Upper Stratospheric Flow Velocities and Data Gathering

Cameron Butler, Lorenzo Narducci, Chukwuma Odigwe, Michael Owca, Blaire Weinberg[*]

The High Altitude Student Platform (HASP) program is a Louisiana Spacegrant (LaSpace) and NASA Balloon Program Office (BPO) sponsored flight on a high altitude balloon at roughly 120,000 ft for 15 to 20 hours. The University of Maryland's Balloon Payload Program (BPP) developed a scientific payload, High Altitude Atmospheric Turbulence-Triggered Release Information Carrier (HAAT-TRIC), which is split into two major subgroups. The first subgroup, HAAT, measured high altitude flow velocities and temperatures with two thin-film anemometer probes extended from the payload. The data from HAAT will help better characterize clear air turbulence in the mid to upper stratosphere where little data currently exists, in an effort to aid the development of hypersonic vehicles that will operate at these altitudes. The other subgroup, TRIC, is a proof of concept data drop module that will store HAAT data until commanded to drop, severing all connections with the main payload and being stored in a hanging net. An actual drop can be used on long duration balloon flights to receive data well before balloon termination and recovery.

Nomenclature

BPP: Balloon Payload Program
Balloonduino: Ballooning Arduino Mega (developed by Camden Miller, UMDBPP)
UMD: University of Maryland (College Park)
MdSGC: Maryland Space Grant Consortium
HASP: High Altitude Student Platform
DRS: Data Relay System
DAS: Data Acquisition System

[*] University of Maryland, College Park, MD, 20742, Department of Aerospace Engineering

I. Introduction

HAAT-TRIC is a two system payload that flew on a zero pressure balloon. Split into two main portions with an inter-section Data Relay System, HAAT-TRIC uses two hot filament probes to measure temperature fluctuations to characterize the flow velocities in the mid to upper stratosphere and collects that data into a separate module that will drop from the main balloon to be recovered mid flight. The data relay system takes the high volume of data collected by the anemometers in the scientific module, HAAT, and sends it to the data module, TRIC.

Little is known about the flow velocities in the mid to upper stratosphere around 100,000 ft where hypersonic vehicles are being designed to fly at cruise. By characterizing these flow velocities, hypersonic vehicle designs can be more accurately designed to avoid turbulence in the flow across the vehicle. The HAAT system is designed to begin to characterize the velocities on a scientific balloon flight. While a weather balloon only reached about 70k ft, the data recorded throughout the flight was promising. By flying on HASP, additional flow velocity data could be guaranteed at or above 100k ft for a minimum 10 hours, neither of which could be guaranteed on the previous flight.

The idea of an early recoverable data module stems from Antarctic NASA balloon flights where the balloon can remain in the upper atmosphere for several months. Data collected on these flights cannot be recovered until after the flight is over as the volume of data is too great to transmit. With a recoverable data module, data can be collected early to be analyzed before the mission is over. The module can also be released in an area of recovery that is easier to access than if a balloon flight were to take its full course and land in an area that could be hard to reach. TRIC was designed as a proof of concept that large volumes of data can be captured on a separate module from the main scientific payload and terminated from the main payload and data source. On this flight, TRIC was dropped into a net as technology demonstration that a data relay system could be properly terminated from the main data unit while in flight, without the risk of losing flight hardware.

II. HAAT

II.A. Introduction

Present-day hypersonic vehicles are being designed for cruise altitudes near and above 100 kft (i.e. mid to upper stratosphere), where they are subject to extreme aero-thermo-acoustic environments which can strongly impact the boundary layer over the vehicle. Transition to turbulence is often accompanied by significant increases to surface heating rates, friction drag and acoustic loading, all of which degrade vehicle performance.[1] It is therefore of utmost importance that transition onset can be accurately predicted during the design process.

For the disturbance environment typically encountered during flight, hypersonic transition is caused by external disturbances exciting unstable modes within the boundary layer through receptivity mechanisms. These unstable modes then grow in amplitude as they propagate downstream until the development becomes nonlinear. At this point, these modes break down into turbulent spots, which coalesce to form a turbulent boundary layer.[2] The key takeaway from this discussion is that transition depends not only on the boundary flow, but also on the freestream disturbance type, intensity and spectrum.[3] Conventional ground-testing facilities have been shown experimentally to generate much stronger freestream disturbance environments than real flight, so results cannot be extrapolated to flight performance[4].[5] Current techniques for predicting in-flight transition location combine linear stability analysis with information regarding the initial disturbance amplitude and breakdown criteria.[6] Thus, in order for the efficacy of these techniques to be assured, there must exist accurate information regarding the disturbance environment in flight. This provides the fundamental motivation behind the design and implementation of the High Altitude Atmospheric Turbulence (HAAT) subsystem.

The stratosphere is generally a region of relative thermodynamic stability because of the positive vertical temperature gradient, but it is nevertheless intermittently populated with regions of clear air turbulence (CAT), which can extend hundreds of meters in the vertical direction, tens of kilometers laterally, and minutes to hours temporally. The ultimate goal of HAAT is to record temperature and velocity fluctuations at HASPs float altitude, facilitating the characterization of CAT in the mid to upper stratosphere, which remains inadequately studied relative to the troposphere.

