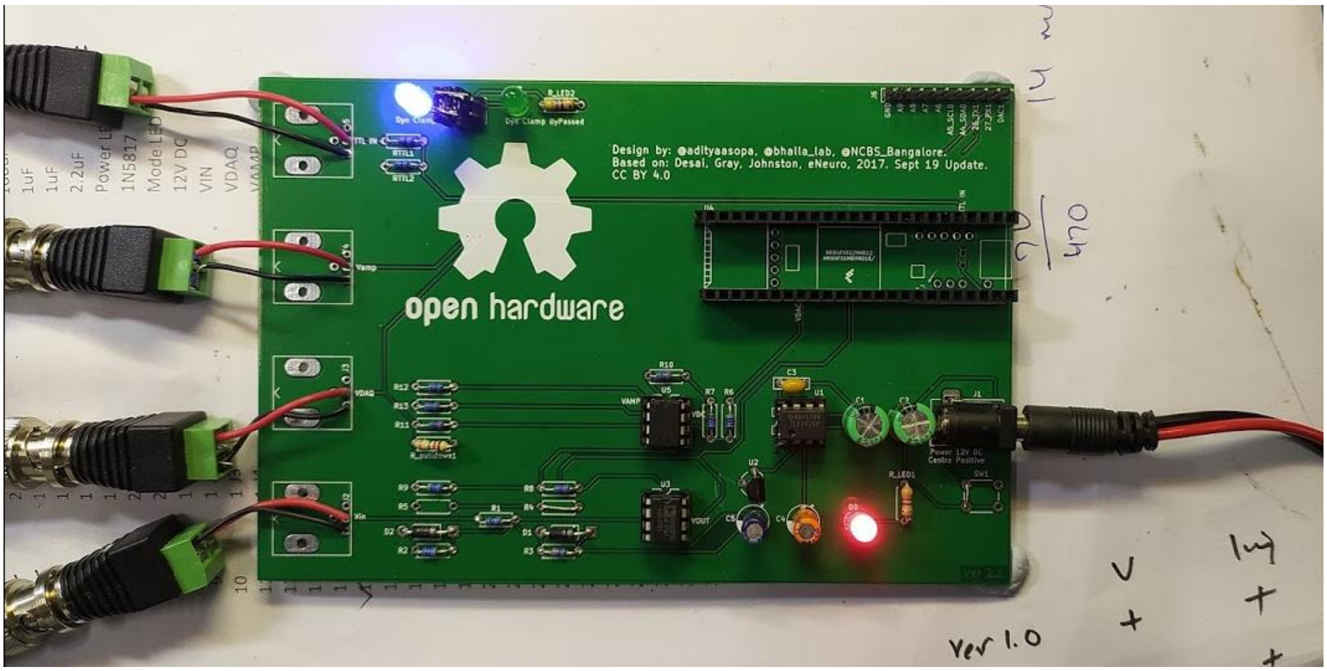
# Dynamic Clamp 2.0: Calibration and Setup

Aditya Asopa, Bhalla Lab, NCBS

Oct 2020



# General Points:

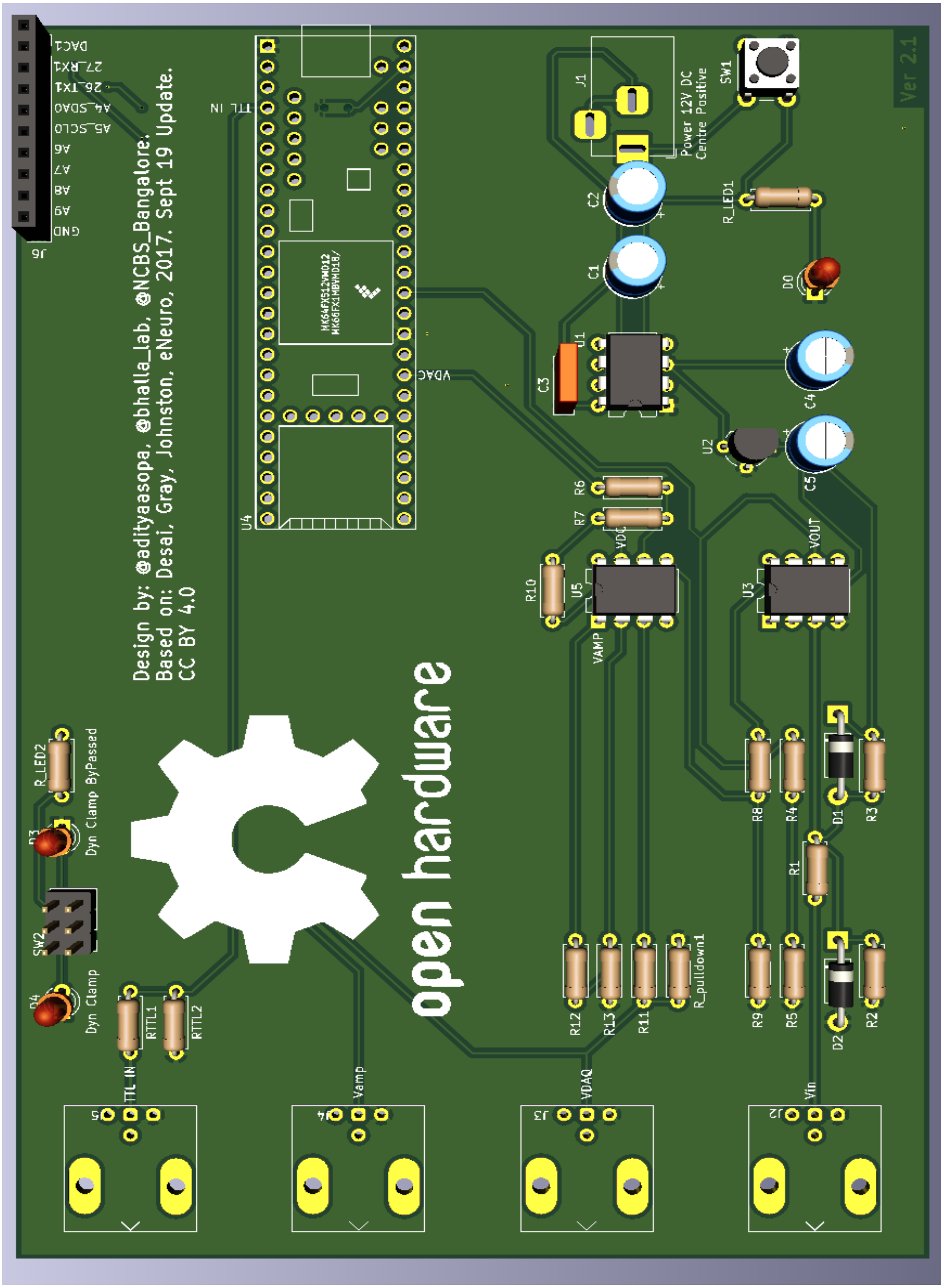
## Testing the board

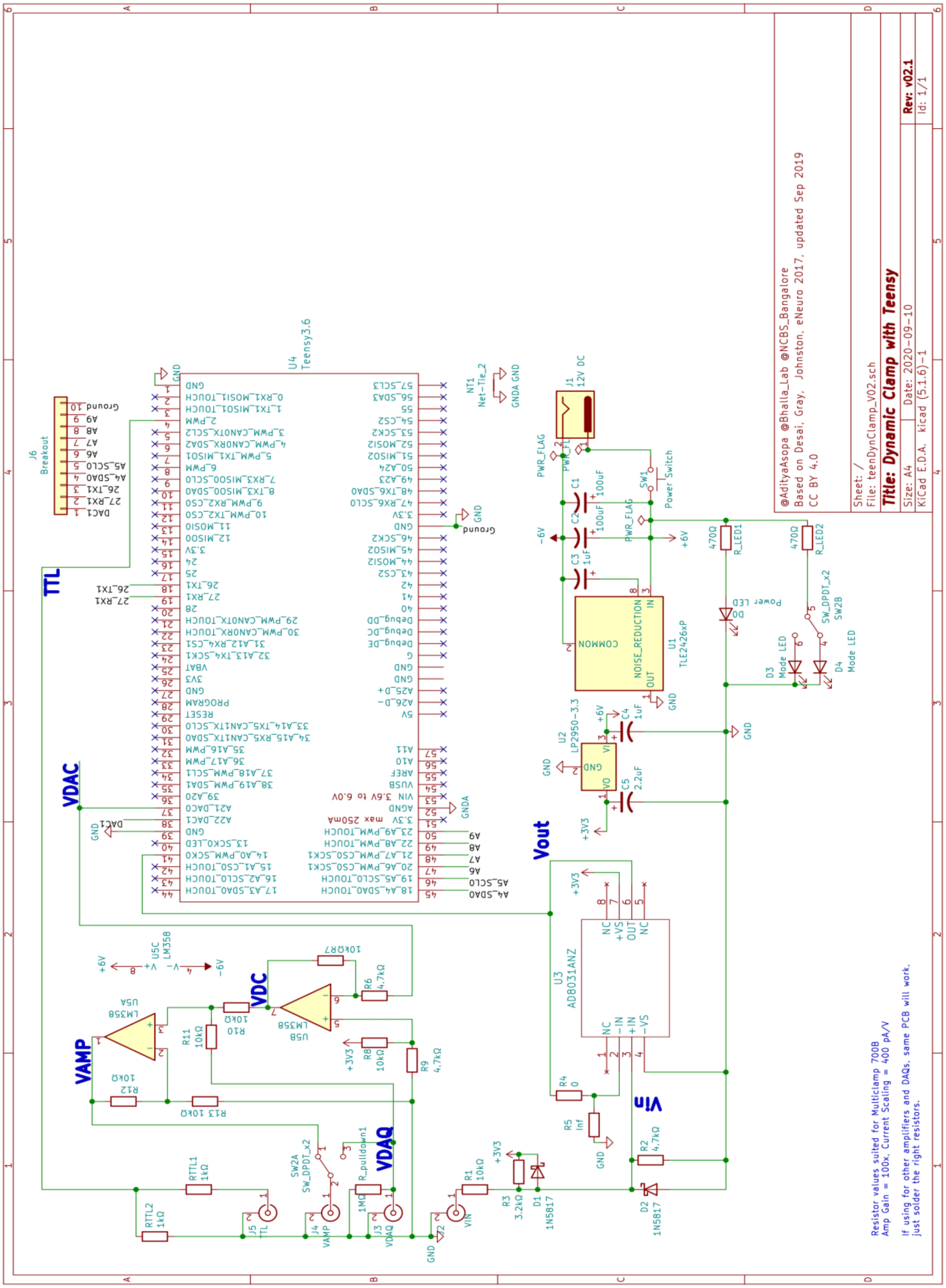
- Do not keep loose metal items on the bench around the board. They can short contacts.
- Keep the board fixed at one place, with blue tack or foam.
- Keep the benchtop tidy.
- Do not touch ICs with bare hands without grounding.
- Be careful around the capacitors even after the power supply is disconnected.
- Always disconnect power supply to the board before changing anything.
- Do not put Teensy before testing all the other stages.

