ECEN 5813 – Principles of Embedded Software

Final Project Proposal

Date	04/11/2022
Prepared By (Name)	Ayush Gupta

1. Overview

Implement Huffman Coding algorithm on FRDM KL25Z development board to compress the messages sent over Serial Port.

2. System Block Diagram

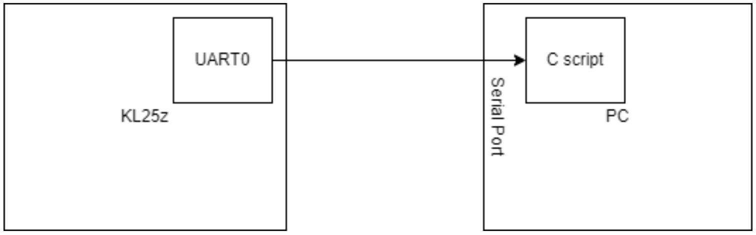

Figure 1 System Block Diagram

3. Hardware Requirements

- FRDM KL25Z Development board
- A PC with ubuntu operating systems

4. Software Requirements

- MCU Expresso IDE version - 1.2.0
- GCC compiler
- Teraterm/Putty to test the UART Transmitter and Receiver

5. Software Modules

Following are the software modules required to realize the proposed solution:

- **UART0 module**: Initialize the UART0 module on KL25Z to be able to sent out data on the serial port.
- **Huffman Coding Algorithm module**: This module will have all the Huffman coding logic and to be called before transmitting the string on to the serial port.
- **Circular Queue Module**: This module will have the implementation of Circular Queue which will act as a buffer, required to store the transmitting data.
- **Test Huffman Module**: This module will have the automated test cases to verify the Huffman Coding Algorithm.
- **Test Circular Queue Module**: This module will have the automated test cases to verify the working of Circular Queue.

6. Technologies

- Huffman Coding (Data Compression Technique)
- Circular Buffers (to store the messages to be sent over the serial port)

