

# PLATO control and robotics

Daniel M. Luong-Van\*, Michael C.B. Ashley, Jon R. Everett, Jon S. Lawrence,  
John W.V. Storey  
School of Physics, University of New South Wales, Sydney NSW 2052, Australia

## ABSTRACT

PLATO, the 'PLATeau Observatory', is a robotic Antarctic observatory developed by UNSW for deployment to Dome A, the highest point on the Antarctic plateau. PLATO is designed to run autonomously for up to a year, providing power, communications and thermal management for a suite of scientific and site-testing instruments. To achieve this degree of autonomy, multiple-redundant Linux-based 'supervisor' computers, each with their own watchdog-timer and Iridium satellite-modem, communicate with each other and with the outside world. The active supervisor computer monitors and controls the PLATO power distribution, thermal and engine management subsystems via a CAN (Control Area Network) bus. High-bandwidth communication between the instruments and the supervisor computers is via a 100 Mbps Local Area Network. Data is stored in cold-verified flash memory. The PLATO computers monitor up to 140 analog channels and distribute electrical power and heating to 96 current-monitored channels via an intelligent load-shedding algorithm.

**Keywords:** Astronomy, Astronomical observatory, Site testing, Automation, Robotics, Control

## 1. INTRODUCTION

PLATO is the third-generation robotic observatory developed by the University of New South Wales for its Antarctic astronomy research program. The previous generations were the AASTO<sup>1</sup> at South Pole, and the AASTINO<sup>2</sup> at Dome C. PLATO was developed to go to Dome A, one of the most inaccessible locations on Earth, with human presence for at most two to three weeks per year. PLATO was therefore designed to be fully self-supporting, with autonomous operation between its yearly servicing intervals.

The PLATO observatory consists of a power-generation module and an instrument module. The modules are separated by 50 metres, and linked via an umbilical cable that carries power and data. This isolates sensitive instruments from the power generation system.

Companion papers<sup>3,4</sup> provide more information on the engine module, science-case, instruments, and the overall design of PLATO.

PLATO's control system is implemented using multiply-redundant industrial-grade PC/104 'supervisor' computers communicating via a CAN (Control Area Network) bus to microcontroller nodes. Each node is able to control aspects of power distribution, thermal control or engine management. The instrumentation suite is controlled via a 100 Mbps LAN. Each instrument is designed to be autonomous, taking data whenever the power is applied. However, the instrument computers can be controlled by the supervisor computers if necessary.

The PLATO control system data storage is implemented solely in USB flash drives that have been verified to work at temperatures down to  $-60^{\circ}\text{C}$ . Crucial elements such as the supervisor computers' operating-system and the microcontroller's real-time program are read-only to minimise the chance of corruption.

Communication is provided by an Iridium satellite modem in each supervisor computer. These provide continuous real-time status of the PLATO observatory, and daily reports and reduced data from the instrument suite. It is possible to use scripted or interactive sessions to update all of the on-board software, apart from the firmware in various microcontrollers. Bulk data storage is the responsibility of each instrument, and is typically done on conventional rotating hard disks. The bulk data will be retrieved during the Polar Research Institute of China 2008–2009 overland traverse.

\*daniel@phys.unsw.edu.au

## 2. PLATO COMMAND AND CONTROL

### 2.1 Hardware overview

An overview of the PLATO command and control system is shown in Fig. 1. Power is provided by two solar arrays that can provide 1 kW when sunlight is available, and two banks of three single-cylinder diesel-engines providing up to 1.8 kW per engine during winter. Electrical power from both the solar and diesel sources is converted to a 28 V nominal DC bus for all control and instrumentation<sup>3</sup>.

