Nora Shao_L7A

December 5, 2023

Name: Shao, Nora

Teammates at Table: [redacted]

Student Number: [redacted]

Lab Section: L7A

[60]:
```
# @title
%%capture
from IPython.display import Image
from google.colab import drive
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

drive.mount('/content/drive/')
path = 'drive/My Drive/UBC ENPH Y2/ENPH259/Lab7/'
file_name = 'Nora Shao_L7A.ipynb'

assert file_name in os.listdir(path)

def img_path(img_name):
  return path + img_name
```

1 Introduction

1.0.1 Wire colour code

- Orange: I/O from AD2
- Blue: internal connections between pins of in an IC or other circuit components OR DIO pins from AD2
- Purple: connections between different circuit components
- Red: power (5V)
- Black: GND
- Blue: $V_{CC}-$

1.0.2 Experiment Overview

In this lab, we build and a test a breadboard-based circuit that lets us control and read the speed of a servo motor via a potentiometer and the Analog Discovery 2 board. I'll be building the individual circuits clockwise on the two breadboards.

For organizational purposes, I attached the two breadboards together and connected all the power and gnd rails (separately), so just one power and gnd input from the AD2 breakout board would power all the power rails and ground the grounding rails.

[61]:
```
Image(filename = img_path("L7A_complete schematic.png"))
```

[61]:

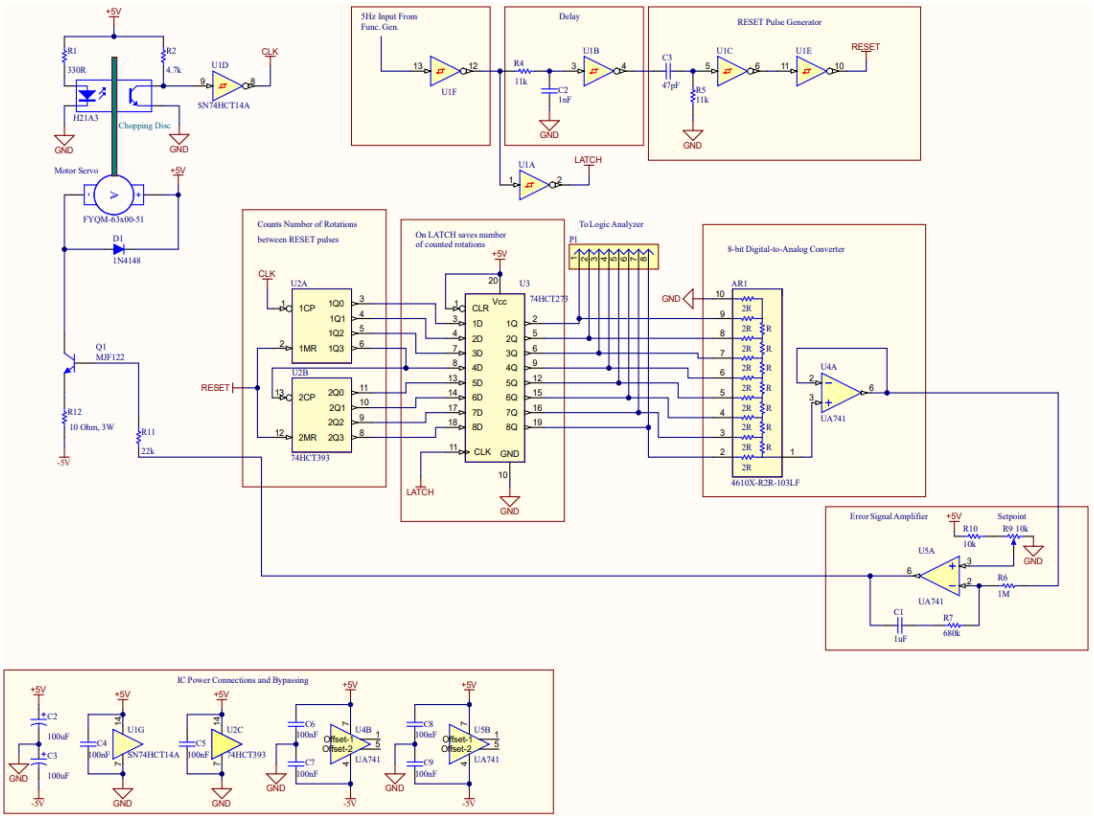


Figure 1: Functional diagram of entire circuit controlling servo motor

2 Experiment

2.1 *LATCH* and *RESET* generator

The latch takes a signal from a 5 Hz square wave input going from 0 to 5 V, and a reset pulse is generated by the series of capacitors and resistors that first create a delay from the input signal, then generates the brief reset pulse.

2.1.1 Delay Circuit

Procedure

- I'll use the 74HC14 Schmitt-Trigger Inverters as our inverters for the latch and reset generator and follow this datasheet: https://www.mouser.com/datasheet/2/308/74HC14.REV1-34947.pdf
- I built the delay and reset generator circuits around one inverter IC, visible in Figure 3
- I test the operation of the delay by inputting the square wave into the input of the U1F inverter and reading the output of the U1B inverter.
  - I used a BNC cable connected to the Wavegen 1 Output of the AD2 breakout board
  - I used BNC cables connected to the Scope 1+ and Scope 2+ inputs on the breakout board to read the output of the U1B inverter (delay circuit) and the input, respectively.

Calculations:

Note: See below in the "Understandings" section where we got the V_{T-} and V_{T+} values.

In a series circuit, we have the time constant

$$\tau = RC,$$

where R and C are respectively the resistance and capacitance of the circuit. Here, we get

$$\tau = R4 \times C2,$$

or

$$\tau = 11k\Omega \times 1nF,$$

which evaluates to $\tau = 1.1 \times 10^{-5}s$.

The signal from the delay circuit (after U1B) goes HI when the input signal to U1B drops below V_{T-}, 1.65 V.

$$V(t) = V_0 e^{-t/RC},$$

so

$$1.65V = 5Ve^{-t/\tau},$$

and

$$t = -\tau \ln(\frac{1.65}{5}),$$

and we plug in our values to get

$$t = \tau \ln(\frac{1.65}{5}).$$

We evaluate to get

$$t = 12.2\mu s.$$

Thus, we expect our delay to be around 12.2 s, since there is some variation in what our actual V_{T+} and V_{T-} values are.

Troubleshooting

#TS1: I set my input as a square wave from 0 to 5 V with am amplitude and offset of 2.5 V, but when I read the input with the scope, I saw that it was going from 0 to 2.5 V instead of 0 to 5 V. After inspecting my circuit with the TA, I found that I had wired the input of U1F to the output, which is bad, since the output state of an inverter should always be different from the input, yet I had effectively forced them to be the same state.

#TS2: I first measured a delay of ~2 s, which didn't fit well at all with my calculations, so I reinspected my circuit and found that I didn't ground the IC. After I grounded the IC, my delay was ~12 s, which made sense as it was (very) close to my expected delay.

