Developing a graphical user interface for the real-time ionosphere scintillation monitor

Clément Drouadaine

École Nationale des Sciences Géographiques

National Mapping Authority

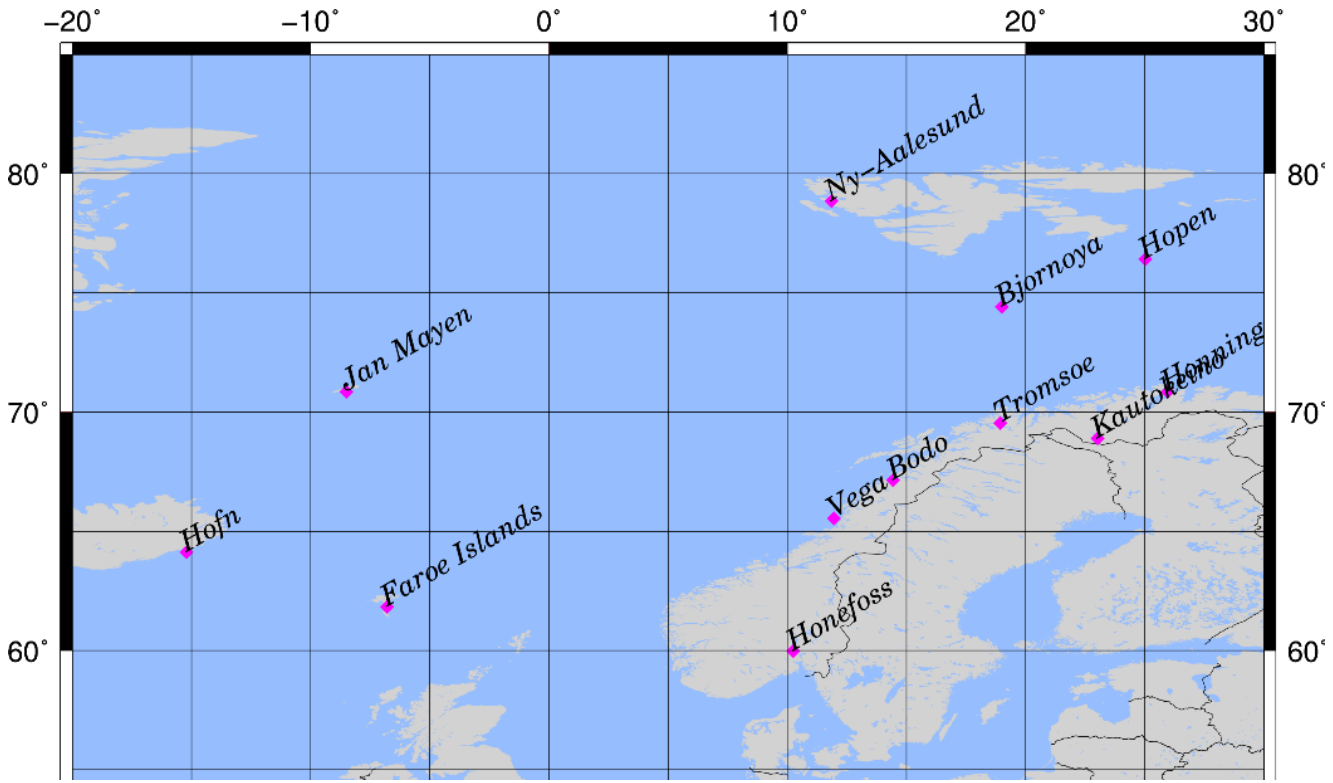
29 September 2015

Overview

1. Brief presentation of the context
2. The architecture
   - Limiting the bandwidth
   - Test again 3
