

# A sensor network for Glaciers

Kirk Martinez, Alistair Riddoch, School of Electronics and Computer Science. University of Southampton,  
UK

Jane Hart, School of Geography, University of Southampton, UK

Royan Ong, Department of Engineering, University of Leicester, UK.

**Abstract** — This paper reports on the design, implementation and results from GlacsWeb, an environmental sensor network for glaciers installed in Summer 2004 at Briksdalsbreen, Norway. The importance of power control, hardware architecture and communication systems are discussed and research issues highlighted.

**Keywords** - *sensor network, radio communications, low power, environmental monitoring, glaciology.*

## I. INTRODUCTION

The continuous advancements in wireless networks and miniaturisation have made the deployment of sensor networks to monitor the environment increasingly feasible. The potential advances to environmental sciences could be as great as the revolution produced by the development of remote sensing during the 1970's. Sensor network technology can provide basic scientific data as well as hazard warning. This is particularly important in remote or hazardous environments where many fundamental processes have never been studied due to inaccessibility. In addition, more accessible environments could be monitored on an unprecedented scale.

Chong and Kumar [1] illustrated that sensor networks combine state of the art technologies from the fields of sensors, communications and computer science. We have suggested [2] that in an environmental sensor network, a holistic design is required including specialist user-domain knowledge. A sensor network comprises sensor nodes which gather data autonomously and usually pass it to one or more *Base Stations*, which forward the data to a Sensor Network Server (SNS).

Due to the innovative nature of the technology, there are currently very few environmental sensor networks in operation. These networks, which includes NASA/JPL's project in Antarctica [3], and Huntington Gardens [4], Berkeley's habitat modelling at Great Duck Island [5], the CORIE project which studies the Columbia river estuary [6], deserts [7], volcanoes [8] as well as the GlacsWeb project discussed below, monitor a variety

of environments. These projects have demonstrated the value of using environmental sensor networks for environmental studies: primarily as they are cheap, reasonably easily configured and installed, inherently redundant, and have the potential to produce data of high spatial resolution. It is foreseeable that this technology could become pervasive and expand to the point where information from numerous networks (e.g. glacier, river, rainfall, avalanche and oceanic networks) are aggregated at higher levels to form a picture of a system at resolutions that are unfeasible or unattainable with current technology.

We are considering a future where there will be large scale repeated deployments of different types of environmental sensor networks around the world and they will become truly pervasive. To move towards this, fundamental issues relating to power, communications, deployment, cost, maintenance and security need to be resolved. Separate sensor networks monitoring different aspects of the environment will require a simple way of automatically interchanging data. We envisage using web services and semantic web technology to expose the data to other sensor networks and the software which derives information from sensor networks. This would enable a more complete view of an environment. In the case of an ice cap, data could be gathered from very different systems: fixed weather stations, fixed river flow stations, mobile sensors in snow, ice or sediment and so on. This should be manageable even though the sub-networks are from different vendors. It should also be possible to retrieve summary data rather than process raw temperature data for example max/min could be all that is required. A relatively small number of nodes might produce a summary of one data stream such as average ice melt per day. While researchers may need in depth data access tourists for example may be interested in accessing more general information about the area they are visiting.

In order to maximize data efficiency from nodes (where a vast amount of data may be potentially supplied), domain knowledge of the environment combined with the generation of specific algorithms will need to be generated. The production of this type of sensor networks is thus an interdisciplinary project.

The GlacsWeb project described here has deliberately focussed on producing a robust platform for one particular environmental area. The lessons we have learnt in two deployments have led us to adapt our architecture and philosophy. The main advances have been in hardware design, which will be described in more detail. We have maintained a general approach throughout with the objective of generating useful designs for a wider range of applications.

## II. THE GLASWEB PROJECT

The aim of the Glacsweb project is to build an environmental sensor network to understand glacier dynamics in response to climate change. Glacier behaviour is moderated by local conditions, in particular the nature of the bed. Currently 90% of the discharge from Antarctica is from the ice streams, whose behaviour is controlled by subglacial processes and not climate. Briksdalsbreen in Norway was chosen as an excellent analogue for Antarctica, as it has similar bed conditions, but is far more accessible, and so allows the development and testing of the first glacial environmental sensor network (Figure 1).

The intention of the environmental sensor network was to collect data from sensor nodes (*Probes*) within the ice and the till (subglacial sediment) without the use of wires which might disturb the environment. The system was also designed to collect data from the surface of the glacier (position, weather). Finally all the data would be combined in a database on the *Sensor Network Server* together with large scale data from maps and satellites. In this way, specific data from the sensor nodes (which reflect point data) are combined with larger scale data to understand the glacier as a whole. The data was collected from within the ice by radio communications (which has never been achieved before). These requirements led to the following research objectives: i) miniaturisation; ii) low power design, iii) ad-hoc networking, iv) autonomous and adaptive behaviour.



Figure 1. Briksdalsbreen with Field site location indicated with arrow

The GlacsWeb system is composed of *Probes* embedded in the ice and till, a *Base Station* on the ice surface, a *Reference Station* (2.5 km from the glacier with mains electricity), and the *Sensor Network Server* (SNS) based in Southampton (Figure 2).

