ECEN 5623 – REAL TIME EMBEDDED SYSTEMS

SRICHARAN KIDAMBI – ASSIGNMENT 5

**PROJECT PROPOSAL**

**Goal and Objective of the Project:** To build a visual synchronization frame capturing system that capture the best frames at 1Hz and 10Hz frequency in a completely glitch less over a period of 30minutes. The goal is to neither skip a tick nor duplicate a tick. The 1800 frames in total should be captured and displayed exactly at 1 second and 10 second each.

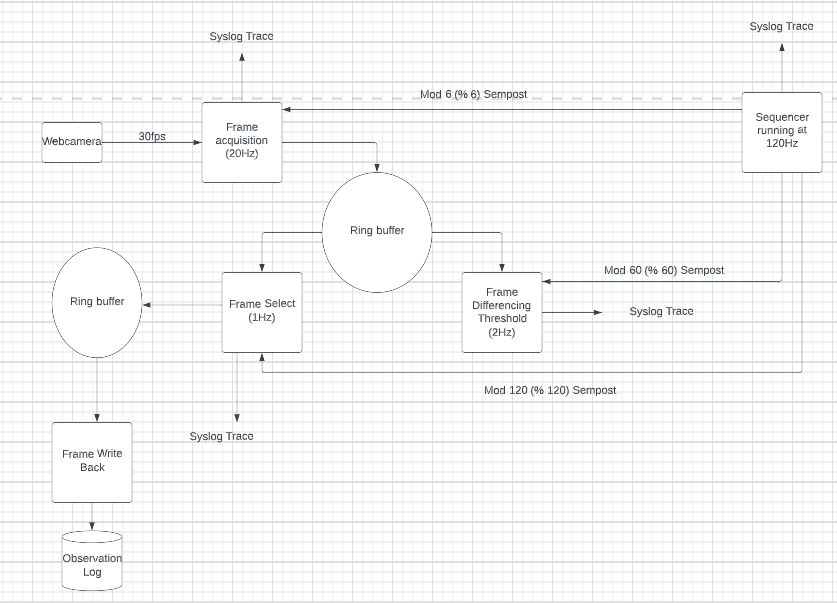
**Components required:** A Camera with UVC driver, A linux compatible Microcontroller, An analog clock for validation and verification.

**Project Requirements:**

This project requires to acquire individual frames on the camera, commanded at an image of 1Hz for the minimum requirement and 10 Hz for the stretched requirement. The restored image is expected to be written to the flash or the non-volatile memory in PPM format. There should be no observable software glitch over the 30minute period which should be verified and validated by calls to syslog tracing every capture using timestamps. The image acquisition is expected to be accurate enough to acquire all 1800 frames at 1 second interval. The requirements should finally be verified by computing the average fram jitter to determine if there is any accumulated latency.

**Project Software Flow Diagram:**

Since this is software-based project, please find the software flow diagram of the project.



The above flow diagram was inspired by Prof Sam Siewart’s explanation video

Source: <https://drive.google.com/drive/folders/13fHc9Xa6TcG7qXqRfIxoLVKXlegG5LAC> (Video L-N9.4)

**Project Procedure:**

When synchronizing a real time embedded system to provide a deterministic, predictable and fault tolerant response. A sequencer module which uses an auxiliary clock that generates an Interrupt every 120 times per second (120 Hz) is utilized. This oscillator can also be duplicated for redundancy. An interval is loaded to the software and if they match, a sequencer service of 120Hz is obtained. The value 120Hz is chosen considering the below 2 factors:

1. The sequencer should run at a highest frequency compared to other driving systems (frame acquisition, frame differencing and frame selection) which runs at a sub rate of this frequency.
2. The sequencer should not steal a lot of CPU time and hence should not increase the overall latency of the system. (That can cause the system to miss deadline).

It posts 3 binary semaphores:

* 1. Mod 6 semaphore for frame acquisition.
  2. Mod 60 semaphore for frame differencing
  3. Mod 120 semaphore for frame selection.

