University of Colorado Boulder

PROJECT VALIDATION AND SIMULATION

BY

SIDDHANT JAJOO
SATYA MEHTA
AYUSH DHOOT

UNDER THE GUIDANCE OF:

PROFESSOR TIMOTHY SCHERR,
UNIVERSITY OF COLORADO BOULDER

5/4/2019

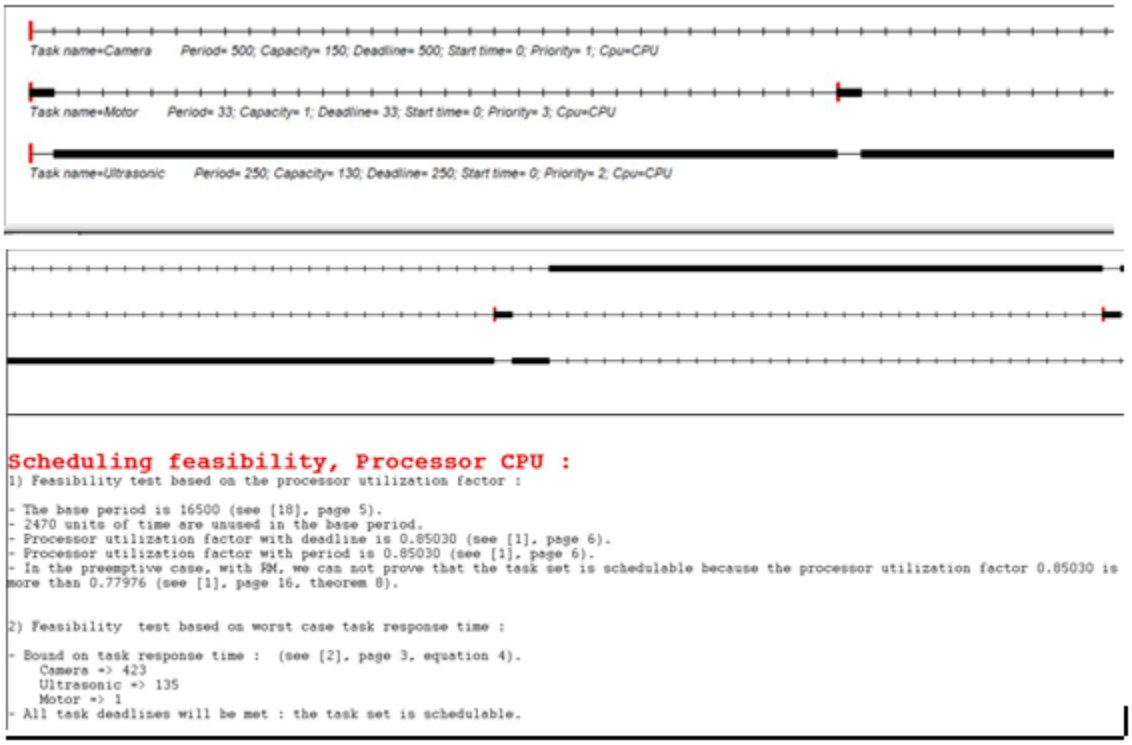
BOARD USED: RASPBERRY PI.

