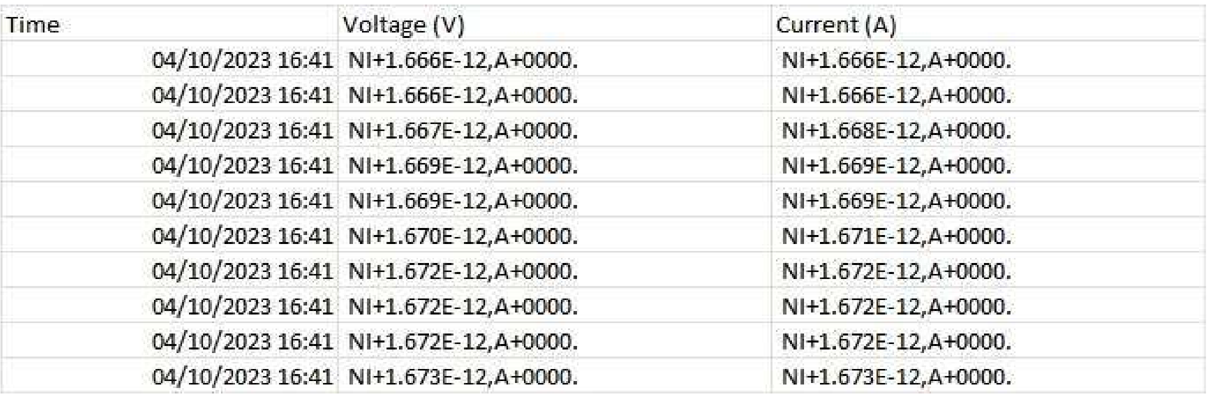
Data Collection code:

```
import pyvisa
import time
import csv
import traceback

# Replace 'GPIB0::10::INSTR' with your HP 4140B's GPIB address
instrument_address = 'GPIB0::17::17::INSTR'

# Initialize VISA resource manager
rm = pyvisa.ResourceManager()
rm.timeout = 5000  # Set timeout to 100 seconds (adjust as needed)

try:
    # Open a connection to the instrument
    instrument = rm.open_resource(instrument_address, write_termination='\n', read_termination='\n')

    # Configure instrument settings (customize these for your measurements)
    instrument.write("VOLT:RANG 02")  # Set voltage range to 10V
    instrument.write("CURR:RANG 1E-6")  # Set current range to 1µA

    # Create a CSV file to save the data
    filename = "hp4140b_data10.csv"
    with open(filename, mode='w', newline='') as csv_file:
        fieldnames = ['Time', 'Measurement']
        writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
        writer.writeheader()

        for _ in range(500):  # Perform 10 measurements
```

```
            # Measure voltage and current
            instrument.write("MEAS?")
            measurement_str = instrument.read()

            # Get the current time
            timestamp = time.strftime("%Y-%m-%d %H:%M:%S")

            # Write the measurement and time to the CSV file
            writer.writerow({'Time': timestamp, 'Measurement': measurement_str})

            # Wait for 0.1 seconds before taking the next measurement
            time.sleep(0.1)

except pyvisa.errors.VisaIOError as e:
    print(f"Measurement failed with error: {e}")
    traceback.print_exc()

finally:
    # Close the instrument connection
    instrument.close()
```



Time	Voltage (V)	Current (A)
04/10/2023 16:41	NI+1.666E-12,A+0000.	NI+1.666E-12,A+0000.
04/10/2023 16:41	NI+1.666E-12,A+0000.	NI+1.666E-12,A+0000.
04/10/2023 16:41	NI+1.667E-12,A+0000.	NI+1.668E-12,A+0000.
04/10/2023 16:41	NI+1.669E-12,A+0000.	NI+1.669E-12,A+0000.
04/10/2023 16:41	NI+1.669E-12,A+0000.	NI+1.669E-12,A+0000.
04/10/2023 16:41	NI+1.670E-12,A+0000.	NI+1.671E-12,A+0000.
04/10/2023 16:41	NI+1.672E-12,A+0000.	NI+1.672E-12,A+0000.
04/10/2023 16:41	NI+1.672E-12,A+0000.	NI+1.672E-12,A+0000.
04/10/2023 16:41	NI+1.672E-12,A+0000.	NI+1.672E-12,A+0000.
04/10/2023 16:41	NI+1.673E-12,A+0000.	NI+1.673E-12,A+0000.

