# 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.

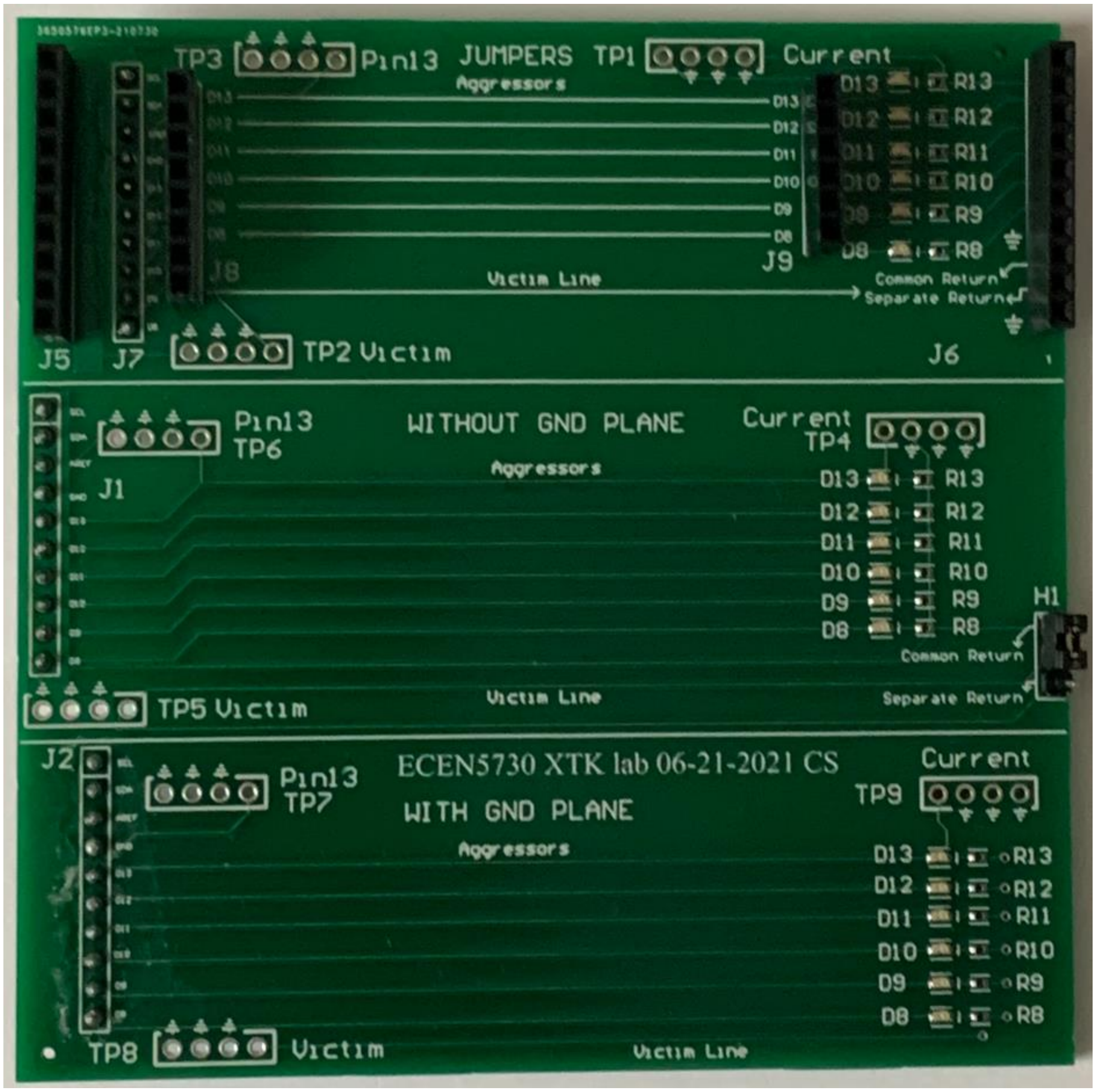
In each case, you will also want to follow the interconnect path of the signal and return conductors of the aggressor and the victim loops.

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.