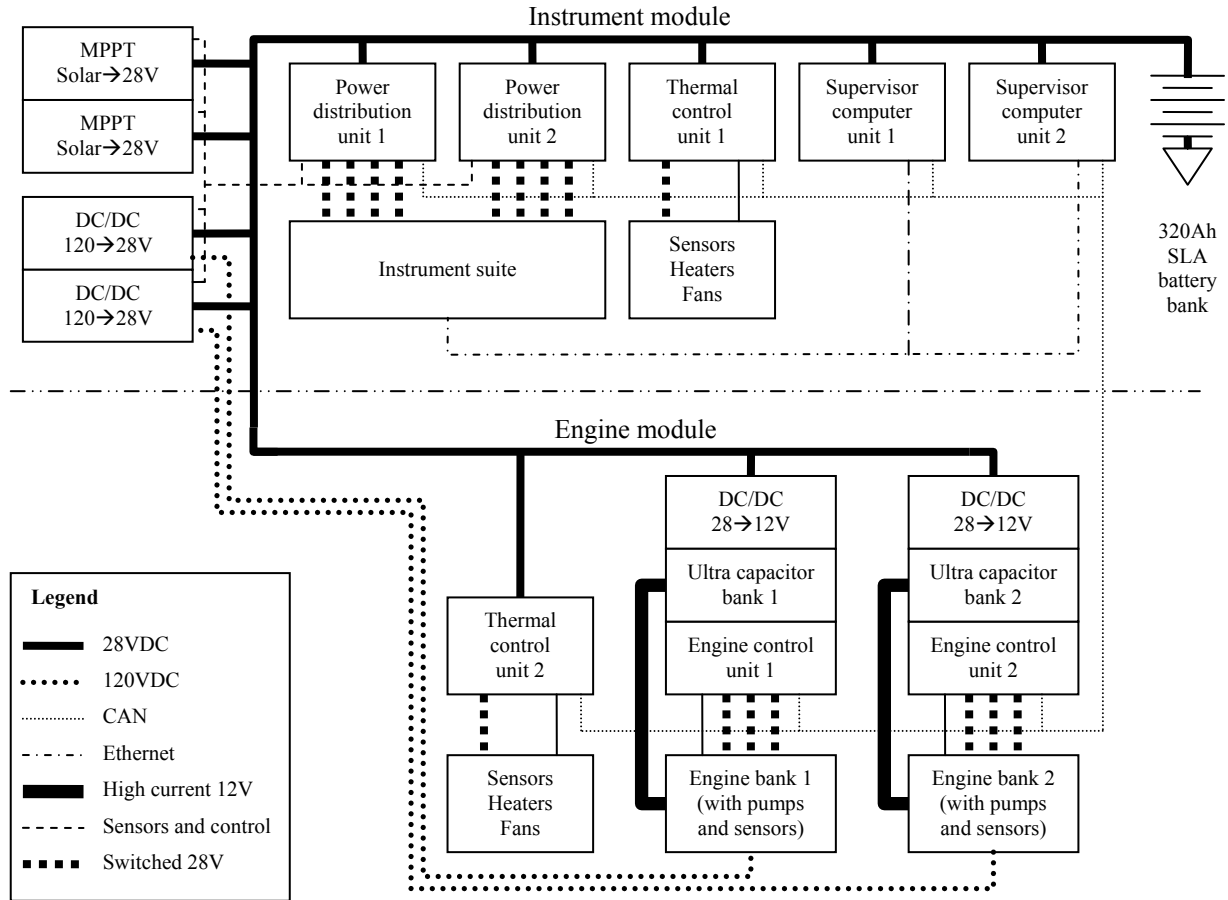


Fig. 1. Overview of the PLATO control system showing the interconnection of all major subsystems and power routing

The umbilical cable that links the engine module to the instrument module carries two 120 VDC cable pairs, a dual-redundant 28 VDC cable pair, and a CAN bus. The 120 VDC cable carries power from the engines to the instrument module DC/DC converter. The engine module electronics are powered via the 28 VDC cable since the instrument module contains the high-capacity DC/DC converters and SLA (Sealed Lead-Acid) battery backup<sup>3</sup>.

### 2.2 CAN microcontroller interface modules

A general purpose CAN I/O board based on the 8-bit Atmel AT90CAN128 microcontroller is used as the foundation for much of the PLATO control electronics. The CAN I/O board uses a wide-input high-efficiency DC/DC converter to provide operation from 9 to 34 VDC. Fig. 2 shows the common interfacing capabilities of the I/O board. Stackable expansion headers allow the CAN I/O board to be connected to various expansion boards that configure it for the different required control tasks.

