Time and Position
Data String
Serial Latency Measurements

HASP Technical Report 2009-01

Michael F. Stewart

February 16, 2009

GPS time and position data is now available to student payload groups via serial communication. They can request to receive these data strings from the HASP flight system at some pre-defined period, specified prior to integration. The string contains a timestamp and GPS coordinate information. The GPS coordinate information could alleviate the need for groups to purchase and deploy their own GPS system which could reduce the costs of student payloads both in materials and development time. The timestamp generated by HASP is accurate to within 1 millisecond and might be used by the payload as an accurate clock. However, there is a latency introduced by the serial communication. This report investigates the latency and its variation in order to better understand the limitations in using the timestamps to discipline payload clocks.

I. Introduction

HASP now provides student payloads with the ability to periodically receive accurate time and position information. Many payload experiments need real-time position information in order to operate as designed. GPS devices can be costly and take time to integrate into the payload's acquisition system. By providing this service through the serial connection, the added cost and development time needed could be eliminated. Some student group payload's also need an accurate clock, locked to GPS time, for time stamping records or scheduling events. Maintaining an accurate clock is not necessarily an easy task. For short flights the clock on some student payload hardware may not drift to a great degree at some constant temperature. However, for balloon flights the temperature range experienced by the student payloads may be extreme causing the internal clock to lose accuracy. For example, the popular BASIC Stamp is often used for student balloon data acquisition systems and the time base on this chip is accurate to ~1% *[1]*. A clock that is accurate to within 1 percent could be off by 14 minutes in a 24 hour period and thus it may be necessary to discipline in order to keep it within 1 millisecond of GPS time. While the timestamps generated by HASP are accurate to 1 millisecond, the latency in the serial communication between HASP and the student payload needs to be taken into account.

It is expected that the communication latency will match the serial latency which can be calculated given the serial baud rate at which the two processes communicate. If communication latency does match the expected serial latency and is consistently repeatable, accurate clock measurements over serial communication is possible. For a 1200 baud connection, 1 byte can be transferred every 8.3 milliseconds. For a 4800 baud connection the rate is 1 byte every 2.1 milliseconds and at 9600 baud the rate is closer to a byte per millisecond. For this test, the size of the GPS time and position string was 95 bytes; so if the communication latency were dominated by the serial latency then at 1200 baud the string should be received in about 790 milliseconds, at 4800 baud it should take about 200 milliseconds and at 9600 baud, about 100 milliseconds.

II. The Communication Latency Test

To test the serial latency, a process used to simulate the student payload was connected serially to the HASP process which communicates with student payloads; the SPSA (Student Payload Serial Agent). For this test, the student payload simulator was run on the same processor as the SPSA and therefore used the same clock. The clock is disciplined by a GPS 1 PPS and is accurate to well within 1 millisecond. The time and position string serial transmission period was set to 2 seconds and the serial communication between the two processes was run at 1200 baud, then 4800 baud and finally 9600 baud collecting at least a couple thousand strings at each frequency. Once the data string was received by the payload simulator process, a timestamp was appended to the data string and the string was written to file. This ensured that each data string contained a timestamp marking the time at which the string was written to the serial port and a timestamp marking the time once the string was read in its entirety from the serial port. The file was imported into Excel where the average communication latency was calculated as well as the deviation across the latency samples.

To test how the communication latency behaved while the SPSA was stressed, the student payload simulator, in addition to reading and writing the GPS time and position strings to file, also serially transmitted bytes to the SPSA as fast as the 9600 baud rate would allow; approximately 7700 bits per second. This action simulated a student payload transmitting data across its serial link as fast as possible.

III. The Communication Latency Results

At 1200 baud, 3,129 data strings were collected and the average communication latency was found to be 802 milliseconds with a standard deviation of 3 milliseconds. This was consistent with the expected previously calculated latency due solely to serial transmission of 790 milliseconds. At 4800 baud, 2,527 data strings were collected and the average communication latency was 201 milliseconds with a standard deviation of 1 millisecond. This also was consistent with the expected previously calculated serial latency of 200 milliseconds. At 9600 baud, 1,889 data strings were collected and the average communication latency of this data was found to be 101 milliseconds with a standard deviation of 1 millisecond which again was very close to the calculated serial latency of 100 milliseconds. The stress test was conducted at 9600 baud and 225 strings were collected. The average communication latency was 101 milliseconds with a standard deviation of 1 millisecond. These results were no different than the results when the testing wasn't stressed.

IV. Conclusion

Communication latency was tested at three different baud rates and at each rate the latency was consistent with the expected serial latencies at that rate. These results show that it is possible to use the timestamp within the GPS timestamp and position string as an accurate clock, even while the SPSA is stressed. If a clock accuracy of about 1 second to within GPS time is desired; this could most likely be achieved without many special student payload process design considerations or extensive testing. However, if sub-second clock accuracy is desired, contact Michael Stewart (stewart@phunds.phys.lsu.edu) to discuss in more detail methods to develop deterministic, time accurate student payload process algorithms. This would require special considerations to be taken when designing and developing the student payload processes.

V. References

[1] Parallax, Inc. "20: An Accurate Timebase." BASIC Stamp 1 Application Notes, BASIC Stamp Programming Manual version 1.9 (pages 167-170), http://www.parallax.com/Portals/0/Downloads/appnt/stamps/bs1Appnotes.pdf