UNIT
13

스레드

로봇SW 교육원
최상훈(shchoi82@gmail.com)

학습 목표

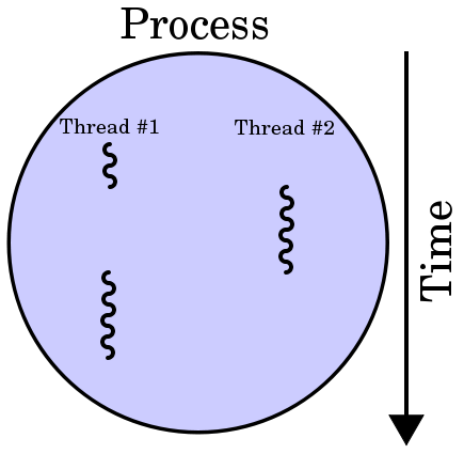
- 스레드의 이해

스레드

- 스레드(thread)
  - 프로세스내의 실행단위

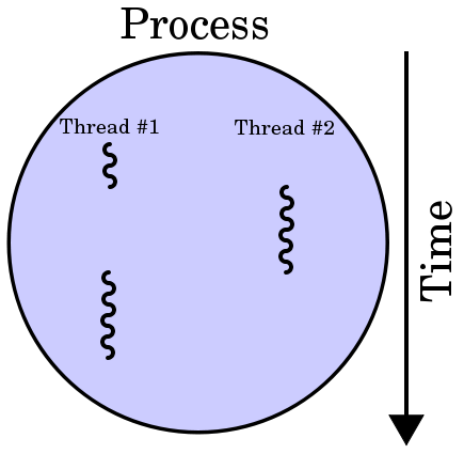
- 단일 스레드 : 프로세스내 단일 스레드로 구성
  - 일반적인 프로세스는 보통 하나의 스레드로 구성됨
  - 프로세스는 한가지 일을 순차적으로 수행

- 다중 스레드 : 프로세스내 여러 개의 스레드로 구성
  - 프로세스 실행 중 여러 개 의 스레드를 생성
  - 프로세스가 한가지 이상의 일을 동시에 수행



스레드

- 다중 스레드의 장점
  - 비동기적 이벤트 처리
  - 프로세스 정보 공유
    - 메모리 영역, 파일 디스크립터 테이블 등등
  - 처리성능 향상
  - 반응시간 향상



pthread_create


```
#include <pthread.h>
int pthread_create(pthread_t *restrict tidp,
                   const pthread_attr_t *restrict attr,
                   void *(*start_rtn)(void *), void *restrict arg);

                                   Returns: 0 if OK, error number on failure
```

- **기능** : **새로운 스레드 생성**
- tidp : **스레드** ID
- attr : **스레드 속성**, NULL - **기본속성**
- start_rtn : **스레드가 실행할 함수의 주소(함수 포인터)**
- arg : start_rtn **함수의 인자**

pthread_self


```
#include <pthread.h>
pthread_t pthread_self(void);
                                    Returns: the thread ID of the calling thread
```

- **기능 : 현재 스레드의 ID를 반환**

pthread_exit


```
#include <pthread.h>
void pthread_exit(void *rval_ptr);
```

- **기능** : **현재 스레드를 종료**
- rval_ptr : **스레드의 종료 값**
  - **종료 값은** thread_join**으로 확인**

pthread_join


```
#include <pthread.h>
int pthread_join(pthread_t thread, void **rval_ptr);
                                    Returns: 0 if OK, error number on failure
```

- **기능** : **스레드의 종료 값 회수**
- thread : **스레드** ID
- rval_ptr : **스레드의 종료 값을 회수할 주소**

실습1-1: 스레드 생성 및 ID

파일명 : threadEx1.c

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
#include <pthread.h>

pthread_t ntid;

void
printids(const char *s)
{
    pid_t pid;
    pthread_t tid;
    pid = getpid();
    tid = pthread_self();
    printf("%s pid %lu tid %lu (0x%lx)\n", s, (unsigned long)pid,
            (unsigned long)tid, (unsigned long)tid);
}

void *
thr_fn(void *arg)
{
    printids("new thread: ");
    return((void *)0);
}
```

실습1-2: 스레드 생성 및 ID


