SwitchDoc Labs

The SmartGarden3
Configuration and Operations Manual

January 2022
Version 1.2

Contents

Why you should build this project

Imaging buying a fully functional computer for $35 and then tracking the weather at your home, and even watering your houseplants from your phone while away on vacation? Imagine learning a little about electronics without messing with soldering, resistors, and transistors? Ever been curious about "coding" but have no interest in taking a class? This great little project will take you less than six hours at your kitchen tables and when you are done, you will know:

- What a solenoid does
- The purpose of a relay
- Python Coding Language
- How a simple computer works
- How stuff connects to the Internet and your phone

This is the ultimate IoT you have been reading about – the Internet of Things. If you are a tinkerer and love to invent stuff, this is for you. Engage your kids and learn something new together!

What is The SmartGarden3?

Imagine building your own remote monitoring and management system for your indoor or outdoor garden? How about sharing your garden and the weather worldwide? You can do all this and learn a little about electronic circuitry without touching a **soldering** iron. You can measure soil moisture and use that as water your plants or garden just the right about of water.

Highly expandable! You can have your Raspberry Pi Base Unit inside your house and have multiple wireless control units in your outdoor garden, greenhouse or in the upstairs bedroom. Up to 250 wireless control units can be connected to one Raspberry Pi base unit. This allows you to control your truly MASSIVE garden. Or your small one. Either way!

This is a perfect highly expandable project kit for kids with some help from the adults and for adults trying to learn some new things This project requires no **soldering** and uses Grove connectors to wire everything up! You cannot reverse them and blow things up. Here is our tutorial on the Grove system.

The SmartGarden3 Features

- Control Valves, Pumps and Lights
- Measure your Soil Moisture
- Measure your Sunlight
- Measure your Air Quality
- Measure your Weather
- Show your results on the Internet and your Phone

Easy to build. Easy to learn about the IOT (Internet Of Things) and the Raspberry Pi.

Cautions when building and using The SmartGarden3

1) **Keep all water away from the electronics and power supply at all times!**
2) **The SmartGarden3 is designed for indoor use only and should be placed in a dry environment where no water or rain can reach to avoid short circuiting the electronics**
3) **Insert the moisture sensor into the CENTER of the flower pot, and keep it near the center of the plant and away from the water coming out of the holes cut in the watering pipe.**
4) **This is not a toy! Keep it out of reach of young children and pets.**
5) **SwitchDoc Labs assumes no liabilities in the use of this kit, beyond the refund of the purchase price.**

Ready to begin!

Find a clean, dry indoor workspace where you can spread out the parts without them being disturbed.

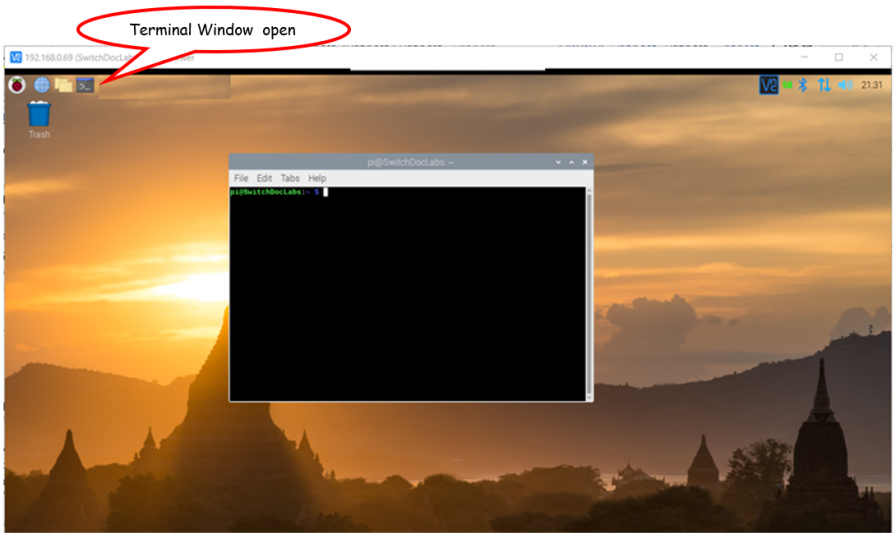
- Static is your enemy. You can control static by touching a grounded object such as a radiator with one hand when touching an electronic component with the other. If you are a geek you can buy an anti-static mat for as little as $25.
- Always keep water away from the electronics and power supply! Keep that coffee cup at a distance!
- The SmartGarden3 electronics are designed for indoor use only and should be placed in a dry environment. Of course, certain parts are designed for outdoor use, such for monitoring soil moisture and watering your yard.
- This is not a toy! Keep it out of reach of young children and pets.
- SwitchDoc Labs assumes no liabilities in the use of this kit, beyond the refund of the purchase price.

Using this Manual:

You do not need to learn how to code. You will be given specific commands to type into the Terminal Window. These commands are written in a popular programming language called Python. Therefore, you are in fact writing code.

Along the way you will pick up some tidbits, for example, the command the entry “sudo” tells the software you are a super-user, allowing you to do stuff not available to normal users. Sudo is often followed by the word “python” to tell the software to act on your command in Python language. Finally, pay close attention to your type case: “Python” and “python” are not the same command.

When we talk about the Terminal Window, we are referring to this on the Raspberry Pi:



When you need to type something in the Terminal Window, we will show what you should type in this font and color:

```
cd SDL_Pi_SmartGarden3
```

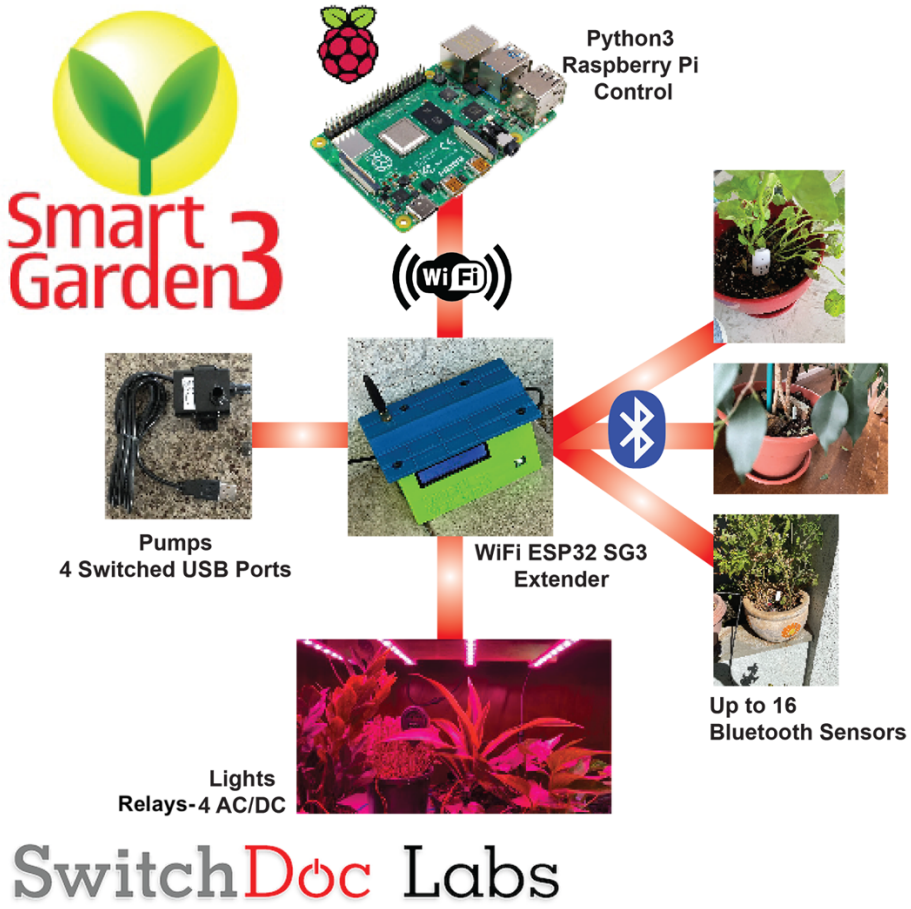
What is displayed in the terminal window is displayed like this:

```
SG3Configure.py:24: DeprecationWarning: AppURLopener style of invoking requests is
deprecated. Use newer urlopen functions/methods
  myURLOpener = AppURLopener()
```

Assembling the SmartGarden3

Begin your project by assembling the SmartGarden3 components. You will need to complete steps outlined in the SmartGarden3 Assembly and Test Manual.

Step 1: With your SmartGarden3 up and running, you now want to make sure your Wireless Extender unit(s) are plugged in and connected to your Wi-Fi network.



Step 2: Determine your Raspberry Pi IP address on your network. Follow the procedures here: https://learn.pimoroni.com/tutorial/raspberry-pi/finding-your-raspberry-pi

In a terminal window, change directories down to:

```
cd SDL_Pi_SmartGarden3
```

Run SG3Configure.py

```
sudo python3 SG3Configure.py
```

You will see something like this:

```
pi@SwitchDocLabs:~/SDL_Pi_SmartGardenSystem2 $ sudo python3 SG3Configure.py
SG3Configure.py:24: DeprecationWarning: AppURLopener style of invoking requests is
deprecated. Use newer urlopen functions/methods
  myURLOpener = AppURLopener()
remi.server      INFO     Started httpserver http://0.0.0.0:8001/
remi.request     INFO     built UI (path=/)
SGS.JSON File does not exist
SGSConfiguration.JSON File does not exist
```

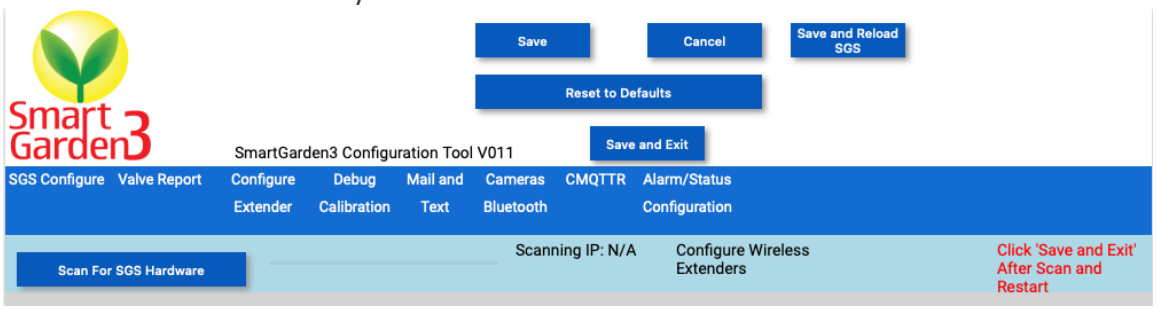
Now open a browser window (either on the Raspberry Pi or on another computer on your local WiFi network) and enter this URL:

On your Raspberry Pi: http://127.0.0.1:8001/

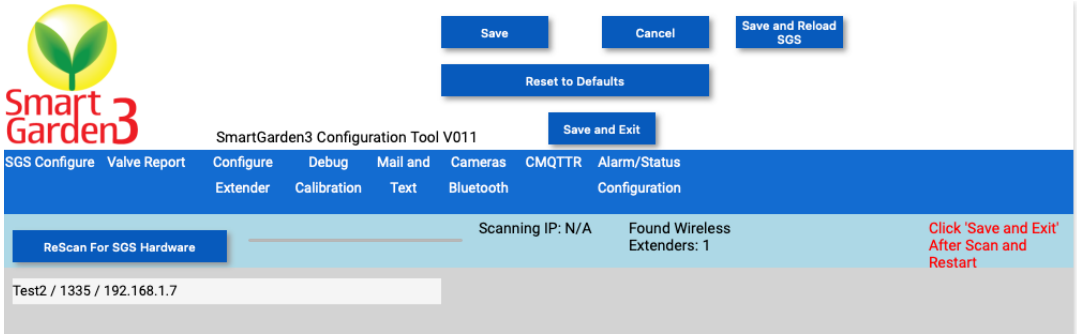
If you do this on another computer type: http://xxx.xxx.xxx.xxx:8001/

NOTE: The "xxx.xxx.xxx.xxx" is the IP address of **your** Raspberry Pi that you wrote down above.

You will see this screen on your browser:



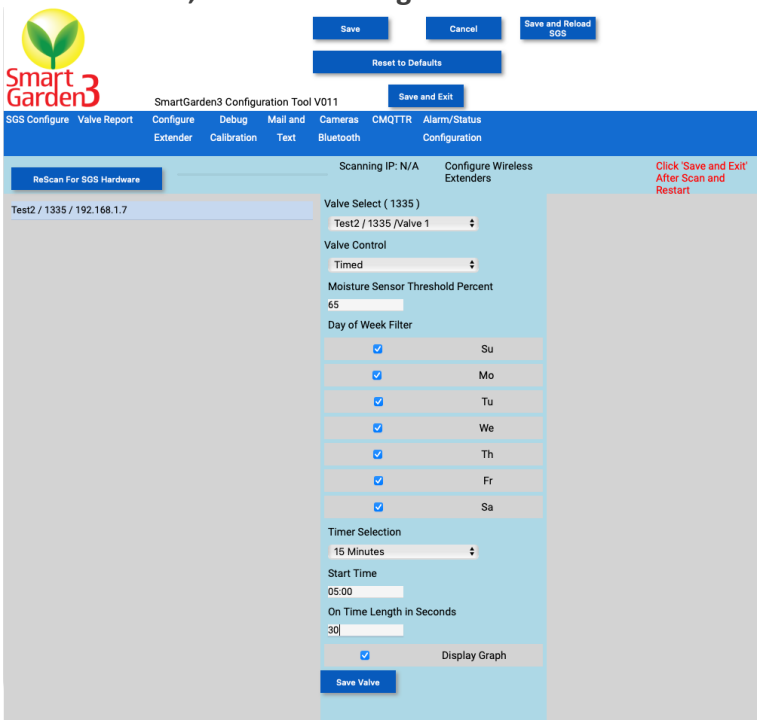
Step 3: Click on the "Scan For SG3 Hardware Button". The first thing we do is have the SG3Configure software scan for all your wireless units. Your base unit comes with one system. This takes a while (15 or 20 minutes). Click on the "Scan For SG3 Hardware Button".



SG3Configure found one SG3 Wireless Unit: "Test2/1335/192.168.1.7". Note that we had named this unit "Test2" earlier. Yours will probably show up blank. Click "Save and Exit" and restart SG3Configure.

Step 4: Click on the Wireless Unit on the screen to open up the Valve configuration menus. "Test2/1335/192.168.1.7"in our example.

Select **valve 1** – using the Valve Select menus (make sure you still have the USB Light Stick plugged into Valve 1) to turn on for 30 seconds every 15 minutes and click Show Graph. Note that you can only select the Timer Selection and Start Time if you have selected "Timed" in the Valve Control dropdown menu. You should see the screen shown here: Finally click the "Save Valve" button. **If you do not do this, the valve changes are NOT saved.**



Step 5: Click on the “Debug Calibration” Tab on your menu.

Click on enable SW Debugging (you can turn this off later).

Click on enable MySQL logging. The default MySQL password on the SDL SD Card is “password”.

Click on Save and Exit which saves your JSON files for SG3 and quits the SG3Configure program.

Note: On some systems, you may have to hit either “ctrl-c” or the “ctrl-|” to get the server to quit in the terminal window.

