# Remaining Firmware Items to Implement for the MOLLER ADC

1. Priority 1: The subsystem\_capture.sv module implement the code to calculate the sum, sum of squares, and the number of samples for a specified period. This is the module that would be called for each individual block within a helicity window.

This module needs to be wrapped in another module that implements the following:

* Receives a trigger signal from the TI that specifies the start of the helicity window sampling period. Helicity windows are usually about $480 μs$ long which would translate into a fixed number of 7000 samples per window, at a sampling time of $68 ns$ . Number of samples needs to be controllable by slow controls, as we won’t know the actual integration window until the experiment. Optionally, have the integration start after a specified number of samples to skip following the trigger.
* We would also like to split these 7000 samples into a fixed number of blocks and call subsystem\_capture.sv for each of these blocks. We would like to be able to specify the number of blocks in slow control (i.e. just like we are specifying the sample prescale currently). For example, if we specify $k=4$ blocks then we would accumulate the sum, sum of squares, and number of samples 5 times, once for the full $7000$ helicity samples, and for each of the $4$ blocks over $1750$ samples. This allows us to perform a sort of check-sum of the data and check for bit errors and transmission errors. Could be specified by # of blocks and # of samples per block. Is there a maximum number of blocks, and can it be at least 10?
* A single trigger signal from the TI would be used to start accumulating the samples into the series of blocks
* We also want to collect the time stamp for the full helicity window and each of the blocks. Timestamps should be reset by the synchronization signal sent at the beginning of the CODA run.
* All of the summed data item should be stored in $64$-bit words to prevent overflow.
* The data items for each block and the total for the full helicity window should be packaged together. I think this means that the packetizer currently implemented in subsystem\_capture.sv would have to move to the wrapper module.

1. Lower priority: The subsystem\_stream.sv module implements the readout of all samples for two selected board channels. This mode is currently used to test the boards. We would like to add/change the following:
* Extend this to readout all 16 channels at a reduced sampling rate and/or only for short periods time with a variable start time within the helicity window.
* Add a time stamped streaming start signal based on the TI trigger and/or the TTL gate input at the back of the board. When using the TI as the start, the “gate” should start with the TI start and last the specified number of samples. Alternately we could just have a bit that was true when the TI trigger is received.
* We would like to have two bits in the streaming data. One that indicates (is high) when the sample falls within a given helicity window and one that indicates whether the helicity window counter is even or odd (requires a trigger counter). The even-odd could be counted from the sync reset at the CODA start run.