ECEN 5623 -Real Time Embedded Systems

PROJECT

TIME LAPSE

By-
RAJ LAVINGIA
UNIVERSITY OF COLORADO BOULDER
EMAIL:- rala9631@colorado.edu
STUDENT ID :- 109123201

TABLE OF CONTENTS

1. INTRODUCTION:-

Aim :- The project "time lapse" is all about getting the perfect accuracy in a system. I have applied my concepts of Real time systems subject in this project. The concept is to generate a very accurate time lapse video.

What is a time lapse ?

It is a technique of photography, when the frequency at which film frames are captured is much lower than what we are seeing in the sequence (Wikipedia).

What is the purpose of time lapse ?

The aim of doing time lapse is to use the scheduling algorithms which are real time. For example, I capture the frames at a slower rate and then those captured frames I run at the faster rate. The video formed with faster rate is called Time lapse video.

The speciality of this video is I can detect a slowly changing object like melting of an ice, rotting of fruits, watch time changing etc in a way that changes are noted quickly. Thus with the help of this video we are able to record the changes which are happening very slowly in real time.

Lets say I want to see the sun rise and sunset of a particular day in just a minute and know the shades of the sky at that time and I don't have the whole day to waste after it, then I will start recording the camera from sunrise till sunset and then will watch the time lapse of that ,with a constant frame rate , thus I will be able to see the sunrise and sunset in a particular time frame. So we can say that time lapse is basically used to detect the slow changes in a fast manner.

Here it is very important that only after the completion of one frame 2nd frame should be started or else I will miss the change that happened in that particular frame. Hence we have to take specific images at a specific frame rate. The frame capture should be without any hindrance. Since this subject was all about making things in real time system , I was able to apply my learnings from the subject.

2. DESIGN OVERVIEW

Hardware Used in this Project :-

1) Raspberry – Pi :-

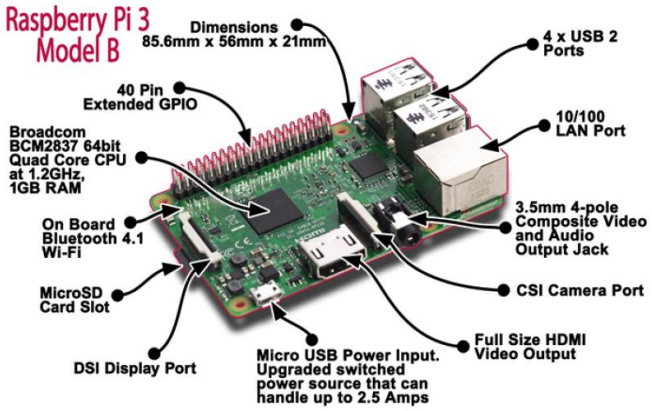


Fig :- 1 Raspberry Pi (Wikipedia)

2) Logitech C200 Camera

Its fps is 30 at 320*240, 640*480 resolution

Fig :- 2 Logitech C200 Camera

3) Mouse and Keyboard :- I have used these for R-pi working environment

3 . Project Requirements :-

1) Resolution :- The image capture done by the camera should be atleast in the resolution of 640*480 (VGA). This is a standard resolution for any TV. We can also use the higher resolution of 720*480.

2) Frame Rate :- The image should be captured by the camera in the resolution of 640*480 at 1 fps. It means in 1 sec the camera should capture 1 image/frame. For this purpose I have used a timer using the POSIX API. CLOCK_REALTIME this API is used for getting the time stamp when the camera captures the image.

3)Saving the frames :- I am saving the captured images from camera at 1fps into a "P6" encoded PPM. The size of this images which are stored in the PPM format are very huge. For my code and capturing the images I am nearly getting the size of Images in PPM as 995 KB. The total size allocated in the R-pi for use is very less. If 1 image acquires so much amount of space in the system then we can store a lot of images. Thus the need of compression arises.

4) Compression of images
I have reduced the size of images from nearly 995KB to 10KB by converting it from PPM to JPG.

5) Adding uname :- In the PPM image data we have to add the uname so that we can know on which system the file is running. We also have to add the timestamp on the image so just by observing the first and last frame of any time period we can find if there is jitter in the system or not.

6) Video :- After the compression of images from PPM to JPG we have to combine these images and make a video in the MPEG2 or MPEG4 format. For running the video on R-pi I am using OMXPLAYER. I have used some functions like Videowriter which comprises of different attributes like running FPS, Size of the Window, Format of the Video. First of all I am making a .avi file which I am converting to MPEG2 format using ffmpeg command.

7) Jitter Graph :-
I am also finding the jitter in the system and finally plotting it on the graph using a .csv file which I am opening in an Excel Format. I am collecting the data of 1800 frames for finding the jitter in the system

8) Running at a higher frame :- I have also run the camera at a higher frame rate that is 10Hz

4 .Real time services used in this project :-

Sequencer :- I have used this service as the first basic service for this project. I am using this to generate a delay for either 1hz or 10Hz. the deadline of the sequencer is kept at 20 % more than the normal deadline just to be on a safer side.