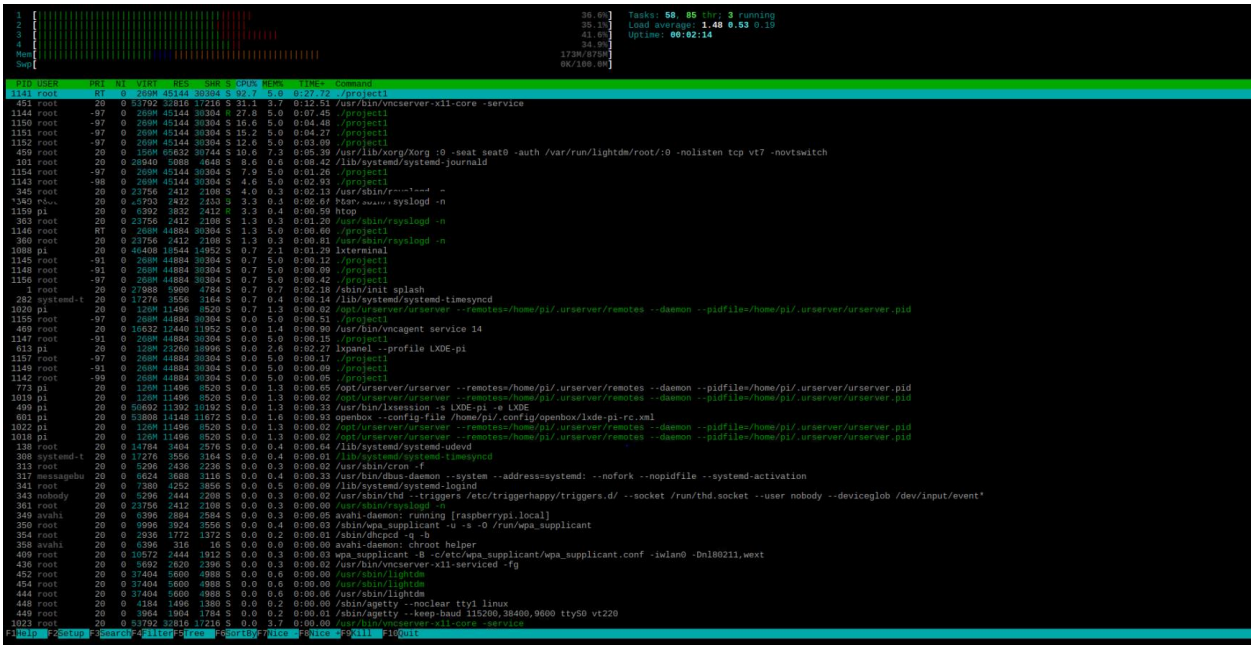
TABLE OF CONTENTS

QUESTION 1

CHEDDAR SIMULATION AND WORST CASE ANALYSIS

The image below is the Cheddar image of the schedule with the values of DCT specified in the table above. It can be seen that the camera task appears at the last after the Ultrasonic task has finished its execution since the camera task has the lowest priority. It can be observed here that the Utilization factor obtained here is around 85%.





The above image shows the CPU loading due to the code that has been run on RPI. The light blue row at the top shows the CPU loading value as 92%. It can be observed that the there is a difference of about 7% between the utilization factor calculated and the one above. This might be due to the background processes or the linux kernel too.

Hand written schedule was not possible because the LCM is too large.

SCHEDULING POINT AND COMPLETION TEST

Scheduling Test and completion test provides Necessary and Sufficient feasibility test for RM Policy. Lehozcky, Sha and Ding introduced this test which is built upon the RM Lub policy.

$$\forall i, 1 \le i \le n, \min \sum_{j=1}^{i} C_j \left\lceil \frac{(l)T_k}{T_j} \right\rceil \le (l)T_k$$

$$(k,l) \in R_i$$

$$R_i = \left\{ (k,l) \middle| 1 \le k \le i, l = 1, \ldots, \left\lfloor \frac{T_i}{T_k} \right\rfloor \right\}$$

If the sum of these times are smaller than *l* period of Sk service than the services are feasible.

