```c
#include <stdio.h>

#include "Roach_Events.h"
#include "roach.h"
#include "Roach_State_Machine.h"
#include "timers.h"

//a list of states that this SM uses:

enum {
    Moving_Forward,
    Reversing,
    Turning_Randomly,
    Hiding,
};

int current_state;

/* This function initializes the roach state machine.
 * At a minimum, this requires setting the first state.
 * Also, execute any actions associated with initializing the SM
 * (that is, the actions on the arrow from the black dot in the SM diagram)*/
void Initialize_RoachStateMachine(void)
{
    current_state = Moving_Forward;
    Roach_LeftMtrSpeed(100);
    Roach_RightMtrSpeed(100);

    //seed rand:
    srand(Roach_LightLevel());
};

/*
 * @briefThis function feeds newly detected events to the roach state machine.
 * @param event:  The most recently detected event*/
void Run_RoachStateMachine(Event event)
{
    switch (current_state) {
        case Moving_Forward:
            printf("Current state:  Moving_Forward\r\n");
            if (event == ENTERED_DARK) {
                current_state = Hiding;
                Roach_LeftMtrSpeed(0);
                Roach_RightMtrSpeed(0);
            }
            if (event == FRONT_LEFT_BUMP_PRESSED || event ==
             FRONT_RIGHT_BUMP_PRESSED){
                current_state = Reversing;
                Roach_LeftMtrSpeed(-100);
                Roach_RightMtrSpeed(-100);
    }

    break;



case Reversing:
    printf("Current state:  reversing\r\n");
    if (event == ENTERED_DARK) {
        current_state = Hiding;
        Roach_LeftMtrSpeed(0);
        Roach_RightMtrSpeed(0);
    }
    if(event == FRONT_RIGHT_BUMP_RELEASED){
        Roach_LeftMtrSpeed(0);
        Roach_RightMtrSpeed(100);
        int x = rand()%1000;
        TIMERS_InitTimer(NAV_TIMER , x);
        current_state = Turning_Randomly;

    }
    if(event == FRONT_LEFT_BUMP_RELEASED){
        Roach_LeftMtrSpeed(100);
        Roach_RightMtrSpeed(0);
        int x = rand()%1000;
        TIMERS_InitTimer(NAV_TIMER , x);
        current_state = Turning_Randomly;
    }

    break;

case Turning_Randomly:
    printf("Current state:  Turning_Randomly\r\n");
    if (event == ENTERED_DARK) {
        current_state = Hiding;
        Roach_LeftMtrSpeed(0);
        Roach_RightMtrSpeed(0);
    }
    if (event == NAV_TIMER_EXPIRED){
        Roach_LeftMtrSpeed(100);
        Roach_RightMtrSpeed(100);
        current_state = Moving_Forward;
    }
    if (event == FRONT_LEFT_BUMP_PRESSED || event ==
     FRONT_RIGHT_BUMP_PRESSED){
        current_state = Reversing;
        Roach_LeftMtrSpeed(-100);
        Roach_RightMtrSpeed(-100);

    }
    break;
```

```
        case Hiding:
            printf("Current state:  Hiding\r\n");
            break;
    }
};
```