[62]:
```
Image(filename = img_path("L7A_delay and reset schematic.png"))
```

[62]:

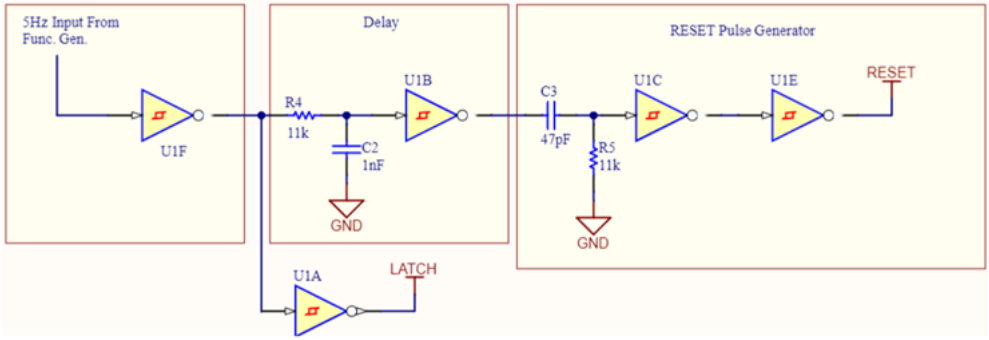
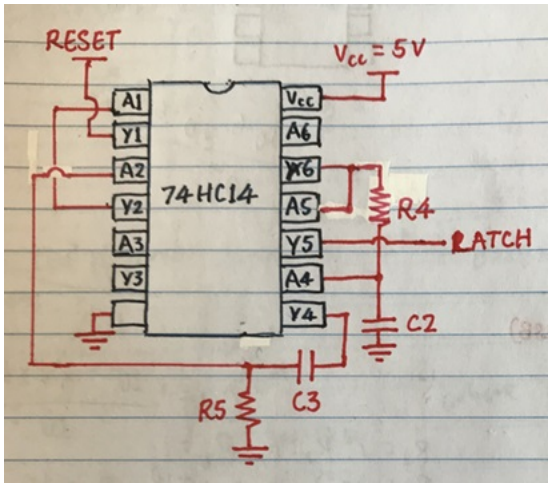


Figure 2: Function diagram of delay and reset pulse generating circuits

[63]:
```
Image(filename = img_path("L7A_inverter IC pinout.jpg"))
```

[63]:



[64]:
```
Image(filename = img_path("L7A_measuring delay.png"))
```

[64]:

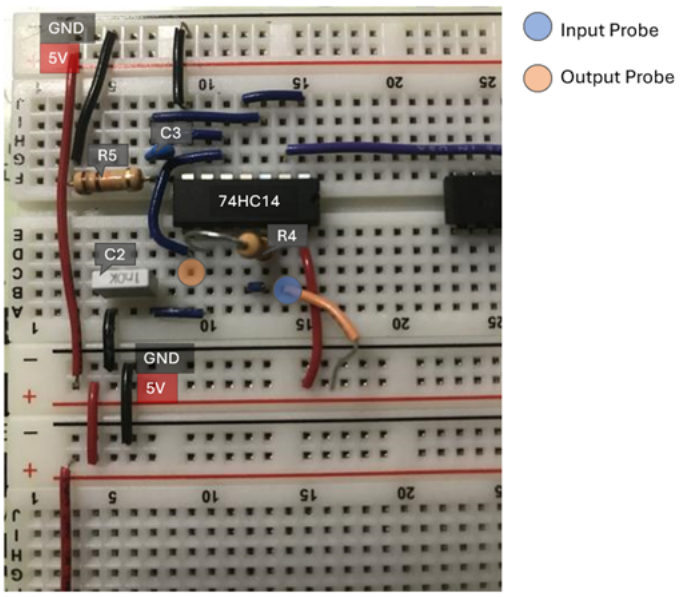


Figure 3: Setup of delay and reset circuit on breadboard for measuring delay generated by delay circuit

Results

I confirmed the delay circuit worked with a delay of ~11.65 s, which is very close to our expected delay.

Understandings

The delay circuit works by utilizing the Schmitt Trigger's inverting operations and the mini RC circuit made up of R4 and C2. The input signal must increase past a certain threshold, V_{T+}, and decrease past a different threshold, V_{T-} for the inverter to invert the signal to LO and HI, respectively. From the Schmitt-Trigger datasheet, we see in Figure 3 that there's a range of values for V_{T+}, between 2.6 and 3.5 V. Taking the middle of the range for a rough value for V_{T+}, we get 3.05 V. The datasheet tells us V_{T-} is between 1.3 and 2 V, which we take the middle of to get a rough value of 1.65 V.

Although the input signal changes abruptly, the capacitor slows down the growth/decay of the signal from the output of U1F such that there is a delay when the voltage at the input of U1B increases past or decreases below V_{T+} and V_{T-}. Looking closer at how the delay works, in Figure [], we see what happens to the input signal into U1B when the input square wave goes HI, and it does take approximately 12 s for the signal to decay to the V_{T-} we're assuming. Now, we switch to look at U1B's output, and see that it indeed takes around 12 s to go HI, but the V_{T-} is actually around 1.75 V instead of the 1.65 V we initially assumed.

[65]:
```
Image(filename = img_path("L7A_schmitt trigger threshold voltages.png"))
```

[65]:

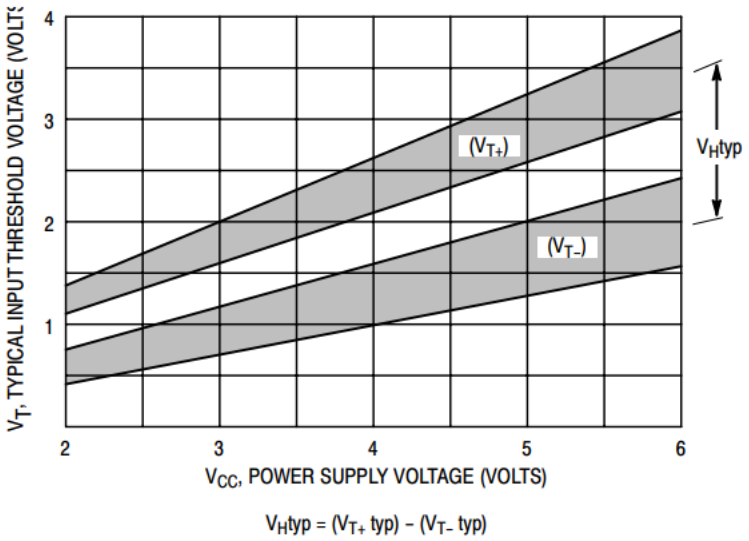


Figure 3. Typical Input Threshold, V_{T+}, V_{T-} versus Power Supply Voltage

Figure 4: Range of V_{T+} (negative switching voltage) and V_{T-} (positive switching voltage) values for Schmitt-Trigger Inverter depending on supply voltage

[66]:
```
Image(filename = img_path("L7A_zoomed out delay.png"))
```

[66]:

