

## THE HISPARC CONTROL SYSTEM

R. Hart, B. van Eijk, D. Fokkema, A. de Laat, Nikhef, Amsterdam, Netherlands

### Abstract

The purpose of the HiSPARC project is twofold. First the physics goal: detection of high-energy cosmic rays. Secondly, to offer an educational program in which high school students participate by building their detection station and analysing their data. Approximately 70 high schools, spread over the Netherlands, are participating. Data are centrally stored at Nikhef in Amsterdam.

The detectors, located on the roof of the high schools, are connected by means of a USB interface to a Windows PC, which itself is connected to the high school's network and further on to the public internet. Each station is equipped with GPS providing exact location and accurate timing.

This paper covers the setup, building and usage of the station software. It contains a LabVIEW run-time engine, services for remote control and monitoring, a series of Python scripts and a local buffer. An important task of the station software is to control the dataflow, event building and submission to the central database. Furthermore, several global aspects are described, like the source repository, the station software installer and organization.

direction of the event. The majority of the HiSPARC stations have two detectors. Figure 1 shows a detector as used by a HiSPARC station.



Figure 1: Typical HiSPARC detector.

The ski-boxes are ideal to store the scintillator, because they are weatherproof and relatively cheap. The two (or four) detectors and the GPS device are connected by cables (max. 30 meters) to the HiSPARC II readout and control box. Figure 2 shows the front of it.

### INTRODUCTION

HiSPARC is an acronym for High School Project on Astrophysics Research with Cosmics. It is a cosmic ray air shower detector on a large surface (approximately the size of the Netherlands) with a significant outreach component towards the Dutch high-school physics community. Every high school in the Netherlands may participate. Special education material is available for both junior and senior students. They build their own detector and analyse the data. At Nikhef, the Dutch institute for sub-atomic and particle physics, all data are collected for further analysis to detect air showers.

### GENERAL DESCRIPTION

The basic design of the detector was originally developed at the Radboud University of Nijmegen, but later on the management, maintenance and development was taken over by Nikhef. It consists primarily of a plastic scintillator with a photomultiplier tube (PMT) mounted inside a ski-box carrier and placed on the roof of the high school. At least two ski-boxes, ten meters apart and each equipped with a scintillator and PMT, are required to have a first-order correlation to detect an event. It reduces the amount of noise significantly. In addition a GPS device is installed in between, to provide the location and, even more important, an accurate timestamp of the event. Some stations use four detectors on one roof, which makes it possible to detect the



Figure 2: Front of the HiSPARC II.

The HiSPARC II box is responsible for the data-acquisition (DAQ), the control of the PMTs (high-voltage) and the control and readout of the GPS device. It is connected to a PC by means of USB, using FTDI chips. One of the requirements of the HiSPARC detector was to keep it cheap. A minimum setup (the hardware) costs around US\$ 6,000, consisting of two ski-boxes with scintillator and PMT, a HiSPARC II unit, cables and installation material. Furthermore the software should use only free or open-source software (FOSS). Another requirement is that the readout software has to run on a Windows XP or Windows 7 PC, because most high schools in the Netherlands use Windows. It has to be connected to the school's intranet, which itself should be connected to the public internet. At this time more than 100 HiSPARC stations are active, as shown in Figure 3.

The HiSPARC setup is not limited to the Netherlands. It is expanding to neighbouring countries like Denmark and England.



Figure 3: HiSPARC in the Netherlands.

## READOUT AND CONTROL

The main task of the DAQ software running on the PC is to readout the HiSPARC II unit, store the data locally and send it to the *datastore* at Nikhef. Figure 4 shows the dataflow of a HiSPARC station PC.

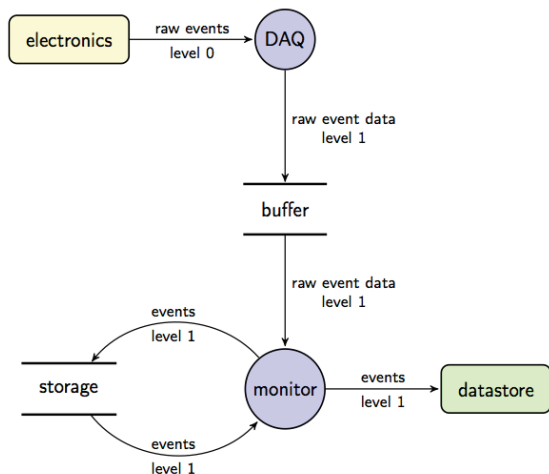


Figure 4: Dataflow of a HiSPARC station.

