Table of Contents

INTRODUCTION 2

IMU Module 2

Experimentation 5

- System Functionality 6
- Compression Subsystem 7
- Encryption Subsystem 7

Results 8

- System Functionality 8
- Compression Subsystem 8
- Encryption Subsystem 9

ATPs 11

Admin Documentation 11

Distribution of Work 11

Project Management Tools 12

Development timeline 12

References **13**

INTRODUCTION

This report will focus on sending data from an IMU sensor to the compression and encryption subsystems. An ICM 20948 sensor will be used to create data by performing different motions on the sensor. This data will be used to measure the performance of the system and can then be extrapolated to fit that of the actual ICM 20649 sensor which will be used on the SHARC BUOY. The data will be passed to the system, that is, the compression and encryption subsystem and the performance of these subsystems will be measured by varying the size and throughput of the data.

IMU Module

An ICM 20948 will be used to collect data which will be used to test the system. The IMU is a low power 9-axis device with a 3- axis gyroscope, 3-axis accelerometer, 3-axis compass, a humidity sensor, temperature sensor, among other components. An ICM 20649 will be used on the SHARC BUOY. It is a 6-axis device with a 3-axis gyroscope, 3-axis accelerometer, humidity sensor, temperature sensor, among other components. The two devices are similar: they both have a VDD operating range of 1.71V to 3.6V and both utilize I2C and SPI communication protocols. They both have on-Chip 16-bit ADCs and Programmable Filters. with a few differences. Below is a table showing some of the differences between ICM 20649 and ICM 20948.

ICM 20649	ICM 20948
Is a 6-axis device	Is a 9-axis device
offers precise analysis of contact sports applications by providing continuous motion sensor data before, during, and after impact	integrates multi powerful sensors such as gyroscope, accelerometer, magnetometer, barometer, temperature and humidity sensor, etc.
No Axis Compass	3-Axis Compass with a wide range to ±4900
Has a 4kB FIFO buffer enables the applications processor to read the data in bursts	Has android support

Host interface: 7 MHz SPI or 400 kHz I2C	Auxiliary I2C interface for external sensors. 7 MHz SPI or 400 kHz Fast Mode I²C.
Extended FSR for gyroscope: ±4000 dps	±2000 dps FSR for gyroscope
Extended FSR for accelerometer: ±30g	±16g FSR for accelerometer

[1][3]

Extrapolation:
The final project would be used in the poles in sharc buoys to capture various forms of data such as wave dynamics. The 2 conditions in these environments are pretty rough. The sensors would be in constant movement due to the wave, ice and wind movement. The first aspect is the harsh weather. Tests were done with the IMU in a freezer. Although the temperatures may not be the same, this would give an almost ideal representation of the weather in the poles. The second

aspect was the vigorous movements that are common in the poles. Tests were done while shaking the IMU to mimic the movements. The IMU responded well by giving accurate results on the accelerometer and gyroscope that reflected that movement. The third aspect was the magnetic forces present in the poles. For this, we drove an electromagnet with current to create a magnetic field. This would give an idea of the magnetic forces that the IMU would have to deal with. The data produced by the IMU accurately represented this.

Since both IMU utilizes SPI protocols at the same rate, the program and sampling rate we use on the ICM20948 can be used when dealing with the ICM20649 with possibly minor adjustments needed.

IMU Validation

Given an ICM20948, in order to verify our IMU is in working condition, we derived test conditions to examine and validate the accelerometer and gyroscope. Since the 3-Axis compass is not available on the SHARC BUOY, we decided not to test its function. Before beginning the tests, we first displayed the IMU readings to the console at a continuous rate to examine the data from the IMU and its changes. We decided to choose a sampling rate of 50ms as it was sufficient to examine the changes in the accelerometer and gyroscope readings. We also decided to use the maximum scaling factor for both the accelerometer and gyroscope as this would increase the sensitivity of these sensors to make their varying readings easily examinable. The IMU rested on a breadboard to ensure its stability before the tests were conducted.

Testing the accelerometer consisted of applying oscillating forces to the IMU, namely side-side push and pull forces for the x axis, forward and backward push and pull forces for the y axis, raising and lowering forces for the z axis. By applying these forces, an acceleration would occur within these axes which would reflect in our console readings. Each axis was tested for approximately 10 seconds which can be seen in Figure 1 which illustrates the readings collected from the accelerometer as a result of the oscillating force testing. When testing the x-axis, we used a gradual motion resulting in its meander shape. When testing the y-axis, we wanted to see what would happen if we applied impact forces, such as flicking the breadboard with a finger, resulting in a shape with a number of large spikes. In testing the z-axis, we pulsated our raising and lowering of the breadboard at a fast rate, which can be seen in the resulting shape. Though the z-axis responds to changes in axis motion, it can be seen to have an offset of around 2000 which can be corrected in software.

