AMA_EOS LAB 1

1.2 ANALYSIS

Event 1: ev_ButtonStart - > button to start the game

Event 2: ev_Button -> button pressed

The state transitions from STATE_START to STATE_DISPLAYSSD (after the random time of 1-3 seconds is over)

Event 3: ev_randTime -> random time between 1 and 3 seconds to transition state

Event 4: ev_timeout -> event triggered after user does not press any button for 1s

1.3 ERIKA ELEMENTS

```
ISR(systick_handler)
{
    CounterTick(cnt_systick);
}

CyGlobalIntEnable; /* Enable global interrupts. */

//Set systick period to 1 ms. Enable the INT and start it.
EE_systick_set_period(MILLISECONDS_TO_TICKS(1, BCLK__BUS_CLK__HZ));
EE_systick_enable_int();
```

***In my code, I have named the “alrm_Tick1ms” as “alarm_measureTime”

```
SetRelAlarm(alarm_measureTime,1,1);
```

There are 2 alarms to be fired only once, I have named them “alarm_randTime” and “alarm_1000”:

```
randTime = (rand() % 3 + 1) * 1000; //random wait time value
CancelAlarm(alarm_randTime);
SetRelAlarm(alarm_randTime,randTime,0); // 2 seconds for test purposes
```

alarm_randTime is a single shot alarm which is ran from 1-3 seconds (random number) before displaying the SSD values

```
SetRelAlarm(alarm_1000,1000,0);
```

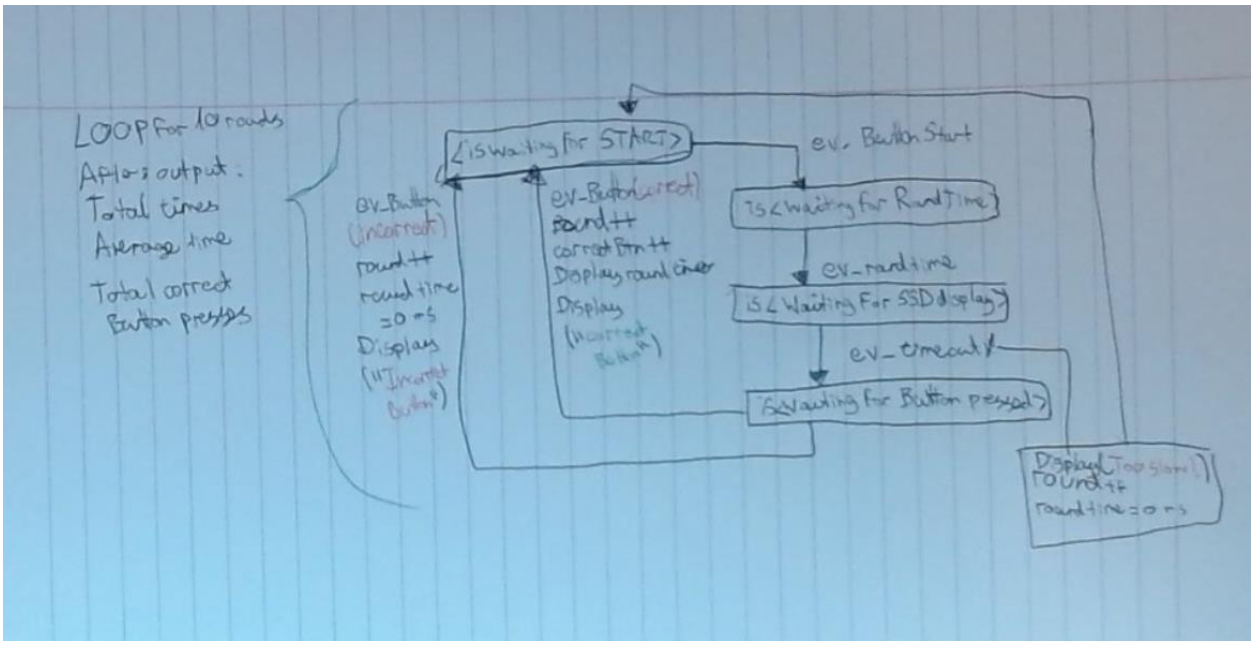
alarm_1000 is a single shot alarm which switches to the STATE_TIMEOUT state if the user does not press a button for 1s.

Neither of these alarms need to be cyclic.

```
ISR2(isr_Button)
{

    if(BUTTON_1_Read() == 1)
    {
        SetEvent(tsk_game,ev_Button);
    }


    if(BUTTON_2_Read() == 1)
    {
        SetEvent(tsk_game,ev_Button);
    }

}
```

The ISR is category 2 because it is a software-based interrupt. Unlike category 1 which would be hardware-based interrupt. In the top design the 4 buttons are OR'd together to the isr_Button interrupt making it a category 1 however for the ERIKA OS, category 2 is used.

State machine diagram



ARCADIAN STYLE

GLOWER

glower tasks would be:

1. “tsk_rgb” changing the parameters each time the task is activated (500ms cyclic task called 3 times)
2. “tsk_white” toggling the RGB led with parameters resulting in a white output. This would be a 100ms cyclic task called 6 times.

```
for(int num = 0; num < sizeof(RG_glowtable); num++)
{
    for(uint8_t time = 0; time <=RG_glowtable_1[num][4]; time++)
    {
       LED_RGB_Set(RG_glowtable_1[num][0],RG_glowtable_1[num][1],RG_glowtable_1[num][2])
    }
}
```

The code would look something like this. Loop through the table and for each element, set the RGB values respectively for the ttime specified before going on to the next.

****I had some errors in my code and was not able to complete the ARCADIAN STYLE SECTION so I removed it form the final code**