Measurements of the temperature and velocity fluctuations are taken by means of a mini Constant Voltage Anemometer (mini-CVA) provided by Tao Systems. CVA operating principles allow for extremely fast-response measurements (450 kHz), which make this type of probe particularly well-suited for measuring turbulent fluctuations. Thermal anemometry takes advantage of the relationship between the change in temperature of a thin film or wire caused by contact with an external flow and the resulting change in that wires resistance. The basic circuit behind a CVA is shown below in Figure 1, where all resistors are fixed aside from R_w, the resistance of the wire, and Vs is the output voltage of the circuit.

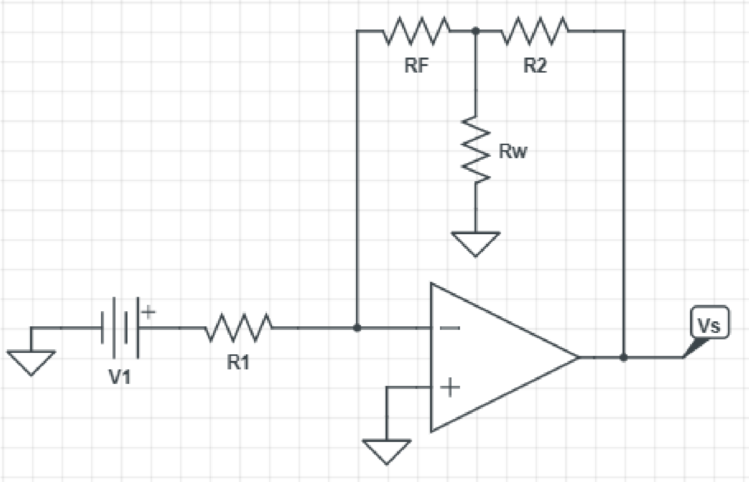


Figure 1: CVA Circuit Diagram

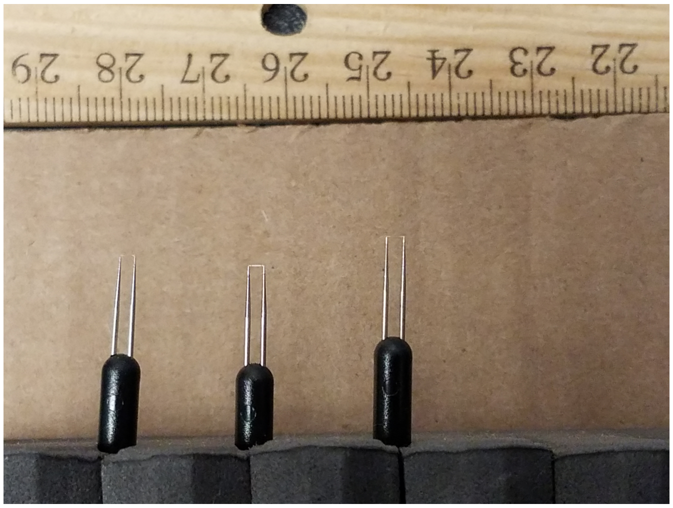


Figure 2: Thin Film Probe

An example of one of the film probes intended for HAAT is shown in Figure 1, having a diameter of 0.002 in. As the wire and the flow exchange heat, the wires resistance will change, which then creates an output voltage signal within the anemometer circuit given by Equations 1 and 2 . Depending on the amount of current applied to the wire, one can make measurements of the flow temperature or velocity. In order to observe the temperature, a very small amount of current is applied to the wire; to measure the velocity, a large current is applied.

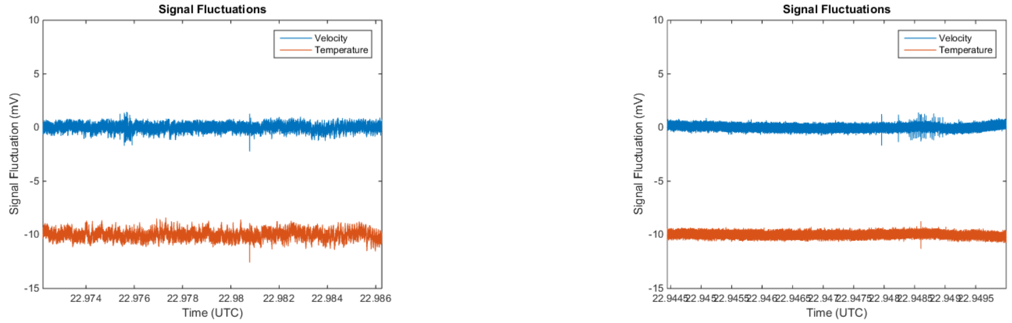
$$V_w = \frac{R_f}{R_w} V_1 \tag{1}$$

$$V_s = (1 + \frac{R_2}{R_f} + \frac{R_2}{R_w}) V_w \tag{2}$$

II.B. Results

Although data reduction is ongoing, preliminary results from the flight are encouraging. Both temperature and velocity channels were sampled at a rate of 30kHz, allowing for resolution of fluctuations up to the Nyquist frequency of 15kHz. The entirety of the flight data is shown in Figure 4a , where the data has been put through a high-pass IIR filter to remove any signal below 0.5 Hz caused by the rocking or rotation of the gondola and the temperature signal has been given an offset to facilitate viewing both signals simultaneously. Converting these signals to physical units requires a complex calibration procedure which has yet to be performed, so the data is reported in terms of voltage. Note that the time scale here is in hours and the green dashed line seen in Figure 4b is when float was reached. Even at these time scales, there are easily-identifiable bursts of more energetic flow seen simultaneously in both signals. Several of these regions have been circled in Figures 4e, 4g, 4h, 4i, 4j.