3. Second Section

Description of RTIS

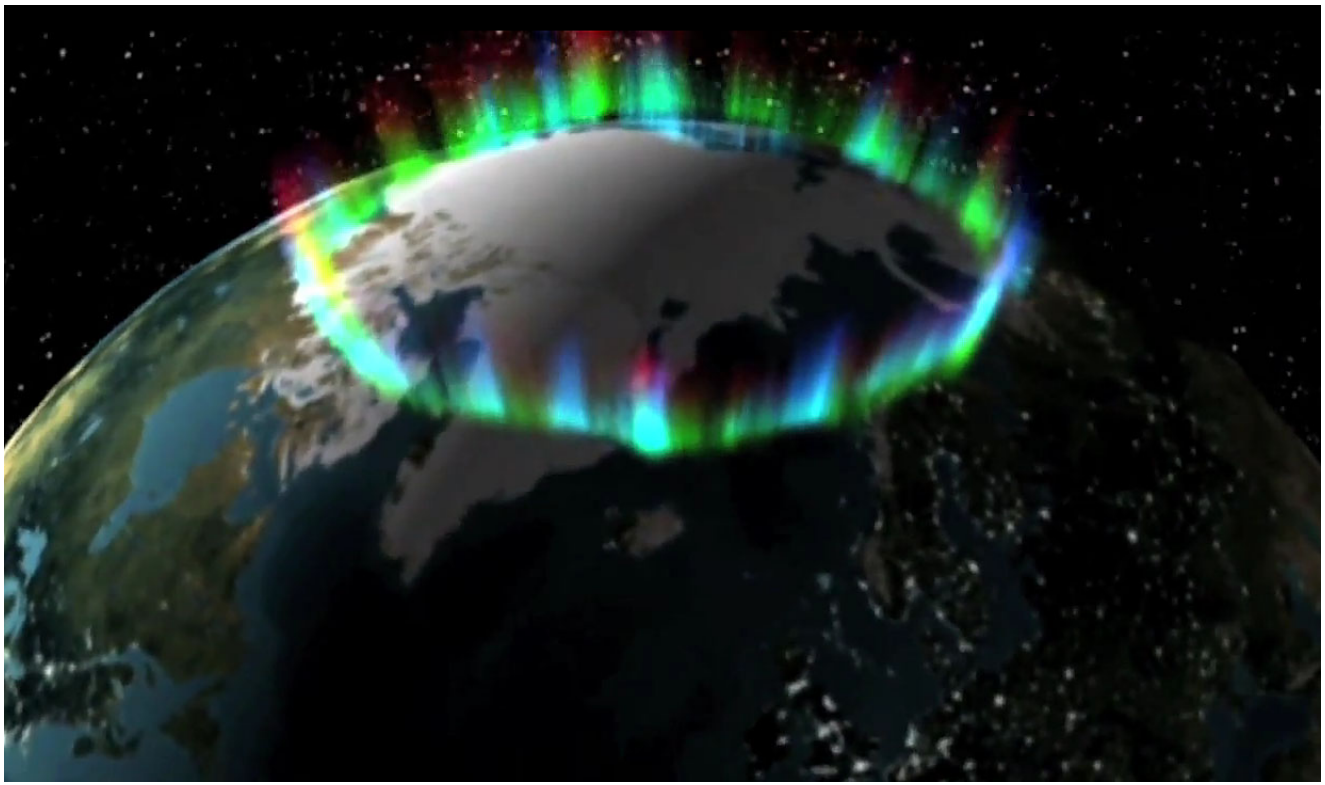
- The Real-Time Ionospheric Scintillation Monitor (RTIS) is a software developed by the Norwegian Mapping Authority (NMA)
- It is deployed in twelve observation stations, spread all across Norway and the Norwegian Sea.



Ionospheric Scintillation

Phenomenon related to solar winds.

