Reaction Game with ERIKA OS

Learning focus:
OSEK, tasks, events, alarms

Your task is to develop a simple reaction game similar to the one developed during the Presemester class using Erika OS and to extend this functionality with some Arcadian light effects.

Marking

Correct Design and Implementation Reaction Game   6 points
Correct Design and Implementation "Arcadian Style"   4 points

You need 4 points in order to pass the lab!

Print the exercise and bring a handwritten solution together with your code to the lab.

---

1   Reaction Game

1.1   Requirements

Your task is to develop an embedded application with the following requirements
- The application runs as an Autosar Application
- Upon startup, the program will show a welcome message via the serial port.
- After pressing one of the two white buttons, the program will wait for a random time. After waiting for 1s to 3s a random value (1 or 2) will be displayed on both 7segment displays
- The user has to press the right button in case a '1' is displayed and the left button in case a '2' is displayed
- In case the correct button is being pressed, the measured reaction time in [ms] will be shown and the game can be started again by pressing one of the two buttons.
- In case a wrong button is pressed, an error message will be displayed and the game can be started again by pressing one of the two buttons.
- In case the user does not press a button within 1 s, the message "Too slow" will appear and the game can be started again by pressing one of the two buttons.
- One game consists out of 10 rounds
- At the end of a game, print the score (i.e. correct number of button pressed), the total time and the average time

Example output via the serial port:

Reaction test program round 1
press one of the two buttons to start...

Great - correct button pressed
Reaction time in ms: 249

===

1.2 Analysis

Events are input signals which can be fired by the user or by the system to initiate a state change.

Example:

- After pressing button1 or button2, the program will wait for a random time.

Which events are mentioned in this requirement?

Event 1: _____

Event 2: _____

Which state transition is mentioned in this requirement?

```



```

In addition to the two user input events, two additional system events can be identified from the requirements.

Event 3: _____

Event 4: _____

1.3 Erika elements

Alarms can be used to implement time signals. They can be compared with timer interrupts but provide more flexibility. Furthermore, several alarms can share one physical timer resource.

Provide the code snippets which will create the 1ms tick for the base counter.

```



```

In our application, we will use the alarm `alrm_Tick1ms` which will be fired every 1ms. Every time the alarm is fired, the task `tsk_Timer` will be called.

Check the API call `SetRelAlarm` and provide the code to configure the alarm to be fired every 1ms.

`SetRelAlarm(_____, _____, _____)`

Alternatively, an alarm can also be configured to be fired only once. Where might we need this feature and how do you configure this?

