Software Meeting - 02/23/2024 - Legacy Code Review/Implementation

General Goals:

- Review 2023 code again, make decisions on what to keep and what to change
- If applicable, clone existing code to 2024 repo
- If time allows, look over GCS code

<u>Code Review</u>

ContainerFSW

1. Hardware.cpp:
    a. Under namespace hardware, they set up the initializations needed for some of the code to function. First camera call is set to true, however, not essentially sure what it does for the time-being.
    b. Init:
        i. Setup the pin modes for output
        ii. Write the camera pin to high, so that it can be written
        iii. cameraHold, unsure what it does as of now
        iv. cameraRecording, unsure what it does as of now
        v. firstCameraCall, unsure what it does as of now
        vi. payLoad, will come back to this later, from lines 28-29
        vii. Wire, need to write the I2C protocol commands to Arduino
        viii. Begin 0x77 for the BMP chip, initializes the chip with a HEX argument
        ix. Set bmp temperature to temperature with a default parameter
        x. Set bmp pressure with a default parameter
        xi. Set bmp IIR filter coefficient with a default parameter
        xii. Set bmp output data rate to 50 Hz with a default parameter
    c. Deploy_payload:
        i. Ignore for now, simply writing to the servo motors with a parameter of 30 (presumably 30 seconds)
    d. update_camera:
        i. Boolean is set for true or false as input parameter, assuming record is true in this case for function to trigger
            1. If the record boolean is true, but the camera is not recording,
                a. Then verify if this the first time camera is recording, if it is then go ahead and put a waiting gap of 0 for the camera, and then set firstCameraCall back to false
                b. Then start recording
            2. Else if the boolean record variable is set to false, and camera is recording

            a. Check the firstCameraCall boolean
                i. Set camera hold waiting time to 0, and apply firstCamerCall to false
            b. Stop recording
   e. Start_recording:
      i. Check if cameraHold is less than 150, which is the latency set for the camera before it starts recording
         1. Write to the camera pin low for stopping to record
         2. Else write high to the camera pin so it starts recording
            a. Set the cameraRecording boolean to true, and firstCameraCall to true as well
   f. Stop_recording:
      i. Check if cameraHold is less than 550, set camera pin to low, so that it can stop recording, since 550 is the max latency
         1. Set camera pin to low,
         2. Else, set camera pin to high
            a. Once camera pin is high, set it so that we can transmit to the camera
            b. Set the camera recording to be false
            c. Then set the camera call to true
   g. Read_sensors:
      i. Pass in the sensor data as the input parameter for the sensors to read
         1. Start to read the data from the bmp
         2. Set the bmp altitude to sea_level

2. ContainerFSW.ino
   a. Initialize hardware in the setup, with a serial process beginning at the given time slot
   b. Initialize the hardware base setup
      i. loop()
         1. Read the altitude data from the bmp
         2. Retrieve the required sensor data
         3. Set the hardware to read the sensor data
         4. Incrementing the altitude by 25 in each iteration, not sure why previous team did this process manually
         5. If sensor altitude data is less than 130, and it has reached the critical apogee altitude, set the alt and iterator both to 0
         6. Print the specific altitude using the serial thread

        7. Not sure what check1, check2 or check3 are doing, maybe this was some type of tester code, but team is unsure what condition is being set to true, and what conditions is setting to false (NEED FURTHER CLARIFICATION as to what is going on here)
        8. State cameraCall boolean to update the camera to true, so that it can turned on/off accordingly
        9. Set lastAlt to get altitude
        10. Add a specific delay for the telemetry

3. PayloadFSW/Hardware.cpp
    a. The updated hardware file contains code for the camera, GPS, BNP and BNO chips. Also contains information tied to the radio XBee chips, however for now the XBee chip will be a later priority. Moving forward, we will transfer the boiler plate code to the new repository, along with this notes file.