 Python Main Start
Initialize Controller

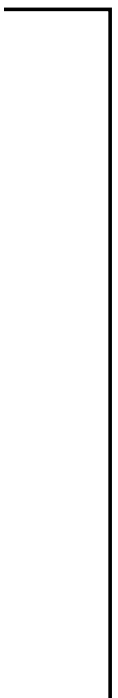
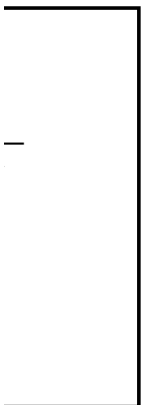
Initialization Phase

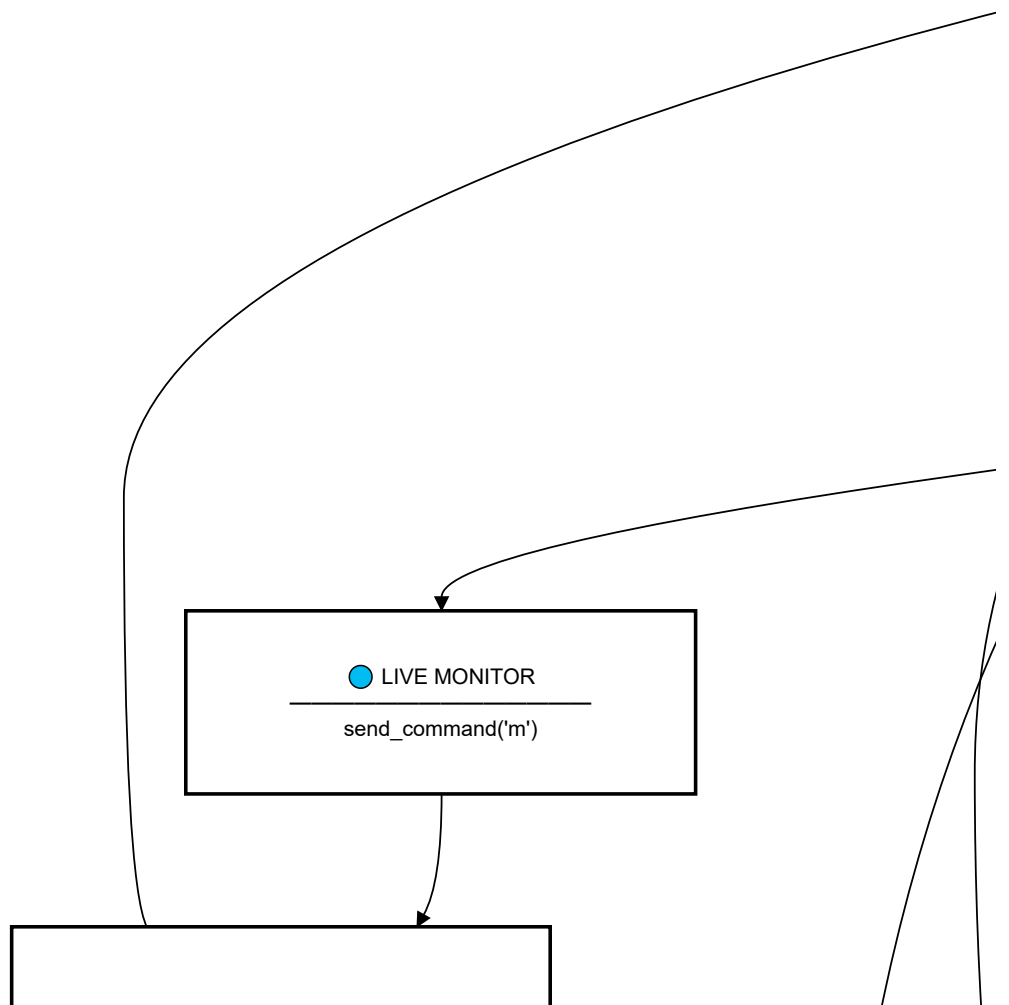
- Create SurfaceTensionController
 - Load test profiles 1-8
 - Detect serial port
 - Connect to Arduino
 - Create output directory
- Load calibration from EEPROM

