아래는 csd_arm.S 코드입니다.

```
#define csd_LED_ADDR 0x41200000
#define csd_SWITCH_ADDR      0x41210000

.extern csd_main

.align 8

// Our interrupt vector table
csd_entry:
        b csd_reset
        b .
        b .
        b .
        b .
        b .
        b csd_irq
        b .

.global main
csd_reset:

main:

        ldr r1, =csd_SWITCH_ADDR // save the address of switch in r1
        ldr r2, =csd_LED_ADDR // save the address status of LED in r2

loop:

        ldr r0, [r1] // load the newest status of switch in r0 (also parameter is passed to csd_main automatically)
        bl csd_main // check the on duration in each iteration based on which switch is in up position

        mov r4, r0 // store the number of iteration (which is already stored in r0) also in r4

        first_loop:

                mov r3, #1 // LD0 will be turned on (based on GPIO 8-bit register)
                str r3, [r2] // LD0 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne first_loop // if times are left, then go to first_loop

        mov r0, r4 // store the number of iteration in r0 (which will be used in next loop)

        second_loop:

                mov r3, #2 // LD1 will be turned on
                str r3, [r2] // LD1 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne second_loop // if times are left, then go to second_loop

        mov r0, r4 // store the number of iteration in r0

        third_loop:

                mov r3, #4 // LD2 will be turned on
                str r3, [r2] // LD2 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne third_loop // if times are left, then go to third_loop

        mov r0, r4 // store the number of iteration in r0

        fourth_loop:

                mov r3, #8 // LD3 will be turned on
                str r3, [r2] // LD3 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne fourth_loop // if times are left, then go to fourth_loop

        mov r0, r4 // store the number of iteration in r0

        fifth_loop:

                mov r3, #16 // LD4 will be turned on
                str r3, [r2] // LD4 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne fifth_loop // if times are left, then go to fifth_loop

        mov r0, r4 // store the number of iteration in r0

        sixth_loop:

                mov r3, #32 // LD5 will be turned on
                str r3, [r2] // LD5 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne sixth_loop // if times are left, then go to sixth_loop

        mov r0, r4 // store the number of iteration in r0

        seventh_loop:

                mov r3, #64 // LD6 will be turned on
                str r3, [r2] // LD6 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne seventh_loop // if times are left, then go to seventh_loop

        mov r0, r4 // store the number of iteration in r0

        eighth_loop:

                mov r3, #128 // LD7 will be turned on
                str r3, [r2] // LD7 is turned on
                sub r0, r0, #1 // count down the remaining number of times
                cmp r0, #0 // check if LED should be turned off
                bne eighth_loop // if times are left, then go to eighth_loop

        b loop // continue iterations

// Normal Interrupt Service Routine
csd_irq:
        b .
```

아래는 csd_main.c 코드입니다.

```
/*
 * csd_main.c
 *
 *  Created on: 2019. 4. 4.
 *      Author: JJ
 */

int csd_main(unsigned char a)
{
	// if and else if statement is for priority of switches

	if (a >= 128){
		return 430000;
	} // if SW7 is in the up position, the on duration is roughly 100msc

	else if (127 >= a && a >= 64) {
		return 860000;
	} // if SW6 is in the up position, the on duration is roughly 200msc

	else if (63 >= a && a >= 32) {
		return 1290000;
	} // if SW5 is in the up position, the on duration is roughly 300msc

	else if (31 >= a && a >= 16) {
		return 1720000;
	} // if SW4 is in the up position, the on duration is roughly 400msc

	else if (15 >= a && a >= 8) {
		return 2150000;
	} // if SW3 is in the up position, the on duration is roughly 500msc

	else if (7 >= a && a >= 4) {
		return 2580000;
	} // if SW2 is in the up position, the on duration is roughly 600msc

	else if (3 >= a && a >= 2) {
		return 3010000;
	} // if SW1 is in the up position, the on duration is roughly 700msc
```

```
        else if (a == 1) {
                return 3440000;
        } // if SW0 is in the up position, the on duration is roughly 800msc

        else {
                return 4300000;
        } // otherwise, the LED's on duration is roughly 1 second
}
```