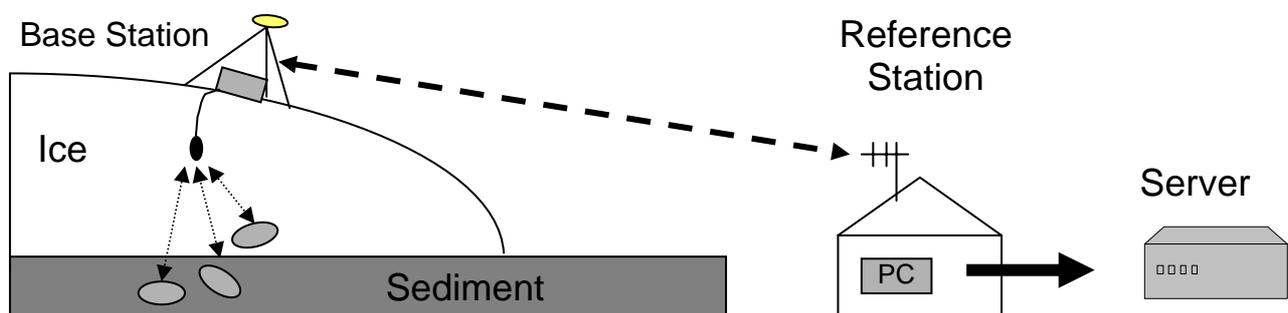


Figure 2. Simplified System overview

Design of the prototype system began in late 2002 and a prototype was installed in the glacier during the summer of 2003. The *Probes* were embedded via holes drilled into the glacier with a hot water drill. In 2003 nine *Probes* were inserted. These *Probes* communicated with the *Base Station* at 868MHz, a licence-free channel that was chosen with consideration for theoretical losses within pure ice and the size of the *Probe*.

However, the amount of water within a temperate glacier such as Briksdalsbreen meant that radio losses were significantly more than anticipated; hence only one *Probe* transmitted data for 10 days [2, 9]. Nevertheless, this showed that the system was feasible with more powerful communication modules.

During 2004 the *Probes* and *Base Station* were redesigned based on the experience of the previous year. Eight *Probes* were embedded in summer 2004 which immediately transmitted their data. The details of the new system and the results are discussed below.

### III. SYSTEM ARCHITECTURE VERSION 2

#### A. *Probes*

The electronics and sensors are enclosed in a Polyester egg-shape capsule measuring 148 x 68mm (Figure 3). Its two halves are permanently bonded with cyanoacrylate and epoxy resin. The *Probes* were made as small as possible to behave like rounded natural stones, which also simplified insertion in the ice. Each *Probe* is equipped with one pressure, temperature and resistivity sensor, two accelerometers (orientation in three dimensions) and a strain gauge. These sensors were glued and potted into the *Probe* to prevent water intrusion. The *Probe*'s electronics were mounted on three octagonal PCBs – one each for the digital, analogue and radio sub-systems – which efficiently utilises the available volume and modularises the design; as shown in Figure 4.

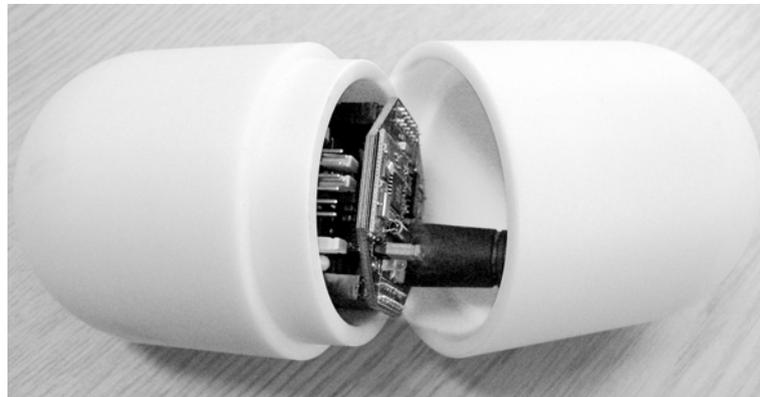


Figure 3. *Probe* housing shown open

At the heart of each *Probe* is a PIC microcontroller (PIC16LF876A), which is used to configure, read and store the attached sensors at user-specified times, handles power management, and communicates with the *Base Station*; as shown in Figure 5. The *Probe*'s orientation is measured using two dual-axis 180 degrees micro-electromechanical accelerometers (from Analog Devices) capable of sensing static G. The microcontroller reads the analogue sensors (pressure, resistivity, strain) with its in-built analogue-to-digital converter (ADC). The temperature sensor, real-time clock (RTC) and a 64kB FlashROM are accessed via the inter-integrated circuit (I<sup>2</sup>C) bus. 3.6V Lithium Thionyl Chloride cells were used due to their high energy

density and good low temperature characteristics. The radio transceiver uses conventional 1/4 wavelength “stubby” helical antennas which – although quite large – were more efficient than dielectric antennas used in the first prototype.