Look at the JSON files to see what you have done! SG3.JSON is the general configuration file, while SGSConfiguration.JSON is the valve/pump/timers configuration file for all the wireless extender units.

```
pi@SwitchDocLabs:~/SDL_Pi_SmartGarden3 $ more SGS.JSON
{"key": "value", "ProgramName": "SmartGarden3", "ConfigVersion": "001", "SWDEBUG
": false, "enable_MySQL_Logging": true, "English_Metric": false, "MySQL_Password
": "password", "mailUser": "yourusername", "mailPassword": "yourmailpassword", "
notifyAddress": "you@example.com", "fromAddress": "yourfromaddress@example.com",
 "enableText": false, "textnotifyAddress": "yournumber@yourprovider", "INTERVAL_
CAM_PICS__SECONDS": "60", "REST_Enable": false, "Camera_Night_Enable": false, "M
QTT_Enable": false, "MQTT_Server_URL": "", "MQTT_Port_Number": "1883", "MQTT_Sen
d_Seconds": "500", "manual_water": true, "Send_Status_Email": false, "Status_Sen
d_Email_Minutes": "300", "Send_Status_Text": false, "Status_Send_Text_Minutes":
"300", "Tank_Pump_Level_Full": "500", "Tank_Pump_Level_Empty": "900", "Infrared_
High_Auto_Gain": false, "Infrared_Low_Auto_Gain": false, "Infrared_Low_Temp": "1
7.0", "Infrared_High_Temp": "22.0", "WirelessDeviceJSON": [{"return_value": 0, "
id": "1335", "name": "Test2", "ipaddress": "192.168.1.7", "hardware": "esp32", "
hydroponicsmode": "false", "return_string": "1335,1,1,1,1,1,1,1,1,0,059", "conne
cted": true, "hydroponics_temperature": "false", "hydroponics_tds": "false", "hy
droponics_ph": "false", "hydroponics_turbidity": "false", "hydroponics_level": "
false"}]}

pi@SwitchDocLabs:~/SDL_Pi_SmartGarden3 $ more SGSConfiguration.JSON
{"SGSConfigVersion": "001", "Valves": [{"id": "1335", "ValveNumber": 1, "Control
": "Off", "MSThresholdPercent": "65", "TimerSelect": "Daily", "DOWCoverage": "YY
YYYYY", "StartTime": "05:00", "OnTimeInSeconds": "10", "ShowGraph": false}, {"id
": "1335", "ValveNumber": 2, "Control": "Off", "MSThresholdPercent": "65", "Time
rSelect": "Daily", "DOWCoverage": "YYYYYYY", "StartTime": "05:00", "OnTimeInSeco
nds": "10", "ShowGraph": false}, {"id": "1335", "ValveNumber": 3, "Control": "Of
f", "MSThresholdPercent": "65", "TimerSelect": "Daily", "DOWCoverage": "YYYYYYY"
, "StartTime": "05:00", "OnTimeInSeconds": "10", "ShowGraph": false}, {"id": "13
35", "ValveNumber": 4, "Control": "Off", "MSThresholdPercent": "65", "TimerSelec
t": "Daily", "DOWCoverage": "YYYYYYY", "StartTime": "05:00", "OnTimeInSeconds":
"10", "ShowGraph": false}, {"id": "1335", "ValveNumber": 5, "Control": "Off", "M
SThresholdPercent": "65", "TimerSelect": "Daily", "DOWCoverage": "YYYYYYY", "Sta
rtTime": "05:00", "OnTimeInSeconds": "10", "ShowGraph": false}, {"id": "1335", "
ValveNumber": 6, "Control": "Off", "MSThresholdPercent": "65", "TimerSelect": "D
aily", "DOWCoverage": "YYYYYYY", "StartTime": "05:00", "OnTimeInSeconds": "10",
"ShowGraph": false}, {"id": "1335", "ValveNumber": 7, "Control": "Off", "MSThres
holdPercent": "65", "TimerSelect": "Daily", "DOWCoverage": "YYYYYYY", "StartTime
": "05:00", "OnTimeInSeconds": "10", "ShowGraph": false}, {"id": "1335", "ValveN
umber": 8, "Control": "Off", "MSThresholdPercent": "65", "TimerSelect": "Daily",
 "DOWCoverage": "YYYYYYY", "StartTime": "05:00", "OnTimeInSeconds": "10", "ShowG
raph": false}]}
```

Initial Testing of your SmartGarden3

Step 6: Start your SG3 system in a terminal window by typing:

```
sudo python3 SG3.py
```

You will see something like this as it scrolls across your screen. By the way, if you turn Software debugging ON (table "DM" - Debug configuration on) in SG3Configure you will see much more on your terminal window.

```
SGSConfiguration.JSON File exists
##############################################
SG3 Version 050  - SwitchDoc Labs
##############################################

Program Started at:2022-01-30 06:21:39

SGS.JSON File exists
JSON value not found - Set to Defaults: USEBLYNK
JSON value not found - Set to Defaults: BLYNK_AUTH
SGSConfiguration.JSON File exists
----------------------
Local Devices
----------------------
----------------------
Checking Wireless SG3 Devices
----------------------
myCommand= assignBluetoothSensors?params=admin,C4:7C:8D:6B:90:39
Test2 - 1335:              Present
myCommand=enableHydroponicsMode?params=admin,0,0 myIP=192.168.1.7
UpdateStateLock Releasing - initializeOneExtender
subscribing to  SGS/1335


----------------------
Plant / Sensor Counts
----------------------
Wireless Unit Count: 1
Valve Count:  8
Bluetooth Sensor Count:  4

----------------------
Other Smart Garden System Expansions
----------------------
GardenCam:          Present
Lightning Mode:            Not Present
MySQL Logging Mode:               Present

----------------------
-----------------
Scheduled Jobs
-----------------
Jobstore default:
    checkForButtons (trigger: interval[0:00:10], next run at: 2022-01-30 06:21:59 PST)
    manualCheck (trigger: interval[0:00:15], next run at: 2022-01-30 06:22:04 PST)
    takeSkyPicture (trigger: interval[0:01:00], next run at: 2022-01-30 06:22:49 PST)
    valveCheck (trigger: interval[0:01:00], next run at: 2022-01-30 06:22:49 PST)
    tick (trigger: interval[0:05:00], next run at: 2022-01-30 06:26:49 PST)
    checkForAlarms (trigger: interval[0:05:00], next run at: 2022-01-30 06:26:49 PST)
    updateDeviceStatus (trigger: interval[0:12:00], next run at: 2022-01-30 06:33:49
PST)
    cleanPictures (trigger: cron[day='*', hour='3', minute='4'], next run at: 2022-01-
31 03:04:00 PST)
```

```
    cleanTimeLapses (trigger: cron[day='*', hour='3', minute='10'], next run at: 2022-
01-31 03:10:00 PST)
    buildTimeLapse (trigger: cron[day='*', hour='5', minute='30'], next run at: 2022-
01-31 05:30:00 PST)
-----------------
    updateDeviceStatus (trigger: interval[0:12:00], next run at: 2020-07-14 11:12:14
PDT)
```

When 15 minutes have passed (after the USB Light Stick has turned on and off) minutes, hit “ctrl-c’ to quit.

With software debugging ON, you would have seen this:

```
pi@SwitchDocLabs:~/SDL_Pi_SmartGarden3 $ sudo python3 SG3.py
SGS.JSON File exists
JSON value not found - Set to Defaults: USEBLYNK
JSON value not found - Set to Defaults: BLYNK_AUTH
SGSConfiguration.JSON File exists
Pi Camera Revision ov5647
################################################
SG3 Version 050  - SwitchDoc Labs
################################################

Program Started at:2022-01-30 06:23:15

SGS.JSON File exists
JSON value not found - Set to Defaults: USEBLYNK
JSON value not found - Set to Defaults: BLYNK_AUTH
SGSConfiguration.JSON File exists
----------------------
Local Devices
----------------------
----------------------
Checking Wireless SG3 Devices
----------------------
myCommand= assignBluetoothSensors?params=admin,C4:7C:8D:6B:90:39
myURL= http://192.168.1.7/assignBluetoothSensors?params=admin,C4:7C:8D:6B:90:39
myURL= http://192.168.1.7/setValves?params=admin,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
return= {'return_value': 0, 'id': '1335', 'name': 'Test2', 'ipaddress': '192.168.1.7',
'hardware': 'esp32', 'hydroponicsmode': 'false', 'return_string': '', 'connected':
True}
Test2 - 1335:              Present
MQTT:  Sending CONNECT (u0, p0, wr0, wq0, wf0, c1, k60) client_id=b'SG3'
MQTT:  Received CONNACK (0, 0)
myCommand=enableHydroponicsMode?params=admin,0,0 myIP=192.168.1.7
myURL= http://192.168.1.7/enableHydroponicsMode?params=admin,0,0
UpdateStateLock Releasing - initializeOneExtender
subscribing to  SGS/1335
MQTT:  Sending SUBSCRIBE (d0, m1) [(b'SGS/1335', 0)]
MQTT:  Received SUBACK


----------------------
Plant / Sensor Counts
----------------------
Wireless Unit Count: 1
Valve Count:  8
Bluetooth Sensor Count:  4

----------------------
Other Smart Garden System Expansions
----------------------
GardenCam:          Present
Lightning Mode:            Not Present
MySQL Logging Mode:               Present

----------------------
-----------------
Scheduled Jobs
```

```
-----------------
Jobstore default:
    checkForButtons (trigger: interval[0:00:10], next run at: 2022-01-30 06:23:35 PST)
    manualCheck (trigger: interval[0:00:15], next run at: 2022-01-30 06:23:40 PST)
    takeSkyPicture (trigger: interval[0:01:00], next run at: 2022-01-30 06:24:25 PST)
    valveCheck (trigger: interval[0:01:00], next run at: 2022-01-30 06:24:25 PST)
    tick (trigger: interval[0:05:00], next run at: 2022-01-30 06:28:25 PST)
    checkForAlarms (trigger: interval[0:05:00], next run at: 2022-01-30 06:28:25 PST)
    updateDeviceStatus (trigger: interval[0:12:00], next run at: 2022-01-30 06:35:25
PST)
    cleanPictures (trigger: cron[day='*', hour='3', minute='4'], next run at: 2022-01-
31 03:04:00 PST)
    cleanTimeLapses (trigger: cron[day='*', hour='3', minute='10'], next run at: 2022-
01-31 03:10:00 PST)
    buildTimeLapse (trigger: cron[day='*', hour='5', minute='30'], next run at: 2022-
01-31 05:30:00 PST)
MQTT:  Sending PINGREQ
MQTT:  Received PINGRESP
--------------------
Garden Cam Picture Taken
--------------------
>>>>>>Valve Check<<<<<<
newValve= {'id': '1335', 'ValveNumber': 1, 'NextTime': datetime.datetime(2022, 1, 30,
5, 15), 'LengthTurnOn': '30'}
nextMoistureValveSensorCheck =  2022-01-30 06:15:00
nextMoistureValveSensorCheck =  2022-01-30 06:30:00
--------------------
Garden Cam Picture Taken
--------------------
>>>>>>Valve Check<<<<<<
valveState Found for 1335 1
MQTT:  Sending PUBLISH (d0, q0, r0, m2), 'b'SGS/1335/Valves'', ... (113 bytes)
MQTT:  Received PUBLISH (d0, q0, r0, m0), 'SGS/1335', ...  (97 bytes)
Wireless MQTT Message received:  b'{"id": "1335", "messagetype": "1", "timestamp":
"01/30/2022 14:25:24", "valvestate": "V10000000"}'
Valve Change Received
Timer Fired!  Next Fire= 2022-01-30 06:30:00
MQTT:  Received PUBLISH (d0, q0, r0, m0), 'SGS/1335', ...  (97 bytes)
Wireless MQTT Message received:  b'{"id": "1335", "messagetype": "1", "timestamp":
"01/30/2022 14:25:54", "valvestate": "V00000000"}'
Valve Change Received
--------------------
Garden Cam Picture Taken
--------------------
>>>>>>Valve Check<<<<<<
valveState Found for 1335 1
MQTT:  Sending PINGREQ
MQTT:  Received PINGRESP
```

Configuration of your SmartGarden3

Step 7: Now we will go through all the configuration screens and describe the buttons and fields. You do not have to follow this in the order presented. Remember, you can pick and choose and always go back and change the configuration later and reload your configuration into SG3 on the fly without even stopping the program.

The first thing to figure out is what your Raspberry Pi IP address is. Follow the procedures here:

https://learn.pimoroni.com/tutorial/raspberry-pi/finding-your-raspberry-pi

Step 8: In a terminal window, change directories to:

```
cd SDL_Pi_SmartGarden3
```

Run SG3Configure.py:

```
sudo python3 SG3Configure.py
```

You will see something like this:

```
pi@SwitchDocLabs:~/SDL_Pi_SmartGardenSystem2 $ sudo python3 SG3Configure.py
SG3Configure.py:24: DeprecationWarning: AppURLopener style of invoking requests is
deprecated. Use newer urlopen functions/methods
  myURLOpener = AppURLopener()
remi.server      INFO     Started httpserver http://0.0.0.0:8001/
remi.request     INFO     built UI (path=/)
SGS.JSON File does not exist
SGSConfiguration.JSON File does not exist
```

Now open a browser window (either on the Raspberry Pi or on another computer on your local WiFi network) and enter this URL:

On your Raspberry Pi: http://127.0.0.1:8001/

On your Raspberry Pi:

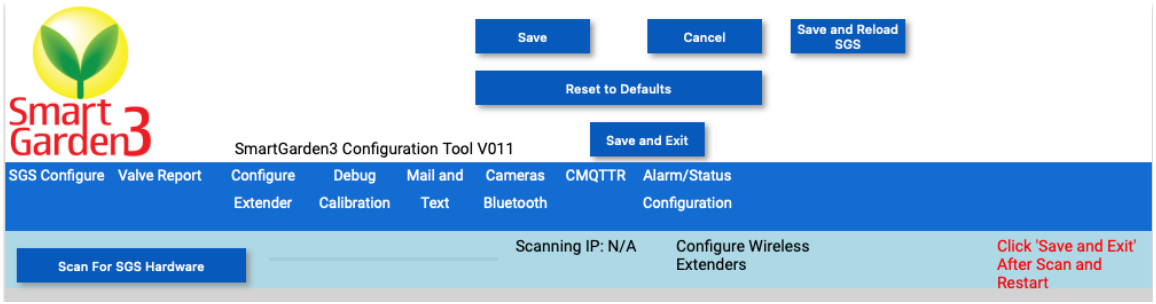
```
http://127.0.0.1:8050/
```

On another computer type:

```
http://xxx.xxx.xxx.xxx:8050/
```

Where “xxx.xxx.xxx.xxx” is the IP address of your Raspberry Pi that you wrote down above.

You will see this screen on your browser (or one of the other pages).



Notice the blue boxes on the top with key commands such as Save, Cancel and Restore to Default. We should understand what they do:

Save Button – Saves your configuration into SGS.JSON and SGSConfiguration.JSON. Does not reload the SG3 program with your values.

