The brown trace is the ends of the long floppy wires shorted to a ground pin. The blue trace is the same shorted condition, but with the short spring ground tips. The dramatic reduction in noise on the shorted pins of channel 3 is due to the reduce loop mutual-inductances between the channel 1 and channel 3 probe tips.



The lesson is use spring ground tips and NOT long floppy wires when you are measuring high speed signals. Otherwise you may be sensitive to the signal distortion and cross talk in the probes, artifacts of the measurement.

18.5.1 Lab report and rubric

It is not necessary to turn in a lab report for this lab. These are best measurement practices you will use in all your measurements going forward.

However, a report about these measurement practices on your portfolio page would look great to a hiring manager.

Chapter 19 Brd 1: bring up, test and evaluate

19.1 Assembly practice of your Brd 1 practice board

You will be given 5 bare boards of your brd 1 and a kit of parts. If you need more parts, just ask. Note: the resistors, LEDs and capacitors are all 1206. If you designed for other size parts, the technical term for this situation is, "you're screwed." We do not have any other size parts in inventory. This is why you must follow the directions for what to use in your design.

If you did not get your board back, ask your TA why. If you turned your design in late, you should have an explanation for why.

Use your POR description of what you expect your board to do as your criteria for “working”.

What are the features your board 1 should demonstrate? What is your test plan?

Be sure to take pictures of your sketch, your schematic, your layout, your bare board and your assembled board with lights on. This progression will look great in your final report.

Your final report is due on the following Monday. It counts as a midterm and is worth 10 points. (also 10% of your grade).

The grading rubric for your report is in the brd 1 section of the lab manual, chapter 8.

The most important section of your report is lessons learned.

Chapter 20 Brd 2: Final CDR

Before coming to class, you should have completed the POR, the schematic and most of the layout.

In class, a few volunteers will be selected to present their layout to the class. While they present their layout, the rest of the class will critically listen and offer recommendations to the designer.

Be on the lookout for:

1. *Any possible hard errors which will prevent the correct operation of the board*
2. *Any soft errors which will increase the noise or the risk of the board not meeting spec*
3. *Any good design features which might be good features to include in everyone else's designs.*

Use the check list in the lab manual and in the textbook to go through the list of possible, or commonly occurring errors to verify none of these are present in the design.

After a few layouts have been reviewed, you will have time to complete your layout in class.

Before you leave, you should review your layout with your TA to have them check off your design.

It is not necessary your design be completed by the end of class time, but it should be far enough along so that any major issues can be identified.

Before you post your three design files on canvas, you must:

1. *Review your schematic and layout against the checklists provided in the lab manual and textbook*

2. *Have another student review your schematic and layout*
3. *Have your TA check off your layout*
4. *Post your design files to the JLC website and complete the DFM and get an email acknowledgement back from them that your design is accepted and ready for ordering. This may take as much as 2 hours to get an email back.*
5. *If you received any error messages from the DFM check, you have corrected them, and resubmitted your design for DFM check.*

Design files are due by 9 am on Thursday morning. This is when they will be ordered. If you miss this deadline, your board will not be ordered. You will have to wait for the next week to have your board ordered.

Turning in your design files counts as 3 points. The grading rubric is:

- 1 point for sign off by your TA in class.*
- 1 point if your design files were turned in on time*
- 1 point if your design was accepted by the vendor for production*

If your design file is rejected by JLC, you may only end up with the 1 point for TA check off for the design assignment. This will only happen if you fail to go through the online DFM check on the JLC web site.

If your board fails the DFM check when the TA places the order for your board, it will slow up the delivery of ALL the boards and may affect the schedule for the rest of the students. DO NOT LET THIS HAPPEN FOR YOUR BOARD.

If your board is not ordered with the other student boards, you are still responsible for getting your design completed. Submit it by the next week.

Chapter 21 **BOARD 3: A Golden Arduino PCB**

This board will be a complete, start to finish project to design a “Golden Arduino” board.

We start with the absolute minimum features we need to be able to use this board as an Arduino, to accept uploaded code from a USB port, run the Arduino IDE and be fully compatible with most Arduino Uno R3 shields.

Graduate students are required to add additional features to their board as described below.

21.1 Purpose of this lab

The purpose of this lab is to gain practice with the entire prototype design flow, including all seven steps. You will have a chance to practice reading datasheets to get useful information, and to realize there is ambiguity in datasheets. Consider adding features in your design to provide options to evaluate design features.

This is a prototype. Be sure to add: test points, indicator LEDs and isolation switches, as needed.

In the first week of this board design, you will start the POR, identify some of the special, non-commodity parts you will need, and build some prototype solderless breadboard circuits to test out some new circuit elements.

As with all of your boards, use best design practices for routing signals to reduce cross talk (ground bounce) and best design practices to reduce power rail switching noise from I/Os switching.

The commercial Redboard Arduino Uno board in your kit works. It is connected correctly. This means that it meets the ***design for connectivity*** requirements.

But, inspect the board carefully. You will find there are many things you can improve based on what you know about cross talk control and power delivery noise control, bring up and test.

In this board design assignment, you will design your own “Golden Arduino” which meets the same connectivity specs, but has features for better noise control, assembly, test and bring up.

You will walk through each of the seven steps in the board design process.

21.2 The POR

In the POR you should articulate what the purpose of your board is, what it means to “work” and any special features you expect to implement. You will not turn in your POR, but include it in your final report. Your final report for brd 3 is equivalent to a midterm for this course.

In the POR, you will sketch out the board design and the risk reduction steps. As a starting place, you should check out some reference designs for Arduino Uno boards, but remember, once you start your schematic, it becomes your design. Do not use a feature from a reference design in your design unless you take ownership of it.

When you get your board back, you will demonstrate the following features:

- *Boot load your Atmega 328 to turn it into an Arduino*
- *Run the Arduino IDE on your board and any standard sketch*
- *You must use the same header pin footprint for the pins so any shield will fit on your board*
- *Using a special switching noise shield we will give you, you will measure the noise on a commercial board and on your board under identical conditions. Your noise should be 20% to 50% of the noise on the commercial board.*
- *You will measure the near field emissions from your board, compared to an identical commercial version and find your near field emissions are << 10% that of the commercial Arduino board.*

As a stretch goal, add test points to the digital buses and sniff the digital signals with a scope. For example, add a test point to the D+ and to the D- lines and to the TX and RX of the UART. This way, you will be able to see the actual bus traffic on these lines.

And, as a stretch goal, add the series resistor for current sensing of the power rail and measure the in-rush current and the steady state current for the board. Note that to measure this voltage, you will need a differential voltmeter. You cannot easily measure this voltage with a single-ended scope. You could use 2, single-ended probes and take the difference using the scope match function, or the Digilent AD2 is perfect to measure the inrush current because its inputs are true differential.

If you do no other report in your portfolio to show prospective employers, you should post the report on your Arduino board. You will demonstrate that just by using the design principles you have mastered, you have reduced the noise on your board by as much as an order of magnitude over commercial versions of your board. If they hire you, you can do the same for their products. What better way of selling your skills could you come up with?

21.2.1 Special features and components to add

For brd 3, all the parts you will use should already be in the integrated libraries.

Some of the features you will use in your design are:

1. *An Atmega 328 microcontroller*
2. *A CH340g USB to UART interface chip*
3. *A 16 MHz resonator for generating a clock. A 12 MHz resonator for the CH340g.*
4. *Appropriate decoupling capacitors*
5. *A connector for the SPI and boot loading pins*
6. *A TVS chip to protect the data pins from ESD*
7. *Power from the power plug or the USB connector, but not both at the same time*
8. *A reset switch with a debounce capacitor*

9. A 3.3 V LDO, not used by a component on the board, but available on one of the header pins.
10. Header sockets that match the location of the standard Arduino board so that you can plug a shield into your Uno board
11. Maximum board size 3.9 inches x 3.9 inches
12. Add a second row of header pins 300 mils center to center spaced on the outside of your digital I/O pins all connected to the ground plane. These will be used to demonstrate the lower switching noise when connections are made off your board.
13. Add a 22 uF decoupling capacitor close to the header pin locations on your board for the 5 V and 3.3 V rails. This will decouple connections from your golden Arduino board to any shields you will add.
14. Consider adding an isolation switch in the power path to your CH340g. (NOT between the VCC pin and decoupling capacitor!). This way you can turn the CH340g off and isolate it if you need to, in the debug process. Consider doing the same for your 328 uC.
15. Optional: test points to sniff the USB lines
16. Optional: test points to sniff the UART lines
17. Optional: test points to sniff the I2C lines
18. Optional: inrush current monitor using 2 single-ended 10x probes 19. Optional: ferrite bead filter to the AVCC of the 328

21.2.2 Graduate students are required to add the following features to their Arduino boards:

1. A ferrite filter on the AVCC pin to the ADC circuit on the 328
2. A 0.5 ohm series resistor in the power rail and connectors to it to measure the supply current. Add two single-ended test points for the 10x probe to use a scope to measure the inrush current
3. Selectable power from either a 5 V external AC to DC converter or from a USB connector
4. Add test points to the digital lines to sniff the bus traffic

Undergraduates can add these features to their board, but they are not required.

21.2.3 Develop and document your POR:

In your POR, you will need to include what it means to “work”. These are your functional requirements which will be used to test against your board to give it the acceptance of “yes, it works.”

We only stock 1206 parts. Since you will be assembling your board yourself, you should plan to only use 1206 parts.

In addition to the schedule, be sure to include in your POR what you will want to implement to reduce the risk and reduce the potential noise, such as:

1. *What special features in the schematic do you want to add to facilitate test and debug?*
2. *What special features do you want to add in the schematic to reduce power rail noise?*
3. *JLC will also assemble your board. Try to select parts in the library JLC also stocks.*
4. *What special features in your layout do you want to include to reduce the risk for:*
 - a. *User interface*
 - b. *Test and debug*
 - c. *Lower cross talk*
 - d. *Lower power rail noise*
 - e. *Lower cross talk for signals coming off the board through the Arduino pins*
5. *Look at the commercial version of the Arduino Uno board in your lab kit. What three features can you think of doing differently to make your board lower noise?*
6. *What potential risks can you imagine and what can you do to avoid them? Remember, “think like Ralphie’s mom.”*

Here is one hint: We want to use the same pin out layout for the pin headers so it is compatible with other Arduino boards and shields.

21.2.4 Do not just copy a reference design

You can start your design based on reference designs. Afterall, you are not the first person to design an Arduino board. However, you will see that many reference designs are either wrong or have features which do not apply to your board. In particular, watch out for blindly adding features such as:

- *A MOSFET switch to control the connection between the 7-12 V input and the USB plug*
- *A 5 V on board regulator*
- *A polyswitch resettable fuse in the power line (they are not worth it.)*
- *Multiple values decoupling capacitors*

Here are some examples of reference design to evaluate:

https://www.arduino.cc/en/uploads/Main/Arduino_Uino_Rev3-schematic.pdf

<https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/>

<https://learn.circuit.rocks/the-basic-arduino-schematic-diagram> <https://www.baldengineer.com/diy->

[arduino-schematic-checklist.html](http://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/RedBoard-V22.pdf) <http://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/RedBoard-V22.pdf> <https://circuitmaker.com/Projects/Details/Troy-Reynolds/Arduino-Uno-Rev3-Reference-Design>

21.3 BOM and non-commodity parts

The minimalist board you will design and build will use the Microchip Atmega 328 micro controller. There are multiple ways of implementing the UART to USB interface. We will use the CH340g

You will use the USB C connector. This is more robust than the micro and a smaller form factor than the large square type A connector.

For powering, you will use the USB connector. If you use an external power jack, implement a feature so that you would NEVER have both connected to power your board at the same time.

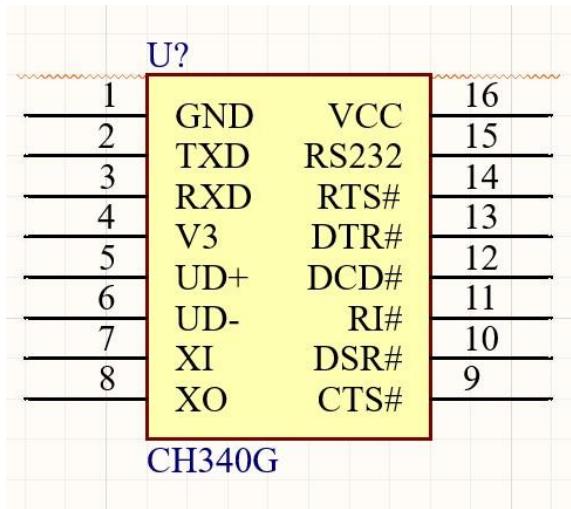
We will use a 2-layer board for the Arduino Uno. The board will be identical in function as the millions of commercial Arduino Uno boards out there, but you will design and build this one the right way for reduced switching noise.

You will be measuring the noise in the commercial boards and in the one you design and build to see the impact from improving the layout on the noise reduction.

While you can use some of the various reference designs on the internet to start, you will take responsibility for all the design decisions. Just because the design is published does not mean it is a recommended design.

21.3.1 The USB to UART chip is the CH340g

The CH340g is the interface between the USB port and the UART to the 328. The symbol is here:



It connects via the TXD and RXD pins. Be sure to note that the TXD on the CH340 connects to the RXD on the 328 and the RXD on the CH340 connects to the TXD of the 328. Pick a labeling strategy for the RX and TX pins and stick to it. Maybe something like CH340_TX, or 328_RX.

The datasheet for this component, translated poorly from the Chinese, is here:
<https://cdn.sparkfun.com/datasheets/Dev/Arduino/Other/CH340DS1.PDF>

You will want to power the Vcc at 5 V. However, there is an internal 3.3 V pin, which connects to an internal 3.3 V regulator on the 340 chip. This 3.3 V rail supplied by the CH340g chip should not be connected to anything else except a decoupling capacitor. It should not connect to the 3.3 V regulator on the board or to power any other components.

This pin is pulled out so you can connect a decoupling capacitor to it. Remember, regardless of what the datasheet says about the values of the decoupling capacitors to use, you know the right way of decoupling.

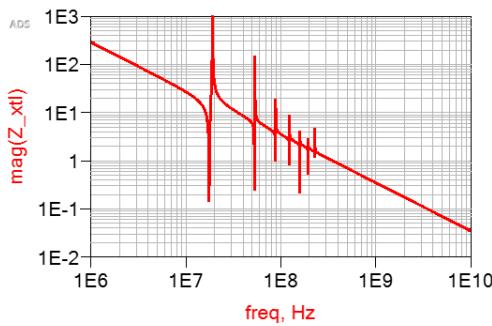
21.3.2 Crystal circuits

There are multiple ways of generating a clock signal on a micro controller. They all follow the common practice of placing a resonating structure across the input and output of an inverter circuit. The inverter will oscillate at the resonant frequency of the resonator.

There are two common types of resonators, quartz crystals and ceramic resonators. The quartz crystal is more stable but has a larger form factor. A ceramic resonator is smaller, and is a little less stable, but still can have a stability within 10 ppm.

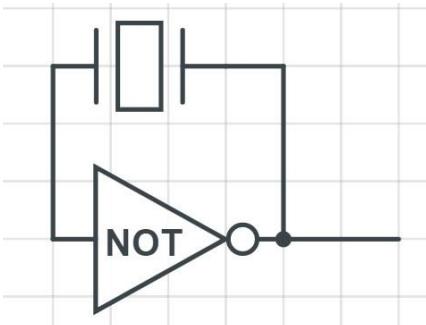
For our boards, we will use ceramic resonators. They are smaller size, lower cost and readily available. They behave very similar to crystals.

A crystal is a slice of quartz between two electrodes. It is piezo electric. A voltage across the end faces causes a mechanical compression and a mechanical compression of the crystal causes a voltage across its ends. Due to this coupling of mechanical motion and electrical signal, it has an impedance which dips to a low value at the resonant frequencies for which it will vibrate. The figure below is an example of the impedance profile of a typical crystal.

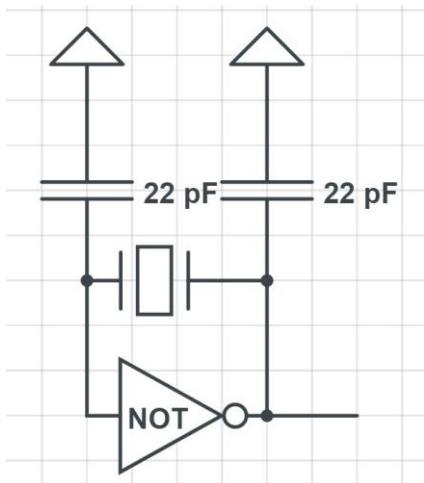


Note that this crystal, as is typical, has multiple resonances above the first harmonic.

When the crystal is in the feedback loop of a high gain inverting amplifier, the frequency where the impedance is lowest, ie, where there is more feedback, and at which there is a high voltage generated by the compression of the crystal, will be the frequency of oscillation. The basic inverting circuit is shown below.



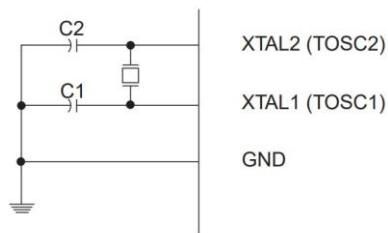
But, the crystal can oscillate at multiple harmonics. In order to use a quartz crystal in a clock generation circuit, we need to suppress the higher frequencies, so they don't resonate. We do this by adding filter capacitors to ground to shunt the higher frequencies to ground. The value of the capacitors should be in the range of 5 pF to 30 pF, with 22 pF a good starting place. The final circuit looks like this:



In rare cases, especially when using a discrete CMOS inverter circuit, it is necessary to kick start the oscillations by adding a 1 Meg shunt resistor across the crystal. This is not usually the case when the inverter is built into a uC, as there is enough noise to kick start the oscillations.

In the [328 schematic](#), (Figure 9-2), shown below, there is no feedback resistor specified. There are just the 22 pF capacitors to suppress higher order modes.

Figure 9-2. Crystal Oscillator Connections



The traces from the crystal pins of the 328 to the crystal and the capacitors are sensitive to noise and trace capacitance. **Try to route the traces as short as practical.** Try to avoid passing the signal traces from the IC pins to the crystal pads and 22 pF filter capacitors through vias, or routing over gaps in the return path.

This is one of the circuits you will build in a solderless breadboard prototype.

21.3.3 Debounce circuits and switches

There is a reset pin on the 328, PC6. The reset pin is normally held high with a pull up resistor you will add in your circuit. This means the reset is normally disabled. When the reset pin is momentarily pulled low, the 328 resets the sketch it is running and starts from the beginning. It only needs to be pulled low for 10 usec to start the reset operation.

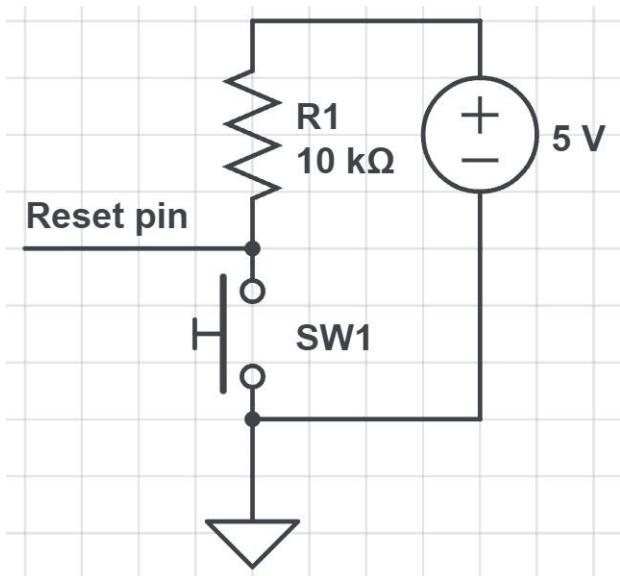
In a typical application, we will pull the reset pin low in two ways: with a manual switch so we can reset the 328 any time we want by pushing a button, and automatically from the CH340g device.

A mechanical switch we commonly use as a reset is this one:

<https://ilcpch.com/parts/componentSearch?isSearch=true&searchTxt=C174049>

When it is pressed, a pair of contacts short to each other.

As an example, the figure below shows a simple pull-down switch circuit. The reset pin is normally pulled high, until the switch is closed. Then the rest pin is pulled low while the switch is depressed.



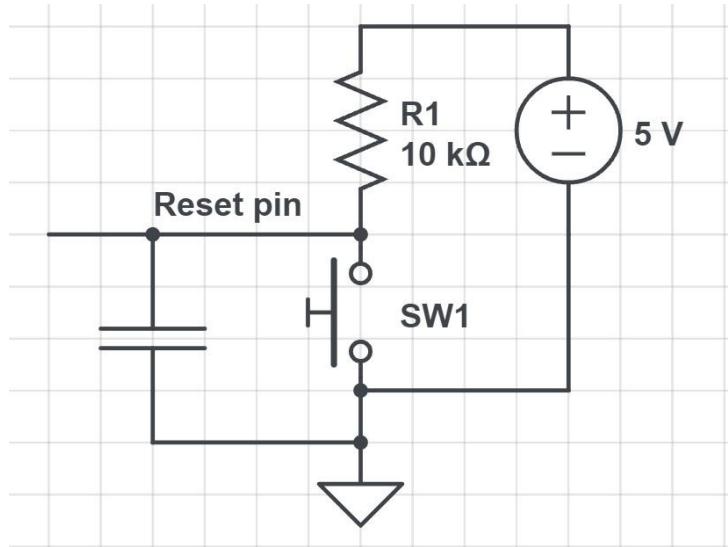
There is one problem with this circuit. Many mechanical switches, after they make contact, "bounce" up and down a few times before they finally stay closed. This is called bouncing. As a little background, check out [this article](#) I wrote about bouncing circuits.

If the bouncing happens on a reset pin, it is possible the multiple contacts will cause multiple resets and potentially a fault condition. The typical spec for a reset pin is that a pin has to be brought low for > 10 usec for the reset pin to read the reset pin going low.

A reset pin would have seen this voltage fluctuate many times and go through multiple reset attempts in the 1 msec time before the switch voltage stabilized.

While this can be fixed in software by adding a delay after a reset is detected before another reset is detectable, it can also be fixed in hardware by adding a debouncing circuit.

The purpose of a debouncing circuit is to hold the reset pin low, after the switch is pressed, and until all the bouncing has stopped. This is as simple as adding a capacitor across the switch, as shown in the figure below.



In this circuit, the resistor keeps the reset line normally high. The capacitor is charged to 5 V as well. When the switch is closed, the reset pin is pulled low and the capacitor is discharged to 0 V as well. When the switch bounces up the first time, the capacitor holds the reset pin low. It has an RC charging time, back up to 5 V, of $R \times C$. If the bouncing time for the switch to settle is short compared to the RC charging time, the reset pin will be kept low.

For example, if $C = 1 \mu\text{F}$ and $R = 10 \text{ k}\Omega$, the RC time constant is 10 msec. As long as all the bouncing time is finished in a time short compared to 10 msec, the reset pin will be kept low. It will not see the bouncing. Any time constant longer than about 1 msec is probably good enough.

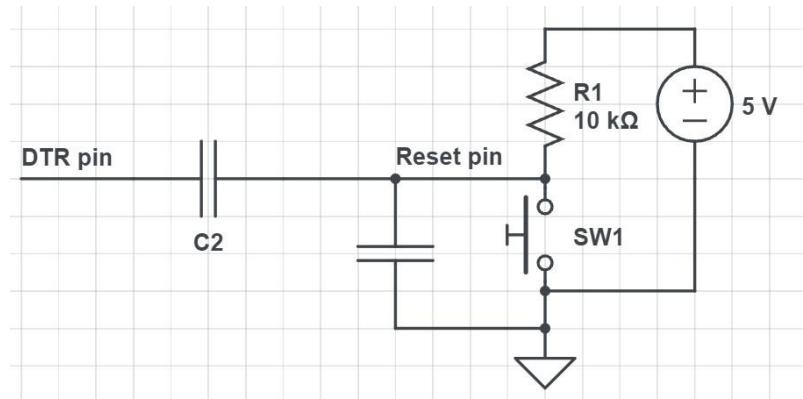
Any capacitor $> 0.1 \mu\text{F}$ would be a suitable debounce capacitor. Since the exact values are not critical, use component values you are already using on your board. For example, a 1 k ohm resistor and a 1 μF capacitor, or a 10k and 0.1 μF , or 10k and 1 μF .

This is for the debounce circuit.

We need to set up the reset pin on the 328 to accept a pull-down signal either from the manual switch or the CH340g DTR pin pulling it low. Here is how we will do it.

The DTR pin on the CH340g chip is normally HIGH. When it starts receiving communications on the USB bus and has code to send to the 328, the DTR pin is pulled LOW. The problem is, the CH340g DTR pin will be pulled low for many seconds, while it is communicating over the USB bus. We only want the initial pulled low signal to trigger the reset of the 328. We do this with a high pass filter. This is a series capacitor to the reset pin of the 328.

We connect the DTR pin to the reset pin with a high pass filter- a series capacitor. This will only let through the high frequency, negative, falling edge. This circuit is shown below:



The one problem is that we have also built a capacitor voltage divider circuit. The voltage on the reset pin is the voltage divider of the C2 series capacitor and the debounce capacitor. If $C_2 = C_{\text{debounce}}$, the voltage on the reset pin will only be $\frac{1}{2}$ the voltage swing of the DTR pin. It may not be enough to pull the reset pin low.

In order to pull the reset pin low enough when the DTR pin pulls down, C2 should be much larger than the debounce capacitor. If the debounce capacitor is 1 uF, then C2 should be at least 22 uF.

If the DTR pin has a 3.3 V signal range, then when it goes low, the reset pin will only go as low as $5 \text{ V} - 3.3 \text{ V} = 1.7 \text{ V}$. This is not low enough to pull the reset pin low. We need to pull the reset pin below 0.8 V.

This is why it is so important to make sure the voltage supplying the CH340g chip, that powers the DTR pin is 5 V and not 3.3 V.

There are lots of important design details to pay attention to.

21.3.4 The ATmega 328 microprocessor

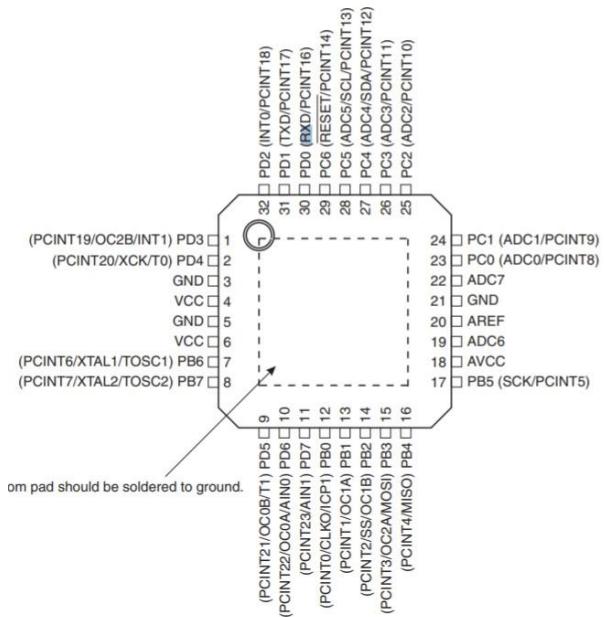
A commonly used 328 uC is this one:

<https://www.digikey.com/product-detail/en/microchip-technology/ATMEGA328P-ANR/ATMEGA328PANRCT-ND/2774230>

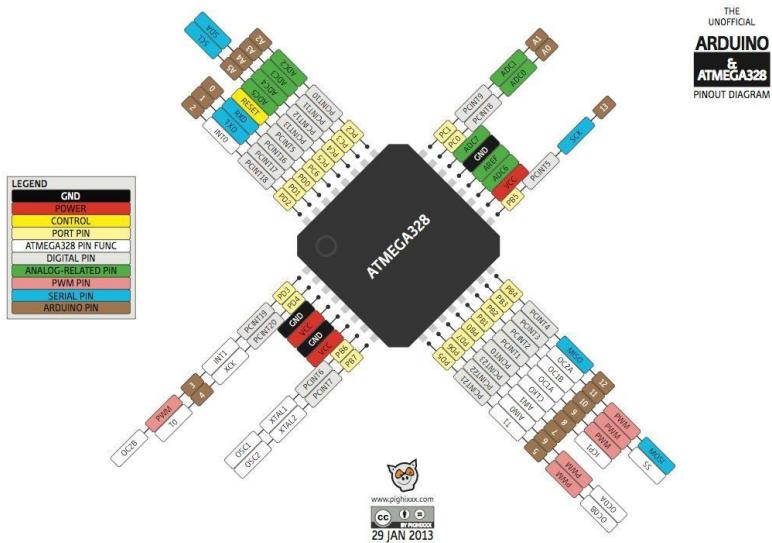
Take a look at its specs. Check out some reference designs that use this uC as well.

The device we will use in this board is a 32 pin package. The pinout is in the datasheet and shown here:

32 VQFN Top View



Here is another diagram of the pin out:



Note, the RXD pin of the 328 is PD0 and the TXD of the 328 is the PD1 pin.

You can put indicator LEDs on the TX and RX pins using a 1k resistor and series LED. What will this circuit look like?

Use a separate connection to the AVCC and VCC pins. They each get their own decoupling capacitor. Ideally, you should add an LC filter between the power rail and the AVCC pin to keep noise on the Vcc rail from getting onto the AVCC rail. This is with a ferrite inductor.

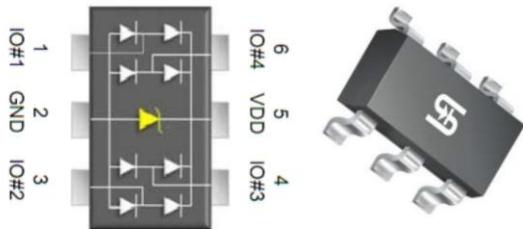
The RX and TX pins connect to the 340 chip. Otherwise, all the pins come out to the header pins. The 16 MHz crystal connects to the XTAL pins. Each Vcc pin should have its own decoupling capacitor.

21.3.5 The TVS chip

An important protection device for your circuit is the transient voltage suppression (TVS) chip. The purpose of this chip is to protect your computer when it is connected to your Arduino board with the USB cable. If someone touches your board and is charged up by ESD, you do not want this high voltage discharge to damage the USB hub chip on your laptop.

The one we are using for this board is this one: <https://www.digikey.com/en/products/detail/taiwan-semiconductor-corporation/TESD5V0V4UCX6-RFG/7623393>

In the datasheet you will see the internal circuit shown below:



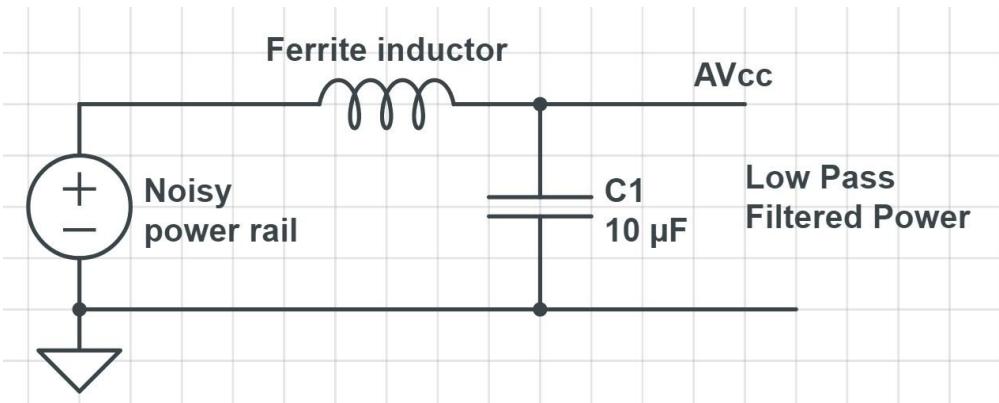
Based on lab 14 you should know about the behavior of how a TVS chip works.

How will you connect the TVS chip to the power and ground connections on your USB connector and to the D+ and D- data lines? All that is necessary is gnd to pin 2, 5 V to pin 5, and D+ and D- go to pin 1, 3, 4, or 6. You will only have two data lines, so you may leave two IO pins open on the package.

21.3.6 Using a ferrite bead to filter noise from the PDN to the AVCC pin of the 328

This topic was covered in the textbook in chapter 14.9.

What ferrite is available in the library? The inductor in the library is 10 μF and the largest 1206 decoupling capacitor available is 22 μF . What pole frequency will you be able to achieve with this ferrite and a large decoupling capacitor? The circuit should be as follows:



21.4 Part 1: Complete the schematic and layout for the Golden Arduino board

21.4.1 Use reference designs as a guide but take responsibility

There is value looking at reference designs to get you started, but always take responsibility for your own design. You can review the reference designs mentioned in the last section.

Be aware that there are features in some reference designs you WILL NOT use and layout principles you SHOULD NOT follow. You are probably more expert at board design than the person doing the layout in some of the reference designs. Chances are the person who did the reference design layout did not understand the root cause of switching noise, for example, like you do.

Take responsibility for your own design.

21.4.2 Complete the schematic

While we will assemble our own boards for cost reasons, you may wish to use parts just from the JLC integrated library so if you order this design in the future, you could have them assemble it for you. Choose these parts if at all possible.

When you select a **surface mount** part to be added to your schematic, wherever possible, be sure to select a part that has an LCSC part number and in particular, a Basic type to keep your costs down.

If you need a unique part like say a 2 uF capacitor, make sure to check with the TA or the professor. We can either substitute parts or do creative things like using two, 1 uF in parallel. I am willing to order parts if there is a compelling reason.

Select parts as needed from your integrated library and place them in your schematic.

Place parts on your schematic page with some flow: power on top, signal flow from inputs to outputs, and components used together in specific functions, in close proximity.

Connect up the terminals in the schematic as required. Use net names when appropriate. Think about which ones you want to connect by direct connection vs by common net names.

Try to complete the schematic as well as you can on your own, without consultation except about using the tool.

21.4.3 Stackup and board outline

You will construct a 2-layer board. This is the same stack up as used in all commercial boards. This way you will be able to compare exactly the same stackup in your board as in the commercial boards.

You can use any board size and shape you want with two constraints:

- *Keep the dimensions under 3.9 x 3.9 in. This will keep the cost down.*

- *Make sure you include the header sockets in the same location as in the commercial Arduino so that shields we add to your board are compatible.*

Even though the bottom layer will be a ground plane, make sure you select the bottom layer as a “signal” layer. You will add a polygon pour to the bottom layer. This way it will be a positive layer.

21.4.4 Start the layout

Your schematic should be complete. It should have gone through a design review with another pair of eyes. You should have completed an Electrical Rule Check (ERC). This does not check if your design is correct, only if there is something obviously wrong like unconnected terminals, or inputs that should be outputs.

Push your schematic to your board. You should see a room with all the parts placed.

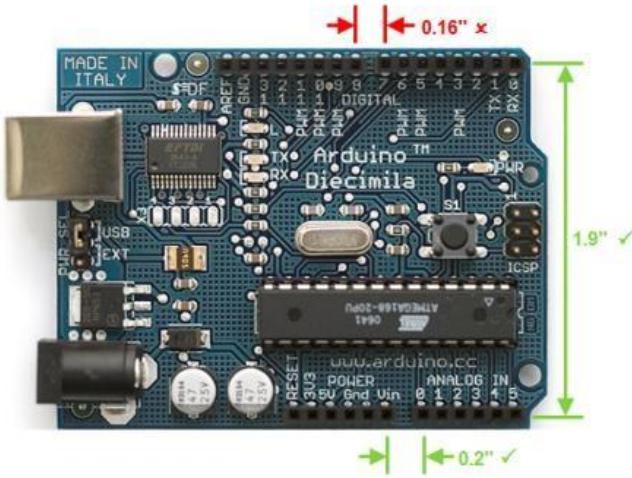
Place the parts on the board in this order:

- *Connectors on the outside edge, as needed. Be sure to use the correct positioning for all the header connectors so they are Arduino compatible.*
- *Place the ICs and rotate them to minimize the routing congestion*
- *Once ICs are placed, place decoupling capacitors where needed- as close to the Vcc pins as practical.*
- *Place the crystal and its components in close proximity to the IC pins.*
- *Place switches and indicator LEDs as needed*
- *Indicator LEDs should be in close proximity to what they are indicating and near any switches to which they refer.*
- *Route ground vias with traces as short as practical*
- *Route signal traces. Try to route all on the top metal layer*
- *Use cross-unders as needed, keeping them as short as practical.*
- *Route power paths on the top layer*
- *You only need thermal relief vias if they are used in a soldered pin or are close to a soldered pad. Otherwise, do not use them, as they take up more area than a regular via. They should be used for all the connector pins. This will make them easier to solder.*
- *Check and verify you have no ghost wires left unrouted.*

21.4.5 Add the header pins in the correct locations on your board

Be sure your header pins meet the standard Arduino format so you can plug our test shield into your board.

The footprint you should use for the header pins is shown below, from <http://brettbeauregard.com/blog/2009/07/arduino-offset-header/>. Your board outline does not have to match this board outline.

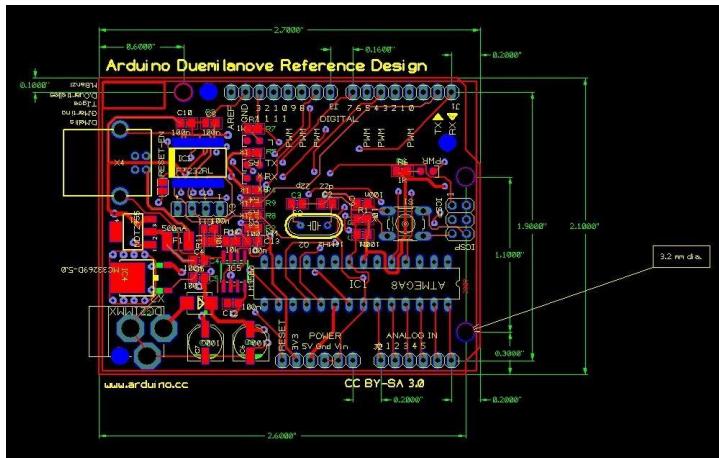


Note that the spacing between centers of the digital pins in the upper part of the illustration is 160 mils. This is not a multiple of 100 mils! Apparently, it was a mistake not caught and left in at the last minute and has been kept in due to legacy shields. You can read about it here. <http://brettbeauregard.com/blog/2009/07/arduino-offset-header/> This

is why a good CDR is important before you release a board!

A more detailed dimension drawing is shown below, taken from here:

<https://forum.sparkfun.com/viewtopic.php?t=24335>



Your board outline does not have to be the standard Arduino shape. Be sure to use the keepout layer for the board outline and include this layer in the gerbers.

It is not essential to add the ICSP 6 pin header. These have special connections for the SPI bus and to connect a header connector to a bootloader to burn the boot load code into your Arduino.

You can use this connector to connect to the bootloader, or you can use the same connections located in the two columns of header pins on the sides of your Arduino board. Your choice if you want to include the ICSP pins.

21.4.6 Layout tips: design for connectivity, signal integrity, assembly, test and bring up

For this design, we will use the same design rules for any low-cost board:

- *Signal lines as 6 mil wide*
- *Closest space to any other metal, 6 mils. Try for 10 if practical.*
- *Top layer: signal and power routing*
- *Bottom layer: continuous ground, and some cross unders (make bottom layer a signal layer and use a copper polygon for the ground net)*
- *Power paths should be 20 mils wide.*
- *Avoid routing signal traces between the mounting pads of components. It will make it easier in tracing the routing if you can see the whole path. Try to make the traces visible with components in place so that you can trace the routing for debug.*
- *If there is no signal routing under the die, consider using the region under the die to connect to ground vias and bus all the power connections between the common power pins of the IC.*
- *Each Vcc pin needs its own decoupling capacitor. Where should the decoupling capacitors be placed?*

- All components have their designator, component value, and orientation marking in the silk screen, in close proximity of the component.
- Place all LED indicators in close proximity to what they are indicating with a polarity indicator.
- Add brief labels to all indicator LEDs in an obvious way. Don't use "LED" as the label, use "3.3 V" or "TX" or "pin 13".
- Keep the silk screen pen width > 6 mils and the character height at least 50 mils high.
- If you have any 3-pin switches with shorting flags label the flag position for on-off or the various options.
- Make sure each test point has a silk screen label that describes the test point in an obvious way. Label the test point so it is instantly obvious what it is measuring.
- BE SURE TO ADD YOUR NAMES AND BOARD NAME AND BRD REV in the silk screen on your board.
- Add some test points to measure the 5 V rail and 3.3. V rail on the board with the 10x probe of the scope. What other circuit nodes do you want to measure?
- Place the crystal and its filter capacitors as close to the IC pins as practical and route the traces as short as practical.

21.4.7 Design for performance

For this golden Arduino board, consider features you can add to reduce the switching noise between signals and for power rails.

This means provide a continuous return path for each signal trace, add adjacent return pins to each signal pin and low loop inductance decoupling caps. Use a ferrite bead filter on noise-sensitive power pins.

- Route adjacent signal lines as far apart from each other as practical for lower cross talk.
- For all signal cross unders in the ground plane, keep the cross under length short. Better to have multiple cross unders in a trace than one long cross under, making a long gap in the return plane. A long gap in the return path is bad.
- If a gap in the return path is longer than $\frac{1}{2}$ inch, add at least one return strap across a gap, in the middle of the gap.
- Do not use copper pour on any layer except ground. This is a bad habit. It solves no problem, makes debug harder, increases the risk that the returns are not well engineered and can result in HIGHER near field emissions and higher cross talk especially for rf signals, than if there were no copper pour on a signal layer.
- Decoupling capacitors should have as low a loop inductance in their power and ground connections to the IC as possible. This means place capacitors as close as possible to the IC they are decoupling, with wide traces from capacitors to power pins.
- Use at least 1 decoupling cap per power pin, more if practical.
- For graduate students: For very noise sensitive voltage pins, like AVCC, add a ferrite bead-capacitor LC low pass filter to filter noise **from** the board **onto** the sensitive pin.

- If there is no routing under an IC, consider routing all the ground lead connections inward, under the die footprint, rather than outward in the region of the board used for signal routing. This can enable shorter connections to the ground plane.

21.4.8 Watch out for these common problems

- 1 . The TX of the 340 should connect to the RX of the 328 Again, TX to RX NOT TX to TX
- 2 . Label the TX pin either as the TX_328 or TX_340 depending on which is the direct connection
- 3 . Check the spacing between the header pins. It must exactly match the commercial Arduino board spacing
- 4 . No need to connect anything expect a pin header to Aref
- 5 . Add your name to the board
- 6 . Label all the header pins
- 7 . Make sure all the ground pins in the header are thermal relief
- 8 . Label all the indicator LEDs
- 9 . Label all the test points
- 10 . The 3.3 V pin on the 340 chip should only be connected to its own decoupling capacitor
- 11 . Each VCC pin should have its own decoupling capacitor
- 12 . Do not share any ground vias.
- 13 . Your ground plane has short length cross unders and an otherwise solid plane
- 14 . You do not add a copper fill on the top signal layer
- 15 . Verify the crystal has 22 pF caps and the decoupling caps are 22 uF
- 16 . Check the dimensions and spacings of the header pins to make sure they conform to the Arduino spec.
- 17 . The TVS chip should have one pin connected to the D+ line and one pin to the D- line.
- 18 . A sense LED on the RX or TX pins should be routed to ground with a 1k series resistor, not in series with the TX and RX lines
- 19 . A decoupling capacitor connects between the Vcc pin to ground with as short a path as possible, NOT in series with the power path.

21.4.9 Review your design yourself

If you think you have completed your layout, take a moment to carefully review the board design against the check list in Chapter 20 and 21 of the textbook and the check lists in the last few sections.

Perform a DFM using the Altium built in tool. This will just check that your layout conforms to the design rules you set up. It will also point out any ghost wires, not connected. It will not check for design errors.

21.4.10 Perform a CDR for brd 3

The purpose of the CDR is not just to verify that your board can be manufactured, but to verify you have the correct electrical designs, connections, components and best layout design features for connectivity, performance, assembly, bring up and test.

Remember, after you submit your board for production, there will be as much as 2 weeks before you get your board back. This means any changes you might want to make will have more than a 2 week feedback cycle. This delay is way too long for a successful board on a tight schedule.

Catch all of your potential problems BEFORE you submit your final design files to be ordered.

When you review another engineer's design this gives you practice in catching problems. Listen to the reasoning of other designers for selecting the design decisions they did. Maybe you will pick up a pointer or two.

We will do a CDR as a group. Everyone should come to the lab with your best layout complete. Individuals will be selected to review your layout to the class.

When reviewing another person's design, look for hard errors that might prevent the board from "working." These are most important. There is a soft error and a personal preference. If you think the design should have been done differently, but it will not result in an error, understand the rationale for the feature. You may be convinced to change your mind.

21.4.11 Export your Gerber and design files

After your design is complete and you've performed a CDR with other pairs of eyes, export the Gerber files and NC drill file.

In your exported Gerber file, be sure to include just:

- *Bottom metal layer*
- *Bottom solder mask*
- *Top metal layer*
- *Top silk screen (overlay)*
- *Top solder mask*
- *Board outline*
- *NC drill file*

If you do not include the bottom solder (mask) layer, it will be almost impossible to solder any pins in the board. Trying to solder pins on the bottom, like the power jack or USB pins, will result in shorts to the ground plane.

We will not be doing any solder paste stencil printing, so no need for a solder paste layer.

No need to include any other mechanical layer. In fact, adding additional mechanical layers may mess up your design if there is additional metal patterns added to the board in these other mechanical layers.

Do not add every file possible and let the fab vendor sort out what they need. It may delay your board, or they may use the wrong files.

After you have created the Gerber files, check them using a tool like [gerby](#) or the online gerber viewer on the PCBway web site: <https://www.pcbway.com/project/OnlineGerberViewer.html>

21.4.12 Submit your board to JLCpcb to use their DFM

You will include a screen capture of the approved DFM check from JLCPCB.

You will include an assembly (Pick and Place) files

You will included a BOM file. Export your BOM in .xls format. Make sure there is **ONLY** one part per row in the spreadsheet. Only then should you export the BOM. Again, this file is useful for future designs where you wish the vendor to provide assembly of your board.

DO NOT change the reference designator values on your layout.

If you select any features in the board set up that are not standard, be sure to check the impact on the price. The board price should be \$2.00.

You will need an account to place an order. It is free and you do not have to add any CC information.

If the quote is more than \$2, check your design and your options. If the price is > \$2 you should have a very good reason for this.

You can request a DFM check be performed before you place the order. DO NOT PLACE THE ORDER You will receive an email within an hour with an ok or a note with the errors. Fix any errors and try again.

When you have gotten the ok, you can submit your three board files to canvas.
It is very important to have the correct file names: lastname_typeOfFile_boardName_otherInfo
The file extensions must be:

- ✓ *Gerber: .zip*
- ✓ *BOM: .xls*
- ✓ *Pick and place: .csv*
- ✓ Screen capture of DFM check

Submit these files on canvas on the day of the deadline. If you miss this hard deadline, your board will not be ordered. All board orders will be placed by the TA deadline and if you miss these deadlines you will have to fund your own order, ouch!.

21.4.13 Part 1 grading rubric

As with all the board design assignments, the grading rubric is:

1 point for layout sign off by TA

1 point if your design files are turned in on time

1 point if your design is accepted by JLC in the order

21.5 Part 2: bring up, boot load and test of brd 3

21.5.1 Assembly and bring-up

In this lab you will complete any final assembly needed for your board 3, do some preliminary debug and bootload the code on the ATmega328 so it will understand the Arduino IDE over the USB.

After you have verified you can run and modify Blink on your ATmega328, measure the switching noise on a quiet hi and quiet low pin, when 1 I/O switches, 2 I/O and 3 I/O switch using the same code and the same switching shield as you used in one of the labs.

Remember the important steps in bringing up your board:

1. *Inspect the bare board*
2. *Test for opens and shorts of the power paths*
3. *Take pictures of your bare board and the assembled board. The report in your portfolio will look great if you can show the sketch of the schematic, your final schematic, your layout, the bare board, and the final assembled, board and then measurements from your final board. What a great example this can be of the complete start to finish flow of designing, building and debugging a board!*
4. *When your Atmega328 uC is ready to test, you will need to burn the bootloader into the uC. See below for this process.*
5. *After your board is running, be sure to measure some of the switching noise features, as you did on the commercial Arduino board.*

21.5.2 Burn the bootloader

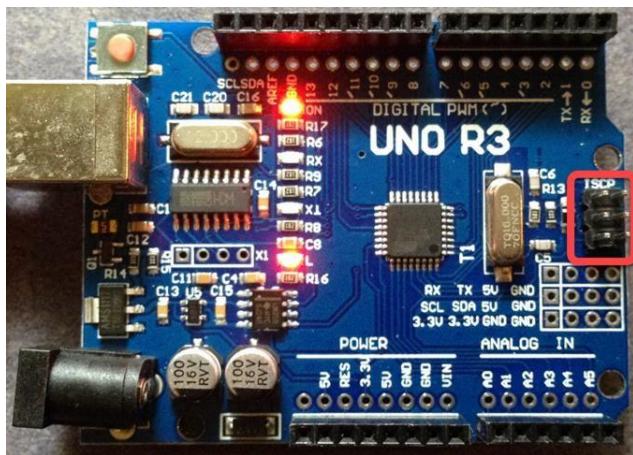
The Atmega 328 comes up dumb. The first step in using the 328 is to turn the 328 into an Arduino. You must teach it how to interpret the UART signals from the CH340g into Arduino IDE speak. This is by burning the boot loader code into the 328.

The program we add to the Atmega 328 is the bootload program. It is like the bios in a microprocessor. It is the first thing the microcontroller runs when it turns on. We need to give it the instructions on how to listen to the UART data lines and how to interpret the commands from the Arduino IDE compiler.

After we burn the bootloader in the ATmega328, your board becomes an Arduino.

The simplest guide to burn the bootloader is on the [Arduino.cc web site](#), and on the [Sparkfun web site](#).

You will use a standard commercial Arduino Uno board as the *programmer*. It will act as the In-circuit Serial Programmer (ISP). It will generate the code and drive the In-Circuit Serial Programmer (ICSP) pins on the *target* board, your Atmega 328 board, which will turn it into an Uno after the boot code is loaded. The ICSP header on an Uno is shown in the figure below.



On your board 3, the Target board, you can connect wires to the ICSP header pins, or to the specific digital pins in the header. If you routed your ICSP pins in the correct way, their pin assignment will be shown in the figure below.

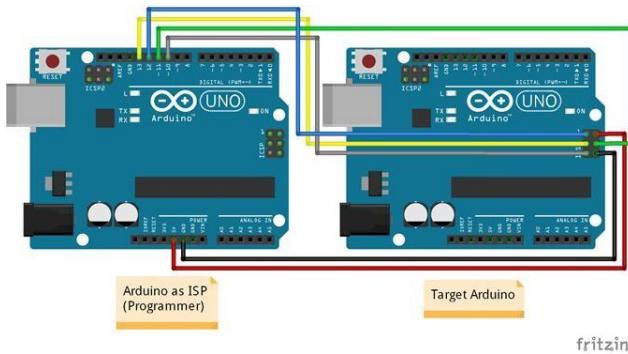


The ICSP header are male pins. The header pins on the digital I/O pins of your board are female sockets. To connect between digital socket header pins and digital socket header pins, you can use standard male to male jumper wires. To connect between ICSP pins on both boards, you will need female to female jumper wires. To connect between ICSP pins on one board to digital pins on another board, you will need jumper wires with female on one end and male on the other.

Regardless of the method you choose, you will need 6 jumper wires.

We will use an Arduino Uno as the in-system programmer (ISP) to burn the bootloader code in our Atmega 328 microcontroller. The Uno is the programmer, our Atmega 328 board is the target.

The pins to connect on the target Atmega 328 board, your board, are the ICSP pins. An example of the wiring between the ISP and your target board, taken from the SparkFun website is shown in the figure below.



