Amplifier-Based Photodiode Analog-to-Digital Converter System

Prof. Higman, TA: Robert Bloom

Eric Minnerath, Akshay Naik, Archit Kalla, and Carl Anderson

12/09/2021

Table of Contents

Abstract

The purpose of this project is to design a transresistance photodiode circuit that uses an analog to digital converter to read the output of the photodiode sensor at a 1nA resolution. The proposed design can be split into 5 main parts: a transresistance amplifier, analog to digital converter (ADC), microcontroller, button tare, and LCD display. Noise reduction, a primary design constraint, is achieved through a filter after the amplifier along with a voltage regulator and buffer from the wall power source. Amplification and conversion are done using off-the-shelf integrated circuits (ICs). System functionality is performed on a microcontroller through the use of serial communication with the other components on a custom-designed printed circuit board (PCB). The design meets specification while remaining under $50.00 USD.

Introduction/Background

This design includes 5 main components:

Transresistance Amplifier - The operational amplifier is the first component that the photodiode interacts with. Its responsibility is to convert and amplify an input current to an output voltage with minimal noise.

ADC - The ADC, or analog to digital converter, takes the voltage output from the amplifier and converts it to a digital signal that our microcontroller knows how to deal with.

Microcontroller - The microcontroller is the brain of the operation. It connects the analog side of our project with the various digital components we are using as input/output. It reads the digital value from the ADC via SPI, displays that value on the display via I^2C, and tares at the current value if the tare button is pressed. For debugging purposes, it also outputs raw ADC data via UART.

LCD Display - The LCD display simply displays what the microcontroller is currently reading from the ADC.

Tare - The tare function acts as a way for the user to zero out the display. Oftentimes there are various input offset errors in the system that require a reset.

Group/Individual Contributions

Group - Part selection, high-level design, hardware debugging

Akshay Naik - Noise analysis, soldering

Archit Kalla - Analog design

Carl Anderson - PCB design, soldering

Eric Minnerath - Firmware design

Product Design Specification

Qualitative Statement

Construct a transresistance amplifier-based analog to digital converter (ADC) system to read the output of a photodiode sensor.

Design Specification

1.) Input signal: Optical photodiode sensor similar to the Excelitas VTP9812FH (talk to us first if you want to use a different diode) which is a photodiode sensitive to visible spectrum light (400nm-700nm). Note that this diode has a peak response at 580nm (yellow) of 34mA/W. The expected output from the photodiode is on the order of 0-4μA.

2.) The input signal should be gained up using a current to voltage (transresistance) amplifier and then converted to a digital signal via a 24 bit A/D. The current generated by the photodiode should be displayed on a decimal display (LCD, 7-segment, etc.) in units of μA, to a usable resolution of 1nA. Strive to have no noise at the nA level. In other words, you will need a display with a minimum of 4 digits plus a decimal point. In addition to the display, the raw 24 bits of ADC data should be made available via UART to show the useful number of bits. Please format the UART transfer as ASCII characters "1" or "0" for each bit, most significant to least significant. If connected to a serial terminal the output should read eg. "001100101101111100100100 /n".

3.) The output of the photodiode can be considered quasi-DC signal and the display and raw data output should update every 200-300msec.

4.) There should be a way to zero out the display initially to compensate for various input offset errors in your system. This tare function should be activated via a user input (button, switch, or other).
5.) Your design should be an all-in-one unit including the photodiode, analog and digital circuitry, display, user input(s) and power source. The design should be mounted in an enclosure/box with cutouts for the photodiode, display and user inputs. It should be powered by either batteries or a wall adapter. Your design, including components and enclosure, should be within a $50 budget. The PCB will be provided and is not counted in the budget.

Proposed Design

Overview

The proposed design below utilizes a variety of ICs and passive circuit elements to be installed on a PCB.