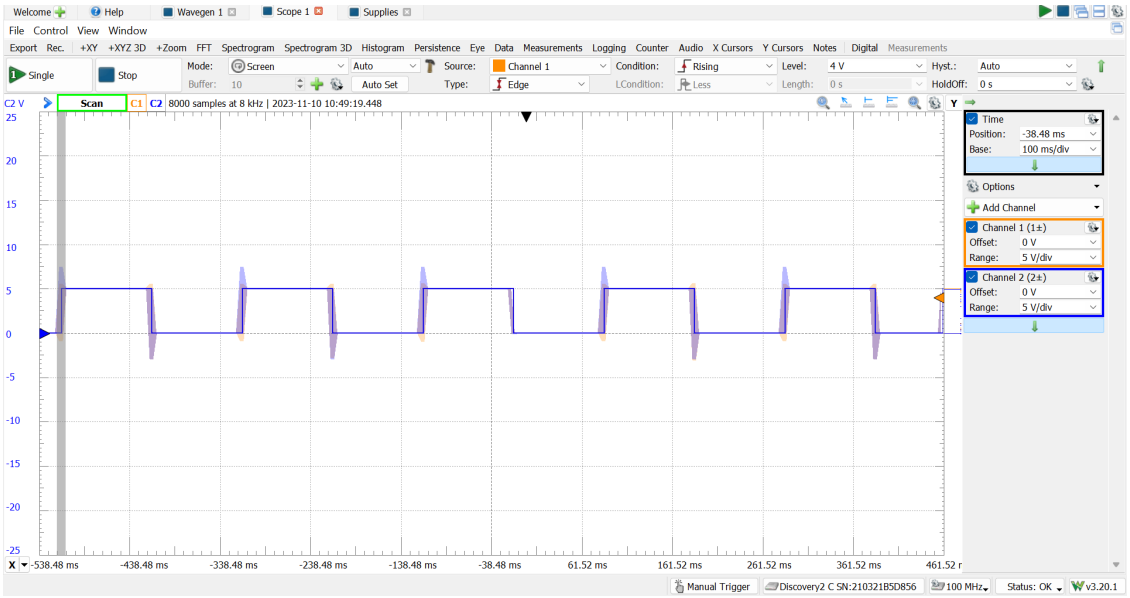
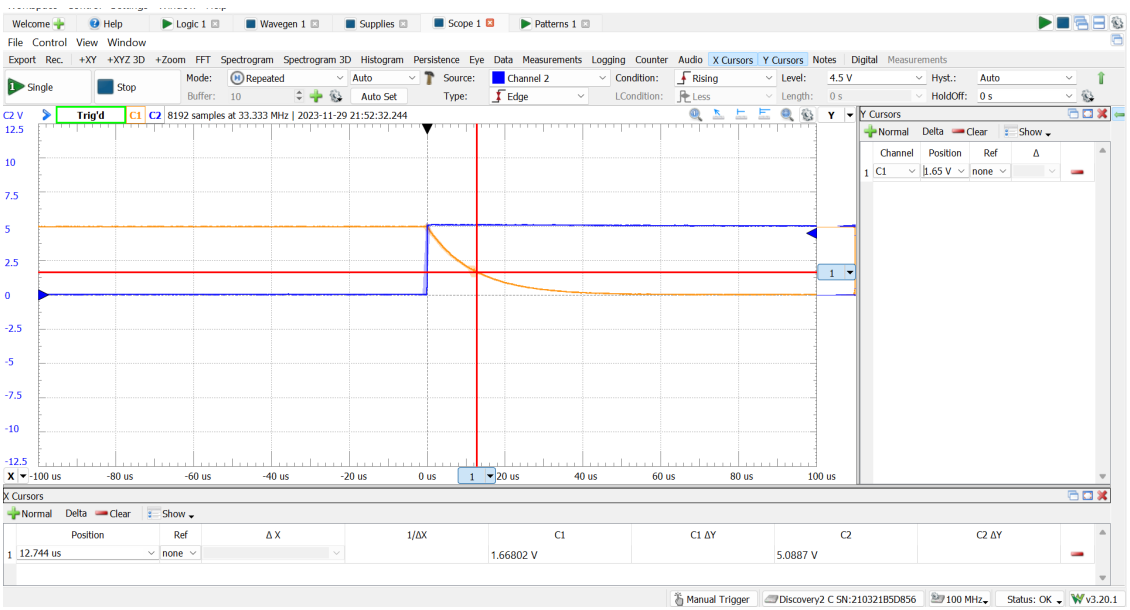


Figure 5: Zoomed out reading of delayed signal (output of U1B; CH1) compared with input signal (CH2)

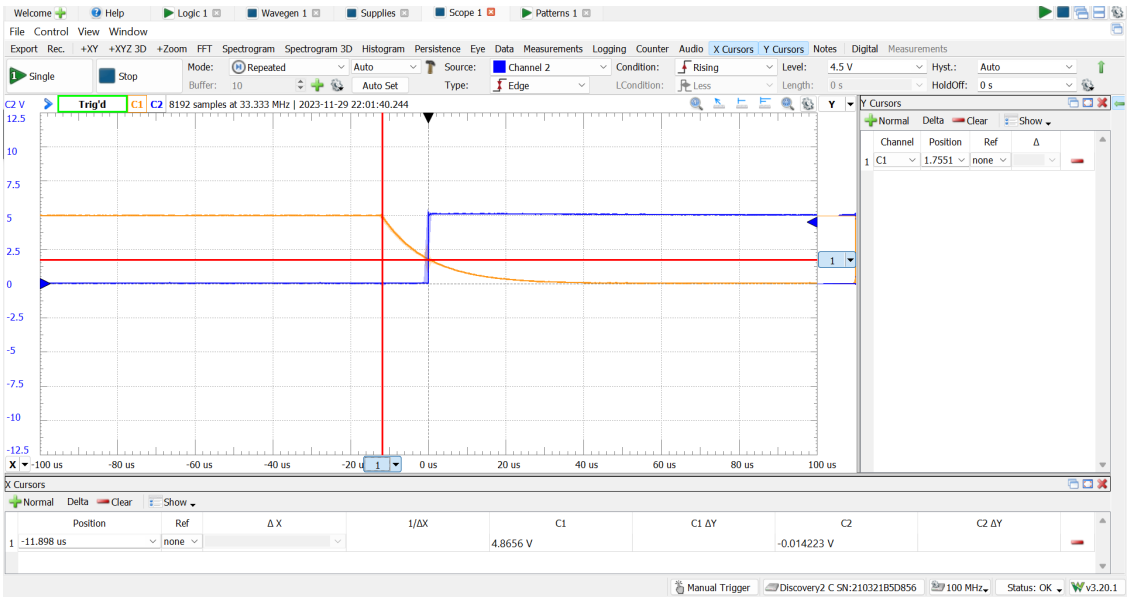
[67]:
```
Image(filename = img_path("L7A_C2 discharging.png"))
```

[67]:



[68]:
```
Image(filename = img_path("L7A_C2 with delay.png"))
```

[68]:



[69]:
```
Image(filename = img_path("L7A_delay measurement.png"))
```

[69]:

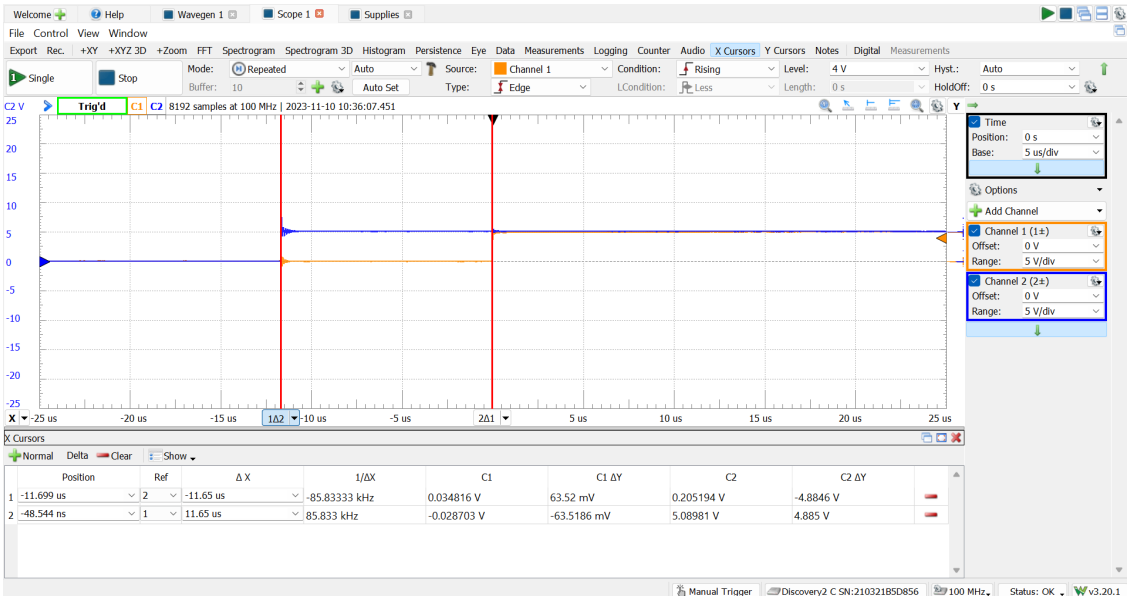


Figure 6: Measurement of delay from rising edge of output signal of delay circuit (CH1) and input signal (CH2)

2.1.2 RESET Pulse Circuit

Procedure

- I set up the RESET circuit on the breadboard around the IC as seen in Figure 7
- I kept the 5 Hz square wave oscillating from 0 to 5 V at the input of the IC

Troubleshooting

#TS3: When I tried to measure the RESET signal output, I measured nothing even though I was inputting the 5 Hz square wave. I moved my output signal scope measurement to trace along the circuit, ensuring the signal was what I expected at every point. I found that past C3 in Figure 2, the signal flatlined at 0. Then I zoomed in significantly on the time scale and realized that the RESET signal pulsed very very briefly before going LO for the rest of the period. However, the delay from the rising edge of the input square wave to start of the pulse was around 2 s, when it should have been 12 s.