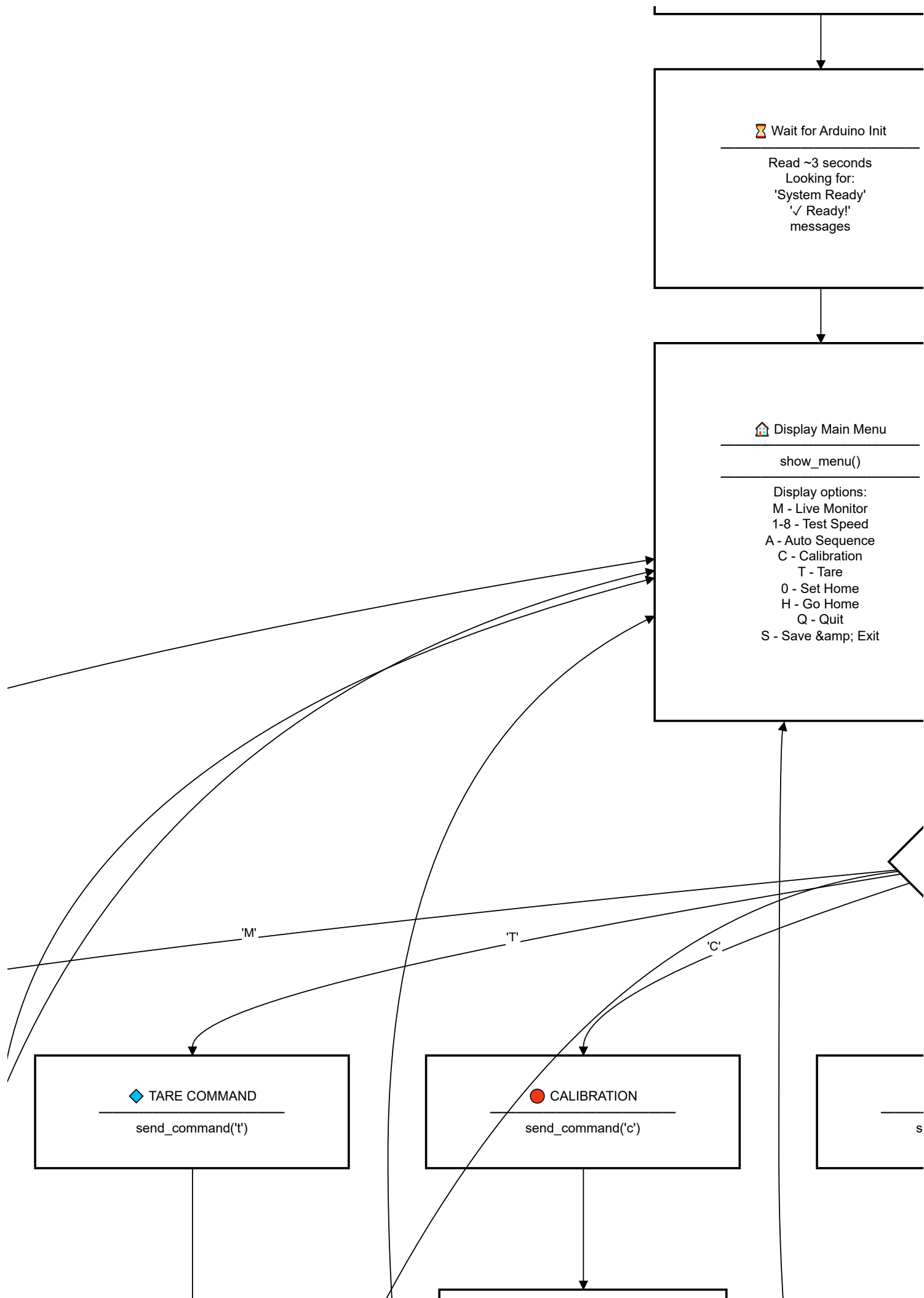
Serial Connection

connect_serial()