Key Components

The analog components of the design serve the purpose of amplifying the output of the photodiode and converting the amplified analog signal to a serial digital signal. With this amplification, we had to account for noise from various sources highlighted in the Noise Analysis section. The first analog component is the transresistance amplifier. We know that the photodiode outputs anywhere between 0-5 microamps in a reverse bias and we need to convert current to a usable voltage range (0-1V). To achieve this, a suitable resistor was required that would provide enough gain to the signal to get us to 1V at the max of 5 microamp input. (note: Because we needed to reverse bias the photodiode, a 1V DC voltage source was needed in the positive input of the opamp, hence at 0 A input, the opamp will output 1V). Using the following equation:

$$-I = \frac{V_{out}}{R_f}$$

$$R_f = \frac{V_{out}}{I}$$

Figure 2: Op Amp feedback resistance equations

Plugging I=-5uA and V_{out}=1V, we found the resistor value to be 200k ohms. After finding this value, we implemented an RC lowpass filter with a corner frequency of 15Hz to compensate for the noise generated by both the op and resistors. We opted for a 1 microfarad capacitor to minimize the size of the resistor which would introduce more noise at larger values.

The output of the amplifier and filter is then fed into the ADC. The ADC will then convert this analog signal to a serial digital signal to be sent over SPI communication to be then used by the microcontroller.

The display selected was the same as the one used in EE 2361. It communicates with the microcontroller via I2C and contains two rows of 8 ASCII characters. This is sufficient to show the number of decimal places of uA along with the text " uA". Above all, software development time is reduced as libraries are already written for it.

The system gets power from a wall outlet. Wall power was the best solution for this design as the noise from something like a 5V USB breakout would be too large and difficult to rectify. The wall power is passed through a wall adapter along with a voltage regulator before supplying VDD and VSS to almost every component on our PCB.

The tare is a button that pulls a pin on the microcontroller low and triggers an interrupt in the firmware. We will perform debouncing logic in software, a tradeoff to the simplicity and price of the button.

The microcontroller connects to everything in the system and had to be considered carefully. As shown in Figure 1 below, the main considerations were pin count and serial interface support.

Function	Required Pins
Vdd, Vss, MCLR, ICSPDAT, ISCPCLK	5
I2C (SDA, SCL)	2
SPI (SCLK, SDI*, SDO, SS)	3-4
UART (TX, RX*)	1-2
Tare	1
Total	14

Figure 1: Microcontroller pinout requirements

*RX implementation is optional since the requirement is only to broadcast data.

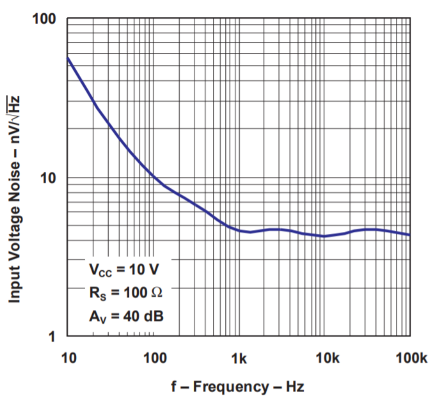
Once there was an interface and pin count requirement, it came down to choose a specific model. The Microchip PICs offered familiarity from E2361. The first candidate was PIC24FJ64GA002, but this was expensive and overpowered for the situation. Then narrowed the scope to 8-bit PICs. The second candidate was PIC16F15243, but it only had one MSSP and preferred to not implement I2C in software via “bit-banging”. For less than 50 cents more, the PIC16F18326 was selected. It had the exact pin count required while containing dedicated peripherals for UART, SPI, and I2C.

<u>Noise Analysis</u>

Within the noise analysis section, we plan to calculate the noisy bits and the usable bits of data. The outcome of these calculations determines the theoretical resolution of the bits. We conduct these calculations in a 7 step approach.

