Guide/Outline for PiCultivator

1. Introduction

Brief Description

The PiCultivator is a comprehensive solution for controlling and monitoring environmental parameters such as temperature and humidity in automated setups like a grow box or greenhouse.

Target Audience

Tech-savvy gardeners with basic knowledge of Raspberry Pi, Linux, scripting, and electronics who want to simplify or partially automate their cultivation process.

Features Overview

- Intuitive control of lighting and climate conditions via a web interface.
- Clear display of current sensor values and relay statuses.
- Visualization of temperature and humidity trends with charts.
- Reduction or replacement of direct control via shell window.

2. System Requirements

Supported Platforms

- Raspberry Pi Zero W Rev 1.1 or more powerful models (e.g., Raspberry Pi Zero 2 W).
- Operating System: Raspbian GNU/Linux 11 (Bullseye).

Hardware Requirements

- Minimum memory: 512 MB.
- Processor: ARMv6.
- Wiring: A properly connected Raspberry Pi with temperature sensors and relays.

Network and User Configuration

- Local access to ports 22 (SSH) and 5000 (web interface).
- User and password setup required.
- WLAN credentials and hostname should be correctly configured.

3. Installation

⚠ **Important Note:** Do not download or run unknown or external scripts without thoroughly reviewing or understanding their contents. Even scripts from seemingly trustworthy sources should be handled cautiously—including this guide. Always review the source code if in doubt.

Step-by-Step Guide

Preparations:

- Connect relays, sensors, cables, etc., correctly (refer to wiring table).
- Install Raspberry Pi OS (see system requirements) and configure network and SSH access.

Download:

- The program can be obtained via the following installation script: repo.systemctl-schaefer.de/autosetup/PiCultivator/get-picultivator.sh
- Download using `wget` in the home directory, make it executable with `chmod +x`, and run it.

3. Installation Process:

- The script downloads and executes several preparatory and setup scripts in the order: Pre-Setup, Main-Setup, Post-Setup.
- Error logs saved in `~/picultivator_setup.log`.
- Temporarily downloaded scripts are moved to `~/obsolete-files` after execution.

Setup Phases

Pre-Setup:

- Perform system update, install essential tools and Python development tools, and download additional files.

Main-Setup:

- Create and activate a virtual Python environment.
- Install Python packages and libraries into the virtual environment.
- Generate and make executable the start script `start-picultivator.sh`.

Post-Setup:

- Display key information for the user.

4. Getting Started

Initial Configuration

1. Adjust storage locations and settings.
2. Installation residues can be removed if needed, though this is optional.
3. Control via the web interface at http://[local-IP]:5000.
4. Predefined thresholds:
   - Maximum temperature: 30°C.
   - Maximum humidity: 65%.
   - Minimum humidity: 50%.
   - Lighting time: 6:00 AM to 10:00 PM.
1. Adjust thresholds via sliders and input fields.

5. Detailed Features

Main Functions

- **Automation:**
- Control of temperature, lighting, and other parameters.
- **Sensor Monitoring:**
- Temperature and humidity monitoring (DHT22 sensor).
- Automatic relay switching based on thresholds.
- **Data Logging:**
- Storing sensor data in an SQLite database.
- **Web Interface:**
- Real-time data visualization.
- Control and monitoring of relays.

Technical Details

- Flask-based API for data retrieval and control.
- SQLite database for sensor data recording.

6. Troubleshooting

Common Issues

- Web interface not accessible: Check if the Flask server is running and port 5000 is available.
- Relays not switching: Verify GPIO configuration and wiring.
- Sensor data not recorded: Check connection to the DHT22 sensor.

Logs and Debugging

- Logs in ~/picultivator_setup.log.
- Analyze Flask server logs.

Contact and Support

- No warranty or support. Self-study required.
- Contact form: contact.systemctl-schaefer.de.

7. Extensions and Customizations

- Modify the web interface (HTML, JavaScript). Note: Programming knowledge required.
- Add additional sensors or actuators. Note: Technical and programming skills necessary.
- Enable encrypted access via Apache or Nginx. Note: Requires server configuration and network security knowledge.

8. Use Cases and Examples

Grow Box and Greenhouse Control

- Lighting control based on schedules or environmental conditions.
- Temperature regulation: Fans, heaters, external climate systems.
- Humidity control: Misters, humidifiers.
- Control of other devices (e.g., water pumps).

9. License and Legal Notes

Licensing Model

- **GNU General Public License v3.0 (GPL-3.0):**
- Free to use, modify, and distribute, provided derivative works are also published under GPL-3.0.
- Disclaimer: Provided "as is" without warranty.
  Full license text available at: gnu.org/licenses/gpl-3.0.html

10. Updates and Upgrades

⚠ **Important Note:** Do not download or run unknown or external scripts without thoroughly reviewing or understanding their contents.

- Update Python packages in the virtual environment.
- Additional maintenance scripts available in the repository at:
  repo.systemctl-schaefer.de
  Current location of updater at the time of writing (may be outdated):
  .../tools/maintenance/client-maintenance.sh

11. Acknowledgments and Credits

Contributors and Tools Used

- **APT Packages:**
- python3-venv: Virtual environment setup.
- RPi.GPIO: GPIO support for Raspberry Pi.
- speedtest-cli: Network testing.
- **Python Libraries:**
- Flask: Web framework for the interface.
- RPi.GPIO: GPIO pin control.
- pytz: Timezone management.
- Werkzeug: Backend support for Flask.
- Jinja2: Template system for the web interface.
- Adafruit CircuitPython DHT: Support for DHT22 sensors.

12. FAQ

- **"What if an error occurs?"** Check logs, e.g., ~/picultivator_setup.log.
- **"Can I use PiCultivator on [Hardware Y]?"** Ensure it meets minimum requirements (e.g., Raspberry Pi Zero W or better).
- **"How do I access the web interface?"** Open http://[local-IP]:5000 in a browser.
- **"What if relays are not switching?"** Verify GPIO configuration and wiring.
- **"How do I fix sensor monitoring issues?"** Check the DHT22 sensor connection and ensure the correct GPIO pin is used.
- **"How do I update PiCultivator?"** Use the maintenance script provided in the repository.
- **"Can I add additional sensors?"** Yes, but this requires programming knowledge and software adjustments.
- **"Why won't the Flask server start?"** Ensure Python dependencies are correctly installed and port 5000 is not blocked.
- **"Can I encrypt web interface access?"** Yes, via a reverse proxy setup with Apache or Nginx.
- **"What happens if an update fails?"** Analyze the logs and ensure all dependencies are correctly installed.

13. Appendix

- gpio-pins.png
- pinout-table.xlsx
- license.txt
- webinterface-example.png