Documentation Plantomation V1.0

Automatic Plant Watering System

Made with LaTeX

PLANTOMATION

Synthron

admin@synthron.de

Contents

1 Prelude

Plantomation is a project started by a friend of mine who had the idea of a self-watering flower pot. After some deliberation and initial drafts he pulled out of it for personal reasons.

I now restart the project because I always forget to water my house plants and eventually they all die. This will be a thing of the past thanks to this small project!

2 Introduction

Plantomation is an automatic plant watering system based on ESP32. It can handle up to four different plants simultaneously and is fully configurable via a simple web interface.

It uses capacitive soil moisture sensors to detect the humidity of the soil in the flower pot. Via a pump and valves, water can be delivered to the plants that need it.

By applying a threshold, the amount of water used for plants can be customized to the plants specific needs.

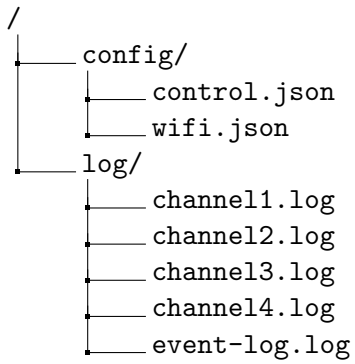
The module is supplied by external 12V DC. Pumps and Valves are driven off 12V directly, moisture sensors are driven by 5V, the spillage sensor is connected to 3.3V.

Config and Log files are saved on the SD card.

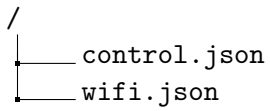
3 File System

The SD card is connected via SPI and is necessary for logging and external config. If no SD-Card is present, the internal config will be used, but no logging will be done. The internal config will always be overridden by the externel config at startup if available.

File Tree SD-Card:



File Tree Internal SPIFFS:



control.json

Contains information about the channels and their operation. Names, humidity thresholds and operation modes (moisture control/time control/disabled) are stored here for the ESP to act upon and to display in the web interface. The log_enable keys can be used to enable or disable logging for this channel. (SD only)
Default is enabled, errors will always be logged if SD is available.

wifi.json

Contains all necessary information for wifi-usage, like SSID, password and if the ESP should be in station mode (client inside existing network), or in AccessPoint-Mode (creating its own network). Default is AP-Mode.

Logfiles

If logging is enabled, the ADC-values (hourly) and watering-events will be logged. This way long-term data about the moisture and water usage can be obtained for further fine-tuning. In the event-log, important events will be logged as well, like page loads, spill detects, container empty etc.

4 Configuration

Plantomation mainly uses two json files for the configuration of the system.

4.1 Channel-Config

The config.json is the main config file for the operation of the system. It contains all necessary infos on what to do with each channel as well as some general settings.

The filestructure is as follows:

```
plant1.json
├── <name>
├── <op-mode>
├── <moisture>
├── <interval_time>
├── <pump_time>
└── <log-enable>
.
.

plant4.json
├── <name>
├── <op-mode>
├── <moisture>
├── <interval_time>
├── <pump_time>
└── <log-enable>
```

json Key Description:

- name: Name given to channel/plant
- op-mode:
  - 0: disabled
  - 1: moisture control
  - 2: time control
- moisture: value between 0..100 of moisture sensor range
- interval_time: duration in-between waterings (time-control mode only)
- pump_time: duration for which the pump should be active
- log_enable: 1 enables logging (SD card only)

4.2 Global Config

The config.json contains all information regarding the operations of the system.

The filestructure is as follows:

```
config.json
├── <log-level>
├── <debug-level>
└── <pump_rate>
```

json Key Description:

- log-level:
  - 0: no logging of events
  - 1: only log errors
  - 2: log errors and page connections
- debug-level:
  - 0: no debug output
  - 1: no cyclic debug messages
  - 2: all debug messages
- pump_rate: flowrate of the pump in ml/min

4.3 wifi.json

The wifi.json contains all information regarding the intended network and mode.

The filestructure is as follows:

```
wifi.json
├── <mode>
├── <ssid>
├── <passwd>
└── <hostname>
```

json Key Description:

- mode:
  - 0: Access Point Mode
  - 1: Station Mode
- ssid: Name of the WiFi-network
- passwd: Password for the network
- hostname: Hostname of the client

5 Getting Started

5.1 Preparing SD-Card

If you want to use an SD-Card, format it to FAT32 and create the filestructure layed out in **3. File System**.