1. Confirm Op-Amp Model

GBW = 12MegHz, Avol = 10000V/V, Rail = 0.1V, Slew = 5MegV/s Vos = 4mV, PhMargin = 60,
En = 4nV/sqrt(HZ), Enk = 500Hz



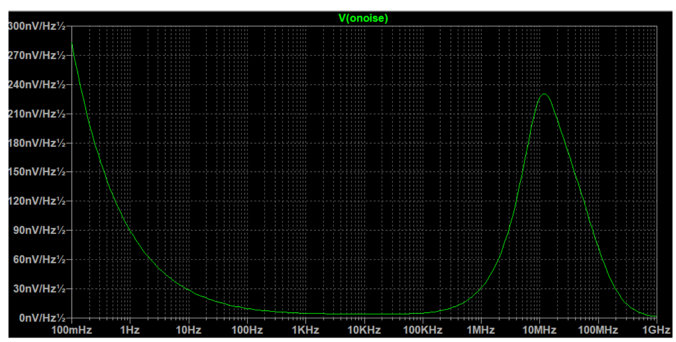


Figure 6. Input Voltage Noise vs Frequency

Hz	Datasheet	LT Spice
10	54.67 nV/sqrt(HZ)	28.8 nV/sqrt(HZ)
100	8.87 nV/sqrt(HZ)	9.78 nV/sqrt(HZ)
10K	3.31 nV/sqrt(Hz)	4.19 nV/sqrt(Hz)
100K	3.6 nV/sqrt(HZ)	5.05 nV/sqrt(HZ)

The LTSpice model has lower values than the Datasheet in some instances and vice versa due to minor discrepancies. In the table above, we see that the initial Input Voltage Noise starts off higher on the datasheet. Once it reaches 100Hz, the values start to conform with the LTSpice simulation showing that the Op-Amp model is consistent with the one being used in the other LTSpice simulation.

2. Simulate Front End

Amplifier = 351.21nVRMS **Simulated using LTSpice*
ADC = 1.5uVRMS **Given in Datasheet*

Total Noise = **1.54uV**

The front end noise is calculated by adding the amplifier noise to the ADC noise. The amplifier noise is calculated by taking the area underneath the noise analysis done on the Op-Amp. Taking this value and adding it to the ADC value given by the datasheet gives the front end total noise.

3. Determine Value of LSB

FS = |-2| + 2 = 4V **Range of ref voltage*
LSB = FS/2^{24} = 4/2^{24} = 238.42nV =~ **238nV**

The full scale input of the ADC is used as the range of the reference voltage. This value is divided by 2^{24} to determine the least significant bit.

4. Number of bits needed to represent noise

Noise bits = $\log_2$(Total Noise/LSB) = $\log_2$(1.54uV/238.42nV) = 2.691 **round up*
Noise bits = **3 bits**

The number of bits needed to represent the noise is calculated by taking the total voltage, dividing it by the least significant bit and taking the $\log_2$ of that value. Rounding this bit value will give the number of noisey bits present.

5. Number of usable bits

Usable bits = Total bits - Noisy bits - ADC lost bits
24 - 3 - 2 = **19 bits**

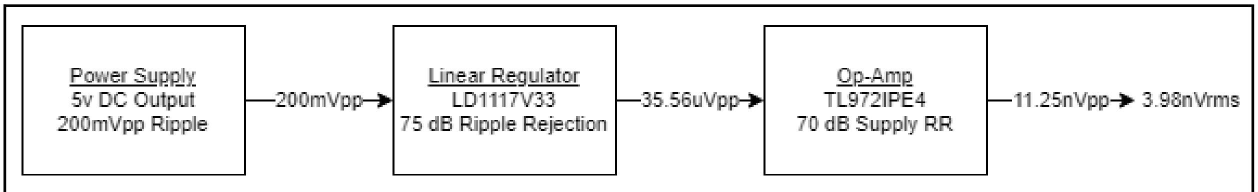
The number of usable bits is represented by the total bits minus the number of noisy bits which leaves us with 19 bits. ADC lost Bits are a result of setting the negative terminal of analog input to ground and our REFIN to 1, we lose 1 bit for negative values and 1 bit for values in between 0V-1V.

