NAME: SURAJ THITE,ATHARV DESAI
PES PROJECT 2 (CREATION OF CUSTOM MAKE FILE)

```
/*
 * main.c
 *
 *  Created on: Sep 28, 2019
 *      Author:SURAJ THITE , ATHARV DESAI
 */
/*******************************************************************************
 * Include Files
 ******************************************************************************/


#include "fsl_rtc.h"
#include "board.h"
#include "fsl_debug_console.h"
#include "fsl_gpio.h"

#include "clock_config.h"
#include "pin_mux.h"
#include "RGBled.h"

/*******************************************************************************
 * Definitions
 ******************************************************************************/
//#define BOARD_LED_BLUE_GPIO BLUE_LED_PORT
//#define BOARD_LED_BLUE_GPIO_PIN BLUE_LED_PIN

/*******************************************************************************
 * Prototypes
 ******************************************************************************/
/*!
 * @brief delay a while.
 */
int delay(int x);
void RTC_get();
void RTC_init();




/*******************************************************************************
 * Variables
 ******************************************************************************/
unsigned int time[]=
{3000,1000,2000,600,1000,400,1000,200,500,100,500,100,500,100,1000,200,1000,400,2000,
600};

rtc_datetime_t date;
rtc_config_t rtcConfig;

/*******************************************************************************
 * Code
 ******************************************************************************/
```

```
/*****************************************************************************/
/* Function name: delay
 * Parameters: time in mili seconds
 * Return : return the no of counts for the delay generated.
 * Description: Function to create delays.
 * no of cycles required for for loop = 1/(24Mhz)*10 = 2400 cycles.
/*****************************************************************************/
int delay(int time_ms)
{
	volatile uint32_t i = 0;
	for (i = 0; i < 2400*time_ms; ++i)
	{
		__asm("NOP"); /* No operation */
	}
	return i;
}


/*!
 * @brief Main function
 */

/*****************************************************************************/
/* Function name: main
 * Parameters: void
 * Return : return
 * Description: Main loop . */
/*****************************************************************************/
int main(void)
{
	/* Board pin, clock, debug console init */
	BOARD_InitPins();
	BOARD_BootClockRUN();
	BOARD_InitDebugConsole();
	RTC_init();
	RGB_init();
	int time_last_event;
	/* Init output LED GPIO. */
	// GPIO_PinInit(BOARD_LED_BLUE_GPIO,BOARD_LED_BLUE_GPIO_PIN,
&led_blue_config);

	unsigned int i = 0;
#ifdef fb_run             //TO be executed for RUN = fb_run flag set.
	for ( i=0;i<20;i++)
	{
		if ((i)%2==0)//Check status for ON condition
		{
			led_switch(i/6);    //Switch to led state according to the data
array
			delay(time[i]);           //Delay according to the current lookup
table.
		}
		else
		{
```

```
                    RGB_OFF();   //Switch LED off
                    delay(time[i]);     //Delay according to the current lookup
table.
             }

      }
#endif

#ifdef fb_debug           //TO be executed for RUN = fb_debug flag set.
      for ( i=0;i<20;i++)
      {
             if ((i)%2==0)//Check status for ON condition
             {
                    led_switch(i/6);
                    status_print_on(i/6);     //Switch to led state according to the
data array
                    time_last_event=delay(time[i]);  //Return value for no of loops
executed for previous delay
                    PRINTF("%d  ",time_last_event);  //Print no of cycles executed
for previous state.
                    RTC_get();   //Print RTC values on the Terminal for the real time
date and time from the RTC.

             }
             else
             {
                    status_print_0ff();//Switch to led state according to the data
array
                    time_last_event= delay(time[i]);//Return value for no of loops
executed for previous delay
                    PRINTF("%d  ",time_last_event);  //Print no of cycles executed
for previous state.
                    RTC_get();   //Print RTC values on the Terminal for the real time
date and time from the RTC.

             }

      }
#endif

#ifdef pc_run//TO be executed for RUN = fb_debug flag set.
      int main (void)
      {
          int i=0;

      for ( i=0;i<20;i++)
          {
             if ((i)%2==0)//Check for ON condition
             {

                    status_print_on(i/6);     // Print the on status for the current
state

                    delay(time[i]);     //Delay according to the current state.
