KEK

量子場計測システム国際拠点
International Center for Quantum-field Measurement Systems for Studies of the Universe and Particles

KEK/QuP Research on Direct Optic Fiber Feedthroughs into Evacuated Systems

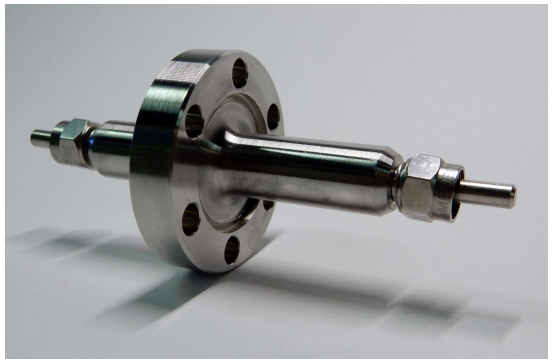
Suerfu Burkhant, Alex Drummond

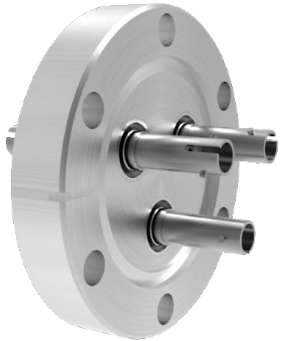
Quick Introduction

VIRGINIA TECH

Research Motivation

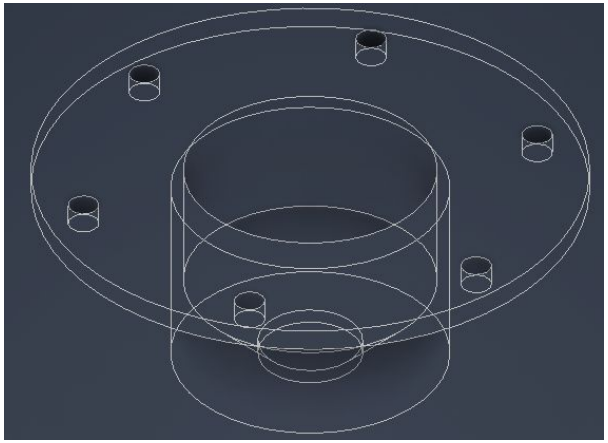
- Existing feedthroughs (right) are limiting
- Cost, installation, complex
- Many use epoxy resin to create a vacuum seal on the cable itself

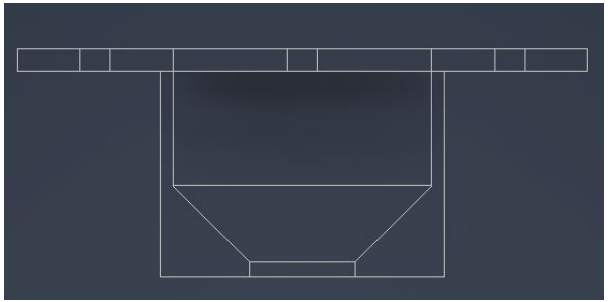




Research Motivation

- We decided to test the effectiveness of a direct silicone feedthrough for optic fiber cables into evacuated systems



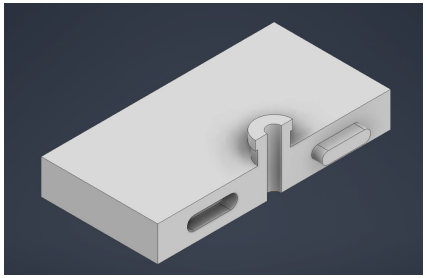


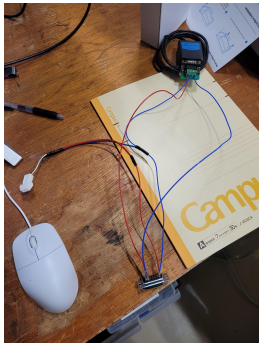
CAD wire representation of silicone filled feedthrough