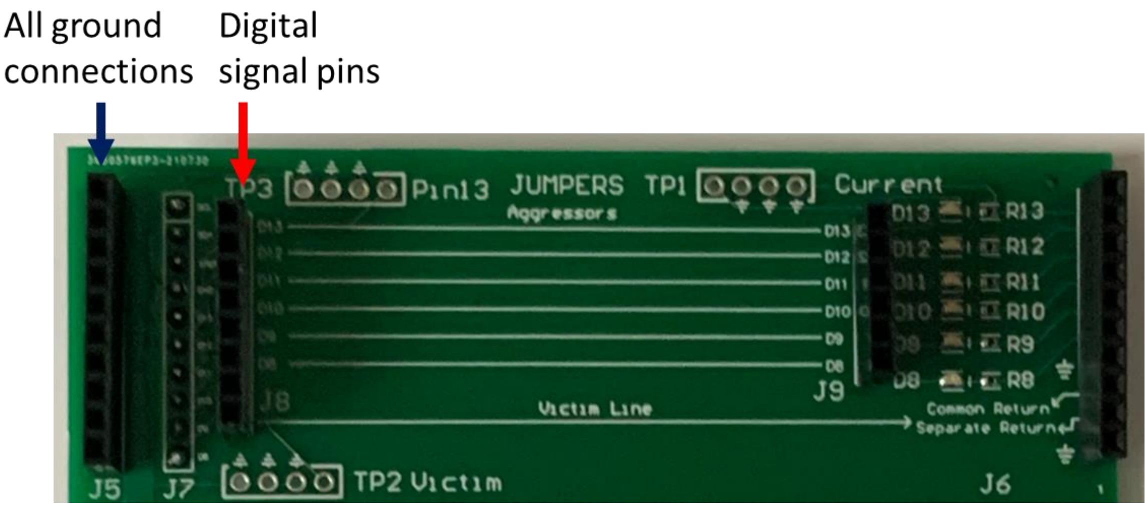
You should write your own code. Here is my sketch:

```
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:

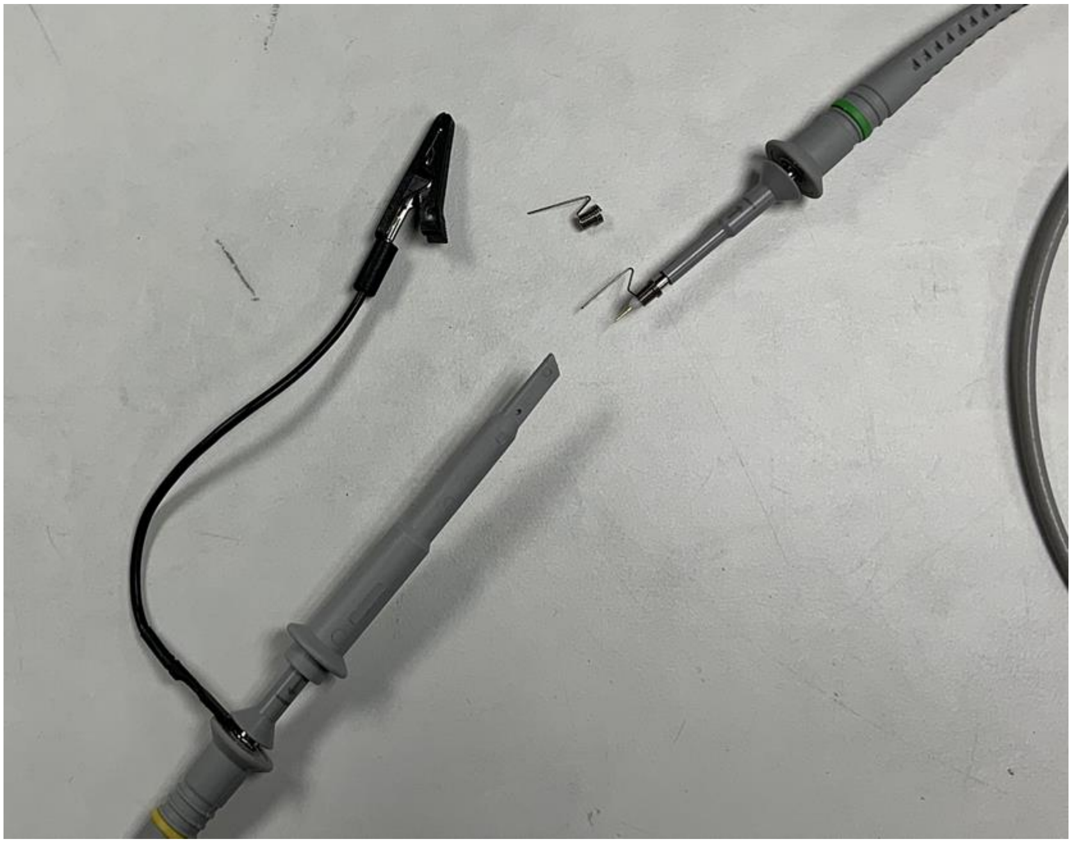
1. *Make sure the probe is set for 10x*
2. *Make sure the scope is set for 10x attenuation*
3. *Use a color code band on the 10x probe to match the color of the trace on the scope*
4. *Check the compensation of the 10x probe*