There are 2 Hard real time services in this project

1) **Capturing the image 40msec**
2) **Saving and Compression of Image 65msec**

1) **Service 1 :-** This service comprises of real time capture of an image. I am used cvquerycapture and IplImage API of opencv to capture it in real time. The difference should be exactly 1sec between them. For this service , I am also making sure that 1 sec does not surpasses before 2nd frame comes in. I am also using cvputtext in this service to print the timestamp on the image.
2) **Service 2 :-** This service comprises of saving of an image and compression of the image. The ppm file which is of very high space is compressed to jpg image of a very little value

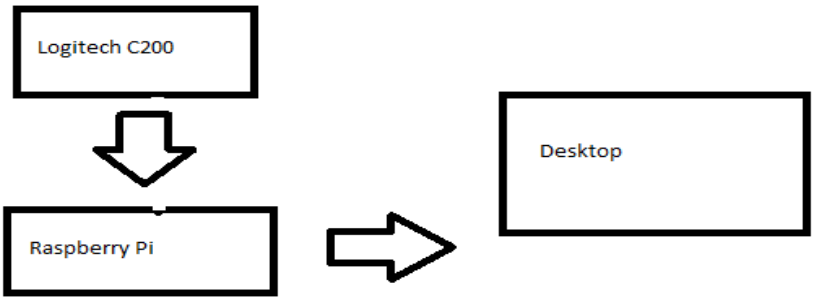
These 2 services will have a common deadline of 1 sec. So it is compulsory to complete the capture, saving and compression should be done in 1sec. Both the services won't be running at the same time. They are running in the loop.

For this continuous looping I am using semaphores and mutex locks in the process.
Semaphore :- A system of sending messages by holding the arms or two flags or poles in certain positions according to an alphabetic code.(Wikipedia)[2]

Mutex Locking and Unlocking Mechanism :- It is a kind of locking and unlocking mechanism used in linux so that whatever value is between is 2 brackets of lock and unlock wont change on its own and jitter will be reduced. While using this I had to make sure that I do not have deadlocks in the system. Deadlock happens when thread A and thread B (programs of a cpu) ,share the same resource and thereby preventing each other to take the resource.

Apart from these services I am also making the video of the images captured , stored as jpg. For that I am using the reference of **"ffmpeg -f image2 -i nameoftheimage.jpg raj.avi"** command
Moreover I am also converting the already made .avi file from avi format to mp4 format that is MPEG2 or MPEG4 format.
For this I am using the following command
"ffmpeg -i input.avi output.mp4"

5.HARDWARE DESIGN



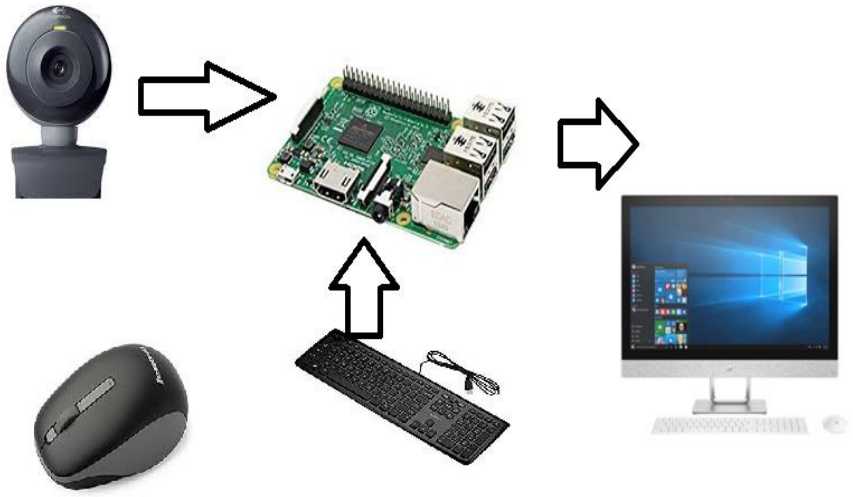


Fig :- 4 Hardware Block Diagram

Hardware Components Used are :-

1) Raspberry-pi
2) Desktop
3) Keyboard
4) Mouse
5) Logitech C200

For this project, my environment comprises of different hardware components made perfect sense for the project. these hardware runs smoothly so that my software code runs perfectly in this hardware. Since the deadline was very crucial in this and if my system comprised of errors or some slow errors in the system electrically or processor wise then I would have faced a lot of issues in debugging them and finally making the project almost impossible to run

