Designing a Modular Communications Bus for High Altitude Balloon Missions

Cesar Parent

School of Arts, Media and Games
University of Abertay Dundee
Dundee, DD1 1HG, UK

ABSTRACT

Context: Current high-altitude scientific balloon missions use single-purpose, custom-made on-board computer and communications systems, whereas satellites use modular buses that host custom-made instruments. Such a bus could lower the cost and complexity of High-Altitude Balloon missions.

Aim: The investigation aims to determine the feasibility of a modular bus to provide data transfer and radio communications for High-Altitude Balloon missions. The goal is to design, document and build a prototype bus that will be used as a platform to test communications protocols, data bus formats and connection to external payloads. Testing will determine whether the design is suitably reliable and fast for in-flight use.

Method: Using Commercial-Off-The-Shelf hardware, a prototype payload bus, its software and its communication protocols will be designed and built. Multiple intra-payload communications protocols, forward-error correction algorithms and radio packet formats will be tested. Then, full scale hardware-in-the-loop ground simulations will be run with multiple test payloads connected to the bus.

Results: The performance and reliability of the bus will be evaluated by measuring the data and power efficiency of the prototype, the robustness and flexibility of its software and by checking that the prototype can handle full-flight simulations.

Conclusion: The project aims at defining a modular bus that High Altitude Balloon missions can be based upon. Provided the prototype passes testing, the documented protocols and software and hardware interfaces could allow wider use of High Altitude Balloons for upper-atmosphere research.

KEYWORDS

Radio Communications, Networking, Forward Error Correction, High-Altitude Balloon, Data Bus, Telemetry Systems

1 INTRODUCTION

Nano-satellites have lowered the cost of access to orbit by allowing light payloads to be launched as secondary passengers on heavy-lift rocket flights. However, the cost of the hardware required and the launch itself — when it is not subsidised by the launch provider or space agencies — still limit their use to well-funded universities and companies.

For applications that require low atmospheric density or medium altitudes, but do not depend on the payload being in a Low-Earth Orbit, High-Altitude Balloons (*HAB*) present an affordable alternative. Such balloons, inflated with low-density gases (usually helium or hydrogen), can reach the stratosphere (altitudes of 30 to 45km). The payload is encased in a thermally insulated container attached under the balloon itself. Depending on the mission, balloons either float at a constant altitude and drift for long durations or climb to the target altitude, then burst under the pressure of the expanding gas. Once the balloon is destroyed, the payload is brought back to the ground under a parachute.

The main component of the payload is the on-board computer (*OBC*), which controls the different instruments through the different phases of flight (ascent, observation, free-fall and parachute descent). Because high-altitude winds can reach speeds upwards of 150km/h and carry the payload over long distances, most balloons carry a Global Navigation Satellite System (*GNSS*) receiver that the OBC can use to keep track of the balloon's position and altitude. To avoid loosing flight data if the payload is lost or lands in an unaccessible area, the OBC also controls a radio transmission system used to forward payload data and metadata (GNSS coordinates, altitude, systems health) to a ground station as it is acquired.

Depending on the mission, various instruments are added to the payload. The OBC is in charge of gathering, storing and transmitting their data to the ground. Most balloon payloads are designed as custom hardware and software to fit the mission, which means common systems like GNSS, on-board data transfer and radio communications are re-implemented every time.

The satellite industry solves this problem by making use of buses for most missions. Instead of designing a custom, monolithic platform for each new satellite, manufacturers sell buses that provide utility (power, data bus, attitude control and space-to-ground communications) to which the customer or science team connect mission-specific payloads.

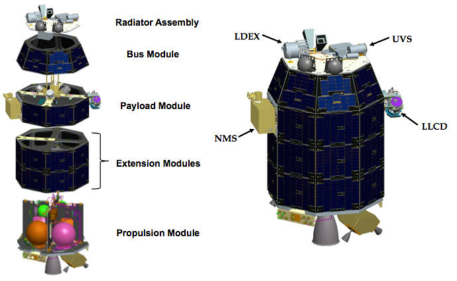

Figure 1: LADEE Spacecraft, based on the Modular Common Spacecraft Bus (NASA Ames)

Such modular designs allow mission specialists and scientists to focus on their experiments and payloads, and help reduce cost by spreading common development costs over many missions. Bringing their use to HAB missions would make high-altitude research accessible to a wider range of users, by taking away the computer science and electrical engineering required to design and build the flight computer and radio telemetry system.