Experimental Setup

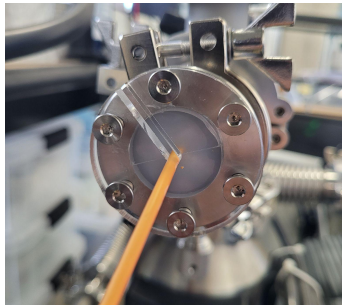
- 3D print pieces to hold cable as silicone cures (CAD image on right)
- Laser cut acrylic for silicone molding, compression
- Wire configuration for RS485 to USB communication

Stereolithography 3D printer, washer, UV cure





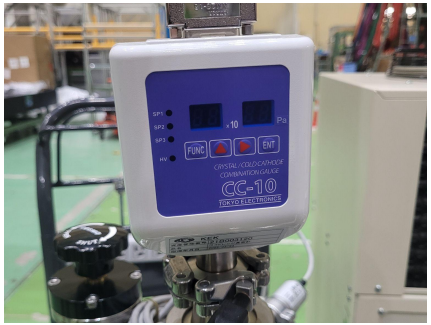
RS485 to USB wiring



Feedthrough example

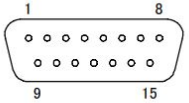
Experimental Setup

- Our Gauge; CC-10 Televac
- RS485 to USB cable (Innomaker)
- Gauge/Innomaker pin configuration
- Connected with DB9/DB15 soldered pieces



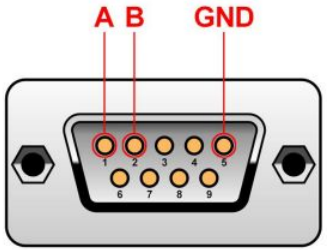
13. I/O コネクタ

外部I/Oコネクタのピン配列、信号内容を示します。(D サブコネクタ 15pin オス)



I/O

ピン番号	名　称	入出力	信号内容	
8	+24V	入力	DC+24V 電源入力	
15	COM		DC+24V COM 入力（0V）	<注1>
7	EXT V_{CC}	入力	外部電源入力（DC+24V）	
14	IN1		大気圧校正トリガ入力	
6	IN2		手動ゼロ校正トリガ入力	<注3>
13	IN3		HV INHIBIT 信号	
9	ANALOG OUT+	出力	アナログ電圧出力	<注2>
1	ANALOG OUT−		アナログ GND	<注1>
5	SP1	出力	セットポイント 1 出力	
12	SP1 COM		SP1 COM	
4	SP2	出力	セットポイント 2 出力	
3	SP3		セットポイント 3 出力	
11	SP2, SP3 COM		SP2, SP3 COM（共通）	
2	RS485+	通信		<注4>
10	RS485−			



Experimental Setup

- Coding for reading from the Gauge
- Coding for analyzing the data from the Gauge

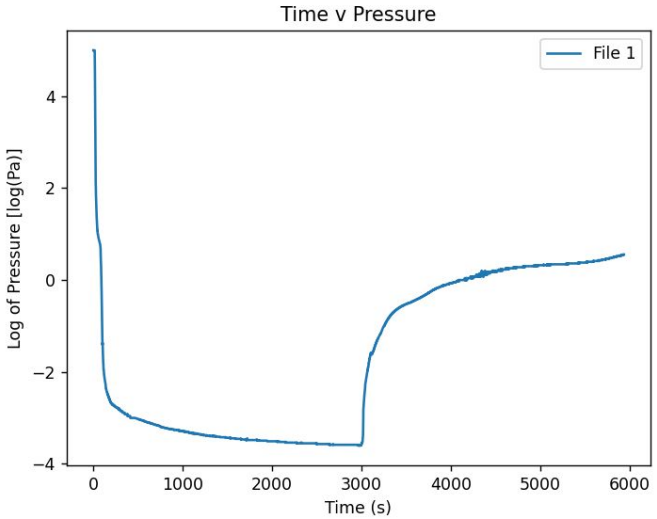
```python
# Serial Reading Thread
def read_serial():
    with open(f'{log_file_path}.txt', 'a') as logfile:
        print("Starting to read from RS485...")
        print("Time (s), Pressure (Pa)")
        logfile.write(f'#Date: {formatted_time}\n#Purpose: Measure Pressure vs Time
        while not stop_event.is_set():
            response = query_gauge(ser, GAUGE_ADDRESS, 'S1')
            if response:
                current_time = time.time()
                elapsed_time = round(current_time - start_time, 2)
                mantissa = float(f"{response[3]}.{response[4]}")
                exp_sign = 1 if response[5] == '1' else -1
                exp = int(response[6]) * exp_sign
                log_line = f"{elapsed_time},{mantissa}e{exp}"
                print(log_line)
                logfile.write(log_line + '\n')
                logfile.flush()
            else:
                current_time = time.time()
                elapsed_time = round(current_time - start_time, 2)
                print(f"{elapsed_time} seconds - No response from gauge.")

            woke_early = wake_event.wait(timeout=recording_speed[0])
            if woke_early:
                wake_event.clear()
```

```
#Date: 2025-07-17 13:07:32
#Purpose: Measure Pressure
#Columns: Time, Pressure
#Units: Seconds, Pascals
1.38,1.0e5
2.75,1.0e5
4.07,1.0e5
5.3,1.0e5
6.52,1.0e5
7.73,1.0e5
9.03,1.0e5
10.24,1.0e5
11.48,1.0e5
12.74,1.0e5
14.0,1.0e5
15.23,1.0e5
16.57,1.0e5
17.85,6.5e4
19.11,3.9e4
20.39,1.6e4
21.68,5.4e3
```