These regions can be better characterized by examining the frequency spectrum of the fluctuations. The region circled in Figure 4j at around UTC 23:00 can be analyzed to show both qualitative and quantitative differences between calm, laminar regions and regions of CAT. In order to avoid averaging out the small time scale fluctuations related to turbulence, a 20s interval was chosen within the high energy pocket of interest. The high pass filtered data for this region is seen compared alongside another 20s interval occurring just before this pocket of fluctuations begins. When looking at such a short segment of data, the differences between calm and energetic regions immediately become clear. The energetic region has clear frequency content, whereas the calm region appears mostly flat.

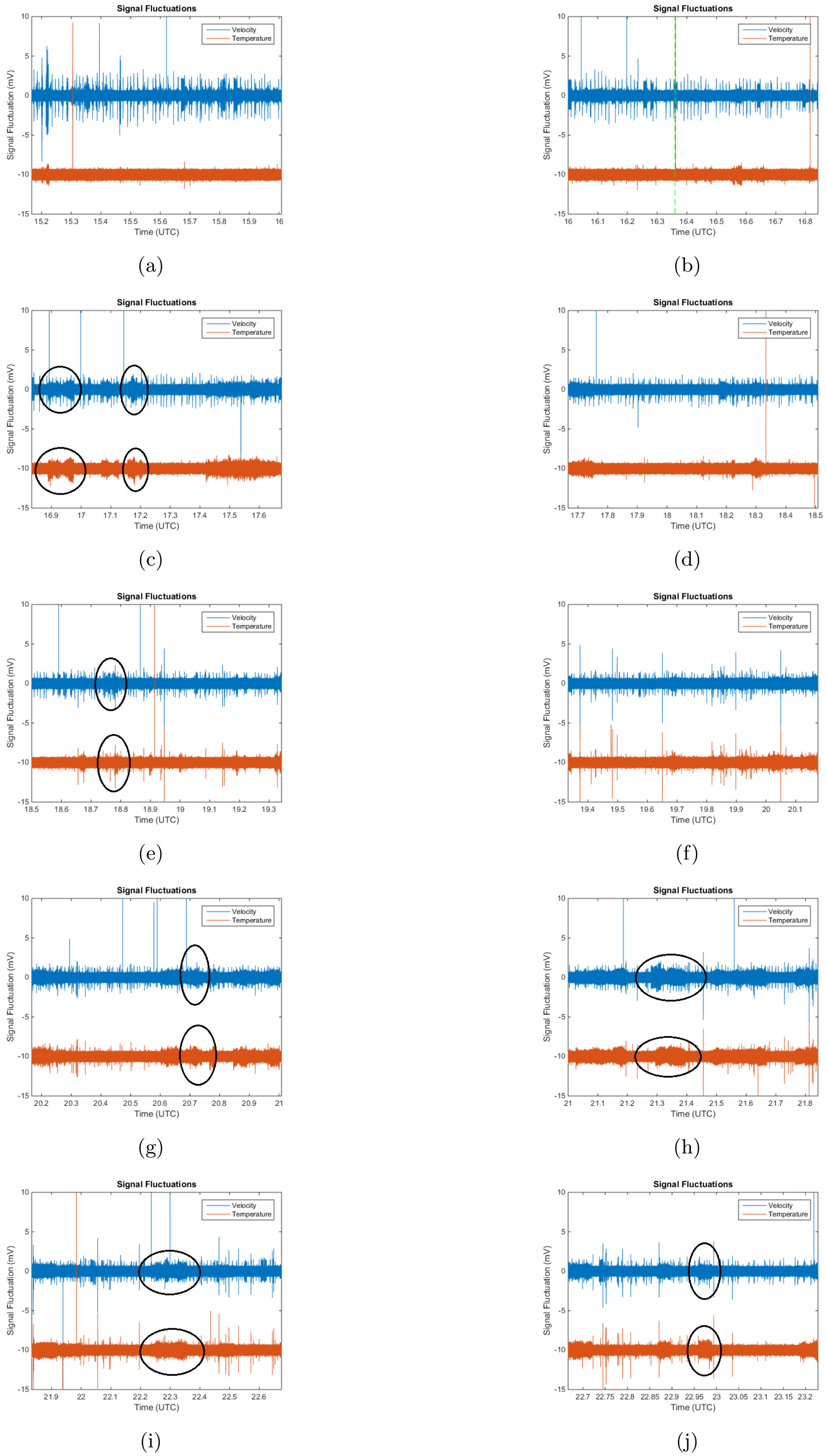


(a)

(b)