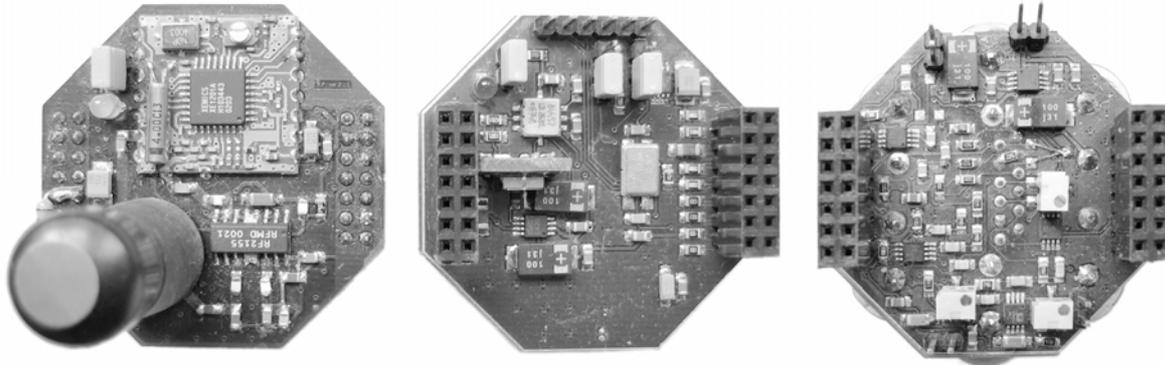


Figure 4. Analogue, Digital and Radio PCBs

The *Probes* “awaken” six times daily from sleep mode to collect data, but only enable the transceiver once each day to conserve power. This acquisition rate is acceptable as the data is not expected to change rapidly. The recorded data (96 bytes/day) is time-stamped and stored in the FlashROM, which is organised as a ring buffer. This setup allows the *Probe* to store up to 682 days worth of data in the event that it loses communication with the *Base Station*.

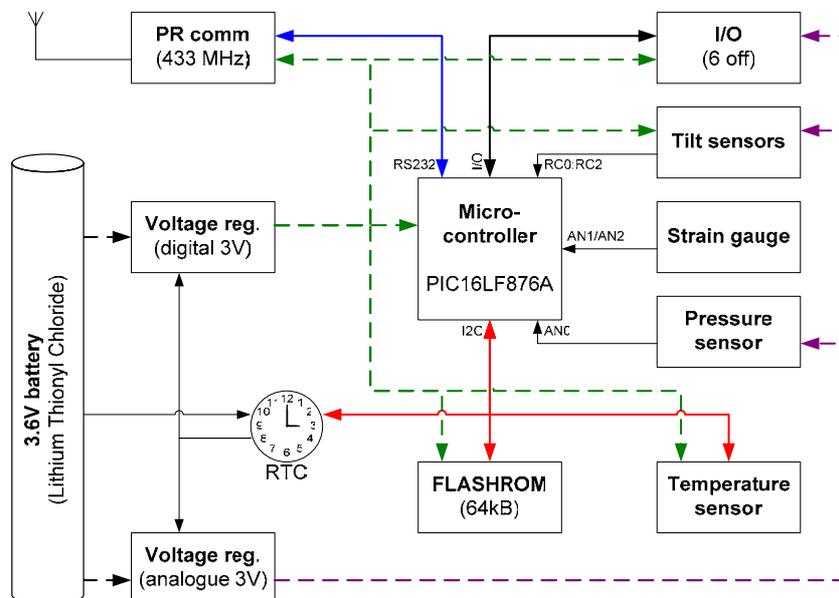


Figure 5. Block diagram of the *Probe* (version 2)

In sleep mode all the electronics, except the RTC and voltage regulators (outputs disabled), are unpowered: consuming only 9µA (32µW). The microcontroller sets the RTC’s alarm to the next “wake up” time before

commanding it to disable the voltage regulator outputs (powering-off the system). Although this is a risky manoeuvre – corruption of the RTC’s alarm registers could result in unpredictable operation – write access to these registers is always verified by the firmware. When powered, the *Probe* consumes only 4.1mA (~15mW) with the transceiver disabled, 24mA (86.4mW) when enabled, and approximately 130mA (~470mW) when transmitting. An estimated consumption of 100mW over the 180 seconds the *Probe*’s transceiver is active is reasonable considering the fact that transmissions occur infrequently. Hence, the daily power consumption is approximately 5.8mWH (15mW x 10s for five wake-up times + 100mW x 180s when transceiver is active + 32 $\mu$ W x 86210s for sleep time), which means its 3.6V 6AH (21.6WH) battery could last for more than 10 years! The *Probe*’s battery voltage is measured once a day as an indication of its energy status.

Based on the outcome of the previous season, the frequency of the link between the *Probe* and *Base Station* was lowered from 868MHz to 433MHz, and the radiated RF power substantially increased, to improve ice penetration. Xemics DP1201A-E433 transceiver modules were used in place of the Adlink 868MHz modules. The new modules allowed us to incorporate a programmable RF power amplifier to boost RF transmission power to over 100mW. Although this is outside the legal power level, the massive attenuation of the ice and water meant that the RF power is well within the 10mW limit for this license-exempted band on the surface of the ice. To further improve communication, the *Base Station* transceivers were buried 30 - 40m under the ice.

The *Probe*’s firmware – stored in the microcontroller’s program memory – is divided in two segments: the program space (5k words) and the user space (3k words). The program space contains the initialisation and control routines that configure the microcontroller’s in-built modules (e.g. I<sup>2</sup>C, ADC) upon starting. It also streams the bytes transmitted to and received from the on-board UART (universal asynchronous receiver transmitter), interprets received packets, and executes commands issued by the *Base Station* (via the packets) and user space. The program space is unalterable.

The user space, which holds the program that is autonomously executed whenever the *Probe* “awakens”, comprises the upper 3k words of the microcontroller’s program memory. This program is written as a C function and calls the same set of commands that could have been sent via the transceiver. Multiple programs could be stored in the user space (memory permitting) and the desired program is selected by issuing a specific command. Programs could also be loaded and removed from the user space as the microcontroller has the ability to reprogram its program memory while running. This feature greatly increases the flexibility of the *Probes* since their automated sequence could be remotely altered from Southampton even when they are buried in a Norwegian glacier. Finally, a mechanism is in place to ensure that rogue programs will be terminated if they exceed some preset timeout, and they will not automatically execute the next time the *Probe* “awakens”.

