CU ECEN4/5730

Workbook for Practical PCB Design and Manufacture

fall 2021

Oct 2021 rev

Eric Bogatin

Addie Rose Press

Second Edition

Sept 2021

ISBN xxxxxxxxxxxxxxx

A Publication of Addie Rose Press

Table of Contents

Chapter 1 About the ECEN 4/5730 labs

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

You can check out your own kit at the beginning of each lab and it should be returned to your TA at the end of the lab.

You should complete each lab on your own, but discussion with other students is encouraged.

You are responsible for all the content in your kit. Please be careful with the parts and be responsible to keep track of the items in your kit. Do not leave any parts on your lab bench, or mistakenly put them in the soldering kits.

Accidents happen, and it is ok if something breaks. If something breaks, or is missing, let you TA know and it will be replenished.

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 can check out your kit until you complete the lab.

If you miss a lab, check in with Prof Bogatin 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 regularly scheduled time.

Chapter 2 Kits and Tools and Accessories

Your Buff card should be set up to open the ECEE 281 lab. This will give you 24/7 access when the bld is open.

Most important lab rule is safety for your self and you 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.

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

2.1 The lab equipment

All the labs will be conducted in the ECEE 281 Circuit Labs. You will use the following equipment and will become expert in their use. These instruments are on each lab bench.

- ✔ *Keysight 4024 scope*
- ✔ *Keysight function generator*
- ✔ *Keysight triple power supply*
- ✔ *Keysight bench DMM or Beckman*
- ✔ *Keysight handheld DMM or ANENG 8008 DMM*

2.2 Solder kits

Each of you will use one of the soldering kits available in the 281 lab. There are a total of 30 kits. You may have to share.

If any of these items are NOT in the solder kit you pick up, please let your TA know and it will be replenished.

The following should be in each solder kit:

- ☐ *Weller solder station and power cord and soldering iron*
- ☐ *Blue silicone rubber mat*
- ☐ *Tweezers*
- ☐ *Needle nose pliers*
- ☐ *Diagonal cutter pliers*
- ☐ *Flux pen*

- ◻ *Roll of lead free solder*
- ◻ *Copper or brass sponge*
- ◻ *Safety glasses*
- ◻ *Copper solder wick*
- ◻ *Tube of solder paste*
- ◻ *Small magnifying glass*
- ◻

2.3 In your lab kit, you should find:

- ☐ *BNC coax cable*
- ☐ *BNC to minigrabber cable*
- ☐ *200 MHz 10x scope probes*
- ☐ *red, banana to grabber cables*
- ☐ *black, banana to grabber cables*
- ☐ *1 USB to mini cable for your Arduino board*
- ☐ *Safety glasses*

2.3.1 Building solderless breadboards:

- ◻ *Solderless breadboard*
- ◻ *Box of 6 colored solid AWG 22 core wires*
- ◻ *Pliers*
- ◻ *Wire snippers*
- ◻ *Jumper wires connected on ribbon cable, M-M*

2.3.2 Specific chips, semiconductors, passives

- ◻ *Resistor kit*
- ◻ *2 each 1 uF and 2200 uF capacitors or equivalent*
- ◻ *555 timer (NE555) and (ICM7555)*
- ◻ *2 2N7414 hex inverters*
- ◻ *MCP601 op amp or equivalent op amp*
- ◻ *(2) TIP31c or equivalent NPN transistors*
- ◻ *Leaded 16 MHz crystals*
- ◻ *Axial lead ferrite*
- ◻ *Assorted LEDS*
- ◻ *1 ADS1115 16-bit ADC module*
- ◻ *2 TMP36 temperature sensors*
- ◻ *An Arduino Uno and USB cable*
- ◻ *A 5 V AC to DC supply*
- ◻ *A 9 V AC to DC supply*
- ◻ *Power jack cable with red and black bare wire ends*

- ◻

2.3.3 Specific boards to be handed out at the start of the lab

- ◻ *Digikey ruler*
- ◻ *Assembly practice board*
- ◻ *Blow traces up*
- ◻ *Brd 2 Chaithra' s design*

- ☐ *Switching noise shield for Arduino*
- ☐ *Cross talk with cables board*

Chapter 3 Wk 1: building a portfolio of your projects

Your next step after you leave school 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 create 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 created 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 will 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 page description. Why not consider writing up to put on your portfolio page. What new thing did you learn? What skill would you want to show off to a potential employer?

Maybe:

- *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 processes.*

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, in your portfolio page.

Chapter 4 Wk 1 Mon: build 555 timer as SBB

The first board project you will build is brd 1, the practice board. Before you commit the time (can be as long as 2 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 PDR happens early in the design cycle to review the plan of action. The CDR happens before we commit major resource. In addition, 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.

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.

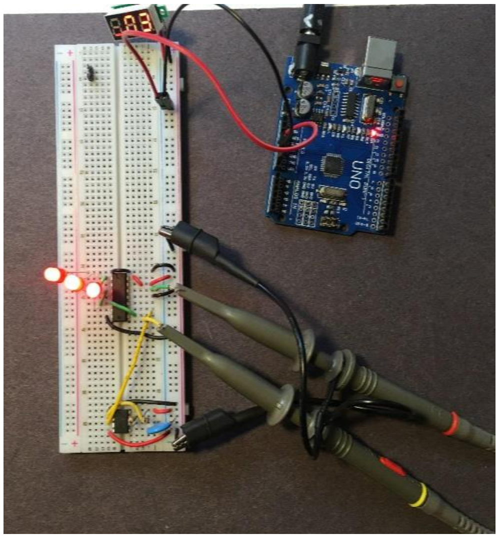
4.1 Before you start this lab

You should have viewed the skill building workshop videos on using the Keysight 4024 scope and 10x probe.

4.2 Purpose of this lab:

1. *Build a prototype version in a solderless breadboard of the brd 1 project you will design and build as a circuit board.*
2. *Try your hand at building a functional circuit in a solderless breadboard.*
3. *Get some hands-on experience with scope measurements.*
4. *Practice finding datasheets of components and reading them.*
5. *Find useful circuits online but make them your own. If you use a specific component in your schematic, know why!*
6. *Make some design decisions, balancing tradeoffs of parts and performance.*
7. *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.

4.3 What you will do in this lab

Goal: design a 555 astable vibrator circuit operating at about 500 Hz and about 50% duty cycle. Use the 555 timer chips and the capacitors in your kit. Pay attention to the circuit design in the SBB and how you will replicate it in your circuit board.

Use a 5 v rail from an Arduino to power the 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 time of the outputs. How well are each of them able to drive the 50 ohm resistor and LED?

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? 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 AFTER you have estimated the figures of merit with your mark I eyeball.*

4.4 Some hints

In this first lab, it is not necessary to use Best Measurement Practices. This lab is just a first step at building and debugging a circuit and performing simple measurements with a scope.

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.)

4.5 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.

4.6 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 get a 0 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?*

4.7 Your lab report

In your 1-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 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.

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.

4.8 Grading rubric:

2 points 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 the scope is not set up properly or if you are using the technical terms incorrectly.

0 points if you turn in nothing or clearly have no clue what you did.

Chapter 5 Wk 1 Wed, wk 2 Mon: Brd 1 design PCB

This lab will be on Wk 1, Wed and carry over to wk 2, Monday

5.1 Purpose of this lab

Complete your first PCB design and submit the board design files.

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. If you wish to assemble the board, you should select parts you can assemble. Otherwise use BASIC parts.*
4. *Add 4 LEDs of all the same color and series resistors: 10k, 1k, 300, and 50 Ohms.*
5. *Use indicator lights, test points and isolation switches as appropriate*
6. *Measure the 5 V input rail, the 555 output voltage and the current through the 50 Ohm LED.*

5.2 What you will need

Altium Designer and the integrated libraries provided online.

5.3 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 What you will do

Before you come to the lab, you should review the videos in SBW-1.

Before you come to the lab, you should make a start at sketching the block diagram, the schematic, the schematic capture and the layout.

Wk 1, Wed: During the lab time, you will finish the schematic capture and the layout under the guidance of the TAs

Wk 2: Mon: come to class with your schematic and layout completed. We will do a CDR in the lab.

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.*

On wk 2, Monday by midnight, students must submit the 3 design files with appropriate names and extensions to the canvas dropbox.

On Tues morning, all board orders will be placed.

5.5 What you will turn in or complete for this lab

The Wk 2, Monday lab assignment is to submit your three design files to canvas.

Your score is 2 if you submit the design files on time.

Your score is 1 if the files are not correct, providing they are re-submitted before midnight, Monday wk 2.

Your score is 0 if they are not submitted. And your board will not be ordered, and you will not be able to complete the brd 1 report.

5.6 The schedule for the completion of brd 1

Wk 1, Wed: finish your design

Wk 2, Monday: complete CDR and submit your design files

Wk 2, Tues: boards will be ordered

Wk 5, Mon: bring up and measure your boards. Assembly ONLY by JLC

Wk 5, Friday: final report due, worth a midterm, 10% of your grade.

Chapter 6 Wk 2 Wed: Using the Scope to measure loop to loop cross talk

This lab will give you practice in using the scope and the function generator to set up and measure one of the most important noise problems with interconnects: loop to loop cross talk.

6.1 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.

In order to start this lab you must first be an expert in using the scope and function generator. If you understand how to use the scope and function generator, you should be able to follow these simple directions and set up the scop and function generator appropriately.

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.

6.1 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.

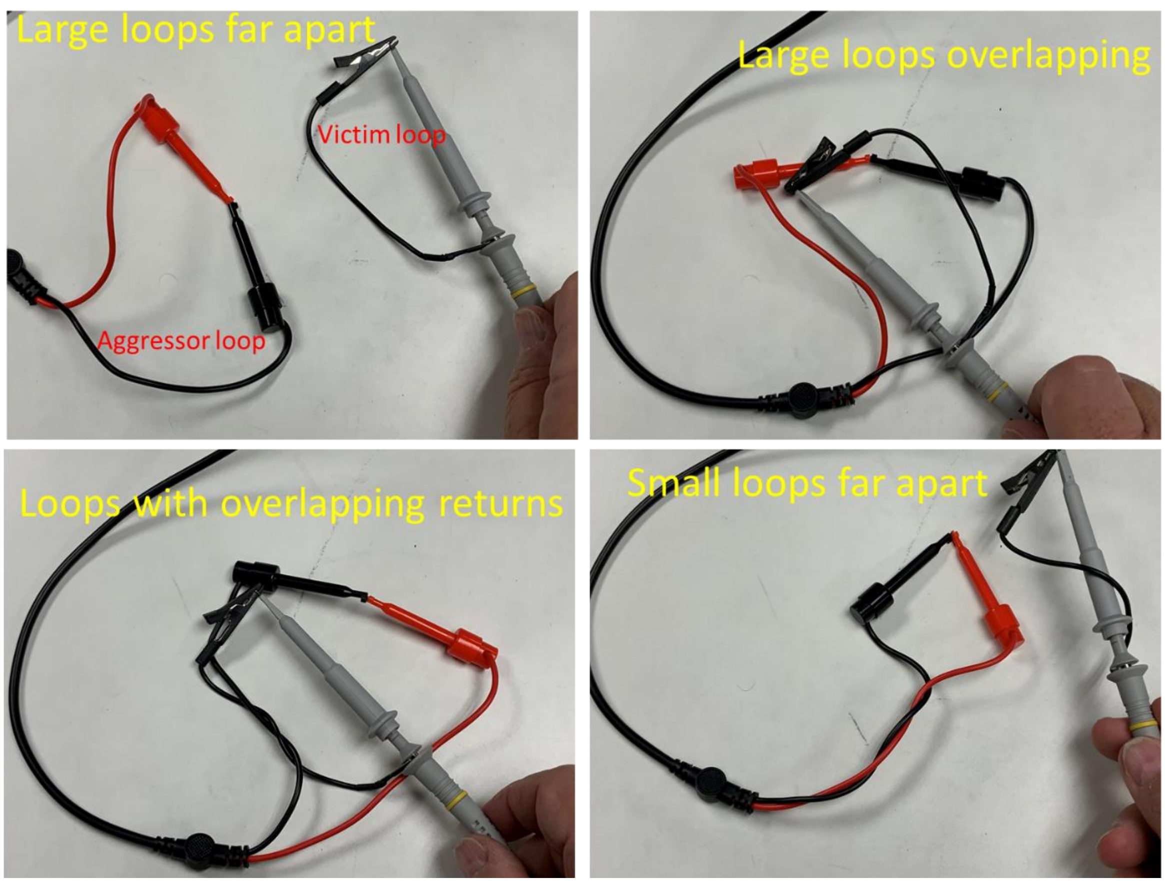
Review the chapters in the textbook on Thevenin models of voltage sources and how to measure them. This is sections: 6.5 and 6.6 of the textbook.

6.2 Setting up the scope and function generator

1. *Set up the scope to measure the output of the function generator, set for a 10 V p-p square wave of 10 kHz. Use the BNC to mini grabber on the function generator and the 10x probe in channel 1 of the scope.*
2. *Make sure the function generator is on High Z output.*
3. *Measure the sync out signal with the scope in channel 2. Trigger the scope on the sync signal. What is the output of the function generator doing when the scope is triggered by the sync signal?*
4. *What do you expect to see as the output signal? (rule #9). What do you actually measure?*
5. *What is the amplitude of the signal, the frequency of the signal and the rise and fall time of the signal you measure with the scope.*
6. *Measure the Thevenin output voltage and source resistance of the function generator by driving a known resistor load. You have just reverse-engineer an important property of this instrument.*
7. *Set the function generator output load to be 50 ohms. What is the Thevenin voltage and the Thevenin resistance?*

6.3 Set up the function generator as an aggressor loop

1. *Use the sync signal to trigger the scope and the function generator on the same square wave setting.*
2. *Short the ends of the function generator minigrabber cable. Knowing the internal Thevenin voltage and the Thevenin resistance of the function generator, what is the current in the minigrabber loop>*
3. *From your measurement of the rise time of the function generator signal, what is the dI/dt in the loop?*
4. *When the signal is rising (what is the sync signal doing on the rising edge of the function generator output?) what is the direction of circulation of the current in the mini grabber?*
5. *Use the 10x probe as the victim loop. Measure the voltage noise induced in the 10x probe synchronous with the function generator edge. How will you trigger the scope to measure the synchronous noise?*
6. *When the 10x probe loop is 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.*
7. *Flip the orientation. Of the loops. What happens to the signature of the noise? Why?*
8. *Explore the cases for the highest noise coupling and the lowest noise coupling.*
9. *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.*



6.4 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 measurements above or to explain any of your measurements.

6.5 The lab report:

Remember, your 1-2 page lab report will make a great example in your portfolio, demonstrating an important electrical effect of inductively coupled noise between an aggressor and a victim loop.

In your lab report, explain your measurement set up (a picture would help).

Show an example of the inductively coupled cross talk between the victim loop and aggressor loop and explain why it has the signature it does. Then show one other example of a different geometry and why it is either larger or smaller based on the geometry of the loops.

Your lab report is due on Canvas by Monday, 9 am.

6.6 Grading rubric:

2 points if you have a clean scope trace and articulate what you measured and what it means

1 point is you did not set up the scope correctly or if you are confused about the interpretation of the measurement

0 points if you do not have a clue or did not complete the lab

Chapter 7 Wk 3 Wed: 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 eats a lot of current to perform some computational operation.

When there is a sudden current draw on the power rail, the current flows through the inductance of the power rail. The dI/dt through this inductance causes a voltage drop. This is a serious type of noise which can be dramatically reduced with a decoupling capacitor.

You will measure the signature and value of this voltage drop and see how much you can reduce it using the local charge storage of a decoupling capacitor.

You will explore the six 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 10 uF of decoupling capacitance, and then as large a capacitance as practical.*
4. *Where you measure the noise on the power path influences how much noise you measure.*
5. *Short rise time signals have a larger dI/dt and show more power rail noise than long rise time signals.*
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.*

7.1 Purpose of the lab

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.*

7.2 Prep before you start this lab

Read the section in the textbook about power rail switching noise: Chapter 13.

7.3 The big picture

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

The slammer circuit is a transistor that turns on and draws a fixed current from the power rail. It will be triggered by either a digital signal from and Arduino digital I/O which has a 5 nsec rise time or by an opAmp output with a rise time of about 1 usec.

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 collector pin for these different rise times with and without decoupling capacitors.

7.4 Install the Arduino IDE

1. *Download the Arduino IDE from Arduino.cc. Then install it.*
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.*

7.5 Set up the two driver signals to the transistor's base

The Arduino will provide the input signal to turn on the transistor's base, 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 transistor 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 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.

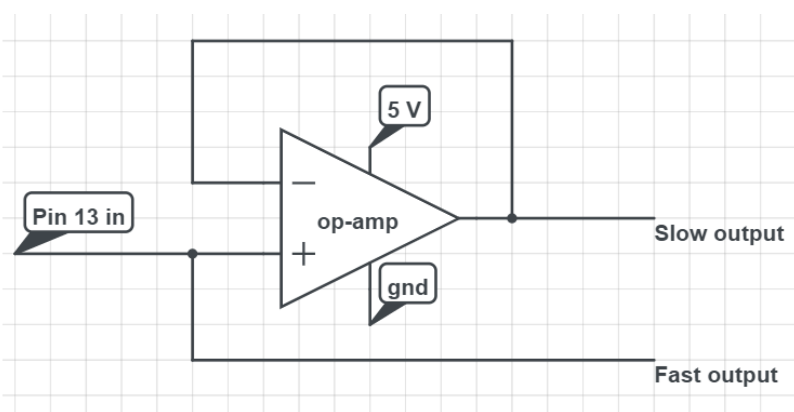
If you are not sure which chip is on your Arduino, look carefully at your board. If you are still not sure, download and install both drivers. You cannot go wrong.

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 identidied 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.

On the solderless breadboard, add the op-amp follower at the bottom end of the solderless breadboard. Set up an op-amp in your board as a follower. Connect the output of pin 13 to the input of the follower circuit.

The circuit you will build is very simple, shown in the figure below



Simple buffer follower to generate two signals from pin 13, one with a fast edge and one with a slow edge.

1. *Use the external 9 V AC to DC supply to power the Arduino.*
2. *Connect the USB cable to the Arduino and make sure you can upload a sketch.*
3. *Write a modified Blink sketch to turn Pin 13 on for 1 msec and off for 20 msec*

4. *Measure this signal with the scope when it is running and verify the duty cycle is less than 10%.*
5. *Be sure to use color coding for the solderless breadboard and the power and ground right hand side columns.*
6. *Did you verify that the top half of the columns are connected to the bottom half of columns on your solderless breadboard?*
7. *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.*
8. *Power the opAmp with 5 V from the Arduino using a black and red wire from the Arduino to the solderless breadboard.*
9. *Did you use the correct color code?*

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 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.

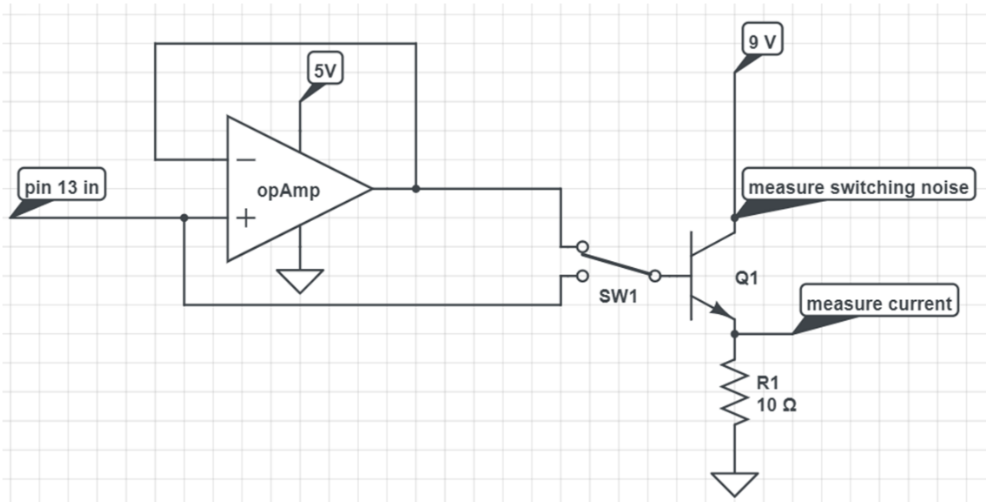
What is the rise and fall times of these two signals?

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 20 msec or longer. If the duty cycle is larger than 10%, the circuit may heat up too much.

Remember if you have large loops at the tip of the 10x probe, you will have the potential of cross talk noise ***in the probes***. What consistency test can you think of to verify how much of your scope measurement is just induced noise in the tip's loop?

7.6 Build the slammer circuit with no decoupling capacitors and the slow edge.

We will build the slammer circuit in stages. First, we start with just the transistor and NO decoupling capacitors. The slammer circuit is shown in the figure below.



Simple slammer circuit with the transistor connected to the power rail.

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

This is also a constant current-source circuit. The transistor will turn on with whatever current flow through it so the voltage across the sense resistor and the base to emitter voltage drop adds up to the voltage on the base.

The only way we get a voltage across the sense resistor is with a current through it. You should derive the current through the resistor as:

$$I = \frac{(5V - 0.6V)}{10\Omega} = 440mA$$

When the signal to the base turns on, there will be about 440 mA current draw from the power rail. This current will turn on with the rise time of source filtered by the response of the transistor.

In the first series of measurement, use the slow edge, from the buffered signal.

Now it should be obvious why it is so important to use a signal with a duty cycle less than 10%. At steady state, the power consumption in the transistor is about 5 V x 0.4 A = 2 watts. This will heat the transistor.

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 440 mA, instantaneous power dissipation through the resistor would be:

$$P = I^2 R = 0.44^2 \times 10 = 1.9\text{watts}$$

The resistor is rated at only ¼ 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 down 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.

7.7 Measure switching noise with a slow rise time

Construct the circuit in this order:

1. *Build the slammer circuit in the solderless breadboard. Use the best practices for the solderless breadboard as described in the text. Use discrete wires, not the long floppy jumper wires.*
2. *Connect the transistor. Check the* datasheet for the transistor.
3. *Add the 10 ohm series resistor from the emitter to ground.*
4. *This is the configuration of a board with no decoupling capacitors.*
5. *Initially, connect the slow rise time signal from the buffered output to the base of the transistor. It should be a 5 V signal, 5% duty cycle.*

Practice Rule #9 for all measurements.

6. *Measure the voltage on the base. What did you expect to see?*
7. *Measure the voltage on the sense resistor. What is the current through the collector?*
8. *What is the rise time of the current and the dI/dt?*
9. *Use the current through the emitter as the trigger for the scope on chan1*
10. *On chan2, measure the switching noise on the power rail of the die- this is the collector pin.*
11. *What is the switching noise on the collector? Why does it have the shape it does? At short rise time and long rise time.*
12. *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?*
13. *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.*
14. *Does the noise change as you move down the power rail on the solderless breadboard?*
15. *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.

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?

7.8 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 base, 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 collector pin for the rising and falling edges.*
4. *Look at the signal on a very fast time scale and a slow time scale.*
5. *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?*
6. *Why is the switching noise so different between the slow edge and the fast edge?*
7. *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 mA and the dt = 50 nsec, then the loop inductance is 8 V/0.4 A x 50 nsec = 1000 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 nH x 0.4 A /2 usec = 0.2 V, too small a value to see compared to the pattern of the changing current.

7.9 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 collector 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 first.*
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 collector pin and measure the switching noise on the collector pin.*
7. *What do you think will happen to the switching noise if we move the capacitor closer to the transistor's collector pin?*
8. *Why does the switching noise decrease as you move the capacitor closer to the collector 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. *Does the value of the decoupling capacitor, either then 1000 uF or the 1 uF capacitor, effect the amount of switching noise?*

7.10 Charge depletion time in the capacitor

When the 440 mA of current flows through the collector on the power rail, there are three features in the voltage response:

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 440 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:

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

How does your estimate of the slope of the voltage droop match what you actually measure?

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

7.11 Check out by your TA

Before you leave the lab, call your TA over for a check out of your experiment. 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 transistor? What is the rise and fall time?*
3. *Where does the power rail switching noise come for?*
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 capcaitor 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 collector pin 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?*

7.12 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 transistor collector.*

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 capcaitor*

4. *With a scope screen capture, illustrate the impact on the switching noise of the 1 uF and 1000 uF capcaitor.*

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****.*

7.13 Grading rubric:

2 points if your scope traces clearly show the features you want to illustrate the cause of switching noise ad your explanations are clear.

1 point if you made an attempt but do not demonstrate understanding of switching noise.

0 if you did not complete the lab, or if you were not checked off by your TA, or if you demonstrated you do not have a clue.

Chapter 8 Wk 4 Monday and Wed lab: measure cross talk between signal-return loops in a special test board

In this lab we are going to explore three 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.

8.1 Purpose of this lab

You will explore three 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.