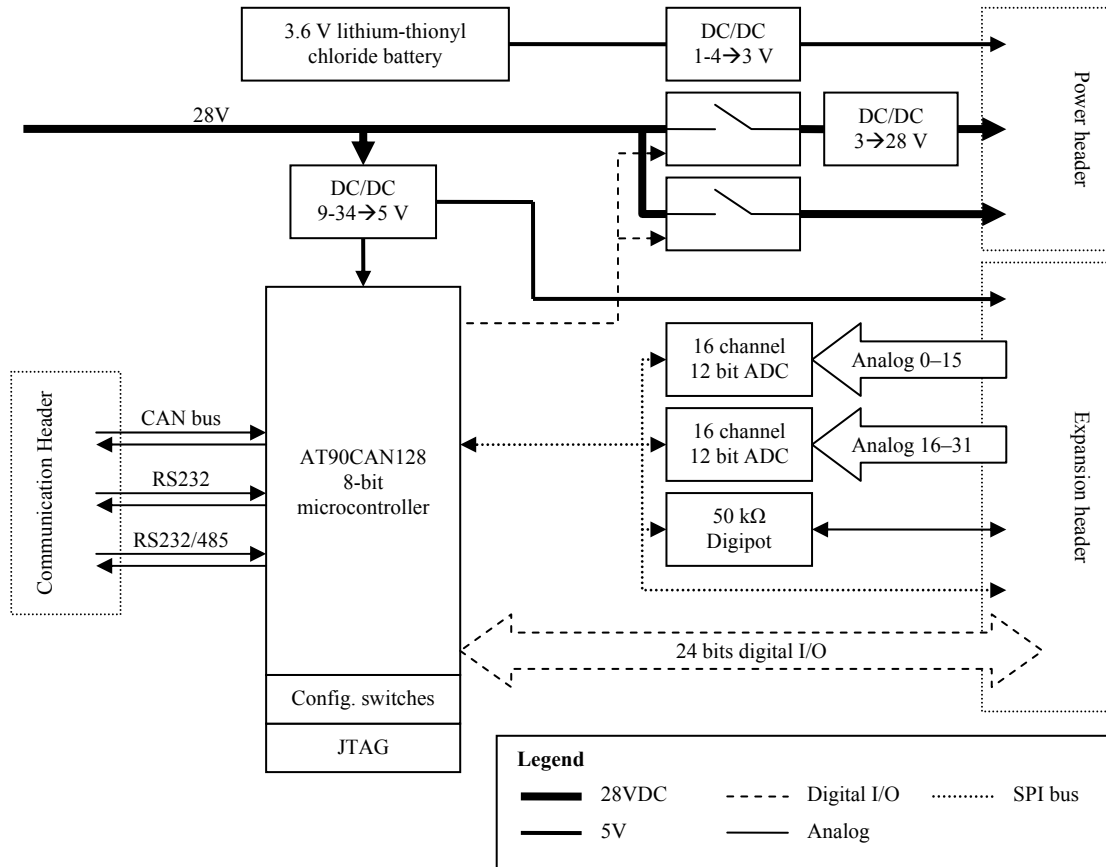


Fig. 2. CAN I/O board common features and signal routing

A total of eight CAN I/O boards were used and a common real-time program runs on all of them. The CAN baud rate is software programmable and defaults to 100 kbps. Jumpers select which functions to execute and the CAN address of the board. Each board has software to act as a supervisor, power distribution, engine or thermal controller. By using a common program, replacement units can be pre-programmed and easily configured for any of the required tasks. The controller beeps its CAN-bus address during power-up to allow for its identification. Lights are not used as indicators as these may interfere with sensitive instrumentation. Further information on the software can be found in section 3.

Two switchable high-current outputs are provided on the I/O board using Infineon BTS442 high side switches to allow for powering of various peripherals. One of the outputs has an optional DC/DC converter to allow for a selectable output voltage.

Thirty-two 12-bit 0–4.096 V analog inputs are provided by two MAX1230 analog-to-digital converters. Any pair of channels can be software configured for differential input and each has an internal temperature-stable band-gap reference. Each channel has a first-order anti-aliasing filter that can be configured for the required sampling rate.

A single 50 kΩ digital potentiometer provides a programmable resistance for adjusting external electronics.

24-bits of 5 V logic I/O and an SPI bus expansion allow for the board to be easily interfaced to other application-specific hardware.

The two RS232/485 channels per board are fully accessible on the CAN bus. Any standard serial device can be connected to a CAN I/O board and communicated with as if it were a local device. Full software baud rate configuration and hardware-handshaking is possible. The serial Modbus protocol timing is supported, which allows standard Modbus devices to appear as CAN devices.

A hardware and software watchdog-timer resets the CAN I/O board to a safe state if either the internal processor or communication protocol is not active in a defined time window.

Another feature of the I/O board is the provision of a high-reliability 3–3.6 V supply for real-time clocks. Standard lithium 3 V button cell performance rapidly degrades at temperatures below  $-20^{\circ}\text{C}$ . Instead, a high efficiency step-up DC/DC converter that can also handle over-voltage is used in conjunction with 3.6 V, 13 Ah lithium thionyl chloride batteries. When the battery voltage lowers due to the decrease in temperature, the step-up converter will provide a stable 3 V output for input voltages as low as 0.9 V. In this way, real-time clocks and battery backed memory can be powered for well in excess of a year at temperatures below  $-80^{\circ}\text{C}$  should PLATO's primary power system fail.

### 2.3 Supervisor computers

Major control decisions are performed by two supervisor computer units in a dual-redundant configuration. PLATO was designed to accept three supervisor computers, but one was held back for development purposes this year. Each supervisor computer contains a Parvus CPU-1452 PC/104 stack, a low-temperature rated NAL 9522A Iridium modem, and a CAN I/O board mounted in a 3U case. A laser-cut chassis plate holds all the components securely and rubber isolation mounts protect the contents against vibration on the  $\sim 1200$  km sled traverse to Dome A from the Antarctic coast. Fig. 3 shows the interconnection of the components and power and signal routing.

