CS5272 – EMBEDDED SOFTWARE DESIGN

GROUP 05 PROJECT REPORT

Team Members:

Darren-Gavin Ho Weiliang (A0087858E)

Yeo Kheng Meng (A0081007U)

Date: April 22, 2015

Tasks Implemented

Task	Description	Priority
Headlights	Changes headlights' light intensity based on a time specific value processed from the potentiometer's value.	1
ADC Initiator	Gets emitted value from the slider, potentiometer, door and engine button and the custom force sensor; issues interrupt and sends the value to the respective mailbox. Its period is set to 100 ms.	2
Mailbox Retriever	Retrieves the respective value(s) from the Mailbox.	3
Speed	Recalculates new speed based on the resultant force on the car. This resultant force calculated using the Drag Force equation, the slider and the force sensor's value.	4
Interior Lights	Regulates the turning on, off and dimming of lights when the door is opened or closed.	5
Engine	Changes the state of the engine (depending on the input from the engine button) and sends the corresponding status message(s) to the LCD for displaying.	6
Door	Changes the state of the door based on the current state of the door and engine. It also triggers the alarm if necessary when the door is open while the engine is turned on	7
LCD	Displays the respective values for speed, door status and the potentiometer values periodically. Its period is set to 100 ms.	8
Alarm	Changes the state of the alarm lights where necessary based on the states of the door and the engine.	9

Assumptions

Some assumptions which we have made for this project are:

1. When the interior lights are turned on, any changes to the ambience or the potentiometer value will not change the state of the lights.
2. We have included the phenomenon of no light as a level of brightness as well.

Drag Force Equation

In this project, we used the Drag Force equation to regulate the effects of the car's acceleration.

$$F_D = \frac{1}{2}\rho u^2 C_D A,$$

where F_D is the drag force experienced by the car as its velocity changes

We substituted the data of a 1992 Toyota Camry into the following variables:

ρ: Density of air where its average is 1.23 kg/m^3. However, we used a larger value of 6 to compensate for the lack of a realistic form of friction with the road.

u: Velocity of the car.

C_D: Drag coefficient of the car which is 0.31.

A: Front surface area of the car.

The engine force of the car is proportionate to the slider value. The resultant force will be the summation of the engine force and the drag force. Using *F = ma*, where *m* = 1300 kg, we can determine the acceleration.

Task Priority and Period Justifications

The Headlights task was given the highest priority because the PWM operation requires an instantaneous access to the task so that the dimming effect will be as fluid as possible.

The next task that should take second highest priority would be the Speed task. However since it depends on the ADC Initiator and Mailbox Receiver tasks for its values, its priority will be lesser than these two tasks.

The Interior Lights task will take the next highest priority as it is similar to the Headlights tasks but it is only significant when the engine is off unlike the Headlights task where the engine must be on.

The Engine and Door tasks will be have the next two respective highest priority as it depends on the existence of user inputs (such as pressing either the Engine or Door buttons). Therefore, compared to the previous tasks, these tasks will have a lower priority. Furthermore, the Door task will have a lower priority than the Engine task as it depends on it for its state change.

Lastly, the Alarm task will have the lowest priority as it is a dynamically created task only when both the engine and door are active. Hence, its occurrence as compared to the other tasks would be significantly lower.

In addition, the period of the ADC Initiator and LCD tasks is set to 100 ms because a higher frequency update is relatively indistinguishable to the human eye.