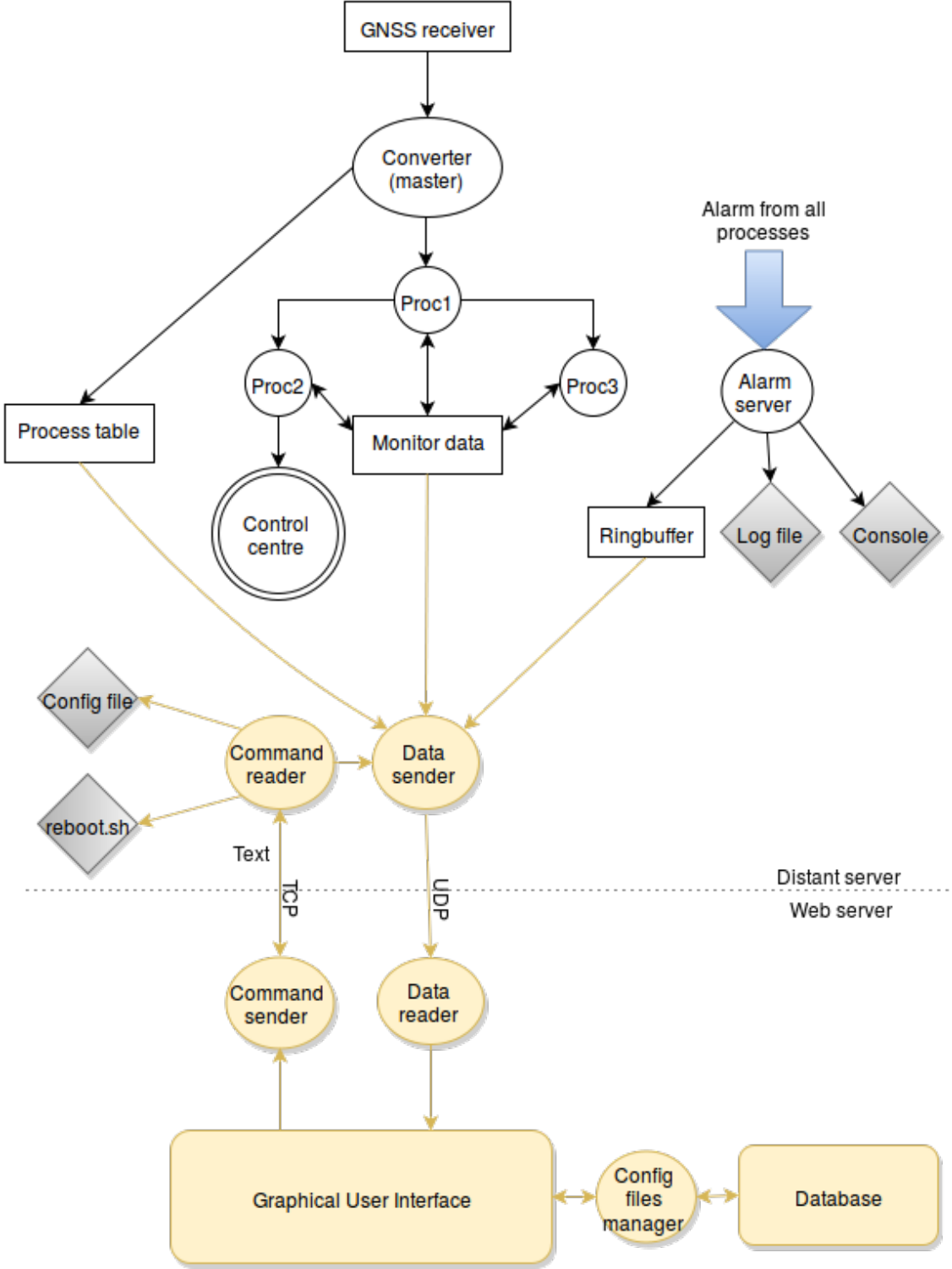
Lessens quality of GNSS observations.



