

テーマ：イベントフラグ

```

/*****
*** reset process      ***
*****/
void uinit(void)
{
    /* initialize */
    clock_switch();
    led_init();
    sw_init();
    trb_init();
    asm("FSET I");
}

/*****
*** tasks              ***
*****/
void tsk01(void) {
    while(1) {
        wai_flg(1,0x0001);
        led_set(0xff);
        tslp_tsk(500);
        led_set(0x00);
    }
}

void tsk02(void) {
    UB k=0;
    while(1) {
        k=sw_read();
        if(k!=0)
            set_flg(1,0x0001);
    }
}

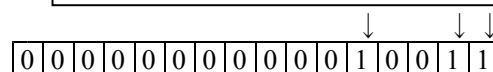
void tsk03(void) {
    while(1) {
        slp_tsk();
    }
}

```

ポイント 1 イベントフラグはタスク間の同期にフラグを使う方法である。

ポイント 2 イベントフラグは 16 ビットで構成されており、一つのイベントフラグで複数の事象の同期を取ることができる。

事象が発生したらフラグを立てる



ポイント 3 `wai_flg(fluid, waiptn)` : fluid で指定したイベントフラグに waiptn で指定したビットがセットされるまで WAITING 状態で待機する。

ポイント 4 `set_flg(fluid, setptn)` : fluid で指定したイベントフラグに setptn で指定したビットをセットする。

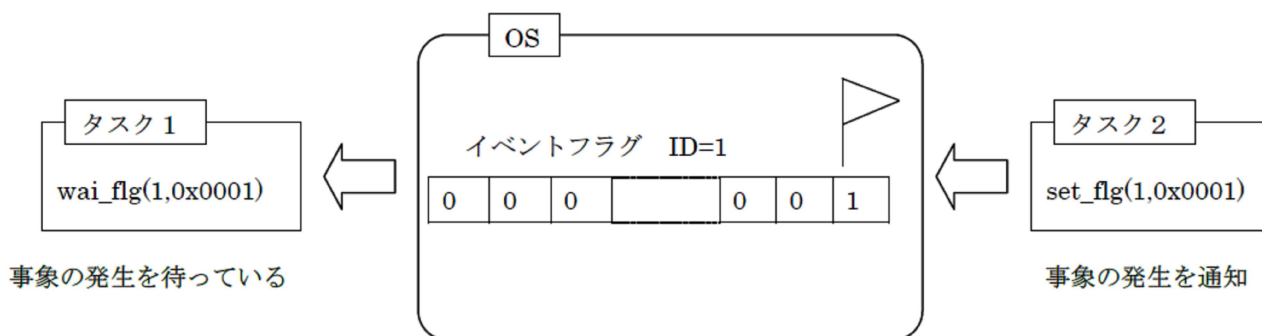
次のプログラムを、イベントフラグを用いて作成しなさい。

【実習 1】 sw が押されたらブザーが短く鳴る。

【実習 2】 1 秒間隔でブザーが短くなる。

【実習 3】 1 秒間隔で LCD に 1 ずつ増加させながら数字を表示する。

【実習 4】 sw が押された回数を LCD に表示する。



```
//実習1
/*****
/**** reset process ****
/****
void uinit(void)
{ /* initialize */
    clock_switch();
    bz_init();
    led_init();
    sw_init();
    trb_init();
    asm("FSET I");
}
/****
/**** tasks ****
/****
void tsk01(void) {
    while(1) {
        wai_flg(1,0x0001);
        short_bz();
    }
}

void tsk02(void) {
    UB k=0;
    while(1) {
        do{
            k=sw_read();
        }while(k==0);
        set_flg(1,0x0001);
        do{
            k=sw_read();
            while(k!=0);
        }
    }
}

void tsk03(void) {
    while(1) {
        slp_tsk();
    }
}
```

```
//実習2
/****
/**** reset process ****
/****
void uinit(void)
{ /* initialize */
    clock_switch();
    bz_init();
    sw_init();
    trb_init();
    asm("FSET I");
}
/****
/**** tasks ****
/****
void tsk01(void) {
    while(1) {
        wai_flg(1,0x0001);
        short_bz();
    }
}

void tsk02(void) {
    while(1) {
        tslp_tsk(1000);
        set_flg(1,0x0001);
    }
}

void tsk03(void) {
    while(1) {
        slp_tsk();
    }
}
```

//実習3

```

/*****
*** reset process      ***
*****/
void uinit(void)
{   /* initialize */
    clock_switch();
    bz_init();
    led_init();
    sw_init();
    trb_init();
    asm("FSET I");
}
/*****
*** tasks              ***
*****/

void tsk01(void) {
    UH k=0;
    lcd_clr();
    while(1) {
        lcd_xy(0,0);
        wai_flg(1,0x0001);
        lcd_int(k);
        k++;
    }
}

void tsk02(void) {
    while(1) {
        tslp_tsk(1000);
        set_flg(1,0x0001);
    }
}

void tsk03(void) {
    while(1) {
        slp_tsk();
    }
}

```

//実習4

```

/*****
*** reset process      ***
*****/
void uinit(void)
{   /* initialize */
    clock_switch();
    lcd_init();
    sw_init();
    trb_init();
    asm("FSET I");
}
/*****
*** tasks              ***
*****/

void tsk01(void) {
    UH k=1;
    lcd_clr();
    while(1) {
        lcd_xy(0,0);
        wai_flg(1,0x0001);
        lcd_int(k);
        k++;
    }
}

void tsk02(void) {
    volatile UB k;
    while(1) {
        do{
            k=sw_read();
        }while(k==0);
        set_flg(1,0x0001);
        do{
            k=sw_read();
        }while(k!=0);
    }
}

void tsk03(void) {
    while(1) {
        slp_tsk();
    }
}

```