Progress report

# Introduction

The Global Positioning System (GPS) is a network of satellites that allows calculation of positional data using triangulation. Distances are calculated by measuring the travel-times of light-speed messages. Naturally, a good measurement of position relies upon the performance of the clocks.

With multiple on-board atomic clocks [1], GPS satellites keep time with great accuracy and precision which is transmitted to the receiver. Calculating the position requires calculating the signal’s travel-time very accurately [2]. Unfortunately, the standard output of GPS receivers – a message in the *NMEA* format, known as a sentence – lists the time only to the second.

Sub-second timing from GPS could be useful in many cases, such performing simultaneous measurements in multiple locations. The aim of this project is to produce a system which calculates sub-second time from the GPS output, presenting the time conveniently to the user.

# Method

Whilst there are a variety of mechanisms for presenting the GPS information, we concern ourselves with receivers that send out data in *serial* format. In this format, the message, encoded by voltage in binary format, is sent one *bit* at once. This can be easily detected and decoded by a microcontroller.

The *Arduino Uno* microcontroller was chosen alongside the *Adafruit Ultimate GPS* receiver, specifically designed for the Arduino. This microcontroller is capable of reading and writing serial, measuring voltages, and writing to SD card. It also houses an internal oscillator clock for timing. Using this device, GPS serial messages can be decoded, logging the time of arrival (ToA) onto a memory card.

Ultimately the device should use the timings and contents of the GPS serial data to deduce the time with sub-second precision and accuracy. As such, there are two different clocks that must be characterised: the Arduino’s oscillator which performs the measurements of serial times and the GPS serial output, which occurs once in each second and can be thought of as “ticks” of a clock.

The Arduino’s oscillator clock can tell the time with millisecond precision, but lacks the accuracy. Furthermore, it is temperature- and voltage-sensitive, so its accuracy could drift due to changes in conditions. In contrast, the GPS output’s time is extremely stable on the long-term, and accurate as its timings originate from atomic clock. However, its precision is limited to the nearest second, and the consistency of the serial output is unknown; how soon is the serial message dispatched after the second turns, and is this time difference constant?

The stability and accuracy of the Arduino can be measured by comparing its time-keeping to the GPS receiver’s over long periods, while the receiver’s short-term stability can be measured by checking its tick lengths against the Arduino’s time. In order to determine how accurately the ticks fall on the second itself an external reference is needed.

Some GPS receivers, including the Adafruit receiver, contain a pulse-per-second (PPS) output. PPS generators produce a high voltage on the turn of each second and can be measured by the Arduino. This acts as the reference clock in the experiments.

|  |
| --- |
|  |
| Figure  The Arduino oscillator clock measures the ToA of each PPS and serial signal in milliseconds |

Once the predictability of the Serial data and Arduino oscillator have been assessed, the

# Bibliography

|  |  |
| --- | --- |
| [1] | “GPS Timing,” GPS.gov, [Online]. Available: http://www.gps.gov/applications/timing/. [Accessed 16 Dec 2015]. |
| [2] | A. Leick, “Scientific American,” Scientific American, 28 Nov 2005. [Online]. Available: http://www.scientificamerican.com/article/how-do-gps-devices-work/. [Accessed 01 Jan 2016]. |