Goal of the internship

Create an interface for monitoring the stations:

- Visualise data
- Get application messages
- Change configuration files
- Reboot the system


GNSS receiver
Converter (master)
Alarm from all processes
Proc1
Proc2
Proc3
Process table
Monitor data
Alarm server
Control centre
Ringbuffer
Log file
Console
Config file
Command reader
Data sender
reboot.sh
Text
TCP
UDP
Distant server
Web server
Command sender
Data reader
Graphical User Interface
Config files manager
Database

Requirements

Low bandwidth on some stations $\Rightarrow$ minimise exchanges.

Blocks of Highlighted Text

Block 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Block 2

Pellentesque sed tellus purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vestibulum quis magna at risus dictum tempor eu vitae velit.

Block 3

Suspendisse tincidunt sagittis gravida. Curabitur condimentum, enim sed venenatis rutrum, ipsum neque consectetur orci, sed blandit justo nisi ac lacus.

Multiple Columns

Heading

1. Statement
2. Explanation
3. Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Table

Treatments	**Response 1**	**Response 2**
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table : Table caption

Theorem

Theorem (Mass–energy equivalence)

$E = mc^2$

Verbatim

Example (Theorem Slide Code)

```
\begin{frame}
\frametitle{Theorem}
\begin{theorem}[Mass--energy equivalence]
$E = mc^2$
\end{theorem}
\end{frame}
```

Figure

Uncomment the code on this slide to include your own image from the same directory as the template .TeX file.

Citation

An example of the `\cite` command to cite within the presentation:

This statement requires citation [**?**].

The End

DEVELOPING A GRAPHICAL USER INTERFACE (GUI) FOR THE REAL-TIME IONOSPHERIC SCINTILLATION MONITOR (RTIS)

C. Drouadaine
ENSG, Marne-la-Vallée - NMA, Hønefoss

What RTIS is

• Due to its high latitude, Norway is a country subject to the effects of solar winds, one of whose most well-known manifestations are aurora borealis. Aside from its aesthetic manifestations, solar weather is also responsible for irregularities in the electron distribution in the ionosphere, that can directly impact the quality of GNSS observations across the country.

• The Real-Time Ionospheric Scintillation monitor software has been developed to measure and quantify this phenomenon, and has been deployed in 12 stations all across continental Norway, Norwegian islands, Faroe Islands and Iceland.

• This work is about creating a graphical user interface to monitor the stations of the RTIS network.

What this work is based upon

• The RTIS software contains an alarm server that can be configured to output alarms and info messages to external devices.

• It also contains a couple of tables accessible from external devices containing data for monitoring a station.

• No graphical interface existed & connections to the stations had to be done via SSH.

Requirements for the solution

• The GUI enables to:
· quickly see the status of every station (i.e. if they are operational or not);
· connect to any one station;
· see any of the monitoring table, the list of active processes on the remote station, and/or the error messages, in near real time.
· visualise the configuration file of the remote station and edit it, replace it by a previous configuration file, or upload a new one;
· reboot the RTIS software on the station;

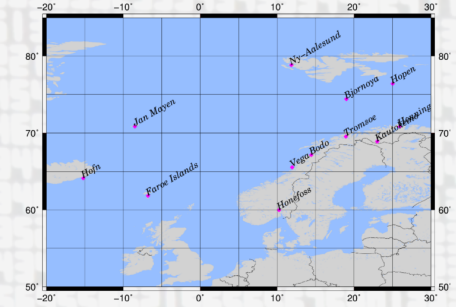
• The frequency of the refresh for the monitoring table and the list of processes are independant and chosen by the user, as is the minimum severity for error messages.

• Given the geographical position of the stations, some of them have to rely on satellite connections to access the internet : the volume of communication between the GUI and the stations is kept to a minimum.