The pin connectivity between the ISP (Uno) and the Target (your board) are:

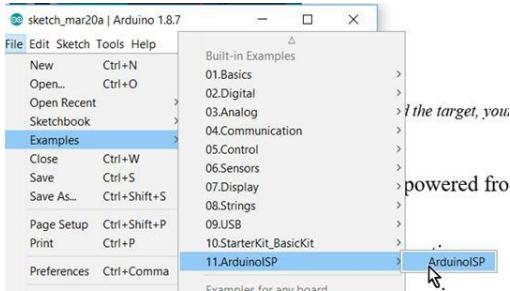
<u>Uno</u>	<u>Atmega328</u>	<u>Comments</u>
5 V	5 V	these are the 5 V rails on each board
Gnd	gnd	ground connections
D10	reset	controls the reset pin of the Atmega328 programmatically
D11/MOSI	D11/MOSI	Master Output, Slave Input
D12/MISO	D12/MISO	Master Input, Slave Output
D13/SCK	D13/SCK	Serial clock

Note: in this configuration, the Atmega 328 board is powered by the ICSP pins.

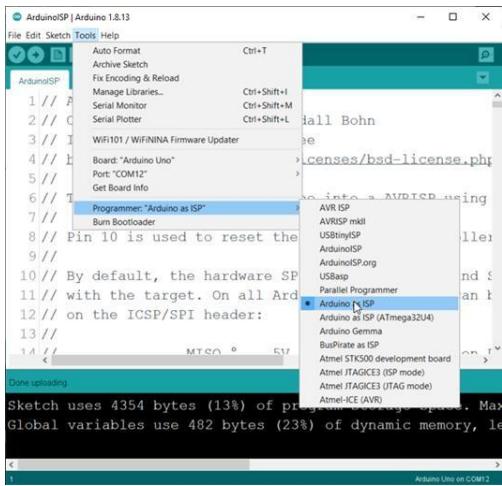
This means you should NOT plug your board into a USB port, OR into an external power jack.

Once the two boards are wired up, the basic process is:

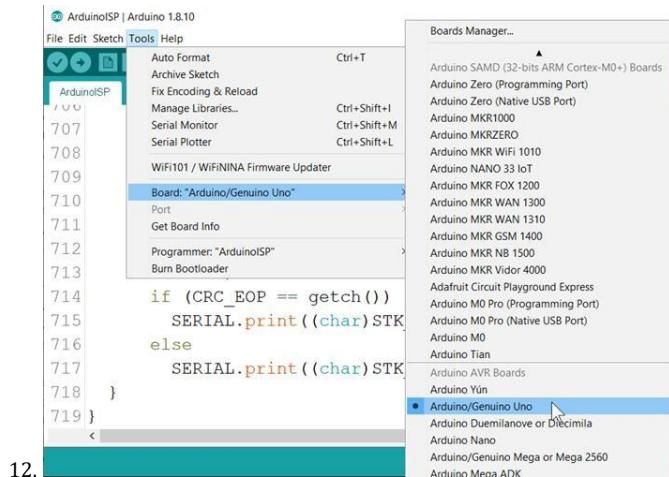
1. Open the ArduinoISP sketch (in Examples) to your Arduino Uno board. It is the item #11 in the files/examples pull down, as shown in the figure. This has the firmware we will burn into the bootloader of the target, your Atmega328.



2. You will need to set up your Uno to talk to your USB port, just like any sketch. Select the items in the Tools > Board and Serial Port menus that correspond to the board you are using as the programmer, the Uno (not your target board).
3. At this point, you can upload the sketch to your Arduino Uno ISP.
4. Upload the ArduinoISP sketch to the Uno that is the programmer. Your Uno is now an Arduino ISP. This code that has turned your Uno into an Arduino ISP is now running on your Uno.
5. This ISP sketch uses three digital pins as diagnostics:
 - ✓ pin 9: Heartbeat - shows the programmer is running
 - ✓ pin 8: Error - Lights up if something goes wrong (use red if that makes sense)
 - ✓ pin 7: Programming - In communication with the slave
 - ✓ Rather than add a resistor with an LED to each pin as a diagnostic, we will use 2 channels of a scope. You can use 2, 10x probes plugged into the UNO acting as the ISP. Note: since we only want to see high level signals, we can get away with the large loop inductance of the ground lead. Use a short wire stuck into pins 7 and 9 to the 10x probe tips, and a wire in one of the ground pins for the ground lead. Try to keep the wires short.
 - ✓ Set up your scope to read the 5 V signals from each channel. Use pin 9, the heartbeat to trigger the scope. You might see a 500 Hz PWM signal from this pin.
6. Make sure your Arduino Uno programmer board is wired correctly to your target Atmega328 board, your board. At this point, the programmer Uno board will be powering your board.
7. Now we begin the process of burning the bootloader code into your board. Select the item in the Tools > Board menu that corresponds to the target board which is an Arduino.
8. Select the "Arduino as ISP" in the Tools>Programmer menu, as shown in the figure.



9. As the final step, select the Burn Bootloader command, also on the File/Tools menu, as the bottom option. This step will actually burn the firmware into your ATmega328. After this step is complete, you are done. Your ATmega328 should now be an Arduino Uno board.
10. Remove all the connections from your new Arduino board. At this point your ATmega328 should now be an Arduino and understand the Arduino IDE commands over the USB.
11. To test your new Arduino board, connect it to your computer with a USB cable. In the Arduino IDE, be sure to select as your board, an **Arduino Uno** as shown in figure, when you try to program your board over the USB connection. And select its correct com port.



12.

13. Upload Blink, change the delay to 100 and see if it blinks quickly. If it does, success!

When your board is boot loaded and it runs the modified Blink code, your board is “working.”

21.5.3 Bring up and test

Once your Golden Arduino board has its bootloader burned, you should be able to run any sketch. Try uploading a simple Blink sketch.

Just to be sure you are talking to your Arduino, change the delay in blink and make the LED or the signal on pin 13 turn off and on at 5 Hz.

When you can demonstrate this call your TA over to proudly show the birth of your Arduino.

Try a few other sketches.

21.5.4 Debugging

But, what if your board does not work? You need to determine as quickly as possible if the problem is a design problem either in the schematic, in the layout or in the choice of components, or is it a manufacturing defect or if it is a defective part.

Is there a hard error, and can you fix it?

If a design problem, can you fix it with an engineering change wire, or replacing a component?

If a manufacturing defect, can you repair the defect? Does the other board work?

Are there any soft errors, that you can tolerate, but will make note of for the future to avoid?

If you have isolation switches, you can try using them to isolate different parts of your circuit to see the impact on the performance.

If the root cause is the wrong part value, you will need to figure out what the wrong part is, what value it is and what value you would like it to be.

Maybe you can replace it with the correct value, if we have it in stock.

Watch the skill building soldering videos on YouTube.

Read the textbook about troubleshooting to learn some of the techniques to find and verify the root cause of a problem.

After the board is working, you can move to characterizing the noise.

21.5.5 Here are some of the common errors you might encounter

- 1 . Do you see 5 V on the 328 VCC pins and on the CH340 pins? Do you see 3.3 V on the CH340 pin labeled as 3.3 V? Use a scope to measure the voltage on the pins so that you can measure the DC component AND any noise. If you do not see these voltages, it is a hard error.
- 2 . Check the solder joints of the CH340 and 328 under a microscope to make sure all the leads have a good solder connection to the board. This is one of the most common problems. Sometimes, the only way of seeing this is with the microscope.
- 3 . Do you see the 16 MHz resonator oscillations on the 328 crystal? Measure one of the xtl pins with the 10x probe and the gnd of the 10x probe connected to a gnd via. If you do not see 16 MHz oscillations, this is a problem with your 328 or the resonator connections. Check the 22 pF caps. Are they really 22 pF? If they are 22 uF, your oscillator will not work. Check the connections to the 328. You do not need a 1 Meg resistor across your resonator. If there is no 16 MHz oscillations, but the resonator connections look good, then it may mean your 328 did not bootload correctly. Try again.
- 4 . Do you see the 12 MHz crystal oscillations on the CH340 crystal? Measure one of the xtl pins with the 10x probe and the gnd of the 10x probe connected to a gnd via. If you do not see 12 MHz oscillations, this is a problem with your CH340 or the resonator connections. Check the 22 pF caps. Check the connections to the CH340.
- 5 . Are you able to boot load the 328 using the external Arduino as the ISP? If so, this suggests the 328 is working. Your 16 MHz crystal should be running AFTER bootload. It will not be running before bootload.
- 6 . When you are boot loading, measure the heartbeat signal and the error signal with a scope.
- 7 . Have you followed the boot loading directions carefully? You should have been successful when you boot loaded the 328 DIP using the external CH340 module.
- 8 . Are you able to see the CH340 on your USB port? If so, this suggests the D+ and D- to the CH340 is correct and most of the CH340 is working.
- 9 . If your computer does not see the port with the CH340, is the CH340 driver installed in your computer? Can you talk to another Arduino board?
- 10 . Carefully inspect the solder joints of the USB connector. Is there a short?

11. Is your TVS connected correctly to your D+ and D- lines and to 5 V and gnd? Check your schematic and check your layout and actual board. Is the TVS chip connected correctly?
12. There is always the possibility your TVS chip is bad and shorting the D+ or D- lines. With the board powered off, are the D+ or D- lines shorted to gnd or to power?
13. Are all the footprints in your layout the correct orientation or could any of them be mirror imaged or reverse flipped?
14. As a last resort, if you cannot get communications between your computer and the CH340, remove the TVS chip and see if your computer sees the CH340. In some cases the TVS chip has been blown by an ESD event and shorts out the D+ and D- lines. It died a glorious death protecting your board.
15. Are all of your decoupling capacitors between the VCC pin and ground, and NOT in series with the 5 V rail?
16. Are any parts missing from your board?
17. Can you communicate with your 328 using the external CH340 module you used to boot load the DIP version of the 328? You will have to turn off the CH340, otherwise its TX pin will be in parallel with the TX of the external module. You will have to figure out a way of disconnecting or powering off the TX of your CH340 on your board. This is why adding a power on switch to your CH340 chip is so important.
18. If the 328 and CH340 are working, but you cannot communicate between them, it could be the DTR pin or the TX and RX.
19. If your computer sees the 340 port, but the 328 does not respond, can you measure the reset signal on the reset pin of the 328? It should be held high and pulled down when the CH340 starts communicating on the USB port. It will pull down once and rise up with about a 200 msec time constant. Look for this using the NORMAL trigger mode to trigger the scope ONLY when the reset pin is pulled down.
20. Look at the TX and RX lines of the CH340. Do they toggle? You can measure them from the digital output pin headers.
21. Is the TX of the CH340 connected to the RX of the 328 and the RX of the CH340 connected to the TX of the 328?
22. Are there any components in the series path of the TX and RX connections. Even though many reference designs show a 1 k series resistor, it is part of the legacy design and is not needed.
23. Is the inductor LED of the TX line connected between the line and ground with a 1 k resistor? It can also go between the TX line and 5 V.
24. Is the DTR pin of the 340 connected with a 22 uF capacitor to the reset pin of the 328?
25. Is the debounce capacitor on the reset pin no larger than 1 uF?
26. Is the CH340 chip using 5 V to power itself?

21.6 Measuring the switching noise on your Arduino board and compare with the commercial board.

There are four measurements you should do and compare with the commercial Arduino board. Insert the noise shield in your Arduino and perform the following measurements. These should have been done using a commercial Arduino board in a previous lab:

1. *With three I/O switching simultaneously: pins 12, 11, 10, measure the quiet hi and quiet low noise.*
2. *Measure the voltage noise on the 5 V power rail on the board.*
3. *Using pin 7 to drive the slammer circuit, measure the voltage noise on the 5 V power rail on the board and on the quiet high pin.*
4. *Using the pin 13 as the scope trigger, use a second 10x probe as a near field pick up loop and place it under your board. Compare the near field emissions of your board and the commercial board. What is the peak noise you measure for each one?*

Be sure to comment on your interpretation of these measurements and the comparison to the commercial board. Is your noise the same, larger, or smaller than on the commercial board?

When you compare the noise, remember that it scales with dI/dt . Is the rise time of the signals on the commercial Arduino and your Arduino the same? Make note of this when comparing the switching noise.

Remember, when you are measuring the digital pins, use as short a connection on the tip of the 10x probe as practical. Use the spring tip.

21.6.1 Measure the inrush current from a USB power rail or a 5 V AC to DC converter

If you added a series resistor, you can measure the real time current in a power rail. You can see the current either during power on or during the operation of your board.

You cannot connect a 10x probe across the series sense resistor on your board. Why not? Always be aware of this limitation. So, how will you measure the voltage across the sense resistor to measure the current flow into your board?

Connect a 10x probe to the high side and a second probe to the low side of the series resistor. Set up the scope to display the voltage of each probe. They should show a very similar pattern, but offset from each other. Use the math function to display the difference. Do they have the same calibration and gain? Try connecting both probes to the compensation signal and adjust the compensation of each very slightly so that they both measure the same voltage on a zoomed in vertical scale.

The difference between the voltages on either side of the resistor is related to the current through your circuit. What is the inrush peak current? This is mostly due to the charging of the decoupling capacitors on your board. Measure this using the normal trigger on your scope when you plug power into your board for the first time.

21.6.2 Near Field Emissions

Using the method described in an earlier lab using the noise shield, measure the near field emissions from the bottom of your board while it is switching. Compare this to the near field emissions from the commercial board.

Take a picture of your measurement set up to include in your final report.

21.6.3 Your report for bring up and test

There is no separate report required for the bring up and test of your brd 3. Instead, include the results of your testing in your final report.

21.7 Step 3: Final report- counts as a midterm

Remember, your final report will look great on your portfolio page. Write your report as though a perspective employer were to read it- as they most likely will!

The last step of your design assignment, which counts as your final, is to document what you have done on this project.

Review the seven steps in the project.

Emphasize what you did that you think worked, what you did that was a mistake and how you would do it differently next time.

Discuss what you learned in this design that you will use in future designs.

Discuss what you learned from this project that you will do differently in future projects.

Take pictures along the way.

Demonstrate best measurement practices.

Show the difference in measured noise between your board and the commercial board and some of your design features that contributed to this difference. Show pictures.

Take pictures of scope traces that are relevant. If you don't include any analysis of a scope trace, don't bother including the scope trace.

21.7.1 Grading rubric

Your final will be graded based on a total score of 10 points. Your final grade is based on three sections:

Section 1: 2 points possible for your schematic and layout design.

Include your POR. What did you do for risk reduction?

What did you do in the schematic and layout that worked well?

What did you do that did not work well- hard errors that were fixed or software errors?

What did you learn from this board design that you will do differently in your next design?

1 point off for each of the following features in your design:

- *copper fill on top layer*
- *no ground plane on bottom*
- *long gaps in the ground plane without ground straps over gaps*
- *high inductance routing to decoupling caps*
- *no name on your board*
- *lacking labels for pins*
- *lacking labeling for parts*
- *incorrect footprint for 6, 8, 10 pin header sockets*
- *no indicator LEDs*
- *late report (1 point off per day late)*

For graduate students:

One point off per feature missing:

Correct ferrite filter to the AVCC pin

Current sense resistor in series with the power line

Alternative power from USB or power jack

Additional rows of gnd pins outside the regular header pins

Section 2: Functioning of your board. 3 points possible:

3 points if you were able to boot load and run a modified blink on your Arduino board

If your board does not boot, then:

1 point if you have the correct power appearing on the various Vcc pins

1 point for showing some evidence your CH340g or Atmega328 chips are working- like crystal activity, RX, Tx lights, port is seen by your computer

1 point if you did not get the boot loading working but figured out the root cause and can demonstrate some evidence of it and what you would do differently to make it work.

Section 3. Performance of your board. 5 points total

Show and discuss the measurements of

the switching noise and the quiet LOW pins

the quiet HIGH and board level power rail noise

the power rail noise on the board and the quiet HIGH pin when the slammer circuit is on the near field emissions from under the board

Compare the measurements of your board and the commercial board.

Show scope traces and your analysis of the difference in the noise between these two boards.

What features of your board contribute to the differences in the noise?

If you are not able to get your board to boot load and cannot measure the switching noise, you can use a board I provide you as your reference design. If you complete the measurements and analysis, you can still receive full credit for this section.

Chapter 22 Lab 10: boot load a 328 uC

Your Board 3 is an Arduino board, using the same circuit as the commercial versions, but with your layout.

At the heart is the Atmega 328 uC chip. When it comes from the vendor, this chip is brain dead. You have to burn the bootloader program to enable it to understand USB speak. You can read this article for some background: <https://www.instructables.com/Bootload-an-ATmega328/>

In this lab, you will bootloader a new 328, in the form of a DIP package. You will then use the CH340g USB to UART chip to interface to the 328.

Step 1 is make sure you have all the parts

Step 2 is you will burn the bootloader into the 328 using another Arduino as the in-circuit serial programmer (ISP)

Step 3 is you will communicate between your new 328 uC and the USB with a CH340g USB to UART translator module.

Step 4 is you will write the code in the IDE to make the digital pin 13 blink like a heartbeat

You will receive a 328 DIP package and a CH340g module from your TA. You will solder the header pins in the 340 module using your best soldering practices. Be sure to [view the videos on soldering](#) and practice soldering header pins on the PCB BEFORE you solder the pins on the CH340 module.

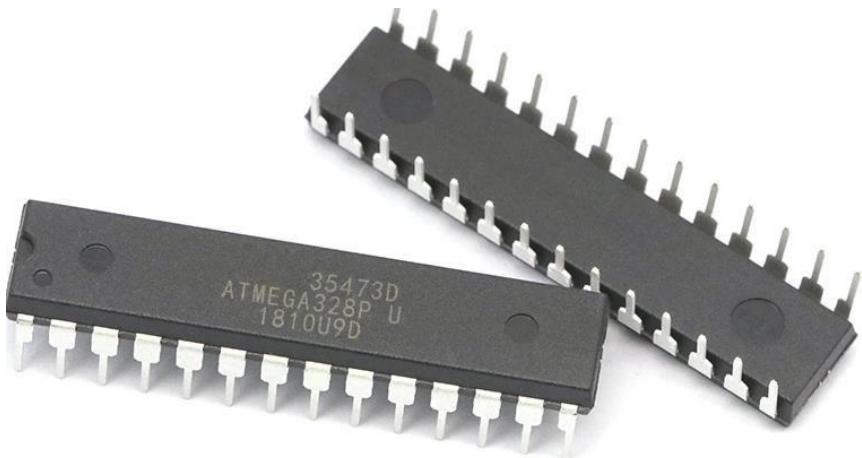
By the end of this workshop, you should have a fully functional Arduino Uno built on a breadboard and understand the necessary connections and components that allow it to be programmed via USB.

To demonstrate that your 328 is alive, program it to flash pin 13 with a heart beat: thump thump...thump thump...

Have it blink a red LED in this pattern.

22.1 The 328 uC

You will receive a 328 uC in the form of a DIP, as shown below.

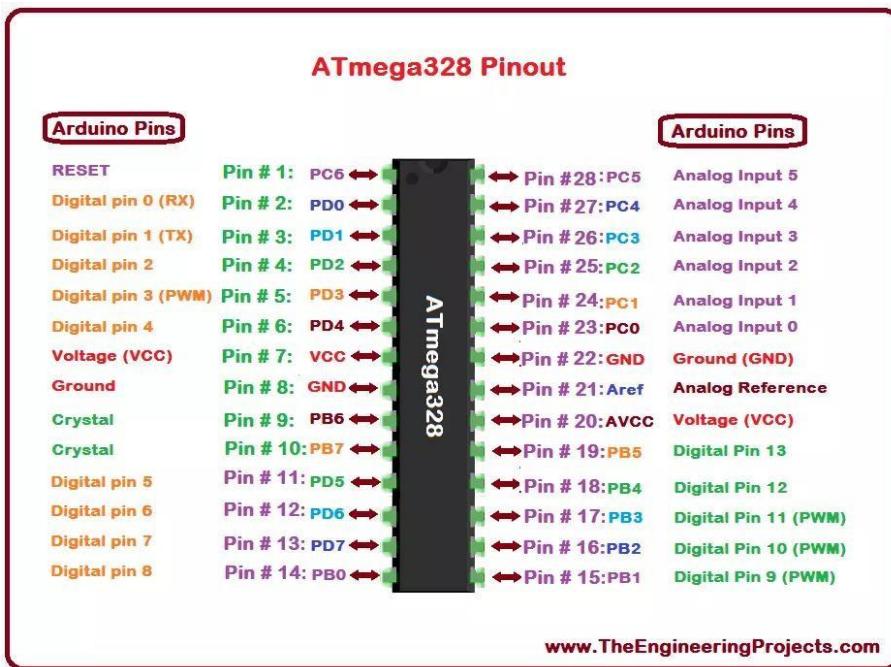


You will use a solderless breadboard to do all the assembly of the 328, the CH340 and all the needed peripherals. You will burn a bootloader to factory-new chip, connect it to a USB to TTL converter module, and program it to blink an LED.

(PCINT14/RESET) PC6	1	28	□ PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	□ PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	□ PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	□ PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	□ PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	□ PC0 (ADC0/PCINT8)
VCC	7	22	□ GND
GND	8	21	□ AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	□ AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	□ PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	□ PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	□ PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	□ PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	□ PB1 (OC1A/PCINT1)

Pinout of ATmega328p. Note the multiple functions that each pin has.

Here is another pinout that describes all the pins on the 328 28 pin DIP package:



You will need the following parts for this lab:

- 328 uC ATmega328 IC ([Datasheet Here](#))
- CH340g module with header pins soldered on ([here is the datasheet](#))
- A USB to micro cable to plug into the CH340 module
- Solderless breadboard
- 16 MHz crystal, with leaded pins
- 2, 22 pF capacitors.
- A 10k resistor for the pull up on the reset pin
- A functional Arduino Uno board as the ISP
- 6 jumper wires for the ISP burning of your 328

22.2 Solder the header pins to the CH340g to UART interface board

Before you begin this lab, you will need to solder the header pins onto the USB to UART board. You might be lucky and be assigned a UART from a previous semester which will have already been soldered.

You should have already viewed the [skill building workshop videos on best soldering practices](#). You should have applied these practices to soldering header pins to the small test boards and have been check off by your TA.

If you have not completed these tasks, you should do so before you start this lab.

22.3 Burn the bootloader

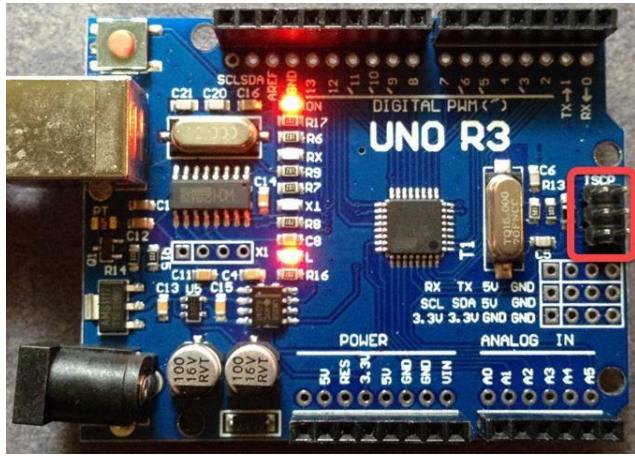
The Atmega 328 comes up dumb. The first step in using the 328 is to turn the 328 into an Arduino. You must teach it how to interpret the UART signals from the CH340g into Arduino IDE speak. This is by burning the boot loader code into the 328.

The program we add to the Atmega 328 is the bootload program. It is like the bios in a microprocessor. It is the first thing the microcontroller runs when it turns on. We need to give it the instructions on how to listen to the UART data lines and how to interpret the commands from the Arduino IDE compiler.

After we burn the bootloader in the ATmega328, your board becomes an Arduino.

The simplest guide to burn the bootloader is on the [Arduino.cc web site](#), and on the [Sparkfun web site](#).

You will use a standard commercial Arduino Uno board as the *programmer*. It will act as the In-circuit Serial Programmer (ISP). It will generate the code and drive the In-Circuit Serial Programmer (ICSP) pins on the *target* board, your Atmega 328, which will turn it into an Uno after the boot code is loaded. The ICSP header on an Uno is shown in the figure below.

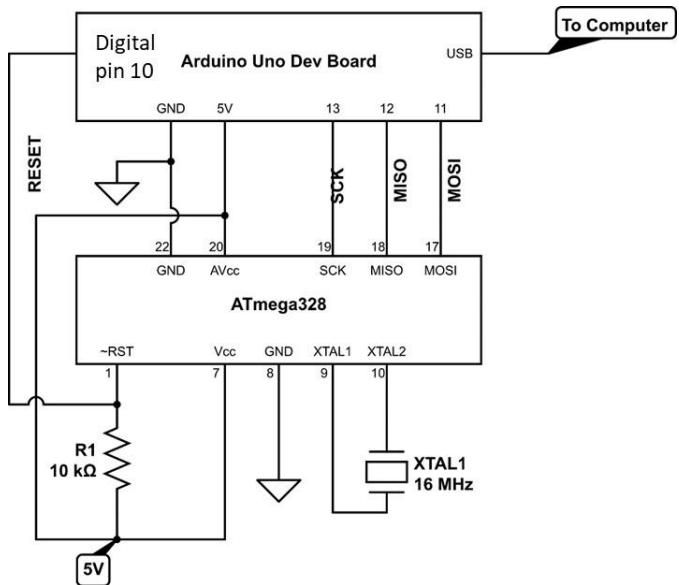


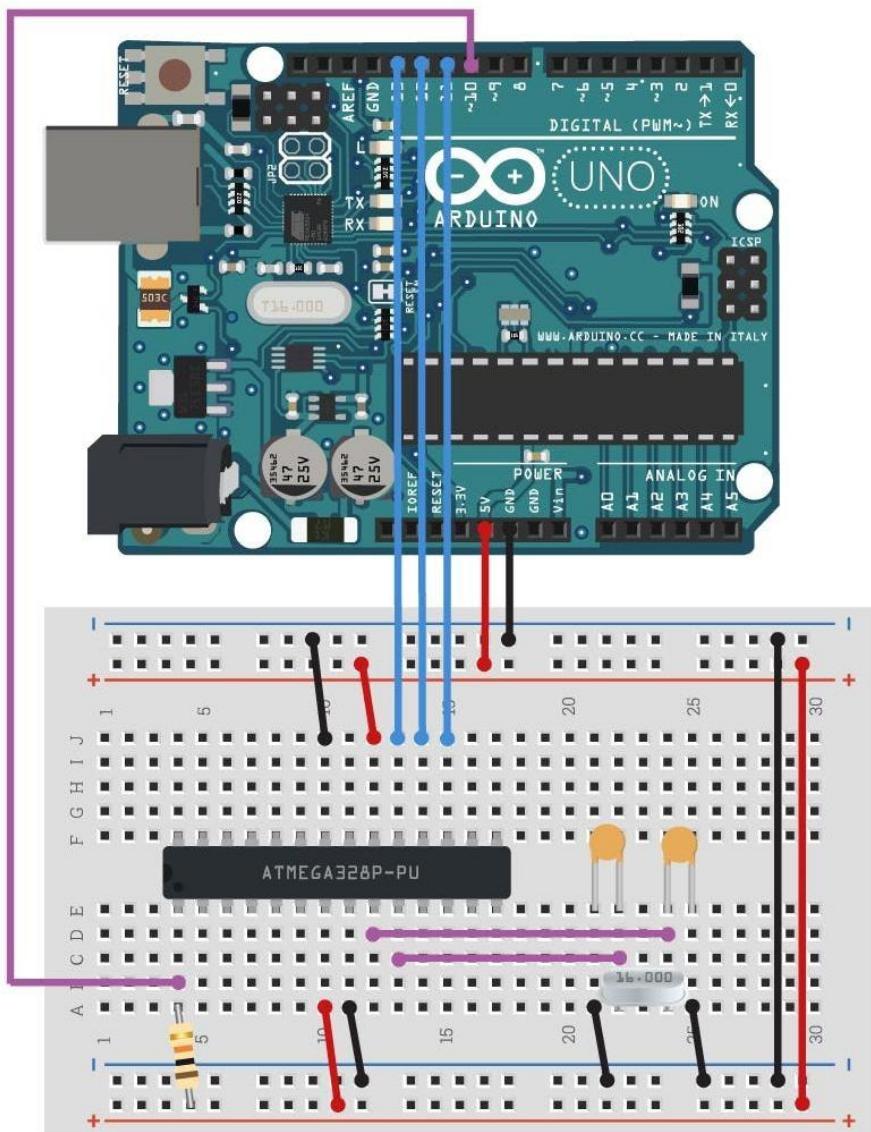
On your 328 in a dip package in your solderless breadboard, the Target board, you can connect wires directly to the pins in the solderless breadboard and the header pins on the Uno or its ISCP pins.

Regardless of the method you choose, you will need 6 jumper wires.

We will use an Arduino Uno as the in-system programmer (ISP) to burn the bootloader code in your Atmega 328 microcontroller DIP package. The Uno is the programmer, your Atmega 328 board is the target.

This is the same process you will use when you program your own board 3 golden Arduino board. It's just that your 328 will already be mounted to your board.





Note: in this example, the crystal is placed farther away from the two xtl pins on the 328 than you should use I your board. You should plug the 16 MHz crystal directly into the holes of the XTL1 and XTL2 pins of the 328.

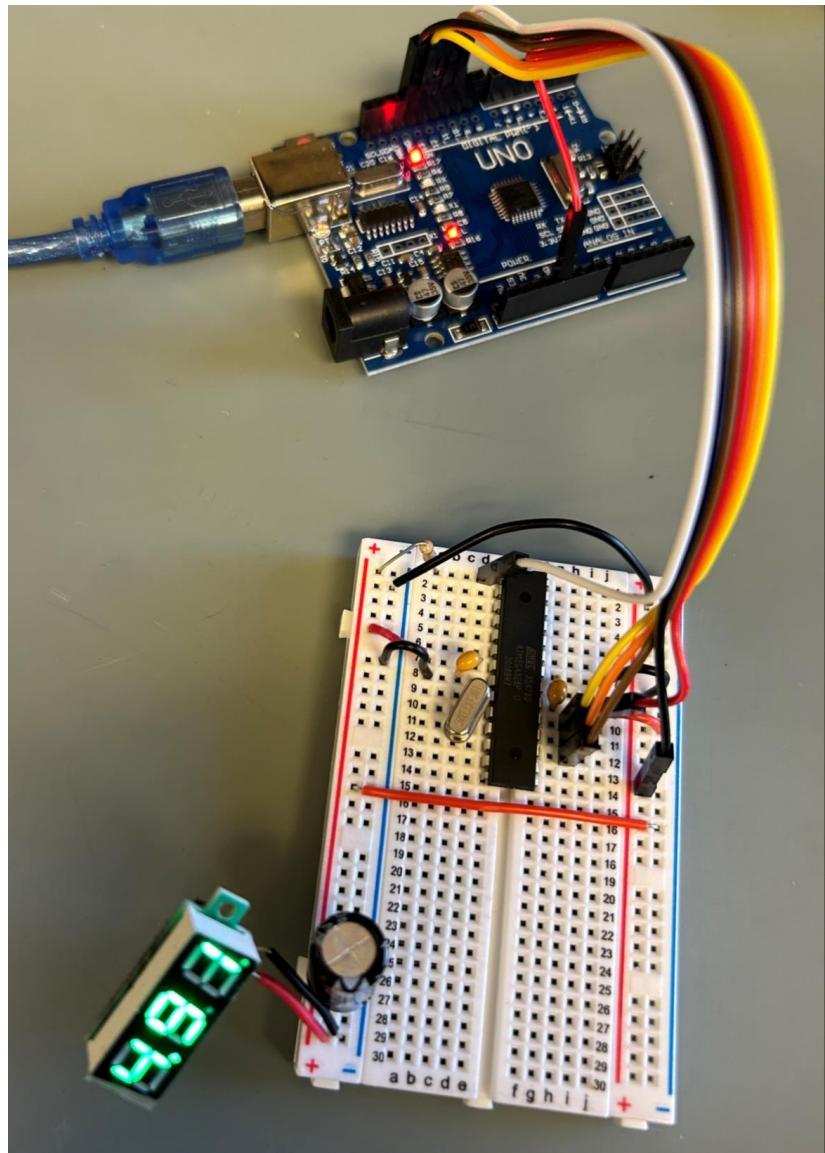
Just plug the crystal into the pins 9 and 10 of the 328. Do not use long leads. You probably do not need the 2, 22 pF capacitors. Try not including them. If you are unable to get the 328 to bootload and do not see a signal on the XTLs, then add them in. I did not need them on my solderless breadboard.

You are powering the 328 DIP chip using the power from the Arduino ISP board.

The RST pin on the Arduino as ISP is really pin 10, one of the digital pins. It is connected to the RESET pin of your 328 DIP package. Continue to follow the guide until you have successfully burned a bootloader to the 328. From this point, you can upload code to the 328 using the CH340 USB to UART chip, described in the next section.

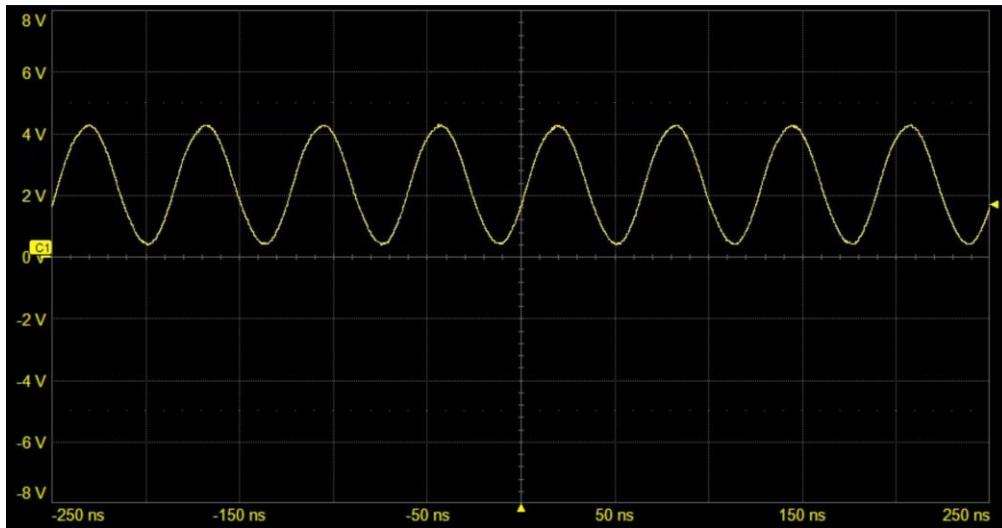
In my board, shown below, I added a few features. I route power on the outside columns and gnd on the inside columns. I routed the gnd wire from the digital pin side along with all the digital pins. The digital pins will be switching. It is more important the digital pins have their return path in close proximity.

I added a voltmeter to make sure I get the 5 V rail connected. I also added a 1000 uF cap on the 5 V rail and a 10 uF cap right on the Vcc and gnd pins of the DIP. Why not? If it is free, adding a low inductance decoupling capacitor close to the pins of the IC is always a good thing to do.



Even before you have booted loaded the 328, you will not see the crystal resonating at 16 MHz. If your 328 shows the 16 MHz signal on one of the XTL pins, then it is because you have a 328 that had been previously booted loaded. That is ok. Proceed anyway.

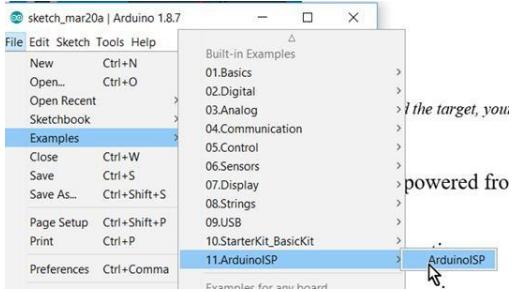
After bootloading, the 16 MHz crystal will oscillate and you will definitely see it on the XTL pin. Look at the voltage on one of the xtl pins with your 10x probe with the gnd of the 10x probe connected to ground. I measured a 16 MHz oscillation on my xtl1 pin, with no 22 pF capacitors. Here is what I saw on my scope:



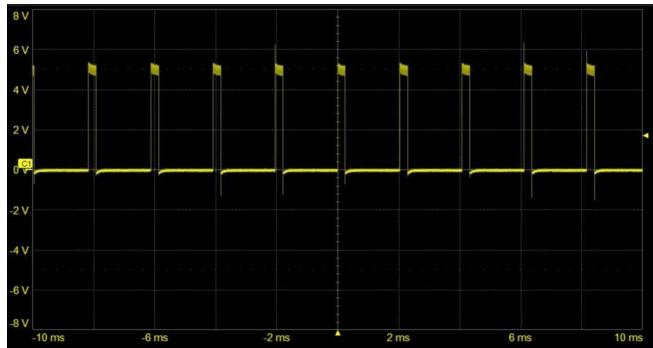
Pretty nice looking 16 MHz oscillation.

Once the two boards are wired up, the basic process is:

1. Open the *ArduinoISP* sketch (in Examples) to your Arduino Uno board. It is the item #11 in the files/examples pull down, as shown in the figure. This has the firmware we will burn into the bootloader of the target, your Atmega328.

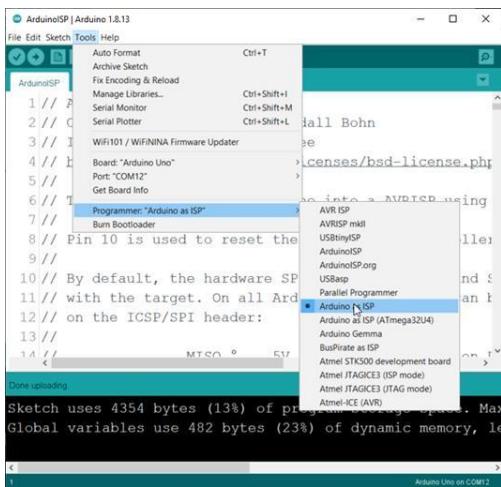


2. You will need to set up your Uno to talk to your USB port, just like any sketch. Select the items in the Tools > Board and Serial Port menus that correspond to the board you are using as the programmer, the Uno (not your target board).
3. At this point, you can upload the sketch to your Arduino Uno ISP.
4. Upload the ArduinoISP sketch to the Uno that is the programmer. Your Uno is now an Arduino ISP. This code that has turned your Uno into an Arduino ISP is now running on your Uno.
5. If you get an error when uploading the ISP bootloader code to the Arduino uno, like "avrdude: stk500_disable(): protocol error, expect=0x14, resp=0xe0", ignore this and continue.
6. This ISP sketch uses three digital pins as diagnostics:
 - ✓ pin 9: Heartbeat - shows the programmer is running
 - ✓ pin 8: Error - Lights up if something goes wrong (use red if that makes sense)
 - ✓ pin 7: Programming - In communication with the slave
 - ✓ Rather than add a resistor with an LED to each pin as a diagnostic, we will use 2 channels of a scope. You can use 2, 10x probes plugged into the UNO acting as the ISP. Note: since we only want to see high level signals, we can get away with the large loop inductance of the ground lead. Use a short wire stuck into pins 7 and 9 to the 10x probe tips, and a wire in one of the ground pins for the ground lead. Try to keep the wires short.
 - ✓ Set up your scope to read the 5 V signals from each channel. Use pin 9, the heartbeat to trigger the scope. You might see a 500 Hz PWM signal from this pin. Here is a snapshot of the heartbeat I saw on my scope:



This just means the boot loader code is running correctly on your Uno.

7. Make sure your Arduino Uno programmer board is wired correctly to your target Atmega328 board, your board. At this point, the programmer Uno board will be powering your board.
8. Now we begin the process of burning the bootloader code into your board. Select the item in the Tools > Board menu that corresponds to the target board which is an Arduino.
9. Select the "Arduino as ISP" in the Tools>Programmer menu, as shown in the figure.



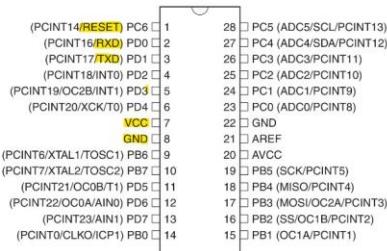
10. As the final step, select the Burn Bootloader command, also on the File/Tools menu, as the bottom option. This step will actually burn the firmware into your ATmega328. After this step is complete, you are done. Your ATmega328 should now be an Arduino Uno board.

11. Remove all the connections from your new Arduino board. At this point your ATmega328 should now be an Arduino and understand the Arduino IDE commands over the USB.

The next step is to try communicating to the RX and TX UART pins on your 328 with the CH340 interface chip.

22.4 Step 3: USB Programming

Now that the bootloader is burned onto the 328 (effectively turning it into an Arduino Uno), it is time to figure out how to program over USB. The Arduino does not natively support USB but does support programming over the UART bus which is what we will leverage using the CH340G module. The CH340 is a family of chips that converts USB to UART. This is often accomplished by FTDI chips or some chips by Silicon Labs. The CH340, however, is a cheap Chinese alternative. You will be using five pins on the 328 to accomplish the connections to the CH340. Below is the pinout with these connections highlighted.



As shown in the pinout, the 328 makes use of the TX, RX, and Reset lines. The TX line is responsible for transmitting data out from the 328 while RX is used to receiving data into the 328. This doesn't work for programming unless the Reset line is appropriately used.

The most common mistake in using a UART interface is that each of the RX and TX lines are one-directional. This means the TX of one device connects to the RX of the other device. DO NOT CONNECT RX of one device to RX of the other device.

The Reset line, when used in conjunction with the UART bus, is responsible for instructing the MCU to listen for new code to be flashed. If any signals are incorrectly handled, then the USB programming will not work.

The reset pin on the 328 should be pulled high with a 10k resistor. A roughly 1-10 uF capacitor should connect in series between the DTR pin of the CH340 and the reset pin of the 328.

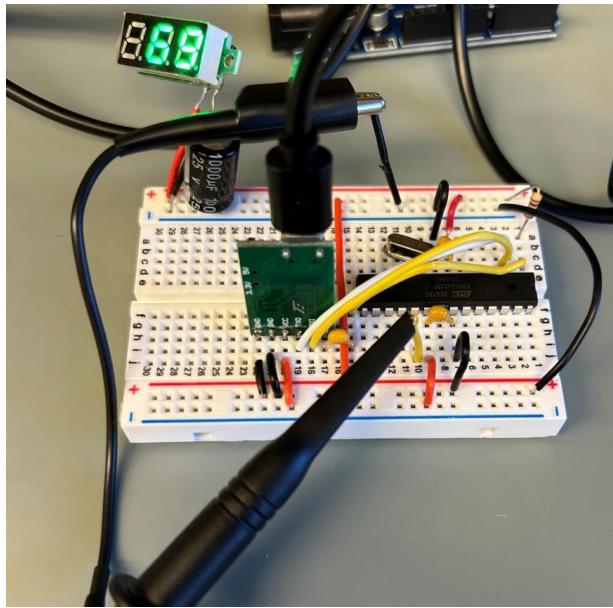
The TX of the CH340 goes to the RX of the 328

The RX of the CH340 goes to the TX of the 328

The 328 is powered by 5 V which is on the 340 module. Do not apply external power to the 328.

Be sure to connect power and ground to the 328 from the 340 module.

Here is my CH340 connected to my bootloaded 328:

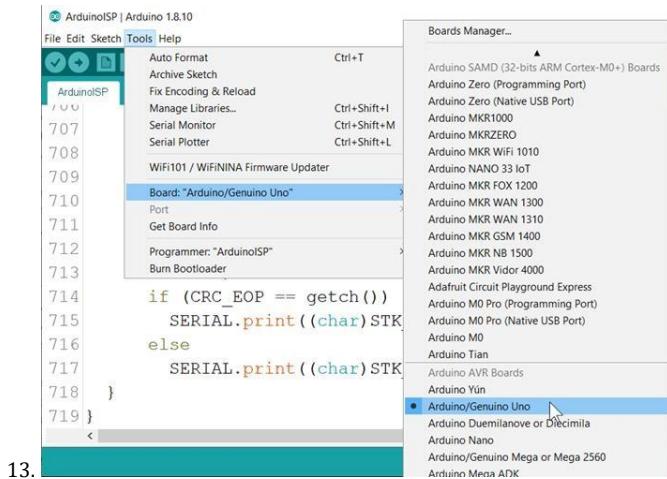


After your 328 has been bootloaded with the other Arduino as ISP, you should be able to flash programs to the 328 over USB using the CH340 module. After that, you have successfully made a custom Arduino Uno on a breadboard.

If you are unable to communicate with the CH340 chip with your computer, [check out this article](#) for information about the drivers.

Once you have communications between the USB port of your computer and the CH340 and are able to program the 328, look at the TX and RX signals on the 328 while there is communication. Use a 10x probe and your scope to capture an example of the data pattern on one of the lines.

12. To test your new Arduino board, connect it to your computer with a USB cable. In the Arduino IDE, be sure to select as your board, an **Arduino Uno** as shown in figure, when you try to program your board over the USB connection. And select its correct com port.

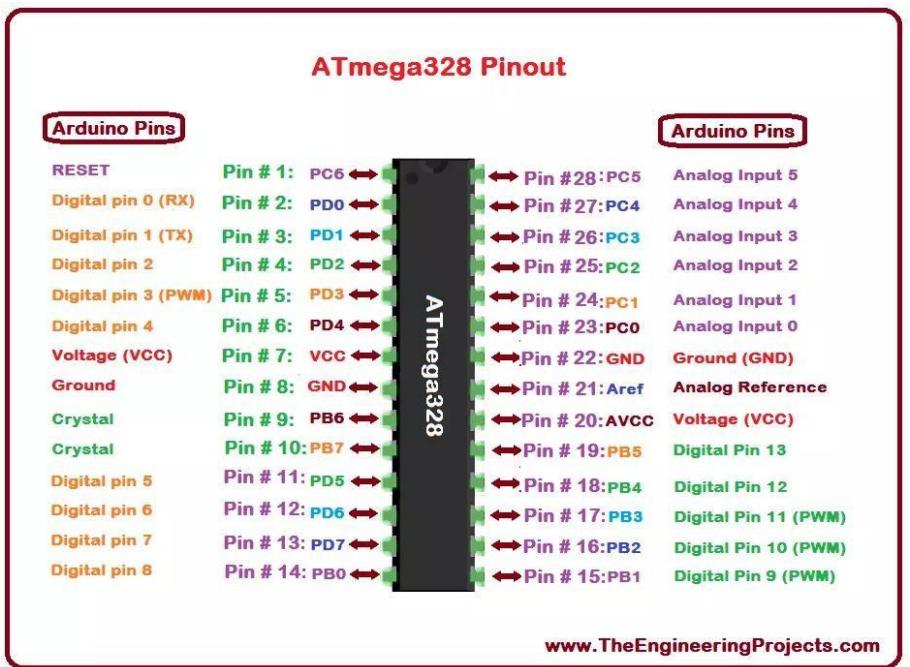


13.

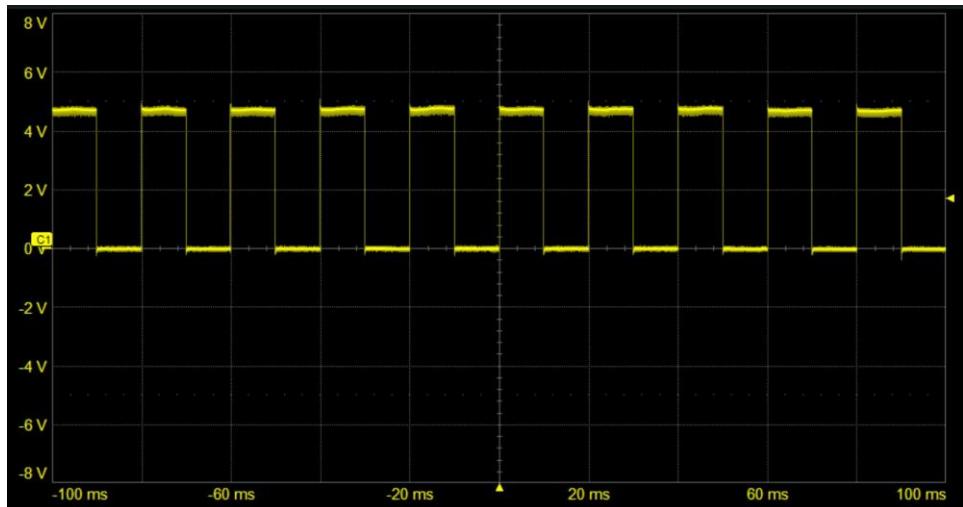
14. Upload Blink, change the delay to 100 and see if it blinks quickly. If it does, success!

Pin 13 is When your board is boot loaded and it runs the modified Blink code, your board is “working”

Recall the pinout of the 328, shown below:



Digital pin 13 is actually pin 19 on the 328 DIP package. Here is my pin 13 output running with a fast blink, 10 msec on and 10 msec off:



Success!!

22.5 Step 4: Confirmation of successful bootloading

As a final test, program your 328 to pulse pin 13 with a heart beat signal. Add an LED with a 300 ohm series resistor to have it flash in the pattern of a heart beat. This is a thump thump....thump thump....thump thump....Show this to your TA for final check off.

When you are done with this lab, you can keep the 328 and CH340 module in your small components kit.

Be sure to turn in any other older boards you still have.

22.6 Grading rubric

This lab is worth 3 points.

1 point for check off by your TA

1 point if your lab report describes the general process you went through to bootload your 328. Be sure to include pictures.

1 point if you are able to include an example of the TX or RX signals between the CH340 and 328.

Chapter 23 Lab 11: measure trace resistance

23.1 Before you come to the lab

You should have read chapters 5 and 6 in the textbook. You should be familiar with the 2-wire and 4-wire method of measuring resistance.

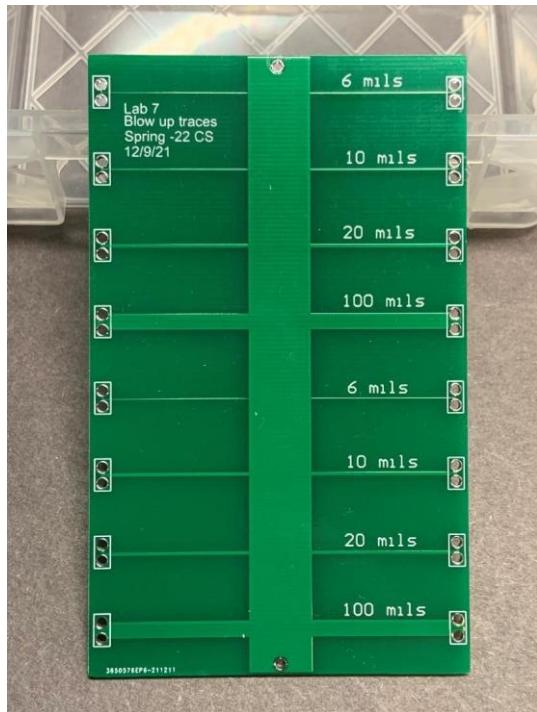
23.2 What you will do in this lab

In this lab, you will learn basic Best Measurement Practices for:

- *DMM measurements of 2-wire resistance*
- *4-wire resistance measurements*
- *A constant voltage and constant current power supply*
- *Estimating the resistance of traces*
- *Measuring the resistance of traces to compare with your expectations*

23.1 Exp 1: Analyze the Test Board

The test board we will use is a simple 2-layer board with similar patterns on the left and righthand sides. One like it is shown below.



You should be able to verify the connections with a DMM testing connectivity. This is called *reverse engineering* and is an incredibly valuable skill.

You perform measurements to deduce a model of what might be going on inside the DUT that is consistent with all your observations. Take every opportunity to practice reverse engineering using all the measurements and consistency tests you can think of doing.

There is a long, wide bar in the center of the board with two contact points on either end. There are two contact pads on either end of the 1 inch long test lines.

