X-CubeSat Controler

Version Pre-Beta

User Guide

Author:
Tiago L. Gimenes

Programmer:
Tiago L. Gimenes

April 2, 2014

Chapter 1

Requirements

The current version of X-CubeSat Controler works only in POSIX compliant operating systems, like GNU/Linux, FreeBSD, etc, but future versions will work also in non POSIX compliant operating systems like Windows. You can find below the libraries and software needed to compile and run this version of XCubeSat Controler. The versions of the libraries and software below were tested, try other versions at your own risk.

- g++ version 4.7.2
- GNU Make 3.81
- Git version 1.7
- GTK+3
- gtkmm version 3.0 or above
- SQLite version 3.7.13
- SQLite Wrapped version 1.3.1
- Gpredict modified version that can be found in the following address `https://github.com/tlgimenes/gpredict`

Chapter 2

Installation Instructions

The installation instructions will cover the compiling and installing phases. It will only shows how to install in the Debian GNU/Linux distribution but the same method should work in Ubuntu, Linux Mint and other Debian based distributions.

You can install this program in any directory that you want. For doing it

2.1 Installing dependencies

- **Install g++**: Open a terminal and type: $ sudo apt-get install -y g++ If you get no errors, you can go to the next item, otherwise google the problem.
- **Install GNU Make**: In the same terminal type: $ sudo apt-get install -y make If you get no errors, you can go to the next item, otherwise google the problem.
- **Install Git**: In the same terminal type: $ sudo apt-get install -y git If you get no errors, you can go to the next item, otherwise google the problem.
- **Install gtkmm version 3.0**: In the same terminal type:

```
$ sudo apt-get install -y libgtkmm-3.0-1 libgtkmm-3.0-dev libgtkmm-3.0-dbg
```

If you get no errors, you can go to the next item, otherwise google the problem.

- **Install SQLite version 3**: In the same terminal type:

```
$ sudo apt-get install -y sqlite3
```

If you get no errors, you can go to the next item, otherwise google the problem.

- **Install SQLite Wrapped version 1.3.1**: Go the a directory that you want with the command **cd** and in the same terminal type:

```
$ wget http://www.alhem.net/project/sqlite/sqlitewrapped-1.3.1.tar.gz
$ tar -zxvf sqlitewrapped-1.3.1.tar.gz
$ cd sqlitewrapped-1.3.1/
$ make && make install
```

If you get errors in this part, go to the url of the SQLite Wrapped project,`http://www.alhem.net/project/sqlite/download.html`, and follow their install instructions, sometimes the installation of this part is a little tricky.

- **Install Gpredict modified version**: Go the a directory that you want to install Gpredict with the command **cd** and in the same terminal type:

```
$ git clone https://github.com/tlgimenes/gpredict.git
$ cd gpredict
$ make && make install $ ./src/gpredict
```

If you get errors in this part, go to the url of the Gpredict modified version project,`https://github.com/tlgimenes/gpredict`, and follow their install instructions, or read the **README** file provided with Gpredict, sometimes the installation of this part is a little tricky.

2.2 Compiling

Usually if you got to this part, you should be ok to compile and install XCubeSat Controler. Open a terminal, go to the folder that you want to install XCubeSat Controler with the **cd** command and type:

```
& git clone https://github.com/tlgimenes/X-CubeSat
$ cd X-CubeSat
$ make
$ cd dataBase
$ sqlite3 XCubeSat_ Controler.db
.read tables.sql
.q
$ cd ..
$ ./XCubeSat_ Controler
```

Here you are running XCubeSat_ Controler \o/.

Chapter 3

Overview

The X-CubeSat Controler is a software for sending, receiving and stoking received data from the X-CubeSat and other satellites from a modem connected to a serial port to the computer. For each satellite we can program many scripts that are going to be put in a FIFO (first in first out) queue to be executed. The first script in the FIFO queue is executed when the satellite is above the horizon (calculated by Gpredict), than the script is reinserted in the last position of the FIFO queue and a new script is ready to be executed in the next pass of the satellite.

The programming language is a simple, interpreted programming language that will allow to perform the act of sending, receiving and stoking data in the SQLite database. For more details about the programming language for this software, take a look in the others documents provided with this software.

The X-CubeSat Controler is divided in many different frames that are going to be presented in the following chapters. Each frame has an specific function, from controlling the port that the modem will be connected to writing scripts and changing their priority. You can have a general view of the graphical interface in the figure 3.1.

Figure 3.1: General view of the graphical interface of X-CubeSat Controler

Chapter 4

Frame: Port Config

The port config frame can be seen in the figure 4.1. This frame allows to change the serial port that the modem is connect to the computer and its speed.

Figure 4.1: View of the PortConfig frame

The red buttons that we can see in the figure above indicates the status of the port and its speed; green means that the serial port is connected and/or the speed for the port is supported and read means the opposite. You can see an example in the Figure 4.2 that the serial port is connected and the speed is supported.