             }
```

```
            else
            {

                  status_print_0ff(); //OFF State

                  delay(time[i]);     //Delay according to the current state
            }

          }

      return 0;
      }
#endif

#ifdef pc_debug
      int main (void)
      {
          int i=0;
          int time_last_event;
      for ( i=0;i<20;i++)
          {
             if ((i)%2==0)
             {

                  status_print_on(i/6);
                  time_last_event=delay(time[i]);
                  printf("%d",time_last_event);
                  delay(time[i]);
             }
             else
             {

                  status_print_0ff();
                  time_last_event= delay(time[i]);
                  printf("%d",time_last_event);
                  delay(time[i]);
             }

          }

      return 0;
      }
#endif
}

/*****************************************************************************/
/* Function name: RTC_init
 * Parameters: void
 * Return : void
 * Description: Function to initialize the RTC  and set current date/time. */
/*****************************************************************************/
void RTC_init()
{
      RTC_GetDefaultConfig(&rtcConfig);      //Default config for RTC
      RTC_Init(RTC, &rtcConfig);//Initialize the RTC
```

```
	CLOCK_EnableClock(kCLOCK_Rtc0);  //Enable CLock
	CLOCK_SetRtcClkOutClock(kCLOCK_Rtc0);  //Set Clock
	date.year = 2019U;
	date.month = 10U;   //Initialize
	date.day = 30U;
	date.hour = 14U;
	date.minute = 0;
	date.second = 0;
	RTC_SetDatetime(RTC, &date);
	RTC_StartTimer(RTC);


}
/*************************************************************************/
/* Function name: RTC_get
 * Parameters: void
 * Return : void
 * Description: This function prints the current date and time from RTC. */
/*************************************************************************/

void RTC_get()
{
	RTC_GetDatetime(RTC, &date);
	PRINTF("Current datetime: %d-%d-%d %d:%d:%d\r\n", date.year, date.month,
date.day, date.hour,date.minute, date.second);

}
```

```c
/*
 * RGBled.h
 *
 *  Created on: Sep 28, 2019
 *   Author:SURAJ THITE , ATHARV DESAI
 */

#ifndef RGBLED_H_
#define RGBLED_H_
void led_switch(int n);   //Function to switch the led_state
void RGB_init();    //Function to initialize the RGB Leds
void RGB_OFF();     //Function to turn off the RGB led off
void status_print_on();   //Function to Print the On status on the UART screen
void status_print_0ff();//Function to Print the Off status on the UART screen

#endif /* RGBLED_H_ */
```

```c
/*
 * RGBled.c
 *
 *  Created on: Sep 28, 2019
```

```
 *      Author:SURAJ THITE , ATHARV DESAI
 */



#include "board.h"
#include "fsl_debug_console.h"
#include "fsl_gpio.h"

#include "clock_config.h"
#include "pin_mux.h"

/*******************************************************************************/
/* Function name:RGB_init
 * Parameters: void
 * Return : void
 * Description: Function to initialize the GPIO RGB Led Pins . */
/*******************************************************************************/
void RGB_init()
{
             gpio_pin_config_t led_blue_config = {
               kGPIO_DigitalOutput, 1,
           }; //Config the pin  for BLUE LED to Digital Output
              GPIO_PinInit(BOARD_LED_BLUE_GPIO,BOARD_LED_BLUE_GPIO_PIN,
&led_blue_config);
           gpio_pin_config_t led_red_config = {
               kGPIO_DigitalOutput, 1,
           }; //Config the pin  for RED LED to Digital Output
           GPIO_PinInit(BOARD_LED_RED_GPIO,BOARD_LED_RED_GPIO_PIN, &led_red_config);
           gpio_pin_config_t led_green_config = {
               kGPIO_DigitalOutput, 1,
           }; //Config the pin  for GREEN LED to Digital Output
           GPIO_PinInit(BOARD_LED_GREEN_GPIO,BOARD_LED_GREEN_GPIO_PIN,
&led_green_config); //Initialize the GPIO Pins
