**Dendro-Pi-Logger**

Program to take log dendrometer data

- --

\*\*This assumes you have already finished the *normal* raspberry pi setup\*\* (To take pictures and upload to dropbox)

**Getting started**

**Copy scripts to Raspberry Pi**

**run the following command (without braces)**

*> scp -r {Path to File(s)} {Destination for copy}*

*Example – Adding Dendrometer logging script -(all one line):*

*> scp – r C:\Users\alanj\Documents\School\MadLab\RaspberryPi\moyenne\_glisante\Alans\_Scripts\*dendro\_monitor\_scheduled\_4.py *madlab@DorvalTest.local:~/dendro-pi-main/dendro\_logger*

*Example – Updated Dropbox Uploader file -(all one line):*

*> scp -r C:\Users\alanj\Documents\School\MadLab\RaspberryPi\moyenne\_glisante\Alans\_Scripts\upload-to-dropbox.sh madlab@DorvalTest.local:~/dendro-pi-main/Dropbox-Uploader/*

*Need to save a copy of upload-to-dropbox.sh with dendrometer logging and picture taking.*

**Configuring Raspberry Pi**

**1. Connect to same network on computer that the raspberry pi is configured to connect to**

**2. If on PC, open the command prompt / Windows PowerShell,**

**SSH into raspberry pi by running the following command (without braces)**

*> ssh username@{hostname}.local*

*EXAMPLE: ssh madlab@DorvalTest.local*

When prompted for password: input Password from OS Customization

*EXAMPLE: Aspen2022*

**3. Run the following command to configure the Raspberry Pi**

*> sudo apt-get update*

**4. Install pip (if not already installed):**

*> sudo apt-get install python3-pip*

**5. Create a Virtual Environment**

> cd dendro-pi-main ?

*> python3 -m venv myenv*

*Activate the Virtual Environment:*

*> source myenv/bin/activate*

**6. Install necessary libraries:**

*> sudo apt-get install libopenblas-dev*

*> pip install adafruit-circuitpython-busdevice adafruit-circuitpython-ads1x15 numpy*

*sudo apt install watchdog*

*sudo systemctl enable watchdog*

*sudo systemctl start watchdog*

*Configure the watchdog by editing /etc/watchdog.conf:*

*watchdog-device = /dev/watchdog*

*watchdog-timeout = 3600 # Reboot after 3600 seconds of inactivity*

*Crontab example:  
  
\*/5 \* \* \* \* /home/madlab/venv/bin/python3 /home/madlab/dendro-pi-main/dendro\_logger/dendro\_monitor\_scheduled\_4.py*

*/home/madlab/dendro-pi-main/venv/bin/python3 /home/madlab/dendro-pi-main/dendro-logger/dendro\_monito*

*r\_scheduled\_4.py*

*Need to update Crontab example / backup file to include both dendrometer logging and picture taking*

***Install RPi.GPIO inside your virtual environment***

*> source /home/madlab/dendro-pi-main/venv/bin/activate*

*> pip install RPi.GPIO****Enable I2C in Raspberry Pi Config****> sudo raspi-config  
Interface Options → I2C → Enable*

*Need to change directories in upload script*

*(from sh to bash)*

***Here's how to plot your dendrometer data on Windows using Python (with pandas/matplotlib):***

***Step 1: Download Data from Dropbox***

1. *Sync your Dropbox folder to your Windows PC, or manually download the .txt files.*

***Step 2: Combine Files (if needed)***

*If you have multiple daily files (e.g., micron\_values\_channel\_0\_2025-02-22.txt), combine them into one file:*

*# In PowerShell, navigate to the data folder and run:*

*> cat micron\_values\_channel\_0\_\*.txt > merged\_channel\_0.csv*

***Step 3: Plot with Python***

1. ***Install Python****: Download from* [*python.org*](https://www.python.org/)*.*
2. ***Install Required Libraries****:*

pip install pandas matplotlib

***Step 1: Prepare Data***

1. *Open a new Excel workbook.*
2. *Import your .txt file:*
   * ***Data Tab*** *→* ***Get Data*** *→* ***From Text/CSV****.*
   * *Select your file and ensure columns are split correctly.*

***Step 2: Format Timestamps***

1. *Highlight the timestamp column.*
2. ***Data Tab*** *→* ***Text to Columns*** *→* ***Delimited*** *→ Choose comma (,).*
3. *Convert text to Excel datetime format.*

***Step 3: Create a Line Chart***

1. *Select both timestamp and microns columns.*
2. ***Insert Tab*** *→* ***Line Chart****.*
3. *Customize axes/titles:*
   * *Right-click chart →* ***Add Trendline*** *for smoothing.*