Finite State Machines

Legend

1. Variables

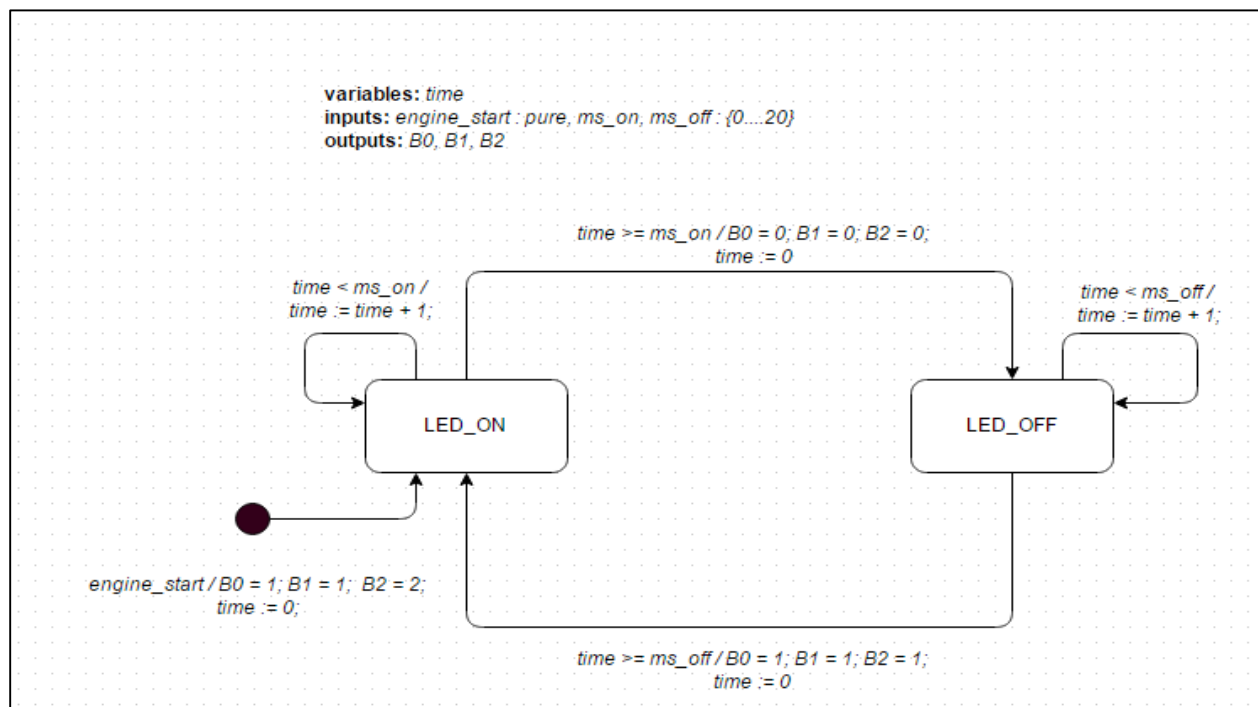|	
Current Speed Value	*current_speed*
Counter	*time*
Time specific value for potentiometer that indicates dark ambience.	*ms_on*
Time specific value for potentiometer that indicates bright ambience.	*ms_off*
10 seconds timer	*timer_10*
20 seconds timer	*timer_20*
5 seconds timer	*timer_5*

2. Input(s)

Slider Value	*slider_value*
Open Door	*door_open*
Close Door	*door_close*
Start Engine	*engine_start*
Stop Engine	*engine_stop*
Car is moving	*moving*
Press Brake	*brake_pressed*
Ambient Light Value	*amb_light*

3. Output(s)

Headlights	*B0, B1, B2*
Turn on interior lights	*lights_on*
Turn off interior lights	*lights_off*
Dimming of interior lights	*lights_dim*
Turn on alarm	*alarm_on*
Turn off alarm	*alarm_off*
Open Door	*open_door*
Close Door	*close_door*

Headlights Module

variables: *time*
inputs: *engine_start : pure, ms_on, ms_off : {0....20}*
outputs: *B0, B1, B2*

time >= ms_on / B0 = 0; B1 = 0; B2 = 0;
time := 0

time < ms_on /
time := time + 1;

time < ms_off /
time := time + 1;

LED_ON

LED_OFF

engine_start / B0 = 1; B1 = 1; B2 = 2;
time := 0;

time >= ms_off / B0 = 1; B1 = 1; B2 = 1;
time := 0

In this module, the state of the headlights (turned on or dimmed down) is dependent on the time specific value (*ms_on* or *ms_off*) processed from the potentiometer value. If these values are satisfied, the LED of the headlights will turn on or off accordingly creating the respective dimming through instantaneous flickering of the light.