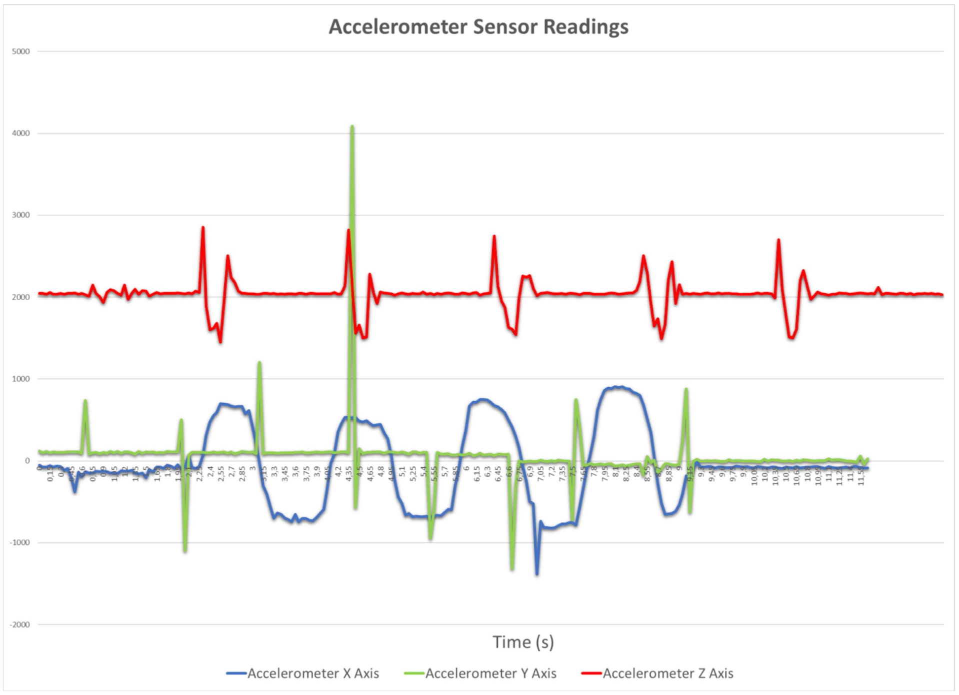


Figure1: Accelerometer Sensor Readings

We followed the same approach when testing the gyroscope, where each axis would be tested for approximately 10 seconds. Testing the gyroscope consisted of applying oscillating rotations within the x, y and z axes. Gyroscope sensors measure angular velocity and by applying these oscillating rotations, varying positive and negative readings should occur within our console readings. Figure 4 illustrates the readings collected from the gyroscope sensor after applying these oscillating rotations, demonstrating that the angular velocity is varying and that the gyroscope sensor is working.

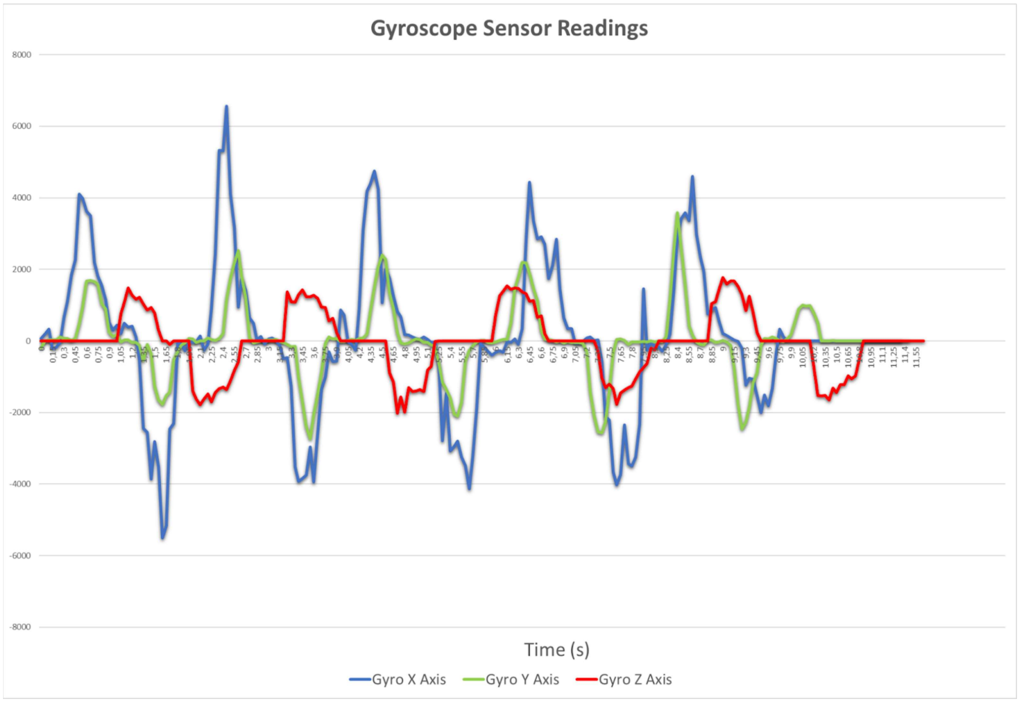


figure2:Gyroscope Sensor Readings

Even though we could not test and confirm that the accelerometer and gyroscope readings are in fact accurate, we can confirm that the accelerometer and gyroscope sensors are indeed functional, and that the IMU is functional.