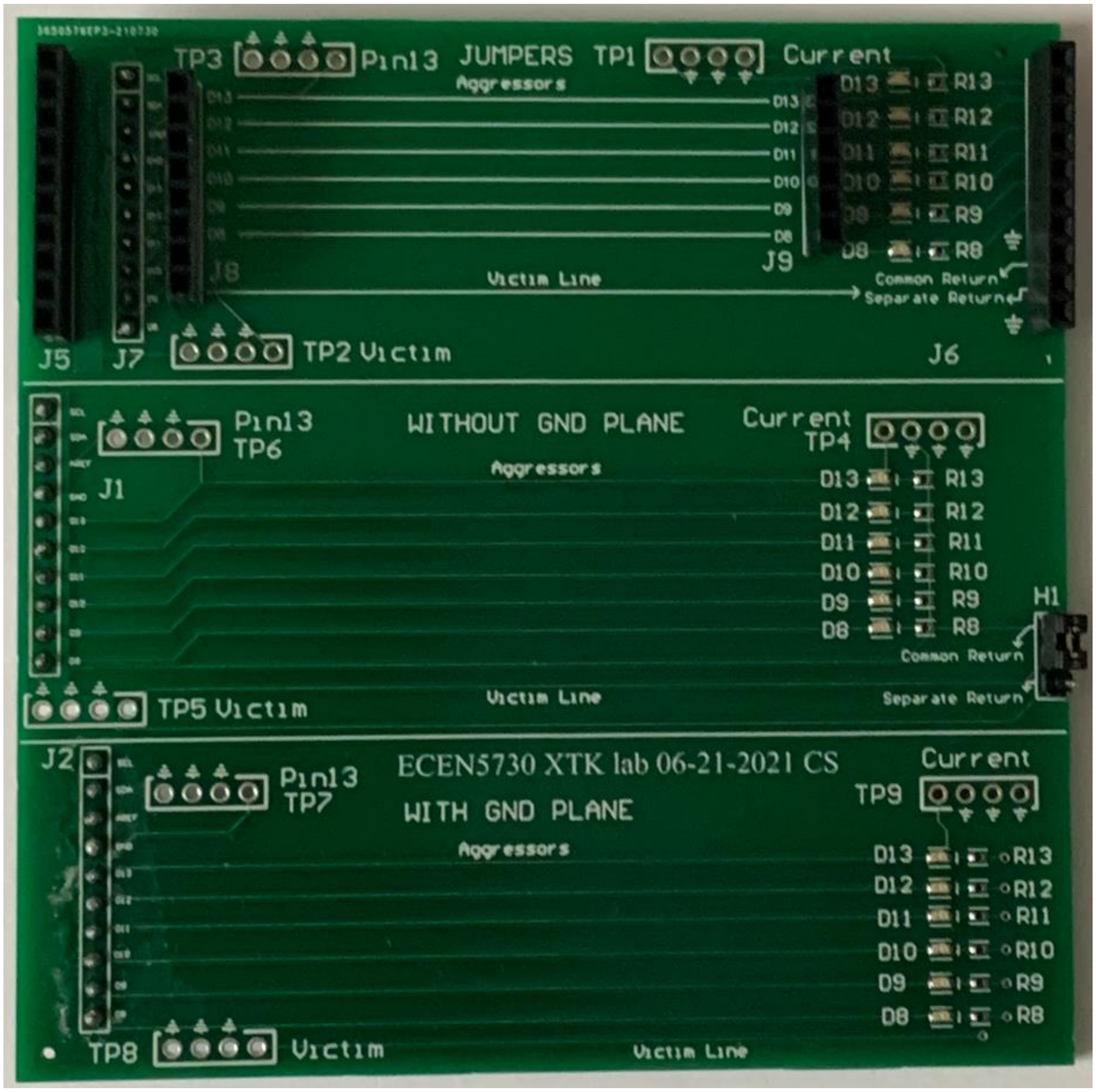
In each case, you will also want to follow the interconnect path of the signal and return conductors of the aggressor and the victim loops.

The purpose of this lab is:

1. *To measure the probe to probe cross talk and best measurement practices to reduce it.*
2. *To gain practice with the best measurement practices to measure switching noise cross talk by triggering the scope on the aggressor signal.*
3. *To look at different wiring options for signal paths and return paths.*
4. *To learn to distinguish and identify the signal path and the return path of any interconnect*
5. *To compare the timing using a digitalWrite command and a PORTB command*
6. *To write the microcode for a pulse train for 1 to 6 I/O that switch simultaneously with different patterns.*
7. *To measure the victim noise signature.*
8. *To evaluate how the switching noise scales with the number of simultaneous switching signals*
9. *To evaluate how the cross talk to the victim line varies as the physical wiring of the aggressor and the victim changes.*
10. *To see the routing geometry that creates the lowest cross talk.*

8.2 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:



8.3 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

Watch the four lab videos about the four different measurements you will do.

8.4 What you will do

There are 4 parts to this lab, spanning two lab sessions.

In part 1 you will measure the cross talk between two 10x probe loops using long loops and using very short loops. This should convince you never to use long floppy loops when you measure a signal with a 10x probe.

In part 2 you will measure the best-case cross talk, when there is a continuous return path.

In part 3, 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.

In part 4, you will measure the huge cross talk between jumper wire loops when the return is shared and how much it can be reduced when it is not shared.

The signal source will be an Arduino generating from 1 to 6 output pins switching simultaneously in various patterns.

8.5 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 on the victim path, then it is sometimes hard to distinguish what is the signal and what is the cross-talk noise. To make our job of measuring the cross-talk noise easier, we will turn on the signal on the victim path so that any voltage we measure will only be cross talk 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 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. Durring 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:

$$V_{victim} = M \times n \times \frac{dI_{aggressor}}{dt}$$

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. In this lab, we will use discrete jumper wires, 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 of 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 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.*

8.6 Part 1: 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 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 the first part of the 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.

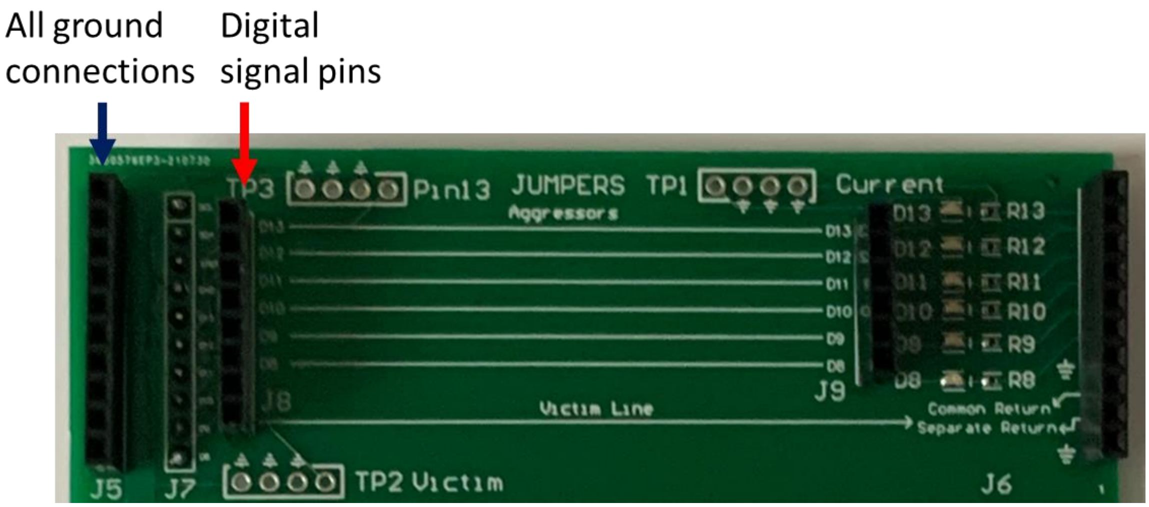
You should write your own code. Here is my sketch:

```
void setup() {
  pinMode(12, OUTPUT);
  pinMode(13, 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);
}
```

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 received 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 pins 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:

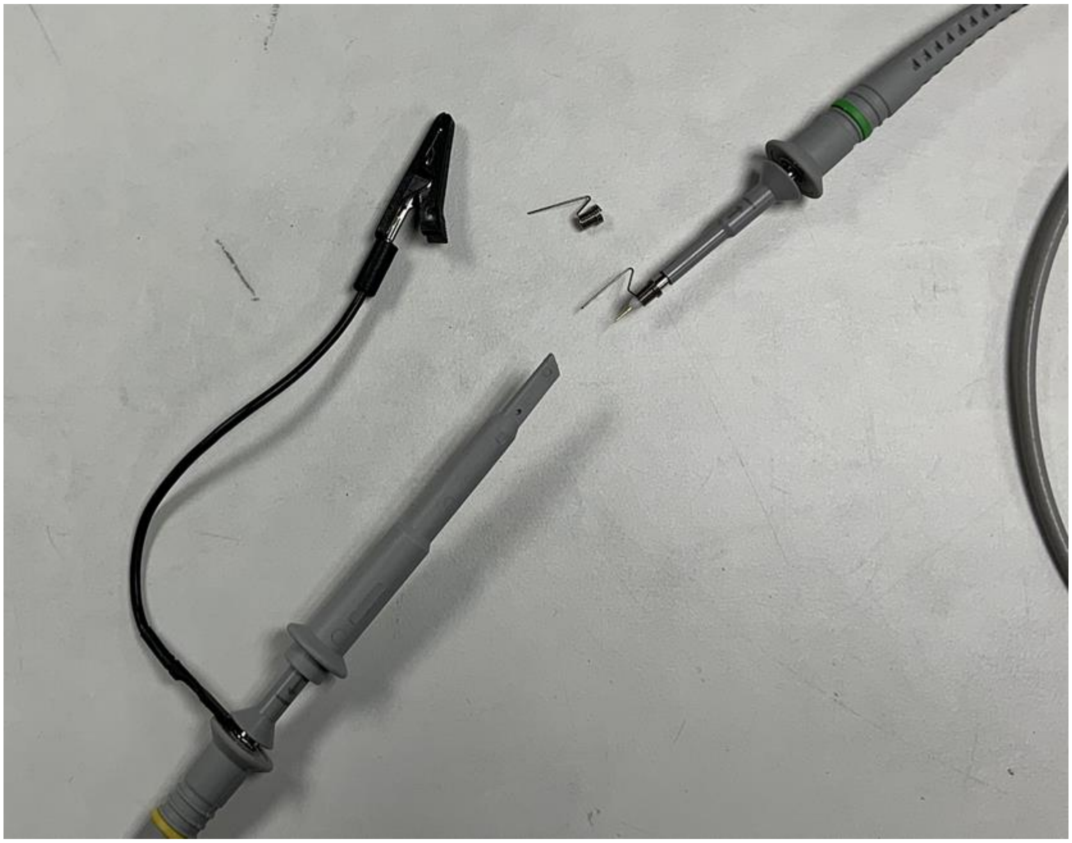
1. *Make sure the probe is set for 10x*
2. *Make sure the scope is set for 10x attenuation*
3. *Use a color code band on the 10x probe to match the color of the trace on the scope*
4. *Check the compensation of the 10x probe*

Step 1: impact of tip loop inductance on signal measurement quality:

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 quality of the measurement?

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.

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 tip exposing the sharp tip of the probe. Screw the spring probe tip onto the scope probe, as shown in the figure below.



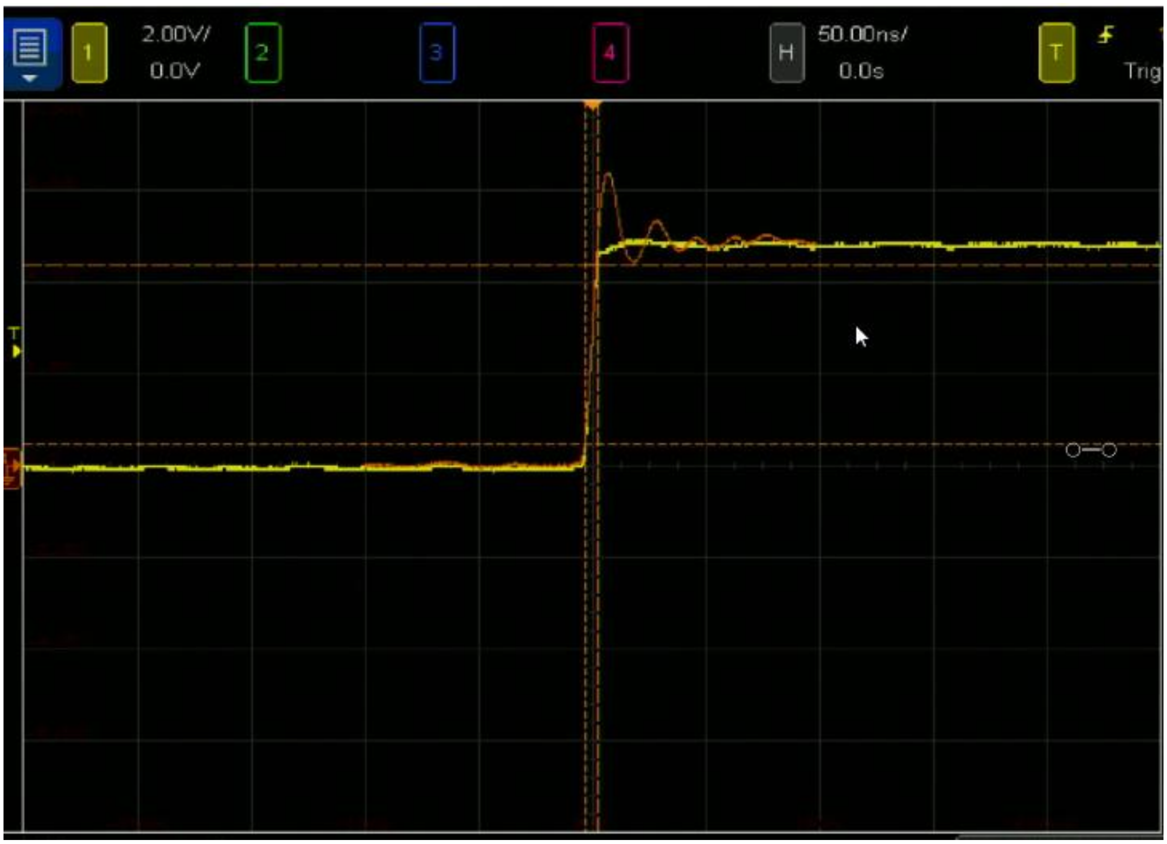
Now you have the shortest practical signal-return loop on the probe tip.

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?

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.



Step 2: 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. 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. Before you connect the second probe into pin 11, which is a LOW signal, connect both the signal and return of the second probe into the ground pins, so they are shorted together. 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? Where does this signal come from? Since it should be 0 V, everything you are measuring is an artifact from your measuring method.

Next, connect the second probe into the pin 11 output. You should see the distorted signal on pin 13 in channel 1. 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 on when the tips were grounded.

In fact, 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 cross talk noise. This is a measurement artifact.

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.

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, the traces on the board and the single ground connection from the Arduino board to the small adaptor board shield.

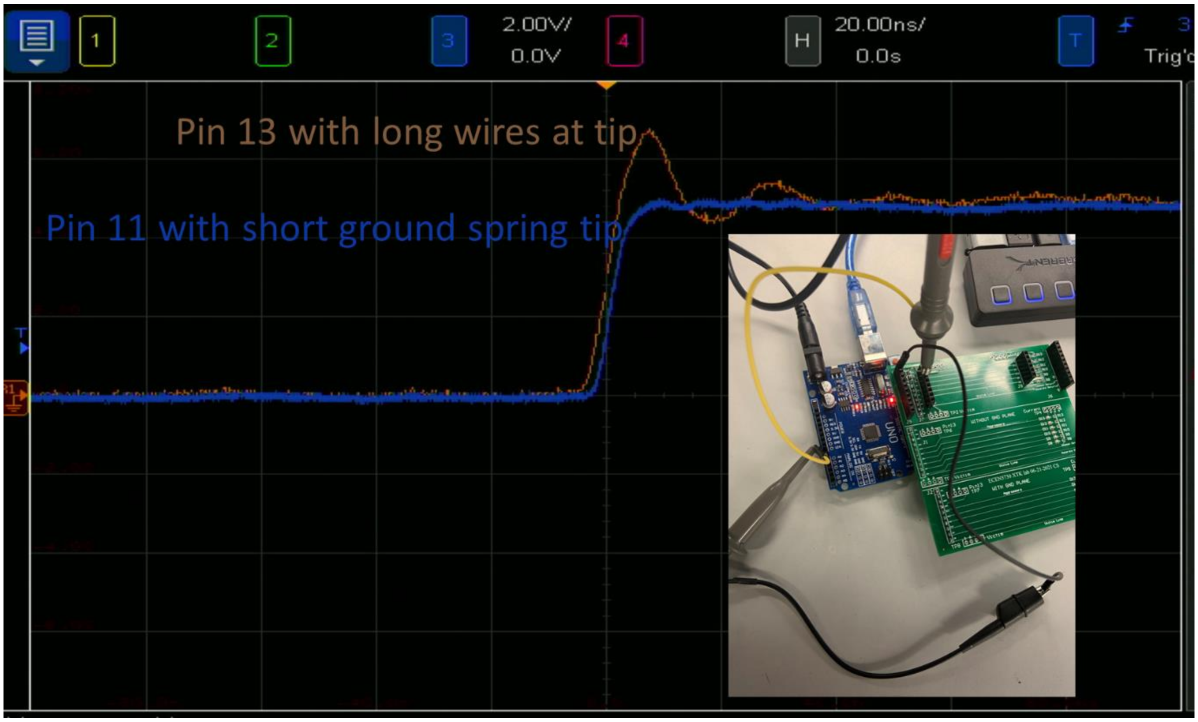
From this exercise, what do you conclude about the best measurement practices for high speed signals?

If your signals have a 100 nsec rise time, this measurement artifact would be dramatically reduced. But if your signals are a few nsec, 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 they 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.

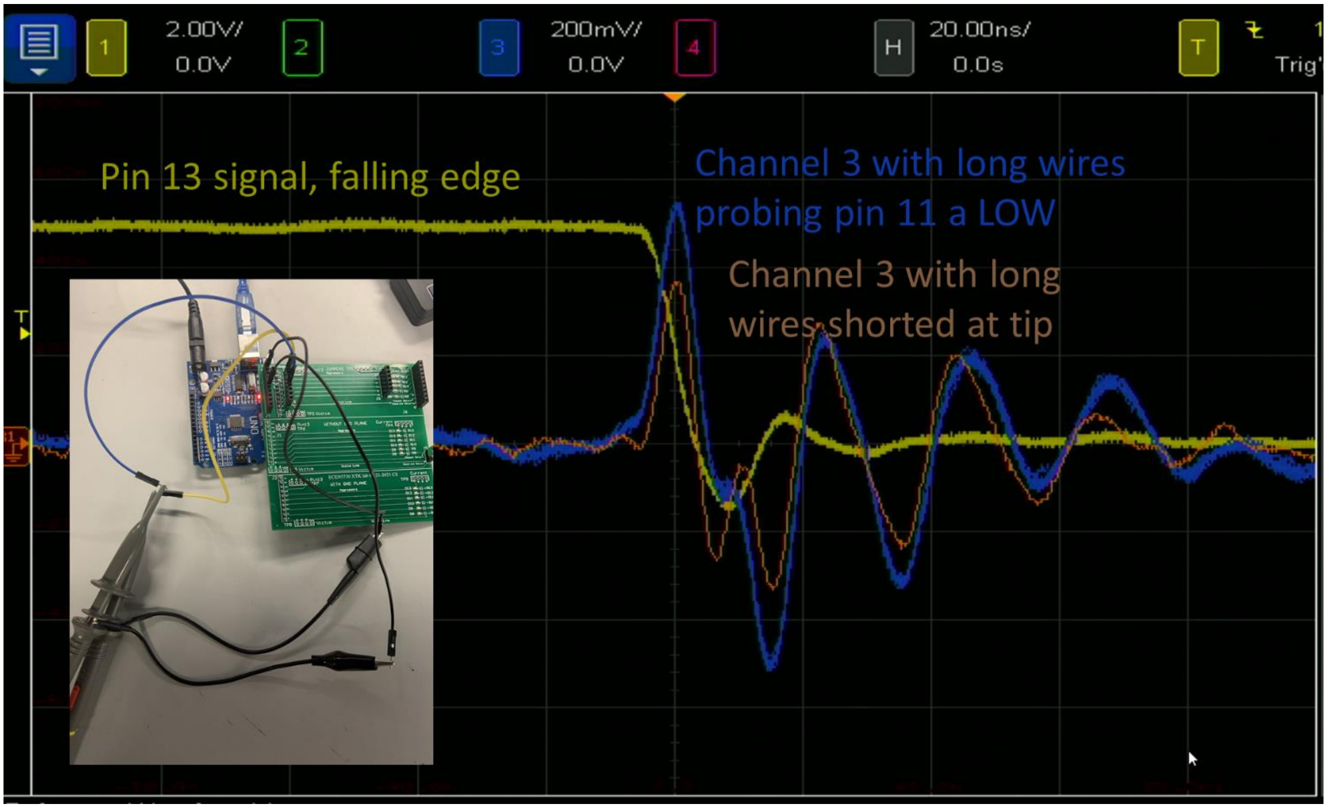
Here are my examples when I did this measurement. Below is the signal from pin 13 with long floppy wires on the tip and the signal from pin 11 with the small spring ground tip. All the ringing and overshoot in the pin

13 measurement was due to the large loop inductance of the wires from the probe to the pin under test. When this inductance was eliminated, the ringing is gone.

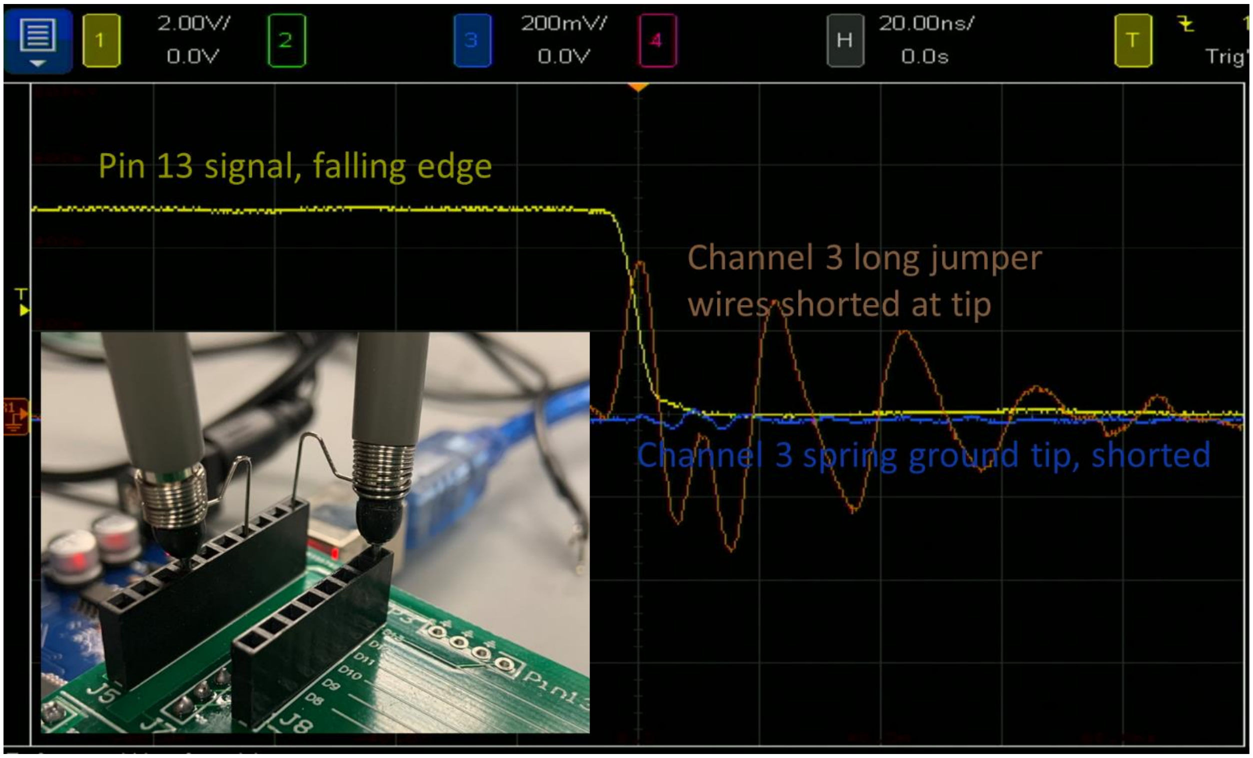


This measurement suggests, when probing high speed signals, use short wires at the tip, preferably the pin and spring ground tip.

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.

8.7 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 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.*

8.8 Part 2: Cross talk with a continuous return plane

In Part 2 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 switch 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 higher current more slowly than it can shut the current off.

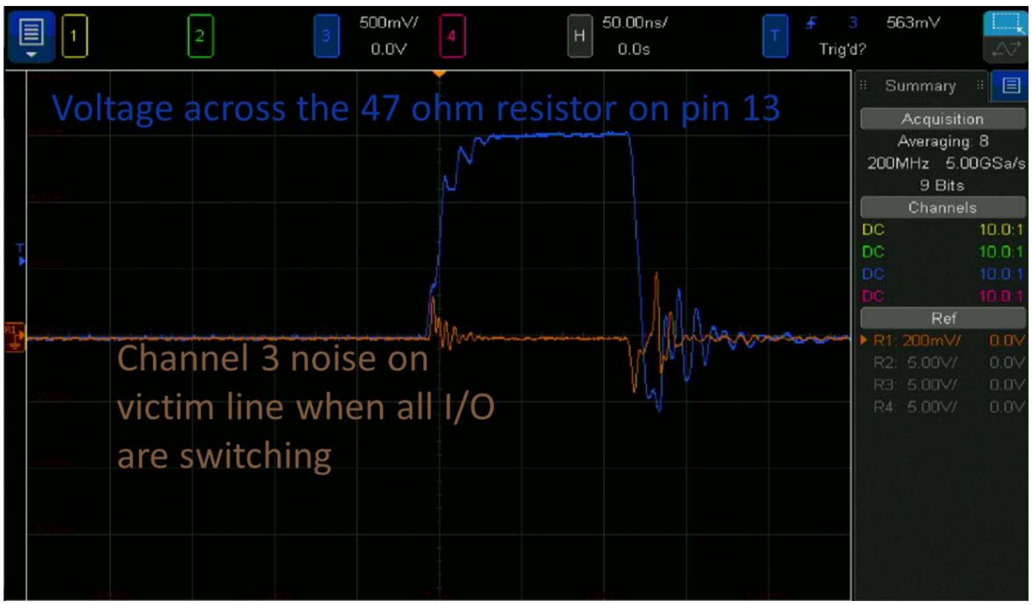
This means the dI/dt will be larger on the falling edge and the inductively generated cross talk noise will be larger on the falling edge.

Using the voltage across the 47 ohm resistor to trigger the scope, measure the voltage on the victim trace. All measurements should be done using the spring ground tips. Save this noise on the victim line as your reference. You will compare this noise with the other noise you measure in other configurations. At least 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 have 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 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 tlak 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.