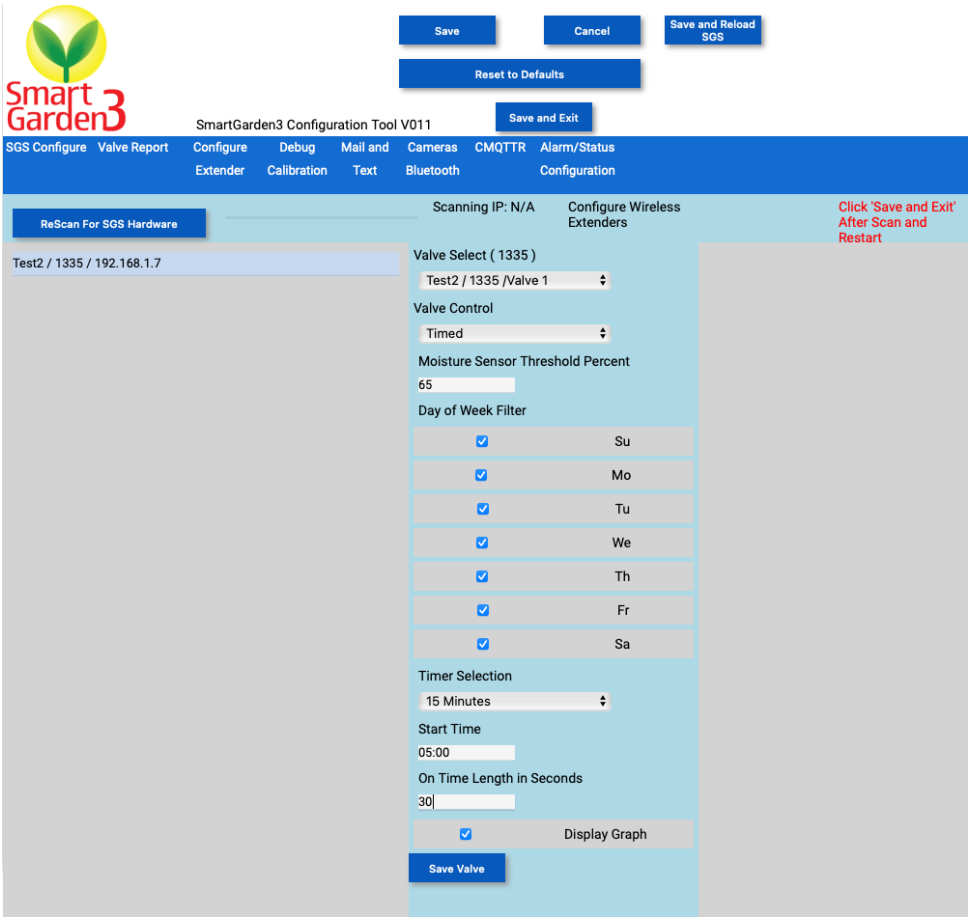
Cancel Button – Clears all your changes and reloads the original JSON files.

Reset to Defaults – Resets the configuration to the factory default values

Save and Exit – Saves your configuration into SGS.JSON and SGSConfiguration.JSON and exits the program. Does not reload the SG3 program with your values.

Save and Reload SG3 – Saves the JSON files and dynamically reloads your SG3 program with the new values. This will take about 20 or 30 seconds for the SG3 program to recognize your request and carefully reinitialize the SG3 program with your new JSON files.

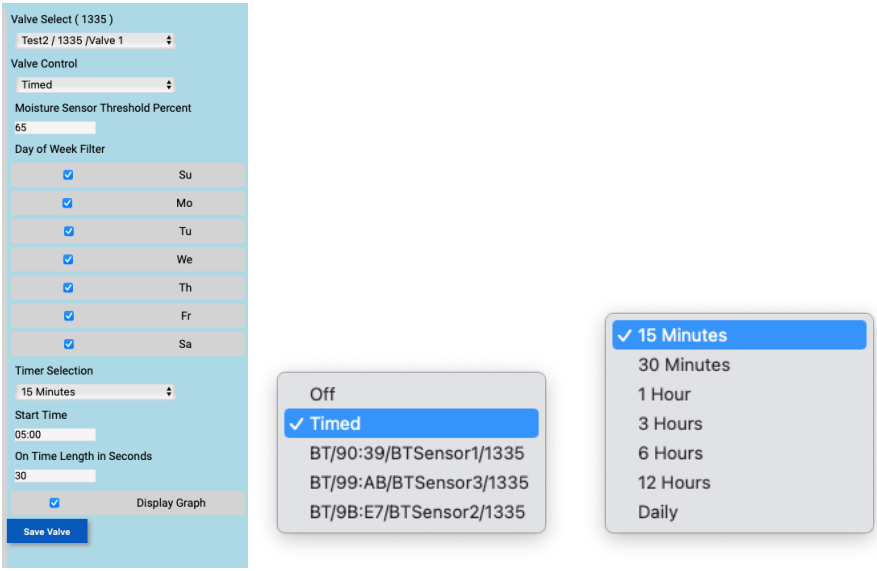
SG3 Valve Configure Window



- **Scan For SG3 Hardware** - (Rescan for SG3 Hardware) – Clicking this button will scan (or rescan) all your local IP numbers on your local network looking for SG3 Wireless Units. It will take about 10 minutes to scan your entire network. You can watch the progress in the Scanning IP and Found Wireless Extenders fields. Rescanning your network does not clear the valve and sensor programming on your current units.

- **The Wireless Extender Buttons** - When you select a Wireless Unit by clicking on the Wireless Extender name, you can then add and edit all the valves or pumps on your Wireless unit. The fields of the name are: Name of Unit / Internal ID of Unit / IP Address of Unit. When you click on the unit, a valve configuration screen comes up.

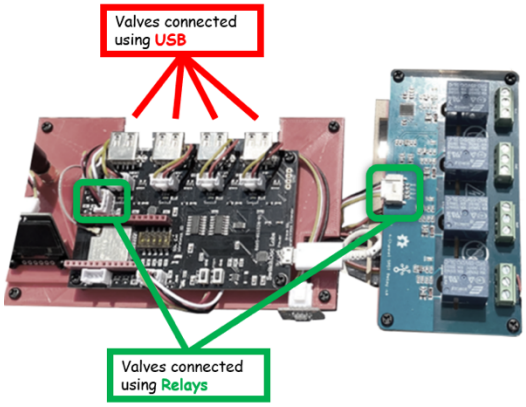
Valve Configuration Menu

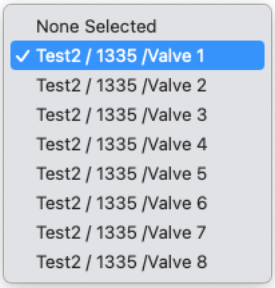


- **Valve Select** – Select Valve 1 -8 on the current Wireless Unit
- **Valve Control** – There are two choices. You can select a "timed" valve (turns on and off at a specified length and a specified timing interval) or have the valve controlled by any of the Bluetooth soil moisture sensors in the entire system. The fields of Timer Selection and Start Time cannot be changed unless you have selected a "timed" value.
- **Moisture Sensor Threshold Percent** – This is the threshold for the pump or valve to be turned on by the selected sensor. ***It defaults to 65 Percent.*** By the way, if you set this above 100%, then the valve will be always on. This is one of the ways to turn a valve or fan on or off permanently.
- **Day of Week Filter** – This allows you to turn your valve control off and on for the days of the week. Great for Orchids!
- **Timer Selection** – This dropdown menu selects the repetition frequency for the timed valve. The first fire of your timer will be after the Start Time and after the current time aligned on the appropriate clock time. For example, selecting a 15-minute interval at 13:12 (use 24 hour time in the field) will start your first on time at 13:15, the next at 13:30, and so on.
- **Start Time** – This is the time to start calculating the next "timed event" or when to do the on and off event in the case of the Daily event.
- **On Time Length in Seconds** – When a timed event happens, this is where you specify how long to turn the valve or pump on in seconds. This is a second way to make a valve or pump turn on permanently. For example, if you select a 15-minute timed event and set them On Time Length In Seconds to something longer than 15 minutes (over 900), then the Valve will never turn off. Very handy!
- **Display Graph** – If you click this, then the Moisture Sensor selected (if you are connected to a moisture sensor event) and the Valve On/Off graph will be displayed on the Dash App, described below.

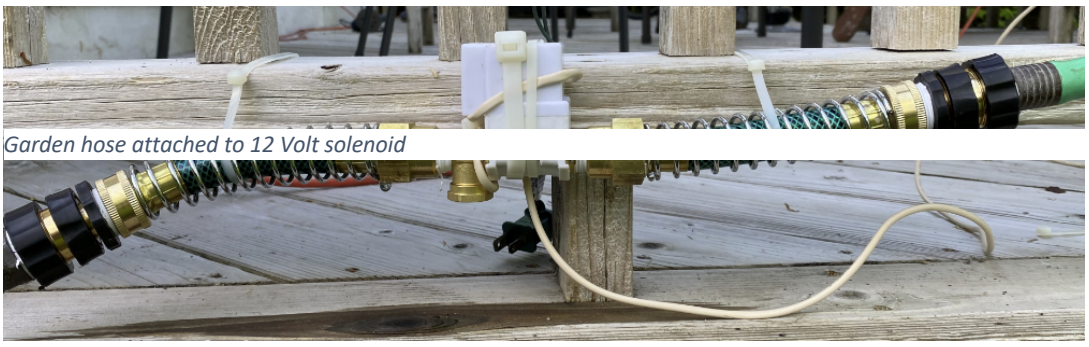
- **Save Valve** – This saves the configuration of the valve. Warning: Selecting another valve will clear your changes if you do not hit "Save Valve".

Note 1: There are eight valve ports in the standard configuration: The setting for valves provide for Valves 1 – 4 to support pumps using the USB connectors on the extender board; such as the pump supplied in the SmartGarden3 kit. Valves 5 – 8 support pumps and solenoids attached to the four blue relays. Solenoids are valves operating at higher, and potentially significantly higher, voltage. Relays separate these higher voltage circuits from the delicate 5-volt powering the Raspberry Pi and extender.





Note 2: Though the four blue relay board is included in the standard kit, we do not supply pumps and solenoids to be connected to this circuitry as this is beyond the scope of most users. Your

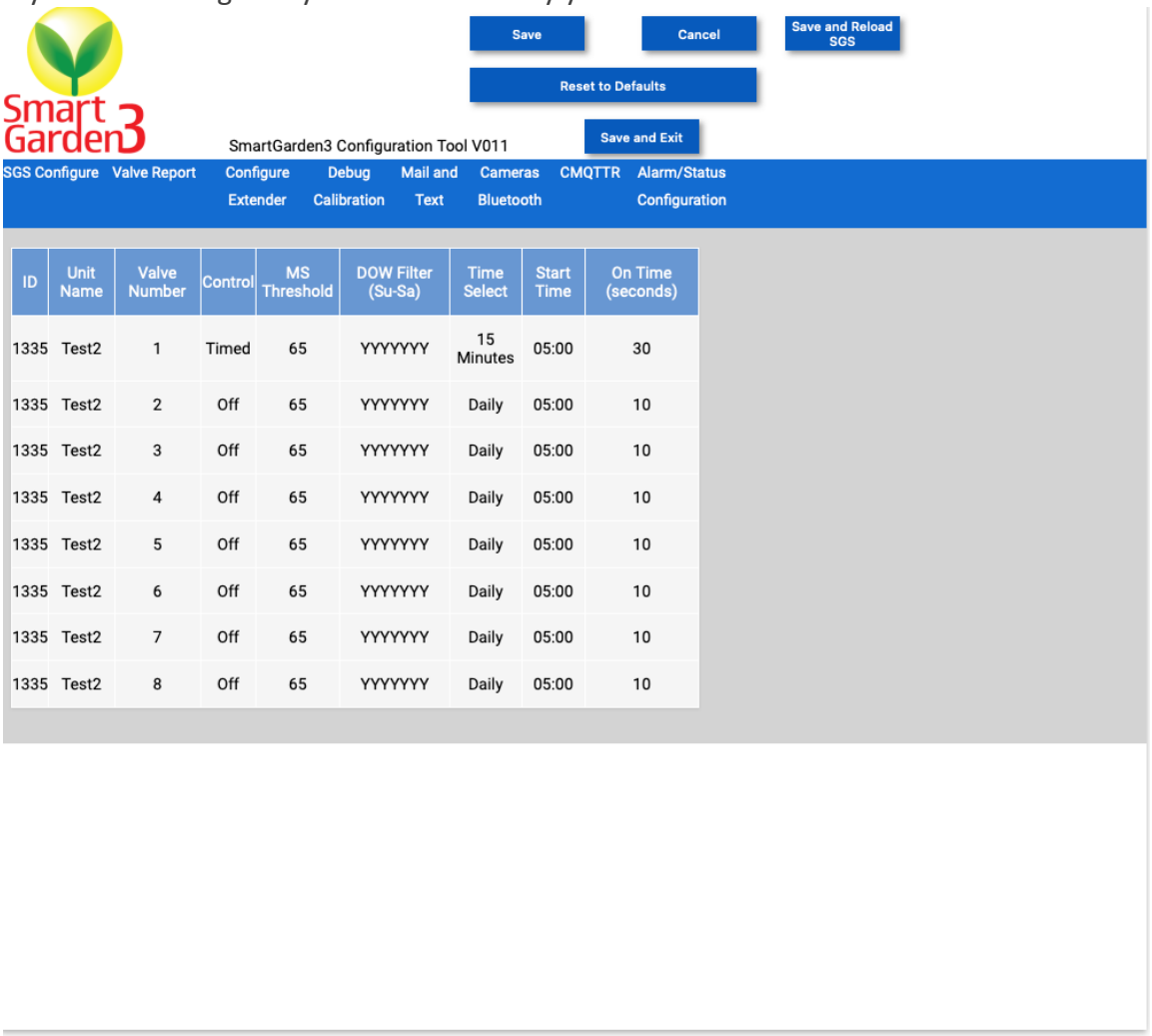


Garden hose attached to 12 Volt solenoid

SmartGarden3 is capable of watering your entire yard, or a golf course if you are inclined. Solenoids are simple switches that can open a hose or pipe to allow water to flow. You would need 12-volt solenoids for your yard, but you could attach a 250-volt solenoid without harming your extender.

SG3 Valve Report

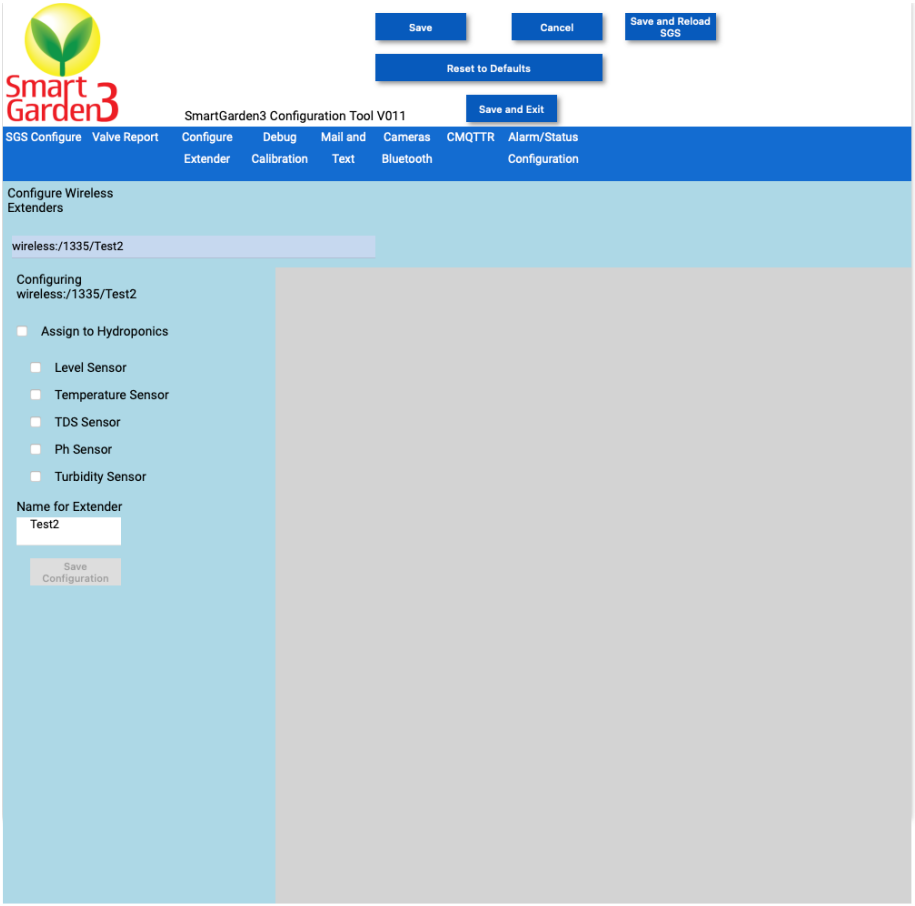
The Valve Report tab shows the current configuration of all your valves. This is a great place to check if you have configured your valves the way you intended.