## Equipment Needed

- Oscilloscope
- Function Generator (extra good if saw tooth function present, we are going to make good use of it)
- Computer with Processing, Teensy and Arduino IDE installed.
- Breadboard Jumpers
- Alligator ended BNCs
- Bench top power supply (±10V range)

## Circuit Schematic





@AdityaAsopa @Bhalla_Lab @NCBS_Bangalore
Based on Desai, Gray, Johnston, eNeuro 2017, updated Sep 2019
CC BY 4.0
Sheet: /
File: teenDynClamp_V02.sch
Title: Dynamic Clamp with Teensy
Size: A4
Date: 2020-09-10
Rev: v02.1
KiCad E.D.A. kicad (5.1.6)-1
Id: 1/1
Resistor values suited for Multiclamp 700B
Amp Gain = 100x, Current Scaling = 400 pA/V
If using for other amplifiers and DAQs, same PCB will work,
just solder the right resistors.
U4
Teensy3.6
U3
AD8031ANZ
U1
TLE2426xP
U2
LP2950-3.3
J6
Breakout
J1
12V DC
SW1
Power Switch
TTL
VDAC
VDC
VAMP
VDAQ
Vin
Vout

## Stages

<table>
<tr><th>Stage</th><th>Chip</th><th>Function</th><th>Input Signal(s)</th><th>Output Signals</th></tr>
<tr><td>1</td><td>TLE2426</td><td>Rail Splitter, create a rail exactly the centre of supply voltage range</td><td>+/- $V_{supply}$<br>(9 to 24V)</td><td>(+V/2) to Gnd to (-V/2)</td></tr>
<tr><td>2</td><td>LP2950-33</td><td>3.3V regulator</td><td>+V/2 to Gnd</td><td>3.3V to Gnd</td></tr>
<tr><td>3</td><td>AD8031</td><td>Amplifier: scales patch amplifier readout of membrane voltage into teensy acceptable range</td><td>$\boldsymbol{V_{in}}$<br><br>Vin is cell voltage $V_m$ amplified by the patch amplifier.<br>(For axon, gain = 100x)<br><br>$V_m$ range = ±100 mV after 100x gain:<br>$V_{in}$ range = ±10 V</td><td>$\mathbf{V_{out}}$<br><table><tr><th></th><th>Vin</th><th>Vout</th></tr><tr><td>Max</td><td>+10V</td><td>3.3V</td></tr><tr><td>zero</td><td>0 V</td><td>1.65 V</td></tr><tr><td>Min</td><td>-10V</td><td>0 V</td></tr></table></td></tr>
<tr><td>4</td><td>Teensy 3.6</td><td>Computing conductances, producing scaled outputs for current injection</td><td>$\mathbf{V_{out}}$</td><td>$\mathbf{V_{DAC}}$</td></tr>
<tr><td>5</td><td>LM358</td><td>Amplifier: produces voltage proportional to the current to be injected in the cell<br><br>5a. scales $V_{DAC}$ to $V_{DC}$<br><br>5b. Add $V_{DAQ}$ to $V_{DC}$</td><td>$\mathbf{V_{DAC}}$<br>signal range: 3.3 - 0 V<br><br>$\mathbf{V_{DC}}$ **and** $\mathbf{V_{DAQ}}$</td><td>$\mathbf{V_{DC}}$<br>range: -3V to +3V<br><br>$\mathbf{V_{AMP}} = V_{DC} + V_{DAQ}$</td></tr>
</table>

The dynamic clamp circuit does not read the membrane potential of the cell directly but reads an amplified copy of it that is generated by the patch clamp amplifier. This amplification gain is kept at 100mV/V. Once calibrated, this gain should not be changed when the dynamic clamp circuit is in operation. For that reason, it is recommended that this signal be taken from secondary output of Multiclamp amplifier. This will allow us to record the membrane potential of the cell at any gain using primary output independent of the dynamic clamp.

$V_{AMP}$ is a voltage that is sent to the patch clamp amplifier that then injects a proportional amount of current into the patched cell. This proportionality constant can be changed in multiclamp. Usually it is set to 400pA/V. This means that if VAMP is 1 V, the patch clamp amplifier will inject 400 pA into the cell.

## Testing on Bench

### Stage 1: Voltage rails.

- Bench top power supply was used.
- Connect the power supply.
- Measure voltage across the power supply jack: 18 V
- Across pin 1 of rail splitter (TLE2426) and both terminals of power supply jack: +9 V and - 9 V
- Measure voltage at pin 4 w.r.t GND: -9 V
- Measure voltage at pin 8 w.r.t GND: +9 V

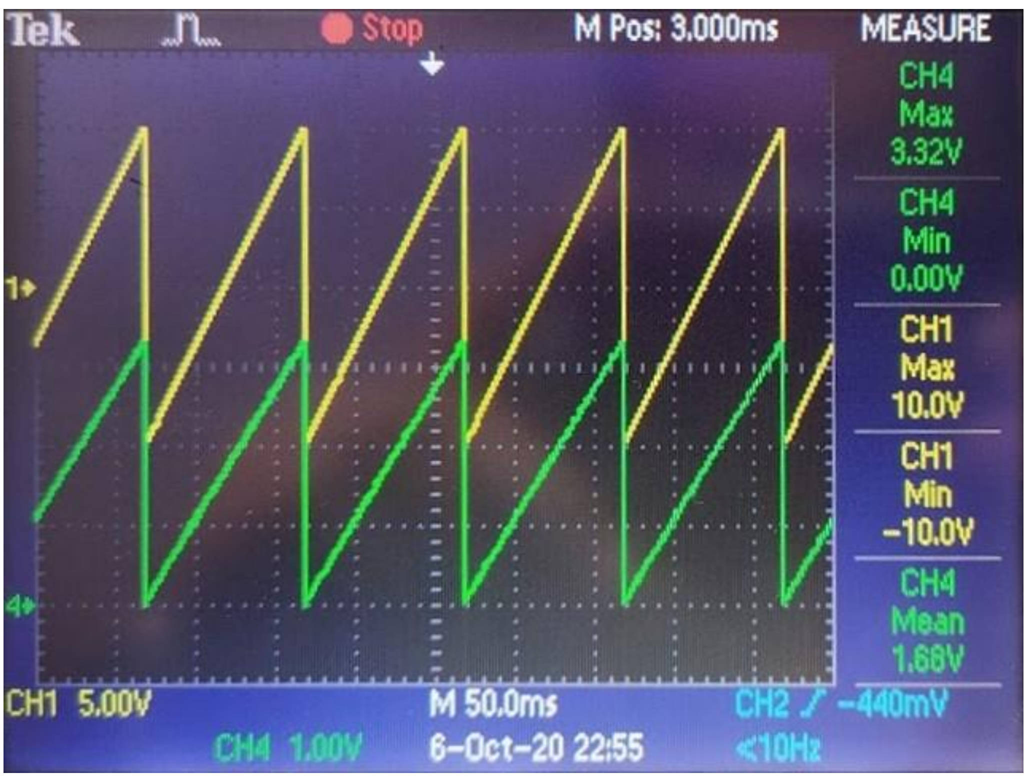
### Stage 2: 3.3V Voltage rail

- Measure the voltage at pin 1 of LP2950 with respect to GND: 3.30 V
- Measure the voltage at pin 7 of AD8031 with respect to GND: 3.30 V
- Measure the voltage at D1: 3.30 V

### Stage 3: Vin -> Vout scaling by AD8031

- Give a linear ramp of voltage from -1 V to +1 V using a function generator at $V_{in}$ BNC.
- Measure $V_{out}$ with oscilloscope using a jumper probe put into pin 12 of Teensy socket (without Teensy present) or a probe touched to pin 6 of AD8031.

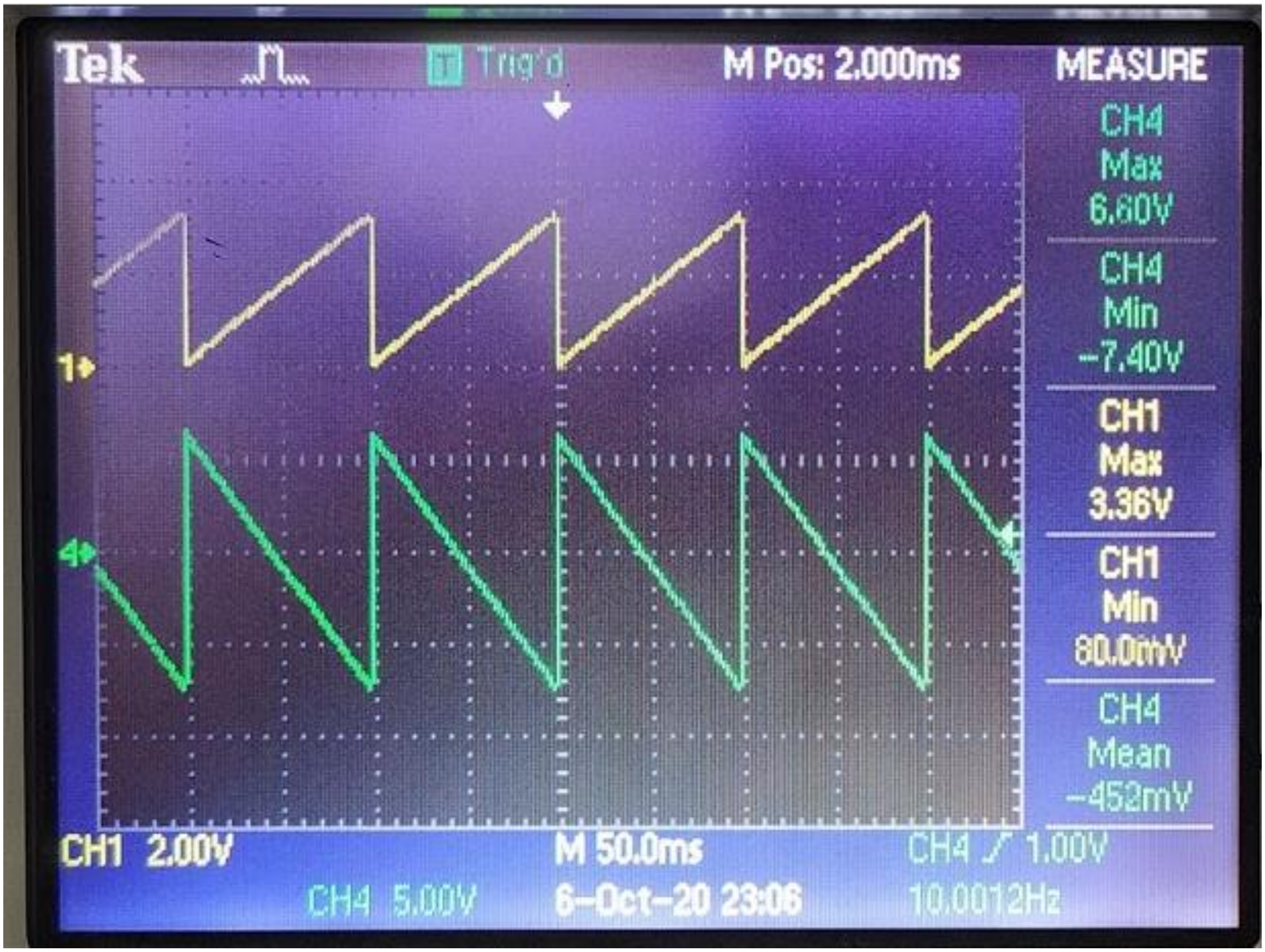
Fig. Vin shown in yellow (-10 to +10V) and Vout (0 to 3.3 V) shown in green, perfectly linear scaling as expected



### Stage 5a: Testing the scaling of VDAC to VDC

- Put the function generator input into VDAC pin of teensy socket w.r.t. GND.
- Input function should be a ramp from 0 to 3.3V, in the range the Teensy would generate.
- Measure output of first opAmp of LM358 pin 7 w.r.t ground.

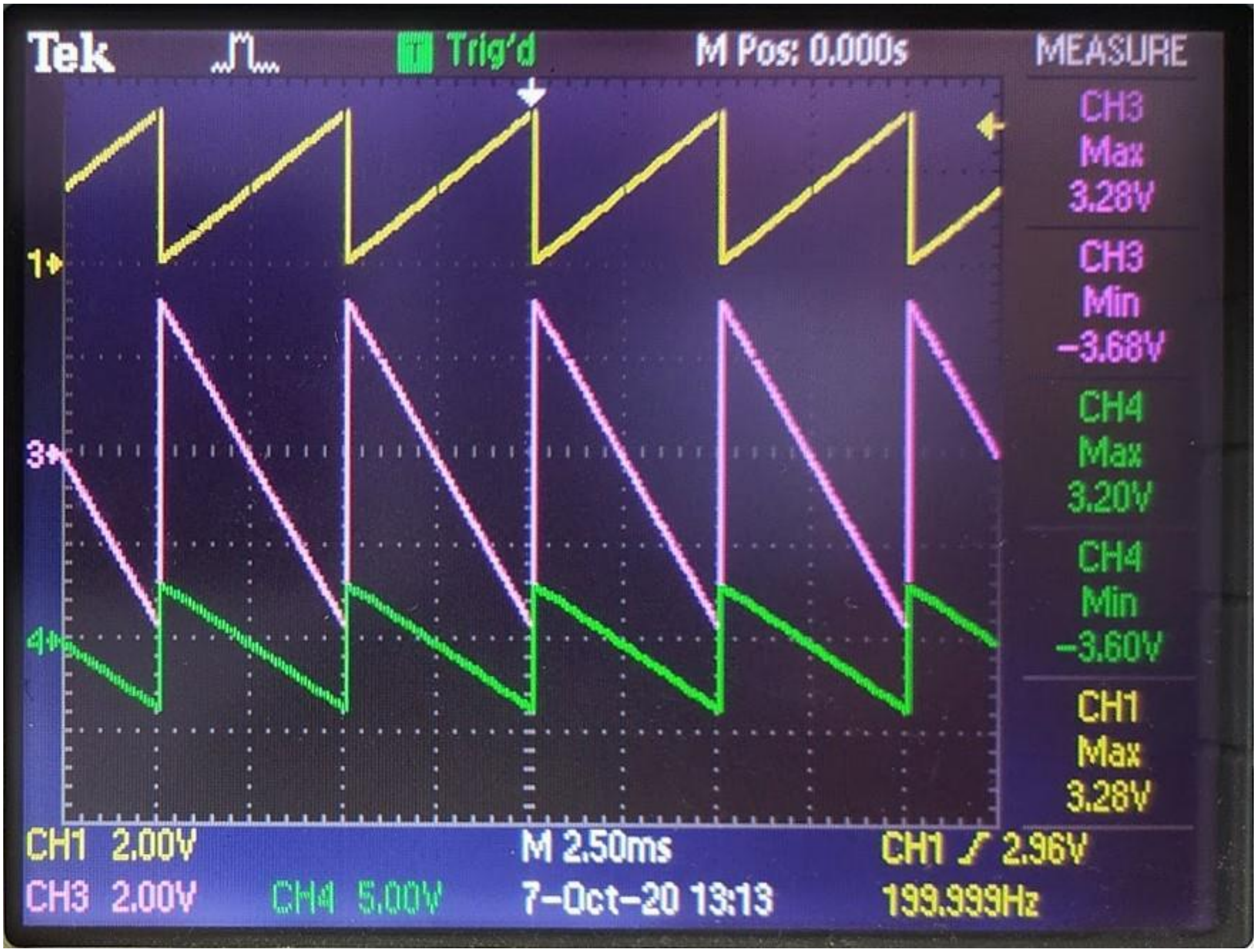
Fig. VDAC in yellow (0.08 to 3.36 V) and VDC (6.6 to 7.4 V) in green showing linear scaling.