```
int
main(void)
{
    int err;
    err = pthread_create(&ntid, NULL, thr_fn, NULL);
    if (err != 0){
        fprintf(stderr, "can't create thread");
        exit(1);
    }
    printids("main thread:");
    sleep(1);
    exit(0);
}
```

```
pi@robotcode ~/ch11 $ gcc -Wall -W -lpthread threadEx1.c -o threadEx1
threadEx1.c: In function ‘thr_fn’:
threadEx1.c:21:14: warning: unused parameter ‘arg’ [-Wunused-parameter]
pi@robotcode ~/ch11 $ ./threadEx1
main thread: pid 2932 tid 3069345792 (0xb6f28000)
new thread:  pid 2932 tid 3067729008 (0xb6d9d470)
pi@robotcode ~/ch11 $
```

실습2-1: 스레드의 종료

파일명 : threadEx2.c

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
#include <pthread.h>

void *thr_fn1(void *arg);
void *thr_fn2(void *arg);

int
main(void)
{
    int err;
    pthread_t tid1, tid2;
    void *tret;
    err = pthread_create(&tid1, NULL, thr_fn1, NULL);
    if (err != 0){
        fprintf(stderr, "can't create thread 1");
        exit(1);
    }
    err = pthread_create(&tid2, NULL, thr_fn2, NULL);
    if (err != 0){
        fprintf(stderr, "can't create thread 2");
        exit(1);
    }
```

실습2-2: 스레드의 종료


```
    err = pthread_join(tid1, &tret);
    if (err != 0){
        fprintf(stderr, "can't join with thread 1");
        exit(1);
    }
    printf("thread 1 exit code %ld\n", (long)tret);
    err = pthread_join(tid2, &tret);
    if (err != 0){
        fprintf(stderr, "can't join with thread 2");
       exit(1);
    }
    printf("thread 2 exit code %ld\n", (long)tret);
    exit(0);
}

void *
thr_fn1(void *arg)
{
    printf("thread 1 returning\n");
    return((void *)1);
}
```

실습2-3: 스레드의 종료


```
void *
thr_fn2(void *arg)
{

    printf("thread 2 exiting\n");
    pthread_exit((void *)2);
}
```

```
pi@robotcode ~/ch11 $ gcc -Wall -W -lpthread threadEx2.c -o threadEx2
threadEx2.c: In function 'thr_fn1':
threadEx2.c:42:15: warning: unused parameter 'arg' [-Wunused-parameter]
threadEx2.c: In function 'thr_fn2':
threadEx2.c:49:15: warning: unused parameter 'arg' [-Wunused-parameter]
pi@robotcode ~/ch11 $ ./threadEx2
thread 2 exiting
thread 1 returning
thread 1 exit code 1
thread 2 exit code 2
pi@robotcode ~/ch11 $
```

실습3-1: 스레드의 인자전달

파일명 : threadEx3.c

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
#include <pthread.h>

struct foo {
    int a, b, c, d;
};

void
printfoo(const char *s, const struct foo *fp)
{
    printf("%s", s);
    printf(" structure at 0x%lx\n", (unsigned long)fp);
    printf(" foo.a = %d\n", fp->a);
    printf(" foo.b = %d\n", fp->b);
    printf(" foo.c = %d\n", fp->c);
    printf(" foo.d = %d\n", fp->d);
}
```

실습3-2: 스레드의 인자전달


```
void *
thr_fn1(void *arg)
{
    struct foo foo = {1, 2, 3, 4};
    printfoo("thread 1:\n", &foo);
    pthread_exit((void *)&foo);
}

void *
thr_fn2(void *arg)
{
    printf("thread 2: ID is %lu\n", (unsigned long)pthread_self());
    pthread_exit((void *)0);
}
```

실습3-3: 스레드의 인자전달


```
int
main(void)
{
    int err;
    pthread_t tid1, tid2;
    struct foo *fp;
    err = pthread_create(&tid1, NULL, thr_fn1, NULL);
    if (err != 0)
        fprintf(stderr, "can't create thread 1");
    err = pthread_join(tid1, (void *)&fp);
    if (err != 0)
        fprintf(stderr, "can't join with thread 1");
    sleep(1);
    printf("parent starting second thread\n");

    err = pthread_create(&tid2, NULL, thr_fn2, NULL);
    if (err != 0)
        fprintf(stderr, "can't create thread 2");
    sleep(1);
    printfoo("parent:\n", fp);
    exit(0);
}
```

실습3-4: 스레드의 인자전달