A CAN I/O board running a supervisor support program powers the PC/104 via its unregulated switched power output, and the Iridium modem via the regulated power output.

The PC/104 computer is a low-power passively-cooled 400 MHz Celeron processor with 256 MB of directly soldered memory to maximize reliability. Traditional battery backed CMOS settings are stored in flash memory to retain all important parameters, eliminating the need for battery-backup. Real-time clock power is provided by the I/O board's high reliability 3 V power supply.

The PC/104 uses the CAN I/O board as a gateway to the CAN network via an RS232 connection, with the CAN I/O board routing commands to and from the CAN bus. A watchdog-timer reset command must be sent by the PC/104 to the I/O board every five minutes otherwise the I/O board will power cycle the PC/104. Logging of the 28 V bus and the 5 V voltage level present on the PC/104 power supply is performed by the I/O board for diagnostics.

The PC/104 sends commands to the I/O board to control power to the Iridium modem and the modem is only powered when communication is active. Section 2.8 details the data transfer capability and strategies.

An AD590 temperature sensor is attached to the heat sink of the PC/104 processor and is monitored by the I/O board. The 3U rack mount case is sealed to retain the 8–12 W of heat produced by the PC/104. This heat is distributed in the case to keep the Iridium modem warm via a fan which is controlled via commands from the PC/104 computer. In practice, the temperature at the PC/104 CPU heat-sink is about  $25^{\circ}\text{C}$  when the internal module temperature is  $-10^{\circ}\text{C}$  and the outside ambient is  $-70^{\circ}\text{C}$ .

An omnidirectional Iridium antenna is used on the first supervisor unit and a high-gain antenna is used on the second. This is the first time a permanent Iridium connection has been used at the geographical location of Dome A, and it was essential that communication could be made at least some of the time. Our tests have shown that reliable communication can be made with the omnidirectional antenna, so we plan that all our further antennas will be of this style.

An internal four-port high-speed USB hub holds four 8 GB cold-rated USB flash memory keys. Cold testing of the USB keys involved continual read-write verification with MD-5 checksums at  $-60^{\circ}\text{C}$ . A front mounted USB port is used to hold the read-only GNU/Linux operating system. This is described in detail in section 3.

### 2.4 Power distribution unit

PLATO provides each instrument with a budget of 100 Watts average electrical power. Power is distributed to up to 16 instruments via two power distribution units, each capable of a peak of 2.5 kW, although this power level is seldom used. Each power distribution unit fits inside a standard 2U case and contains a CAN I/O board piggybacked to a large solid-state switching board. Fig. 4 shows how the internals of the power distribution unit are interconnected.

Each instrument has a separate instrument-power and heater-power line paired into a shared connector. The instrument-power is high priority and all attempts are made to shut down an instrument gracefully before it is switched off. The heater power may be switched at any time if power needs to be conserved. For further information on the power-shedding algorithms, refer to section 3.