Now, the frame acquisition will be running at 20Hz, which will be driven by the stable oscillator.

The acquired frames are fed into a ring buffer which is RAM based, this is added to the remove the IO coupling which can increase the IO latency.

The frame differencing module will select the most stable frame from a differencing algorithm. For this design, it will select the most stable frames out of 20 best frames from the ring buffer every 500msec.

The most stable frame selected at 1Hz frequency created due to the Mod 120 semaphore from the sequencer in the frame select system. This is again fed to a ring buffer for safe writes to the non-volatile memory. Since the write back does not involve processing, it can still be a best effort system while the rest of the systems must be a real time system. This design thereby creates a fault tolerant, predictable response real time system to capture frames at exactly 1 second and exactly 100msec (stretched goal – 10Hz) without software glitches throughout the execution time. Syslog traces are used for verification and validation of all the involved subsystems.

**Cheddar Analysis**

For obtaining the cheddar analysis, we have 4 services

1. Sequencer - (S1)
2. Frame Acquisition - (S2)
3. Frame Differencing - (S3)
4. Frame Selection. - (S4)

Write back is considered a best effort system and is hence not taking place in the cheddar analysis.

Time periods for the 4 operations to capture 1800 frames

1. Sequencer period – T1 = (1/120) \* 1800 = 15 seconds (--120 Hz frequency)
2. Frame acquisition period – T2 = (1/20) \* 1800 = 90 seconds (--20 Hz frequency)
3. Frame Differencing period – T3 = (1/2) \* 1800 = 900 seconds (--2 Hz frequency)
4. Frame Selection period – T4 = (1/1) \* 1800 = 1800 seconds (--1 Hz frequency)