### Stage 5b: Adding VDC to VDAQ

- VDAQ is the voltage that comes from the DAQ as the scaled version of command current that is going to injected into the cell by the amplifier. We want to add our VDC (dynamic clamp scaled voltage) to it.
- Give an analogue signal at VDAQ (BNC terminal on board) while giving another signal at VDAC (Pin 8 of teensy socket).
- Measure the resultant signal VAMP at VAMP BNC terminal or at the mode jumper terminal.

Fig. VAMP = VDC + VDAQ. Linear and inverted scaling of VDAC (0 to 3.3 V, yellow) to VDC (-3.7 to 3.3 V, pink), with VAMP (-3.6 to 3.3 V, green), when VDAQ = 0.




- Now turn VDAC to be zero and supply a voltage signal at VDAQ. Read output at VDC and VAMP.
- As VDAC is zero, VDC should be zero too, and VAMP should just be a scaled version of VDAQ signal.

Fig. VDAC is zero (yellow) and a signal is supplied at VDAQ (blue, 0 to 3 V). As a result VDC is zero and VAMP is a linearlly scaled signal proportional to VDAQ.


These quick tests show that the relationship between different signals, their ranges, scaling, and inversions are as expected.

## Testing the Stagewise IO Relationship/Scaling Using an Oscilloscope

These tests are done to test the individual modules or stages of the dynamic clamp circuit on the board. Please refer to the circuit schematic for labels and naming conventions.

### Stage 3: AD8031 Vin --> Vout

**Theoretical relation:**

$$\mathrm{V_{OUT}} = \frac{\frac{\mathrm{V_{IN}}}{\mathrm{R_1}} + \frac{3.3}{\mathrm{R_3}}}{\left(\frac{1}{\mathrm{R_1}} + \frac{1}{\mathrm{R_2}} + \frac{1}{\mathrm{R_3}}\right)} \tag{1}$$

$or$

$$\mathrm{V_{OUT}} = \frac{1}{R_1}\left(\frac{1}{\frac{1}{\mathrm{R_1}} + \frac{1}{\mathrm{R_2}} + \frac{1}{\mathrm{R_3}}}\right) * \mathrm{V_{IN}} + \frac{1}{R_3}\left(\frac{1}{\frac{1}{\mathrm{R_1}} + \frac{1}{\mathrm{R_2}} + \frac{1}{\mathrm{R_3}}}\right)$$

Given: $R_1$ = 22kΩ, $R_2$ = 10kΩ, $R_3$ = 3.2kΩ

$$\mathrm{V_{OUT}} = 0.15993 * \mathrm{V_{IN}} + 1.6493 \tag{2}$$

Here:

$$m_{in} = \frac{1}{R_1}\left(\frac{1}{\frac{1}{\mathrm{R_1}} + \frac{1}{\mathrm{R_2}} + \frac{1}{\mathrm{R_3}}}\right) = 0.15993 \tag{2a}$$

And

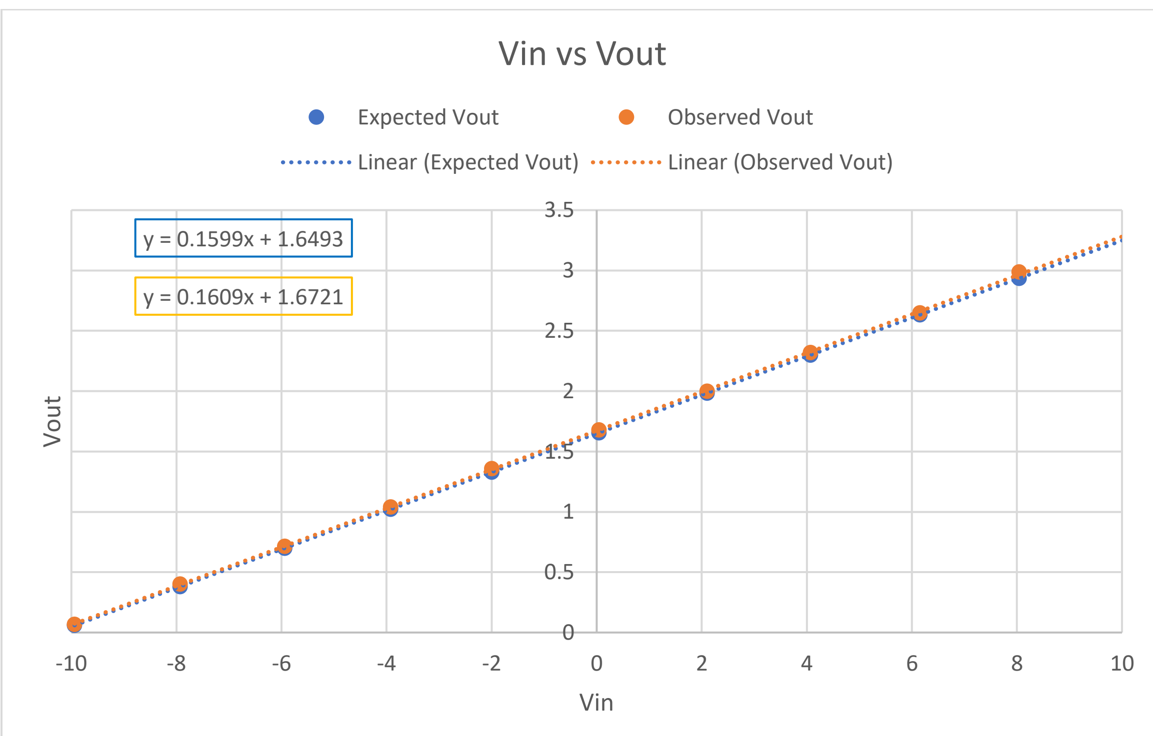
$$c_{in} = \frac{1}{R_3}\left(\frac{1}{\frac{1}{\mathrm{R_1}} + \frac{1}{\mathrm{R_2}} + \frac{1}{\mathrm{R_3}}}\right) = 1.6493 \tag{2b}$$

**Setup:**

Use a function generator to produce voltage levels and feed it to Vin and measure Vout using an oscilloscope.

**Measurement: Table 1**

| Vin (V) | Expected Vout (V) $V_{OUT} = 0.15993 * V_{IN} + 1.6493$ | Observed Vout (V) | Error |
|---|---|---|---|
| -9.940 | 0.060 | 0.068 | -14.102 |
| -7.930 | 0.381 | 0.403 | -5.759 |
| -5.940 | 0.699 | 0.716 | -2.386 |
| -3.920 | 1.022 | 1.040 | -1.724 |
| -2.000 | 1.329 | 1.360 | -2.299 |
| 0.041 | 1.656 | 1.680 | -1.458 |
| 2.100 | 1.985 | 2.000 | -0.748 |
| 4.070 | 2.300 | 2.320 | -0.860 |
| 6.150 | 2.633 | 2.650 | -0.651 |
| 8.040 | 2.935 | 2.990 | -1.869 |
| 10.100 | 3.265 | 3.290 | -0.778 |
| **Slope ($m_{in}$)** | **0.1599** | **0.1609** | |
| **Intercept ($c_{in}$)** | **1.6493** | **1.6721** | |



## Stage 5a: LM358a, VDAC+VDAQ --> VAMP, with VDAQ = 0

**Theoretical relation:**

$$\mathrm{V_{DC}} = -\mathrm{V_{DAC}} * \frac{\mathrm{R_7}}{\mathrm{R_6}} + 3.3 \; * \; \left(\frac{\mathrm{R_9}}{\mathrm{R_8} + \mathrm{R_9}}\right)\left(\frac{\mathrm{R_6} + \mathrm{R_7}}{\mathrm{R_6}}\right) \tag{4}$$

Given: $R_6$ = 4.7kΩ, $R_7$ = 10kΩ, $R_8$ = 10kΩ, $R_9$ = 4.7kΩ

$$\mathrm{V_{DC}} = -2.1277 * \; \mathrm{V_{DAC}} + 3.3$$

Here:

$$m_{out} = \frac{\mathrm{R_7}}{\mathrm{R_6}} = -2.1277 \tag{4a}$$

And

$$c_{out} = \; 3.3 \; * \; \left(\frac{\mathrm{R_9}}{\mathrm{R_8} + \mathrm{R_9}}\right)\left(\frac{\mathrm{R_6} + \mathrm{R_7}}{\mathrm{R_6}}\right) = 3.3 \tag{4b}$$

And

$$\mathrm{V_{AMP}} = \; \mathrm{V_{DAQ}} + \; \mathrm{V_{DC}}$$

$$\mathrm{V_{AMP}} = \; \mathrm{V_{DAQ}} - 2.1277 * \; \mathrm{V_{DAC}} + 3.3 \tag{5}$$

If VDAQ is GND i.e., 0 V

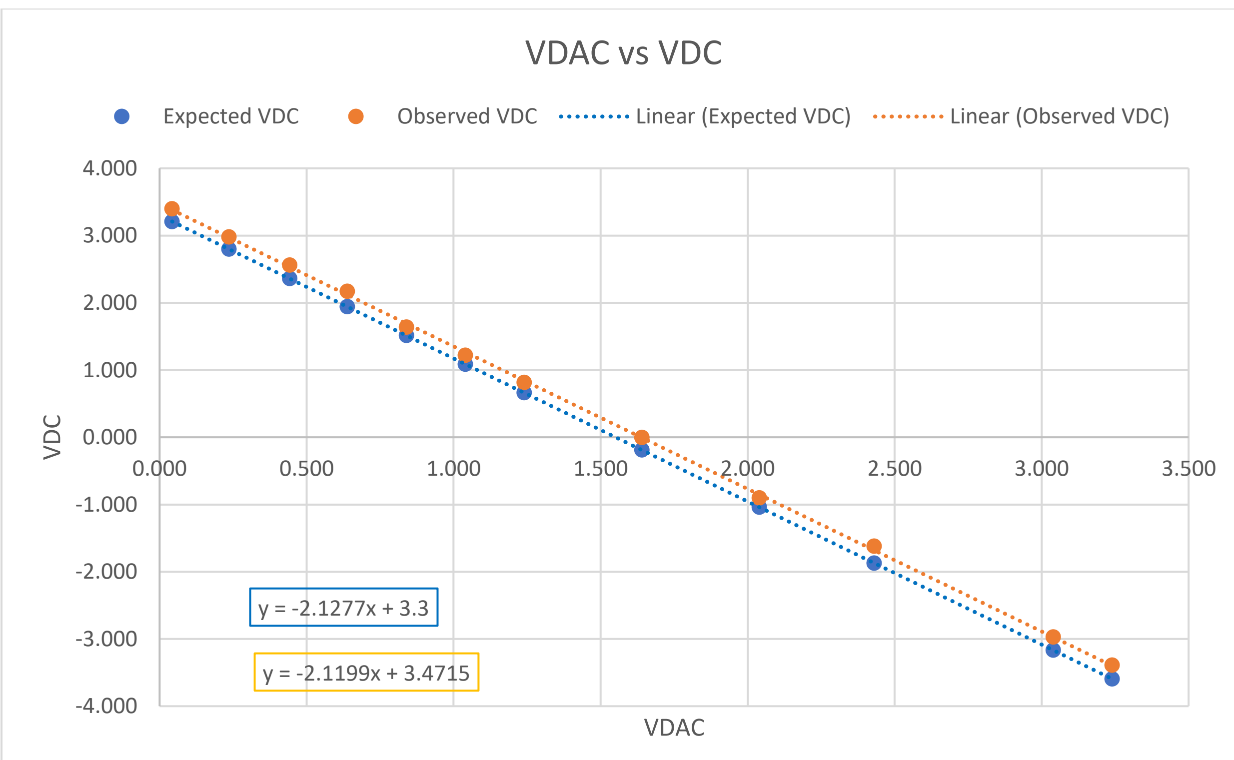
$$\mathrm{V_{AMP}} = \; \mathrm{V_{DC}} = -2.1277 * \; \mathrm{V_{DAC}} + 3.3$$

