Use of HalCoGen with Golf and PacSat Flight Software

What is HalCoGen

Both systems on a chip used in the Fox and Golf IHUs supply some means of dealing with the complex register setting and peripheral access that is required to make peripherals and error routines work. The STM32L supplies a library which can be called relatively easily. While this takes up a lot of source (and probably code) space it is especially useful when you want to ask questions of the "community". Nearly everything is asked and answered in terms of these routines. That said, you could read the code (or the docs) and it would not be TERRIBLY hard to code your own as Altus Metrum has done in there software (and we have borrowed some of that).

The TI TMS570 instead uses a GUI called HALCoGen (Hardware Abstraction Layer Code Generator). Using this GUI, you set up everything you need and it generates basic code to access peripherals and to handle interrupts. This code also includes places where you can add your own code (and HALCoGen remembers the added code so when you change something in the GUI, your added code will still be there.

How do we use it for Golf and PacSat?

Again, it would doubtless be possible to hand-code most of this, but the TMS570 appears to have a lot more interacting pieces that the STM32. I have just used HALCoGen V4.07.01, the latest as of January 2023. There are a couple of routines that needed customizations in places where HALCoGen did not allow it (in particular in order to send SPI messages without allocating a buffer to receive simultaneously). In that case, we have "spi-replacement.c" which is a file copied from HAL's spi.c and modified as I required.

HALCoGen runs only on Windows. I have a QEMU/KVM Virtual Machine running Windows 10 on my Linux build machine and that is where I run HALCoGen. There are only two files that describe the TMS570 setup shown in the GUI: Those are a .dil file and a .hcg file. In addtion the code it generates goes into "include" and "source" directories. I have placed these in a DropBox folder so that I can run HALCoGen when I need to and then just copy all the files into Code Composer Studio. In other words, all the code required to set up HALCoGen is actually in the git repositories for Golf and PACSat.

And note again that yes, it could be made more efficient by hand coding. And I also think that HALCoGen does not really deal well with DMA. But this has been a good way to learn and to avoid subtle errors. (That said, there are a few errors in HALCoGen itself! The bug I have found is that you can't change the SPI speed using the GUI. It always come out to the same value. This was a known problem 10 years ago, and appears not to have been fixed, at least not for all the SPI devices).