INTERNAL REPORT

Real time atmospheric monitoring system for the Sardinia Radio Telescope site: A self-built meteorological station.

Francesco Nasir, Franco Buffa, Gian Luigi Deiana (OAC),
Giannina Sanna, Carlo Vadilonga (Università di Cagliari,
Dipartimento di Ingegneria Civile)
e Antonio Vivanet (Geodesia & Tecnologia srl)

Report N. 19, released: 07/08/2012

Reviewer: I. Porceddu

OAC Osservatorio Astronomico di Cagliari

1 Introduction

It is known that radioastronomical observations are strongly affected by atmospheric conditions, especially water vapor, liquid droplets present in clouds and winds [3]. The Sardinia Radio Telescope (SRT), which observes in the 0.3-100 GHz frequency range, will be backed-up by an atmospheric monitoring system (AMS) with particular care given to on-site atmospheric transparency conditions.

The AMS for SRT will integrate different types of sensors and will make use of atmospheric models developed for this application. Through a project supported by the regional Sardinian government (RAS) we are developing an integrated water vapor (IWV[1]) monitoring system which will be able to accomplish two important tasks: firstly perform two dimensional IWV tomographic measurements over the Sardinian region and in particular over the SRT site and secondly perform such measurements in real time.

With this aim, we are taking advantage of a preexisting infrastructure named SARNET, which is managed by the private company: Geodesia & Tecnologia and the University of Cagliari's Engineering Department. SARNET consists of a network of geodetic GPS stations which enable the user to have millimetric GPS precision anywhere in Sardinia: more details will be given in the next section. Furthermore, our task is to install a surface meteorological station next to each geodetic GPS SARNET station. In fact it is renown from literature that if the stations positions are known with precision, as in our case, then GPS measurements coupled to surface meteorological measurements can be used to infer the IWV content in the portion of atmosphere directly above [1].

Furthermore, if there are many geodetic GPS-meteorological stations spread out on a large territory, through interpolation it is possible to obtain a two dimensional IWV tomography over the region of interest [6]. This information is important because it permits to monitor the transit of water vapor over a large region and detect the movement of weather fronts which are strongly characterized by high water vapor content.

Our set of meteorological stations is constituted by Vaisala off-the-shelf stations and by our own self-built stations. The reason for constructing some of the meteorological stations by ourselves is that off-the-shelf meteorological stations are very expensive therefore we have decided to reduce costs greatly by buying the sensors (e.g., barometer, temperature and humidity sensors) separately and assembling them, including the microcomputer that will read sensor data and save it and send it via a serial port through a telnet/ssh connection protocol. Furthermore, by utilizing the SARNET network, costs are highly reduced: all the framework already exists, we only have to position the meteorological stations and resolve some software issues.

This report focuses especially on the procedure of building our own meteorological stations: we firstly explain the preexisting SARNET project, then we briefly describe how IWV estimates are obtained from geodetic GPS and surface meteorological station measurements. We then explain how we assembled and programmed our self-made meteorological stations. Next we illustrate how the data is collected and sent from the stations to a central serve, finally some conclusions are made.

1 The total amount of water vapor integrated from the surface to the top of the atmosphere, usually measured in mm or equivalently in g/cm^2.

2 SARNET

SARNET is a project managed by the private company: Geodesia & Tecnologia. It comprises 14 GPS stations which have been more or less equally spaced over the whole Sardinian region. The GPS stations send data to the central server via internet or cell phone connection. The user of the SARNET system may connect his rover GPS to the central server through a GPRS or UMTS link, and send his position. The central server generates a virtual reference station (VRS) optimized with respect to the area in which the user is located. The server uses the VRS position in order to create an extremely precise RTCM[2] signal, the user receives the correction data on his real time kinematic receiver.
Extremely precise positioning with the use of a simple non-geodetic receiver has many applications, amongst which topography, land registry and public housing vigilance.

