REPORT ON
EXERCISE 6

By

SIDDHANT JAJOO
AYUSH DHOOT
SATYA MEHTA

University of Colorado
Boulder

UNDER THE GUIDANCE OF:

PROFESSOR TIMOTHY SCHERR,
UNIVERSITY OF COLORADO BOULDER

DATED:04/23/2019.

Question 3:

The Project is aimed at designing a bot that captures certain symbols and subsequently carries out certain actions based on the symbols. The project design includes motors, IR sensor (GP2D12 - Distance Sensor) and Pi-Camera Module V2. The board used for processing is the Raspberry Pi.
The basic idea is that the bot runs in the direction specified by the symbols. The Camera would search for these signs in its frame:

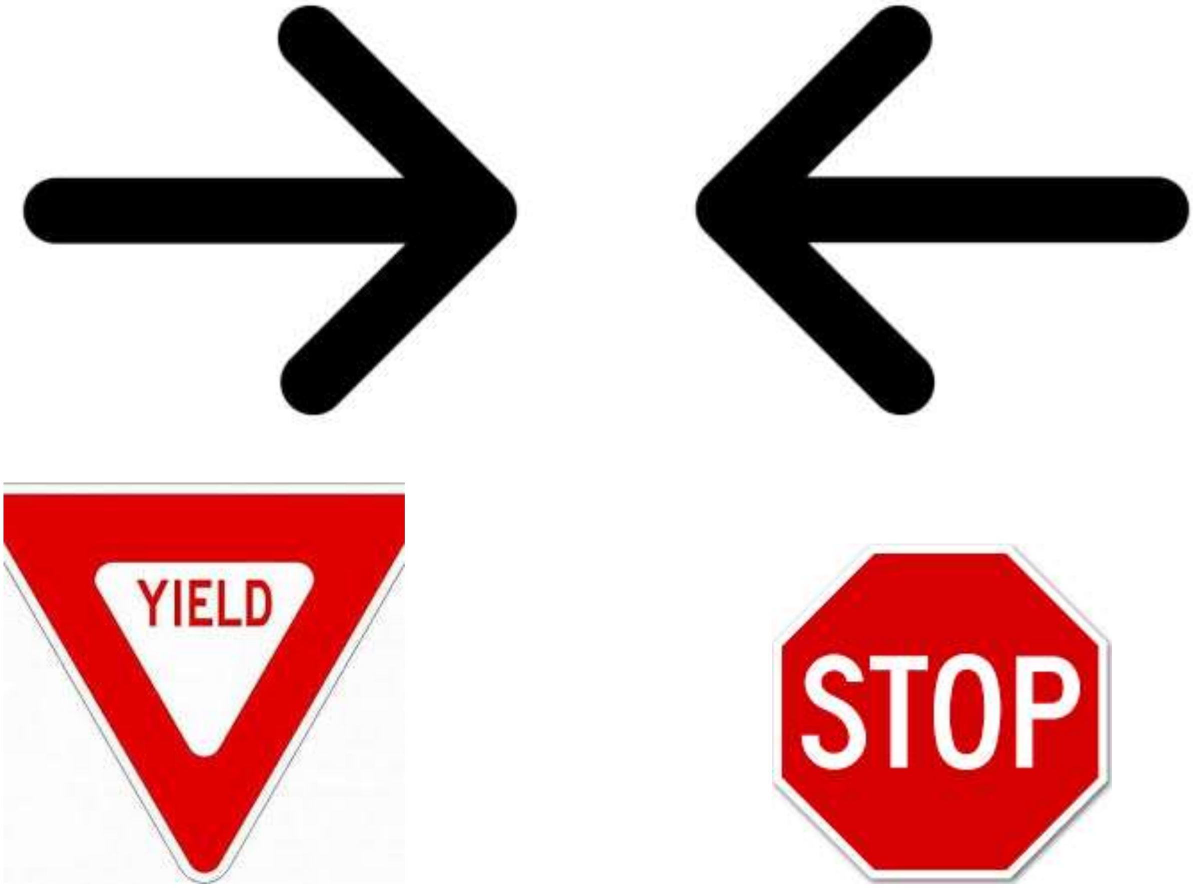


Figure 1: Road Signs

When the Camera detects such signs, the motors stop, turns and moves in the direction of the specified sign. The system would take care of 3 such kinds of signs - Right, Left and Forward. In case of an obstacle in the path of the bot, the care would be taken that it does not crash onto the object. For this purpose, an IR Sensor would be attached onto the bot which would constantly detect for any obstacles in its path. If an obstacle is detected, it would immediately stop and resume its operation if the object is removed from its path.

The Scheduling policy that would be used for this project on Linux is SCHED_FIFO. The priorities would be assigned on the frequency of operation of different tasks, thus following the Rate Monotonic Policy.
The different tasks are:

1. Motor Task - Second Highest Priority.
2. Camera Task - Third Highest Priority.
3. IR Sensor Task - Lowest Priority.
4. Scheduler - Highest Priority.

In order to determine which task runs at a particular point of time, a scheduler would be designed which would post the semaphores for the respective tasks. All the tasks would be blocked on their respective semaphores which would be released in the scheduler.

This project is being done in a group of 3 people. Different tasks are being designed by different individuals:

1. Motor Task - Satya Mehta.
2. Camera Task - Ayush Dhoot.
3. IR Sensor Task - Siddhant Jajoo.
4. Scheduler Task - Satya, Siddhant, Ayush.