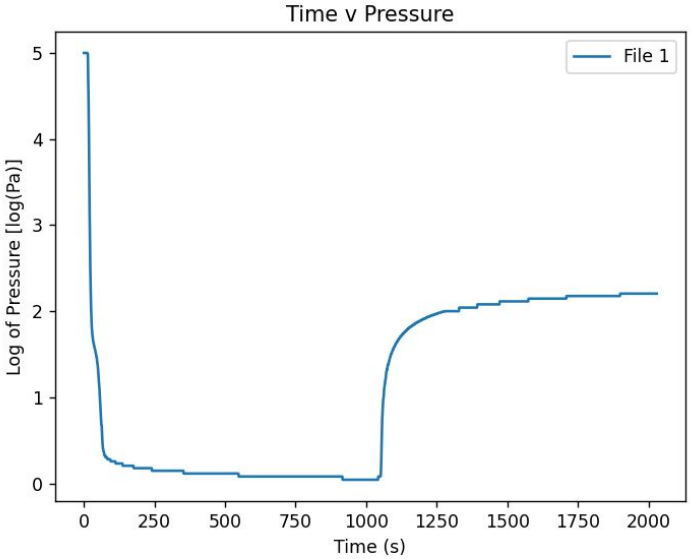
```python
# Turn parse strings into definitions for passing into curve_fit
function_map = {
    'exponential': Exponential,
    'polynomial': Polynomial,
    'gaussian': Gaussian,
    'linear': Linear
}
func = function_map.get(args.type.lower())

# If args.simple == False, performs fit. If True, only plots the data
if not args.simple:
    parameters, covariance = curve_fit(func, xdata[0], ydata[0], p0 = guess, maxfev=2000)
    fit_A = parameters[0]
    fit_B = parameters[1]
    if args.type != 'linear':
        fit_C = parameters[2]
    if args.type == 'gaussian':
        fit_D = parameters[3]
    if args.type == 'linear':
        equation = f'y = {fit_A:.3f}x + {fit_B:.3f}'
        fit_y = Linear(xdata[0], fit_A, fit_B)
        print(f'Fit parameters are A = {fit_A:.3f} and B = {fit_B:.3f}')
        print(f'Guess parameters are A = {guess[0]:.3f} and B = {guess[1]:.3f} respectively.')
    elif args.type == 'exponential':
```

Snippets:
Serial reading (above)
Data analysis (left)
Example output file (right)