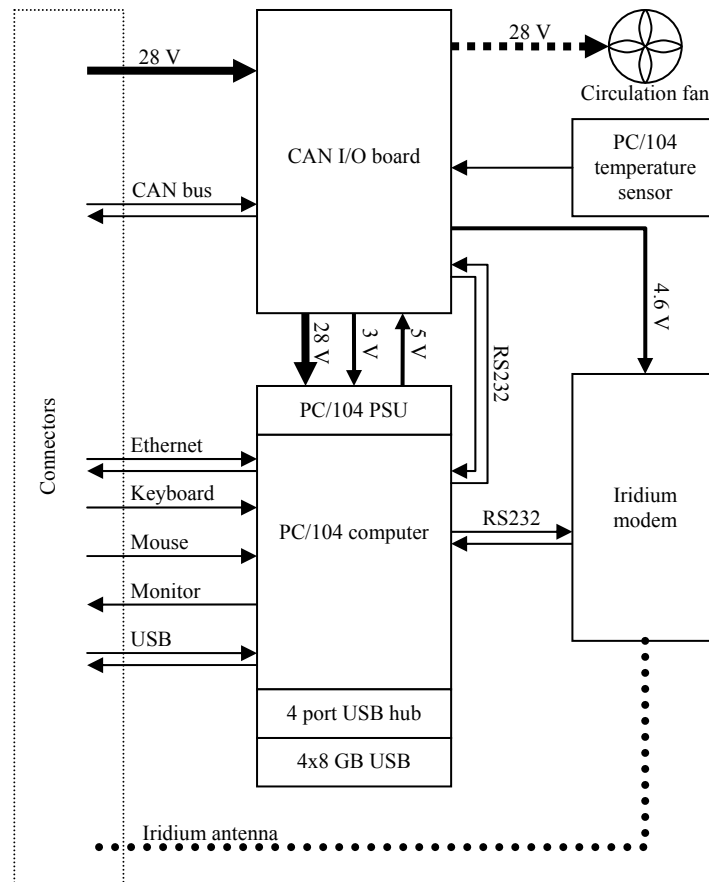


Fig. 3. Supervisor computer unit components and signal routing

There are eight output connectors per power distribution unit, using a total of 16 BTS442 solid-state high-side switches controlled by digital outputs from the CAN I/O board. Each instrument-heater switch pair is monitored by a 0.5 mΩ shunt resistor and high-side differential amplifier. Each amplifier output is monitored via an analog input on the CAN I/O board. In this way, the total power used by each instrument is recorded.

One of the design philosophies used in PLATO was to eliminate as many hard-fuses as possible, thus current limiting is implemented electronically. If the combined current of an instrument port goes above 12 A, the port is shut down for 1 second before retrying. The CAN I/O board performs the output reset and the supervisor computers can decide whether to continue to power the port.

Each switch can be programmed with an asynchronous period and PWM (Pulse Width Modulation) duty cycle. This allows for heaters and frequency driven devices such as solenoid pumps to be powered directly from the power output connectors. Software interlocks prevent devices such as computers from being PWM driven.

Each switching board has a large prototype area for constructing sensor conditioning circuits so that various sensors can be read using the spare ADC channels of the CAN I/O board. The CAN I/O board provides 5 V to power any of the custom circuits as well as the current sense amplifiers.

An additional responsibility of the two power distribution units is to monitor the solar MPPT (Maximum Power Point Tracker) units<sup>3</sup>. These communicate on RS485 bus which is connected directly to the CAN I/O board, which then rebroadcasts the RS485 messages as CAN packets. There are two MPPT units and two power distribution units, so redundancy is possible.

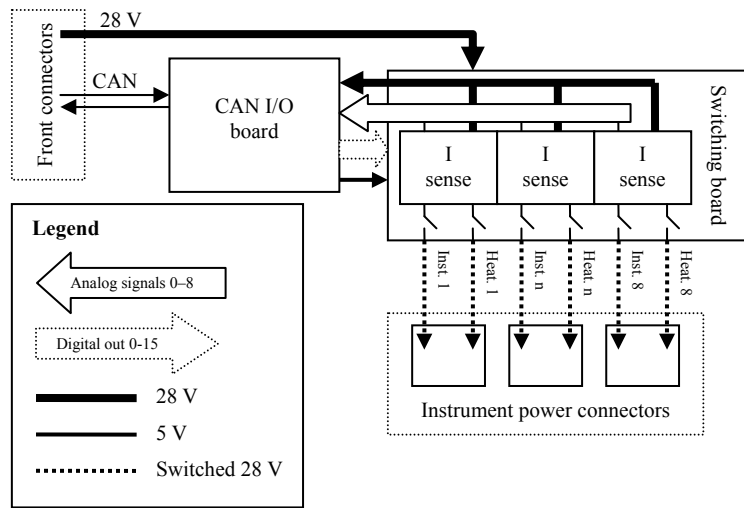


Fig. 4. Power distribution unit

## 2.5 Engine management unit

The engines are started from a high current 12 V bus, with ultra-capacitors providing the bulk energy storage<sup>3</sup>. The CAN I/O and power-switching boards are fully 12 V compatible, so no circuitry changes are required.

There are two banks of three engines, each requiring medium-current supplies for the starter solenoid and glow plugs, and frequency outputs for the fuel and oil circulation pumps. Two engine management units are used, each controlling one bank of three engines. The prototype area is used to provide 12 K-type thermocouple inputs to measure air intake, oil, cylinder-head and exhaust temperatures. The CAN I/O board also measures the generator output voltages and the main fuel tank level. These signals are conditioned on the prototype area and read using the available analog inputs on the CAN I/O board.

A piezo-resistive absolute-pressure gauge is connected to one of the analog inputs of the CAN I/O board to measure the internal pressure of the engine module. This sensor correlates very well with the Australian Antarctic Division's Dome A automated weather station<sup>5</sup>.

## 2.6 Thermal control unit

Both the engine and instrument module require a thermal control unit to prevent overheating and unacceptably low temperatures. For further information on the thermal properties of PLATO, refer to Lawrence et al.<sup>4</sup>. To heat the instrument module, twelve heaters are paired into six heater-units each consisting of two heater-elements and one 80 mm circulation fan. Each heater can be PWM driven to allow the heater power to be adjusted from 0–360 W per heater pair. One set of heaters is attached to the SLA batteries and does not require a fan, while the rest are strategically placed around the module to provide heating to the electronics-racks and door-seals. Waste heat from the diesel-engines provides heating for the engine module.