Doors Module

inputs : *moving, door_open, door_close : pure*
outputs : *open_door, close_door*

!moving && door_open / open_door

DOOR_CLOSE

DOOR_OPEN

door_close / close_door
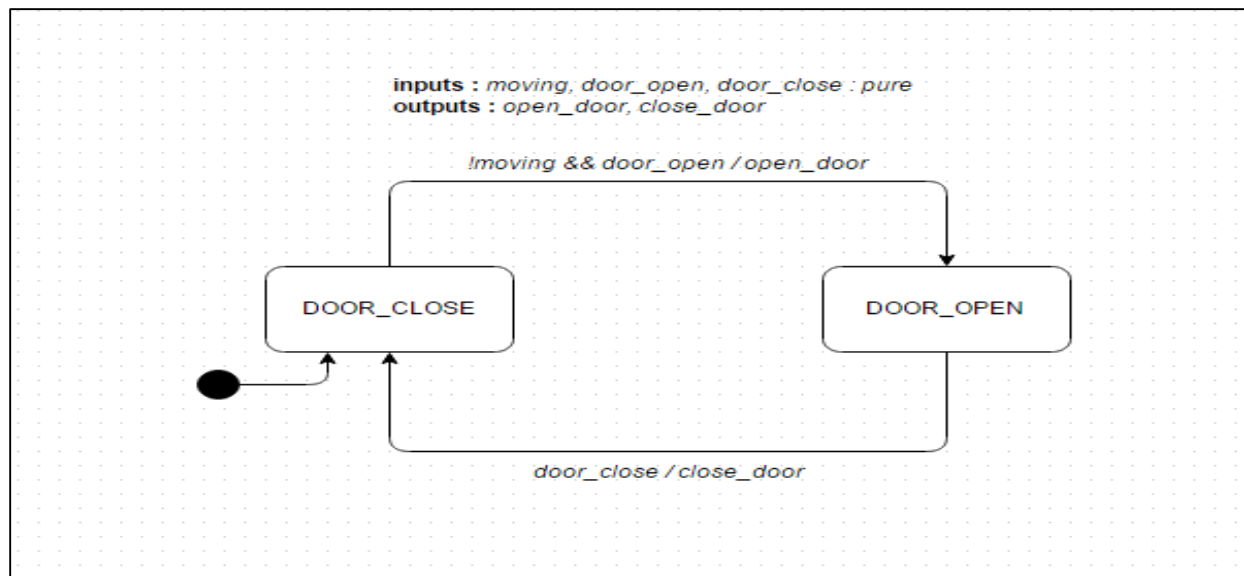
In this module, the state of the door is changed based on the state of the car or the engine and the current state of the door. The final state of the door (open or close) is required to determine whether the alarm should be raised (as seen in the Alarm module of this system).