[70]:
```
Image(filename = img_path("L7A_measuring reset pulse.png"))
```

[70]:

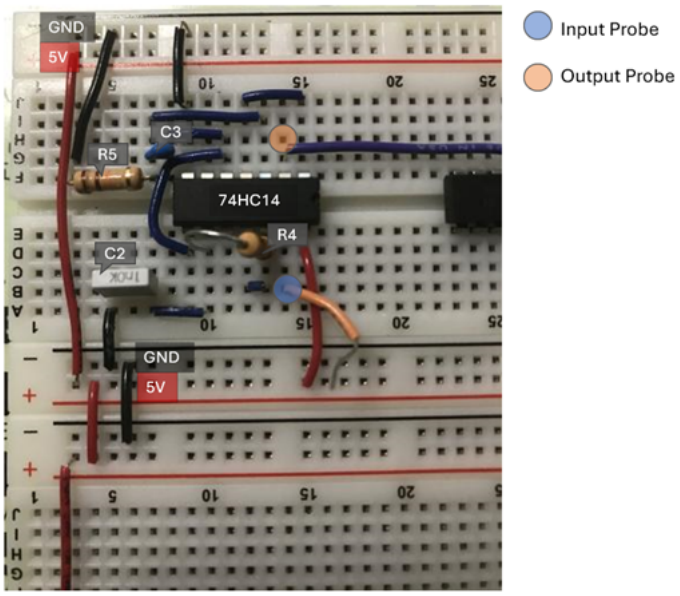


Figure 7: Setup of delay and reset circuit on breadboard for reading the reset pulse generated

Results:

Just qualitatively, the reset pulse visible in Figure 8 makes sense as, well, it's a brief pulse. It has the same magnitude as the input signal 5 V, and is delayed by the delay circuit's ~12 s.

Understandings:

The delay circuit results in the delay in the RESET pulse we see in Figure 8. This is further confirmed by the fact that the delay from the rising edge of the input signal to the rising edge of the RESET pulse is also around 12 s. In fact, it's approximately 11.72 s, which is very close to the 11.65 s delay measured directly from the delay circuit earlier.

Analyzing the behaviour of the circuit between U1B and U1C explains the shape of the reset pulse. When the input signal from the AD2 has been HI for a long time, the output of U1B is at 5 V, and the capacitor C3 will be an open circuit, so the input to U1C is LO (0 V), and its output is 5 V. Thus, RESET will be LO. Then, the input signal switches to LO, and at this instant, since there is still 5 V across C3, the input to U1C is -5 V, so the output stays the same (since the input is still below V_{T-}). However, the capacitor will discharge and eventually the input to U1C will go to GND (0 V, or LO). Then, the input signal switches to HI again. Now, briefly, the capacitor will act as a short-circuit, so the input to U1C will be above ~3 V, V_{T+}, and in that brief time before the capacitor charges and the input to U1C decays below V_{T+}, U1C will output LO, so RESET is HI, hence the brief pause in the HI state before the cycle restarts and RESET goes LO again.

[71]:
```
Image(filename = img_path("L7A_reset pulse.png"))
```

[71]:

Figure 8: Reset pulse generated by reset circuit (CH1) compared with input signal (CH2)

[72]:
```
Image(filename = img_path("L7A_reset pulse measurements.png"))
```

[72]:

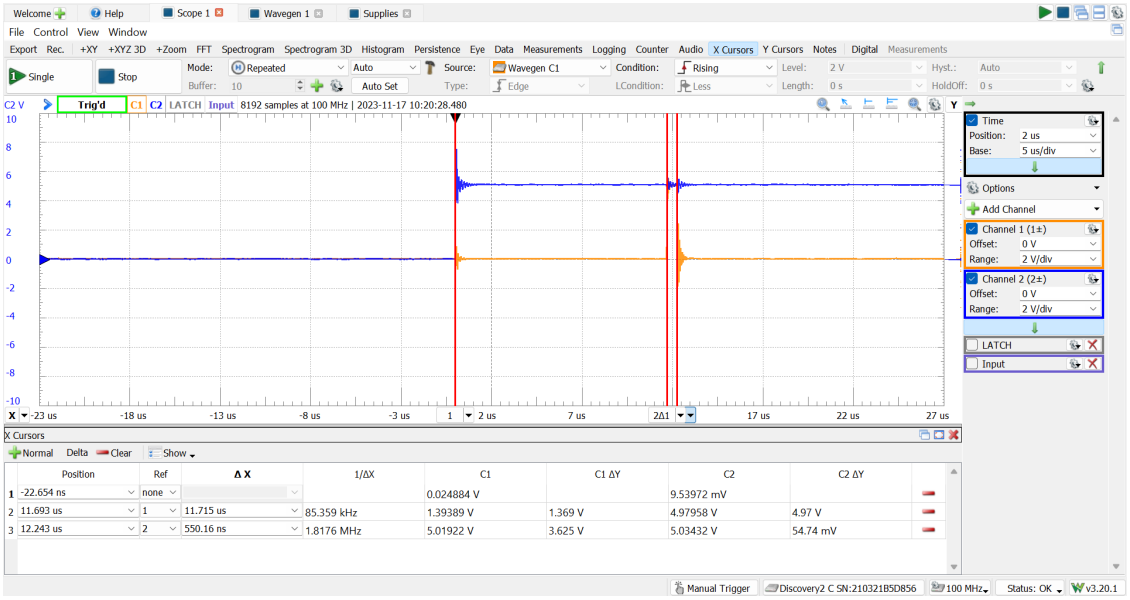


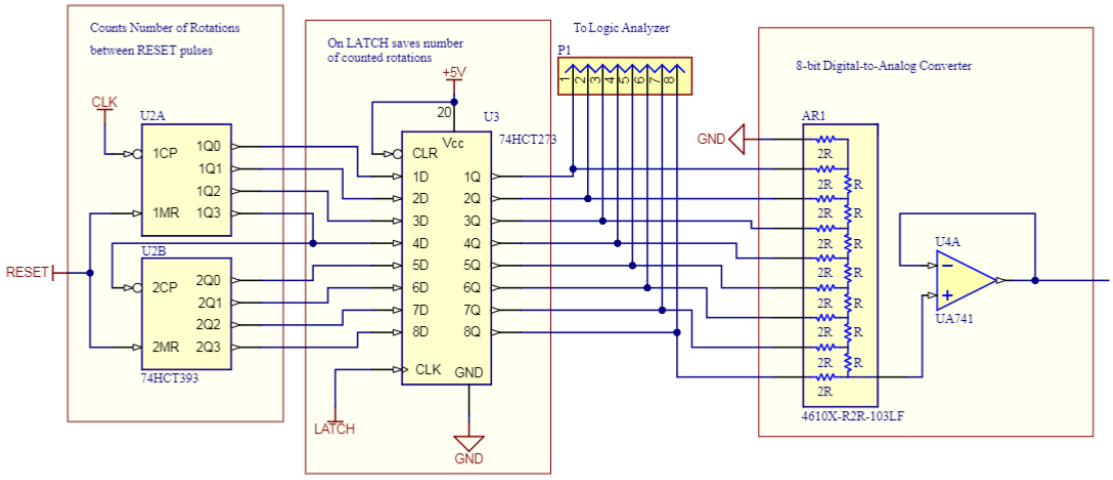
Figure 9: Measuring the delay between the rising edge of the reset pulse and the input signal and the width of the reset pulse

2.2 Counter, D-latch, DAQ

blah blah

[73]:
```
Image(filename = img_path("L7A_counter latch dac.png"))
```

[73]:



2.2.1 Counter

Procedure

- I followed this datasheet for the dual 4-bit 74HCT393 counter IC: https://www.jameco.com/Jameco/Products/ProdDS/243205.pdf
- On WaveForm's Logic app, I set up DIO pins 0-7 in the BUS signal form, with 7 as the Most Signficant Bit (MSB) and DIO 0 as the Least Significant Bit (LSB) mapped as seen in Figure 10
- I used WaveForm's Patterns app to set up the clock and reset functions of the counter IC.
  - For the RESET pin, I connected it to DIO 8 and set it to be constantly low (so I won't get any unwanted resetting of the counter)
  - For the CLK pin, I connected it to DIO 9 and set it as a 1 Hz clock, so I could watch the counter increase value at a decent pace.
- When determining the behaviour of the counter past its max value
  - I configured Logic to display the output from the counter in decimal form, so it was easier for me to understand the values
  - I adjusted the clock frequency to 10 Hz so it would reach the max value possible and over flow quicker

Troubleshooting

TS: My counter was acting weird, randomly jumping between decimal values. But, once I configured the value to display in decimal instead of binary form, I saw the digits of the binary value increasing from left to right, and not in a linear fashion, so I realized the endian of the diodes was reversed. I had set DIO 7 as the Least Signficant Bit and DIO 0 as the Most Signficant Bit when it should have been the other way around. I fixed this by reversing the endian of the diodes in the WaveForms Logic Bus window.