6.Software Requirements :-

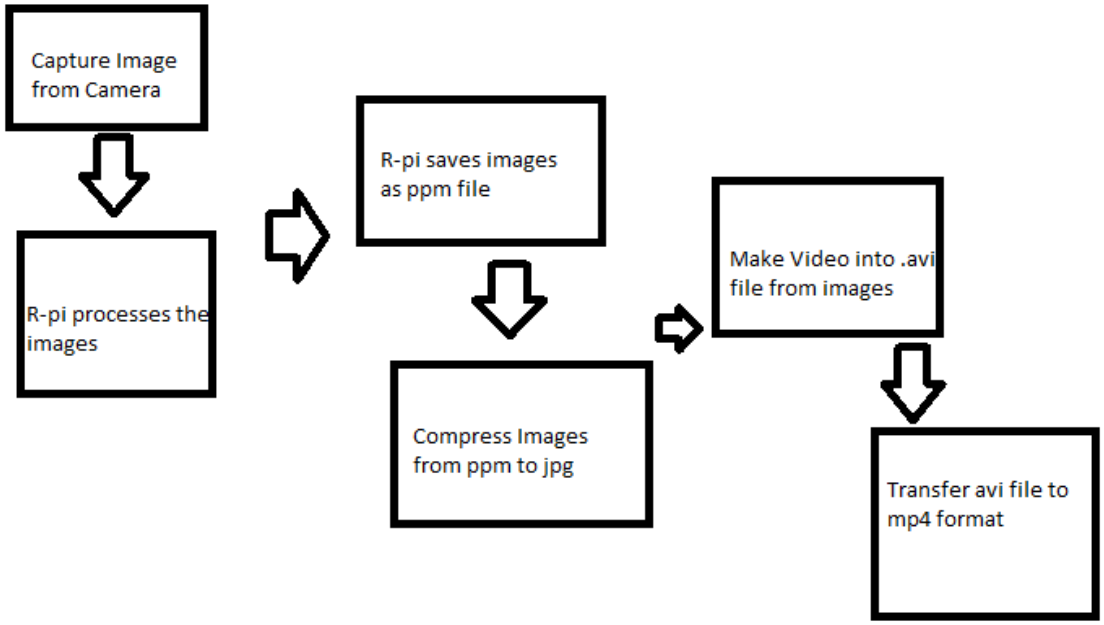


Fig 5 :- Software Block Diagram

I have explained my whole code with this software diagram
First priority I have given to sequencer. The sequencer is followed by service 1 and Service 1 is followed by service 2. This is the order of my priority.
Initially I decided to use 2 Services. 1st frame comprises of Capture of an image. For that I am using Logitech C200 camera , this camera generates the image into a raw image format. It captures an image and the digital signature is then transferred to the my processing system that is raspberry pi. When the image is being captured or the camera is turned on by the controller, it shows a yellow ambient light on the top of the camera. This indicated that the camera is ON.
After getting the images in the raw format, the controller processes these images which are coming at around 30-40 msec to the controller. These images are processed by the controller in nearly 60-70 msec time frame.
Moreover I am also putting a timestamp on the image. It will display the frame number and the time at which the image was taken this was an important step for the verification

of images for 1hz frequency. By subtracting the first and last frame timestamp I was able to know the exact time at which my capturing was happening.
These images are of very large size. The controller does not have a lot of memory in it. Thus we have to compress the images. I needed to store more than 8000 frames in total for 1Hz and 10Hz capture of frames. These images are compressed to jpg files which are hardly 9-10kb per frame which was nearly 1Mb for 1 ppm frame.
The process of compression and then saving the compressed and uncompressed frames were going on at the same time. I was defining a specified path where I wanted my images to be stored in the system.

After this, I took all the jpg images and using ffmpeg command I made a .avi file video. This video was running at 20 fps speed. After converting these images into video I used OMXPLAYER to run it on the linux terminal with the help of this command "omxplayer video.avi"

The requirement also suggested to provide the video in the high defination format of mp4.
Thus i converted the .avi file into mp4 using the ffmpeg command.

Opencv Commands & others I used in the Code :-

IplImage:-It is taken from Intel processing library and it is one of the very few commands used by Opencv for the image format.
Cvcapture:- It is used to in capturing of an image.
Syslog :- Using this command we can see whatever is running on the kernel end. To check the logs we have to print on the terminal "less /var/syslog".
cvSetCaptureProperty :- This command is used to set the parameters of the resolution Eg Width and Height.
cvNamedWindow :- This command is used to open the window on the terminal of the image capture.
Clock_gettime(CLOCK_REALTIME,&var) – This command is used to get the timestamp in real time.
Mat :- This command is used to convert normal frame (raw image) into a mat format. It can be done by using **cvarrToMat(frame).**
Imwrite :- This command I am using for saving a particular image to a particular file format. For example in this case from ppm to jpg.
sprintf:- It Composes a string with the same text that would be printed if *format* was used on printf, but instead of being printed, the content is stored as a *C string* in the buffer pointed by *str*. [4]

7.DEBUG METHOD AND PROBLEMS FACED

Problem Faced & Solved Method 1 :-

While I was making the services of this project, initially I used all the services in 1 service only. I was using the capture, saving, compression and formation of the video in 1 service only. This time I observed that I am getting a lot of jitter in the process because the video forming in a real time system was very had to produce without a jitter. So i seperated the services into 3 parts.
First service was capture , 2nd comprised of saving and compression of an image and 3rd one consisted of forming the video. Now for the 3rd Service when I use the 1Hz that is 1 sec/frame I was not having the issue on a larger scale. I was getting the jitter around 5-6 sec for 1800 frames. But for the 10Hz frequency signal, i.e. 10 Frames in 1 sec, here I was getting a lot of jitter in the compilation. At the end of 1800 frames I was getting the jitter of 70-80 sec. All these jitter formations are not at all acceptable since the requirement was to get for 1hz jitter in +-1sec. Thus I decided not to use video in any of the services and just run services 1,2 and excluded the service3 which comprised of the video.

Problem Faced & Solved Method 2 :- (Jitter Removal Techniques)

For 10Hz, I was initially getting a lot of jitter in the system. The jitter was around 200 sec which was very high from my expectations. A number of changes I did to improve that jitter bringing it to 0 are:-

1) Removing real time video capturing from the service 3:- After applying this my jitter reduced directly from 200 sec to 70 sec approximately. Compression of images and storing it in the video format was not happening very fast. It might be because the R-pi is not capable of working at such a high frame rate.
2) Removal of Timestamp from the image :- I was using “imshow “ command to open the window of the camera on the desktop and printing a time stamp on it. I was also printing the Image number on the window opened on the desktop. This was also creating jitter in my system. I was not sure if “cvputtext” command was generating the jitter or it was the opening of an window on the desktop or both. By removing these functions from the code I was again able to compensate my jitter.
3) On 10 Hz , use of extra elements in the system :- While the capturing of images were happening I was also working parallely on a different code and compiling it but not running it since my capturing was happening on the other terminal. This was also creating jitter in the system since I was not using all the 4 available cores for the code. Thus parallel processing was creating a little bit of jitter in my run time.