```
pi@robotcode ~/ch11 $ gcc -Wall -W -lpthread threadEx3.c -o threadEx3
threadEx3.c: In function 'thr_fn1':
threadEx3.c:23:15: warning: unused parameter 'arg' [-Wunused-parameter]
threadEx3.c: In function 'thr_fn2':
threadEx3.c:31:15: warning: unused parameter 'arg' [-Wunused-parameter]
pi@robotcode ~/ch11 $ ./threadEx3
thread 1:
 structure at 0xb6da1de0
 foo.a = 1
 foo.b = 2
 foo.c = 3
 foo.d = 4
parent starting second thread
thread 2: ID is 3067749488
parent:
 structure at 0xb6da1de0
 foo.a = -1225992320
 foo.b = -1226149360
 foo.c = 0
 foo.d = -1227217704
pi@robotcode ~/ch11 $
```

미션 1

- **실습 3 프로그램의 문제점이 무엇인지 분석하고 프로그램을 올바르게 수정하시오.**

time / localtime


```
#include <time.h>
time_t time(time_t *calptr);
                                        Returns: value of time if OK, -1 on error
```

- **기능** : 1970년 1월 00:00:00 **이후로 지난 시간(초)**

```
#include <time.h>
struct tm *localtime(const time_t *calptr)
                        Both return: pointer to broken-down time, NULL on error
```

- **기능** : time_t **의 시간, 날짜정보를 추출해** tm **구조체로 변환**

tm 구조체


```
struct tm { /* a broken-down time */
    int tm_sec; /* seconds after the minute: [0 - 60] */
    int tm_min; /* minutes after the hour: [0 - 59] */
    int tm_hour; /* hours after midnight: [0 - 23] */
    int tm_mday; /* day of the month: [1 - 31] */
    int tm_mon; /* months since January: [0 - 11] */
    int tm_year; /* years since 1900 */
    int tm_wday; /* days since Sunday: [0 - 6] */
    int tm_yday; /* days since January 1: [0 - 365] */
    int tm_isdst; /* daylight saving time flag: <0, 0, >0 */
};
```

실습4: 현재 시간 출력하기(sec)

파일명 : time1.c

```
#include <stdio.h>
#include <time.h>

int
main(void)
{
    time_t t;
    struct tm *tmp;

    setbuf(stdout, NULL);

    while(1){
        time(&t);
        tmp = localtime(&t);
        printf("%02d:%02d:%02d\r", tmp->tm_hour, tmp->tm_min, tmp->tm_sec);
    }
    return 0;
}
```

clock_gettim


```
#include <sys/time.h>
int clock_gettim(clockid_t clock_id, struct timespec *tsp);
                                                Returns:0 if OK, -1 on ERROR

#include <sys/time.h>
int clock_getres(clockid_t clock_id, struct timespec *tsp);
                                              Returns: 0 if OK, -1 on error
```

- clock_gettim : **시간정보**
- clock_getres : **시간정보의 정밀도**
- struct timespec **구조체**

```
struct timespec {
    time_t    tv_sec;         /* seconds */
    long      tv_nsec;        /* nanoseconds */
};
```

- **컴파일시 rt 라이브러리 옵션 추가**
  - Ex) **gcc -Wall -W time2.c -o time2 -lrt**

실습5-1: 현재 시간 출력하기(nanosec)

파일명 : time2.c

```
#include <stdio.h>
#include <time.h>
#include <sys/time.h>

int
main(void)
{
    struct timespec tsp;
    struct tm *tmp;

    setbuf(stdout, NULL);

    clock_getres(CLOCK_REALTIME, &tsp);
    printf("the resolution of the clock(tv_nsec):%ld\n", tsp.tv_nsec);

    while(1){
        clock_gettime(CLOCK_REALTIME, &tsp);
        tmp = localtime(&tsp.tv_sec);
        printf("%02d:%02d:%02d:%02ld\r", tmp->tm_hour, tmp->tm_min, tmp-
>tm_sec, tsp.tv_nsec/10000000);
    }
    return 0;
}
```

실습5-2