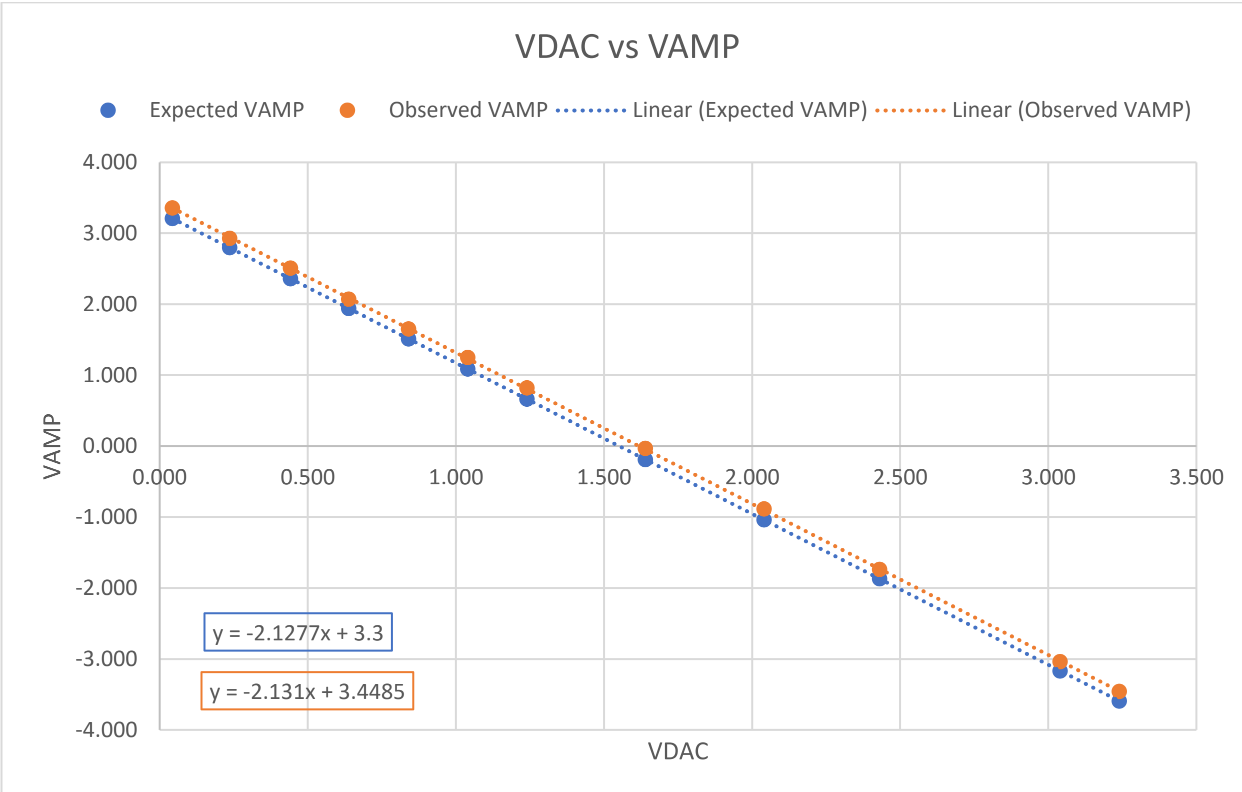
**Setup**:

Feed voltage using a function generator (for ex, using a very low frequency square pulse), to VDAC, and connect VDAQ to Gnd (0 V) and measure using a function generator at VAMP.

**Measurement: Table 2**

| VDAQ | VDAC | Expected VDC $= -2.1277 * V_{DAC} + 3.3$ | Observed VDC | VDC Error | Expected VAMP $= -2.1277 * V_{DAC} + 3.3$ | Observed VAMP | VAMP Error |
|---|---|---|---|---|---|---|---|
| 0.000 | 0.043 | 3.209 | 3.400 | -5.940 | 3.209 | 3.360 | -4.694 |
| 0.000 | 0.236 | 2.798 | 2.980 | -6.510 | 2.798 | 2.930 | -4.722 |
| 0.000 | 0.442 | 2.360 | 2.560 | -8.494 | 2.360 | 2.510 | -6.375 |
| 0.000 | 0.638 | 1.943 | 2.170 | -11.709 | 1.943 | 2.070 | -6.561 |
| 0.000 | 0.840 | 1.513 | 1.640 | -8.411 | 1.513 | 1.650 | -9.072 |
| 0.000 | 1.040 | 1.087 | 1.220 | -12.211 | 1.087 | 1.250 | -14.971 |
| 0.000 | 1.240 | 0.662 | 0.814 | -23.016 | 0.662 | 0.821 | -24.074 |
| 0.000 | 1.640 | -0.189 | -0.005 | 97.360 | -0.189 | -0.032 | 83.101 |
| 0.000 | 2.040 | -1.040 | -0.901 | 13.401 | -1.040 | -0.884 | 15.035 |
| 0.000 | 2.430 | -1.870 | -1.620 | 13.379 | -1.870 | -1.740 | 6.962 |
| 0.000 | 3.040 | -3.168 | -2.970 | 6.253 | -3.168 | -3.040 | 4.043 |
| 0.000 | 3.240 | -3.594 | -3.390 | 5.666 | -3.594 | -3.460 | 3.718 |
| **Slope** | | **-2.128** | **-2.120** | | **-2.128** | **-2.131** | |
| **Intercept** | | **3.300** | **3.472** | | **3.300** | **3.449** | |






## Stage 5b: VDAQ + VDAC --> VAMP, with VDAC = 0

**Theoretical relation:**

From Eq 4

$$V_{AMP} = V_{DAQ} - 2.1277 * V_{DAC} + 3.3$$

If VDAC is GND i.e., 0 V

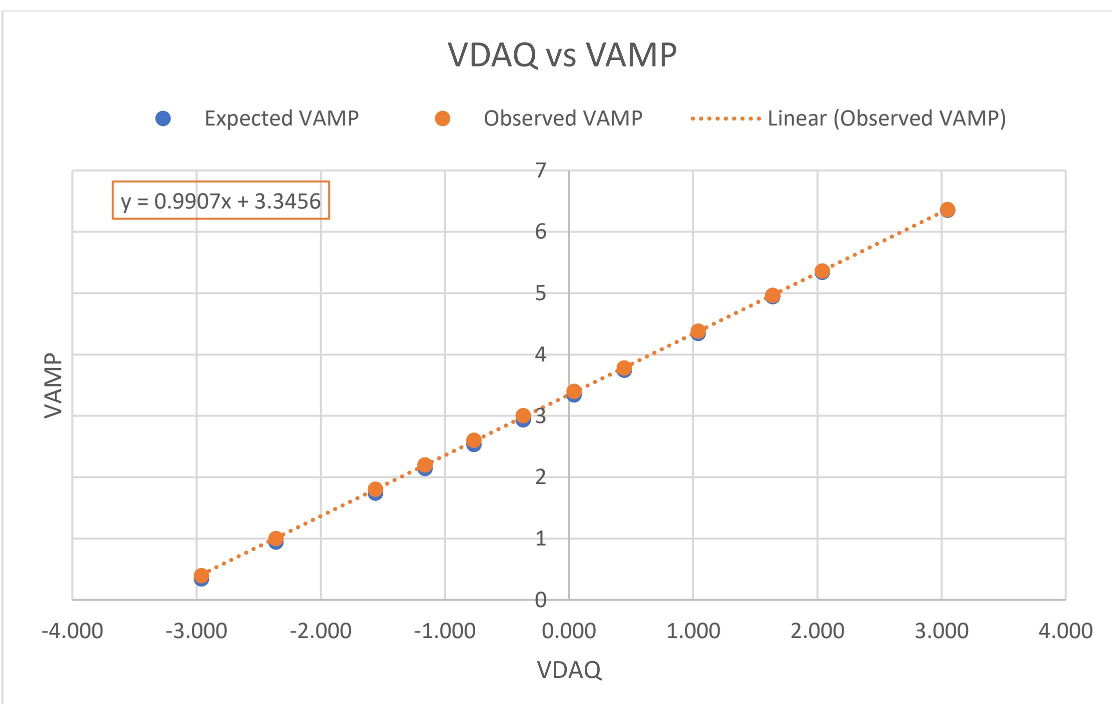
$$V_{AMP} = V_{DAQ} + 3.3$$

**Setup**:

Now switch the function generator to input into VDAQ and connect VDAC to Gnd (0 V) and measure from VAMP.

**Measurement: Table 3**

| VDAQ | VD AC | Expected VDC $= V_{DAQ} - 2.1277 * V_{DAC} + 3.3$ | Observed VDC | VDC Error | Expected VAMP $= V_{DAQ} - 2.1277 * V_{DAC} + 3.3$ | Observed VAMP | VAMP Error |
|---|---|---|---|---|---|---|---|
| -2.960 | 0.0 | 3.300 | 3.410 | -3.333 | 0.340 | 0.395 | -16.176 |
| -2.360 | 0.0 | 3.300 | 3.410 | -3.333 | 0.940 | 1.000 | -6.383 |
| -1.560 | 0.0 | 3.300 | 3.410 | -3.333 | 1.740 | 1.800 | -3.448 |
| -1.160 | 0.0 | 3.300 | 3.410 | -3.333 | 2.140 | 2.200 | -2.804 |
| -0.766 | 0.0 | 3.300 | 3.410 | -3.333 | 2.534 | 2.600 | -2.605 |
| -0.369 | 0.0 | 3.300 | 3.410 | -3.333 | 2.931 | 3.000 | -2.354 |
| 0.040 | 0.0 | 3.300 | 3.410 | -3.333 | 3.340 | 3.400 | -1.796 |
| 0.444 | 0.0 | 3.300 | 3.410 | -3.333 | 3.744 | 3.780 | -0.962 |
| 1.040 | 0.0 | 3.300 | 3.410 | -3.333 | 4.340 | 4.380 | -0.922 |
| 1.640 | 0.0 | 3.300 | 3.410 | -3.333 | 4.940 | 4.960 | -0.405 |
| 2.040 | 0.0 | 3.300 | 3.410 | -3.333 | 5.340 | 5.360 | -0.375 |
| 3.050 | 0.0 | 3.300 | 3.410 | -3.333 | 6.350 | 6.360 | -0.157 |
| **Slope** | | **0.000** | **0.000** | | **1.000** | **0.991** | |
| **Intercept** | | **3.300** | **3.410** | | **3.300** | **3.346** | |



## Theoretical Calibration

Calibration sets the relationship between amplifier and Teensy.

Input stage: relation between cell membrane voltage (Vm in mV) and the ADC values read by Teensy.

Output stage: relation between DAC values sent by Teensy and current injected by the amplifier into the cell.