Figure 3: Signal Fluctuations

Further insight can be obtained by looking at the Power Spectral Density (PSD) for each of these regions. At lower altitudes, it has been shown that turbulent fluctuations in temperature and velocity display a $m^{-\frac{5}{3}}$ scaling in the wavelength band from 10cm-1m, where m is the wavenumber of the disturbance, and a m^{-7} scaling from 10cm down to the Kolmogorov scale, which tends to be on the order of 1mm within the atmosphere.[7] Unfortunately, the lack of information regarding the freestream velocity of the flow relative to the probes makes converting the PSD from frequency to wavenumber impossible, but these scaling laws may still be observed. The PSD for the temperature and velocity fluctuations in the two regions shown above are given in Figure 5a, 5b, 5c, 5d, with the -5/3 and -7 scaling lines shown in red and green, respectively. Immediately it can be seen that, for the disturbed region, there is a -5/3 decay in the signal up to about 8Hz, after which the decay follows the -7 line down to about 20Hz. As we would expect, this is not the case for the undisturbed flow. There is almost no frequency content in the signal over the entire spectrum.



(a)

(b)

(c)

(d)

(e)

(f)

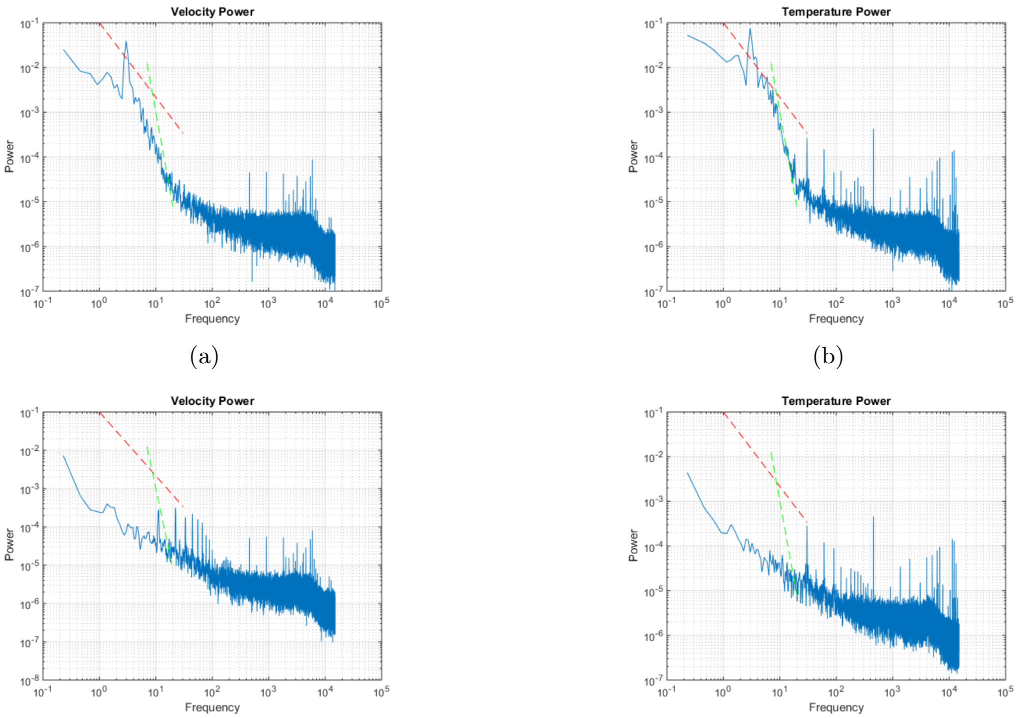
(g)

(h)

(i)

(j)

Figure 4: Data Graphs



(a)

(b)

(c)

(d)

Figure 5: Velocity and Temperature Power

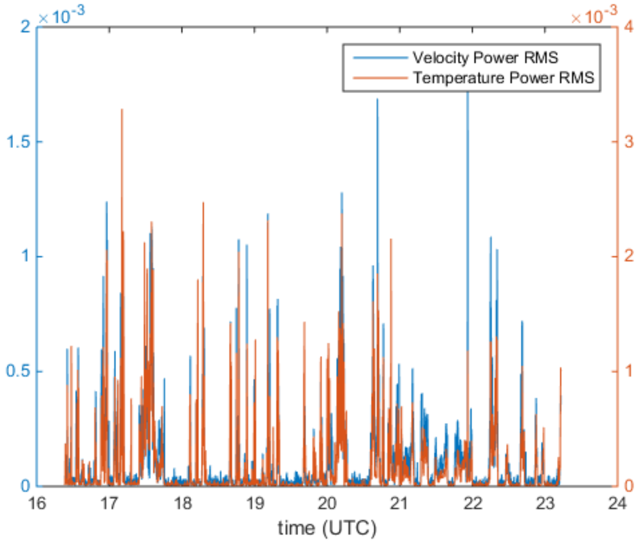


Figure 6: RMS

The striking difference in the spectra of these two regions suggests that the RMS of the spectrum may be used to automatically identify pockets of significant CAT. We can see from Figure 6 that there are clear spikes in the signal rms. This plot gives us significant information regarding both the strength and intermittency of the freestream disturbances occurring over the course of the flight. Based on this plot, additional regions of CAT can be easily identified for further analysis.

