Irrigation System

Admin GUIDE

|  |  |
| --- | --- |
| **1 Raspberry Pi Setup** | Page 2 |
| **2 Software Configurations** | Page 2 |
| **3 Required Materials** | Page 2-3 |
| **4 Wiring Diagrams** | Page 3 |
| **5 INITIALIZING DEVICES** | Page 3-4 |
| **6 First Time Configuration(Moisture Sensor)** | Page 4 |
| **7 Installing System** | Page 5 |
| **8 Trouble Shooting** | Page 5-6 |
| **9 Updating** | Page 6 |
| **10 Image Key** | Page 6-7 |

**1 Rasberry Pi Setup**

1A) The Admin will need to install proper python libraries using pip install manager onto the raspberry pi to be used.

* paho-mqtt

1B) Ensure that the entire Irrigation System folder is downloaded and placed on desktop or easily accessible location where users can access.

1C) Ensure that Device has valid internet connection

**2 Software Configurations:**

The following files have credentials and settings that only the admin should have access to. It is highly recommended that the Admin set permissions for these files after initial configuration is finished.

2A) creds\_and\_settings.py

* This will store the MQTT Broker information, coordinates of system, and other credentials. If a request is made to use a new broker, the change for new devices can be made here

2B) configTemplate.json

* This is the config File that will be loaded to all newly created devices, the Admin can give any wifi/password combination in the file to be used.
* The Admin can also change the Mqtt broker information to be loaded to future devices.

*Note: If the wifi or broker change after a device has been created. Then the devices will need to have its config file updated manually by opening the config file on the device via instructions in section 7*

**Hardware Configuration**

**3 Required Materials:**

*Note: If required, specific link supplied, else generic devices will do*

3A) Moisture Sensors

* 1. Pico-w(optional headers to install if not wanting to solder directly to board)
     1. <https://thepihut.com/products/raspberry-pi-pico-w?variant=41952994754755>
     2. https://thepihut.com/products/male-headers-for-raspberry-pi-pico
  2. Moisture Sensor
     1. https://www.amazon.com/gp/product/B07SYBSHGX/ref=ppx\_yo\_dt\_b\_asin\_title\_o06\_s00?ie=UTF8&psc=1
  3. Micro-Usb Power supply
  4. Female -> Male Jumper wires if not soldering to board

3B) Valve System(Good for 4 relays)

* 1. Pico-w (Male-Headers both Sides)
     1. https://thepihut.com/products/raspberry-pi-pico-w?variant=41952994787523
  2. 4-Relay board
  3. 4 valves
     1. https://www.amazon.com/dp/B000A16QSO?ref\_=pe\_1815430\_211938580
  4. Water Meter
     1. <https://www.amazon.com/dp/B01EA3CA7I?ref_=pe_1815430_211938580>
        1. Ensure this meter is used, as the system is calibrated for this size
  5. 24 vac Transformer
     1. https://www.amazon.com/dp/B07FT2RCG9?ref\_=pe\_1815430\_211938580
  6. Waterproof Wire Nut
  7. PVC Pipe/Hose cut to size
  8. PVC Glue/Primer
  9. Wiring(Length dependent on User needs)

**4 Wiring Diagrams**

**4A) Moisture Sensor:**

Admin can either use jumper cables on a Pico-W with male-headers or solder the wires directly to the pico in the following positions from the moisture sensors to the Pico-W

**1) Yellow Wire -> GP26\_AO**

**2) Black Wire -> GND**

**3) Red Wire -> 3V3**

**4B) Valve Controller:**

Utilizing the relay board, wire up as in diagram in Image 2 on last page.

Wire water meter using jumper cables in the following configuration to the Relay Board pinouts

**1) Yellow Wire -> GP13**

1. **Black Wire -> GND**
2. **Red Wire -> 3V3\_0**

**5 INITIALIZING DEVICES**

After a configuration file has been created from User Guide 3A.1), connect the appropriate raspberry pi from 3A or 3B to the raspberry pi via the micro-usb port on the pico-w.