You will be measuring the resistance of a few of the 1 inch long test lines using both the 2-wire and 4-wire method.

23.1 Exp 2: Estimate the resistance of each trace

Remember Rule #9. Before you do a measurement anticipate what you expect to see.

Before you do any measurements, estimate the resistance of each of the traces. It is literally as easy as counting squares and multiplying the number of squares by your estimate of the sheet resistance.

Assume the copper is 1 oz copper. The traces are under the solder mask so are probably not plated up and do not have a HASL coating.

The trace widths are 6 mils, 8 mils, 10 mils, 20 mils, 100 mils.

The trace lengths are all 1 inch.

Estimate the resistance you expect each trace to be. DO NOT GOOGLE this answer or use an online calculator. The whole purpose of this lab is to learn how to do this estimate yourself. It is incredibly easy if you understand the principle of sheet resistance.

Create a table to record your estimate and the 2-wire and 4-wire resistance measurements. Add your estimates to your table. Your table can be something like this:

Line width	Estimate	2-wire	4-wire with 1 A current
6 mil			
10 mil			
20 mil			
100 mil			

You must complete your estimates BEFORE you perform measurements. Otherwise, how will you know what to expect? This is the essence of Rule #9.

[Watch this video and I walk you thru estimating the resistance of a trace.](#)

23.2 Exp 3: Measure the resistance of each trace using a 2-wire DMM

Use a handheld or a bench top DMM to measure the 2-wire resistance of each trace and add these values to your table.

Make note of which DMM you used. Where will you make the connections to the ends of the trace to measure the 2-wire resistance? The central bus trace is 400 mils wide. In the worst case, when a section of it is part of the 2-wire resistance measurement, what would you estimate to be its resistance contribution to the 2-wire trace resistance measurement?

Always answer the “so what?” question. You just did an estimate and a measurement. So what? How do you interpret your ability to accurately predict the trace resistance and measure it using the 2-wire method? How close is your agreement?

If the values are off, can you offer any comment as to why? Just saying “measurement error” without an estimate of the source of the measurement error and an estimate of its magnitude, is an empty statement.

23.3 Exp 4: Measure the resistance of each trace using the 4-wire method

For the 4-wire method, you will need the constant current power supply and the cables to connect the hi current and lo current ends to each trace.

[Watch this video about how to use the power supply in the constant voltage or constant current mode.](#)

Before you connect your test board to the power supply, practice using the power supply in both modes, CC and CV. Be sure to set the maximum current to less than 1 A.

NOTE: the default max current is 10 A. If you forget to adjust the current and use the 10 A current limit and the output voltage is set for 5 V, and then connect the wires and turn on the power supply, you will blow up the traces! Do not do this more than once. (wear safety glasses!!)

Use a current setting of 1 A or below. How will you make the connections to the ends of your 1 inch long test wire to force a 1 A current through the entire trace? Where will you connect to the trace to pick off the voltage drop across it?

Use the special banana to mini grabber cables in your kit. They are designed of high current. The cables hanging on the racks in the lab are not.

Connect the current leads so that you force the 1 A current through your trace.

Use two other leads and two other connections to a DMM to measure the voltage across the trace. Even though the power supply has a voltmeter in it, do not use it to measure the voltage across the trace. Why is this not a good idea?

Does it matter if the current and voltage probe points are exchanged?

[Watch this video and I walk you through the principles of the 4-wire measurement method.](#)

Your TA will ask you to trace the current path through the conductors and identify the location at which the voltage is measured and how you interpret the voltage measurement.

From the current through the trace and the voltage across it, what do you calculate as the 4-wire resistance of the trace? Add this value to your table. How does it compare to the estimate and the 2-wire measurement? Be prepared to answer the so what question. Why do you see the results you are seeing?

Very important! Perform all the estimates and calculations AS YOU TAKE THE MEASUREMENTS. Do not wait until after the get home to do the calculations. How will you know what to expect unless you do the estimates in real time? If you wait, and you get home and do the calculations and they are way off, you do not have a chance to redo a measurement or check a result.

Fill out your table for each trace line width, the estimate, the 2-wire measurement and the 4-wire measurement.

How do you interpret the result?

Stretch exercise: using the 4-port method, measure the resistance of other conductors:

- *A long column in the solderless breadboard*
- *A 6 inch length of jumper wire*
- *A 6 inch length of AWG 24 wire*
- *A paper clip*
- *One of the banana to mini grabber leads*

23.4 Grading rubric

This lab report will be combined with the lab report for blowing up traces. There will be one lab report for both labs.

Chapter 24 Lab 12: blow up traces

In this lab, you will measure the maximum current carrying capacity of narrow traces by putting too much current through them and having them blow up.

Very important! Because there is a chance some traces may explode, EVERYONE should wear safety glasses for the entire lab. No exceptions.

24.1 Exp 1: estimate the max current handling capacity of a trace.

Before you blow any trace up, use the calculator in [the Saturn PCB tool](#), as described in the textbook, to estimate the maximum current for the 6 mil wide trace and the 20 mil wide trace. Make a note of these currents, before the trace temperature exceeds 40 deg C temperature rise over ambient.

Now you will have an idea of what maximum current to expect to see before the traces get warm to the touch.

24.2 Exp 2: increase the current through a trace and feel it get warm

We can use the power supply to force a current through a trace and a voltmeter to measure the voltage across the trace. Use the connections based on a 4-wire measurement, as described in the textbook. This will enable you to measure the resistance of the trace using the 4-wire method while you are blowing it up, at the same time we are forcing a large current through it.

[Watch this video and I will walk you through driving current through a trace to blow it up.](#)

Note that for copper, the resistivity increases with temperature. Its temperature coefficient of resistance, TCR, is 0.4% per degC. As the temperature of the copper trace increases, its resistance will increase. View the video about thermal run away in the textbook.

While we increase the current through the trace, we will touch the trace to feel when it starts getting warm. We are looking for the current to create three temperature levels:

- *noticeably warm to the touch*
- *hot to the touch*
- *smoking*

Connect up the power supply, using a max current setting of 1 A. The current is forced from one end to another end of the trace. Follow the path the current is taking between each contact point. How does the current change throughout the circuit, especially when the interconnect width changes?

Connect to the 6 mil trace and crank up the current and feel the trace.

1. *At what current does it get noticeably warm?*
2. *At what current does it get hot to the touch?*
3. *At what current does it begin to smoke?*

Once it smokes a little, please shut the current off. The smoke detectors in the lab will go off if everyone generates a lot of smoke.

Make sure the power supply is turned off.

What do you conclude from this lab as the maximum current capacity for a 6 mil and 20 mil wide trace? How does your estimate of what you measured compare with the IPC recommendations?

24.3 What you will turn in or complete for this lab

Before you complete this lab, be sure to be checked off by your TA. They will ask you questions such as:

1. *What is the connection to each trace on your board? How do you know this?*
2. *When set up for 4-wire measurements, what is the current path and what is the voltage path?*
3. *What is the difference between a 2-wire and 4-wire resistance measurement?*
4. *When is it better to use a 4-wire measurement?*
5. *How does a power supply switch between constant current or constant voltage?*
6. *How did you estimate the resistance of the 6 mil wide trace?*
7. *How did you estimate the maximum current the IPC recommends through the 20 mil wide trace?*
8. *How did you set up the constant current through your 20 mil wide trace?*
9. *When did the 6 mil wide trace start to smoke?*
10. *When you are increasing the current, what mode was the power supply in?*
11. *When you were putting a constant current into the trace, you saw the voltage across the trace increase a little bit- why do you think this was?*
12. *At what current do the wires connecting the traces to the power supply start to feel warm?*
13. *How did your estimates based on the IPC spec for the max current compare with your measurements?*

14. What is the difference between constant voltage mode and constant current mode?
15. What is the max current you can output in this power supply?
16. How do you switch between constant current and constant voltage mode?
17. What could happen if the power supply were connected across a very thin conductor and you turned the output on with the default values of 6 V and 10 A as the set points?
18. What is the output impedance of an ideal voltage source if its output voltage is always constant, independent of the current drawn?
19. How do you change the output current of the power supply?
20. What is the difference between the set point current and the actual current the power supply is outputting?

24.4 In your lab report

As always, write your lab report so it will look great on your portfolio page. These experiments and your observations are gold to a hiring manager. What other candidates know how to use the 4-wire resistance method or how much current it will take to blow up a trace?

Use 1 page to describe the 2-wire and 4-wire resistance method, the description of your table and what you conclude about the ability to predict the resistance of a trace. If you did not get good agreement, try to evaluate an explanation. Just saying measurement accuracy is not enough. Where are the uncertainties coming from? Copper resistivity? Trace width? Copper thickness? Measurement artifact?

Use 1 page to write up your experiment measuring the max current before the wire gets hot or smokes. Pictures would be very useful.

Add your recommendation for the max current you would feel safe putting through each trace and why.

Again, keep in mind this report will look great on your portfolio.

24.5 Grading rubric:

1 point for check off by your TA

1 point if you included data with an explanation in your report

1 point if you have a coherent explanation for the resistance measurements and the max current capacity of a trace.

Chapter 25 Lab 13: Build a crystal oscillator in a solderless breadboard

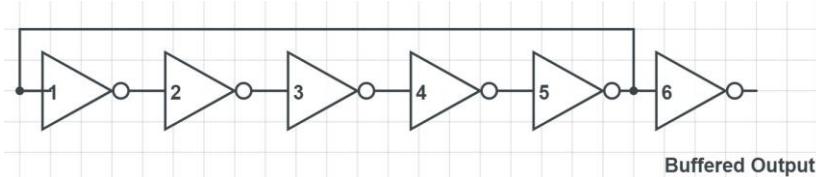
A crystal is a component made of quartz which will ring at a very fixed frequency when excited by a voltage. Using positive feedback, we can set the circuit into oscillation at the resonant frequency of the crystal. Newer circuits also use a ceramic resonator. These are smaller size than quartz crystals but behave the same way.

25.1 Exp 1: A ring oscillator

In lab 7, you built our own ring oscillator based on the chaining the outputs of the hex inverter chip. The speed of the oscillator is based on the individual hex inverter propagation delay. When connecting all six hex invertors in a daisy chain (series) array, you should measure a frequency of about 20-50 MHz, depending on specific IC you have on your board.

Build a ring oscillator circuit with a hex inverter. You should have an SN74AHC14 hex inverter in your kit as a 14 pin DIP package. Add a decoupling capacitor between the 5 V power and gnd connections. 1 uF or larger should be fine. Since these are through hole, leaded parts, the decoupling capacitor will be most effective using short leads, and preferably a small MLCC yellow capacitor.

Connect the output of the ring oscillator to another inverter used as a buffer. Is there a difference in the output waveforms from these two inverters? Here is a ring oscillator with a buffer in the figure below:



Remember you want to use an odd number of inverters so it self-oscillates. You also want to buffer the output with the last inverter. Again you should measure a frequency of about 20-50 MHz, depending on specific IC you have on your board. Remember, the oscillation frequency will be about $1/(2 \times \text{prop delay} \times 5)$.

You can power the device at either 5 V or 3.3 V.

25.2 Experiment 2: Adding a 1 Meg resistor in the feedback loop

The input impedance of the CMOS inverter inside the hex inverter is very high. It does not take much current at the input to switch it from a high to a low state or low to high state. When there is a direct short between the input and output, it will definitely oscillate.

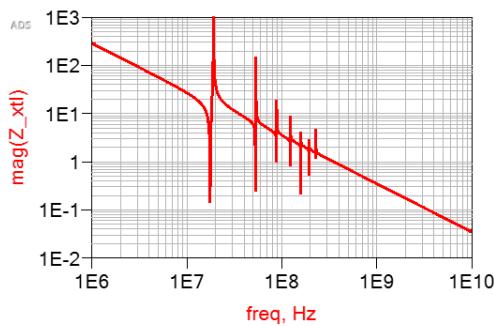
Add a resistor to the feedback loop. This will cut down the input current from the feedback. What is the largest value resistor you have in your kit? Connect it in feedback instead of the short. Does the frequency of the ring oscillator change?

In a CMOS inverter, the input impedance of the inverter is very high. A 1 Meg resistor in the feedback loop is still a low impedance compared to the input resistance of the gate of the inverter input. A CMOS inverter will probably self oscillate with a 1 Meg resistor in the feedback. A bipolar inverter may not. A bipolar inverter is not a suitable inverter to use with a crystal.

The frequency of the ring oscillator may slow down. You have built an RC oscillator. There is some input capacitance of the 7414. It may be on the order of 10 pF. The RC time constant is $1 \text{ Meg} \times 10 \text{ pF} = 10 \text{ usec}$. It may oscillate at about 100 kHz.

25.3 How a crystal works

A crystal is a slice of quartz between two electrodes. It is piezo electric. A voltage across the end faces causes a mechanical compression and a mechanical compression of the crystal causes a voltage across its ends. Due to this coupling of mechanical motion and electrical signal, it has an impedance which dips to a low value at the resonant frequencies for which it will vibrate. The figure below is an example of the impedance profile of a typical crystal.



Note that this crystal, as is typical, has multiple resonances above the first harmonic.

While we are only showing the impedance profile, a crystal is not a passive component. It generates a voltage when it is mechanically stressed. At the resonant frequency, the combination of the impedance and voltage response generates a low impedance.

When the crystal is in the feedback loop of a high gain inverting amplifier, the frequency where the impedance is lowest, ie, where there is more feedback, and at which there is a high voltage generated by the compression of the crystal, will be the frequency of oscillation. At the frequency of lowest impedance, there is more current that flows through the crystal providing the feedback current to switch an inverter.

When a crystal is added between the input and output of an inverter, the inverter will resonate at one of the frequencies where the impedance is lowest. To select the lowest frequency, we need to add a filter to remove the higher frequency components.

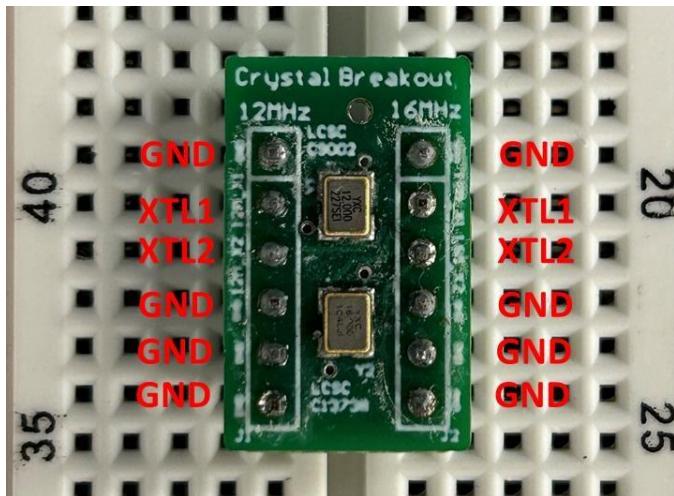
Sometimes, the inverter will not start into self-oscillation when the crystal is added. Sometimes, we have to add a resistor in feedback to get it to self-oscillate.

25.4 Experiment 3: Measure the impedance of the crystal or ceramic resonator

Generally, to measure the impedance of a crystal is difficult. It requires a network analyzer. However, we are fortunate and our digital oscilloscopes have a built in function generator and software that allows us to mimic a much more costly network analyzer.

We will use the built in function generator to generate a swept frequency over a narrow frequency range, around the resonant frequency of the crystal. We can plot the magnitude of the swept frequencies as they pass through the crystal seeing which frequencies pass through the crystal and which ones are attenuated.

You can do this experiment with either the leaded crystal which is in your electronics kit, or with a special resonator break out board you will be given. Here is the board:

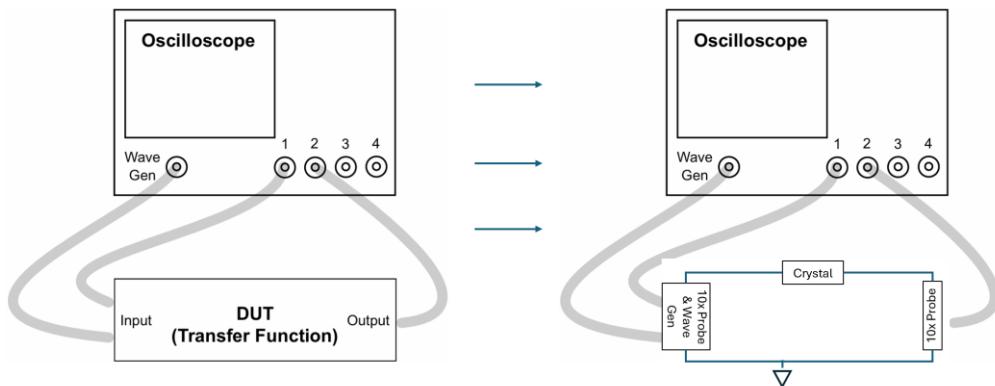


You will have to solder the pins in this board.

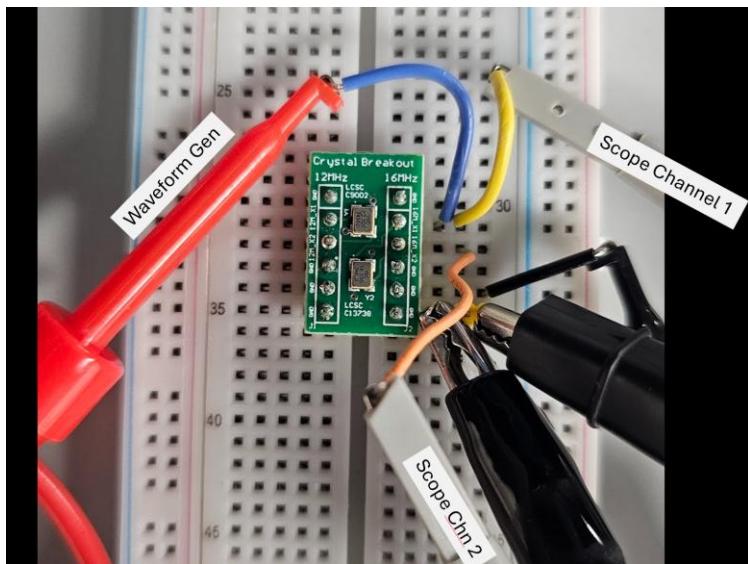
The connections to the ceramic resonator for the 12 MHz resonator is on the left side of the board and the connections for the 16 MHz resonator is on the right side. You will be using both of these on your board 3.

All the GND pins are connected to the plane on the bottom of the board. For the 12 MHz resonator on the left side, the second and third pins down are the connections across the resonator. The same pin connections are mirrored on the right side for the 16 MHz resonator.

You will build a circuit with the voltage output of the function generator shorted by the crystal. You can use either the leaded crystal part or the small break out board with a ceramic resonator. We want to find the frequency at which the series impedance of the resonator is lowest. We can do this by simply measuring the crystal as shown:



Here is my set-up



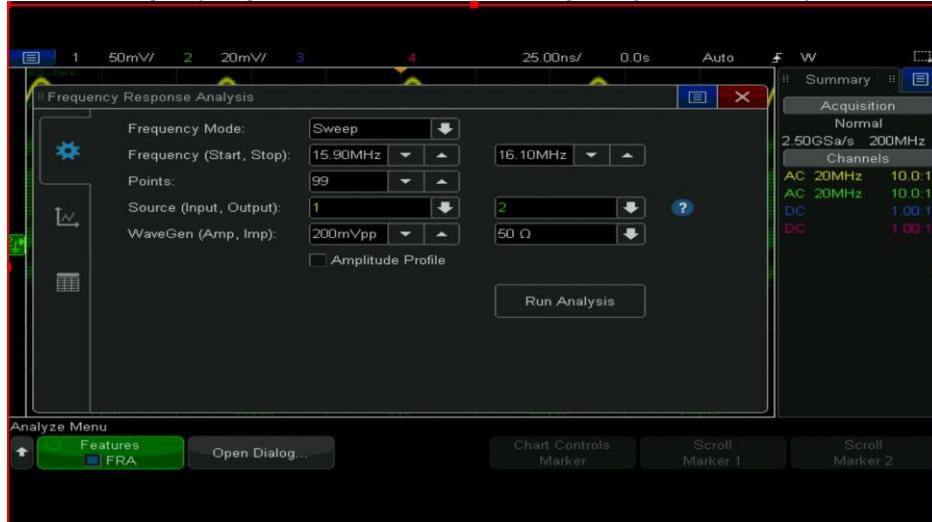
To enter the Frequency Sweep Mode:

- 1 Choose **Main Menu > Analyze > Analyze Menu**.
- 2 Click **Features**; then, select **Frequency Response Analysis**.
- 3 Click **Features** again to enable the feature.



- 4 Click the **Open Dialog...** softkey to open the Frequency Response Analysis dialog box.
- 5 Select the setup tab on the left side of the dialog box (gear icon).

Once in the Frequency Response Menu, I entered the following setting for the 16 MHz crystal:



After entering the start, stop and number of points (which approximately set the step size), click **Run Analysis**. You should get a sweep plot like this (I disabled the phase):



You can manually scope the cursors with the touch screen to see which frequency the crystal allows the most energy to pass, this is the resonance frequency of the crystal. To push in deeper to the results, you can look at the table format of the data and find with frequency the had the highest gain. You could iterate to "zoom in" to find the exact frequency. My crystal resonance was at 15.99867 MHz.



You can save the analysis results by choosing **Main Menu > File > Save Menu**, clicking **Format**, and selecting the **Frequency Response Analysis data (*.csv)**

Measure the resonance of the 12 and 16 MHz ceramic resonator.

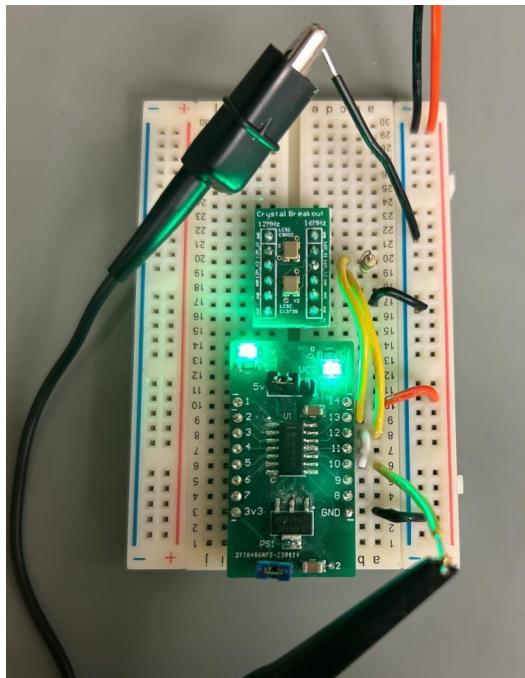
25.5 Experiment 4: Add a crystal in the feedback loop

When a crystal is added to the feedback loop of an inverter, the frequency of oscillation of the inverter depends on the frequency at which the feedback impedance is lowest.

Add just a crystal to the feedback circuit. Does the inverter oscillate? What is the frequency of oscillation? Use the buffer on the output of the inverter.

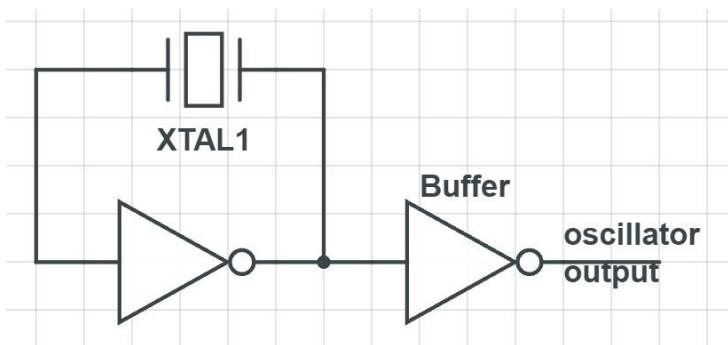
Be sure to keep all wires as short as possible.

If you still have your hex inverter breakout board that you assembled and tested, you can use it to drive the crystal. Otherwise, you can use the 74AHC14 DIP package of the hex inverter you have in your electronics kit. Here is my circuit using the hex inverter I assembled:



I used the 5 V rail from my Arduino to power this inverter. The Arduino was plugged into a 9 V AC to DC wall supply.

The frequency in a ring oscillator depends on the propagation delay between a change in state at the input causing a change in state at the output which changes the input, etc. Here is the basic circuit:



Only one inverter is needed to generate the oscillations. However, it is good practice to add another inverter at the output of the oscillator to buffer the oscillator. This way if the output is loaded down with a low impedance, like 1 k ohms, the buffer will drive the load without impacting the inverter on the oscillator.

In principle, this circuit will drive either a crystal or a resonator. In practice, we need to add two features to make the oscillations more stable and robust.

At other than the resonant frequencies, the impedance of a crystal is very high. It is like an open. If you connect an open across the output and input of the inverter, will the inverter oscillate? If the feedback resistor is an open, there is no feedback and no oscillation.

This means that sometimes, the circuit will not oscillate. Something has to kick start the feedback, like broadband noise or an initial transition. Often times, if the inverter does not oscillate, just turning the power off and on will generate enough transient at the output to start the oscillation. Try unplugging the power to your hex inverter and turning it on again. Do you get oscillation?

In my circuit, I did not get oscillation with just the crystal in the feedback loop.

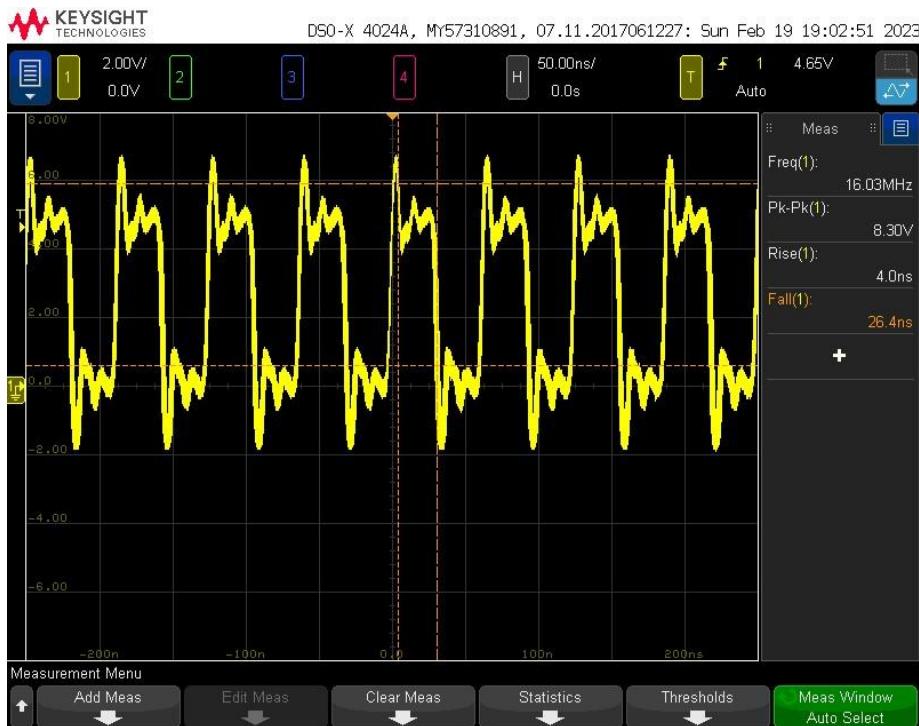
25.6 Experiment 5: Add a large value resistor to the feedback loop

To guarantee your circuit oscillates, we can add a high impedance resistor, like 1 Meg ohm, across the output and input. This will start the circuit into oscillation. Then, the frequency will be set by the frequency at which the crystal impedance is lower than the feedback resistor.

You want the feedback resistor as high as possible so that by itself, it will drive the ring oscillator into oscillation. Once it begins to oscillate, the frequency will shift to when the crystal has a lower impedance than the feedback resistor.

Add a 1 Meg resistor in parallel across the crystal. At what frequency does your ring oscillator oscillate. How does this compare with the ring oscillator without the crystal?

I was able to get my resonator to drive the inverter at 16 MHz after I added the 1 Meg resistor in feedback across the crystal. Here is what I measured on the buffered output of my inverter:

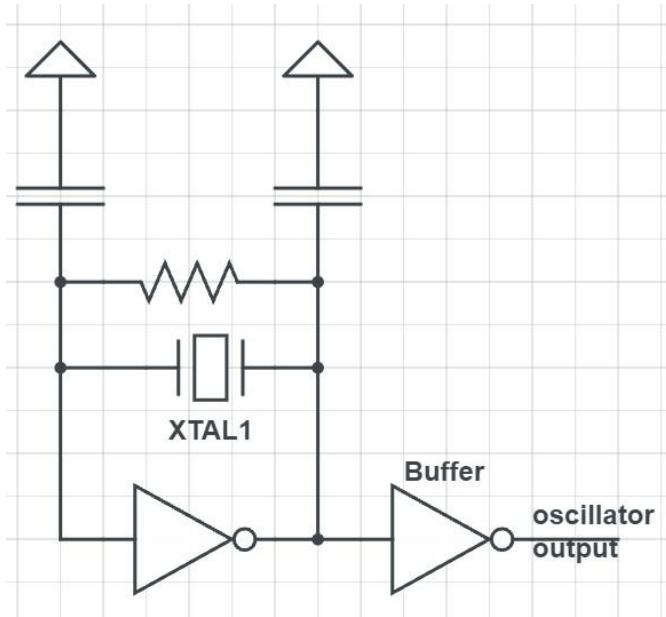


25.7 Experiment 6: add small capacitors to suppress higher order modes

Sometimes, if the highest frequency capability of the inverter ring oscillator is higher than the second or third modes of the crystal, the ring oscillator will oscillate at one of these higher frequencies. To suppress these higher order modes, we want to filter out the higher frequencies from providing feedback.

To do this, we add some filter capacitors to ground at the input and output.

We need to suppress some of the higher mode frequencies that can resonate, so that only the lowest frequency oscillates. This means we need to add two filter capacitors on either side of the crystal. Their value should be in the 7 pF to 30 pF range, depending on the resonator. Values of 22 pF work well for our applications. This practical circuit is shown below:



Try another crystal. What is the frequency of oscillation of each crystal? Was there any change to the resonant frequency of your ring oscillator?

In applications with a crystal used with a microcontroller, you are connecting an external crystal to an inverter circuit inside the uC. Sometimes, there is already a 1 Meg resistor across the feedback of the inverter, so you do not always need an external resistor. However, you generally need the 22 pF capacitors across the crystal to suppress higher order modes.

In your lab report, describe your circuits and the frequency at which you measured the oscillations. Include a scope trace of the waveform on the output of the buffer.

25.8 Grading rubric:

1 point for check off by your TA

If there is a report on this lab, your report should also include:

- 1-point if you included screen shots of the scope oscillation and a photo of your circuit that oscillates
- 1-point if you have a coherent explanation for frequency of your oscillator.

Chapter 26 Lab 14: The TVS diode array to protect against ESD events

If your board will connect to your computer, for example, though the USB port, there is the possibility an ESD generated high voltage event might send a high voltage pulse into your computer, damaging your USB hub.

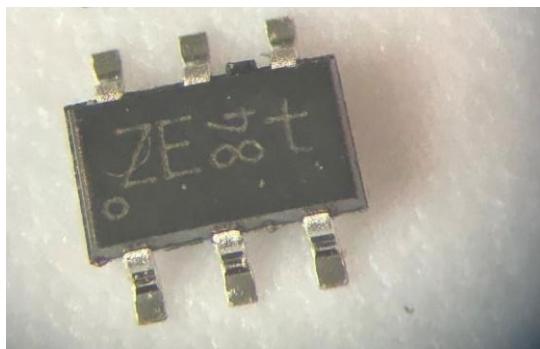
To prevent this, we use a transient voltage suppression (TVS) chip to prevent excess voltage from appearing on your data lines or power lines into your USB cable to your computer.

This device is basically just an array of diodes which will forward conduct if the voltage on the D+, D- or 5 V rail gets too high.

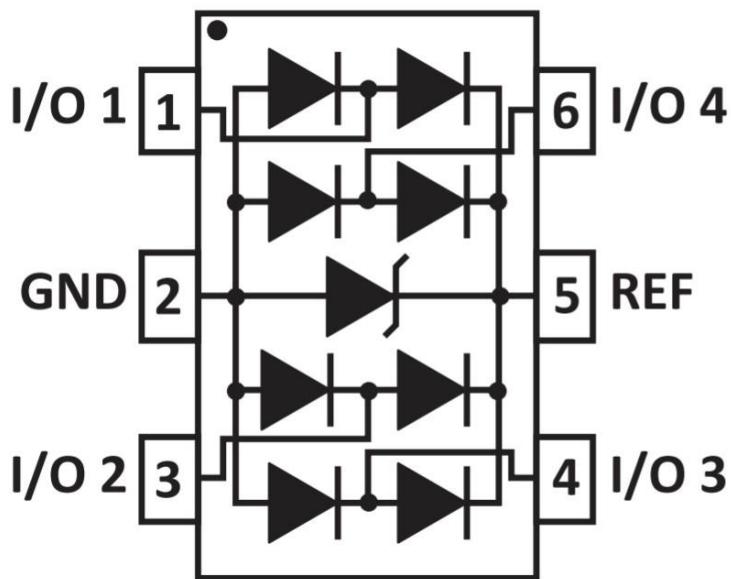
In this lab you will explore how effective this TVS chip is in suppressing high voltage transient.

The one we are using for this board is this one: https://datasheet.lcsc.com/szlcsc/ProTek-Devices-SRV05-4P-T7_C85364.pdf

Here is a closeup of the part:



In the datasheet you will see the internal circuit shown below:



Based on what you know about the behavior of a diode and a Zener diode, you should be able to figure out how this device works, what its purpose is and how to connect it into your schematic.

Take a few minutes to do your own analysis.

Here is my analysis.

Pin 5, labeled as REF is the 5 V rail.

Between pins 2 and 5 is a Zener diode. The breakdown voltage, as listed in the datasheet, is 6 V. This means that if the voltage between pin 2 and pin 5 exceeds +6 V, the Zener will turn on and reduce the voltage. Since the gnd will always be less than the 5 V rail, the Zener will never be conducting in the forward direction. This Zener limits the voltage range that can appear on the 5 V rail and ground to less than 6 V. Any ESD pulse that gets on the power rail to exceed this voltage will be suppressed. This is also why you should never apply a VCC > 5.5 V.

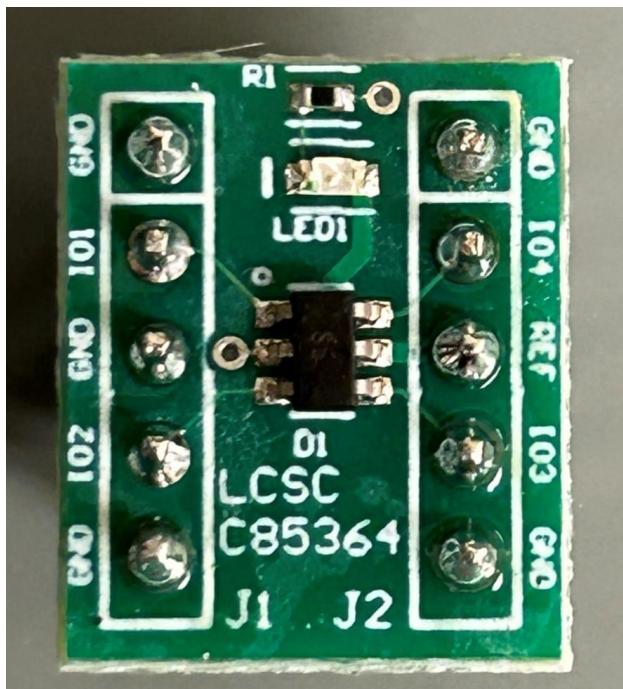
Trace the circuit connection between pin 1 and pin 6. This is a trick question. There is no connection.

Where does pin 1 connect? Just between the two diodes. If the voltage on pin 1 exceeds 5 V or goes below 0 V, the diodes turn on. As long as the voltage stays in this range, as it should, the diodes are reverse biased and play no role.

How will you connect the TVS chip to the power and ground connections on your USB connector and to the D+ and D- data lines?

All that is necessary is gnd to pin 2, 5 V to pin 5 and D+ goes to pin 1 and D- goes to pin 3. The TVS chip just touches the data lines- there is no connection passing through the TVS chip.

In this lab, you will use a TVS chip that has been assembled on a small breakout board with all the pins brought out. An example is shown below. You may have to add your own header pins to your board.



Wire this up in a solderless breadboard and power it with the 5 V from an Arduino board. Note, the pin labeled ref is the 5 V rail.

Use the function generator set for a sine wave with a 20 V peak to peak value and about 1 kHz frequency. The frequency is not critical. How do you set the output voltage to 20 V peak to peak? How do you verify this voltage?

Measure the output signal from the function generator before you connect it into one of the data lines. Use the mini grabber coax cable to connect the function generator to gnd and one of the IO pins of your TVS chip.

When you connect it to an IO pin, what do you expect to see as the waveform? If the voltage on the IO pin goes below about gnd - 0.7 V, the diode will forward conduct.

If the voltage on the IO pin goes above about 5 V + 0.7 V, a diode will forward conduct.

What is the waveform you see?

Why should you never use a 10x probe to connect your function generator to any other device you want to drive?

In your lab report, describe the circuit, what you did, the voltages you measured on the function generator before and after connecting it to the TVS chip. Explain why you see the waveforms you do.

Remember:

You must connect the signal AND ground reference of the function generator to your solderless breadboard circuit.

DO NOT use the 10x probe to connect from your function generator to the TVS chip. Why is this a bad idea?

Use a BNC cable with mini grabber ends to connect your function generator into your solderless breadboard

Use the 10x probe to measure the voltage on the output of the function generator.

26.1 Grading rubric:

1 point for check off by your TA

If there is a report on this lab, your report should also include:

- 1-point if you included screen shots of the scope waveform with your TVS chip attached to the function generator and you can explain the shape.
- 1-point if you have a coherent explanation for why you should use the TVS chip and how it would be hooked up to the D+ and D- lines of the USB data lines.

Chapter 27 Lab 15: my good-bad switching noise board

This lab will give you a chance to explore what to expect with your board 2 design project. This board will demonstrate the two important design principles to reduce switching noise:

Use a continuous return path under the signal traces

Use low inductance decoupling capacitors in close proximity to the IC power pins.

This board has a 555 timer as a clock and two different hex inverter circuits, one circuit designed following best design principles and, in another region of the board, the same circuit but with really bad design practices.

Some inverters in the circuit will generate the dI/dt . The switching noise will be measured on quiet HIGH and quiet LOW inverter pins.

This board has been built for you and you will have a chance to measure my version, before you get your board back to measure.

Very important: remember that the switching noise depends on dI/dt . To evaluate the amount of noise, you should also measure the dI/dt of all the switching signals. It will be different for different circuits.

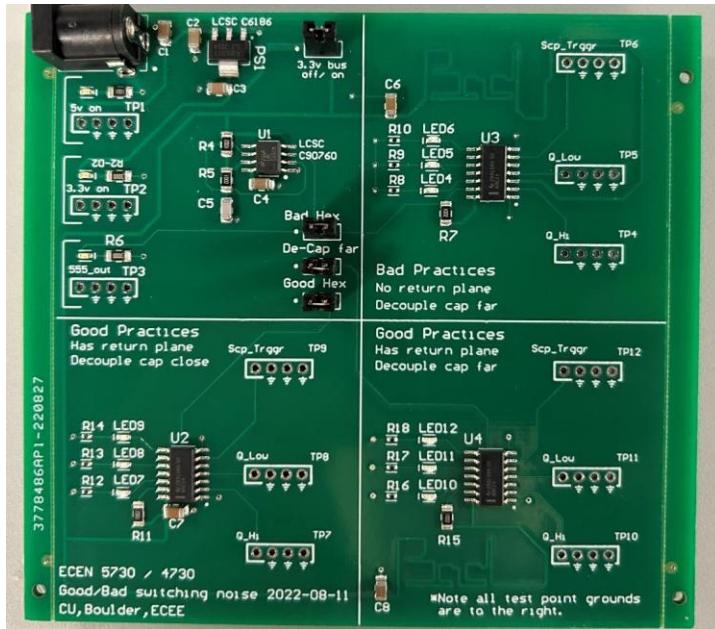
27.1 Purpose of this lab

In this lab, you will get to measure some of the switching properties of a version of your brd 2 that we built for you.

This board will give you an idea of how the board 2 design will be implemented. Review its layout carefully. Note the test points, switches and indicators.

27.2 The board you will measure

For this lab, you will use a version of brd 2 that Darren Schultz designed. Your board does not have to be exactly the same as this one. In fact, you should be able to identify at least three important changes you would want to implement in your board compared to this one.



In this board, the 555 and the hex inverters are all powered by 3.3 V. The switch selector in the upper left quadrant connects power to the hex inverters. When the jumper is removed, none of the hex inverters are powered on.

Note: this board uses the [CMOS version of the 555 timer](#). It can operate from a power source as low as 1.5 V to up to 15 V. We are using 3.3 V.

For this board, there are actually 3 regions. The upper left is the power and clock. The clock is sent to the three different regions using the three jumper switches in the upper left quadrant.

The upper right quadrant has no return plane and the capacitor, C6, very far away from the IC it is decoupling. The worst thing you could do.

The lower right quadrant has a ground plane, but you can see the decoupling capacitor, C8, is very far away.

The lower left quadrant is how you want to design all of our boards. There is a ground plane and the decoupling capacitor, C7, is close to the IC pin.

You will measure the quiet low and quiet high on each of the these three quadrants using the scope trigger test point to trigger the scope so you can see the switching noise.

Before you do the measurements, apply rule #9. Think about what you expect to see in each case.

27.3 Beyond “working”

Just because a circuit “works,” in other words, meets the initial engineering specs as defined in the plan or record, does not mean it is a well-designed circuit or board. We evaluate the quality of the layout in terms of the circuit performing the intended function AND the amount of noise.

Whether our design requires low noise or not, we should always follow good design practices that results in the lowest noise possible, at no additional cost. If it is free, it is always a good thing to do as it will add operating margin to your circuit.

The design guidelines you follow to reduce noise at no additional cost should become habits. The maximum noise your design creates should become a measure of your skill as a designer and a feature you become proud of.

In my circuit (your circuit’s criteria may be a little different), “working” means:

The 5 V rail produces a DC level of 5 V

The 3.3 V rail produces a DC level of about 3.3 V

The 555 timer produces an output of about 3 V peak to peak and about 1 kHz with about a 60% duty cycle

The three hex inverters create a trigger signal and drive the three LEDs with a modulated signal.

The quiet high and quiet low on the board and on the die show switching noise

You can measure the noise on the 3.3 V rail near the LDO and see it is much less than the quiet high noise on either inverter IC.

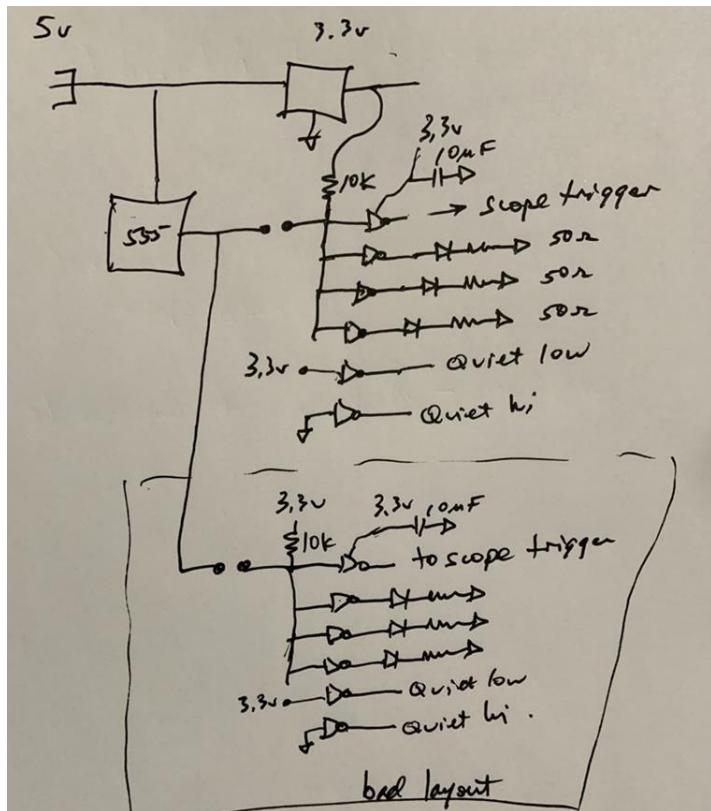
The first step is to perform the measurements to verify the board “works”.

27.4 Circuit Design for brd 2

The circuit design for brd 2 is very similar to board 1, the practice board. 5 V will come in from the power jack and power a 555 timer. We add a few more circuit elements.

The timer signal will connect to three different 7414 hex inverter chips. In each chip, there are 6 inverters. An inverter flips the input level to the output level. A LOW input creates a HIGH output. These 6 inverters in each chip are independent.

The starting place for any design is a rough sketch with as much detail as necessary to convey the important design features. Here is my rough sketch, in the figure below.



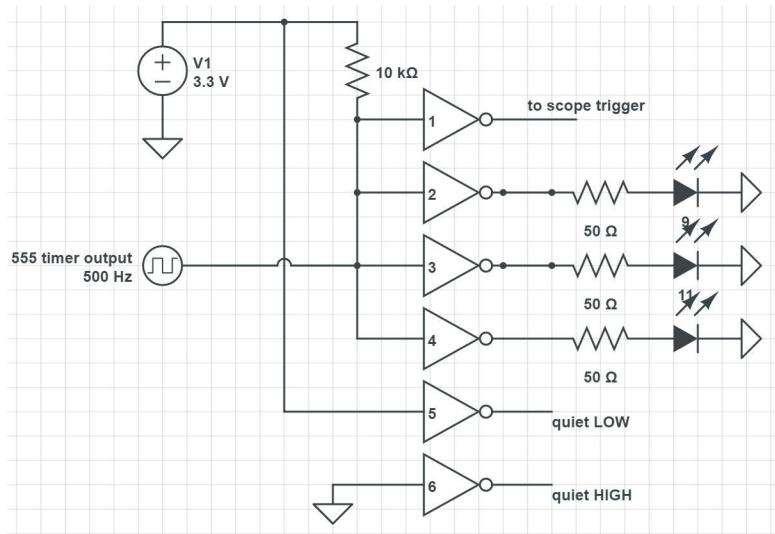
The hex inverter is a collection of 6 independent inverters that share the same power rails in the IC package.

In each circuit, four of the hex inverters are driven with switching signals. The output of one of these inverters is used to trigger the scope.

The output of the other three inverters will drive LEDs with current limiting resistors, so that we have some dl/dt that switches.

Two of the inverters are special.

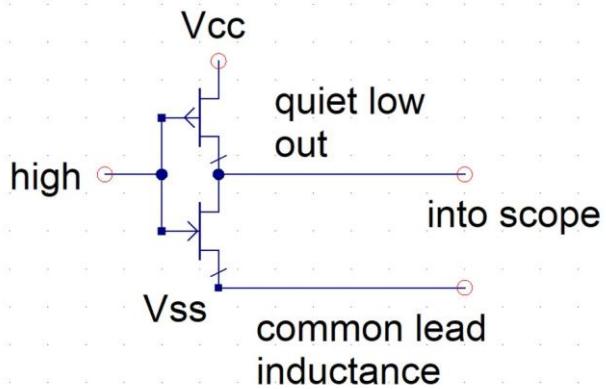
Here is another version of the schematic for each region we built in this board:



Note, there is a 10k pull up resistor to all the switching inputs. Why is there here? Hint: if the 555 is disconnected to the inputs, what do you want the inverters to do? They should stay off. If there were no pull up connection to the 3.3 V rail, and the inputs were floating, what would they do?

27.5 Quiet LOW and quiet HIGH pins

When the input to one inverter is tied to 3.3 V. This means its output is always a low. The actual transistor level circuit for one inverter is shown below:



When the input is a high, the p-channel MOSFET connecting the output pin to the VCC rail on the die is open and the n-channel MOSFET on the low side is shorted. This connects the output pin to the Vss rail on the die.

We measure the voltage on the quiet low pin at the test point on the board. This is the voltage on the Vss pin relative to the local ground on the board where the 10x probe connects to the test point.

If the Vss rail on the die bounces above the voltage on the board ground, we see it as noise on the quiet pin at the test point. This is how we measure ground bounce noise or cross talk between the quiet output signal return loop that is not switching and the other signal-return loops that switch.

Likewise, a quiet HIGH output connects the output pin of the hex inverter to the internal VCC rail on the die. This allows us to measure the switching noise on the power rail when the other I/Os switch.

We call these two pins quiet HIGH and quiet LOW because nominally, these pins are not switching, and they should have no changing voltage on them. Any voltage we measure must be switching noise.

In the circuit you will build, and that we have already built and provided you, there are three signals to measure for each circuit:

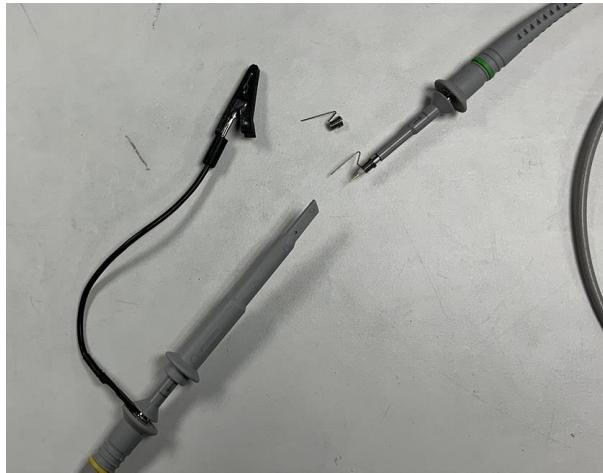
- *Trigger output for the scope*
- *A quiet LOW*
- *A quiet HIGH*

27.6 What you will measure on our board

27.6.1 Use the 10x scope probes

Remember to use the 10x probes with the ground spring tips. Whenever you use the 10x probes, remember to:

1. *Set the channel on the scope to DC coupling*
2. *Make sure the 10x probe is set to 10x and not 1x*
3. *Compensate the 10x probe*
4. *Use the spring tip, as shown in the figure below*



Be sure to watch out for probe to probe cross talk.

27.6.2 Measure the I/O switching on the good and bad layout sides

Measure the rise and fall time of the trigger signal to the scope. Trigger the scope on this I/O switching.

Measure the switching noise on the quiet LOW and quiet HIGH pins. This is the noise synchronous with the I/Os switching.

There are other test points available on this board.

Here is a special note: the input to an inverter is very high impedance. If there is no connection to the inverter, stray ESD fields present in your hands will trigger the inverter. It is always important when you are not connecting to an inverter, to always tie it high thru a 10k resistor. This way the output is always a low.

In the circuit for brd 2, a switch connects the 555 output to the various inverters. When the switch is closed, the 555 will drive the inverters. When the switch is open, and the 555 is not connected to the inverters, they should be tied to 3.3 V through a 10k resistor. This keeps the inverters from switching unless you want them to.

27.6.3 Analyzing the noise

Identify the good layout part of the board. Turn on the hex inverter on the good side of the board and turn off the hex inverter on the bad side.

Measure the signal on the output pin of the hex with the 10x probe. What is the rise and fall time? Use the rising edge as the reference.

Use this output to set the trigger for the scope. Measure the switching noise on as many test points as you can, such as:

- *The 5 V rail*
- *The 3.3 V rail*
- *The quiet LOW*
- *The quiet HIGH*

Then, turn off the good hex circuit and turn on the bad hex circuit. Repeat these measurements on the hex inverters on the board layout side. How does the noise on each test point compare in the bad layout region compared to the good layout region?

What can you conclude about how the different layout features in the good layout region and the bad layout region affects the switching noise?