## B. Base Station

The *Base Station* controls the *Probes*, gathers surface data from the weather station and GPS, and links to the *Reference Station* in the valley. This system is powered from lead-acid gel batteries with a total capacity of 96AH in parallel with two solar panels (15W in total). A permanent weather and movement tolerant pyramid structure holds sensors, antennas and tethers the equipment, as shown in Figure 6. The weight of the batteries inside the box stabilizes the *Base Station* by creating an even surface as the ice beneath melts. A differential GPS unit is read on a daily basis to measure its location in conjunction with data from the *Reference Station's* GPS. This is carried out by taking 10 minute recordings at both the base and reference stations. A 500mW radio modem provides a 9600 baud link 2.5km down the valley to the *Reference Station* PC.



Figure 6. *Base Station* on the glacier with pyramid support

The *Base Station* uses a commercially available StrongARM-based (PXA255) embedded computer (*BitsyX*) with 64 Mbytes of RAM, running Linux. It has a custom board comprising a microcontroller (PIC16F876A) for power management and sensor interfaces. It also uses a compact flash card as a permanent data store. The controller, temperature and tilt sensors, signal and power isolators, RS232 level converter, RS232 splitter/combiners and switch-mode power supply (DC-DC 3V and 5V) regulators, as shown in Figure 7, are part of a separate PCB. The GPS, GSM, Weather and LR comm modules are external devices connected to the *BitsyX's* case. The Host and Debug ports are connected to the *BitsyX's* second and third serial port: these are just to aid development. A third “breakout” PCB – mounted above the *BitsyX's* case – was used to

connect other external devices to the *BitsyX* via the ADSmartIO, PR1 comm, PR2 comm, Camera and Port modules.

The controller and the temperature sensor are accessed by the *BitsyX* via the I<sup>2</sup>C bus. This microcontroller controls the isolators (solid state relays and analogue switches), which physically disconnects the power and signals to all modules. Such a design meant large power savings were achievable. Only the *BitsyX*, power and signal isolators, RS232 splitter/combiners and RS232 level converter are always on, which consumes very little power (when the *BitsyX* is in sleep mode). The controller also acts as an I<sup>2</sup>C bridge for the tilt sensor. This sensor was found to be useful in determining the stability of the *Base Station*.

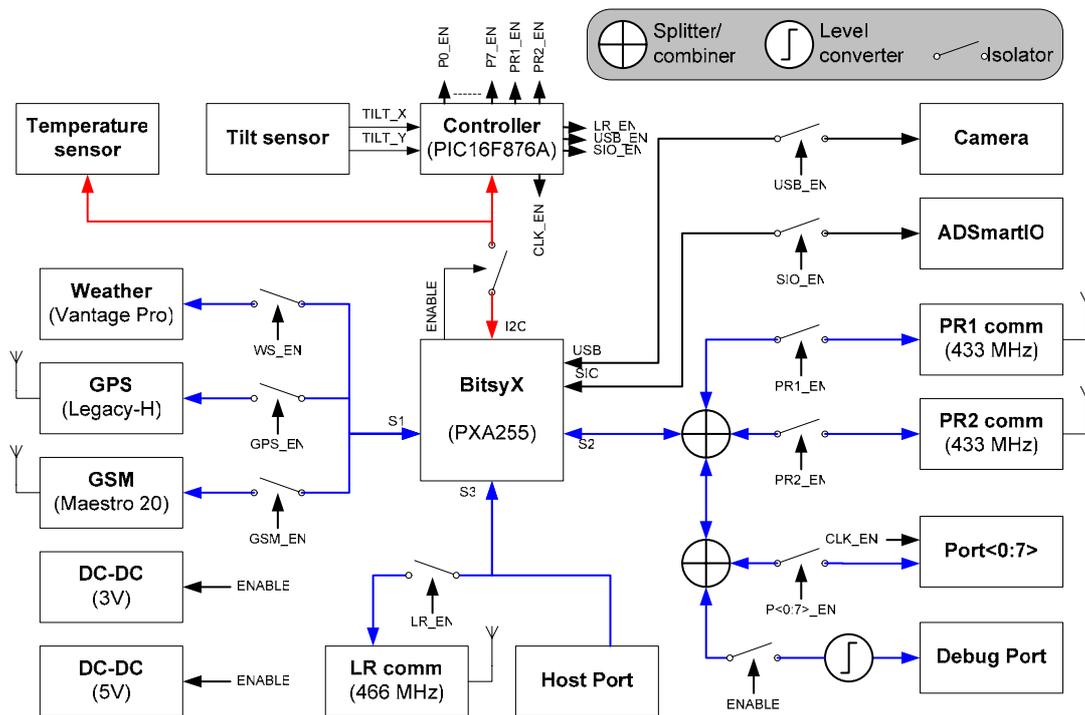


Figure 7. Block diagram of the *Base Station* (communication)

*BitsyX* is powered directly from the 12V batteries, and 5V is generated for the devices that are continuously powered, as shown in Figure 8. The remaining modules are either powered from the 12V batteries or the 5V/3V DC-DC converters (on the PCB) – depending on their voltage rating – via the power isolators. Both 5V and 3V sources are DC-DC converted from 12V for efficiency, and there are two of each module for redundancy. The DC-DC converted outputs also pass through linear voltage regulators to minimise power rail noise. The outputs of these modules can be disabled by the *BitsyX* since they may become unstable when no load is present (i.e. in sleep mode).