4) Use of resources in the system :- I have connected my keyboard, mouse, camera all together to R-pi. Converting the digital signatures of these appliances was generating a jitter in the system. While my camera was capturing some images the processor was also processing the data of the devices . Clashing of digital signatures might be generating the jitter while capturing the images.

Problem Faced & Solved Method 3:-

My raspberry-pi got crashed once may be due to overheating or corruption data and I was not able to login into my system. Even if I enter the correct password it was just rebooting itself. I tried taking the data from the SD Card of the R-pi by inserting it into my laptop but it asked me to erase all data first in order to write something on it. I finally replaced the R-pi and solved the problem

Problem Face But Not Solved :-

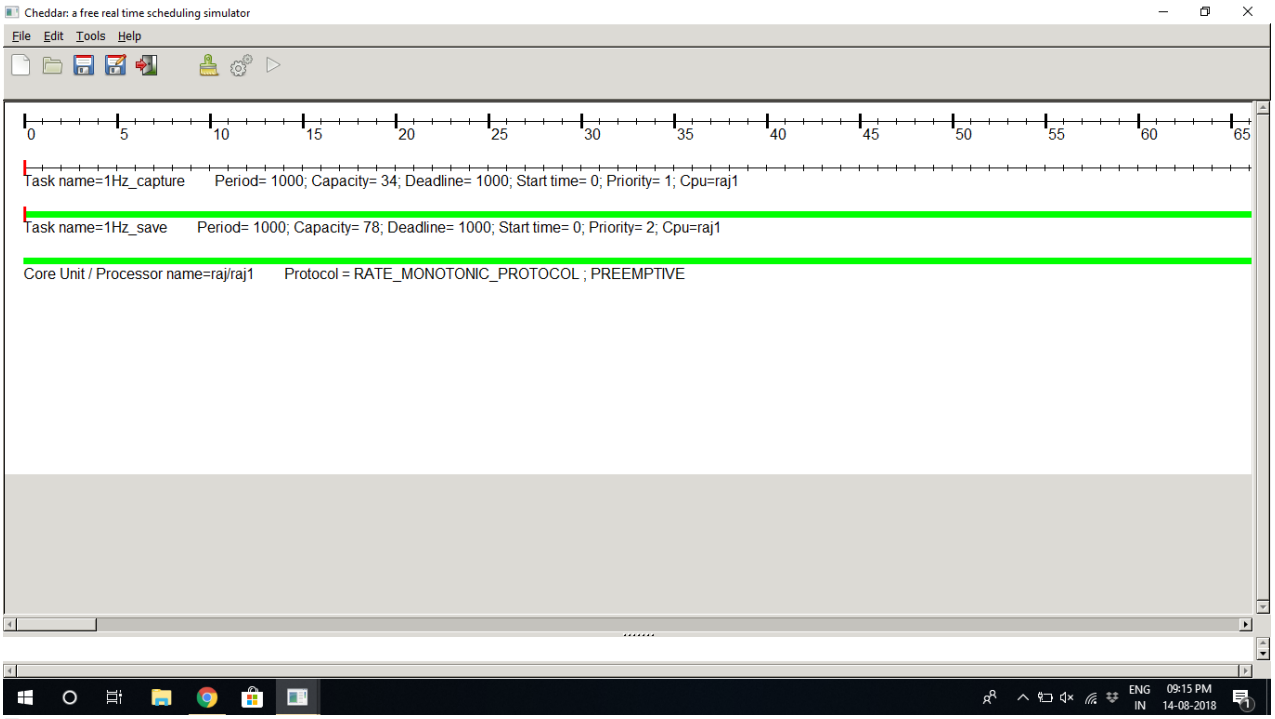
I tried doing the uname. For this I was copying the data from one image to another image. Before doing that I was using printing the P6 and resolution of the image in a dummy file and then printing all the data of that original image into the dummy image. After getting the data into the dummy image I thought it would open with the timestamp and User name. It happened also but then I was not able to open the ppm image. I was successful in printing the data but not successful in opening the image after printing the data on the ppm image. Just first image was opening up and all the other images were not opening up. I was finally not able to do it successfully.

Problem Face But Not Solved :-

I tried using the socket program which I took from professor Sam Siewert's website and also some help taken from StackOverFlow website, I managed to transfer 1 image from R-pi to Laptop using client server socket code. My client was my R-pi and my server was Laptop. The requirements suggested that I had to do a continuous download of images but I was able to do just 1. In the end I was not able to transfer the real time images from R-pi capturing and transferring it into the laptop.
Moroever I was able to transfer a bunch of images not in real time but through wired or wireless connection if both are connected on the same network for example common internet. This process was done very quickly but it is not real time.

