WhitSat Technical Report Series

WHITWORTH SPACE EXPLORATION

High Altitude Balloon Command Module Software

WhitSat TR-002

John M. Larkin

Version 2.0, 2021-07-26

Table of Contents

Summary 1
1. Iridium Short Burst Data 2
2. Persistent Memory 3

Summary

The High Altitude Balloon Open Source Collection (HABOSC) is a set of interrelated tools for transmitting telemetry from a high altitude balloon, processing and storing that data, and providing user access to the stored data both during and after the flight. The system is designed to be flexible so it can adapt to the needs of various users.

The components of the collection are:

Command Module (hardware)

This is the essential hardware system carried by the balloon. The core tracking features are implemented using a GPS receiver and an Iridium satellite modem. The operations of the Command Module are coordinated by a microcontroller (in the ARM Cortex-M family). Bluetooth communication allows a local computer using Mission Control (described below) to configure the Command Module for the flight. Communication with experiment pods containing a Flight Computer (see below) is possible through a XBee 802.15.4 radio network operating at 2.4 GHz.

Command Module (microcontroller software)

Written in C++ using the ARM Mbed API and RTOS (real-time operating system), this is the code running on the microcontroller that ensures measurements are made and then transmitted to the Iridium satellite network.

Mission Control (server software)

The Python code running on the server listens for telemetry packets relayed by the Iridium ground station. The telemetry data is processed and stored in a MongoDB database. Users interface with Mission Control through a web frontend based on HTML and Javascript. Close to real-time mapping and graphical display of key telemetry is provided. Recent versions of several web browsers allow access to a local computer's serial ports, enabling Mission Control to use a computer's Bluetooth connection to send mission configuration details to the Command Module. This also allows authorized users to monitor communication between the Command Module and linked Flight Computers during the prototyping and debugging phase on the ground.

Flight Computer (hardware)

This printed circuit board holds an Arm Cortex-M3 microcontroller. The microcontroller supports reading sensors and logging data to the attached SD card. An XBee radio modem allows the microcontroller to synchronize its real-time clock with the Command Module as well as to relay data to the Command Module for transmission to Mission Control. An integrated rechargeable battery provides a reliable power source. A version of the Flight Computer also provides a base for attaching a breadboard if custom circuitry is required.

Flight Computer (software library)

This Mbed-based library allows users to easily integrate communication with the Command Module into their C++ code.

The remainder of this document describes the software portion of the Command Module.

1. Iridium Short Burst Data

Bytes	Description	Type
0-1	Mission ID (positive = in-flight, negative = pre- or post-flight)	`int16`
2	Bits: 0 = GPS fix, 1 = heading sign, 2-7 = pod active	—
3-6	Time of GPS update (Unix time?)	`int32`
7-10	Latitude (in 100,000th of a minute)	`int32`
11-14	Longitude (in 100,000th of a minute)	`int32`
15-16	Altitude (in meters)	`uint16`
17-18	Vertical velocity (in tenths of m/s)	`int16`
19-20	Ground speed (in tenths of km/h)	`uint16`
21	Heading relative to true north (0-180 deg, sign in bit byte)	`uint8`
22	Command module battery voltage (in units of 0.05 V)	`uint8`
23-24	Internal temperature (in units of 0.01 deg C)	`int16`
25-26	External temperature (in units of 0.01 deg C)	`int16`

2. Persistent Memory

Bytes	Description	Type
0-1	Mission ID	`uint16`
2	Flight status 0 = lab 1 = launch	`uint8`