Time Synchronization Notes

The following was posted by Tom Nolan to Slack in November 2015.

The ExoMars rover doesn't use the SCLK terminology. In some other missions there is an SCLK, which is a counter onboard the spacecraft that counts up at some frequency. For those missions, you usually use a toolkit to convert SCLK to some usable time format.

The rover uses Rover Elapsed Time (RET), which is a 6-byte quantity: 4 bytes of seconds, and 2 bytes of subsecond ticks (65536 ticks = 1 second). They have not specified what time corresponds to 0 RET, but at some point they presumably will.

The FSW uses 3 different tick counts in various places: 100-microsecond ticks, 10-millisecond ticks, and 1-second ticks. The 100-us ticks start from power-on, because they're kept in a register in the FPGA. The other two start from FSW boot time, because they're kept in the flight software.

The telemetry packets contain 100-us ticks (what you're calling "FSW ticks"). The correspondence between ticks and RET is recorded in the digital status packet: whenever I get the time command from the rover, I record the RET contained in the command and the 100-us ticks when I got the command (a "time at the tone" concept). From this information, you can reconstruct the RET for each packet, but then you still need to do the RET-to-time conversion.

We're currently using a convention embedded in the EGSE for the RET-to-time correspondence (I think it's seconds since 1/1/2000, but I don't know for sure). But we need to anticipate that there will be other RET-to-time conventions when we integrate with the rover. I suspect that there is no requirement at the rover level to make RET unique. So during rover I&T, they will probably start RET=0 at power-on, and we'll potentially get data sets with overlapping times every time the rover is power-cycled.

There is also the possibility that RET will start over again at 0 on Mars, which would add to our headaches.

Revision #1
Created 22 March 2023 17:30:26 by Nick Dobson
Updated 24 March 2023 13:46:57 by Nick Dobson