Real Time Analysis :-

Rate Monotonic Scheduling Algorithm :-



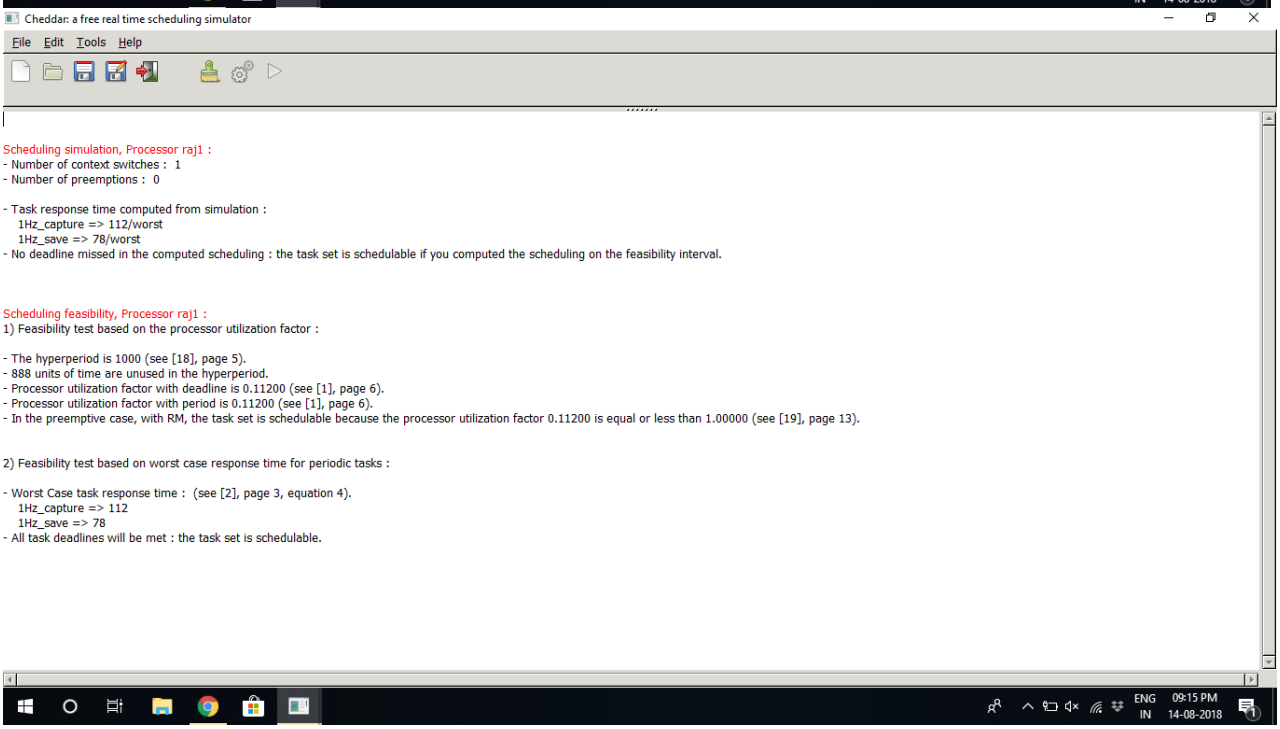


Fig 7:- RM Scheduling for 1Hz

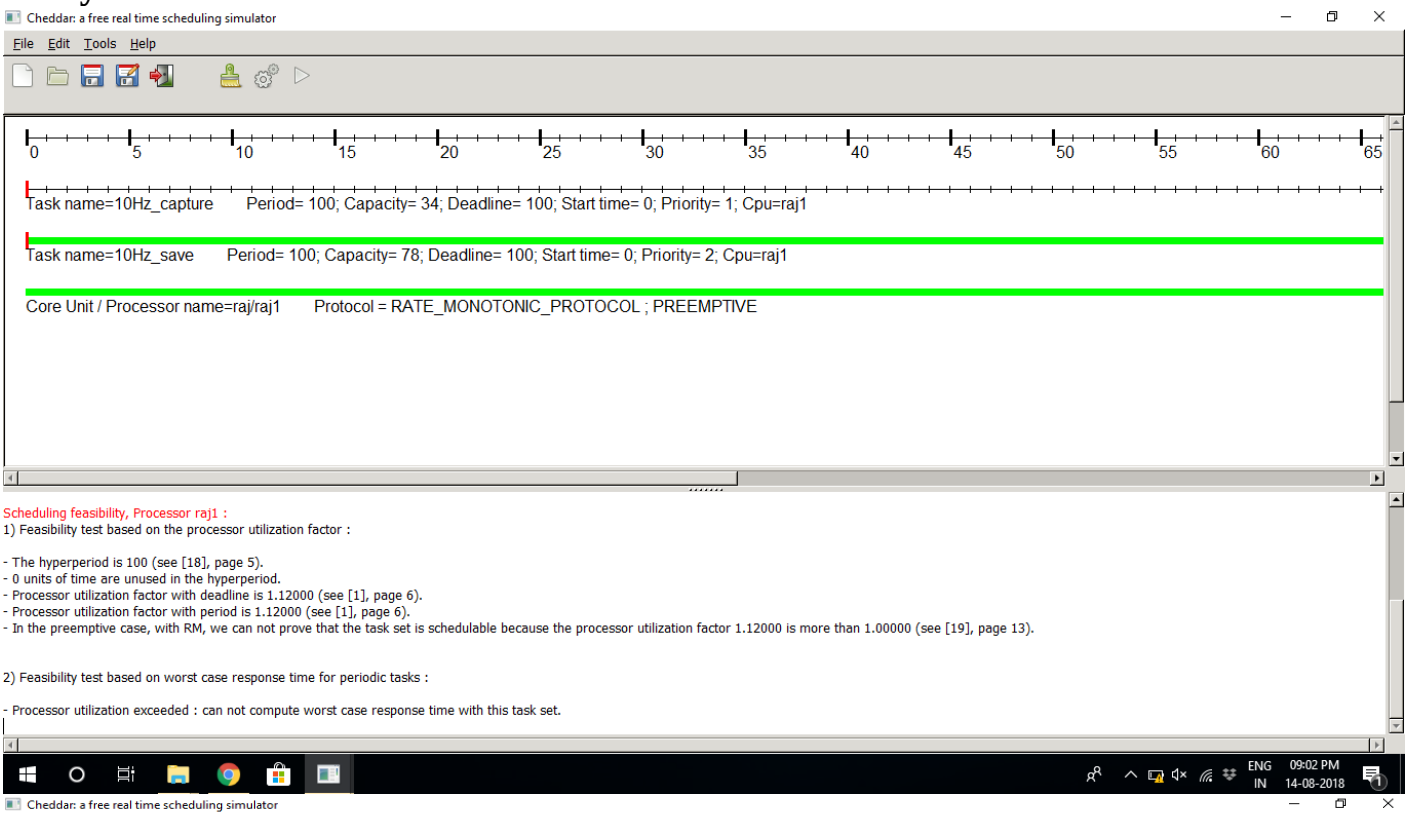
For the images shown above , I was getting the C1 ranging from 33-34 to 38-39msec. Thus I decided to take 34 as my C1. For Service 2 I was getting from 77-78 to 84-85 msec thus I decided to take 78msec. For this cheddar, I have used Rate Monotonic Anaylysis. Here the Deadline is calculated as whose C is less, that Service will have the highest priority. As shown in the image even if I do not set the priority here, then also the RM of cheddar is capable of generating that priority on its own.