Each thermal control unit has signal conditioning for up to nine AD590 temperature sensors. The sensors are conveniently plugged into the front of the unit and measure the temperature profile of the engine and instrument modules, as well as the lower fuel tank and external temperatures.

Cold air is drawn from outside the modules via cold air intake fans to provide cooling during the summer months and when there may be an excess of heat from the instruments or engines. Ducts for the fans are designed so that no air is transferred if the fans are not active.

A Eurotherm 3200 PID (Proportional Integral Derivative) temperature controller is integrated into each thermal control unit. The Eurotherm controls the cold-air intake fans and is able to auto-tune its PID parameters. The Eurotherm communicates via Modbus and is interfaced into the CAN I/O board which then translates to/from CAN messages. The

Eurotherm outputs a digital signal which is interfaced into the switching board to control one current-monitored channel. Regulation only occurs if the module temperature rises above a configurable set point, but if a stable temperature is required, the heaters can be used in conjunction with the intake fan. By design, the engine module always has enough waste heat to allow the Eurotherm to keep the internal temperature at 15°C and the internal temperature is typically regulated to within  $\pm 1^\circ\text{C}$ .

## 2.7 Communication

To allow for redundant supervisor computers, the communication hardware and protocol must allow for multiple masters. By design, CAN bus networks have priority and collision-detection. The two supervisor computers are capable of both sending and receiving to and from all the CAN I/O based hardware. The CAN bus uses a 100 kbps baud rate which is well in excess of what is required, and allows for further expansion of the network. The CAN transceivers are slew rate limited and filtered, and this, in conjunction with shielded differential twisted-pair cabling, maximises signal integrity. The low baud rate also increases the effectiveness of the digital filtering present in the CAN hardware and reduces the effects of clock drift. The clock drift measured at Dome A between the CAN I/O modules is less than 2 ppm, which is well within the 0.5% design requirement, and no signal integrity issues were measured before deployment.

A local-area-network using twisted-pair 100 Mbps Ethernet allows for high bandwidth communication between the supervisor computers, the instruments, and other network based hardware via TCP/IP. An industrial-grade dual-power-supply Ethernet-switch that can be power-cycled via the CAN bus was used to interconnect the LAN devices. A 54 Mbps wireless access-point with internal and external antenna is used during deployment to minimise the number of physical connections made to the PLATO hardware. This configuration reduces the chance of ESD damage and allows for communication from the comfort of the base camp. The wireless access point can be switched off via the power distribution board to conserve power and to ensure that the radio does not interfere with sensitive instruments.

Details on how the multiply redundant systems determine which system is in control are detailed in section 3.2.

Communication to the outside world is performed via the two NAL 9522A Iridium modems in the supervisor units. Each modem is capable of 300 Bps data-rates and is fully independent. Software described in section 3.2 is used to pool the available bandwidth, and at the time of writing over 2.0 GB of bzip2 compressed data has been transferred in 4 months.

## 2.8 Video cameras

PLATO has four Watec video cameras that feed into an Axis video digitiser. The cameras output composite video. One of them is full-colour, while the other three are low-light black and white versions. The cameras have been cold tested to below  $-80^\circ\text{C}$  and do not require any heating, making them ideal for use in Antarctica. The four cameras are mounted on the roof of the instrument module and are used to monitor the external experiments and engine module. For further information on the cameras refer to Lawrence et al.<sup>4</sup>.

# 3. SOFTWARE

## 3.1 CAN I/O board program

The firmware that runs on each CAN I/O board was developed using the GNU C cross-compiler for the 8-bit AVR architecture. The program takes advantage of hardware-accelerated features such as 15 CAN message filtered mailboxes to achieve good performance with an 8 MHz clock. The AT90CAN128 micro-controller has a hardware watchdog timer that must be periodically refreshed, otherwise the processor is reset. A software watchdog timer also resets the CAN I/O board if a valid CAN message is not sent and received at a minimum of once every five minutes. At the time of writing, 10 M seconds have passed without any unintended resets.

When the CAN I/O board boots, it reads a series of switches that set the CAN address and configure the functions to a supervisor, power, engine or thermal controller. The initialisation takes approximately one second and the CAN address is indicated via a series of different pitch beeps. By using common code where possible, software development time was decreased; this also has the advantage that each board could be flashed with the same software image.