You should make at least the following measurements on your board:

555 timer output signal

5 V rail

3.3 V rail

Trigger out to the scope, good and bad sides

Quiet low, good and bad side

Quiet high, good and bad sides

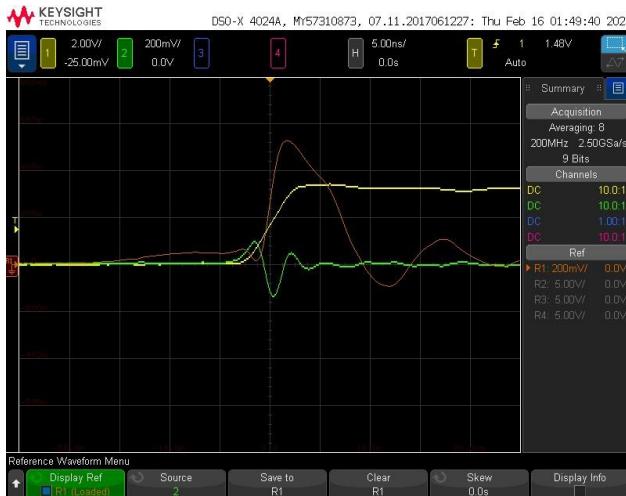
27.7 My results

Here are some examples of the measurements I took on this board:

Bad-bad layout, quiet low, tallow trace is the scope trigger signal. Note the rise time about 5 nsec. The quiet low noise is about 0.5 V.

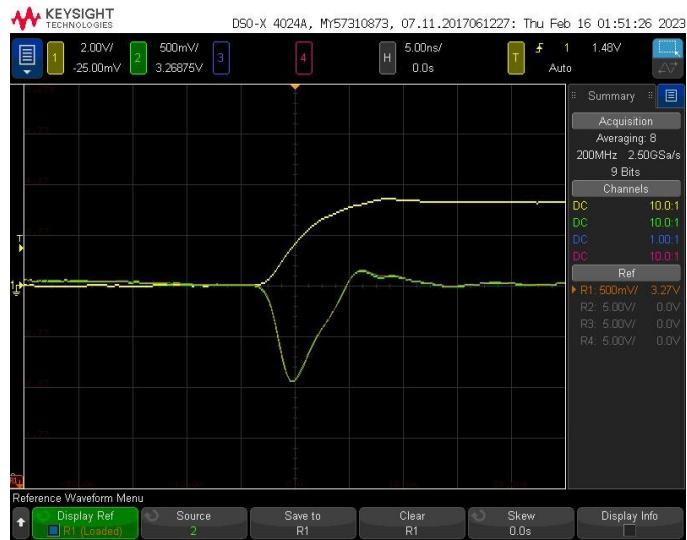


On the same scale, here is the quiet low noise with the good-good layout:



Note: the rise time is shorter, maybe 3 nsec, and the ground bounce noise is less, only 200 mV peak to peak.

Here is the quiet HIGH of the bad-bad quadrant. Note the rise time of the signal is close to 10 nsec. There is a 1 V drop in the 3.3 V rail voltage. This is huge!



Here is the quiet HIGH for the good-good layout. Note the rise time of the signal is only 3 nsec, and the rail collapse noise is much less.



Be aware that the rise time of the switching inverters is affected by the rail noise. When there is a lot of rail collapse noise, the rise and fall times of the inverter changes. The lower the voltage on the rail, the longer the rise time. The output transistor is effectively turning off.

This means when you compare the switching noise, you also have to compare the rise times. When the rise time is longer AND the switching noise is higher, this means the switching noise is much higher on the bad layout section than the better layout section.

What a huge impact just the board layout has on the switching noise!!! This is why layout is so important.

27.8 In your report, you should include

1. *Analyze the measurements to compare the switching noise on the good layout and the bad layout sections.*
 - a. *The rise time of the hex scope trigger output and the 555 timer output. What is the difference in their rise times?*
 - b. *Q_hi outputs on good and bad layouts, and why they are different*
 - c. *Q_low outputs on good and bad layouts, and why they are different*
2. *What features on this board contributed to the reduced noise in the good layout section?*
3. *Switching noise on the 5 V and 3.3 V rails (zoom in to see any synchronous noise)*
4. *Based on what you measured, what do you recommend as best design guidelines in your next design to reduce switching noise?*

27.9 Special notes for your lab report

Remember: your report will look great on your portfolio page. Few hiring managers have ever seen the sorts of measurements you did with this board.

Take pictures along the way so you can document what you did.

Always adjust the scales of the scope to most effectively show the measurement to illustrate your interpretation

NEVER use cursors or measurement functions if you haven't gotten an estimate with your mark I eyeball. If you can't take a reading from the screen, how do you expect the cursors to know how to take a reading?

Never show a screen shot without including your analysis of it. Where are you measuring, under what condition are you measuring, what is the information you gain from the measurement and what does it mean? What do you conclude from the measurement?

Take every opportunity to practice articulating the technical terms to describe what you measure and what you interpret.

Include your analysis and interpretation of the noise you measured.

*What features in the circuit or layout did you find particularly useful in the bring up of your board?
What features would you want to include in your next board?*

27.10 Grading rubric

1 point for check off by your TA

1 point for including scope shots done correctly with a good explanation

1 point if you provide a coherent explanation of how design features affect the noise you measured.

Chapter 28 Your Brd 2 bring up, test, evaluate

28.1 Assemble your brd 2

Before you start assembly, look at your bare board. Make note of labels you might want to add to future board to make the assembly process easier. Remember, whenever you think "easier" think "lower risk".

You might consider testing parts of your board as you assemble them, rather than assembling all the components on your board and then doing the testing. Of course, if you used isolation pins to turn off some of the circuits this will facilitate test and debug.

28.2 Testing your brd 2.

Follow the methods you did in lab 15 to measure the switching noise on the good and bad layout side.
Measure the following and include in your final board report:

Quiet high and quiet low

On the rising and falling edge

On your good layout and on your bad layout region.

28.3 Rubric

Follow the rubric in the brd 2 description you read when you started this project.

Chapter 29 Brd 3: final CDR

Before coming to class, you should have completed the layout.

In class, a few volunteers will be selected to present their layout to the class. While they present their layout, the rest of the class will critically listen and offer recommendations to the designer.

Be on the lookout for:

1. *Any possible hard errors which will prevent the correct operation of the board*
2. *Any soft errors which will increase the noise or the risk of the board not meeting spec*
3. *Any good design features which might be good features to include in everyone else's designs.*

Use the check list in the lab manual and in the textbook to go through the list of possible, or commonly occurring errors to verify none of these are present in the design.

After a few layouts have been reviewed, you will have time to complete your layout in class.

Before you leave, you should review your layout with your TA to have them check off your design.

It is not necessary your design be completed by the end of class time, but it should be far enough along so that any major issues can be identified.

Before you post your three design files on canvas, you must:

1. *Review your schematic and layout against the checklists provided in the lab manual and textbook*
2. *Have another student review your schematic and layout*
3. *Have your TA check off your layout*
4. *Post your design files to the JLC website and complete the DFM and get an email acknowledgement back from them that your design is accepted and ready for ordering. This may take as much as 2 hours to get an email back.*
5. *If you received any error messages from the DFM check, you have corrected them, and resubmitted your design for DFM check.*

Design files are due by 9 am on Thursday morning. This is when they will be ordered. If you miss this deadline, your board will not be ordered. You will have to wait for the next week to have your board ordered.

Turning in your design files counts as 3 points. The grading rubric is:

- 1 point for sign off by your TA in class.*
- 1 point if your design files were turned in on time*
- 1 point if your design was accepted by the vendor for production*

If your design file is rejected by JLC, you may only end up with the 1 point for TA check off for the design assignment. This will only happen if you fail to go through the online DFM check on the JLC web site.

If your board fails the DFM check when the TA places the order for your board, it will slow up the delivery of ALL the boards and may affect the schedule for the rest of the students. DO NOT LET THIS HAPPEN FOR YOUR BOARD.

If your board is not ordered with the other student boards, you are still responsible for getting your design completed. Submit it by the next week.

Chapter 30 BOARD 4: A 4-layer Instrument Droid board

Board 4 will be a 4-layer board with a microcontroller and data acquisition system. This is an instrument droid. Graduate students will build a complete instrument droid including the 328 uC. It will be a standalone instrument droid and is not meant to be Arduino Uno R3 compatible.

However, think about how you will bootload your 328 to turn it into an Arduino. You have already build a solderless breadboard version of the 328 and a board version of the Arduino. You have to bootloader the 328 in each case. If you do not include access to at least the pins you need to bootloader your 328, how will you turn it into an Arduino? What other header pins do you want to include in your instrument droid?

Undergraduates will build this instrument droid as a 4-layer shield that plugs into any Arduino uno R3 board. Your shield may not require 4-layers, but undergraduates will still design a 4-layer shield to gain practice at a 4-layer design.

You will use your instrument droid to measure the output impedance of any voltage source, such as a power supply or even just a digital output pin. You will measure the Thevenin output resistance of any supply as the current load changes. You will plot the Thevenin resistance vs the current load and use this instrument to characterize multiple voltage sources such as:

- *A 5 V wall wart*
- *A 9 V wall wart*
- *A 12 V wall wart*
- *The 6V power supply*
- *An AA battery*
- *A coin cell battery*
- *A LiPo battery*
- *The digital output pin of an Arduino*
- *The output of a 601 opamp*
- *The output of a TLV4110 opamp*
- *The output of a 555 timer*
- *The output of a hex inverter*
- *The output of the 5 V rail of an Arduino*
- *The output of the 3.3 V rail of an Arduino*
- *The output of a 3.3 V LDO*
- *The output of a DAC*

You will design the board, assemble it, boot it, test it, write the code for it, and then use it to characterize a few voltage sources.

Many of the details for this instrument droid are described in Lab 21, which is the solderless breadboard version of this instrument droid. This lab is basically an external shield that the commercial Arduino Uno talks to and performs the same function as your brd 4. Review this lab for the design details.

30.1 The purpose of the board

This board will satisfy three goals.

1. *First, it will be a chance to design and build a 4-layer board.*
2. *Second, you will have many different components you will have to assemble to your board by hand.*
3. *And finally, it will perform a very special function which will make you aware of the limitations of every power source.*

You will have to write some code to exercise the important functions of this board.

This board will use some new parts. If you would like to get practice creating a new symbol and footprint, you can create your own library elements. Otherwise, you can use the library elements that are in the integrated library. There are a series of videos in the skill building workshop that discuss the library in Altium: [SBW-2](#) [Altium using libraries and generating new symbols and footprints](#)

30.2 Principle of operation

You will design and build an instrument droid: an intelligent data acquisition system with a very specific purpose or “killer app”. The killer app in this lab is to automatically measure the Thevenin output resistance of a power source as a function of current draw.

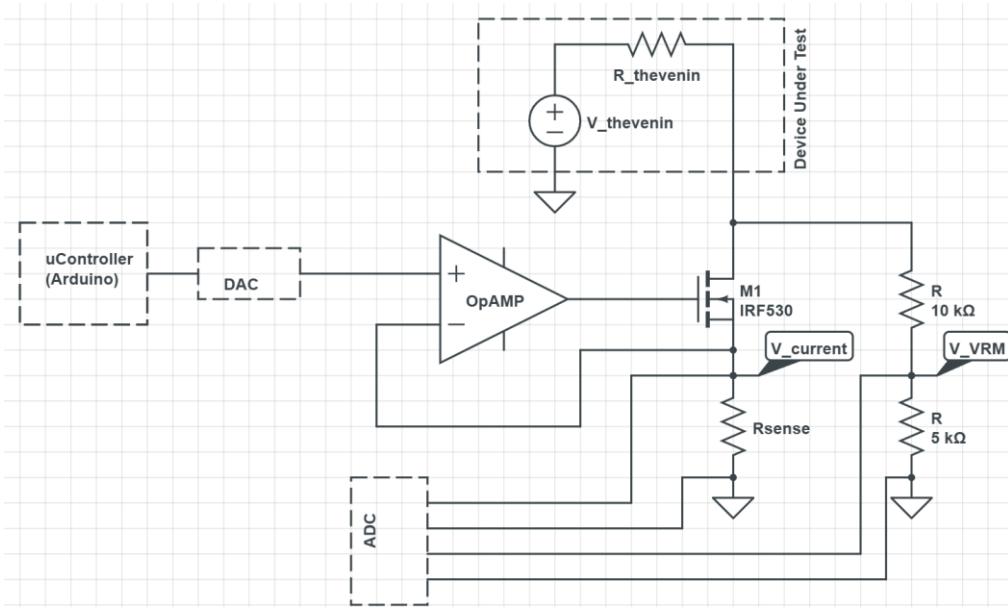
This is the same measurement you did when evaluating the Thevenin resistance of the function generator in an earlier lab. Only this time, you will measure the output resistance over a wide range of current loads and determine how linear or non-linear it is. In addition, you will see if there is a specific current clamp on the output.

This board consists of two parts, the Arduino uC part and the functionality which implements the VRM characterization. Graduate students build this as one board.

Undergraduates will use either their own Arduino board or the commercial one as the uC but will build just the shield that performs all the specialized killer app functions. This shield will be a 4-layer board that plugs on top of the Arduino Uno board using the header pins on every Arduino Uno.

The instrument droid uses the same principle as was used to measure the output impedance of the function generator, but implemented in a way that can be controlled by a uC.

Here is the schematic of the droid you will build:



The goal is to measure the Thevenin voltage and the Thevenin resistance of any power source you connect.

The uC will tell the DAC to output a voltage pulse. The opAmp will turn the voltage at its non-inverting input to an output voltage that will turn on the MOSFET enough to let through enough current so that the voltage across the sense resistor is exactly the same as the DAC voltage.

This current through the MOSFET is also through the VRM (voltage regulating module) you are measuring. There will be a voltage drop on the output of the VRM due to the current flowing through the Thevenin resistance of the VRM.

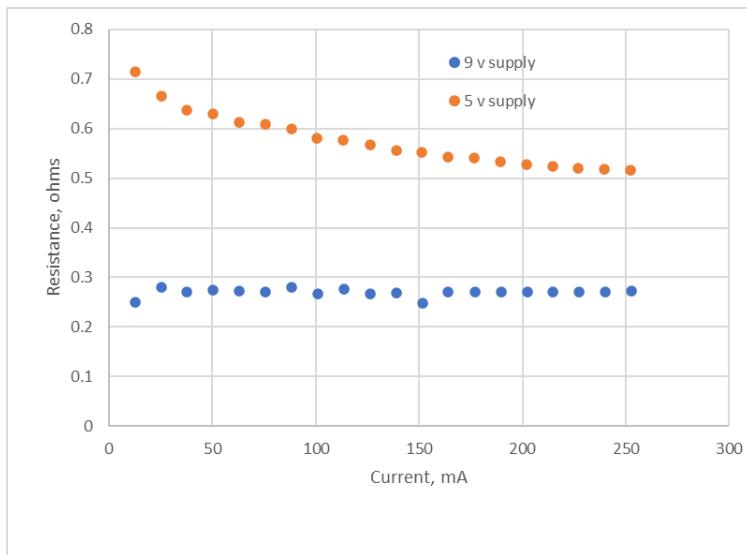
You will measure the current through the MOSFET, as a voltage across the sense resistor and the voltage drop on the power rail using a 2-channel ADC. The ratio of the rail voltage drop to the current draw is the Thevenin resistance.

The ADC has a voltage range of only about 0 to 5 V. I want you to measure a rail voltage as high as 12 V. This means you will need to add a voltage divider on the rail so that you bring the highest rail voltage you will measure below 5 V. I do not stock a 5 k ohm resistor. How will you achieve a 3:1 voltage division?

You may be turning on currents as high as 1 A through the MOSFET. There may be as much as a 12 V drop across the MOSFET. Its instantaneous power draw may be as high as $12 \text{ V} \times 1 \text{ A} = 12 \text{ watts}$. If this is on for more than 1 sec, the MOSFET will burn up or smoke. To avoid this problem, you will pulse the DAC with a signal with a duty cycle of no larger than 10%. This will drop the average power to 1.2 watts, more manageable.

You will write the code (provided for you in lab 21), to turn on the DAC, measure the voltages, calculate the Thevenin resistance, increase the current and map out the Thevenin resistance as a function of the current draw from a low current to a high current.

Your uC will add the intelligence to this system to know at what current to start, and when to stop the ramping, display the current, the Thevenin voltage and the Thevenin resistance. You will then copy this data from the Arduino IDE and paste it into excel where you will plot the Thevenin resistance vs current, such as shown here.



You will design your system to apply currents as small as 1 mA up to about 3 A. What sense resistor will you use?

30.3 The design goals

Here are some of the features of your brd 4 design:

- You will use a 4-layer board
- The uC is the 328. You can literally copy your Arduino board design, if you want
- Add a few smart LEDs and a speaker to provide indication of the start of the current pulse, and when the current is on
- Use a MCP4725 DAC to control the output current
- The current source can be a MOSFET and opamp with input from the DAC

- In principle, it does not matter which opAmp or which MOSFET you use. In practice you want to use the MCP6002 opamp because this is the one we have in stock and available for assembly. And use the SMT MOSFET in the JLC library, the AO3400A. We only stock these parts.
- The current through the VRM will be pulses with a 10% duty cycle. This will keep any parts from overheating.
- The current will ramp in steps from 0 A to a max of 3 A max.
- There will be a switch to set the max current between 300 mA and 3 A based on the sense resistor used
- The current ramp up will stop if the loaded voltage on the VRM drops below 75% of the unloaded voltage
- The VRM voltage could be as large as 12 V
- You will measure the VRM voltage unloaded and under current load with the ADS1115 as a differential voltage
- You will measure the voltage across the sense resistor with the ADS1115 as a differential voltage
- You should plan to average a few consecutive points for the loaded and unloaded voltages and current.
- Export the Thevenin resistance, the current load and the loaded voltage of the VRM and plot this in excel.
- The input from the VRM to your board will be by USB mini, power jack and screw terminal
- The ADS1115 and the MCP4725 will be powered by the 5 V from the input to the Arduino board
- Note that the ground of the VRM under test must be connected to your local ground on your board

30.4 The POR for brd 4

The purpose of this board is to create an electronic load to draw some current from a voltage source and then measure the voltage drop from the voltage source. This circuit is sometimes called a slammer circuit, very similar to what you already built the PDN lab.

Knowing the current load and the voltage drop from the power source will enable you to calculate the Thevenin resistance of the source. This is also sometimes referred to as the output resistance of the source.

When you worked with the function generator, you measured the Thevenin output resistance of the function generator and found it to be 50 ohms. Generally, for signal sources, the output resistance will be from 10 to 100 ohms. For power sources like batteries or regulators, the output resistance will be 0.1 ohms to 10 ohms.

30.5 New components

This board will use all the same components you used for the Brd 3 Arduino board, with the addition of a few new parts. These will be:

ADS1115 16-bit ADC

MCP4725 12-bit DAC. It is equivalent to this part: <https://www.adafruit.com/product/935>

MCP6002 opamp https://datasheet.lcsc.com/lcsc/1811081213_Microchip-Tech-MCP6002T-I-SN-C7377.pdf

RGBW digital smart LED such as this one:

https://datasheet.lcsc.com/lcsc/2106062036_WorldsemiWS2812B-B-W_C2761795.pdf

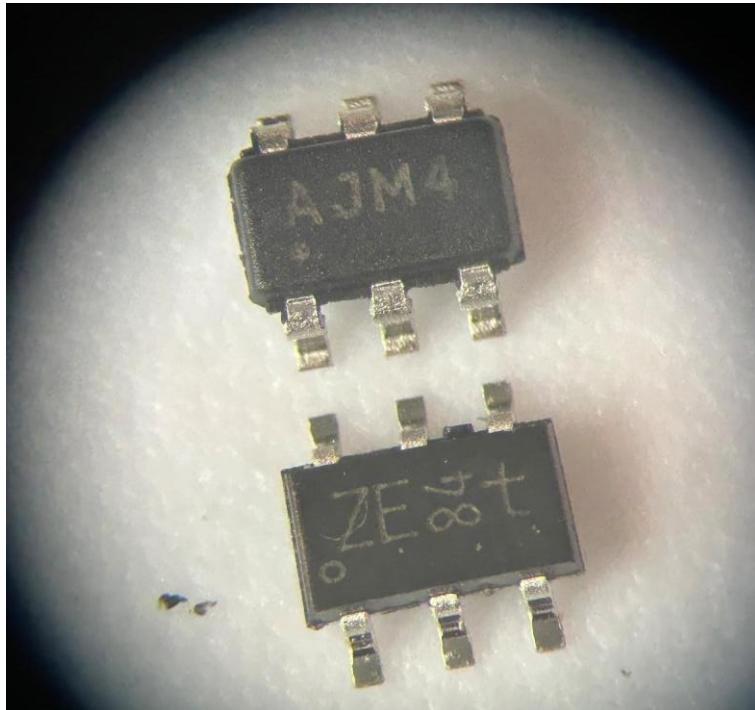
a simple buzzer such as this one: https://www.digikey.com/en/products/detail/pui-audio-inc/AT-1224TWT-5V-2-R/5011404?utm_adgroup=Battery%20Products&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Product&utm_term=&utm_content=Battery%20Products&gclid=CjwKCAjwwL6aBhBIEiwADycBIMjkXjMLpLKZR7P0hpDTBnFSnAb5LHGvVBG3Rcbn6lw08_BdvrBkFhoCVaIQAvD_BwE

an n-channel MOSFET such as this one: https://www.digikey.com/en/products/detail/alpha-omegasemiconductoric/AO3400A/1855942?utm_adgroup=Semiconductor%20Modules&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Product&utm_term=&utm_content=Semiconductor%20Modules&gclid=CjwKCAjwwL6aBhBIEiwADycBIBv6WIxvAvjV6ectl1lcPlajx906R5kl64p9VoxMpWaZRa85FzzKphoCajlQAvD_BwE

All of these parts can be found in the JLC integrated library.

For this board, while you will use the parts in the JLC integrated library, we have all of them in stock and you will hand assemble them to your board. Be sure to use the specific parts referenced above, as these are parts we will have in stock.

Just as a note, there are two parts which have a similar footprint. They are 6 pin SO-23 style. Superficially, they look very similar and are easily confused. Under a magnifier, you can see the difference in their markings. Here are these two parts. The AJM4 is the DAC while the ZE84t is the TVS part. Make note of this when it comes time to assemble your parts to the board. They will both be in your kit of parts. Don't confuse them.



30.6 The schematic and layout

Design your schematic and layout for brd 4.

What is the stack up you want to assign for each of the 4-layers? Here is a hint: Do not define any layer as a plane layer. You should use a signal layer for each layer so the layer is a positive layer.

Even though you will route a ground plane on one layer, select the layer type in the stack up manager as a signal layer. All this does is select a layer as a positive or negative layer.

Remember, wherever there is a signal via, you will need to add a return via connecting between the two ground planes. All your vias will be through hole vias.

If you are designing a shield, be sure to stick your header pins from the bottom of your shield so the pins are sticking down and will insert into the header pins of the Arduino board. When you set up your BOM, make sure you DO NOT include an LCSC catalog number for these header pins. JLC will only assemble component on one side of your board. If you include the catalog number for the header pins in your BOM, JLC will assemble them facing up from your board. This will make your board unable to plug into an Arduino Uno board. You will hand assemble the header pins so they are inserted from the bottom of your shield.

30.7 Submitting the design files and grading rubric

After your board is designed, you will need to submit the gerber file, pick and place and BOM on canvas. The deadline for submission is 9 am on the day the boards will be ordered.

If you miss this deadline, you will not get your board back on time to complete for this class.

The grading rubric for submitting the design files is the same as with the other boards:

1 point for sign off by your TA

1 point if your gerber files are submitted on time

1 point if they are accepted by the vendor for fabrication

30.8 Part 2: assemble and test your board

When your bare board arrives, collect all the parts you will need for your board from your TA. Verify you have all the parts against your BOM, before you begin assembly.

Determine which order will be lowest risk to assemble your parts. Wherever possible, test each part after it is assembled, and before other parts are added. Write some basic code to exercise each part as it is added.

For the fine pitch parts, look at the leads under a microscope to verify they are soldered well with no shorts.

Where possible, use a socket for a leaded part rather than soldering the pins into the board. A socket can be an array of female header pins.

30.9 Part 3: document your board and rubric

The final report on your board is the final for this class. It is the last assignment.

It is worth 20 points, and 20% of your grade. The grading rubric is:

5 points for writing up a summary of what worked and did not work, how you would do the board over again if you could and what you learned in this project that you will avoid doing, or try to do in your next design.
Need only be a 1-2 page summary. Consider writing this up to post on your portfolio page.

1 point each for demonstrating each of the following and showing some detail (screen shot, picture, serial plotter, etc., in your final report:

- *The 328 is booted and responds to commands*
- *The DAC output*
- *The ADC measures some voltage*
- *The constant current signal on the sense resistor*
- *The MOSFET turns on*

5 points for demonstrating the buzzer and smart LEDs turn on and perform some function.

5 points for demonstrating that the board measures the output resistance of a voltage source that is plugged into it.

Chapter 31 Lab 16: diff or SE signaling and ground noise

Purpose of this lab is to compare the quality of the analog signal measurements from a sensor using a single ended and differential pair measurement and explore signals on the I2C bus.

You will need a solderless bread board, your Arduino to provide power and to drive the I2C signals, an ADS1115 module and a TMP36 temperature sensor. These should all be in your kits.

The ADS1115 module is a 16-bit ADC. You will use this same chip, not on a module, on your sensor shield, brd 4. It has four channels and communicates over the I2C bus.

31.1 Set up the SBB

We will use a TMP36 temperature sensor as a voltage source for the ADS1115. By placing it at the opposite end of a solderless breadboard as where the ADS1115 module will go, we will be able to add some noise to the ground return path and see the difference between a differential and a common signal measurement.

We will measure the voltage on the temperature sensor using a single-ended measurement and simultaneously, a differential measurement.

The single-ended measurement will be more sensitive to noise (IR drop) on the ground line while the differential measurement will not be sensitive to this noise.

31.1.1 Wire up your solderless breadboard

As a good habit, we will always try to use a consistent color-coding habit:

Ground is black, sometimes grey or green

Power is red

Signals are other colors.

Use a consistent strategy for power or ground routing on the vertical columns.

In this case, we will be powering the ADS1115 and the TMP36 from the 5 V rail of the Arduino.

Connect the power lines from the Arduino to the solderless breadboard.

Remember, always try to make your wires as short as convenient, rather than use long floppy jumper wires.

31.1.2 The TMP36 temperature sensor

We will use as our sensor, a TMP36 temperature sensor. This is a simple component. The datasheet can be [downloaded from here](#). It has three terminals, as shown in Figure 26.1.

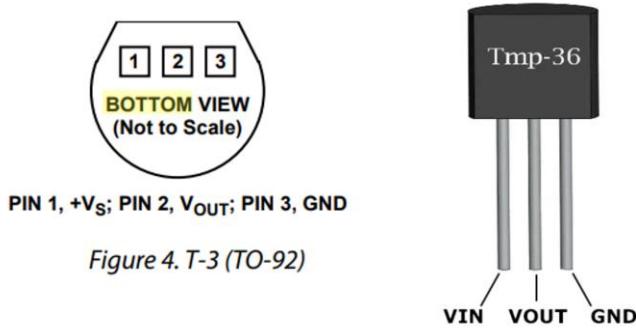


Figure 26.1. The pin configuration of the TMP36. Note this pin out when assembled into the solderless breadboard. Note, this is the **BOTTOM VIEW**, looking at the pins from the bottom of the package.

The calibration curve for the temperature and output voltage is shown in Figure 26.2. Its sensitivity is 10 mV/degC.

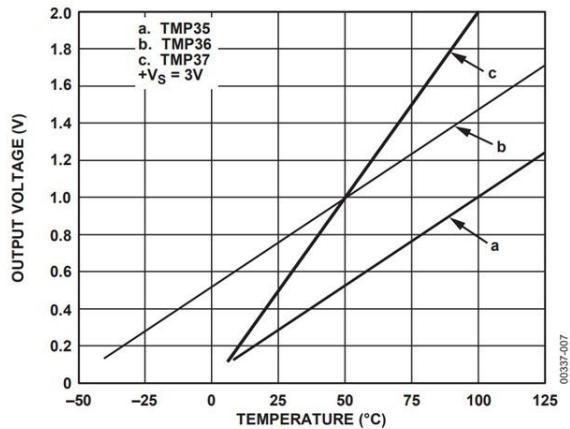


Figure 6. Output Voltage vs. Temperature

Figure 26.2. Calibration curve for the TMP36 sensor. It is curve b.

The temperature and voltage are related by:

$$T \text{ [degC]} = V \text{ [volts]} \times 100 \text{ [degC / V]} - 50 \text{ [degC]}$$

and

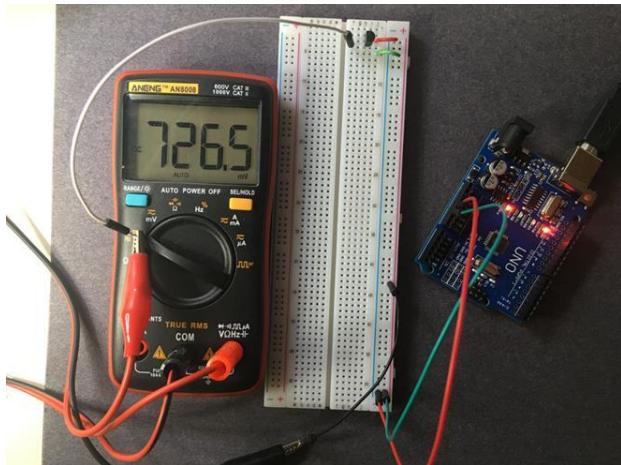
$$V \text{ [Volts]} = T \text{ [degC]} \times 0.01 \text{ [Volts / degC]} + 0.5 \text{ [Volts]}$$

This means, at roughly 20 degC, the voltage on the sensor is about 0.7 V.

Insert the TMP36 into the solderless breadboard at the opposite end of the board as where the ADS1115 will go. This will enable us to add noise in the ground return path.

To wire up the TMP36, we will apply 5 V from the Arduino board to the TMP36, pin 1, and gnd to pin 3. Pin 2 will have a voltage on it related to the temperature.

This should be about 0.7 V. Once configured, verify the voltage with a DMM or your scope. This configuration is shown below.



31.1.3 Assemble your ADS1115 module

The ADS1115 module may come with the header pins not connected to the circuit board. If they are already assembled, you are all set. If they are not, you will have to solder the header pins into the board. When you have completed the assembly, your module will look like the unit shown below.



The simplest way to solder the pins is to first apply solder flux to the holes. Then insert the pins with the long ends down, into a solderless breadboard. Place the module on top of the short end pins so it is fully seated. Liberally coat solder flux on all the holes.

Solder one pin in by using a clean solder tip, wetting it with a blob of solder and moving it in contact to the pin and board. The solder will wick onto the pin and fill the hole. Remember, the solder tip temperature should be about 700-725 degF.

Once one pin is soldered, move from pin to pin soldering the others. This process is described in one of the skill building workshops on Master Soldering. [SBW-3 Secrets to Great Soldering](#). Be sure to view

these videos BEFORE attempting any soldering operations.

Here is the link to the [Adafruit web site](#) for this specific ADC. The block diagram for this part is shown below. This is taken from the TI datasheet which you can [download from here](#).

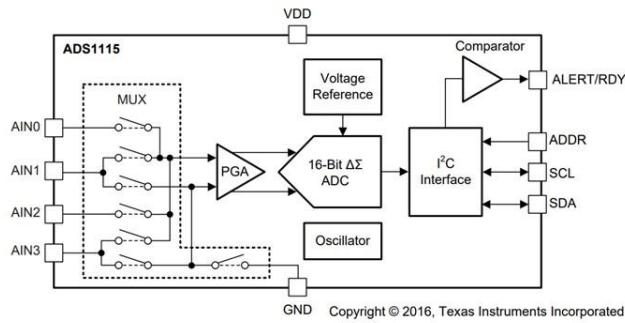


Figure 22. ADS1115 Block Diagram

Note, it can be used as either 4, single-ended inputs or 2-differential inputs, or some combination, reconfigured for each measurement. If it is used as 4, single-ended inputs, the – input to the programmable gain amplifier (PGA) is set to ground, at the ground lead of the IC.

If it is set as 2 differential inputs, the + and – input to the PGA are the differential inputs. The output of the PGA, is a measure of the voltage difference between the two inputs and this amplified difference voltage goes to the ADC and is digitized at 16-bit resolution.

The PGA can be set for six different ranges, as listed below.

Table 3. Full-Scale Range and Corresponding LSB Size

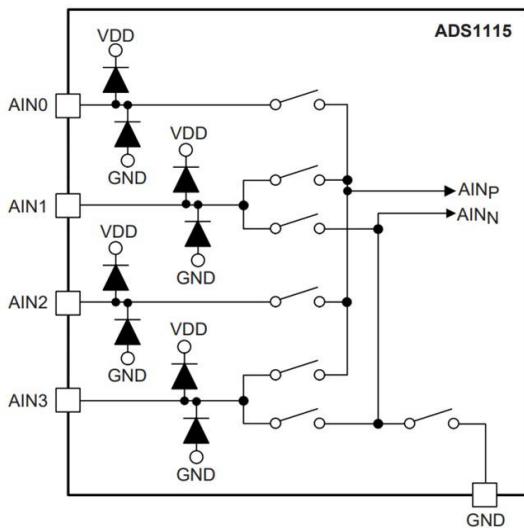
FSR	LSB SIZE
±6.144 V ⁽¹⁾	187.5 µV
±4.096 V ⁽¹⁾	125 µV
±2.048 V	62.5 µV
±1.024 V	31.25 µV
±0.512 V	15.625 µV
±0.256 V	7.8125 µV

(1) This parameter expresses the full-scale range of the ADC scaling.
Do not apply more than VDD + 0.3 V to the analog inputs of the device.

This means, when the scale is set for ±2.048 V, for example, each ADU bit level is 62.5 uV. This value is just:

$$\frac{\text{Volts, full scale}}{\text{ADU}} = \frac{4.096 \text{ V}}{2^{16} - 1} = \frac{4.096 \text{ V}}{65535} = 62.5 \frac{\mu\text{V}}{\text{ADU}}$$

The ADS1115 also has some ESD protection, as shown below. If any of the input pins go below gnd or above Vdd from an ESD event, the diode will protect the amplifier. But they are not designed to handle much DC current. Do not apply more than Vdd + 0.3 V or the diodes will turn on and may be destroyed.



Copyright © 2016, Texas Instruments Incorporated

Figure 25. Input Multiplexer

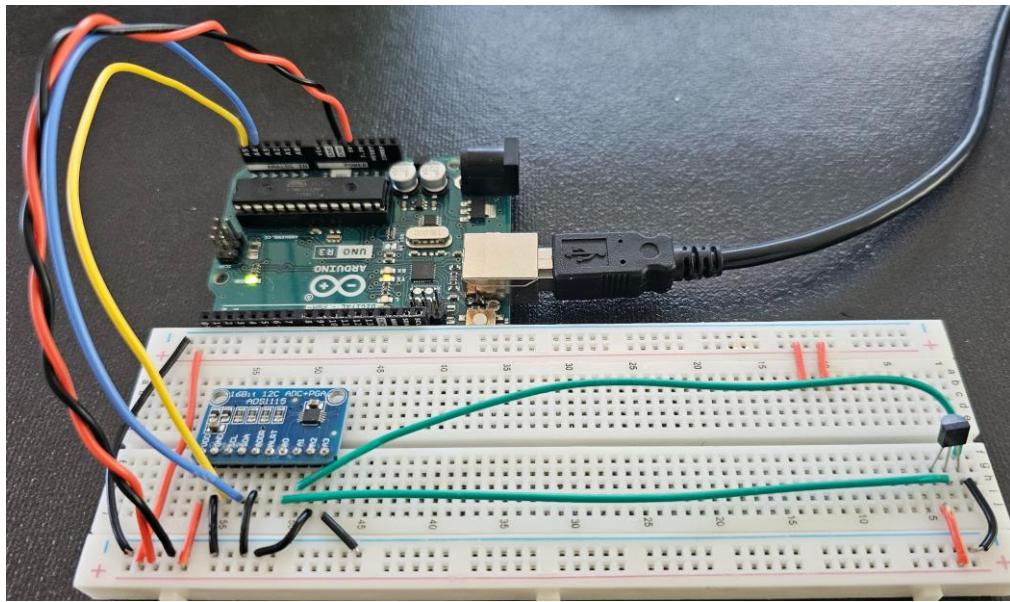
The interface to the ADS1115 is an I2C interface. This means you need to connect the serial clock, SCL, and serial data, SDA, pins to the SCL and SDA pins of the 328.

In the Atmega 328 chip, the SCL pin on the die is connected to both the analog A5 pin and a separate pin in the header strip above pin 13. The SDA pin on the die is also connected to the analog A4 pin and a separate pin in the header near pin 13. You can use either connection on the Arduino board.

To drive the ADS1115 module, you can use the commercial Arduino.

For this lab, you will not need to add pull up resistors to the I2C bus. The 40k Ohm pull up resistor on the Arduino pins is small enough to charge the I2C pins to meet the timing requirements for the 100 kHz standard bus clock.

Wire up the ADS1115 at the opposite end of the solderless breadboard than the TMP36, as shown in the figure below.



31.1.4 Wiring in the ADS1115 module

Here are the pin connections to use for the ADS1115 module.

VDD ---- +5,

GND ---- Gound

SCL ---- Arduino Uno A5

SDA ---- Arduino Uno A4

ADDR --- GND

ALRT --- no connect

AIN0 ----- TMP36 output voltage pin

AIN1 ----- gnd connection **at the location of the TMP36**

AIN2 ----- local gnd

AIN3 ----- local gnd

31.1.5 Setting the address of the ADS1115

The ADS1115 module communicates over the I2C bus. There are four different addresses each module can be set for. This means you can use 4 modules or 16 total SE or 8 total diff channels on any single board.

The address for the ADS1115 is set using just one pin, the ADR pin. To set the address of the ADS1115 module use the following connections to the ADS pin:

- 0x48 (1001000) ADR -> GND
- 0x49 (1001001) ADR -> VDD
- 0x4A (1001010) ADR -> SDA
- 0x4B (1001011) ADR -> SCL

The default address is 0x48. Unless you have a strong compelling reason otherwise, use this address for your module. If you use a different address than the default, then after you instantiate the object, ads1, for your ADS1115, use the ads1.begin(0x49) command to assign this object the address 0x49, for example.

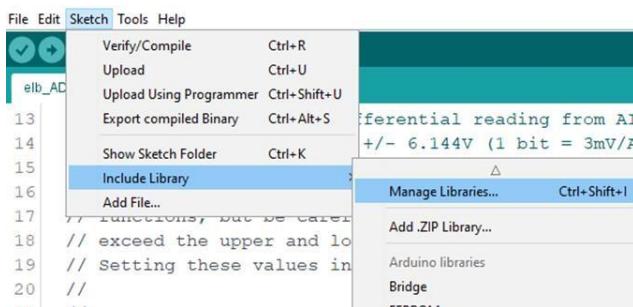
Note: the ALRT pin is only used when the ADS1115 is set for comparator mode, or for high-speed acquisition handshaking. This will send a HIGH signal when the input exceeds a set threshold. We will not use this pin in this example.

31.1.6 Libraries for the ADS1115

The simplest to use library for the ADS1115 is from Adafruit. The description of using the library [is here](#). Providing the correct pins to the SCL and SDA pins are connected to the ADS1115, you can use the same library as provided by Adafruit for the Arduino IDE.

While you can download the .zip file from the [GitHub link](#), it is much simpler to install the library in your Arduino IDE.

Under Sketch, select Include library and ManageLibraries, as shown below:



The library manager will open up and you can enter *ADS1x15* in the search box. You will then see the only item available, which is the Adafruit ADS1x15 driver. Select install. It is now installed in your IDE.

To practice using it, we will open up one of the examples and hack it.

Under File, select examples and scroll all the way to the bottom of the list of examples until you see the Adafruit ADS1X15 list and then the three options as shown in Figure 26.10.

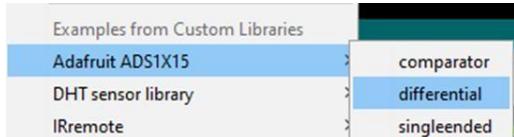


Figure 26.10. Select the differential example.

We will modify this sketch slightly to use with the ADS1115. All we want to do is set up the ADS1115 and read the differential signal between channels A0 and A1 and then the single-ended signal on A0.

In the included sketch, we modify line 4 and remove the comment marks and line 5 and add comment marks.

On line 8, I prefer using a baud rate of 2000000, because we can.

We remove the various print statements on lines 9, 11, 12, and the comments in lines 14 thru 17.

We need to uncomment the line that sets the gain of the PGA.

The voltage of our signal is about 0.73 V. This means the highest gain we can use is with a full-scale voltage of ± 1.024 V which is a gain of 4x. We select line 23 to uncomment to use this gain value.

This is a scale factor of 0.03125 mV/ADU. This is on line 23.

The actual command to read the differential channel that is the $V_{in0} - V_{in1}$, is:

```
ads.readADC_Differential_0_1()
```

To read a single-ended channel, the command is

```
ads.readADC_SingleEnded(0);
```

where the channel number, 0 thru 3, is the parameter in the () .

The output of the temperature sensor should be connected into AIN0.

In addition, AIN1 should connect with another wire to the pin 3 of the TMP36. This is the local gnd near the TMP36. The TMP36 sensor voltage is actually the voltage difference between the signal pin and its local ground.

31.2 Exp 1: setting up both Differential and single-ended measurements

It is the voltage difference between the two pins of the TMP36, the pin 2 and the pin 3, that is the actual temperature signal.

When we read a single-ended voltage on pin AIN0 of the ADS1115 chip, inside the ADS1115 chip, we are really measuring the differential voltage between the AIN0 pin and the local gnd pin near the ADS1115 chip.

This means in a single-ended signal, the voltage we measure is actually the voltage between the TMP36 pin 2 and the local ground of the ADS1115. This voltage includes any voltage in the ground path from the local ground of the TMP36 pin 2 and the local gnd near the ADS1115 module.

When we measure a differential voltage between the AIN0 and AIN1 pins, we measure the voltage between the pin 2 and pin 3 of the TMP36. This should be a better measure of the actual sensor voltage.

If there is little difference in the ground voltage between the local ground of the TMP36 and the local ground of the ADS1115, the ADS1115 should measure the same voltage either in single-ended or differential mode.

We will just read the values, convert them to mV values and print them to the serial monitor where we can see the numbers or plot them.

Whenever we use an ADC, we always want to sample at the highest sample rate possible and then average the consecutive values together to give the time resolution you require for the application. This will give the lowest measurement noise. This means we need to know the fastest sample rate and how to achieve it.

31.2.1 Measuring the execution time of the ADS1115 to sample measurements

We can perform a precision measurement of the sample rate for the ADS1115 by measuring the time to read 100 samples. The entire sketch to measure the time to run 100 operations of a read from the ADS1115 module is here;

```
#include <Adafruit_ADS1X15.h>
#include <Wire.h>

Adafruit_ADS1115 ads; // instantiates the object
float mV_per_ADU;
long n_operations = 100;
long iTimer0_usec;
long iDummy;
float Time_X_usec;

void setup() {
  Serial.begin(1000000);
  delay(1000); //lets the serial port get initialized
  ads.begin(0x48); //default address with addr pin connected to gnd
  ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V 1 bit = 3mV      0.1875mV (default)
  mV_per_ADU = 0.1875; //for ADS1115
}
void loop() {
  //Serial.println(ads.readADC_SingleEnded(0)*mV_per_ADU);
  iTimer0_usec = micros();
  for (int i = 1; i <= n_operations; i++) {
    iDummy = ads.readADC_SingleEnded(0);
  }
  Time_X_usec = (micros() - iTimer0_usec) / (n_operations * 1.0);
  Serial.print(" Xtime_usec: ");
  Serial.print(Time_X_usec, 3);
}
```

```

Serial.print(" Frequency (Hz): ");
Serial.print(1e6 / (Time_X_usec), 3);
Serial.println();
}

```

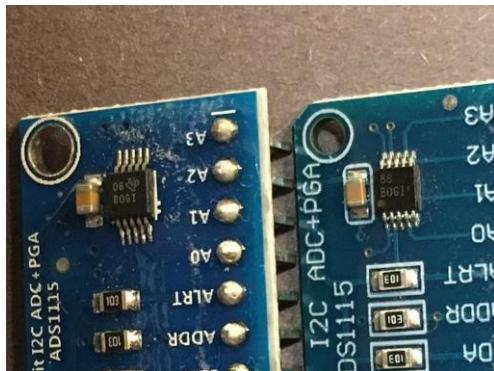
The execution time for one read is measured as 9.22 msec, which is a sample rate of 108.5 Samples/sec. The specified default sample rate in the ADS1115 library is 128 S/s. The actual measured rate is slower than this. This is probably due to some computational overhead in the Adafruit library. There is also a small variation due to chip-to-chip internal RC clock frequency.

Your ADS1115 module may not sample quite this fast.

31.2.2 Not all ADS1115 modules are created equal

While measuring the execution time of a collection of 10 different ADS1115 modules, an interesting effect was observed. Among the 10 samples, seven showed the same sample rate of 108.5 S/s, while three of them showed a much slower rate of 30.9 S/s.

In surveying the parts, I noticed a distinctive difference: the faster parts had the TI logo stamped on their chips, while the slower parts did not. It may be that the faster parts are genuine TI parts, while the slower ones are clones or counterfeit. All of these parts had been purchased from AliExpress. Examples of these chips are shown here.



Examples of two ADS1115 chips on modules. The one on the left, with the TI logo was noticeably faster than the one on the right, probably a clone.

One aspect of the timing of an ADS1115 is related to an internal clock. There is no crystal associated with the part. Instead, there is an onboard RC clock which drives the timing for the chip's operations. There is some variation in this clock from part to part. It may be that the counterfeit parts have a much slower internal clock, or some other pre-set register values that make them run slower using the default conditions in the Adafruit library.

The Adafruit library sets the sample rate register value to 128 S/s. This should give 128 S/s, but as we measured, the TI parts gave 108.5 S/s and the clone parts, 30.9 S/s.

However, with the addition of one line of code, this default rate can be dramatically increased for both parts. There is still another step required to get to the highest sample rate, but this is a simple step that should be done.

31.2.3 Faster Sample Rates in the Default set up

We can add one line of code in the set-up loop to change the register value and increase the intrinsic sample rate of the ADS1115 modules.

The line to add in the set up loop is:

```
ads.setDataSource(RATE_ADS1115_860SPS);
```

This is added after the PGA gain setting command.

With this command, the genuine parts increased their sample rates from 108.5 S/s to 373.3 S/s. The clone parts increased their sample rates from 30.9 S/s to 129.5 S/s.

If a sample rate of 373 S/s is sufficient, then it is possible to use the genuine TI parts with this additional command using the standard default library.

Using the interrupt method and the alert pin, the maximum sample rate that can be achieved is 860 S/s with the genuine TI parts. This is only 2.3 x higher than the default rate.

If a higher sample rate than 100 samples/sec is not critical, then any part can be used in your application. Be aware of this sample rate for your part by measuring it. You should measure the sample rate after adding this important line. No other changes to your code or the hardware are needed.

31.2.4 Averaging your ADS1115 readings

Instead of just taking a single voltage value and plotting it, we are going to take consecutive values and average them. We set the averaging time for 100 msec. This is long enough to get some reduction in noise but short enough to get good time resolution.

In addition, we are adding a few dummy values to print to the serial plotter in order to set the scale and fix it.

The complete sketch to do all this is here. Note, when I create a variable, I ALWAYS add the units of the variable to the name of the variable. This way I know exactly what I am measuring and in what units. This is critical to reduce confusion.

```
#include <Wire.h>
```

```
#include <Adafruit_ADS1X15.h>

Adafruit_ADS1115 ads; /* Use this for the 16-bit version */
int iTime_ave_msec = 100; //averaging time per point
long iTime_stop_msec;
int iCounter1 = 0; //used in counting number of points
float V_diff_mV; //used for diff voltage
float V_SE_mV; // used for SE measurement

void setup(void) {
Serial.begin(2000000);
// ADS1015 ADS1115
// -----
// ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V 1 bit = 3mV 0.1875mV
(default)
// ads.setGain(GAIN_ONE); // 1x gain +/- 4.096V 1 bit = 2mV 0.125mV
// ads.setGain(GAIN_TWO); // 2x gain +/- 2.048V 1 bit = 1mV 0.0625mV
ads.setGain(GAIN_FOUR); // 4x gain +/- 1.024V 1 bit = 0.5mV 0.03125mV

// ads.setGain(GAIN_EIGHT); // 8x gain +/- 0.512V 1 bit = 0.25mV 0.015625mV
// ads.setGain(GAIN_SIXTEEN); // 16x gain +/- 0.256V 1 bit = 0.125mV 0.0078125mV

ads.begin(); //with the ADR pin set to gnd.

ads.setDataRate(RATE_ADS1115_860SPS);

} void
loop(void) {

V_diff_mV = 0.0;
V_SE_mV = 0.0;
iCounter1 = 0;
iTime_stop_msec = micros() / 1000 +
iTime_ave_msec; while (micros() / 1000 <=
iTime_stop_msec) {
V_diff_mV = V_diff_mV + ads.readADC_Differential_0_1() * 0.0312;
V_SE_mV = V_SE_mV + ads.readADC_SingleEnded(0) * 0.0312;
iCounter1++;
}
V_diff_mV = V_diff_mV / iCounter1;
V_SE_mV = V_SE_mV / iCounter1;

Serial.print(720);
Serial.print(", ");
Serial.print(760);
}
```

```
Serial.print(",");
Serial.print(V_diff_mV);
Serial.print(",");
Serial.println(V_SE_mV);
}
```

You can literally copy and paste this code into a blank sketch and it will run. You may have to do some minor editing to clean it up.

The ADU values are converted into mV by multiplying the ADU values $\times 0.0312 \text{ mV/ADU}$.

When this sketch is run, the first two numbers printed are the lower and upper scales. The third number is the differential voltage in mV and the fourth is the single-ended voltage in mV. These should be the same if there is no voltage difference between the local grounds.

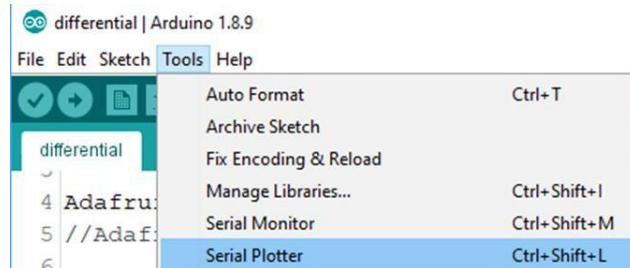
An example of the signal recorded on the serial monitor from a TMP36 is shown below. Remember to use the same baud rate in the serial monitor as in the `Serial.begin()` command.

```
720, 760, 738.42, 733.62
720, 760, 739.49, 734.14
720, 760, 737.29, 736.56
720, 760, 738.19, 734.63
720, 760, 738.62, 736.36
720, 760, 737.39, 738.43
720, 760, 738.41, 741.68
720, 760, 739.15, 743.68
720, 760, 740.51, 743.33
```

By closing the serial monitor and selecting the serial plotter, as shown below, we can plot the two channels of data as it comes in. In the Arduino IDE 2.x, the serial plotter only plots the last 50 data points. This means we have a limited time range.

I strongly recommend you use the Arduino IDE 1.8.x which plots 500 points on the serial plotter.

When we use an averaging of 100 msec, 50 points is 5 seconds plotted on the screen using the 2.x IDE. When we use the 1.8.x IDE, there are 500 points on the serial plotter, and we are plotting $100 \text{ msec} \times 500 = 50$ seconds full scale on the screen.



The measurements will be averaged for 100 msec, or whatever you selected, and the differential signal, along with the single-ended signal plotted. An example of a plot of 50 points is shown below.



In this example, even though there is a long path between the TMP36 local ground and the ADS1115 local ground, the TMP36 sensor voltage is measured as 724 mV in both methods.

