Integration of ecowitt Sensors with HCA using Hubitat

Ecowitt (ecowitt.com) makes a variety of sensors that can be used with their weather stations. These devices can also be used with just their app and connected to Hubitat using their GW1100 WiFi gateway. When integrated with Hubitat, data from the sensors can be forwarded to HCA. This Tech Note will describe the integration of an ecowitt WH51 Wireless Soil Moisture Sensor. Up to 8 such sensors can be supported by one gateway. These sensors could, for example, be used to control an automatic irrigation system connected to HCA. Other sensor types are available and could be integrated using the approach in this tech note. One interesting sensor ecowitt offers is the WH57 Outdoor Wireless Lightning Detection Sensor.

The Hubitat drivers can be found at:

https://github.com/sburke781/ecowitt#readme

(It should be noted that there are variants of these drivers on GitHub. This article specifically used the ones at the above URL).

The first step is to install the ecowitt WiFi gateway using the ecowitt app. The install instructions on GitHub say to use the "WS View" mobile app and shows an Android app. This may work, but if you are using iOS, you need to use the "ecowitt" app from the Apple store with this icon:
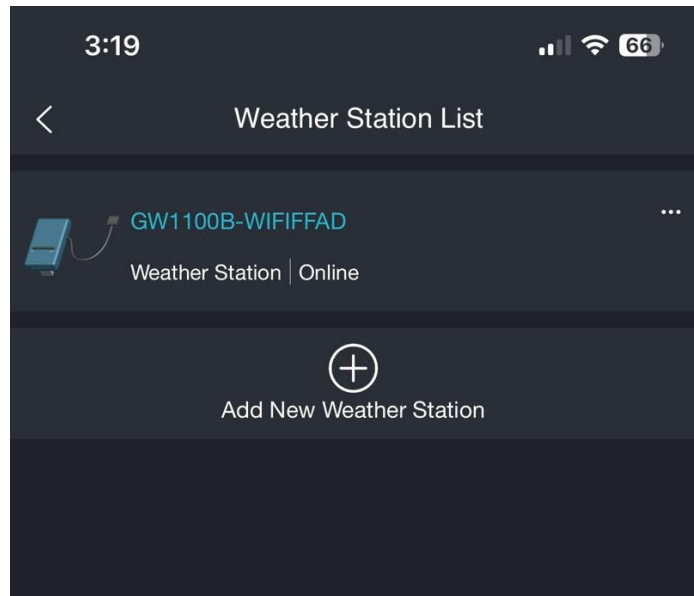

Once this app is installed, you should use the "Add New Weather Station" function to add the WiFi gateway. You can also setup an ecowitt account, although the cloud is not necessary for the gateway to forward information to the Hubitat, i.e. everything can operate via the local network. This screenshot shows the app with the gateway installed:
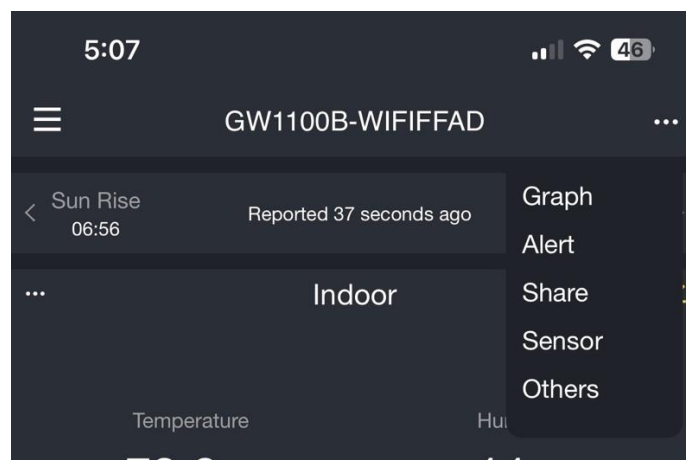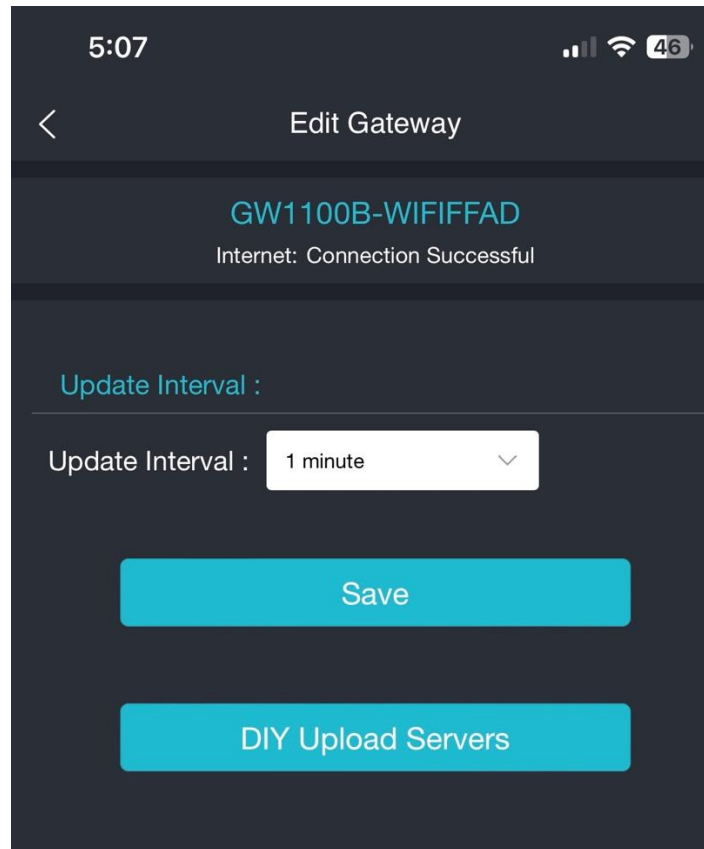
Once the GW1100 gateway has been added to the app, you need to set it up so that it will forward data to the Hubitat. To do this, select the "…" menu next to the gateway's name. This will pop up another menu:

Select "Share" which will take you to this menu:

Set the interval at which you want to update the server. Then select "DIY Upload Servers", which takes you to this menu, where you should select "Customized":

Protocol should be "Ecowitt", the Server IP should be that of the Hubitat and the port is 39501. Path should be "data", then select "Save". Note that there is no way to set a static IP on the gateway, so you should use your router to assign a fixed IP based on the WiFi gateway's MAC address. In order to find the MAC address of the gateway, click on the hamburger menu on the main app page and then select "Weather Station" which will show you the installed WiFi gateway. Select the "…" menu and this will bring up a page that will show you the MAC address (see below). Since this is the only way to find the MAC address of the gateway, this means you
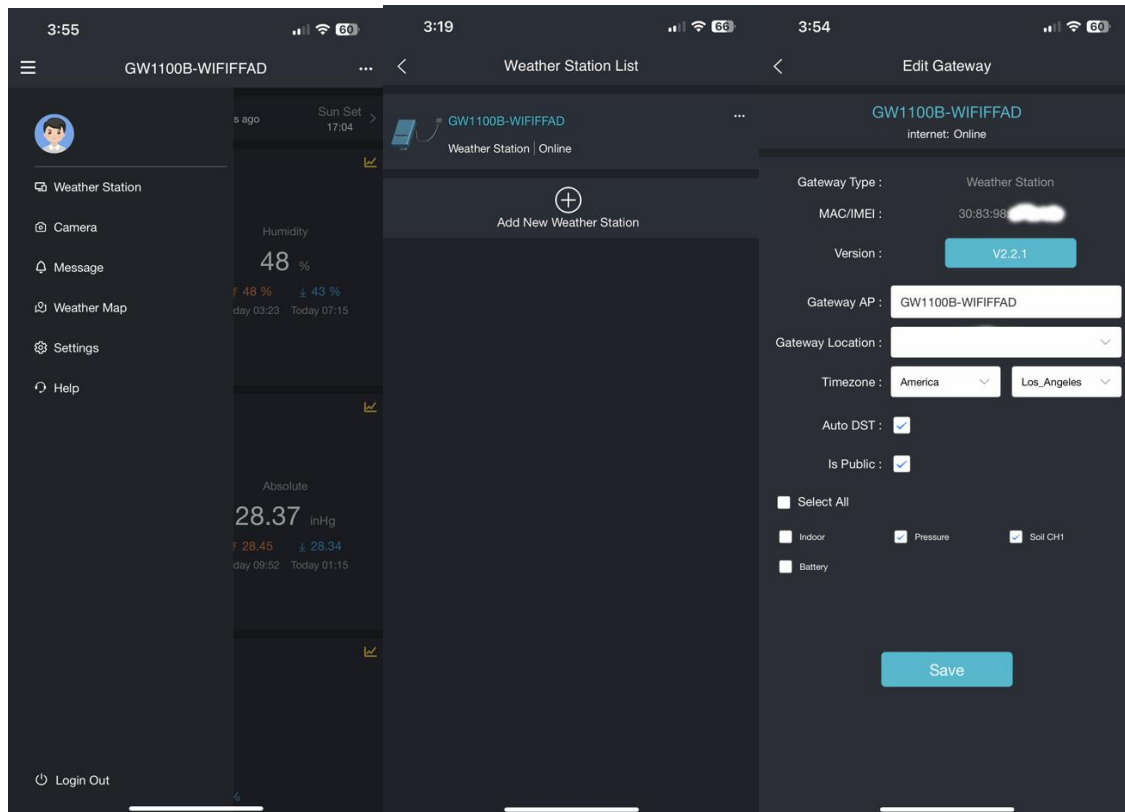
TechSup@HCATech.com

www.homecontrolassistant.com

need to install the gateway and let your router assign it a random address, *then* set a fixed IP into the router for the gateway once you can see the MAC address.

