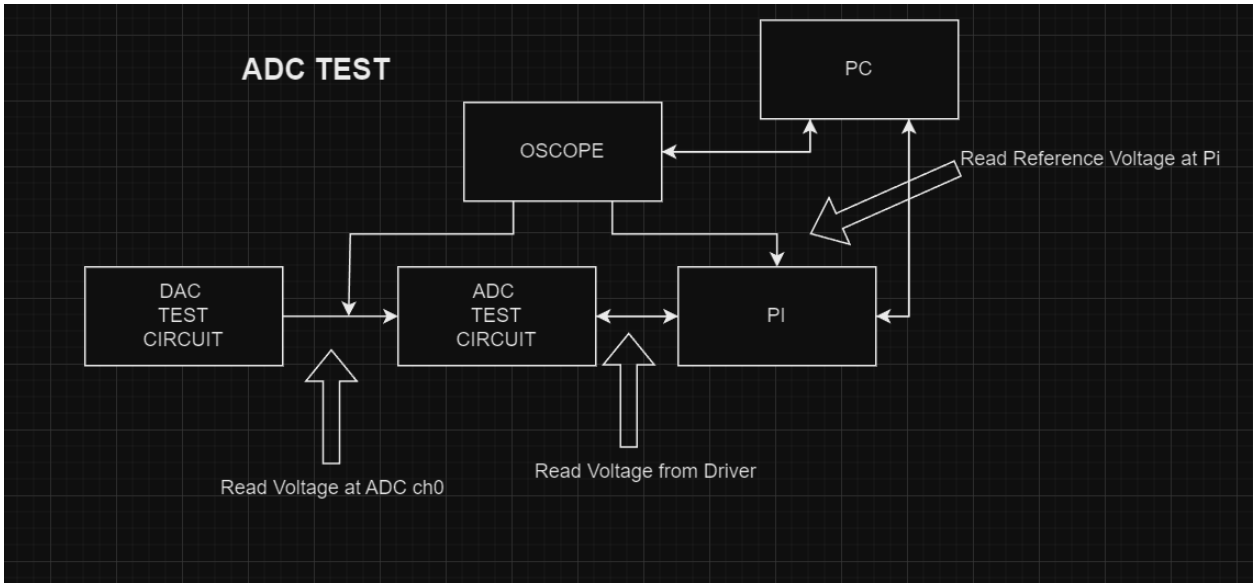
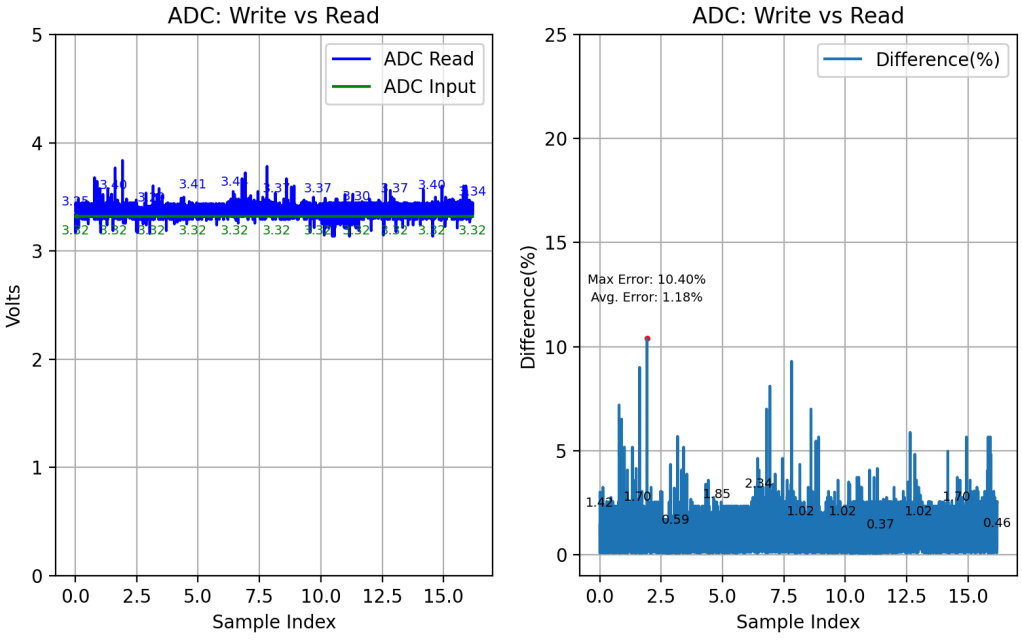
ADC Verification



Updates made to test

- Pi script for ADC time is 40ms -> 0.2ms with c++ file handling
- Python script improved +20ms -> 2ms read time with separate read and processing

Concluded First Test and Got These Results



Even though avg error is 1.18% doing some excel math I got

	avg	std
3	1.18	0.92
5	upper avg	lower avg
3	2.10	0.26
2	mode(% error)	median
9	5.07	5.07

Showing that even though the average is 1.18% avg is considered to >2% which exceeds our limits we will add hardware improvements before looking into software filtering methods