These different tasks would be serviced based on Rate Monotonic Policy. RM policy was chosen as opposed to any dynamic deadline policy due to the fact that all services would be independent of its execution of each other. Suppose if 1 of the service fails to meet its deadline, there would not be any cascading effect on the other tasks. The other higher priority tasks would preempt the lower ones to carry out their functionality. In addition to this, debugging is easier for a deterministic schedule.

The IR Sensor Task

This task will check any object, if detected in its range as and when deployed by the scheduler. If an object is detected, the task will raise the obstacle flag which would be checked in the motor task. If no object is detected the flag is set low.

The Camera Task

This task will check any signs - Left, Right and Forward, if detected in its range as and when deployed by the scheduler. If any sign is detected, the task will raise the corresponding motor direction flag which would be checked in the motor task. If no object is detected the flags are set low.

The Motor Task

This task will check for all the flags set by the other two tasks and take the corresponding actions to move the bot forward, right, left or stop.

The Scheduler Task

This task will periodically be invoked to check which task should be deployed next. The different tasks would be deployed by releasing the semaphores based on the frequencies of the different services.

REFERENCES:

- http://www.cse.unt.edu/~rakl/KAH08.pdf
- https://users.ece.cmu.edu/~koopman/des_s99/real_time/
- https://github.com/opencv/opencv/blob/master/samples/cpp/facedetect.cpp
- http://wiringpi.com/
- https://autottblog.wordpress.com/programming-the-car/opencv/

Individual Roles:

Satya Mehta (Motor Task) :-

- Interface motors to Raspberry Pi through motor drivers.
- Study datasheet of motor driver L293D.
- Write code for motors, checking flags, turn on gpio to drive the motors, etc.
- Contribute in making scheduler, deciding parameters like T, D,C.
- Generate PWM on gpio of Raspberry Pi.

Ayush Dhoot (Camera Task) :-

- Interface Pi Camera Module v2 with Raspberry Pi.
- Identify road signs (stop, yield, traffic, directions) from the live feed using OpenCV and Haar Cascade classifiers.
- Modify the state of the motor according to the sign detected.
- Contribute in calculating WCET using timestamps or other profiling tool like KernelShark.
- Determine the priorities according to Rate Monotonic Policy and create threads for each task in the main loop.

