Thrust Stand Software Manual

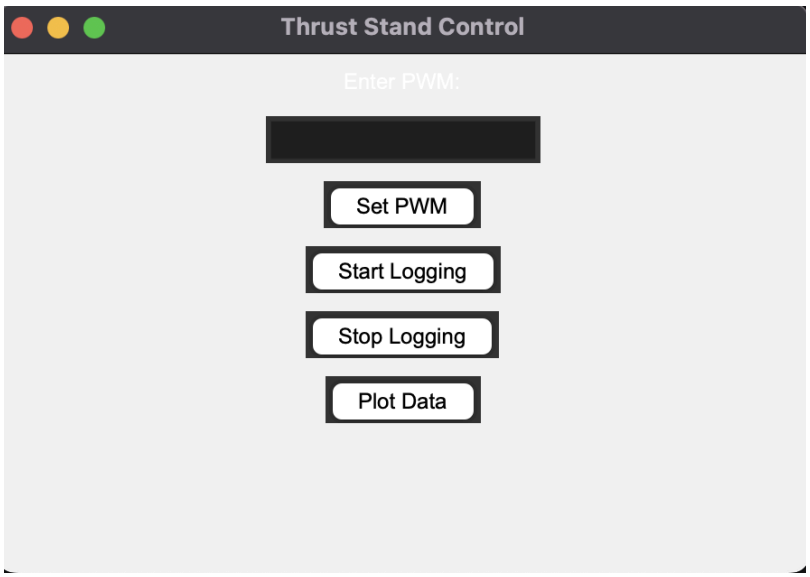
Last Updated: May 6, 2025

This manual guides first-time users through setting up and operating the software for the Arduino-based thrust stand, which measures motor performance (thrust, torque, RPM, power). It assumes the hardware is set up (see Hardware Manual). The software includes an Arduino script (`Arduino_ThrustStand.ino`) and a Python script (`Python_log_data.py`) for data logging and visualization.

1. Software Setup

1. **Install Required Software**:
    - **Arduino IDE** (Version 2.3.6 or later):
    - **PyCharm** (Version 3.5):
2. **Download Scripts**:
    - Visit the GitHub repository: ThrustStandProject.
    - Download:
        - `code/Arduino_ThrustStand.ino`
        - `code/Python_log_data.py`
3. **Set Up Arduino**:
    - Connect the Arduino (e.g., Uno) to your computer via USB cable (ensure hardware is wired per Hardware Manual).
    - Open `Arduino_ThrustStand.ino` in Arduino IDE.
    - Install required libraries (listed at the top of the script):
        - `HX711.h` (for load cells)
        - `Servo.h` (for ESC control): Included with Arduino IDE.
    - Click **Upload** to compile and upload the script to the Arduino.
        - The motor may beep briefly and stop (normal ESC initialization).

4. **Set Up Python Script**:
    - Open `Python_log_data.py` in PyCharm.
    - Install required Python libraries:
    - Modify the script:
        - **Line 11**: Update the serial port name to match your computer (e.g., `COM3` on Windows, `/dev/ttyUSB0` on Linux/macOS).
            - Find the port in Arduino IDE under Tools > Port or Device Manager (Windows).
        - **Line 17**: Change the output CSV filename (e.g., `thrust_data_2025.csv`) to your preference.
    - Save the script.
5. **Run the Python Script**:
    - In PyCharm, click **Run**
    - A GUI window should open (check behind other windows if it doesn't appear).
        - The ESC may beep frequently, then stop after ~5 seconds (normal wakeup process).
    - The thrust stand is now ready for testing.



2. Testing Operation

1. **Daily Testing**:
    - For subsequent tests, only run `Python_log_data.py` (no need to re-upload the Arduino script unless modified).
    - Ensure the Arduino is connected and powered.
2. **ESC Wake-Up Process**:
    - The ESC starts in **sleep mode** (PWM = 0), causing beeping.
    - The Arduino script sends a weak signal (PWM = 800) to wake the ESC, stopping the beeps after ~5 seconds.
    - If a high PWM (>800) is sent before waking, the ESC beeps continuously and rejects the input. Reset to PWM = 0 and wait.
    - The Python script automates this wake-up process.
3. **Running a Test**:
    - Use the GUI to send PWM commands (default starts at 0 after wake-up).
    - Click **Start Logging** to begin recording data (thrust, torque, RPM, power).
    - Adjust PWM via the GUI to test motor performance.
    - Click **Stop Logging** to end the session (resets time to 0 for clean CSV data).

3. Using the GUI

- **Controls**:
    - **PWM Input**: Set motor speed (PWM range defined in Arduino script, e.g., 800–2000).
    - **Start/Stop Logging**: Begins/ends data collection. Use once per CSV file to avoid messy data.
    - **Plot Data**: After stopping, click to view plots (PWM vs. Thrust, Torque, RPM, Power vs. Time).
- **Output**:
    - Data saves to a CSV file (name set in Line 17 of `Python_log_data.py`).

    - Move the CSV to a separate folder for analysis (e.g., `data/thrust_data_2025.csv`).
- **Notes**:
    - Expect a slight delay (~1–2 seconds) when starting commands due to serial communication.
    - Plots and data are unaffected by the delay.

4. Troubleshooting

- **Arduino Upload Errors**:
    - Ensure libraries (`HX711.h`, `Servo.h`) are installed.
    - Verify Tools > Board and Tools > Port in Arduino IDE.
    - Check USB cable and Arduino connection.
- **Python Script Fails**:
    - Confirm `pyserial` and `matplotlib` are installed (`pip show pyserial matplotlib`).
    - Check the serial port name in Line 11 (e.g., `COM3`).
    - Ensure the Arduino is connected before running the script.
- **GUI Doesn’t Appear**:
    - Look for the window behind other applications.
    - Restart PyCharm and re-run the script.
- **ESC Beeps Continuously**:
    - Reset PWM to 0 in the GUI and wait 5 seconds for wake-up.
    - Check ESC wiring (signal to pin 11, ground to Arduino GND).
- **Data Issues**:
    - If thrust/torque readings are noisy, verify HX711 calibration (see Hardware Manual).
    - Ensure the RPM sensor is aligned (within 1–2 mm of reflective tape).

5. Modifying the Software

- **Adding Sensors**: Edit `Arduino_ThrustStand.ino` to include new sensors Update `Python_log_data.py` to log new data.
- **Customizing Plots**: Modify `Python_log_data.py` to change plot types or CSV format.

Appendix A: Software Dependencies

- **Arduino Libraries**:
  - `HX711.h`: Load cell amplifier control.
  - `Servo.h`: ESC PWM control.
- **Python Libraries**:
  - `pyserial`: Serial communication with Arduino.
  - `matplotlib`: GUI and plotting.
- **Files**:
  - `code/Arduino_ThrustStand.ino`
  - `code/Python_log_data.py`
  - Example CSV: `data/sample_data.csv`