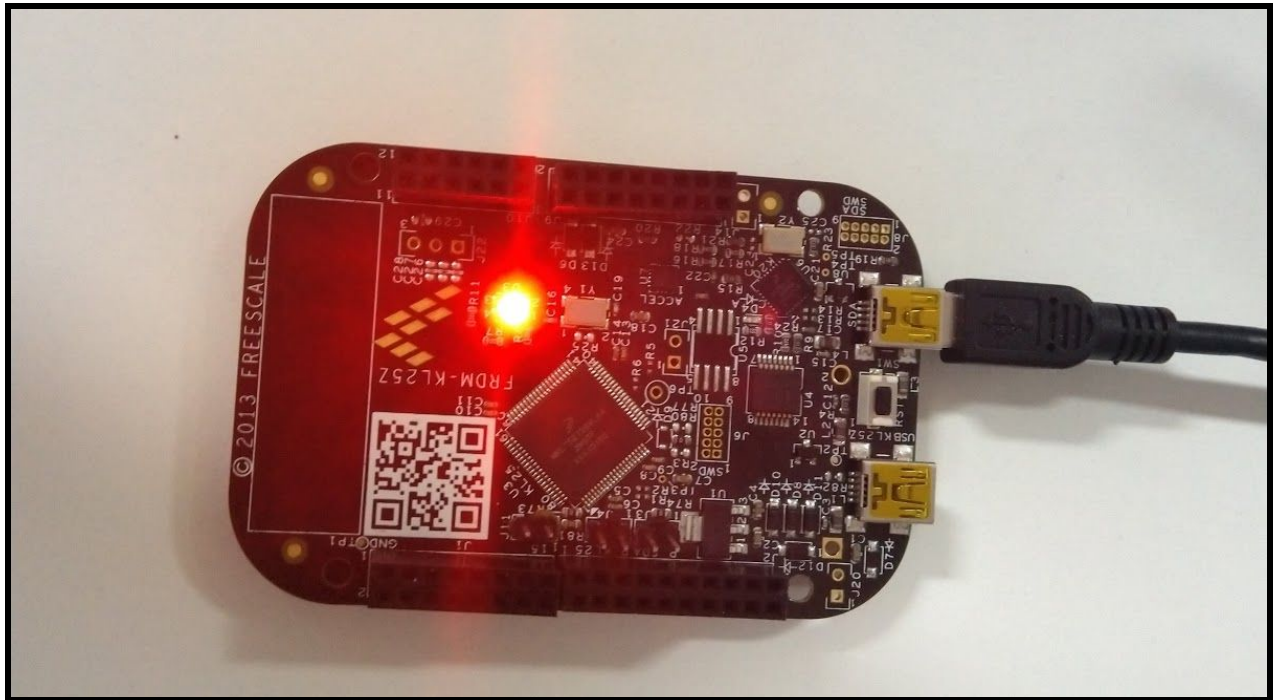


Lab Report



Embedded Systems

Himanshu Rai

111601032

30th September, 2019

Objective

To light up RGB lights on FRDM-KL25Z board in succession in a loop, using the GPIO ports.

Theory

The following tables give the correspondence between the LEDs and port pins associated with them

LED	GPIO PIN
Red	PortB18
Green	PortB19
Blue	PortD1

First we will have to follow the following procedure to configure the ports/pins corresponding to the LEDs, before using them

- By default, GPIO modules are disabled to save power. So, we should first enable the clocks to the ports corresponding to the LEDs to prevent hardware fault. The *system clocking gate control register 5* (*SIM_SCGC5*) gates the clocks to GPIO ports. We can enable the clock for these ports by setting the respective bits in *SIM_SCGC5* control registers - which are bit 10 (for port B - red and green LEDs) and bit 12 (for port D - blue LED).
- Next configure the pins for GPIO . For this we must write 001 in the corresponding *program control registers (PCR)* at bit positions 10 to 8.
- Now set the pins as output pins by setting the corresponding pin position in the *port data direction register (PDDR)* of the corresponding ports.

After setting up the pins, we can turn the LEDs on and off by setting the corresponding bits in *port clear output register (PCOR)* and *port toggle output register (PTOR)*.