Siddhant Jajoo (IR Sensor Task) :-

- Interface IR distance sensor GP2D120 to the RPi
- Study datasheet of GP2D120.
- Initialize and use ADC peripheral on Rpi.
- Write code for this task which will check adc data, set/clear obstacle flag, etc.
- Contribute in calculating WCET, schedule the task sets, etc.

Question 4:

Major requirements: -

1. The scheduler should run at a decided frequency and it should schedule as per the theoretical schedule we obtained using RM analysis.
2. The system should detect signs and move the car as per the sign detected.
3. The system should detect obstacles and stop the motor.
4. If there is an obstacle and sign both detected than the system should give the priority to the obstacle and stop the motor.
5. The IR sensor task must be accurate enough to detect the obstacle at proper distance which can give the motor task enough time to stop the motor.

Question 5:

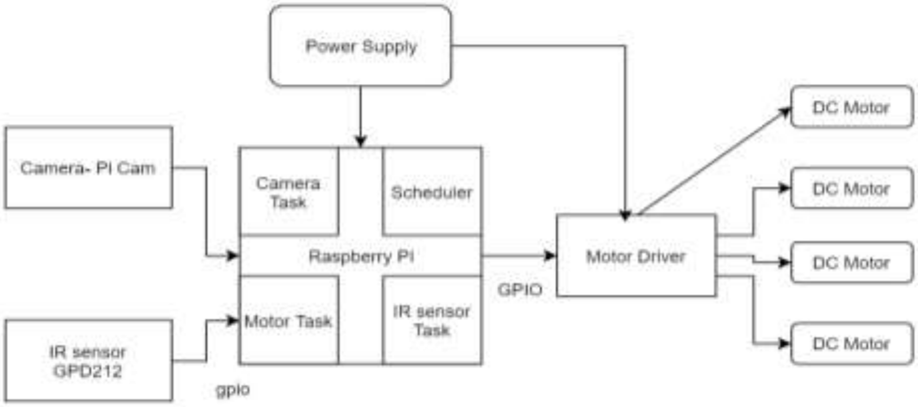


Figure 2: Block Diagram

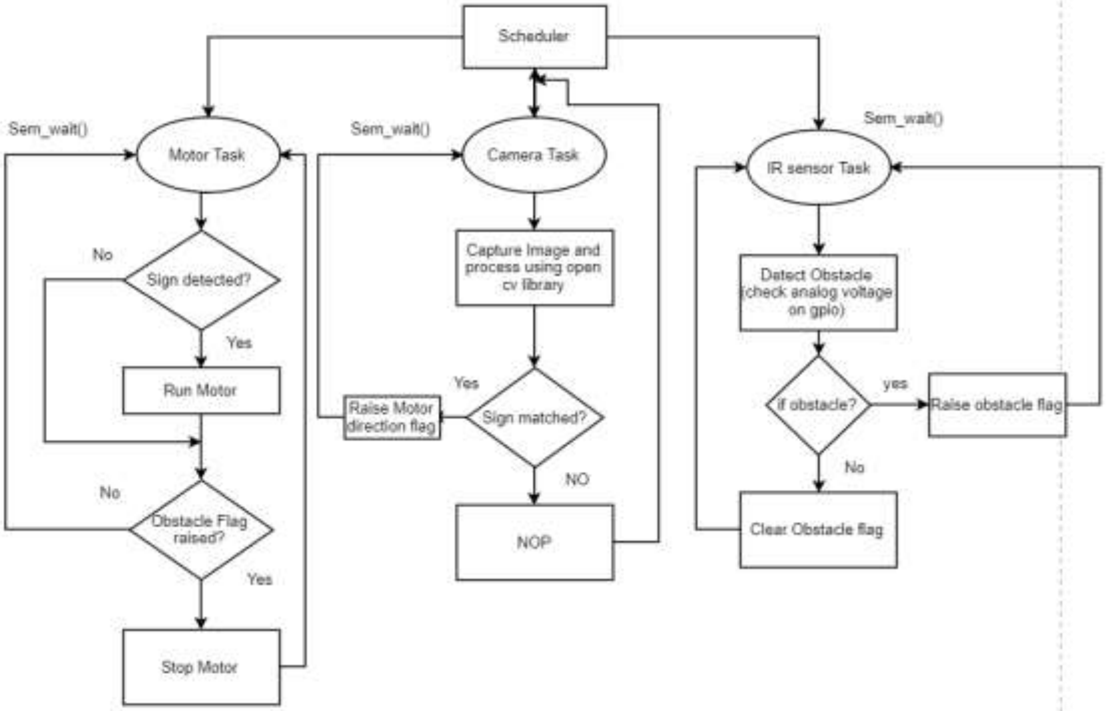


Figure 3: Flowchart

Scheduler

This software flow chart represents scheduler task incorporated in the project.