8.9 Part 3: Cross talk with no plane, but an adjacent return trace

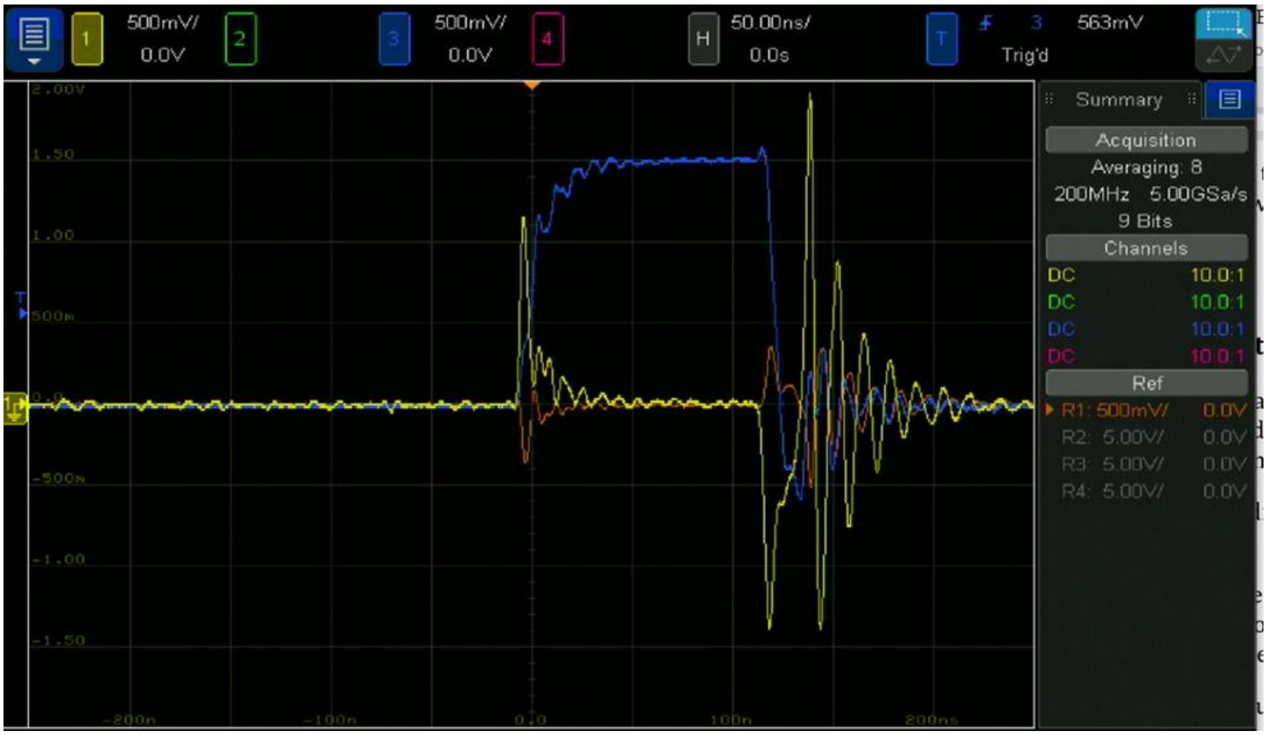
Part 3 of this lab uses the middle section of the board. Again, all the signal pins are connected to traces that span the length of the board and connect to LEDs and resistors which are all bussed together through a common return trace. There is no plane on the bottom of the board.

Trace the signal paths and their returns on this board. You can drive these traces with the same signals as in the last section.

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.

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?

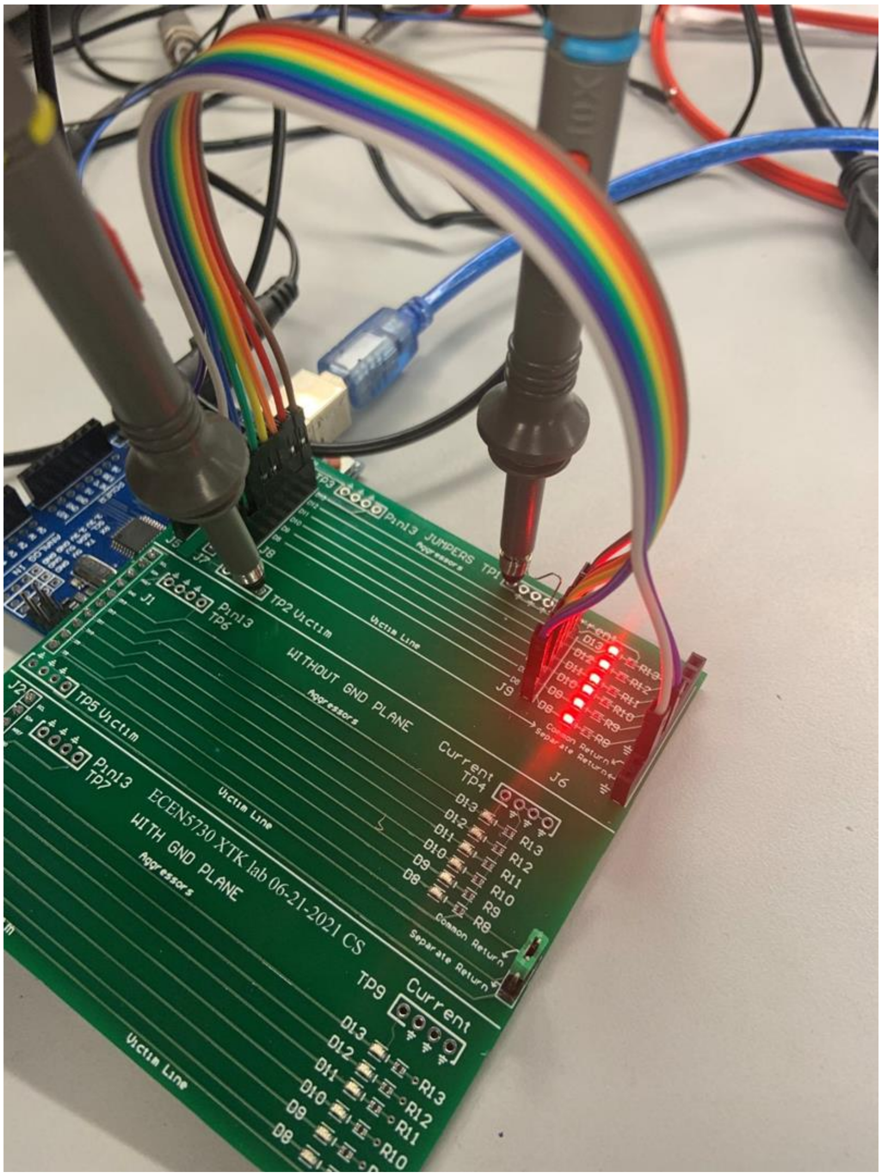
8.10 Part 4: Limitations of the Solderless breadboard

In the final 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.

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.



The ground holes on the far right edge have a special configuration. You should reverse enigneer 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.

8.11 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)*

Grading rubric is the same as always:

2 points if your scope measurements tell the correct story and you analyze and articulate the principles

1 point if you come close

0 points is you do not get checked off, do not complete the lab or do not have a clue.

Chapter 9 Wk 5 Mon: testing the switching noise on our version of your brd 2

This is the beginning of your brd 2 design: demonstrating good and bad layout for switching noise.

This board will demonstrate the two important design principles to reduce switching noise:

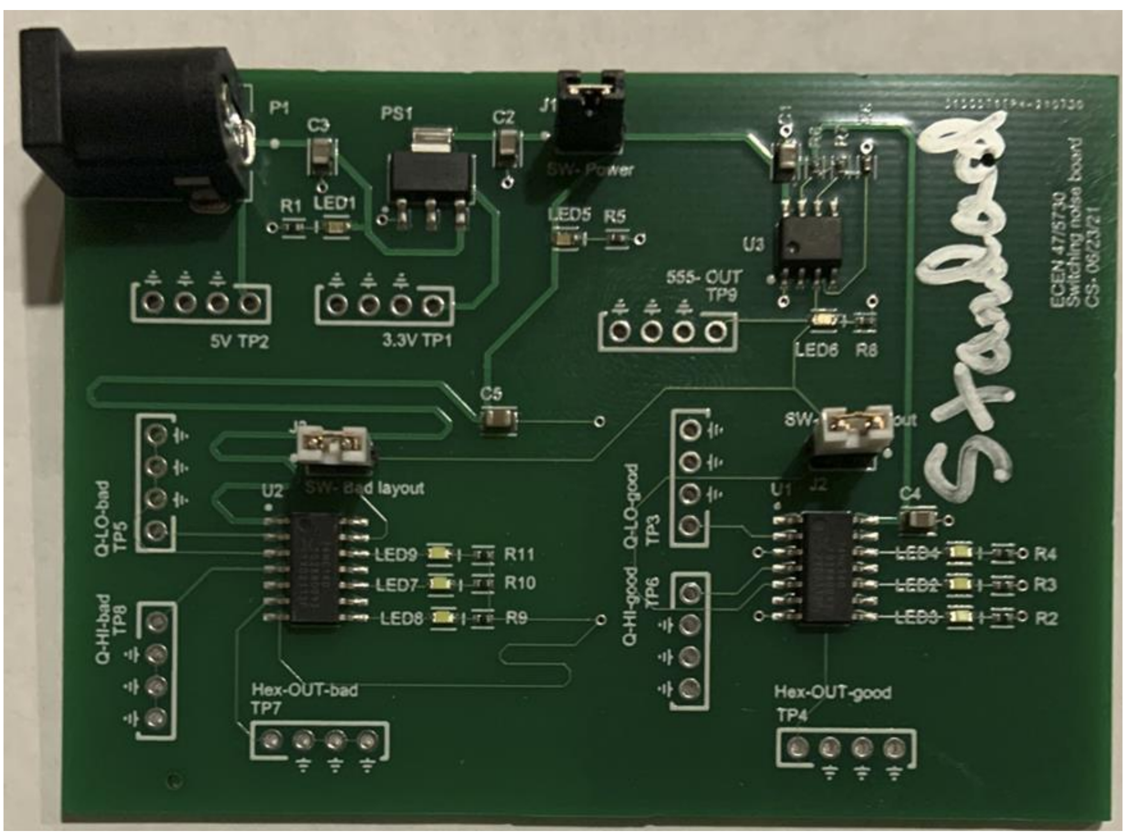
1. *Use a continuous return path under the signal traces*
2. *Use low inductance decoupling capacitors in close proximity to the IC power pins.*

In this board, you will use 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.

Three of the output signals from the hex will drive red LEDs with series resistors to limit the current. These will generate the dI/dt for the signal cross talk and the power rail noise.

For this lab, you will use a version of brd 2 that we 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.

Here is the board you will use for the switch noise measurements for this lab:



9.1 The schedule

In wk 5 (this week) you will get a heads up on the brd 2 design.

In wk 6, you will complete the schematic and layout of board 2. On Thurs Sept 30, we will order all the brd 2 fabs.

In wk 9, your brd 2 boards will come in and you will do the analysis of them.

9.2 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.

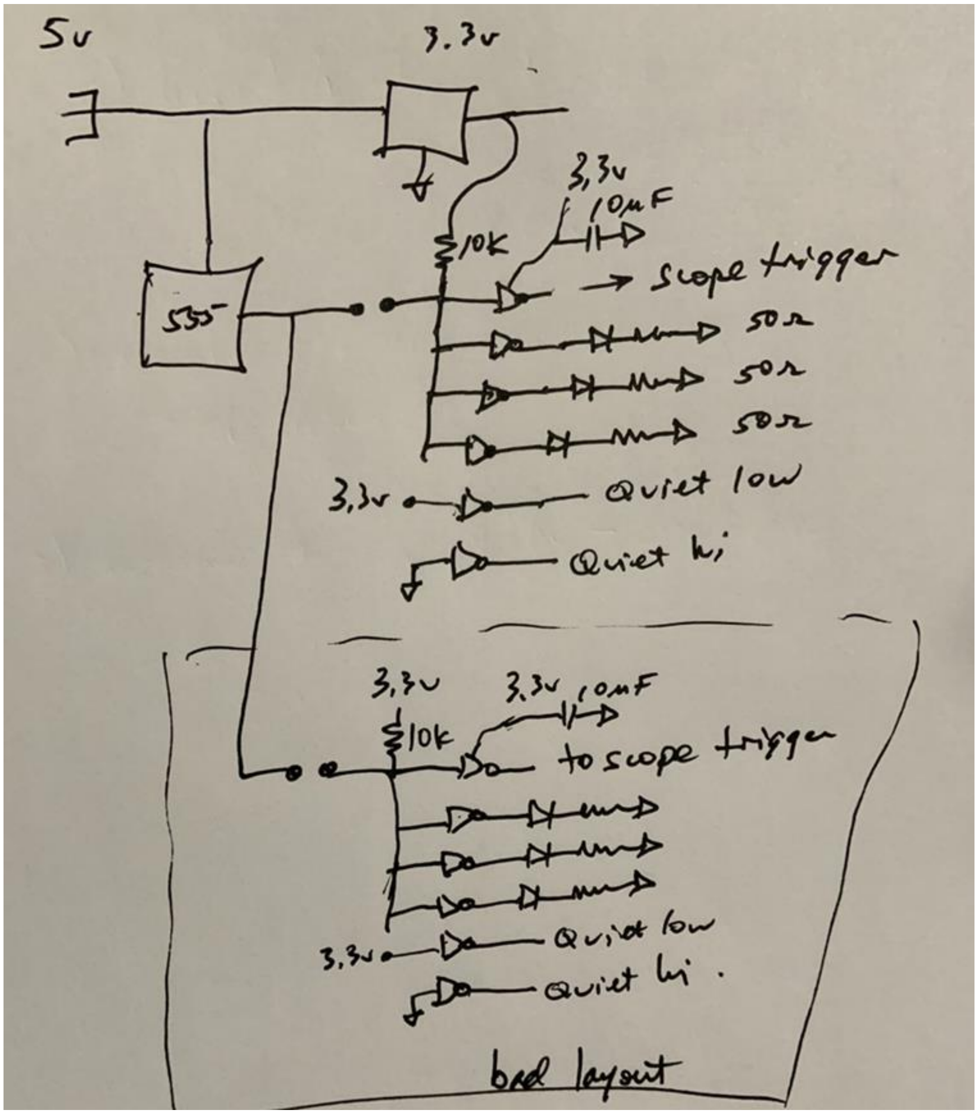
This board will give you an idea of how the board 2 design will be implemented. Review its layout carefully. Note the test points, switches and indictors.

9.3 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 two 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.



In addition to the 555 timer, you will use two new parts in your board 2: a low voltage dropout (LDO) 3.3 V regulator and a 7414 hex inverter.

These two new parts can be found in the newly released integrated libraries JLC_2021_08 release. There is just one LDO option, the AMS1117. It converts the 5 V in to 3.3 V out. You will need a capacitor on its output, not for decoupling, but to filter out the high frequency noise and prevent the LDO from going into oscillation.

There are three different 7414 options in the library. The parts we will use for the brd 2 design are the 74HC14 chips, if they are available. You can open up the specs for these parts. Be sure to use the latest 2021_08 release as this library has the latest footprints for the hex inverters.

You will add a decoupling capacitor to the 555 IC and each of the 7414 ICs. In the case of the good layout circuit, of course, the capacitor should be placed as close as practical to the IC it is decoupling. In the case of the bad layout section, you can place the decoupling capacitor anywhere you like.

In addition to these components, you will also add various test points, indictor LEDs and switches.

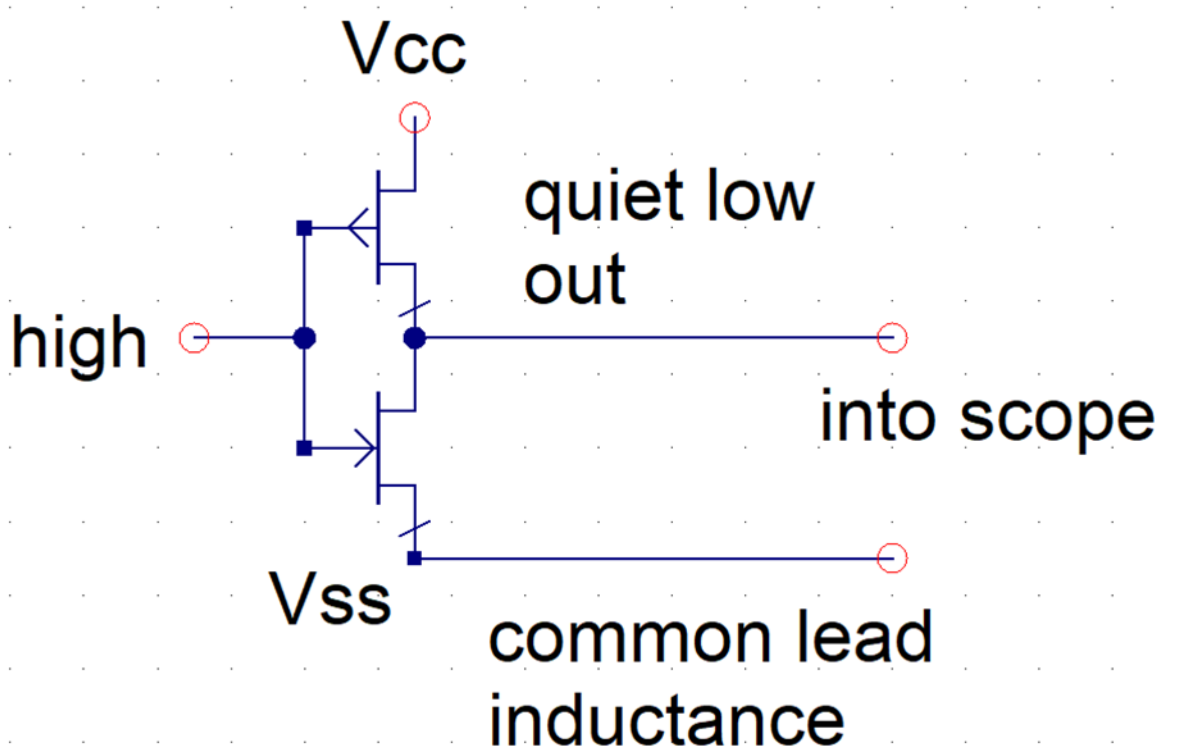
The hex inverter is a collection of 6 independent inverters that share the same power rails in the IC package.

In each circuit, four of the hex inverters act like normal inverters. The output of one inverter is used to trigger the scope.

The output of three of the inverters will drive LEDs with current limiting resistors, so that we have some dI/dt that switches.

Two of the inverters are special.

When the input to one 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.

When we measure the voltage on the output pin on the board, with the 10x probe, we are measuring the voltage of the output pin, relative to the local ground on the board. If there is any ground bounce noise from the local ground where the 10x probe is to the Vss rail on the die, it will appear as a voltage on the probe. 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 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 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*

Use the scope trigger to measure the switching of one of the I/O. Using this to trigger the scope, 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, stary 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 form switching unless you want them to.

On our board, we did not add the 10k pull up resistor. This means when the inputs to the inverters float, we can change their state by waving our hand over the inputs- like magic.

9.4 What you will measure on our board

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. 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?

9.5 In your report

Remember, this report will look good in your portfolio. This is a classic example of how layout affects switching noise. If you write this up well, you will impress any hiring manager with how much you understand about switching noise.

Show a scope trace comparison of the I/O signal that is switching and the synchronous noise on the quiet high and quiet low of the good layout and bad layout. Try to use the same scales so you can do a direct comparison of the magnitude of the noise.

Comment on what features in the layout account for this difference.

Be sure to have your TA check off your measurements and your explanation. Be prepared to show an example of your measurements and the features on this board that contributes to more, or less switching noise.

Remember. The two circuits are identical, yet, they have a very large difference in the amount of noise, just because of their layout.

9.6 Grading rubric

2 points if you show the difference in switching noise between the good layout and bad layout and include an explanation of the layout features that contribute to the noise differences.

1 point if you only look at one section or do not have a clue what you are doing

0 points if you do not submit a report or do not get checked off by your TA.

Chapter 10 Wk 5 Wed: bring up and test of brd 1

You should receive your brd 1 example in the lab on Wed. Review your board, take pictures, perform measurements and write up your report. This counts as your midterm. It is due on Monday Sept 27, 9 am.

10.1 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 V AC 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 indictor LEDs and switches.*

10.2 Measure key signals

You should measure at least the following signals:

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%.*

10.3 Final report for brd 1 and grading rubric:

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

2 points for:

- *1 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)*

2 points for:

- *An analysis of the performance of your board compared with what you expected (the engineering specs). Include scope measurements as appropriate.*

1 point for:

Your evaluation of what went right and what went wrong in your board design process. What would you do differently next time if you could do this board over? Were there any hard errors you had to fix? Any soft errors? What would you do differently to make your board easier to test?

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.

Chapter 11 Wk 6 Monday and Wednesday: Design Brd 2

This is the beginning of the design process for brd 2. You got a taste of the board design for brd 2 in the measurement of switching noise on our version of brd 2. Now you get to design your own version.

11.1 Brd 2 POR

In this second 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*
2. *Create a clock signal of about 500 Hz and about 50% duty cycle*
3. *Drive four of the inputs of two hex inverters used to demonstrate good layout and bad layout*
4. *Use red LEDs and 50 ohm resistors as the load to three of the switching outputs of each hex inverter*
5. *Estimate the current you expect to draw from the inverter for the LED and 50 ohm resistor load.*
6. *Stretch goal: if you measure the voltage on the output of an inverter with the LED and resistor compared to the inverter with no load into the scope, you can extract the Thevenin resistance of the inverter output.*
7. *Connect the output of the fourth switching inverter to a test point to act as a trigger for the scope*
8. *Set up one output of each hex inverter as a quiet HIGH and one output as a quiet LOW*
9. *Demonstrate indictor LEDs, test points and circuit isolation switches as appropriate*
10. *Engineer the layout on one side of the board with best design practices and the other side of the board with bad layout practices.*
11. *Keep the part placement and routing identical for the two regions of the board, except for the location of the decoupling capacitor.*
12. *Plan test points for at least:*
    a. *The scope trigger output*
    b. *The quiet high*
    c. *The quiet low*
    d. *The 555 output signal*
    e. *The 3.3. V rail on the board*
    f. *The 5 V rail on the board*
13. *Use best scope measurement practices when you test your board*

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

11.2 The schedule for Design Assignment Brd 2

We will complete the design steps in these phases:

Wk 6 : Monday

- Create POR
- Create schematic, starting from your brd 1 design.
- In-class lab on Monday will be schematic design reviews.

Wk 6: Wed

- In-class lab on Wed will be to finish your layout and complete a CDR of your layout before release to fab
- Create the design files and sign off by your TA brd design files should be submitted by Wed, Sept 29, 9 pm
- Your brd design files will be ordered on Thurs, Sept 30.

Wk 9: Wed

- *When the boards are back, you will characterize the noise on the good layout section and bad layout section.*
- *The report for brd 2 is your midterm, due Monday Oct 25, at the latest.*

11.3 New components for brd 1

11.3.1 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 and only an LDO would work.

An LDO is generally very easy to us. There is often an enable pin. This should be tied high to the input voltage to enable or turn on the output voltage.

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. It's value is not critical. Any value in the 1 uF to 22 uF range will filter the high frequency noise to prevent oscillation.

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.

If you are interested in a cool experiment, use a 1206 filter capacitor so you can manually take it off your board and then an isolation switch to isolate your 3.3 V LDO from the rest of the circuit. With the capacitor removed, you may see the oscillations at the output of your LDO. With the capacitor in place, there should be no oscillations. The oscillations will be large voltage fluctuations at about 1 MHz.

The LDO we have in the LCSC library is this one: LCSC Part# C6186. You can do a google search and find the datasheet for this part.

11.3.2 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 the required 3.3 V of the power rail? Or should it be driven by the 5 V rail?*
5. *Can it handle the current it needs to source or sink to drive the hex inverter inputs?*
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 that 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.

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

11.3.3 Hex inverter

The Hex inverter, 7414, is a very simple chip. It 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 3.3.V. 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?

There are many versions of the hex inverter. We have two in the LCSC library:

LCSC Part# C5482

LCSC Part# C6065

Which part will you use in your board?

You can use the same criteria as for the 555 to make a decision.

For our application to show off the noise, we want the fastest switching part so that we can see the most noise. The switching noise varies with dI/dt. Of course, if you wanted lower switching noise and the rise time of the signal was not important, then you would want the slowest part.

Wouldn't it be great if the datasheet explicitly offered the rise and fall times? If it doesn't list this important parameter, and you wanted the fastest part, how will you decide which part to use?

Just for your reference, the letters, LVC, or HC, or AHC in the part name refer to the IC technology for which the parts were built. This also affects the switching rise or fall time. Generally, an LVC part is faster than an AHC part, which is faster than an HC part.

Do you want to run your hex inverters at 5 V or at 3.3 V? If you are not sure, you can add a 3-way switch to select the 5 V rail or the 3.3 .V rail for each inverter chip. This is one way of selectively turning the chip off or on and isolating it from the rest of the circuit. This approach of providing an option to the voltage rail of a device is sometimes an important feature to test in the evaluation or prototype phase of a circuit design.

Whichever part you decide to use, it is very important that you:

- ✓ *Use the same part for BOTH the good and the bad layout circuits.*
- ✓ *Use the same power connections for both parts*
- ✓ *Make note of the part you are using so you can include it in your POR and your final report.*
- ✓ *Maybe find another classmate using a different part from you so you can compare notes after you get your boards back.*

11.3.4 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 an 0603 part size. Each color will have a slightly different forward voltage drop.

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.

For the other indicators, just be sure we can drive them with the 3.3 V 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? From your practice board, you will see that even 3 mA, as when you used a 1k resistor, was plenty bright.

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 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 reference 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.

11.3.5 Test points

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

11.3.6 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 JLC library. We will let JLC do the assembly for us. You will just need to add the shorting flag as needed.

You can sue 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.

11.4 Design Assignment, brd 2, wk 5 Monday, before you come to the lab: create POR

Write up a brief summary outline of you POR. The POR is a living document. It may change as you learn more about the design. It will also be part of your final report.

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.

Be sure to include:

1. *Rough schematic or block diagram*
2. *List of the significant parts*
3. *Find datasheets for them*
4. *Definition of what it means to "work"*
5. *The schedule*
6. *The test, characterization plan*
7. *The power budget and how you will supply power*
8. *Potential risk sites and how you might avoid them*

11.5 Design Assignment brd 2, wk 6 Monday: complete schematic

You can start with your brd 1 schematic. In addition, use the latest released libraries, the 2021-08 versions for all additional parts.