6. Effective LSB Calculation

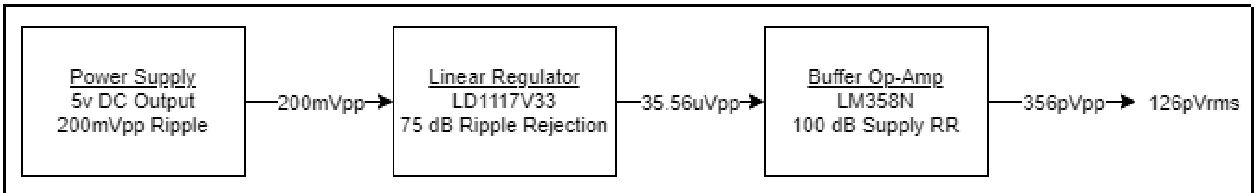
LSB = $FS/2^{\text{Usable Bits}} = 4/2^{19}$ = **7.629uV**

7. Determine Resolution

Resolution = LSB / Gain of Amplifier = 7.629uV/ 200k = **38.1pA**



Power supply ripple noise analysis at transresistance amplifier
3.89 nV_{RMS} << Noise at output of op-amp



Power supply ripple noise analysis at buffer op-amp
126 pV_{RMS} << Noise at output of op-amp

Budget

Name	Qty.	Purpose	Price
PIC16F18326-I/P	1	uC	$1.56
AD7789BRMZ	1	ADC	$7.95
TL972IPE4	1	OP Amp	$0.95
10.6k resistor	1	LPF	$0.26
1uF Capacitor	1	LPF	$0.32
AQM0802 (ECE DEPOT)	1	Display	$7.75
Bud Box (ECE DEPOT)	1	Box	$7.00
1k resistor	2	Voltage divider	$0.10
1.5K RESISTOR	1	Voltage divider (2V)	$0.10
2.2k resistor	1	Voltage divider (1V)	$0.10
LD1117V33	1	Volt Reg	$0.70
Button	1	Button	$1.04
200k resistor	1	Rf	$0.12
VEL05US050-US-JA	1	Wall adapter	$6.50
Terminal Block	1	Connection	$1.17
Capactior (0.1uF Ceramic)	2	Decoupling	$0.32
Capacitor (10uF Electrolytic)	1	Decoupling	$0.10
uC Capacitor (10nF Ceramic)	2	uC Circuit	$0.32
uC Resistor R1 (10K ohm)	1	uC Circuit	$0.12
uC Resistor R2 (100 ohm)	1	uC Circuit	$0.10
LM358N	1	Op Amp for Buffer	$0.30
PA0027	1	Breakout for adc	$4.09
R1 10k	1	R1 for microcontroller	$0.10
R2 100Ohm	1	R2 fo microcontroller	$0.10
VTP9812FH (ECE DEPOT)	1	Photodiode	$1.02
		Total:	**$42.93**

Final Design

Analog

A major difference between the proposed design and the final design for the analog components was how the input into the amplifier. Initially we designed the amplifier such that the input signal was going through the positive terminal. This would have resulted in positive feedback which means that there would have had outputs that were either 0V or VDD rather than an 0-1V. To rectify this problem the schematic was modified and made the input and feedback go to the negative terminal of the amplifier.

Testing the analog design on a breadboard we found that when presented with no light the amplifier output was 1V which is what we expected. Testing the higher end of the system, showing light to the photodiode we found that when our amplifier displayed 2V we calculated that the output of the photodiode was 5 uA thus matching our equations. Placing this circuit on the PCB also had no effect on the output of the amplifier, further proving that the schematic was correct.

Digital

The digital system didn’t change significantly between our proposed design and the final design. As planned, we used a PIC16F18326 microcontroller to connect our ADC, LCD, and tare functionalities. The ADC uses SPI and the LCD uses I2C. The only change is that we chose to poll the tare button in the main loop instead of map to an interrupt. An interrupt would have worked fine, but we didn't want to deal with coordination on where in the main loop the processor was. On startup, the LCD is initialized and the ADC is reset. The main loop goes as follows:

1.) If the tare button has been pressed, adjust the tare offset variable
2.) Wait for ADC to signal data ready. (falling edge of MISO)
3.) Get data from the ADC
4.) Transmit data to UART, scale data down, and display to LCD
5.) Delay for 2 ms.