II.C. Conclusions

Preliminary data reduction from the 2017 HASP flight shows promise for identifying and characterizing regions of CAT in the mid to upper stratosphere. By examining the RMS of the signal PSD, a threshold criterion can be employed to determine the duration and intermittency of turbulent spots. By combining this with GPS data provided by the HASP team, the horizontal span of CAT regions may be determined. Additionally, by performing a calibration procedure on the CVA, the signals may be converted from mV to physical quantities. This will allow for the amplitude of the disturbances to be determined in more useful units. However, these conclusions rely upon the assumption that the turbulence being observed is indeed CAT and not the result of the sensors entering the wake of the balloon or gondola. In continuing to analyze this flight data, some effort must be made to determine when the probes are facing into the freestream and theoretically outside the balloon wake. This may be possible by studying the unfiltered velocity signal.

III. TRIC

III.A. Data Module

TRIC is the droppable module. Data is transferred via a magenetic USB cable, and stored on a Raspberry Pi. TRIC also took pictures using a Raspberry Pi Camera board. After four hours of flight time, the release mechanism actuated, dropping the data module into a plastic mesh netting set up around HAAT-TRIC. The netting was secured to the payload structure by four quarter-inch eye bolts. The module successfully disconnected from the main payload, falling into the net and breaking the data tether while still taking pictures for the rest of the flight.

Previously, the drop module had been flown multiple times on BPP flights to varying degrees of success. Lessons learned on those flights helped lead into the successful drop of the module; mostly with the actuating system. On typical BPP flights, the mechanical actuator was a separate payload above the payload it was dropping. For the HASP flight, the mechanical system was conjoined with the data module via two U bolts secured by the actuators until the drop.

III.B. Release Mechanism

Current BPP payloads utilizing similar methodologies for dropping payloads utilize internal timers or radio command signals to actuate dropping. The HASP release mechanism was set to drop after either receiving a signal from the gondola or after a set four hours. As later discussed in §VI.B, future iterations of TRIC aim to utilize serial communication to command the drop of the payload.

The system consisted of two 10 mm stroke actuators wired to a Balloonduino to both move forwards to lock the drop module in place, and move in reverse to drop the module. The actuators used quarter inch aluminum pins to hold the u bolt mounting brackets. When signaled, the Balloonduino would retract the pins, allowing the drop module to fall.

The mechanical release mechanism was adapted from Mechanically Actuated Release System (MARS), a BPP project meant to drop other payloads from the payload string mid flight. As the command center of the BPP payload strings are a separate unit flying on the payload string, MARS utilizes either internal timers or radio signals via XBee to communicate between the ground, command module, and MARS payload. The HASP system, however, utilized the contained gondola command center to signal a drop via a discrete signal (a line pulled low for 100 ms) or an internal timer. As mentioned in §VI.B, serial commanding is also being investigated for future iterations of the payload. Additionally, MARS utilizes only a single actuator, while the HASP drop module utilized two for redundancy and safety.

IV. Data Relay

Raw data acquired from the hot filament probes was an analog signal split to be stored in both the DAS (located in HAAT) and on a Raspberry Pi located in TRIC. While the DAS signal was logged as is, the raspberry pi signal was passed through an ADC to store digital signal data on the Raspberry Pi, such to provide a proof of concept for in-flight data conversions to aid in ease of post processing. A Functional Block Diagram of the data relay system can be seen in Fig 7.

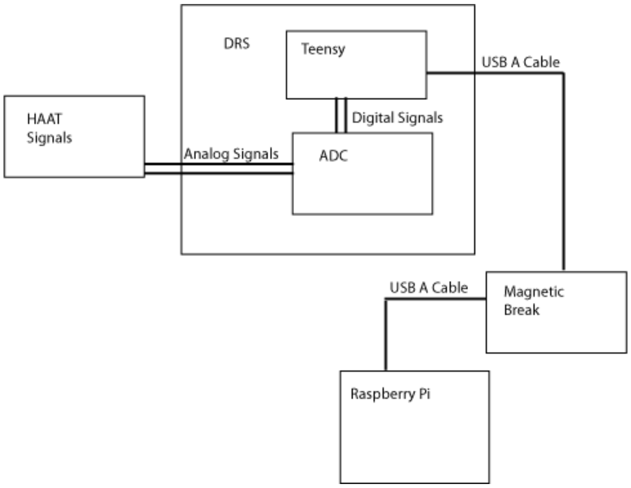


Figure 7: Functional Block Diagram, Data Relay System

IV.A. Data Conversion

Analog signals from HAAT were converted using the Analog Devices ADAS3022 ADC. This ADC is capable of operating at speeds up to 1 MSPS and was capable of receiving either differential or single-ended signals. This versatility and speed was attractive for converting data from the the CVA because of the need for the sampling of temperature and velocity data at a speed of at least 100 kHz. In order for the ADAS3022 ADC to operate properly, a system of components was needed to successfully power and calibrate it.

A microcontroller was also needed in order to enable the ADC to convert incoming data and then send it to the Raspberry Pi in the TRIC drop module. The TEENSY 3.2 microcontroller was an attractive option because it allows for a range of operation similar to the Arduino Uno microcontroller but is only 1.4 inch by 0.7 inch in area, reducing necessary space. The DRS printed circuit board would house the ADAS3022 ADC, components for its operation such as the Analog Devices ADP1613 step up DC-to-DC switching converter to power the ADC at 15V, other components such as decoupling capacitors and resistors, and the entire TEENSY 3.2 microcontroller itself. The newly converted data from the ADC would then be sent from the TEENSY 3.2 microcontroller to the Raspberry Pi via a magnetic USB cable. The full DRS schematic can found in the Appendix.