Try to use only basic parts if possible. This will keep the cost down. If you need extended parts, just justify your reasoning.

Use the 10x probe test points.

Plan to present your design in the in-class design review at the end of the lab period and to review others' designs.

11.5.1 Design reviews

In class on Monday and Wednesday, we will have peer design reviews both as a class and individually in pairs. In a design review be sure to always be respectful.

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.

11.6 Design Assignment brd 2 wk 6 Wed: complete the layout

Use any board size you want, but keep its dimensions below 3.9 in x 3.9 in. Do not exceed this dimension. What happens if the board is larger?

Use a 2-layer board.

In the region of the board with the good layout, keep the decoupling capacitors close to the Vcc pin of the hex and use a continuous return plane under the routing for the traces. This reduces the switching noise.

In the region of the board with the bad layout, do not include a ground plane. Move the decoupling capacitors far away from the Vcc pin of the hex.

Otherwise, the circuits should be identical.

Set up the design constraints as 6 mil signal traces, 20 mil power traces, vias with 13 mil drill hole, 25 mil capture pad.

Complete the part placement and layout.

Add all the important silk screen indicators, especially your name and the brd name.

Before you leave the lab on Wed, be sure to have your design checked off by your TA. If it is not complete, at least have what you have completed reviewed.

During the wk 6 Wed in-class lab, we will do a peer design review of each individual's layout.

Before 9 pm on Wed, Sept 29, you should complete the design, create the three design files and submit them to your TA to place the orders.

To ensure the highest chance of success and getting your fabricated board back on time, complete a design review of your board design files by going thru the submission process on the JLC web site, but DO NOT PLACE THE ORDER.

11.6.1 Grading rubric for the wk 6 lab assignment:

2 points if you submit your design files and the order is accepted

1 point if you submit your design files but there is an error and your board order does not get placed on time.

0 points if your board order is not submitted

Chapter 12 Wk 7 Monday measure trace resistance and blow traces up

This lab is really two parts. In the first part, you will measure the resistance of narrow traces on a circuit board. In the second part, 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.

12.1 Before you come to the lab

You should have read chapters 5 and 6 in the textbook.

12.2 What you will do in part 1

In part 1 of 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*

In part 2, you will estimate the maximum current for a 6 mil and 20 mil wide trace, using the calculator in the Saturn PCB tool. Then you will apply enough current to make the traces warm, hot and then smoking.

12.3 The Test Board

The test board we will use is a simple 2-layer board with similar patterns on the top and bottom layers. One like it is shown below.

The connections are not obvious, but you should be able to figure it out 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.

Before you begin to use this board, use a DMM on ohms setting and figure out which connections connect between the pads and the traces.

Here is a hint, on the top layer, there is a long, continuous bar. The connections to the bar are by holes on opposite edges. The other end of the trace is connected to specific pads.

There will be two holes on each side of the trace to enable 4-wire measurements.

Watch this video and I walk you through figuring out the connectivity of this test board.

12.4 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 it 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.

Create a table to record your estimate and the 2-wire and 4-wire resistance measurements. Add your estimates to your table.

You must complete this table 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.

12.5 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.

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.

12.6 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 and then 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. 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 probes 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 t 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.

Flesh 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*

12.7 Part 2: Blow some traces up

You should still be wearing the safety glasses you had on for part 1.

The purpose of this lab is to measure the maximum DC current you can put through a trace before it gets noticeably warm and then very hot, and then smokes. The values you measure should be compared with the estimates based on the IPC specifications.

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.

We can use the power supply to force a current through a trace and a voltmeter to measure the voltage across the trace. This will enable us to measure the resistance of the trace using the 4-wire method.

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 latest released version of 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?

12.8 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?*

Turn in to canvas a pdf copy of your spreadsheet of the estimated, and measured resistance using the 2-wire and 4-wire methods and your analysis of your observations. Answer the so what questions.

Include the estimated maximum current through a trace based on the IPC recommendation and your observation of the current for barely warm, hot and smoking for the 6 mil wide trace and the 20 mil wide trace.

Add your recommendation for the max current you would feel safe putting through each trace.

Again, keep in mind this report will look great on your portfolio.

Chapter 13 Wk 7 Wed Soldering and Assembly practice board

13.1 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.

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.

13.2 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*

Check out a soldering kit from you TA. Your kit will have a check list in it. Verify all the parts on the list are in your kit. If they are not, let your TA know and they will replace the parts.

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

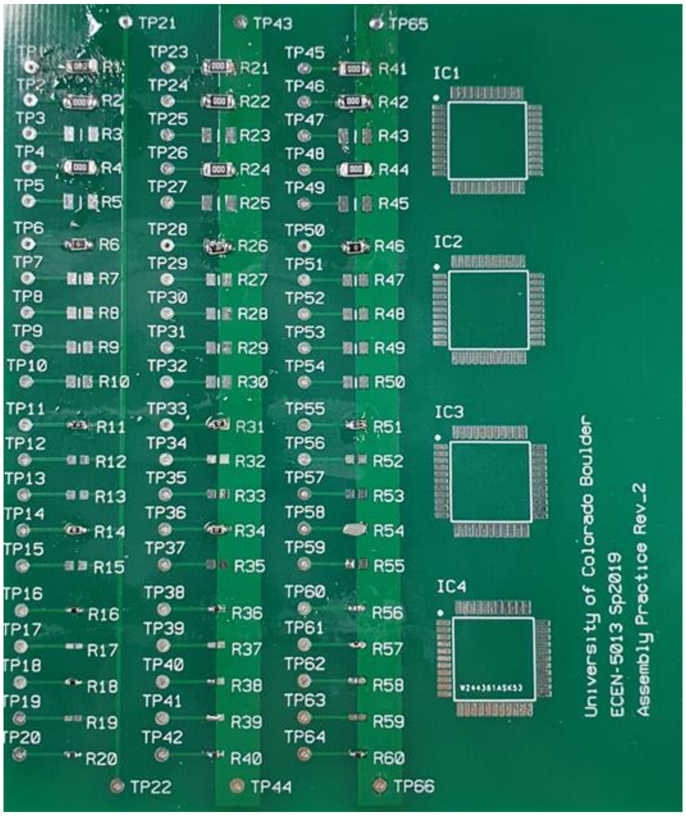
When you turn in your soldering kit, it should be complete.

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.

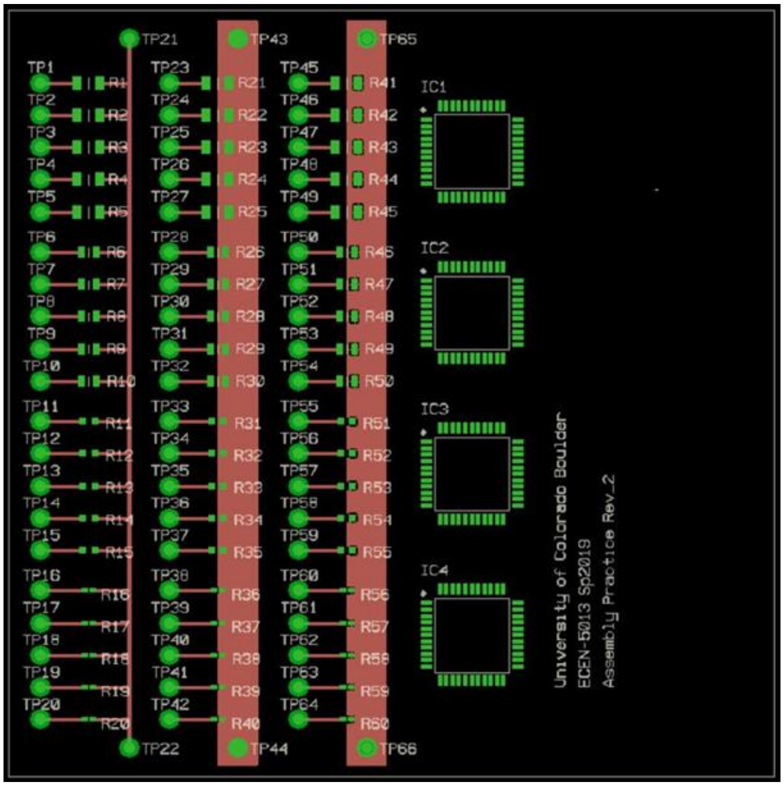
13.3 What you will do

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 types 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 bused test pad, labeled TP21 or to TP22.

The middle column of pads, starting with TP23, have their right pads embedded in part of a plane. This copper plane will suck heat from the solder 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.

The third column from the left, starting with TP45, has the same pads in the middle of a plane, but using a special connection to the pads called a thermal relief. Figure 3.1 shows a closeup of pads with a thermal relief.

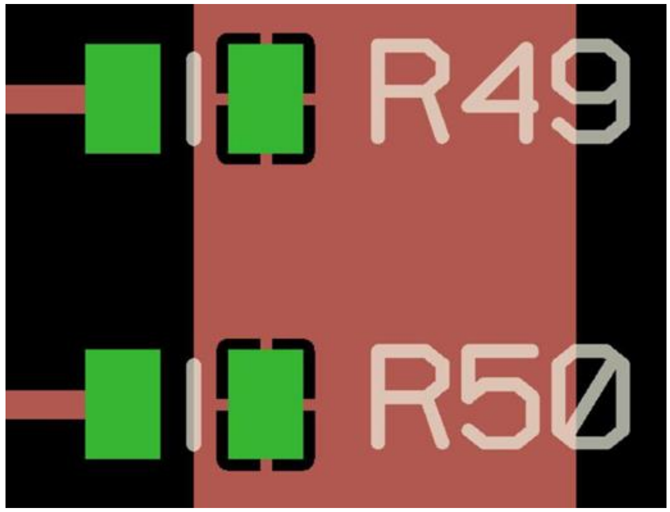


Figure 3.1 The pads in the far right column are in the middle of a plane, but separated from the plane using thermal reliefs.

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 mOhm/4 = 0.12 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.

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.*
12. *Try the 0805 and at least one of the 0603 parts.*
13. *If you are feeling ambitious, try an 0402 part.*
14. *On the 32 lead QFN footprint, practice removing a solder short.*
15. *Add enough flux and solder to the pads so they are shorted together.*
16. *Using the copper wick and solder flux and an extra hot iron, suck up the extra solder to clean the pads.*
17. *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.*

When you practice soldering parts, the goal is not to master putting a few 0402 parts down, but to master soldering 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 an 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.*

Practice tinning your soldering iron. This will be a trick you use over and over again.

You should melt solder wire that contains a resin core, on the tip until the solder flows over the surface of the tip, alternately, cleaning the tip in the brass sponge and applying new solder.

Alternatively, add a pea sized blob of solder paste to a small region of the blue silicone rubber matt. This solder paste is mostly solder flux with small spheres of lead-free solder.

Touch the hot soldering iron tip to the blob of solder paste to coat the tip in solder flux. Roll the tip around. Periodically clean it off on the brass sponge until the tip is coated with solder and the solder flows over the surface.

Note, when the hot soldering iron encounters the solder flux, the flux will vaporize. Do not breathe in the solder flux vapor.

If the tip is kept at 700 degC, you will have to re-tin the tip and clean off the tin oxide every 5 to 10 minutes.

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.

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. You will not assemble parts, but will practice removing solder shorts.

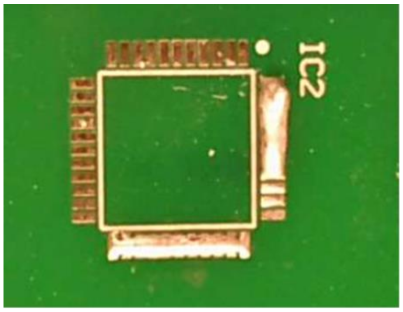
First, create some shorts. Add enough solder to the pads to short out a few adjacent pads.

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.

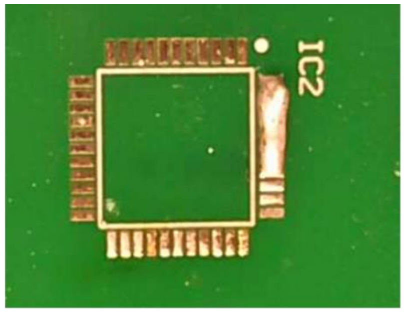
After you have tried your own methods, try this method:

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.*
5. *Add solder flux to the pads.*
6. *Place the 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 attract 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.*

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

13.4 What you will turn in or complete for 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.

Take a picture of your board and turn it in on canvas.

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 soler 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.

Chapter 14 Wk 8 Monday Brd 3: A Golden Arduino PCB

In this week, we start the new board 3 design assignment. This 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.

14.1 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 this week, 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.

In wk 8, you will complete your POR and preliminary BOM. This version of the BOM should include just those parts you are concerned about that are not commodity, common parts.

You will not turn this POR in, but you will use it for each step of the design process and include it in the final report.

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 will use your Arduino board to communicate with your brd 4 which will be a multi-sensor shield*
- *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 boards.*
- *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.*

This is a huge reduction in noise, just from using best design practices!

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?

14.1.1 Schedule

In the ***POR***, you need to be aware of the schedule and the due dates for reaching each specific milestone. Adjust your design and project plan accordingly.

In the POR, be sure to include the items listed in the textbook about the POR. In particular, map the 5-week schedule into your calendar.

Due dates:

Wk 8, POR and preliminary BOM and schematic should be completed

Wk 9, layout is due

Wk 10, CDR of schematic and layout, boards ordered

Wk 13, board 3 comes back, boot load, complete characterization, final due

14.1.2 What you will do in this wk 8 Monday lab

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 your final for this course.

For brd 3, all the parts you will use should already be in the integrated libraries. In board 4, you will learn to create your own symbols and footprints.

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 crystal 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 USB plug*
8. *A reset switch with a debounce capacitor*
9. *A 3.3 V LDO*
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*

Graduate students are also required to add the following features to their Arduino boards:

12. *A ferrite filter on the AVCC pin to the ADC circuit on the 328*
13. *A 0.5 ohm series resistor in the power rail and connectors to it to measure the supply current*
14. *Selectable power from either a 5 V external AC to DC converter or from a USB connector*
15. *Additional rows of ground pins adjacent to and spaced 300 mils center to center from your digital I/O switching pins to reduce cross talk for I/O pins switching.*

Undergraduates can add these features to their board, but they are not required.

14.1.3 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.”

Include in your POR the non-commodity parts you will use. Given the supply chain issues these days, also include price and availability on the JLCpcb LCSC library or on Digikey.

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. *What special features in the parts selection should you consider making your product easier to assemble?*
4. *What special features in the parts selection should you consider to make your product more robust to mechanical stresses?*

5. *What special features in your layout do you want to include to reduce the risk for:*
    a. *Assembly*
    b. *User interface*
    c. *Test and debug*
    d. *Lower cross talk*
    e. *Lower power rail noise*
    f. *Lower cross talk for signals coming off the board through the Arduino pins*
6. *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?*
7. *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.

14.1.4 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*
- *Multiple value decoupling capacitors*

Here are some examples of reference design to evaluate:

https://www.arduino.cc/en/uploads/Main/Arduino_Uno_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

https://circuitmaker.com/Projects/Details/Troy-Reynolds/Arduino-Uno-Rev3-Reference-Design

14.2 Step 2: 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 mini 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 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.

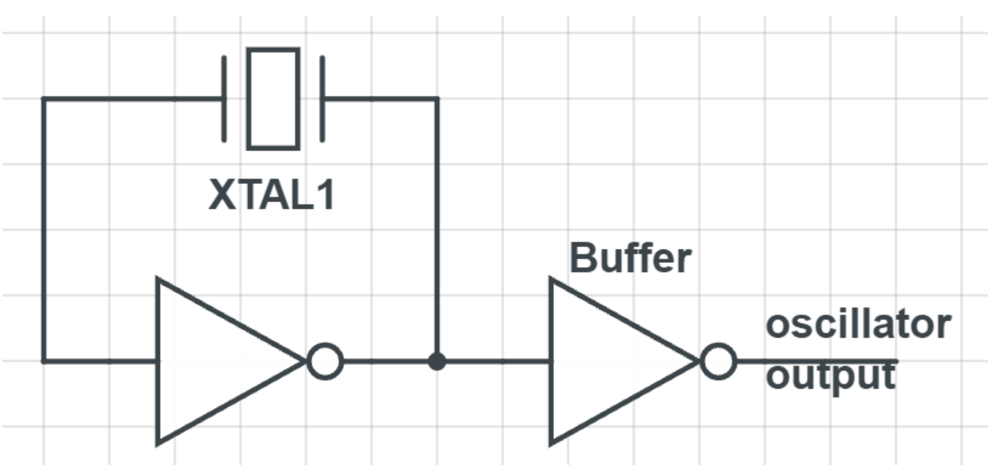
14.2.1 Crystal circuits

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.

In principle, a simple inverter circuit with feedback between its output and input will self-oscillate. The frequency 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:

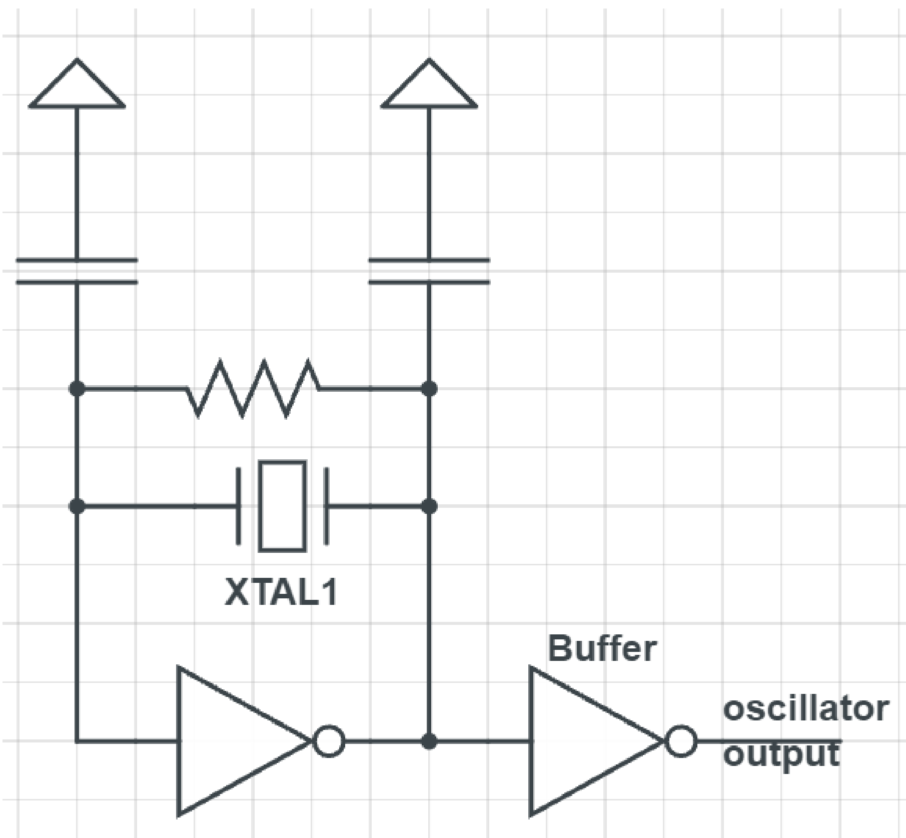


Only one inverter is needed to generate the oscillations. However, it is good practice to add another inverter at the output of the oscillator to buffer the oscillator. This way if the output is loaded down with a low impedance, like 1 k ohms, the buffer will drive the load without impacting the inverter 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.

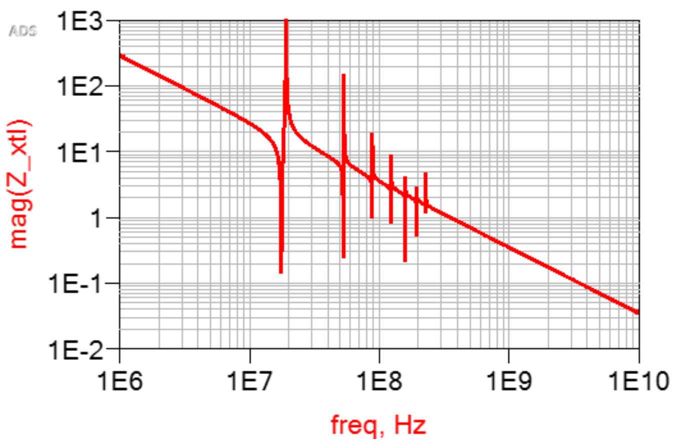
First is a feedback resistor to start the circuit into oscillation. This can be in the range of 1 M to 10 k. With the resonators selected for your board, we have found a value of 10 k works best. For many crystals, 1 M works best.

Secondly, 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. 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:



14.2.2 How they work

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. The frequency of the low impedance drives the oscillation of the inverter.

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.

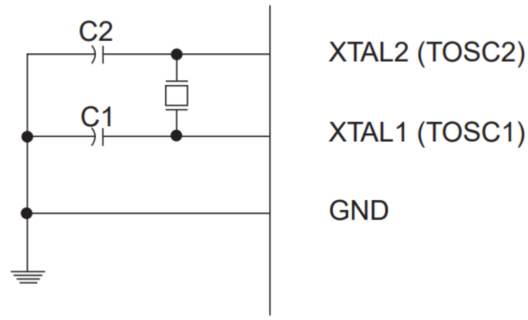
But, the crystal can oscillate at multiple harmonics. In order to use a quartz crystal in a clock generation circuit, we need to implement two important features:

1. *The inverter sometimes needs to be kick started by adding some feedback between the input and the output. This way, the initial transitions of the inverter will drive high frequency components, some of which will overlap the resonant frequency of the crystal. The frequency component with the largest voltage component and the strongest feedback, i.e., the lowest impedance, will be the oscillation frequency. Using a high resistance resistor across the input to output of the inverter will "**kick start**" the inverter into oscillation.*

2. *We need to suppress the higher frequency components of the crystal so that only the first harmonic, the lowest frequency, will drive the resonance. We do this by adding small filter capacitors on either side of the inverter to filter out the higher frequency components at which the crystal might oscillate.*

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 10k resistor to jump start the oscillations is optional. In our lab experiments, sometimes the resonator self oscillates with no problem when connected to the 328. Sometimes it does not. This is one of those cases where you will want the option of adding the 10k resistor. You can decide if you want to take the low-risk path and add it to your board, or just place the pads for it so you can add it to your board if you do not get oscillation without it.

If you plan to add it, you may want to use a 1206 part so you can manual assemble it to the board.

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.

In wk 8 Wed, you will be building a solderless breadboard version of an oscillator circuit.

14.2.3 Debounce circuits and switches

There is a reset pin on the 328, PC6. The reset pin is normally held high. This means the reset is off. When the reset pin is momentarily pulled low, the 328 resets the sketch it is running and starts from the beginning.

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. The DTR pin on the CH340g chip will pull the reset pin low when it receives an incoming message from the USB port and has code to transmit to the 328.

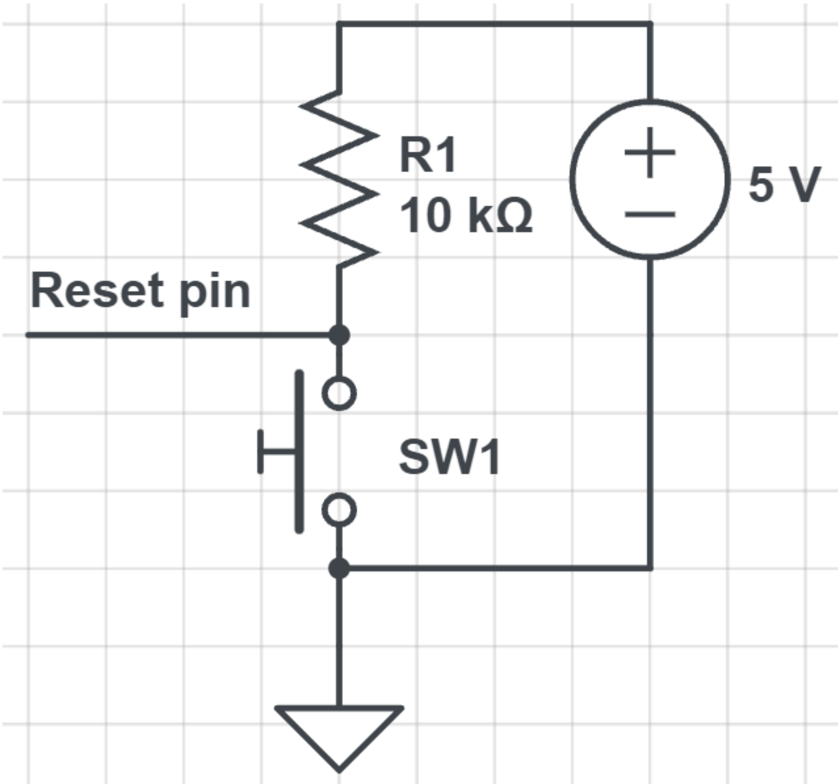
We need to set up the reset pin on the 328 to accept a pull down signal either from the manual switch or the CH340g DTR pin pulling it low. Here is how we will do it.

A mechanical switch we commonly use as a reset is this one:

https://jlcpcb.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.