### Input Stage Calibration

$V_{in}$ i.e., $V_m$ after amplification by multiclamp:

$$\mathrm{V_{in}} = \mathrm{V_m} * \mathrm{g_{in}}$$

Vout produced by AD8031, from Eq. 2:

$$\mathrm{V_{out}} = \mathrm{m_{in}} * \mathrm{V_{in}} + \mathrm{c_{in}}$$

Vout digitized by Teensy into ADC:

$$\mathrm{ADC} = \frac{4096}{3.3} * \mathrm{V_{out}}$$

Solving for $V_m$ and ADC:

$$ADC = \left(\frac{4096 * m_{in} * g_{in}}{3.3}\right) * V_m - \left(\frac{4096}{3.3}\right) * c_{in} \quad (5)$$

Here:

$$\text{Slope Vm to ADC} = \left(\frac{4096 * \mathrm{m_{in}} * \mathrm{g_{in}}}{3.3}\right)$$

And

$$\text{Intercept Vm to ADC} = \left(\frac{4096}{3.3}\right) * \mathrm{c_{in}}$$

Teensy should know the ADC to Vm conversion. In the dynamic clamp program that goes on the teensy for operation, we need to feed the input slope and input intercept. When we put together the whole board and do calibration, we will change the Vm and record the ADC values from Teensy. From that we will get the "Intercept Vm to ADC" and "Slope Vm to ADC" values. Using that we must obtain the inverse relationship i.e. slope and intercept to get Vm values from ADC values.

$$V_m = \left(\frac{3.3}{4096 * m_{in} * g_{in}}\right) * ADC - \left(\frac{c_{in}}{m_{in} * g_{in}}\right) \tag{6}$$

Here:

$$\text{Input Slope} = \left(\frac{3.3}{4096 * m_{in} * g_{in}}\right) \tag{6a}$$

$$\text{Input Intercept} = \left(-\frac{c_{in}}{m_{in} * g_{in}}\right) \tag{6b}$$

**Calculation: Table 4**

| Quantity | Value | Unit |
|---|---|---|
| Vm | -65.00 | mV |
| | | |
| Gain of Amp | | |
| stage 1 | 10 | x |
| Stage 2 | 10 | x |
| Total Input Gain | 100 | mV/mV |
| | | |
| Digitization Max Voltage | 3.3 | V |
| Digitization Bit depth | 4096 | levels |
| | | |
| IO relation (from Table 1) | Expected | According to Observed Scaling |
| Slope Vout to Vin (m_in) | 0.1599 | 0.1609 |
| Intercept Vout to Vin (c_in) | 1.6493 | 1.6721 |
| | | |
| Vm to ADC | | |
| Slope Vm to ADC | 19.84698182 | 19.9711 |
| Intercept Vm to ADC | 2047.131152 | 2075.431 |
| | | |
| Inverse relationship | | |
| Input Slope | 0.050385495 | 0.050072 |
| Input Intercept | -103.1457161 | -103.922 |

## Output Stage Calibration

$$V_{AMP} = V_{DC}$$

$$\mathrm{V_{DC}} = \mathrm{m_{out}} * \mathrm{V_{DAC}} + \mathrm{c_{out}}$$

$$\mathrm{V_{DAC}} = \left(\frac{3.3}{4096}\right) * \mathrm{DAC}$$

$$\mathrm{I_{inj}} = \mathrm{g_{out}} * \mathrm{V_{AMP}}$$

Combining all:

$$\mathrm{I_{inj}} = \left(\frac{3.3 * \mathrm{m_{out}} * \mathrm{g_{out}}}{4096}\right) * \mathrm{DAC} + (\mathrm{c_{out}} * \mathrm{g_{out}}) \tag{7}$$

And inversely:

$$\mathrm{DAC} = \left(\frac{4096}{3.3 * \mathrm{m_{out}} * \mathrm{g_{out}}}\right) * \mathrm{I_{inj}} + \left(-\frac{4096 * \mathrm{c_{out}}}{3.3 * \mathrm{m_{out}}}\right) \tag{8}$$

Here:

$$\text{Output Slope} = \frac{4096}{3.3 * \mathrm{m_{out}} * \mathrm{g_{out}}} \tag{8a}$$

$$\text{Output Intercept} = -\frac{4096 * \mathrm{c_{out}}}{3.3 * \mathrm{m_{out}}} \tag{8b}$$

During calibration, using processing calibration program with a model cell, the teensy wants to figure out the output relationship i.e., how much DAC value gets converted to how much current injection into the cell. For that, teensy generates an array of DAC values, and then measures the Vm of the model cell (in form of the ADC values). From Vm and resistance of the model cell, it calculates how much current is sent to the model cell.

We have to empirically find Input and Output slopes and intercepts.

**Table 5**

| Quantity | Value | Unit |
|---|---|---|
| $I_{inj}$ | -100.00 | pA |
| Total Output Gain | 400 | pA/V |
| Digitization Max Voltage | 3.3 | V |
| Digitization Bit depth | 4096 | Levels |
| | | |
| **IO relation from Table 2** | Expected | According to Observed Scaling |
| Slope ($m_{out}$) | -2.1277 | -2.131 |
| Intercept ($c_{out}$) | 3.3 | 3.4485 |
| | | |
| DAC to $I_{inj}$ | | |
| Slope | -0.68568457 | -0.68675 |
| Intercept | 1320 | 1379.4 |
| | | |
| Inverse relationship | | |
| Output Slope | -1.458396533 | -1.45614 |
| Output Intercept | 1925.083423 | 2008.597 |

## Direct Calibration with a Function Generator and an Oscilloscope

In direct calibration, we can give a certain known value on the input side of the stage and measure the value on the output side. From the measurements, if we fit a line, we can get the slope and intercept of that relationship. This can be done using a function generator and an oscilloscope without the need of a model cell.

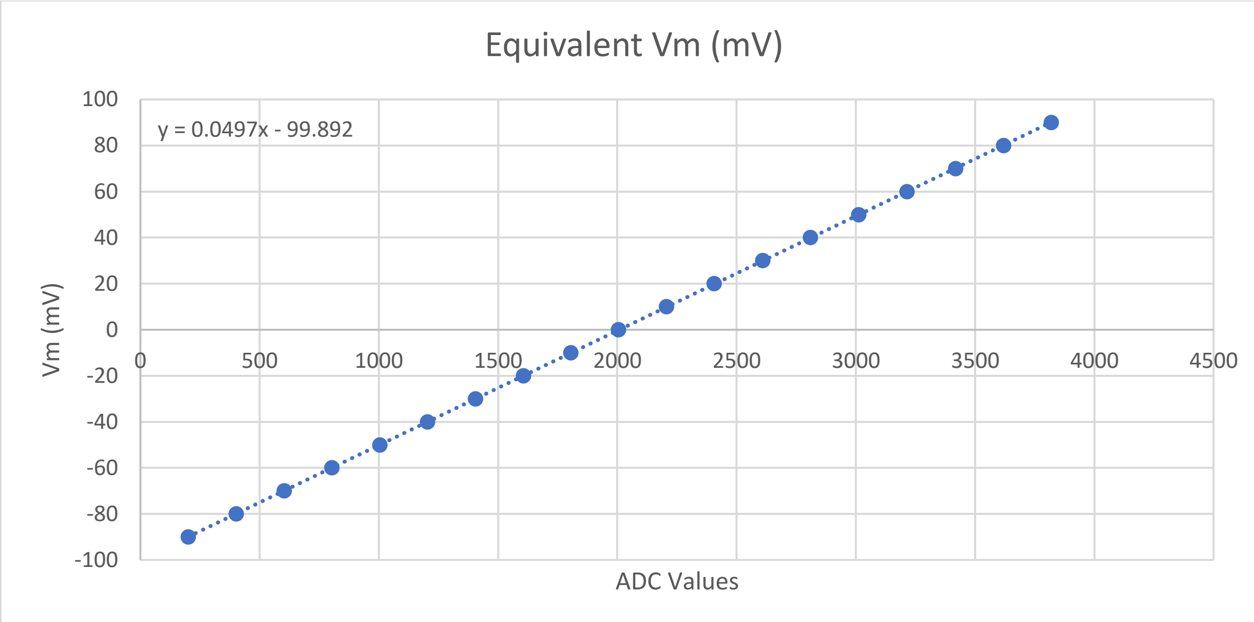
### Input Stage Calibration: Vin → ADC

**Setup**:

Put Teensy into circuit. Using a function generator, give known constant voltages to Vin and read ADC values recorded by Teensy. Measure the ADC to Vin relationship.

**Measurement**: Table 6, Compare with theoretical values from table 4

| Input Vin Using a function Gen (V) | Equivalent Vm (mV) | Measured ADC Value |
|---|---|---|
| -9 | -90 | 200 |
| -8 | -80 | 402 |
| -7 | -70 | 603 |
| -6 | -60 | 803 |
| -5 | -50 | 1004 |
| -4 | -40 | 1204 |
| -3 | -30 | 1405 |
| -2 | -20 | 1606 |
| -1 | -10 | 1805 |
| 0 | 0 | 2005 |
| 1 | 10 | 2206 |
| 2 | 20 | 2406 |
| 3 | 30 | 2610 |
| 4 | 40 | 2810 |
| 5 | 50 | 3012 |
| 6 | 60 | 3215 |
| 7 | 70 | 3419 |
| 8 | 80 | 3620 |
| 9 | 90 | 3820 |
| Input Slope | 0.0497 | |
| Input Intercept | -99.892 | |

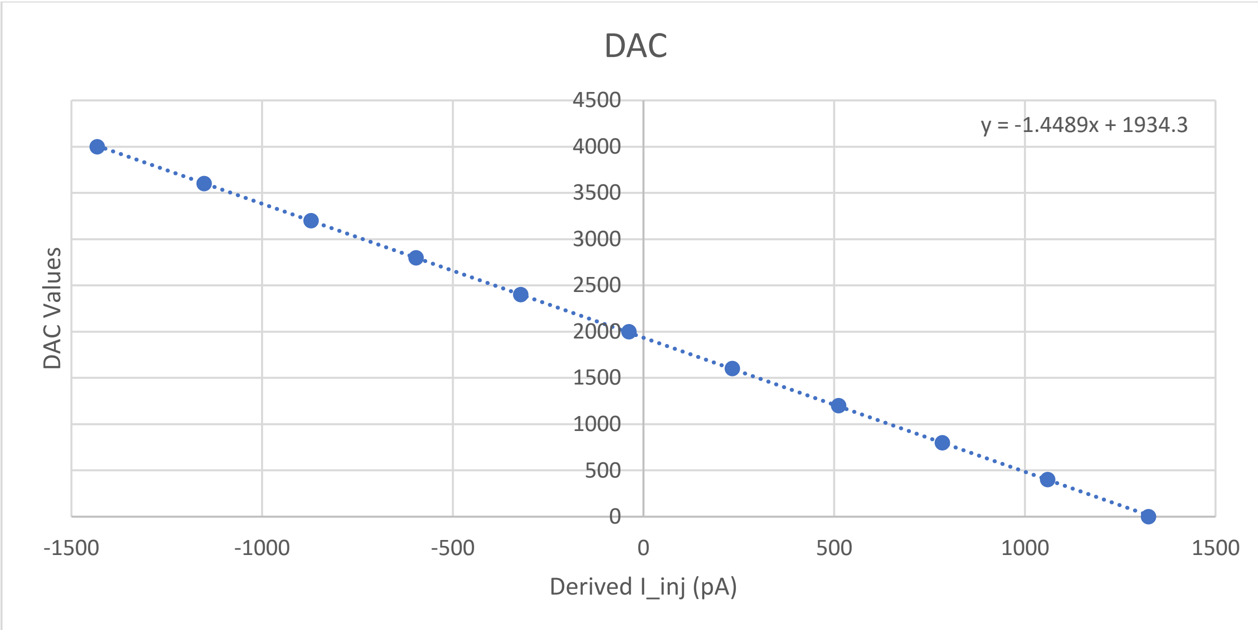



## Output Stage Calibration:

**Setup**: For output calibration, I used a teensy program to give a number of DAC values and recorded the VDAC and VAMP using an oscilloscope. Program script is in appendix.

**Measurement**: Table 7, Compare with theoretical values from Table 5

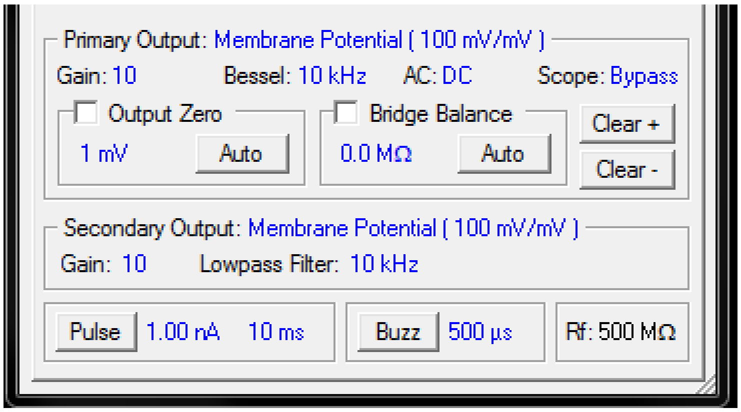
| i | DAC | VDAC | measured VAMP (V) | I_inj (pA) derived from measured VAMP |
|---|---|---|---|---|
| 0 | 0 | 0.065 | 3.31 | 1324 |
| 1 | 400 | 0.348 | 2.65 | 1060 |
| 2 | 800 | 0.702 | 1.96 | 784 |
| 3 | 1200 | 1.03 | 1.28 | 512 |
| 4 | 1600 | 1.36 | 0.583 | 233.2 |
| 5 | 2000 | 1.68 | -0.095 | -38 |
| 6 | 2400 | 2.01 | -0.805 | -322 |
| 7 | 2800 | 2.31 | -1.49 | -596 |
| 8 | 3200 | 2.64 | -2.18 | -872 |
| 9 | 3600 | 2.96 | -2.88 | -1152 |
| 10 | 4000 | 3.3 | -3.58 | -1432 |
| Output Slope | -1.4489 | | | |
| Output Intercept | 1934.3 | | | |



DAC
y = -1.4489x + 1934.3
4500
4000
3500
3000
2500
2000
1500
1000
500
0
DAC Values
-1500
-1000
-500
0
500
1000
1500
Derived I_inj (pA)

## Calibration with a Model Cell Using Processing

Now dynamic clamp can be boxed and put into connection with the Axon devices. Remember the dynamic clamp stage 1 is designed to read a 100x amplified Vm between the range ±10V. Multiclamp, the patch clamp amplifier sends a secondary output that can be selected to be a scaled copy of membrane potential Vm. It should have the same gain settings of 10, which gives an amplification of 100 mV/mV (i.e., 100x)[1]. We will use this output from multiclamp to read cell Vm.,




**Setup**:

Instructions are described in the original documentation. Plug in a model cell, switch to current clamp on multiclamp and remove any offsets so that model cell Vm is 0 mV at 0 pA current injection.