The tasks are ran at particular frequency to make the set schedulable

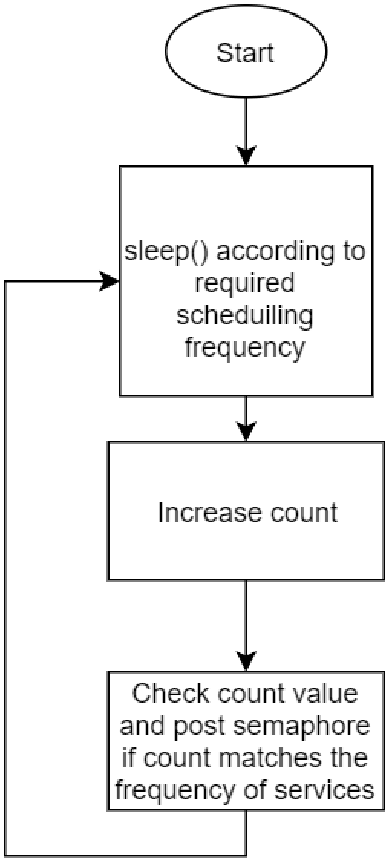

Start
sleep() according to required scheduiling frequency
Increase count
Check count value and post semaphore if count matches the frequency of services

Figure 4: Scheduler Flowchart

Question 6:

The project has three services Camera, IR and Motor. The camera service will wait for the semaphore to be posted by the scheduler. Once it acquires the semaphore it will capture images from the live feed through Pi Cam and detect the sign using cascade.load() and cascade.open() opencv apis. If the sign or the frame matches the xml file of the sign we stored in the Rpi it will set the appropriate flag. The IR sensors service will fetch the data from the ADC channel which gives the measurement of the distance from any object which will be helpful in detecting the obstacle. The flag will be raised if we find any such obstacle. The motor service will constantly check the flags which will be used to drive the motor depending on the value of the flag. Every services will wait on its semaphore and execute for a unit of time when provided by the scheduler