For completion test feasibility determination it is required that the a(n) is required to be less than or equal to the Sn.

$$\forall i, 1 \le i \le n, \min \sum_{j=1}^{i} C_j \left\lceil \frac{(l)T_k}{T_j} \right\rceil \le (l)T_k$$

$$(k,l) \in R_i$$

$$R_i = \left\{ (k,l) \middle| 1 \le k \le i, l = 1, \ldots, \left\lfloor \frac{T_i}{T_k} \right\rfloor \right\}$$

For testing purpose we implemented and learnt in Exercise 2 about how to implement Completion Test and Scheduling Point Test in C code. We used that implementation to test our projects real-time requirements. We analyzed that the Completion feasibility test and Scheduling Point test shows that our services are feasible.

```
satya@satya:~/Satya/Spring19/RTES/EX/EX-2$ ./feasibility_tests
******** Completion Test Feasibility Example
Services U=0.85 (C1=1, C2=130, C3=150; T1=33, T2=250, T3=500; T=D): FEASIBLE


******** Scheduling Point Feasibility Example
Services U=0.85 (C1=1, C2=130, C3=150; T1=33, T2=250, T3=500; T=D): FEASIBLE
satya@satya:~/Satya/Spring19/RTES/EX/EX-2$
```

SAFETY MARGIN

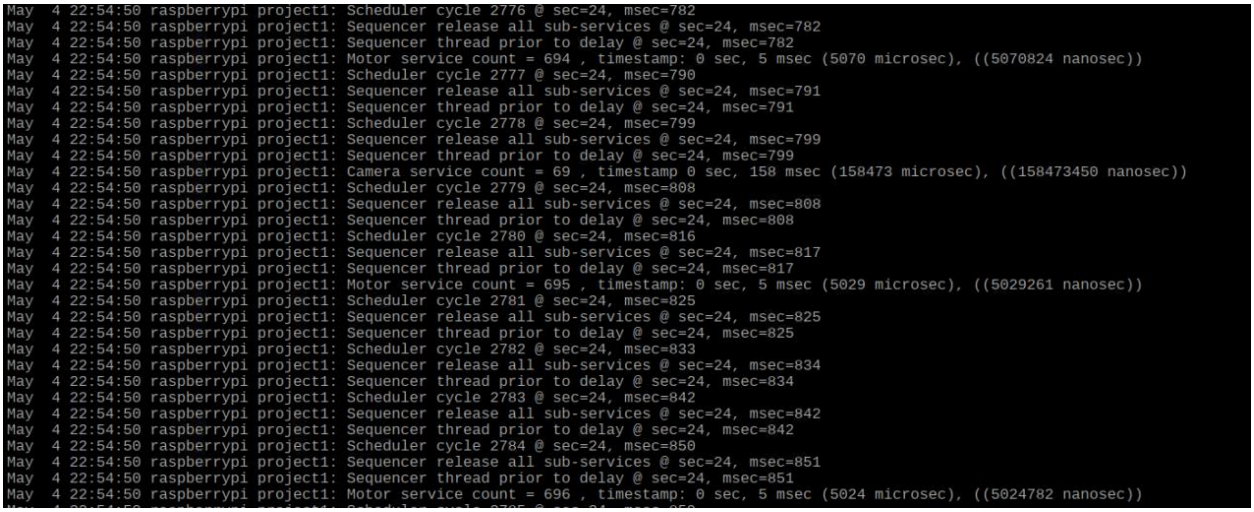
Safety margin should be 30% but as our utilization is 85% we have only 15% margin available. We tried reducing the utilization by trying to decrease the frequency of the services but as our prototype relates to the automobile application it was required to keep the camera frequency at 3-4 Hz so that the sign gets detected when the bot is moving.