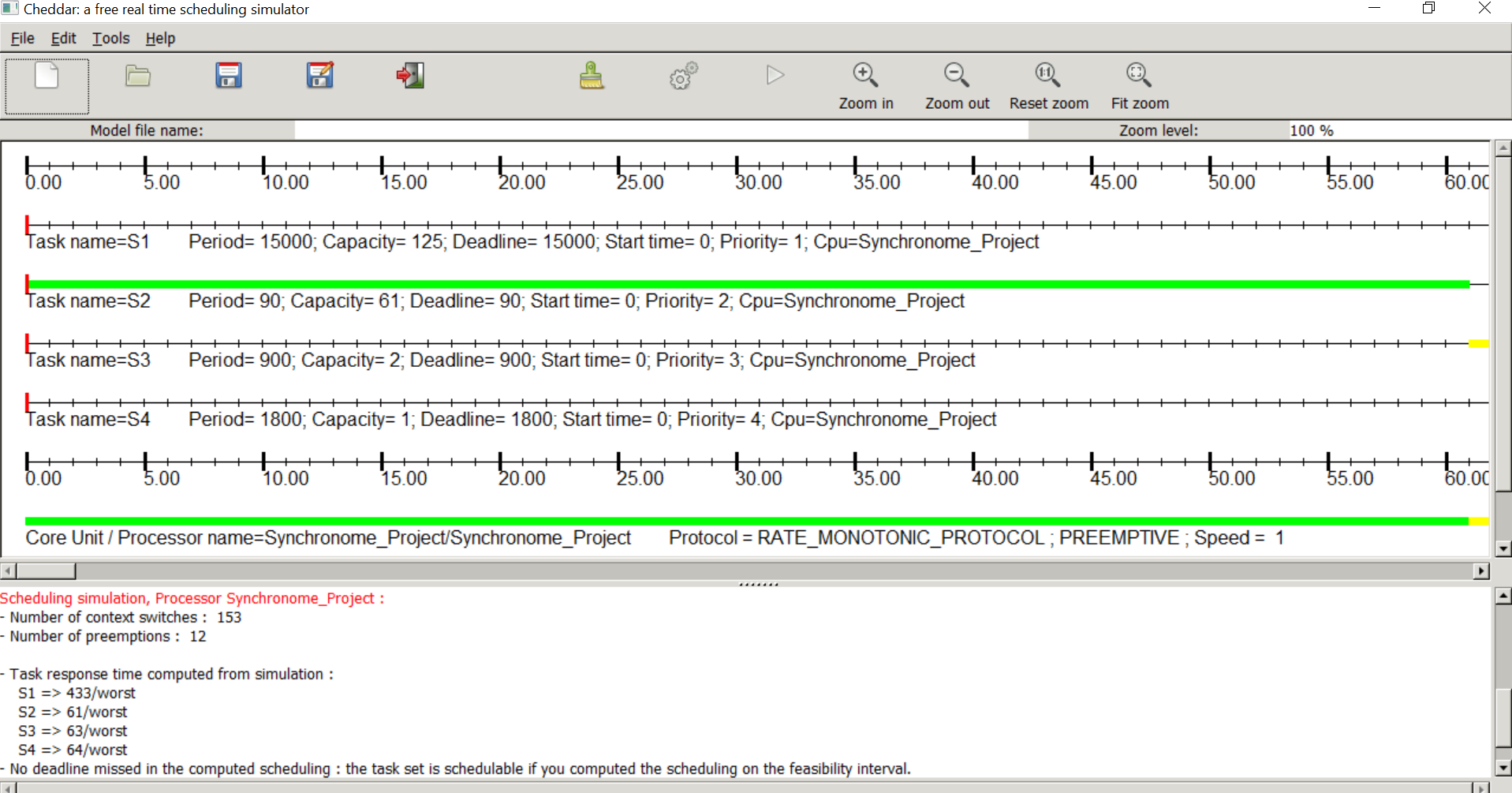
C – Capacity for all the 4 services

1. Sequencer capacity: (C1): Running a sequencer program with the foreseen

Conditions. – total time consumed = 125msec, 0.125 sec

1. Frame acquisition capacity: (C2): Frame acquisition time taken found out to be 61sec.
2. Frame differencing capacity (C3): For analysis assuming this to be 2 second
3. Frame selection capacity (C4): For analysis assuming this to be 1 second

