

EECS 3215
Embedded Systems
Lab 8 - Timers
Michael Founk 213641204
Sean O'Brien 213735741

Problem Statement

The problem statement of this lab is to simulate the seatbelt warning systems of a vehicle. The goal is to create inputs which simulate monitors of the state of the vehicle, with inputs simulating the engine state, whether the seat is occupied, and whether the belt is buckled. The system should respond to these inputs by responding with outputs simulating a buzzer and a warning light. The solution should utilize the systick timer to limit the time in which an output buzzer is active.

Top-Level Design

Our design is straightforward, we initialize the inputs and outputs in the same way we have in previous labs. What is different is the need to initialize the systick timer. We initialize the systick timer using the control register, and we set the reload time for 5 seconds using the reload register. The system moves between states depending on the input values. The state of interest is the one utilizing the systick timer. The system waits for the systick timer to equal zero before resetting the timer by writing to the 16th bit of control register. Inside this area, a flag is set to turn the green LED off while keeping the red LED on. The other states reset this flag so that the green LED will remain on for five seconds when the system returns to the unbuckled state.

Code

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "MKL43Z4.h"
#include "fsl_debug_console.h"
#include <stdlib.h>

int main(void) {

    BOARD_InitBootPins();
    BOARD_InitBootClocks();
    BOARD_InitBootPeripherals();
    BOARD_InitDebugConsole();

    SysTick->LOAD |= 0xFF;
    SysTick->CTRL |= 0x05;

    SIM->SCGC5 |= SIM_SCGC5_PORTE_MASK; // enable clock to port E
    SIM->SCGC5 |= SIM_SCGC5_PORTD_MASK; // enable clock to port D
    SIM->SCGC5 |= SIM_SCGC5_PORTC_MASK; // enable clock to port C
```

```
SIM->SCGC5 |= SIM_SCGC5_PORTA_MASK; // enable clock to port A
```

```
PORTD->PCR[5] |= 0x100; // enable green led  
PTD->PDDR |= 1<<5u; // set green led to output  
PTD->PSOR |= 1<<5u; // off by default
```

```
PORTE->PCR[31] |= 0x100; // enable red led  
PTE->PDDR |= 1<<31u; // set green led to output  
PTE->PSOR |= 1<<31u; // off by default
```

```
PORTC->PCR[3] |= 0x103; // enable pushbutton 1 (belt buckled) left side  
PTC->PDDR &= 0<<3u; // set to input
```

```
PORTA->PCR[4] |= 0x103; // enable pushbutton 2 (seat occupied)  
PTA->PDDR &= 0<<4u; // set to input
```

```
PORTA->PCR[1] |= 0x103; // enable pin for third input signal (engine on)  
PTA->PDDR &= 0<<1u; // set to input
```

```
PORTA->PCR[2] |= 0x103; // enable the second wire  
PTA->PDDR &= 0<<2u;
```

```
int on = 0;  
while(1) {
```

```
    if (PTA->PDIR & 0b10){  
        //if engine is off, both LEDs should be off  
        printf("engine off\n");  
        PTE->PSOR |= 1<<31u;  
        PTD->PSOR |= 1<<5u;  
        on = 0;  
    }
```

```
    else if(!(PTA->PDIR & 0b10) && PTA->PDIR & 0b100 && !(PTC->PDIR & 0b1000)){  
        // if seat is occupied, and belt is not buckled,  
        // turn on warning light and set buzzer for 5 seconds  
        printf("unbuckled\n");  
        if(on == 0){  
            PTE->PCOR |= 1<<31u;  
            PTD->PCOR |= 1<<5u;  
            while(!(SysTick->VAL == 0)){  
                SysTick->CTRL |= 0x10005;  
                PTD->PSOR |= 1<<5u;  
                on = 1;  
            }  
        }
```

```

    }

    else{
        PTE->PCOR |= 1<<31u;
        PTD->PSOR |= 1<<5u;
    }
}

    else if (!(PTA->PDIR & 0b10) && !(PTA->PDIR & 0b100) &&
!(PTC->PDIR & 0b1000)){
    // if the seat is occupied and the belt is buckled
    // the warning light and buzzer should be off
    printf("proper\n");
    PTE->PSOR |= 1<<31u;
    PTD->PSOR |= 1<<5u;
    on = 0;
}

    else {
        // the system is in an illegal state
        // the alarms should be set
        printf("Illegal\n");
        PTE->PCOR |= 1<<31u;
        PTD->PCOR |= 1<<5u;
        on = 0;
    }
}
}

```