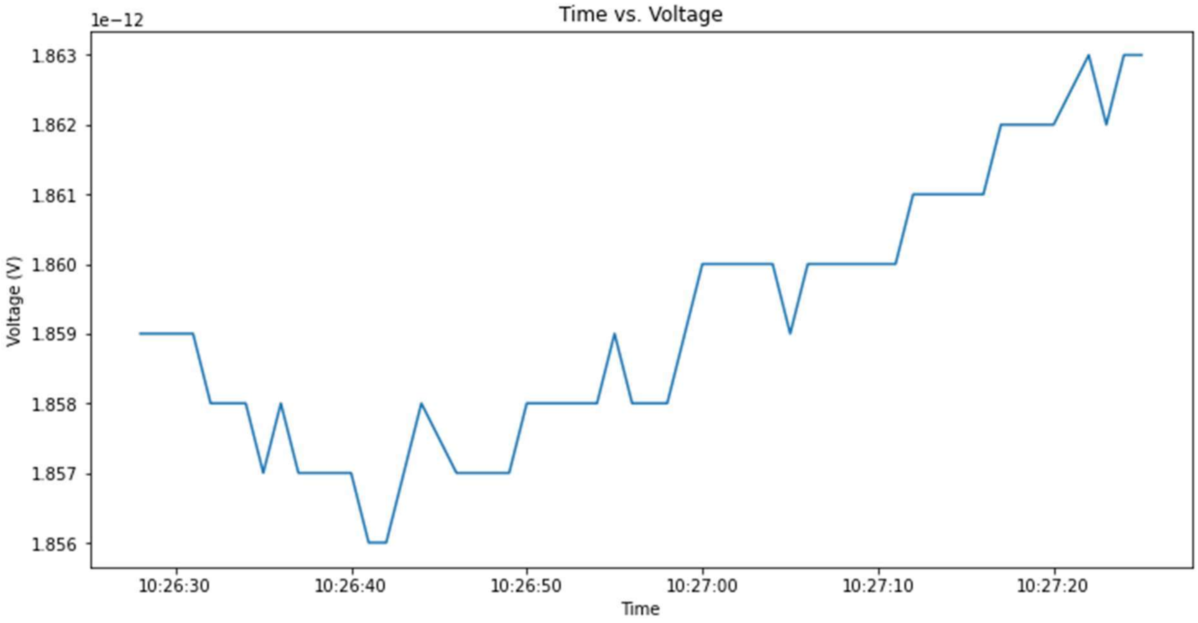
Data Converstion code:

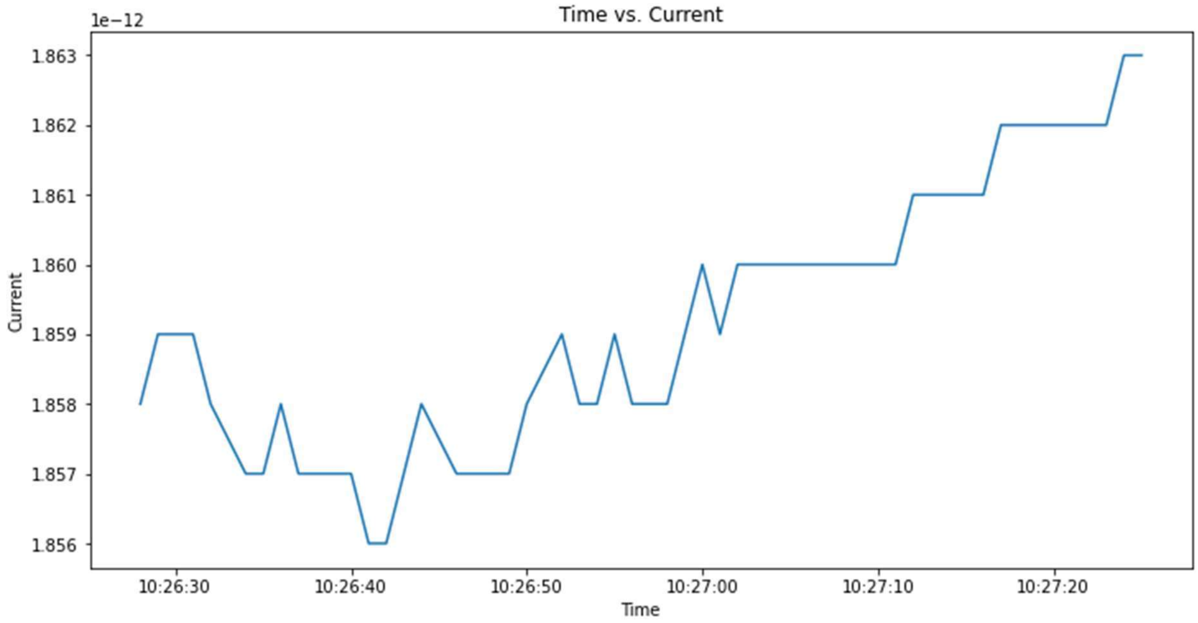
```
import csv


def extract_current_voltage(measurement_str):
    """Extracts the current and voltage values from a measurement string.


    Args:
        measurement_str: A string containing the measurement data.


    Returns:
        A tuple of two strings, containing the current and voltage values as strings.
    """
    # Split the measurement string using a comma as the separator.
    parts = measurement_str.split(",")


    if len(parts) == 2:
        # Extract the first part as current (skipping the first two characters)
        current_str = parts[0][3:]


        # Extract the second part as voltage (remove 'A' and leading/trailing whitespace)
        voltage_str = parts[1].replace("A", "").strip()


        return current_str, voltage_str
    else:
        raise ValueError("Invalid measurement string format")


# Create lists to store the current and voltage values as strings
current_values = []
voltage_values = []
```

```
# Open the CSV file containing the measurement data.
with open("hp4140b_data10.csv", mode='r', newline='') as csv_file:
    # Create a CSV reader object.
    reader = csv.DictReader(csv_file)

    # Iterate over the rows in the CSV file.
    for row in reader:
        # Extract the current and voltage values from the measurement string.
        current, voltage = extract_current_voltage(row['Measurement'])

        # Append the values to their respective lists
        current_values.append(current)
        voltage_values.append(voltage)

# Now you have two lists: current_values and voltage_values containing the current and voltage
values as strings.

# Write the current and voltage values to a new CSV file.
output_filename = "current_voltage_values1.csv"
with open(output_filename, mode='w', newline='') as output_csv_file:
    fieldnames = ['Current', 'Voltage']
    writer = csv.DictWriter(output_csv_file, fieldnames=fieldnames)
    writer.writeheader()

    # Write the values to the CSV file
    for current, voltage in zip(current_values, voltage_values):
        writer.writerow({'Current': current, 'Voltage': voltage})

print(f"Current and voltage values saved to {output_filename}")
```

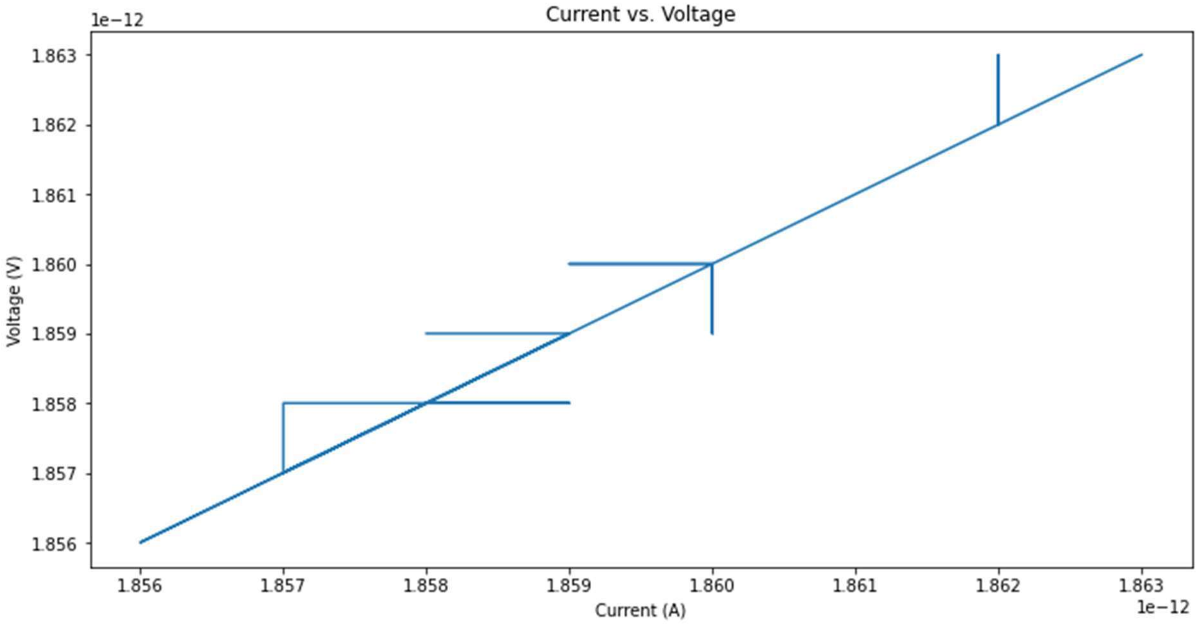
Graphical representation code:

```
import csv
import matplotlib.pyplot as plt

def read_csv_data(filename):
  """Reads data from a CSV file.

  Args:
    filename: The name of the CSV file.

  Returns:
    A list of tuples, where each tuple contains a current and voltage value.
  """

  data = []
  with open(filename, mode='r', newline='') as csv_file:
    reader = csv.DictReader(csv_file)
    for row in reader:
      current = float(row['Current'])
      voltage = float(row['Voltage'])
      data.append((current, voltage))
  return data

csv_filename = "current_voltage_values1.csv"
data = read_csv_data(csv_filename)


plt.plot([x for x, _ in data], [y for _, y in data])

# Set the axis labels and title.
plt.xlabel("Current (A)")
plt.ylabel("Voltage (V)")
plt.title("Current vs. Voltage")

# Show the graph.
plt.show()



plt.plot([y for _, y in data], [x for x, _ in data])

# Set the axis labels and title.
plt.xlabel("Voltage (V)")
plt.ylabel("Current (A)")
plt.title("Voltage vs. Current")

# Show the graph.
plt.show()
```