[74]:
```
Image(filename = img_path("L7A_counter IC pinout.jpg"))
```

[74]:

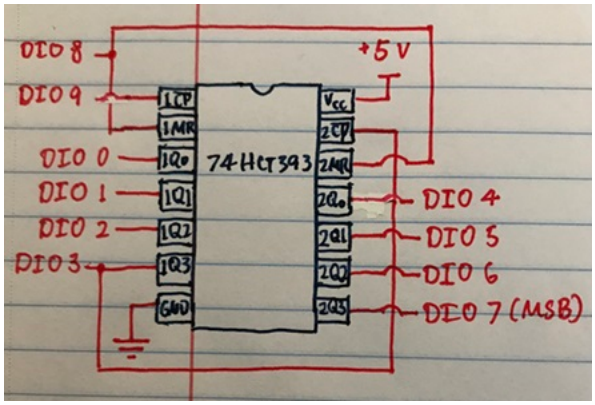


Figure 10: IC pinout diagram of counter for testing operation of counter

[75]:
```
Image(filename = img_path("L7A_counter pinout setup.png"))
```

[75]:

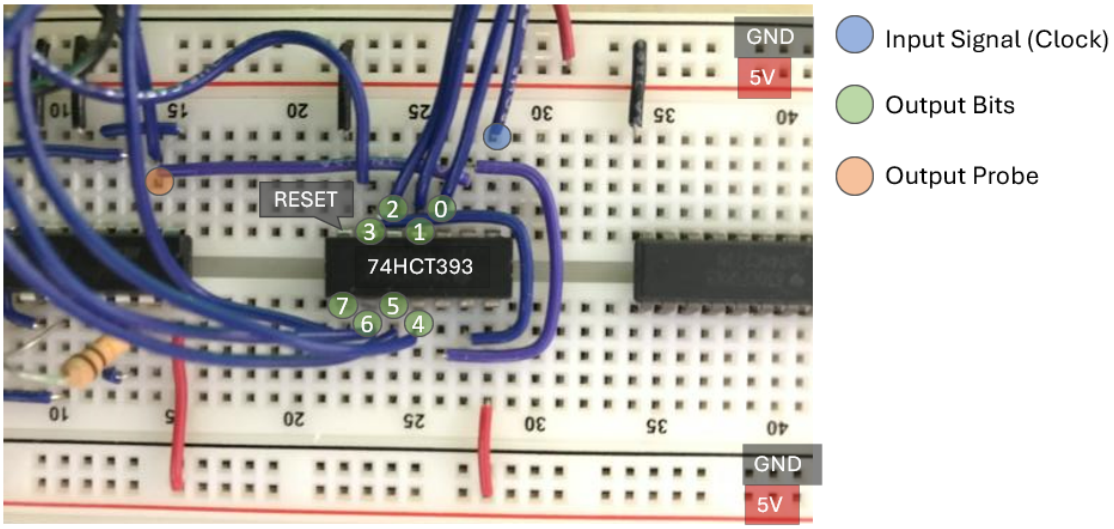


Figure 11: Setup of counter IC wired to test operation on breadboard

[76]:
```
Image(filename = img_path("L7A_counter logic setup.png"))
```

[76]:

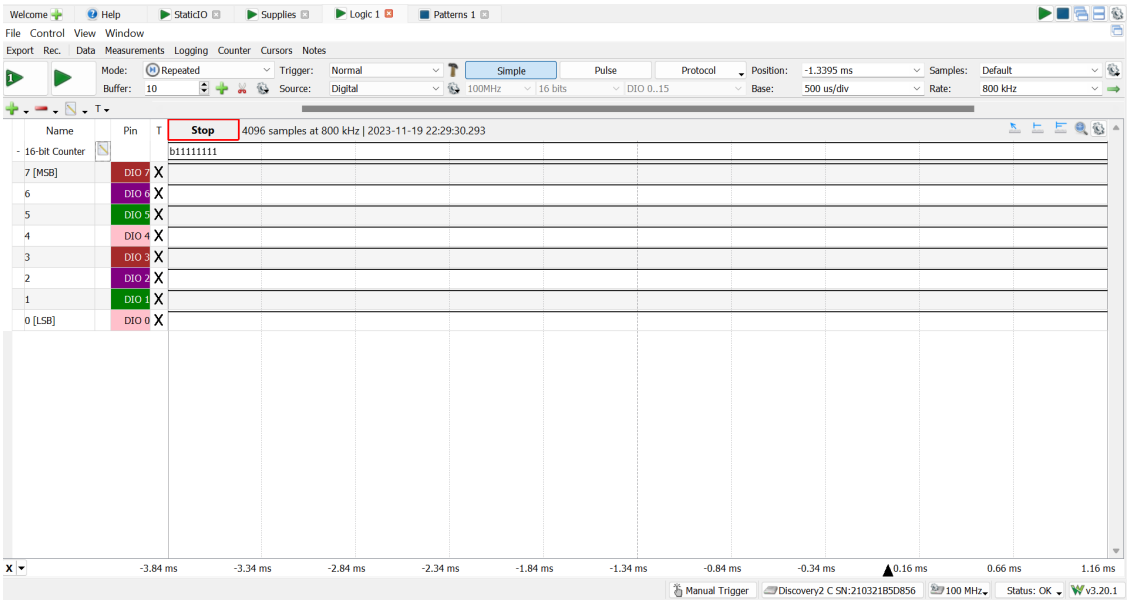


Figure 12: Setup of DIO output pins on Logic to read output (value) from counter

Results

The counter worked as expected, where each bit of increasing significant would switch states (from LO to HI or vice versa) at frequencies decreasing by a factor of 2.

At a binary value of 11111111, or 255 in decimal form, the counter overflowed and just reset back to 00000000 (0 in decimal form).

Understanding:

Since we connected two 4-bit counters, when the individual counters can count to 2^4, 15, the now 8-bit counter can count to 2^8, 255 (since the counter starts at 0, not 1). Hence, at 11111111, or 255, the counter overflowed or went back to 0.

When integrated with the circuit, this counter is responsible for measuring the servo motors speed by counting the number of partial turns it makes - this will be explained in further detail in the Motor Sensor and Driver section.