ID	Unit Name	Valve Number	Control	MS Threshold	DOW Filter (Su-Sa)	Time Select	Start Time	On Time (seconds)
1335	Test2	1	Timed	65	YYYYYYY	15 Minutes	05:00	30
1335	Test2	2	Off	65	YYYYYYY	Daily	05:00	10
1335	Test2	3	Off	65	YYYYYYY	Daily	05:00	10
1335	Test2	4	Off	65	YYYYYYY	Daily	05:00	10
1335	Test2	5	Off	65	YYYYYYY	Daily	05:00	10
1335	Test2	6	Off	65	YYYYYYY	Daily	05:00	10
1335	Test2	7	Off	65	YYYYYYY	Daily	05:00	10
1335	Test2	8	Off	65	YYYYYYY	Daily	05:00	10

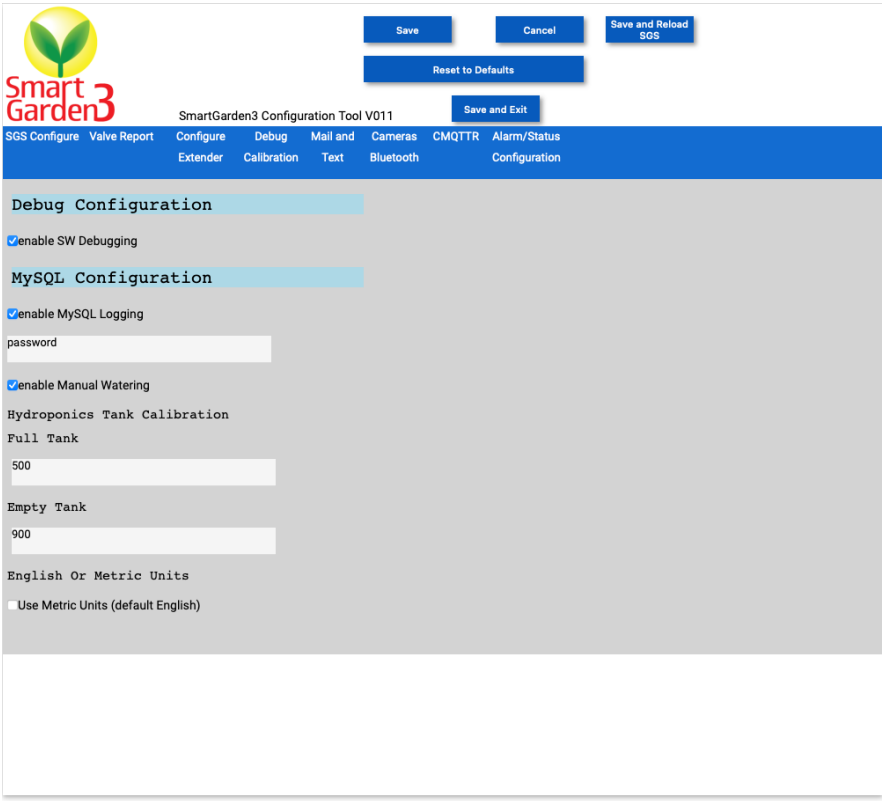
Configure Extender Tab

The Configure Extender tab is used to configure your Wireless Extender. Click on the name of the extender to bring up the configuration column.



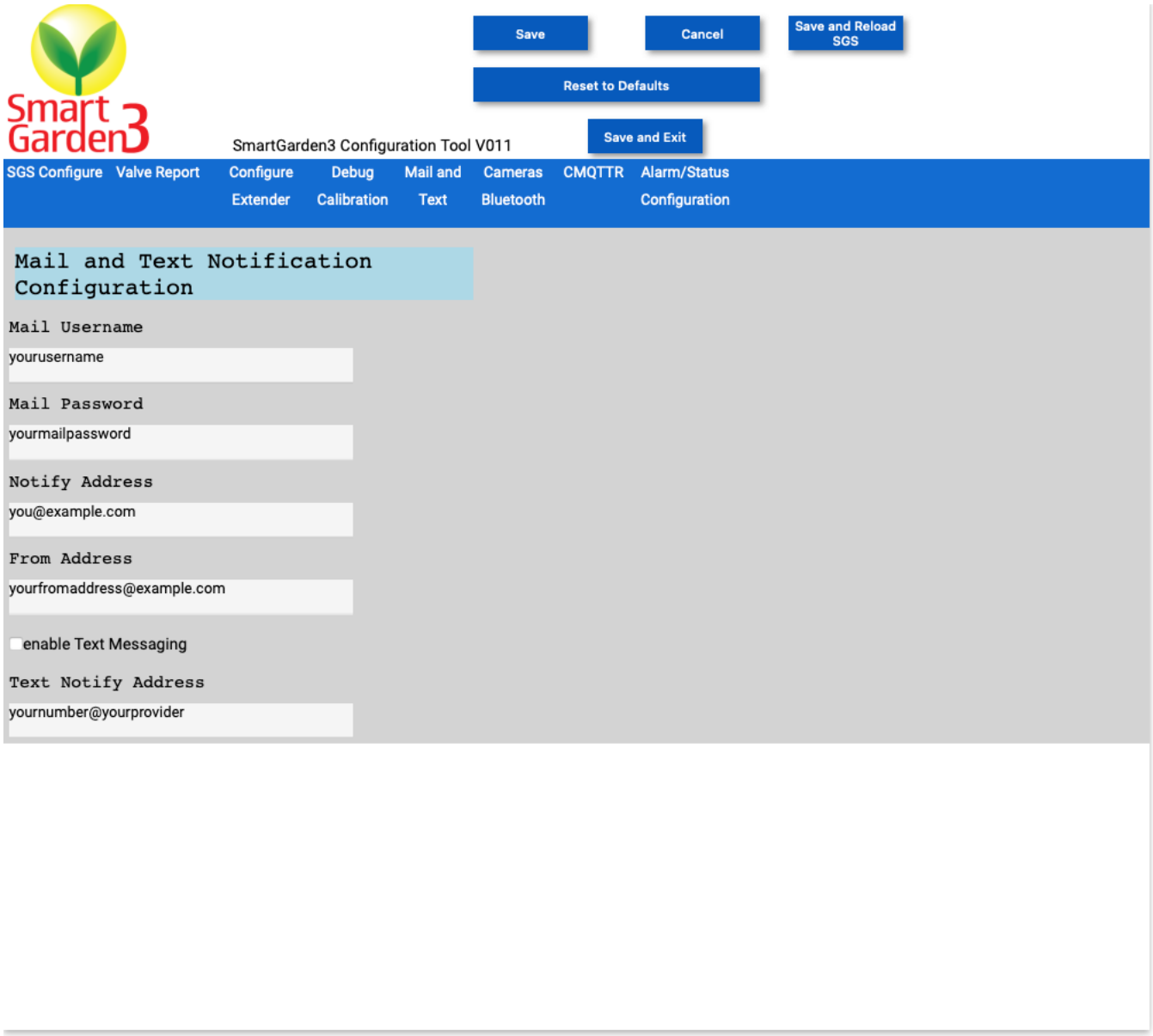
- **Assign to Hydroponics** – Select if your unit has the SDL Hydroponics Kit
  - **Level Sensor** – Click if you have the Hydroponics Water Level Sensor
  - **Temperature Sensor** – Click if you have the Hydroponics Temperature Sensor
  - **TDS Sensor** – Click if you have the Hydroponics Total Dissolved Solids Sensor
  - **Ph Sensor** – Click if you have the Hydroponics Ph Sensor
  - **Turbidity Sensor** – Click if you have the Hydroponics Water Turbidity Sensor
- **Name for Extender** – Type in the name for your Extender
- **Save Configure** – Click to save your wireless extender configuration

SG3 DM (Debug / MySQL / Manual Watering / Metric / English



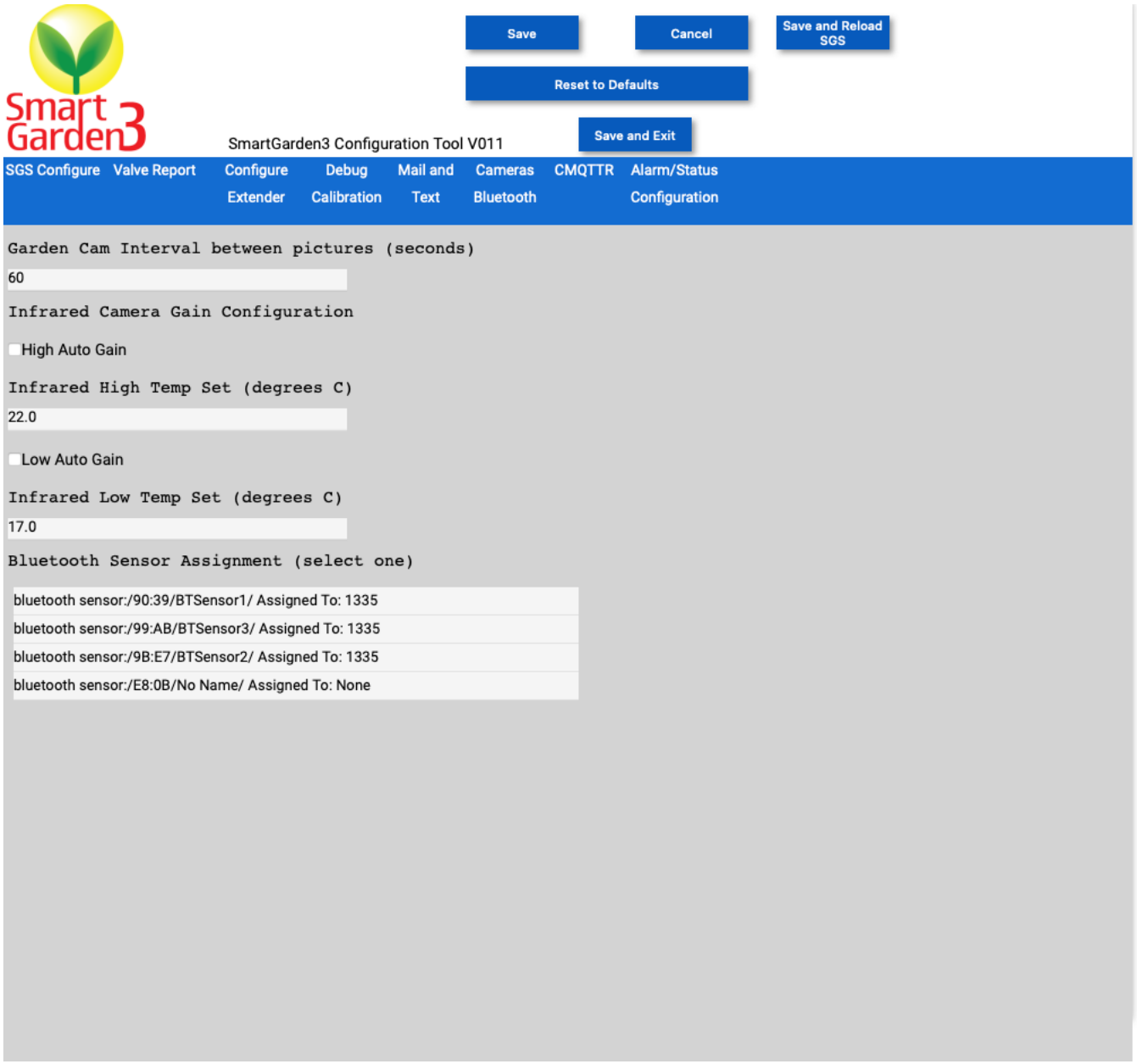
- **Debug Configuration** - enable SW Debugging – Checking this box enables software debugging on the SG3 program and produces substantially more output to the console terminal window where you are running the SG3 program.
- **MySQL Configuration** – Enable MySQL Logging – This tells the SG3 program to log lots of information to the Raspberry Pi MySQL database for the SmartGarden3. Very useful and needed to produce graphs and charts in the Dash App.
- **MySQL Configuration** – Password – This is the password on the "root" user of your MySQL application on your Raspberry Pi. Defaults to "password" which is the default password on the MySQL on the SwitchDoc Labs SD Card.
- **Enable Manual Watering** – Future Enhancement
- **Hydroponics Tank Level Calibration** – The Full Tank and Empty Tank values are used to calibrate the Hydroponics Tank for the SDL Hydroponics System
- **Use Metric Units –** Checking this box will set the units used by the SmartGarden3 to metric (versus the default of English units).