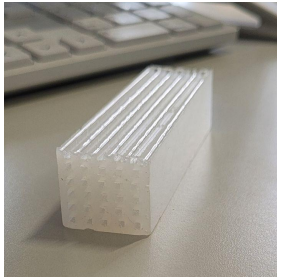
Test Measurements



Ultimate vacuum/base leak rate

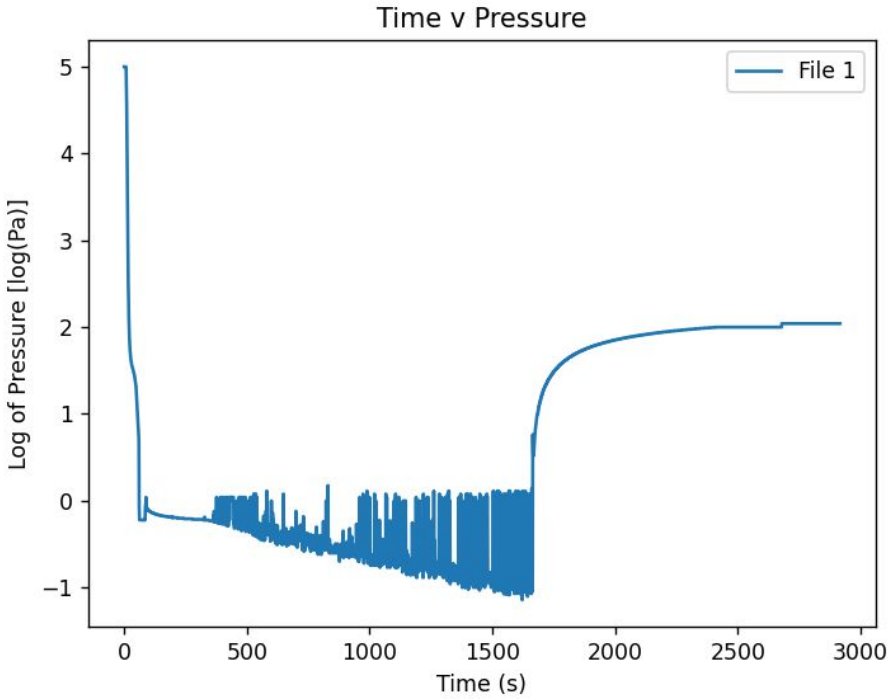


Silicone degas rate



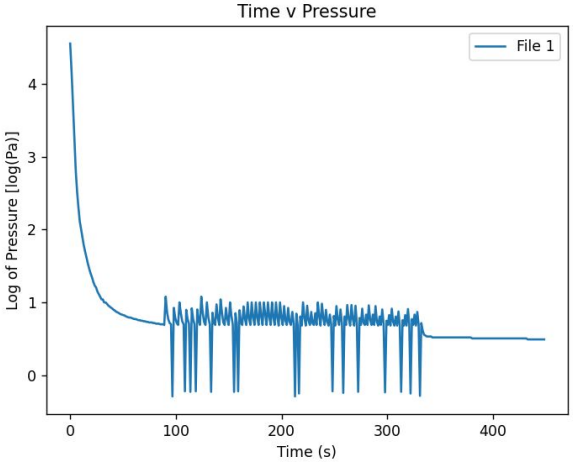
Test Measurements

- Silicone degas rate high; attempt to cure in vacuum
- Cured silicone in vacuum, then tested degas rate (right)
- Believe that curing the silicone in vacuum caused condensate to enter pump/gauge, causing errors

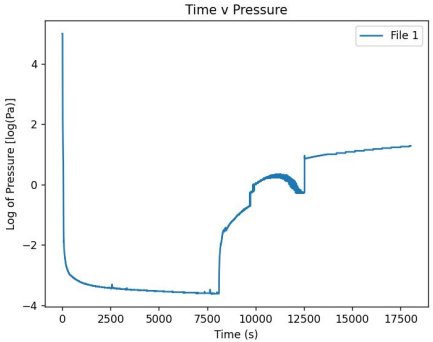


Currently

- Troubleshooting gauge errors
- Gauge has errors between 3-5 Pa and when below 1 Pa
- Purging vacuum
- Replacing vacuum system



One purge cycle; example of gauge misreading between 3-5 Pa



Measure of vacuum after replacing gauge, vacuum system

Thank you!
Any questions?