Add ONE ISR function for the button pressed. Will this function be an ISR1 or ISR2 category ISR? Explain.
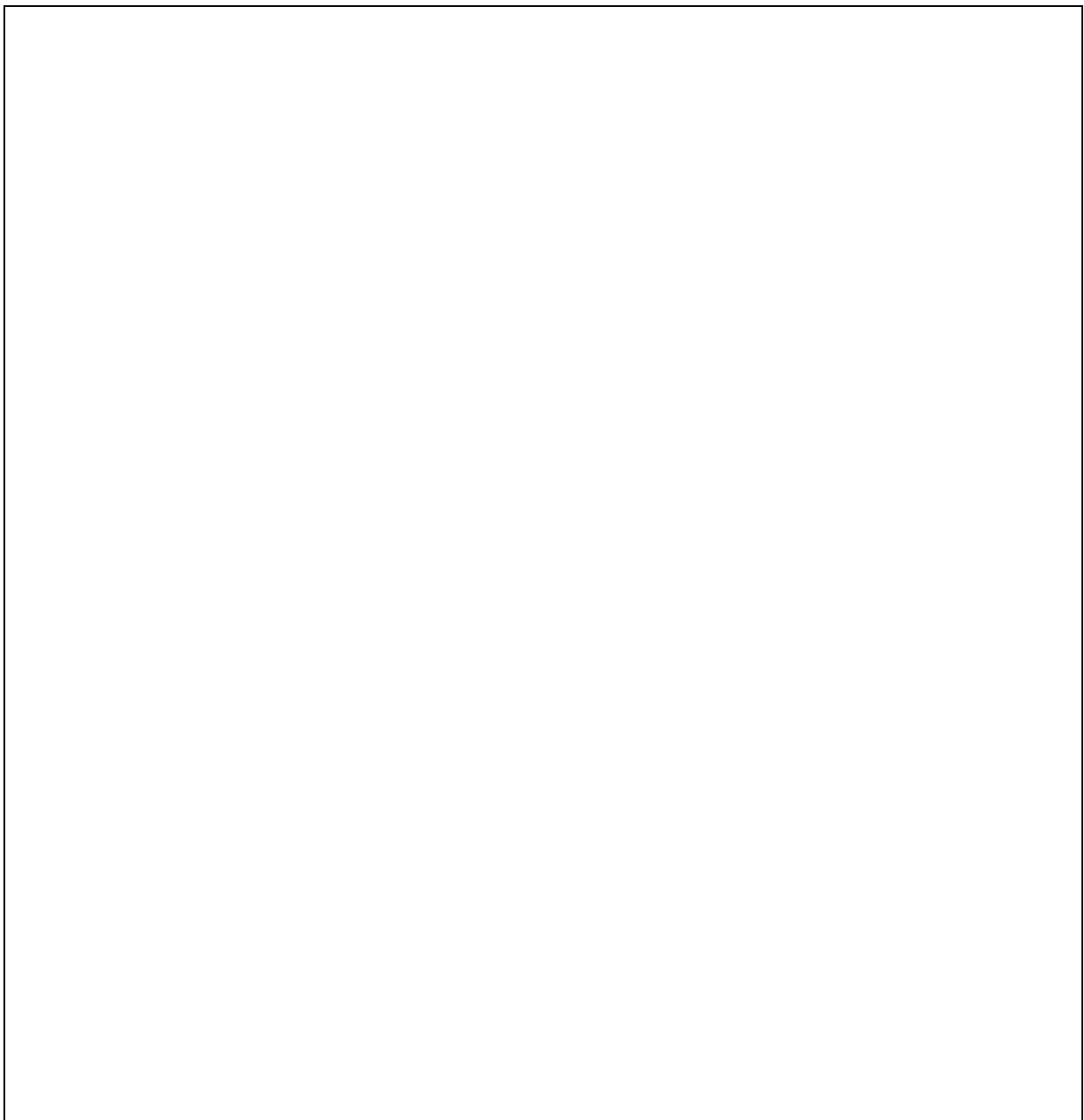
1.4   State Maschine

Draw a state machine of the system.

Describe the states using the **Is<WaitingforSomething>** mini sentence convention.

Use the following convention to describe a transition between two states:

Event [optional guard] / Action

A guard is an additional condition which is checked when an event occurs. In our game, this could be for example a check, if the correct button has been pressed. Guards are very often functions returning a Boolean value.

Implement the code for the program based on the given requirements and design.

2 Arcadian Style

2.1 Requirements

In order to make the game a bit fancier, we want to add some Arcadian style light effect. The light effects may not interfere with rest of the functionality.

Fader:

In a first step, we want to create a fading traveling light. The three LED's are glowing using the following pattern:
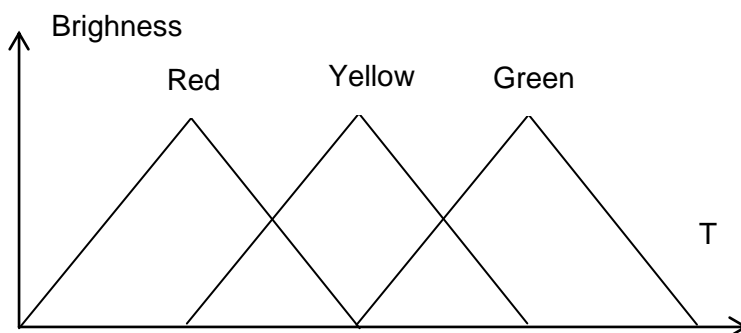
You might want to check some 80ies Knightrider movies to get into the grove, e.g. https://www.youtube.com/watch?v=Mo8Qls0HnWo

Glower:

Using the RGB LED, we want to implement an easily configurable glowing function. Using a const - table like the following (pseudo code)

```
const RG__Glow_t RG_glowtable_1[] = {
//Red    Green Blue  TimeInMS
{255,   0,    0,    500},
{0,     255,  0,    500},
{0,     0,    255,  500},
{0,     0,    0,    100},
{255,   255,  255,  100},
{0,     0,    0,    100},
{255,   255,  255,  100},
{0,     0,    0,    100},
{255,   255,  255,  100}
};
```

will create the sequence:

- 500ms red
- 500ms green
- 500ms blue
- 100ms off
- 100ms white
- 100ms off

- 100ms white
- 100ms off
- 100ms white

This sequence will be repeated permanently.

Provide a good declaration for `RG__Glow_t`

How many tasks do you need for the new Arcadian functions? Explain.