Receive imaginary automatization

Feasibility study

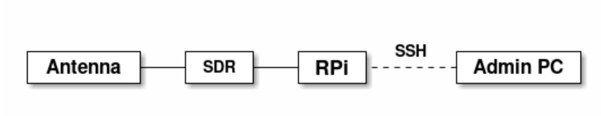
Sławomir Figiel Tomasz Mrugalski Ewelina Omernik

2019-12-09

Main objectives

We need a software for automatically receive imaginary from satellites. We want to track NOAA satellites and receive APT transmissions. Our station is built on Raspberry Pi 4. We connect to this using SSH protocol.
Software should fetch data about next transition, adjust SDR, manage record the signal, extract picture and save this to specific target location

Physical architecture



Requirements

Functional requirements

1. Predict satellite transits
2. Adjust SDR radio
3. Recording signals
4. Extract picture from signal
5. Save picture to disk

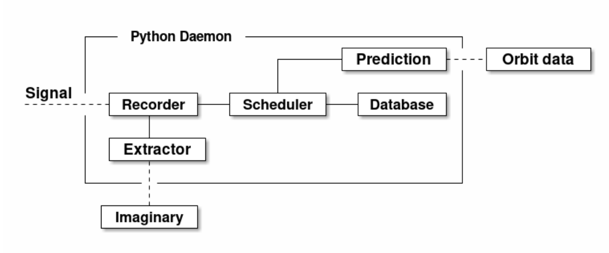
Non-functional requirements

1. Automatically receiving
2. Failsafe

State of art

- raspberry-noaa
- weather-satellites

Software components



I propose that most components should be realized using external command line tools. These tools should be called by core Python script. This core script may be installed in operating system as SystemD daemon for autostart, work without session and OS-level fail procedure support.

1. Prediction - component for calculate next prediction time. It needs actual data about orbits from Internet.
   Proposed software: Predict, others
2. Database - store frequencies of tracked satellites. At start it may be realized as dictionary in source code
3. Recorder - component for record signal. It should save data in lossless compressed formats.
   Proposed software: rtl_fm, GQRX
4. Scheduler - component for call the record flow on specified time. It may be realized with Python "sched" module. It need to contains algorithm to decide which satellite should be tracked when many satellites are visible.
5. Extractor - component for convert recorded signal to graphic data (and text metadata). Probably we will needed preprocess signal before extraction

(for example resampling).
Proposed software: sox (resampling), noaa-apt, wxImage

Possible issues

1. Only one process can access to SDR in the same time
2. Transitions can overlaps
3. We don't know how long will processing take. Scheduler may call next task before end previous
4. SDR device may be busy
5. Predictor must to update orbit data periodically from Internet or other source
6. Script may be stopped unexpectedly

Conclusions

A lot of possible components are ready and well tested. We need only connect them.
Also exists ready solutions which use the same (or similar) utilities. We can use them too. They include a fancy interface and ready share server, but they may to have a problems with handle fails.

Completing the task is possible. We need max 2 weeks to do it.