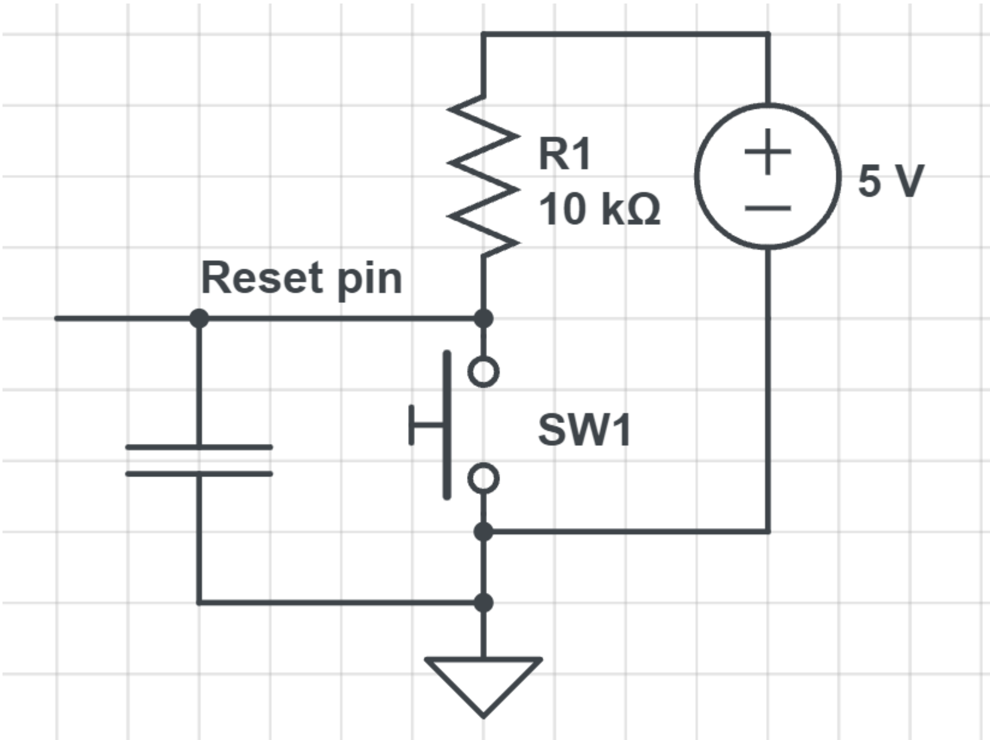
There is one problem with this circuit. Many mechanical switches, after they make contact, "bounce" up and down a few times before they finally stay closed. This is called bouncing. As a little background, check out this article I wrote about bouncing circuits.

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 resent pin is that a pin has to be brought low for > 1 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 rest 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.



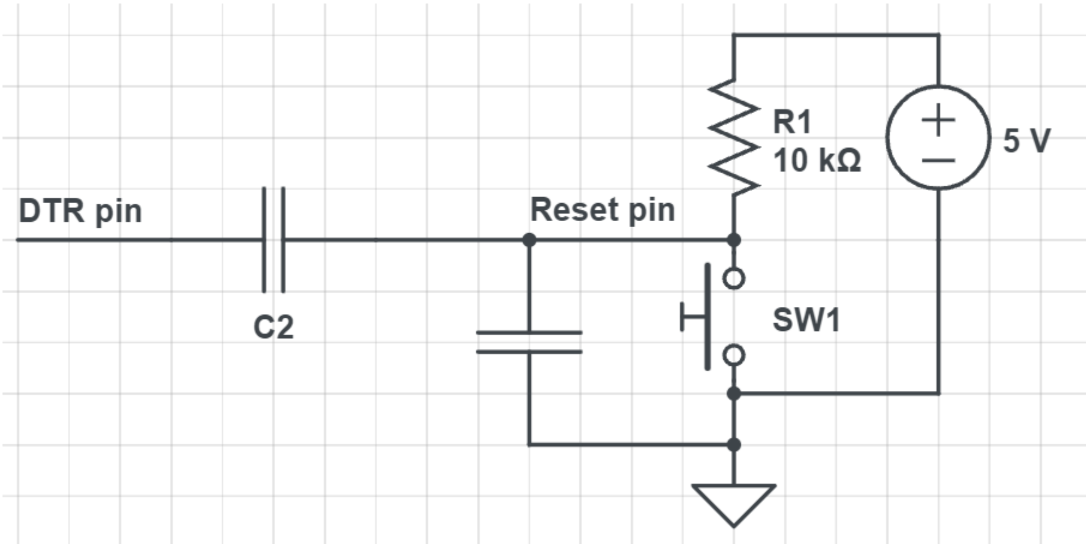
In this circuit, the resistor pulls the reset line high normally. The capacitor is charged to 5 V as well. When the switch is closed, the reset pin is pulled low and the capacitor is discharged to 0 V as well. When the switch bounces up the first time, the capacitor holds the reset pin low. It has an RC charging time, back up to 5 V, of R x 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 uF and R = 10 k, 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 uF 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.

For the DTR pin, we need another circuit. 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. 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 coupling capacitor and the debounce capacitor. If C2 = Cdebounce, the voltage on the reset pin will only be ½ 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 capcaitor. It this is 1 uF, then the C2 should be at least 22 uF.

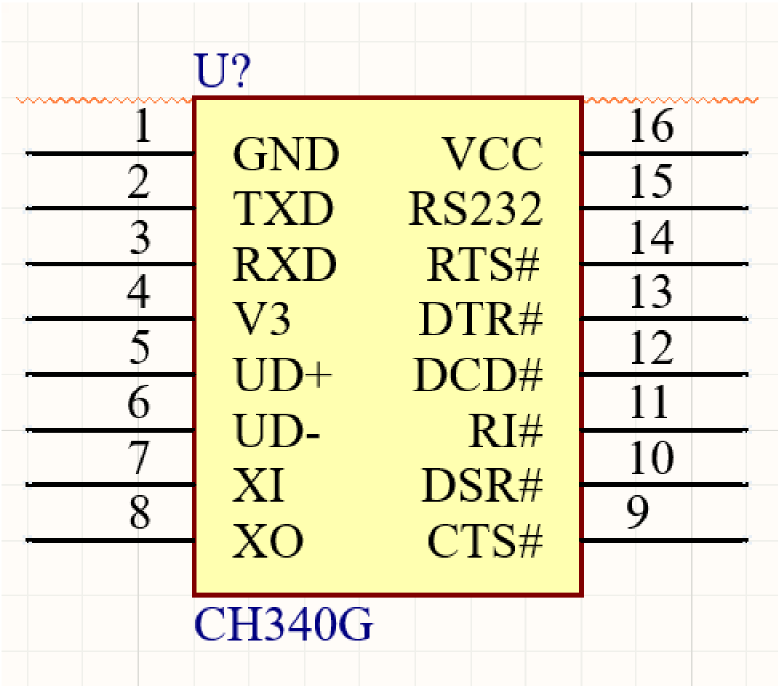
If the DTR pin has a 3.3 V signal range, then when it goes low, the reset pin will only go as low as 5 V – 3.3 V = 1.7 v. This is not low enough to pull the reset pin low. We need to pull the reset pin below 0.8 V.

Make sure the voltage suppling 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.

14.2.4 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.

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. 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.

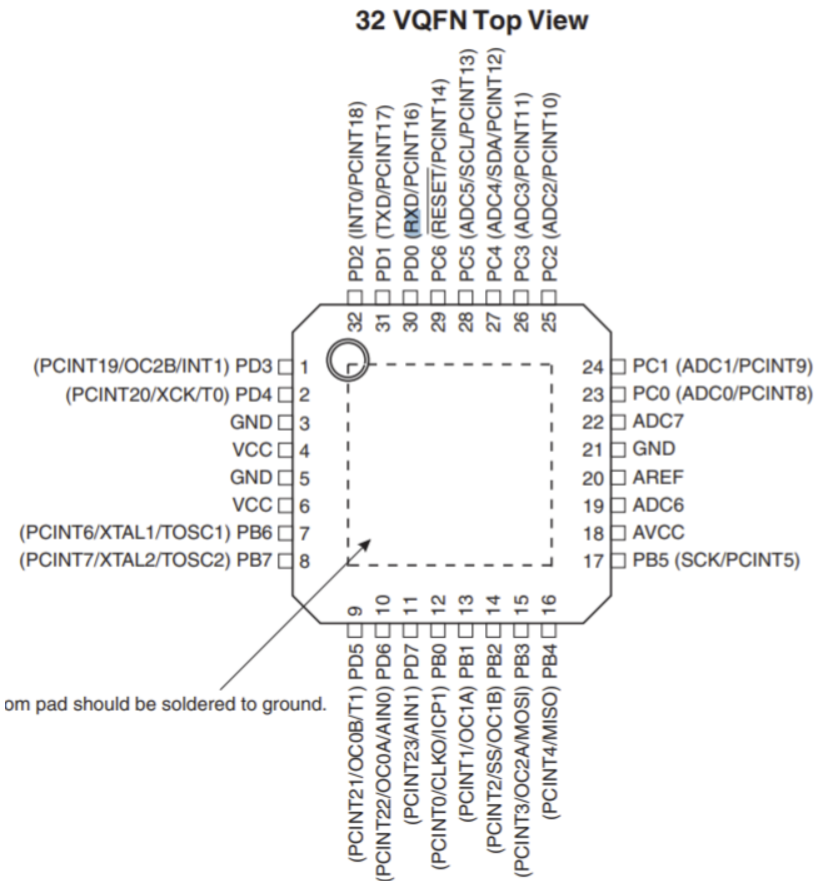
14.2.5 The ATmega 328 microprocessor

A commonly used 328 uC is this one:

https://www.digikey.com/product-detail/en/microchip-technology/ATMEGA328P-ANR/ATMEGA328P-ANRCT-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 pin out is in the datasheet and shown here:



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 series resistor and LED. What will this circuit look like?

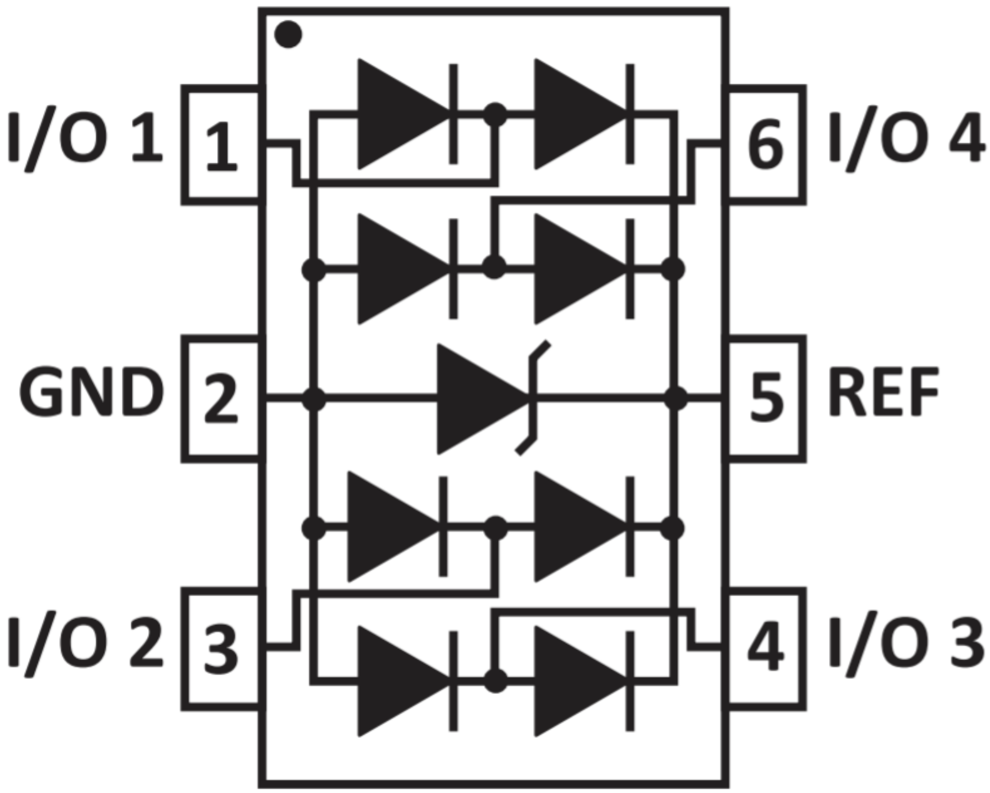
Use a separate connection to the AVCC and VCC pins. They each get their own decoupling capacitor. Ideally, you should add an LC filter between the power rail and the AVCC pin to keep noise on the Vcc rail from getting onto the AVCC rail. This is with a ferrite inductor.

14.2.6 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://datasheet.lcsc.com/szlcsc/ProTek-Devices-SRV05-4-P-T7_C85364.pdf

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.

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 exceeds +6 V, the Zener will turn on and reduce the voltage. If the voltage is larger than -0.6 V, the Zener will turn on in the forward direction. This Zener limits the voltage range that can appear on the 5 V rail and ground.

Trace the circuit connection between pin 1 and pin 6. This is a trick question. There is not 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- datalines?

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 datalines- there is no connection through the TVS chip.

14.2.7 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? What pole frequency will you be able to achieve with this ferrite and a large decoupling capacitor?

What other questions do you have about the components and the circuits you will build?

Chapter 15 Wk 8 Wed: complete the schematic for the Golden Arduino board

In this week, you will complete the schematic design of the Golden Arduino Board.

Note, you will be using the 2021-08_ integrated libraries for your schematic.

The ***POR*** has been completed. It is a living document. As you learn more about the project, you should periodically update the POR. Most important, you should have identified the important risk sites, and mitigation strategies. Part of the POR is the schedule, budget and power requirements.

The initial ***BOM*** was completed, identifying the most important, non-commodity parts. At the same time, you should have reviewed their datasheets.

This week, you complete the schematic. This should include:

- *All non-commodity parts*
- *All other parts*
- *Their connectivity*
- *Including, the power distribution path, test points, LED indicators, isolation jumpers*

In the next week, you will turn the schematic into the layout.

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 other designs you WILL NOT use and layout principles you SHOULD NOT follow. Take responsibility for your own design.

Normally, in your designs, you would

1. *Find the parts you want to use on digikey*
2. *Download the CAD models- schematic symbol and layout footprint, into a library package project in Altium*
3. *Compile the library package as required, into an integrated library, which will be automatically installed into your board project.*

However, in this project, you will only use parts in your schematic that the assembler we will use, JLCpcb, has in their library. To start your design, use the latest integrate library package. (Note, this is a good starting place for other future designs that might use these parts.)

Some parts in this library are identified as having an LCSC part number. This means they are in the JLCpcb library.

All the parts in the JLC integrated library are available from JLC and can be assembled by them. Choose these parts if at all possible. You should only need to grab the 10x probe test pads from the manual library.

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. We are only allowed 10 extended parts in any board, but we can have many more basic parts.

Since JLCPCB will be doing the assembly, it is really ok to use an 0402 part, especially if it is available as a basic part.

If you need a 2 uF part, but we only have a 1 uF part and it is a basic type, figure out a way to use the 1 uF part if at all possible.

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.

Create a pdf copy of your schematic and post it on the canvas drop box. This is due on the Monday at the end of the week.

By the end of wk 8, your schematic should be complete

In week 9, you will experiment with a solderless breadboard circuit to test out:

- *A crystal oscillator*
- *A ferrite filter*
- *A reset circuit*

Then you will complete the layout.

Chapter 16 Wk 9 Monday: layout of brd 3

In this week, we complete the layout of board 3 and the CDR and release the design files to the fabricator. Once again, we will use JLCpcb as a “turnkey” manufacturer to do the board fab, acquire or kit the parts, and perform the assembly.

16.1 Step 4: Complete 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.

Prepare your board to accept the parts from the schematic.

16.2 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.*

16.3 The order for placement

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*
- *Place the crystal and its components in close proximity to the IC pins.*
- *Place switches and indictor 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 close to a soldered pad. Otherwise, do not you 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.*

16.4 Layout tips: design for connectivity, signal integrity, assembly, test and bring up

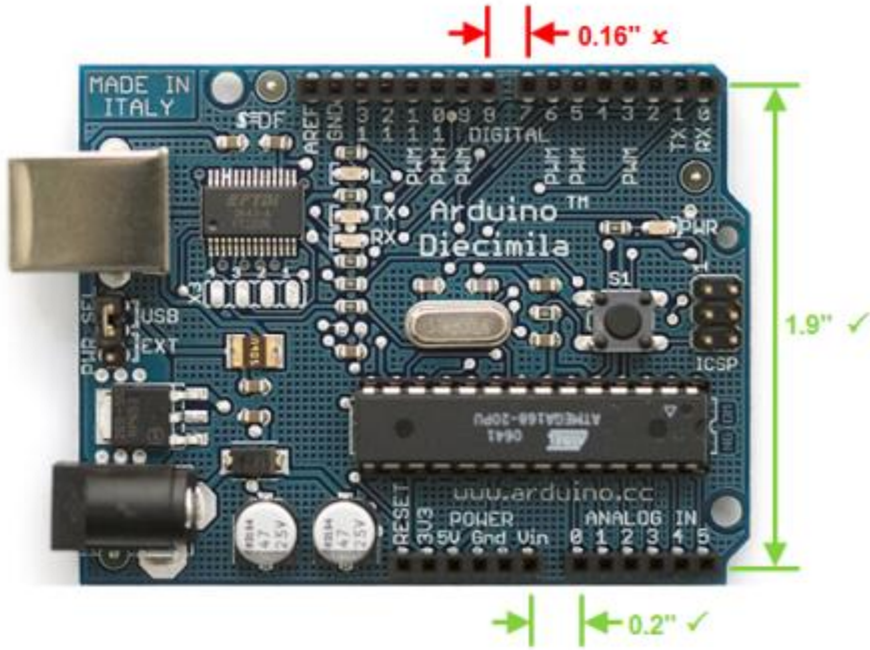
For this design, we will use the same design rules for any low-cost board:

- *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.*
- *All components have their designator, component value, and orientation marking in the silk screen, in close proximity of the component.*
- *Place all LED indictors in close proximity to what they are indicating with a polarity indicator.*
- *Add brief label 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 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.*

16.5 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.

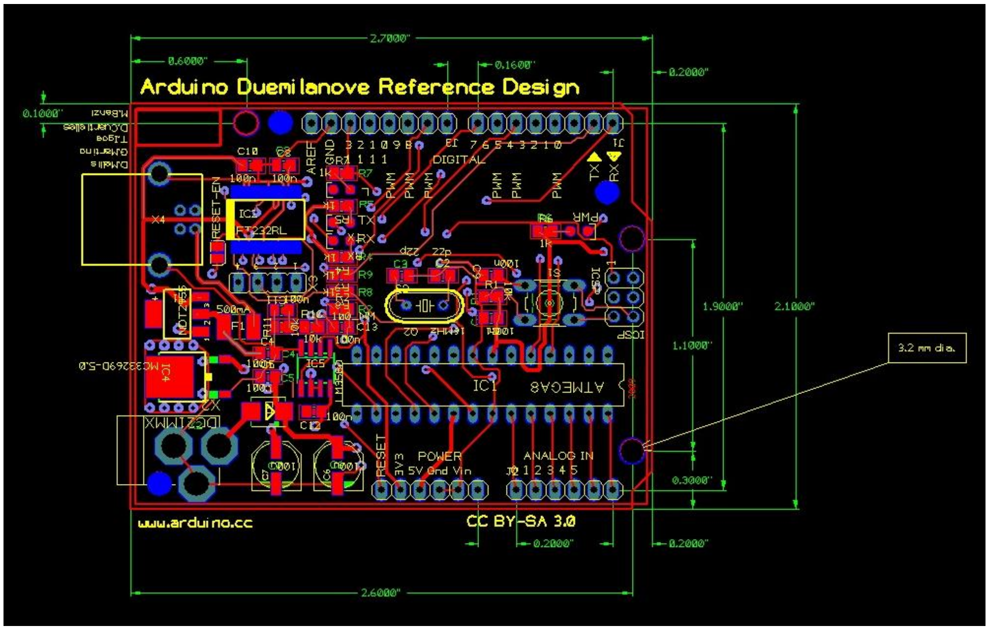
The footprint you should use for the header pins is shown below, from http://brettbeauregard.com/blog/2009/07/arduino-offset-header/. Your board outline does not have to match this board outline.



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.

16.6 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.

- *For graduate student designs: Add a row of ground pins on the outside of each pin header as adjacent return connection for reduced switching noise*
- *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 ½ 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.*

Your layout should be complete by the end of the lab on Wk 9, Monday.

Chapter 17 Wk 9, Wed: Perform a CDR

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 make will have more than a 2 week feedback cycle. This delay is way too long for a successful board.

Catch all of your potential problems BEFORE you submit your final design files to be ordered.

17.1 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.

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.

17.2 In class CDR

In the lab on Wed, wk 9, 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.

17.3 Peer design review

After the in-class design review, break up into pairs and review each other's layout as a final check.

17.4 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 gerbv or the online gerber viewer on the PCBway web site: https://www.pcbway.com/project/OnlineGerberViewer.html

17.5 Submit your board to JLCpcb to use their DFM

In order to have your boards assembled, we will need the pick and place file and the BOM.

When you export your BOM in .xls format, make sure the LCSC part number is one of the columns selected to display in the BOM. Make sure there is one part only per row in the spreadsheet. Only then should you export the BOM.

When we submit your design, JLCpcb will read the BOM and add the LCSC parts to your board.

If their server cannot find the LCSC part, we will have to find a substitute. We may find the incorrect part. Your board may not work. This would not be a good thing.

After your design is complete and you have verified the gerber files, goto www.JLCpcb.com and submit your board file, with your last names and brd2 in the file name, as though you were placing the order. But, DO NOT PLACE THE ORDER.

Select 5 boards as quantity.

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-$4.

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 $4, check your design and your options. If the price is > $4 you should have a very good reason for this.

Submit your Gerber files, then select the assembly option at the bottom of the page. Upload your BOM and pick and place files.

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*

Submit these three files by 9 pm on Wed of wk 9 on canvas. Orders will be placed at 9 am on Thurs.

Note: we are using JLCpcb as a turnkey manufactures. They should assembly ALL the parts. In the event a part is not in stock and we cannot find a substitute, we will assemble the part in the lab from the stock I have.

17.6 Brd 3 final grading rubric

Remember, the final report for brd 3 counts as your final for the class. It counts 20% of your grade. Here is a heads up how we will grade your board design in the final report:

10 points possible for your schematic and layout design. 1 point off for each of the following:

1. *copper fill on top layer*
2. *no ground plane on bottom*
3. *long gaps in the ground plane without ground straps over gaps*
4. *high inductance routing to decoupling caps*
5. *no name on your board*
6. *lacking labels for pins*
7. *lacking labeling for parts*
8. *incorrect footprint for 6, 8, 10 pin header sockets or their location*
9. *late board*
10. *no indicator LEDs*
11. *no isolation jumpers*
12. *missing key parts*

13. *no way of separating the 5 V and USB power*

10 points for the bootload and bring up and test. 1 point off for each of the following:

1. *Power is not distributed to your parts*
2. *TVS is connected incorrectly*
3. *Board will not bootload*
4. *Header pins will not accept a shield*
5. *USB connector will not plug in*
6. *No test points for power*
7. *No test points for any signals*
8. *Board won't run a modified fast blink*
9. *Scope trace sees a clean edge on pin 13 when it switches*
10. *Board is rejected by vendor and not manufactured*

5 points for final report, 1 point off if it does not include the following

1. *A few details about each of the seven steps*
2. *Pictures of your schematic, layout, bare board, assembled board*
3. *Scope trace showing an actual signal from your board and its description and analysis*
4. *A list of what went right and what went wrong. If you could do this design again, what would you do differently*

Of course, your final report on brd 3 is not due until after your board comes back from fab.

Chapter 18 Wk 10 Mon: bring up and test of brd 2

There are two last steps in completing brd 2: bring up, test and characterization and then writing up your final report.

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

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.

If your board is missing the hex inverters, you can pick up a few ICs that you can hand solder to your board.

Make sure you note the orientation of pin 1.

Regardless of whether you put the hex inverter on your board, or it came assembled, make note of which hex inverter you have on your board, and what its rise time is.

18.1 Inspect your bare board

Before you power on your populate board, inspect the board. 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.

18.2 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, and 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 following criteria.

18.3 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.

18.4 Switching noise and layout

The real 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 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 a 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?

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.

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 and the quiet hi voltage?

18.5 Rail compression- a stretch goal

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. 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 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.

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

18.6 Rubric for your final report

Your final report for brd 2 is a midterm. It is worth 5 points total. Here is the scoring:

1 point off if your board was late or did not show switching signals, as expected

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 have a description of the seven steps for circuit board production and any special considerations for your board design.

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.

Chapter 19 Wk 10 Wed special circuits

In this lab you will build two 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:

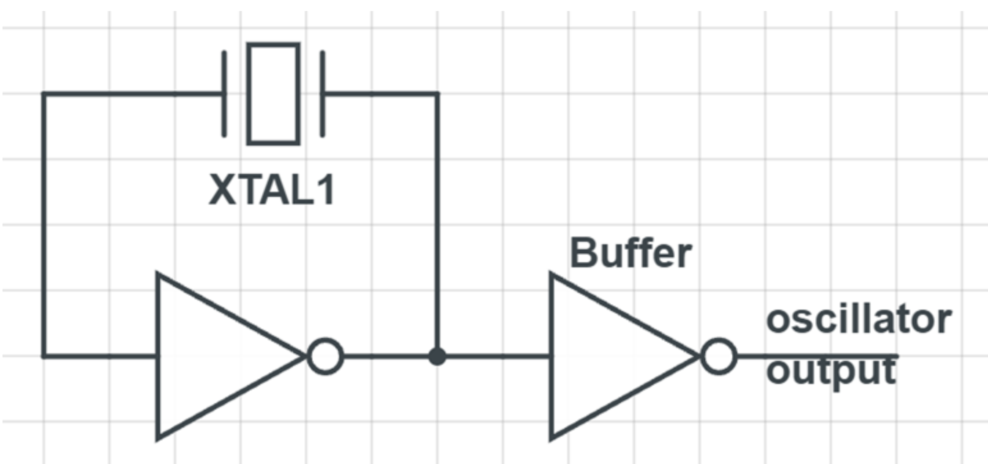
- *A crystal oscillator*
- *A reset circuit*

19.1 Crystal oscillator circuit

You will build a simple inverter circuit using a 7414 hex inverter. In your kit is a 7414 and a crystal. Based on what is written on the outside of the crystal, at which frequency do you expect the circuit to oscillate?

Set up the 7414 with 5 V power from the Arduino, which has a 9 V AC to DC power supply powering it. Use a local decoupling capacitor in close proximity and tie all unused inputs of the hex inverter directly high so their outputs are low.

Connect the crystal across the input and output of one inverter and use a second inverter as a buffer, as shown in the figure below.