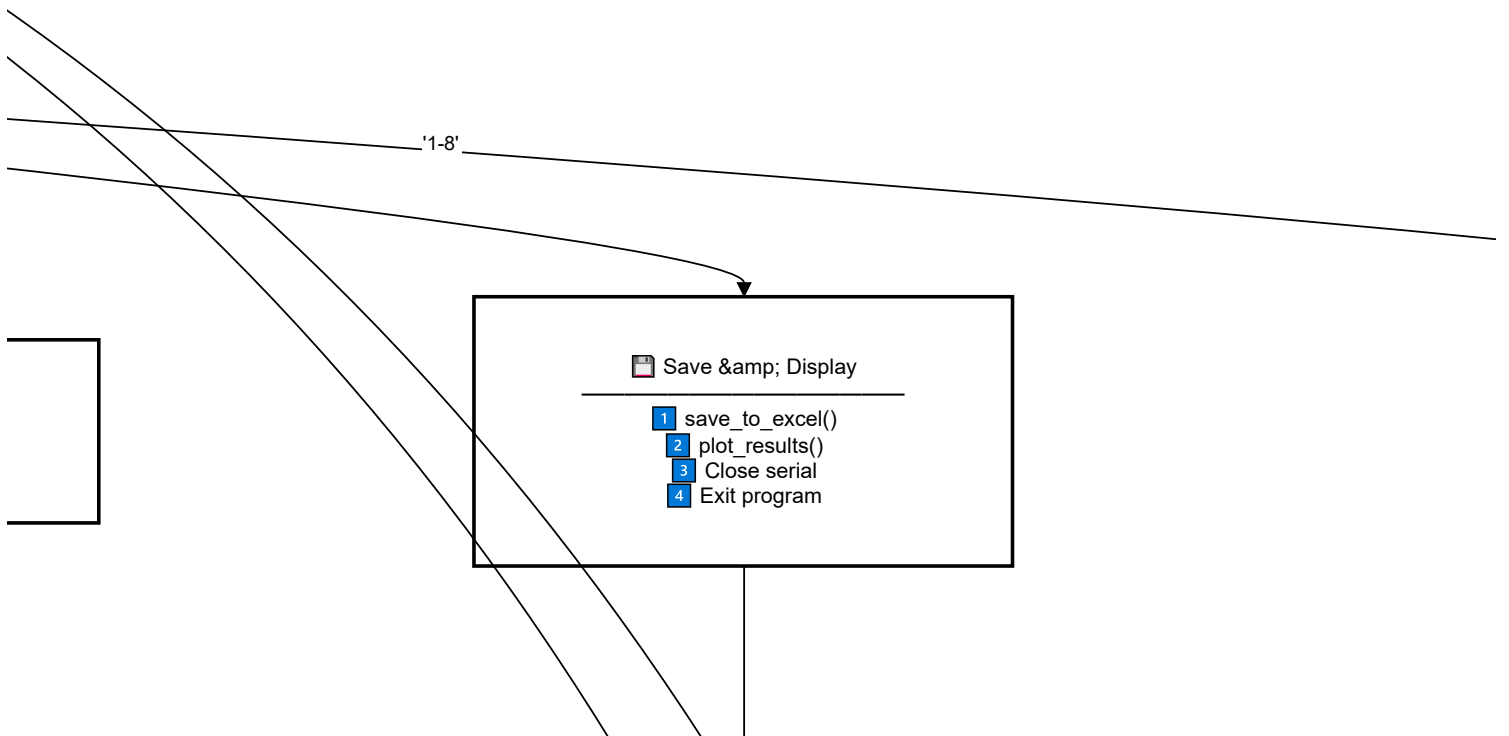
- 1 Try connect port
(auto-detect: COM5)
↓
- 2 Set baudrate: 115200
↓
- 3 Wait 2 seconds
(Arduino hardware reset)
↓
- 4 Check connection
if success: continue
else: list_ports()

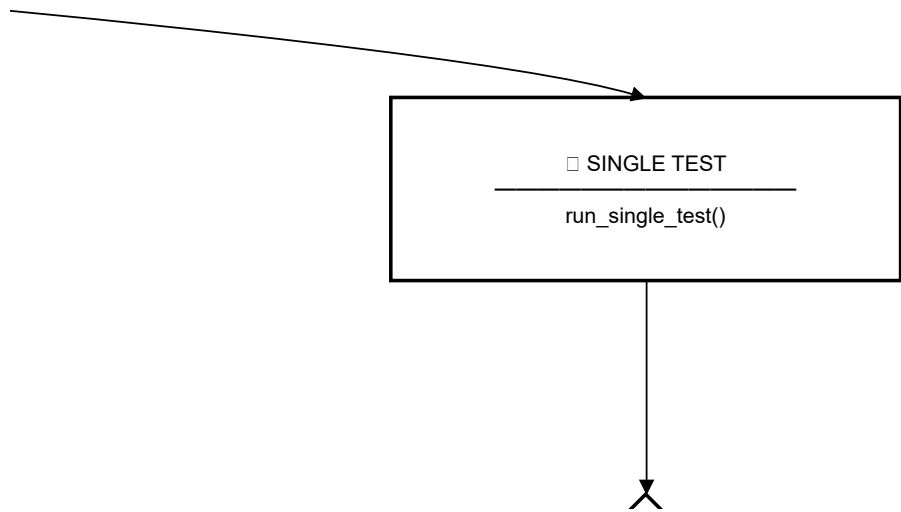












● Live Monitor Loop

```
Read from serial:  
  if 'Force:' in line  
    print(line, end='\n')  
← Overwrite each update  
  if Ctrl+C pressed:  
    send_command('q')  
    return to menu
```