Tests needed

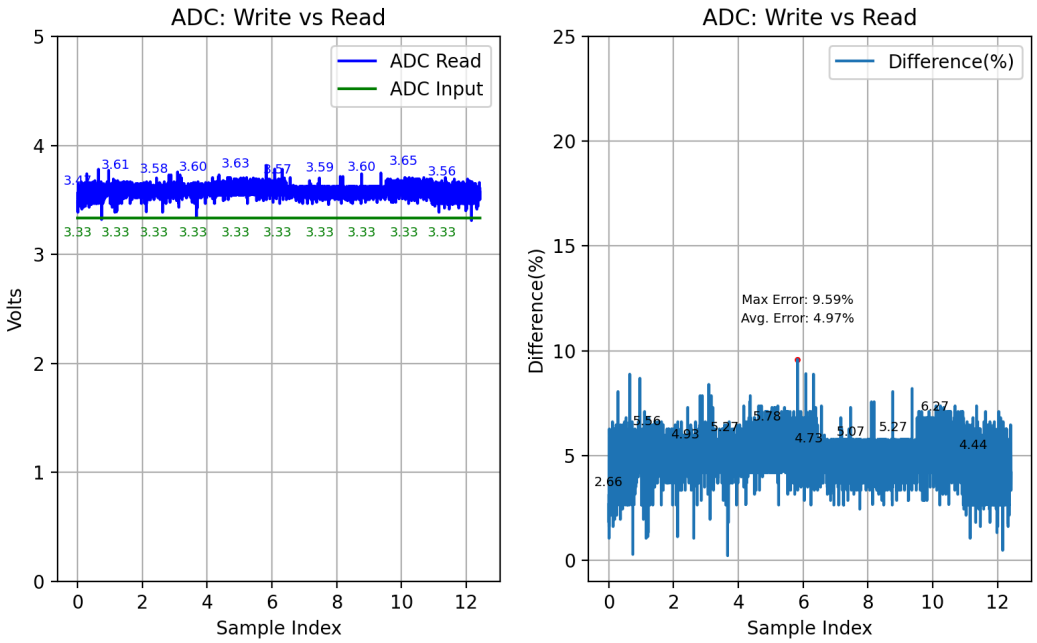
- steady DC Voltage with 1.1%
-sweep DC Voltage with 1.1%
- Transient AC response with 1.5%
- Max Frequency > 3kHz

Better Measurements

- better grounding about 10-15mV off
- better reference voltage
- Did 60Hz coupling cause an issue
- Wiring can be an issue

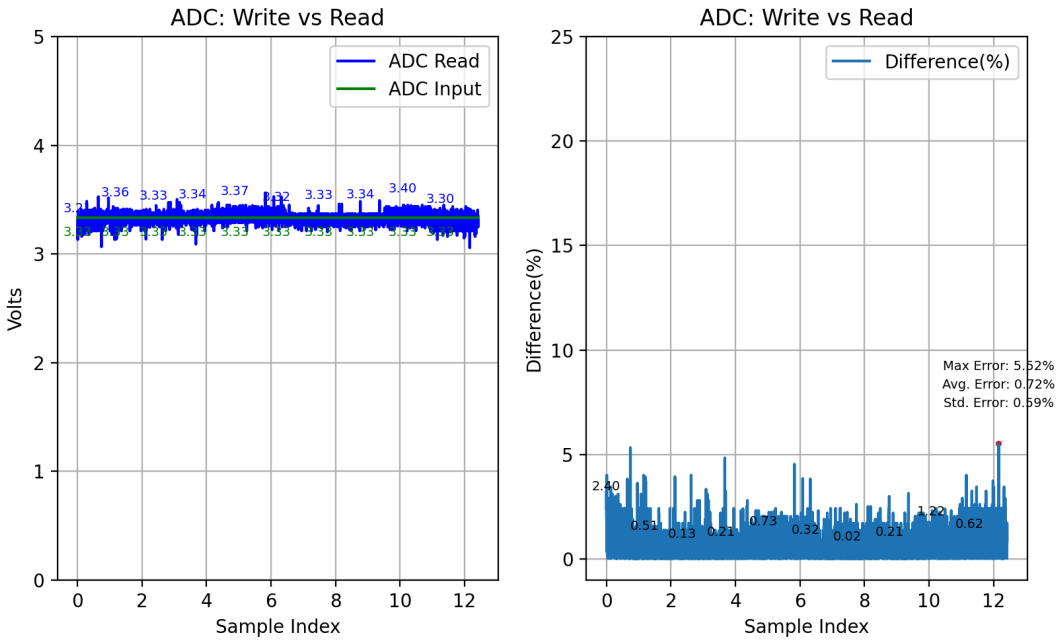
We are close enough that I feel comfortable moving forward with other tests and hardware

This was my second attempt at testing with a focus on improved grounding by connecting more ground and twisting ground wires with other signals



These are the results. Why it is more inaccurate I am not sure but excel math tells us that the std from the average is significantly better

First test had a an average error of 1.18 % std dev of 0.92% so double the % error error so that so that 2% variations This test has average error of 5.101% but std of 0.05% meaning voltage was extremely steady in the set up with a variation of 0.05% or 2.5mV while the old test saw a so a change of 46mV. If we subtract each entry by the average amount we get an error rate of 0.72%.



These are well within our limits so are next move is to find a way to guarantee a constant voltage reference and voltage output.

So I know we can get a error rate of 0.72% and a std of .05
Look on datasheet about reference voltage regulation-> bypass caps will come in handy