July HASP status report from the SKC Wide Field Camera team

Overall status:

The SKC Wide Field Camera is ready for integration. SKC will fly its primary camera in payload position 4, and it's backup camera (which is essentially identical to the primary) in payload position 6. The primary will be left in Palestine after integration, and the backup will be brought back to Montana and reintegrated before flight. Bringing the backup back to SKC allows the team to practice flight operations during August.

Accomplishments:

(1) The issue with the SD card writes not working mentioned in the June report was resolved early in July. The problem turned out to be two things. The Altera University Program SD card IP core running on our FPGA only works with SD cards made by certain manufacturers. We've found it works with 2 GB SanDisk and Toshiba non-HC microSD cards. We will have to make a slight reduction in the rate of image acquisition (one full resolution image every 90 seconds rather than every 60 seconds) because 4 GB non-HC microSD cards apparently are no longer made by these manufactures. Larger capacity cards are only available in the HC format, which is unsupported by the Altera SD card IP core. At the 90 second rate we can store 750 images on one 2 GB card. For an 18-hour flight at 40 images an hour each camera would acquire 720 images. The other problem was the Altera SD IP core is incompatible with the most recent versions of Quartus (the software used to compile configurations for Altera FPGAs). The fix was to downgrade to Quartus version 9.0.
(2) The writing of the flight software is completed.
(3) Flight PCB boards were ordered and received. They worked as designed out of the box.
(4) We had some difficulty in getting the FPGA configuration and C code onto the EPCS device and then load onto the FPGA at power on. Having that capability enables the payload to automatically come up running at power on. This is needed in case the payload needs to be powered down in flight, perhaps for cooling the electronics, and eases starting up the payload on the flight line. We now have this capability working. The issue was the rather brief Altera documentation describing the process required extensive experimentation to fill in the details missing from the documentation.

Issue:

The DC-to-DC converter on the interface PCB runs rather hot to the touch, and we have some concern it might get too hot sealed inside the payload box. We plan on asking for the advice of people at CSBF during integration on how to deal with this, and we will test these ideas during the Wednesday thermal/vac.

PSIP addendum:

We would like to make two changes to the PSIP.

(1) We've added three bytes to both the downlinked thumbnail serial record and downlinked temperature record that shows the address on the SD card of where the last full resolution image ends. If this address is observed to be changing at a rate of once every 90 seconds it provides evidence that images are being written to the SD card as expected. The changed serial records

are shown below.

Byte	Description
1	Record type (value = 0x01)
2-7	Timestamp (lowest six base-10 digits of the ten base-10 digits specifying the number of integer seconds of the last Unix timestamp received from the HASP system, ASCII hex values from 0x30 to 0x39)
8-10	Highest three bytes (of four) of the address on the SD card of where the last image data written to the card ends
11	Image line number (value from 0x01 to 0x80)
12-331	Pixel data (two bytes per pixel, 160 pixels in one line, values from 0x00 to 0xFF)
332	Line checksum (8 least significant bits of the sum of the values of bytes 11-331)
333-41226	Bytes 11-332 repeated 127 more times for one complete thumbnail image

Serial Data Record for a 160 x 128 Thumbnail Image Downlink

Byte	Description
1	Record type (value = 0x02)
2-7	Timestamp (lowest six base-10 digits of the ten base-10 digits specifying the number of integer seconds of the last Unix timestamp received from the HASP system, ASCII hex values from 0x30 to 0x39)
8-10	Highest three bytes (of four) of the address on the SD card of where the last image data written to the card ends
11-18	Temperature data in order from sensor 1 to 8, one byte from each sensor (values from 0x00 to 0xFF)
19	Command number of the last uplinked command (value from 0x00 to 0xFF)
20	Last uplinked command status (value from 0x00 to 0xFF, the interpretation of the value is customized for each command)
21	Checksum (8 least significant bits of the sum of the values of bytes 1-20)

Serial Data Record for Temperature Data Downlink

(2) With the last image address information coming down in the serial records we would like to dedicate both analog lines to providing temperature information (rather than one line for temperature status and one line for image acquisition status as specified in our PSIP). We will hardwire one thermistor to each analog line to give a direct read of the DC-to-DC converter temperature and FPGA temperature. With this approach we get temperature data from these key components even if the payload is powered off for cooling, and more frequent temperature readout when the payload is powered on.

Question:

We plan on requesting a downlink of a 41226-byte thumbnail serial record every 30 minutes, and more frequent downlink of the 21-byte temperature record. Would it be easier for the HASP operator if our flight software just automatically sent out the thumbnail and the temperature data at predetermined intervals (a thumbnail every 30 minutes, and a temperature record every 4.5 minutes)? Our current software requires an uplink command to trigger the thumbnail and temperature record transmissions. We can easily change the software to automate this and that would substantially reduce the number of our uplink command requests to the operator.