IV.B. Data Storage

Data sent from the Teensy 3.2 was to be transferred via a magnetic USB Micro-B - USB A connection, and stored on a Raspberry Pi. The magnetic interface between the cables was used to disconnect the data line without damaging hardware when TRIC was dropped, as the impulse from dropping was enough to free the two halves of the magnet, ending the transmission of data.

Received digital signals was stored on a Raspberry Pi, as well as images from a connected Raspberry Pi Camera. Both systems were run by a script that was programmed on the Raspberry Pi to run immediately upon successful power on and boot of the system. During initial testing of the Raspberry Pi and Teensy interface, the Raspberry Pi would store only 10 lines of data, and then proceed to only store images.

By the date of the HASP flight, a conclusive reason for why the data was not storing had not been determined. The software for data reading was not starting on boot, while the image capture software was. The decision was then made to only capture images on the 2017 flight. Once the Raspberry Pi was received, it was unable to log into the system upon power on although the system could maintain stable power. After returning from flight, the Pi was able to be run from a separate TTY window that did not require lockfile generation upon password input. It was determined that the reason for the difficulty in logging into the Raspberry Pi was due to the fact that the SD card had been filled with images and there was no memory available to open the login lockfiles.

V. Flight Results

V.A. Software

The Raspberry Pi took a total of 3967 images throughout the flight. Table 1 shows the timestamps of significant events throughout the flight.

Time	**Event**	**File Number**
Sept. 1, 1947h	Pi turns on	0
Sept. 2, 0155h	TRIC drops	3311
Sept. 2, 0230h	Pi Memory fills	3624
Sept. 2, 0308h	Battery dies	3967

Table 1: Table of Flight Images

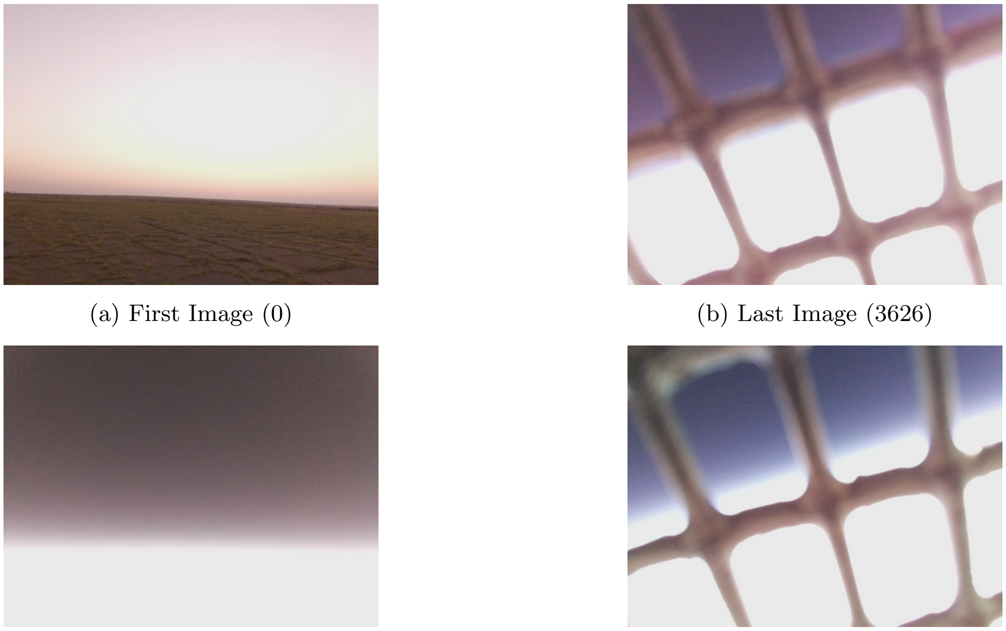
While the dates and times listed in 1 are not consistent with the dates and times of the launch (14:04:25 UTC Sept. 4)[8] , it is assumed that the clock on the pi is internally consistent with itself and the times shown accurately depict the passage of time.

From Table 1, it is noted that TRIC was dropped 6 hours and 8 minutes after the Raspberry Pi was turned on. After 0230, the image size drops to 0 bytes, although the image timestamp and file name are recorded. From that file size and the reason the pi could not be logged into was that the memory was full, it was concluded that the memory filled up on the raspberry pi 6 hours and 47 minutes after the pi turned on. The last image timestamp was recorded 7 hours and 21 minutes after the Raspberry Pi turned on, which is within a reasonable estimate for shutoff due to a low voltage battery. The battery was returned bloated in appearance and at significantly lower voltage than the LVC was set to, so it is still undetermined whether pi shutdown was a result of the battery dying or the LVC shutting off the power supplied to the pi.

