PRINCIPLES OF EMBEDDED SOFTWARE PROJECT PROPOSAL

Name: Sricharan Kidambi
Course: ECEN 5813 Principals of Embedded Software.
Date: 13th April 2022

Project Objective.

The main aim of this project is to compress and decompress the messages displayed on an UART. This enables us to print messages at a quick rate and lossless transmissions can be observed.

Overview:

A text communication is the most basic form of communication. Communicating something through text is the easiest way to let someone know the crux of the information at lower cost, bandwidth, and power. However, that comes with certain added responsibilities as well. The message must be sent without any losses. Whereas video and audio streaming losses are ok, compared to text. But there is a tradeoff between efficient transmission and speed. To resolve this, we use a technique called **data compression and decompression.**

Data Compression

The data compression is a technique used to reduce the size of the transmitted data. A message consists of alphabets, numerals, special characters. Which when converted to ASCII code, has character codes up to 128, hence in general conditions the ascii length is 1 byte. Now, we encode it with a fixed length. The number of bits of that character is largely related to frequency of occurrence of that character in the message by the below formula.

Character size in bits (C) = frequency of that character * message length (8 in general).

Message size = Sum of character size in bits of all characters (∑(C)).

This is then decompressed in the receiver end to get back the original message.

Frequency of a character is inversely proportional to the message code for that particular character.

Technologies Used

We use the UART0 module to communicate between the KL25Z Freedom board hardware and the host system. Once both the transmitter and receiver agree on the basic parameters of communication. Communication occurs between the transmitter and receiver. A circular buffer-based technique to store and transfer the bits is used by us in this coursework. The improvement done here is implementing Huffman encoding in this module to print the debug printouts from the input and output. MCU Xpresso IDE tool used regularly for FRDM KL25Z by NXP semiconductors is the IDE tool used here.

Test Suite:

Encoding techniques generally demand good test suites to check intermediate stuff. I am planning to implement Systick Timers in this module to regularly test the timing of the tool and that will facilitate as a timing modulating tool of the software.
Initially we can test by sending a single character and check the timing in the output module. Slowly improve the size of the string and the number of unique characters sent via STDOUT.
In production code, there is a function that stores the frequency of the characters, and the Huffman encoding is based on this function. Hence the more the distinct characters sent, the greater the speed of the transaction. For this function I am planning to develop a good test suite for that function before delving into integrating UART and this compression function.

Further Possible Enhancements

A further possible encoding can be integrated in addition to the existing Huffman function which an start to operate only when the necessity arises beyond the possibility of Huffman encoding. This can be tried to implement if time permits before the actual submission of the project.

Hardware

No external hardware is required for this since the KL25Z FRDM board which is serving as the transmitter for this interface consists of timing monitor equipment's like Systick and interface communication equipment's like UART inside the device architecture itself.

Key Learning Outcomes

1. Understand how compression and decompression works in embedded systems.
2. Configure UART0 module with compression techniques.
3. Incorporate timing (Systick) with a asynchronous serial communication.
4. Minimum level of optimization might be needed to match with the required speed.

References:

1. https://www.youtube.com/watch?v=iEm1NRyEe5c Learn basics of Huffman Encoding.
2. KL25Z reference manual for NXP Semiconductors.