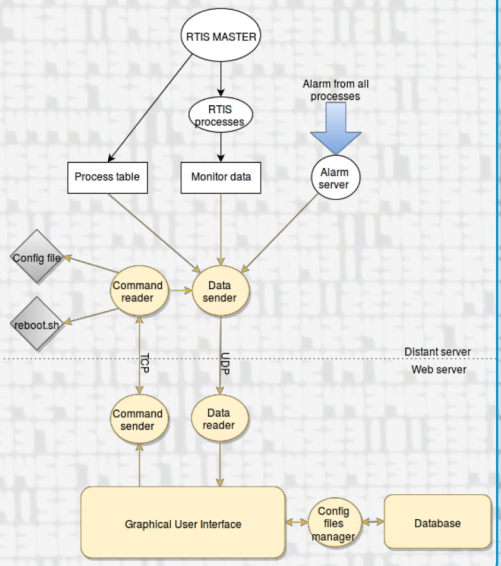
• The GUI can be visited from any kind of device (Linux, Windows, mobile...), and adapts itself to the screen size (responsive design).

• The GUI is developed in PHP, while the communications with the RTIS software & the modules added to RTIS are written in C.

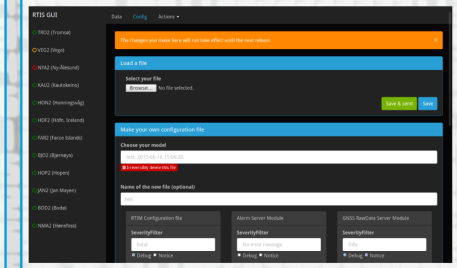
Map of the RTIS stations:



Global architecture of the solution:



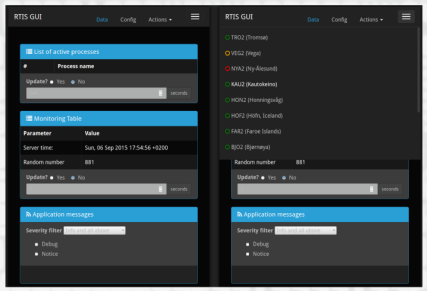
Screenshot of the configuration tab:



On the menu on the left, we have the list of the stations, each of which is assigned a circle whose colour indicates the status of the station. The menu on top is used for navigation within the different actions.

The configuration page itself contains a file upload panel (the upper one), as well as form to manually enter the values (underneath), which is prefilled with the selected file in the dropdown.

Screenshot of the data tab, as seen on a mobile device:



On mobile devices, the "stations" menu, on the left on the previous screenshot, has been minimised in the top right corner, as shown by the left screenshot. On the right one, the menu is deployed.

Transfering data between the GUI and the remote stations

• The commands are sent from the GUI to the client using the TCP protocol and ask for acknowledgement of receipt, to be sure the message has arrived.

• On the other hand, the data going from the client to the GUI is sent via UDP protocol, without any sort of verification, in order to be as fast as possible. The reduction of latency matters here, since we are dealing with real time data.

• All messages exchanged between the GUI and the remote stations are formatted as SATREF messages. This format is a NMA standart which is a highly efficient and lightweight way of packing data.

• User input is treated in Command Senders and sent to the Command Reader of the remote station. Commands can be of different types:
· Get the monitor table data;
· Get the process table;
· Get the error messages;
· Reboot the station;
· Update the configuration file.
When these messages are received by the station, they are obeyed to, and immediately forgotten. For example, to refresh process data every 5 seconds, the GUI sends a message every 5 seconds. The only exception to that are error messages, that are sent continuously, as soon as they are generated by the RTIS software.

• Data from the RTIS software is sent to the Data Readers in the GUI, that are in charge of reading the data and displaying it in the interface.

About the interface: dealing with configuration files

• To prevent unnecessary data exchanges between the GUI and the remote station, the history of configuration files for each station is kept in a database on the web server, which is updated everytime a file is added or edited. It also keeps track of the current active file on every station.

• It is possible to either upload a configuration file directly to the interface, or to enter it via a form.

• The user has access to a dropdown list of the configuration files in the history. By default, the active one is selected. When selecting a file, the form prefills with the values it contains, making it easy to visualise and edit it quickly.

Poster made with Scribus, background image credits to Shervinafshar