[77]:
```
Image(filename = img_path("L7A_counter operation.png"))
```

[77]:

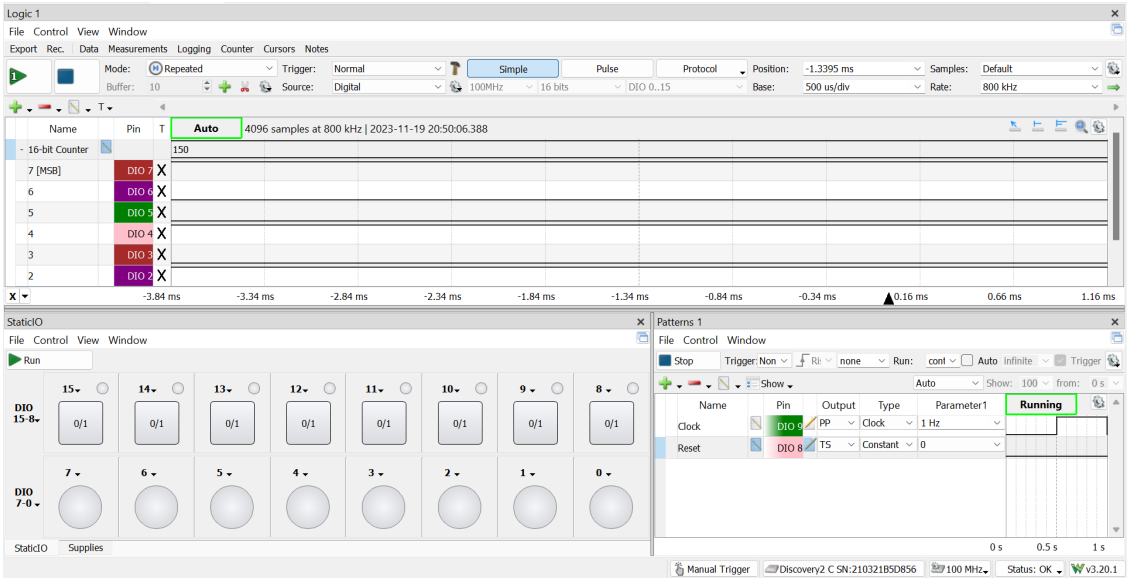


Figure 13: Working counter at decimal value of 150 with clock and reset inputs visible on bottom right

2.2.2 LATCH Circuit

Procedure

- I followed the datasheet for the 74HCT273 octal flip-flop here: https://assets.nexperia.com/documents/data-sheet/74HC_HCT273.pdf
- This time I used th AD2's AWG instead of WaveForms' Patterns to generate the input pulses
  - I connected the Counter IC's clock terminal to the breakout board's WaveForm Gen 1
  - I connected the input terminal of the latch and reset pulse generator to the breakout board's Waveform Gen 2
- I played with the frequencies of the clock input into the counter and the clock input to the latch.

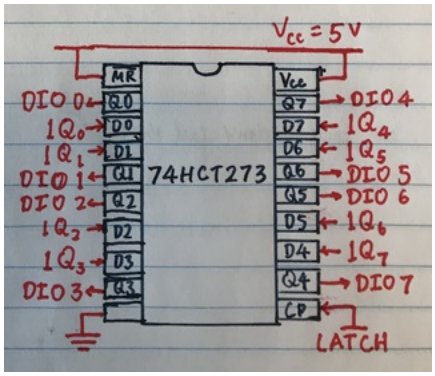
Troubleshooting

#TS I used the wrong pinout from an incorrect datasheet. :(Luckily, before I actually did any testing, the TA informed us about this

#TS I couldn't understand why []. With Professor Jones's help, I realized that I couldn't set the counter's clock input, and the reset and latch input to the same to the same frequency, since that meant the counter effectively reset itself at the same rate it counted. Thus, I tried inputting the clock at frequencies significantly higher than the 5 Hz of the reset and latch clock signals, which fixed this issue.

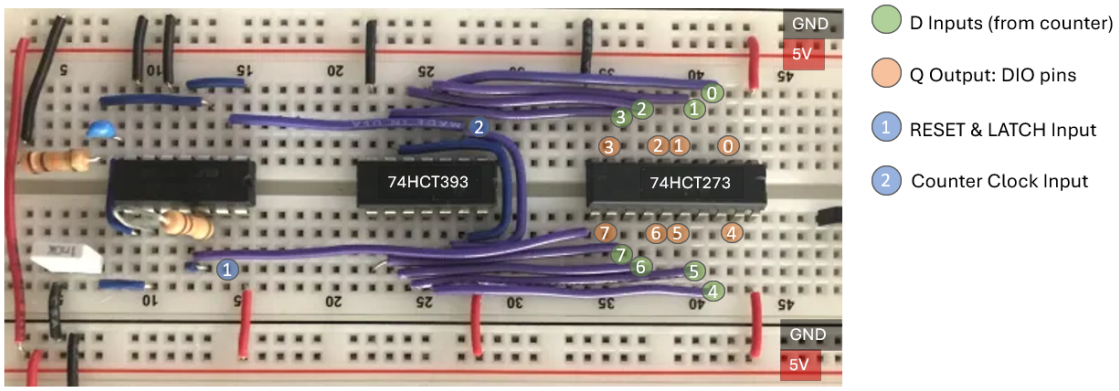
[78]:
```
Image(filename = img_path("L7A_latch IC pinout.jpg"))
```

[78]:



[79]:
```
Image(filename = img_path("L7A_testing latch setup.png"))
```

[79]:



[80]:
```
Image(filename = img_path("L7A_testing latch outputs.png"))
```

[80]:

Results

Initially, I only saw the value the DIO 0 pin was reading change; in Figure [], only the signal from DIO 0 fluctuated up and down; the rest stayed in the LO state. After I fixed this in the TS[] described above, I found that varying the frequency of the signal I input into the clock's changed the value the latch read from the counter and outputted, but for one frequency, this value stayed constant.

In Table 1 in the DAC section, I explore the relationship between the counter's clock frequency and the latch's output value (the value it reads from the counter) further.

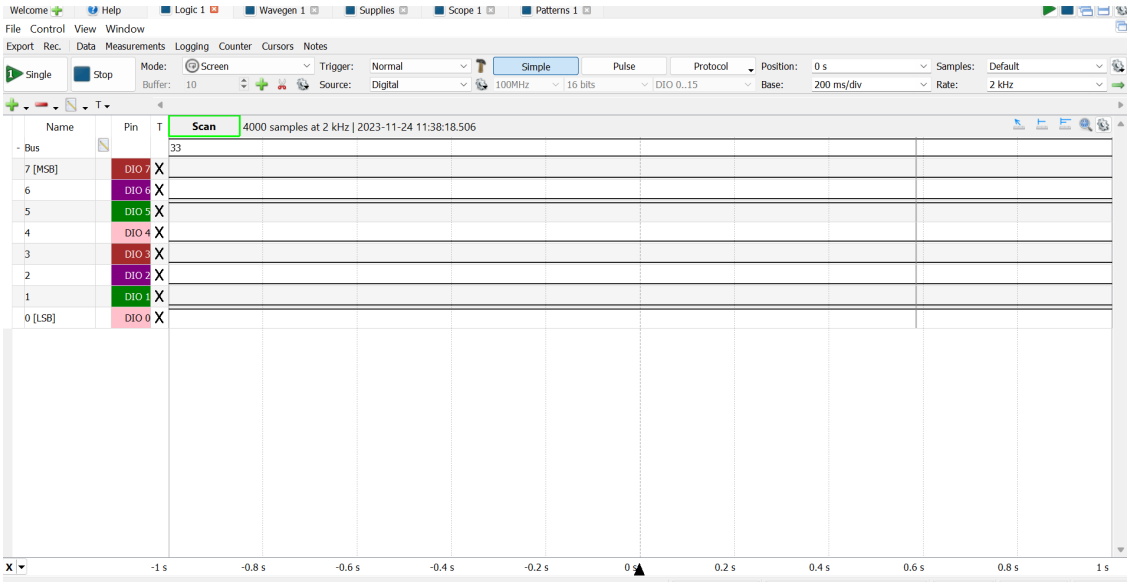
Understanding

Since I also set up the latch Q outputs with WaveForm Logic's bus feature in the decimal form with the same endian I configured the counter to count at, the latch would just output the decimal value the counter was counting to before it reset.

The relationship between the frequency of the RESET and LATCH signals and the counter clock means that the counter value and output of the latch reset at the same rate (the RESET and LATCH signal frequency), and the counter output values count up in the time between the resets, which doesn't change, thus the latch always records one value for a counter clock frequency.