Calibration will be done using the graphical interface written in processing and can be found in the Dynamic Clamp directory. First, we have to upload the *teensy_calibration.ino* file to Teensy and then use Processing program *processing_calibration.pde* to measure input and output slopes and intercepts that we have been doing manually so far.

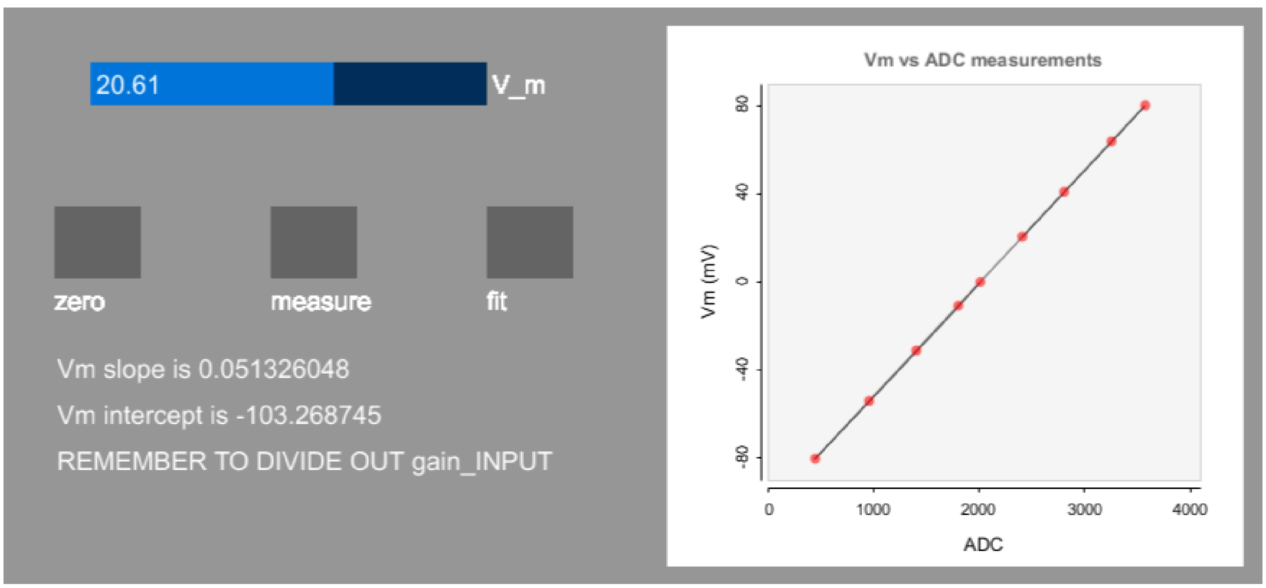
If you go through the theoretical calibration steps and formulae as described in earlier sections, it will be clear that the calibration values, intercepts, and slopes are entirely a function of resistors and amplifiers used in the circuit and ADC – DAC range of the Teensy and do not depend on the multiclamp amplifier and DAQ.

[1] Multiclamp has an internal gain of 10x that cannot be changed. So, when we select a gain of 10 in the user interface, the actual gain is 10 *10 = 100. Refer to table 4.

## Input Stage calibration with Processing Program

**Table 8**

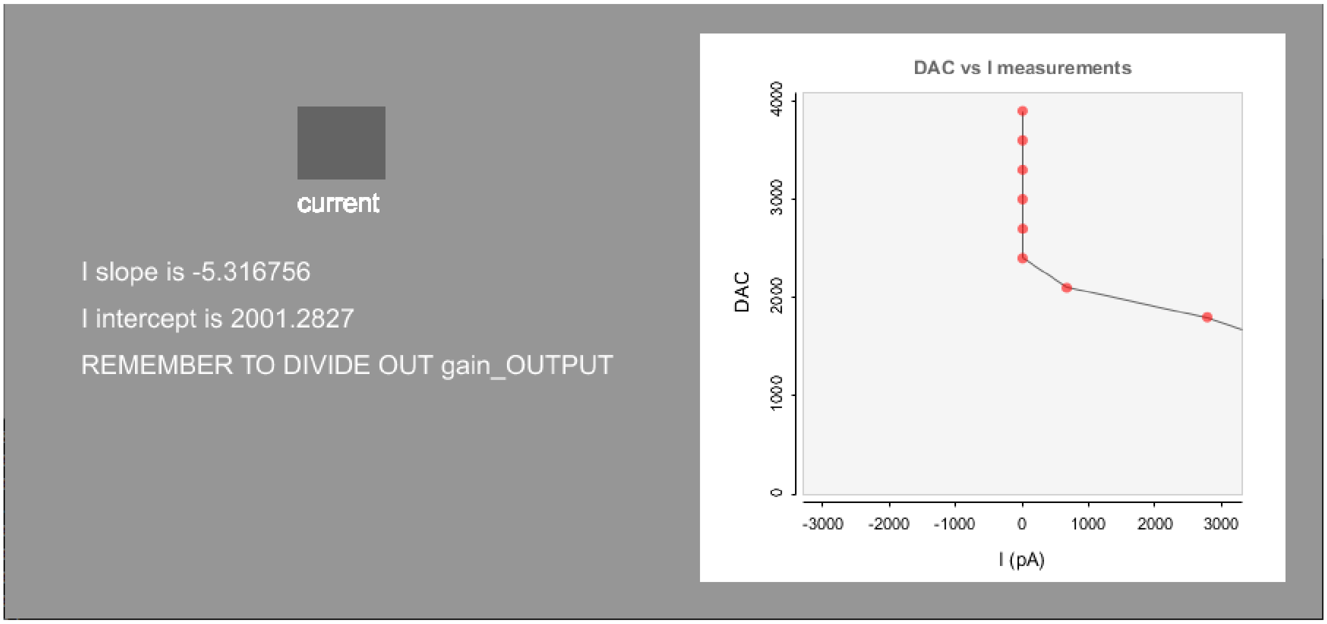
| | |
|---|---|
| Input Gain | 100 |
| **Vm (mV)** | **ADC reading** |
| 0.000 | 2014 |
| -10.800 | 1803 |
| -31.091 | 1407 |
| -54.000 | 960 |
| -80.182 | 449 |
| 80.182 | 3576 |
| 63.818 | 3255 |
| 40.909 | 2807 |
| 20.618 | 2411 |
| | |
| Input Slope | 0.0513 |
| Input Intercept | -103.27 |




The output of slope and intercept have to fed into the dynamic clamp main program *dynamic_clamp.ino* after multiplying with the gain (100).

```
const float inputSlope = 5.13/gain_INPUT.
const float inputIntercept = -10327/gain_INPUT.
```

## Output stage calibration with Processing Program

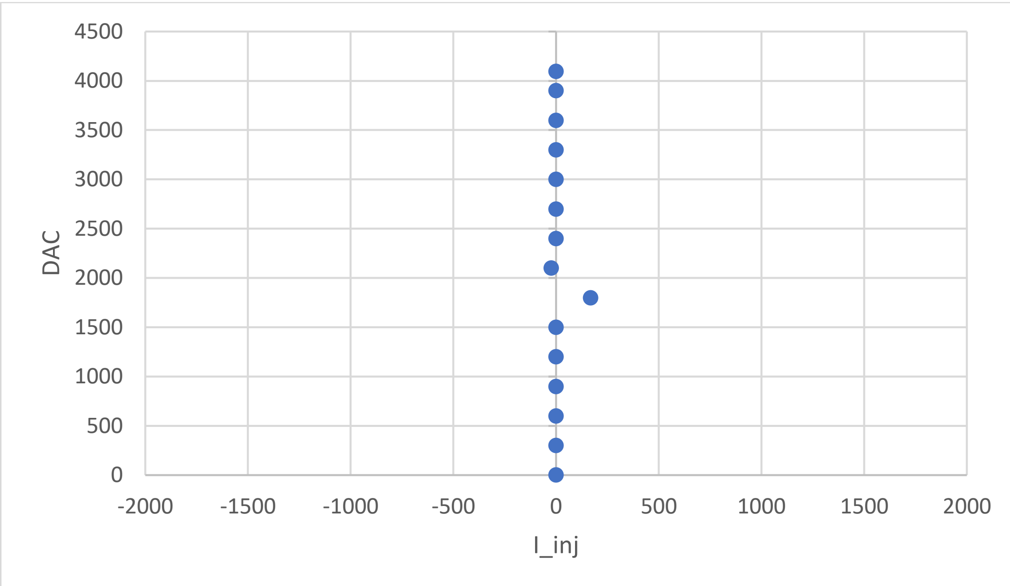



The processing program generates an error. The reason is that in the backend, the output calibration generates DAC values from 300 to 3900, which in turn would create a VAMP of ±1.4nA. The program then measures the Vm, derives $I_{inj}$ from Vm (using user supplied model cell resistance). Normally, for a patched cell (Rm ~ 100 MΩ, this current would create a Vm of around ±100 mV. But when a model cell is used (Rm ~ 500 MΩ) the Vm goes beyond 600 mV which is far beyond the range of dynamic clamp input and thus for most of the DAC values, Vm on model cell is unmeasurable.

**Table 9: Problem with output slope measurement**

| DAC | Expected VAMP (approx.) | Corresponding I_inj | Vm on Model cell (mV) | Vin going into DynClamp (V) | In Range of Teensy? (±10V) |
|---|---|---|---|---|---|
| 0 | 3.3 | 1320 | 671.88 | 67.188 | FALSE |
| 300 | 2.82 | 1128 | 574.152 | 57.4152 | FALSE |
| 600 | 2.34 | 936 | 476.424 | 47.6424 | FALSE |
| 900 | 1.86 | 744 | 378.696 | 37.8696 | FALSE |
| 1200 | 1.38 | 552 | 280.968 | 28.0968 | FALSE |
| 1500 | 0.9 | 360 | 183.24 | 18.324 | FALSE |
| 1800 | 0.42 | 168 | 85.512 | 8.5512 | TRUE |
| 2100 | -0.06 | -24 | -12.216 | -1.2216 | TRUE |
| 2400 | -0.54 | -216 | -109.944 | -10.9944 | FALSE |
| 2700 | -1.02 | -408 | -207.672 | -20.7672 | FALSE |
| 3000 | -1.5 | -600 | -305.4 | -30.54 | FALSE |
| 3300 | -1.98 | -792 | -403.128 | -40.3128 | FALSE |
| 3600 | -2.46 | -984 | -500.856 | -50.0856 | FALSE |
| 3900 | -2.94 | -1176 | -598.584 | -59.8584 | FALSE |
| 4095 | -3.252 | -1300.8 | -662.107 | -66.21072 | FALSE |

Resulting into the above plot or the one below




## Solution: Output stage calibration with a Script

There are two ways to solve this problem. Either we can use a model cell with lower resistance, or we can change the range of DAC values that are used in the processing program. There is a bonus method to solve the problem describer in the appendix.