What are the advantages, issues and obstacles in designing a modular intra-payload and long-range communications platform for High-Altitude Balloon scientific missions? The aim of the proposed investigation is to design, document, implement and evaluate the performance of a low-cost, modular platform to provide data transfer and long-range radio telemetry to payloads carried on High-Altitude Balloon missions. The objectives of the proposed project are:

- Evaluate existing platforms and protocols used for communications between payloads and over radio in high-altitude balloons and nano-satellites.
- Evaluate the feasibility of a modular High-Altitude Balloon bus based on low-cost, open-source, off-the-shelf hardware.
- Design the required protocols and build a prototype for an open-source high-altitude balloon flight computer and communication platform.
- Evaluate the power, data throughput and data loss performance of the prototype through testing.
- Evaluate the robustness, maintainability and ease of use of the bus' software compared to that of single-purpose payload firmwares.

2 BACKGROUND

Most HAB missions led by universities, schools or amateur are built around commercially available small computers or micro-controller boards: Redland Green School students used a Raspberry Pi (Raspberry Pi Foundation 2014) running python scripts as an OBC for their HAB mission (Hinschelwood et al. 2015), while other projects have been ran on AVR micro-controller based boards like the Arduino Mega (Atmel Corporation 2015).

In the United Kingdom, radio licenses do not cover airborne transmission, making packet radio protocols like AX.25 unusable for HAB missions (UKHAS 2016). However, the Office of Communications allows the use of certain *license-exempt* frequencies like the 70cm wavelength band, provided that the transmission power does not exceed 10mW (Ofcom 2014). For this reason, most missions launched from the United Kingdom use the Radiometrix NTX2 transmitter (Radiometrix Ltd. 2012). Because of the constraints, these missions usually transmit their telemetry as text using the Radio-Teletype (RTTY) protocol.

While most HAB projects make use of commercial-off-the-shelf components, the finished payloads are usually custom, single-purpose designs that are little, if at all modular. In the design document for the Titan-1 HAB mission, Bombasaro describes a hardware bus based on tightly coupled sensors which communicate over two different data buses, SPI and I2C, and a custom-made, single-purpose flight software (Bombasaro 2015).

Because of the risks and cost involved, nano-satellites rely more on modular designs, allowing each team to work on their module independently. In his paper, Volstad describes the design of the data bus of the NTNU CubeSat: while the OBC is a based on a custom made circuit board, it is designed to provide a standard power interface, as well as access to an I2C standard bus (NXP Semiconductors 2014) to each payload (Volstad 2011). The I2C protocol was chosen because of its low power consumption, and because if only requires two lines (clock and data). Thomas Clausen describes in his 2001 paper how a simple packet protocol built on top of I2C itself is used for data transfer and error detection in Aalborg University's CubeSats (Clausen 2001).

Satellite missions last a lot longer than High-Altitude Balloons' and the requirements for radio transmissions differ: while satellites can only communicate when their orbit passes above a ground station and require precise speed and data volume planning, as described by Sandy Anthunes in his book (Antunes 2015), non-floating HAB missions allow for line-of-sight communications from liftoff to late into the descent of the payload.

There are however some technologies that can be adapted from satellites to be used in HAB applications. The University of Arizona uses a custom packet format over 434MHz radio, containing raw binary values for each instrument's measurements (Eatchel et al. 2002). The BRITE-Austria CubeSat mission uses the AX.25 packet radio protocol (used by amateur radio users), which allows the use of off-the-shelf transmission and reception hardware rather than custom-made circuits (Traussnig 2007).

Some nano-satellites use custom telemetry data format, which allows the team to minimise the volume of data to be sent over radio. The Planetary Society's Lightsail mission has an uplink connection that allows the ground station to request specific logs or data. By default, The spacecraft only communicates a custom beacon containing a summary of the its housekeeping data (Ridenoure et al. 2016). Uplink being impractical with the transmission power limits imposed in the United Kingdom, such a selective telemetry system cannot be relied upon for HAB missions.

To improve collaboration and allow the deployment of large networks of satellites, probes and other space vehicles, the Consultative Committee for Space Data Systems has defined multiple communication standards that for the different types of networks encountered in spacecrafts.