[81]:
```
Image(filename = img_path("L7A_latch q values with 330 Hz clock.png"))
```

[81]:



2.2.3 DAC

Procedure

- Datasheet for resistor network DAC: https://www.mouser.ca/datasheet/2/54/r2r-1018811.pdf
- Datasheet for op-amp: https://www.ti.com/lit/ds/symlink/ua741.pdf
- I'm not entirely sure what the negative voltage supply for the op amp should be, but considering we expect all the voltage outputs to be

Troubleshooting

#TS: Initially, I had the DAC set up on one side of the gap on the breadboard and the wires connecting the bits from the latch to the DAC via the connectivity of the breadboard's columns, but I read nothing from the output terminal of the resistor network. Then I realized columns across the gap in the breadboard are not connected. Once I bridged the gap with wires, I read expected values from the DAC

Results

I tested the output of the DAC with varying input frequencies to the counter's CLK terminal, and found that the DAC output was indeed proportional (somewhat) to the output value of the latch/counter and thus clock frequency.

Table 1. Counter and DAC output with varying counter clock frequencies

Clock Frequency	Counter Output	DAC Output
165 Hz	34	0.65 V
330 Hz	64	1.27 V
660 Hz	132	2.52 V

Clock Frequency	Counter Output	DAC Output
900 Hz	180	3.42 V
1000 Hz	192	3.72 V
1320 Hz	0	0.182 V

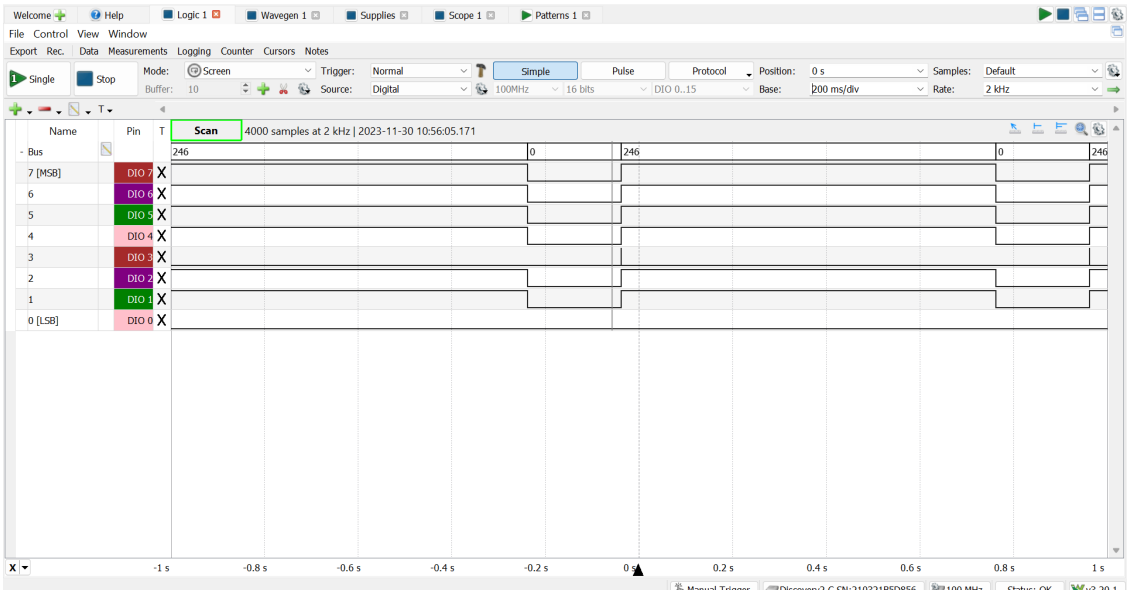
Understandings

Since the counter and latch reset at a frequency of 5 Hz, and I wanted to test the maximum value the DAC could read, I multiplied 5 by 255, the maximum value the counter could counter to, which gave me 1275 Hz, the frequency I set the counter CLK input to.

Since the op amp read 3.7 V, its upper voltage output bound, I knew that the latch succesfully outputted the maximum voltage.

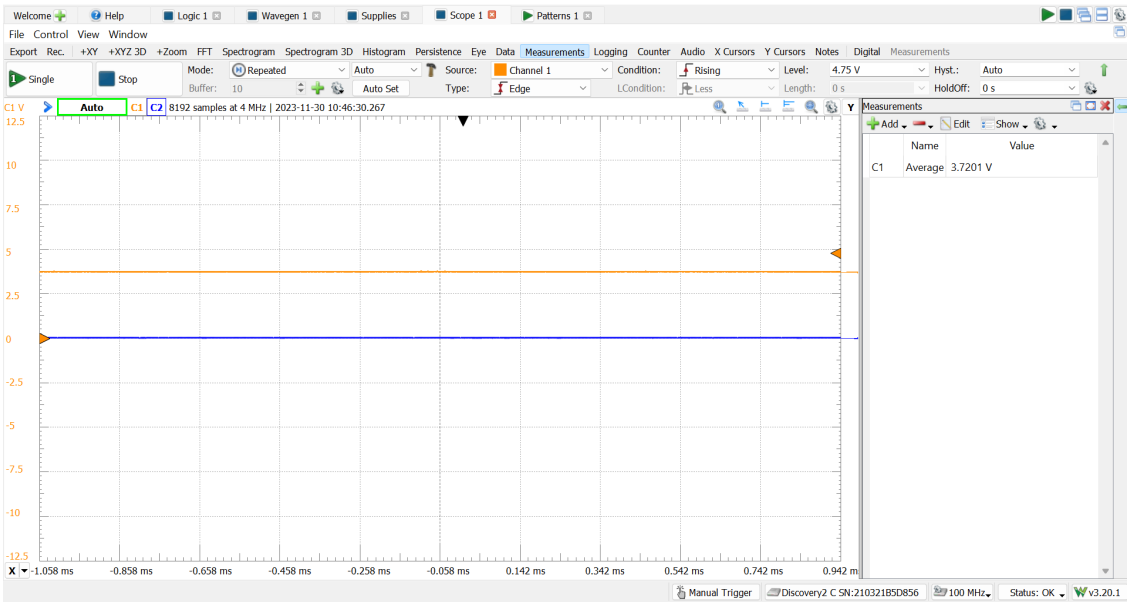
[82]:
```
Image(filename = img_path("L7A_12.76 kHz counter.png"))
```

[82]:



[83]:
```
Image(filename = img_path("L7A_max op amp output at 12.5 kHz.png"))
```