Consideration was done to use interrupts for ADC data acquisition, but polling was easier to implement and remained within specification of the project.

Power

The power system was also implemented in the final design the same way which we planned in our preliminary design. We used an AC to DC wall adapter to step down the 120VAC wall power to 5V DC power which was then fed into a low dropout regulator which brought our 5V DC rail down to the 3.3V DC that is required for the LCD. All of the other components were 3.3V-5V tolerant but given the LCD was on 3.3V and we did all of our analog calculations with GND and 3.3V rails it made our life much easier to just use 3.3V for all of our devices.

<u>User Interface</u>

The final user interface design was slightly different than proposed. We had to change our button out at the last moment because we accidentally broke off one of the pins while soldering it so we just replaced it with a standard cheap breadboard button, the functionality of the tare button is unchanged and works to add a tare to the current measurement. The LCD was implemented just as expected, a post message first appears once the initialization of the microcontroller is completed and then it goes to displaying the measured photodiode current in the format “X.XXX uA” until the device is powered down.

Results

<u>Analog</u>

In our noise analysis we expected to have 19 accurate bits of data, however we wound up with only 17 bits of usable data. After presenting the system with a constant light source and manually reading the output of the UART we found that the least significant 7 bits were noisy, which leaves us with 17 bits of precision. This was 2 bits off of our predicted 19 bit precision leaving us with a maximum of 4 times the noise that we predicted. The lost two bits in our tests can be attributed to human error. First there was a fan that would slightly move the photodiode, and because the photodiode is directional-sensitive, the data could have fluctuated. We also used a fluorescent lamp rather than a stable LED, providing a slightly inconsistent amount of light increasing the chance of error.

Digital

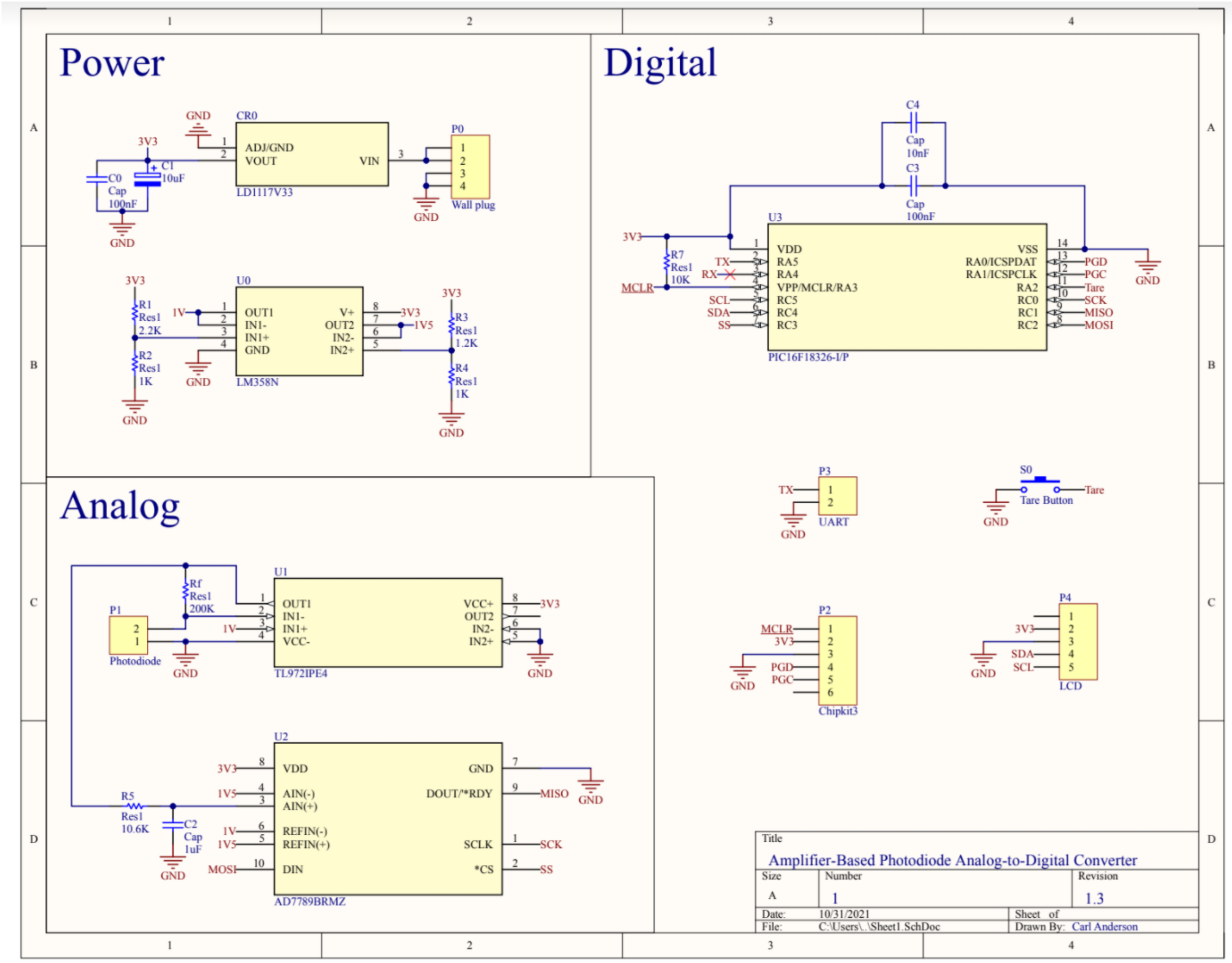
The rate of data acquisition in the main loop can fluctuate a little based upon how recently the ADC pulled the data ready line low. Our loop waits for a falling edge, so if that just happened one cycle earlier (the worst case), we have to wait 60ms to get data.

