Using Helium GPS Tracker Raspberry Pi Image File

Table of Contents

tags: `IoT`

Software Access Credentials

Here is a list of the account info for the various software applications. When installing you should configure your own unique passwords.

SSH
User: helium
password: iot

NodeRed
Port: 1880
User: admin
password: iot

Grafana
Port: 3003
User: admin
Password: heliumiot

Mosquitto MQTT Broker
Port: 1887

MQTT account for Helium Console
user: heliumconsole
password: heliumRocks

MQTT account for Node-Red
user: nodered
password: nodered33

MQTT account for mqtt explorer or other MQTT monitor app
user: mqtt_explorer
password: mqtt_explorer33

Test account with full read/write access
user: helium
password: iot

Onboard GPS Tracker onto Helium Network and Connect to Community Helium Mapper Project.

Use the following guide to help you onboard the LGT-92 tracker.
The guide also shows how to forward data from your GPS tracker to the open source Mapper Project. This community mapper project anonymoizes the data source and shows potential sensor companies the "real" coverage provided by the netork.
m/
https://docs.helium.com/use-the-network/coverage-mapping/mappers-quickstart/

If you do not want to submit data to the community mapper project then complete all the steps except for step 2(Create HTTP Integration).

If you have successfully onboarded the sensor, added a decoder function, and are receiving data to the console than we can proceed to add the MQTT integration between Helium Console and our Raspberry Pi.

Create Helium Console MQTT Integration

Determine your external IPv4 address.
https://whatismyipaddress.com/

Create MQTT Integration

1. go to https://console.helium.com/integrations
2. Select Custom MQTT Integration and fill in the following parameters
   Endpoint: mqtt://heliumconsole:heliumRocks@YOURIPADDRESS:1887
   Name of Integration: LGT92 MQTT on Pi Integration
3. Copy the following into the Template Body section of the JSON Message Template and press save.
   This template will flatten the JSON and extract just the data we are interested in.
   I have found that many applications do not really like JSON with multiple levels of hierarch.

```
{
  "name": "{{name}}",
  "dev_eui": "{{dev_eui}}",
  "device_id": "{{id}}",
  "reported_at": "{{reported_at}}",
  {{#hotspots}}
  "hotspot_name": "{{name}}",
  "rssi": "{{rssi}}",
  "snr": "{{snr}}",
  "spreading": "{{spreading}}",
  "raw_data": "{{payload}}",
  {{/hotspots}}
  {{#decoded}}{{#payload}}"ALARM_status": "{{ALARM_status}}",
  "battery": "{{BatV}}",
  "accuracy": "{{Accuracy}}",
  "altitude": "{{Altitude}}",
  "firmware": "{{FW}}",
  "LON": "{{LON}}",
  "lat": "{{Latitude}}",
  "lon": "{{Longitude}}",
  "PDOP": "{{PDOP}}",
  "pitch": "{{Pitch}}",
  "roll": "{{Roll}}",
  "MD": "{{MD}}"{{/payload}}{{/decoded}}
}
```

4. Finish by pressing "Add Integration" button

Create a label to connect device, integration, and functions

1. Go to Labels->Add Label
2. Set label name = MQTT_Pi
3. Add label attachments:
    - Integration: LGT92 MQTT on Pi Integration
    - Function: Dragino LGT92 Decoder
4. Press button "Add Label & Manage"
5. Press button "Add this Label to Device"
   Select the LGT92 device and press "Add Label to Devices"

The MQTT integration should now be setup and data should be published to your MQTT broker running on the Pi.

Configure Port Forwarding and DHCP Reservation

You should first use DHCP reservation to give the Pi a static local IP address.

You will need to open up one or two ports on your internet provider's modem/router.

1. MQTT Broker Port 1887 (Required)
2. Grafana Dashboard port 3003 (optional)

The MQTT broker port needs to be open so that the Helium Console router is able to send data packets to your MQTT broker.

If you want to view your dashboard from outside of your home network then you will need to also open up the Grafana port. The default port for Grafana is 3000. During setup we modifed to the port to be 3003 so open up the port that you configured.

tags: `IoT`