Experimentation

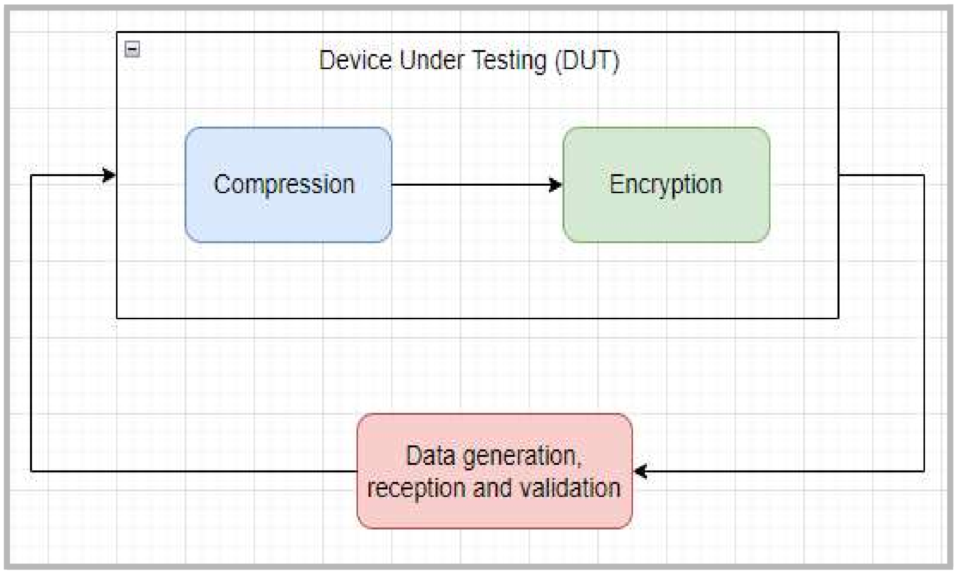
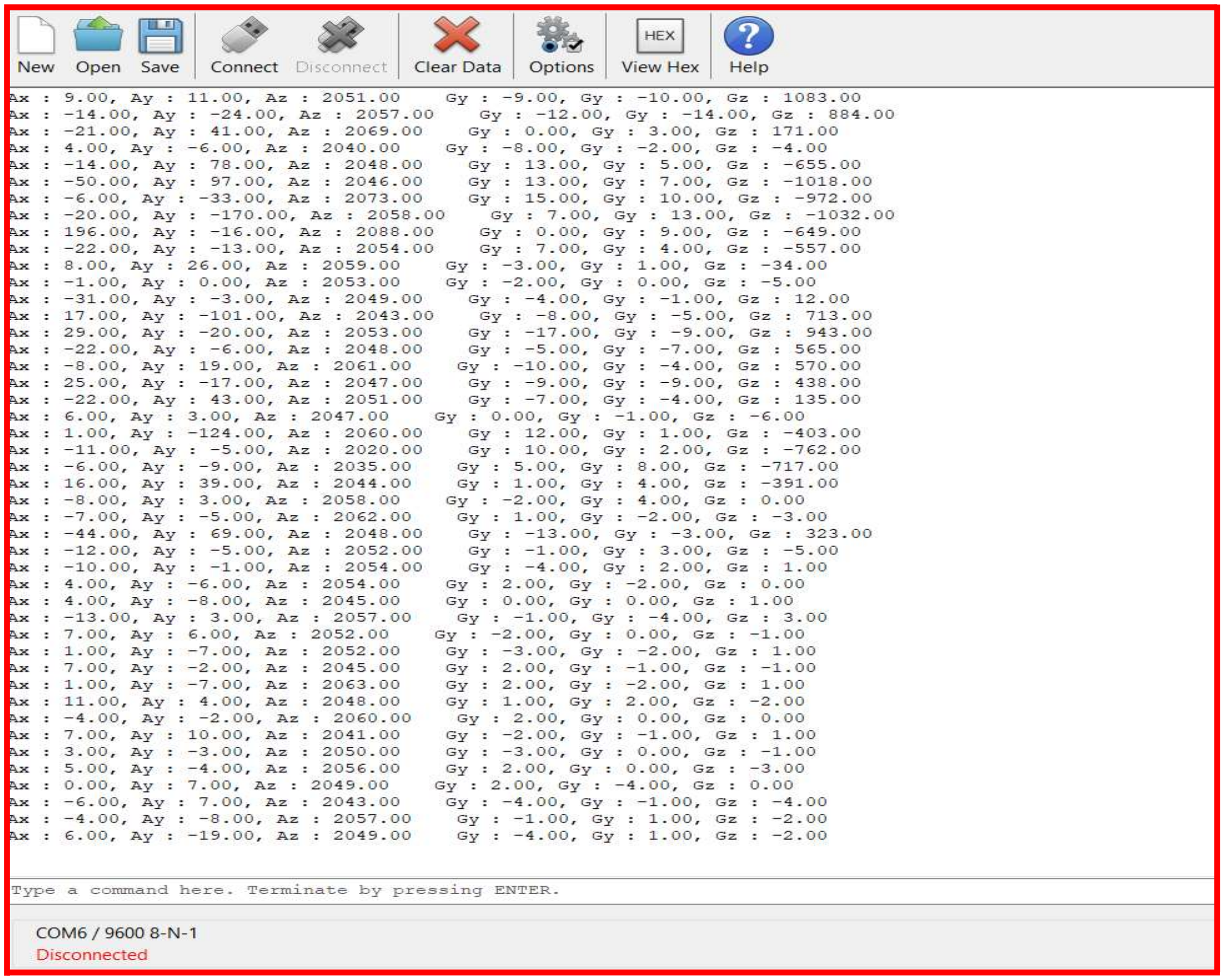

Device Under Testing (DUT)
Compression
Encryption
Data generation, reception and validation

