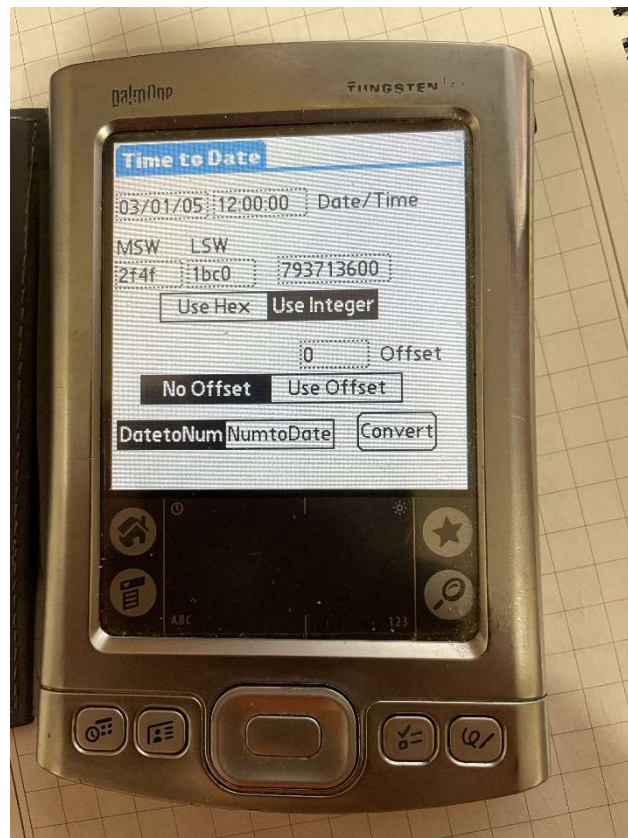


Time to Date

When I started doing software verification on ISS (International Space Station) software, I was initially responsible for verifying the command processing functionality in the Node 1 computers. Commands to the ISS were sent in sixty-four 16 bit words. Words 4 and 5 of a command contained time information that was expressed in the number of seconds that had elapsed since 00:00:00 6 January 1980. This is also the beginning of the GPS epoch.

I wrote a console application in C to convert hours, minutes, seconds, month, day and year to GPS time expressed as two 16 bit hex words. This was quite handy during our testing. I soon wrote a GUI based version for my Palm PDA after acquiring Pocket-C for Palm. The following is a photo of the PDA screen:

Here is a photo of the PDA screen:



I continued to use my Palm PDA for ISS software verification tests. The TimetoDate program was very useful when we were doing reboost tests. This is when the Russians fire their thrusters along the orbital path to increase the orbital radius. The US GNC (Guidance Navigation Computer) propagates the state vector in the event it is unable to get updates from its navigation sources. For the propagation to work properly, it is necessary to send a command with a maneuver plan to the GNC. The maneuver plan command contains the start and stop time of the burn along with the expected acceleration resulting from the burn. When we performed joint software tests with the Russians, the Russians

would give us the start time of the burn and the burn duration. The offset field of my program is used for entering the burn duration. When a NumtoDate conversion is performed, the offset is added to existing time (presumably the burn start time was previously computed) and the conversion will give you the burn stop time. The maneuver plan command was then updated with the start and stop times and sent to the ISS command and control computer (C&C) which would route it to the GNC. Our simulation for the C&C (CES – C&C Environment Simulation) included a simulation of S-Band data. This allowed us to simulate commands from MCC-H (Mission Control Center – Houston).

We would perform joint software verification tests whenever the US GNC software changed or the Russian software changed. Both the US and Russians would typically do a software update once a year but not necessarily at the same time. This usually meant a big software test at least once a year. The following is a picture taken of a test readiness review held on July 15, 2013. I am the guy in the blue shirt at the head of the conference table. I was presenting power point charts describing our software interface tests to our NASA customer, our international partners, quality assurance, requirement owners and several other interested (and maybe not so interested) people.

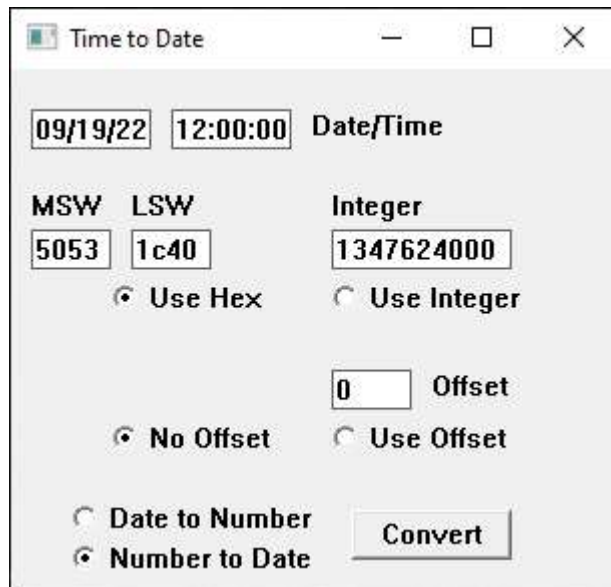


After writing the time2date program for my Palm PDA, I did not do any C GUI programming for nearly twenty years. After retiring from Boeing in August 2020, I started dabbling in various programming tasks. One of these tasks was to learn to do Win32 GUI programming.

I finally had a breakthrough when I went through chapters 10-13 of Jan Bodnar's book, *Introduction to Windows API Programming* including building and executing each of the examples using the Pelles C IDE recommended by the author.

You can find the Pelles C IDE at <http://www.smorgasbordet.com/pellesc/>. The copy of Jan Bodnar's book I have is dated January 7, 2021. I have no idea where I downloaded it from. Anyway, after completing those chapters, I had confidence that I could piece together several of the examples from Jan Bodnar's book and implement a PC based version of my Palm PDA using code that I wrote long ago for my PDA.

Here is a screen shot:



The program works. I tested it using data from some of the as-run 670 (reboost) procedures I had on my USB drive. I also tested this program with Visual Studio 2015. I copied the source file into Visual Studio after asking for a new project. I then used the Visual Studio wizard to create a template and replaced everything in the wizard generated code with my code (except for the first couple of includes). It works there as well. The source code that is on GitHub contains these two include files but they are commented out.

I'm not sure who will have a use for this tool except for maybe a few of my former coworkers. I had already provided them with a copy of the source code for them to look at, modify or use as they desire. If someone else finds this useful, great. Otherwise, thank you for reading.