The full CAN 2.0B 29-bit address specification is used, with the 11-bit base identifier selecting the board, and the lower 18-bits selecting the sub-functions of the board. Each CAN I/O feature is individually addressed per switch, pump, DAC-channel, current-sensor, etc. All the board functions are exposed in atomic messages, with no multiple-state message

sequences. This allows for any number of supervisor units to read and write to all the peripherals. As each supervisor receives all the CAN messages, it can tell what actions the other machine is performing and behave appropriately.

When the CAN I/O board is configured for use inside a supervisor unit, it uses a checksum on all serial traffic to minimise the risk of incomplete or corrupted messages being processed. A simple message packet is used consisting of a send or receive header, CAN address, data and full message checksum. The same protocol is used on the RS232 interface to the PC/104 computer and on the CAN network. The CAN I/O board effectively bridges the two buses as all messages are rebroadcast. This allows for the supervisor computers to be fully redundant with either machine having full communication capability to all other I/O boards. It also allows the supervisor computers to monitor each other, for instance, supervisor-1 can monitor supervisor-2's internal temperature, control its fan or power cycle it. This allows a supervisor unit to power down its peers if one becomes unresponsive. So far this has not occurred in 4 months of operation.

The CAN I/O boards are also the last 'intelligent' agent to be in control of PLATO if the engine module fails and there is no solar power. The boards can be commanded to power off all peripherals for a fixed period of time. As only the supervisor computers can send the power-down messages, it is possible for them to boot up, send a short command message, then power down. The five minute software watchdog timer prevents a deadlock situation if a computer fails to boot, while still allowing greatly reduced total power consumption.

### **3.2 Supervisor computer software**

The supervisor software has the responsibility for all the high level control decisions. It provides commands to all the CAN I/O boards in the control units, decides on the instrument scheduling, heat allocation, engine bank power, battery maintenance and communication.

A GNU/Linux operating system derived from Debian etch i386 is used as the platform for all the supervisor control software. The operating system has been modified to use a completely read-only root file system that can be booted from a USB flash-memory drive. The volatile files in '/var', '/tmp', '/etc' and '/home' are symbolically-linked to a virtual-memory file-system which is uncompressed during boot. In this way non-persistent changes can easily be made, and if they are unsuccessful, a power-cycle is all that is needed to restore the system. The system can be powered down at any time without warning and still be guaranteed to cleanly boot. A script that allows updating of the read-only file system allows for permanent changes to be made if desired.

The operating system image is constructed using a script that uses the 'debootstrap' program to automatically create a complete USB bootable installation. There are no individual customisations for each supervisor computer, and instead the network MAC address is registered and used against an internal database of known PC/104 computers. This allows any USB drive to be used in any of the supervisor computers.

The operating system is kept on its own USB key, as the wear-levelling algorithm used inside flash-memory drives makes partitioning ineffective against file-system corruption.

A client-server model is used to allow programs to talk to the CAN bus. Only the server is allowed to directly communicate with the CAN I/O board, and anti-flood detection is used to prevent a run-away process from overly loading the CAN bus.

As each CAN message is fully atomic, addressed, and identifiable, there is no need to track the state of the CAN bus communications. Client programs can send and receive messages to any of the devices on the CAN network and an agreed device naming and data format is returned. The command and data format is exposed in a shared library so all the programs utilise a consistent scheme. The server will reject any communication that is not completely valid. The Perl 5 scripting language is used in all the control programs, and the communication scheme is designed to be easily processed by the inbuilt regular expression and evaluation engine. A Perl library that handles many of the high-level common tasks ensures a high degree of code re-use between scripts.

The core power-management program is implemented in Perl. It monitors all the solar, engine and battery states. If there is insufficient power, it attempts to run a diesel-engine and if more power is needed a second engine can be operated. The script can selectively choose which engine to run depending on the engine wear strategy. At the time of writing, the current strategy is to run a single engine for as long as possible.



The script also attempts to generate the most efficient use of fuel. The engines provide the best efficiency when they are producing 600–800 Watts of electrical power. The power-management script will first attempt to run as many experiments as possible, then start prioritising heaters. The most important heaters are the ones present on the instruments, followed by the battery heaters, then the room heaters. Depending on the observing conditions and an ephemeris, it is possible to schedule instrument power. For this year, this has not been needed since ample electrical power has been allocated to the instruments to allow continuous operation.

The script continuously monitors the battery voltage and ensures that, if possible, the batteries are always fully charged. The exception to this is when there is ample solar power; in this situation the batteries are used to power the module as there is a high probability the sun will be able to recharge them<sup>3</sup>. The script will periodically move the battery float voltage into the soak-charge region to ensure the batteries have a full charge, which is essential for the health of SLA batteries.