Figure3:A loop-in test was used in this experiment

Given an stm32f0 microcontroller and usb-to-uart module, we were able to form a serial connection between our PC and the microcontroller. The serial connection allows for our IMU data to be retrieved ,illustrated in Figure 4 below,as well as its compressed and encrypted forms. Below we will describe how we aim to run our experiments to test our overall system functionality, as well as discuss the experiments set for the compression and encryption submodules.



```
Ax : 9.00, Ay : 11.00, Az : 2051.00    Gy : -9.00, Gy : -10.00, Gz : 1083.00
Ax : -14.00, Ay : -24.00, Az : 2057.00    Gy : -12.00, Gy : -14.00, Gz : 884.00
Ax : -21.00, Ay : 41.00, Az : 2069.00    Gy : 0.00, Gy : 3.00, Gz : 171.00
Ax : 4.00, Ay : -6.00, Az : 2040.00    Gy : -8.00, Gy : -2.00, Gz : -4.00
Ax : -14.00, Ay : 78.00, Az : 2048.00    Gy : 13.00, Gy : 5.00, Gz : -655.00
Ax : -50.00, Ay : 97.00, Az : 2046.00    Gy : 13.00, Gy : 7.00, Gz : -1018.00
Ax : -6.00, Ay : -33.00, Az : 2073.00    Gy : 15.00, Gy : 10.00, Gz : -972.00
Ax : -20.00, Ay : -170.00, Az : 2058.00    Gy : 7.00, Gy : 13.00, Gz : -1032.00
Ax : 196.00, Ay : -16.00, Az : 2088.00    Gy : 0.00, Gy : 9.00, Gz : -649.00
Ax : -22.00, Ay : -13.00, Az : 2054.00    Gy : 7.00, Gy : 4.00, Gz : -557.00
Ax : 8.00, Ay : 26.00, Az : 2059.00    Gy : -3.00, Gy : 1.00, Gz : -34.00
Ax : -1.00, Ay : 0.00, Az : 2053.00    Gy : -2.00, Gy : 0.00, Gz : -5.00
Ax : -31.00, Ay : -3.00, Az : 2049.00    Gy : -4.00, Gy : -1.00, Gz : 12.00
Ax : 17.00, Ay : -101.00, Az : 2043.00    Gy : -8.00, Gy : -5.00, Gz : 713.00
Ax : 29.00, Ay : -20.00, Az : 2053.00    Gy : -17.00, Gy : -9.00, Gz : 943.00
Ax : -22.00, Ay : -6.00, Az : 2048.00    Gy : -5.00, Gy : -7.00, Gz : 565.00
Ax : -8.00, Ay : 19.00, Az : 2061.00    Gy : -10.00, Gy : -4.00, Gz : 570.00
Ax : 25.00, Ay : -17.00, Az : 2047.00    Gy : -9.00, Gy : -9.00, Gz : 438.00
Ax : -22.00, Ay : 43.00, Az : 2051.00    Gy : -7.00, Gy : -4.00, Gz : 135.00
Ax : 6.00, Ay : 3.00, Az : 2047.00    Gy : 0.00, Gy : -1.00, Gz : -6.00
Ax : 1.00, Ay : -124.00, Az : 2060.00    Gy : 12.00, Gy : 1.00, Gz : -403.00
Ax : -11.00, Ay : -5.00, Az : 2020.00    Gy : 10.00, Gy : 2.00, Gz : -762.00
Ax : -6.00, Ay : -9.00, Az : 2035.00    Gy : 5.00, Gy : 8.00, Gz : -717.00
Ax : 16.00, Ay : 39.00, Az : 2044.00    Gy : 1.00, Gy : 4.00, Gz : -391.00
Ax : -8.00, Ay : 3.00, Az : 2058.00    Gy : -2.00, Gy : 4.00, Gz : 0.00
Ax : -7.00, Ay : -5.00, Az : 2062.00    Gy : 1.00, Gy : -2.00, Gz : -3.00
Ax : -44.00, Ay : 69.00, Az : 2048.00    Gy : -13.00, Gy : -3.00, Gz : 323.00
Ax : -12.00, Ay : -5.00, Az : 2052.00    Gy : -1.00, Gy : 3.00, Gz : -5.00
Ax : -10.00, Ay : -1.00, Az : 2054.00    Gy : -4.00, Gy : 2.00, Gz : 1.00
Ax : 4.00, Ay : -6.00, Az : 2054.00    Gy : 2.00, Gy : -2.00, Gz : 0.00
Ax : 4.00, Ay : -8.00, Az : 2045.00    Gy : 0.00, Gy : 0.00, Gz : 1.00
Ax : -13.00, Ay : 3.00, Az : 2057.00    Gy : -1.00, Gy : -4.00, Gz : 3.00
Ax : 7.00, Ay : 6.00, Az : 2052.00    Gy : -2.00, Gy : 0.00, Gz : -1.00
Ax : 1.00, Ay : -7.00, Az : 2052.00    Gy : -3.00, Gy : -2.00, Gz : 1.00
Ax : 7.00, Ay : -2.00, Az : 2045.00    Gy : 2.00, Gy : -1.00, Gz : -1.00
Ax : 1.00, Ay : -7.00, Az : 2063.00    Gy : 2.00, Gy : -2.00, Gz : 1.00
Ax : 11.00, Ay : 4.00, Az : 2048.00    Gy : 1.00, Gy : 2.00, Gz : -2.00
Ax : -4.00, Ay : -2.00, Az : 2060.00    Gy : 2.00, Gy : 0.00, Gz : 0.00
Ax : 7.00, Ay : 10.00, Az : 2041.00    Gy : -2.00, Gy : -1.00, Gz : 1.00
Ax : 3.00, Ay : -3.00, Az : 2050.00    Gy : -3.00, Gy : 0.00, Gz : -1.00
Ax : 5.00, Ay : -4.00, Az : 2056.00    Gy : 2.00, Gy : 0.00, Gz : -3.00
Ax : 0.00, Ay : 7.00, Az : 2049.00    Gy : 2.00, Gy : -4.00, Gz : 0.00
Ax : -6.00, Ay : 7.00, Az : 2043.00    Gy : -4.00, Gy : -1.00, Gz : -4.00
Ax : -4.00, Ay : -8.00, Az : 2057.00    Gy : -1.00, Gy : 1.00, Gz : -2.00
Ax : 6.00, Ay : -19.00, Az : 2049.00    Gy : -4.00, Gy : 1.00, Gz : -2.00

Type a command here. Terminate by pressing ENTER.

COM6 / 9600 8-N-1
Disconnected
```

Figure4:IMU data collection

System Functionality

Using our IMU we will collect gyroscope and accelerometer data. Using HAL_Delay we can control the sampling frequency (The rate at which we collect data)We store the data in a buffer and after a certain period of time we will have an interrupt that takes the data in the buffer , compress it followed by encrypting it and sending it to the computer(Note we also send the original data ,compressed only data). That is to simulate the transmission process , To simulate the Receiving process we had to send the compressed plus encrypted data to the STM board where it undergoes decryption and decompression before sending to it back to the computer (Note we again we also send only decrypted data , and uncompressed data).

- We will first compare the file sizes of the raw data and compressed only data to prove that compression occurred.
- Secondly we will compare the (encrypted + compressed) data and the compressed only data to prove that encryption has occurred.
- Thirdly we will compare the compressed data before encryption and after decryption to check if the encryption maintains data integrity.
- Fourth we will check how much of the compressed data has been retrieved by comparing the (decrypted + uncompressed) data and the raw unprocessed data .
- Lastly, without transmitting any other data we will run the system to measure the time it takes to compress , encrypt and send data and time to decrypt ,uncompress and retransmit the data to the computer again.

Compression Subsystem

The main criteria used to test the compression subsystem was compression ratio. For compression, we only tested the compression submodule by only compressing the IMU data and sending it to the computer where it is sent back for decompression. In the end, we compare the file size of the raw data and compressed data to ensure that data compression has occurred, also by comparing the uncompressed data and the raw data we can check the data losses. We also tested the time it takes for the compression and transmission of data as well as the decompression of data of the system. By increasing and decreasing the sampling frequency we were also able to determine whether the compression ratio is affected by the volume of data being operated on.

