Tasks that I have completed:

- Created the ability to encode ASCII to CAM, and printing the output to the serial monitor.
- Created the ability to decode CAM to ASCII, and printing the output to the serial monitor.
- Created the ability to transmit the CAM, when encoded from ASCII, on the red LED.
- Created the ability to transmit the CAM, after decoding it to ASCII, on the blue LED.
- When receiving the string, “IR*” or “ir*”, in CAM on the serial port, it will emit Infra-red at tone 38,000Hz from the IR emitter for 2000ms.
- When receiving the string “RXIR” or “rxir”, in CAM on the serial port, it will print to the serial port “HIGH” or “LOW” if the value on the IR Receiver is HIGH or LOW respectively. An interesting point to note is that the IR receiver in testing always was at HIGH by default until it received an input, and only with an input would it be set to LOW.
- When receiving the word “PV” or “pv”, in CAM on the serial port, it will print out the current analogue value on the potentiometer and transmit it to the serial port encoded into 4 digits of CAM.
- When receiving the word “LP” or “lp”, followed by 12 number digits it will then light up the red, orange, yellow and green LEDs with the relative brightness that has been given in the 12 digits.

Problems I encountered:

I encountered a few problems with nearly every function I made leading to in some cases hours of attempted Debug, hence the massive amount of pre-processing for DEBUG, but the function and part of the task that gave me the most problems is LP and the following 12 digits. I found that making that function to be difficult, and when integrating it with the set up requiring an entirely different function to integrate with the current setup in the loop() part of my code.

For example, when running the testing harness for the first time, my code worked on my more up to date version of Arduino but didn’t compile on the harness, so I had to re-write a few lines of my code to use different methods to achieve the same ends.

Another example of a problem I encountered when testing the code is when I received my test back after it was sent through the harness I had three glaring errors, one of which is that CAM translated into lowercase ASCII. This was easy to fix by changing the output of my cam2char function to have uppercase instead of lowercase.

The test harness’ second error, was with a string input which returned ‘#’ on the harness but we hadn’t been given that string input for anything in terms of the translation, I fixed this issue in a seemingly haphazard way by adding the ‘#’ to the translation and adding the string that seemingly corresponded to it into my function, cam2char.

The test harness’ third error, was with having a double space at the start of the string that was to be translated into CAM, this was because of my lack of foresight, I fixed it by adding extra clauses to an if statement to run the CAM translation, so that it actually ran based on the first ‘non-space’ character in the string/char array. This was instead of just leaving it to the first character in the string. I then had another error where the CAMstring2string function (which translated a string of CAM characters into an ASCII string) would not translate spaces at the start of the CAM properly, this was fixed by adding an extra else if statement, taking into regard previous positions being null or a space, before doing the just if space check. This extra else if statement allowed the space to be

added to the returned value of CAMstring2string without a letter previously being there, being required.

Before using the test harness, I had this curious error that was fixed with a seemingly random number, now this error came about by using a memory pointer as an integer. It was caused by using the strstr() function from C++. However, I managed to get the program to work despite this, by taking away 2171 from the memory address. This caused an error further down the line fixed by Serial.println(output of strstr() – 2171), I believe this was an error in how the Arduino was storing the variables in memory. I fixed this by converting my array of characters into a string and then using string.indexOf(), this allowed for basically the same thing as I was achieving using strstr() – 2171, without the errors I was getting previously.

What do I think that I deserve for my work?

In regards to work that I have put so much effort into I would say that I fulfil the requirements entirely to receive between 60 and 69% (a 2-1), however, I would hate to have undersold myself and because I fully understand how my code works and if I had to write it again I could. I believe that my code deserves between 70 and 79% however I am not confident in my ability to produce a nice document to go with it.

Overall I believe that it would not be unfair to say the work is of a 2-1 standard (60-69%), however, I think that it would not be out of order to receive a 1st (70-79%), for this piece of work.