Procedure

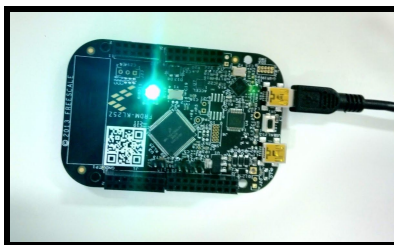
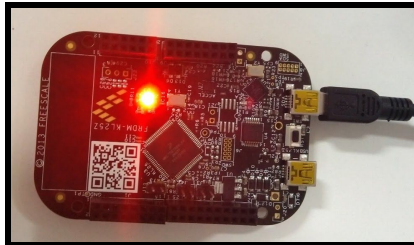
According to the steps mentioned above, it would suffice to write three functions for each LEDs - one to initialise the ports and the other two to switch the LEDs on and off. The initialisation functions for each LEDs need to be called only once, at the beginning of the program. Thereafter call the on and off functions of each LEDs in a loop to light them in succession.

Also, to clearly see each LED glowing we a suitable delay would be required between turning the LEDs on and off.

The source code is at the end of the report.

Results

The LEDs worked as desired lighting up in succession.



Source Code

```
#include<MKL25Z4.h>
```

```
void led_red_init() {  
    SIM->SCGC5 |= 1 << 10;           // will activate the port B  
    // set the pin as gpio  
    PORTB->PCR[18] &= 0xFFFFF8FF; // set 8th, 9th and 10th bit of PCR[18] to 0  
    PORTB->PCR[18] |= 1 << 8;       // set 8th bit of PCR[18] to 1  
    // set the port as output port  
    PTB->PDDR |= 1 << 18;           // set the 18th bit of PDDR to 1 for output  
}
```

```
void led_red_on() {  
    // clear the 18th bit of PDOR register  
    PTB->PCOR |= 1 << 18;           // set the 18th bit of PCOR to 1  
}
```

```
void led_red_off() {  
    // toggle the 18th bit of PDOR register  
    PTB->PTOR |= 1 << 18;  
}
```

```
void led_green_init() {  
    SIM->SCGC5 |= 1 << 10;           // will activate the port B  
    // set the pin as gpio  
    PORTB->PCR[19] &= 0xFFFFF8FF; // set 8th, 9th and 10th bit of PCR[19] to 0  
    PORTB->PCR[19] |= 1 << 8;       // set 8th bit of PCR[19] to 1  
    // set the port as output port  
    PTB->PDDR |= 1 << 19;           // set the 19th bit of PDDR to 1 for output  
}
```

```
void led_green_on() {  
    // clear the 18th bit of PDOR register  
    PTB->PCOR |= 1 << 19;           // set the 19th bit of PCOR to 1  
}
```

```
void led_green_off() {  
    // toggle the 18th bit of PDOR register  
    PTB->PTOR |= 1 << 19;  
}
```

```

void led_blue_init() {
    SIM->SCGC5 |= 1 << 12;           // will activate the port D
    // set the pin as gpio
    PORTD->PCR[1] &= 0xFFFFF8FF; // set 8th, 9th and 10th bit of PCR[1] to 0
    PORTD->PCR[1] |= 1 << 8;        // set 8th bit of PCR[1] to 1
    // set the port as output port
    PTD->PDDR |= 1 << 1;            // set the 1th bit of PDDR to 1 for output
}

```

```

void led_blue_on() {
    // clear the 18th bit of PDOR register
    PTD->PCOR |= 1 << 1;           // set the 1th bit of PCOR to 1
}

```

```

void led_blue_off() {
    // toggle the 18th bit of PDOR register
    PTD->PTOR |= 1 << 1;
}

```

```

void delay(int cycles) {
    while(cycles--);
}

```

```

int main() {
    SystemCoreClockUpdate();           // updating the clock from PLL
    led_red_init();
    led_green_init();
    led_blue_init();
    while(1) {
        led_red_on(), delay(1000000), led_red_off();
        led_green_on(), delay(1000000), led_green_off();
        led_blue_on(), delay(1000000), led_blue_off();
    }
}

```