SG3 Mail and Text Notifications



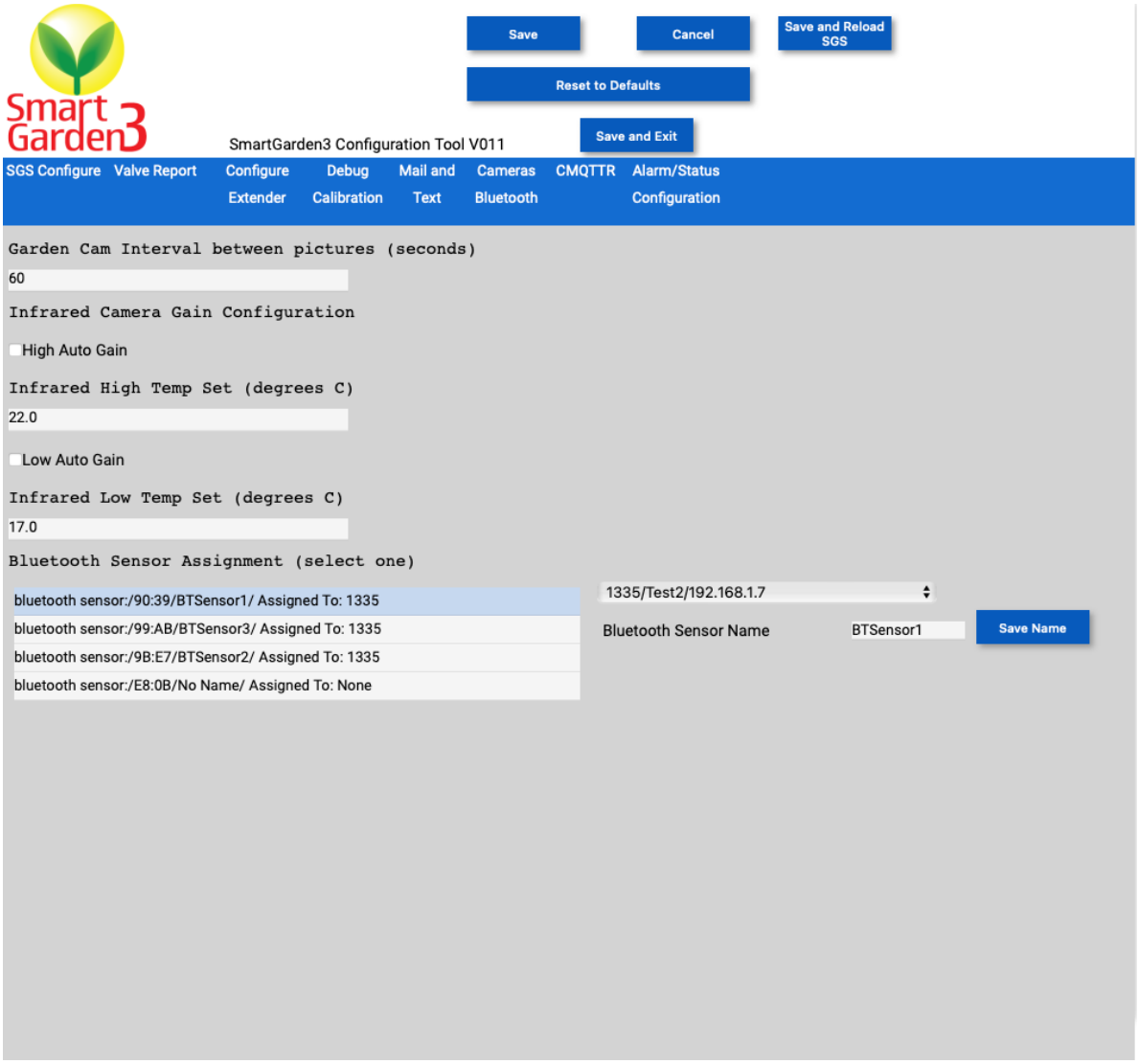
- **Mail Configuration** – Mail Username – The username for your email account – for Gmail (a good choice!) would be myname@gmail.com.
- **Mail Configuration** – Mail Password – The password for your email account.
- **Mail Configuration** – From Address – Enter the email address you would like your notifications and email to come from. Note: For some mail servers, this will need to match your account information.
- **Text Configuration** – Text Notification Address – Text updates from SmartGarden3 will be sent to this email address, customized for your mobile text provider (see this website: https://www.dialmycalls.com/blog/send-text-messages-email-address).

Cameras and Bluetooth Tab



- **Garden Cam Interval between pictures (Seconds)**– How often to take a picture of your garden with the optional Raspberry Pi Camera.
- **Infrared Camera Gain Configuration (Hydroponics) –** Only used for the SDL Hydroponics Kit
  - **High Auto Gain –** Sets the Infrared High Temperature to the maximum read by the sensor
  - **Manual Infrared High Temp Set (Degrees C) –** Fix the "High" temp for the Infrared Sensor
  - **Low Auto Gain –** Sets the Infrared High Temperature to the minimum read by the sensor
  - **Manual Infrared Low Temp Set (Degrees C) –** Fix the "Low" temp for the Infrared Sensor

- **Bluetooth Sensor Assignment** – Used to assign the Bluetooth Soil Sensors to a Wireless Extender. Click on sensor to be assigned.

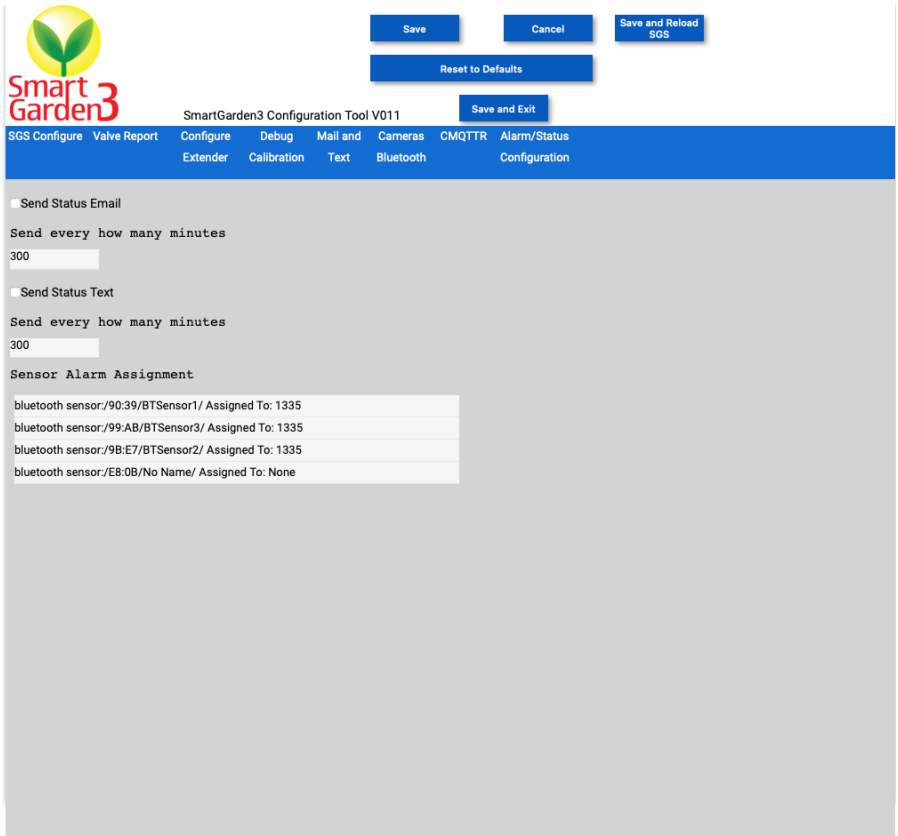


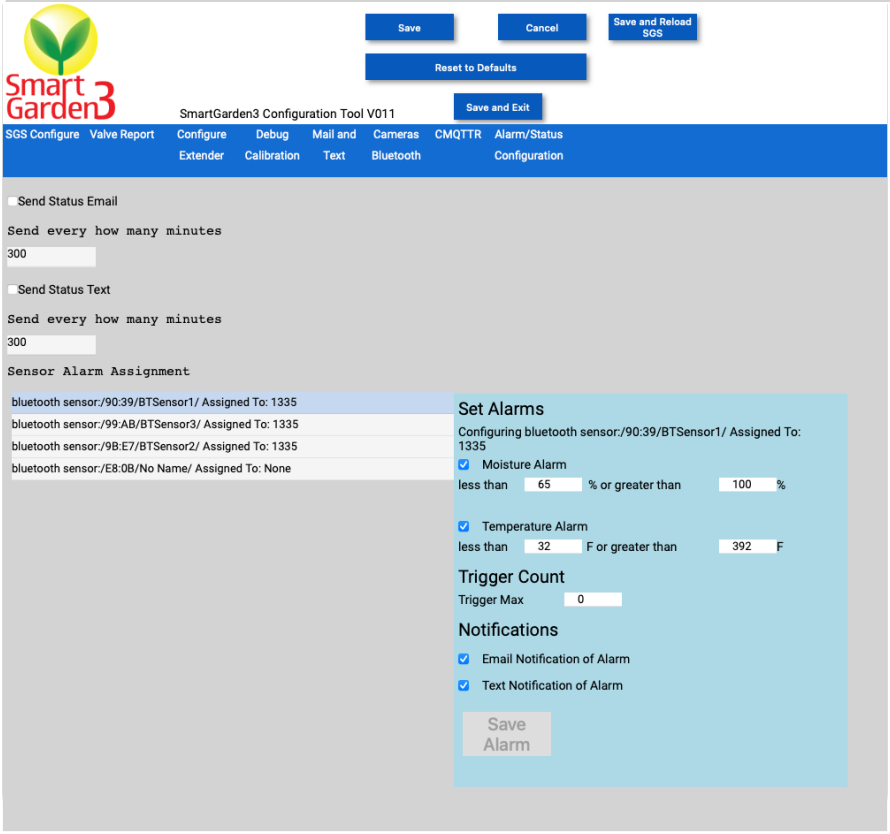
Select an Extender from the drop down menu. This is where you can name your Bluetooth Sensor ("Flower Pot" for example)

SG3 Camera / Rest / MQTT Configuration (CMQTTR) Tab

- **Night Camera Enable –** For future enhancement.
- **REST Interface – REST Enable** – Enables the REST interface to the SmartGarden3 on the Raspberry Pi – Future enhancement.
- **MQTT Configuration** is for the use of an MQTT interface FROM the SmartGarden3 to another computer of your choosing. For future enhancement.
- **MQTT Configuration – MQTT Enable** – For future enhancement.
- **MQTT Configuration – MQTT Server URL** – For future enhancement.
- **MQTT Configuration – MQTT Server Port Number** – For future enhancement.
- **MQTT Configuration – How Often MQTT Send in Seconds** – For future enhancement.

Alarm Status Configuration Tab





- **Send Status Email –** For future enhancement.
- **Send every how many minutes –** How often to send status email
- **Send Status Text –** For future enhancement.
- **Send every how many minutes –** How often to send status text
- **Sensor Alarm Assignment –** Assign alarms to specific Bluetooth Sensors – Click on sensor to bring up the Set Alarms Menu
- **Set Alarms Menu –** Alarm configuration for selected Bluetooth Sensor
- **Moisture Alarm – Enable Moisture alarm for selected Bluetooth Sensor**
    - **Less Than Box – Alarm if the moisture value is less than this number**
    - **Or Greater Than Box – Alarm is the moisture value is more than this number**
- **Temperature Alarm – Enable Temperature alarm for selected Bluetooth Sensor**
    - **Less Than Box – Alarm if the Temperature value is less than this number**
    - **Or Greater Than Box – Alarm is the moisture value is more than this number**
- **Trigger Count** – Maximum number of times this alarm can be triggered. 0 means no limit.
- **Email Notification of Alarm** – Send an email when this alarm is triggered
- **Text Notification of Alarm** – Send a text when this alarm is triggered

Note that alarms are cleared on startup and will be reset if the alarm has not been triggered for over 1 day. Alarm information is logged to the System Log.

Dash Application Screens

The SmartGarden3 uses Dash to display live information about what is going on in the SmartGarden3 and in your Garden. Dash is a Python framework for building web applications. It built on top of Flask, Plotly.js, React and React Js. It enables you to build dashboards using pure Python. Dash is open source, and its apps run on standard Web browsers.

Step 9: You will open a second Terminal Window in your Raspberry Pi to start Dash by typing the following:

```
cd SDL_Pi_SmartGarden3
cd dash_app
sudo python3 index.py
```

Step: 10 Next, you will open your favorite browser to see live streaming data!

On your Raspberry Pi:

http://127.0.0.1:8010/

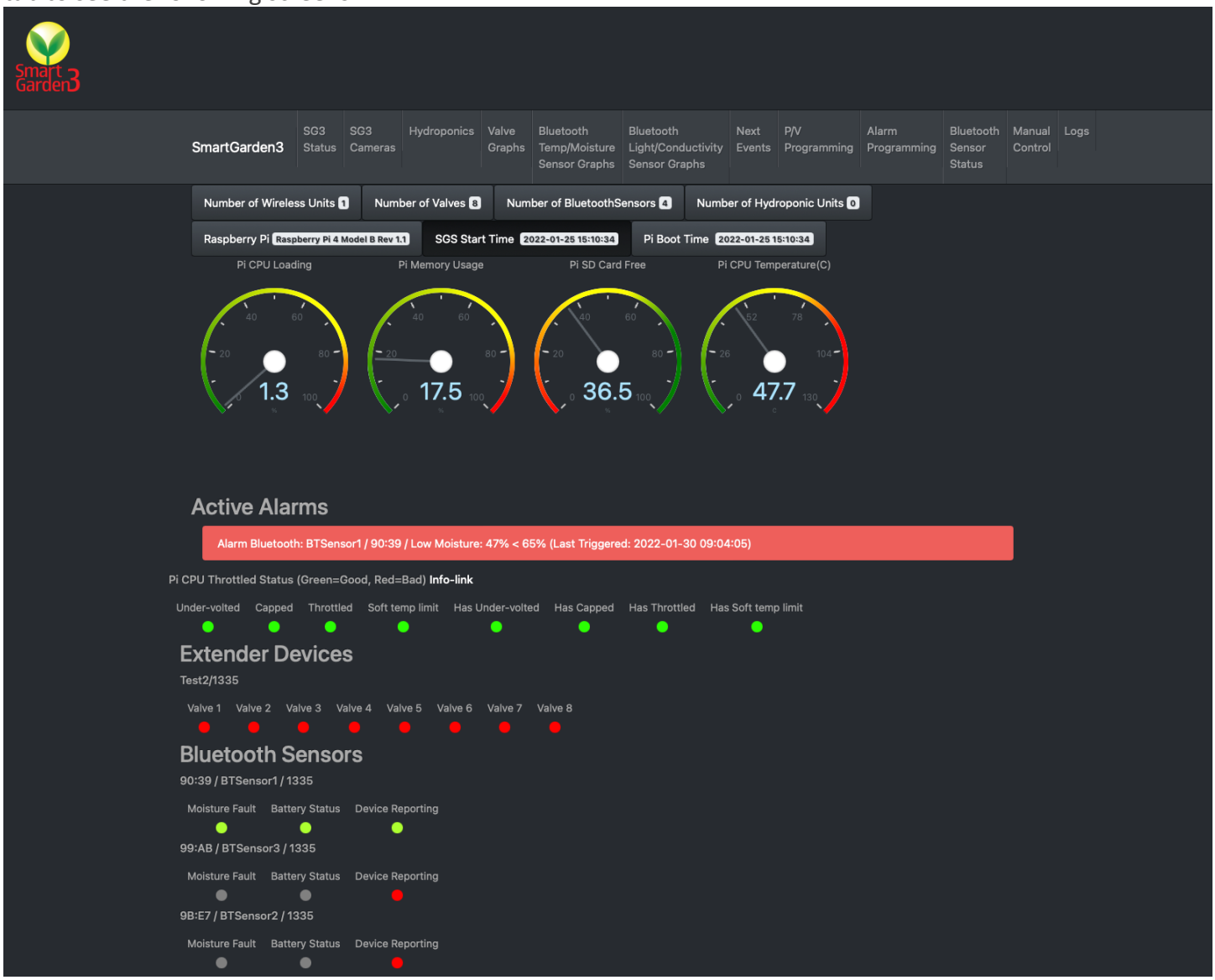
On another computer type:

http://xxx.xxx.xxx.xxx:8010/

Where “xxx.xxx.xxx.xxx” is the IP address of your Raspberry Pi that you wrote down above.