MARGIN OF ERROR

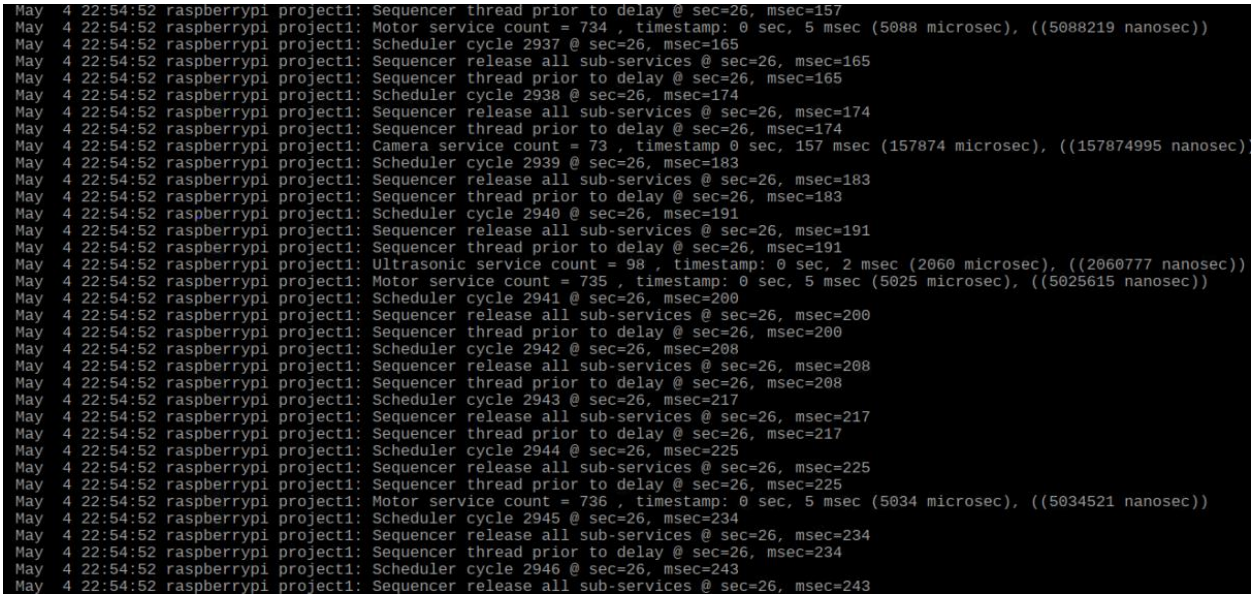
TASK	WCET (ms)	Ci (ms)	ERROR MARGIN WCET - Ci
Motor	1	0.01	+ve
Ultrasonic Sensor	130	10	+ve
Pi Camera	150	120	+ve

The above table shows that the margin is always positive signifying that the Overload situation never occurs. The Ci is always less than WCET.