**Cheddar Diagram for Rate monotonic Feasibility**

****

**Result of cheddar diagram:**

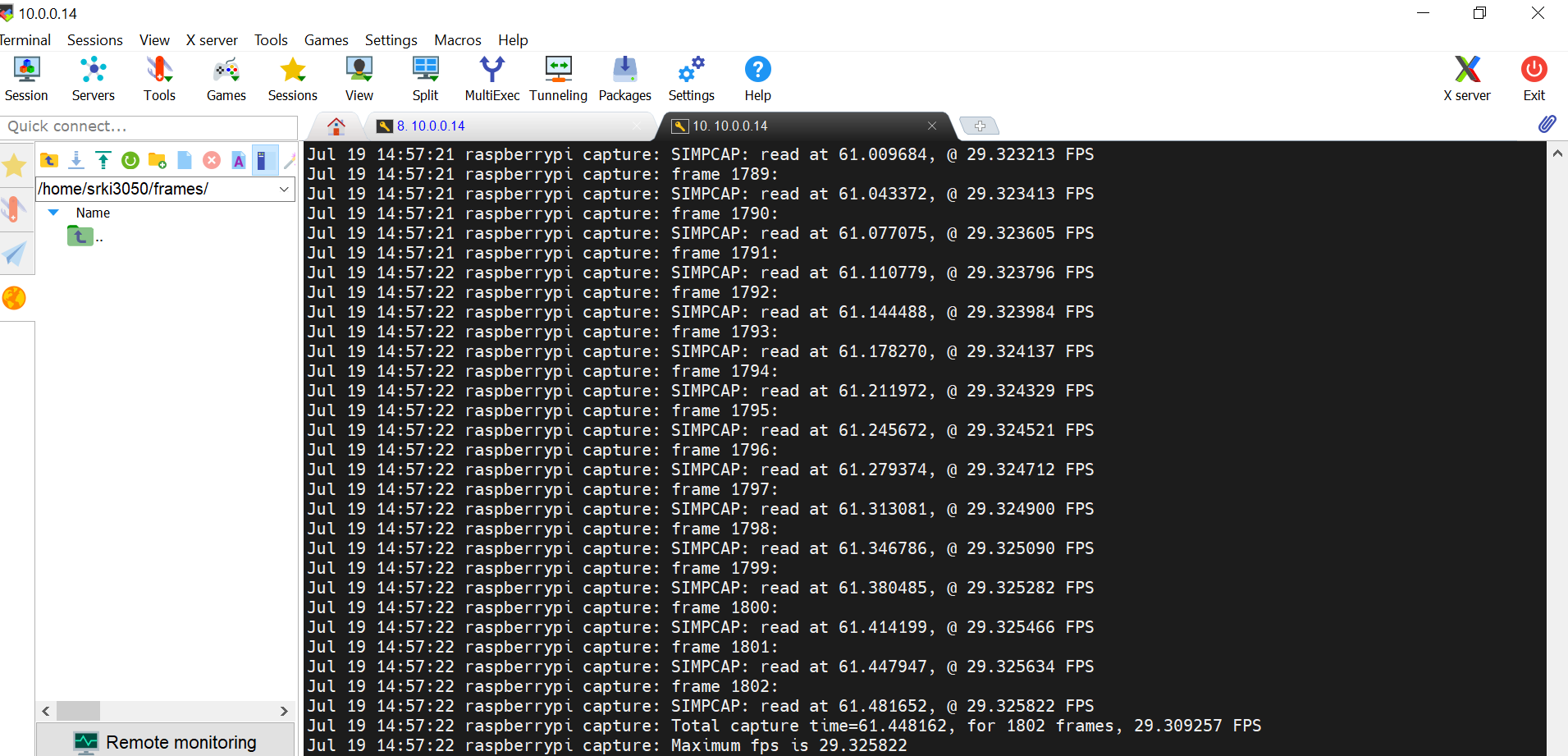
The RMA analysis for feasibility tests were validated and found to be feasible, hence comfortably proceeding with rate monotonic analysis for this project.

**Schedule Plan**

|  |  |
| --- | --- |
| Date | Target Completion of the process |
| 07/28/2022 | Create the makefile and understand the dependencies of all the involved processes  + Report Preparation. |
| 07/29/2022 | Interface a simple capture module with the sequencer program to run frames at 120Hz  + Report Preparation. |
| 07/30/2022 | Interface circular buffers to the program to remove IO coupling.  + Report Preparation. |
| 07/31/2022 | Implement frame differencing at 2Hz frequency that selects the most stable frames  + Report Preparation. |
| 08/01/2022 | Implement frame selection at 1Hz frequency that selects the correct frame.  + Report Preparation. |
| 08/02/2022 | Implement the final ring buffer that implements the write back function to the flash. |
| 08/03/2022 | Testing and Debugging with Professor Sam Siewart |
| 08/04/2022 | Correction of errors and potential failures stated by professor in the meeting. |
| 08/05/2022 | Correction of errors and potential failures stated by professor in the meeting. |
| 08/06/2022 | Testing the 1Hz situation against an external analog clock |
| 08/07/2022 | Implementing all the above scenarios at 10Hz situation once 1Hz is done to perfection. |
| 08/08/2022 | Final demonstration with professor for the project and report validation |

**References used for the proposal**

1. [simple-capture-1800-syslog - Google Drive](https://drive.google.com/drive/u/0/folders/1jv7MJEAvtHw_urJbL6wlrraNb60fhT1f) Ran this program for to calculate the frame acquisition time. (61 seconds) Please see the screenshot below for reference.

****

1. Ran the [Folder - Google Drive](https://drive.google.com/drive/folders/1-EfpRuEKd8w_FwnmIME4gx0fDr6N7mgY) (consisting of the sequencer C program) for three functions at requested frequencies.

Frame acquisition – 20Hz

Frame difference – 2Hz

Frame selection – 1Hz

The corrected c file is attached with this.