The noise on the voltage measurements is also apparent. It is about $0\text{+/- }2\text{ mV}$ noise. It may be different on your module.

31.3 Exp 2: Generating a voltage drop between the local grounds

In this example, there is only a small voltage difference in the local grounds. There is no advantage in using a differential measurement over a single-ended measurement.

The ground path from the local region of the TMP36 and the ADS1115 is a narrow trace inside the solderless breadboard in one of the vertical columns.

You should measure this resistance. Note that it will be about 0.04 Ohms. This means you will have to measure it using the 4-wire method introduced in an earlier lab.

You can use any current you want to measure the DC resistance of a column in the solderless breadboard, but be sure it is less than 0.5 A. Otherwise, there is a slight danger of melting it.

Measure the series resistance of the ground path in any of the columns. Drive a known current in a column and measure the voltage between one end of the column and the other. This is the resistance in the path from the TMP36 local ground to the ADS1115 local ground.

The reason there is normally no voltage drop in this path between the TMP36 local ground and the ADS1115 local ground is that there is normally less than 1 mA of current flowing through this roughly 0.04 ohm path. This would create a voltage drop on the order of $1 \text{ mA} \times 0.04 \text{ A} = 40 \text{ uV}$ of drop!

We will generate some noise in the ground path by driving some current through it using the function generator.

What is the output resistance of the function generator? If you are not sure, go ahead and measure it again using the scope and the scope's 1 Meg input and 50 Ohm input.

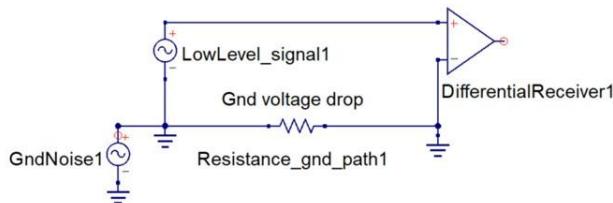
When we set up the function generator for a 0.5 Hz, 20 V peak to peak signal and short the ends of the signal, what is the current that flows from the function generator when the output of the function generator is shorted by the ground wire in the solderless breadboard?

Connect the function generator between the two ends of the ground return path between the TMP36 and the ADS1115. This generates a periodic voltage drop between the two ends of the ground path.

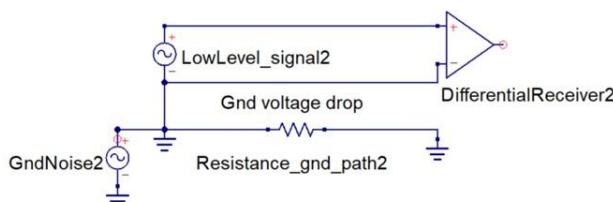
This is why it is sometimes confusing calling this conductor "ground". It is the same conductor, but due to the DC currents flowing in it, we will have a different voltage at different points of the conductor. Which point on the conductor we select as our "reference" point will influence the voltage we measure relative to that reference point.

The difference in routing topology between the single-ended and differential measurements of the TMP36, are shown below. This illustrates how the voltage noise between the local ground points of the TMP36 and the ADS1115 can contribute to noise on the single-ended measurement, but not on the differential measurement.

Single-ended Voltage Measurement

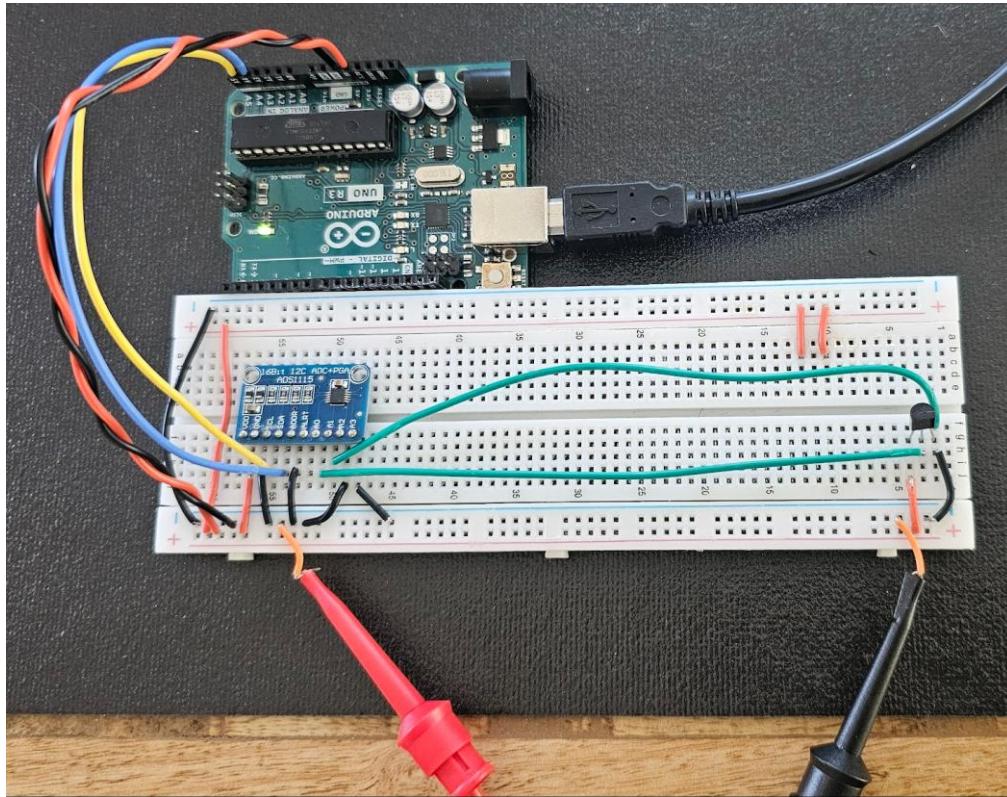


Differential Voltage Measurement



The routing topology of the low side of the ADS1115 when it is internally connected to the local ground or connected to the local ground of the TMP36.

For example, if the return path resistance were 0.04 Ohms and the current were 200 mA peak to peak, the voltage drop would be $0.04 \text{ Ohms} \times 200 \text{ mA} = 8 \text{ mV}$ peak to peak. We should see a periodic noise of 8 mV peak to peak on the single-ended signal. The differential signal should be insensitive to this periodic noise. The actual wiring set up for this experiment is shown below.



The wiring to drive an external current in the ground return path of the TMP36 and ADS1115.

Note that the AIN1, the minus input to the differential amplifier, is always connected to the local ground of the TMP36, at the location of the TMP36.

When the ADS1115 is programmed to measure a differential signal, the minus input to the differential amplifier is connected to the local ground of the TMP36.

When the ADS1115 is programmed to measure a single-ended signal, the minus input to the differential amplifier is connected to the internal local ground of the ADS1115 by an internal connection through the chip multiplexer.

When a 1 Hz square wave of 200 mA peak to peak current flows on the ground return path, the noise voltage is seen on the single-ended measurement of about 8 mV peak to peak voltage noise, but the differential measurement is completely insensitive to this noise. An example of this measurement is shown below.



The modulated voltage noise is about 8 mV peak to peak, exactly as expected based on the resistance of the return path and the current through the return path.

Now imagine this is current from the power returns, or dI/dt through the inductance of gaps in the return path. While the noise in the return path may create small differences between the local grounds at different locations, when the signal you are trying to measure is small, it is easy to see and be influenced by the ground noise. If there is not current in the ground path, there will be no voltage drop on the ground path.

This is the value of a differential measurement. When the two inputs to the sensor are located in close proximity of each other, the differential signal is not sensitive to variations in the local ground voltage at different locations.

Whenever possible, use a differential measurement for a sensor so that the measured voltage is insensitive to noise on the ground return path.

31.4 Layout considerations for differential pairs

The whole reason to use a differential measurement is to carry the difference voltage between the high and low end of the sensor back to the input of the analog amplifier and ADC.

When we route these two signal lines, refer to one as the positive line or p line and the other as the negative line, or the n line. Sometimes they are called the p and the m line, for minus.

The p and n lines are each a single-ended transmission line. They have their signal conductor and they have their return path, the plane on the adjacent layer.

We call the two, single-ended lines used to carry a differential signal, one differential pair. A differential pair is any two transmission lines that are used together to transport a differential signal.

In this application, the purpose of the differential pair is to carry the differential signal without adding additional noise. We want to pay attention to what we can do in the routing and layout to minimize the additional noise we might add to the differential signal.

The following are important guidelines to reduce the additional noise that could be picked up by the differential pair:

Route them over a continuous return path, following the guidelines to reduce ground bounce on either line.

Route the two lines as far away from other signal lines as practical.

Route the two lines as close together as practical so that they share the same, common environment. If there is some noise source from outside the board, both lines might see the same noise and the difference will be smaller than the common noise.

31.5 A "Noisy" Ground

You will hear many engineers refer to a "noisy" ground. A voltage is ALWAYS between two points. A noisy ground means there are voltage drops across different locations of the same conductor labeled as ground. The ONLY thing that causes voltage noise across different locations on the same ground conductor is large currents in the ground conductor, which flow across the impedance of the ground conductor. A current through an impedance generates a voltage drop.

A noisy ground is one in which there are large currents flowing. These currents, in combination with the impedance of the ground conductor, which means a series resistance and series total inductance, can generate a voltage difference between different locations on the ground, as we saw in this lab.

Be aware of the currents that are flowing in your ground conductor and the typical impedances they might encounter. In this lab, there were currents on the order of 200 mA. They were flowing through a ground conductor with a series resistance on the order of 40 mohms. This resulted in a voltage drop between two locations on the same conductor labeled as ground of about $0.2 \text{ A} \times 0.04 \text{ ohms} = 8 \text{ mV}$, which we were able to measure.

If a ground plane using 1 oz copper were used, the series resistance would be on the order of a few squares or about 1 mohm. Even if currents on the order of 1 A were flowing, this means the voltage drops in the ground plane, the voltage noise that would be seen between different locations, might be on the order of $1 \text{ A} \times 0.001 \text{ ohm} = 1 \text{ mV}$. Is noise between two locations on the same ground conductor on the order of 1 mV important in your application? If it is, then watch out for "noisy" ground conductors.

Note that these currents that flow over wide regions of the ground plane, and NOT directly under the path of the originating signal currents, are on the order of 10 kHz or lower frequency. All currents with frequency

components higher than this, will flow under the signal paths, and will not generate voltage drops throughout the plane. See chapter 12 in the textbook where currents flow in return paths is covered.

Noisy ground is really about the currents in the ground plane. When there are large currents, at low frequency, these currents can create voltage drops in the ground conductor. This means signals that are referenced to a local ground point may have a different voltage difference when measured at a remote ground connection due to the difference in voltage between different round locations.

This problem is fixed by using a differential pair connection, in which the local ground connection is carried by one of the conductors in the differential pair which will not have current flowing in it and will maintain the same voltage from one end to the other. The difference voltage between the signal and its local return at the sensor location will be the same as at the location where the input to the ADC is and where the sensor voltage is actually measured. This sort of measurement is insensitive to the noise on the ground conductor.

It is also possible to fix this problem by splitting the ground path and isolating the regions of the board with the large, low frequency currents to keep the large, low frequency currents from spreading throughout the rest of the board. However, this is very dangerous and should be avoided unless you are very careful and know exactly what you are doing. If any signals cross this gap in the return path, they will generate far more noise on the signal conductors which will couple over a larger number of signals, than if there were a continuous plane.

Rarely is a split ground plane to confine large, low frequency currents to a limited region of the ground plane, an acceptable alternative. Use this method ONLY if you know exactly what you are doing, you are very careful, and you make sure no signal paths cross this split ground path. It is a much lower risk and more effective solution to use differential signaling when low noise measurements are important.

For more information about noisy grounds and where return currents flow and when to use a split ground, [check out this article I wrote](#).

31.6 Check off by your TA

Before you complete this lab, be sure to call your TA over and be checked off. Be prepared to describe your measurement set up and the interpretation of the results. You should be able to answer the following questions:

1. *Describe the measurement set up used to compare a single-ended measurement and a differential measurement of the TMP36 sensor.*
2. *Why is the differential measurement so much less sensitive to ground noise than the single-ended measurement?*
3. *In your brd 4 design, you will use differential pairs to route from the current sensor on your board to the ADS1115 chip. You will also use a differential pair connection between the voltage divider on the power rail to the ADS1115 chip, because you can. Describe how you will do the routing of these two analog signals.*

31.7 The Lab Report

If there is a report for this lab, then describe what you did in this experiment. Include a picture of your set up and the plot of the measured single-ended and differential measurements of the TMP36 with a modulated current in the return path.

In particular, answer the following questions:

1. *With no current in the ground connection between the TMP36 and ADS1115, what was the temperature read by the TMP36?*
2. *With no current in the ground return path, what was the voltage difference between the differential measurement and the single-ended measurement of the TMP36?*
3. *What was the current from the function generator in your ground return between the TMP36 and ADS1115 in your set up?*
4. *What is the voltage difference you measured between the single-ended and differential measurements when there was this current flowing through the ground path?*
5. *How would you recommend routing the differential pair from the sensor to the ADS1115 for the lowest noise pick up?*

Remember, your report will look great on your portfolio page. Hiring managers will eat this stuff up.

31.7.1 Grading rubric

As is usual, the scoring for this report is

1 point for check off by your TA

1 point if you show examples of what you built, measured and your analysis

1 point if you review the principles behind the circuit and the measurements coherently.

Remember, this report will look great in your portfolio.

Chapter 32 Lab 17: I2C communications

In this lab, you will use the same set up as you did for the single-ended and differential measurements. But, in this case, we will only look at the I2C bus pins.

32.1 The I2C bus

The Inter-IC (IIC or I2C) bus is a common digital bus. There is a minimum of two communications wires associated with the I2C bus, the serial data (SDA) and serial clock (SCK) lines. However, this is misleading. In addition to these two lines that carry digital signals, there are two other lines required to be connected in common to all devices: power and ground. This means there are really four pins associated with the I2C bus:

1. *SDA: the serial data line*
2. *SCL: the serial clock*
3. *Gnd: ground*
4. *Vdd: power*

There are two types of devices on the I2C bus: a *controller*, formerly called a *master* that generates the clock and initiates communication and a *minion*, formally called a *slave*, that listens to the bus and sends data back.

The SCL is generated by the controller and is measured by the minions. This line has data (the clock signal) that travels only in one direction. It is unidirectional transport, from the controller to the minion.

The SDA line is bidirectional. Each device on the I2C bus is connected in parallel to all the other devices AND the controller. Each minion device has an address which it listens for on the data line. The communications protocol has to be carefully designed so that there are no collisions of bits on this bus. The controller manages all the handshaking of who is allowed to talk at any one time.

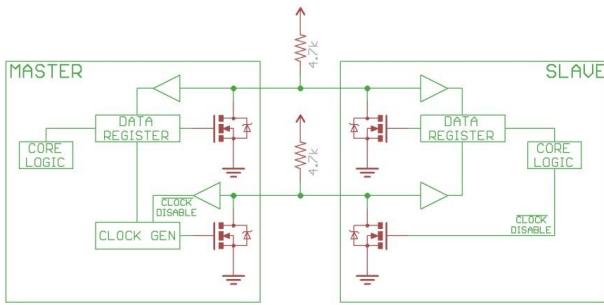
The standard rated frequency for the bus is 100 kbytes/sec. In fast mode, this can go up to 400 kHz.

A very good tutorial on the I2C bus can be [found at Sparkfun](#).

The SCL and SDA drivers for all minions and some controllers are “open collector” or “open drain”. This means the drivers connected to the bus connect by an open collector, if bipolar. This means when the driver wants to write a 0-bit, it turns the transistor on, pulling the line down.

But, it has no way of writing a 1-bit, or a HIGH signal. Instead, it relies on a pull up resistor to pull the line HIGH when the transistor is turned off. Without a pull up resistor, there is no way for a line to reach the Vdd level of a digital 1.

This is why all I2C buses show a pull up resistor to the VDD line attached to the SCL and SDA lines. An example of the circuit is shown below



In the [I₂C spec](#), the max rated sink current for any drive transistor is about 3 mA. If the bus voltage is 3.3 V, then we can reach the 3 mA limit with about a 1 K resistor. There is a voltage drop from the transistor when it is on of about 0.2 to 0.6 V depending on the type of transistor, so the voltage across the resistor is actually 3.3 V - 0.4 V = 2.9 V.

But, if the bus voltage is 5 V, then the lowest resistance still in spec is

$$R = \frac{5V - 0.4V}{3mA} = 1.5 k\Omega$$

The smallest recommended pull up resistor is 1.5 k. It does not mean that a 1 K will not work, it just means that there will be a little more power dissipation in the transistors due to sinking the 5 mA of current if a 1 k resistor is used.

What value R should we use as a pull up resistor on the bus? This is where design tradeoffs come in. A larger value means the pull up time constant gets longer. It has to be short enough so that this rise time is << the unit interval of a bit. A lower value means more power dissipated, important in low power circuits, but not as critical in wall powered circuits.

Since we want to minimize the number of unique components, we also want to consider other value parts we are using in our circuit. A 1k resistor is used for indicator LEDs. Unless we have a compelling reason otherwise, we should consider using a 1k resistor as the pull up resistor rather than a 1.5 k resistor, because we are already using them.

32.2 Impact from the capacitance on the I₂C bus

The value of the pull up resistor determines the charging time of the bus. Whatever capacitance there is in the bus has to be charged up through the pull up resistor.

The spec for the input capacitance of any device connected to the I₂C bus is 10 pF per device. The 328 pin plus one minion is a 20 pF load on the SDA and the SCL lines. The interconnects on the bus also have some capacitance.

For a few components, it may be about 30 pF of total capacitive load on the SDA and SCL lines.

The typical bus frequency is 100 kHz. This is a period of about 10 usec. If we want the charging or discharging time to be no longer than 10% of the period, which is 1 usec, as the 10-90 rise time, then the pull up resistance criteria needs to be:

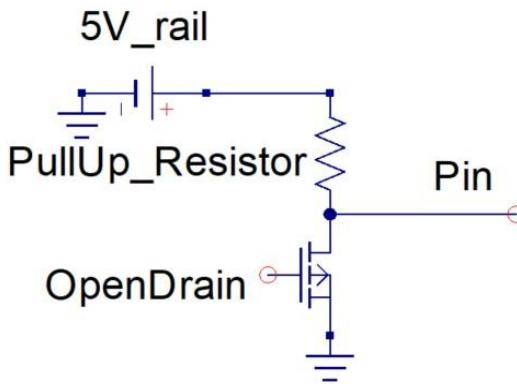
$$2.2 \times RC < 1 \text{ usec} \rightarrow R < \frac{1 \text{ usec}}{2.2 \times 30 \text{ pF}} = 16k\Omega$$

This is why many circuit diagrams show a pull up resistor on an I2C bus as 10k resistors. This is a good compromise between low power consumption and fast enough charging.

Use a smaller value pull-up resistor and the line will charge faster, so better timing margin, but more power might be dissipated.

32.3 The internal pull up resistor on the 328

When an Atmega 328 microcontroller drives the SCL and SDA pins, the output pins on the 328, are in an INPUT_PULLUP pinMode state. This means their output is connected as open drain (it is a CMOS device) with an internal pull up resistor, as shown below.



Note that not all microcontrollers have an internal pullup resistor when they drive an I2C line. This is why all generic I2C bus circuit schematics include a pull up resistor.

The specification for the 328 pullup resistor associated with the on-die pull up resistor is 10k to 50k. What value is it? It is very easy to measure, by modeling the pin as a Thevenin source. The open circuit voltage is the Thevenin voltage and the loaded voltage is related to the external load resistor and the Thevenin resistance.

Since the Thevenin resistance is high, rather than using the 50 Ohms of the scope as the test load, we will use an external 10k resistor as the load. Doing this measurement, we find:

$$V_{open} = V_{Thevenin} = 5 \text{ V}$$

$$V_{load} = 1 \text{ V}$$

This means the Thevenin voltage is 5 V and the Thevenin resistance is 40k. This would make the voltage across the 10k external load resistor in this test as:

$$V_{Load} = V_{Thevenin} * \frac{10k}{10k + 40k} = 0.2 * V_{Thevenin} = 1\text{V}$$

If no external pullup resistors are added to the I2C bus, the Arduino 328 drivers for the SDA and SCL lines will have internal 40 k pullups on them. This would give a 10-90 time constant of about

$$2.2 \times RC = 2.2 \times 40k \times 30\text{pF} = 2.6\text{usec}$$

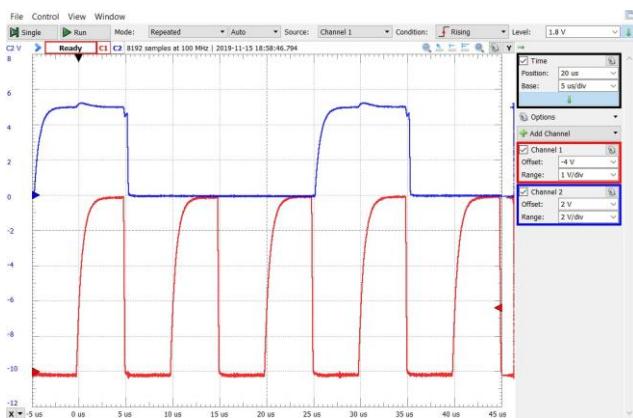
This is marginal for the 100 kHz data rate, with a period of 10 usec. And it is better if there is only 1 minion on the bus so the capacitive load is 10 pF to 20 pF.

This is why some I2C buses work just fine without an external pull up resistor due to having an internal one already on the 328 output pin, though rather high.

But, if the capacitance is much higher than 30 pF, then the 40k pull up resistor of the Arduino pin will slow the rising edge and the line may not come up in time. In this case, using a lower value pull up resistor is important.

32.4 Exp 1: measure the I2C signals

By measuring the SCL or SDA lines while the Arduino is driving I2C communications, we can look at the impact on the rise time of no pull up resistor and comparing to a 1k pull up resistor. For example, the figure below shows the measured voltage on the SCL and SDA lines with no external pull up resistor. The line is pulled up using the internal 40k pullup resistor of the Arduino digital pin. The 10-90 rise time is about 1 usec.



Use the 10x probe to measure the signals on the SCL and SDA lines. Use the SCL line to trigger the scope.

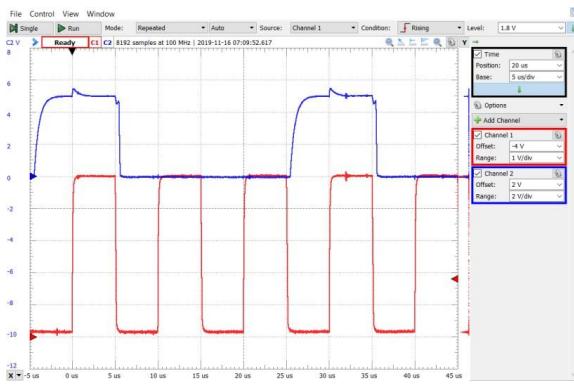
What is the 10-90 rise time and fall time of each signal?

Measure the Thevenin resistance of the output pin of the SCL or SDA pin.

If the pull-up resistor is 40k, what is the total capacitance of the data line or clock line based on the rise time?

32.5 Exp 2: change the pull up resistance and see the change in the rise time of the signals

When a 1k pullup resistor is added to the SCL line, the rise time immediately decreases. This shorter rise time is shown below.



Generally, to provide some performance margin, when power consumption is not an issue, it is always a good idea to add an external pullup resistor to the SCL and SDA line.

You should reproduce these measurements using your Arduino board and the SDL and SCL lines communicating to the ADS1115.

On your board 4 design, you will have to decide how to manage the I2C bus. Will you add a 1 k or 10 k pullup resistor (or no resistor at all)?

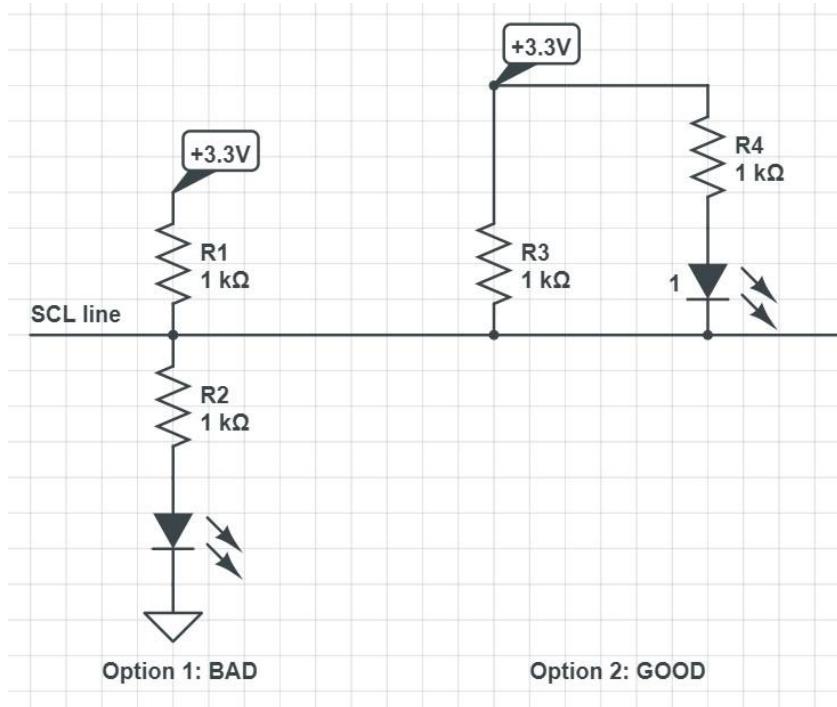
If you have 3 minions connected on your I2C bus, should you add just one pull up resistor to your I2C bus or multiple pull up resistors?

32.6 Adding test points and indicator LEDs to the SCL and SDA lines

On a circuit board, it is always a good idea to add test points to the SCL and SDA lines so you can see the signal quality of the signals and to decode the information to aid in debug.

If all the minions work fine, of course, there is nothing to debug. But we add test points, inductor LEDs and isolation switches as insurance. If there is a problem, these features will dramatically reduce the debug time and effort.

However, we have to be careful how we add inductor LEDs to the SCL and SDA lines. Below are two ways of attaching an inductor LED to the SCL line. The same option could be used for the SDA line. One circuit is a very bad idea and the second option is a good option.



Why is option 1 a bad idea? If you connected your indicator LEDs this way to the SCL or SDA lines, why could this prevent any communications on the line? Hint: what is the DC voltage on the SCL line when none of the devices pull the SCL line low? What is the minimum input high required for all the components to read a 1 on the SCL or SDA lines?

Why is option 2 a better way of adding indicator LEDs?

After you evaluate the signals on the SCL and SDA lines, add indicator LED circuits as shown above and measure the voltages on the SCL or SDA lines. Which is a better approach?

32.7 Cross talk between the SDA and SCL lines

The worst thing you can do when routing the cables from the uC to the device is twist the SDA and SCL wires together. While this makes neat wiring from the Arduino to your SBB, it also couples these two signals tightly together and increases the cross talk between them.

It is much better to tightly couple each signal line with its return path (ground). This will reduce the switching noise.

You can convince yourself of this in a simple experiment.

First, route the wires correctly. Make the connection between the Arduino and your SBB with twisted pairs composed of the SDA line and a ground line, and between the SCL line and another ground line. Route these two twisted pairs from your Arduino to the SBB connection.

Using the SCL line to trigger the scope, measure the noise on the data line, when the SCL Line switches. This is the minimum cross talk from the cabling.

Next, route the cables in the worst case. This is by twisting together the SDA and SCL wires, with a separate ground connection from the Arduino board to the SBB. In this configuration, you will have the maximum cross talk.

Do the same measurement as before, triggering the scope on the SCL line and measure the switching noise on the SDA line.

You will see a much larger amount of noise.

When you use twisted pair, you always want to twist signal and return wires together, never two different signals.

32.8 In your report

There is no report for this lab. Focus on a short and concise report on the SE and DiFF lab (lab 16)

32.8.1 Grading rubric:

1 point for check off by your TA

Chapter 33 Lab 18: Measure the in rush current and operation current of a board

In this lab you will explore a simple method of measuring the in rush current and steady state current draw that a board draws from the power supply.

The basic method is using a sense resistor in series with the power rail. This converts the current into a voltage which can be measured by a scope.

The challenge is that you need to be able to measure a small voltage across the resistor, not a voltage referenced to ground.

There are three simple methods to measure this small voltage difference:

1. *Using two single-ended scope probes*
2. *Using an AD2 scope*
3. *Using an instrumentation amplifier (such as an AD623)*

In this lab, we will illustrate using two single-ended probes. You can experiment on your own with the AD2 scope which has true differential inputs, or with the third method using a simple AD623 instrumentation amplifier.

33.1 The principle of a current sense resistor

Every circuit will draw some current. This current will change depending on the load the circuit represents to the power supply. Initially, when the circuit is plugged into the power supply, the current draw by the circuit can be very large, as all the decoupling and other capacitors are charged up by the power supply.

We call this initial current transient the "in rush" current. To first order, it is related to the Thevenin resistance of the voltage supply, which can be as low as 1 ohm or less. If the rail is 5 V, the initial current can be as high as 1-5 A. This is huge. There is usually something else that limits it, but it can still be very large.

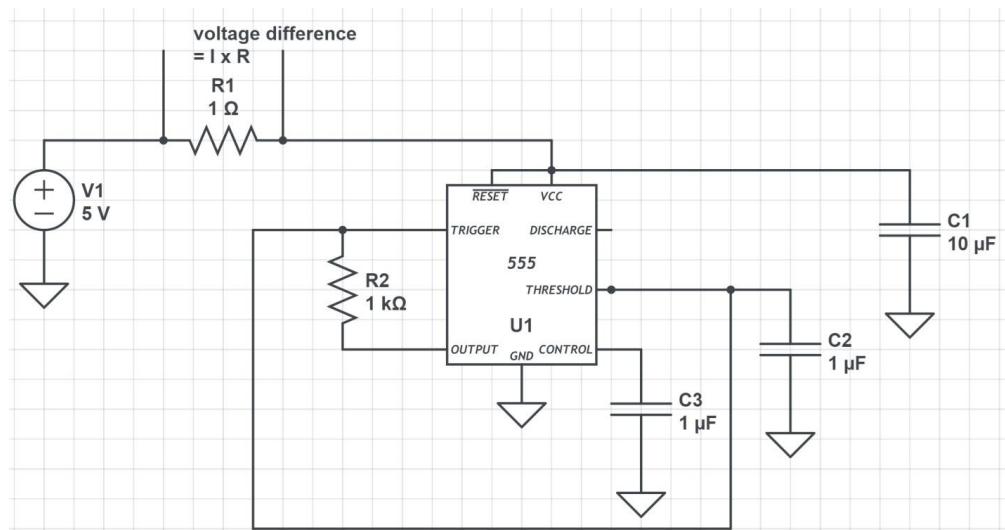
Once the initial in rush current has dropped, the circuit will draw some amount of current based on its operation. If it is doing something periodic, you may see some periodic current flow in the power rail.

The way we measure the current flow in the power rail is by adding a series resistor as a current sense resistor. Its value should be large enough to produce a measurable voltage, but small enough to have this voltage be a small value compared to the power rail. Afterall, the voltage drop on the resistor is the DC voltage drop in the power rail.

For example, if we expect a steady state current of 0.01 A, and we can reasonably measure a voltage of 20 mV, then the resistance might be: $R = V/I = 20 \text{ mV}/10 \text{ mA} = 2 \text{ ohms}$. If the current were to go up to 1 A, the voltage drop in the sense resistor would be $V = I \times R = 1 \text{ A} \times 2 \text{ ohm} = 2 \text{ V}$. This would be intolerable for steady state, but might be ok for the transient inrush current.

Selecting the value of sense resistor to use is a little tricky and requires knowledge of the properties of your circuit. Sometimes, you have to use trial and error to find the best series sense resistor.

The sense resistor should be placed between the external power source and the circuit being powered. This is shown in the figure below:



Make sure that the power supply connects through the sense resistor.

You might have a series resistor built into your brd 3 golden Arduino. In this case, you can use this board to measure the inrush and the steady state current.

33.2 Build or select a circuit in which to measure the current draw

The method described here, of adding a series sense resistor on the high side of the power rail, can be applied to any circuit. You could add a series sense resistor in series with a power jack to measure the current draw when any device is plugged into the power jack.

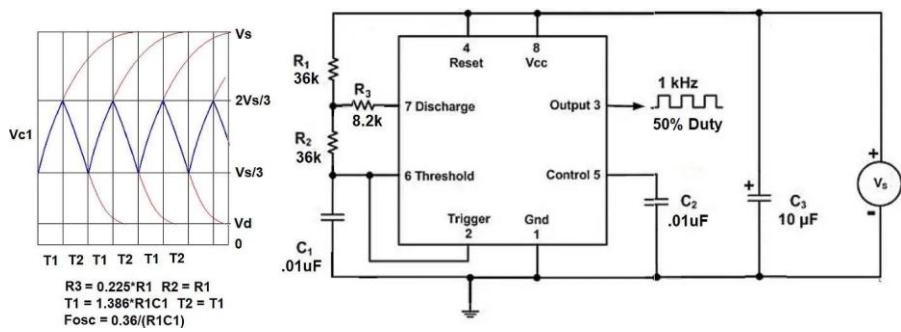
For example, you could plug your commercial Arduino board into a circuit that has a power jack cable with a series resistor. You measure the voltage across the series resistor with two 10x probes, using the ground lead of the power cable as the reference for the two 10x probes.

If you want to use this method, you can ask for a power jack cable and solder a resistor in series with the power line.

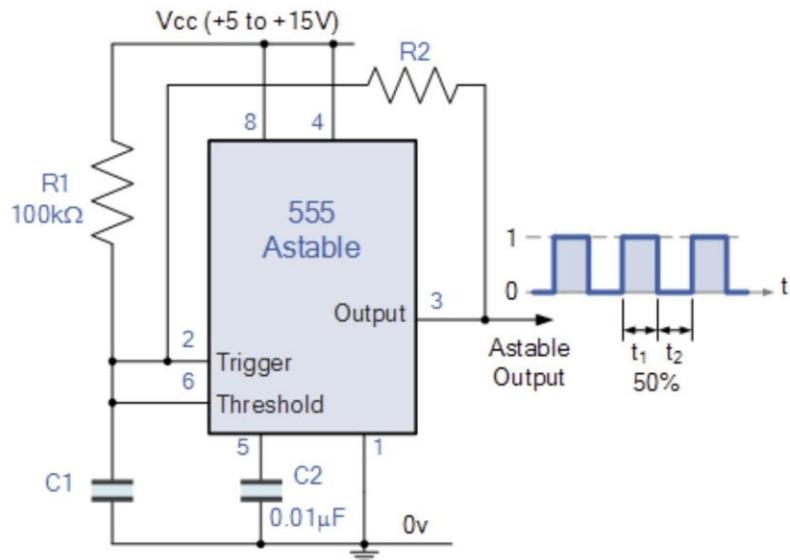
33.3 A simple 555 timer circuit

You can use any circuit you want as a test circuit to measure the current. Feel free to use your 328 28 pin DIP, your commercial Arduino board, or a hex inverter circuit, or a simple 555 timer circuit. One example of a 555 circuit is described here.

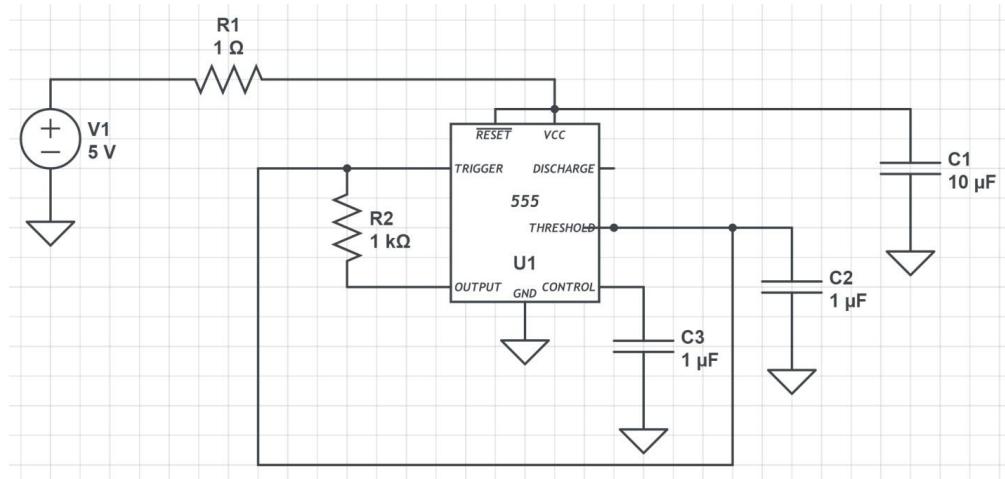
You can try this circuit from an [EDN article](#) that will give you a true 50% duty cycle square wave. This circuit is shown in the figure below.



This is an even simpler circuit that also gives a 50% duty cycle, [from this source](#):



It can be simplified to:



The output signal will always be inverted from the voltage on the trigger pin. By applying feedback, we have built a ring oscillator. The output will switch state when the trigger level oscillates between 1/3 and 2/3 of the Vcc value. If the output voltage goes between 0 V and Vcc, the charging and discharging time will be the same.

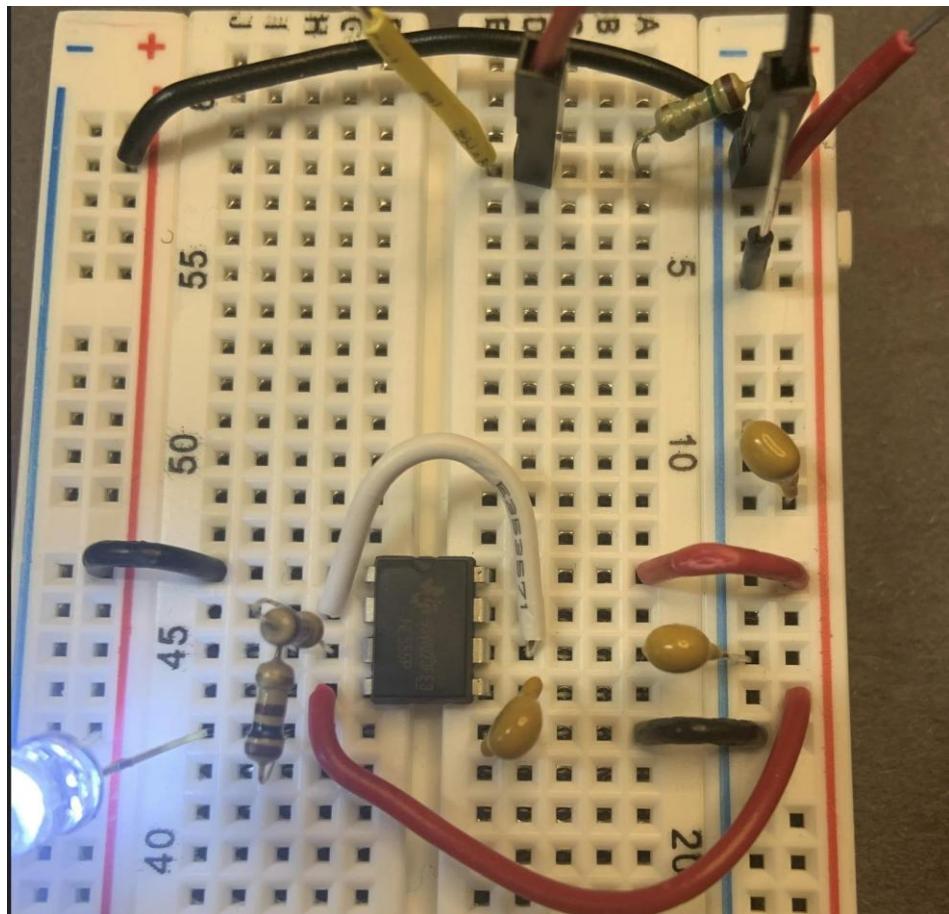
For about a 1 msec charging and discharging time, I am using a 1 k resistor and 1 uF capacitor. This is about a 500 Hz frequency.

However, this circuit is based on having the output voltage level being the Vcc voltage. If you use an NE555, the output voltage is rated at only 3.3 V with a 5 V rail. This means the charging time will be longer than the discharge time and the output will not be 50% duty cycle. But it will be close enough.

The details are not important. This circuit will be the reference circuit from which to measure the supply rail current.

In order to generate some reasonable current in the circuit to actually measure, I connected the output of the 555 into an LED through a 100 ohm resistor. In this circuit, the high output voltage of the 555 is measured as 3.45 V. The forward drop across the white LED I am using is about 2.75 V. The current limiting resistor in the LED circuit is 100 ohms. I expected a current of about $(3.45\text{ V} - 2.75\text{ V})/100\text{ ohms} = 0.7\text{ V}/100\text{ ohms} = 7\text{ mA}$.

I used the 5 V rail of an Arduino to power the 555 circuit. This circuit is shown in the figure below.



33.4 Measuring a differential voltage with two single-ended probes

In previous labs we identified that single-ended signals, or voltages, are measured with respect to a local ground. This is what the 10x scope probe measures. The voltage on the tip is always relative to the local ground of the cable, which is connected to the chassis of the scope, which is connected to the earth connection in the power plug where the instrument is plugged into the wall.

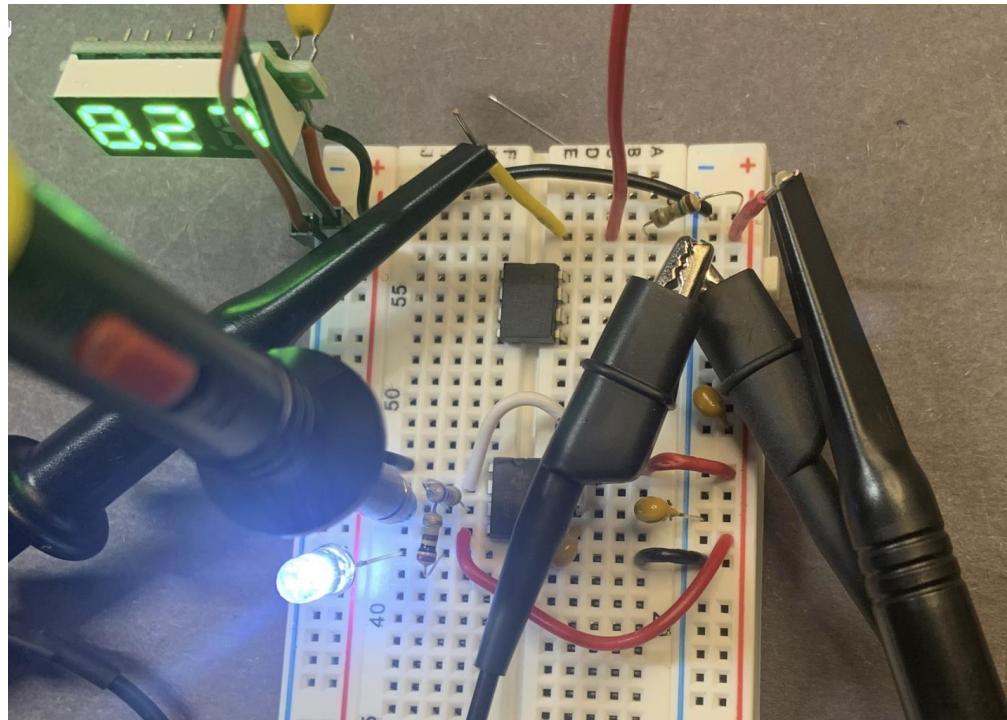
If you try to add the 10x probe tip on the high side of the resistor and the ground strap to the low side you will short out the power rail to the board, and effectively drive the short circuit power supply current through the ground strap of the 10x probe.

In addition to not measuring what you expect, you will probably damage the 10x scope probe. **DO NOT DO THIS.**

However, we can use two scope probes to measure the high side of resistor, relative to local ground, and the low side of the resistor, relative to local ground. The difference between these voltages is the voltage across the resistor.

In principle, this is a perfectly fine way of measuring the differential voltage across the resistor. In practice, what we are doing is measuring two large voltages, close to 5 V and taking the difference to measure a small voltage. Due to the absolute accuracy variations and the noise on each channel of the scope, this is not a very effective measurement method. It will have a lot of noise. But, it is really easy to do and should be the first thing to try.

This configuration with the two probes attached to either side of the sense resistor is shown in the figure below.

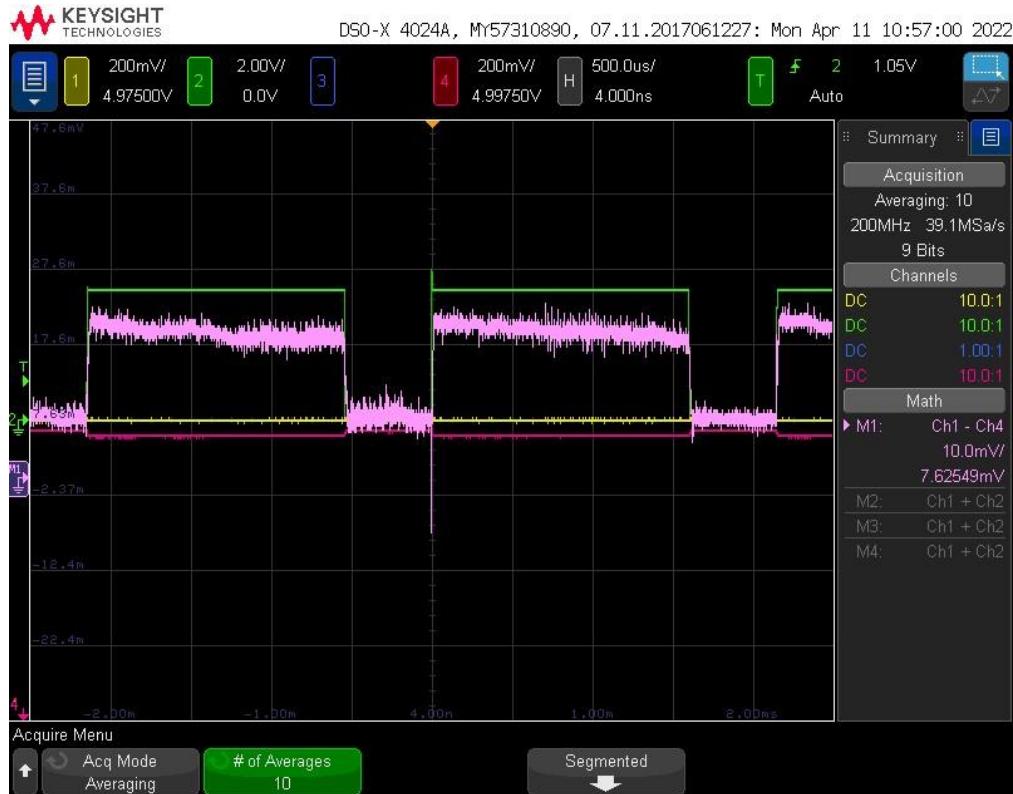


Two separate probes are used to measure the voltage across the series resistor. Each probe is referenced to local ground.

The difference between the two channels, in this case, channels 1 and 4, is calculated using a math function. This difference is displayed on the scope. Due to the large amount of noise, I averaged 10 consecutive acquisitions.

I used the output voltage of the 555 to trigger the scope. This way I could see the voltage across the sense resistor synchronous with the output switching.

The voltage across the 1.5 ohm sense resistor when the 555 switched, is shown in the figure below.



The yellow is the voltage on the high side of the sense resistor, the red is the voltage on the low side of the resistor. The green is the output of the 555 which triggers the scope. The pink trace is the voltage difference, on a scale of 10 mV/div. This is the voltage across the sense resistor.

When the 555 is on and the LED is lit, there is a voltage of about 11 mV across the 1.5 ohm resistor. This is a current of about $I = 11 \text{ mV}/1.5 \text{ ohms} = 7.3 \text{ mA}$. This is very close to our expectation of 7 mA.

33.5 Measuring the inrush current with two single ended probes

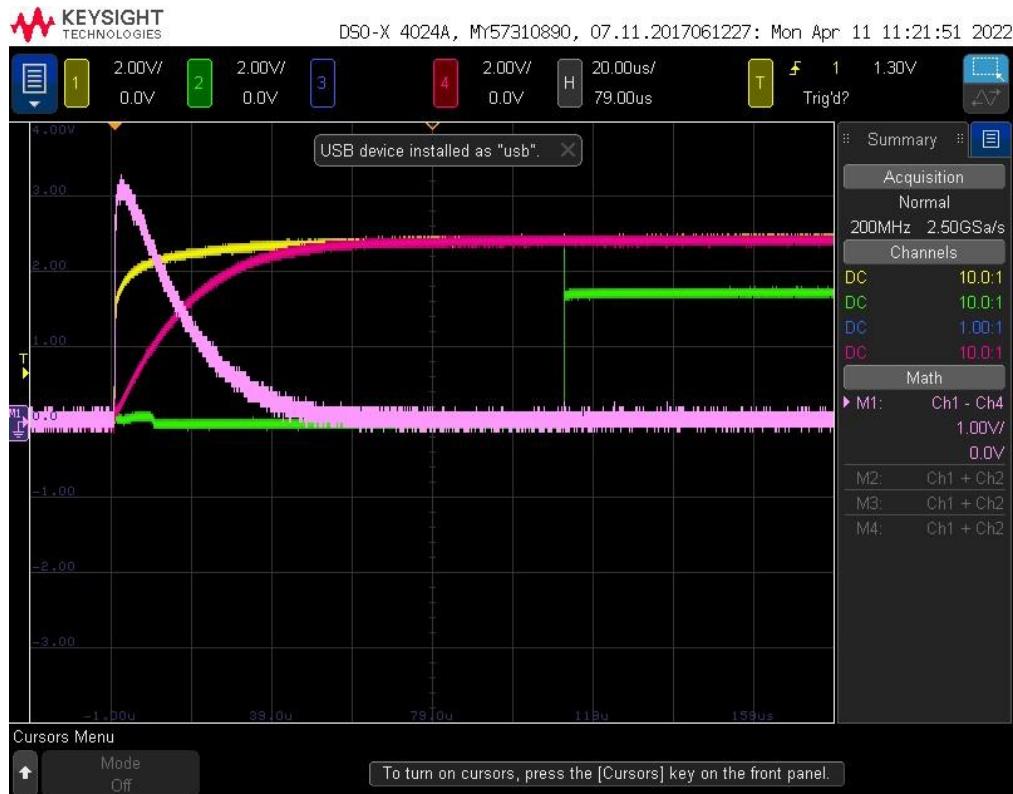
With this same circuit configuration, we can measure the inrush current from the power supply when we plug the power plug into the 555 circuit. The reason we expect inrush current is because we have a 10 uF decoupling capacitor on the power rail. This will be charged every time we plug the circuit into the power supply.

The secret to measuring inrush current is to trigger the scope on the rising voltage on the resistor and display the voltage difference which is the inrush current.

To only trigger the scope when the voltage goes up, and not all the time when there is no trigger event, we use the Normal mode. The scope will only trigger when the voltage on the high side of the resistor goes above our set threshold, about 1 V in this case. This will happen once when we plug the power plug into the circuit.

Remember, when the scope is set on Auto mode, the scope will trigger only on a trigger event, unless no trigger event comes in within about 50 msec, and then the scope will automatically trigger showing what is currently measured. When you start out, use the Auto mode to set the trigger, and then use Normal mode for a transient event. DO NOT use the stop button.

The figure below is the scope measurements of the difference voltage in pink when the voltage on the high side of the resistor, on channel 1, exceeded about 1 V.



In this measurement, I turned the averaging off, since this is NOT a repetitive signal, but a once only transient signal.

It takes a little trial and error to adjust the scales to capture this transient event. The measured voltage across the 1.5 ohm sense resistor, the pink trace, is measured on a scale of 1 V/div. The peak voltage is 3 V. This is a peak current of $3\text{ V}/1.5\text{ ohms} = 2\text{ A}$. There is 2 A of current flowing into the power rail to charge up the capacitance on the power rail!