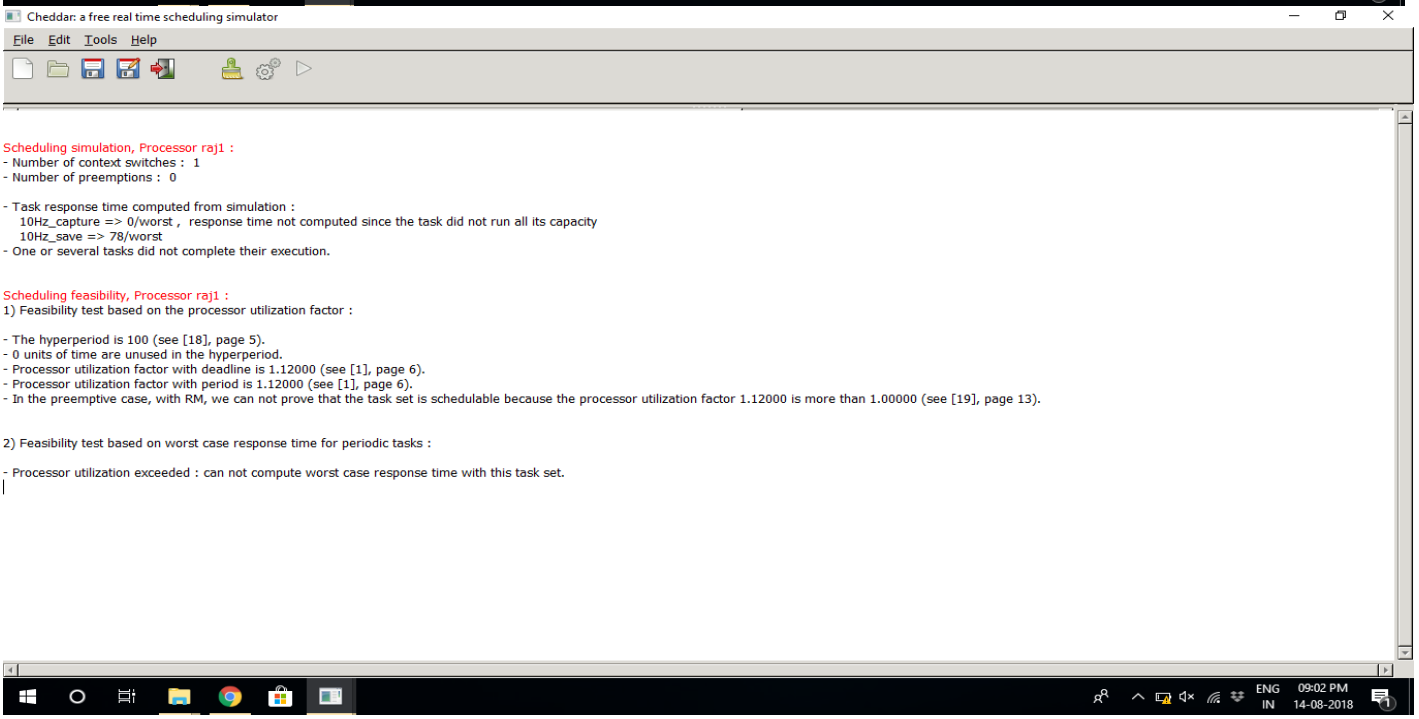
Here my C1=34, T1=1000
C2=78, T2 =1000

U=C1/T1 +C2/T2+C3/T3=0.034+0.078=0.112=11.2%

Since our deadline is 1000 msec and the services are getting completed in almost 112 msec, almost none of the deadlines will be missing and my system will be a very feasible one.