The USB port connects to a web-cam mounted on the pyramid support for observing the general condition of the site. In practice the Video for Linux drivers were found to be too unstable for this to be used regularly. The ADSmartIO module consists of eight I/O lines – four of which have ADCs – and power (3V, 5V, 12V)

lines. The I/O lines are controllable by *BitsyX* and were connected to a plough meter (measuring strain) and tilt cells (measuring glacier advancements relative to its bed). These more traditional instruments will be used to compare the *Probe* performance to conventional wired sensors.

Communication between *BitsyX* and the other modules are via its three serial ports, S1, S2 and S3. A four-way RS232 splitter/combiner was designed to split/combine the RS232 signals from/to serial port S2 of *BitsyX*. This allowed four different modules to share a single serial port – which is possible in this design – as these modules interpret packetised data and are slaves with respect to *BitsyX* (i.e. the modules only transmit packets when commanded). Although the Weather Station (Weather in Figure 7), GPS and GSM modules are connected to the same serial port, these modules are never used together. Their respective isolators guarantee that only one device is connected to *BitsyX* at any time. On the other hand, the Host Port and LR Comm could be enabled concurrently (for debugging) and it is the user’s responsibility to only connect one device at a time.

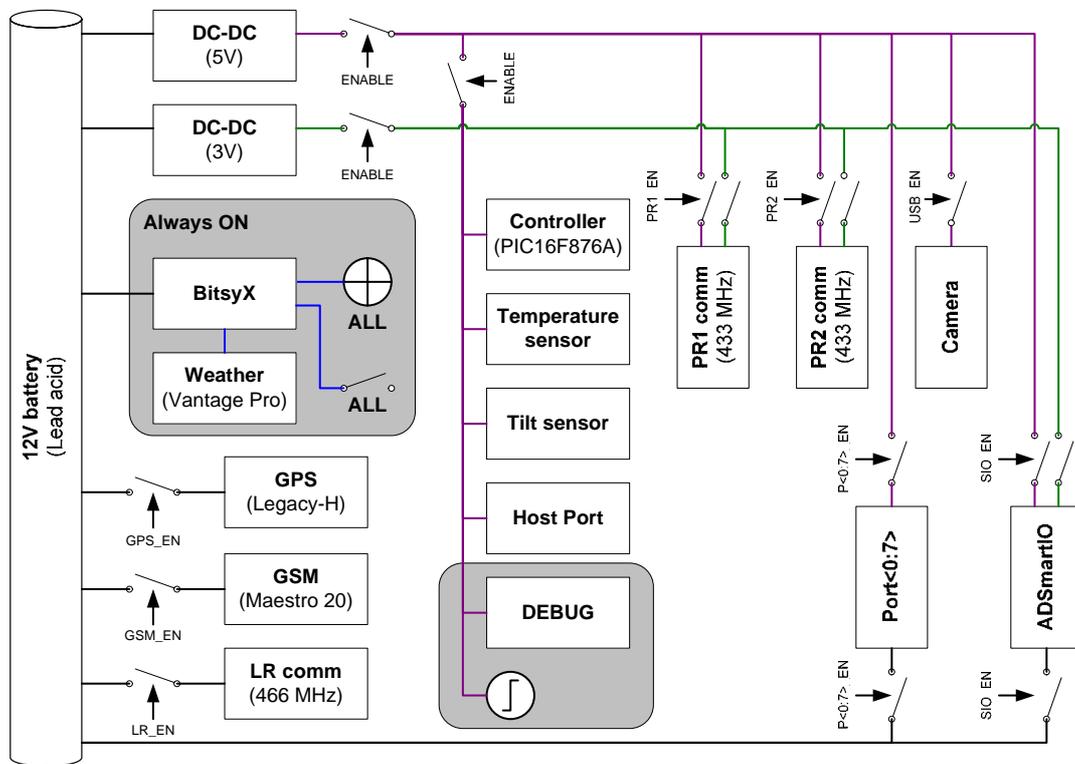


Figure 8. Block diagram of the *Base Station*'s power control

PR1 comm and PR2 comm modules connect to the transceivers which communicate with the *Probes*. Two independent channels were designed as this is a vital link as one of the serial cables in the ice could be damaged in the course of a year. The Debug Port is primarily used for monitoring the packets during development.

A GSM modem was implemented in the prototype *Base Station* (2003) which allowed data to be sent directly to the UK via text messages (SMS), if the long-range link fails. This was an important piece of

equipment in that system (which had less efficient antennas) and was also included in the current design as a backup link. Only the *Probe* and *Base Station* data are transmitted in ASCII coded hexadecimal characters: daily GPS data is too large (typically > 10kB) for SMS transfers. Apart from sending SMS, the GSM modem could be used for data communications in the future (i.e. a backup dial-up into the base station).

The differential GPS module is used in conjunction with an identical unit in the *Reference Station* to measure the exact position of the *Base Station*, which is moving at roughly 15cm/day in summer. For this technique to work both GPS units must record at the same time. The two files are processed off-line using Topcon's software and an accuracy of one centimetre is typical (it depends on satellite coverage).

The Weather Station is a commercial unit that measures precipitation, pressure, temperature, humidity, wind speed and direction, solar radiation and ultra-violet radiation. This module is powered continuously (45mW idle, ~80mW reading) and uses a "tipping-bucket" method to measure precipitation. The sensors are read periodically (varies according to sensor) and the data is stored on a console, which is accessed by *BitsyX* once daily.

Long range communication is handled by a pair of 500mW radio modems operating at 466MHz. One of these is connected to port S2 of *BitsyX* (LR comm module). The LR modem can be isolated from *BitsyX*, but the Host Port is permanently connected and always powered: this provides a means of logging on to *BitsyX* via a PPP (point-to-point) connection. This has been used on the glacier with a laptop to monitor system activity and reconfigure the system. The prototype used a custom protocol but the use of tcp/ip has greatly enhanced the usability of the link for administration.