But, this current lasts for only a short period of time. In this case, it is down to a very low level within 40 usec.

Once this circuit and the scope is set up, you can change the decoupling capacitance on the circuit's power rail and explore the impact on the in rush current.

33.6 In your lab report, you should:

1. Explain the principle of measuring the power rail current
2. Show an illustration or photograph of how you implemented the series resistance circuit and how you set up the scope and its probes to measure the voltage across the resistor.
3. Show an example of the circuit you built in which to measure the current draw. Describe how you estimated the value of the series resistor to use.
4. Show an example of the steady state current draw of your circuit while it is doing something
5. Show a measurement of the inrush current and then when the decoupling capacitor is changed. 6.
Summarize the so what of what you observed

As always, write up your lab report so that a hiring manager will be impressed with what you know and so that you can teach them an important principle. Very few engineers have done these sorts of measurements of circuits.

Add pictures and scope screens.

33.7 Grading rubric

As with all the labs, the rubric is

1 point for check off by your TA

1 point for turning in your report on time

1 point if you have a coherent explanation of what you did with good illustrations and figures correctly analyzed.

Chapter 34 Lab 19 4-layer via to via cross talk board

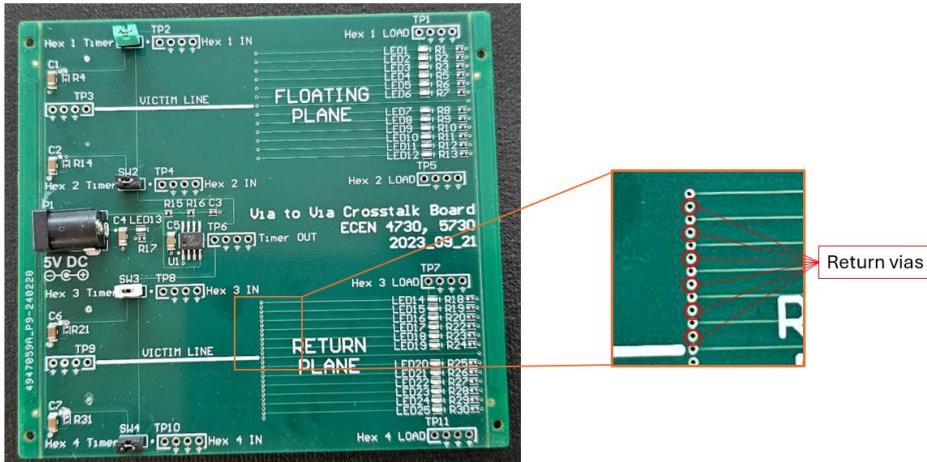
This lab will give you a chance to measure the impact on cross talk noise when signals pass through vias from layer 1 to layer 4 with and without return vias.

The principles of designing four layer boards is described in chapter 23 of the textbook. You should review this chapter before doing this lab.

Two boards will be provided for you. They are both 4-layer and are of identical construction. The stack up is:



Here is the test board:



The bottom section of the board labeled with, "return plane" has a return via adjacent to every signal via. These return vias are connected to layer 2 and layer 3. The return via provides a way for return currents to transition between the two planes. Without these return vias, the return current would need to search for a way to transition between these two return layers.

The top section of board labeled with, "floating plane" uses layer 2 as the ground and layer 3 is a floating separate plane meant to mimic a power plane in a design. When layer 3 is a power plane and floats, the return current has to couple through the impedance of the cavity and we see the long range cross talk as noise coupled in the cavity of layer 2 and 3. In practice, there would be capacitors between layers 2 and 3, and these would decrease the impedance and decrease the cross talk. However, the plane floats to show a clear effect and make the cross-talk easy to see.

On the top signal layer are the signals from the bottom layer going to LEDs and resistors. On the bottom layer are the hex inverters which drive the signals. They have signals that route from their I/O to the center of the board, and then to the LEDs. The resistors are about 47 ohms. The voltage across them is about 1.5 V when the LEDs are on. This is a current of about $1.5 \text{ V} / 47 \text{ ohms} = 30 \text{ mA}$ through each channel.

There are 12 hex inverters that switch. This is a total current of 0.36 Amps that flows from the bottom of the board to the top of the board. The return current starts on the layer 3 plane and then transitions to the layer 2 plane. When the signals pass through the signal vias, the return current flows through the impedance of the planes.

With no return vias, this creates noise across the impedance of the cavity. This noise is picked up in the victim trace. With a return via between the planes adjacent to each signal via, the impedance between the two planes is very small. This means there should be very little noise on the victim trace.

Note that this hex inverter is the 74AVLC14, which is rated for only 3.6 V max Vcc. We are using it at 5 V. It may get warm to the touch and its lifetime will be shorter. But, it will also have a very short rise time. It will live a short but glorious life.

34.1 Exp 1: Measuring the cross talk when there are no return vias

Using the board with no return vias, measure the current through the resistor with the LED. Compare this signal with the output from the 555 timer.

Note the inverted signal.

The falling edge is typically a little shorter than the rising edge. We will see more switching noise on the falling edge of the inverters.

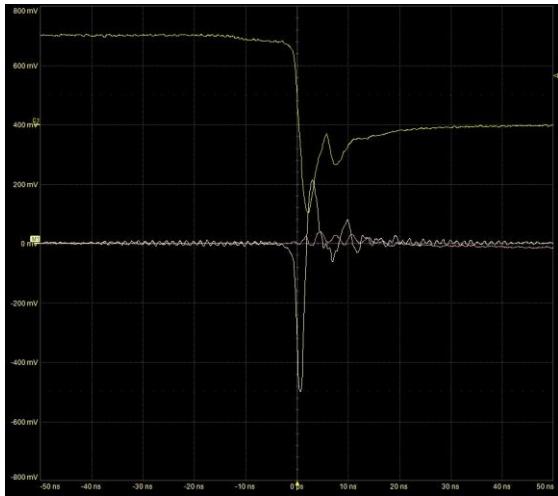
Measure the voltage on the victim line with the falling edge of the current. What is the peak to peak value of the noise?

34.2 Exp 2: Measuring the cross talk when there are return vias

In the second board, the layout and the hex inverter circuits are identical. The only difference is that there are return vias between the two planes in this board, adjacent to every signal via. This means the impedance the return current passes through from the bottom plane to the top plane is much lower. With less noise between the two planes, there will be less noise picked up on the quiet, victim line.

Using the current through the signal line to trigger the scope, measure the noise on the victim line when there are adjacent return vias, compared to the case with no return vias.

Here is my measurement comparison:



The top trace is the current through one of the 47 ohm resistors. The straw colored trace is the noise on the victim trace with no return vias. This is about 700 mV peak to peak. The rose colored trace is the noise on the victim line with the return vias, on the same scale. It is about 50 mV peak to peak.

34.3 In your report

There is no report for lab 19, however you must have a TA checkoff for this lab (1 point).

Chapter 35 Brd 3 assemble, bring up, boot load, test

Chapter 36 Lab 20: Switching noise in commercial Arduino and golden board

36.1 What you will do in this lab

In this lab, we will analyze the switching noise on a **commercial Arduino board** using a special, custom shield. I will walk you through the measurements and then you will do the same measurements with the Keysight 4024 scopes in the lab.

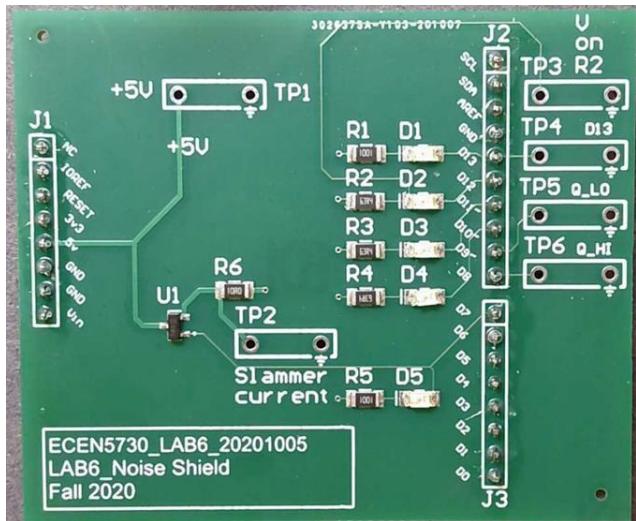
After you have completed the measurements on a commercial Arduino board you will repeat the exact same measurements on your brd 3 Arduino board and compare the results.

Very important to note: The noise you measure in this experiment is all about switching noise. This sort of noise is always driven by the dI/dt . While the dI will probably be the same between the commercial Arduino board you are using and your Golden Arduino board you will measure in a later lab, the dt may not be the same.

In order to do a good comparison and to interpret the results you are seeing, be sure to measure the rise and fall time of the I/O signals and the changing currents, when you can. If the rise time is shorter in your golden Arduino board than the commercial board, you would expect to see larger switching noise, if all things are equal. This is important in your interpretation.

36.2 The switching noise shield

The shield will enable driving significant current through three I/Os and measuring the switching noise. The figure below shows the top view of the shield. The bottom layer is a solid, continuous ground plane.



In this boards, we will use:

- D13 (TP4) as a trigger output to trigger the scope. All I/O will be synchronous with pin 13. Connected to pin 13 is an LED, 1k resistor, and a 10x scope probe test point, TP4.
- D12, D11, D10 each connect to an LED and a 49 ohm resistor. These will drive some substantial current when these pins switch. They will generate the switching noise.
- D9 is connected to a test point, TP5. We will use D9 as a quiet low and keep its output always low.
- D8 is connected to a test point, TP6. We will use it as a quiet HIGH test point.
- TP1 is connected to the 5 V rail on the board.

D7 is another I/O we will use to trigger a MOSFET. This pin drives the gate of a MOSFET with a 10 ohm resistor between the source and ground. When the MOSFET is on, it will connect the 5 V rail to the 10 ohm resistor, drawing about 400-500 mA of current.

We will measure the switching noise on the board and on the quiet-HIGH when we draw the current and generate the voltage drop on the 5 V rail. This MOSFET circuit is often called a slammer circuit.

36.3 Exp 1: Write the microcode

We are going to write the code to trigger pin 13. We will use this as the trigger for the scope.

Then we will turn on D12 and look at the switching noise, then D12 and D11, so two outputs switch simultaneously. Then D12, D11, D10 all three switching simultaneously.

So that we can see these happening all at once we will use the port command and have them switch right after each other so we can see this all on one screen, at one time.

Here is the simplified code:

```
void setup() {    DDRB =  
B00111111;    pinMode(7,  
OUTPUT);  
digitalWrite(7, LOW);  
}  
void loop() {    PORTB =  
B00111101;  
delayMicroseconds(4);  
PORTB = B00000001;  
delay(1);  
    digitalWrite(7, HIGH);  
delayMicroseconds(400);  
digitalWrite(7, LOW);  
delay(10);  
}
```

All we do is turn on D13, D12, D11, D10 at the same time using the port command. Then we set D9 as a quiet LOW, pegged low, and D8 as a quiet HIGH, pegged HIGH.

The second part of the sketch is to turn on the D7 pin which will drive the slammer circuit. Note: on some of your boards, the signal to the MOSFET gate may be pin 1. You will have to trace these connections for your board.

36.4 Exp 2: Measuring the switching noise on the quiet LOW

You will use pin13 to trigger the scope and use channel 2 to measure the other pins in turn.

You will measure:

The current thru the 49 ohm resistor

The quiet low

On the rising and falling edge

You can modify the code and turn off all the higher current switching I/O and only see the noise when pin 13 switches.

From these measurements:

Analyze the ground bounce switching noise and offer any interpretation from these measurements.

These are the very same measurements you will do on your Arduino board.

Don't forget to apply rule #9: always anticipate what you expect to see, then look at the results and compare them with your expectations.

36.5 Exp 3: Measure the noise on the power rail when the microcontroller itself is the aggressor

When all the I/O switch their currents, this current ultimately comes from the 5 V power rail on the board. But, it must travel through the inductance of the power rail and through the package leads of the 328 uC. This inductance will cause a voltage response on the on-die power rail.

When the I/O switch, measure the voltage on the quiet HIGH. This is connected to the power rail on the die.

The chip and its I/O are the aggressor source of noise on the power rail. How much of this noise gets to the board power rail?

Compare the voltage on the Quiet HIGH pin and the voltage on the board level power rail when the chip is the aggressor.

36.6 Exp 4: Measure the on-board power rail noise when the board is the aggressor

The digital pin D7 connects to a MOSFET. The MOSFET is just going to draw some current on the 5 V rail from the board. This is the same circuit you built in Lab 3 when we studied the PDN. In that lab you used a NPN transistor instead of the MOSFET.

You should be able to draw the schematic of this slammer circuit based on my description above and what you did for the PDN noise lab at the beginning of the semester.

When D7 turns on, the MOSFET turns on and current flows through the 10 ohm resistor. You will measure the current through the 10 ohm resistor, its rise time and the voltage noise on the 5 V rail on the board and on the Quiet HIGH pin, on the die.

Trigger the scope with the voltage across the 10 ohm sense resistor. In this case, the noise is being generated on the board level power rail. The 5 V rail on the board is the aggressor. The on-die 5 V rail is the victim. You will measure how much noise is on the 5 V rail on the board and how much is on the die using the quiet HIGH pin, on both the rising and falling edges of the current.

Very important to note: in the PDN switching noise lab, you found that the droop on the power rail you measured at the collector of the switching transistor was due to the inductance from the collector to the nearest decoupling capacitor or back to the VRM source. It is the dI/dt thru this inductance that causes the switching noise on the power rail.

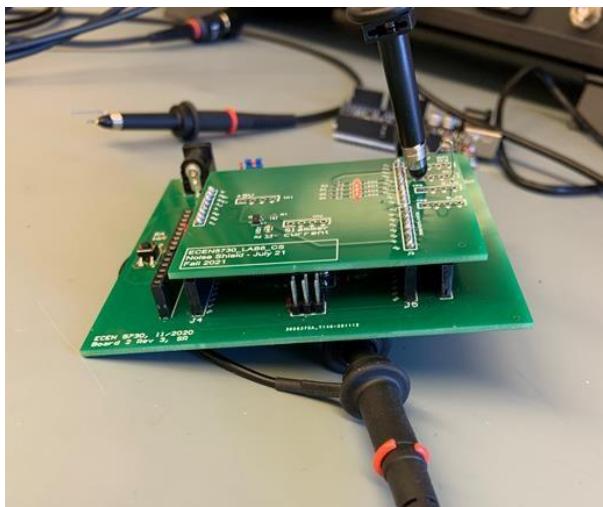
In the noise shield, trace the path from the drain of the MOSFET, which is where you are measuring the board level 5 V rail, back to the nearest decoupling capacitor. The dI/dt and this inductance will determine the power rail switching noise you measure on the 5 V rail. This is the value you will compare with your Golden Arduino.

36.7 Exp 5: Measuring the near field emissions from bottom of the board

As a final measurement, we will use the 10x probe with the tip shorted to the ground return making a small loop as a pick-up probe to measure the near field emissions from the board.

While the slammer shield is connected and the code is running, use pin 13 to trigger the scope. Move the pickup probe around underneath the board. Look at the magnitude of the noise picked up. Can you find a place to position the 10x probe loop under the board that gives the most amount of rf pick up noise?

This is near field emissions. It will often turn into far field emissions and may cause EMI failures. Here is the set up to place the 10x probe as a pickup loop under the board while the scope is triggered by the trigger from pin 13.



36.8 Exp 6: generic measurements to characterize any Arduino Board

These measurements are standard measurements for any Arduino board. You will perform these measurements on your commercial Arduino board and then on your version of the Golden Arduino board.

These same measurements will be used for both boards. Use the commercial Arduino board to develop the skills in performing the measurements. When you have an idea of how to do these, then you can set up your golden Arduino and perform these same measurements.

36.9 Your Lab Report

This lab report will most likely be part of your bigger board-03 report. Use the following guidelines to include these data into your board 3 report.

Be sure to be checked off by your TA. This is of benefit to you as you will need to understand these measurements in order to include them in your final report.

In the lab report for this lab, you only need to include the measurements from the commercial Arduino board. You can include these same measurements in your final report of your Golden Arduino board.

The following are some questions to use as a guideline of what to cover in your lab report. You will need to understand these answers to be checked off by your TA before the end of the lab. Pick a few of these to focus on in your lab report.

When the three I/O were switching, what was the total current switching, the duration and the rise time? Use a plot of the measured current to justify your analysis.

1. *What was the quiet HIGH and quiet LOW noise on the die for the rising edge? Compare these on the same plot.*
2. *Do the same analysis for the falling edge. Any comments about this noise?*
3. *When the I/O current was switching, what was the difference in switching noise on the 5 V rail on the die and on the board? Compare the two measurements on the same plot.*
4. *When the slammer circuit triggered, what was the current flowing through the 5 V rail on the die? What was the duration and the rise time?*
5. *What was the voltage drop on the 5 V rail at steady state and the current flow? What does this suggest as the output Thevenin source resistance of the 5 V power rail?*
6. *What was the switching noise on the 5 V rail during the rising edge? On the quiet HIGH rail on the die?*
7. *What was the switching noise on the 5 V rail during the falling edge? On the quiet HIGH rail on the die?*
8. *What do you conclude about noise on the die getting onto the board and noise on the board getting on the die?*

Remember, in your report, when you show scope captures:

Adjust the scales to read important information right off the front screen

Only use cursors or measurement functions if the scale is set up conveniently and your measurement matches your mark 1 eyeball estimate.

If you are looking at switching noise, always trigger the scope on the switching signal so you know if it is switching low to high or high to low.

Always include your analysis of what the measurement tells you- answer the "so what?" question.

Pictures of the board and the measurement set up will be very useful to illustrate what you are measuring.

36.9.1 Grading rubric:

Most semesters, these data are used in your bigger board 3 report. If there is a report, the scoring for this report is:

- 1 point for check off by your TA
- 1 point if you show examples of what you measured and your analysis
- 1 point if you review the principles behind the circuit and the measurements coherently.

Remember, this report will look great in your portfolio.

Chapter 37 Lab 21 SBB version of brd 4

In this lab, you will build a solderless breadboard version of your brd 4 to demonstrate the basic operation and write some simple code to perform the measurements.

You are building an intelligent measurement system, called an instrument droid. It is a custom instrument specifically designed to characterize any voltage source, or voltage regulator module (VRM) by measuring its Thevenin voltage and the Thevenin resistance as a function of output current.

You will build an electronic load to draw some controlled current and measure the voltage drop on the VRM with the current load. With no current, the voltage on the VRM is the Thevenin voltage. With a current load, the voltage drop is the voltage across the internal Thevenin resistor. Knowing the voltage drop and the current through it, you can calculate the Thevenin resistance.

As you increase the current load, you will print out:

Current, Thevenin voltage, Thevenin resistance, loaded voltage.

You can either print these numbers to the serial monitor and copy and paste them directly into excel to plot the Thevenin resistance vs current load, or you can stream the measurements directly into excel to print them up as you go.

You will measure the Thevenin resistance vs current for a few different VRM sources to test out your system.

37.1 Power consumption considerations

This instrument droid may be capable of drawing as much as 3 A of current from the VRM. If this current flows through a 1 ohm sense resistor, the power dissipation could be as high as 9 watts. This will blow up a resistor rated for only 0.2 watts.

To avoid this potential problem you will not keep the current on for very long. If you keep the duty cycle down to 2%, the maximum average power dissipation would be $9 \text{ watts} \times 0.02 = 0.18 \text{ watts}$, not a problem. When the current is on, there will be 9 watt of power consumption. The resistor and MOSFET and other components may heat up. But, as soon as the current is turned off, the temperature will drop. It is important to keep the average power dissipation below about 0.2 watts.

The MOSFET may have a worst case power consumption of $12 \text{ v} \times 3 \text{ A} = 36 \text{ watts}$. If you keep the duty cycle to 2%, this is only $36 \text{ watts} \times 0.02 = 0.7 \text{ watts}$. This may be acceptable for the MOSFET on your board. Always perform a worst case analysis of the power consumption of any component when dealing with potentially high currents.

In your code, you will write an algorithm to pulse a current on, perform the measurements, turn the current off, wait some time and then do it again with a different current.

If the max current is just 0.3 A, the peak power consumption in a 10 ohm resistor is 1 watt. Even a 10% duty cycle would be fine.

37.2 Setting a max current, or max voltage drop

We want to sweep the current from a low value to some maximum value. This will be determined in a loop. But, if the voltage on the VRM drops below some value, compared to the start value, we also want to stop the measurements. These limits will be added to the code.

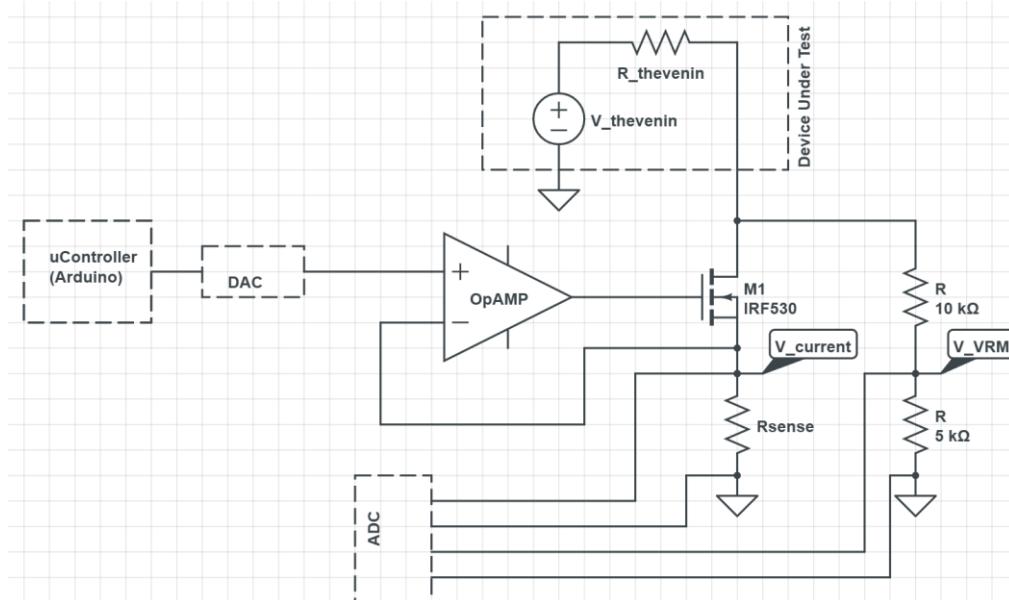
The DAC is 12-bit, or 4095 levels and goes from 0 V to 5 V. This is about $5 \text{ V}/4095 = 1.2 \text{ mV}/\text{level}$. This is our voltage resolution. If we want the smallest current change to be 1 mA, then we want a sense resistor of $R = 1 \text{ mV}/1 \text{ mA} = 1 \text{ ohm}$. If we want 0.1 mA resolution, then we want 10 ohms sense resistor.

If we use a 1 ohm sense resistor, then the maximum voltage across it might be $5 \text{ V} - 2 \text{ V} = 3 \text{ V}$. The 2 V is roughly the MOSFET threshold. This would be a max of $3 \text{ V}/1 \text{ ohm} = 3 \text{ A}$ as the current we could get through the system.

You can decide if you want a system that will go from 1 mA to 3 A or 0.1 mA to 0.3 A. This determines the sense resistor and the duty cycle so you do not burn anything out. I recommend using either a 10 ohm sense resistor or maybe a switch to go between a 1 ohm or a 10 ohm sense resistor.

37.3 The circuit

The actual circuit you will build is shown below.



You will use an external DAC, the MCP4725 to generate a voltage which will match the voltage across the sense resistor. If the sense resistor is 10 ohms, then a 1 V out of the DAC will turn on the MOSFET so that there is 1 V across the 10 ohm resistor, or a current of $1\text{V}/10\text{ ohms} = 100\text{ mA}$.

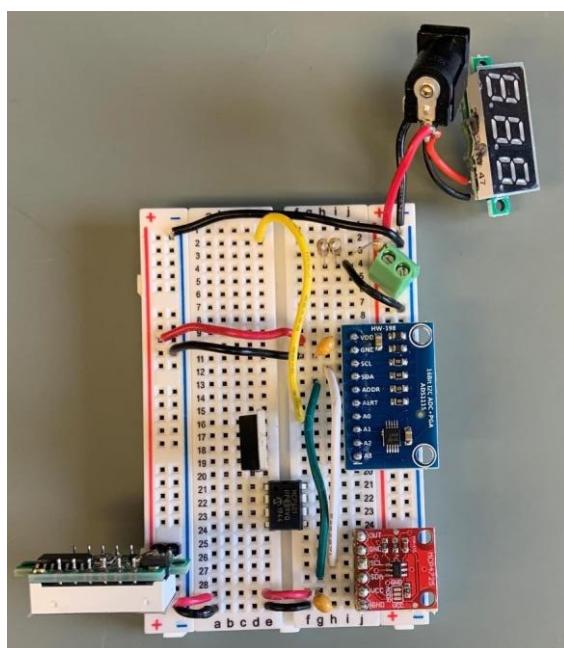
You will measure the voltage across the sense resistor with a 16 bit ADC, the ADS1115. You will also measure the loaded voltage of the VRM.

However, since we want to measure VRM voltages as high as 12 V, and the max voltage the ADS1115 can measure is 5 V, we will add a 3:1 voltage divider on the VRM output voltage and measure this voltage.

It is better to use a MOSFET than a transistor because the current through the sense resistor will be the same as the current through the VRM. Using a transistor, there will be additional base current that flows through the sense resistor but not through the VRM.

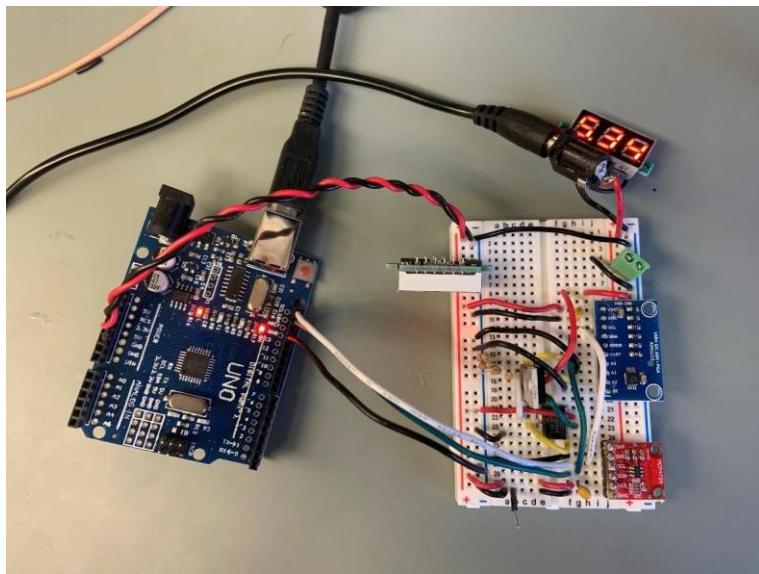
The specific opAmp you use is not critical. Any opAmp will do. You may have an MCP601 or 602 or even a 6001. Just be sure to select the correct footprint and pin out from the library.

You will build this circuit in a solderless breadboard first. It will be similar to that shown below:



The 5 V power for the ADC and DAC and op amp will be supplied by the Arduino board. The VRM will be plugged in either into the power jack or the terminal block. These voltages are separate, but the ground is common.

Here is my version of the final board and my Arduino:



37.4 Building the circuit and the code

When building this circuit, you can do it piece by piece and test it out element by element, or connect the whole thing up and then write the code to test it. It is a personal preference.

If you do not have a lot of experience building circuits, you should do it step by step. This will make it easier to debug.

37.4.1 Set up the DAC

First, connect the DAC and the I2C connections to the Arduino Uno. The DAC is the MCP4725. You can find information about it here: <https://www.adafruit.com/product/935>

Once wired up, to start using the MCP4725, you'll need to install the Adafruit_MCP4725 library. The library is available through the Arduino library manager.

From the Arduino IDE, open up the Library Manager and search for Adafruit MCP4725. Install the latest version. Normally, the examples in a new library are great, but the included examples in this library are too complicated. Instead, you can just run this code:

```
// vrm characterizer board
```

```
#include <Wire.h>
#include <Adafruit_MCP4725.h>

Adafruit_MCP4725 dac;
float DAC_ADU_per_v = 4095.0 / 5.0;           //conversion from volts to ADU
int V_DAC_ADU;                                // the value in ADU to output on the DAC

void setup() {
  Serial.begin(115200);
  dac.begin(0x60);                // address is either 0x60, 0x61, 0x62, 0x63, 0x64 or 0x65
  dac.setVoltage(0, false); //sets the output current to 0 initially
}

void loop() {
  V_DAC_ADU = 0 * DAC_ADU_per_v;    dac.setVoltage(V_DAC_ADU, false); //sets the output
  current to 0 initially
  V_DAC_ADU = 3.0 * DAC_ADU_per_v;  dac.setVoltage(V_DAC_ADU, false); //sets the output
  current to 0 initially
}
```

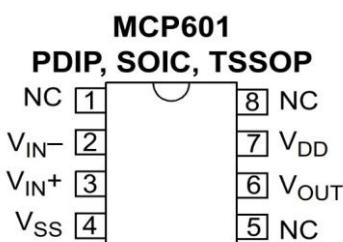
The first thing you will have to do is determine the address of your MCP4725. Mine used the default address of 0x60. You can reverse engineer this address. Just try an address and see if you can change the output voltage on the DAC, as measured by a scope. Keep trying an address until the output voltage changes.

Now you can talk to the DAC.

37.4.2 Set up the opAmp.

In my example, I am using an MCP601. Check the pinout of the one you are using. Note: this is a through hole part. In your board, you will be using a SMT part, and it may not be this exact op amp. Any opAmp will do.

Here is the pinout for my opAmp:



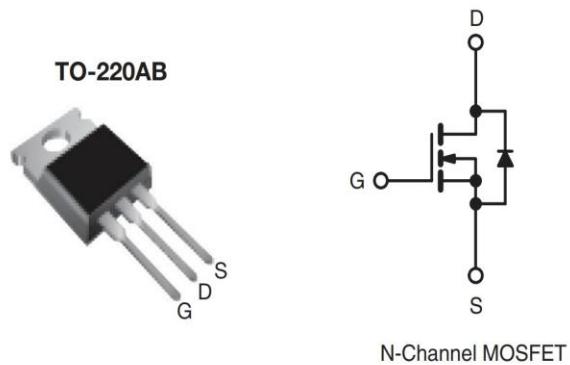
37.4.3 Set up the MOSFET

The specific MOSFET I am using in this SBB circuit is the IRL520. It doesn't matter which MOSFET you use.

Almost any will work the same in this application. But, in your brd 4, you will use a SMT MOSFET that is in the JLC library. This is the one I will stock in inventory. Do not use the 520 through hole part. I have a limited supply of these and they should stay in your lab kit.

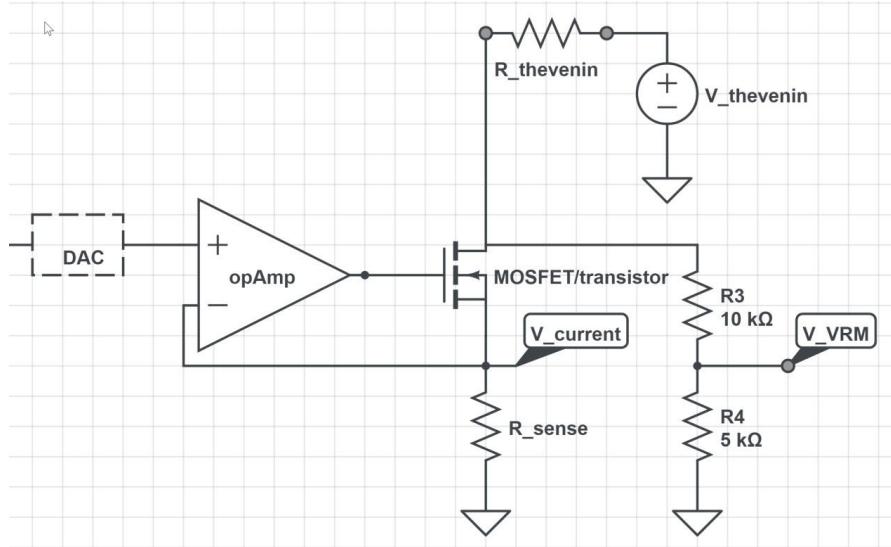
Here is the datasheet:
<https://www.vishay.com/docs/91298/sihl520.pdf>

Any MOSFET will work in this application. Just be careful of the pinout. The pinout for this MOSFET is:



This particular one has an on resistance of about 0.27 ohms and a turn on threshold of about 1-2 V.

Note, the circuit with the DAC, opamp and MOSFET is here:



37.4.4 Connecting the VRM under test

As a general guideline, whenever you are testing a new instrument, always start measuring something you know the answer to. As a test of our system, we are going to connect a resistor of 10 ohms in series with the

VRM. Effectively, we are creating a VRM with a known Thevenin resistance. This will also limit the max current of the MOSFET, should our code screw things up.

I am using an external VRM of 9 V. You can start with any VRM you have lying around to test this circuit.

When you think you have the DAC driving the opAmp, measure the voltage across the sense resistor and see if you get the voltage you are telling the DAC.

We are using the DAC to drive a certain amount of current. Remember, computers work for us, we do not work for computers. Write the code to select the current you want through the MOSFET and then have the DAC output the voltage you want to see across the sense resistor.

Here is my code:

```
// vrm characterizer board
#include <Wire.h>
#include <Adafruit_MCP4725.h>

Adafruit_MCP4725 dac;

float R_sense = 10.0; //current sensor
float time_on_msec = 20.0; //on time for taking measurements float
time_off_msec = time_on_msec * 10.0; // time to cool off
float DAC_ADU_per_v = 4095.0 / 5.0; //conversion from volts to ADU
int V_DAC_ADU; // the value in ADU to output on the DAC
int I_DAC_ADU; // the current we want to output
float I_A = 0.0; //the current we want to output, in amps

void setup() {
  Serial.begin(115200);
  dac.begin(0x60); // address is either 0x60, 0x61, 0x62, 0x63, 0x64 or 0x65
dac.setVoltage(0, false); //sets the output current to 0 initially }

void loop() {
  I_A = 0.0;
  I_DAC_ADU = I_A * R_sense * DAC_ADU_per_v;
dac.setVoltage(I_DAC_ADU, false); //sets the output current to 0 initially
delay(time_off_msec);

  I_A = 0.02;
  I_DAC_ADU = I_A * R_sense * DAC_ADU_per_v;
dac.setVoltage(I_DAC_ADU, false); //sets the output current to some value
delay(time_on_msec);
}
```

I am using a 10 ohm sense resistor. I want 20 mA through it. This means the DAC will output 200 mV. The opAmp will output whatever voltage is needed by the MOSFET to let through 20 mA of current. In my circuit, the voltage on the gate to drive the 20 mA through the MOSFET is 2 V. This means the threshold for my MOSFET is about 1.8 V. This is consistent with the spec.

37.4.5 Connect the ADS1115

We will measure the voltage across the voltage divider of the VRM with channel A0 and A1 and across the sense resistor with channels A2 and A3.

When given the chance, it is always better to use a differential measurement than a single ended measurement.

In my circuit the voltage on the VRM divider is A0-A1. The voltage across the sense resistor is A2-A3.

We will use the same library to drive the ADS1115 as we did in the single-ended and differential signal lab.

37.4.6 The code for the VRM Characterizer Instrument Droid

The code is structured into three parts. The beginning are the libraries and the variables used in the sketch. Here is what I ended up with:

```
// vrm characterizer board
#include <Wire.h>
#include <Adafruit_MCP4725.h>
#include <Adafruit_ADS1X15.h>

Adafruit_ADS1115 ads;
Adafruit_MCP4725 dac;

float R_sense = 10.0; //current sensor
long itime_on_msec = 100; //on time for taking measurements
long itime_off_msec = itime_on_msec * 10; // time to cool off
int iCounter_off = 0; // counter for number of samples off
int iCounter_on = 0; // counter for number of samples on
float v_divider = 5000.0 / 15000.0; // voltage divider on the VRM
float DAC_ADU_per_v = 4095.0 / 5.0; //conversion from volts to ADU
int V_DAC_ADU; // the value in ADU to output on the DAC
int I_DAC_ADU; // the current we want to output
float I_A = 0.0; //the current we want to output, in amps
long itime_stop_usec; // this is the stop time for each loop
float ADC_V_per_ADU = 0.125 * 1e-3; // the voltage of one bit on the gain of 1 scale
float V_VRM_on_v; // the value of the VRM voltage
float V_VRM_off_v; // the value of the VRM voltage
float I_sense_on_A; // the current through the sense resistor
float I_sense_off_A; // the current through the sense resistor
float I_max_A = 0.25; // max current to set for
int npts = 20; //number of points to measure
float I_step_A = I_max_A / npts; //step current change
float I_load_A; // the measured current load

float V_VRM_thevenin_v;
float V_VRM_loaded_v;
float R_thevenin; int i;
```

The next part is the set up for the ADC and the DAC:

```

void setup() {
Serial.begin(115200);
dac.begin(0x60); // address is either 0x60, 0x61, 0x62,0x63, 0x64 or 0x65
dac.setVoltage(0, false); //sets the output current to 0 initially

// ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V 1 bit= 3mV 0.1875mV (default)
ads.setGain(GAIN_ONE); // 1x gain +/- 4.096V 1 bit = 2mV 0.125mV
// ads.setGain(GAIN_TWO); // 2x gain +/- 2.048V 1 bit = 1mV 0.0625mV
// ads.setGain(GAIN_FOUR); // 4x gain +/- 1.024V 1 bit = 0.5mV 0.03125mV
// ads.setGain(GAIN_EIGHT); // 8x gain +/- 0.512V 1 bit = 0.25mV 0.015625mV
// ads.setGain(GAIN_SIXTEEN); // 16x gain +/- 0.256V 1 bit = 0.125mV 0.0078125mV
ads.begin(); // note- you can put the address of the ADS111 here if needed
  ads.setDataRate(RATE_AMPS_860SPS); // sets the ADS111 for higher speed
}

```

Then comes the loop region. This is where the current steps are created and looped through, using a for loop:

```

void loop() {
for (i = 1; i <= npts; i++) {
  I_A = i * I_step_A;
  dac.setVoltage(0, false); //sets the output current
  func_meas_off();
  func_meas_on();
  dac.setVoltage(0, false); //sets the output current

  I_load_A = I_sense_on_A - I_sense_off_A; //load current
  V_VRM_thevenin_v = V_VRM_off_v;
  V_VRM_loaded_v = V_VRM_on_v;
  R_thevenin = (V_VRM_thevenin_v - V_VRM_loaded_v) / I_load_A;

  if (V_VRM_loaded_v < 0.75 * V_VRM_thevenin_v) i = npts; //stops the ramping

  Serial.print(i);
  Serial.print(", ");
  Serial.print(I_load_A * 1e3, 3);
  Serial.print(", ");
  Serial.print(V_VRM_thevenin_v, 4);
  Serial.print(", ");
  Serial.print(V_VRM_loaded_v, 4);
  Serial.print(", ");
  Serial.println(R_thevenin, 4);
}

Serial.println("done");
delay(30000);
}

```

After the current value is established, I call two functions. One function measures the voltage and current with the current off. This is func_meas_off. I always label my functions with func_. This way I can tell at a glance if a name is a variable or a function.

I placed this function in a separate tab to make the code neater:

```

void func_meas_off() {
    dac.setVoltage(0, false); //sets the output current
    iCounter_off = 0; //starting the current counter
    V_VRM_off_v = 0.0; //initialize the VRM voltage averager
    I_sense_off_A = 0.0; // initialize the current average
    itime_stop_usec = micros() + itime_off_msec * 1000; // stop time

    while (micros() <= itime_stop_usec) {
        V_VRM_off_v = ads.readADC_Differential_0_1() * ADC_V_per_ADU / v_divider +
        V_VRM_off_v;
        I_sense_off_A = ads.readADC_Differential_2_3() * ADC_V_per_ADU / R_sense +
        I_sense_off_A;
        iCounter_off++;
    }

    V_VRM_off_v = V_VRM_off_v / iCounter_off;
    I_sense_off_A = I_sense_off_A / iCounter_off;
    // Serial.print(iCounter_off);Serial.print(", ");
    // Serial.print(I_sense_off_A * 1e3, 4); Serial.print(", ");
    // Serial.println(V_VRM_off_v, 4);
}

```

The output of this function is the measured `V_VRM_off_v` and the `I_sense_off_A`. This is the unloaded value of VRM voltage and current.

The second function, also in a tab, is `func_meas_on`. It outputs these same measured values, but with the current on. Here it is:

```

void func_meas_on(){
//now turn on the current

    I_DAC_ADU = I_A * R_sense * DAC_ADU_per_v;
    dac.setVoltage(I_DAC_ADU, false); //sets the output current

    iCounter_on = 0;
    V_VRM_on_v = 0.0; //initialize the VRM voltage averager
    I_sense_on_A = 0.0; // initialize the current averager
    itime_stop_usec = micros() + itime_on_msec * 1000; // stop time
    while (micros() <= itime_stop_usec) {
        V_VRM_on_v = ads.readADC_Differential_0_1() * ADC_V_per_ADU / v_divider +
        V_VRM_on_v;
        I_sense_on_A = ads.readADC_Differential_2_3() * ADC_V_per_ADU / R_sense +
        I_sense_on_A;
        iCounter_on++;
    }
    dac.setVoltage(0, false); //sets the output current to zero

    V_VRM_on_v = V_VRM_on_v / iCounter_on;
    I_sense_on_A = I_sense_on_A / iCounter_on;
    // Serial.print(iCounter_on);Serial.print(", ");
    // Serial.print(I_sense_on_A * 1e3, 4);Serial.print(", ");
    // Serial.println(V_VRM_on_v, 4);
}

```

}

37.4.7 The output and examples

The output of this function is the V_VRM_on_v or loaded output voltage and the I_sense_on_A current when loaded.

These two sets of values are used to compute the VRM on and off voltages, currents, and Thevenin resistance, back in the main loop. These values are printed to the serial monitor.

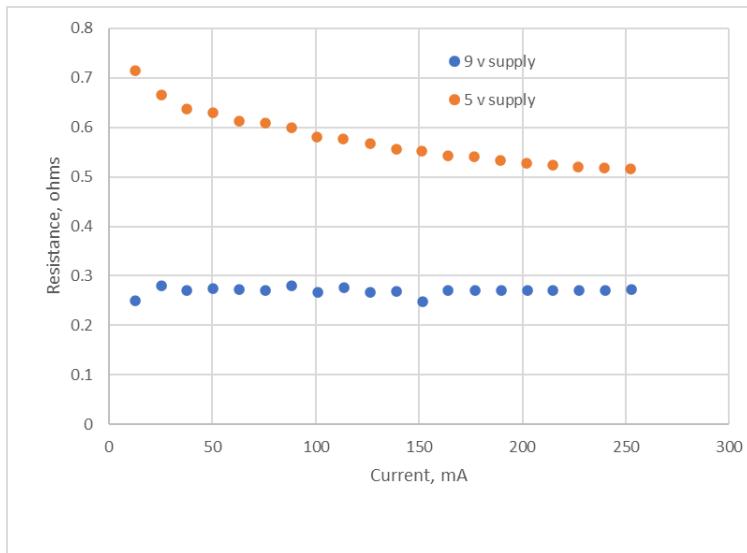
Remember, the first example you should run to test your circuit and code is a VRM with a known series resistor added. You should measure the added series resistor as the Thevenin resistance.

Here is an example of the print out for the case of a 5 V AC to DC supply wall wart. The columns are:

Index, current in mA, V_thevenin, V_loaded, R_thevenin

```
1, 12.702, 5.4683, 5.4593, 0.7141
2, 25.122, 5.4682, 5.4515, 0.6653
3, 37.697, 5.4682, 5.4442, 0.6364
4, 50.236, 5.4682, 5.4366, 0.6291
5, 63.016, 5.4682, 5.4295, 0.6135
6, 75.655, 5.4681, 5.4220, 0.6095
7, 88.066, 5.4681, 5.4153, 0.5997
8, 100.698, 5.4680, 5.4095, 0.5811
9, 113.343, 5.4680, 5.4027, 0.5765
10, 126.169, 5.4679, 5.3964, 0.5668
11, 138.831, 5.4680, 5.3909, 0.5555
12, 151.348, 5.4679, 5.3844, 0.5518
13, 163.864, 5.4679, 5.3789, 0.5435
14, 176.623, 5.4680, 5.3725, 0.5406
15, 189.514, 5.4678, 5.3668, 0.5330
16, 202.157, 5.4678, 5.3611, 0.5282
17, 214.568, 5.4677, 5.3554, 0.5233
18, 227.126, 5.4678, 5.3498, 0.5194
19, 239.611, 5.4678, 5.3435, 0.5188
20, 252.202, 5.4677, 5.3373, 0.5170
```

This is the data that can be plotted directly in excel. Here are two example plots of a 9 V and 5 V supply.



You can take the printed values from the serial monitor, copy them and paste them directly into excel. Or, you can use the [data streamer function in excel](#) and print them directly into excel. Note: this only works for a microcontroller using a CH340g or FTDI USB to UART interface chip. It does not work for a SAMD21 microcontroller. Alternatively, you can use a terminal emulation tool like [PuTTY](#) or [Yet Another Terminal emulator, YAT](#) to collect the data in a csv file and then import it into excel and plot it.

Before you finish your lab, you should be checked off by your TA.

37.6 Grading rubric:

Most semesters, there is no report. If there is a report, the scoring for this report is:

- 1 point for check off by your TA
- 1 point if you show examples of what you set up, measured and your analysis
- 1 point if you review the principles behind the circuit and the measurements coherently.

Remember, this report will look great in your portfolio.

Chapter 38 CDR of Brd 4

Before coming to class, you should have completed the layout.

In class, a few volumeters will be selected to present their layout to the class. While they present their layout, the rest of the class will critically listen and offer recommendations to the designer.

Be on the lookout for:

1. *Any possible hard errors which will prevent the correct operation of the board*
2. *Any soft errors which will increase the noise or the risk of the board not meeting spec*
3. *Any good design features which might be good features to include in everyone else's designs.*

Use the check list in the lab manual and in the textbook to go through the list of possible, or commonly occurring errors to verify none of these are present in the design.

After a few layouts have been reviewed, you will have time to complete your layout in class.

Before you leave, you should review your layout with your TA to have them check off your design.

It is not necessary your design be completed by the end of class time, but it should be far enough along so that any major issues can be identified.

Before you post your three design files on canvas, you must:

1. *Review your schematic and layout against the checklists provided in the lab manual and textbook*
2. *Have another student review your schematic and layout*
3. *Have your TA check off your layout*
4. *Post your design files to the JLC website and complete the DFM and get an email acknowledgement back from them that your design is accepted and ready for ordering. This may take as much as 2 hours to get an email back.*
5. *If you received any error messages from the DFM check, you have corrected them, and resubmitted your design for DFM check.*

Design files are due by 9 am on Thursday morning. This is when they will be ordered. If you miss this deadline, your board will not be ordered. You will have to wait for the next week to have your board ordered.

Turning in your design files counts as 3 points. The grading rubric is:

- 1 point for sign off by your TA in class.*

- *1 point if your design files were turned in on time*
- *1 point if your design was accepted by the vendor for production*

If your design file is rejected by JLC, you may only end up with the 1 point for TA check off for the design assignment. This will only happen if you fail to go through the online DFM check on the JLC web site.

If your board fails the DFM check when the TA places the order for your board, it will slow up the delivery of ALL the boards and may affect the schedule for the rest of the students. DO NOT LET THIS HAPPEN FOR YOUR BOARD.

If your board is not ordered with the other student boards, you are still responsible for getting your design completed. Submit it by the next week.

Chapter 39 Lab 22: Applications of brd 4 using the SBB version

Your brd 4 is a specialized instrument droid which will characterize any voltage source. In this lab, you will have a chance to finalize your SBB version, fine tune the code and conduct some measurements in anticipation of receiving your final, functional brd 4.

Once you understand how your instrument droid works, and what to expect in all your measurements, you can move on to look at some unknown voltage sources.

As you have seen in many labs, it is very easy to get a measurement. Virtually all instruments will give you numbers. But you can never be sure the measurements are not just artifacts. You need to verify your instrument droid is working as you expect with few measurement artifacts. This is part of the testing of any measurement system.

39.1 Exp 1: finalize the hardware and understand your code

The very first step is to finalize the hardware of your solderless breadboard version of the brd 4. To test your instrument droid, you might consider using as a reference voltage source to measure, the output of the function generator set on DC output.

You can measure this output voltage with a scope, so you know the Thevenin voltage. From your previous measurements in other labs, you should know the Thevenin output resistance is always 50 ohms. Be sure to use a voltage of at least 5 V output as the source.

You should test each of the parts of your SBB circuit and look at the voltage on the gate of the MOSFET, the voltage across the sense resistor and the voltage on the VRM. As the code steps up the voltage on the DAC, do you see a DC signal on each of the test points?

Sometimes, there is oscillation at about 1-20 MHz due to the large inductance in the path from the VRM under test and the drain of the MOSFET. This starts due to the capacitive coupling between the drain and the gate and the fast dV/dt on the gate. If you see oscillation due to this feedback, you can suppress it by adding a small capacitor between the gate and ground, on the order of 1 μF or less.

While you do not have to create any original or additional code to run your instrument droid, you should understand what the code is doing.

Are the voltages you see consistent with what you expect to see (rule #9) based on the functioning of the code you are using?

39.2 Exp 2: Measure something for which you know the answer

The very first thing you should measure is something for which you know the answer. The function generator is a good first device to measure. You can measure the Thevenin voltage and you know the Thevenin resistance.

Given the sense resistor value you are using, you can also calculate the maximum current you can get before the output voltage of the function generator drops to 75% of its initial value. This is hard coded as when to stop the sweeps.

For example, if the sense resistor is 10 ohms, and the VRM voltage is set for 5 V, then 75% of 5 V is 3.75 V. This should be the lowest voltage your instrument droid measures. The current would be $1.25 \text{ V}/50 \text{ ohms} = 25 \text{ mA}$.

Are the measurements reported by your instrument droid consistent with what you expect for this VRM?

Are the voltages you measure at various nodes consistent with what you expect?

Be sure to copy the columns of measurements into excel and plot the Thevenin resistance vs current for your VRM. What do you expect the resistance to do as a function of current? What do you actually measure?

The more familiar you are with the functioning of your instrument droid using the solderless breadboard version, the better you will be able to debug your brd 4 when it comes back.

39.3 Exp 3: Measure another VRM for which you know the answer

As a final verification, use the bench top power supply which has a very low output resistance. Add a series 10 ohm resistor to it so you know what to expect in your measurements. Select its out voltage to be whatever you think appropriate.

Look at the output voltage into your instrument droid with a scope while you are taking measurements with your instrument droid. Are the voltages at various test nodes what you expect to see?

What should you expect to measure as the Thevenin resistance of this modified VRM?

Be sure to plot in excel the Thevenin resistance as a function of the output current. What do you expect to see? What do you actually measure?

When you have verified your instrument droid is working as you expect, you can apply this instrument droid to some unknown voltage sources.

Only move to the next step when you are confident your instrument droid is working as expected, otherwise you will have no idea if what you are measuring is real or is just an artifact.

39.4 Exp 4: Measure some unknown VRMs

The whole purpose of your instrument droid is to give you insight into the behavior of some unknown voltage sources. Once you have confidence your instrument droid is working as you expect, it is time to measure some real sources.

Think about all the voltage sources you have lying around and consider measuring them. Remember, they have to have a constant voltage during the time of your measurement, so using a source that is a square wave probably is not going to give you a measurement that makes any sense.

Keep track of the VRM sources you measure with your solderless breadboard version so that can remeasure them with your brd 4 version and compare the results.