For 10hz
Analysis :-





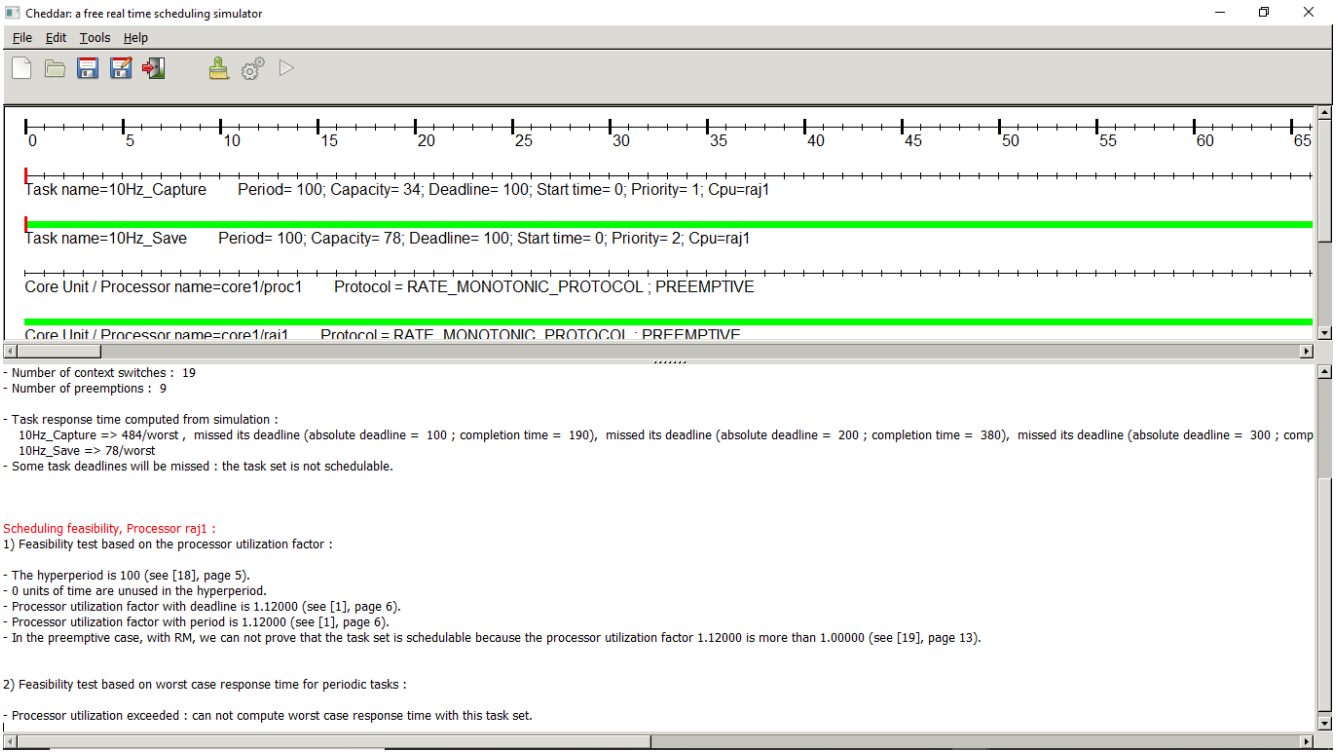


Fig 8 :- 10Hz Cheddar Analysis

For the images shown above , I was getting the C1 ranging from 33-34 to 38-39msec. Thus I decided to take 34 as my C1. For Service 2 I was getting from 77-78 to 84-85 msec thus I decided to take 78msec. For this cheddar, I have used Rate Monotonic Anaylysis. Here the Deadline is calculated as whose C is less, that Service will have the highest priority. As shown in the image even if I do not set the priority here, then also the RM of cheddar is capable of generating that priority on its own

C1=34 T1=100
C2=78 T2=100
U= C1/T1+C2/T2+C3/T3 = 0.38+0.78 = 1.12=112%

It clearly shows that the deadlines are missing and it not a feasible system.

9.Jitter Analysis :-

For 1Hz :-

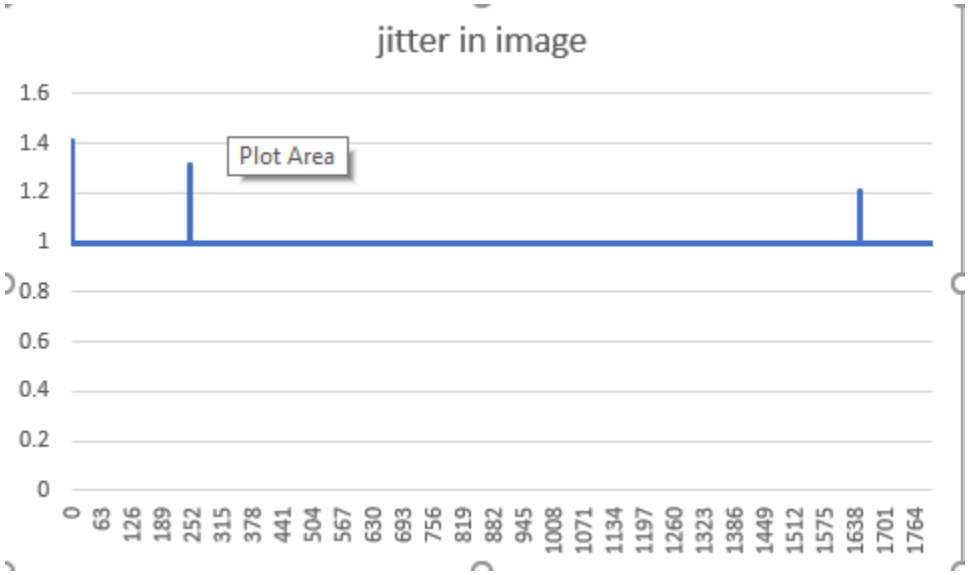


Fig 9 :- 1Hz Jitter Graph

As seen in the picture few of the deadlines are missing here. The commulative jitter of these deadlines is not more than 1 and thus our 2000 frames gets completed in the given time frame.