**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.



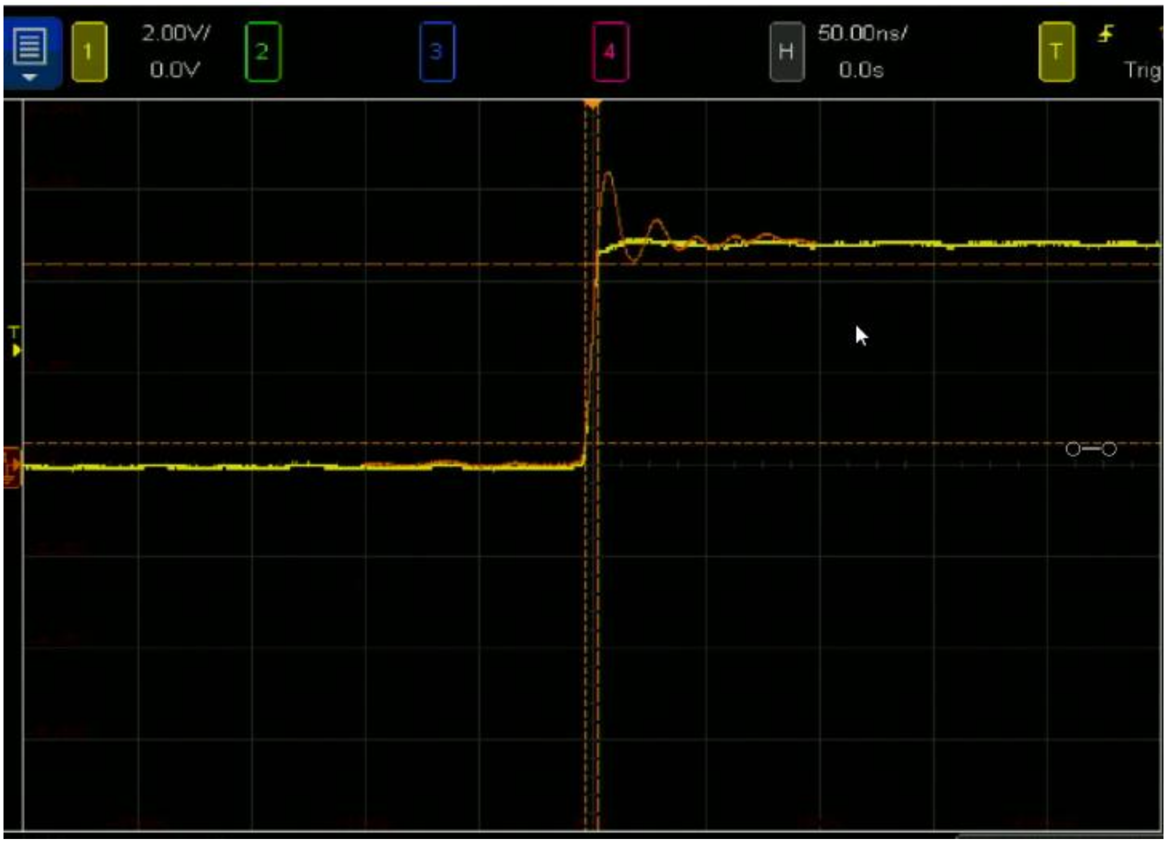
Now you have the shortest practical signal-return loop on the probe tip.

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.

