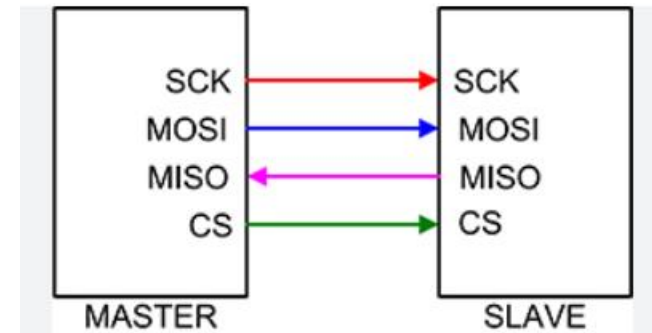


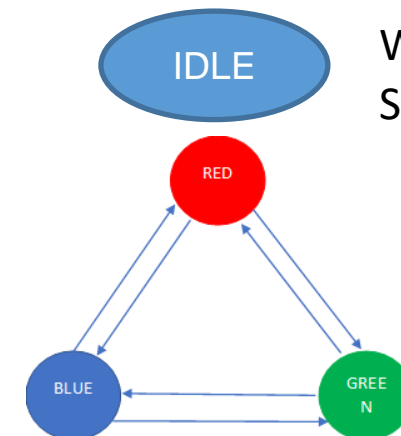
System Overview

- 2 ECUs using SPI
- ECU 1 should control the state machines
- ECU2 should execute the output of the state machine
- On ECU1 when a switch is pressed, ECU2 will execute either state machine at 1000ms depending on the SPI message that ECU1 sent.
- On ECU1 UART messages are sent to PC indicating the states

Read on-board switches and send command



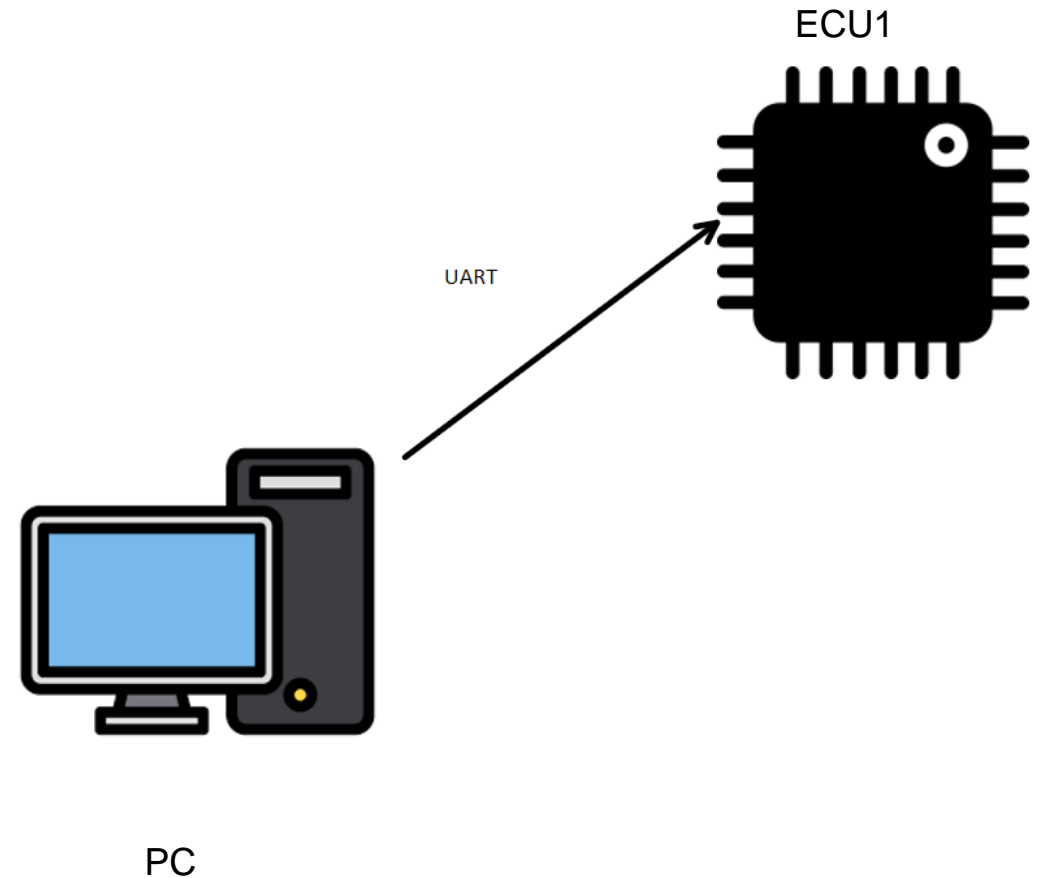
Led State Machine



White led on State(IDLE)

UART to PC connections and Statuses

- ECU1 shall be connected to PC through UART and the messages sent to PC should indicate which state ECU2 is running along with indicating any error message sent from ECU2



Functional Requirements

1) State Machine:

ECU 1 has the following functionality:

- a. Read the input switches (the on-board switches).
- b. SW1 & 2 controls ECU2.
- c. ECU1 sends an SPI message each **50 ms**
- d. Sends UART messages to PC each **State change**