7. Software Flow Diagram

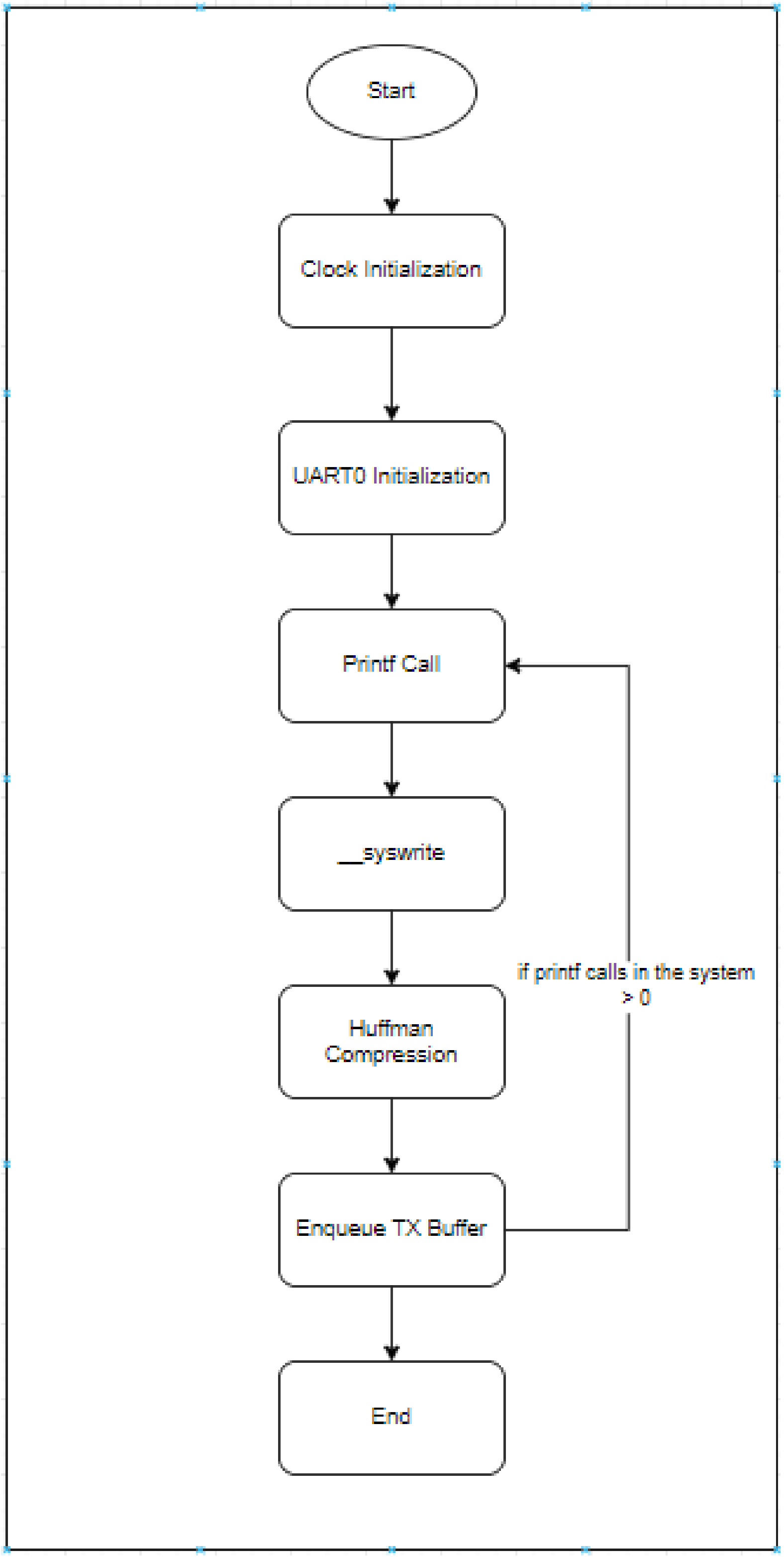

Figure 2 Software flow diagram on the Transmitter side (KL25Z)