The DAQ process is a LabVIEW program, called *hisparcdaq*, which communicates with the hardware using the FTDI chipset, which acts as a USB interface. Data are retrieved, sanitized and stored to disk in the *buffer* (implemented as a *MySQL* database). The *monitor* program, a Python script, gets the raw data from the buffer, structures it and puts it into a temporary database

*storage* (implemented as a *SQLite* database). At certain intervals the data from storage is fetched and uploaded over the internet (using the HTTP protocol) to the *datastore* at Nikhef, which itself is implemented as a binary file, using the HDF5 data format.

The hisparcdaq and monitor program are automatically started at a reboot or start-up of the HiSPARC PC. Before the DAQ can be used, the PMT and its ADC has to be calibrated once. The settings are saved and the acquisition can start. Figure 5 shows a typical screenshot of the hisparcdaq program.

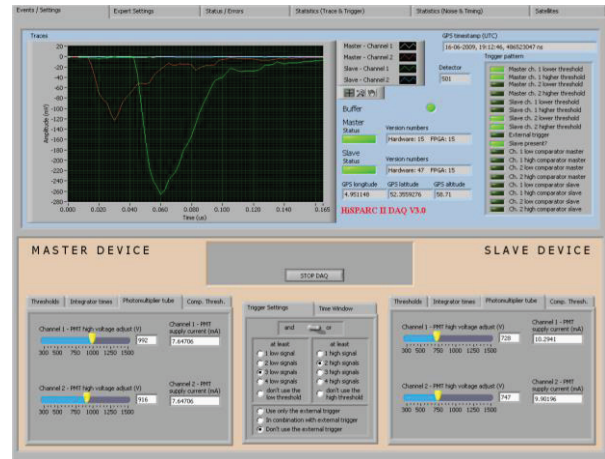


Figure 5: HiSPARC DAQ program.

## MONITORING AND CONTROL

In order to maintain, control and monitor the stations, a *Virtual Private Network* (VPN) is created. For this purpose the *OpenVPN* implementation is used. Two types of connections are defined (see Figure 6):

1. Remote desktop connection between a HiSPARC station and a so-called administrator station. For this purpose *TightVNC* was chosen.
2. Diagnostics data between a station and the VPN server, which runs an instance of the *Nagios* monitoring system.

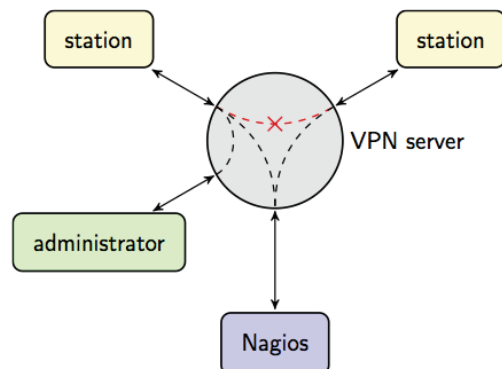


Figure 6: Allowed and forbidden VPN connections.