You can set the necessary configurations directly before startup with the mentioned JSON files.

If you don't want to use an SD-Card, skip this step.

5.2 Starting up

Now connect the Plantomation-Module to power and let it boot. By observing the LEDs, you can see the status of the module. Each LED is indicating a specific event. These boot messages will be displayed for 3 seconds before operation continues.

LED 1 OK/NOK System	LED 2 OK/NOK Wifi	LED 3 OK/NOK SD	LED 4 OK/NOK JSON	Status Description
0	0	0	0	Filesystem ERROR - HALT
0	0	1	0	JSON config corrupt - HALT
1	0	0	1	Boot OK - Wifi config (Flash) failed - use Default
1	0	1	1	Boot OK - Wifi config (SD) failed - use Default
1	1	0	1	Boot without SD OK
1	1	1	0	Boot OK with Default - Empty SD formatted
1	1	1	1	Boot with SD OK

5.3 Wifi-Connection

If booting with your custom Wifi-Configuration succeeded, you can now connect to the IP sent out via USB after booting.

If you didn't specify any Wifi-Configuration or the configuration failed, Plantomation will create its own Wireless Network which you can connect to. The "XX-XX-XX" is referring to the last three bytes of the ESPs MAC-address.

SSID: Plantomation-XX-XX-XX
Password: Planto1!
IP-Address: 4.3.2.1

5.4 Configuration

After successfully booting, you can now connect to the IP address and visit the “Configuration”-Tab on the Webinterface. Here you can graphically configure all general settings like channel modes, log levels etc.

After setting the channel modes, go back to “Control” and set up your channels.

Congratulations, you are now done!.

6 Troubleshooting

6.1 Spill Sensor

If the Spill sensor activated, Plantomation will go into a Lockdown Mode, where all valves will get closed and the pump shut off.

Additionally, all four LEDs will blink at 1Hz.

To Reset the alarm, either reset the board or press the “Boot0”-Button. This will acknowledge the alarm and resume normal operation. Make sure to have dried up the previous spill, otherwise the Lockdown Mode will happen again.

6.2 Empty Reservoir Warning

If an empty water supply is detected, the LEDs will blink in an alternating pattern.

If this happens, refill the water reservoir and press the “Boot0”-Button.

6.3 Booting Problems

During the Boot-Phase an error can occur halting the module and preventing a successful boot.

6.3.1 Filesystem ERROR

In main.cpp, set “**#define** FORMAT_SPIFFS_IF_FAILED” to `true` and reupload the code. Then set it to `false` and reupload the code again.

This should reset and format the SPIFFS partition to a usable state.

6.3.2 JSON config corrupt

This error occurs, when one or more config files from the SD are corrupt, i.e. not in a readable state by the firmware. This can happen due to poor formatting or typos.

If this is the case, remove the SD-Card and check the files. If the error continues to show up, delete all files from the SD-Card and insert it blank into Plantomation. On first startup, valid files will be created and a default network will be opened. You can then either use the Web-Interface or a local text-editor for changing the config.

7 Operation Description

7.1 General Operation

During booting up, the config files from the SD (if available) are loaded into RAM, checked for integrity and transferred into internal Flash memory. If no SD is available, Configs will be loaded from Flash directly.

Then the system will be initialized, starting with pin definitions, Wifi-Network and Webserver.

The webserver and the OTA-handler are allocated on Core0, while the application firmware is running on Core1. This way, the response time of the webserver is not influenced by the application and vice-versa.

After a successful boot, the application will start running.

If Plantomation is connected to a network with internet access, it will periodically (once per day) synchronize its internal calendar via NTP. This way log files will have the correct time stamps.

7.2 Plant Control

Plant control can have 3 modes of operation: disabled, moisture-controlled and time-controlled.

When channels are disabled, they won't do anything.

7.2.1 Moisture Control

On Moisture Control, the capacitive Soil Moisture will be read 6 times every 10 minutes with a delay of 10 seconds between measurements.

The readings will be averaged and compared to the set moisture level. If the level is below the desired value, the valve will be opened and the pump will dispense a set amount of liquid.

If after 5 cycles the moisture sensor value doesn't change, a "Water Reservoir Empty"-Error will be triggered and operation will be halted until it gets acknowledged.

7.2.2 Time Control

When time controlled, the channel will be activated according to the specified time interval. The pump will then deliver the set amount of water and the valve will close again.

7.3 Calibration