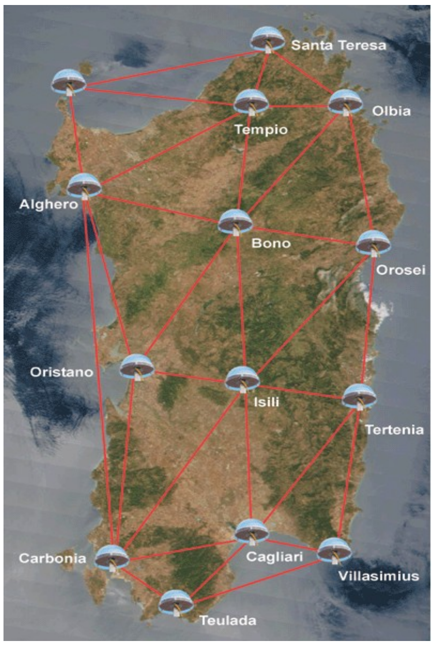

Figure 1: (left) The SARNET network stations located in various cities of Sardinia (ITALY). (right) A trimble geodetic GPS receiver placed in one of the locations one the map next to a Vaisala meteorological surface station.

In order to implement a tomographic IWV monitoring system over a large region we must take advantage of the preexisting SARNET network, this represents the starting point.

3 Retrieving IWV with a GPS and a surface meteorological station

As mentioned before, when a geodetic GPS receiver is coupled to a surface meteorological station, it is possible to measure the IWV present in the atmospheric layer straight above the station itself. In fact if the position of the GPS station is already known, one can fix this information and use the GPS measurements to infer the excess of optical path due to refractivity in the zenith direction, also known as zenith total delay (ZTD). In addition to this through meteorological surface measurements it is possible to separate the excess of optical path in the dry component and the wet one, the latter is directly proportional to IWV.

---

2 Radio Technical Commission for Maritime Services.

Usually zenith tropospheric delay ZTD, is subdivided in the hydrostatic dry component ZHD and in the wet component ZWD [1]:

$$ZTD = ZHD + ZWD = 10^{-6} \int_s N ds;$$

where $N = 10^{-6}(n-1)$ represents refractivity and \$n\$ is the refraction index. The hydrostatic component is given by the next equation:

$$ZHD = 10^{-6} k_1 \int \frac{P_h}{T} ds;$$

where P_h is the partial pressure of dry air. The wet component of the tropospheric delay [2] is given:

$$ZWD = 10^{-6}\left(k_2 + \frac{k_3}{T_m}\right) \int_s \frac{P_w}{T_m} ds;$$

where k_1 , k_2 and k_3 are refractivity constants, P_w is the water vapor partial pressure and

$$T_m = \frac{\int_s \frac{P_w}{T} ds}{\int_s \frac{P_w}{T} ds};$$

is the mean atmospheric temperature. To implement the previous relationships, surface temperature and pressure are needed, they can be obtained with a meteorological station. Furthermore, T_{mr} can be recovered through an empirical relationship which uses surface temperature and T_m .

$$ZWD = \Pi \cdot IWV;$$

where

$$\Pi = 10^{-6}\left(k_2 + \frac{k_3}{T_m}\right) R;$$

and R is the universal constant of gasses. As mentioned before geodetic GPS can measure ZTD, to find ZWD we must measure ZHD and T_m independently, this can also be done with good approximation with empirical models taken from literature and a ground meteorological station positioned close to the geodetic GPS receiver.

In our case we have many stations that can measure IWV content directly above them (point measurements), therefore by combining this data and interpolating in regions of space between stations one can obtain real time IWV tomographies (fig. 2).