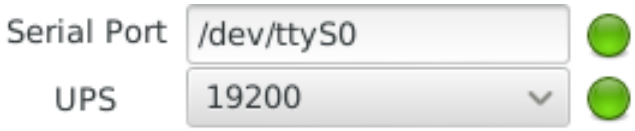

Figure 4.2: View of the PortConfig frame green

In GNU/Linux systems the usual serial port is **/dev/ttyS0** for the motherboard ports and **/dev/ttyUSB0** for USB adapters.

Chapter 5

Frame: Sats

In the Sats frame you can find the satellites that are available in the current Gpredict configuration. If you add a new satellite in Gpredict, it wont be displayed until the next restart of this program.

In a tree path, we can see the scripts related to a satellite. You can rename them just by clicking over it and renaming it. Rename a script is essential for adding a new one, because a satellite cannot have two scripts with the same name.

Warning: If the satellites shares a script with the same name with another satellite, there will be a race condition while saving the file and when re-opening then, one of them will be lost.

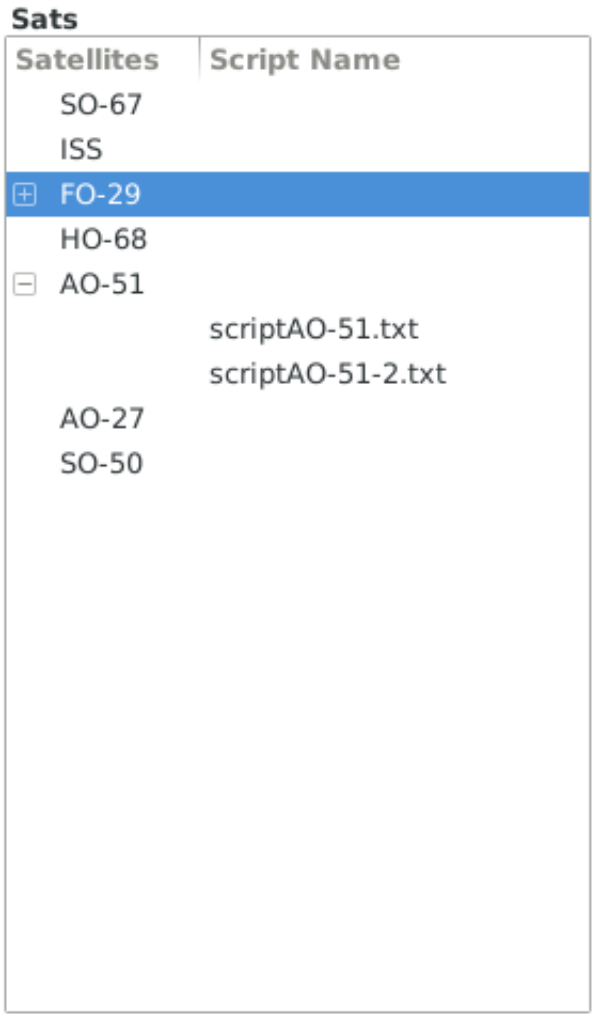

Figure 5.1: View of the Sats frame

Chapter 6

Frame: CurrSat

In the CurrSat frame you can find information concerning the next satellite that X-CubeSat Controler will run the scripts.

The satellite name will appear as it is defined in Gpredict.

The fields elevation and azimuth shows the current elevation and azimuth of the satellite.

The field **Communication Status** can be **Idle** or **Communicating**, and it will change from **Idle** to **Communicating** when the script that is being executed asks for X-CubeSat Controler to send/receive data to/from the satellite.

The progress bar doesn't work yet in this version but it will show the progress while sending/receiving data to/from the satellite.

Figure 6.1: View of the CurrSats frame

Chapter 7

Frame: Config

In the config frame, you will be able to write and edit the script with the name in the field **Script Name** from the satellite with the name in the field **Configuration of Satellite**. In the figure 7.1 you can see the complete frame.