The next step is to install a battery in the moisture sensor. The gateway should auto discover the sensor once it begins transmitting (as shown by the flashing LED on the sensor). You can verify the sensor has been found by selecting "Sensor" from the "..." menu. As you scroll down, you should see the first sensor (CH1 ID) shows an antenna with signal bars and has been assigned the name "D2002":

TechSup@HCATech.com

www.homecontrolassistant.com
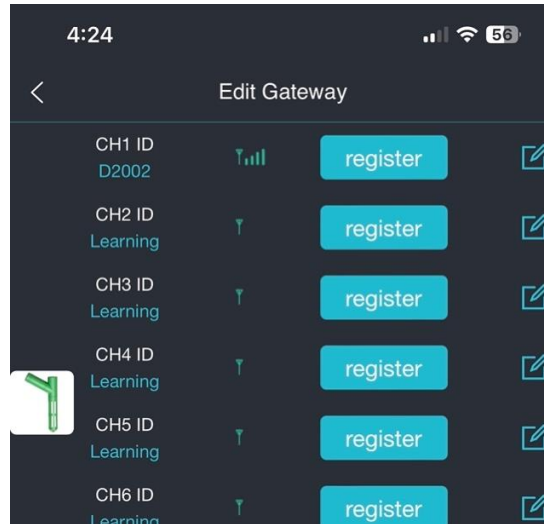
Once the gateway is setup in the app, the next step is to verify data is getting forwarded to the Hubitat from the gateway as described in the GitHub instructions. If that is working, you can then install the two Hubitat drivers for the Ecowitt gateway (ecowitt_gateway.groovy and ecowitt_sensor.groovy). These can be manually installed using the cut and paste procedure used to install the HCA Hubitat drivers and described in HCA Tech Note 700. You can verify the device is in Hubitat by looking at "Devices" for Ecowitt WiFi Gateway. Note the gateway also contains a temperature sensor. It may take a few minutes for data to initially be pushed to the Hubitat from the gateway, based on the time interval selected. This shows the gateway and installed sensors in "Devices".

It is probably a good idea to reboot the Hubitat after the drivers have been installed and configured. It was noted that for some unknown reason data wasn't getting forwarded to the Maker API until the reboot occurred. To verify data is getting to the Maker API, you can turn on logging in the MakerAPI ("Enable Logging

TechSup@HCATech.com

www.homecontrolassistant.com

for debugging") and look in "Events" for the sensor device which should show data is getting forwarded under "Triggered apps". *Also, don't forget to add your new devices into Maker API so HCA can find them.*