Yes

Send 'T' to Arduino

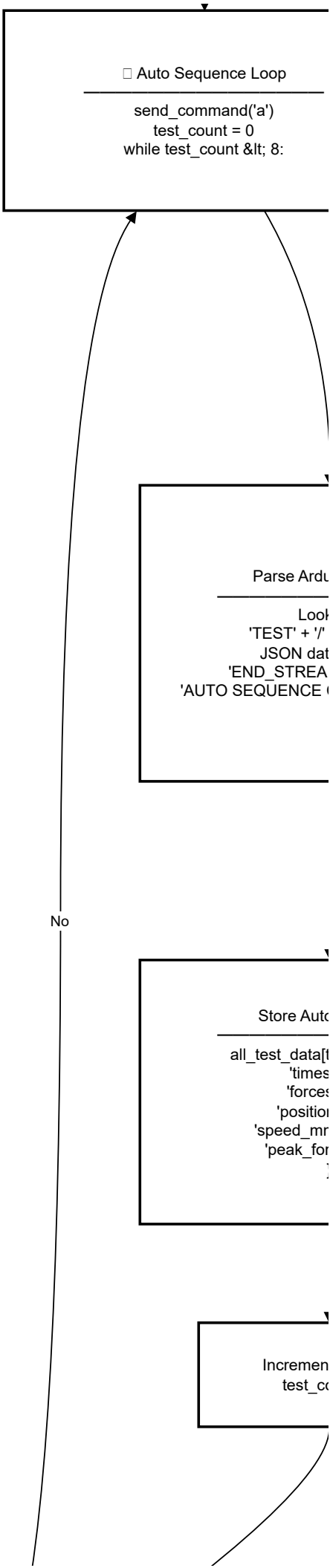
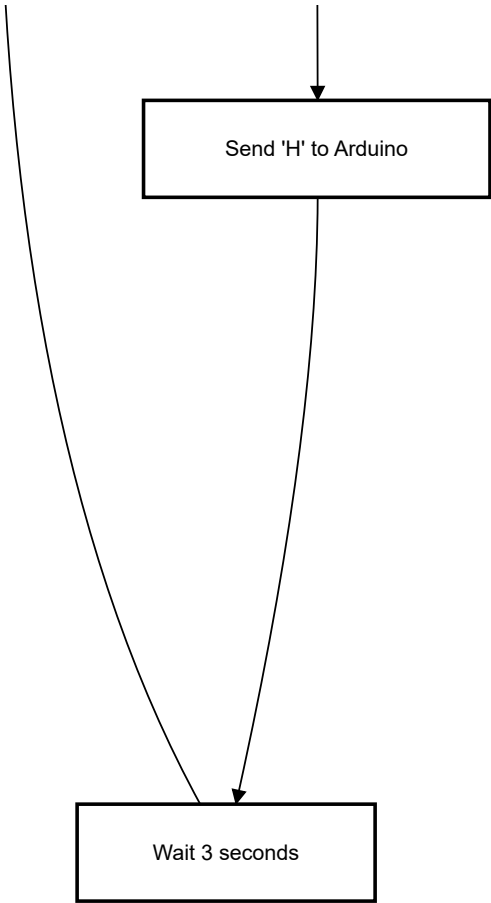
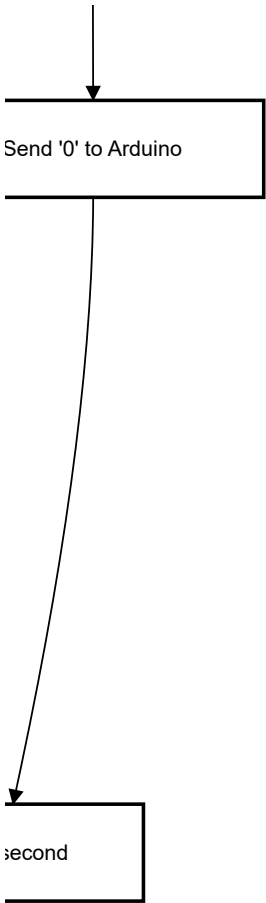
Send 'C' to Arduino
Pass-through mode


User follows on-screen
instructions from Arduino

Wait 2 seconds
Arduino handles tare


Monitor Arduino Output
Print all lines until:

Wait 1 s



 Save to Excel


save_to_excel()

 EXCEL PHYSICS 1

Generate Excel Report

For each test_name:
Create sheet[test_name]_Summary
↓
Write test info:
Speed (µm/s)
Number of runs
↓
Write statistics:
avg_peak = $\Sigma \text{peak} / N$
std_peak = $\sqrt{\Sigma (\text{peak} - \text{avg})^2 / N}$
rsd = $(\text{std} / \text{avg}) \times 100\%$
min, max
↓
For each run:
Create sheet[test_name]_R#
Write detailed data:
Time (s)
Force (N) ← raw values
Position (mm)
↓
Format with borders & colors