Service ★★	Status ★★	Last Check ★★	Duration ★★	Attempt ★★	Status Information
Buffer size	OK	02-09-2013 16:17:33	35d 6h 25m 39s	1/6	Buffer DB contains 7 events
CPU Load	OK	02-09-2013 16:22:33	3d 14h 4m 15s	1/6	CPU Load 16% (5 min average)
Drive Space C:	OK	02-09-2013 16:17:33	35d 6h 25m 5s	1/6	c: - total: 232.88 Gb - used: 27.68 Gb (12%) - free 205.20 Gb (88%)
EventRate	OK	02-09-2013 16:26:13	4d 0h 19m 1s	1/3	Event rate for a period of 60.10 seconds is 0.87
LabviewUsage	OK	02-09-2013 16:25:26	0d 6h 11m 22s	1/6	Memory usage: 30.7 Mb
Memory Usage	OK	02-09-2013 16:18:49	35d 6h 27m 59s	1/6	Memory usage: total:2441.01 Mb - used: 578.54 Mb (24%) - free: 1862.47 Mb (76%)
Pulseheight MPV plate 1	OK	02-09-2013 07:10:00	3d 11h 25m 34s	1/3	Pulseheight OK - Fitted MPV is within bounds (-230.0:690.0): 261.2
Pulseheight MPV plate 2	OK	02-09-2013 07:10:00	3d 11h 25m 8s	1/3	Pulseheight OK - Fitted MPV is within bounds (-230.0:690.0): 253.8
Pulseheight MPV plate 3	OK	02-09-2013 07:10:00	3d 11h 45m 10s	1/3	Pulseheight OK - Fitted MPV is within bounds (-230.0:690.0): 287.9
Pulseheight MPV plate 4	OK	02-09-2013 07:10:00	3d 11h 43m 42s	1/3	Pulseheight OK - Fitted MPV is within bounds (-230.0:690.0): 286.4
StorageGrowth	OK	02-09-2013 16:26:14	4d 0h 19m 0s	1/3	Storage growth: 0.866667 Hz
StorageSize	OK	02-09-2013 16:26:12	4d 0h 19m 2s	1/3	Storage size: 82 events
TriggerRate	OK	02-09-2013 16:26:11	4d 0h 19m 3s	1/3	Trigger rate: 0.58. Last update: 16 seconds ago
Uptime	OK	02-09-2013 16:18:18	35d 6h 28m 30s	1/6	System Uptime - 35 day(s) 6 hour(s) 25 minute(s)

Figure 7: Screenshot of the Nagios webpage, showing the status of a particular station.

Stations and administrators use certificates (strongly encrypted) to connect. Every station gets a unique certificate, issued by the HiSPARC support team at Nikhef. It is impossible to make a connection between common HiSPARC stations. Only an administrator station is allowed to connect to a HiSPARC station. For this purpose a remote control software package called TightVNC is used. The Nagios server connects to the VPN server on the local network. The packages OpenVPN, TightVNC and Nagios (all FOSS) are part of the HiSPARC distribution and run after installation as a Windows *service*.

The Nagios monitoring system provides a clear and real-time status overview of the HiSPARC system. At Nikhef a dedicated web-interface (based on the Nagios template) was built to have such functionality. The stations could be shown in total, per cluster or individually. Figure 7 shows a screenshot of the service status details of a particular HiSPARC station. The Nagios server performs many checks. Some are basic, like uptime, disk and memory usage, others are more HiSPARC specific, like the number of events and performance of the hisparcdaq program. The Nagios webpage, presenting the HiSPARC status, is public and can be found at: <http://vpn.hisparc.nl/>.

## INSTALLER

An important requirement of the HiSPARC installer is to make it independent from other packages. This implies that everything had to be put into one installer, including the packages and binaries. Not only the Python source of the monitor program is installed, but Python itself as well.

The HiSPARC installer is generated by and based upon the *Nullsoft Scriptable Install System* (NSIS). It consists of three installers:

1. *adminUpdater*: the administrator installer which requires administrator rights.
2. *userUnpacker*: the user part for which no administrator rights are necessary.
3. *hisparcInstaller*: the overall or main installer which requires administrator rights

The *hisparcInstaller* can be downloaded from HiSPARC's homepage. It contains implicitly the *adminUpdater* and *userUnpacker*. The reason to have three installers is to optimize the distribution of automatic updates as much as possible. The *updater* program (a Python script and like the *monitor* part of the station software) checks at regular intervals the installer repository at Nikhef for newer versions of the *adminUpdater* and *userUnpacker*. If a more recent version is found, it is copied to the local station and in case of the *userUnpacker* automatically executed. In case of the *adminUpdater* a message box will appear indicating an update is available, which should be executed by an administrator.

The *adminUpdater* installs the packages: OpenVPN, TightVNC, Nagios, NI Run-Time Engine, ODBC and the FTDI drivers. Then it sets the services OpenVPN, TightVNC and Nagios to automatic and starts them. Finally it changes a couple of firewall settings by opening some ports.

The *userUnpacker* installs Python and the necessary Python scripts, like the *monitor* and *updater*.