QUESTION 2

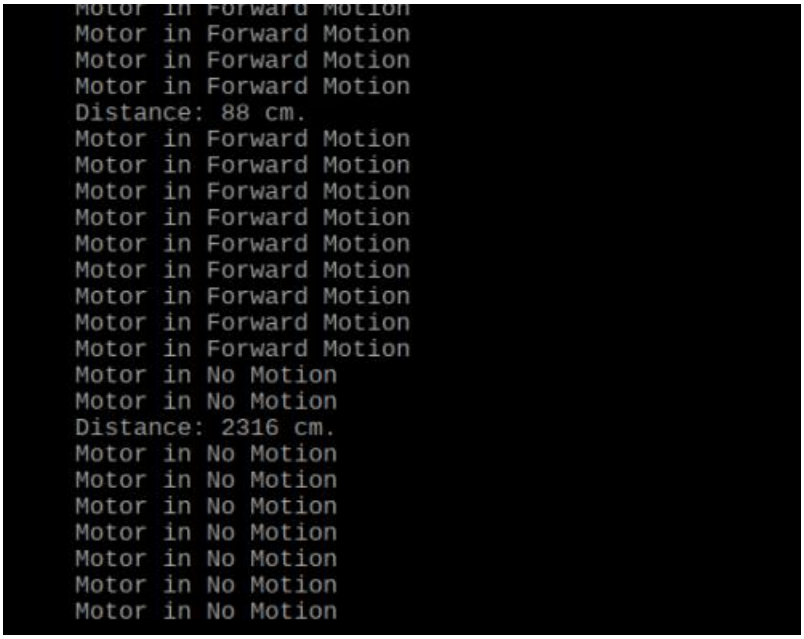


Syslog screenshot

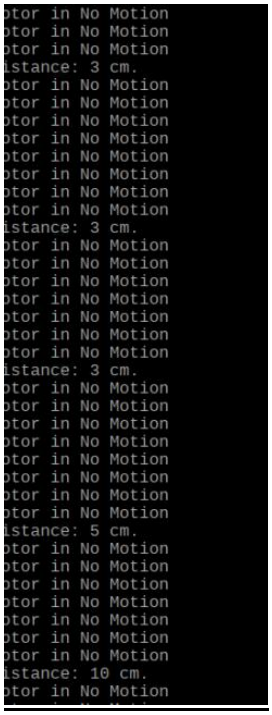


Syslog screenshot

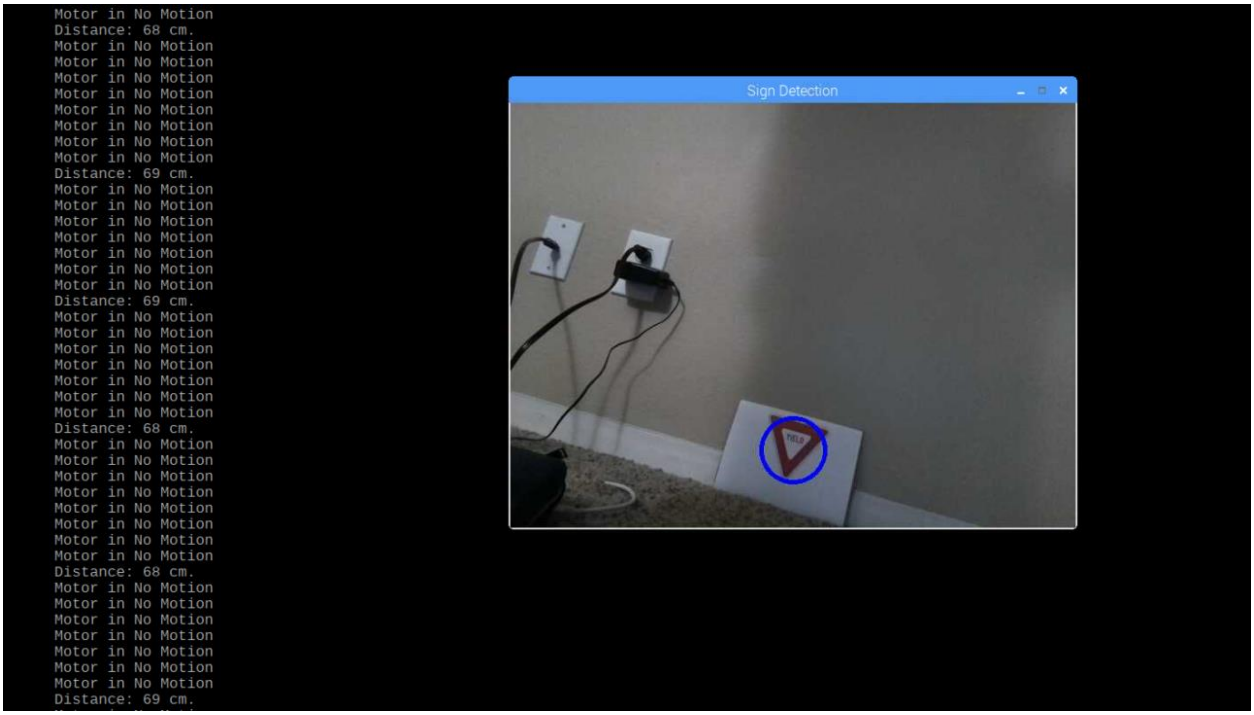
Test Cases:



Forward motion



Ultrasonic Detect



Sign Detection

The working of the entire code and its implementation was demonstrated in class. Syslog Screenshots have been attached as proof of concept code.

The Demo Video can be found over here:
https://www.youtube.com/watch?v=5pxvPxIWPSY&feature=youtu.be