Consider the following sources for example, as mentioned in the description of brd 4:

- *A 5 V wall wart*
- *A 9 V wall wart*
- *A 12 V wall wart*
- *The 6V power supply*
- *An AA battery*
- *A coin cell battery*
- *A LiPo battery*
- *The digital output pin of an Arduino*
- *The output of a 601 opamp*
- *The output of a TLV4110 opamp*
- *The output of a 555 timer*
- *The output of a hex inverter*
- *The output of the 5 V rail of an Arduino*
- *The output of the 3.3 V rail of an Arduino*
- *The output of a 3.3 V LDO*
- *The output of a DAC*

Do you get the same Thevenin resistance vs output current for the same VRM that others measured?

39.5 Grading rubric:

Of this lab, you can write up the details of how your instrument droid works, what you expect as the internal voltages and what you measure and then some examples of the Thevenin resistance and output current of various sources.

As usual, the 3 points are from:

1 point for TA check off

1 point for coherent description of what you measure

1 point for good analysis of what you measured and the so what?

Chapter 40 Lab 23 SBB circuits: smart LEDs

The purpose of this lab is to get comfortable programming the smart LEDs before incorporating them in your board 4 design.

You will add three or more smart LEDs on your board for whatever purpose you would like. They can make cool lighting effects. No more than 10 total on a board.

For this lab, you will use similar LEDs on a small strip you can insert into a solderless breadboard to experiment with the code. You will connect 3 or more smart LEDs in a solderless breadboard to program the LEDs using one digital pin from your Arduino.

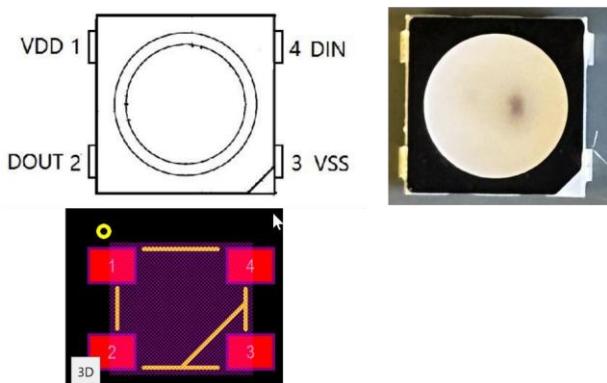
You can use a few discrete smart LEDs on a breakout board, or a strip of smart LEDs.

40.1 The Smart LED Component in your board 4 project

The actual smart LED component you will be adding to your brd 4 project is this one:

https://datasheet.lcsc.com/lcsc/2106062036_Worldsemi-WS2812B-B-W_C2761795.pdf

It is the part in the JLC library we use for the class. Note that the pin assignment of this part is a little funny. Here is the pin assignment of the part, the part footprint and a top view of the actual part.



Note that pin 1, identified in the footprint with the small yellow circle, is not the pin that is marked on the part. The small triangle in the lower right corner of the part is actually pin 3, which is the Vss or gnd connection. Just be aware of this orientation when mounting your smart LED to your circuit board.

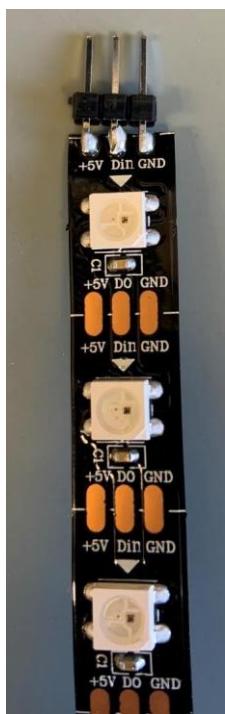
While the mark on the board or in the footprint is usually identifying pin 1 of the part, not always is the identification on the actual part the pin 1 of the part. Read the datasheet carefully for the part.

Note also that the bits transmitted are in the order of GRB. You will need to know this when you modify your code to drive this part.

40.2 The smart LED strip used in this lab

The purpose of this lab is to get familiar with the code to drive the smart LED. We will use a similar but not identical part which we can get in short strips.

You can connect a series of smart LEDs that come already in a strip of smart LEDS. You will need to solder three header pins to the input end to connect power, ground and a digital control pin, as shown below:



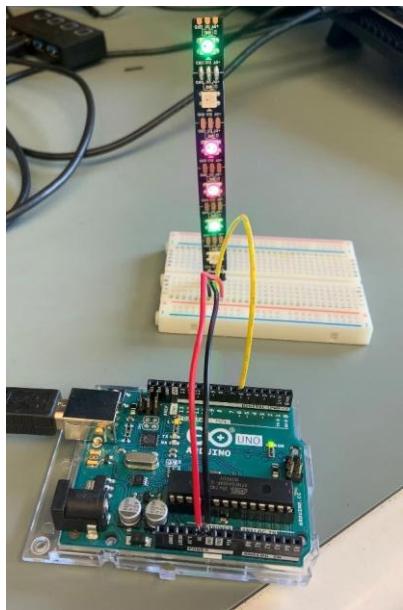
Be sure to solder the header pins to the input side of the strip. Now, this strip can be plugged into a solderless breadboard and connected to power, ground and the digital pin of an Arduino.

Note- many of the example sketches use digital I/O pin 6 to drive the neopixel array.

A simple example sketch to try is the RGBWStrandtest, for example.

You only need to change the digital pin and the number of pixels in the array.

Here is an example of the simple set up required.



40.3 The smart digital LEDs

The specific smart LEDs in the small strips [are from this larger strip](#). Note that these are GRB and NOT RGBW LEDs. The datasheet for these digital LEDs can be [found here](#). Make note that this is a 5050 type RGB smart LED. These are the same smart LEDs as in the WS2812 series. You will need to know this when selecting the Arduino libraries to drive it.

This smart LED is the same as the one you will use on your brd 4. The libraries are the same. This particular part you will use in this solderless breadboard version of the lab is an GRB. This will apply to the color mapping you will use in the library.

The pin out for each pixel is shown in the figure below.

Pin Configuration

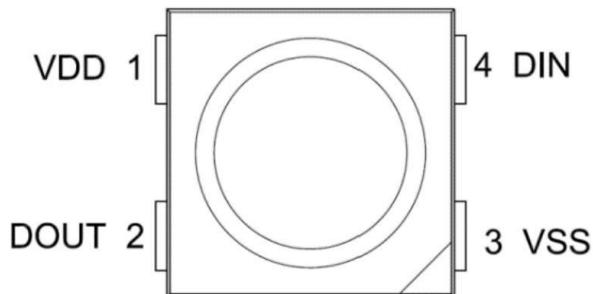
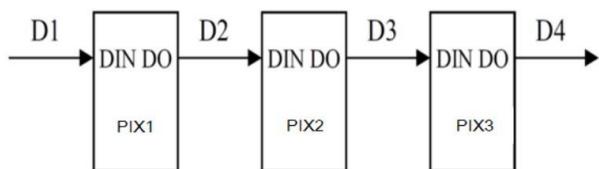


Figure 2. IN-PI55QATPRPGPBW-XX Pin Configuration

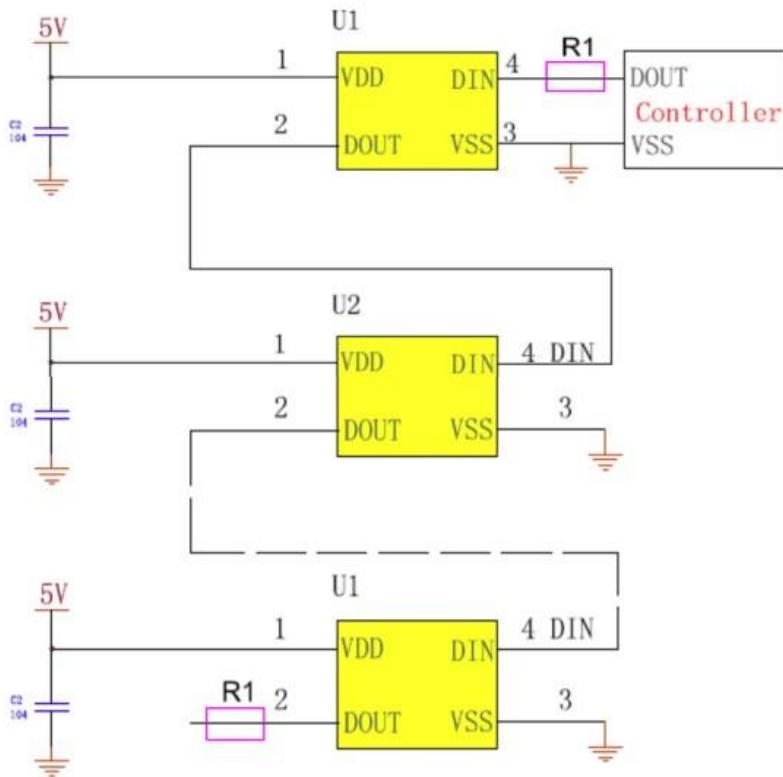
They can be powered by 5 V DC as Vdd.

These are digitally controlled LEDs which can be cascaded in series, as shown below.



40.4 Reading datasheets and design tradeoffs

In some datasheets for these sorts of smart LEDs, you will see a series resistor added to the digital pin, such as [this one below](#):



Product signal input and output must be connected in series with protection resistor R1. R1 depends on the size of the cascade amount, the greater the number of cascade, the smaller R1. The general recommended value is between 200-2KΩ, usually the recommended value is typical 500Ω.

This describes a “protection” resistor. This datasheet recommends 200 to 2k series resistance. Other datasheets suggest a 1 k resistor in series. What is the problem this resistor solves? And why is there one added to the last DOUT pin that just floats? This is an example of legacy code which got into the datasheet somewhere along its history and no one ever questioned it and so it is in all future datasheets and engineers who do not take responsibility for their designs keep including it.

The original intent of the input resistor was to limit the inrush current into the DIN pin. Some tech editor misread a datasheet somewhere and decided to add an additional resistor on the far end and no one questioned it, always assuming that the last person to edit the datasheet was smarter than they are and obviously had a reason to add it, so left it in.

Of course, the resistor at the end that is open serves no purpose at all. It was a mistake that has been copied for generations of datasheets.

While there will be no difference in the performance of the LED strip or the noise if you include this 1 k resistor or keep it in, it is a bad habit to just propagate legacy code that solves no problem. In each design that includes this resistor, the next engineer who looks at it assumes it has value and includes it in their design and it becomes established best practice. Stop the chain.

This is an example of the difference between in principle and in practice. In principle, adding the 1 k resistor at the DIN will limit any current from the digital pin. Afterall, if the output voltage is 5 V, the maximum current you will get from the pin is $5\text{ V}/1\text{k} = 5\text{ mA}$. When it turns on this will generate a dI/dt and create switching noise. The large inrush current will not damage any part. It might create switching noise, in principle.

Without this current limiting resistor, the current would be limited to the internal resistance of the digital pin, about 50 ohms. This would result in a potential current of $5\text{ V}/50\text{ ohms} = 100\text{ mA}$ and would turn on in the rise time of the output driver, which for the 328 uC is about 5 nsec.

Adding the 1 k resistor would increase the rise time of the signal and decrease the dI/dt . The RC charging time for the roughly 5 pF of input capacitance would be about $5\text{ pF} \times 1\text{k} = 5\text{ nsec}$. The max current would be about $5\text{ V}/1\text{k} = 5\text{ mA}$.

When the digital pin turns on, in principle, it is better to limit the current from the pin to 5 mA rather than 100 mA. This will create less dI/dt switching noise. In principle.

However, in practice, the input impedance to the DIN pin is very high. It is the gate to a receiver. This means that as long as the input capacitance of the pin and the interconnect is small, the pin driver will see an open and the drive current will be small. There will be little dI/dt and little switching noise whether the resistor is there or not.

While a SPICE simulation including the pin model, the input capacitance of the DIN pin, and the interconnect capacitance would enable an estimate of the actual current from the pin, and the dI/dt and the switching noise, this is a significant effort. Sometimes an OK answer now! is more important than a good answer late. Using a few simple principles, we can estimate the current from the uC driving pin.

For example, in the worst case, if the input capacitance were 5 pF, and the output impedance of the driver were 50 ohms, very conservative values, the RC charging time constant of the DIN pin would be $50\text{ ohms} \times 5\text{ pF} = 0.25\text{ nsec}$. This is way smaller than the rise time of the digital pins in most controllers. On the 328 uC, it is about 5 nsec. This means the rise time of the signal will be limited by the digital pin on the uC and not the RC of the DIN pin.

The driver will output a total charge of $5\text{ pF} \times 5\text{ V}$ to charge up the DIN pin to 5 V and will charge up in a time of about 5 nsec. This is an average current of about $5\text{ pF} \times 5\text{ V}/5\text{ nsec} = 5\text{ mA}$.

In practice, the rise time of the digital signal will be longer than 5 nsec and its output current will be less than 5 mA. But, if there is a longer interconnect path to the DIN pin then this current could grow.

Is the series resistor really needed in practice?

This is where engineering judgement plays a role. In principle, there is a real problem it could reduce. In practice the value of adding the series resistor is small, so it is not worth paying much for it. If it is free and does not impact other "unintended consequences" of the product design, such as layout, availability, reliability, then it could be included. If it has a cost in schedule, \$\$\$, or risk, then it should be carefully evaluated.

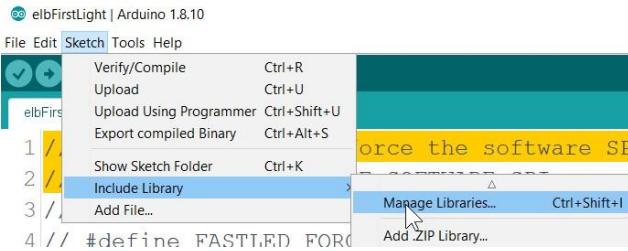
As an engineer, be aware of the reason for your design decisions and make your decisions based on your engineering judgement. Sometimes you don't always have all the information you need but you have to make decisions. This is what engineering judgement is all about.

You decide if you want to add a series resistor to the DIN pin in your design. If you have good reasons to support it and consider the design tradeoffs, then it is the right decision for your design.

40.5 Installing the Library

Adafruit has a great tutorial on using the Neopixels. [Check this link out](#). You can use this [library from Adafruit](#) to drive the digital LEDs. Note the use of 5050 pixels.

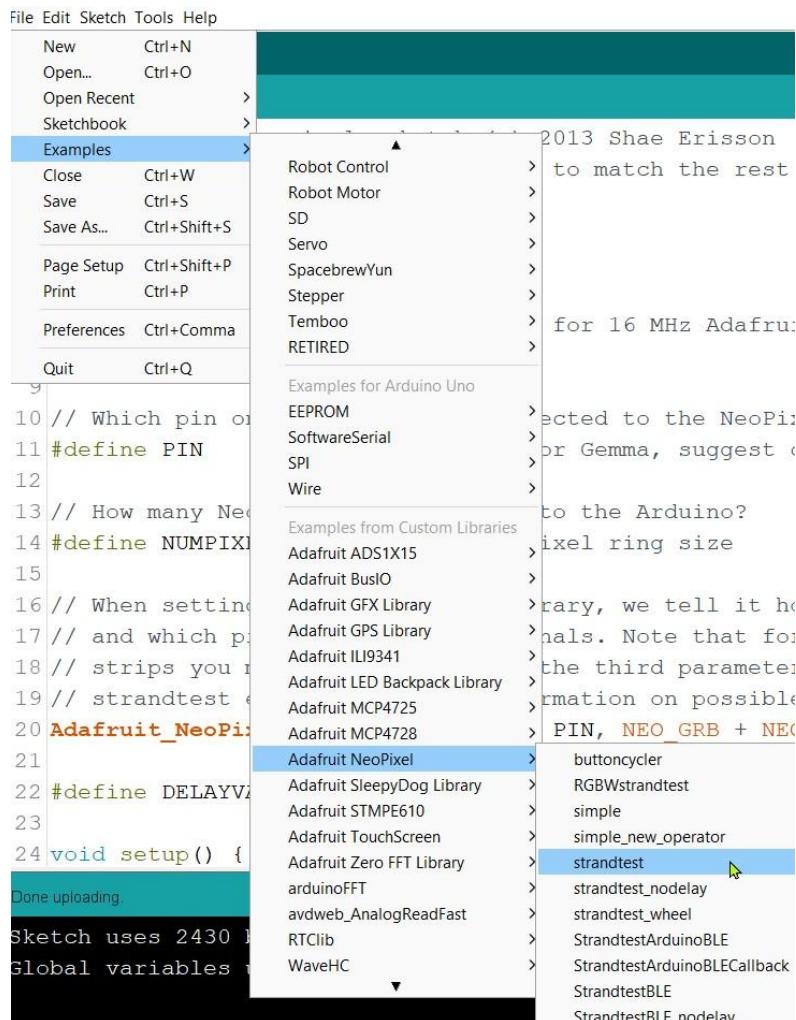
The simplest method is to just install the Adafruit Neopixel library from the Sketch/manage library menu, as seen below. There are a number of Adafruit libraries for neo pixels. If you have never used these before, start with the Easy Neopixels library. If you want to do more advanced operations, use the FastLED NeoMatrix library.



Then, in the list of library options, search under Neopixel and select the Adafruit Neopixel library, as shown below.



After you have installed this library, you can go back to the File/examples menu and scroll down to find the examples for the Adafruit Neopixel. Pick one of these sketches, like the Simple, sketch and modify it for your configuration. This list of sketches are shown below:



The screenshot shows the Arduino IDE interface. The menu bar at the top includes File, Edit, Sketch, Tools, and Help. The 'Sketch' menu is currently active, with its submenu visible. The submenu contains options: New (Ctrl+N), Open... (Ctrl+O), Open Recent, Sketchbook, Examples, Close (Ctrl+W), Save (Ctrl+S), Save As... (Ctrl+Shift+S), Page Setup (Ctrl+Shift+P), Print (Ctrl+P), Preferences (Ctrl+Comma), and Quit (Ctrl+Q). The 'Examples' option is highlighted with a blue selection bar. Below the menu, the code editor displays a sketch named 'strandtest'. The code includes comments for pin setup, number of pixels, and strandtest configuration. It defines a PIN constant and sets up the NeoPixel library. The 'setup()' function is shown with a call to 'strandtest.begin()'. The code editor also shows the status 'Done uploading.' and 'Sketch uses 2430 bytes (18%) of program storage space'. At the bottom, there is a section for 'Global variables'.

```

File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Open Recent >
Sketchbook >
Examples >
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
9
10 // Which pin controls your NeoPixel strip?
11 #define PIN 10
12
13 // How many NeoPixels are connected?
14 #define NUMPIXELS 10
15
16 // When setting up the NeoPixel library, we need to know how many
17 // pixels you have. This is used to calculate the pixel ring size
18 // strips you have connected.
19 // strandtest example sketch for Adafruit NeoPixel library
20 Adafruit_NeoPixel strandtest(10, PIN, NEO_GRB + NEO_KHZ800);
21
22 #define DELAYVAL 100
23
24 void setup() {
  strandtest.begin();
}
Done uploading.
Sketch uses 2430 bytes (18%) of program storage space.
Global variables: 0

```

2013 Shae Erisson
 Robot Control > to match the rest
 Robot Motor >
 SD >
 Servo >
 SpacebrewYun >
 Stepper >
 Temboo > for 16 MHz Adafruit
 RETIRED >
 Examples for Arduino Uno
 EEPROM > ected to the NeoPi:
 SoftwareSerial > or Gemma, suggest <
 SPI >
 Wire >
 Examples from Custom Libraries
 Adafruit ADS1X15 > pixel ring size
 Adafruit BusIO >
 Adafruit GFX Library > rary, we tell it ho
 Adafruit GPS Library > nals. Note that fo
 Adafruit ILI9341 > the third paramete
 Adafruit LED Backpack Library > rmation on possible
 Adafruit MCP4725 > PIN, NEO_GRB + NEO_KHZ800
 Adafruit MCP4728 >
 Adafruit NeoPixel > buttoncycler
 Adafruit SleepyDog Library > RGBWstrandtest
 Adafruit STMPE610 > simple
 Adafruit TouchScreen > simple_new_operator
 Adafruit Zero FFT Library > strandtest
 arduinoFFT > strandtest_nodelay
 avdweb_AnalogReadFast > strandtest_wheel
 RTClib > StrandtestArduinoBLE
 WaveHC > StrandtestArduinoBLECallback
 StrandtestBLE
 StrandtestBLE_nodelay

You should only need to set the digital pin number and the number of pixels you have in series.

Download some code that makes patterns in your LED array.

When you design these neopixels into your board, you have to decide on a specific digital pin to drive the smart LEDs. I recommend using pin 6, for example. DO NOT use pins 0 or 1 as these are reserved for the RX or TX of the UART.

Just be aware of the specific type of smart LEDs you have in your SBB and in your brd 4. Some are RGB and some are RGBW. The libraries are the same, but you will have to edit the commands to select either RGBW or GRB in some locations.

Both the LEDs on your brd 4 and in this SBB version are GRB smart LEDs.

40.6 The experiment

Once you have installed the library and connected to the smart LED strip to your circuit board and then Arduino, you should run one of the example sketches to drive patterns of lights on your LED strip. Experiment to change the LED pattern.

On your board 4, you can decide how you want to use the LEDs as indicators to show progress in the measurements or to translate measurement values into colors, or just as a cool display between measurements.

40.7 In your report

Before you finish your lab, you should be checked off by your TA. Show your TA some of the light patterns you were able to create in your smart LED array.

Write up a brief description of how you set up the smart LEDs and what patterns you used to program them.

40.8 Grading rubric:

As is usual, the scoring for this report is

1 point for check off by your TA

1 point if you show examples of what you built, measured and your analysis

1 point if you review the principles behind the circuit and the measurements coherently.

Remember, this report will look great in your portfolio.

Chapter 41 Lab 24 SBB circuit with the Buzzer

The purpose of this lab is to get comfortable using the buzzer which will be incorporated in your brd 4.

This buzzer will click every time a 5 V signal is connected across the two pins. One pin connects to a digital I/O and the other to ground.

41.1 The buzzer

This is a very simple component. When you change the voltage across the two pins, from 0 to 5 V, the buzzer clicks. If you drive the buzzer with a modulated digital signal, it will make a sound with a tone at the modulation frequency.

You can drive it with a digital pin and manually modulate the pin voltage in your code. You can insert the buzzer directly between the ground pin and pin 12 in an Arduino. It is not a perfect fit, but can be shoved in. This is shown below:



You decide how you want to drive it and which digital pin you want to use. You can use it as an alarm for example.

You can use the tone() function in the IDE .

Or, you can use a PWM signal and the buzzer will buzz at the frequency of the PWM signal.

Or, you can play tones on it. [Here is a project](#) in which someone created dozens of songs that can be played on your buzzer.

Here is a very simple sketch to get started, using pin 12:

```
void setup() {  
pinMode(12, OUTPUT);  
} void loop() { for (int i = 1;  
i <= 100; i++) { tone(12, 100  
* i); delay(100);  
}  
noTone(12);  
delay(5000);  
}
```

41.2 Reverse engineering and experimenting

The specs on the buzzer are hard to come by, other than the pin configuration. How does it work and what are its specs? If you can't find this information and you need it to design your circuit, you should purchase a sample and do the measurements yourself. That's why we build rapid prototypes. The buzzers are cheap enough that we can sacrifice some of them to break them open to see how they work. The figure below is one I opened up to show the coil and a small clapper membrane which is pulled down whenever the current is energized.



The sound is made when this membrane with the small weight hits the center core of the coil. The current is turned off and the clapper spring back up. When the coil is turned on, the clapper comes down again. One pulse is made per one pulse of the signal. A square wave at 1 kHz would generate a tone at 1 kHz.

What is the resistance of the coil that is inside the buzzer? How much current will it draw when it is connected to a digital I/O?

What is the resistance of the coil? Try measuring it with an ohmmeter. I measured about 40 ohms for my buzzer.

When connected to a 5 V source, what will be the current load? It will be a little higher than you think is safe for an Arduino pin, but keep in mind that there is also a roughly 50 ohm output resistance on the Arduino digital pin itself.

The buzzer resistance would be in series with the roughly 50 ohms the Arduino pin, so the total current from the pin might be $5 \text{ V}/(40 \text{ ohms} + 50 \text{ ohms}) = 50 \text{ mA}$.

Select a digital pin and use the `tone()` function to play some sounds on the buzzer. You can search for songs to play.

Does the polarity of how you connect the digital pin to the buzzer matter? It is just a coil of wire. Should it matter which pin is the digital pin and which pin is ground?

41.3 In your report

Before you finish your lab, you should be checked off by your TA. Play some tones for your TA.

Write up a brief description of how you set up the buzzer and what tones you played.

41.4 Grading rubric:

As is usual, the scoring for this report is

1 point for check off by your TA

1 point if you show examples of what you built, measured and your analysis

1 point if you review the principles behind the circuit and the measurements coherently.

Remember, this report will look great in your portfolio.

Chapter 42 Lab 25 SBB circuits: ferrites

A ferrite is just an inductor with a little bit of built in loss and resistance. It is simply an inductor with some inductance and some series resistance, of about 0.1 to 1 ohm. This means it is NOT suitable as a general purpose filter inductor that would be used on a SMPS, for example. Why would its high resistance make it not suitable for a SMPS filter?

The purpose of a ferrite filter is to filter noise from the power rail getting on to the power pin of a very sensitive analog device. It is **NOT** a general-purpose filter for all power rails. What would happen to the power rail to an IC if there were a 1 ohm series resistor in its path, let alone a 10 uH inductor?

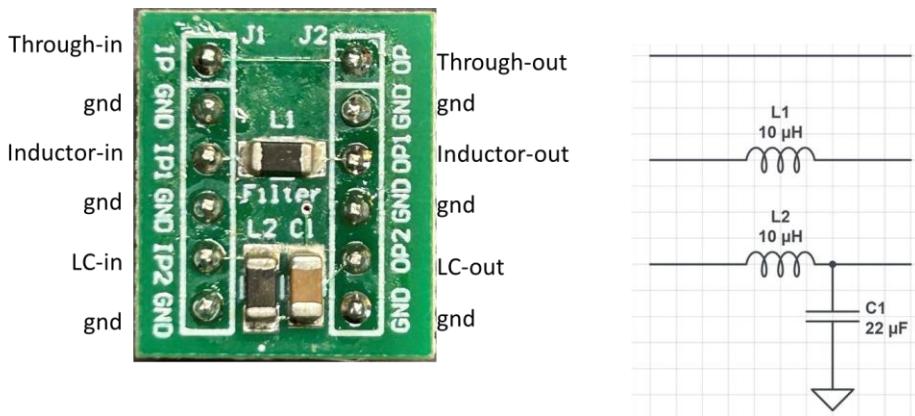
If your IC has a lot of current switching, the noise comes FROM the device and you use decoupling capacitors to reduce the noise generated on the device from the inductance of the board level power rail. The decoupling capacitor localizes the current during the di/dt period so there is no transient current passing through the inductance of the board level power rail inductance.

You will only use a ferrite filter on a device which will not draw much current, but you want the power pin of the device to have low noise. These are usually to power analog components and are often labeled as AVCC. Of course, if your power rail on your board is already low noise you may not need a ferrite filter.

Be sure to read the section in the textbook about using ferrite filters.

42.1 The Ferrite Filter test board

In this lab, you will get a small breakout board with 3 different circuits on it. An example is shown below with the equivalent circuit.



Your board should have pins in the board so you can plug the board in your solderless breadboard with the components on the top side.

This board has three different, independent circuits on it. The bottom layer, of course, is a ground plane.

From the top, the first pins on the left and right side are a through connection.

The next row of pins are gnd connections to the plane on the bottom layer.

The next row of pins are connections to the 10 uH ferrite in series. It connects from one end to the other end of the ferrite.

The next row are gnd pins.

The next row of pins connect to an LC filter you will implement to filter noise FROM your power rail on your board, TO your IC device that is sensitive to noise. Note the orientation. The noisy side is the left side. The filtered side is the right side. What would happen if you flipped the filter?

In this lab you will measure the filtering ability of the three different circuits by sending in noise from a power rail through each circuit.

42.2 Experiment 1: build up your slammer circuit and measure the rail noise

You will build the same slammer circuit you used in a previous lab to generate switching noise on the power rail. You will then measure the noise coming through your three filter circuits to see the impact of the ferrite filter on reducing the noise on the other side of the filter.

In this slammer circuit, you will use a transistor, such as the TIP41. The specific transistor you use will influence the turn on time at which it can drive current from the power rail. If the turn on time is long, the di/dt on the power rail will be small and there will not be much switching noise. In this experiment, we want a large switching noise to show off the ability of the ferrite LC filter to filter this noise.

Different transistors have different turn on rise time. Unfortunately, this is sometimes a poorly documented feature in a datasheet and sometimes if this is important, you just have to measure the turn on time in a circuit.

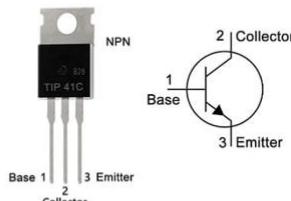
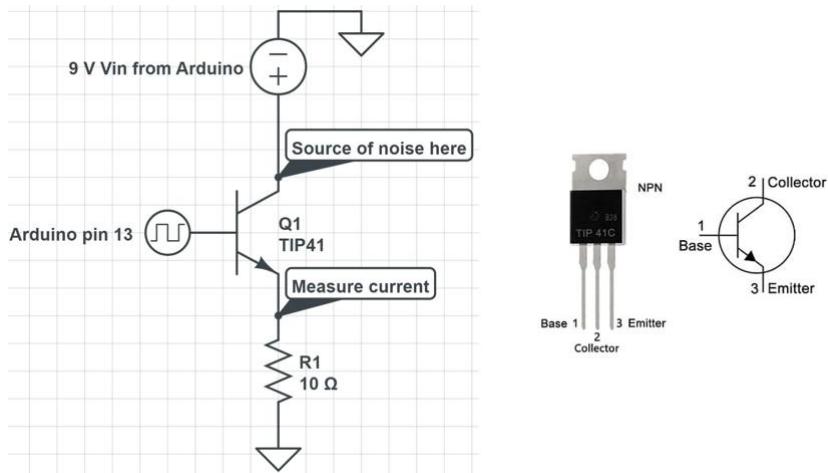
For example, the TIP31c and the TIP41c transistors both show a current-gain bandwidth product of 3 MHz in their datasheets. This is roughly the frequency at which the current gain is 1. However, the TIP41c has a shorter rise time than the TIP31c transistor. The 2N3904 transistor has a current-gain bandwidth of 300 MHz. However, its maximum current is limited to 200 mA, compared to 6 A for the TIP41C.

This circuit we use in the lab is a good test circuit to compare the turn on current rise time and switching noise behavior of different transistors. The signal that switches the transistor on is pin 13 from an Arduino. Its rise time is close to 5 nsec, faster than the transistor can switch. This means this circuit will be sensitive to the transistor turn on time.

Once the circuit is built, we can literally just plug in different transistors and compare their current rise time and their switching noise on otherwise identical conditions.

In many circuits, we want to reduce the switching noise on the power rail. This means, it is a good principle to always use the longest rise time we can get away with, if short rise time is not important. This will always reduce the switching noise.

Here is the circuit you will build in this lab. I used a TIP41c for these measurements.

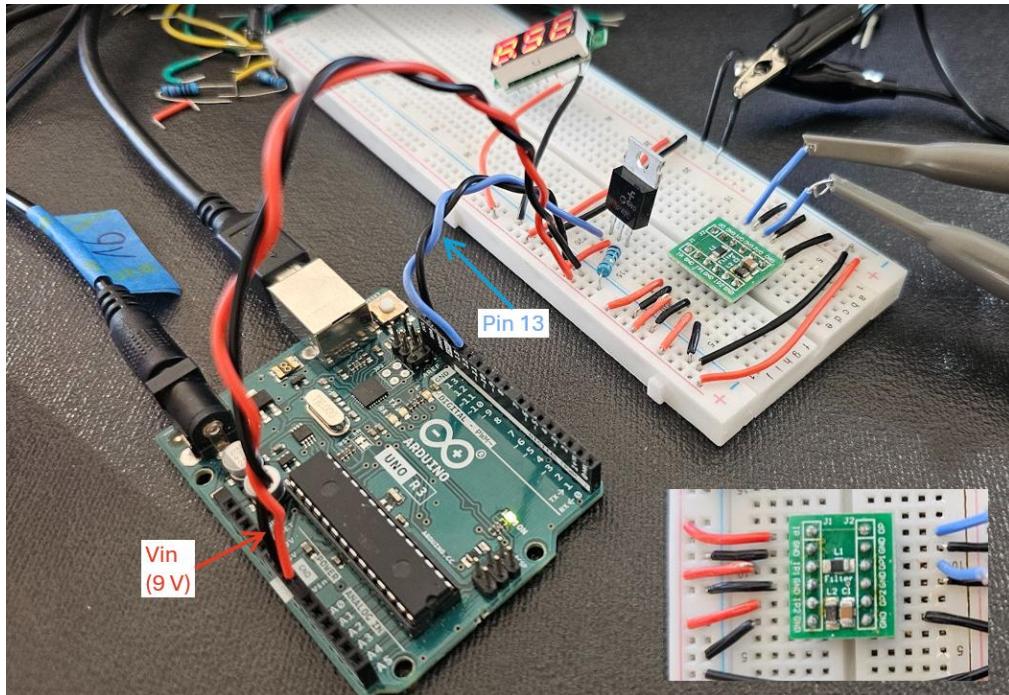


The slammer circuit will be driven by pin 13 from an Arduino. Remember, you should use a low duty cycle, like 10% so that you do not heat up the resistor or the transistor. A program with on-time of 1 msec and off-time of 20 msec should be fine.

The code is simply:

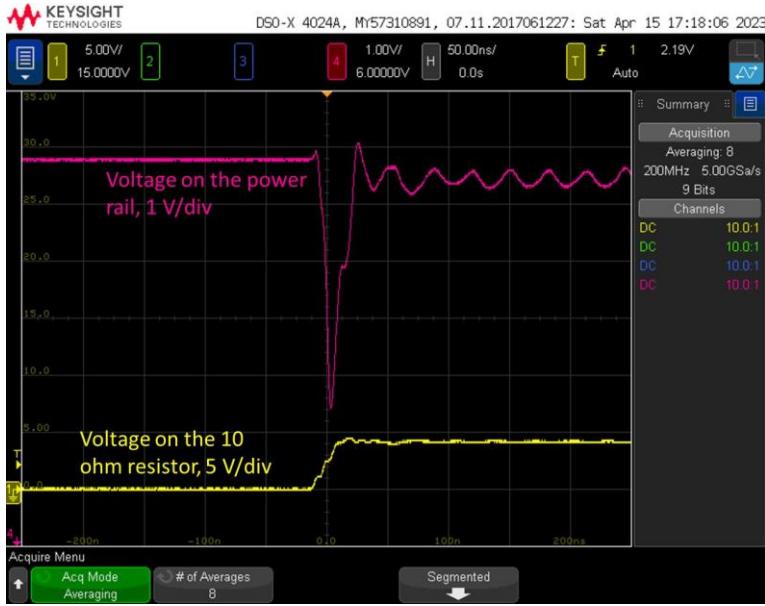
```
void setup() {
  pinMode(13, OUTPUT);
} void loop() {
  digitalWrite(13, HIGH);
  delay(1);
  digitalWrite(13, LOW);
  delay(20);
}
```

You can power the slammer circuit using the 9 V rail from the Arduino board, if you plug a 9 V wall wart into the Arduino board. This entire circuit on a solderless bread board is shown below.



When the signal on pin 13 turns on, it turns the transistor on and the voltage across the 10 ohm resistor is about $5\text{ V} - 0.6\text{ V} = 4.4\text{ V}$. This means the current is about 440 mA. This current turns on with the rise time of the Arduino, about 10 nsec. It is slowed down from the intrinsic rise time of the signal from the Arduino due to the transistor's output voltage dropping and almost turning the transistor off.

When this large dI/dt turns on, there will be a large voltage drop on the power rail due to the inductance in the power rail. There is no decoupling capacitor used in this circuit. The current through the 10 ohm resistor and the voltage on the power rail is shown in the figure below.



This is a large voltage droop on the power rail. In this case, almost 4.5 V. This is the noise that could be on your board due to noise getting on the power rail from devices which switch. Of course, with good use of decoupling capacitors placed close to your device that switches, there should not be much noise on the power rail.

However, sometimes noise gets on the power rail from someone else's circuit. It is this noise which we will use the various filters to reduce getting onto the sensitive device.

Note the ripple noise on the power rail after the Arduino pulse is turned on. Where is this coming from? What is its frequency? It is not ringing noise. Remember, when the Arduino digital pin is on, it is outputting a direct connection to the 5 v rail on the die to the Arduino pin. This is effectively a quiet HIGH pin. What will the noise on the on-die power rail be on the 328 uC? This clock edge noise will modulate the current through the transistor and the voltage noise on the power rail from the 9 V power supply due to its output Thevenin resistance.

42.3 Exp 2: expectations

The LC filter circuit is a 2-pole low pass filter. The pole frequency is

$$f_{pole} = \frac{1}{2\pi\sqrt{LC}}$$

In this example, the L = 10 uH and the C = 22 uF. You should calculate the pole frequency. This means it will filter frequency components above the pole frequency, but not below. It will only filter high frequency components of the noise. Be aware of the pole frequency and be aware of the frequency components of the noise.

When the noise is generated by switching noise, which has a rise time of about 10 nsec in this example, the frequency components of the switching noise are roughly at and above a frequency of

$$f_{noise} > \frac{0.35}{RT} \sim \frac{0.35}{10 \text{ nsec}} = 35 \text{ MHz}$$

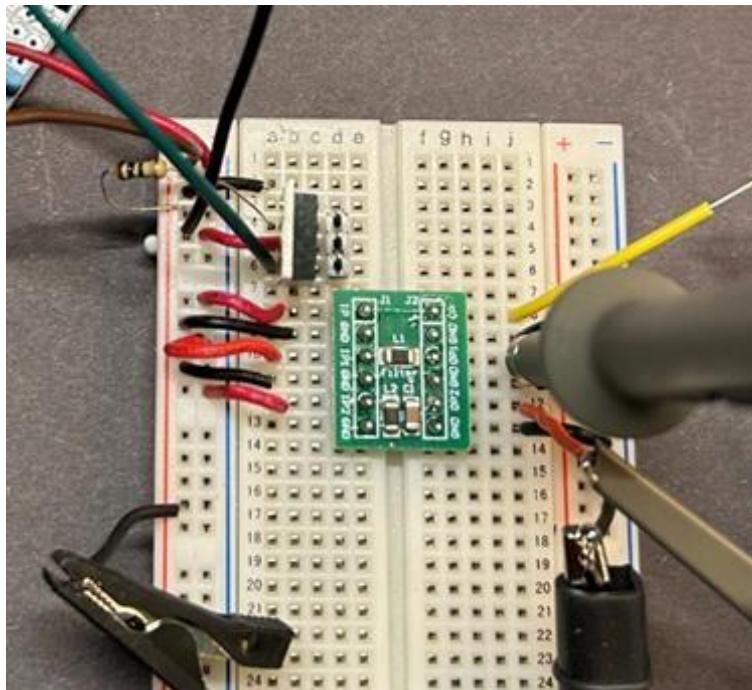
Where is the pole frequency of the filter compared to the frequency components on the noise on the slammer circuit?

Ideally, you want to engineer the pole frequency of the LC filter to be well below the frequency components of the noise. If this is the case, what do you expect to be the noise voltage that gets through the LC filter, onto the sensitive VCC rail of the device you are protecting?

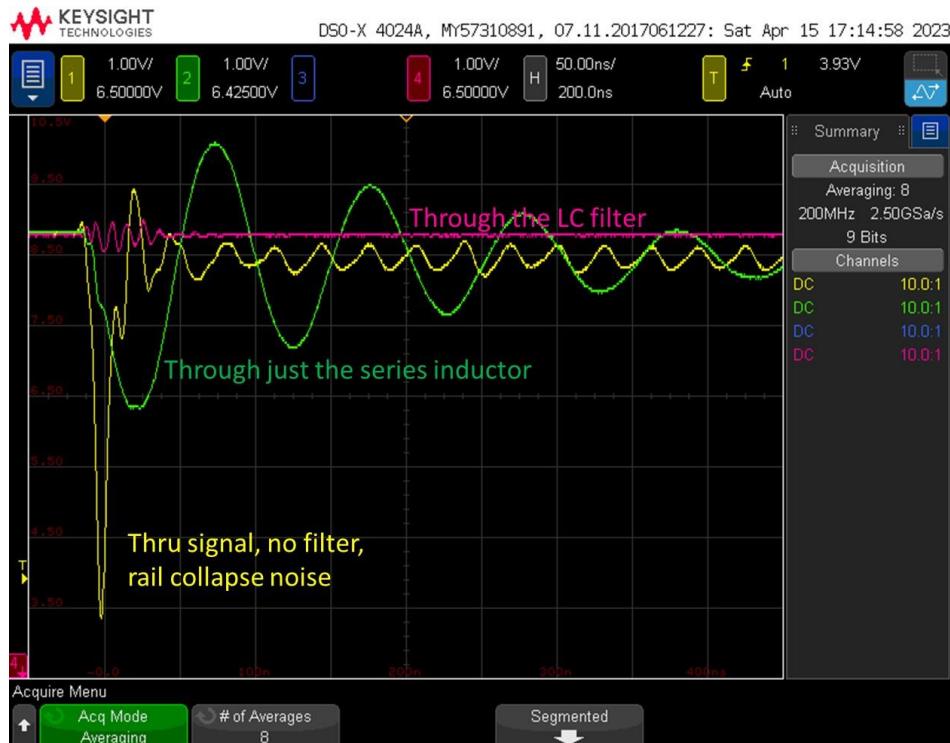
42.4 Exp 3: measure the noise through the filters

Connect the filter board into your solderless breadboard so that the inputs to the left edge pins can connect to the ground rail and the power rail with the noise on it. We will measure the voltage noise on the right side of the filter board.

Note that the orientation of the filter is very important. If the LC filter is reversed and is set up as a CL filter, it will not be very effective as a low-pass filter. This orientation is shown in the figure below.



Measure the signal on the right side of each circuit element: the thru, the series inductor and the LC filter. Here is what I measured on my board:



The rail collapse noise on the board is about 5 V peak to peak. The noise that gets through to the other side of the ferrite filter is about 0.3 V peak to peak. This is a reduction to about 6% of the noise on the other side of the filter. This is significant.

This is why we use ferrite filters to protect noise on the power rail from getting onto to sensitive voltage rails of devices.

You should be aware of the problem a ferrite filter would generate if there were significant current that has to go thru it to power a device. Its series R and L would generate more noise on the device generating the large dI/dt than if the ferrite were not used. This is not a general purpose component to use in all power rails, only in specific applications.

42.5 So what?

In your TA check off and in your lab report, consider answering the following questions:

1. *What is the pole frequency of the LC filter?*
2. *If the series resistance of the ferrite is 1 ohm, what is the Q of the RLC circuit?*
3. *In what case would a ferrite filter be useful to use?*
4. *In what case would a ferrite filter not be recommended?*
5. *Why is it important to have the capacitor on the device side of the filter and the ferrite on the board side of the device?*
6. *What is the difference between a ferrite and a power inductor?*

42.6 Grading rubric

As is usual, the score for this report is worth 3 points:

1 point for TA check off

1 point for good measurements and analysis of your measurements

1 point if you include a description of why a ferrite would be useful and in what cases it would not be appropriate.

Chapter 43 Lab 26 ESD measurement and mitigation

Electrostatic Discharge (ESD) and electrostatic damage (also ESD) are common occurrences in any lab with people unless special care is taken. Whenever you move and rub insulators together, you create net static charges on surfaces which generate large fields which can induce charges in other surfaces. These induced charges, even if you don't touch the source can cause damage. If you are charged up and touch an IC and discharge your charge, stored at high voltage, you can damage the IC if it is not protected by diodes.

In this lab you will explore what creates charges and how charged up you are using an Arduino as a static field detector.

We will use the serial plotter in the IDE to display the ESD measurements. I strongly recommend you use the Arduino IDE 1.8.19 or earlier than 2.x. The serial plotter in the v2.x is brain dead and only allows 50 points to plot. This is too few. The earlier version 1.8.19 allows 500 points to plot. It is much better in this lab.

43.1 Purpose of this lab

In this lab you will use an Arduino as a static field detector. This sensor will allow you to "see" the invisible static electric fields all around you. The Arduino E-field meter will allow you to measure the static fields from charged surfaces which are created all the time.

This will illustrate how pervasive and common ESD events are. You can compare the impact of these ESD events if you ground yourself.

43.2 Exp 1: Build an Arduino E-field meter

The Arduino ADC can take ADC readings in about 112 microseconds. This is about 8,000 times a second. Using a slightly enhanced library, we can read ADC signals even faster. Install the library avdweb_AnalogReadFast, and we can speed the ADC readings up by a factor of 5 or more. This library is shown below:



In the Arduino default library, read the voltage on any pin, we use the command,

```
analogReadFast(pinNumber);
```

The pin labels shown on the side of the pins on the board, are the pin numbers we use in the command to read that pin. This command will return the value read by the ADC as a number from 0 to 1023. I call these units analog to digital units, or _ADU.

Every time we call this `analogReadFast()` command, we take a reading from the specified ADC pin and store this number in the command. We could just print this value each time we read it.

To take a reading on analog pin A0, and print the value to the serial monitor, is literally one line of code:

```
Serial.println(analogReadFast(A0));
```

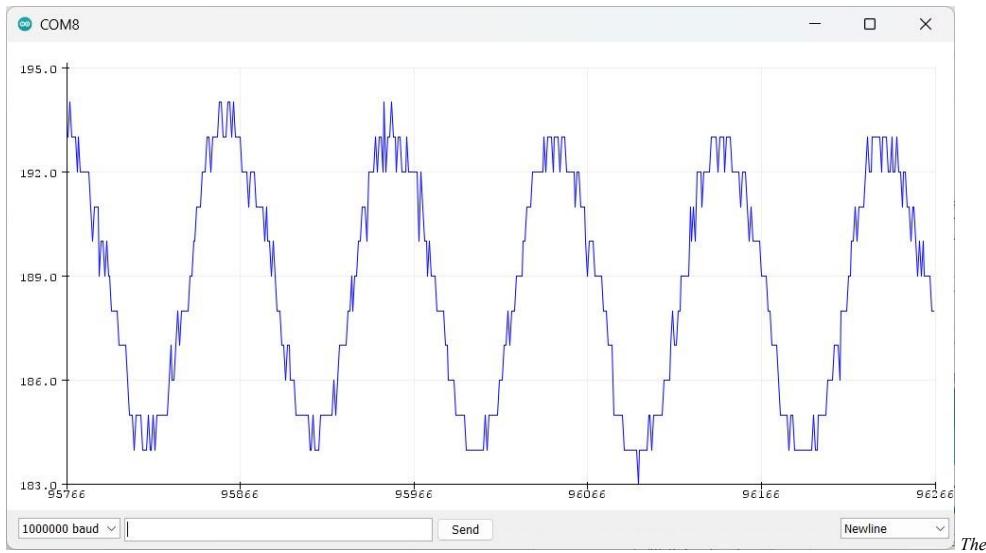
Write a sketch that sets up the Arduino to print the value on the analog pin A0 and plots it on the serial plotter, as fast as it can. What do you think you are measuring?

The whole sketch is only six lines. Here is my version:

```
#include <avrweb_AnalogReadFast.h> void
setup() {
    Serial.begin(1000000);
}
void loop() {
    Serial.print(analogReadFast(A0));
    Serial.println();
}
```

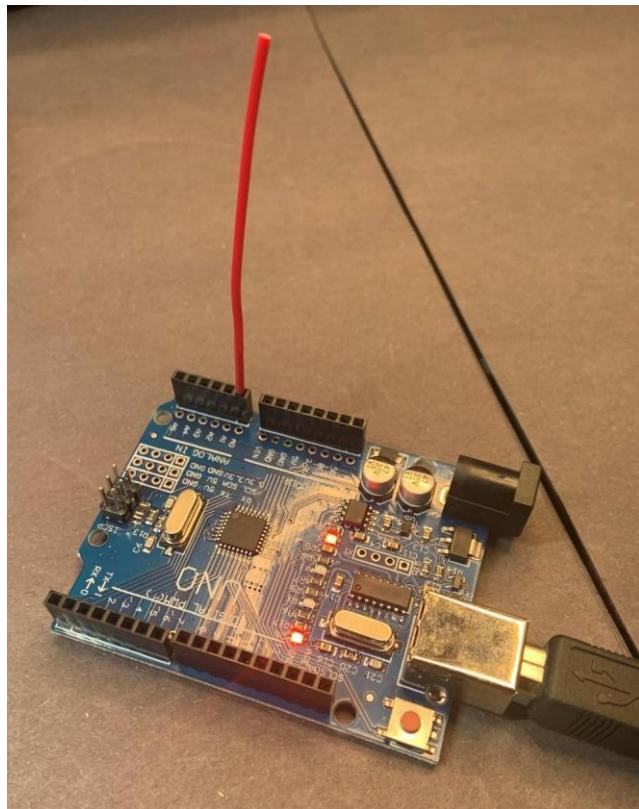
When you run this sketch, you will plot on the serial plotter (don't forget to set the baud rate to 1000000) the voltage appearing on the A0 pin of the Arduino board. Note, depending on the quality of your cables and your computer, you may not be able to run at 2000000 baud. If you get gibberish reduce the baud rate in the set up loop and in the serial plotter to 1000000. On my computer and cables, I could not run at 2000000.

Even with nothing connected, you will see a voltage. What I measured when I did this experiment is shown below.



voltage I measured on analog pin A0 with nothing connected and plotted on the serial plotter. Note the scale is in ADU from 0 to 1023

To increase your sensitivity to stray fields, you can stick a short wire into the A0 pin, no longer than 3 inches. This will act as an antenna and boost the sensitivity of the ADC to stray fields. My Arduino with external antenna is shown below.



What are these sine waves you are measuring?

You are measuring the 60 Hz pick up from stray electric fields in the room around you, radiating from all the power lines nearby. The power line voltage has a frequency of 60 Hz. This means there are 60 cycles of the voltage every second. The period of each complete cycle is $1/60$ Hz or 16.67 msec. This is a good number to remember. The sine wave we see in this measurement has a period of 16.67 msec.

This is the pollution noise that will keep us from seeing the lower frequency static electric fields from static charges. We will need to do two important things to minimize the 60 Hz pollution.

The first and most important thing is to move your Arduino away from any nearby power cords. The power cords and the wiring inside your walls are like bright lights shining on your Arduino. Turn off any lights that might be nearby as these will have 60 Hz currents and contribute to the 60 Hz pollution noise.

Try this experiment: make a note of the peak-to-peak value of the waves on your screen. Then move a power cord close to your Arduino. By how much does the peak-to-peak value increase?

Once you isolate your Arduino as best as you can, the second important technique we will apply is using a digital filter to average out the 60 Hz noise.

43.3 Write a digital filter to average over n power line cycles (PLC)

The 60 Hz voltage pattern has just as many voltage values above the average as below. If we were to average all the voltage readings over one 60 Hz cycle time, with the same pattern above 0 as below 0, we should average to nearly 0. The more cycles we average, as long as it is over an integer number of complete cycles, the closer to the 0 average we will get.

Of course, the more cycles we average over, the longer the time between average values and the slower the plotting. There is a tradeoff.

We are going to write the code to take measurements as fast as we can. Then we are going to average all the voltages we can take over n power line cycles, each lasting 16.67 msec. It doesn't matter when we start as long as we stop averaging exactly 16.67 msec, or an integer multiple, after we start.

This process is called digital filtering. It is a powerful technique to eliminate periodic noise, used in many communications products.

After we implement our digital filter the remaining voltage should be due mostly to the influence from static electric fields.

Think about how you would implement a digital filter to measure as fast as you can, as many data points as possible for an integer number of cycles and display the result.