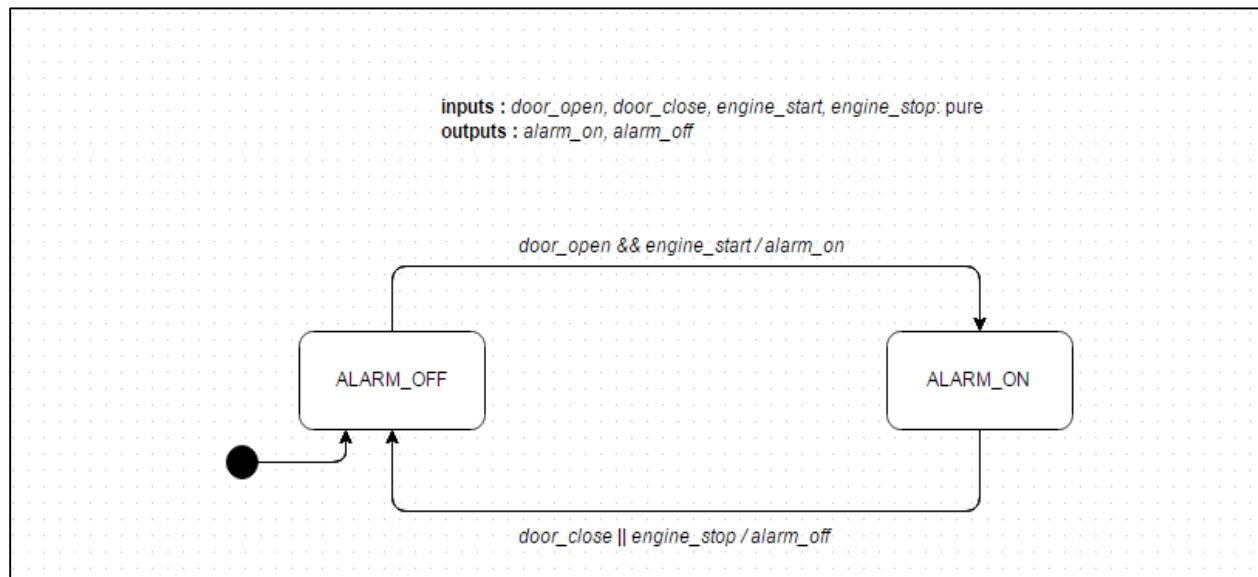
Speed Module
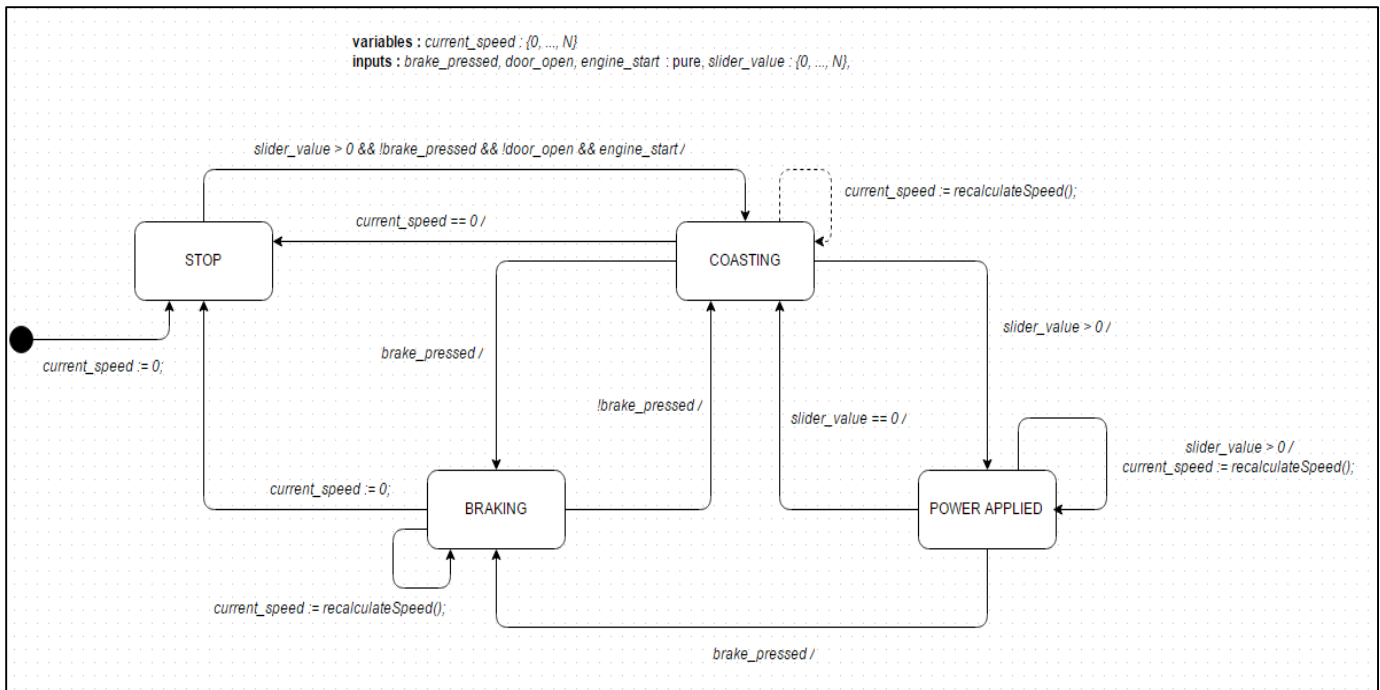
In this module, the relative speed is displayed based on the value emitted from the slider. The displayed value will depict the current state of the car which is either in coasting, acceleration or deceleration. The *recalculateSpeed()* function basically uses the Drag Force equation, the slider and the force sensor's value to calculate the emitted speed.