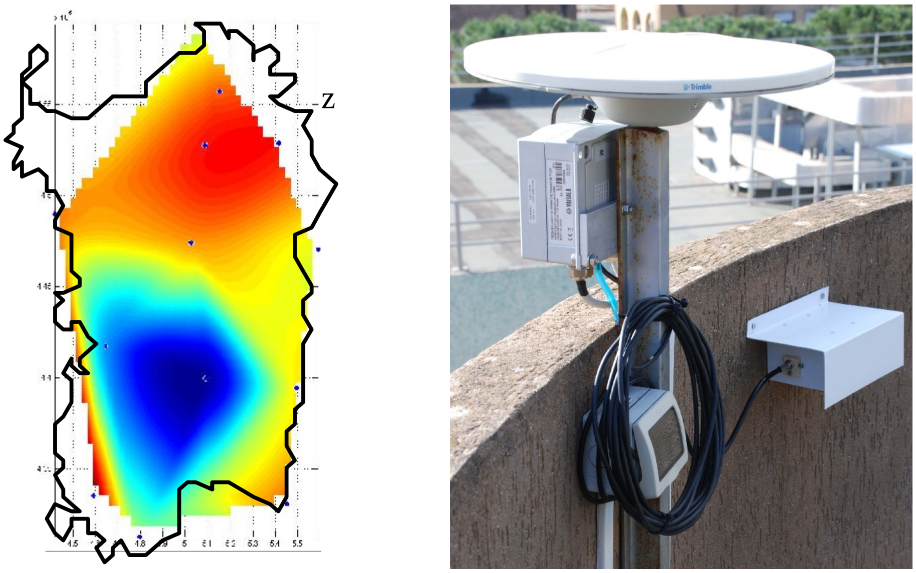

Figure 2: (left) A snapshot of zenith tropospheric delays over Sardinia as measured by SARNET, blue are as indicated higher values. Data is corrected for orographic effects. (right) GPS and surface meteorological station.

4 Instrumentation

We have purchased Vaisaila off-the-shelf surface meteorological stations, they are very reliable but are expensive. In order to reduce costs greatly, we have designed and assembled our self-made stations that mime the exact functioning of the Vaisala stations. In the following sections we first describe a Vaisala station then we describe the fundamental components of our self-made meteorological station starting from the micro-computer FoxBoardG20.

4.1 Off-the-shelf meteorological station

The off-the-shelf meteorological stations which are being used are Vaisala PTU300. These stations comprise of pressure, humidity and temperature sensors.

Figure 3: Vaisala surface meteorological stations.

The pressure sensor is a Barocap Vaisala sensor, it has an accuracy of 0.1 hPa, the relative humidity sensor is a Humicap with and accuracy of 1% and the temperature sensor has an accuracy of 0.2 °C. The stations have also an RS-232 interface and a WAN/LAN interface.

4.2 Our self-built meteorological station

For our own meteorological station we had to acquire the components and sensors and assemble them. Furthermore, code had to be written in order to manage sensor data, all this is described in the following sections.

4.2.1 Fox Board G20

The controller of the meteorological station is the FoxBoardG20, a single board computer built around the ARM9@400MHz Atmel CPU AT91SAM9G20.
The computer has a 400 MHz CPU and a 64 MB RAM, and a default 2GB flash memory microSD. It also has various connectors as indicated in fig. 4. We used the micro-computer to read, write, control and send sensor data through a TCP/IP connection and/or a serial one.

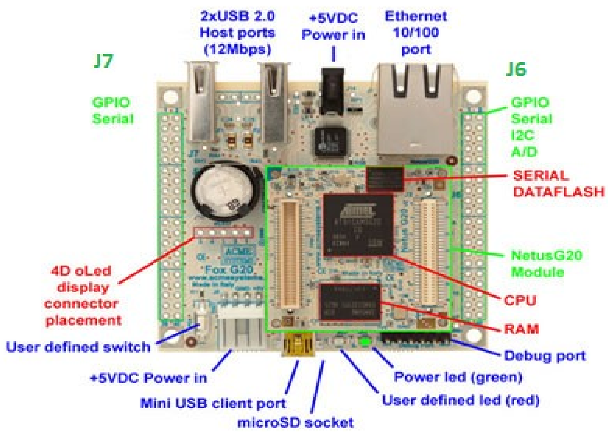

Figure 4: Schematic picture of the FoxBoardG20 and all its components and connectors, the dimensions are roughly 6x6 cm.

The connections that were used were the GPIO (General Purpouse Input/Output) connections for the humidity and temperature sensor, in particular we used the left side pins:

- the J6.2 for the power supply (3V);
- the J6.17 for the serial data line;
- the J6.18 for the serial clock line which is used for synchronization purposes.

We connected the barometer to the FoxBoardG20 through a serial to USB adaptor: the second USB 2.0 port starting from the left in fig. 4. The barometer has its own power adaptor, nevertheless in order to reduce the use of space we decided to power the barometer directly from the FoxBoardG20. Unfortunately the FoxBoardG20 can only power up to 5V but the barometer needs at least 7V.

To overcome this problem we built a mini power transformer that connects to the barometer and to the following FoxBoardG20 GPIO pins:

- the J7.1 ground pin;
- the J7.29 5V pin.

