RV Media Center setup

Raspberry Pi 4 with 1.5 TB of storage running the Plex Media Server software. This hosts all of our media (Music, Movies and TV Shows)

# Hardware

* Raspberry Pi 4B/3B+/3B X735 Power Management with Safe Shutdown Auto Cooling Expansion Board Compatible with Raspberry Pi 4 Model B
* X829 Matching Metal Case/Enclosure + Power Control Switch + Cooling Fan for Raspberry Pi 4B & X829 & X735 (Not Support X825)
* Geekworm DC 5V 4A Power Adapter with Only US Plug DC 5.5x2.5 for Raspberry Pi X820/X825/X828 SATA Expansion Board/ X705/ X725/ X750 UPS/ X735 Power Management Board/Jetson Nano/T300/T100/T200
* WD Green 1TB Internal PC SSD - SATA III 6 Gb/s, 2.5 Inch /7mm - WDS100T2G0A
* Geekworm Raspberry Pi 4 NAS Dual SATA, X829 Dual 2.5 inch SATA HDD/SSD Storage Expansion Board Compatible with Raspberry Pi 4 Model B Only
* DC Buck Module, DROK Adjustable Buck Converter Step Down Voltage Regulator 6V-32V 30V 24V 12V to 1.5-32V 5V 5A LCD Power Supply Volt Reducer Transformer Module Board with USB Port Protective Case (Don’t like this, every power on you have to reset it to USB power)
* CanaKit Raspberry Pi 4 4GB Starter Kit with Clear Case (4GB RAM)

# Installation

I installed the Raspberry Pi in the cabinet, then installed the cabinet next to the One Control router and connected the Pi to the router via a network cable because the metal case containing the Pi kills the wireless connectivity.

I used the DC Buck module to power the Pi from the 12volt in the RV. This way the Pi is powered with or without shore power. This is important since the custom software is most useful when shore power is lost. The Buck Module converts 12 volt to USB 5 volt, but the convert has to be configured to USB power each time it is powered up. I may re-think powering the Pi so that it is always powered by 5 volts.

# Custom Software

I used Python to build a custom application (RVtemp.py) that queries the One Control, via the HAB API, to check for the RV temperature and the status of the generator.

The Application reads a .ini file so the user can customize their configuration without having to make code changes.

The application sends emails alerts when the temperatures are outside of the temperatures defined in the THRESHOLDS section of the .ini file.

The application also creates a log file journaling all of the temperatures each time the app runs.

The application creates a new log file for each day, then cleans up old logs old than the days specified in log retention date entry.

The application is started by cron. I’ve configured cron to run the application every 15 minutes. Of course the application interval can be changed, but I figure the temperature cannot change that drastically in 15 minutes.

Email notifications:

The application sends two emails.

* The notification\_address is the email address that will receive notifications when the temperature is either too high or too low. Originally, I set this up to be a text address, but then changed it to a email distribution list that contains two text addresses. This way we both are notified if the temperature is out of bounds.
* The log\_address is the email that will receive an email every time the application runs. I setup a different email address that will be emailed every 15 minutes with the output even if the temperatures are within bounds. I did this so that I know the application is running as its supposed to run. I thought about creating a phone based app that could talk back to the application, but didn’t feel the complexity of building the application was worth the value it would provide. If you specify a log\_address of ‘none’, the logs won’t be sent each time the application runs.

Here is the format of the email sent to both email addresses:

Garage 74

Main 74

Bedroom 71

Batt V 13.3

Gen State OFF

Hi limit 80

Low limit 60

Again, this same message format will be sent to two different email addresses. The notification\_address is only emailed when we need to be alerted, this address texts both our cell phones.

I setup a new email address for the log\_address and configured my cell phone to sync with that email address for the logs. I can quickly check this email and view the last time the application ran and the current temp in the coach. This is a push-based method so it will always be on my phone.

# Email setup

I have an office 365 subscription, so I used Office 365 as my messaging platform. The configuration should also work with outlook.com based emails, but I haven’t tested that yet. Originally, I tried to use gmail for messaging, but gmail kept blocking my emails due to security.

I created a distribution list that receives the emergency notifications, my email is [rv@robhome.com](mailto:rv@robhome.com)

I created an email account, [rvlogs@robhome.com](mailto:rvlogs@robhome.com) that receives the log message every 15 minutes.

# Logging

The log file stored on the Pi will show a log of all of the runs for the day. Each days log is stored in a separate log file. The format of the log data is as follows:

2020-08-25 21:30:02,805 CRITICAL \*\*

2020-08-25 21:30:02,806 CRITICAL Starting the RV Temp Monitor

2020-08-25 21:30:02,806 CRITICAL \*\*

2020-08-25 21:30:03,236 INFO No need to send email, temps are acceptable

2020-08-25 21:30:03,236 INFO

Garage 73

Main 74

Bedroom 72

Batt V 13.3

Gen State OFF

Hi limit 80

Low limit 60

2020-08-25 21:30:07,585 INFO Gathered temps and emailed to log address :rvlogs@robhome.com

2020-08-25 21:30:07,586 CRITICAL \*\*

2020-08-25 21:30:07,590 CRITICAL Ending run of the RV Temp Monitor

2020-08-25 21:30:07,591 CRITICAL \*\*

### Logging levels

Different logging levels can be set, my recommendation would be to leave the log level at either 10 or 30. A level of 30 will ensure the temperatures are logged each run. A log level of 50 will only log the start and stop of the application, but no temperature data. Don’t worry about the word “CRITICAL”, that’s just the only way I could ensure the start and stop entries are always journaled.