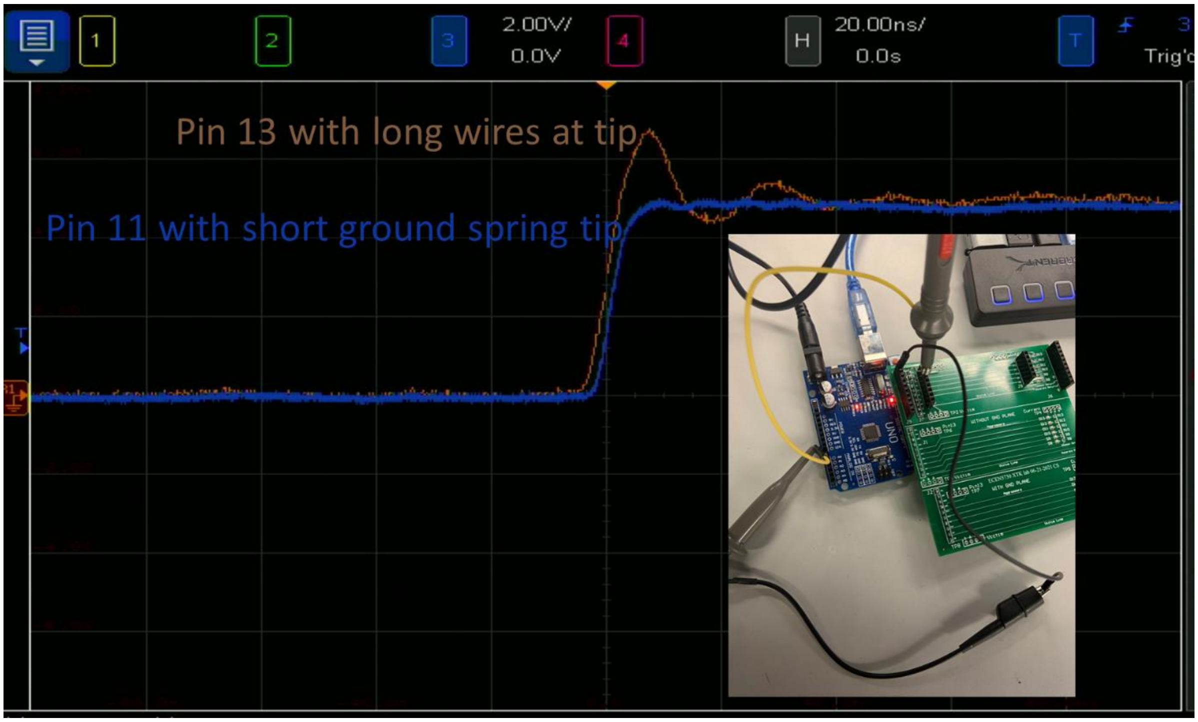
From this exercise, what do you conclude about the best measurement practices for high speed signals?

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.

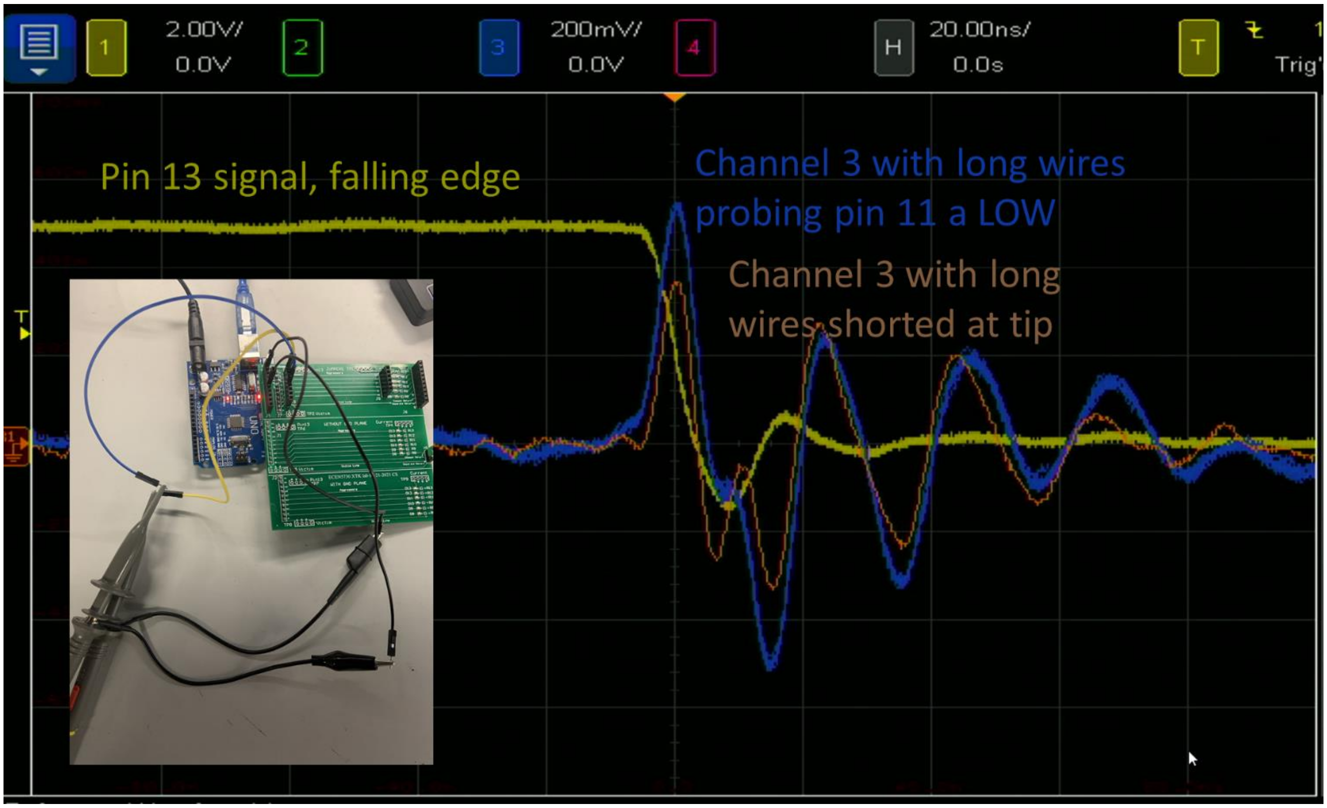
Here are my examples when I did this measurement. Below is the signal from pin 13 with long floppy wires