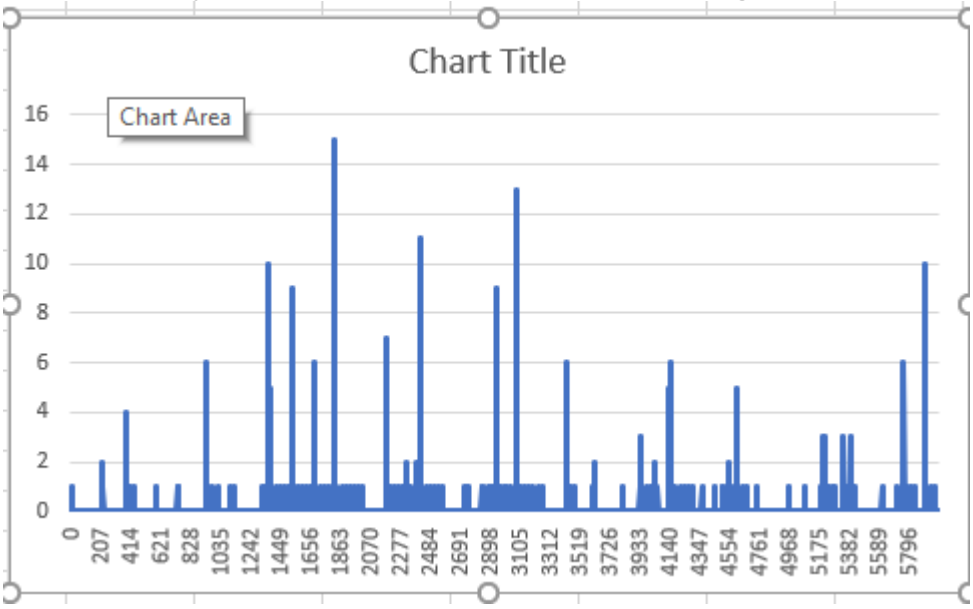
There are total 1800 frames in this. I have adjusted the scale accordingly so that I can see my deadline missing very clearly.

As seen in the image total jitter of 1800 frame 0.23s+0.21s = 0.44s. This is my total jitter for 1800 frames

Thus jitter per frame is 0.00024s

For 10Hz

For 10Hz, my almost all the deadlines are missing.



As seen in the image many deadlines are missing as cheddar has mentioned in the analysis. Since many deadlines are missing it is not feasible to use.

CONCLUSION :-

From this project I have concluded that, for 1Hz I am not getting a lot of jitter. My wcet is in the range of 30-40msec and 75-85msec. Thus my total time frame of 1000msec is almost never going to miss. As shown in the jitter plot diagram, only a few deadlines are missing and the total jitter is also very less. As shown in the cheddar anaylysis , it is clearly shown that my system is feasible.
For 10Hz, there is a lot of jitter in the system and the deadline are also missing as compared to 1Hz and thus I am not considering it as a feasible option. It is also because as seen in the cheddar anaylysis also the deadline are missing since the execution time is 112msec but the deadline 100msec so it is bound to miss the deadline.

This is happening because of the processor of the controller. The R-pi 3B+ is having very less space and the processor speed is also less. It is not able to do things parallely at a higher speed. If I was working on Jetson or any other device then my roblem for 10Hz could have been solved to some extent.

Thus using all my concepts of Real Time systems I was successful in doing the Time Lapse Project using R-pi and opencv with the help of Camera. I used cheddar and jitter graph for the anaylysis of the project. For 1Hz, I have captured 1800 frames and for 10Hz I have captured 6000 frames.

Proof of that :-

Fig :- 10 1st frame of 6000 frames for 10Hz

Fig 11:- 6000th Frame out of 6000 frames for 10 Hz

For 1Hz:-

Fig 12 :- 1st frame out of 1800 frames for 1hz

Fig 13:- 1800th frame out of 1800 frames for 1Hz

Future Objectives and Goals :-

1) Transfer data over the ethernet in real time to meet different objectives like monitoring some desktops from a common place.
2) Run the camera frequency to its maximum efficiency that is 15fps by using a more reliable and faster controller to avoid jitter in the system.

References and Attributions :-

I would like to thank Professor Sam Siewert for teaching me the concepts of Real Time system in Summer Semester. I would also like to thank the TA's of the subject, Mr Anirudh Tiwari and Mr Sandeep Raj Kumbargeri for their support throughout the semester. I would like to thank Vignesh Iyer for the analog watch for the timeframe capture and also for debugging of my code.

1) **https://stackoverflow.com/questions/2393345/how-to-append-text-to-a-text-file-in-c**
2) **https://www.google.com/search?q=timelapse+defination&rlz=1C1CHZL_enUS801US801&oq=timelapse+defination&aqs=chrome..69i57j0l5.3482j0j4&sourceid=chrome&ie=UTF-8**
3) **https://people.cs.clemson.edu/~dhouse/courses/405/notes/ppm-files.pdf**
4) **http://www.cplusplus.com/reference/cstdio/sprintf/**