Save: surface_tension_results_
YYYYMMDD_HHMMSS.xlsx

 Plot Results

plot_results()

uino Output

k for:
→ new test
ta → store
.M' → save run
COMPLETE' → done

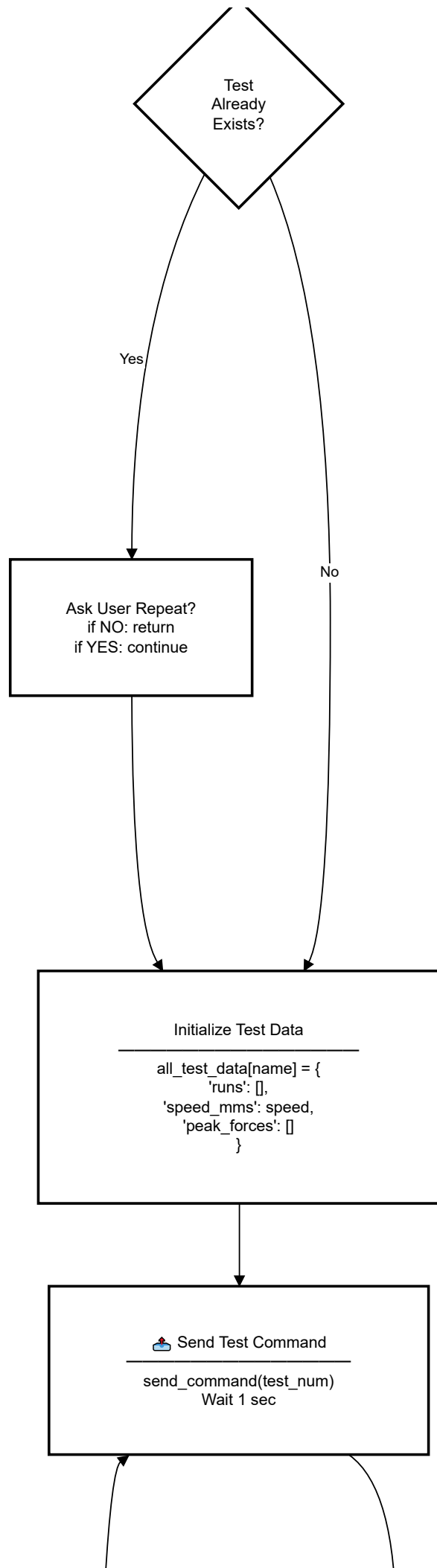
o Test Run

test_name] = {
s': [...],
s': [...],
ns': [...],
ms': speed,
rce': peak
}

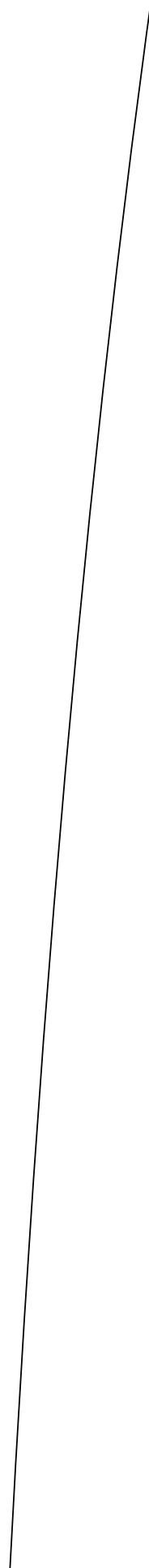
it Counter
ount++

Quit

Next Test

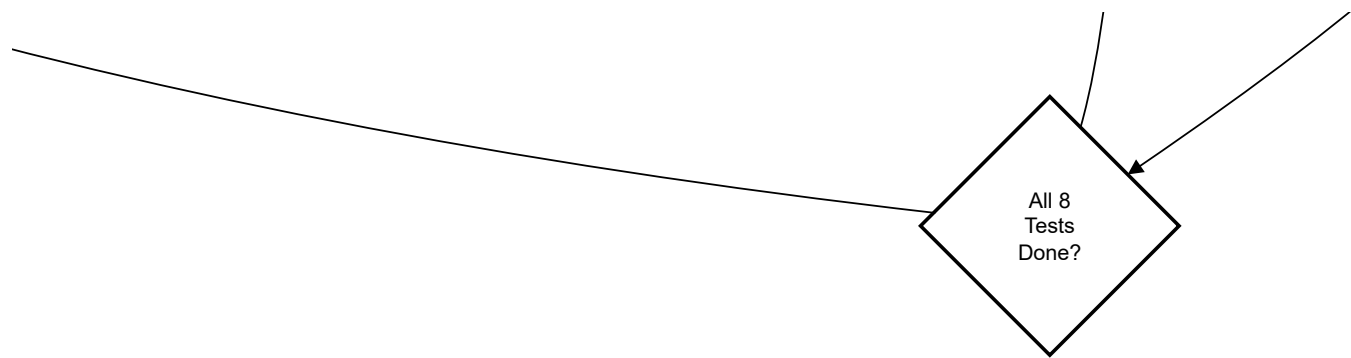


'Q'