SpaceWire defines how instruments' data can be accessed by having the OBC remotely poll their memory (Parkes and Mcclements 2005). A similar system is used to fetch data from ROM chips over I2C and could be used to build a simple data bus.

The CCSDS Space Packet protocol defines a packet format used to encapsulate application data (from instruments for example) that can be sent over a "Space Link", an analogue of the OSI model's link layer (Stallings 1987) using CCSDS frames to encapsulate packets originating from multiple instruments, ground stations and spacecrafts (Consultative Committee for Space Data Systems 2003).

While HAB applications do not require the level of complexity of CCSDS protocols — addressing is not needed since the network only contains two endpoints, communicating in a single direction — some details of the Space Packet protocol are worth reusing (Forward Error Correction, different Application IDs for each payload).

3 METHOD

The proposed project will consist of three main phases:

1. Design and documentation of the hardware and software interfaces, and the packet radio protocol of the payload bus.
2. Implementation of the payload bus design in prototype form (development boards).
3. Testing and evaluation of the prototype bus and its software architecture.

3.1 Technology

The payload bus prototype will be designed and built around an ATmega328p micro-controller (Atmel Corporation 2016). An Arduino Uno board will be used for convenience during development since it allows flashing firmware onto the micro-controller through USB (Arduino LLC 2010).

The OBC flight software will be written with the C programming language (Kernighan and Ritchie 1988) and the AVR C Library, part of the AVR GNU C toolchain (Gudger et al. 2016).

3.2 Planned Architecture

The main role of the payload bus will be to gather data made available by each connected payload, packetise it along with GNSS metadata and forward it to the ground station.

To decode the received data on the ground, a modified version of dl-fldigi (UKHAS and W1HKJ 2016) will be used to forward binary data to a decoding program, which will then decode packets and display each instrument's data and metadata either as text, or in a graphical user interface.

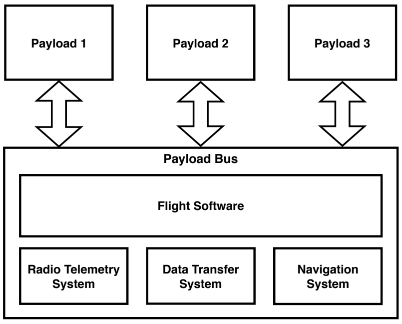

Figure 2: Planned payload bus architecture

The OBC flight software will consist of an event loop running throughout the flight. The major components of the loop would be:

GNSS tracking The tracking subsystem reads the GNSS receiver's output, and updates the last known position if the output is considered sane.

Payload polling Each payload — or the payload that have requested to transfer data — is polled for new available data. The data is transferred to the OBC.

Packaging The data from each payload is put into packets, and the latest GNSS data is inserted into each packet's headers.

Radio transmission The new data packets are added to the radio transmission queue and sent when the link is available.

At this time, two main issues are anticipated with this architecture:

- A payload generating large amounts of data could hold the bus for long periods of time, thus blocking other payloads. This could be mitigated using time or volume limits and a priority system — higher priority payloads would be given more time or data.
- Given the limited data rate that can be achieved over low-power radio links, a large number of payloads could produce data at a higher rate that could be sent to the ground station. Here again a data volume limit could prevent congestion.

3.3 Testing

The testing phase will aim to determine whether the bus prototype is reliable and fast enough to be used on High-Altitude Mission. Testing will be done in an incremental fashion: each module of the bus (hardware, flight software, radio link) will be tested individually at first, before systems are integrated together and larger-scale tests can be ran — up to full flight simulations can be done on the ground. The areas that will be tested are:

Throughput the maximum data throughput achieved by the radio link and the data bus.

Speed the average duration between the time a payload's data is available and the time it is sent to the radio transmitter.

Reliability The data loss ratio of the radio link and the data bus.

Power The current drawn by the OBC and the data bus depending on the number of payloads connected.

Since realistic radio ranges (40-200km) would be inconvenient to reproduce during ground testing, distance will be simulated by attenuating the radio signal artificially (different antennae). The performance of the bus will be determined by comparing the measurements against published numbers of previous HAB missions.

3.4 Risk Management

The main risks for the project that were identified are loss of project data, hardware failures and malfunctions, inaccurate time estimations and unforeseen health or external issues.

Loss of data will be mitigated by the use of regular and redundant local and on-line backups of the project data.