Encryption Subsystem

In our previous report we tested and validated that a compressed text could be encrypted and decrypted. Within this encryption subsystem, we now test the IMU and encryption pair, determining if the IMU data can be collected, encrypted and decrypted. In our previous report we used CoolTerm to display our encrypted and decrypted data, which will be used in this case again for experimentation. For a successful encryption subsystem, we first need to collect the IMU data within the buffer and process the buffer, sending the string buffer into our encryption function and determining if the data inside the buffer has indeed been encrypted. The plan is therefore to collect a subset of the IMU data (displaying the subset as well, to the console using CoolTerm for future reference) and concatenation the subset into a buffer. Once completed we shall send the buffer to the operational encryption function, thereafter the program will print the encrypted buffer data to the console serial port. Since the buffer before encryption was displayed to the console earlier, we will be able to compare the displayed encrypted data without determining if there is no match between them. After encryption, the program will initiate a decryption process where it will take the encrypted buffer data and send it to the operational decryption function, which will in turn display the decrypted buffer data to the console. It is expected that the decrypted displayed data should match the IMU data collected before, which will be examined.

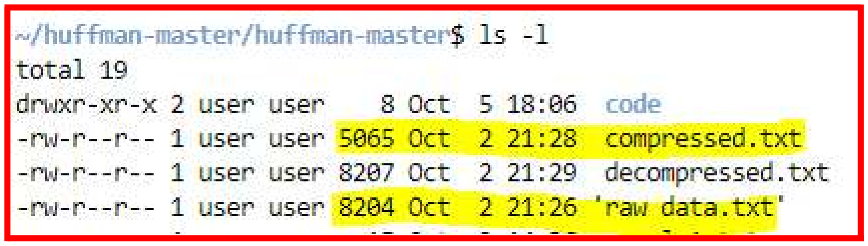
Results

System Functionality

The ICM20948 IMU functions as expected on the stm32 discovery, resulting in expected changes in the accelerometer and gyroscope sensor readings when forces and rotations are applied to and about its axes, represented in Figures 1,2 and 4. Though the IMU is functional, due to time constraints the overall system functionality combining both compression and encryption could not be tested. We discussed that it would be best to first tackle and perfect our IMU and subsystems pairs before merging the subsystems, as it would ensure little to no issues will be present when merging our subsystems. Even though we have yet to tackle our overall system functionality, our IMU subsystems pairs are in working conditions which will be further discussed below.

Compression Subsystem

- The compression of our compression submodule, It works as show by looking at the different file sizes of the raw data and compressed data in fig5.



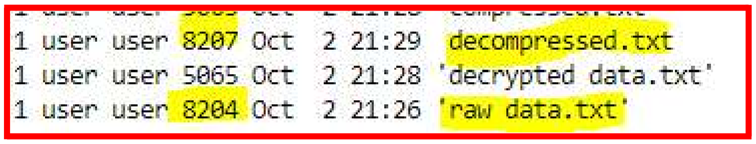
```
~/huffman-master/huffman-master$ ls -l
total 19
drwxr-xr-x 2 user user    8 Oct  5 18:06  code
-rw-r--r-- 1 user user 5065 Oct  2 21:28  compressed.txt
-rw-r--r-- 1 user user 8207 Oct  2 21:29  decompressed.txt
-rw-r--r-- 1 user user 8204 Oct  2 21:26 'raw data.txt'
```

figure5:raw vs compressed data

$$\text{Compression Ratio} = \frac{\text{Uncompressed Size}}{\text{Compressed Size}}$$

= 1.62

- The check of how much of the compressed data has been retrieved by comparing the (decrypted + uncompressed) data and the raw unprocessed data. ended up showing us some data losses, when we performed compression and decryption on our STM board the resultant data always had extra data added to it. The decompressed file is even slightly large that the compressed data.



```
1 user user 8207 Oct  2 21:29  decompressed.txt
1 user user 5065 Oct  2 21:28 'decrypted data.txt'
1 user user 8204 Oct  2 21:26 'raw data.txt'
```

figure7: data Integrity Test with compression

As shown in figure7 the compression algorithm did work, we were able to retrieve the majority of the data but the rest seems to be corrupted on the STM board this is further shown by

- The results of the test on the compression subsystem are shown in the table below to investigate the relationship between file size and compression ratio

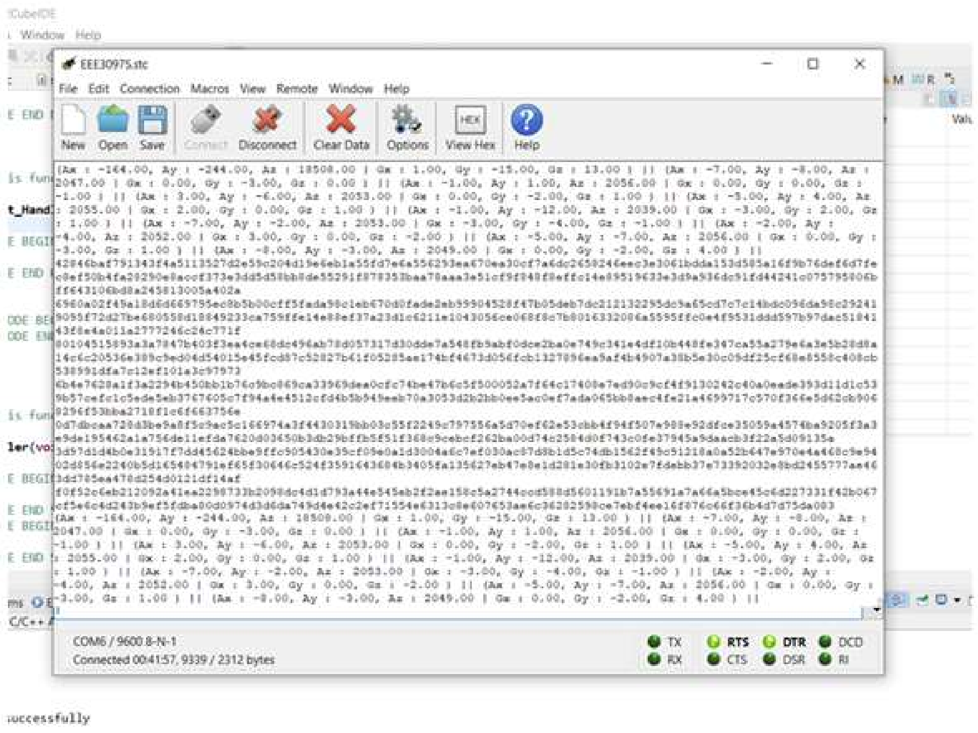
Data Set	Initial File Size (bytes)	Compressed File Size (bytes)	Decompressed File Size (bytes)	Compression Ratio (bytes)
50	1 052	1 012	1 057	1.0009514
500	10 743	9 878	10 748	1.0875683
1000	21 910	21 864	21 916	1.0020134
2000	42 982	42 782	42 986	1.0046749
5000	91 346	90 393	91 349	1.0105428