/

/



PLOTTING PHYSICS 1

Find Contact Point

```
find_contact_point():  
for i in range(len(forces)):  
    if forces[i] < 0 AND  
       forces[i+1] ≥ 0:  
        ← Force crosses 0  
        contact_idx = i + 1  
    return contact_pos
```

Physics meaning:
Ring enters liquid
Buoyancy creates
negative force

PLOTTING PHYSICS 2

Find Peak Force

```
find_peak_force():  
    peak_idx = argmax(forces)  
    peak_force = forces[peak_idx]  
    return peak_force, peak_idx
```

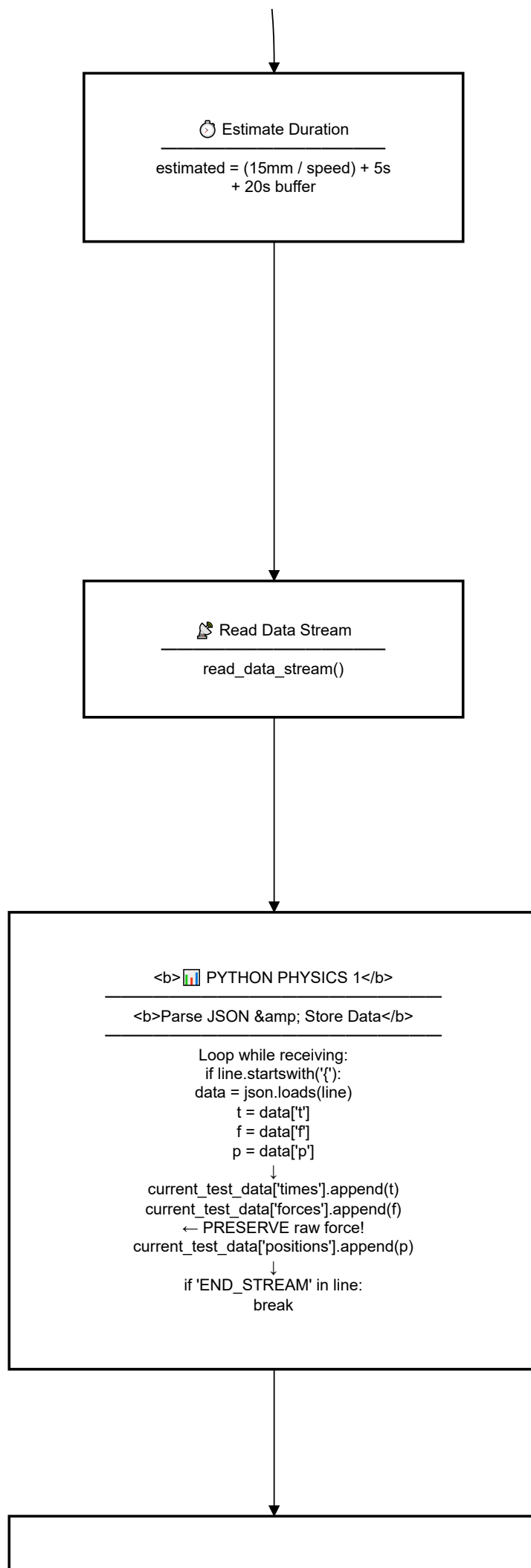
Physics meaning:
Maximum pull force
= Surface tension effect

