**TO-DOs:**

* From: <https://github.com/openaps/docs/blob/master/docs/getting-started/rpi.md>
  + change user login password or disable, only use SSH keys
  + figure out accessing Git on the device (necessary if any changes are being made, probably also to store me-specific files
* Read this: <https://github.com/openaps/openaps/blob/master/README.md>
* ~~From:~~ [~~https://github.com/openaps/docs/blob/master/docs/Log-clean-analyze-with-openaps-tools/using.md~~](https://github.com/openaps/docs/blob/master/docs/Log-clean-analyze-with-openaps-tools/using.md)
  + ~~configure OPENAPS on the Pi~~
* From: <https://github.com/openaps/docs/blob/master/docs/Build-manual-system/Using-oref0-tools.md>
  + re-invoke and review the reports with insulin pump settings, after I get old pump setup
  + starting doing steps after here: $ openaps report add monitor/glucose.json JSON cgm iter\_glucose 5
* draw out flow chart of reports, checks, and enacts for me to follow

**NOTES:**

* OPENAPS raspi:
* 192.168.1.132
* pi, raspberry

$ openaps use pump -h





$ openaps use cgm –h:



git push to Bitbucket:

Top of Form

Bottom of Form

Repository setup

Your repository is empty — let's put some bits in your bucket.

[I have an existing project](https://bitbucket.org/mikestebbins/openaps00#command-line-existing)

Already have a Git repository on your computer? Let's push it up to Bitbucket.

cd /path/to/my/repogit remote add origin https://mikestebbins@bitbucket.org/mikestebbins/openaps00.gitgit push -u origin --all # pushes up the repo and its refs for the first timegit push -u origin --tags # pushes up any tags

In order to eliminate needing to enter the Bitbucket password every time I “git push”, use the following instead of above, “… add origin https://mikestebbins: [BeRtensg@bitbucket.org](mailto:BeRtensg@bitbucket.org)...”

Crontab

Examples: <https://www.raspberrypi.org/documentation/linux/usage/cron.md>

<https://www.raspberrypi.org/forums/viewtopic.php?f=91&t=39344>

Access via:

$ crontab –e

Enter in Nano:

\*/5 \* \* \* \* (cd ~/openaps00/ && git push)



Nightscout integration

Followed these instructions: <https://github.com/openaps/docs/blob/dev/docs/Automate-system/vizualization.md>

Set-up clock zoned from line in script here: <https://github.com/openaps/oref0/blob/master/bin/ns-uploader-setup.sh#L51>

**More CRON to be set-up:**

live4sw 02:10

I think I may need to add something to retry-loop to deal with this scenario. My retry-loop is currently the default, which is retry-loop = ! bash -c "until( ! mm-stick warmup || openaps loop); do sleep 5; done". Has anyone here encountered a similar issue?

Basically, if I reboot, or reset the USB, it gets unstuck, but otherwise it just throws off that error every 5 minutes and doesn't even get to the preflight, even though everything is connected and i'm a few feet away from the stick

scottleibrand 03:48

openaps alias add preflight '! bash -c "rm -f monitor/clock.json && echo -n \"PREFLIGHT \" && openaps report invoke monitor/clock.json 2>/dev/null >/dev/null && grep -q T monitor/clock.json && echo OK || ( ( mm-stick warmup 2>&1 || sudo oref0-reset-usb ); echo FAIL; sleep 120; exit 1 )"'

openaps alias add retry-loop '! bash -c "openaps wait-loop || until( ! mm-stick warmup 2>&1 | egrep -v \"^ \" || ! openaps preflight || openaps loop); do sleep 10; done"'

live4sw 04:47

What do you use for your wait-loop alias? I didn't see that discussed in the docs.

scottleibrand 08:48

openaps alias add wait-loop '! bash -c "openaps preflight && openaps gather && openaps enact && openaps report invoke monitor/temp\_basal.json 2>/dev/null >/dev/null && openaps upload && openaps get-settings 2>/dev/null >/dev/null && openaps wait-for-bg && openaps enact && openaps upload-ns-status >/dev/null"'

live4sw 09:28

Thanks Scott, unfortunately this brings up wait-for-bg and several aliases not discussed in the docs. I see Jason uses these too. What is the purpose of the "wait" aliases? I think I may need to spend some time this weekend really trying to understand how these aliases do things differently.

jasoncalabrese 09:31

the wait-for-bg causes the loop to wait until BG changes, and just keeps polling

you then need a cron to killall openaps processes older than 10 minutes

\* \* \* \* \* killall -g --older-than 10m openaps

live4sw 09:40

Oh I see, so is the logic here that by using the wait, you can make sure that your loop runs immediately when there is a new bg? I assume your cron still runs retry-loop every 5 minutes, you're just killing old processes to make sure they don't snowball?

jasoncalabrese 09:49

the cron runs every minute, but checks if it's already running first

I use \* \* \* \* \* cd /home/edison/indy && ( ps aux | grep -v grep | grep -q 'openaps retry-loop' && echo OpenAPS already running || openaps retry-loop ) 2>&1 | logger -t openaps-loop

**Must go back and fix report monitor/upload-status.json, as I removed the enacted items just to test communication with Nightscout for CGM upload.**

**Python file in Linux**

Create new file from terminal:

* touch /path/to/file for an empty file

Just put this in the first line of your script :

~~#!/usr/bin/env python~~

Actually, change it to this:

#!/usr/bin/python

Then make the file executable with

chmod +x myfile.py

Execute with

./myfile.py

Use subprocess to run it in the shell:

<https://docs.python.org/2/library/subprocess.html>

<http://stackoverflow.com/questions/89228/calling-an-external-command-in-python>