```
pi@robotcode ~ $ gcc -Wall -W time2.c -o time2 -lrt
pi@robotcode ~ $ ./time2
the resolution of the clock(tv_nsec):1
16:27:11:17
```

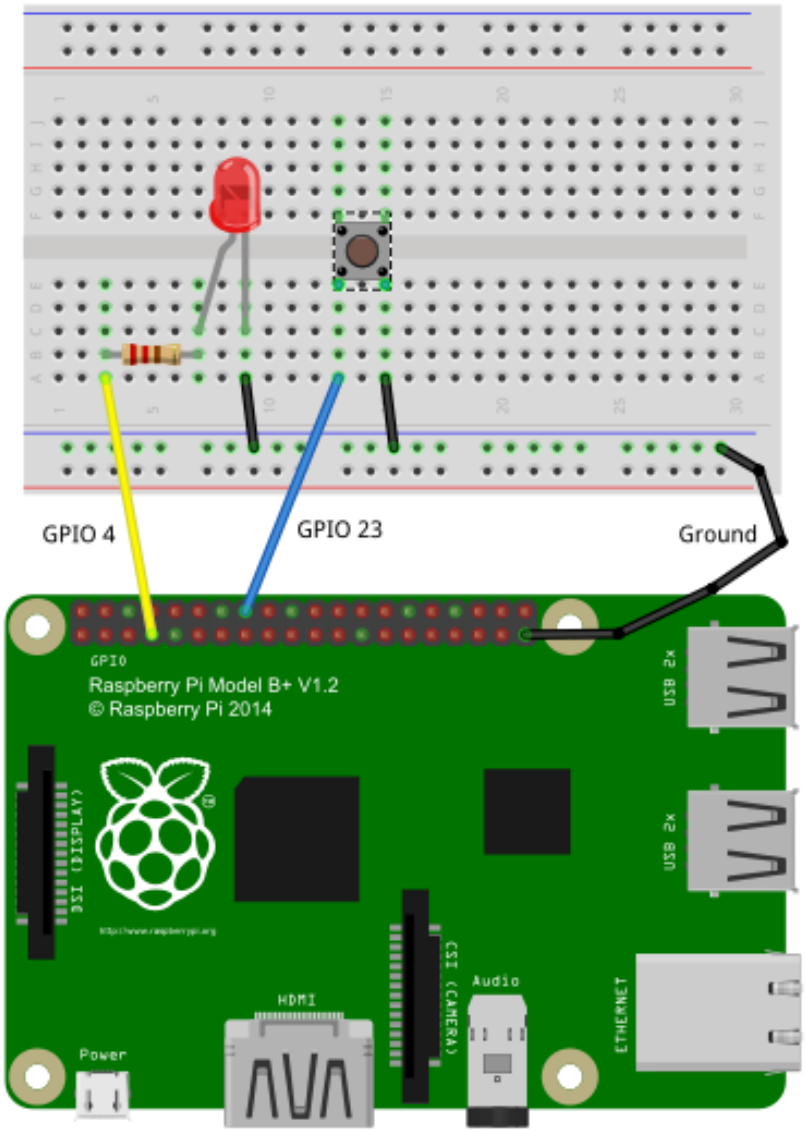
sleep / usleep


```
#include <unistd.h>
unsigned int sleep(unsigned int seconds);
                                        Returns: 0 or number of unslept seconds
```

```
#include <unistd.h>
int usleep(useconds_t usec);
                                        Returns: 0 on success, -1 on error.
```

실습6-1 : 타이머

- 구성
  - **스위치**(GPIO 23)
    - **내부 풀업**(pull up) **저항 사용**
  - LED(GPIO 4)



실습6-2

파일명 : timer.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <pthread.h>
#include <time.h>
#include <unistd.h>
#include <sys/time.h>

#define STARTSW      23
#define FINISHLED    4

int isOn = 0;
struct timespec timerts;
struct timespec startts;     // 시작 시점
struct timespec curts;       // 현재 시점

pthread_t ntid;      // thread ID

// struct timespec 시간 비교
struct timespec diffTimespec(struct timespec t1, struct timespec t2);
```

실습6-3

파일명 : timer.c

```
void *
thread_startbtn(void *arg)
{
    while(1){
        if(digitalRead(STARTSW) == 0){
            if(isOn == 1) continue;
            clock_gettime(CLOCK_REALTIME, &startts);     // 시작 시간 저장
            isOn = 1;
            printf("\nstart!!\n");
        }
    }
    return ((void *)0);
}
```