on the tip and the signal from pin 11 with the small spring ground tip. All the ringing and overshoot in the pin

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.

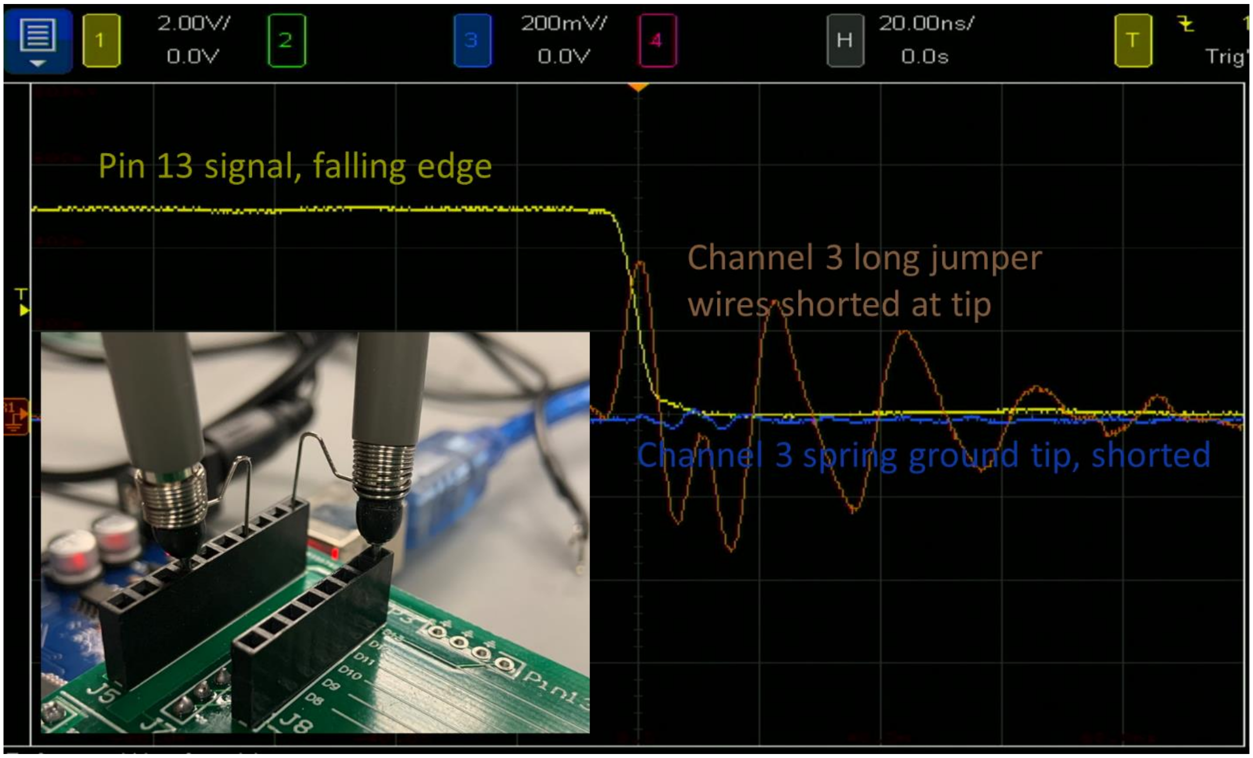



This measurement suggests, when probing high speed signals, use short wires at the tip, preferably the pin and spring ground tip.