Table 1 showing results of compression subsystems

From the table above, it can be shown that increasing the amount of data has little effect on the compression subsystem as seen by the small differences in the compression ratios as the size of the files increased

Encryption Subsystem

From the encryption experiments performed, we can validate that the IMU and encryption subsystem pair is deemed functional. As mentioned in our encryption experiment, we would first store our IMU subset data in our buffer and display the buffer contents to the console and as can be seen in Figure 8 below, the first 7 lines on the CoolTerm console consists of the IMU data where Ax,Ay and Az represent the accelerometer readings in it's 3 axes and Gx, Gy, and Gz represent the gyroscope readings in it's 3 axes. Our buffer size was set to store 1024 bytes worth of IMU data readings and as a result only 10 accelerometer and gyroscope data pairs could be stored at a time. The buffer is expandable, as our stm board still has around 2.5kb left worth of RAM, thus an extra 1024 would be able to store an additional 10 and possibly more after removing unnecessary "Ax,Ay,Ax" labels and brackets. The 10 samples were easily encryptable as can be seen in the lines 8-19 in Figure 8 below, representing hex cipher text of the IMU data. It can also be seen that the hex cipher text and the original IMU data do not match. The initial experiment was to see if the entire 1024 byte concatenated buffer could be sent for encryption however, the encryption algorithm had an dealing with that much data and as a result, the concatenated buffer string had to be broken into chunks of 256 bytes, where each chunks would encrypted and displayed to the console. Similarly, decrypting the cipher could only be done in 256 byte chunks, however the decryption was still deemed functional as can be seen from line 20-26 in the figure below, representing the decrypted cipher text which can be seen to match the original IMU data in lines 1-7. Encrypting and decrypting larger chunks of IMU data will be investigated further.



```
(Ax : -164.00, Ay : -244.00, Az : 18508.00 | Gx : 1.00, Gy : -15.00, Gz : 13.00 ) || (Ax : -7.00, Ay : -8.00, Az :
2047.00 | Gx : 0.00, Gy : -3.00, Gz : 0.00 ) || (Ax : -1.00, Ay : 1.00, Az : 2056.00 | Gx : 0.00, Gy : 0.00, Gz :
-1.00 ) || (Ax : 3.00, Ay : -6.00, Az : 2053.00 | Gx : 0.00, Gy : -2.00, Gz : 1.00 ) || (Ax : -5.00, Ay : 4.00, Az
: 2055.00 | Gx : 2.00, Gy : 0.00, Gz : 1.00 ) || (Ax : -1.00, Ay : -12.00, Az : 2039.00 | Gx : -3.00, Gy : 2.00, Gz
: 1.00 ) || (Ax : -7.00, Ay : -2.00, Az : 2053.00 | Gx : -3.00, Gy : -4.00, Gz : -1.00 ) || (Ax : -2.00, Ay :
-4.00, Az : 2052.00 | Gx : 3.00, Gy : 0.00, Gz : -2.00 ) || (Ax : -5.00, Ay : -7.00, Az : 2056.00 | Gx : 0.00, Gy :
-3.00, Gz : 1.00 ) || (Ax : -8.00, Ay : -3.00, Az : 2049.00 | Gx : 0.00, Gy : -2.00, Gz : 4.00 ) ||
42846baf791343f4a5113527d2e59c204d19e6eb1a55fd7e6a556293ea670ea30cf7a6dc2658246eec3e3061bdda153d585a16f9b76def6d7fe
c0ef50b4fa28290e8accf373e3dd5d58bb8de55291f878353baa78aaa3e51cf9f848f8effc14e89519633e3d9a936dc91fd44241c075795806b
ff643106bd8a245813005a402a
6960a02f45a18d6d669795ec8b5b00cff5fada98c1eb670d0fade2eb59904528f47b05deb7dc212132295dc5a65cd7c7c14bdc056da98c29241
9095f72d27be680558d18849233ca759ffe14e88ef37a23d1c6211e1043056ce068f8c7b8016332086a5595ffc0e4f9531ddd597b97dac51841
43f8e4a011a2777246c24c771f
80104515893a3a7847b403f3ea4ce68dc496ab78d057317d30dde7a548fb9abf0dce2ba0e749c341e4df10b448fe347ca55a279e6a3e5b28d8a
14c6c20536e389c9ed04d54015e45fcd87c52827b61f05285ae174bf4673d056fcb1327896ea9af4b4907a38b5e30c05df25cf68e8558c408cb
538991dfa7c12ef101a3c97973
6b4e7628a1f3a2294b450bb1b76c9bc869ca33969dea0cfc74be47b6c5f500052a7f64c17408e7ed90c9cf4f9130242c40a0eade393d11d1c53
9b57cefc1c5ede5eb3767605c7f94a4e4512cfd4b5b949eeb70a3053d2b2bb0ee5ac0ef7ada065bb8aec4fe21a4699717c570f366e5d62cb906
9296f53bba2718f1c6f663756e
0d7dbcaa728d3be9a8f5c9ac5c166974a3f4430319bb03c55f2249c797556a5d70ef62e53cbb4f94f507e988e92dfce35059a4574ba9205f3a3
e9de195462a1a756de11efda7620d03450b3db29bffb5f51f368c9cebcf262ba00d74c2584d0f743c0fe37945a9daacb3f22a5d09135a
3d57d1d4b0e31517f7dd45624bbe9ffc905430e35cf05e0a1d3004a6c7ef030ac87d8b1d5c74db1562f49c51218a0a52b647e970e4a468c9e94
02d856e2240b5d165404791ef65f30646c524f3591643684b3405fa135627eb47e8e1d281e30fb3102e7fdebb37e73392032e8bd2455777ae46
3dd785ea478d254d0121df14af
f0f52c6eb212092a41ea2298733b2098dc4d1d793a44e545eb2f2ae158c5a2744ccd588d5601191b7a55691a7a66a5bce45c6d227331f42b067
cf5e6c4d243b9ef5fdba00d0974d3d6da749d4e42c2ef71554e6313c8e607653ae6c36282590ce7ebf4ee16f876c66f36b4d7d75da083
(Ax : -164.00, Ay : -244.00, Az : 18508.00 | Gx : 1.00, Gy : -15.00, Gz : 13.00 ) || (Ax : -7.00, Ay : -8.00, Az :
2047.00 | Gx : 0.00, Gy : -3.00, Gz : 0.00 ) || (Ax : -1.00, Ay : 1.00, Az : 2056.00 | Gx : 0.00, Gy : 0.00, Gz :
-1.00 ) || (Ax : 3.00, Ay : -6.00, Az : 2053.00 | Gx : 0.00, Gy : -2.00, Gz : 1.00 ) || (Ax : -5.00, Ay : 4.00, Az
: 2055.00 | Gx : 2.00, Gy : 0.00, Gz : 1.00 ) || (Ax : -1.00, Ay : -12.00, Az : 2039.00 | Gx : -3.00, Gy : 2.00, Gz
: 1.00 ) || (Ax : -7.00, Ay : -2.00, Az : 2053.00 | Gx : -3.00, Gy : -4.00, Gz : -1.00 ) || (Ax : -2.00, Ay :
-4.00, Az : 2052.00 | Gx : 3.00, Gy : 0.00, Gz : -2.00 ) || (Ax : -5.00, Ay : -7.00, Az : 2056.00 | Gx : 0.00, Gy :
-3.00, Gz : 1.00 ) || (Ax : -8.00, Ay : -3.00, Az : 2049.00 | Gx : 0.00, Gy : -2.00, Gz : 4.00 ) ||
```