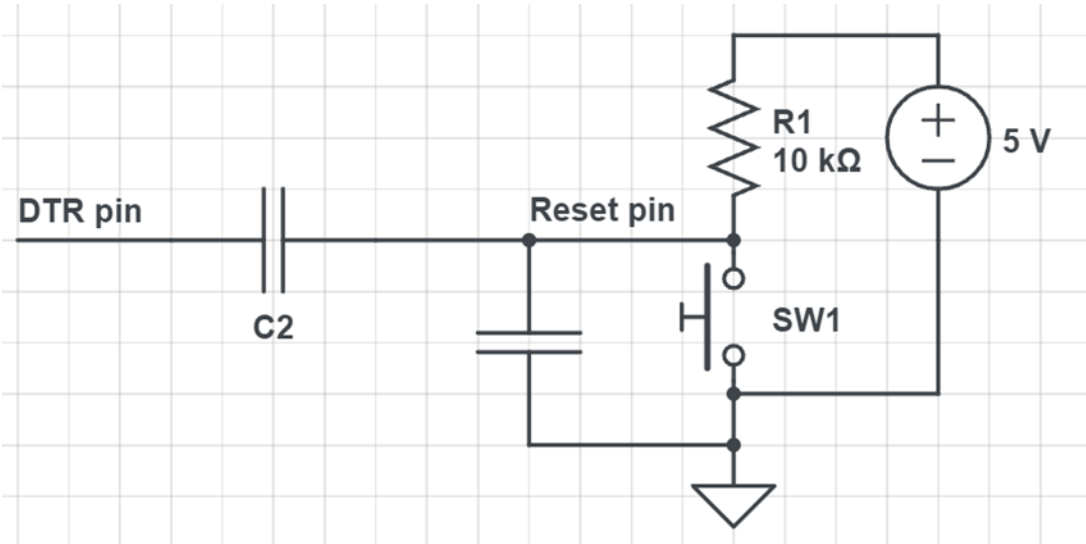
Start with this simple circuit. If your oscillator does not oscillate, then successively add a feedback resistor with a value of 1 Meg, or 100 k or 10k. Then add the 2, 22 pF capacitors, if needed.

You can also use more than 1 buffer in series with your output to clean up the signal so it looks more like a square wave.

In your lab report, show a picture of your circuit, your schematic and a screen shot of the output with the measured clock frequency. Does it oscillate at the frequency you expect?

19.2 Reset circuit

You will build a reset circuit, identical to the one you will use in your brd 3 design, but in stages:



First, insert the switch you will be given into your solderless breadboard and using an ohmmeter, measure between which pins are normally open and then shorted when you press the button.

Next, connect it up in a circuit with the 10k pull up resistor and the 5 V rail powered by the Arduino board.

Measure the bounces on the switch as you press it to short the 10k resistor to gnd. What do you measure when the switch is open? Trigger the scope on the falling edge and look for the < 1 msec bounces on the falling edge. Can you record a screen capture of multiple bounces? (hint, once you can see the bounces, use the scope trigger in normal mode)

Why don't you see bounces on the rising edge when you release the button?

Add the 1 uF capacitor as a debounce capacitor. What does the falling edge look like? If the bounces are still there, you may not have the circuit correct.

Now that you have the debounce circuit working, add the coupling capacitor. Use an Arduino digital I/O as the DTR pin. Create a square wave with a period of no shorter than 200 msec. The only higher capacitor in your kit to use as a DC blocking capacitor is 100 uF or 1000 uF. Try one of these as the DC blocking capacitor.

What signal do you see on the switch node in your circuit? How low does the signal go when you use a large capacitor? Now try using a 1 uF capacitor as the DC blocking capacitor. How low does the reset signal go?

This is why you should watch out for making sure the DC blocking capacitor is large enough.

19.3 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.

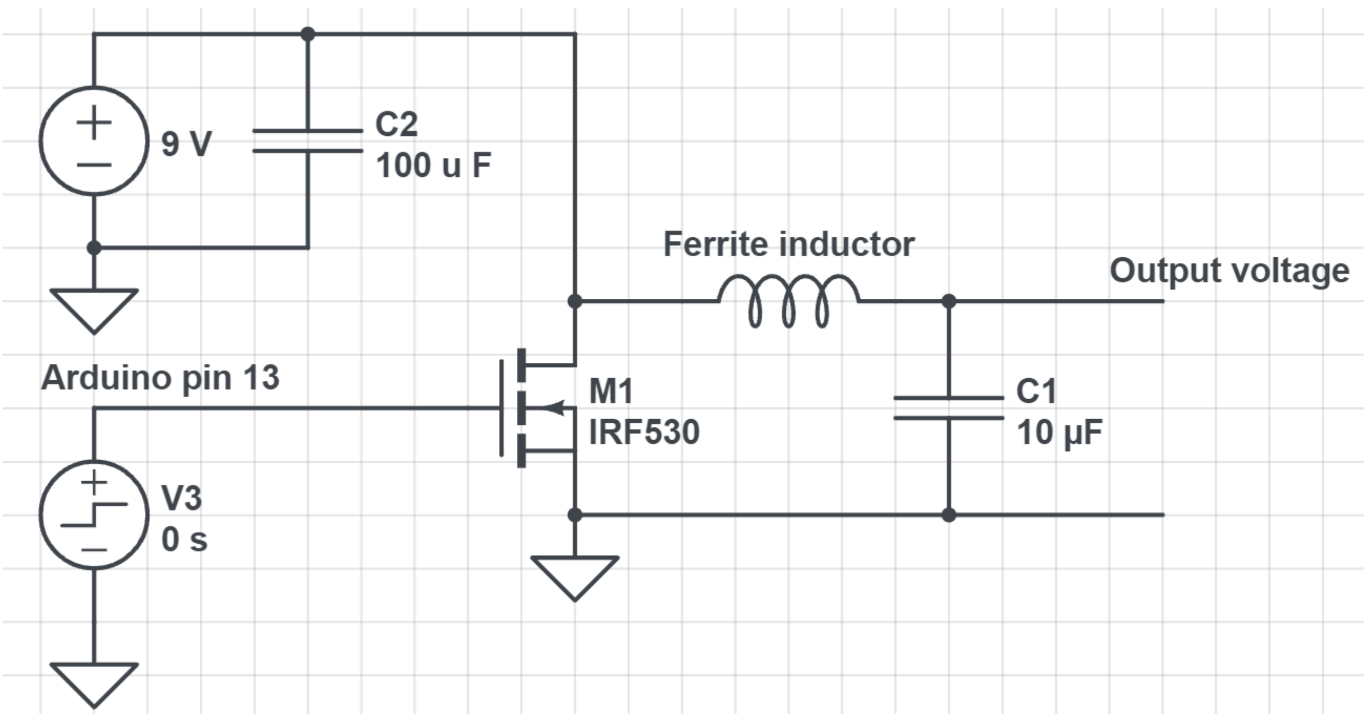
To filter the noise from a power rail getting onto a sensitive power rail, we want to build a low pass filter between the two power rails. The pole frequency should be as low as practical.

We achieve this by adding a large series L with a large C at the IC pin. The L is achieved by using a ferrite bead. The other feature of the ferrite bead is that it adds extra series resistance and this will help to damp any parallel resonances.

To test the low pass frequency, you will build a simple circuit to create noise on the power rail. Then you will see how effective the simple filter capacitor is by itself, and then after adding the ferrite inductor. The metric of filtering will be the rise time of the signal into the filter and the rise time after the filter.

Use the Arduino digital pin as the input to the MOSFET to turn the MOSFET on and off to generate the initial noise. Be sure to use a duty cycle of no more than 5% to keep the power dissipation down. Try a 1 msec on time and 20 msec off time. This is identical to the PDN noise circuit you used in an earlier lab. In use, the output voltage, after the filter will be the AVCC power rail pin, for example.

After you build this circuit, measure the rise time of the falling edge or rising edge on the power rail, the input signal to the filter, and the rise time of the output signal after the filter. The circuit is shown below:



You will need to know the inductance of the ferrite filter. It is 100 uH. If you use a 10 uF capacitor, what is the pole frequency of the LC low pass filter? If the input rise time is 100 nsec, what do you expect the output rise time to be?

19.4 In your optional report, you should include

In wk 10, the report due is your final report on brd 2. This is the only report due this week that you have to turn in. It counts as your midterm.

However, these two circuit examples would make great reports on your portfolio web site. These are very common circuits and showing your ability to design and build them, and analyze the results will show off your expertise at hardware design.

You should include the schematics and take some pictures of the SBB circuit you built and the signals you measured, with your interpretation of the results and post it on your portfolio.

This will further impress any hiring manager with your expertise.

Chapter 20 Wk 11 Monday POR for Brd 4: the sensor shield

This week, we begin the design of board 4. This will be a shield you will insert into a standard Arduino Uno board, either the commercial one, or your own Arduino board.

It will be special for five reasons:

- *It will be 4-layers*
- *It will have through hole and surface mount parts*
- *You will do all the assembly of all the components.*
- *You will have to create some of the symbols and footprints yourself*
- *It will have a number of different sensors you can experiment with*

In this lab, you will develop the POR. While there is nothing to turn in, your POR will guide the work you will do in designing your sensor shield. It will be in the final report for board 4 which will count as a mid term.

20.1 The purpose of the sensor shield

This board will satisfy three goals. First, it will be a chance to design and build a 4-layer board. Second, you will have many different components you will have to assemble to your board. And finally, it will perform many functions. You will have to write some code to exercise each of the different functions.

This board will use new parts for which you may not have symbols and footprints. You will have to figure out either where to get them, or how to build them. This will give you experience managing the library. There are a series of videos in the skill building workshop that discuss the library in Altium.

There are six different sensors and four other components you will add to your board 4:

- *A heartbeat sensor*
- *A temperature-humidity sensor*
- *A Hall effect magnetic field sensor*
- *A CO, carbon monoxide sensor*
- *A microphone*
- *At least 4 smart or digital LEDs*
- *A buzzer*
- *A 12-bit DAC*
- *A 16-bit ADC*

You will use the 16-bit ADC to measure the heartbeat sensor as a differential signal and the Hall effect and CO sensor as a single ended signal.

You will use an Arduino analog pin to measure the output of the DAC and the microphone input.

Here are the parts you will add to board 4:

Part description	Part ID	Link
Hall effect	DRV5053RAQDBZT	Digikey link
Temp/humidity	DHT11	AliExpress link
Gas sensor	MQ-7	AliExpress link
Ambient light sensor (heart)	ALS-PT19-315C/L177/TR8	Digikey link
Buzzer	AT-1224-TWT-5V-2-R	Digikey link
Microphone	CMEJ-0415-42-P	Digikey link
Smart LED	IN-PI55QATPRPGPBPW-60	Digikey link
DAC	MCP4725A0T-E/CH	Digikey link
ADC	ADS1115IDGSR	Digikey link

You should evaluate each of these parts, read the datasheets and think about how you will integrate them into your schematic. Some of them will require additional components, such as resistor and capacitors.

In some cases, you should consider using sockets (female header pins) to insert the senor into our board. This will make assembly and debug easier.

Don't forget all the principles of designing for debug and bring up. What features will you want to add to your board?

We will use the ADS1115 16-bit ADC in the lab on Wed. You will see the details on how to use it and the drivers to talk to it in this lab.

20.2 Details on some specific sensors

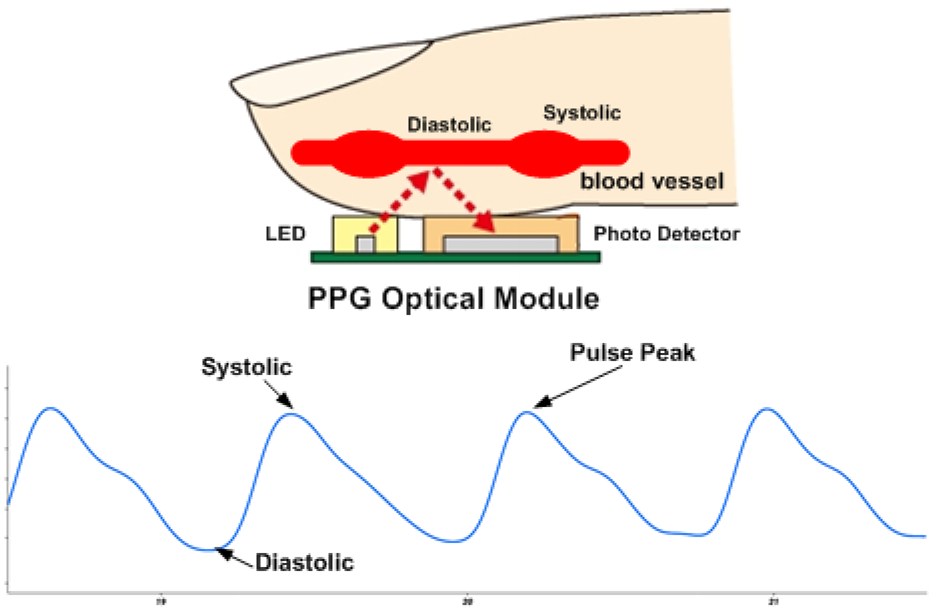
When you connect your sensors to the various power and ground connections and digital and analog pins, you should consider indictor LEDs, test points and isolation switches to assist in the bring up and debug.

The following are some additional notes on each sensor.

20.2.1 The heartbeat sensor:

We will use two smart, programmable LEDs on either side of a photo transistor to create a heartbeat sensor, using the principle of photoplethysmography. This is used in many Fitbits and the Apple iWatch.

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.

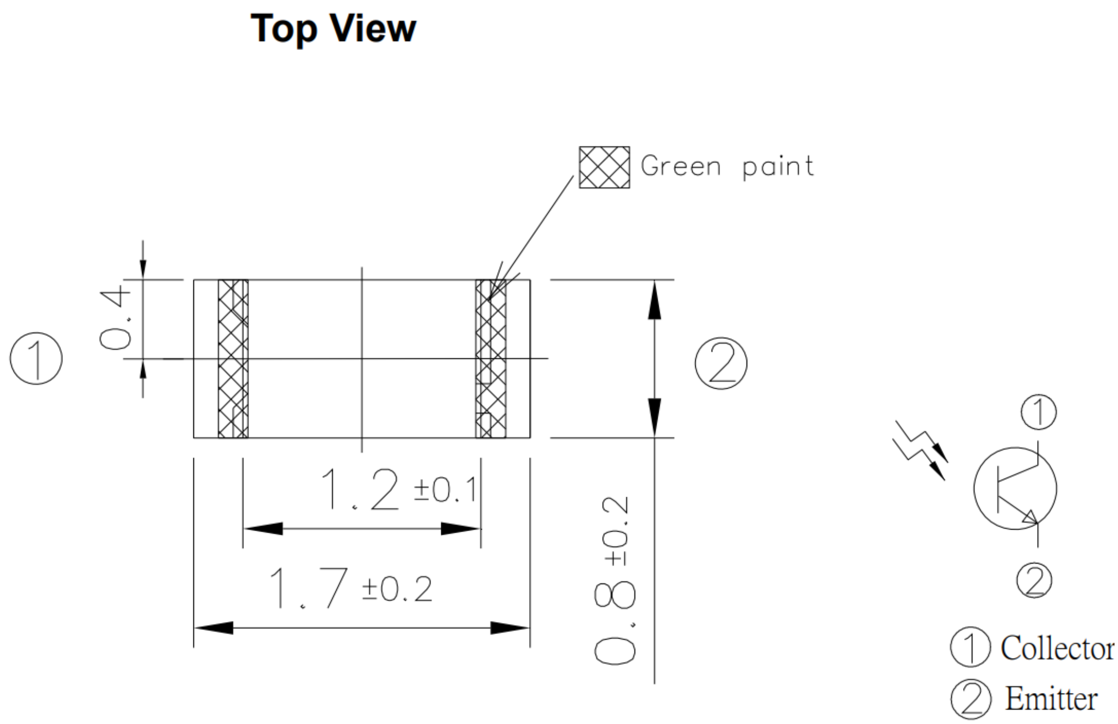


An illustration of the principle of photoplethysmography. From https://www.richtek.com/Design%20Support/Technical%20Document/AN057

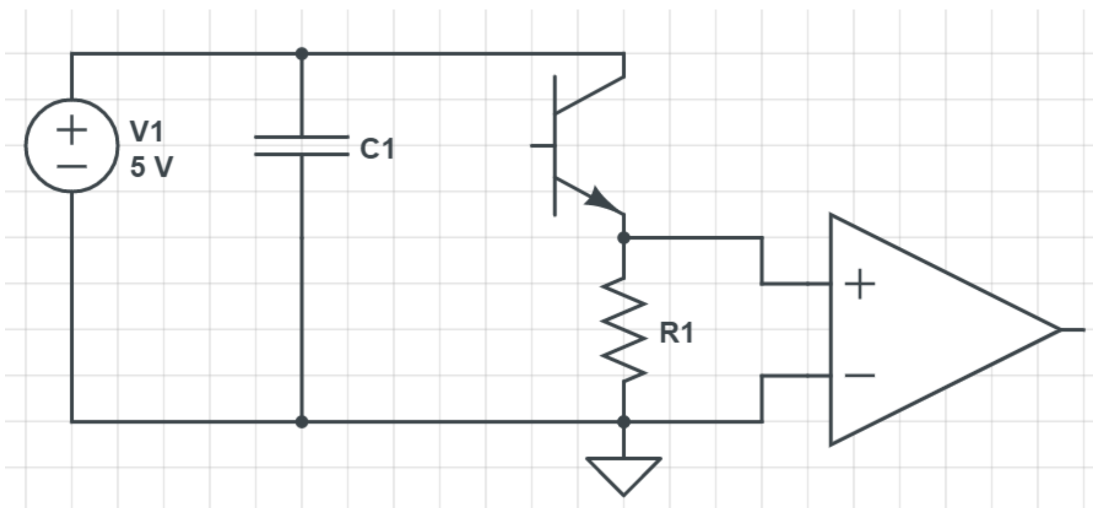
By measuring the current from the photo transistor, you can measure the blood flow in your finger. This is called photoplethysmography (PPG). You can read this article I wrote that used an even simpler configuration to detect the blood flow and some of the code I wrote.

Using a programmable digital LED will enable you to try different colors to see which color is most effective to monitor your blood flow. 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:



The circuit to turn the current through the phototransistor into a voltage is very simple, shown below:



The C1 capacitor is a 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 uA/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 current 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 uA/lux x 1000 lux = 500 uA. For a maximum voltage of 5 V with 500 uA, this would be a sense resistor, R1 = 5 V/0.5 mA = 10k ohms.

Without the chance of doing any experimenting, a good starting value for R1 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 differential input. This will use channels A0 and A1 of the ADC. Routing the + and – inputs from the sensor resistor to the ADC will be very important.

20.2.2 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/296-38849-1-ND/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.

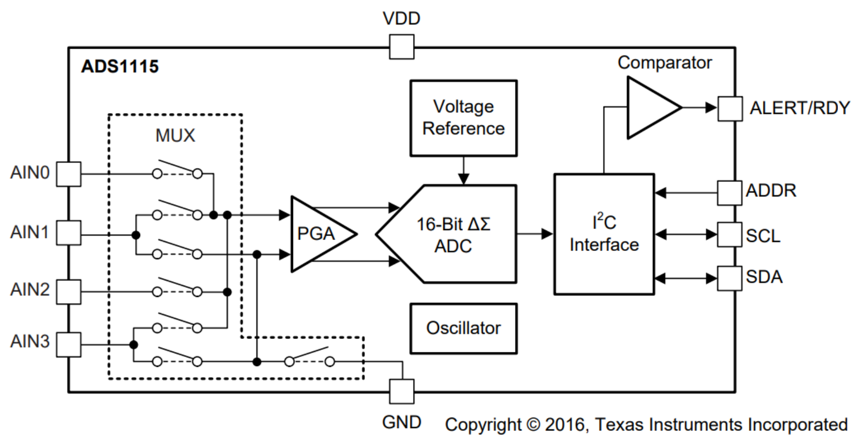


Figure 22. ADS1115 Block Diagram

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:

$$\frac{\text{Volts}}{\text{ADU}} = \frac{4.096\text{ V}}{2^{16}-1} = \frac{4.096\text{ V}}{65535} = 62.5\frac{\text{uV}}{\text{ADU}} \qquad (0.1)$$

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 connections 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.**.

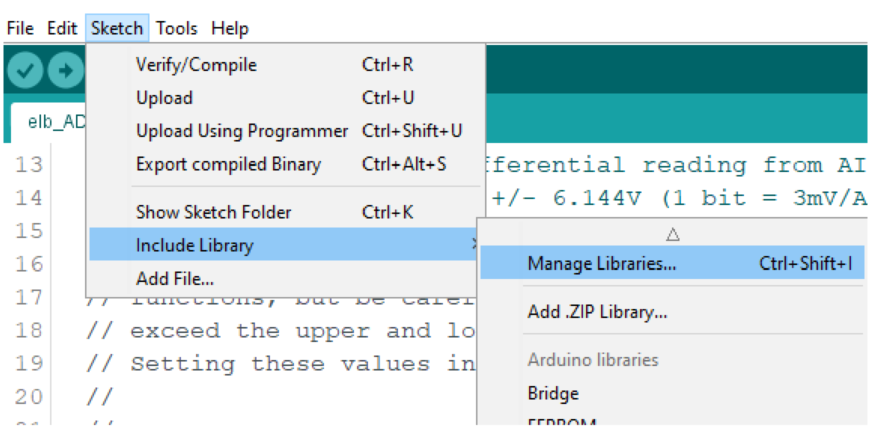


Figure 20.1. To install the driver for the ADS1115, select Manage Libraries under Include Library.

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.

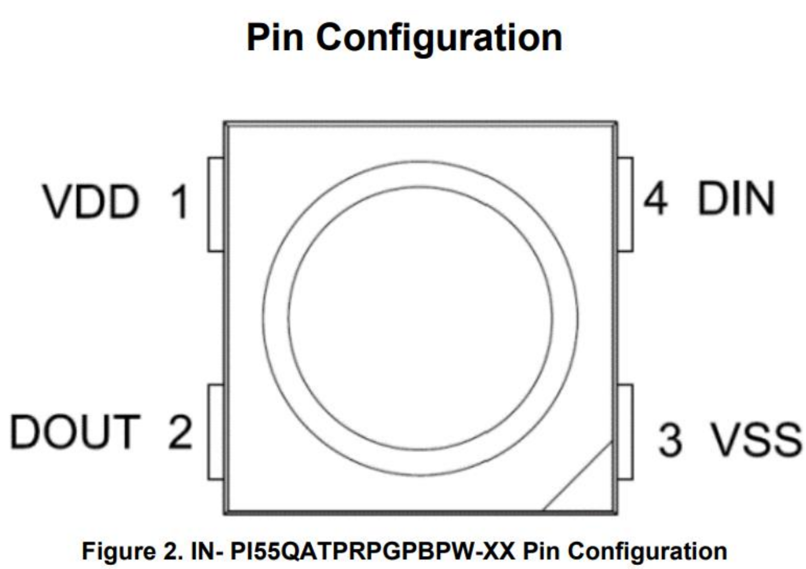
We will use this ADC in the lab on Wed and you will get practice installing the driver and sniffing the I2C bus while we perform single-ended and differential measurements.

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.

20.2.3 The digital LEDs, Inolux

The datasheet for the digital LEDs can be found here. Make note that this is a 5050 type RGBW smart LED. You will need to know this when selecting the Arduino libraries to drive it.

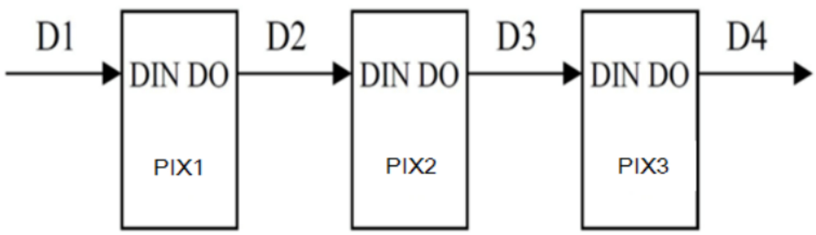
The pin out for each pixel is shown in Figure 20.2.



5. Figure 20.2. Pin configuration, top view for each LED.

They can be powered by 5 V DC as Vdd.

These are digitally controlled LEDs which can be cascaded in series, as shown below.



The digital signal entering D1 from a digital pin should have a 1k Ohm series resistor.

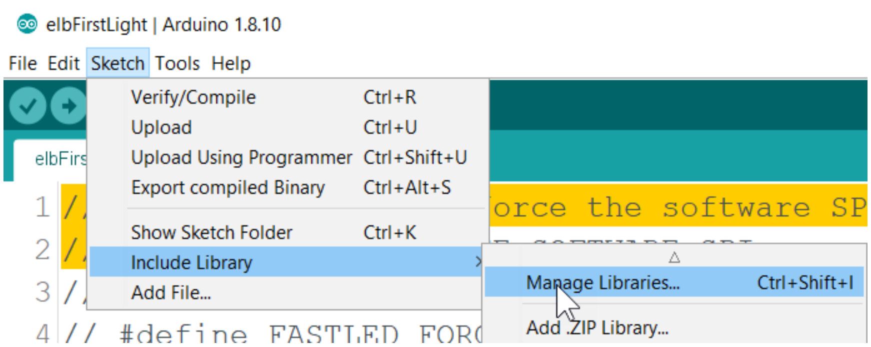
The Arduino IDE library for these LEDs can be downloaded from Adafruit. They are very similar to neo pixels.

You can install the Adafruit NeoPixel library using the library manager and then select one of the examples and hack away.

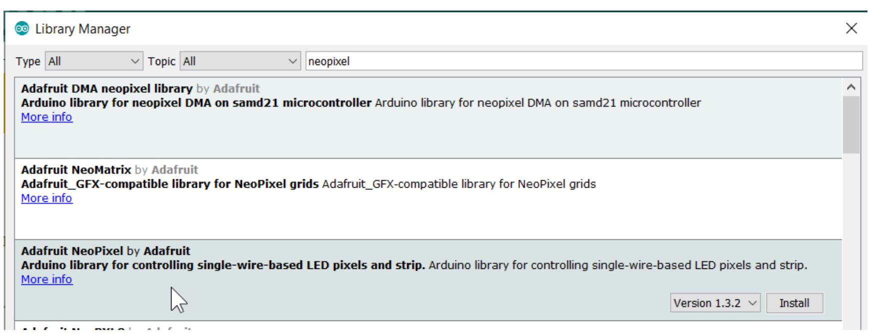
You will use 2 of these for the heartbeat sensor. You will 4 or more additional smart LEDS on your board for whatever purpose you would like. They can make cool lighting effects. No more than 10 total on a board.

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.



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.

You should only need to set the digital pin number and the number of pixels you have in series.