We also used the debug, console port by using a serial to mini USB adapter in order to enable connections to an external desktop computer for initial configurations.
Once the initial configurations were performed (in particular setting the static ip on the microcomputer), connections to external computers where possible through Ethernet 10/100.
In addition to this we also used the second FoxBoardG20 USB port to connect to an external desktop computer in order for the two devices to exchange data serially.

The following list illustrates all the connectors which were attached to the FoxBoardG20:

- Serial to USB connector used for the barometer;
- Serial to USB console port used for initial configuration of the FoxBoardG20;
- Power supply for the FoxBoardG20;
- I_2C (data, clock, power, ground) that connect the humidity and temperature sensor to the FoxBoard GPIO pins.
- USB to serial adapter for communication with external computer/GPS;
- Ethernet RJ45 cable used by the FoxBoardG20 to communicate via internet with other computers and networks.

4.2.2 Barometer sensor

The barometer sensor that was used for the meteorological station was a YOUNG barometric pressure sensor, it measures pressure in the range 500-1000 hPa, its accuracy is roughly 0.2 hPa and it outputs data with a frequency that goes from 1.8 Hz to 1 per minute. The voltage supply is 7 to 30 VDC.

Figure 5: Young barometric pressure sensor

Data is sent to the micro-computer through a serial connection, simply using the receive, send, the voltage and ground pins. Furthermore, the serial connection is in turn linked to a serial to USB adaptor in order to be connected to the FoxBoardG20 USB port. While the power supply connections are linked to the FoxBoardG20 micro-computer by using a transformer to raise the voltage output needed by the barometer (fig. 6).

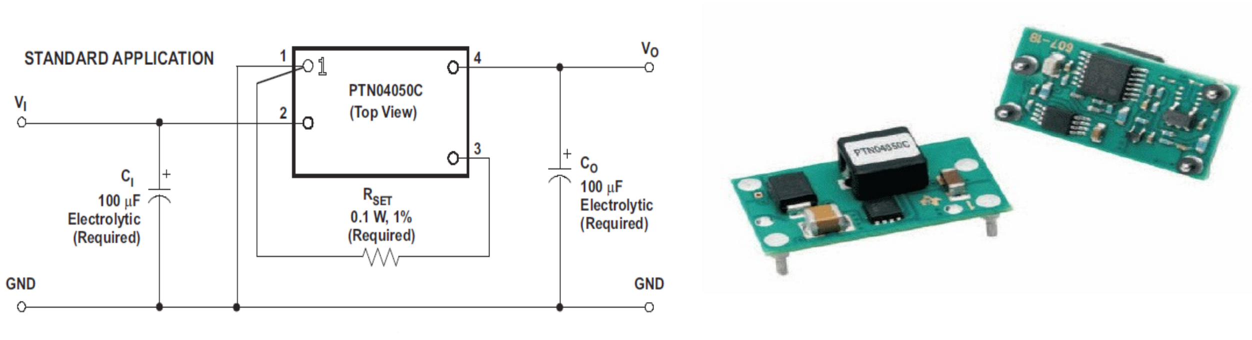

Figure 6: (left) Electric circuit of the barometer power supply adaptor, the two capacitors are needed for stabilization purposes while the PTN04050C is a power transformer used together with the resistance to change the FoxBoardG20 3V GPIO output to the at least 9V required by the barometer (right) photograph of the adaptor.

The circuit developed for the barometer's power supply can be seen in fig. 6 (left). The resistance is used to obtain the right voltage with the Texas Instruments power transformer PTN04050C that can be viewed in fig. 6 (right) [5].

4.2.3 Temperature and Humidity sensor

We have used a Sensirion SH10 humidity and temperature sensor. The sensor is connected to the FoxBoard G20 GPIO (general purpose input/output interface) through a digital 2 wire interface I_2C type connection.
The voltage input of the sensor is 3V and it performs 1 measurement per second, the relative humidity operating range is 0 – 100%, and the temperature one is -40 to +125°C [4].

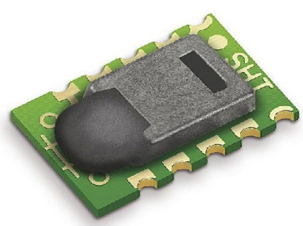

Figure 7: Sensirion temperature and humidity sensor.