Using the total number of pictures and total time recorded it was determined that a picture was taken every 6 seconds, or roughly 10 images per minute. This is a significantly more frequent rate of image taking than was set in the software (once every minute), and can likely explain some of the issues with the data storage of the Raspberry Pi, most of which can be summed as a buffer overflow. The Raspberry Pi would successfully read the data from the USB cable for the first 9 data transfers, and then take a picture. After taking a picture, the pi was supposed to record data for another minute, before taking another picture. In actuality, the pi would be unable to read more data from the buffer after taking the initial picture. Instead of taking a picture every 60 seconds, taking one every 6 seconds left little to no time for the Raspberry Pi to read and store data from the serial buffer. Initial investigations are unsuccessful in understanding why this occurred, as the image rate is set to 60 seconds in the software, and more analysis needs to be done to fully understand where this image rate came from.

V.B. Images

As noted in §V.A, 3967 images were saved on the Raspberry Pi, with 3624 actually containing valid images.



(a) First Image (0)

(b) Last Image (3626)

(c) Image Before Drop (3309)

(d) Image After Drop (3310)

Figure 8: Flight Images

The successful drop of TRIC can be seen in Figures 8c and 8d. Unfortunately, the camera was oversaturated with light, and the images that were taken (see Figure 8c) only show a white earth and the blackness of space.

VI. Future Work

VI.A. Data Relay and Acquisition

Most of the future work planned for TRIC involves improved iterations of the DRS. These changes aim to avoid some of the communications problems prevalent with the current version (Rev. 2), as well as expand capabilities for a broader range of use in an effort to eventually use as a standalone electronics board for a early termination data recovery system.

The most drastic changes to be made to the DRS system involve making it a self-sufficient system: adding a microcontroller and onboard storage system to avoid having to interface with the TEENSY 3.2 and Raspberry Pi, respectively. The addition of the microcontroller would allow for more functionality and options when adding additional sensors or components to the system (e.g. radio antenna and GPS). Switching to an onboard data storage method and replacing the Raspberry Pi would alleviate many of the processor problems that occurred during the 2017 HASP flight, mostly with respect to data storage. Additionally, many functionalities of the Raspberry Pi such as commanding signals with the GPIO pins were not utilized, and the Raspberry Pi was only used for image capture and data storage to a MicroUSB chip. The same functionality can be achieved on the DRS board, alleviating the necessity for the Raspberry Pi.

In order to achieve the goal of a more generally applicable board that can be used for many different projects, the DRS needs to accommodate both digital and analog data streams. The data line of the MCU would have two branches; one for digital input and the other for analog, with the latter routing to an ADC before both go to the MCU. With inputs for both data types, the physical signal cable can be routed between the two payload components, instead of having to add an ADC board specifically for analog signals. This also eliminates the need for the TEENSY 3.2 microcontroller, which operated the ADC on the previous board revisions, as this can be done by the board's MCU.

Without needing the TEENSY 3.2 or Raspberry Pi processors, the cable between the two payload components has no requirements, whereas the TEENSY 3.2 needed a USB Micro-B termination and the Raspberry Pi cable needed a USB - A termination. The magnetically detachable USB-A cable head used to terminate and detach the drop module occasionally got stuck in one of the boxes and did not properly detach from the module. While this is unlikely to happen in a real drop as movement from the balloon will likely tear the cable out in case of a lock, better solutions can be found with no cabling requirements between the two payloads.

Any major functionality changes that may need to be made by a user are to be hardware-based instead of software (e.g. changing the input voltage for a low voltage cutoff system). In doing so, this maintains the software that needs to be written as a stable source code, and does not need to be edited after being verified on the system and uploaded to the board. It allows for ease of end user functionalities such that there is no need to install drivers, libraries, etc, in order to use the board as needed for any one application.

Due to changes in the DRS, a new release system will need to be put in place to terminate the connection between TRIC and HAAT. While the magnetic USB cable worked well between the serial interface of the Teensy and the Raspberry Pi, a new system will need to be created to interface the raw signals coming from HAAT with the new DRS. This system has yet to be developed, and continues to be a main point of research for upcoming revisions.

VI.B. Telemetry

The two methods of telemetry provided on a HASP flight are serial communications and discrete signals. HAAT-TRIC utilized the latter of these to signal the drop of the drop module. Due to inconsistencies in the balloonduino's ability to properly check and execute a command during the 100 ms low signal generated from a discrete signal, future payloads would utilize serial command capabilities.

Serial uplink command capabilites have many uses in the entire HAAT-TRIC payload. TRIC could use uplink to trigger the release of the drop module, and HAAT could use them to reset, clear, or turn on the DAS system. These functionalities come as a result of both increased versatility with serial uplink, as well as the signals' repeatability and duration.

Downlinking telemetry also has many uses for HAAT-TRIC. This functionality was implementable for the past flight, but a hardware problem on the balloonduino prevented serial from being utilized. Both environmental data from balloonduino's onboard sensors and status updates from HAAT and TRIC could be sent down to the ground to inform ground team members about current payload conditions. If the drop module were ever to be truly dropped completely from the gondola, this information can help determine when to drop the module.

VII. Closing

Flow velocities in the 100k ft regime have not been explored in great detail and HAAT-TRIC has begun to collect data in this area, specifically CAT events. By learning more about the mid and upper atmosphere's flow, hypersonic vehicles will be able to design around experiencing turbulence so that the structure does not overheat. Early experiments with HAAT proved to be promising and the HASP flight recorded several CAT events. Further flights will help back up the initially findings.