실습6-4

파일명 : timer.c

```
int
main(int argc, char *argv[])
{
    struct timespec tmpts;
    int i, err;

    if(argc < 1){
        printf("usage : a.out second\n");
        return 1;
    }

    setbuf(stdout, NULL);

    if(wiringPiSetupGpio() == -1) {          // wiringPi 초기화
        printf("wiringPiSetupGpio error\n");
        return 1;
    }
    pinMode(STARTSW, INPUT);
    pinMode(FINISHLED, OUTPUT);
    pullUpDnControl(STARTSW, PUD_UP);    // 내부 pull UP 저항 ON

    timerts.tv_sec = atoi(argv[1]);

    printf("%ld seconds timer\n", timerts.tv_sec);
```

실습6-5

파일명 : timer.c

```
    err = pthread_create(&ntid, NULL, thread_startbtn, NULL);

    if (err != 0){
        fprintf(stderr, "can't create thread");
        exit(1);
    }

    while(1){
        if(isOn){
            clock_gettime(CLOCK_REALTIME, &curts);
            tmpts = diffTimespec(curts, startts);
            printf("%3ld seconds\r", timerts.tv_sec - tmpts.tv_sec);

            if(tmpts.tv_sec >= timerts.tv_sec){
                isOn = 0;
                printf("\ncomplete!!\n");
                for(i = 0 ; i < 10 ; i++){
                    digitalWrite(FINISHLED, HIGH);
                    usleep(100000);
                    digitalWrite(FINISHLED, LOW);
                    usleep(100000);
                }
            }
        }
        else
            printf("%3ld seconds\r", timerts.tv_sec);
    }
    return 0;
}
```

실습6-6

파일명 : timer.c

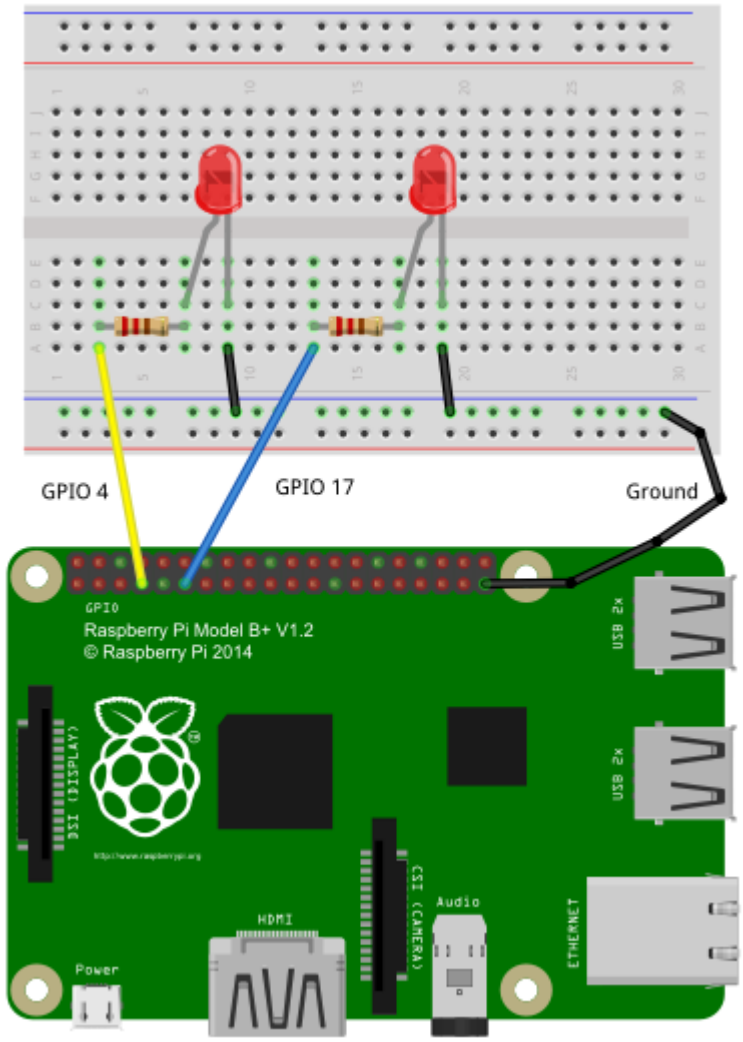
```
struct timespec
diffTimespec(struct timespec t1, struct timespec t2)
{
    struct timespec diff = {0,0l};

    // t1 - t2
    if(t1.tv_sec > t2.tv_sec)
        diff.tv_sec = t1.tv_sec - t2.tv_sec;
    else  // t2 - t1
        diff.tv_sec = t2.tv_sec - t1.tv_sec;


    return diff;
}
```