Hardware failure is the main risk: the electronic components used to build the prototype could damaged by simple errors in wiring or operations. This will be mitigated by the availability of spare development boards and careful handling of rare components.

To mitigate the impact of time estimation and health issues, the project will follow a three step process: In the worst-case scenario, achieving only the first step would still provide an acceptable minimal product. If the project runs well, the second and third steps will yield better, more complete results.

1. The radio packet format and the data bus' protocol are designed, and can be run in a simulated flight software (in-software) environment.
2. The flight software is complete and can drive a hardware payload bus and radio transmitter.
3. The payload bus prototype is complete and drives multiple test payloads, hardware-in-the-loop full mission tests can be run.

4 SUMMARY

High-Altitude Balloons are used to provide affordable access to near-space conditions to scientific experiments and teaching efforts, but still require significant engineering and computing knowledge to design. If successful, the proposed project would bring concepts used to reduce the cost and complexity of satellite operations by providing a modular interface to a standard payload bus. This would allow students and scientists not familiar with the problems of embedded programming and radio-communications to focus on their scientific payloads, thus reducing the barrier of entry to High-Altitude research and education.

REFERENCES

Antunes, S. (2015). *DIY comms and control for amateur space : talking and listening to your satellite.* 1st. San Francisco: Maker Media. ISBN: 1680450476.

Arduino LLC (2010). *Arduino - Arduino Uno R3.* URL: `https://www.arduino.cc/en/Main/ArduinoBoardUno` (visited on 09/21/2016).

Atmel Corporation (2015). *Sensing the atmosphere with an Arduino-based high-altitude balloon.* URL: `http://blog.atmel.com/2015/09/11/sensing-the-atmosphere-with-an-arduino-based-high-altitude-balloon/` (visited on 10/30/2016).

– (2016). *ATmega328/P Datasheet.* URL: `atmel.com`.

Bombasaro, E. (2015). *Titan 1 design and mission documentation.* In:

Clausen, T. B. (2001). *The Cubesat Internal bus: The I2C.* Aalborg.

Consultative Committee for Space Data Systems (2003). *Space Packet Protocol.* Washington, DC.

Eatchel, A. L. et al. (2002). *Development of a Baseline Telemetry System for the CubeSat Program at the University of Arizona.* In: *International Telemetering Conference Proceedings.* Tucson, AZ: International Foundation for Telemetering.

Gudger, K. et al. (2016). *AVR-GCC Toolchain.* URL: `http://www.nongnu.org/avr-libc/`.

Hinschelwood, J. et al. (2015). *A Raspberry Pi Weather Balloon.* In: *Young Scientists Journal* 17, pp. 20–24.

Kernighan, B. W. and Ritchie, D. M. (1988). *The C programming language.* Prentice Hall. ISBN: 9780131103627.

NXP Semiconductors (2014). *UM10204 - I2C-bus specification and user manual Rev. 6.*

Ofcom (2014). *IR 2030 - UK Interface Requirements 2030 Licence Exempt Short Range Devices.* London.

Parkes, S. and Mcclements, C. (2005). *SpaceWire Remote Memory Access Protocol.* In: *DASIA 2005-Data Systems in Aerospace.* Vol. 602. European Space Agency, pp. 18.1–18.9.

Radiometrix Ltd. (2012). *NTX2/NRX2 Data Sheet.* Harrow. URL: `http://www.radiometrix.com/files/additional/ntx2nrx2.pdf`.

Raspberry Pi Foundation (2014). *Raspberry Pi 1 Model A+.* URL: `https://www.raspberrypi.org/products/model-a-plus/`.

Ridenoure, R. W. et al. (2016). *Testing The LightSail Program: Advancing Solar Sailing Technology Using a CubeSat Platform.* In: *Journal of Small Satellites* 5.2, pp. 531–550.

Stallings, W. (1987). *Handbook of computer-communications standards.* Macmillan. ISBN: 002948071X.

Traussnig, W. (2007). *Design of a Communication and Navigation Subsystem for a CubeSat Mission.* Graz.

UKHAS (2016). *UKHAS - Frequently Asked Questions.* URL: `https://ukhas.org.uk/guides:faq?s[]=license`.

UKHAS and W1HKJ (2016). *dl-fldigi.* URL: `https://ukhas.org.uk/projects:dl-fldigi`.

Volstad, M. L. (2011). *Internal Data Bus of a Small Student Satellite.* Ålesund.