43.4 My sketch: display measurements averaged over n power line cycles

Here is my sketch. Try writing your own first, then take a look at mine.

```
#include <avrweb_AnalogReadFast.h>
// Measure static electric fields for n PLC
int pinADC = A0; int nPLC = 1;
long iTime2Average_usec = (1000000.0 * nPLC) / 60.0; // time we want to average float
V_ADU;
long nCountsActual;// number of actual measurements averaged
long iTimeStart_usec; // start of averaging time long
iTime2Stop_usec; // stop of averaging time
/////////////////////////////// void setup()
{
    Serial.begin(1000000);
    /// this routine will get the input to the ADC into a steady state value
    for (int i = 1; i < 3000; i++) {
        V_ADU = analogRead(pinADC);
```

```

    }
}

void loop() {
  //initialize variables at start of loop///
V_ADU = 0.0; nCountsActual = 0;
iTimeStart_usec = micros();
iTime2Stop_usec = iTimeStart_usec + iTime2Average_usec;
/////////////////////////////// while
(micros() < iTime2Stop_usec) {
  V_ADU = V_ADU + analogReadFast(pinADC) * 1.0;
nCountsActual++;
}
V_ADU = V_ADU / nCountsActual;
//Serial.print(nCountsActual); Serial.print(", ");
Serial.print(V_ADU);
Serial.println();

////plot with fixed scales
Serial.print(V_ADU); Serial.print(", ");
Serial.print(0); Serial.print(", ");
Serial.print(200);
Serial.println();
}
}

```

I used three tricks in this sketch that may not be obvious.

When we start using an ADC channel, in the way we are, with nothing connected, there is some initial charge stored on the input of the ADC that gives some initial funny values; a start-up transient *artifact*. After taking a bunch of readings, the ADC has eliminated this residual charge and the artifact is eliminated.

First, to get rid of this artifact before we actually take the measurements we want, I added some dummy steps in the setup() function. I told the Arduino to make 3,000 ADC measurements and just throw the values away. This cleans out the residual artifact. Here are those lines:

```

for (int i = 1; i < 3000; i++) {
V_ADU =analogReadFast(pinADC);
}

```

You might have seen these glitches in your measurements. *Figure 43.1* shows the plotter measurements if I do not have this initial clearing out process, and then adding it in. The difference is remarkable.

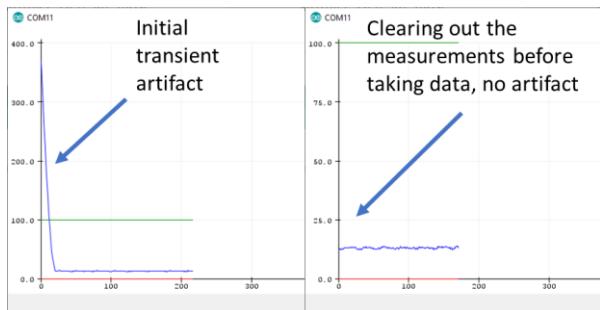


Figure 43.1. Adding the for loop in the setup() function clears the data and there is no initial transient artifact.

Secondly, I wanted to be able to see how many measurements I took in each average. I was curious, when we take data as fast as we can, how many data points are we averaging in one or 2 power line cycles.

When I average over 1 power line cycle, and take data at the rate of about 30,000 measurements per second, I expected about $30,000 \text{ measurements/second} \times 16.7 \text{ msec/cycle} = 500$ measurements averaged together per cycle.

I added the print statements to print the nCountsActual and the V_ADU values:

```
Serial.print(nCountsActual); Serial.print(", ");
Serial.println(V_ADU);
```

When I ran this code, I saw on the serial monitor that I was taking about 512 measurements in 1 power line cycle. This is very close to what I expected and is an important consistency test. It gives me a little more confidence I understand what the code is doing.

Once I know this value, I comment this line out and just print the ADC values I want to plot.

Third, if all I did was plot the voltage values on the plotter, the serial plotter would continually auto scale the plot for me. When we use this to show the impact from static charges, I don't want the scale to change, I want the scale to be fixed so I can see the relative static fields from different sources.

I added two special print lines to force the serial plotter to auto scale on these values. As long as my data to plot is within these limits, the serial plotter scale will be fixed and not change. Here are those lines:

```
////plot with fixed scales
//Serial.print(V_ADU); Serial.print(", ");
//Serial.print(0); Serial.print(", ");
//Serial.println(100);
```

After I see the count and ADU values on the serial monitor, I comment out those print lines and uncomment these three print lines. I can then switch to the serial plotter and see the voltage on the pin, now with the 60 Hz pick up dramatically reduced. The ADU counts at this point are shown in *Figure 43.2*.

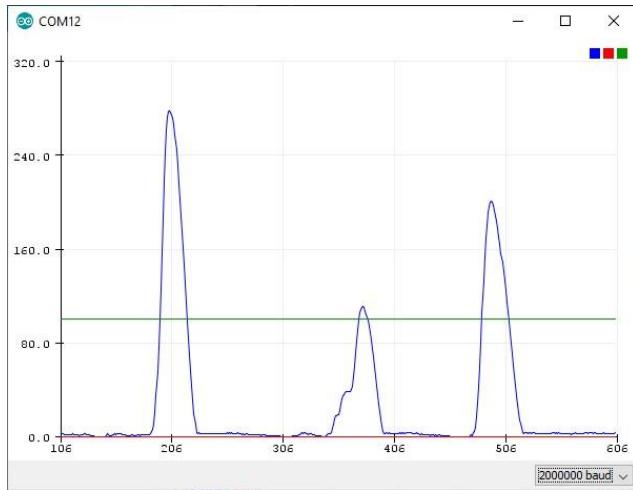
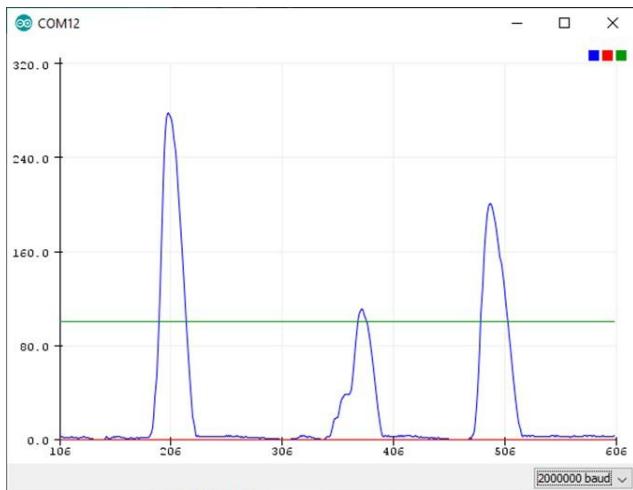


Figure 43.2. The measured ADU values on the ADC channel with 1 power line average and a fixed scale of 0 to 100 ADU full scale. The peaks are from me moving closer or farther away. I am charged up from moving in my chair.

The digital filter removes some of the 60 Hz pickup. And we are sensitive to static electric fields. I charged myself up a little and moved closer to the Arduino and farther away, generating the peaks we see there. Add an antenna to increase the sensitivity to stray electric fields.

Move your hand closer to and farther away from the antenna. Do you see a voltage displayed on the serial plotter? Here is an example of what I measured:



If you want to slow the trace down a little, you can increase the nPLC value to 2.

43.5 What you will do for the lab

Now that you have a static E-field meter, you will use it to measure some of the actions that increase or decrease your static charge build up.

1. *Rub your hands on your clothes or slide on a chair and measure your sign and how charged you are.*
2. *What things can you do will increase your charge build up?*
3. *Touch an earth grounded surface. What happens to your static fields? Do you have to be continually connected?*
4. *How long does it take to be discharged?*

In your report, describe your static E-field meter and some of your observations on what will charge you up and what will discharge you. Remember, chances are a hiring manager will have no idea you could do these sorts of simple experiments with a low-cost Arduino.

43.6 Grading rubric

The grading rubric for this lab is the same as all our labs:

- 1 point for check off by your TA
- 1 point if your lab report is turned in on time
- 1 point if you clearly articulate what you did, what you observed and your interpretation.

Your report does not have to be longer than 1-2 pages, with figures. Feel free to take pictures of your set up as well. Remember, you are explaining to a hiring manager what you did.

43.7 How the ADC measures electric field

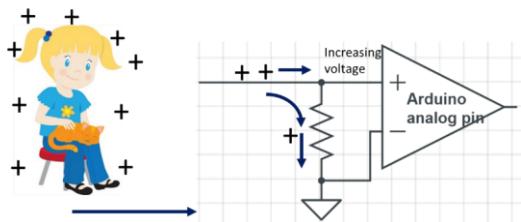
Part of the measurement process in any experiment is thinking through what exactly is being measured. This is called *situational awareness*.

Fundamentally, what is really measured by an ADC is a voltage. How is the presence of static charge converted into a voltage to measure?

To think about this, we are going to look at three simple models of the ADC with increasing complexity.

As a starting place to analyze how an ADC converts a static electric field into a voltage, we will approximate the ADC as an ideal amplifier that just measures the voltage at its front and converts this into

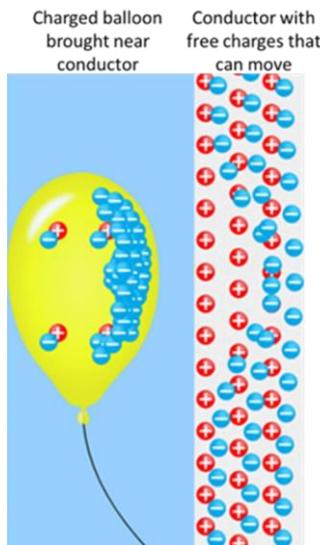
a digital signal. It has a big resistor at its front. The value of this resistor is very large, like on the order of 100 Meg Ohms. A current through this resistor generates a voltage and this is measured by the ADC. This is illustrated below.



When nearby static charges move closer to the antenna, same sign currents in the antenna are pushed through the internal, large resistor of the ADC which creates the voltage we measure.

When we rub a balloon in our hair or over a sweater, the balloon gets charged up, usually negatively. It has a lot of excess negative charges on it.

When this charged balloon approaches a conductor, which has some charges free to move, the charged balloon will push same-sign-charges away from it. The figure below illustrates this.



Negative charges on the insulating balloon are stuck and do not move. When it is brought close to a conductor, it repels negative charges into the conductor.

When we push the balloon closer, the freely moving negative charges in the conductor are pushed away into the conductor. When we move the charged balloon away from the conductor, these free negative charges flow back.

This is what is going on in the antenna:

1. *The motion of the charged balloon induces a current in the antenna sticking out of the ADC pin, due to the balloon's excess charge and its motion.*
2. *This current in the antenna flows through the large resistor in front of the ADC amplifier generating a voltage across the resistor.*
3. *This voltage is measured by the ADC.*

In this simple view, when I moved freshly unrolled tape closer to the antenna, the excess +++ charge on it moved closer to the antenna. These excess +++ charges repelled the plus charges in the antenna and pushed a little bit of positive current through the large resistor at the front of the ADC. This positive current created a positive voltage across the resistor and was read as a positive voltage, for a short time, by the ADC.

When the static charges just sit there motionless in front of the antenna, they are not pushing charges in the antenna and the current through the resistor is 0 and there is no voltage measured. It is only changes in the position of the static charges that are measured.

When the external +++ static charges are pulled away, the plus charges that had been pushed away in the antenna, flow back and the current direction through the resistor changes, so the voltage polarity across it changes. Normally, we should see a negative voltage read by the ADC when we pull the +++ charges away.

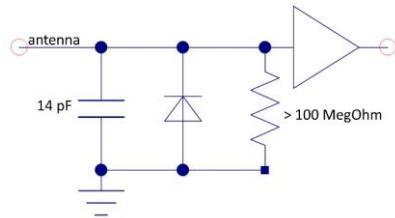
But, in addition to the large resistor across the ADC, there is also a diode that prevents any negative voltage from appearing on the front of the ADC. We can only measure positive currents through the resistor.

In addition, there is one other circuit element across the ADC.

If there were only a resistor across the ADC then the instant the static charges stopped moving, the current would stop and the measured voltage would immediately go to 0 V.

But this is not what we see. When you suddenly stop moving the static charges, the ADC voltage does not immediately drop to 0, it drops over a short time interval, a fraction of a second.

The spec sheet for the Arduino Atmega 328 ADC shows a 14 pF capacitor at the input to the ADC. A slightly more advanced, second order, model for the input to the ADC includes this capacitor and the diode to show that only large positive voltage changes will appear on the ADC. This second order model is illustrated below.



A second order model of the ADC including the diode and small capacitor.

The combination of this capacitor and the large resistor means that if we move the external charge close to the antenna quickly, the induced charge in the antenna moves into the capacitor to charge it up. If the external static charge stops moving, the charge on the capacitor discharges through the large resistor.

The time for the capacitor to discharge, and the voltage to drop to zero is related to the value of the resistor and the capacitor. This model helps to understand the general features of the detector:

1. The ADC only measures motion of the static charge.
2. If the static charges stop moving, the ADC reads 0 V.
3. If a plus charge moves closer, we measure a positive voltage.
4. If a plus charge moves away, we would measure a negative voltage but the diode keeps the voltage from dropping below 0 V.
5. If a negative charge moves closer, negative charges would be pushed through the resistor, but the diode prevents us from measuring a negative voltage.
6. If a negative charge moves away, we measure a positive current in the resistor.

All we can measure is a positive voltage. This is how we can use these principles to measure the sign of the static charge:

7. If we move the charge closer and we see a positive voltage change in the ADC, it's a positive charge we are moving closer.
8. If we move the charge away and it is a positive voltage, the excess static charge on the object is negative.

43.8 A simple static charge experiment with a cat

Static charges are all around us. We cannot prevent static charges from building up on insulators.

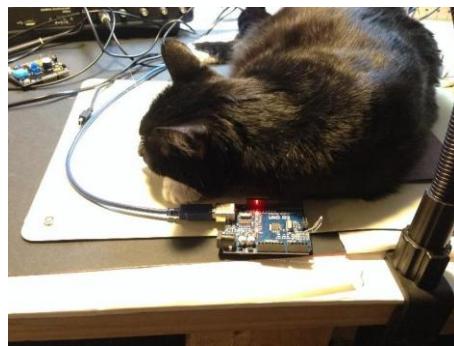
In a humid climate, on every surface is a thin film of water. This makes the surface a little conductive, which will allow excess static charges to flow around and equalize. It's harder to build up static charges when it's humid.

The opposite is also true. In a dry climate, most surfaces are insulating, and static charges can build up creating very high voltages. In fact, the voltages built up from static charges can reach thousands of volts, high enough to breakdown air. This is why you will sometimes get a spark in air after walking on a carpet or petting a cat and touching something grounded.

Usually, when any two insulating surfaces rub over each other, static charges are built up through a process called *triboelectricity*. When you rub a balloon on your sweater or in your hair, the balloon picks up static charges.

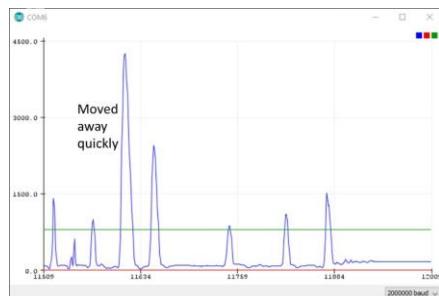
A notorious source of static charge is cat fur. Petting a cat leaves you each charged up.

With the assistance of my lab-cat, Schrodinger, I had a convenient source of static charge right next to my Arduino. The figure below shows Schrodinger in position to offer static charge whenever I pet him. Even though he happened to be resting on a static dissipation mat, he was powerful enough to still provide significant charge.



Schrodinger, my lab-cat, working hard to provide a local source of static charge.

After a few pets, I was charged up. Moving my hand close to the antenna, and then quickly away, resulted in large positive peaks in voltage, only when I moved away. This was very repeatable. The figure below is an example of a series of motions, moving closer slowly and pulling away quickly.



Measured response of moving my hand away from the antenna after having petted a cat and getting charged negatively.

In this experiment, I measured a positive voltage when I moved my hand away from the antenna. This means that I was charged negatively. When I moved forward, I induced a negative current in the resistor of the ADC, which the ADC did not measure because the negative current went through the diode, with a small negative voltage which could not be measured by the ADC.

When I pulled away, I allowed the negative charges to move back into the antenna, now generating a positive voltage across the large resistor connected across the ADC.

43.9 Static charge experiments

The combination of the antenna sticking out of the ADC pin and the sketch we wrote, is a static charge detector. By measuring whether we got a positive voltage by moving closer (positive charge) or moving away (negative charge), we can measure the sign of the charge on the objects moved closer and farther away.

Just as with sniffing 60 Hz stray fields, which opened up a window into the invisible world of ac electric fields all around us, we now have a window opened up to sniff the static electric fields all around us.

Here are some other experiments to try:

Two really common sources of static charge are unrolling scotch tape or packing tape and pulling apart two pieces of Velcro. If you take either one of these examples, there is a good chance it will be very charged.

Take one of these objects and move it closer or farther from the antenna and see if it can detect you. Think about what is actually being measured and why you see this signature. For each object you move closer, it is positively or negatively charged?

Rub your hand over a shirt or sweater and bring it closer and farther from the Arduino. What is the sign of the charges on your hand? Try other surfaces to find a positive charge, and a negative charge source.

If you do not move, you should see the voltage on the serial plotter not change. This should convince you that it is *changes* in the induced charge, flowing through the 100 MegOhm resistor that we are really measuring. Don't move and there is no change in the induced voltage.

When we sit in a chair, when we move around, when we rub our arms, we build up a net, excess, static charge by *triboelectricity*.

When different types of insulating surfaces rub each other, opposite charges are pulled apart and one surface gets charged negatively and the other positively.

We see a striking example of this when we walk along a rug on a dry day and touch a piece of metal, seeing a large spark. We get charged up positively and leave a trail of negative charges on the rug behind us. Depending on what type of clothes we are wearing, we can charge up with an excess negative or positive charge.

It is incredibly difficult to NOT have an excess charge. We have to wear special clothes that are slightly conductive, and our body needs to be connected to a grounded outlet with a wire. It also helps if the room humidity is greater than 50%. Not likely in Colorado where we live.

43.10 Reducing static charge build up

When dealing with sensitive electronic parts, your static charges can potentially blow them up. After you practice building up static charges, practice reducing your static charge build up.

How can you reduce the static charge you carry around? You can use your static charge sniffer to first charge yourself up and then see how effective you can be discharging yourself.

Try some of these methods to reduce your static charge build up:

1. *Touch a surface that is connected to earth ground. This is usually the outside chassis of any instrument plugged into the wall outlet with a 3-prong plug.*
2. *How much static field do you have if you momentarily or continually touch a grounded surface?*
3. *Wear a metal wrist strap that is connected to a good earth ground to bleed off some of your static charge*
4. *Use a tabletop conductive pad, called an ESD mat, connected to a good earth ground to keep static charges from accumulating on the table.*
5. *Wear special clothes which do not have a strong triboelectric effect.*
6. *Rub a wire connected to ground over your arms and clothes to draw off excess charges.*
7. *Keep the room humidity > 50%. Above this humidity, most surfaces have a very thin layer of water condensed on them. This layer is slightly conductive and bleeds off static charge to any other conductive surface.*
8. *Other methods?*

Can you make yourself invisible to the Arduino electric field detector?

Unless some of these precautions are taken, everyone will always have a net charge. This charge will generate an electric field around us. We can't see or feel this electric field, but we can use the Arduino analog pin to measure this invisible presence for us.

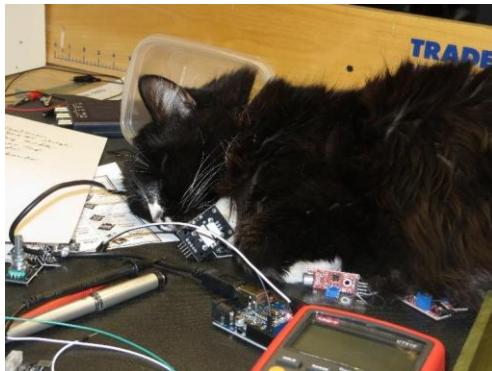
43.11 Electrostatic damage (ESD)

The static charge we build up just due to our normal activities can be disastrous to some electronic components that are not protected. When we are charged up and then touch an electronic component and our charge flows through the component, the large electric field and current burst can damage or destroy the component.

We call what happens when our charge bleeds off onto another device, electrostatic discharge, ESD. When this causes a part failure, we call it ESD damage.

Cat fur has a notoriously high triboelectric effect. As cats move, they build a large static charge. If we pet a cat, we get highly charged.

My other cat, Maxwell, is a great ESD tester. As he walks around my lab bench, touching his nose to my circuit boards, he often delivers a large ESD pulse to the electronics. Parts which are internally well protected with diodes survive. The figure below shows Maxwell hard at work testing the ESD sensitivity of an Arduino Uno board. One reason I like using the Uno board is that it is very robust to all of Maxwell's ESD tests.



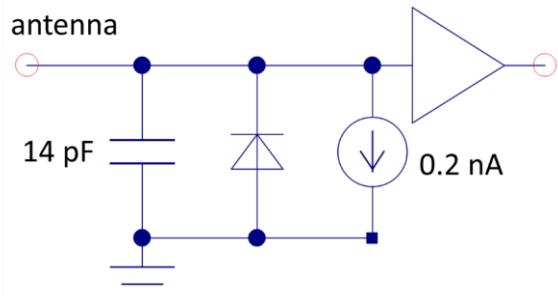
My lab-cat, Maxwell, is a very good ESD tester. His fur builds up a large static voltage which he discharges through his nose when he touches electronic components. The Arduino Uno is very robust to ESD damage which is why I like to use it for my workshops.

43.12 An advanced note

The second order model we created for the front of the ADC is a pretty good model which will help us interpret the measurements we see and how the ADC responds to extra static charges. But, it is not exactly what is going on in the front of the ADC.

In our simple model, we assumed there was a 100 Meg Ohm across the front of the ideal voltmeter of the ADC. In most situations, it's really okay to just think it is 100 Meg Ohms. But in reality, it's not really a resistor. It is a current source created as part of the transistors that make up the ADC circuit.

A better model, a third order model, to describe the front of the ADC is the circuit shown below.



A third order model of the ADC using a current source.

The current source is the input bias current for the amplifier that feeds the ADC. In its normal operation, it draws a steady 0.2 nA or 200 pA of current. If we charge up the 14 pF capacitor by moving some static charge in front of the antenna, this current source will discharge the capacitor in a time of about

$$t = \frac{CV}{I} = \frac{14\text{ pF} \times 5\text{ V}}{0.2\text{ nA}} = 0.4\text{ sec}$$

The current source will bleed the charge from the capacitor linearly in time. This is why the drop off in voltage after I stop moving the charge is more of a linear drop in about 0.4 seconds, rather than the decaying exponential we would expect from an RC circuit.

The third order model is not important to know when using the Arduino ADC to measure static charges. Our simple second order model explains the effects we see with static charges.

Chapter 44 Brd 4: assemble, bring up, bootload, test

Final applications of your brd 4

Chapter 45 Lab extra: SBB circuits: debounce circuit

In this lab you will build three circuits using a solderless breadboard. This will give you some experience in designing and using some of the sub circuits you will incorporate in your board 3.

The circuits you will build are:

- *A crystal oscillator*
- *A reset circuit*
- *Ferrites and filtering of noise*

45.1 Debounce circuits and switches

A mechanical switch we commonly use as a reset is this one:

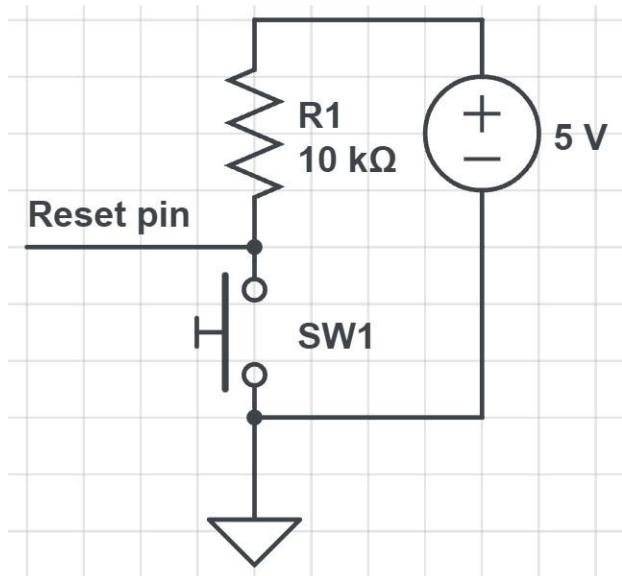
<https://jlcpcb.com/partes/componentSearch?isSearch=true&searchTxt=C174049>

However, for our solderless breadboard lab, we will use this one:

<https://www.sparkfun.com/products/15326>

When it is pressed, a pair of contacts short to each other.

As an example, the figure below shows a simple pull-down switch circuit. The reset pin is normally pulled high, until the switch is closed. Then the rest pin is pulled low while the switch is depressed.



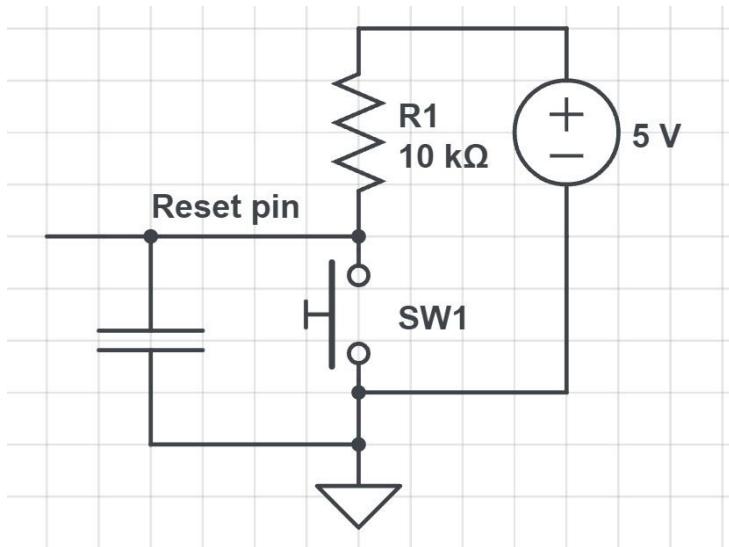
There is one problem with this circuit. Many mechanical switches, after they make contact, “bounce” up and down a few times before they finally stay closed. This is called bouncing. As a little background, check out [this article](#) I wrote about bouncing circuits.

If the bouncing happens on a reset pin, it is possible the multiple contacts will cause multiple resets and potentially a fault condition. The typical spec for a reset pin is that a pin has to be brought low for > 10 usec for the reset pin to read the reset pin going low.

A reset pin would have seen this voltage fluctuate many times and go through multiple reset attempts in the 1 msec time before the switch voltage stabilized.

While this can be fixed in software by adding a delay after a reset is detected before another reset is detectable, it can also be fixed in hardware by adding a debouncing circuit.

The purpose of a debouncing circuit is to hold the reset pin low, after the switch is pressed, and until all the bouncing has stopped. This is as simple as adding a capacitor across the switch, as shown in the figure below.



In this circuit, the resistor pulls the reset line high normally. The capacitor is charged to 5 V as well. When the switch is closed, the reset pin is pulled low and the capacitor is discharged to 0 V as well. When the switch bounces up the first time, the capacitor holds the reset pin low. It has an RC charging time, back up to 5 V, of $R \times C$. If the bouncing time for the switch to settle is short compared to the RC charging time, the reset pin will be kept low.

For example, if $C = 1 \mu\text{F}$ and $R = 10 \text{ k}\Omega$, the RC time constant is 10 msec. As long as all the bouncing time is finished in a time short compared to 10 msec, the reset pin will be kept low. It will not see the bouncing. Any time constant longer than about 1 msec is probably good enough.

Any capacitor $> 0.1 \mu\text{F}$ would be a suitable debounce capacitor. Since the exact values are not critical, use component values you are already using on your board. For example, a 1 k ohm resistor and a 1 uF capacitor, or a 10k and 0.1 uF, or 10k and 1 uF.

This is for the debounce circuit.

It is probably not essential to use a debounce circuit, but it is a good habit when using mechanical switches, especially relays that close hard and often bounce for a long time.

45.1.1 Exp 3: build a switch circuit

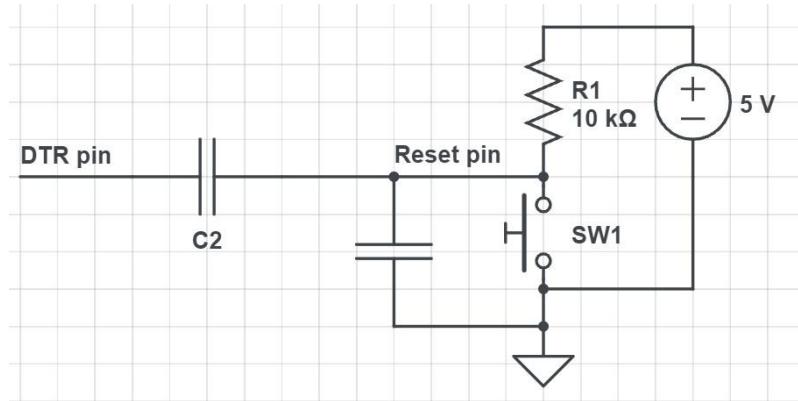
Build the switch circuit above, without the debounce capacitor. Measure the voltage on the reset pin when the switch is pushed. Set up the scope to see the bounces. Hint: use the normal mode to trigger the scope to have it trigger only when the switch is pressed.

Record a few examples of the bounces on the switch.

Add the debounce circuit. What is the expected charging time constant? What do you actually measure? What happened to the bouncing?

45.1.2 Exp 4: debounce circuit with capacitive coupling

The DTR pin on the CH340g chip will also need to pull the reset pin of the 328 down. But, it must be capacitively coupled. The DTR pin may be pulled down for a long time. We want to use the initial falling edge to pull the reset pin down. We connect the DTR pin to the reset pin with a high pass filter- a series capacitor. This will only let through the high frequency, negative, falling edge. To emulate this, we need another circuit, shown below:



The one problem is that we have also built a capacitor voltage divider circuit. The voltage on the reset pin is the voltage divider of the C2 coupling capacitor and the debounce capacitor. If $C2 = C_{debounce}$, the voltage on the reset pin will only be $\frac{1}{2}$ the voltage swing of the DTR pin. It may not be enough to pull the reset pin low.

In order to pull the reset pin low enough when the DTR pin pulls down, C2 should be much larger than the debounce capacitor. If this is 1 uF, then the C2 should be at least 22 uF.

If the DTR pin has a 3.3 V signal range, then when it goes low, the reset pin will only go as low as $5 \text{ V} - 3.3 \text{ V} = 1.7 \text{ V}$. This is not low enough to pull the reset pin low. We need to pull the reset pin below 0.8 V.

Use the digital pin 13 on an Arduino board to emulate the DTR pin of the 340. Set up the digital signal so it is mostly high, but drops down to a low for 10 msec at a time. Use this digital signal to drive the reset pin with a series capacitor.

What is the signal you see on the reset pin? How will you implement this circuit on your board? Design?

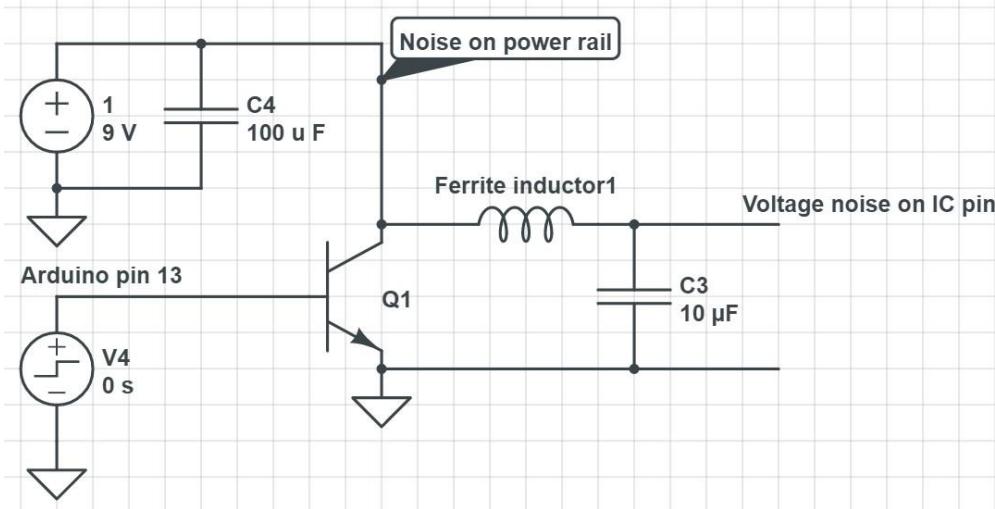
45.2 Ferrites and Filtering

Another problem we often want to avoid in the power distribution path is to filter noise from the board level PDN onto the VCC rail of an IC. This is a form of mutual aggression noise.

This is often the case when we have a noise sensitive part, such as an ADC and a noisy power rail that is polluted with switching noise.

To filter the noise from a power rail getting onto a sensitive power rail, we want to build a low pass filter between the two power rails. The pole frequency should be as low as practical.

We achieve this by adding a large series L with a large C at the IC pin. The circuit and what you will build is shown in the figure below:



The switching noise we would see on the power rail due to higher currents from other devices switching will be filtered by the time it gets through the low pass filter.

If this is all we do when we create the low pass filter, there may be large ringing at the pole frequency of the filter. This will possibly increase the noise on the IC pin at the pole frequency.

To reduce this ringing noise, we reduce the Q of the circuit by adding series damping resistance. This is done for us by using a special inductor that has built in series resistance and higher loss. These inductors are called ferrites. They behave the opposite of low loss inductors used in many power supply circuits that require low loss. Be aware of these two very different applications.

When you select an inductor you will have two choices, a power inductor and a ferrite. Which one will you use in this application?

Note, the ferrites for this experiment look like resistors. Their inductance is about 10 μH and their series resistance, which you can measure using the 4-wire method, is about 1 ohm.

45.2.1 Exp 5: estimate the series resistance needed to damp out the LC ringing

If the decoupling capacitor is 22 uF and the inductance in series of 10 uH, how much series resistance is needed to achieve a Q of 1?

The 10 uH ferrite we have has a series resistance of about 1 ohm. What do you expect the Q to be?

What do you expect the pole frequency to be?

45.2.2 Exp 6: build an LC filter and measure the response

To test the low pass filter, you will build the circuit and drive it with an Arduino digital pin to measure the filtered response.

After you build this circuit, measure the rise time of the falling time for the input signal and the filtered response.

Try driving the Arduino signal at the highest data rate possible. What is the output signal through the filter?

How does the rise time you see compare with what you expect based on your estimated pole frequency?

45.3 In your report, you should include

Write up a report about at least one of these circuits. What did you build, what did you expect to see and what did you measure? Show pictures. Feel free to include all three but include at least one.

When you have your circuits working, have your TA sign off on your lab.

45.3.1 Grading rubric

As is usual, the scoring for this report is

1 point for check off by your TA

1 point if you show examples of what you built, measured and your analysis

1 point if you review the principles behind the circuit coherently.

Remember, this report will look great in your portfolio.

Chapter 46 Lab xx Cross talk in ribbon cables and return connections

46.1 Limitations of the Solderless breadboard

In this experiment, we will use the top section of this board to explore how much cross talk there is using jumper wires as used in a solderless breadboard.

Do this with the golden Arduino using the extra row of gnd connections or with this special noise shield board

46.1.1 Exp 1: cross talk in ribbon cable

Peel off a section of 8 jumper wires, all still connected to the ribbon cable. The first 6 of these will be signal wires. Connect the six signal wires between the left edge of sockets from the digital I/O of the Arduino, to the far right end sockets. While you connect the signal pins, there is no connection to the return path. Even with the I/O from the Arduino pins switching, the LEDs will not turn on because they do not have a complete path to ground.

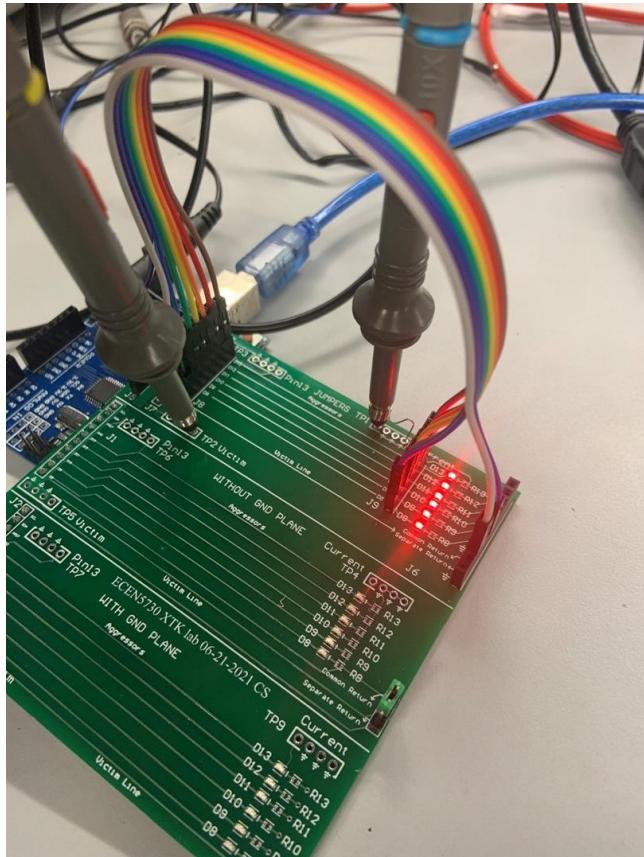
Use the seventh jumper wire to connect the column of ground holes on the left edge to the column of ground sockets on the right edge. The LEDs will now turn on.

The 8th wire will be the victim connection between the victim socket on the left edge to one of two return connection options on the right edge. This configuration is shown in the figure below.

46.1.2 Exp 2: interleaving gnd returns in a ribbon cable

Use the golden Arduino board with the adjacent return pins.

Compare the noise in ribbon cables with the return wire located at one end and using interleaved returns



The ground holes on the far right edge have a special configuration. You should reverse engineer their connectivity using an ohmmeter. The bottom 2 holes are isolated from the top 8 holes. The top 8 holes are all connected together. They are all connected to the common connection to the LEDs and resistors.

If the victim wire is connected to one of the top 8 return holes, the victim wire is essentially shorted to the common return of all the signal I/O. Its return is a common return.

If the victim wire is connected to the 9th hole, it is isolated from the return of the other jumpers. You will use a second jumper wire to connect the 10th hole back to ground hole on the left side. This way, the return of the victim jumper wire is separate from the common return of the aggressors.

In this way, depending on how you connect the victim jumper wire, you can engineer a victim line with a shared return, or its own return.

Measure the relative cross talk in these two configurations.

Circuit uses 2 hex inverter as the source. One of the inverters I set up as an RC oscillator at about 10 kHz.

Use 2 different pin configurations:

The ribbon cable has 20 pins. Use 9 of the hex inverter outputs as switching I/O so that one pin is freed up as the victim line. You will use 3 different configurations for the signal- ground connections:

1. *9 switching I/O with the victim line located in the middle of them, using the first 10 wires and the last 1 wire in the ribbon cable as the one ground.*
2. *Adding 10 ground wires all at one end of the wire*
3. *The right way, with every other wire a ground wire.*
4. *There is a matching connector on the second board with LEDs and resistors to drive current, with the same signal configuration.*
5. *Each of the three connectors can share the same I/O on each of the two boards. It is just the gnd connections in the connectors which are different,*

Use a scope trigger output and a quiet hi and quiet low as the other 3 I/O.

Build ribbon cables with IDC between the signal and the test board.

On the test board, we have matching connectors and test points and LEDs

Using IDC connectors like this: https://www.amazon.com/Keadic-Pieces-Female-Connector-Connectors/dp/B07WHFWMYQ/ref=sr_1_3?crid=RYSWPSQ7QYJ2&keywords=PC+Accessories+idc+connector+kit+latching&qid=1642787464&sprefix=pc+accessories+idc+connector+kit+latching%2Caps%2C93&sr=8-3

Using ribbon cable like this: https://www.amazon.com/Eowpower-16-5Ft-20Pin-RainbowRibbon/dp/B01NC2L2WB/ref=pd_bxgy_img_1/136-35053220795348?pd_rd_w=06td&pf_rd_p=c64372fa-c41c-422e-990d-9e034f73989b&pf_rd_r=XCHXR50B3JMTYC84S30K&pd_rd_r=d2338fcf-cdf1-4f8e-98be-040adde961bb&pd_rd_wg=PCEnV&pd_rd_i=B01NC2L2WB&psc=1

Or pcb to cable connector like this:

<https://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?storeId=10001&langId=1&catalogId=10001&productId=753539>

46.2 In your report, you should include

Before you complete this lab, be sure to be checked off by your TA. You will not receive credit of the lab unless you are checked off.

Based on the measurements you performed, show measurement examples to support the following conclusions:

1. *Cross talk happens mostly when aggressor signals switch their current, at the edges where the current is changing with the largest slope.*
2. *Use short connections at the probe tip to reduce measurement artifacts.*
3. *The lowest cross talk is when there is a continuous return plane under all the traces.*
4. *When individual PCB traces are used, do not share return paths.*
5. *The worst case switching noise will be with jumper wires, even in the case when the return of the victim is not shared.*
6. *(never include any measurement without analysis of how you interpret it and what it tells you)*

46.2.1 Grading rubric is the same as always:

1 point if checked off by your TA

1 point if your scope measurements tell the correct story and you analyze and articulate the principles

1 point if you come close to a coherent explanation of how design features influence crosstalk.

Chapter xx Lab xx: The heartbeat sensor

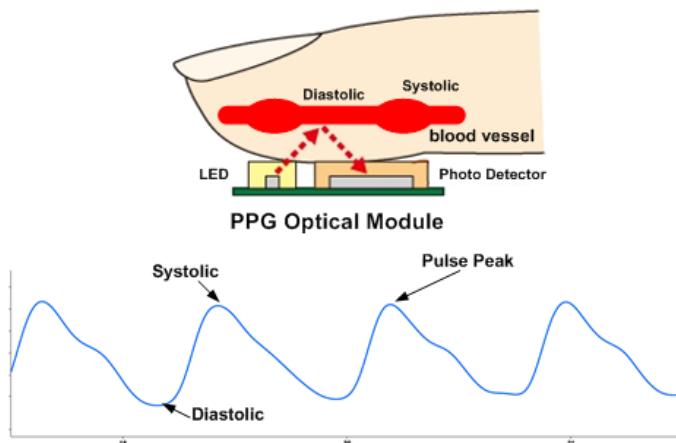
We will use two red LEDs on either side of a photo transistor to create a heartbeat sensor, using the principle of photoplethysmography. This is the principle behind many Fitbits and the Apple iWatch.

Note, the LEDs we will use as the light source will be simple red LEDs, but driven as bright as we can manage. This way we'll have a good reflected signal to detect. We want the light to be relatively constant so that we can detect the small changes due to the blood flow in our finger.

The smart LEDs are modulated by PWM signals. This means they are flashing off and on at about 1 kHz. If we were to use as the light source, the small, reflected light from our finger would be modulated at 1 kHz. We would have to use a lot of averaging to eliminate this modulation.

Plan on using 1206 red LEDs. I have other color 1206 LEDs if you want to try a different color. In order to increase the reflected signal, I recommend using 2 LEDs, one on either side of the photo transistor detector.

When you place your finger over the LEDs, the light will scatter from your finger back into the photo transistor. The amount of light that scatters into the sensor will depend on the blood flow in your finger. This is illustrated in the figure below.



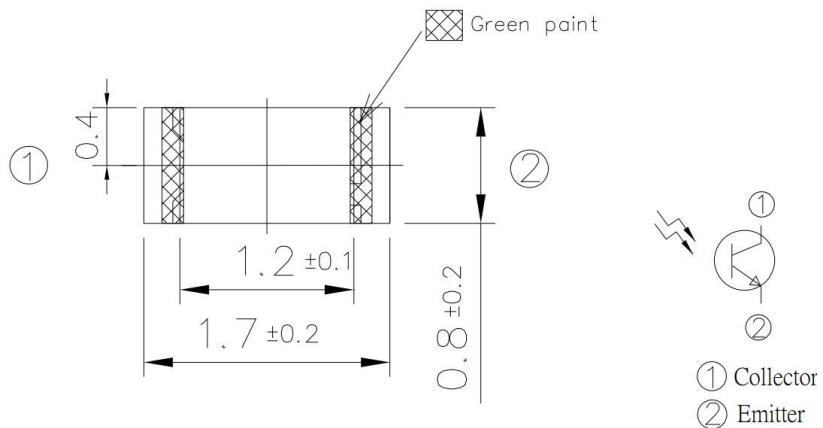
An illustration of the principle of photoplethysmography. From <https://www.richtek.com/Design%20Support/Technical%20Document/AN057>

By measuring the current from the photo transistor, you can measure the blood flow in your finger. This is called photoplethysmography (PPG).

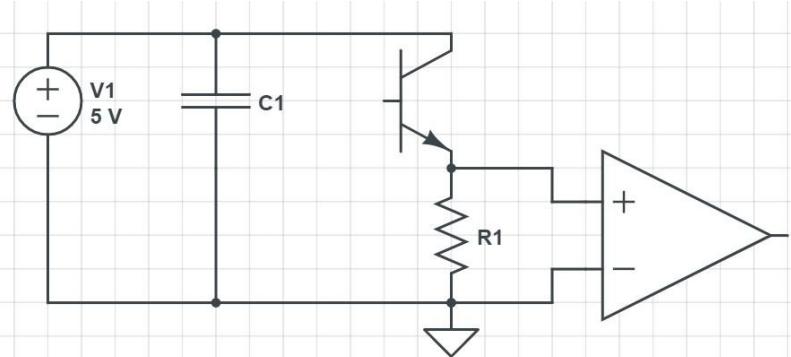
The color you use will be a balance between the sensitivity of the photo transistor and the color that scatters most of the light from the blood in your finger.

The photo transistor is the light sensor. When it is biased correctly, the current through it will be directly proportional to the light intensity it sees. The footprint for this two terminal device is shown below:

Top View



The circuit to turn the current through the phototransistor into a voltage is very simple, shown below:



The C1 capacitor is to provide a local source of charge to reduce the switching noise when the transistor turns on. What value will you want to use?

The resistor, R1, converts the current through the photo transistor into a voltage. What value should you use? You want it large enough to give a measurable signal, but not so large that it saturates for your typical measurements. This is sometimes difficult to determine from the datasheet and will depend on the application. Here are some guidelines:

The specs show that for 10,000 lux from an incandescent light, the output current is 5 mA. This is a sensitivity of about 0.5 μ A/lux of brightness. To interpret this, you have to know how bright a lux is. [Here is a brief guide.](#)

A brightness of 10,000 lux is bright daylight. This is really bright! The maximum brightness you will probably ever see from your finger is less than 1,000 lux. This means the maximum current you will ever measure from your finger is probably about $0.5 \mu\text{A}/\text{lux} \times 1000 \text{ lux} = 500 \mu\text{A}$. For a maximum voltage of 5 V with 500 μA , this would be a sense resistor, $R_1 = 5 \text{ V}/0.5 \text{ mA} = 10\text{k ohms}$.

Without the chance of doing any experimenting, a good starting value for R_1 would be 10k. You should probably use a 1206 size resistor for this part so you can have the chance of replacing it if needed. If you use an odd value, like 9.16k ohms, don't expect to find one in my inventory.

The voltage across the sense resistor will be measured by the 16-bit ADC using a single-ended input. This will use channels A0 of the ADC, for example.

58.1 The 16-bit ADC

A popular 16-bit ADC that communicates over the I²C bus is this one:

<https://www.digikey.com/product-detail/en/texas-instruments/ADS1115IDGSR/296-38849-1ND/5142969>

This chip will communicate over the I²C bus and can be configured to measure combinations of 4 input channels as single-ended inputs or two differential inputs.

The block diagram for this part is shown in the figure below. This is taken from the TI datasheet which you can [download from here](#).

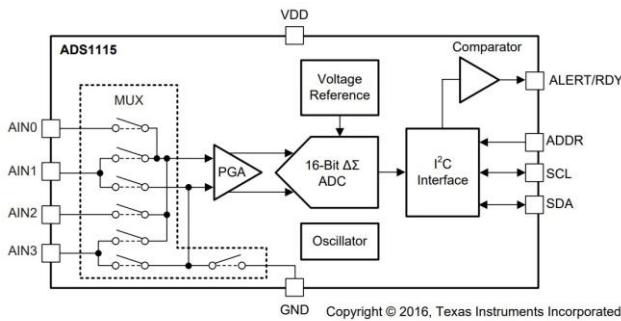


Figure 22. ADS1115 Block Diagram

Note, it can be used as either 4, single-ended inputs or 2, differential inputs. If it is used as 4, single-ended inputs, the – input to the programmable gain amplifier (PGA) is set to ground, at the ground lead of the IC.

If it is set as 2 differential inputs, the + and – input to the PGA are the differential inputs. The output of the PGA, is a measure of the voltage difference between the two inputs and this amplified difference voltage goes to the ADC and is digitized at 16-bit resolution. These settings can be changed on the fly at the time of a measurement.

The PGA can be set for six different ranges, as listed in the figure below.

Table 3. Full-Scale Range and Corresponding LSB Size

FSR	LSB SIZE
±6.144 V ⁽¹⁾	187.5 µV
±4.096 V ⁽¹⁾	125 µV
±2.048 V	62.5 µV
±1.024 V	31.25 µV
±0.512 V	15.625 µV
±0.256 V	7.8125 µV

(1) This parameter expresses the full-scale range of the ADC scaling.
Do not apply more than VDD + 0.3 V to the analog inputs of the device.

This means, when the scale is set for ±2.048 V, for example, each ADU bit level is 62.5 uV. This value is just:

$$\frac{\text{Volts}}{\text{ADU}} = \frac{4.096 \text{ V}}{2} = \frac{4.096 \text{ V}}{65535} = 62.5 \text{ } \mu\text{V} \text{ ADU} \quad (0.1)$$

The interface to the ADS1115 is I2C. This means you need to connect the serial clock, SCL, and serial data, SDA, pins to the SCL and SDA pins of the 328.

In the Atmega 328 chip, the SCL pin on the die is connected to both the analog A5 pin and a separate pin in the header strip above pin 13. The SDA pin on the die is also connected to the analog A4 pin and a separate pin in the header near pin 13. You can use either of these connections. You can use either connection into the Arduino board.

The simplest to use Arduino library for the ADS1115 is from Adafruit. The description of using the library [is here](#). Providing the correct pins to the SCL and SDA pins are connected to the ADS1115, you can use the same library as provided by Adafruit for the Arduino IDE.

While you can download the .zip file from the [GitHub link](#), it is much simpler to install the library in your Arduino IDE.

Under Sketch, select Include library and ManageLibraries, as shown in **Error! Reference source not found..**

Figure 58.1. To install the driver for the ADS1115, select Manage Libraries under Include Library.

The library manager will open up and you can enter *ADS1x15* in the search box. You will then see the only item available, which is the Adafruit ADS1x15 driver. Select install. It is now installed in your IDE.

We will use this ADC in a separate lab on differential measurements and you will get practice installing the driver and sniffing the I2C bus while we perform single-ended and differential measurements.

One of the advantages of using this ADC is that it has a programmable gain amplifier and with 16-bit resolution, has a large dynamic range. As long as the 10k resistor does not create a voltage greater than about 4 V, the photo transistor will not saturate, and you can adjust the gain of the PGA to bring the signal into range of the ADC.

58.1.1 The DAC

This is a 12-bit DAC. It uses the I2C bus, along with the ADC. Because they have different addresses, they can share the same I2C bus connections.

Here is an [excellent tutorial on using this DAC](#). You should connect its output to an Arduino analog pin so that you can measure its output with the Arduino. You may also want to connect it to a test point so you can measure it with a scope.