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

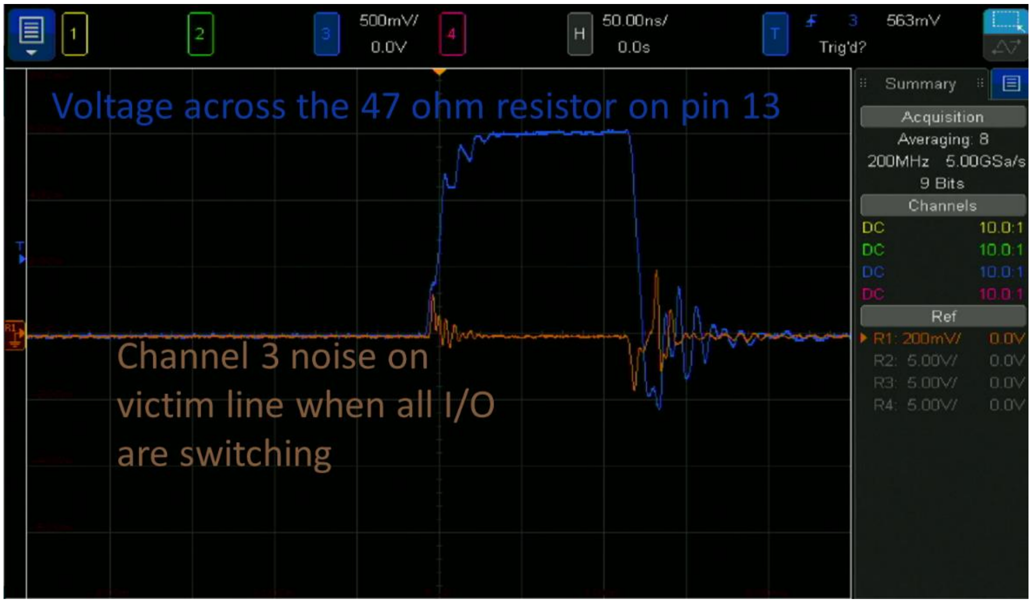
In Part 2 of the lab, use the bottom third section of the board. Follow the signal paths from the pin connections to the LEDs and the series resistors. The return connections are through the via at the end of the trace to the bottom plane. The far end of the victim line is shorted to the bottom plane 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.

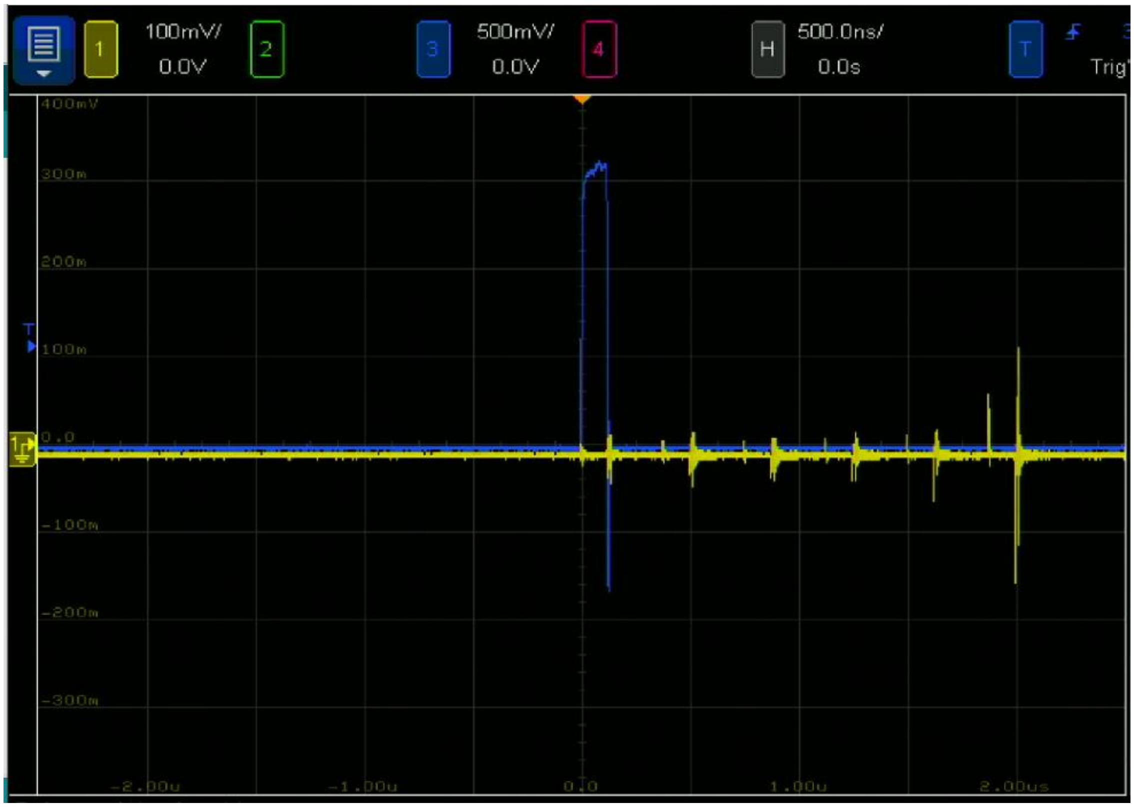
Using the voltage across the 47 ohm resistor to trigger the scope, measure the voltage on the victim trace. All measurements should be done using the spring ground tips.  