There are 4 connectors following the I_2C protocol:

- the power connection;
- the GND;
- the serial clock line (SLC);
- the serial data line (SDL);

The humidity and temperature accuracy is ± 3 % and 0.5 °C respectively.

5 Initial configurations

On the FoxBoardG20 microSD there is a Debian6.0 GNU/Linux operating system with a 2.6.38 kernel version. The disk is partitioned as follows: 24 MB relative to FAT16 file system dedicated to the boot partition, 1.8 GB dedicated to the system partition and 1.2 GB dedicated to the swap partition.

In order to initially access the FoxBoardG20 we connected it to an external desktop computer through a serial connection. The terminal emulator used on the external desktop computer was minicom. The following settings were used:

- baud rate 115200;
- serial device ttyUSBX (X is a generic number that depends on which USB port is used)
- Hardware Flow Control: No;
- Software Flow Control: No;
- parity: none;
- databits: 8.

Initially one may log on the FoxBoardG20 through user:root and password: netusg20. Once the first login has been accomplished than it is possible to change settings. We set a static ip address: 192.168.1.90 , netmask: 255.255.255.0, gateway: 192.168.1.1, this is done by accessing the file "/etc/network/interfaces". By setting a static ip, the next time we were able to login through an ethernet cable RJ45, which has a much faster data rate transfer than a serial connection. We also changed the default password, this was simply done by logging in as root and typing the "passwd" command which enables us to change password.

It is important in order to access the micro-computer to enable services such as telnet and ssh, this is done simply by typing the following command in the terminal: "chkconfig --level 2345 ssh on"

For security reasons it was important to limit the network access to the FoxBoardG20. For this scope we used the gnu/linux firewall iptables which is managed by a firewall rules file in which ports, domains and ip addresses are denied or allowed accordingly. In particular we closed all access to the FoxBoardG20 except for port 22 which corresponds to the ssh service, we also decided to permit access only from very few networks. The file is then managed through the "/etc/init.d" directory so that it may be run at boot.
See appendix A for details.

6 Programmes and Methods

In order to acquire meteorological data from the sensors we wrote some scripts which reside on the FoxBoardG20 linux/debian operating system's root directory. The scripts collect data from the barometer and from the humidity and temperature sensors and write them in appropriate files. Furthermore another program may read the data from the files and send it remotely to a central server through a ssh/telnet connection or to the GPS sensors through a serial connection which in turn will send it to the central server.
The main driver is *drv_meteostat*, it is a bash script that starts up all the other processes. When the script is run, it kills all previous processes related to it and starts them up again. This is done through a

function called *kill_zombie* which is defined in the script itself. Secondly *par2* is launched in the background: it is a C program that executes a loop and spawns a child process called *barometer*, a C program, which listens up for data originated from the barometer and sent to the FoxBoard through a serial to USB connection. Once the data has been received *par2* writes it on an ASCII file named *barometer.txt*.

After this, *drv_meteostat* runs a program called *use_sensirion*, a C program which runs a loop which launches another C program called $sensirion$, the latter has the job to collect data from the humidity and temperature sensor which is connected to the FoxBoardG20 through GPIO pins. The program *use_sensirion* then writes the data in an ASCII file called *temp_hum.txt.*

The next programs that are run are either *portServer.py*, a python script program which also runs a loop in the background, it collects the ASCII meteorological data which resides in the two text files, it formats it accordingly and sends it to a specific network port in order to be sent to a central server by using telnet or ssh protocols.

In actuality the meteo-GPS station might need to work in a different way: the GPS receiver software will pole the meteorological station in order to receive from it the meteorological data in NMEA[3] format, then it will send the GPS and meteorological data to a central server. The two receivers are connected to each other via serial connection. In this case the program used is *serialMet.py* a python program that responds to a polling signal by reading the meteorological data in the ASCII files and sends i via serial connection to the GPS.

Once *drv_meteostat* has finished running it has launched the programs described above which continue to run in the background in a continuous manner until further notice.