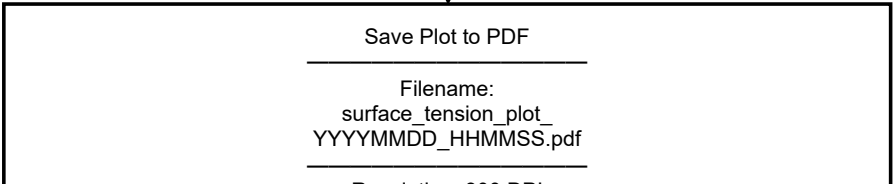
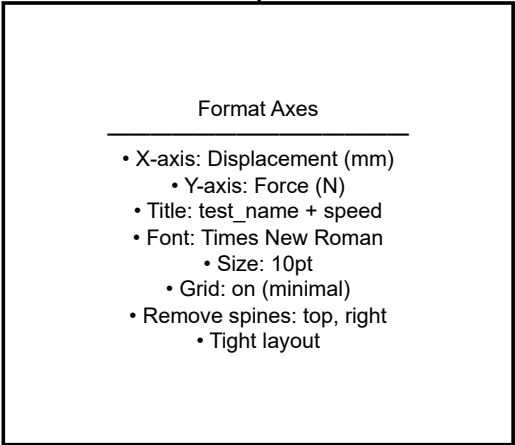
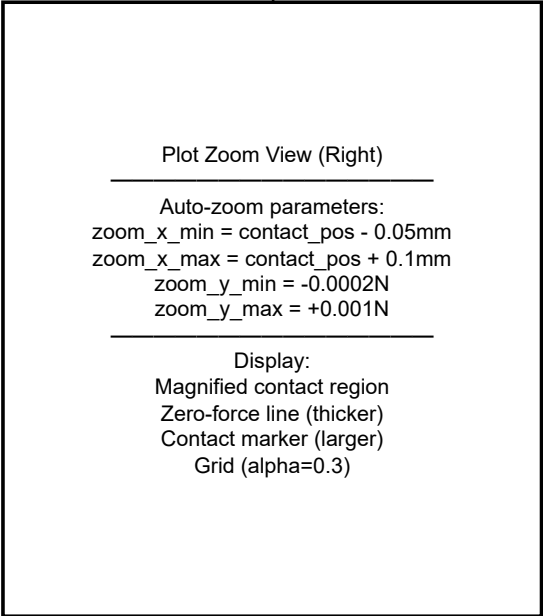
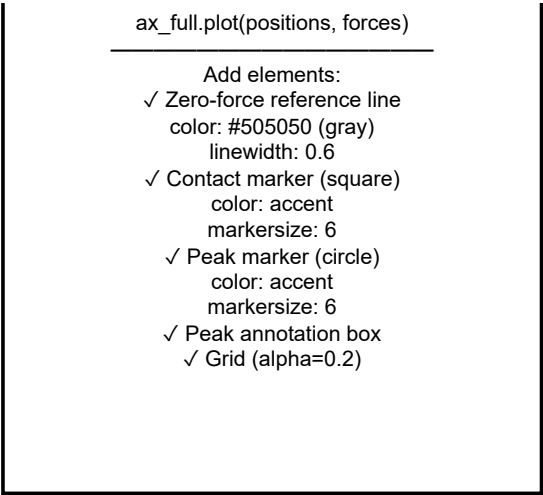
Plot Layout (2 columns per test)

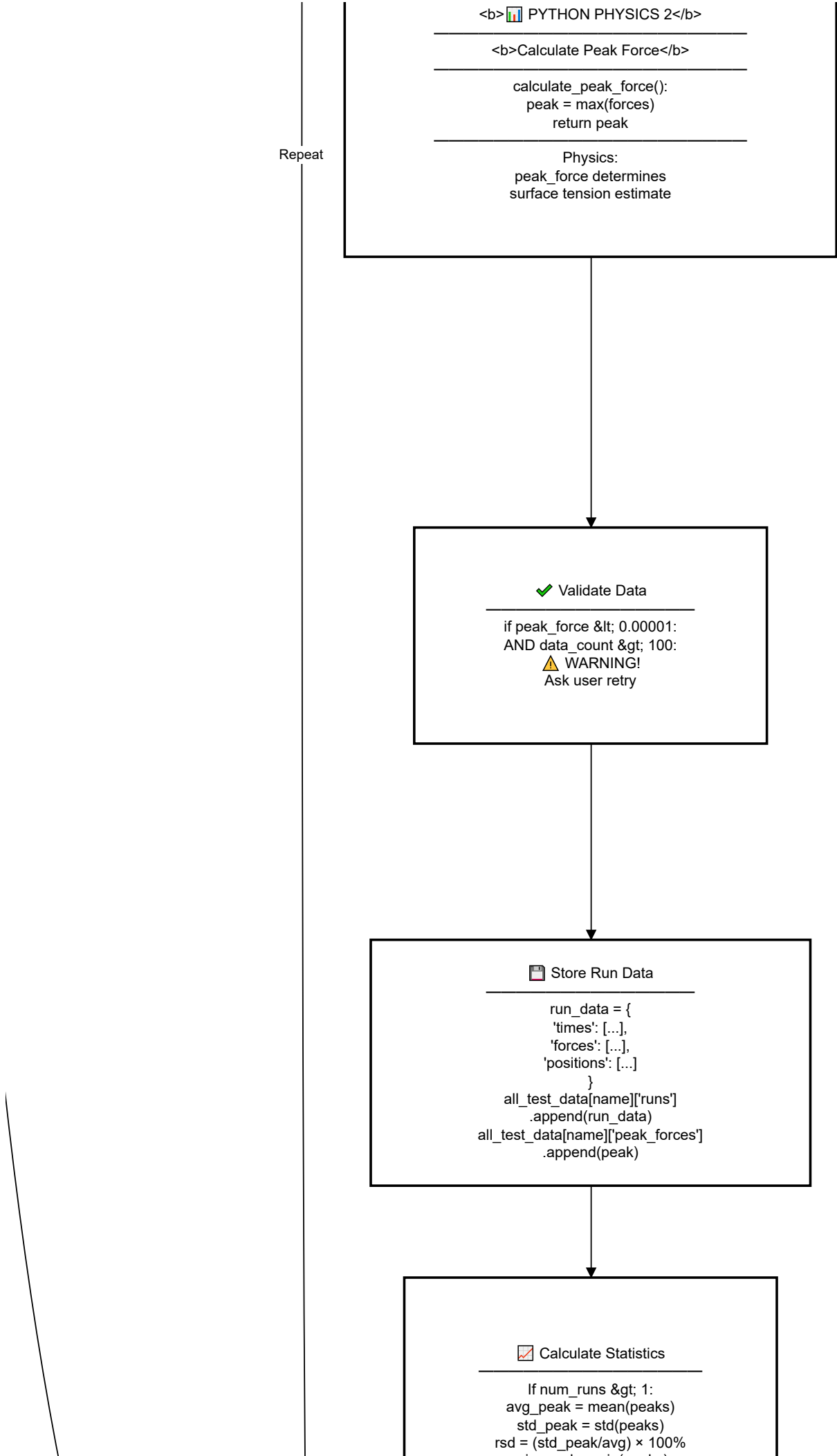
Left: Full curve view
Right: Zoomed contact view