Save this noise on the victim line as your reference. You will compare this noise with the other noise you measure in other configurations.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

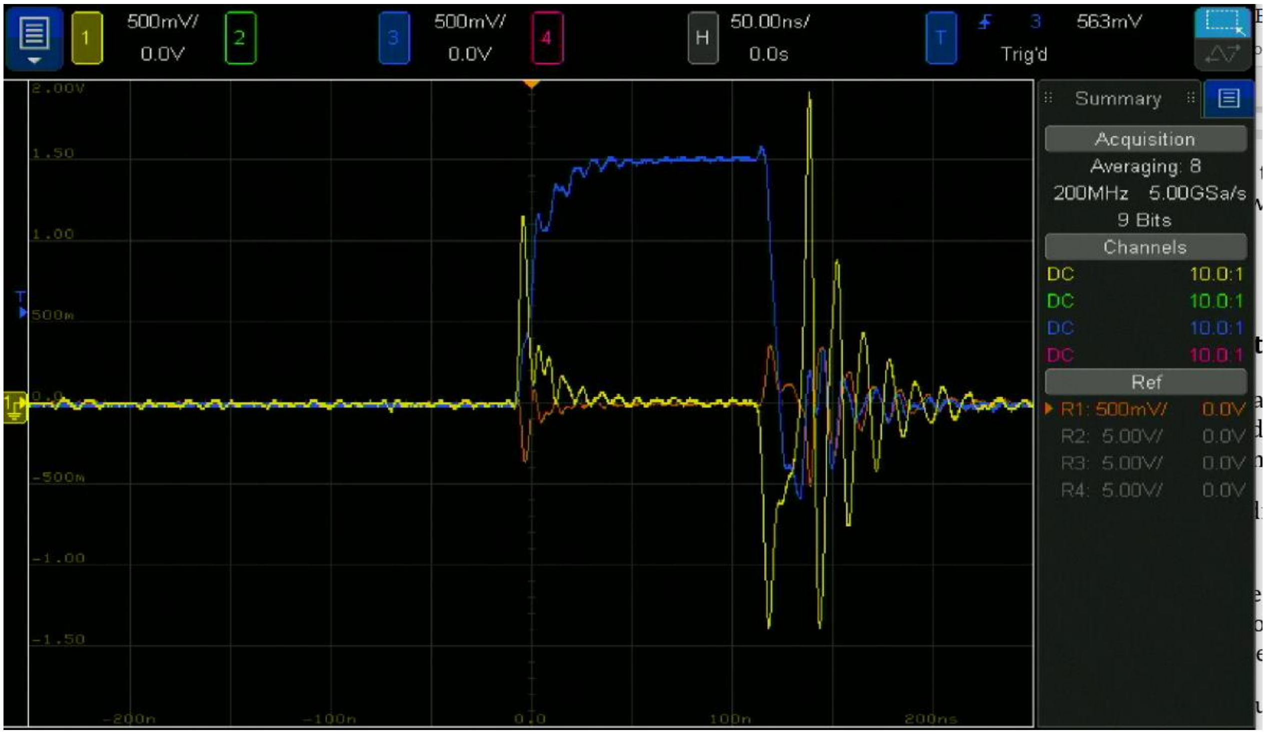
Part 3 of this lab uses the middle section of the board. Again, all the signal pins are connected to traces that span the length of the board and connect to LEDs and resistors which are all bussed together through a common return trace. There is no plane on the bottom of the board.