The *hisparcInstaller* needs some information before it continues. It concerns the station number, password and the certificate used by OpenVPN. Figure 8 shows this



initial dialogue. Then it calls the two other installers, creates a default user *hisparc*, a user *admhisparc* with administrator privileges, its own registry and finally it configures the PC so that on start-up the user *hisparc* is logged on and the DAQ processes (i.e. *hisparcdaq*, *monitor* and *updater*) are started.

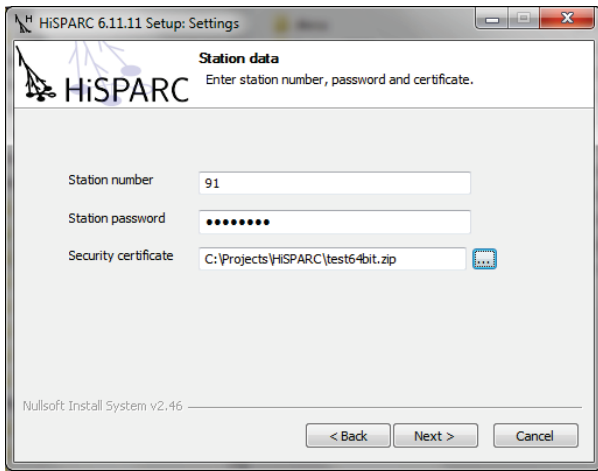


Figure 8: Installer dialogue.

## MANAGEMENT AND CONTROL

The HiSPARC management system, also known as the *HiSPARC public database*, is based on the *Django* open-source framework, a web application framework according to the model-view-controller-model. It contains and provides services like:

1. Administration of stations, passwords, certificates, contact information, etc.
2. Data (histogram) quick looks
3. Analysis results pages

The passwords, certificates (and generation of them) are of course protected. The online histograms are a valuable indication to monitor the data quality of the HiSPARC station. Figure 9 shows an example.

The HiSPARC software, including the sources and documentation, is in itself also FOSS and available on *GitHub*, a cloud repository based on *git* (a distributed version control system). The home page for HiSPARC on GitHub is located at < <http://github.com/HiSPARC/>>.

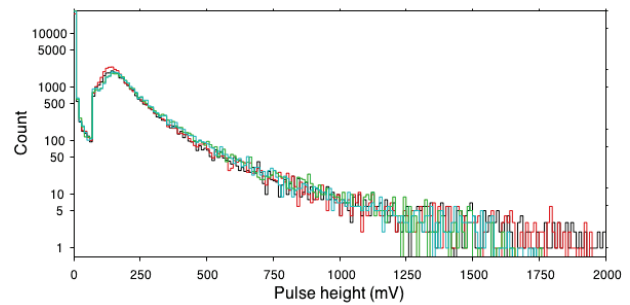


Figure 9: Example of a quick-look histogram.

## FUTURE AND CONCLUSION

The HiSPARC project is now running for almost ten years and is still expanding, especially to the neighbouring countries. HiSPARC is not only a cosmic-ray detector, but could also be extended with other type of detectors, such as: weather station, magnetic field detector and lightning detector. This makes it possible to better explore correlations with the environment. The control system is already prepared for these upgrades. At this moment about a dozen weather stations are up and running.

The HiSPARC control system contains a large number of packages, which makes it complex. The Windows operating system itself is an additional complicating factor. Errors are difficult to understand, because the number of interactions and dependencies between the different packages is large. However, thanks to the excellent remote debug and monitor facilities (VPN and Nagios), the system is well maintainable and the average number of stations, up and running, is more than 90%.

A 64-bit version is expected this year. Other detectors (lightning and magnetic field) are under development. A Linux version is not yet considered.

## REFERENCES

- [1] HiSPARC home page: <<http://www.hisparc.nl>>
- [2] David B.R.A. Fokkema, “The HiSPARC Experiment”, PhD thesis, Nikhef 2012, ISBN 978-90-365-3438-3
- [3] Robert G.K. Hart, “HiSPARC Installer”, implementation report, <[http://docs.hisparc.nl/station-software/doc-dev/\\_static/HiSPARC\\_Installer.pdf](http://docs.hisparc.nl/station-software/doc-dev/_static/HiSPARC_Installer.pdf)>