Note: you will use two of these LEDs on either side of the phototransistor as your heart sensor. You can use a separate digital pin to control the other smart LEDs. You can even write the code to drive the other LEDs with some pattern based on your heartbeat.

20.2.4 Temperature humidity sensor

The sensor is manufactured by Aosong for use in air conditioning systems and inside environments. The datasheet can be downloaded here. When mounted onto a circuit board, as we purchased the sensor, the pin 3, which is a no connect, was not pulled out from the circuit board. Only the +5 V, gnd and the data pin come off the board. You can just use a 3-pin socket to attach the sensor to your board 4.

In the code you will write to drive this sensor, you will need to determine which digital pin you will use to talk to this sensor, connect the wiring accordingly and write the code for your board.

This is not the only smart temperature humidity sensor, but it is probably the lowest cost. In low volume, a few units can be purchased for about $0.50 each.

The DHT11 is an old-style sensor, not very fast, not very high performance, but very low cost and easy to use.

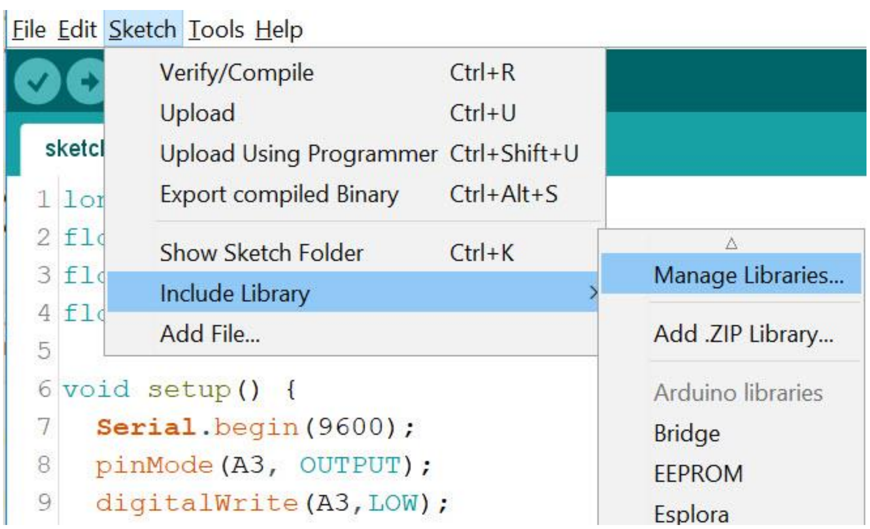
Every smart sensor communicates using digital signals with the microcontroller. Generally, the language it uses is very specialized and custom to that sensor.

Some smart sensors use an I2C interface, some an SPI interface. These are standard interfaces which the Arduino understands.

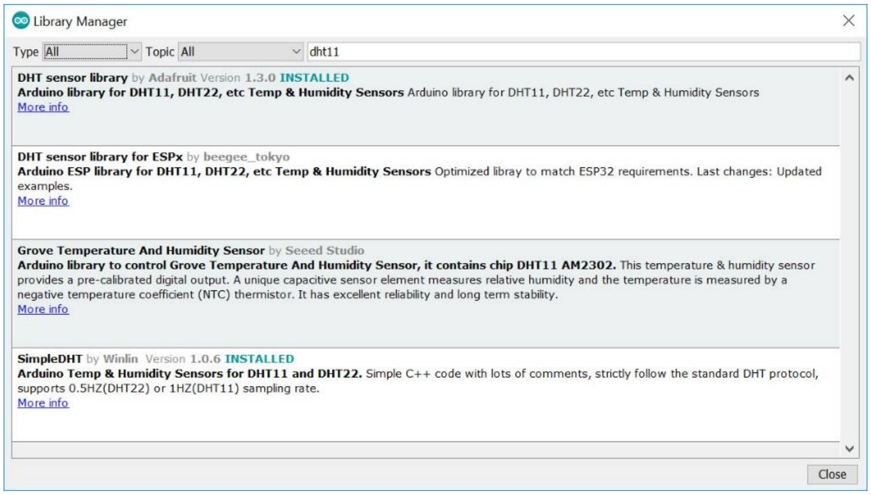
The DHT11 uses its own language and transmits its signals to and from the microcontroller using just one digital pin, measured relative to a ground connection. It receives its power from the 5 V power rail of the Arduino. This means there are a minimum of three wires to the DHT11. Decide in your schematic which digital pin you will use to talk to this sensor.

The library for the DHT11, can be accessed right from the Arduino IDE.

Under the menu item *sketch*, find the line item *Include Library*, near the bottom of the list and select it. This item is shown selected in the figure below. This will open up a new menu with a list of many libraries which can be automatically installed in the IDE.



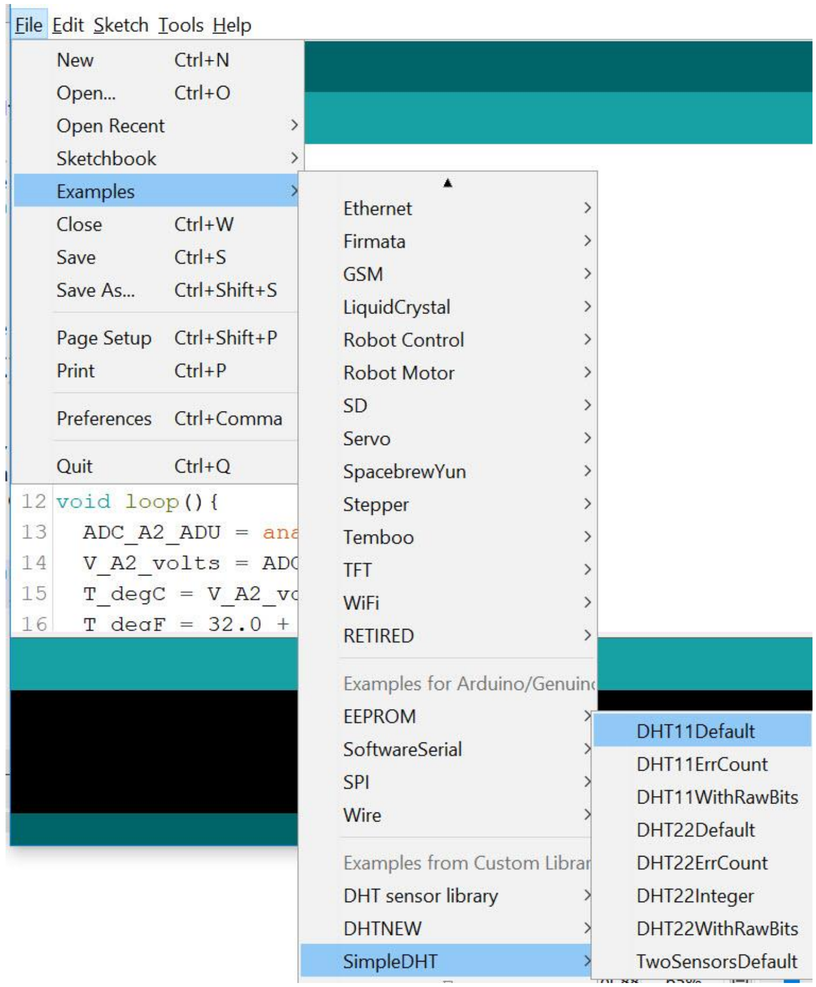
In the search box of this new page, you should type in *DHT11*. When you click on search, you will find a few options, as shown below.



While all of these libraries may work, some are easier to use than others. I recommend selecting the last one, labeled as *Simple DHT*. Select this box and you will see an *Install* button on the lower right of the box. Click this button and you are done.

The simplest way of using this library is to run an example and then modify it to meet your needs. From the *file* menu, select *Examples*. You will have to scroll way down to the bottom of the list.

When you installed the SimpleDHT library, it installed a few new example sketches in the example list. The best one to start with is labeled as DHT11 Default. This is shown below.



Selecting this option will open up an example in a new sketch. Before we run it, we need to make an important modification. Take a look at the complete sketch.

There is a lot of error checking and commenting going on in this sketch. The most important change we need to make is the identification of the digital pin the DHT11 data pin connects. Change this to the pin you select on your board.

When you drive this sensor with this code, below is an example of the output.

```
==================================
Sample DHT11...
Sample OK: 25 *C, 36 H
==================================
Sample DHT11...
Sample OK: 25 *C, 36 H
==================================
Sample DHT11...
Sample OK: 25 *C, 36 H
==================================
Sample DHT11...
Sample OK: 25 *C, 36 H
==================================
Sample DHT11...
Sample OK: 25 *C, 36 H
```

If you want, you can literally plug the sensor directly into an Arduino Uno, set the various +5 V and gnd pins, and test out this sketch prior to designing your brd 4.

You can get a little fancy in your sketch and plot the temperature or humidity directly on the serial plotter.

In Your sketch, there are really only two high level commands you need to pay attention to. The first is:

```
SimpleDHT11 dht11(pinDHT11);
```

It is very obtuse and hard to see what this command does. Basically, the first word, SimpleDHT11 is the command in the library to create an object which can perform a few actions and is connected to the specific pin number inside the ().

The *object* this command creates is a thing, a *noun*. The noun can perform some actions which are verbs.

This command, SimpleDHT11, creates an object which we named *dht11*.

To perform an action, we append a verb to the end of its name, after a period.

The object *dht11* has only one action verb we will use in this sketch. It's rather obtusely hidden in the if statement, but here it is:

```
dht11.read(&temperature, &humidity, NULL))
```

The action verb is to .read the internal settings of the DHT11 sensor, which is connected on the digital pin with the variable name *pinDHT11*. The first term that is sent to the dht11 object is the pin value to which the sensor is connected. The .read command will read and return three variable values: the *temperature*, the *humidity* and another term which we don't care about.

Inside the library, the variables *temperature* and *humidity* are defined. These are special terms. The "and sign", &, in front of the terms, turns these terms into *pointers*. This means that somewhere in memory are the locations of the *temperature* and *humidity* variables.

The dht11.read() command will point to the location where the temperature and humidity numbers are stored and return in the variable names *temperature* and *humidity*, these values.

This is a very convoluted way of just grabbing the temperature and humidity measurements. But we have to do this, based on how the library was written.

And this library is one of the easier ones for the DHT11 sensor.

The DHT11 sensor cannot respond with a measurement faster than every 1.2 second. This means we have to add at least 1.2 second delay after taking a measurement before we can ask for another measurement. This is why there is a delay of 1.5 seconds.

While the code in this example works, it is way overkill for what we need. We can dramatically simplify it.

We strip off all the non-essential code and just use the minimum to set up the sensor and read these two data points.

The complete, minimum sketch is:

```
#include <SimpleDHT.h>
SimpleDHT11 dht11;
byte temperature;
byte humidity;
int pinDHT11 = 4;

void setup() {
  Serial.begin(2000000);
}

void loop() {
  dht11.read(pinDHT11, &temperature, &humidity, NULL);
  Serial.print(temperature); Serial.print(" degC, ");
  Serial.print(humidity); Serial.println(" rH");
  delay(1500);
}
```

This is all the code we need to read temperature and humidity values from the smart sensor. We can use the terms temperature and humidity as variables. Stored in these variables are the latest values of the temperature and humidity.

Remember, this code will only run if you have already installed the `<SimpleDHT.h>` library.

When we run this sketch, we will print the current temperature and humidity. Below is an example of the serial monitor showing the measured temperature and humidity.

```
COM5
23 degC, 29 rH
23 degC, 29 rH
23 degC, 29 rH
23 degC, 29 rH
23 degC, 30 rH
23 degC, 36 rH
23 degC, 40 rH
23 degC, 43 rH
23 degC, 46 rH
23 degC, 49 rH
24 degC, 51 rH
24 degC, 48 rH
24 degC, 45 rH
24 degC, 43 rH
24 degC, 41 rH
24 degC, 39 rH
24 degC, 38 rH
24 degC, 37 rH
24 degC, 36 rH
```

20.2.5 Sketch: Plotting the temperature and humidity

Now that we have a way of measuring these two environmental parameters and we can display them every 1 ½ seconds, we can plot either one on the serial plotter.

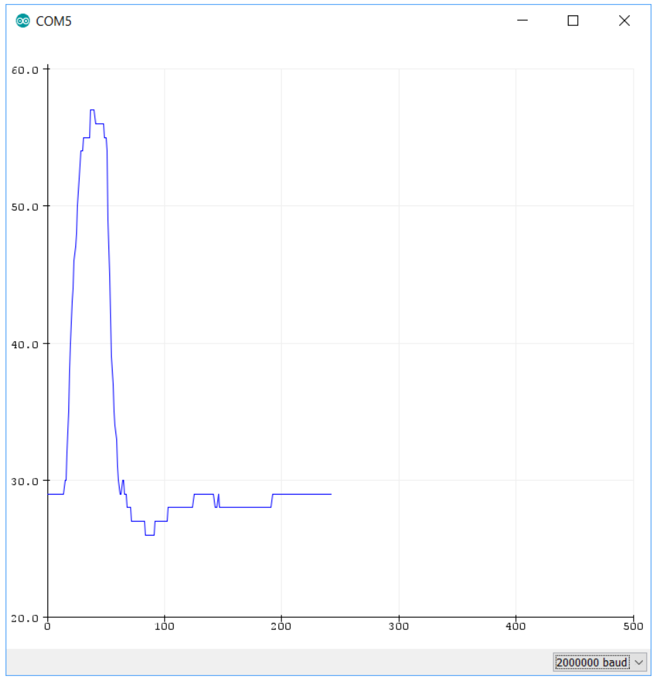
We need to select which term to plot and then print out each data point on a separate line.

Here is my sketch:

```
#include <SimpleDHT.h>
SimpleDHT11 dht11;
byte temperature;
byte humidity;
int pinDHT11 = 4;

void setup() {
  Serial.begin(2000000);
}

void loop() {
  dht11.read(pinDHT11, &temperature, &humidity, NULL);
  //Serial.println(temperature);
  Serial.println(humidity);
  delay(1500);
}
```

When I plotted the humidity, I placed my finger over the sensor for a few minutes and then moved it away. This recorded a spike in the humidity.

The time base corresponds to 100 points per division, at 1.5 sec/point, or a total time of 150 seconds per division, which is 2.5 minutes per division. The figure below shows the plotted humidity after I held my finger over the sensor.



20.2.6 The Hall effect sensor

This is a very easy to use sensor. Read the datasheet. Be sure to identify the specific part we will be using, what the ID number describes and which part in the data sheet to pay attention to. This is very important.

the output voltage of this sensor is proportional to the magnetic field going through it. The sensitivity is about 45 mV/mT. The units of magnetic field density are Tesla, the same as the car.

The earth's magnetic field is about 25 to 65 uT, depending on where you are and the geomagnetic activity. This means, the typical voltage we will measure from the sensor may be only 45 mV/mT x 0.05 mT = 2.2 mV. This is pretty small. If you bring a small magnet nearby, which is a very cool thing to try, you will measure a larger voltage.

This small voltage from this sensor is why we are going to use the 16-bit ADC to measure the voltage. You may have to use a high gain setting on the PGA in order to see the 2 mV signal from the Hall effect sensor. What should you add to the sensor to reduce the noise?

You can decide if you use a single-ended connection or a differential connection to the ADC for this sensor.

20.2.7 The CO sensor

The Co sensor, the MQ7, is part of the MQ series. It is a resistive sensor who's resistance changes with the ambient concentration of CO. You can find a good datasheet for the MQ-7 sensor from Sparkfun, for example.

It is also sensitive to other gases, as shown below, taken from the datasheet:

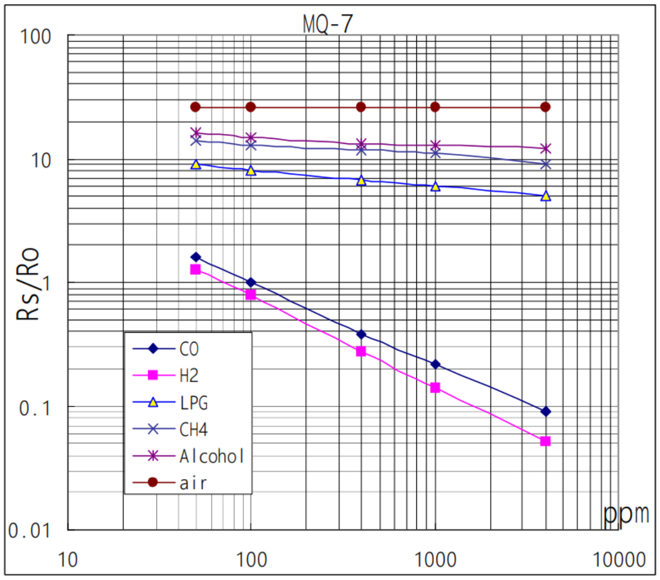


Fig.3 is shows the typical sensitivity characteristics of the MQ-7 for several gases.

in their: Temp: 20℃、

Humidity: 65%、

O_2 concentration 21%

RL=10k Ω

Ro: sensor resistance at 100ppm CO in the clean air.

Rs: sensor resistance at various concentrations of gases.

Fig.3 sensitivity characteristics of the MQ-7

The sensitivity of a gas concentration affects the resistance of the sensor. The higher the cocnetration of the gas, the lower the resistance.

The sensor we are suing is already mounted onto a circuit board. It is not the best circuit, but it si very low cost.

There are four connections: +5 V, gnd, the voltage out of a resistance divider and a digital pin from a comparator. Ignore the digital pin from the comparator. The circuit on the small board is a simple voltage divider with a 10k resistor on the low side, connected between the gas sensor and ground. The analog output pin is the voltage across this 10k resistor. As the sensor resistor decreases, ie, more CO detected, the voltage across the 10k resistor increases.

Use one of the Arduino analog pins to measure the voltage across the 10k on-board resistor.

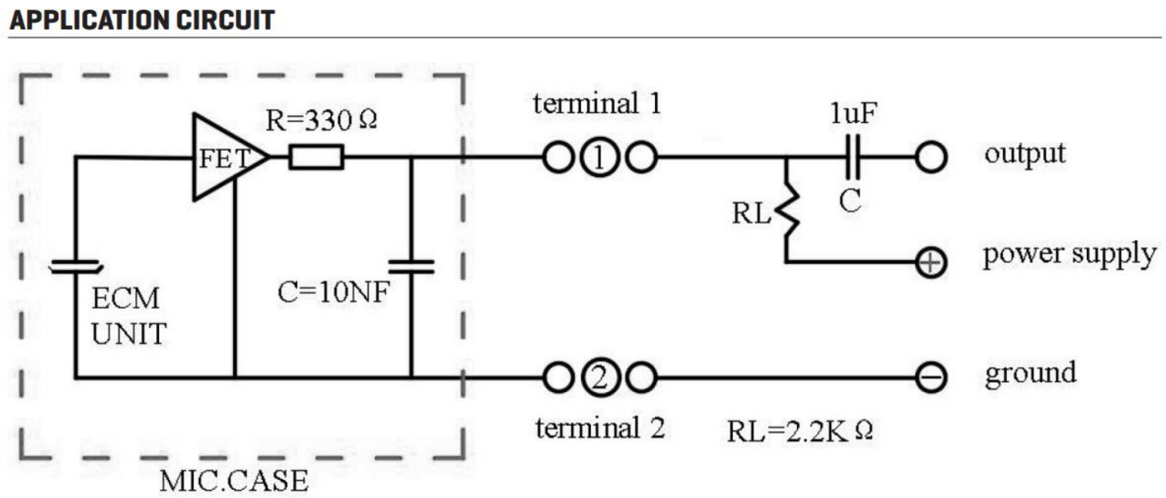
To operate effectively, this sensor also needs 5 V to power an on-board heater. This is directly connected to the 5 V pin on the small PCB the sensor comes attached to.

20.2.8 The buzzer

This is a very simple component. When 5 V is applied to it, it buzzes at 2.4 kHz, with a very loud, annoying tone. You can drive it with a digital pin from the Arduino. You pick which one. You can use it as an alarm for example.

20.2.9 The microphone

This microphone is special in that it has a built in FET amplifier. The microphone is connected with only two pins, connected to 5 V and gnd, but there is additional circuitry on your board required to extract the analog voltage corresponding to the sound picked up. The circuit diagram is shown in the figure below, from the datasheet:



On your board, you will need to add the AC coupling capacitor and the load resistor. What value of RL should you use? Look at the complete circuit. There is a low pass filter and a high pass filter. What value does the datasheet suggest?

Based on this circuit, you should estimate the low pass and the high pass pole frequencies, so you know what bandwidth range to expect from the mic.

Use one of the Arduino analog pins to measure the output from this mic.

20.2.10 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.

Chapter 21 Wk 11 Wed Single-ended and differential measurements and the I2C communications

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.

21.1 Why differential measurements with 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.

21.2 Wire up your solderless breadboard

As a good habit, we will always try to use a consistent color coding habit:

- ✓ *Ground is green, grey or black*
- ✓ *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.

21.3 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.

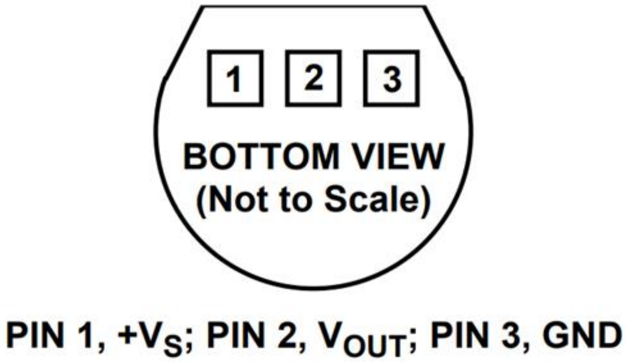


Figure 4. T-3 (TO-92)

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.

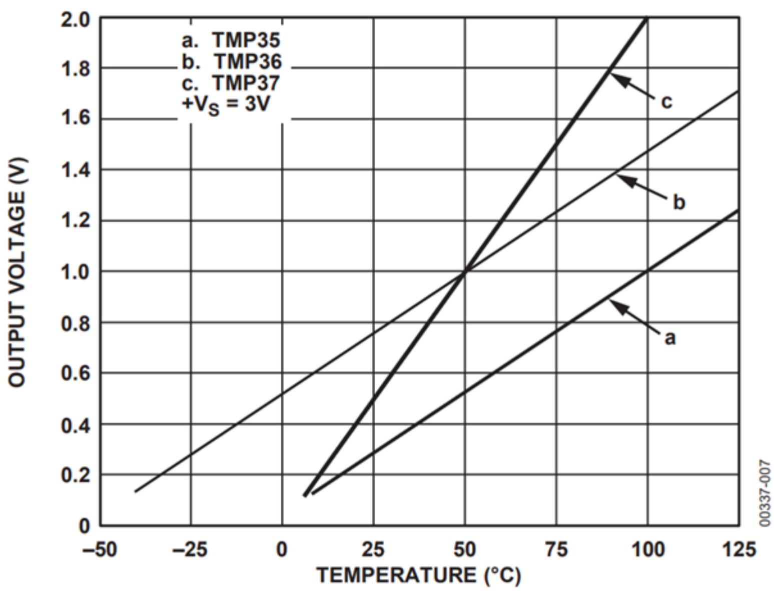


Figure 6. Output Voltage vs. Temperature

Figure 26.2. Calibration curve for the TMP36 sensor. It is curve b.

The temperature and voltage are related by:

$$T[\deg C] = V[\text{volts}] \times 100[\deg C/V] - 50[\deg C]$$

and

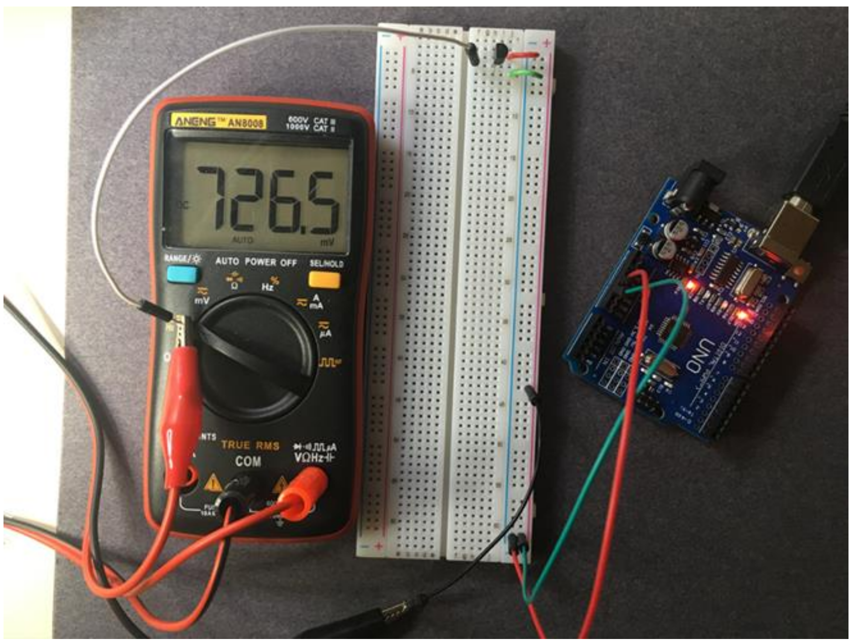
$$V[\text{Volts}] = T[\deg C] \times 0.01[\text{Volts}/\text{decC}] + 0.5[\text{Volts}]$$

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.



21.4 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.

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.

Once one pin is set, move from pin to pin soldering the others. This process is described in one of the skill building workshops on Master Soldering.

Here is the link to the [Digikey page]() 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]().

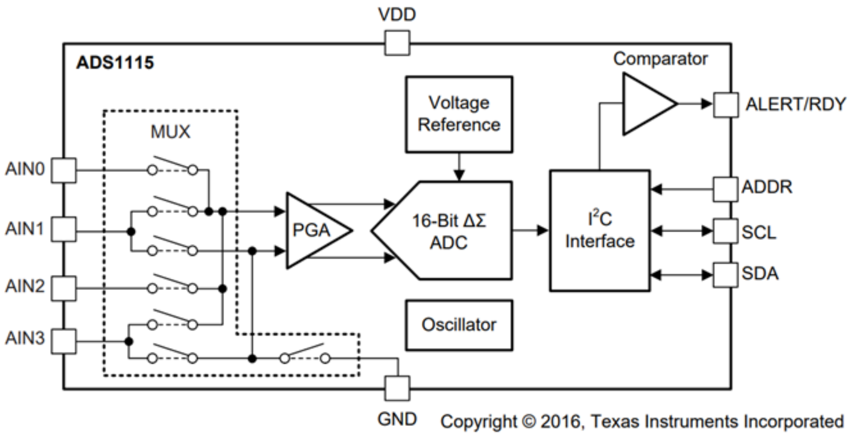


Figure 22. ADS1115 Block Diagram

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:

$$\frac{\text{Volts}}{\text{ADU}} = \frac{4.096\text{ V}}{2^{16}-1} = \frac{4.096\text{ V}}{65535} = 62.5\frac{\text{uV}}{\text{ADU}}$$

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.