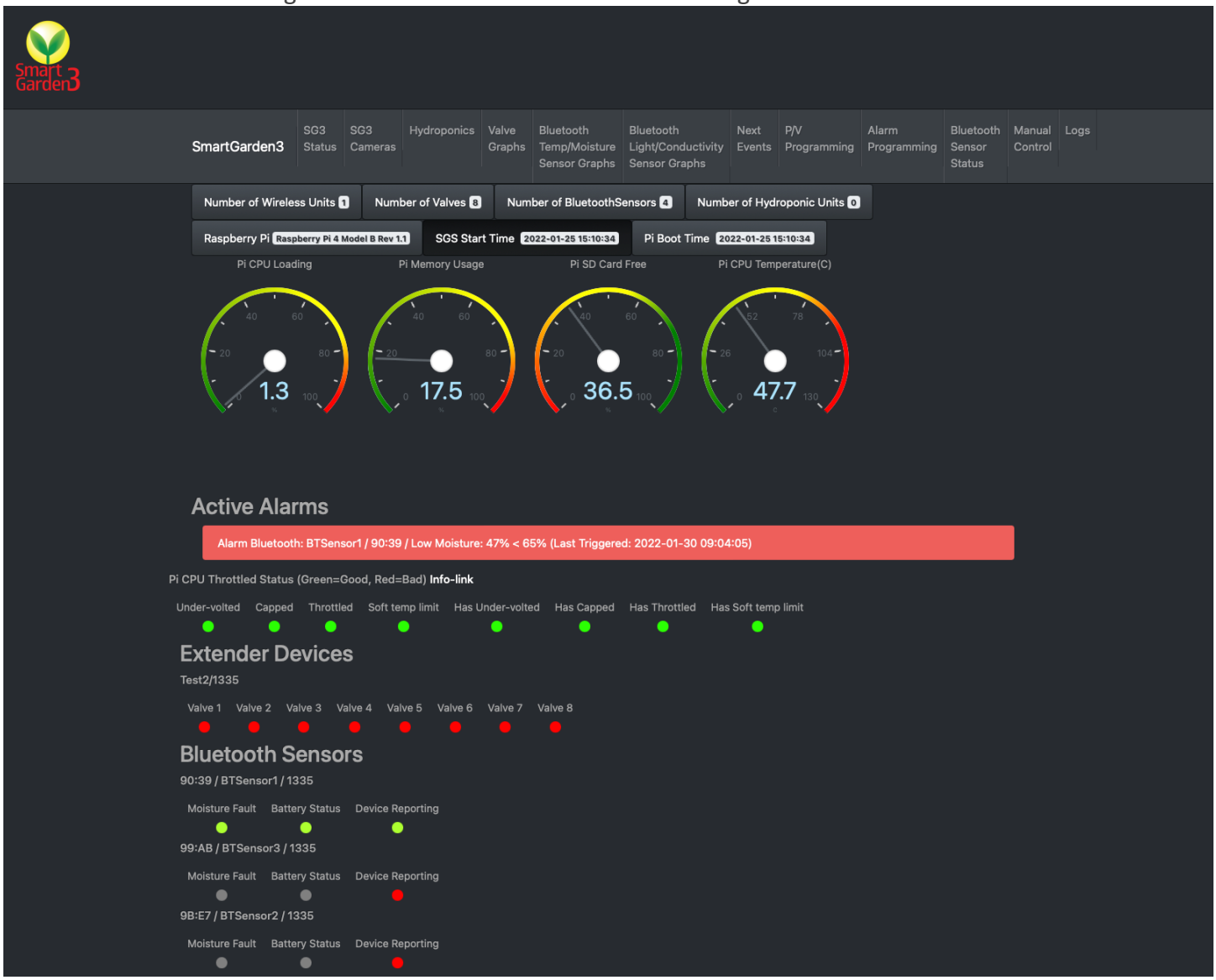
Dash Application Screens

After you start the dash app, you will have a screen with several tabs in your browser. Click on each respective tab to see the following screens.



SG3 Status Tab

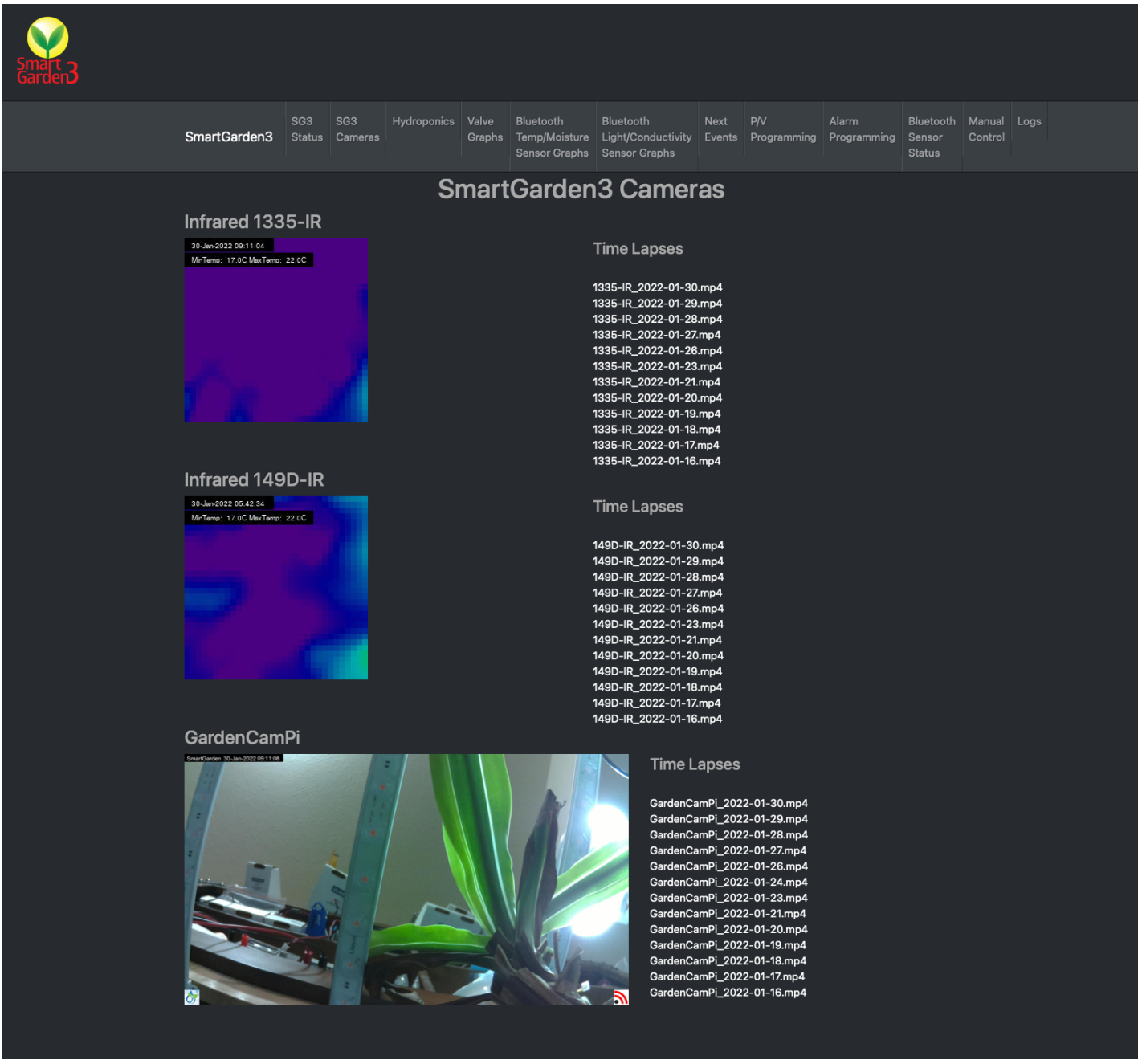
The SG3 Status tab has a set of text boxes giving information about the current state of the Raspberry Pi and the SmartGarden3 running. It is broken down into five different regions:



- **Text Blocks** Containing information about the SG3 System and the type of Raspberry Pi running including time of the last boot of the Raspberry Pi.
- **Raspberry Pi Dashboard Gauges** showing the dynamic state of the Raspberry Pi.
- The **Alarm Reporting** Region – This shows active alarms.
- **Current Valve State** for all the GSG Wireless Extenders
- **Bluetooth Moisture Sensors** Status.

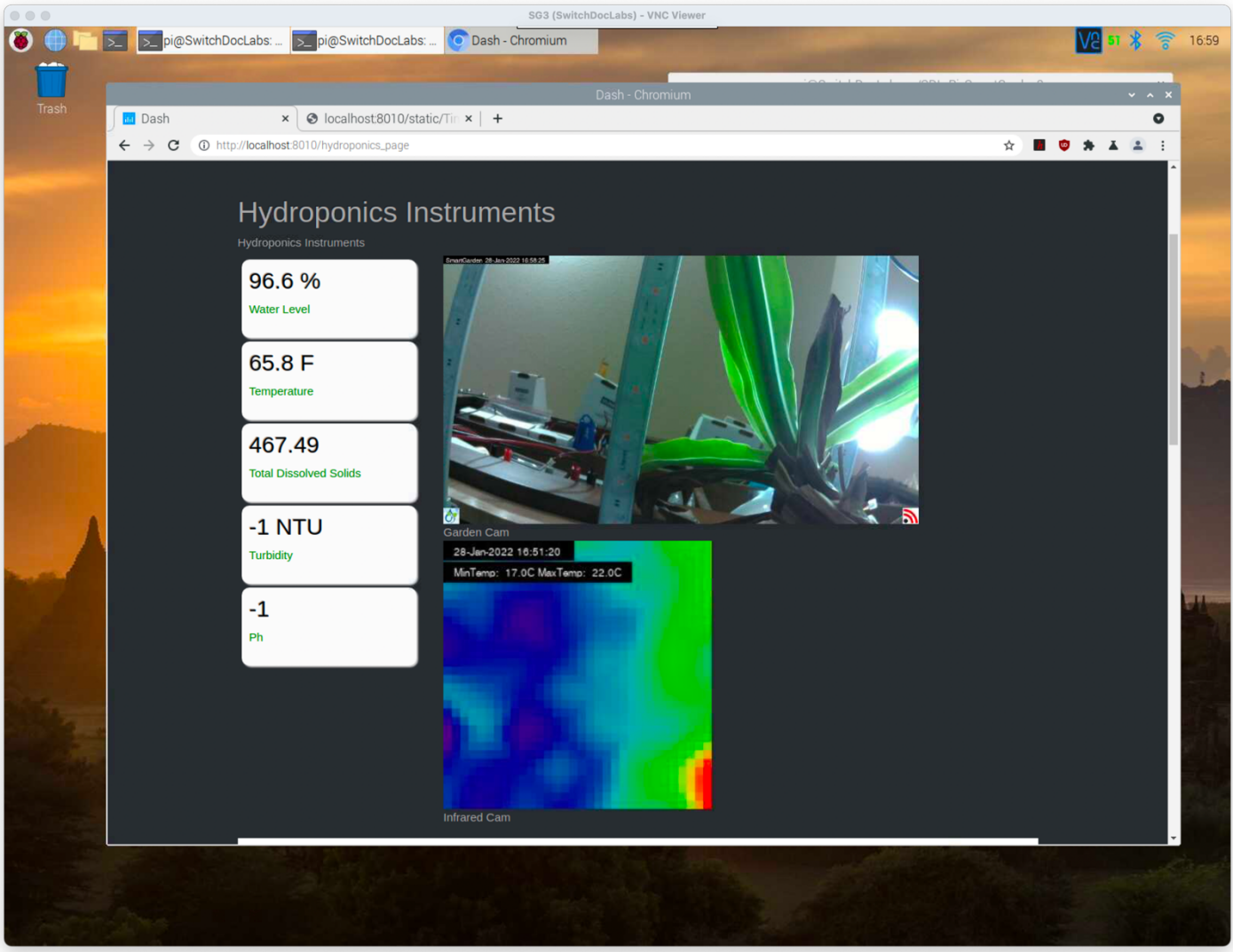
SG3 Cameras Tab

This shows the Infrared Sensors Cameras (part of the SDL Hydroponics) and the optional Raspberry Pi Camera attached to the Raspberry Pi where you are running SG3. SG3 also generates daily timelapse movies allowing you to see the change in your garden!



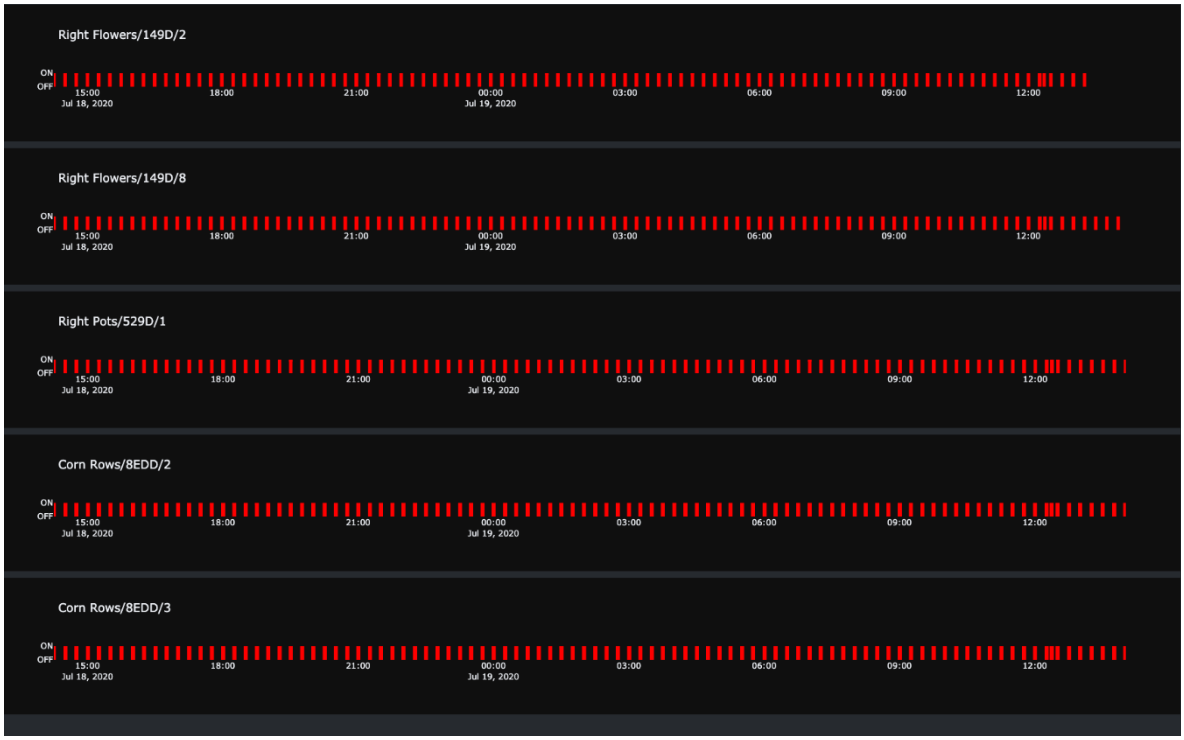
Hydroponics Tab

This tab shows when the status of a connected SDL Hydroponics kit (Optional) and also the graphs of the sensors.