}


/*******************************************************************************/
/* Function name:led_switch(int n )
 * Parameters: current state n
 * Return : void
 * Description: Function to initialize the GPIO RGB Led Pins . */
/*******************************************************************************/
void led_switch(int n)
 {
       GPIO_SetPinsOutput(BOARD_LED_GREEN_GPIO, 1u << BOARD_LED_GREEN_GPIO_PIN);
       //Clear the Pins
       GPIO_SetPinsOutput(BOARD_LED_RED_GPIO, 1u << BOARD_LED_RED_GPIO_PIN);
       GPIO_SetPinsOutput(BOARD_LED_BLUE_GPIO, 1u << BOARD_LED_BLUE_GPIO_PIN);

       switch (n)
       {
       // Switch LED BLUE ON and TURN OTHER LEDs OFF
       case 0:
              {
```

```
            GPIO_ClearPinsOutput(BOARD_LED_BLUE_GPIO, 1u <<
BOARD_LED_BLUE_GPIO_PIN);
            GPIO_SetPinsOutput(BOARD_LED_RED_GPIO, 1u << BOARD_LED_RED_GPIO_PIN);
            GPIO_SetPinsOutput(BOARD_LED_GREEN_GPIO, 1u <<
BOARD_LED_GREEN_GPIO_PIN);
            }
            break;
            // Switch LED RED ON and TURN OTHER LEDs OFF
      case 1:
      {

                  GPIO_ClearPinsOutput(BOARD_LED_RED_GPIO, 1u <<
BOARD_LED_RED_GPIO_PIN);
                  GPIO_SetPinsOutput(BOARD_LED_BLUE_GPIO, 1u <<
BOARD_LED_BLUE_GPIO_PIN);
                  GPIO_SetPinsOutput(BOARD_LED_GREEN_GPIO, 1u <<
BOARD_LED_GREEN_GPIO_PIN);
      }
                  break;
                  // Switch LED GREEN ON and TURN OTHER LEDs OFF
      case 2:
      {
                  GPIO_ClearPinsOutput(BOARD_LED_GREEN_GPIO, 1u <<
BOARD_LED_GREEN_GPIO_PIN);
                  GPIO_SetPinsOutput(BOARD_LED_RED_GPIO, 1u <<
BOARD_LED_RED_GPIO_PIN);
                  GPIO_SetPinsOutput(BOARD_LED_BLUE_GPIO, 1u <<
BOARD_LED_BLUE_GPIO_PIN);
      }
                  break;
      case 3:
            {
                  // Switch LED BLUE ON and TURN OTHER LEDs OFF
            GPIO_ClearPinsOutput(BOARD_LED_BLUE_GPIO, 1u <<
BOARD_LED_BLUE_GPIO_PIN);
            GPIO_SetPinsOutput(BOARD_LED_RED_GPIO, 1u << BOARD_LED_RED_GPIO_PIN);
            GPIO_SetPinsOutput(BOARD_LED_GREEN_GPIO, 1u <<
BOARD_LED_GREEN_GPIO_PIN);
            }
            break;
      }
 }


/*************************************************************************/
/* Function name:RGB_off
 * Parameters: void
 * Return : void
 * Description: Function to turn off the RGB Led Pins . */
/*************************************************************************/

void RGB_OFF()
{     // Clear all the  LEDs.
      GPIO_SetPinsOutput(BOARD_LED_BLUE_GPIO, 1u << BOARD_LED_BLUE_GPIO_PIN);
      GPIO_SetPinsOutput(BOARD_LED_GREEN_GPIO, 1u << BOARD_LED_GREEN_GPIO_PIN);
```

```
	GPIO_SetPinsOutput(BOARD_LED_RED_GPIO, 1u << BOARD_LED_RED_GPIO_PIN);
}
/*************************************************************************/
/* Function name:status_print_on(int n)
 * Parameters: current index  n
 * Return : void
 * Description: Function to print the current state on the UART . */
/*************************************************************************/

void status_print_on(int n)
 {
	switch (n)
	{
	// Print the state for  Switch LED BLUE ON and TURN OTHER LEDs OFF
	case 0:
		{
			PRINTF(" BLUE LED ON ::");
		}
		break;
		// Print the state for Switch LED RED ON and TURN OTHER LEDs OFF
	case 1:
	{

		PRINTF(" GREEN LED ON ::");
	}
			break;
			// Print the state for Switch LED GREEN ON and TURN OTHER LEDs
OFF
	case 2:
	{
		PRINTF(" RED LED ON :: ");
	}
			break;
	case 3:
		{
		// Print the state for  Switch LED BLUE ON and TURN OTHER LEDs OFF
		PRINTF(" BLUE LED ON :: ");
		}
		break;
	}
 }
/*************************************************************************/
/* Function name:status_print_off()
 * Parameters: void
 * Return : void
 * Description: Fucntion to print the off status on the Terminal screen .