COM6 / 9600 8-N-1
Connected 00:41:57, 9339 / 2312 bytes

EEE3097S.stc
File Edit Connection Macros View Remote Window Help
New Open Save Connect Disconnect Clear Data Options View Hex Help
TX RX RTS CTS DTR DSR DCD RI

Figure 8: Encryption & Decryption Experiment Result

ATPs

Table 2: ATPS from Milestone 1

Specification(s) ID	Specification(s)	Review
DS003	We shall use the Huffman encoding algorithm to compress our data from the IMU	It is lossless so that we are able to retrieve at least 25% of the Fourier transform coefficients
DS004	The Huffman encoding scheme takes advantage of the disparity between frequencies and uses less storage for the frequently occurring characters at the expense of having to use more storage for each of the more rare characters.	Fast as it is efficient in compressing text or program files
DS005	The microcontroller will communicate with the IMU in order to receive the data & a serial connection between the microcontroller and PC will be established	A serial connection was successfully established between the microcontroller and PC, allowing for data packets to be stored on and received from the microcontroller
DS006	The use of an AES library to encrypt and decrypt the compressed IMU data and ensure security.	AES-128 was used to successfully encrypt and decrypt the IMU data ensuring 100% security
DS007	Choosing an encryption algorithm that limits the number of processors done on the microcontroller.	The AES library provided by kokke[5] is a simplified version of the general AES library, intended for areas where memory and processing power are constrained. Efficiency testing is yet to commence

Admin Documentation

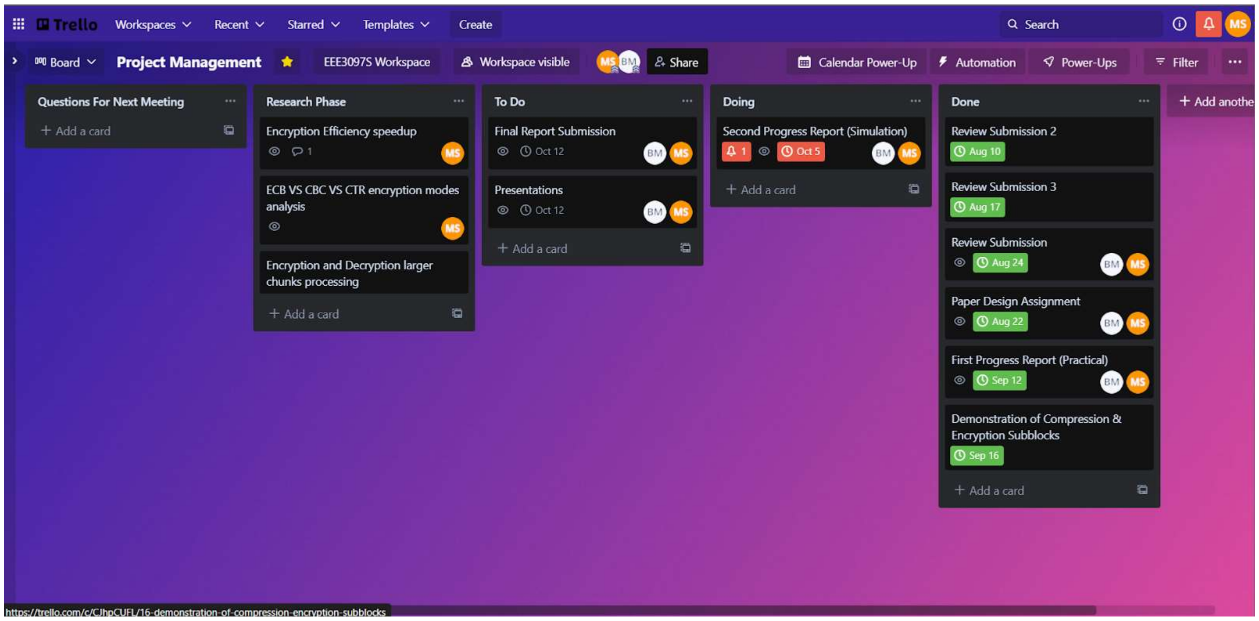
Distribution of Work

Table 3: contribution of each of the team members

Work	Member
Admin documents	MTMBRE002 & STLMOU001

IMU Module	STLMOU001 & MTMBRE002
Experimentation & System Functionality	MTMBRE002
Compression Subsystem	MTMBRE002
Encryption Subsystem	STLMOU001
Results & System Functionality	STLMOU001
Compression Subsystem	MTMBRE002
Encryption Subsystem	STLMOU001
ATPs	MTMBRE002 & STLMOU001