Modifying the *processing_calibration.pde* program, line 29-31.

```
// DAC outputs between 300 and 3900 in increments of 300
29 for (int z=0; z<nOutputs; z++) {
30     outputs[z] = 300*(z+1);
31 }
```

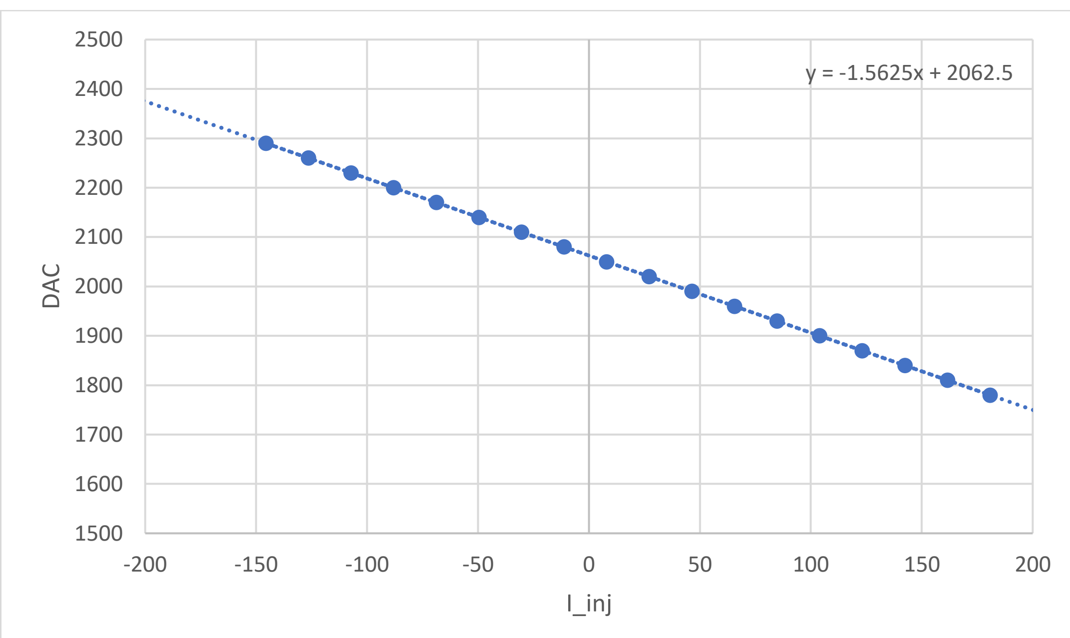
Change to:

```
// DAC outputs between 1800 and 2300 in increments of 30
29 for (int z=0; z<nOutputs; z++) {
30     outputs[z] = 1750+30*(z+1);
31 }
```

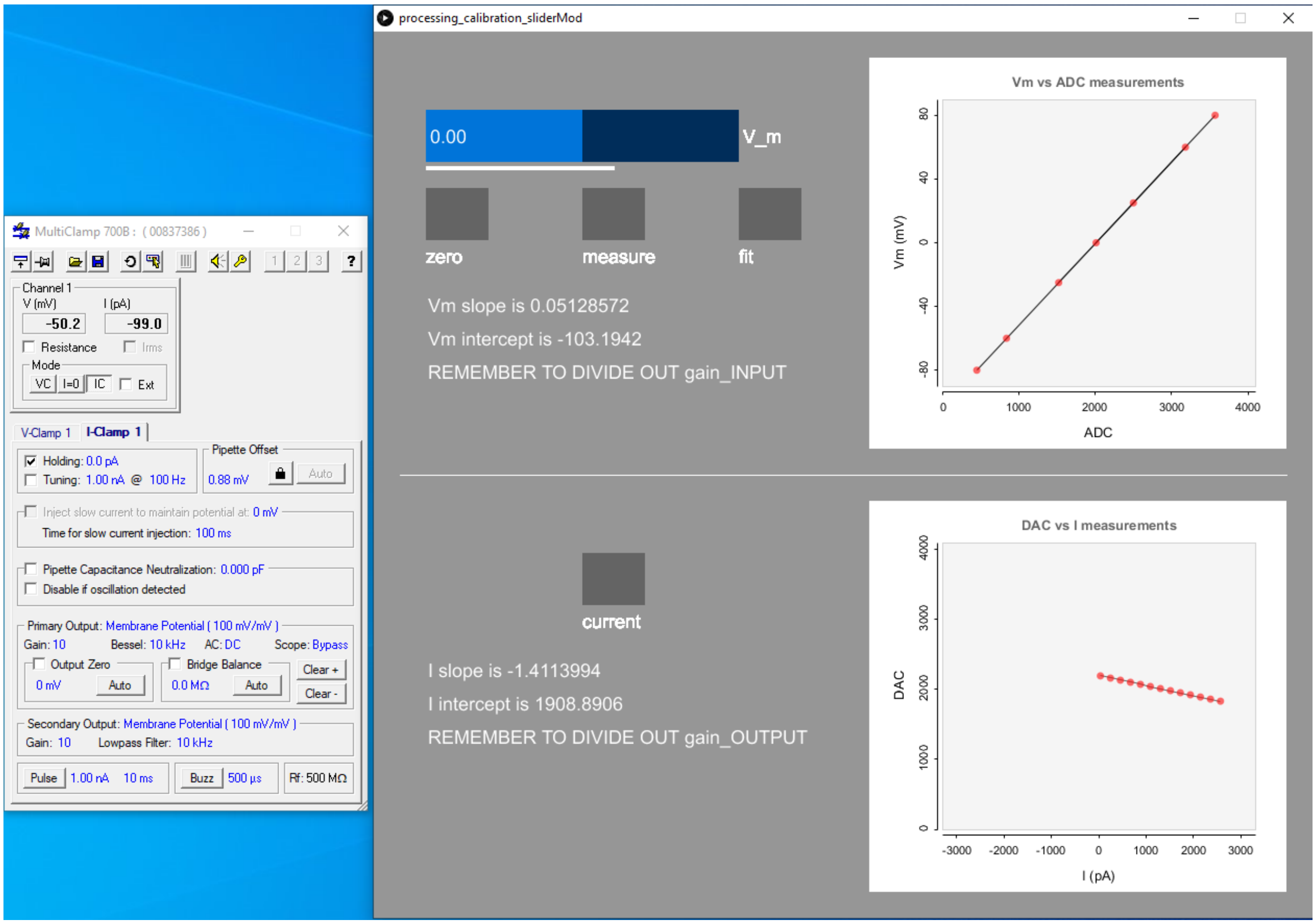
Note: You may need to tweak the increment value a bit to get a range in which all values come out right. Between 1800 and 2300, we need 13 values as nOutputs is fixed at 13, therefore, the right increment would be 30. Do a check, nevertheless.

**Table 10: Measurement of output slope and intercept after tweaking the script**

| DAC | Expected VAMP | Corresponding I_inj | Vm on Model cell (mV) | Vin going into DynClamp (V) | In Range of Teensy? (±10V) | I_inj |
|---|---|---|---|---|---|---|
| 1780 | 0.452 | 180.8 | 92.0272 | 9.20272 | TRUE | 180.8 |
| 1810 | 0.404 | 161.6 | 82.2544 | 8.22544 | TRUE | 161.6 |
| 1840 | 0.356 | 142.4 | 72.4816 | 7.24816 | TRUE | 142.4 |
| 1870 | 0.308 | 123.2 | 62.7088 | 6.27088 | TRUE | 123.2 |
| 1900 | 0.26 | 104 | 52.936 | 5.2936 | TRUE | 104 |
| 1930 | 0.212 | 84.8 | 43.1632 | 4.31632 | TRUE | 84.8 |
| 1960 | 0.164 | 65.6 | 33.3904 | 3.33904 | TRUE | 65.6 |
| 1990 | 0.116 | 46.4 | 23.6176 | 2.36176 | TRUE | 46.4 |
| 2020 | 0.068 | 27.2 | 13.8448 | 1.38448 | TRUE | 27.2 |
| 2050 | 0.02 | 8 | 4.072 | 0.4072 | TRUE | 8 |
| 2080 | -0.028 | -11.2 | -5.7008 | -0.57008 | TRUE | -11.2 |
| 2110 | -0.076 | -30.4 | -15.4736 | -1.54736 | TRUE | -30.4 |
| 2140 | -0.124 | -49.6 | -25.2464 | -2.52464 | TRUE | -49.6 |
| 2170 | -0.172 | -68.8 | -35.0192 | -3.50192 | TRUE | -68.8 |
| 2200 | -0.22 | -88 | -44.792 | -4.4792 | TRUE | -88 |
| 2230 | -0.268 | -107.2 | -54.5648 | -5.45648 | TRUE | -107.2 |
| 2260 | -0.316 | -126.4 | -64.3376 | -6.43376 | TRUE | -126.4 |
| 2290 | -0.364 | -145.6 | -74.1104 | -7.41104 | TRUE | -145.6 |



Measurement after processing code change:




Note: The fitting values are different from the table as this sample plot is generated from another dynamic clamp setup for demonstration.

## Alternative Calibration with a Teensy Program

Another way to measure the input and output slopes and intercepts is to do it manually.

**Setup**: Script to do that can be found in the appendix.

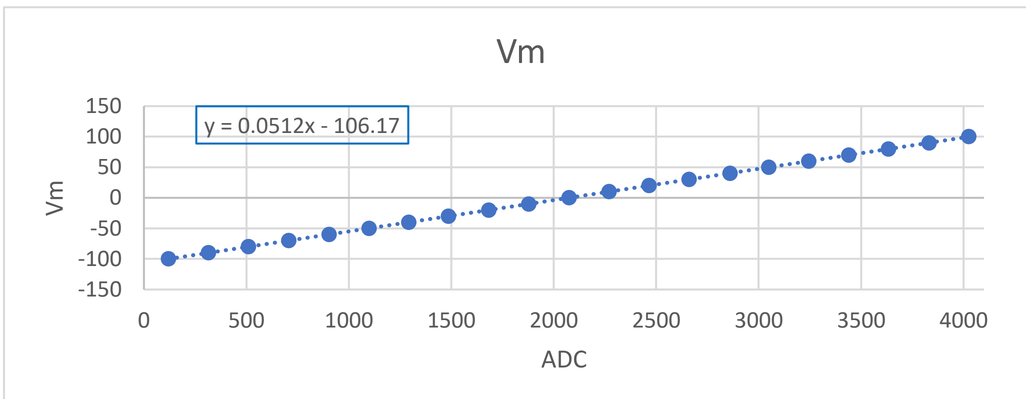
i) Change Vm in current clamp in the multiclamp commander, measure ADC values. Plot and fit a line. Get intercept and slope of the fit. These will be input intercept and slope.

ii) Then, use another script to change DAC values, and measure ADC values. From ADC values get Vm using the relationship obtained in step (i). Get I_inj by dividing Vm with model cell resistance. Plot DAC vs I_inj. Fit a line and get the slope and intercept values from the fit. These will be your output intercept and slope.

**Measurement**: **Input Calibration**

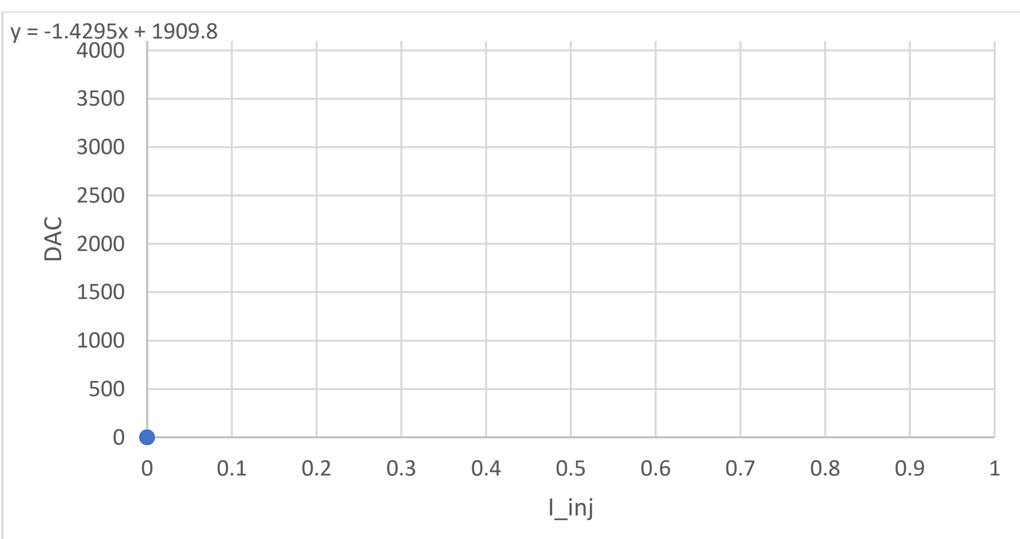
**Table 11**

| ADC | Vm |
|---|---|
| 2075 | 0 |
| 2270 | 10 |
| 2465 | 20 |
| 2660 | 30 |
| 2860 | 40 |
| 3050 | 50 |
| 3245 | 60 |
| 3440 | 70 |
| 3634 | 80 |
| 3832 | 90 |
| 4025 | 100 |
| 1878 | -10 |
| 1683 | -20 |
| 1486 | -30 |
| 1292 | -40 |
| 1098 | -50 |
| 903 | -60 |
| 706 | -70 |
| 510 | -80 |
| 315 | -90 |
| 120 | -100 |
| input slope | 0.0512 |
| input intercept | -106.17 |