As a whole, *BitsyX* itself consumes about 120mA (~1.45W) when operating, and 10mA (120mW) in sleep mode. When the second PCB is enabled, power consumption rises to approximately 165mA (~2W – excluding GPS, GSM and LR comm modules). The estimated power consumption when the *Base Station* executes its job is approximately 4W over 15 minutes (1WH hour per day). Combined with a consumption of 170mW (120mW *BitsyX* + 50mW Weather Station average) in sleep mode, the total estimated daily consumption is 5WH. This means the 96AH (1152WH) batteries should last approximately 230 days. The extra power is derived from the solar panel, which is estimated to produce about 15WH per day during summer (approximately 100 days); hence the *Base Station* should have enough power to run for a year. The overall battery voltage is recorded daily to record the power status. Experience from 2003 shows that the base station is subjected to temperatures down to -9C in the winter and that the batteries remain in a reasonable condition.

### C. Reference Station

The *Reference Station* is a mains-powered 500MHz EPIA-PC running Linux (RedHat 9) located in a café in the valley. This low-power PC format was chosen as can be supported for longer on a UPS as well as being small (micro-ATX form factor) and near-silent. It is connected to the *Base Station* via the radio modem, and

periodically to the internet via an ISDN router. Data is sent daily (as a compressed tar file) to the data server in Southampton. As its name implies, it also acts as the reference point for the DGPS system.

The distance between the Base and *Reference Stations* is 2.5km without a clear line-of-sight. Hence, high-powered low-frequency radio modems employed. In addition, a short directional YAGGI-type antenna increases the sensitivity to the signal and the low (9600) baud rate reduces bit errors.

#### D. Communication Protocol

Data transmission is the most power consuming task for the *Probes*: hence it is important to not only minimise their time awake, but also their transmission time. As it is unfeasible to continuously power the entire system or to stagger the communication window of each *Probe*, the entire system thus relied on a unified time schedule. This decision meant that there has to be a communication protocol between *Base Station* and *Probes*. In addition, this communication link must be suitably robust to maintain data integrity. Clearly an ad-hoc protocol would bring benefits such as probe data hopping back through other probes in order to get to the base station and possibly a power-saving from inter-probe communications being over smaller distances. However the PIC processor used in the probes would constrain the amount of memory available and in the time available implementing an ad-hoc network would have been risky. So a star network topology was chosen and furthermore the probes would be polled from the base station rather than autonomously send data. This would have required either precise timing or a MAC protocol to avoid power-consuming collisions. Similarly security protocols were not used as these would add complexity although base station links used Linux security.

A packet-based communication protocol with identifiers (ID) and error detection (checksum CS) was devised for *Probe* communications. Although many packet-based protocols such as UDP and TCP are widely available, they are deemed unsuitable for the PIC microcontroller with limited resources. Hence, a simple protocol – shown in Figure 9 – was developed.

<b>0</b>	<b>1</b>	<b>2</b>	<b>3 ... 18</b>	<b>19</b>
HD/SZ	ID	CMD	DATA	CS

Figure 9. Communication packet format (maximum size)

Each packet varies between 5 and 20 bytes and has six fields. Although variable packet sizes increase the complexity of the packet interpreter, this approach substantially reduces power consumption since most command and information packets are significantly shorter than 16 bytes. Each byte is transmitter least-significant-bit first, and the packet is transmitted least-significant-byte first. The gap between each transmitted byte has to be less 3ms: this feature ensures that spurious data will not greatly inhibit valid communication. A communication error occurs if this condition is not met and the base has the option of retrying.

The first byte contains the packet header (HD – upper nibble) and the size (SZ – lower nibble) of the DATA field. The header determines if the packet is a command or information packet. When a command

packet is sent by a source (the *Base Station* in this case), it expects an information packet to be transmitted by the sink (typically a *Probe*) within a preset duration. If a packet is not received in time, it is classified as a communication error. Unlike command packets, information packets could be sent by any device at any moment, and they do not expect a reply. These were used successfully for debug information.

The ID field is that of the target device in command packets; or a device's own ID in the case of an info packet. The addressed *Probe* responds with an information packet with the ID set as its own. If the *Base Station* receives an information packet from another source, it is classified as a communication error. No devices will respond to a command packet if the ID is set to the broadcast value. This exception allows the *Base Station* (or any controlling device) to "broadcast" commands (e.g. set RTC time and sleep) to all devices simultaneously.

The command (CMD) field allows up to 256 different enumerated commands to be defined. The data field varies between 1 and 16 bytes and the checksum (CS) is used to check the packet's integrity. When any communication or packet errors occur, the sender can resend the command packet. The number of retries is a compromise between reliability and power consumption and we currently use 3.

#### *E. Sequence of Events*

A traditional way to run such a sensor network is to power up all the systems and let the nodes send data autonomously. This approach was not taken as it was felt that reducing the power-waste of collisions would be critical in a large-scale deployment and that simpler software could be designed. This led to a design where the nodes only respond to commands and (at the moment) do not inter-communicate. The entire system works on a unified daily schedule relying on synchronised clocks. The *Probes* wake up six times daily: five for Data log periods (1s each) and once for up to 180s during the communications (Comm) period. The *Base Station* also wakes up for the Comm period and controls everything rather than acting as a router. The *Reference Station*, which is permanently powered, transfers its data to Southampton during the transfer period.

