# Automated NanoVNA Sweep Data Capture Guide (Multi-Sweep Version)

This guide details the prerequisites, setup, and execution steps for using the nanovna\_capture.py script to automatically connect to your NanoVNA, perform a **sequence of calibrated frequency sweeps**, and save all the resulting S-parameter data into a single, labeled CSV file.

## Prerequisites

Before running the script, ensure you have the following installed and configured on your laptop:

1. **Python 3.x:** Installed and accessible from your command line/terminal.
2. **NanoVNA Connection:** The NanoVNA must be connected to your laptop via a USB data cable and powered on.
3. **Required Python Libraries:** The pynanovna, numpy, and standard libraries (time, csv) are required.  
   **Action:** Install the necessary libraries using pip:  
   pip install pynanovna numpy
4. **Calibration (CRITICAL):** For the data to be useful for analysis or machine learning, you **must** perform a full SOLT (Short, Open, Load, Thru) calibration on your NanoVNA for the exact frequency range defined in the script (default: 100 MHz to 1 GHz) *before* running the script. Uncalibrated data is highly unreliable.

## Script Configuration and Usage

The automation script is designed to be easily configurable for bulk data collection.

### Step 1: Review and Configure the Script

Open your nanovna\_capture.py file and adjust the following parameters under the SCRIPT CONFIGURATION section as needed for your specific measurement task:

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Default Value** |
| NUM\_SWEEPS | **The total number of samples/sweeps to capture.** | 10 |
| START\_FREQ | The beginning frequency of the sweep (in Hz). | 100e6 (100 MHz) |
| STOP\_FREQ | The end frequency of the sweep (in Hz). | 1000e6 (1 GHz) |
| POINTS | The number of measurement steps in the sweep. | 101 |

### Step 2: Save the Script (Multi-Sweep Enabled)

Ensure the following code is saved as nanovna\_capture.py in your working directory. This script contains a loop and will append all sweep data to a single file.

import pynanovna  
import numpy as np  
import time  
import csv  
  
# ====================================================================  
# SCRIPT CONFIGURATION  
# Adjust these values to set the frequency range, resolution, and sweep count.  
# ====================================================================  
NUM\_SWEEPS = 10 # Total number of sweeps/samples to collect.  
START\_FREQ = 100e6 # Start Frequency in Hz (e.g., 100 MHz)  
STOP\_FREQ = 1000e6 # Stop Frequency in Hz (e.g., 1 GHz)  
POINTS = 101 # Number of data points per sweep (resolution)  
  
# Output file name includes a unique timestamp for the entire batch  
OUTPUT\_FILE = f"nanovna\_batch\_sweep\_{time.strftime('%Y%m%d\_%H%M%S')}.csv"  
  
# ====================================================================  
# HELPER FUNCTION  
# Converts complex S-parameter (Real + Imaginary) into Magnitude (dB) and Phase (degrees).  
# ====================================================================  
def s\_to\_mag\_phase(s\_complex):  
 # Magnitude (dB) = 20 \* log10(|S|)  
 mag\_db = 20 \* np.log10(np.abs(s\_complex))  
 # Phase (degrees)  
 phase\_deg = np.angle(s\_complex, deg=True)  
 return mag\_db, phase\_deg  
  
# ====================================================================  
# MAIN AUTOMATION LOGIC  
# Connects, sets sweep, loops through N measurements, and saves the file.  
# ====================================================================  
print("Starting NanoVNA Multi-Sweep Automation...")  
vna = None # Initialize vna variable  
  
try:  
 # 1. Initialize and Connect to the NanoVNA  
 vna = pynanovna.VNA()  
 print("Connection successful.")  
  
 # 2. Set Sweep Parameters once for the entire batch  
 print(f"Setting sweep: {START\_FREQ/1e6:.1f} MHz to {STOP\_FREQ/1e6:.1f} MHz, {POINTS} points.")  
 vna.set\_sweep(START\_FREQ, STOP\_FREQ, POINTS)  
 time.sleep(0.5)  
  
 # Prepare CSV file: Open it and write the header row only once  
 with open(OUTPUT\_FILE, 'w', newline='') as f:  
 writer = csv.writer(f)  
   
 # NOTE: Added 'Sample\_ID' column for ML labeling  
 header = ["Sample\_ID", "Frequency (Hz)", "S11 Mag (dB)", "S11 Phase (deg)",   
 "S21 Mag (dB)", "S21 Phase (deg)",  
 "S11 Real", "S11 Imag", "S21 Real", "S21 Imag"]  
 writer.writerow(header)  
  
 # 3. Loop through the desired number of sweeps  
 for sweep\_id in range(1, NUM\_SWEEPS + 1):  
 print(f"\n--- Ready for Sweep {sweep\_id} of {NUM\_SWEEPS} ---")  
   
 # --- PAUSE FOR MANUAL INTERACTION ---  
 # This is CRITICAL: Wait for the user to change the physical sample/DUT.  
 input(f"ACTION REQUIRED: Attach Sample ID {sweep\_id} to ports 1 and 2. PRESS ENTER to start the sweep.")  
   
 # 4. Perform the Sweep and Capture Data  
 print(f"Sweeping Sample ID {sweep\_id}...")  
 s11\_complex, s21\_complex, frequencies = vna.sweep()  
  
 # 5. Append Data to the CSV file  
 with open(OUTPUT\_FILE, 'a', newline='') as f: # Use 'a' for append mode  
 writer = csv.writer(f)  
   
 for i in range(POINTS):  
 f\_hz = frequencies[i]  
   
 # Convert complex S-parameters  
 s11\_mag\_db, s11\_phase\_deg = s\_to\_mag\_phase(s11\_complex[i])  
 s21\_mag\_db, s21\_phase\_deg = s\_to\_mag\_phase(s21\_complex[i])  
   
 # Construct the row, starting with the Sample\_ID  
 row = [sweep\_id, f\_hz, s11\_mag\_db, s11\_phase\_deg, s21\_mag\_db, s21\_phase\_deg,  
 s11\_complex[i].real, s11\_complex[i].imag,  
 s21\_complex[i].real, s21\_complex[i].imag]  
 writer.writerow(row)  
   
 print(f"✅ Data for Sample ID {sweep\_id} appended successfully.")  
   
 print(f"\n✨ MULTI-SWEEP BATCH COMPLETE! All data saved to {OUTPUT\_FILE}")  
  
except Exception as e:  
 # Catch any connection or communication errors  
 print(f"❌ An error occurred. Ensure the NanoVNA is connected, powered on, and the pynanovna library is installed.")  
 print(f"Error details: {e}")  
  
finally:  
 # Clean up the connection if it was established  
 if vna and vna.connected:  
 print("Disconnected from NanoVNA.")

### Step 3: Execute the Script (Interactive Mode)

1. Open your terminal and navigate to the directory where you saved nanovna\_capture.py.
2. Run the script:  
   python nanovna\_capture.py
3. **Interactive Prompt:** The script will pause after the header is written. It will display a prompt asking you to attach the current sample and **wait for you to press ENTER**. This pause is essential for labeling your data correctly.
4. **Repeat:** Repeat the process of changing the physical sample and pressing ENTER until all NUM\_SWEEPS are complete.

### Expected Output

If successful, a single, large CSV file (e.g., nanovna\_batch\_sweep\_...csv) will be created, containing data from all your sweeps. Every row will be identified by the **Sample\_ID** column, which is perfect for training your ML model!