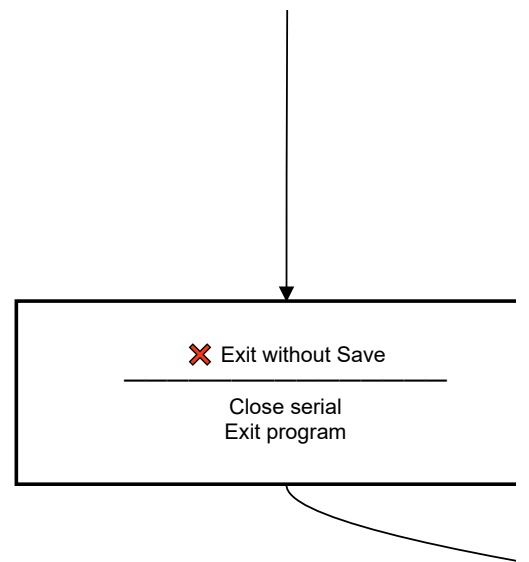
Sort by speed HIGH→LOW

Plot Full View (Left)

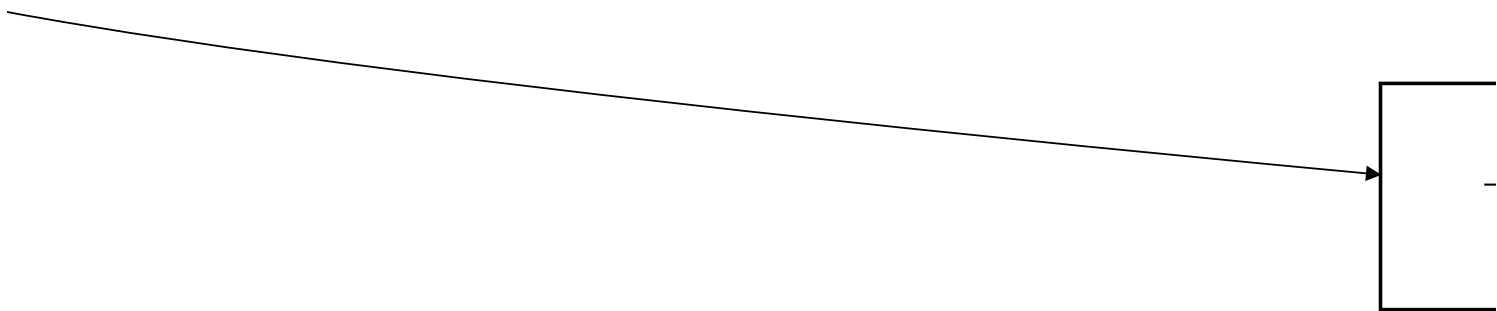






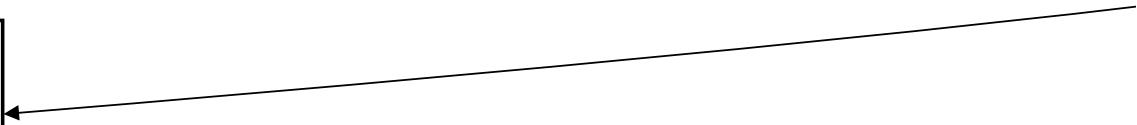


]



✔ Program Complete

Close serial connection
Exit




Resolution: 300 DPI
Format: PDF
Path: output_dir

Display on Screen
plt.show()

```
min_peak = min(peaks)
max_peak = max(peaks)
```



 Display Results

Print:
Peak Force (N)
Data Points
Statistics (if multiple)