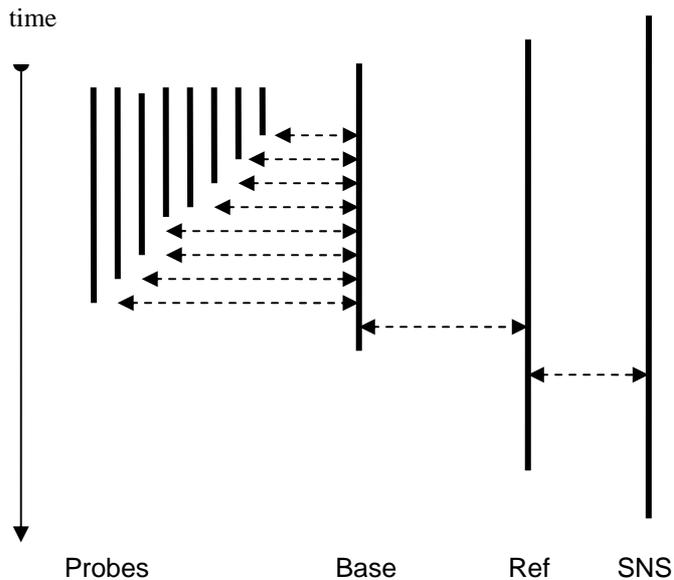


Figure 10. Sequence of events at communications time showing probes waking simultaneously, talking to the base station then sleeping.

Upon “waking up” during a Data log period, the *Probe* reads and stores its sensors, sets its RTC alarm for the next period, then goes to sleep. This procedure is repeated during the Comm period (1200 UTC) apart from it staying awake for up to 180s after setting its RTC alarm. It is this window that the *Base Station* uses to communicate with the *Probes* as shown in Figure 10.

The *Base Station*’s Comm period starts shortly before the *Probes* “awaken” when it reads and store its own sensors. It then goes into a polling sequence broadcasting the GPS system date/time and interrogating each *Probe* in turn. Tests have shown that the RTCs do drift up to  $\pm 2s$  per day relative to the GPS time; hence time broadcasting is vital to keep the system synchronised. If a *Probe* replies, the *Base Station* reads all its unread data before commanding it to sleep (hence the staggered sleep times of the probes as shown in Figure 10). The *Base Station* could also perform other task such as extracting old data or modifying the *Probe*’s user-space programs – if necessary – by modifying the *Base Station*’s job file (a shell script) or by commands issued from Southampton via the internet and *Reference Station*.

After the *Probe*’s Comm period, the *Base Station* starts a 15 minute GPS recording. A separate GPS period was used in 2003 that could be shifted manually because satellite visibility varies considerably over a year. For simplicity and to make use of the *Base Station*’s “idle” time when GPS recording takes place, GPS recording is fixed for 2004. Instead, the *Base Station* sends the previous day’s GPS file to the *Reference Station* while GPS recording takes place. The day’s data is transferred to a server in Southampton during the *Transfer* period, which occurs in the late evening to avoid hogging the dial-up ISDN line that is shared with the internet café.

#### IV. EXAMPLE RESULTS

Eight *Probes* were installed in August 2004, and immediately relayed their data. By the final day of the fieldwork we obtained results from six of the eight *Probes* using only one wired transceiver. The results can be summarised as follows. The temperature was a relatively constant 0.18-0.31°C. The *Probes* which remained

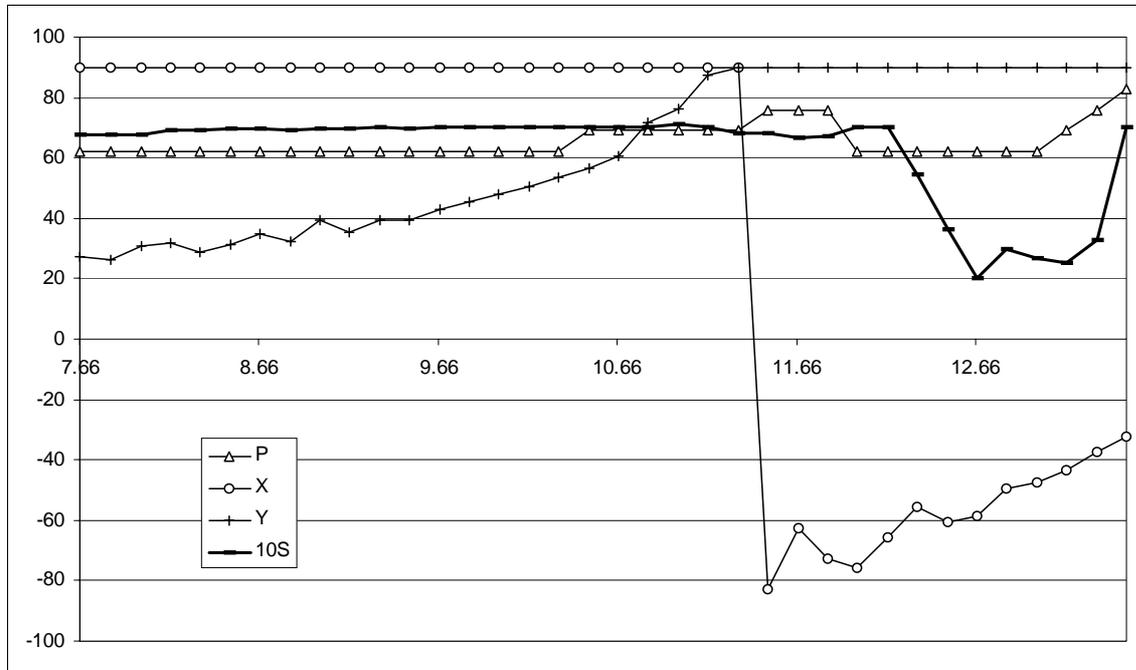


Figure 11. Sample *Probe* data. P – pressure in kPa, X and Y are tilt in degrees, 10S is 10 x strain gauge reading. X axis shows days in August 2004.

in air, as indicated by the external resistance, have a constant low pressure and stress as well as stable position (vertical). One *Probe* under 6 – 8 m of water showed fluctuations in pressure and stress (see Figure 11). The tilt data shows the *Probe* settling until day 11 when water pressure rose and the *Probe* rolls on its axis. The strain gauge shows it is subject to a mild pressure around day 12.