In order to make sure that these programs are up at boot time, *drv_meteostat* will be placed in the "/usr/sbin" directory and it will have a reference start up file with the same name in the "/etc/init.d" file so that it is started automatically at boot time. The latter is formatted taking as an example from the "/etc/init.d/skeleton" file. Finally one must run the "update-rc.d scriptame defaults" command in order to create the correspondent kill or start soft link files in the "/etc/init.d/rcx.d" runlevel directories.

3 National Marine Electronics Association protocol.

The following flow chart illustrates the procedure.

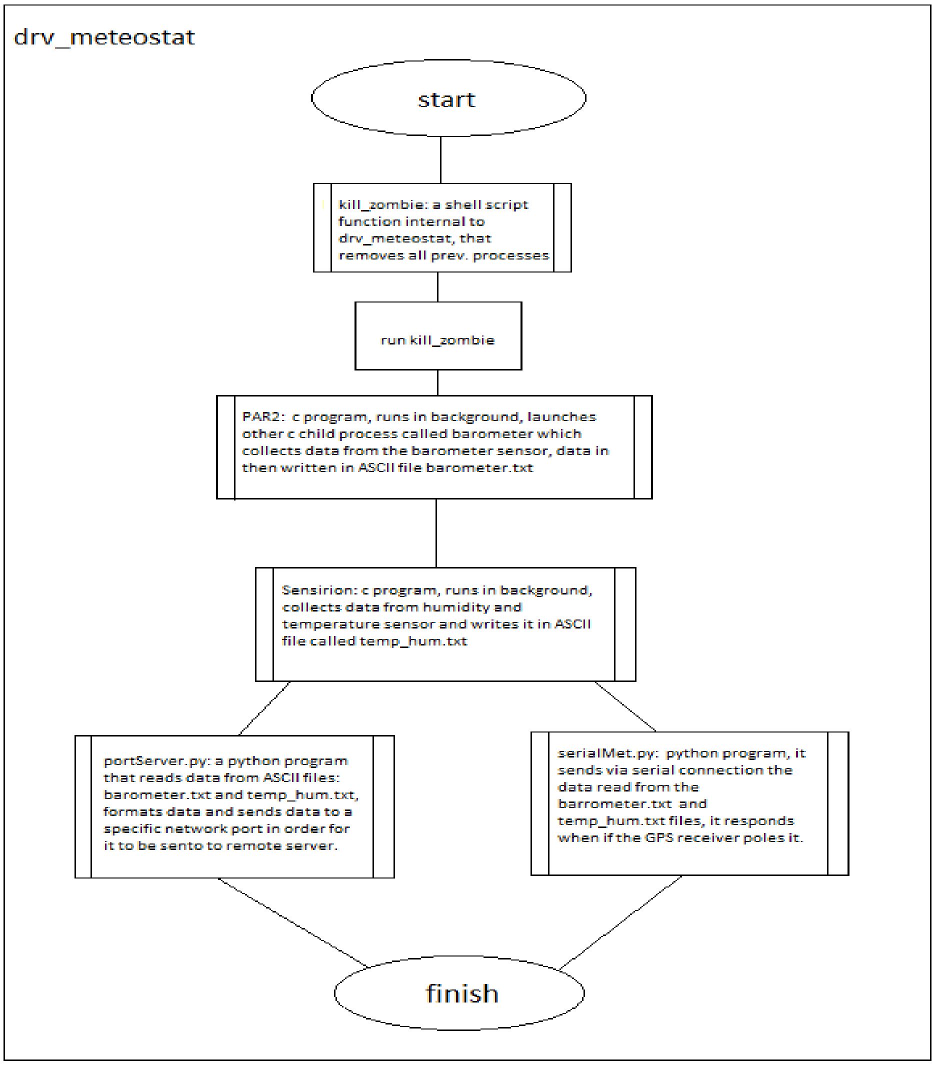

Figure 8: Flow chart of the code used by the FoxBoardG20 in order to collect data from barometer, humidity and temperature sensors and send it to a central server or to another receiver via a TCP/IP network connection or via a serial connection; par2.c, sensirion.c, portserver.py and serialMet.py are programs which run a forever loop.

