Iteration 3: Homework 4

We're going to do this homework a little differently than in the past. Because this is all review and this is a good chance to test what you've learned, we're going to dump some requirements on this page. On the next page, we'll give some hints. On the third page, we'll get into more detail about our solution.

Your job is to work at it and not peek until you need to. Get stuck somewhere in the requirements? Check the second page for hints. Still not enough? No problem… you can dive into more details. As always, we also have a snapshot of our Iteration 3 final implementation. You can check it out when you are done or if you really need to peek.

- Implement CommandHandlerResults
- Digital and Analog Results should be returned.
    - Just return the latest value whenever asked.
    - Don't forget to follow the protocol as specified in the Command Specification for the Data Logger (see Results Response page)

Hints:

- Implement CommandHandlerResults
    - Assembling results is going to get bigger than most handlers… maybe push the hard work into its own module and function like Capture_GetResults
    - You'll want to avoid yet another set of buffers… so maybe pass the buffers in your message to be filled directly?
- Digital and Analog Results should be returned.
    - Just return the latest value whenever asked.
    - Don't forget to follow the protocol as specified in the Command Specification for the Data Logger (see Results Response page)
        - Digital comes first. It's a single byte between 0x00 and 0x3F
        - Analog comes next. It starts with a header 0xC4, followed by a length (in this case 0x0C since we are returning 6 channels for now), and finally 16-bit unsigned ints in big-endian format for each channel (another 12 bytes).
        - The final message will be too long for the existing Packer len field… therefore we will need to expand our message protocol to support 0-9 and A-Z (giving us a maximum of 35 bytes) (see Framing section)

Test Ideas:

- Implement CommandHandlerResults
    - Did you test the happy path, where a valid message is returned?
    - Did you test the situation where STATUS NONE YET was returned because no data has yet been gathered?
    - Did you handle the situation where the user requested the results command and included extra data (which isn't currently supported?)
    - Don't forget about _ExpectAndArgs and _ReturnThruPtr.
    - There is a variation on _ReturnThruPtr called _ReturnArrayThruPtr which accepts a pointer and a LENGTH to the number of elements being passed.
- Capture_GetResults
    - Because there are no options yet, and because the data is always available at the moment, there is really only a single test case that is needed to test this function for now.
    - Don't forget that you can call the same mock numerous times to queue up multiple expectations and return values.
- Packer
    - Did you keep that 9-byte message around to test? It's a good one to keep because it's on the border of a logic change
    - Did you add another test for a 35 byte message?
    - Did you verify that the output lengths are upper case?
    - Did you verify that the number of nibbles are still correct?
- Parser
    - No change should be necessary. We aren't RECEIVING anything larger than we ever have.