Conclusion/Summary

In conclusion, the proposed system above will perform conversion and amplification of a photodiode signal. Due to our component selection and noise reduction measures, we will meet the precision and noise requirements. The system will also cost less than $50 in components.

Appendices

Schematic

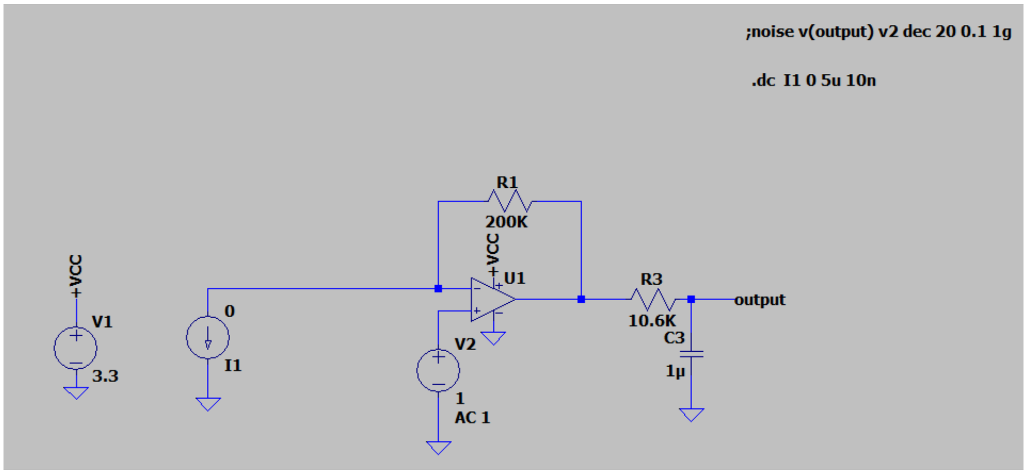


Datasheets

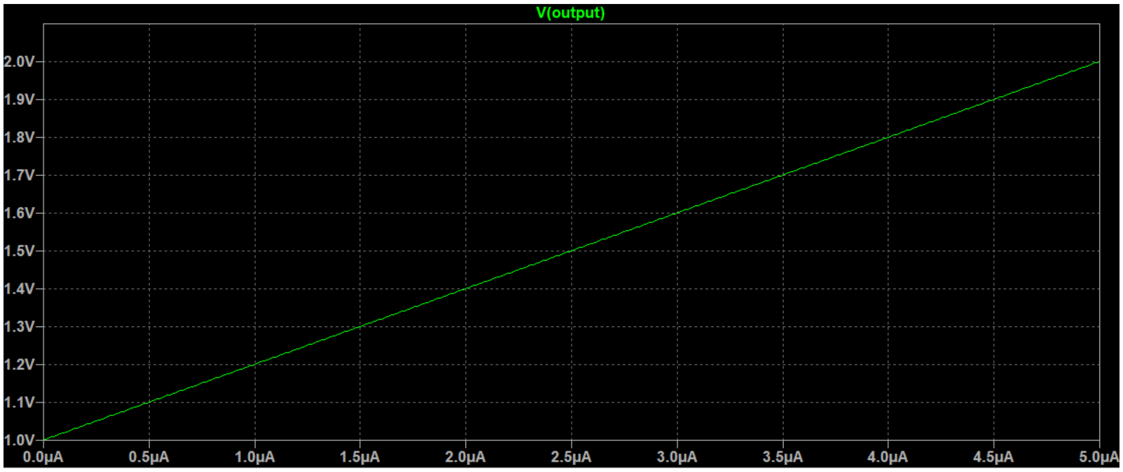
PIC16F18326-I/P	Button	PA0027
AD7789BRMZ	200k resistor	R1 10k
TL972IPE4	VEL05US050-US-JA	R2 100Ohm
10.6k resistor	Terminal Block	LD1117V33
1uF Capacitor	Capactior (0.1uF Ceramic)	LM358N
AQM0802	Capacitor (10uF Electrolytic)	