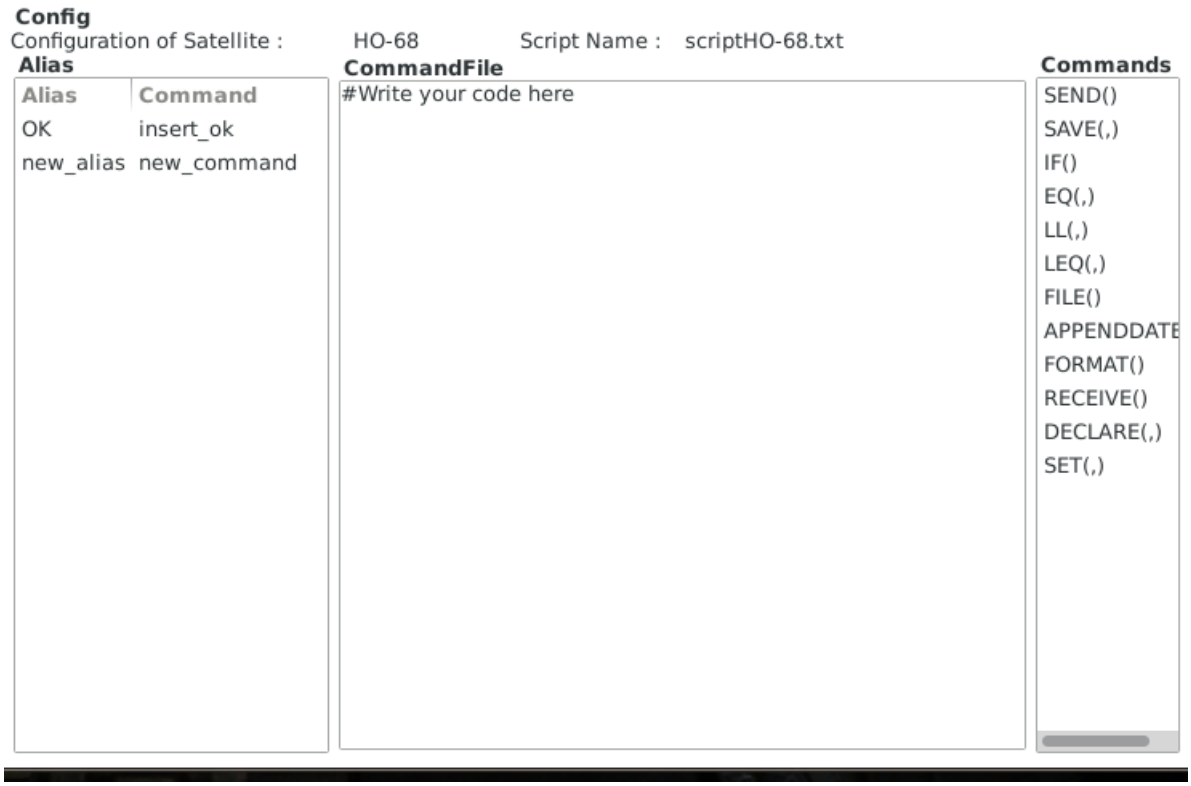

Figure 7.1: View of the Config frame

7.1 Sub Frame: Alias

In the sub frame alias you can find the alias for the current script that is being edited. An alias allows the programmer to define "nicknames" for a standard command or to give names to constants.

An example of the use of an alias is to define the constant string that is going to be sent and than, instead of writing **SEND("12 32 42 25 d2 f4")** you can create an alias, and in the column **Alias** you can write **mode 1** with the corresponding column **Command** as **12 32 42 25 d2 f4** and than, write in the script **SEND("mode 1")**. It gives the code clearer and easier to understand.

You can find a view of this frame in figure 7.2.

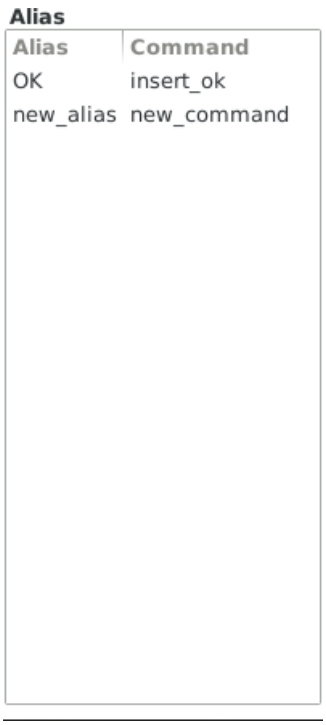

Figure 7.2: View of the Alias sub frame

7.2 Sub Frame: CommandFile

The CommandFile frame is the text editor where you can write the script for the selected script for the selected satellite.

You can find a view of this frame in figure 7.3.

7.3 Sub Frame: Commands

In this frame you can find all the commands that can be written in the script. To add a certain command to the current script that is being edited, just double click in a row that the command will be added at the position of the cursor at the CommandFile frame.

You can view this frame in figure 7.4

Figure 7.3: View of the CommandFile sub frame

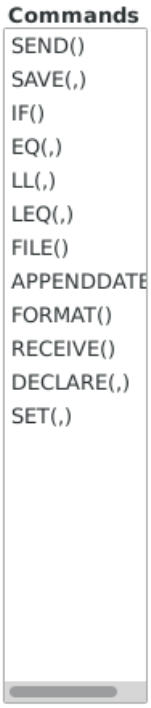

Figure 7.4: View of the Commands sub frame

Chapter 8

Frame: Scripts

In the scripts frame you will be able to add a new script and to change their priority order in the FIFO queue.

The button **New Script** adds a new script to the selected satellite. The new script created is named **new_ script.txt** and it's highly recommended to rename the script for avoiding race conditions while saving the script, causing the lost of the script.

The **UP** and **DOWN** buttons increase/decrease the priority of the selected script in the FIFO queue.

You can view this frame in figure 8.1.

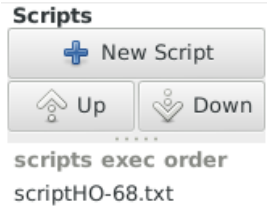

Figure 8.1: View of the Scripts sub frame

8.1 Sub Frame: Scripts Exec Order

The scripts exec order shows the execution order of the scripts. The higher is the script, closer it is of being executed.