ADC code documentation:

\*\*I didn’t explain what the setup code does because it seems pretty standard across all applications for this Launchpad and I honestly don’t fully understand it myself.

ConfigureADC():

* Most of the code in here is pretty standard and was taken from other examples. What this function is power up and do the first part of configuring the ADC. The important part to take away from this is an understanding of what ADCSetMode() does. It lets us pick which ADC we are going to use (there are several on our Launchpad) set the resolution we are going to use and the input mode. I chose 12-bit resolution because we are going to use a single input signal for our input, and this can only be done in 12-bit resolution.
* The adc has two options for reading in data. It can take differential, which takes the difference between two signals and uses that difference as the “data” that’s read in. This is done by using two input pins. We are using single input mode, which means that we need only one pin for each hall probe sensor.

SetUpADCSoftware()

* In this function, we set the acquisition period for our ADC i.e. how long it has to read in data as well as select which channel (pin) we want to use on the ADC.
* SOC stands for “start of conversion.” An SOC is the method TI uses to trigger an ADC read. There are multiple different SOCs and therefore multiple different ways to trigger a read. I have arbitrarily chosen SOC1 since we are using channel 1. Note that an SOC is not something that was created in software. This is something that exists within the hardware, so we have to use it if we want to trigger our ADC.
* There are many ways we can set up the SOC to trigger the ADC. In this scenario, I opted for a force trigger using software.
* Refer to the comments in the code for what each register does. There is also more information in the technical reference manual in chapter 9.The manual can be found in ControlSUITE under Devices->Delfino F2837xS->User Guides. It should be the first document listed.

Sample()

* This function actually reads in the data sampled by the ADC. I’ve commented the code pretty heavily so it should be self-explanatory, however, this line will need more explanation:
* AdcaRegs.ADCSOCFRC1.all = 0x0002;: This line lets us pick which SOC we are using. Specifically, the “FRC” at the end of the name stands for “force” so we have to use this register to forcibly start the ADC reading through code. Each bit in this 16 bit register corresponds to 16 different SOC’s and in order to access them, you need to set the corresponding bit to 1. The indexing for the SOC’s goes from 0-15, so SOC1 is technically the second SOC represented by this register. In order to activate this SOC, I needed to set the second bit in the register to 1, however, since this would make the bit value of the register 0000000000000010, which is two in binary, I set the register equal to 2 in the code. If I wanted to set both SOC0 and SOC1 to high, I would set the register equal to 3 (0000000000000011).