TRIC has shown the ability to collect large amounts of data and to separate from the main balloon gondola to be recovered before the flight ends. The mechanical system has been proven to be consistent, even with extended exposures at high altitudes with high radiation and low temperatures. While there are many challenges to yet to overcome, lessons learned on the 2017 HASP flight will help drive designs for the future data acquisition system. Creating a self-sufficient system designed internally should help many of the interface problems encountered on the current design of TRIC. Eventually, a radio and GPS tracking system as well as a parachute system will be integrated for full release and recovery of the data module.

Appendix

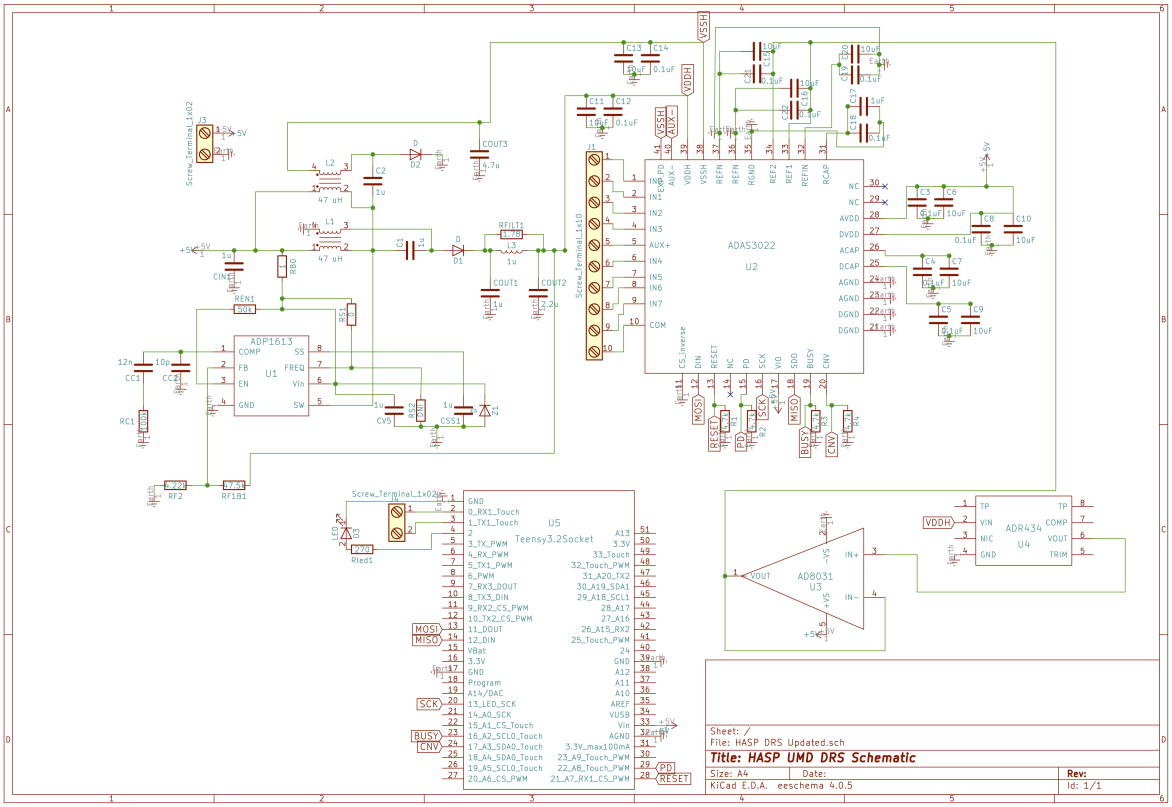


Figure 9: DRS Schematic

Acknowledgments

We would like to thank the Louisiana Spacegrant Consortium who helps run the HASP program as well as the Maryland Spacegrant Consortium who helped fund the HAAT-TRIC project. We would also like to thank our faculty advisor, Dr. Mary Bowden for her help in directing the team with her invaluable experience.

References

[1] Steven P. Schneider. "Flight Data for Boundary-Layer Transition at Hypersonic and Supersonic Speeds", Journal of Spacecraft and Rockets, vol. 36, no. 1, pp. 8-20, 1999.

[2] M. V. Morkovin, E. Reshotko, and T. Herbert, Transition in open flow systems: a reassessment, Bulletin of the American Physical Society, vol. 39, no. 9, pp. 1-31,1994.

[3] L. M. Mack, Transition and laminar instability, NASA-CP-153203, Jet Propulsion Laboratory, Pasadena, California, 1977.

[4] E. Reshotko, Boundary-layer stability and transition, Annual Review of Fluid Mechanics, vol. 8, pp. 311349, 1976.

[5] S. P. Schneider, Effects of high-speed tunnel noise on laminar-turbulent transition, Journal of Spacecraft and Rockets, vol. 38, pp. 323333, 2001.

[6] X. Zhong and X. Wang, Direct Numerical Simulation on the Receptivity, Instability, and Transition of Hypersonic Boundary Layers, Annual Review of Fluid Mechanics, vol. 44, pp. 527-561, 2012.

[7] A. Theuerkauf, Stratospheric turbulence observations with the new balloon-borne instrument LITOS. PhD thesis, University of Rostock, 2012.

[8] http://laspace.lsu.edu/hasp/Flightinfo.php