Project Management Tools



Questions For Next Meeting
- Add a card

Research Phase
- Encryption Efficiency speedup
- ECB VS CBC VS CTR encryption modes analysis
- Encryption and Decryption larger chunks processing
- Add a card

To Do
- Final Report Submission — Oct 12
- Presentations — Oct 12
- Add a card

Doing
- Second Progress Report (Simulation) — Oct 5
- Add a card

Done
- Review Submission 2 — Aug 10
- Review Submission 3 — Aug 17
- Review Submission — Aug 24
- Paper Design Assignment — Aug 22
- First Progress Report (Practical) — Sep 12
- Demonstration of Compression & Encryption Subblocks — Sep 16
- Add a card

https://trello.com/c/CJhpCUFL/16-demonstration-of-compression-encryption-subblocks

Figure 9: Our Project Management Tool

- Here is the link to our github repository here

Development timeline

The project is still planned to be completed within the course timeline. Changes were made when the Second progress report got an extension causing the deadline for the final report to be pushed back by the same number of days as the extension for the second progress report. The demo deadline remains the same.

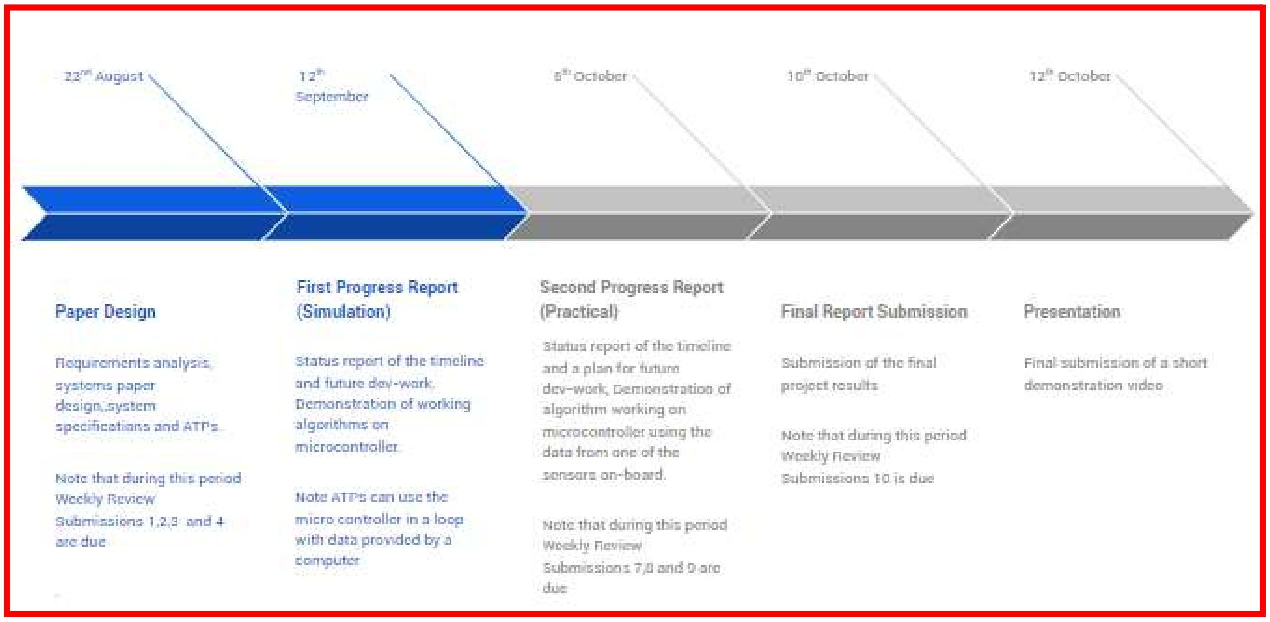

22nd August
12th September
5th October
10th October
12th October
Paper Design
Requirements analysis, systems paper design, system specifications and ATPs.
Note that during this period Weekly Review Submissions 1,2,3 and 4 are due
First Progress Report (Simulation)
Status report of the timeline and future dev-work. Demonstration of working algorithms on microcontroller.
Note ATPs can use the micro controller in a loop with data provided by a computer
Second Progress Report (Practical)
Status report of the timeline and a plan for future dev-work. Demonstration of algorithm working on microcontroller using the data from one of the sensors on-board.
Note that during this period Weekly Review Submissions 7,8 and 9 are due
Final Report Submission
Submission of the final project results
Note that during this period Weekly Review Submissions 10 is due
Presentation
Final submission of a short demonstration video

Figure 10: Our Project Development Timeline

References

[1]"Sense HAT (B) for Raspberry Pi, Onboard Multi Powerful Sensors .." https://www.waveshare.com/sense-hat-b.htm (accessed: Oct. 05, 2022).

[2] Lones and k5054, "File compression and decompression and C," *CodeProject*, 22-Dec-2021. [Online]. Available: https://www.codeproject.com/Questions/5320638/File-compression-and-decompression-and-Cv. [Accessed: 05-Oct-2022].

[3]"ICM-20649 Datasheet | TDK." https://invensense.tdk.com/download-pdf/icm-20649-data-sheet/ (accessed: Oct. 05, 2022).

[4] Admin, "How to use UART to transmit data in STM32 || poll || inetrrupt || DMA," *ControllersTech*, 14-Dec-2021. [Online]. Available: https://controllerstech.com/uart-transmit-in-stm32/. [Accessed: 05-Oct-2022].