/*************************************************************************/
void status_print_0ff()
{
	PRINTF(" OFF STATE :: ");
}
```

```
##############################   MAKE FILE FOR PES PROJECT 2
################################################################################################
##############################    Authors : Atharv Desai , Suraj
Thite#########################################################################################
##############################    Run Modes : fb_run ,fb_debug , pc_run
,pc_debug.#####################################################################
#Reference :   https://mcuoneclipse.com/2017/07/22/tutorial-makefile-projects-with-
eclipse/
################################################################################################
############
# Defining command for removing files
RM := rm -rf

################################################################################################
############
# Include files for building the target
INCLUDES := \
       -I"CMSIS" \
       -I"source" \
       -I"board" \
       -I"drivers" \
       -I"utilities" \
       -I"startup"

#Check the RUN flag to for defining compilers and linkers
ifeq ($(RUN),fb_debug)
Flag = -Dfb_debug
CC := arm-none-eabi-gcc
LL := arm-none-eabi-gcc
endif

ifeq ($(RUN),fb_run)
Flag = -Dfb_run
CC := arm-none-eabi-gcc
LL := arm-none-eabi-gcc
endif

ifeq ($(RUN),pc_debug)
Flag = -Dpc_debug
CC := gcc
LL := gcc
endif

ifeq ($(RUN),pc_run)
Flag = -Dpc_run
CC := gcc
LL := gcc
CFLAGS = -std=c99
endif


################################################################################################
############
# Binary Output  to build
EXE := \
```

```
  ./debug/PES_PROJECT2.axf

################################################################################
############
# Dependencies enumerated below
C_DEPS = \
       ./debug/startup_mkl25z4.d \
       ./debug/system_MKL25Z4.d \
       ./debug/board.d \
       ./debug/clock_config.d \
       ./debug/peripherals.d \
       ./debug/pin_mux.d \
       ./debug/fsl_clock.d \
       ./debug/fsl_common.d \
       ./debug/fsl_flash.d \
       ./debug/fsl_gpio.d \
       ./debug/fsl_lpsci.d \
       ./debug/fsl_smc.d \
       ./debug/fsl_uart.d \
       ./debug/fsl_rtc.d \
       ./debug/fsl_debug_console.d \
       ./debug/main.d \
       ./debug/RGBled.d

################################################################################
############
# Object files enumerated below
OBJS := \
       ./debug/startup_mkl25z4.o \
       ./debug/system_MKL25Z4.o  \
       ./debug/board.o \
       ./debug/clock_config.o \
       ./debug/peripherals.o \
       ./debug/pin_mux.o \
       ./debug/fsl_clock.o \
       ./debug/fsl_common.o \
       ./debug/fsl_flash.o \
       ./debug/fsl_gpio.o \
       ./debug/fsl_lpsci.o \
       ./debug/fsl_smc.o \
       ./debug/fsl_uart.o \
       ./debug/fsl_rtc.o \
       ./debug/fsl_debug_console.o \
       ./debug/main.o \
       ./debug/RGBled.o


################################################################################
############
# Include generated dependency files
ifneq ($(MAKECMDGOALS),clean)
ifneq ($(strip $(C_DEPS)),)
-include $(C_DEPS)
endif
endif
```

```
################################################################################
############
# Linker Options for FB_RUN ,FB_DEBUG
LL_OPTIONS := -nostdlib -Xlinker -Map="debug/PES_PROJECT2.map" \
                    -Xlinker --gc-sections -Xlinker -print-memory-usage \
                    -mcpu=cortex-m0plus -mthumb -T linkerfile.ld -o $(EXE)

################################################################################
############
# Compiler options for FB_run,FB_DEBUG
CC_OPTIONS := \
       -c \
       -std=gnu99 \
       -O0 \
       -g3 \
       -ffunction-sections \
       -fmessage-length=0\
       -fno-common \
       -fdata-sections \
       -fno-builtin \
       -mcpu=cortex-m0plus \
       -mthumb \

################################################################################
