

Software

Bestandteile

Die Software besteht aus kleinen Modulen, die zentral von einer Ablaufsteuerung koordiniert werden.

- Ablaufsteuerung
- PiFM (Python-Wrapper)
- APRS (Python-Wrapper)
- Robot36 (Python-Wrapper)

PiFM

- übergebene WAV-Datei über HF aussenden
- **TODO** hinterher HF abschalten

APRS

- **TODO** übergebene Position (Lat, Lon, Höhe, Temperatur) in WAV verpacken
- via PiFM versenden

Robot36

- übergebene Bilddatei in wav wandeln
- via PiFM versenden

Ablaufsteuerung

Im Vorbereitungsbetrieb:

- Start der Software
- Warten auf GPS Fix
- LED an
- Testaussendungen (APRS, Pause, SSTV, Pause, APRS)
- LED blinken
- 1 Minute warten
- Missionsstart (LED aus)

Im Missionsbetrieb:

- Speichern aktuelle Positionsinformation
- Aussendung APRS auf 144.800 MHz (2 Sek)
- 1 Bild speichern
- Ansage 3xLAT, 3xLON auf 145.200 MHz (10 Sekunden)
- Aussendung SSTV auf 145.200 MHz (36 Sekunden)

- Pause (10 Sekunden)
- Aussendung APRS (3 Sek)
- Pause 20 Sekunden
- nach X Minuten (je nach Version) wird die Nutzlast abgesprengt
- nach X + TBD Minuten wird der Raspberry Pi heruntergefahren und/oder abgeschaltet

Absprengung:

- Mikrotaster abfragen
- wenn geschlossen: Heizung an bis er offen ist
- wenn offen: Heizung für ?? Sekunden an
- Aussenden Signalton!

offene Fragen

- Muss man den Raspberry Pi von Spannung trennen, oder braucht er nach Herunterfahren nur noch wenig Strom? -> scheinbar ist es so, dass er nach dem runterfahren noch mehr strom braucht - also besser anlassen.

Aufsetzen des Produktivsystems

Installation des Grundsystems

- <http://distribution.hexxeh.net/raspbian/raspbian-r3.zip> herunterladen und per dd auf SD-Karte schreiben
- Raspberry Pi starten lassen, IP per DHCP zuweisen → pi:raspberry
- root-Passwort ändern
- `rm /etc/ssh/ssh_host_* && dpkg-reconfigure openssh-server`
- `apt-get update`
- `apt-get install ntp fake-hwclock`
- `apt-get install tasksel`
- `tasksel install standard`
- `apt-get update && apt-get dist-upgrade`
- `rpi-update`

Vorbereitung Temperatursensor

- `apt-get install i2c-tools lmsensors`
- `modprobe i2c-dev echo i2c-dev » /etc/modules`
- `i2cdetect -y 1 -> Anzeige des LM75`
- `echo lm75 0x48 > /sys/class/i2c-adapter/i2c-1/new_device`

From:

<http://loetlabor-jena.de/> - **Lötlabor Jena**

Permanent link:

<http://loetlabor-jena.de/doku.php?id=projekte:xplorerer:software&rev=1391359874>



Last update: **2014/02/02 16:51**