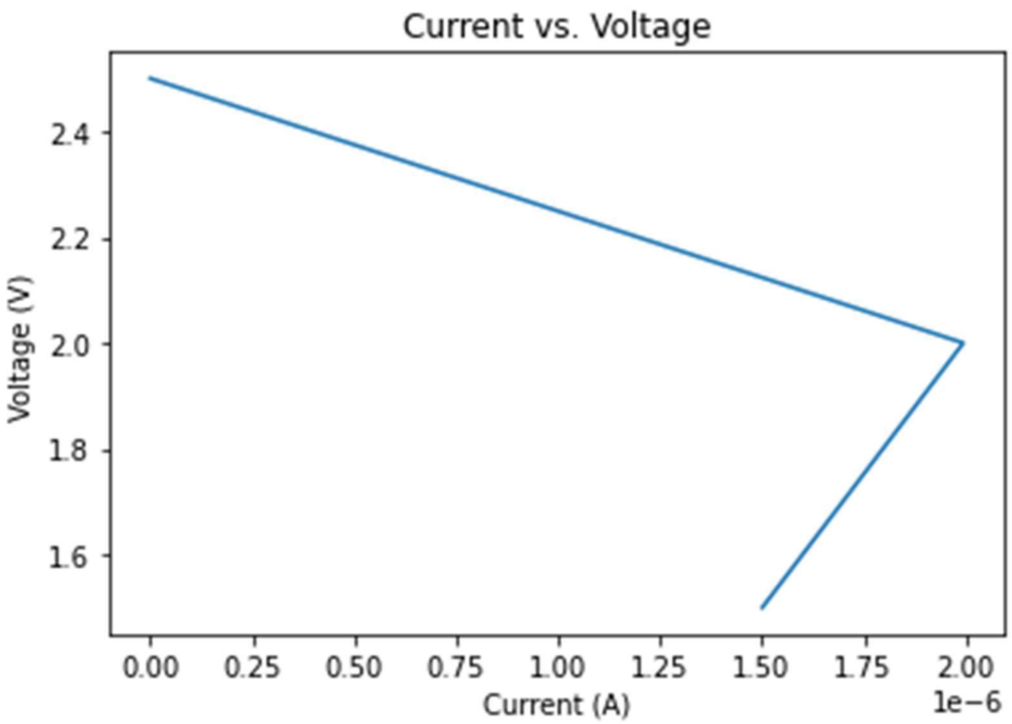




Time vs. Current



Current vs. Voltage


Current vs. Voltage
Voltage (V)
2.4
2.2
2.0
1.8
1.6
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
1e−6
Current (A)