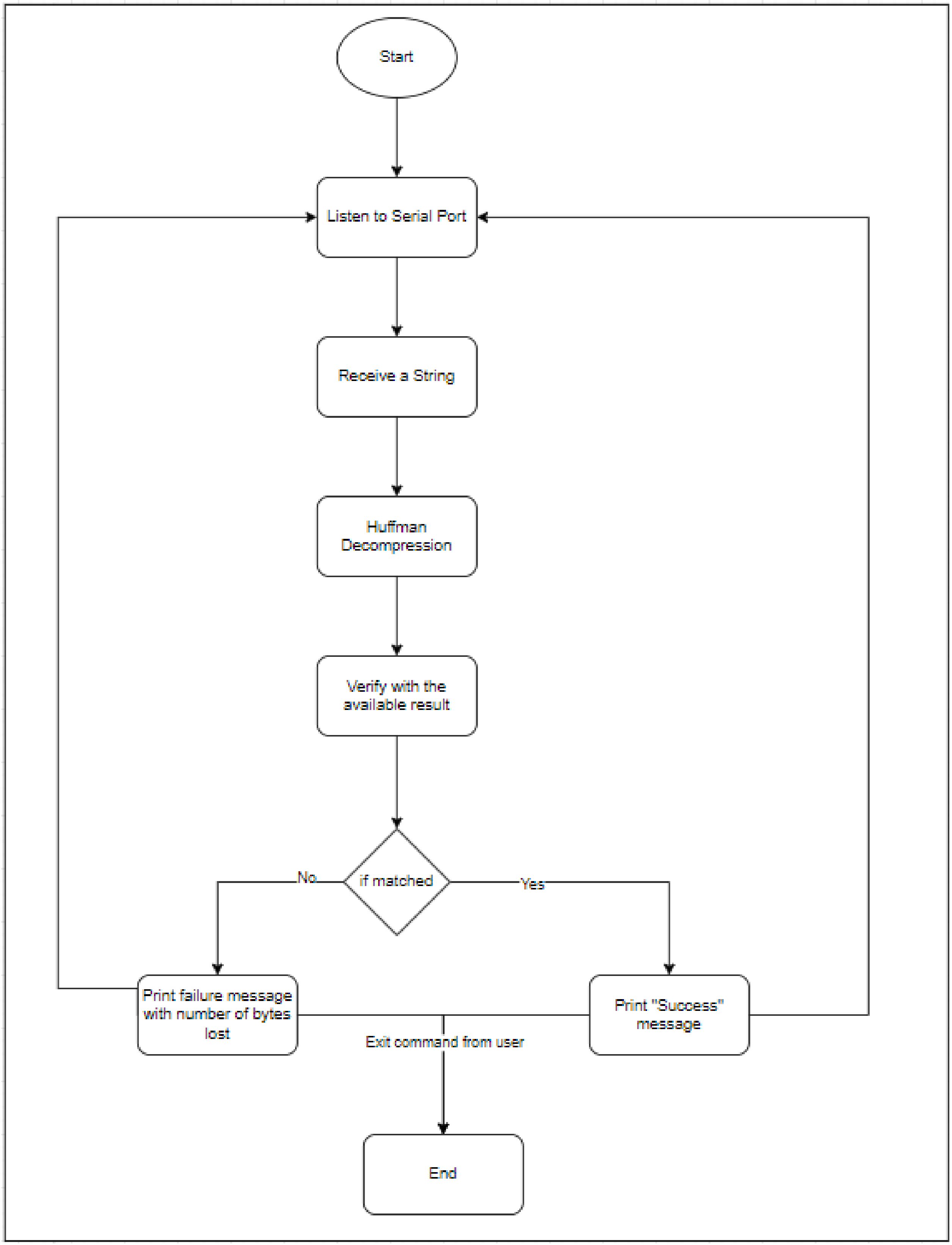

Figure 3 Software flow diagram on the receiver side (PC)

8. Functionalities

- Develop Huffman algorithm and gluing up with Redlib library to compress data and clearly demonstrate the data compression advantages (in form of a compression statistic).
- Will develop a C program on the receiving end (PC) to receive debug messages over UART to decode the received message and match it with the expected output. This program could also keep track of compression statistics, since it can track bytes in and bytes out.

9. Anticipated Learnings

- Working of Huffman Coding Algorithm.
- Learn to implement Huffman compression algorithm on the KL25Z by gluing Huffman coding with Redlib library (__syswrite API), i.e., sending out the compressed data on serial port.
- Learn to access the serial port using C and integrate it with the decompressing algorithm, to verify the results.

10.Future Work

- A bi-directional communication channel to communicate between the PC – KL25Z or between the 2 KL25Z MCUs.

11.Testing Strategy

To verify the above system, most of the testing will be done using automated test cases written on the receiving end. Test cases includes the below mentioned happy, corner and appropriate error handling cases. Initial tests will be performed functionalities wise (refer table 1) with the test cases from message strings (refer Table 2).

Table 1 List of functionalities tests

Test #	**Functionality Tests**
1	Verify the working of FIFOs (TX FIFO)
2	Test the working of Huffman Coding compression algorithm
3	Test the accuracy of Huffman Coding decompression algorithm
4	Integration of Huffman Coding with Redlib & verify the output with TeraTerm/Putty
5	Test the communication between C script and serial terminal and verify the Huffman decompression output with TeraTerm/Putty
6	Integration of both the systems to test the system

Table 2 List of test cases to be performed at every step to verify working of Huffman Coding Algorithm

Test #	**Test case Description**	**User Input Examples**	**Automated/Manual**
1	General debug message	“PES Final Project Test”	Automated
2	Message with a single character	‘r’	Automated
3	Message with a single character repeating over its length	“rrrrrrrrrrrrrrrrrrrrrrrrrrrrrr”	Automated
4	Message with punctuations included	“aaaaaas2232!2@323@!#! !#@#!!!#$!@!#$@#!aaaaaaaaaaaa”	Automated
5	Message without character repetition	“abcdefghijklmnopqrstuvwxyz”	Automated
6	Blank string		Automated
7	*UART message capacity test		Manual

* Test #7 will be performed for various string input sizes to test the capacity manually.

12.References

[1] https://en.wikipedia.org/wiki/Huffman_coding

[2] https://brilliant.org/wiki/huffman-encoding/

[3] https://www.cyberciti.biz/faq/find-out-linux-serial-ports-with-setserial/

[4] https://www.mouser.com/pdfdocs/FRDM-KL25Z.pdf