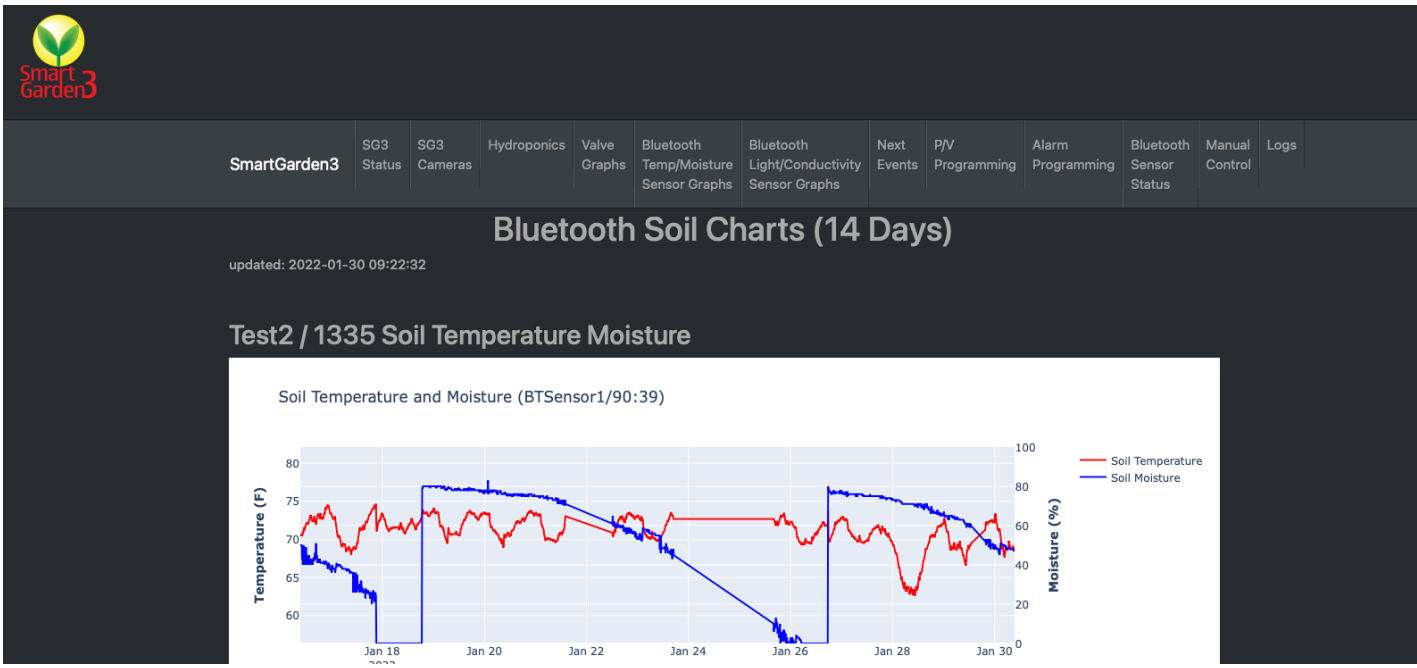
Valve Graphs Tab

This set of graphs shows when the selected valves (are turning on and off again, selected on the SG3Configure Valve Screen).



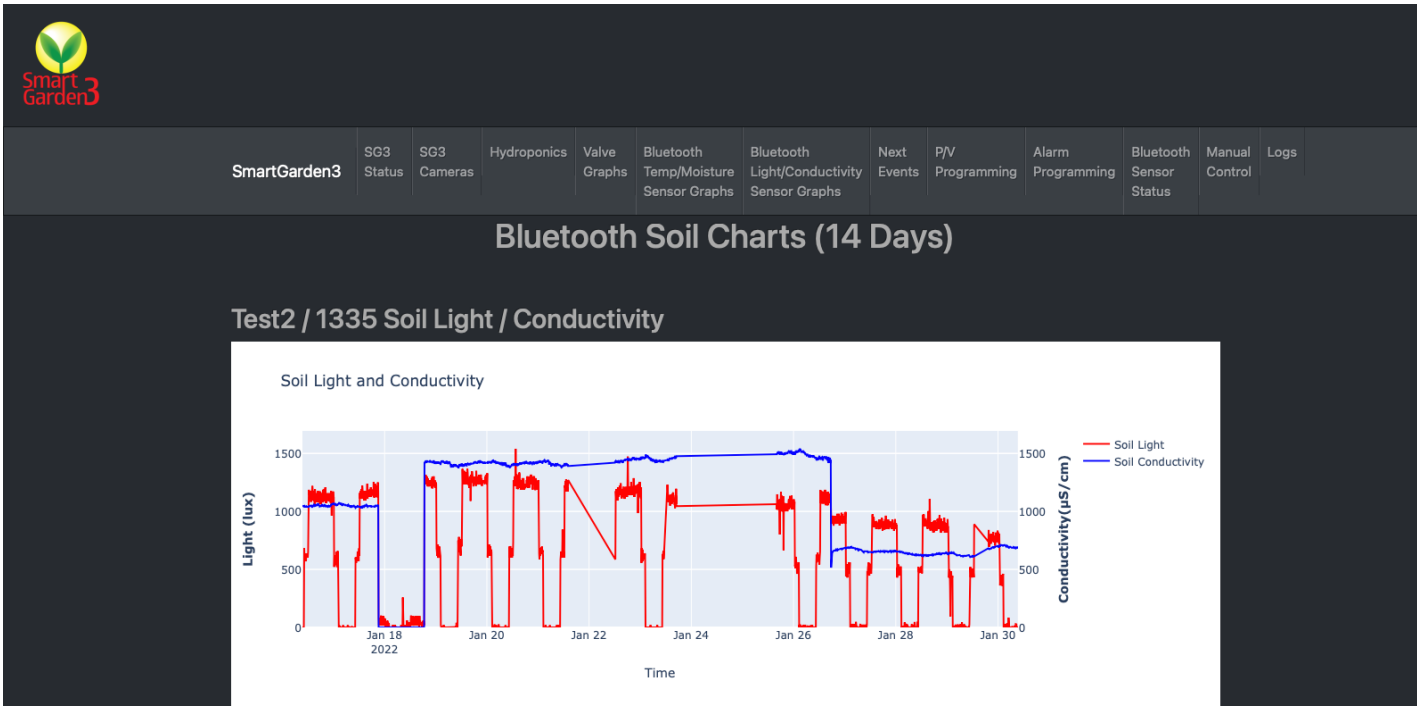
Bluetooth Temperature and Moisture Sensor Graphs Tab

Temperature and Moisture Charts of connected Bluetooth Soil Sensors are show on this tab.



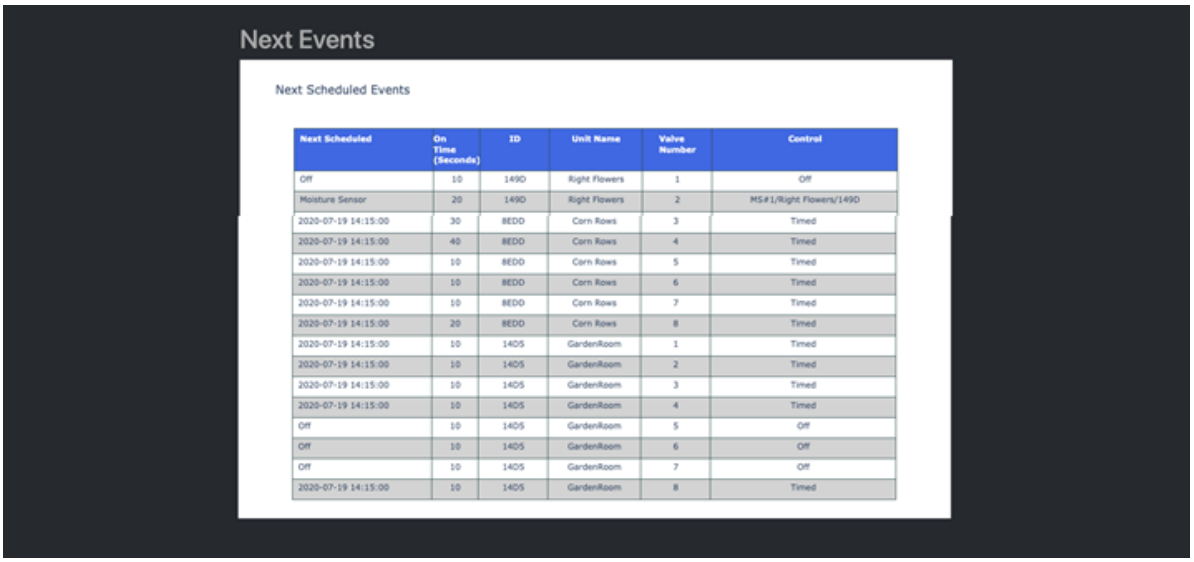
Bluetooth Light and Conductivity Sensor Graphs Tab

Light and Soil Conductivity Charts of connected Bluetooth Soil Sensors are show on this tab.



Next Events Tab

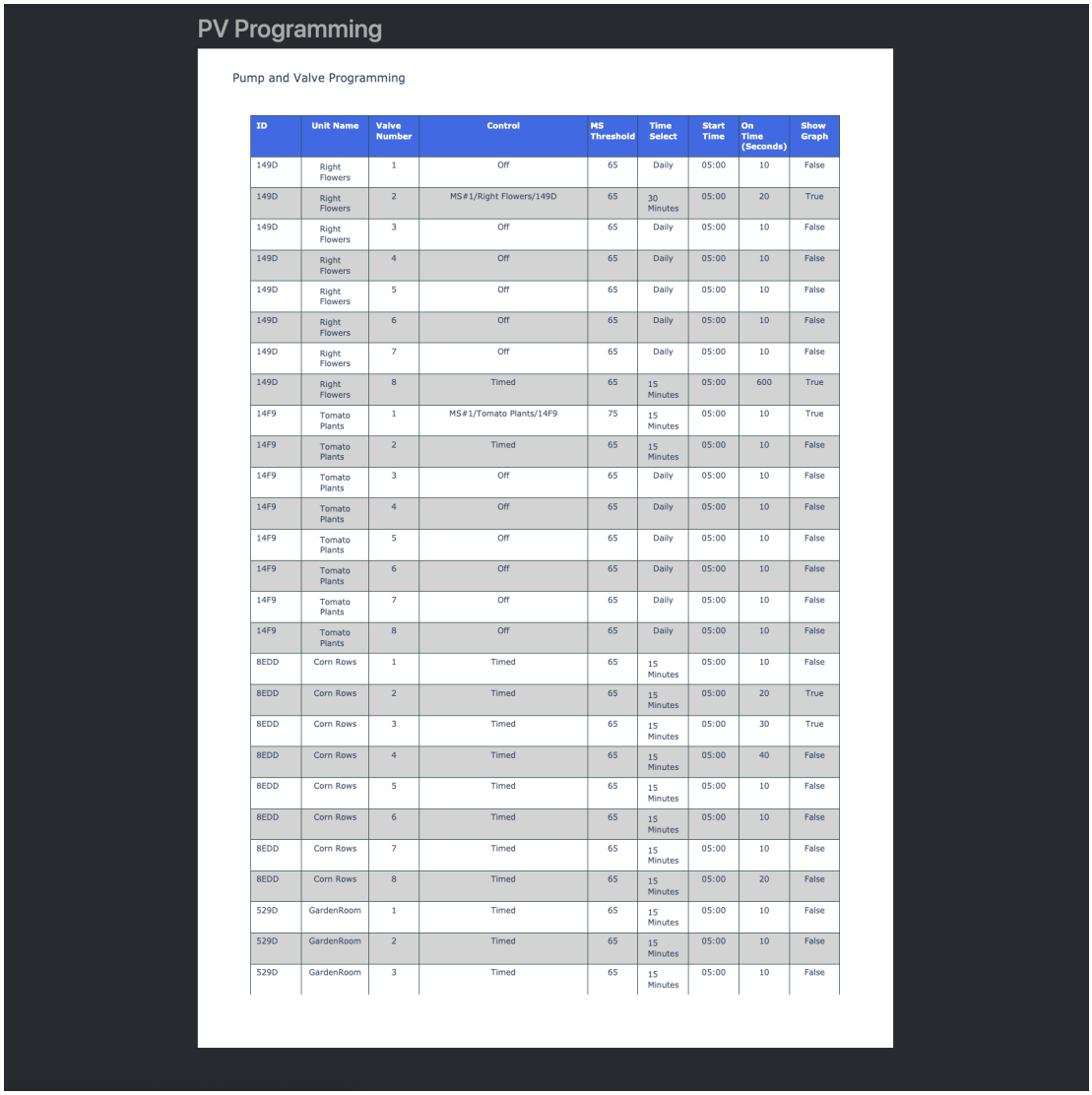
The next events tab shows a summary of all the valves in your SmartGarden3 and is specifically focusing showing the next timed events for all of your valves. This is a great way of checking that your timed events are going to happen at the time you intended.



Next Scheduled	On Time (Seconds)	ID	Unit Name	Valve Number	Control
Off	10	149D	Right Flowers	1	Off
Moisture Sensor	20	149D	Right Flowers	2	MS#1/Right Flowers/149D
2020-07-19 14:15:00	30	8EDD	Corn Rows	3	Timed
2020-07-19 14:15:00	40	8EDD	Corn Rows	4	Timed
2020-07-19 14:15:00	10	8EDD	Corn Rows	5	Timed
2020-07-19 14:15:00	10	8EDD	Corn Rows	6	Timed
2020-07-19 14:15:00	10	8EDD	Corn Rows	7	Timed
2020-07-19 14:15:00	20	8EDD	Corn Rows	8	Timed
2020-07-19 14:15:00	10	14D5	GardenRoom	1	Timed
2020-07-19 14:15:00	10	14D5	GardenRoom	2	Timed
2020-07-19 14:15:00	10	14D5	GardenRoom	3	Timed
2020-07-19 14:15:00	10	14D5	GardenRoom	4	Timed
Off	10	14D5	GardenRoom	5	Off
Off	10	14D5	GardenRoom	6	Off
Off	10	14D5	GardenRoom	7	Off
2020-07-19 14:15:00	10	14D5	GardenRoom	8	Timed

P/V Programming Tab

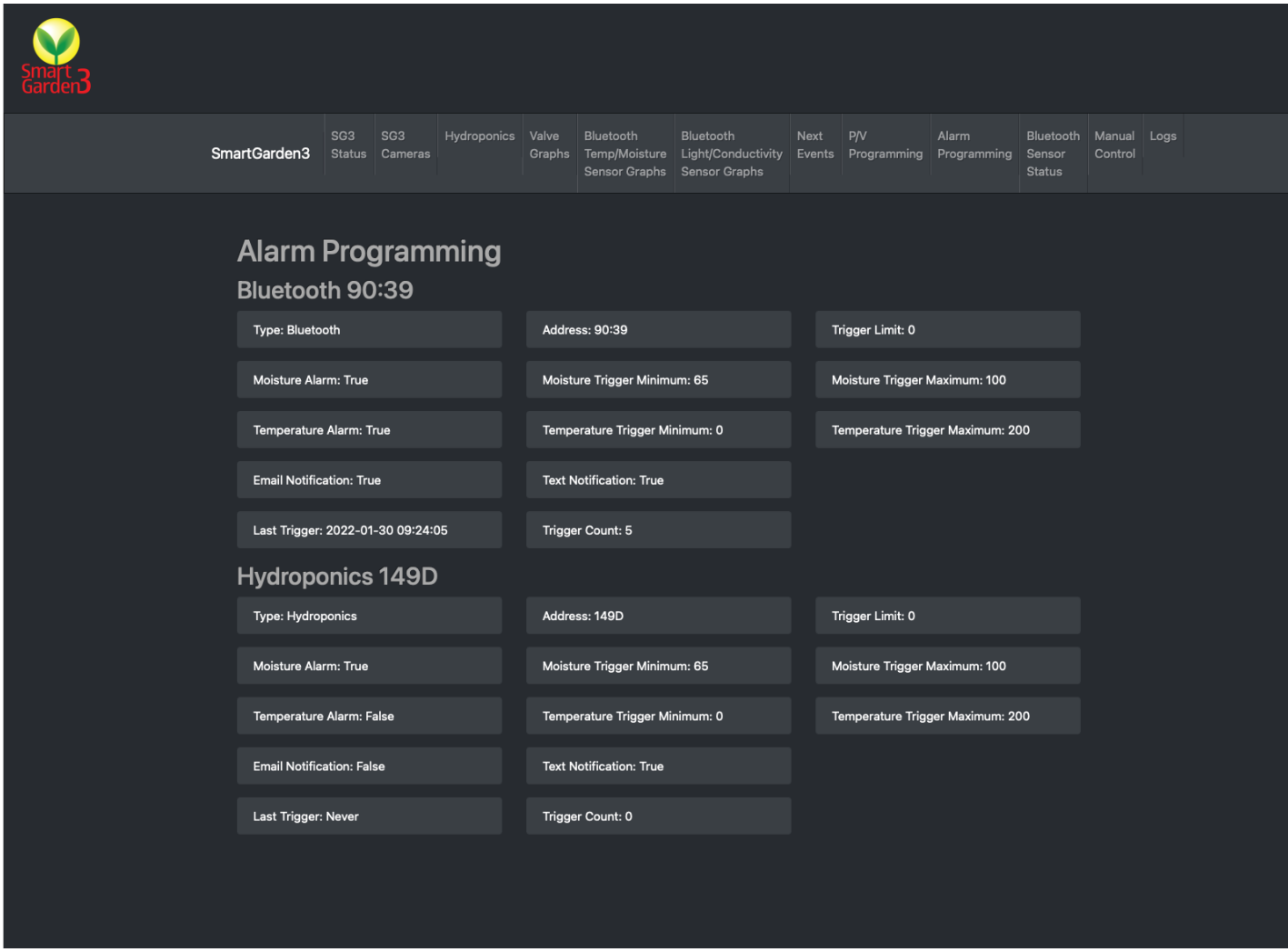
The PV Programming tab is designed to show at a glance all the programming of your valves. Again, a great way to check your programming of your SmartGarden3.



