Example usage of ADC

One way of accessing of the booster pack is using the analog to digital converter (ADC) unit of the TM4C129 chip (you are encouraged to use the functions defined in "driverlib/adc.h"). Here we will show you an example of how to read the values of a sensor using ADC.

1. Enable an ACD module, for example ADC0 module with `SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);`
2. Next we have to wait for the module to be ready with `SysCtlPeripheralReady()`.
3. Next we need to enable the appropriate GPIO as ADC for the specific sensor we are looking for. To find out which GPIO port and pins to enable, we can refer to the user guides for the main board and booster board for the information. Since we connect the Booster board to the main board on BoosterPack 1, we can see on which GPIO port and pins sensors are connected.
4. The next step is to configure the ADC sequencer with `ADCSequenceConfigure()`. For example, we can configure ADC0 and sample sequencer 3 with highest priority 0 and make the processor the trigger
5. The next step is to configure the step in the sequencer with `ADCSequenceStepConfigure().` It is important to specify `ADC_CTL_END` on the last step of the sequence to indicate it is the last one and `ADC_CTL_IE` to trigger and interrupt if an errors occurs. To see which channel, you can see it on the user guide for the main board.
6. Then you need to able the sequencer with `ADCSequenceEnable()`
7. Next, you need to trigger the sample sequence with `ADCProcessorTrigger()`
8. Then you have to wait until the sequence sampling has been completed with `ADCIntStatus()`
9. At the end, you can read the value from the ADC with `ADCSequenceDataGet();`

For more information we refer to the datasheet TM4C129 on Section 18 and TivaWare Peripheral Driver Library on Section 4 for the detailed information regarding the functions used.