**Output Calibration**: Table 12

| i | DAC | ADC | Multiclamp Vm | Multiclamp Im | Vm | I |
|---|---|---|---|---|---|---|
| 0 | 1650 | 3844 | 91 | 181.9 | 90.6428 | 181.7217 |
| 1 | 1675 | 3674 | 82.3 | 164 | 81.9388 | 164.2719 |
| 2 | 1700 | 3504 | 73.6 | 146.9 | 73.2348 | 146.822 |
| 3 | 1725 | 3332 | 64.9 | 129.7 | 64.4284 | 129.1668 |
| 4 | 1750 | 3162 | 56.1 | 111.7 | 55.7244 | 111.7169 |
| 5 | 1775 | 2992 | 47.4 | 94.3 | 47.0204 | 94.26704 |
| 6 | 1800 | 2818 | 38.6 | 76.4 | 38.1116 | 76.40658 |
| 7 | 1825 | 2654 | 29.8 | 59.1 | 29.7148 | 59.57257 |
| 8 | 1850 | 2482 | 21.1 | 41.5 | 20.9084 | 41.9174 |
| 9 | 1875 | 2312 | 12.4 | 24.7 | 12.2044 | 24.46752 |
| 10 | 1900 | 2142 | 3.6 | 6.6 | 3.5004 | 7.017642 |
| 11 | 1925 | 1969 | -5.2 | -11.1 | -5.3572 | -10.7402 |
| 12 | 1950 | 1798 | -13.9 | -28.5 | -14.1124 | -28.2927 |
| 13 | 1975 | 1629 | -22.6 | -45.9 | -22.7652 | -45.6399 |
| 14 | 2000 | 1459 | -31.3 | -63.5 | -31.4692 | -63.0898 |
| 15 | 2025 | 1288 | -40 | -81.2 | -40.2244 | -80.6423 |
| 16 | 2050 | 1118 | -48.8 | -98.6 | -48.9284 | -98.0922 |
| 17 | 2075 | 945 | -57.6 | -116.1 | -57.786 | -115.85 |
| 18 | 2100 | 776 | -66.3 | -133.8 | -66.4388 | -133.197 |
| 19 | 2125 | 607 | -75 | -151.8 | -75.0916 | -150.545 |
| 20 | 2150 | 437 | -83.8 | -168.6 | -83.7956 | -167.994 |
| 21 | 2175 | 267 | -92.5 | -186.2 | -92.4996 | -185.444 |
| 22 | 2200 | 97 | -101.2 | -203.6 | -101.204 | -202.894 |
| | | | | | | |
| | | | | | | |
| Output Slope | | -1.4295 | | | | |
| Output Intercept | | 1909.8 | | | | |



## Conclusion

The whole objective of all the step by step and stage-by stage exercise was to finally, determine the relationship between what Dynamic Clamp measures and what Dynamical Clamp maps it to. It has to map the readout on it is pin 0, i.e., ADC to $V_m$ and also map its output from DAC onto the injected current $I_{inj}$.

In the dynamic clamp setup that was made along with the writing of this manual, the input-output intercepts and slopes came out to be:

**Table 13**

| | |
|---|---|
| Input Slope | 0.0513 |
| Input Intercept | -103.27 |
| Output Slope | -1.4321 |
| Output Intercept | 1913.6 |

Finally, we have to put these values into the *dynamic_clamp.ino* file.

**Summary of Values, errors in different measurements:**

Either of the green shaded column values can be used.

| | Theoretical | Expected From Individual Modules | Direct From Individual Modules with Function Gen | Whole Setup with Processing GUI and Model Cell | Whole Setup Direct with Code and Model Cell | % Error Between Theoretical and Direct with Code |
|---|---|---|---|---|---|---|
| Input Slope | 0.050 | 0.050 | 0.050 | 0.051 | 0.051 | 1.617 |
| Input Intercept | -103.146 | -103.922 | -99.892 | -103.270 | -106.170 | 2.932 |
| | | | | | | |
| Output Slope | -1.458 | -1.456 | -1.449 | -1.432 | -1.430 | -1.981 |
| Output Intercept | 1925.083 | 2008.597 | 1934.300 | 1913.600 | 1909.800 | -0.794 |

## Feeding the calibration values in Dynamic Clamp

The input and output intercepts and slopes obtained in sections above and summarized in the last section need one more modification before being put into the dynamic clamp program. Recall from Equation 6 and Table 13 for input that:

$$\text{Input Slope} = \left(\frac{3.3}{4096 * \mathrm{m_{in}} * \mathrm{g_{in}}}\right) = 0.0513$$

And

$$\text{Input Intercept} = \left(-\frac{\mathrm{c_{in}}}{\mathrm{m_{in}} * \mathrm{g_{in}}}\right) = -103.27$$

As one can see, both quantities are dependent on amplifier input gain $g_{in}$. To make the code more general to be used with different values of input gain, we can take $g_{in}$ out and use it as another variable.

$$\text{Input Slope} = \left(\frac{3.3}{4096 * \mathrm{m_{in}} * \mathrm{g_{in}}}\right) = \frac{5.13}{g_{in}}$$

And

$$\text{Input Intercept} = \left(-\frac{\mathrm{c_{in}}}{\mathrm{m_{in}} * \mathrm{g_{in}}}\right) = -\frac{10327}{g_{in}}$$

Similary, to make the dynamic clamp code more general to be used with different values of gout, we need to factor out gout from equations 8a and 8b.

$$\text{Output Slope} = \frac{4096}{3.3 * \mathrm{m_{out}} * \mathrm{g_{out}}} \qquad (8a)$$

$$\text{Output Intercept} = -\frac{4096 * \mathrm{c_{out}}}{3.3 * \mathrm{m_{out}}} \tag{8b}$$

From 8b, we can see the output intercept does not depend on gout. The final value from 8a and Table 13:

$$\text{Output Slope} = \frac{4096}{3.3 * \mathrm{m_{out}} * \mathrm{g_{out}}} = -\frac{1.430 * 400}{g_{out}} = \frac{572}{g_{out}}$$

$$\text{Output Intercept} = -\frac{4096 * \mathrm{c_{out}}}{3.3 * \mathrm{m_{out}}}$$

Rest all constants, min, mout, cin, cout, 4096, and 3.3 are property of the circuit and are not changeable from outside.

Feed the final values in *Dynamic_Clamp.ino* file:

```
const float inputSlope = 5.12/gain_INPUT;
const float inputIntercept = -10617/gain_INPUT;
const float outputSlope = -571.8/gain_OUTPUT;
const float outputIntercept = 1909.8;
```

## Appendix

### Script for alternative calibration using a Teensy program

```
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/*
  Analog input, analog output, serial output
  Aditya Asopa, Bhalla lab, NCBS Oct 2020
*/

// These constants won't change. They are used to give names to the pins
used:
// Analog input pin that the Vm is read from
const int analogInPin = A0;
// Analog output pin that the DAC goes out from
const int analogOutPin = A21;

int sensorValue = 0;        // value read from the ADC
int outputValue = 0;        // value output to the DAC

void setup() {
// initialize serial communications at 115200 bps:
  Serial.begin(115200);
  analogWriteResolution(12);
//very important to set the bit resolution
  analogReadResolution(12);
}
/*Teensy cant read any value of Vm that is beyond +/-100mV.
 * Since the model cell has a resistance of 500MΩ, a Vm of 100mV is
produced with a current of 200pA.
 * Therefore, IDC has to be between +/- 200pA.
 * For that, corresponding DAC has to be between 1800 - 2300 (check the
previous sheet for scaling).
 * Therefore, the right range of DAC to give here is 1750 to 2300.
 */
void loop() {
  for (int i = 0; i <= 20; i++) {
//outputSlope * injectionCurrent + outputIntercept;
    int Iinj = i*30+1750;
// make sure the output is an integer between 0 and 4095 (12 bits)
    int DAC = constrain((int)Iinj,0,4095);
    analogWrite(analogOutPin,DAC);
    delay(1000);
    int adc = analogRead(analogInPin);
```

```
    Serial.println(i);
    Serial.print("\t");
    Serial.print(DAC);
    Serial.print("\t")
    Serial.print(adc);
    delay(1000);
  }
}
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

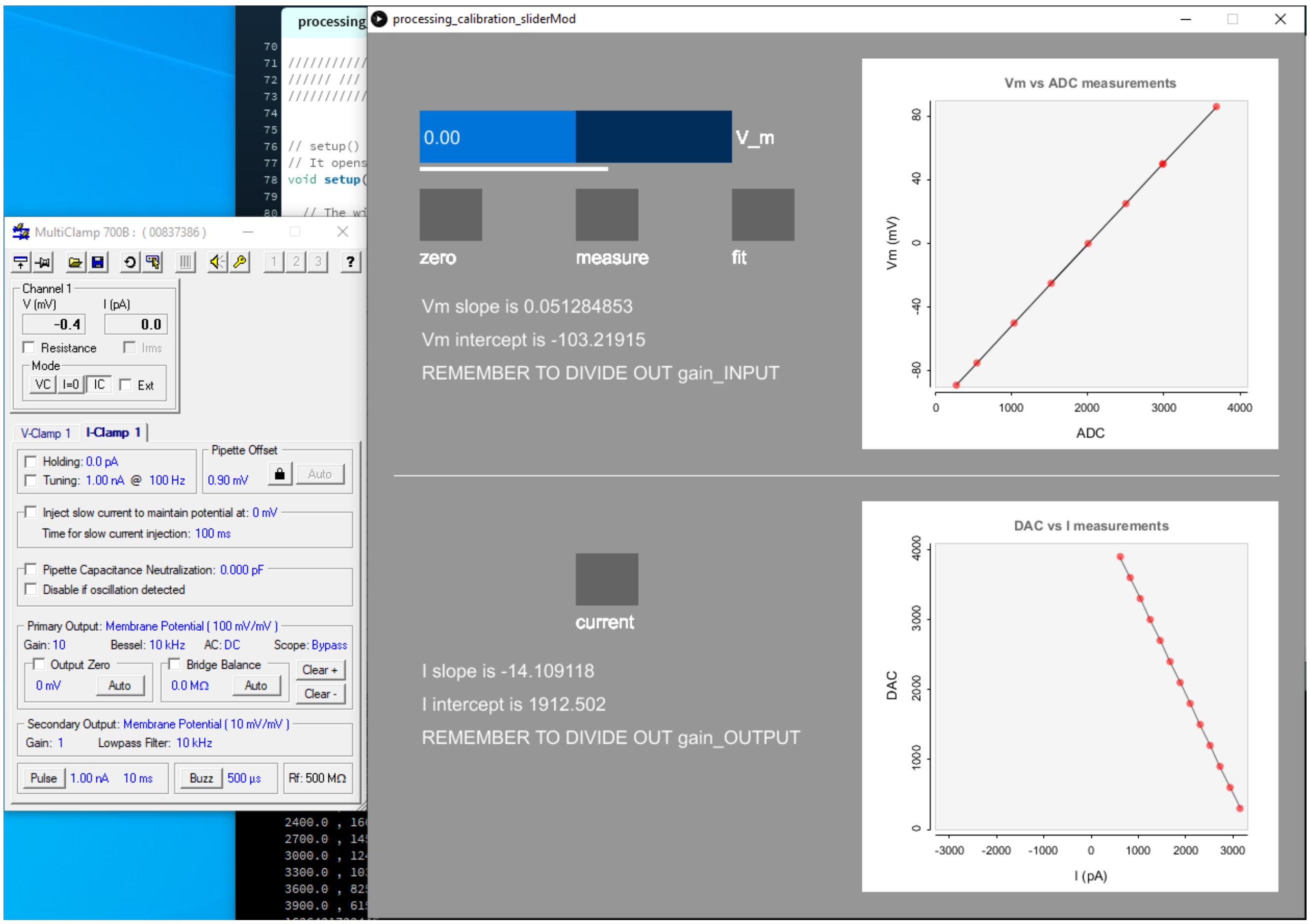
## Bonus method to solve the output calibration problem with the processing calibration program

As described in the section, the problem arises due to huge resistance of the model cell, the Vm drop after amplification goes out of the range of Teensy ADC. We can therefore also solve the problem by changing not the resistance of model cell but by changing the gain of the amplifier and then factoring it back in (recall Eq. 8)

Steps:

1. Keep the input gain at 10 (100mV/mV). Measure the input slope and intercept like usual.
2. Change the gain to 1 (10 mV/mV). Run the output stage calibration.
3. Factor back the gain into the calculation before plugging into dynamic clamp program.

Sample result:




So, the correct output values to be put into the *dynamic_clamp.ino* program will be:

Output slope = -1.410911/gout and
Output intercept = 1912.502

*See if you can figure out why output slope had to be divided by 10 and intercept was left as is. Hint: input gain.*