Internal Lights Module
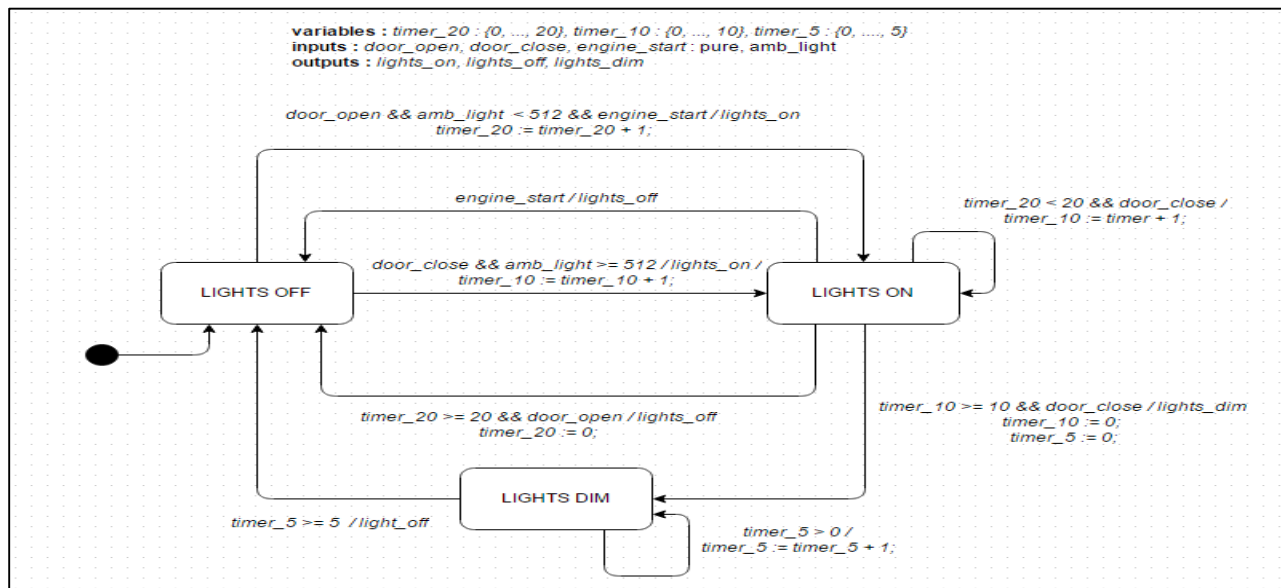
In this module, the state of the interior lights is dependent on the states of the engine and the door. Also, depending on the duration, these lights will either turn off immediately or dimmed down till there is no light.

Alarm Module

inputs : door_open, door_close, engine_start, engine_stop: pure
outputs : alarm_on, alarm_off

door_open && engine_start / alarm_on

ALARM_OFF

ALARM_ON

door_close || engine_stop / alarm_off

In this module, the state of the alarm is changed depending on the states of the engine and the door. The alarm can be turned on when both engine and door states are active. However, it gets turned off once the door is closed.

Additional Features

In this project we have the following additional features integrated and implemented into the main system. These features can be categorized into a few types, namely (1) Movement Features; (2) Engine Features; and (3) Door Features.

In Movement Features, we have the following:

1. At the moment when the brake is pressed via the force sensor, the slide value will be ignored and the car will decelerate at a rate proportionate to the force sensor's value. This will continue until the brake is released or until the speed of car is at zero. After the brake is released, the slider value will cease to be ignored.
2. Application of the Drag Equation to simulate the non-linear acceleration of the car.
3. When the car is moving, the engine cannot be turned off.

In Door Features, we have the following:

1. The door cannot be opened when the car is moving.
2. The car cannot move if the door is opened.

In Engine Features, we have the following:

1. The headlights can only be turned on when the engine is turned on.