General Meeting Minutes

Date: 15-09-2023
Time: 16:15
Attendees: BMJO, Lal
Notes: Explaining how to process data coming through web socket
AudioMarker:0:0

Main Points

- Adjust pointer to a specific position (zero) once a certain limit is reached on the buffer.
- Upon reaching the limit, reset the pointer and start filling the buffer again from zero.
- If a packet larger than the current buffer space is encountered, fill the available space and add the remaining packet to the beginning of the buffer.
- Presentation of data on a graph with a refresh rate of 10 times per second. Using a timer for this repeated refreshing process.
- Recording mechanism is used for future reference.
- The size of the ring buffer can be decided based on specific requirements like sampling rate and display time.
- Ring buffer operates on a first-in, first-out basis and when the end of the buffer is reached, it wraps around and starts from the beginning.
- The subsequent step involves clearing a small block in the buffer to create a visual gap between current and previous data.
- Creation of a ring buffer for data storage and a timer for periodic refresh and update of the graph.
- The ring buffer and refresh timer are critical steps in organizing and visualizing data effectively.

Discussion Details

The discussion started with an explanation of pointer adjustment in a buffer system. As per the discussion, the pointer is initially adjusted to a specific position. Then the system moves along this adjusted pointer and upon reaching its endpoint, resets the pointer back to zero.

Once this occurs, the system begins to fill this buffer from the starting point. The buffer continues to fill up until it gets completely filled. Once it reaches its capacity, the system resets and starts filling the buffer again from zero.

However, it was mentioned that there may be occasions where a packet larger than the remaining buffer space is obtained. In such instances, the system fills the available buffer space and then begins filling the remainder from the start of the buffer again. An example given was when a 100-byte packet is received, but only 20 bytes of space is available. In this case, the system fills up from 0 to 20 and then the remaining 80 bytes

is filled in at the beginning of the buffer.

The discussion also touched upon the visualization of these processes, through a graph which would be refreshed ten times a second using a timer. To aid understanding, it was suggested that the process could be explained again.

It was clarified that the conversation had already been recorded to be reviewed later. The usefulness of this 'ring buffer mechanism' was stressed, as this is the procedure used in most scenarios. The size of the ring buffer can be decided based on specific needs. An example was given where for a waveform display of 10 seconds with a sampling rate of 255, a buffer of 255x20x10 would be maintained.

Continuing the explanation, it was stated that once the buffer starts filling, the process can be observed in a waveform display. It's noted that there's a continuous flow where once the end of the buffer is reached, it restarts from the beginning. In order to prevent confusion, it was advised to fill up the buffer and display it. The next step would involve clearing a small block in the buffer with zero's, creating a gap between the current and previous data points. This gap will subsequently appear to move across the graph as buffer refills and graph refreshes.

The conversation concluded with the understanding that a ring buffer needed to be created according to the explained procedure. It would then need to seamlessly integrate with a timer that refreshes the graph 10 times per second. The entire process involved using pre-existing data and employing it to refresh the graph for capturing its latest version continuously. The last statement is in Telugu and translates to "I bought, I bought, bought, did the action of buying".