If the power sources are inactive for long enough to allow the batteries to discharge, the heaters are immediately powered down; if the batteries continue to discharge, instruments are sent a power-off command before they too are powered down. PLATO can then enter a low power consumption mode with only one supervisor computer periodically powering up. At the time of writing, all power sources have been reliable.

Important scripts such as the power control program are run simultaneously on all the supervisor units. As each supervisor unit can monitor all the CAN I/O parameters, timer-based control tokens are left on the bus. This allows the supervisor units to know which unit is in control so that competing outputs are avoided. By designing the PLATO control system to use stateless messaging, even if two supervisor units decided to command a parameter, minimal if any harm would be done.

The communication software that controls the Iridium modems is also implemented in Perl and fully supports IP data communication and SBD (Short Burst Data) messaging. The bulk of the data transfer is transferred via TCP/IP using an SSH tunnelled connection. In times of unreliable communication, the SBD messages have an excellent success rate and have a latency of approximately 5 seconds.

A special data transfer program that constructs self-assembling messages and has the ability to execute arbitrary commands can be used via SBD messaging if the TCP/IP communication is unreliable. The SBD messages are encrypted using GPG to ensure uncorrupted and secure communication.

As the data rate is extremely low, standard file transfer programs such as ‘rsync’ are not ideal as they are designed for large transmission buffers. A custom data transmission protocol was written that uses 4 kB packets. It implicitly allows for retransmission, and all data files are verified using an MD-5 checksum. The process of transmitting files between the outside world and PLATO is efficient and is as simple as copying the desired data to be transmitted to PLATO into a specially named directory on a computer at UNSW, and retrieving data transmitted from PLATO in another specially named directory. It is easy for any shell script to utilise the file transfer system, and it allows for one supervisory program to allocate the valuable bandwidth. At the time of writing over 2.0 GB of data has been transferred from PLATO back to its collaborators.

Another useful feature of the transmission protocol is the ability to send ‘out-of-band’ data alongside the primary data. Near real-time status information regarding the engines, batteries, heater and total current consumption is transmitted as out-of-band data. The status information is monitored in real-time by the PLATO team, and if any anomalous data are detected, monitoring programs will raise an alarm and send an SMS message to a mobile phone.

#### 4. CONCLUSION

At the time of writing, the PLATO observatory has been operating autonomously for over 120 days. All the systems are fully functional and real-time statistics, remote tuning, and updates of the control system software have been successfully performed. The control and automation system has been flexible enough to allow for full control over the power, heating, instrument and communication systems. Many failure modes can occur without loss of functionality, and most other failure-modes have a low probability of disturbing surrounding systems. Instruments from our collaborators have been easily integrated into the control structure, and there is enough capability for significant expansion in the future.

## 5. ACKNOWLEDGEMENTS

The authors wish to thank the Australian Research Council, the Australian Antarctic Division, and the University of New South Wales, for their financial contributions, without which this project would not be possible. The PLATO team is in great admiration of the Polar Research Institute of China Dome A expedition, that safely deployed PLATO to Dome A. Many thanks must also go to all those students at UNSW who volunteered their time in the construction of PLATO.

## REFERENCES

- [1] Storey, J.W.V., Ashley, M.C.B. and Burton, M.G., "An automated astrophysical observatory for Antarctica," *PASA*, 13, 35–38 (1996).
- [2] Lawrence, J.S., Ashley, M.C.B. and Storey, J.W.V., "A remote, autonomous laboratory for Antarctica with hybrid power generation," *Aust. Journal Elec. Electronics Engineering*, 2(1), 1–12 (2004).
- [3] Hengst, S., Allen, G.R., Ashley, M.C.B., Everett, J.R., M.C.B., Lawrence, J.S., Luong-Van, D.M. and Storey, J.W.V., "PLATO Power—a robust, low environmental impact power generation system for the Antarctic plateau," *Proc SPIE*, 7012, in press, (2008).
- [4] Lawrence, J. S., Allen, G., Ashley, M. C. B., Bonner, C., Bradley, S., Cui, X., Everett, J. R., Feng, X., Hengst, S., Hu, J., Jian, Z., Kulesa, C. A., Li, Y., Luong-Van, D., Moore, A. M., Pennypacker, C., Qin, W., Riddle, R., Shang, Z., Sun, B., Suntzeff, N., Tothill, N. F. H., Travouillon, T., Walker, C. K., Wang, L., Yan, J., Yang, J., Yang, H., York, D., Yuan, X., Zhang, X., Zhang, Z., Zhou, X., Zhu, Z., "The PLATO Antarctic site testing observatory," *Proc SPIE*, 7012, in press, (2008).
- [5] Australian Antarctic Division weather station, <http://www.aad.gov.au/weather/aws/dome-a/index.html>