############
#all target
all: $(EXE)
       @echo "*** Yeah!!! Finished Building ***"

################################################################################
############
# Clean target
clean:
       -$(RM) $(EXECUTABLES) $(OBJS) $(EXE)
       -$(RM) ./debug/*.map
       -@echo "*******Clean Finished!!!*******"

################################################################################
############
#Rules to include all build targets
ifeq ($(RUN), fb_run)
################################################################################
############
# Build Options for fb_run
B_OPTIONS := \
       -D__REDLIB__ -DCPU_MKL25Z128VLK4 -DCPU_MKL25Z128VLK4_cm0plus -
DSDK_OS_BAREMETAL \
       -DFSL_RTOS_BM -DSDK_DEBUGCONSOLE=1 -DCR_INTEGER_PRINTF -DPRINTF_FLOAT_ENABLE=0
\
       -DSCANF_FLOAT_ENABLE=0 -DPRINTF_ADVANCED_ENABLE=0 -DSCANF_ADVANCED_ENABLE=0 \
       -D__MCUXPRESSO -D__USE_CMSIS -DDEBUG -DFRDM_KL25Z -DFREEDOM -Dfb_run -
specs=redlib.specs \
```

```
./debug/%.o: ./source/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) $(Flag) -MMD -MP -
MF"$(@:%.o=%.d)" -MT"$(@:%.o=%.o)" -MT"$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '

else ifeq ($(RUN), fb_debug)
################################################################################
############
# Build Options for fb_debug
B_OPTIONS := \
	-D__REDLIB__ -DCPU_MKL25Z128VLK4 -DCPU_MKL25Z128VLK4_cm0plus -
DSssDK_OS_BAREMETAL \
	-DFSL_RTOS_BM -DSDK_DEBUGCONSOLE=1 -DCR_INTEGER_PRINTF -DPRINTF_FLOAT_ENABLE=0
\
	-DSCANF_FLOAT_ENABLE=0 -DPRINTF_ADVANCED_ENABLE=0 -DSCANF_ADVANCED_ENABLE=0 \
	-D__MCUXPRESSO -D__USE_CMSIS -DDEBUG -DFRDM_KL25Z -DFREEDOM -Dfb_debug -
specs=redlib.specs \

./debug/%.o: ./source/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) $(Flag) -MMD -MP -
MF"$(@:%.o=%.d)" -MT"$(@:%.o=%.o)" -MT"$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '
################################################################################
############
# Build Options for pc_run
else ifeq ($(RUN), pc_run)

	$(CC)  main.c -o main.o
################################################################################
############
# Build Options for fb_debug

else ifeq ($(RUN), pc_debug)

	$(CC)  main.c -o main.o

endif

################################################################################
############

$(EXE): $(OBJS) linkerfile.ld
	@echo 'Building target: $@'
	@echo 'Invoking: Linker'
	$(LL) $(LL_OPTIONS) $(OBJS)
	@echo 'Finished building target: $@'
	@echo ' '

################################################################################
############
# Rule to build the files in the source folder
```

```
./debug/%.o: ./source/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) $(Flag)  -MMD -MP -
MF"$(@:%.o=%.d)" -MT"$(@:%.o=%.o)" -MT"$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '
################################################################################
############
# Rule to build the files in the CMSIS folder
./debug/%.o: ./board/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) -MMD -MP -MF"./$(@:%.o=%.d)" -
MT"./$(@:%.o=%.o)" -MT"./$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '

./debug/%.o: ./CMSIS/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) -MMD -MP -MF"./$(@:%.o=%.d)" -
MT"./$(@:%.o=%.o)" -MT"./$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '

./debug/%.o: ./drivers/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) -MMD -MP -MF"./$(@:%.o=%.d)" -
MT"./$(@:%.o=%.o)" -MT"./$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '

./debug/%.o: ./utilities/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) -MMD -MP -MF"./$(@:%.o=%.d)" -
MT"./$(@:%.o=%.o)" -MT"./$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '

./debug/%.o: ./startup/%.c
	@echo 'Building file: $<'
	$(CC) $(CC_OPTIONS) $(B_OPTIONS) $(INCLUDES) -MMD -MP -MF"./$(@:%.o=%.d)" -
MT"./$(@:%.o=%.o)" -MT"./$(@:%.o=%.d)" -o "$@" "$<"
	@echo 'Finished building: $<'
	@echo ' '
```