7 Conclusions

In this report we illustrated the preexisting SARNET project and how the latter can be exploited in order to create a two dimensional Integrated water vapor monitoring system. To do so it is necessary to position surface meteorological stations next to the SARNET GPS stations, furthermore in order to reduce costs we constructed our own meteorological station starting from the single components such

as barometer, humidity and temperature sensors and micro-computer.
The components was connected to the micro-computer FoxBoardG20, a package of scripts and programs was written in order to collect automatically data from the sensors and write it on the disk and send it to the GPS sensor through a serial connection or send it to the central server via network.
In future we want to build a sufficient number of meteorological stations in order to utilize the whole SARNET network and have a full IWV tomography that covers all of Sardinia. This will help monitor water vapor and observe the movement of weather fronts across the region. In last analysis this will also be useful for scheduling radioastronomical observations at the Sardinia Radio Telescope site when the weather is most suited for the experiment that is being conducted.

Appendix A

The FoxbBoardG20 has a debian operating system, which implements the iptables firewall. We created an iptables rule script file in order to drop and block most types of network connections except the ones which were strictly necessary (fig.9).

```
$IPT -P OUTPUT ACCEPT
$IPT -P INPUT DROP
$IPT -P FORWARD DROP

# allow unlimited traffic on loopback

$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT

# allow all incoming/outgoing SSH

$IPT -A INPUT -i eth0 -p tcp -s 192.168.1.0/24 --dport 22 -m state --state NEW,ESTABLISHED -j
ACCEPT
$IPT -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# allow all incoming/outgoing telnet (outgoing from non standard 2323 port)

$IPT -A INPUT -i eth0 -p tcp -s 192.168.1.0/24 --dport 2323 -m state --state NEW,ESTABLISHED
-j ACCEPT
$IPT -A OUTPUT -o eth0 -p tcp --sport 23 -m state --state ESTABLISHED -j ACCEPT
```

Figure 9: The iptables firewall rule file that we used for the FoxBoardG20.

In order for the rules script file be loaded automatically we created a start up file in the “init.d” directory named myIpRules starting from the template skeleton file. The script file (fig. 9) is also named myIpRules, was placed in the “/usr/sbin” directory and its mode was changed to executable. We than run the “update-rc.d myIpRules defaults” command in order to automatically create the soft link start up and kill files that go in the rc0.d to rc6.d runlevel directories.

Acronyms

AMS: Atmospheric Monitoring System

GPIO: General Purpose Input Output

GPRS: General Packet Radio Service

GPS: Global Positioning System

GND: Ground

IWV: Integrated Water Vapor

PTU: Pressione Temperatura Umidità

SRT: Sardinia Radio Telescope

RAS: Regione Autonoma Sardegna

RTCM: Radio Technical Commission for Maritime Services.

SDL: Serial Data Line

SCL: Serial Clock Line

SRT: Sardinia Radio Telescope

UMTS: Universal Mobile Communication System

VRS: Virtual Reference Station

ZTD: Zenith Total Delay

ZHD: Zenit Hydrostatic Delay

ZWD: Zenith Wet Delay

Bibliography

[1] Bevis M., Businger S., Chiswell S., Herring T., Anthes R., Richard A., Rocken C., Ware R. H.: “GPS Meteorology: Mapping Zenith Wet Delays onto Precipitable Water”. Journal of Applied Meteorology. 33, 379 (1994).

[2] Mendes V. B., Prates D., Santos L., Langley R. B.: “An Evaluation of Models for the Determination of Mean Weighed Temperature of the Atmosphere”. Proc. of the 2000 National Technical Meeting. Institute of Navigation, USA, 443, (2000).

[3] Nasir, F. T., Buffa, F., Deiana, G. L.: “Characterization of the Atmosphere above a site for millimeter wave astronomy”. Exp. Astron. 29(3), 207 (2011).

[4] Sensirion, the sensor company, 2012: Digital Humidity and Temperature Sensors (RH\&T) - Overview.
http://www.sensirion.com/en/01_humidity_sensors/, last accessed 06/05/2012.

[5] TexasInstruments, 2012: PTN04050C Datasheet.
http://www.ti.com/product/ptn04050c, last accessed 06/05/2012.

[6] Xu, W. B., Li, Z.,W., Ding, X., L., Zhu, J., J.: “Interpolating atmospheric water vapor delay by incorporating terrain elevation information”. J. Geod. 85, 555 (2011).

"This project has been supported by the regional Sardinian government law L.R. 7 - 2007"