실습6-7


```
pi@robotcode ~ $ gcc -Wall -W -lrt -lpthread -lwiringPi timer.c -o timer
timer.c: In function ‘thread_startbtn’:
timer.c:23:23: warning: unused parameter ‘arg’ [-Wunused-parameter]
pi@robotcode ~ $ sudo ./timer 7
7 seconds timer
  7 seconds
start!!
  0 seconds
complete!!
  7 seconds
```

실습7-1 : LED

- 구성
  - LED(GPIO 4)
  - LED(GPIO 17)



실습7-2

파일명 : threadEx4.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include <sys/time.h>
#include <pthread.h>
#include <wiringPi.h>

#define LED1    4
#define LED2    17

void *thr_fn1(void *arg);
void *thr_fn2(void *arg);

struct timespec diffTimespec(struct timespec t1, struct timespec t2);

int
main(void)
{
    int err;
    pthread_t tid1, tid2;
    struct timespec tsp;
    struct tm *tmp;

    setbuf(stdout, NULL);

    if(wiringPiSetupGpio() == -1) {          // wiringPi 초기화
        printf("wiringPiSetupGpio error\n");
        return 1;
    }
```

실습7-3

파일명 : threadEx4.c

```
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);

    clock_getres(CLOCK_REALTIME, &tsp);
    printf("the resolution of the clock(tv_nsec):%ld\n", tsp.tv_nsec);

    err = pthread_create(&tid1, NULL, thr_fn1, NULL);
    if (err != 0){
        fprintf(stderr, "can't create thread 1");
        exit(1);
    }
    err = pthread_create(&tid2, NULL, thr_fn2, NULL);
    if (err != 0){
        fprintf(stderr, "can't create thread 2");
        exit(1);
    }

    while(1){

        clock_gettime(CLOCK_REALTIME, &tsp);
        tmp = localtime(&tsp.tv_sec);
        printf("%02d:%02d:%02d:%02ld\r", tmp->tm_hour, tmp->tm_min, tmp-
>tm_sec, tsp.tv_nsec/10000000);
    }
    return 0;
}
```

실습7-4

파일명 : threadEx4.c

```
void * thr_fn1(void *arg)
{
    while(1){
        digitalWrite(LED1, LOW);
        sleep(1);
        digitalWrite(LED1, HIGH);
        sleep(1);
    }
    return((void *)0);
}

void * thr_fn2(void *arg)
{
    struct timespec startts;
    struct timespec curts;
    struct timespec diffts;
    int ledStatus = 0;

    clock_gettime(CLOCK_REALTIME, &startts);

    while(1){
        clock_gettime(CLOCK_REALTIME, &curts);
        diffts = diffTimespec(startts, curts);
        if(diffts.tv_sec >= 1){
            ledStatus = ledStatus == 0 ? 1 : 0;
            digitalWrite(LED2, ledStatus);
            startts.tv_sec += 1;
        }
    }