To add the device to your HCA design, you should run the "#1 Do Discover" program in the Hubitat HCA package. This should find the ecowitt device(s) and place them in your Hubitat folder. You can then do a Get Status on the device(s) which will populate the tags for the device. Unfortunately, the ecowitt drivers send data fields for all possible ecowitt devices, not just for the particular sensor you have installed. The number of data fields sent exceeds the number of tags HCA supports. To deal with this, you need to determine the tags you want for the particular sensor, delete all the unneeded tags from the device, enter the tags you want, and do another Get Status. HCA will again fill the unused space with irrelevant tags that have been sent, but the tags you need will be there. To determine what is being sent by the sensor, you would use HCA's "Tools-Aux Log Setup" and select "Logs for Programs elements-HTTP element" with the Hubitat sensor class selected:
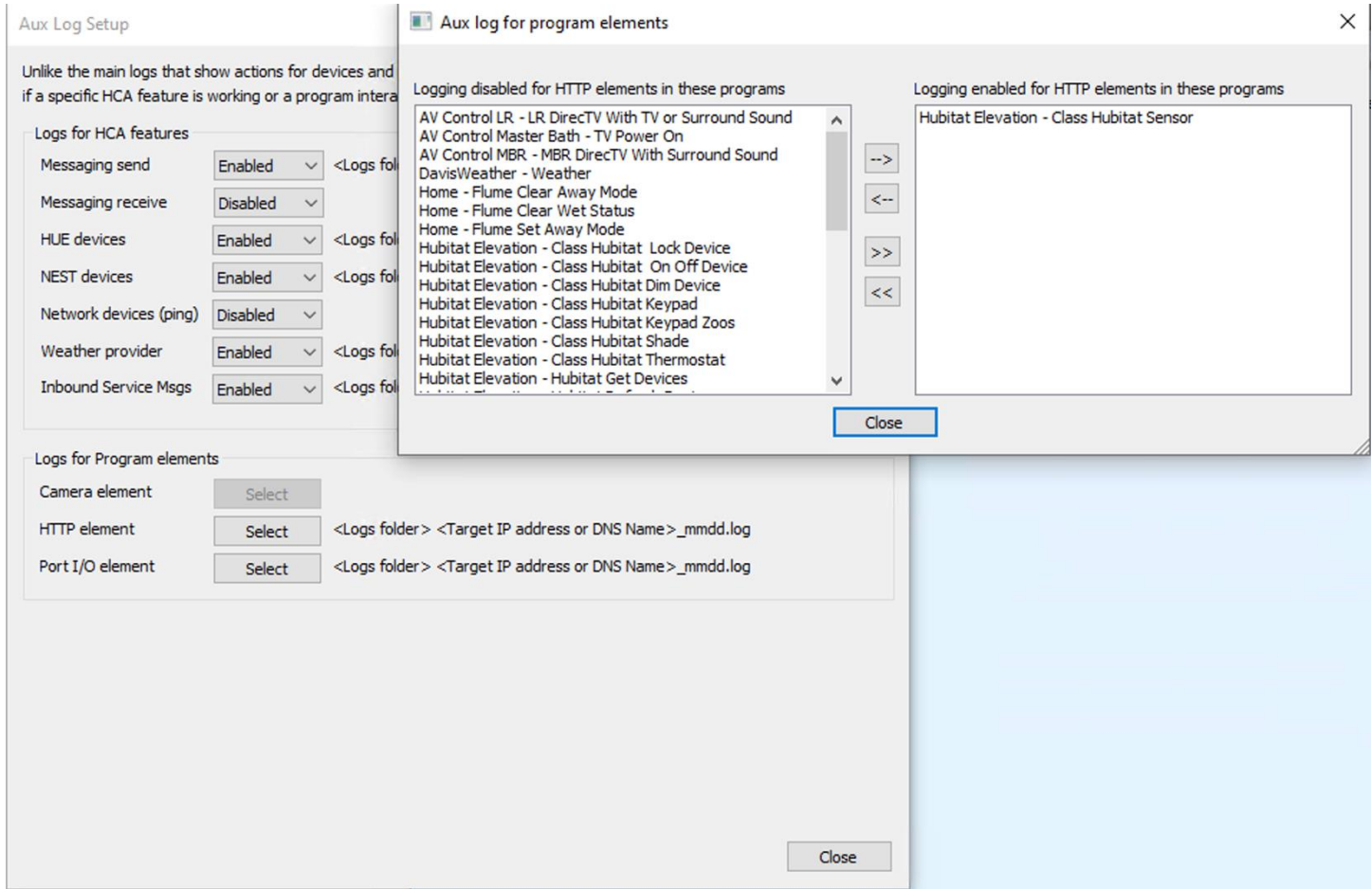
The resulting log file will contain JSON of all the fields being sent back from the ecowitt after a Get Status on the device. in the case of the soil moisture sensor, two fields are of interest: "Humidity" (soil moisture) which would go into the device tag "HE_Humidity" and "Battery" (percent) which would go into the device tag "HE_Battery". You also want to create a tag HE_LastEventAt, as HCA will populate this with the time that the sensor last reported data. The timestamp is useful in determining battery status in the case where the battery became too weak to send a report.

"Sprinklers - Soil Moisture Sensor 1" Properties   ✕

| Name | Notes | Room | Type | User Class Object | Options | Triggers | Restart | Icon | Display | Poll | Transmission |
| Log | | Groups | | Schedule | | Referenced By | | Power Track | | Green | Tags | Statistics |

Tag Name	Current Value		Tag Name	Current Value
HE_type	Ecowitt RF Sensor		HE_UltravioletIndex	null
HE_devType	Sensor		HE_BatteryWindIco	null
HE_devSubType	water		HE_PressureAbs	null
HE_LastEventAt	1/17/2023 9:16:46 AM		HE_WindGust	null
HE_Humidity	35		HE_Orphaned	false
HE_Battery	57		HE_BatteryWindOrg	null
HE_Temperature	null		HE_LightningEnergy	null
HE_Pm10	null		HE_RainEvent	null
HE_AqiColor	null		HE_SimmerIndex	null
HE_RainYearly	null		HE_BatteryTemp	null
HE_WaterColor	null		HE_LightningCount	null
HE_CarbonDioxide	null		HE_LightningDistan	null
HE_OrphanedTemp	null		HE_Status	null
HE_SimmerColor	null		HE_HumidityAbs	null
HE_WindDanger	null		HE_UltravioletDang	null
HE_HeatDanger	null		HE_WaterMsg	null

"Tags" are name-value pairs associated with devices, programs, groups, folders, rooms, and displays that your programs can manipulate.
You can view and modify them here but they are usually manipulated by programs.

OK   Cancel

##end##