Bud Box	uC Capacitor (10nF Ceramic)	
1k resistor	uC Capacitor (10nF Ceramic)	
1.5K RESISTOR	uC Resistor R1 (10K ohm)	
2.2k resistor	uC Resistor R2 (100 ohm)	

Analog Front End SPICE Simulation

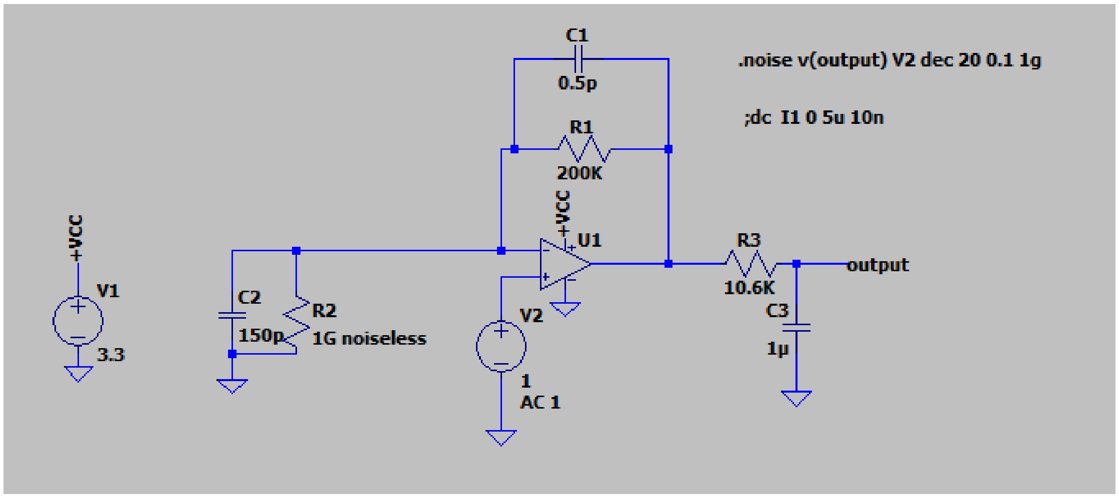
The circuit first tests the analog components within the circuit. A DC sweep of this circuit is done and the results are shown below.



Noise analysis clipping plot. The integral is taken over this range to calculate the amplifier noise.



Complete analog component circuit with noiseless resistors and capacitors included.



The output of this circuit is measured and calculated to create a noise output graph seen below.