ECU 2 has the following functionality

- a. ECU 2 State machine
- b. Process the commands from the SPI very **10ms**.
- c. Execute the state machine as the following:
 - Move to the next state (clockwise) whenever the SW1(message from SPI is received) (each **1000ms**)
 - Move to the next state (anticlockwise) whenever the SW2 (message from SPI is received) (each **1000ms**)
 - If both are pressed. The output on both ECUs will return to initial state(RED)
- d. If no state change or no switches were pressed was identified for more than **10 seconds** ECU1 shall command ECU2 to go to IDLE state(WHITE)

NOTE:

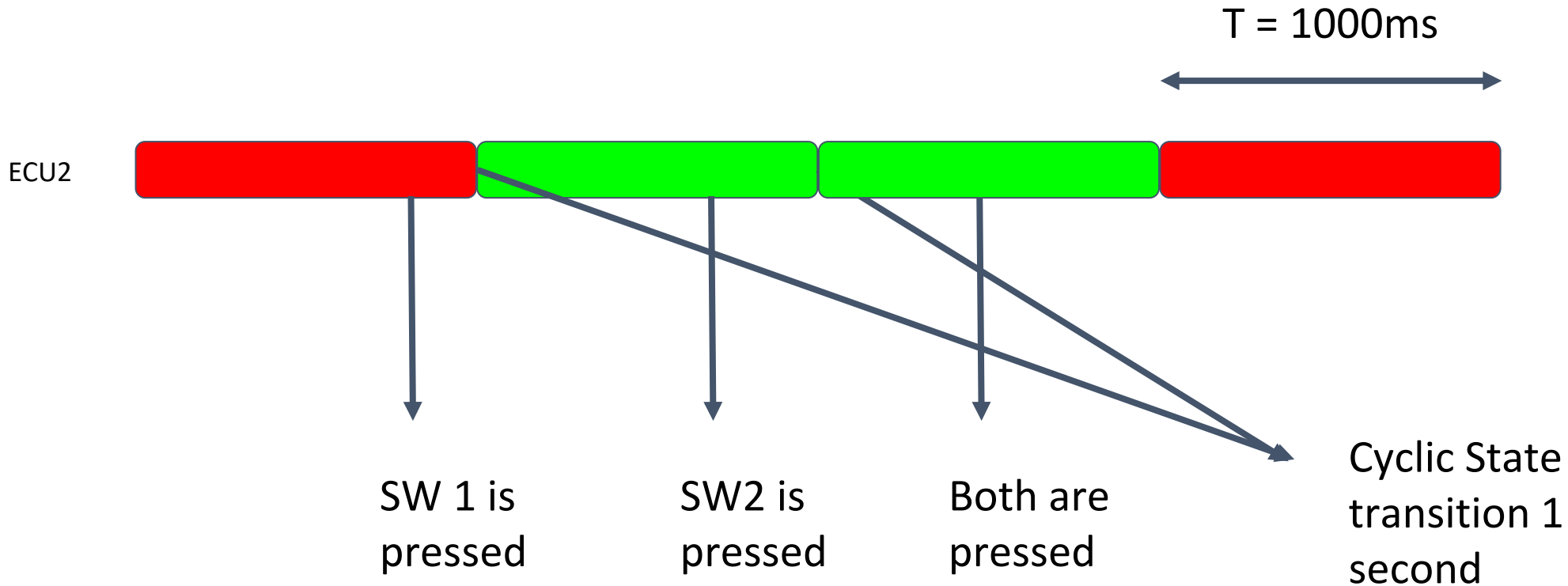
- I. You are not allowed to use delay functions
- II. Switch debouncing is required
- III. Error Handling is a must
- IV. Architecture shall be defined
- V. Any timings that are not mentioned shall be defined and OS shall act accordingly

Functional Requirements

1. All drivers are expected to follow configuration concept(each module will have cfg.h and .c for the specific module)
2. OS and any other services shall follow configuration concept and layered architecture
3. Application state machine shall reside in Application layer and violation of layered architecture is prohibited

Functional Requirements

Timing requirements:



ECUs have periodicities of 1 second so they cannot change colors unless after the 1 second window.

Non-Functional Requirements

- Each component shall have:
 - Source file.
 - Header file.
- Code should be free from compilation warnings(excluding registers files).
- Component description and interfaces shall be documented

Appendix A

Comments

☐ File Comment:

```
/******  
/*Module Name: */  
/*Author : */  
/*Purpose: */  
/******
```

☐ Line Comment:

```
/* Comment Here */
```

☐ Function Comment:

```
/******  
/*Function Name : */  
/*Inputs : input parameters */  
/*Outputs : Return Parameters */  
/*Reentrancy : Reentrant/Non-Rentrant */  
/*Synchronous : Synch / Asynch */  
/*Description : What Function Do ? */  
/******
```

Appendix B

Documents

- Each ECU should have a document following the below structure:

1. Each component should has this table for each API:

Function Name	
Input Parameters	Parameter1 Description: Parameter2 Description:
Output Parameters	Parameter1 Description: Parameter2 Description:
Input/Output Parameters	Parameter1 Description: Parameter2 Description:
Fun. Description	

2. Flowcharts