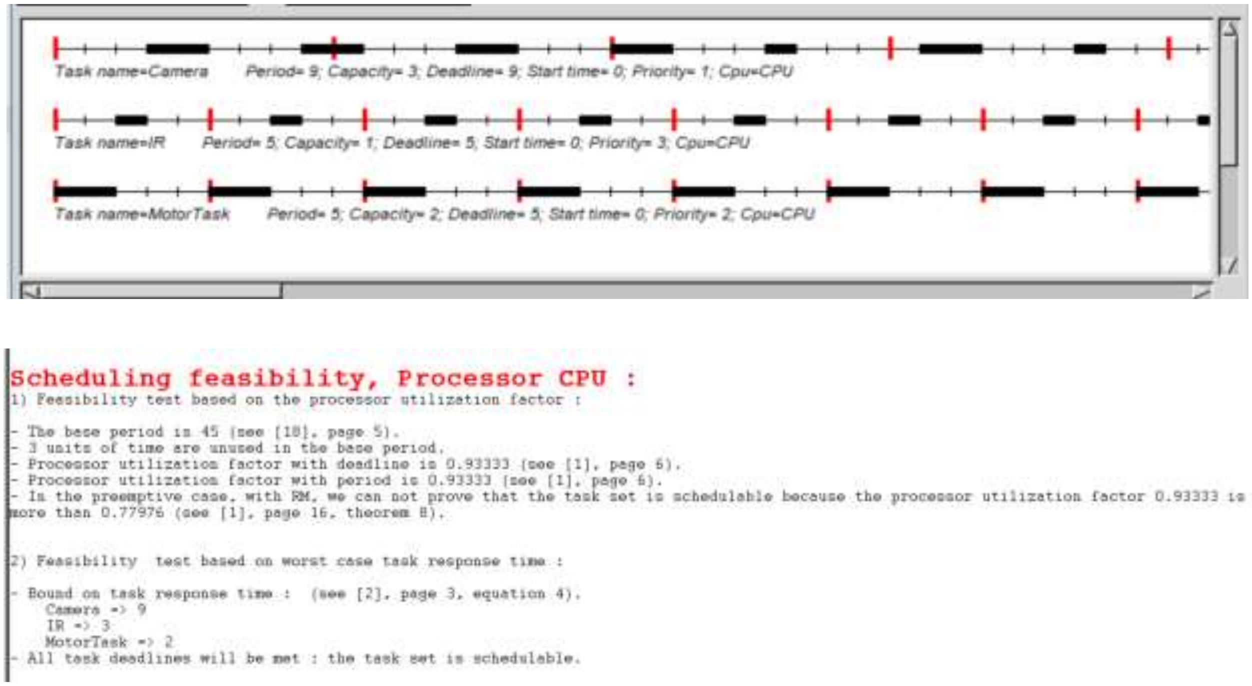
We presumed that the camera task is going to take more time as it has some processing which is required to be carried out which includes Opencv function calls. Hence, we took WCET of camera task to 3. Motor task will have some flag checking and command the motor which bounds to have more instructions than the IR task. Hence, we assumed the WCET of the motor task to be 1. IR task just reads the gpio and set/reset the flag hence, we assumed to have least computation time.

By running different combinations of time, deadline we came upto below solution which is feasible and quite schedulable. Currently, time has been taken as units and depending on the specific time of the scheduler all the times will be scaled during the project implementation.

	Motor	IR	Camera
C	2	1	3
T	5	5	9
D	5	5	9

Table 1 : D,C and T table

All measurements are related to unit of time.



```
Task name=Camera      Period= 9; Capacity= 3; Deadline= 9; Start time= 0; Priority= 1; Cpu=CPU
Task name=IR      Period= 5; Capacity= 1; Deadline= 5; Start time= 0; Priority= 3; Cpu=CPU
Task name=MotorTask      Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=CPU
```

```
Scheduling feasibility, Processor CPU :
1) Feasibility test based on the processor utilization factor :

- The base period is 45 (see [18], page 5).
- 3 units of time are unused in the base period.
- Processor utilization factor with deadline is 0.93333 (see [1], page 6).
- Processor utilization factor with period is 0.93333 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 0.93333 is
more than 0.77976 (see [1], page 16, theorem 8).


2) Feasibility  test based on worst case task response time :

- Bound on task response time :  (see [2], page 3, equation 4).
    Camera => 9
    IR => 3
    MotorTask => 2
- All task deadlines will be met : the task set is schedulable.
```

Figure 5: Cheddar for above D,C and T table

Priorities: -

- We have three services where we have assigned the Motor has highest priority as it must be stopped any time when obstacle is detected.
- IR sensor is given second priority as detection of obstacle should have higher priority than sign detection.
- Camera has been given lowest priority.

As we knew that according to RM Policy, highest priority is given to the least occurring services, we decided the deadlines and period which came feasible on cheddar using the priority we already knew.