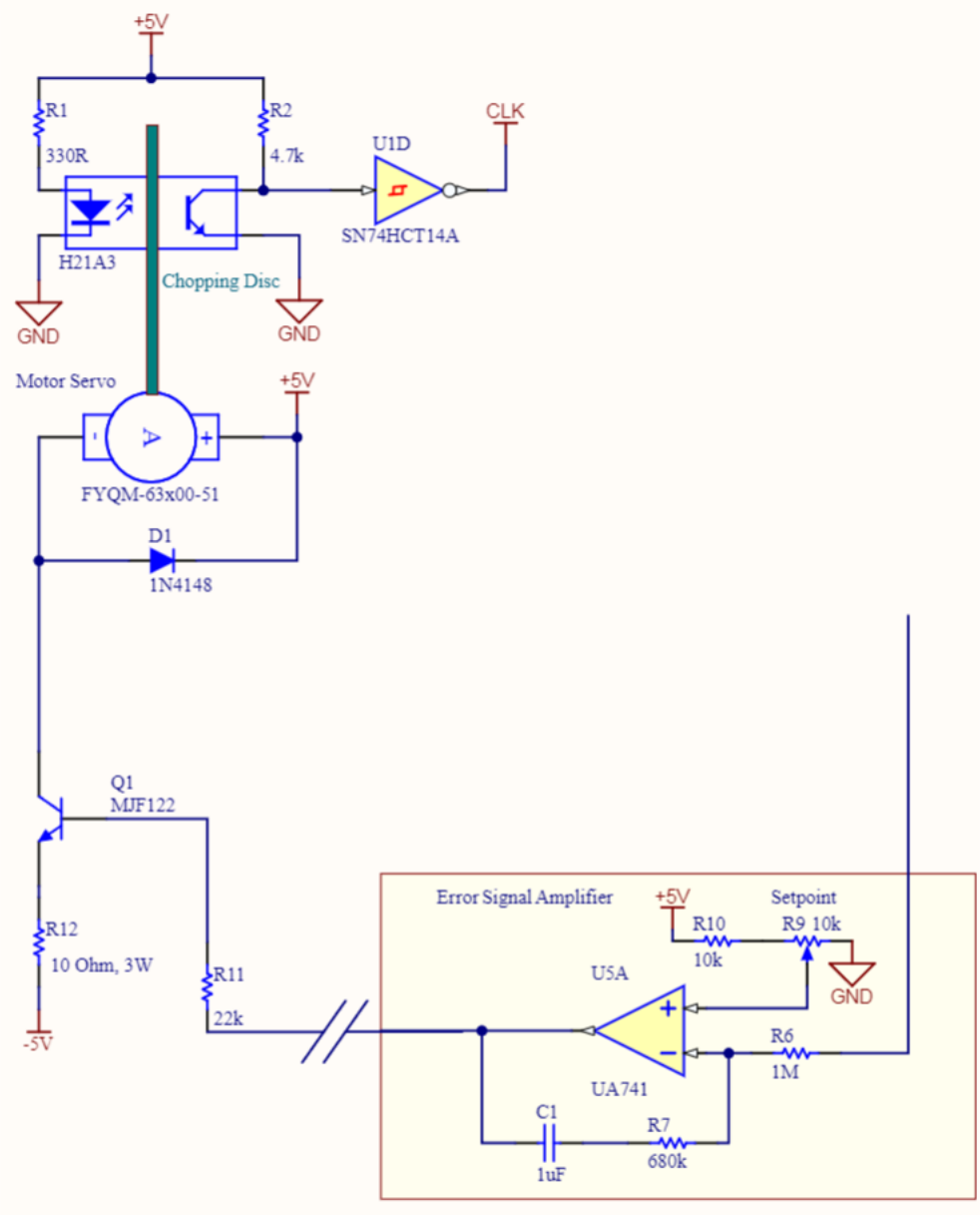
[83]:



2.3 Motor Sensor and Driver

[84]:
```
Image(filename = img_path("L7A_esa and motor schematic.png"))
```

[84]:



2.3.1 Error Signal Amplifier

Derivation of output voltage as function of input voltage

If we have V_{in} as the signal the DAC outputs, from Figure [], we decide that the current flows from V_{in} to V_{out}, so we have current

$$I = \frac{V_{in} - V_{+}}{R_6}.$$

Via nodal analysis, we realize that

$$V_+ - IR_7 - V_C = V_{out}.$$

Since we have the current through a capacitor,

$$I = C\frac{dV_C}{dt},$$

we know

$$V_C = \frac{1}{C_1}\int I(t)\ dt.$$

Plugging these back into our nodal equation, we get

$$V_+ - \frac{V_{in} - V_+}{R_6}R_7 - \frac{1}{C_1}\int \frac{V_{in} - V_+}{R_6}dt = V_{out},$$

which we can rearrage to

$$V_{out} = \frac{R_6 + R_7}{R_6}V_+ - \frac{R_7}{R_6}V_{in} - \frac{1}{C_1 R_6}\int V_{in} - V_+ dt$$

Troubleshooting

#TS: I realized my potentiometer was broken

#TS: The op-amp was railing

Results

Understandings

The error signal amplifier is integral (no pun intended) to enabling us to adjust the motor's speed.

We have some

$$E(t) = V_d(t) - V_c(t)$$

2.3.2 Motor Control with BJT - Manual Motor Test

Procedure

- Datasheet for phototransistor used in motor: https://www.farnell.com/datasheets/13244.pdf

Troubleshooting

TS:

Understandings

The motor counts how many revolutions it has turned by using a phototransistor to keep track of how many times one of the holes in the motor disk passes through the phototransistor housing, switching its state from off to on, or LO to HI. Although the number of these pulses doesn't

correspond exactly to the number of full motor revolutions, they are related by a constant of proportionality. We can understand the number of full rotations, R, as

$$R = \frac{C}{\text{number of holes on disk}},$$

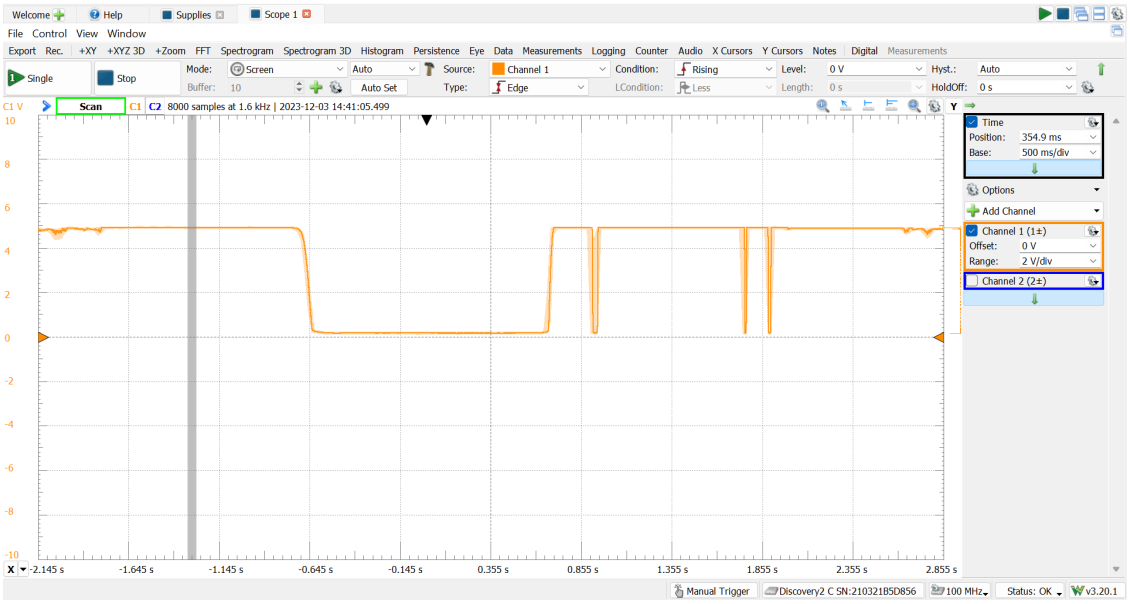
where C is the number of pulses the phototransistor outputs.

We can see that the phototransistor is connected to a transistor connecting 5 V to gnd, so when it goes HI, the transistor closes and connects the 5 V and R2 to gnd, so the input signal to the U1D inverter goes LO, and the inverter outputs HI. When the hole fully passes the phototransistor, the phototransistor outputs LO again, so the transistor opens and input signal to the U1D inverter is 5 V again, so the U1D inverter outputs goes from HI to LO. This signal is connected to the counter's CLK terminal, and since the counter is HI-to-LO edge-triggered, the counter increments once when the hole fully passes the housing.

We see in Figure [] the voltage output of the motor going low when the hole passes into the housing since the phototransistor going HI connects the output to GND. Otherwise, the output is connected to 5 V (there's a resistor, R2, but no current, so it's still 5 V)

[85]:
```
Image(filename = img_path("L7A_manual motor test.png"))
```

[85]:



2.3.3 Motor Control with BJT - Running Motor

Procedure

- I removed the grounding wire from the unused inverter terminal on the inverter IC

2.3.4 Motor Control with BJT - Actual Motor

[86]:
```
# @title
%%capture
!apt-get install texlive-xetex texlive-fonts-recommended␣
  ↪texlive-latex-recommended texlive-plain-generic pandoc
```

[ ]:
```
# @title
# Capture to prevent lots of output... Remove this if troubleshooting!
%%capture

pdf_file_name = file_name.split('.')[0]+'.pdf'  # Same as file_name with .ipynb␣
  ↪changed to .pdf
!rm $file_name
!rm $pdf_file_name

import os

full_path = os.path.join(path, file_name)
!cp  "$full_path" ./

!jupyter nbconvert "$file_name" --to pdf
```