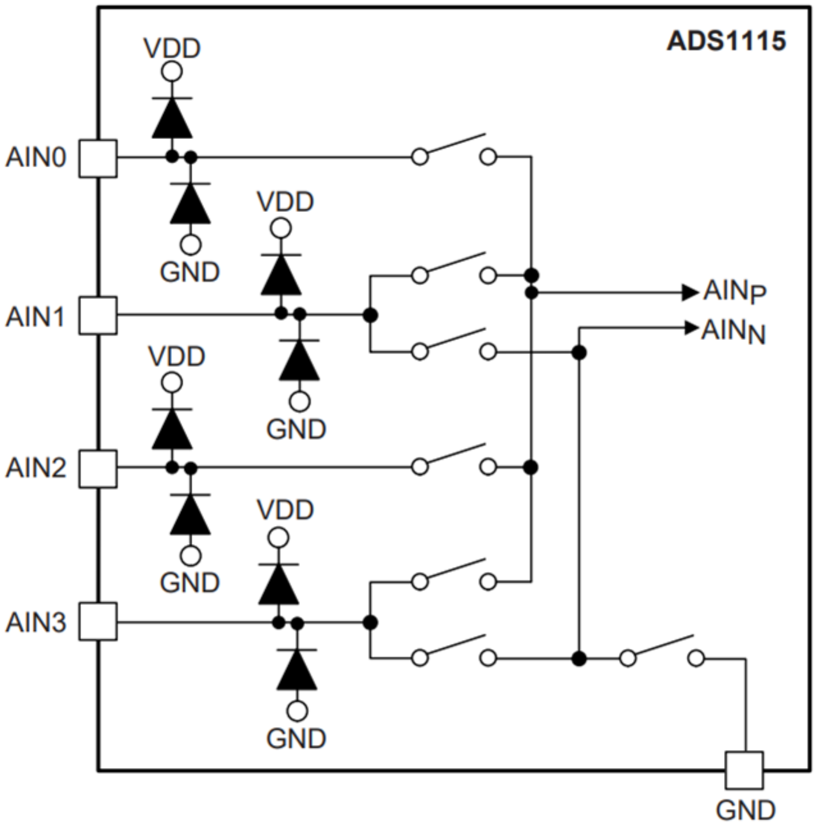


Figure 25. Input Multiplexer

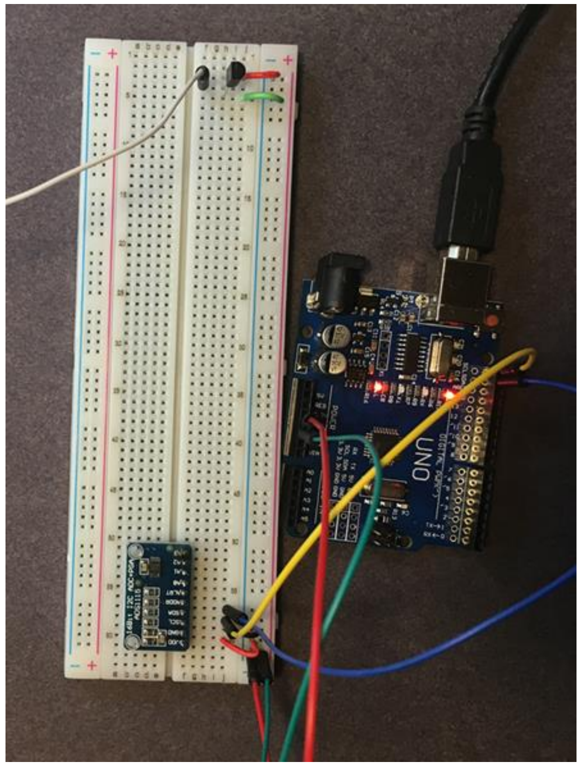
The interface to the ADS1115 is an I2C interface. This means you need to connect the serial clock, SCL, and serial data, SDA, pins to the SCL and SDA pins of the 328.

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 of these connections. You can use either connections into 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.



21.5 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

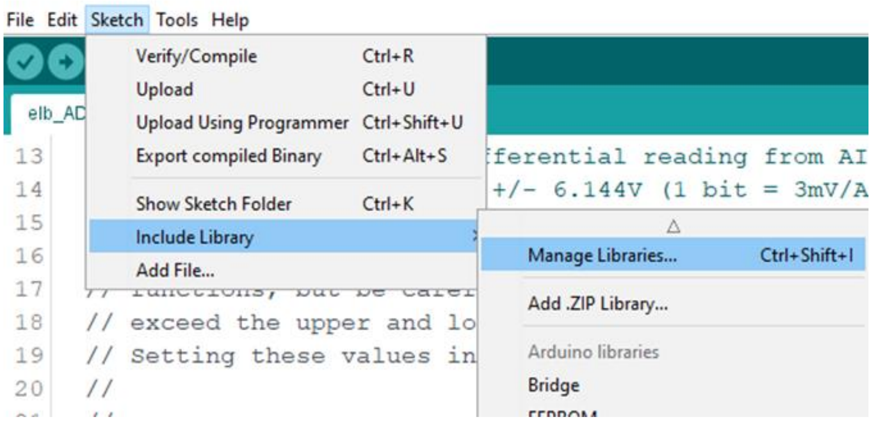
AIN3 ----- local gnd

21.6 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.

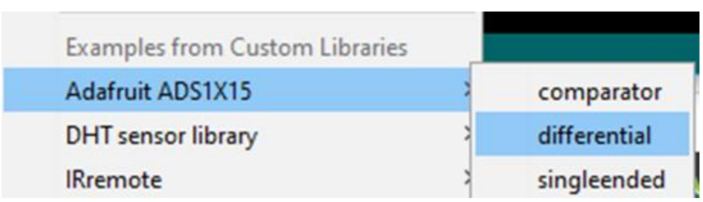


Figure 26.10. Select the differential example.

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 9, I prefer using a baud rate of 2000000, because we can.

We remove the various print statements on lines 10, 12, 13, and the comments in lines 15 thru 18.

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 24 to uncomment to use this gain value.

This is a scale factor of 0.03125 mV/ADU. This is on line 24.

The actual command to read the differential channel that is the Vin0 – Vin1, 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. It is this voltage that is really the voltage of the sensor.

21.7 Differential vs 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 uV values and print them to the serial monitor where we can see the numbers or plot them.

The complete sketch to do all this is here:

```
#include <Wire.h>
#include <Adafruit_ADS1015.h>

Adafruit_ADS1115 ads;  /* Use this for the 16-bit version */

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
```

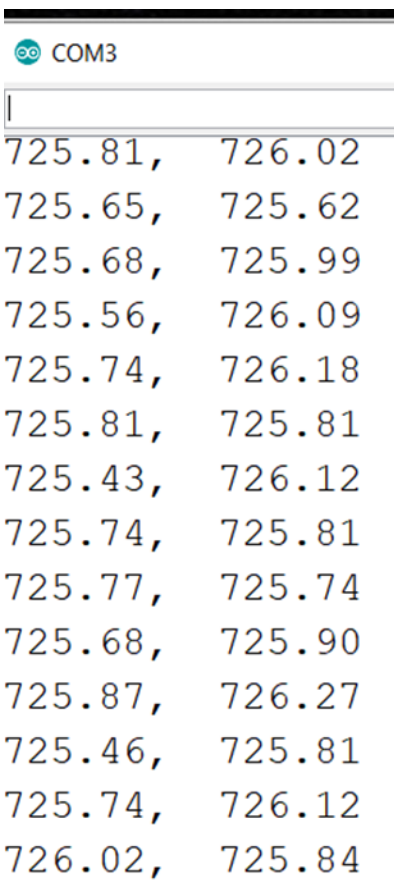
```
  // ads.setGain(GAIN_EIGHT);      // 8x gain   +/- 0.512V  1 bit = 0.25mV   0.015625mV
  // ads.setGain(GAIN_SIXTEEN);    // 16x gain  +/- 0.256V  1 bit = 0.125mV  0.0078125mV
  ads.begin();
}

void loop(void)
{
  Serial.print(ads.readADC_Differential_0_1() * 0.0312); Serial.print(",  ");
  Serial.println(ads.readADC_SingleEnded(0) * 0.0312);
}
```

You can literally copy and paste this code into a blank sketch and it will run.

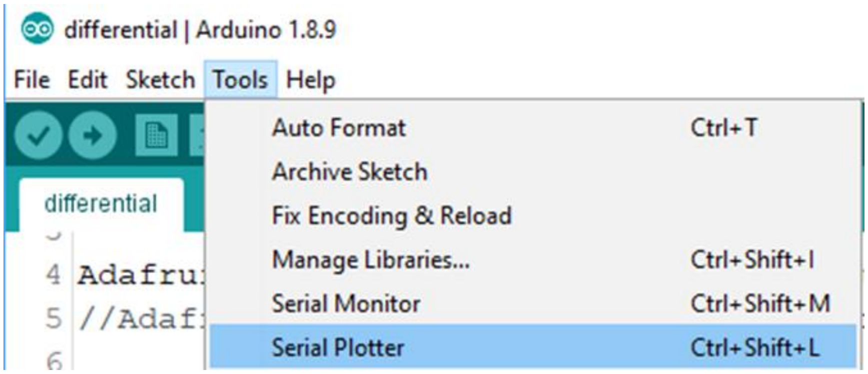
The ADU values are converted into uV by multiplying the ADU values x 0.0312 mV/ADU.

When this sketch is run, the first number printed is the differential voltage in mV and the second is the single-ended voltage in mV. These should be the same if there is no voltage difference between the local grounds.

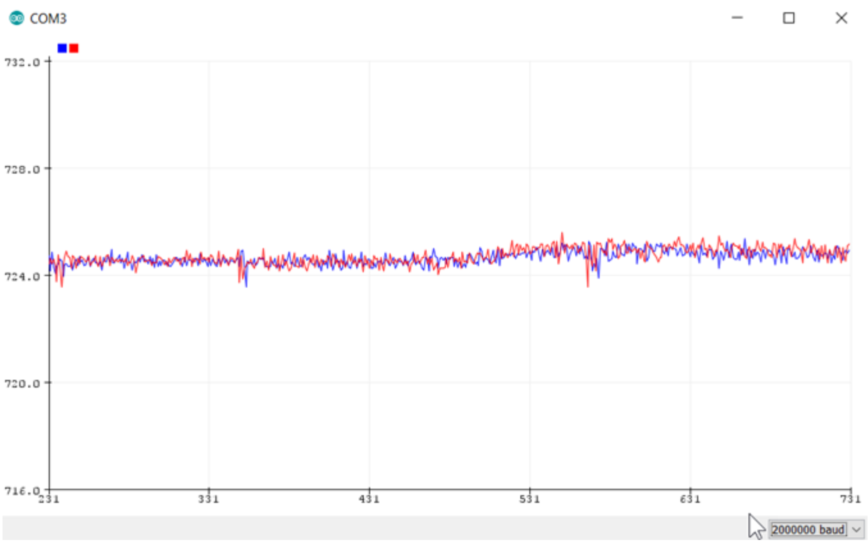
An example of the signal recorded on the serial monitor from a TMP36 is shown below. Remember to use the same baud rate in the serial monitor as in the Serial.begin() command.



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.



The measurements will be taken as fast as they can and the differential signal, in blue will be plotted along with the single-ended signal in red. An example of a plot of 500 points is shown below.



The differential signal in blue and single ended signal in red. They are measuring the same sensor. The vertical scale is 4 mV/div, which is 0.4 degC/div.

In this example, even though there is a long path between the TMP36 and the ADS1115 local grounds, the TMP36 sensor voltage is measured as 724 mV in both methods.

The noise on the voltage measurements is also apparent. It is about 0.4 mV noise. This can be reduced by applying some averaging, if you want.

21.8 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.

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 little current flowing through this roughly 0.04 ohm path.

We will generate some noise in the ground path by driving some current through it using the function generator.

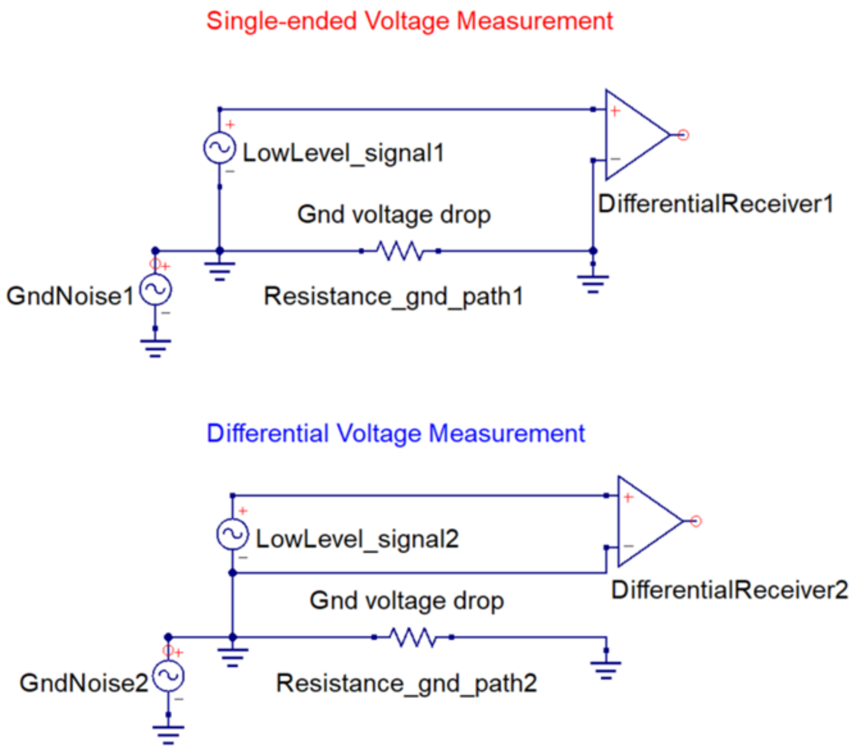
What is the output resistance of the function generator? If you are not sure, go ahead and measure it again using the scope and its 1 Meg input and 50 Ohm input.

When we set up the function generator for a 1 Hz, 1 V peak to peak signal and short the ends of the signal, what is the current that flows from the function generator?

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.

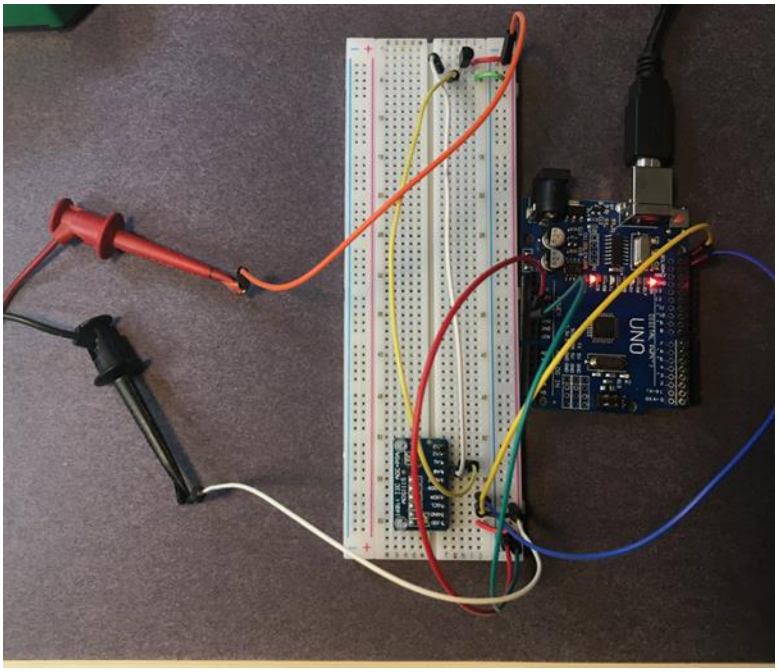
This is why it is sometimes confusing calling this conductor "ground". It is the same conductor, but due to the DC currents flowing in it, we will have a different voltage at different points of the conductor. Which point on the conductor we select as our "reference" point will influence the voltage 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.



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 40 mA peak to peak, the voltage drop would be 0.04 Ohms x 40 mA = 1.6 mV peak to peak. We should see a periodic noise of 1.6 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.



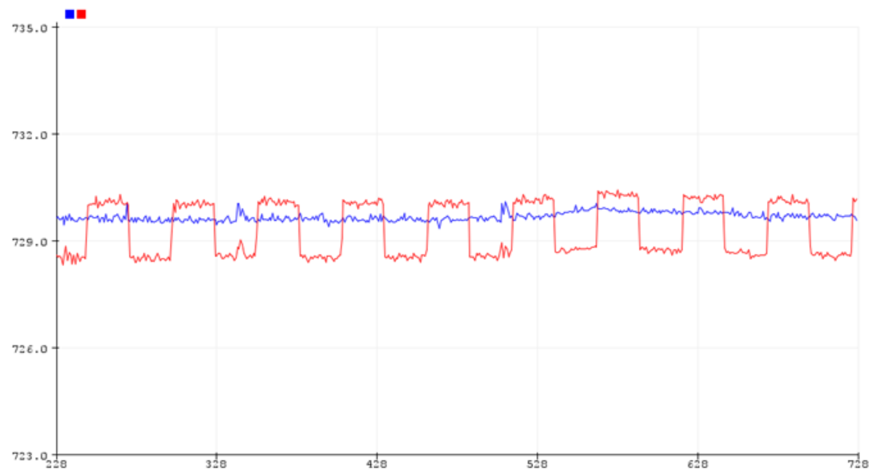
The wiring to drive an external current in the ground return path of the right most column which is the ground return of the TMP36 and ADS1115.

Note that the AIN1, the minus input to the differential amplifier, is always connected to the local ground of the TMP36.

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 local ground of the ADS1115 by an internal to the chip multiplexer.

When a 1 Hz square wave of 40 mA peak to peak current flows on the ground return path, the noise voltage is seen on the single-ended measurement of about 1.5 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 single-ended and differential measurements of the TMP36 sensor with a 1 Hz periodic current in the return path. The scale is 3 mV per div.

The modulated voltage noise is about 1.5 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.

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.

21.9 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 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.*

21.10 Important 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 TMP36 sensor on your board to the ADS1115 chip. You will use a differential pair between the photo transistor of the heartbeat sensor to the ADS1115 chip. Describe how you will do the routing of these two differential pairs.*

21.11 The Lab Report

In your report 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?*

21.12 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. The SDA line is bidirectional. Each device on the I2C bus is connected in parallel to all the other devices. Each minion device has an address which it listens for on the data line.

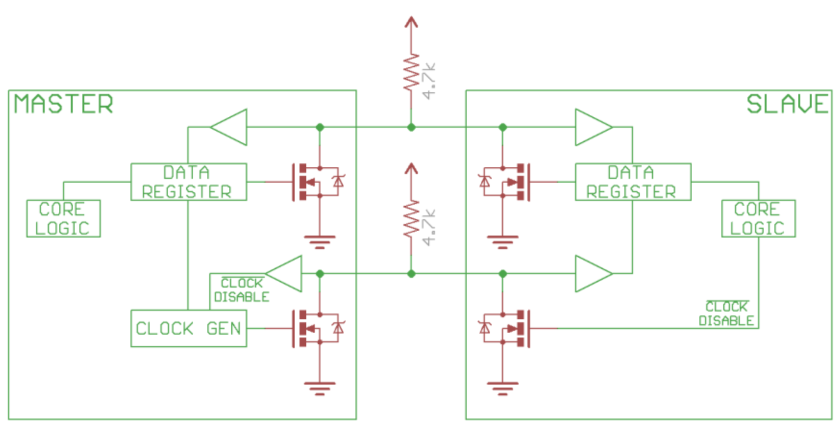
The standard rated frequency for the bus is 100 kbits/sec. In fast mode, this can go up to 400 kHz.

A very good tutorial on the I2C bus can be found at Sparkfun.

The SCL and SDA lines 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

$$R = \frac{5V - 0.4V}{3mA} = 1.5k$$

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.

Since we want to minimize the number of unique components, rather than using a 1.5 k resistor, we will use 1 k resistors as pull up resistors.

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 slave is a 20 pF load on the SDL 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.

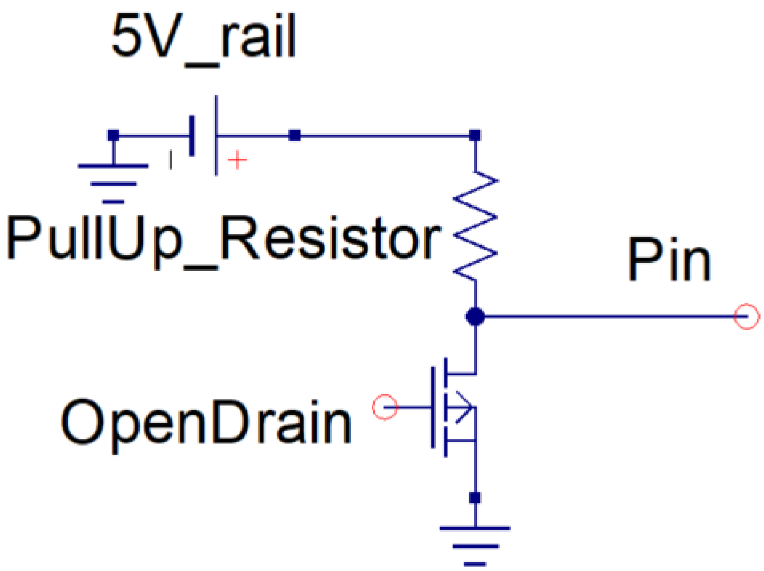
The typical bus frequency is 100 kHz. This is a period of about 10 usec. If we want the charging or discharging time to be no longer than 10% of the period, which is 1 usec as the 10-90 rise time, then the pull up resistance criteria needs to be:

$$2.2 \times RC < 1u\sec \quad \text{and} \quad R < \frac{1u\sec}{2.2 \times 30pF} = 16k$$

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.

A small value pull-up resistor and the line will charge faster, so better timing margin, but more power might be dissipated.

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 a pull up resistor, as shown below.



The specification for the 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:

Vopen = 5 V

Vload = 1 V

This means the Thevenin voltage is 5 V and the Thevenin resistance is 40k. This would make the voltage across the 10k load as:

$$V_{load} = V_{Thevenin}\frac{10k}{10k + 40k} = V_{Thevenin}\frac{1}{5} = 1V$$

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

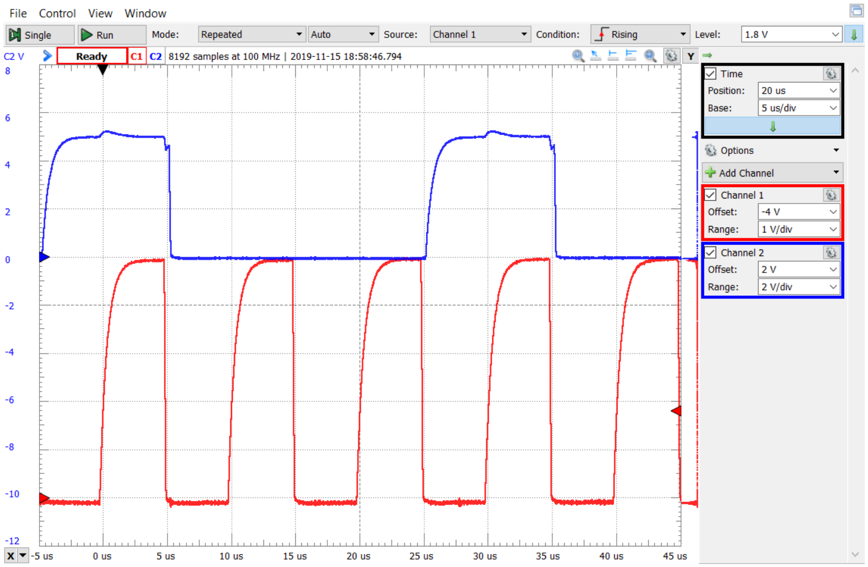
$$2.2 \times RC = 2.2 \times 40k \times 30pF = 2.6u\sec$$

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.

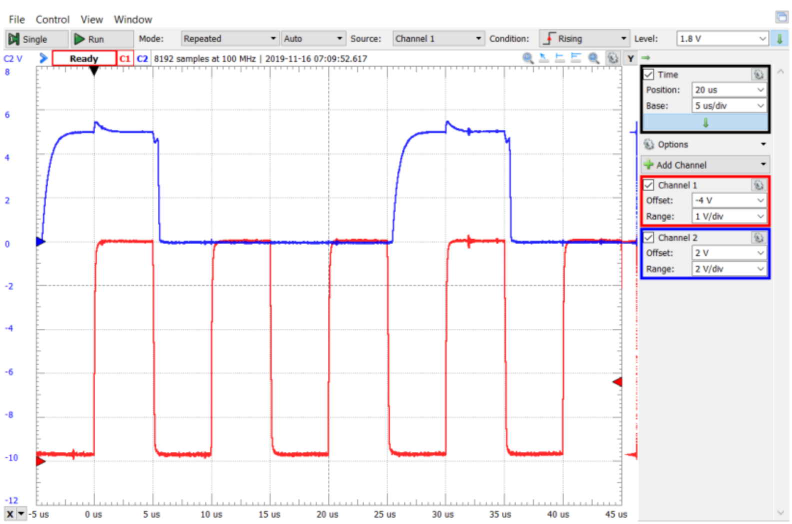
This is why some I2C buses work just fine without an external pull up resistor due to having an internal one already on the 328 output pin, though rather high.

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.

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 not adding a pull up resistor, or adding 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.



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 sensor board, will you add a 1 k or 10 k pullup resistor or no resistor at all?