Then, do the following:

**5A)** **Flashing Device:**

Download micro python to device utilizing official Raspberry Pi Documentation:

* + <https://www.raspberrypi.com/documentation/microcontrollers/micropython.html>
  + *Note: Be sure to install the Raspberry Pi Pico-W UF2*

**5B)** **INITIALIZING Thonny Environment:(First Time Running Dev Environment)**

The following will be used for creating your development environment for the first time.

If using the Raspberry Pi, Thonny will already be installed and Admin will start at 5B.2)

1. Install Thonny from <https://thonny.org/>

2. Connect Flashed Raspberry Pi Pico-W to computer and go to Tools>Options>Interpreter and select MicroPython (Raspberry Pi Pico)

3. Click OK on bottom of page.

**5C) Loading Config File:**

Take the correctly named config file and save each of the following files/folder to the

device

* + Lib/
  + Config(changes depending on device type)
  + Main.py
  + Valve.py or Moisture.py(depends on 3A or 3B)
  + Calibration.py(if 3A)

**6 First Time Configuration(Moisture Sensor):**

After flashing and loading the config file, the admin will need to run a calibration test on the moisture sensors the first time they are run.

To do this:

1) Connect devices to the computer and open Thonny

2) Select “MicroPython (Raspberry Pi Pico)” on bottom right-hand corner of screen.

3) Select open file, and the user will be given the choice between the computer, and the

Raspberry Pi Pico. Select the Raspberry Pi Pico.

4) Open main.py from device.

5) Click the green run button at the top of the screen

After going through these steps, the user will be prompted to go through calibration. Once completed, the user can disconnect the device and use it.

**7 Installing System:**

* It is up to the user to ensure that each Moisture Sensor System is being setup with its configured Valve/Relay Number. This information can be found when creating the device, and an example is shown under User Guide 3A.1)
* When installing plumbing, it is recommended to let the pipes sit for 2-3 hours depending on glue/primer used to ensure proper drying and avoid glue getting into electronic valves/meters.
* Be sure to install Moisture meters before all water Valves on the plumbing such as in image 1. This way there is only one meter needed per 4 valves with the supplied board.
* It is highly recommended to use waterproof containers for the relay board/moisture sensor Pico-W to avoid damage.
* When installing the actual moisture sensor into the ground, it is recommended to have it be inserted up-to the V2 Marking on the sensor.
  + It is also recommended to install waterproofing around the circuits of the moisture sensor(NOT PICO) to avoid water damage. This can be done with heat-shrink around the circuits, and silicon around the edges.

**8 Trouble Shooting:**

If errors occur, they can be found in one of three locations, and have the potential to be debugged from there

8A) Subscribe and waterQueue: From here, if an issue occurs the Admin will be able to see

the error that occurred and what happened.

* + The most likely causes will be database errors, be sure that devices are properly setup and saved to the database when creating new devices.
  + If manually making adjustments to the database through tools like dbBrowser, ensure that you write/save any changes as soon as they are made. If this does not happen, the database remains open and it is unable to be written to from the scripts and will cause an insert failure.
  + Occasionally due to network errors the system may receive more requests than it can handle, this has a catch that will reset the system, so this occurrence can be noted if it happens.

8B) Pico-W Errors:

* + Due to the nature of the devices, it can be difficult to debug the systems if an error occurs. Because of this there is an implementation of a logging system that will save any issues to a logfile.txt on each device. To access this file, go through steps 7.1-4, but open the logfile.txt instead of main.py. This will show what happened before the system crashed.
    - Most errors are caused by Connection Timeouts, and Mqtt Client disconnects.
    - If it is found that a system is not working by using statusChecker.py, the Admin can disconnect power to the devise and force a restart.

**9 Updating:**

Currently there is no system in place to update a device once it has been created (i.e. Changing a devices plant type or valve) If a user would like to request this change the best course would be to add additional functionality to the system to allow this, or manually make changes within the database via tools such as dbBrowser (<https://sqlitebrowser.org/>).

**Image Key**

Image 1(Middle pipe has water coming from water source, to meter, to water valves):

A close up of a pipe

Description automatically generated with medium confidence

Image 2(Wiring Diagram for valves):

**A diagram of a machine

Description automatically generated**