The results show that the radio communication, *Base Station* and *Probes* are functioning as they should and the *Probes* have entrained their sensors to provide consistent and reliable data. A log file is created showing useful system status such as how many retries occurred, which probes communicated and a trace of packet data. This will allow a more detailed system analysis as more data is received.

#### V. CONCLUSIONS AND FUTURE WORK

Our pragmatic choices of hardware and protocols allowed us to install the first prototype in less than one year as well as understand the challenges ahead. Understanding the properties of the system and any malfunctions has been greatly enhanced by our hand-crafted approach. Future research is needed in a probe location system, miniaturisation, ad-hoc network, web service development for the data and simpler system

administration. Although an ad-hoc network was not used we intend to implement our own in the next version within the constraints of power-saving and the base-driven architecture. Scalability was also a future aim because networks in the future will cover large areas and involve hundreds of nodes. In practice these would involve many base stations with relatively few nodes so our 256 node limit per base station is not seen as restrictive. The capability to add new nodes and manage decaying old nodes relies on maintenance of the protocol or backwards compatibility efforts. Clearly using a standardised protocol would greatly help in this area however using a simple, efficient protocol and having full control over its source code is beneficial to the research. For multi-vendor systems of the future a standard will be needed however. An issue which will be present in such systems for a while is simplicity of use so that sensor networks can be installed and maintained by end-users in the future. Similarly the availability of data from environmental sensor networks will require standards such as data Ontologies so that researchers can gather data globally in the future.

Environmental sensor networks represent a new step forward in understanding and monitoring the environment. This system has produced useful data never seen before by glaciology researchers and has generated considerable interest in environmental sensor networks. Developing such a system requires the latest technology in sensors and communications. This study is one of the first in a glacial environment and the research has demonstrated that the system is robust and is beginning to transmit results which will be vital for environmental science applications. We have found that an interdisciplinary approach is essential and GlacsWeb has involved people with expertise in radio communications, embedded systems, electronics, Linux, mechanical engineering, GPS, system administration, physics and glaciology. One contribution this research has made is to build a complete system for a specific purpose and it will become a platform for studying real-world implementations of theoretical concepts in sensor networks research.

#### ACKNOWLEDGMENT

The authors thank the Glacsweb partners Intellisys and BTEExact. Topcon and HP for equipment support. Thanks also to Paritosh Padhy, Harvey Rutt, Sue Way, Dan Miles, Joe Stefanov, Don Cruickshank, Matthew Swabey, Ken Frampton and Mark Long. Thanks to Inge and Gro Melkevol for their assistance and for hosting the *Reference Station*.

#### REFERENCES

- [1] C-Y Chong and S.P. Kumar, "Sensor Networks: Evolution, Opportunities and Challenges," Proc. IEEE, vol 91, no 8, 2003, pp. 1247-1256.
- [2] K. Martinez, J.K Hart and R. Ong, "Environmental Sensor Networks", Computer, Vol. 37, No. 8, pp 50-56, 2004.
- [3] K.A. Delin, R.P. Harvey, N.A. Chabot, S.P. Jackson, Mike Adams, D.W. Johnson, and J.T. Britton, "Sensor Web in Antarctica: Developing an Intelligent, Autonomous Platform for Locating Biological Flourishes in Cryogenic Environments," 34th Lunar and Planetary Science Conference, 2003.

- [4] [http://sensorwebs.jpl.nasa.gov/resources/huntington\\_sw31.shtml](http://sensorwebs.jpl.nasa.gov/resources/huntington_sw31.shtml)
- [5] R. Szewczyk, et al., "Lessons from a Sensor Network Expedition," Proceedings of the 1st European Workshop on Wireless Sensor Networks (EWSN '04), January 2004, Berlin, Germany, pp 307-322.
- [6] D.C. Steere, et al., "Research Challenges in Environmental Observations and Forecasting Systems," Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking (MOBICOMM), 2000, pp. 292-299.
- [7] K.A. Delin, S.P. Jackson, D.W. Johnson, S.C. Burleigh, R.R. Woodrow, M. McAuley, J.T. Britton, J.M. Dohm, T.P.A. Ferré, Felipe Ip, D.F. Rucker, and V.R. Baker, "Sensor Web for Spatio-Temporal Monitoring of a Hydrological Environmental," 35<sup>th</sup> Lunar and Planetary Science Conference, League City, TX, 2004.
- [8] K. Lorincz, D. Malan, Thaddeus R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, and M. Welsh, "Sensor Networks for Emergency Response: Challenges and Opportunities", Special Issue on Pervasive Computing for First Response, Oct-Dec 2004.
- [9] K. Martinez, R. Ong and J.K. Hart, "GlacsWeb: A sensor network for hostile environments", Proceedings of First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, pp. 81-87, 2004.