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?

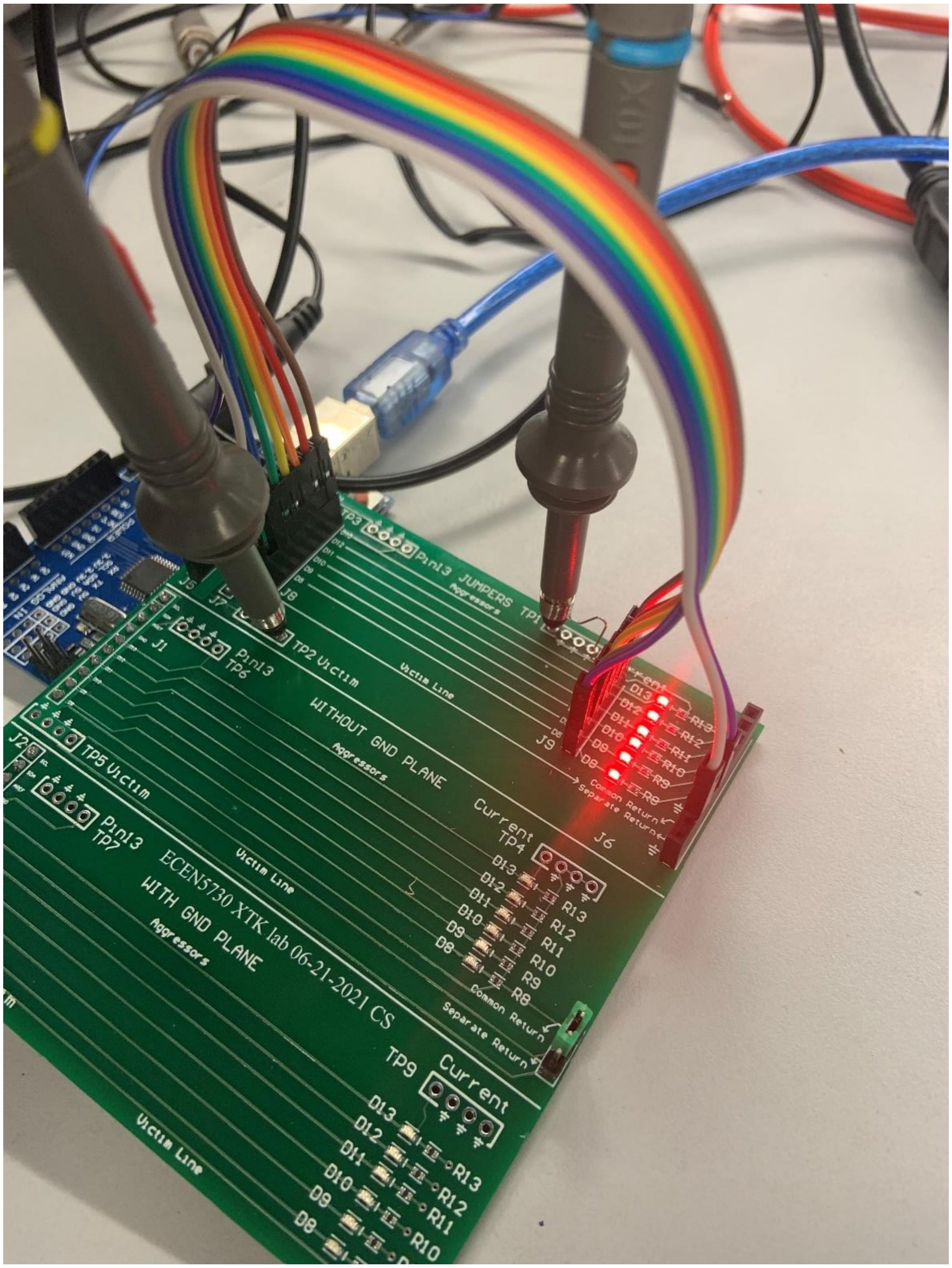
## 8.10 Part 4: Limitations of the Solderless breadboard

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.