ID	Unit Name	Valve Number	Control	MS Threshold	Time Select	Start Time	On Time (Seconds)	Show Graph
149D	Right Flowers	1	Off	65	Daily	05:00	10	False
149D	Right Flowers	2	MS#1/Right Flowers/149D	65	30 Minutes	05:00	20	True
149D	Right Flowers	3	Off	65	Daily	05:00	10	False
149D	Right Flowers	4	Off	65	Daily	05:00	10	False
149D	Right Flowers	5	Off	65	Daily	05:00	10	False
149D	Right Flowers	6	Off	65	Daily	05:00	10	False
149D	Right Flowers	7	Off	65	Daily	05:00	10	False
149D	Right Flowers	8	Timed	65	15 Minutes	05:00	600	True
14F9	Tomato Plants	1	MS#1/Tomato Plants/14F9	75	15 Minutes	05:00	10	True
14F9	Tomato Plants	2	Timed	65	15 Minutes	05:00	10	False
14F9	Tomato Plants	3	Off	65	Daily	05:00	10	False
14F9	Tomato Plants	4	Off	65	Daily	05:00	10	False
14F9	Tomato Plants	5	Off	65	Daily	05:00	10	False
14F9	Tomato Plants	6	Off	65	Daily	05:00	10	False
14F9	Tomato Plants	7	Off	65	Daily	05:00	10	False
14F9	Tomato Plants	8	Off	65	Daily	05:00	10	False
8EDD	Corn Rows	1	Timed	65	15 Minutes	05:00	10	False
8EDD	Corn Rows	2	Timed	65	15 Minutes	05:00	20	True
8EDD	Corn Rows	3	Timed	65	15 Minutes	05:00	30	True
8EDD	Corn Rows	4	Timed	65	15 Minutes	05:00	40	False
8EDD	Corn Rows	5	Timed	65	15 Minutes	05:00	10	False
8EDD	Corn Rows	6	Timed	65	15 Minutes	05:00	10	False
8EDD	Corn Rows	7	Timed	65	15 Minutes	05:00	10	False
8EDD	Corn Rows	8	Timed	65	15 Minutes	05:00	20	False
529D	GardenRoom	1	Timed	65	15 Minutes	05:00	10	False
529D	GardenRoom	2	Timed	65	15 Minutes	05:00	10	False
529D	GardenRoom	3	Timed	65	15 Minutes	05:00	10	False

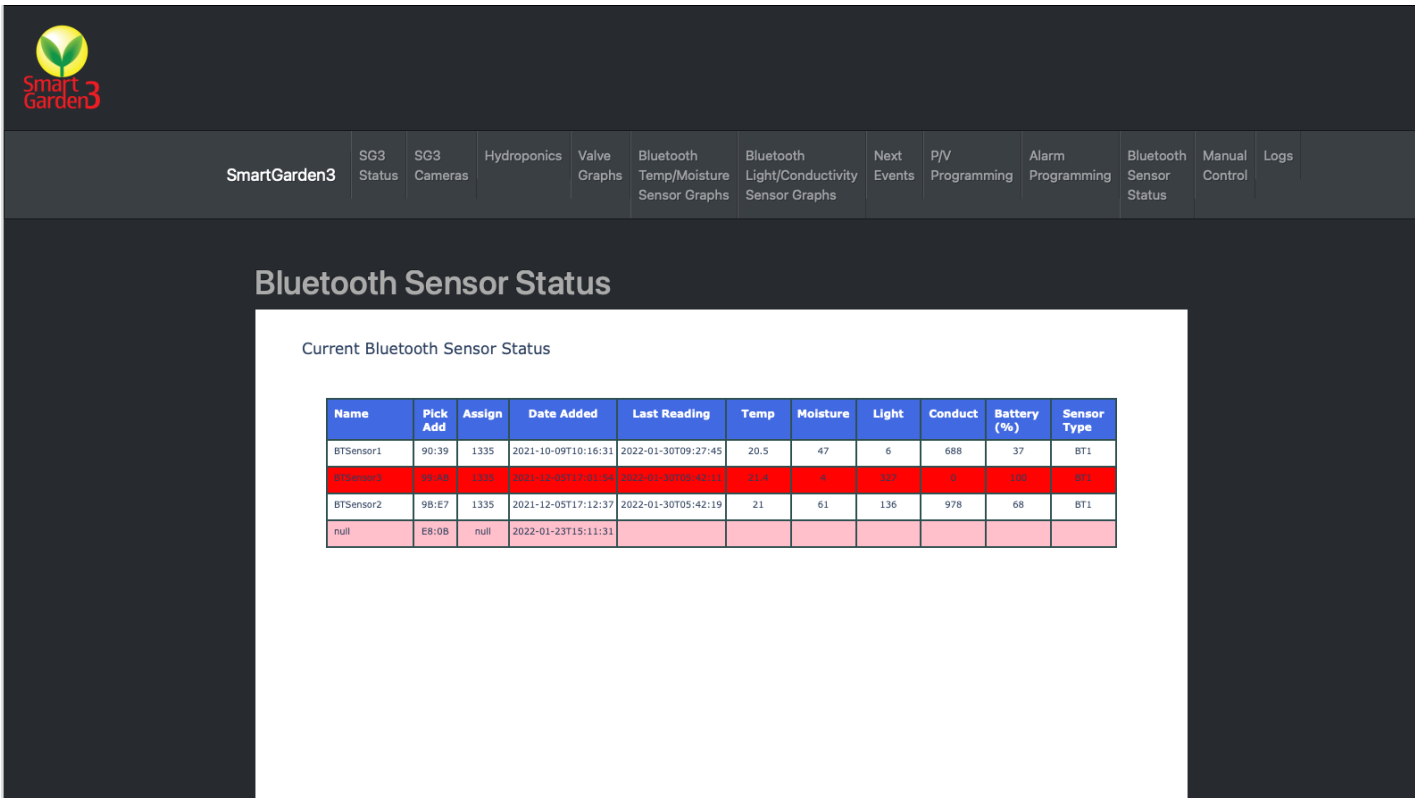
Alarm Programming Tab

The PV Programming tab is designed to show at a glance all the programming of your alarms. Again, a great way to check your programming of your SmartGarden3.



Bluetooth Sensor Status Tab

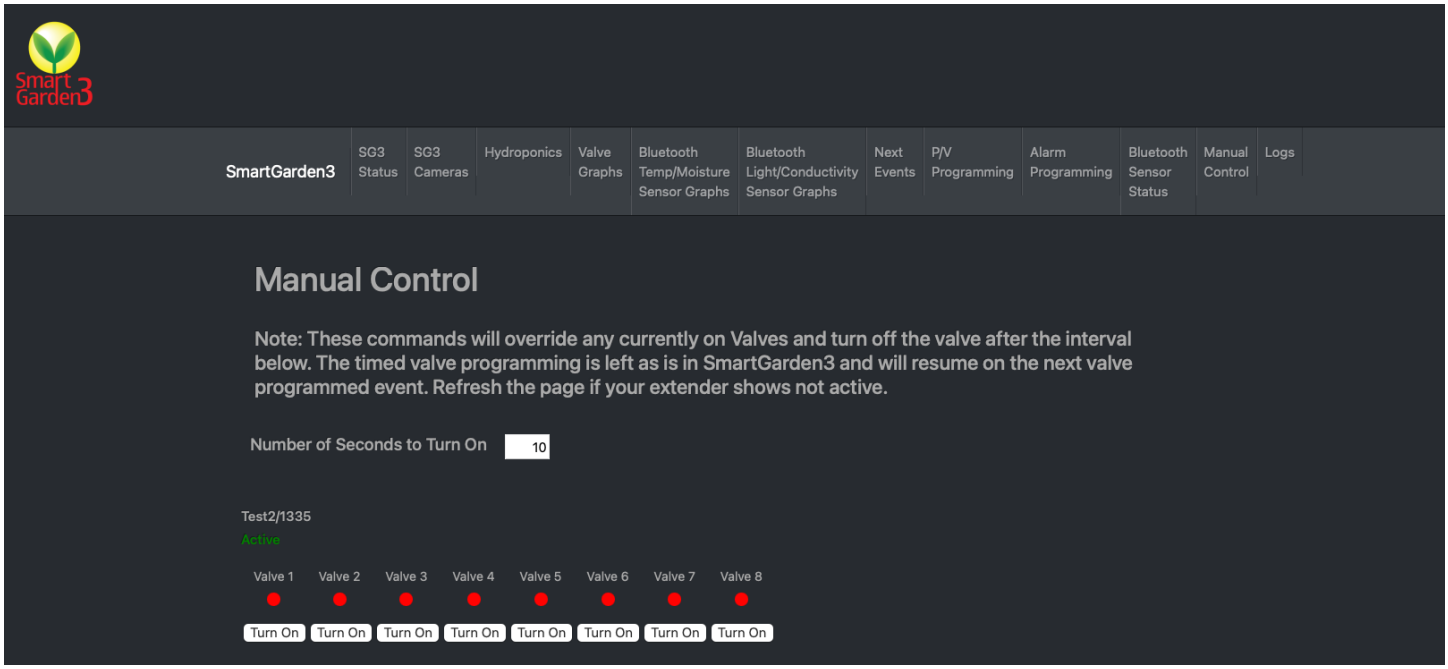
The PV Programming tab is designed to show at a glance the status of your Bluetooth Sensors. Pink shading shows you that the sensor battery is low, Red shading shows a potential moisture fault (moisture % < 5).



Name	Pick Add	Assign	Date Added	Last Reading	Temp	Moisture	Light	Conduct	Battery (%)	Sensor Type
BTSensor1	90:39	1335	2021-10-09T10:16:31	2022-01-30T09:27:45	20.5	47	6	688	37	BT1
BTSensor3	69:AB	1335	2021-12-05T17:01:54	2022-01-30T05:42:11	21.4	4	327	0	100	BT1
BTSensor2	9B:E7	1335	2021-12-05T17:12:37	2022-01-30T05:42:19	21	61	136	978	68	BT1
null	E8:0B	null	2022-01-23T15:11:31							

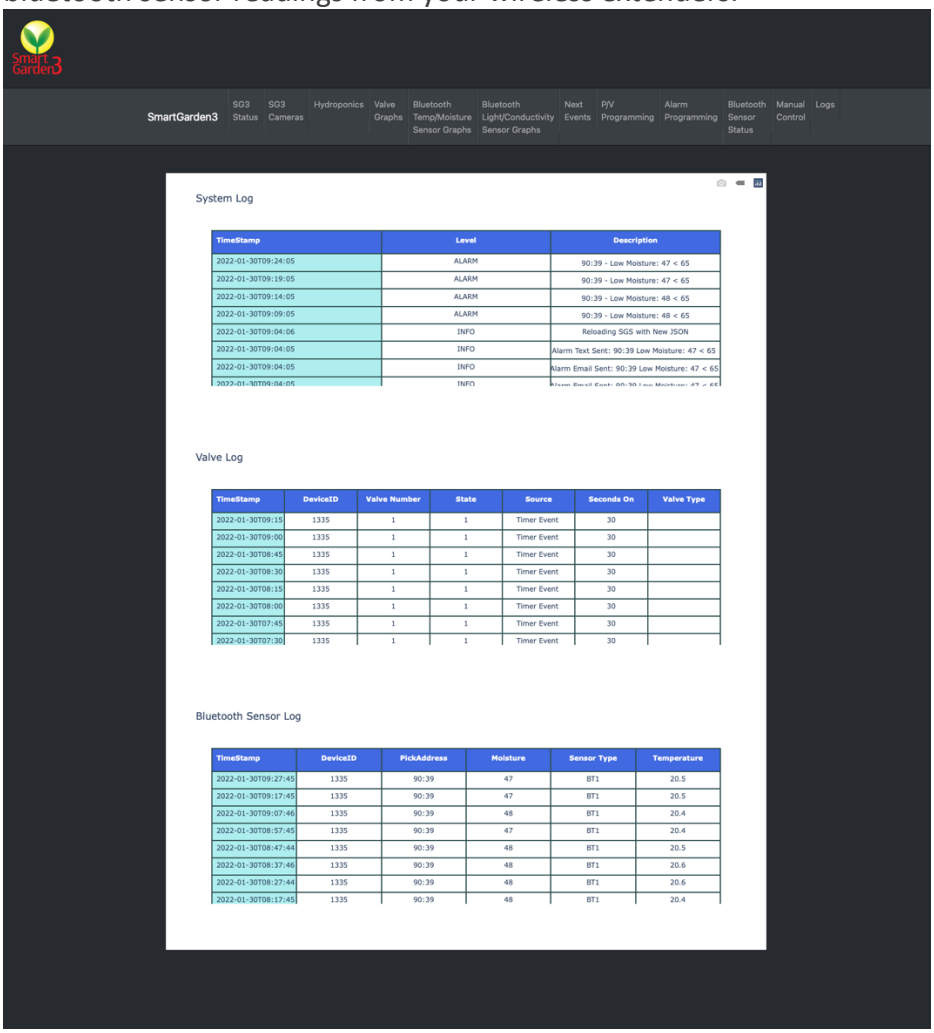
Manual Control Tab

This tab allows you to directly control valves on your Wireless Extenders. Note: These commands will override any currently on Valves and turn off the valve after the interval below. The timed valve programming is left as it is in SmartGarden3 and will resume on the next valve programmed event. Refresh the page if your extender shows not active.



Logs Tab

The Logs tab shows the last lines in the Log files maintained by the SmartGarden3. The main system log shows events, alarms, debugging events and status of your SG3 wireless extenders. The Valve Log shows valve events, their source and any criterial. The Sensor log shows you the latest bluetooth sensor readings from your wireless extenders.



Disclaimer

SwitchDoc Labs, LLC takes no responsibility for any physical injuries and possession loss caused by those reasons which are not related to product quality, such as operating without following the operating manual and cautions, natural disasters, or force majeure.

SwitchDoc Labs, LLC has compiled and published this manual which covers the latest product description and specification. The contents of this manual are subject to change without notice.