    return((void *)0);
}
```

실습7-5

파일명 : threadEx4.c

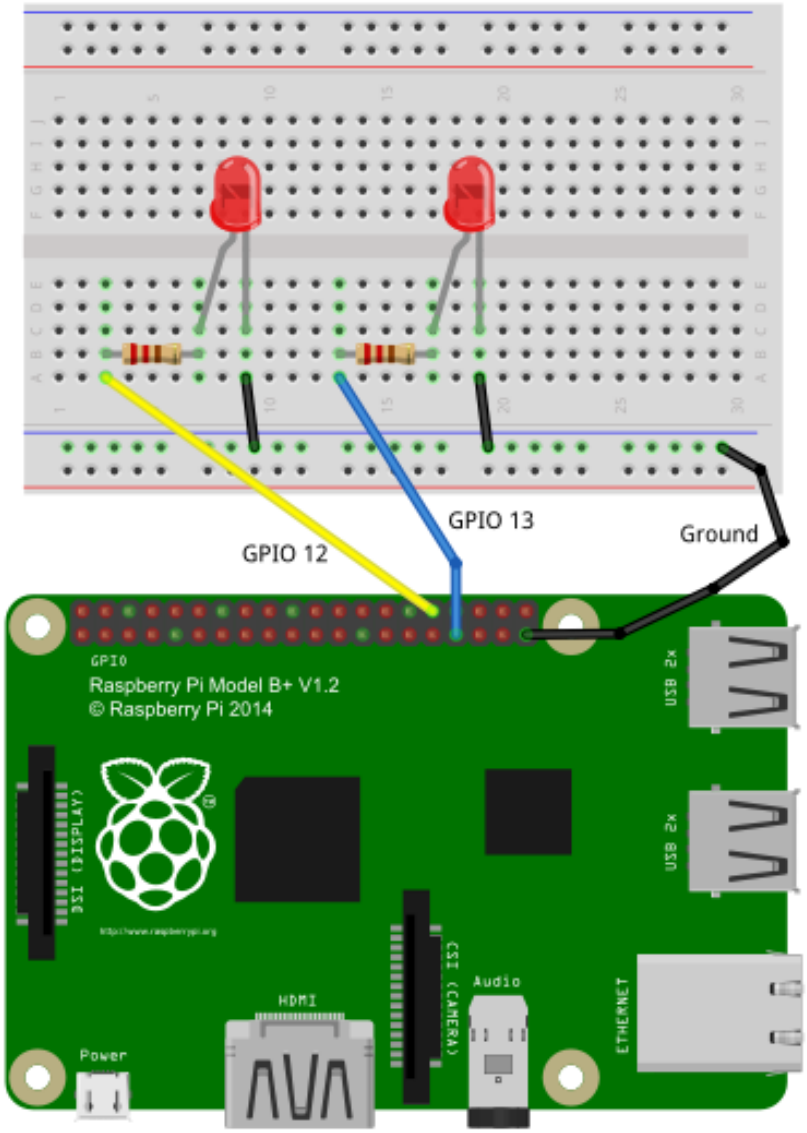
```
struct timespec
diffTimespec(struct timespec t1, struct timespec t2)
{
    struct timespec diff = {0,0l};

    // t1 - t2
    if(t1.tv_sec > t2.tv_sec)
        diff.tv_sec = t1.tv_sec - t2.tv_sec;
    else  // t2 - t1
        diff.tv_sec = t2.tv_sec - t1.tv_sec;


    return diff;
}
```

실습7-6


```
pi@robotcode ~ $ gcc -Wall -W -lrt -lpthread -lwiringPi threadEx4.c -o
threadEx4
threadEx4.c: In function 'thr_fn1':
threadEx4.c:59:15: warning: unused parameter 'arg' [-Wunused-parameter]
threadEx4.c: In function 'thr_fn2':
threadEx4.c:71:15: warning: unused parameter 'arg' [-Wunused-parameter]
pi@robotcode ~ $ sudo ./threadEx4
the resolution of the clock(tv_nsec):1
13:32:30:80
```

실습8-1 : PWM

- 구성
  - LED(GPIO 12 pwm0)
  - LED(GPIO 13 pwm1)



실습8-2

파일명 : threadEx5.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <wiringPi.h>

#define LED1    12
#define LED2    13

void *thr_fn1(void *arg);
void *thr_fn2(void *arg);
int pwm_i;
int pwm_j;

struct timespec diffTimespec(struct timespec t1, struct timespec t2);

int
main(void)
{
    int err;
    pthread_t tid1, tid2;

    setbuf(stdout, NULL);

    if(wiringPiSetupGpio() == -1) {          // wiringPi 초기화
        printf("wiringPiSetupGpio error\n");
        return 1;
    }
```

실습8-3

파일명 : threadEx5.c

```
    pinMode(LED1, PWM_OUTPUT);
    pinMode(LED2, PWM_OUTPUT);

    err = pthread_create(&tid1, NULL, thr_fn1, NULL);
    if (err != 0){
        fprintf(stderr, "can't create thread 1");
        exit(1);
    }
    err = pthread_create(&tid2, NULL, thr_fn2, NULL);
    if (err != 0){
        fprintf(stderr, "can't create thread 2");
        exit(1);
    }

    while(1)
        printf("pwm_i:%3d, pwm_j:%3d\r", pwm_i, pwm_j);

    return 0;
}
```

실습8-4

파일명 : threadEx5.c

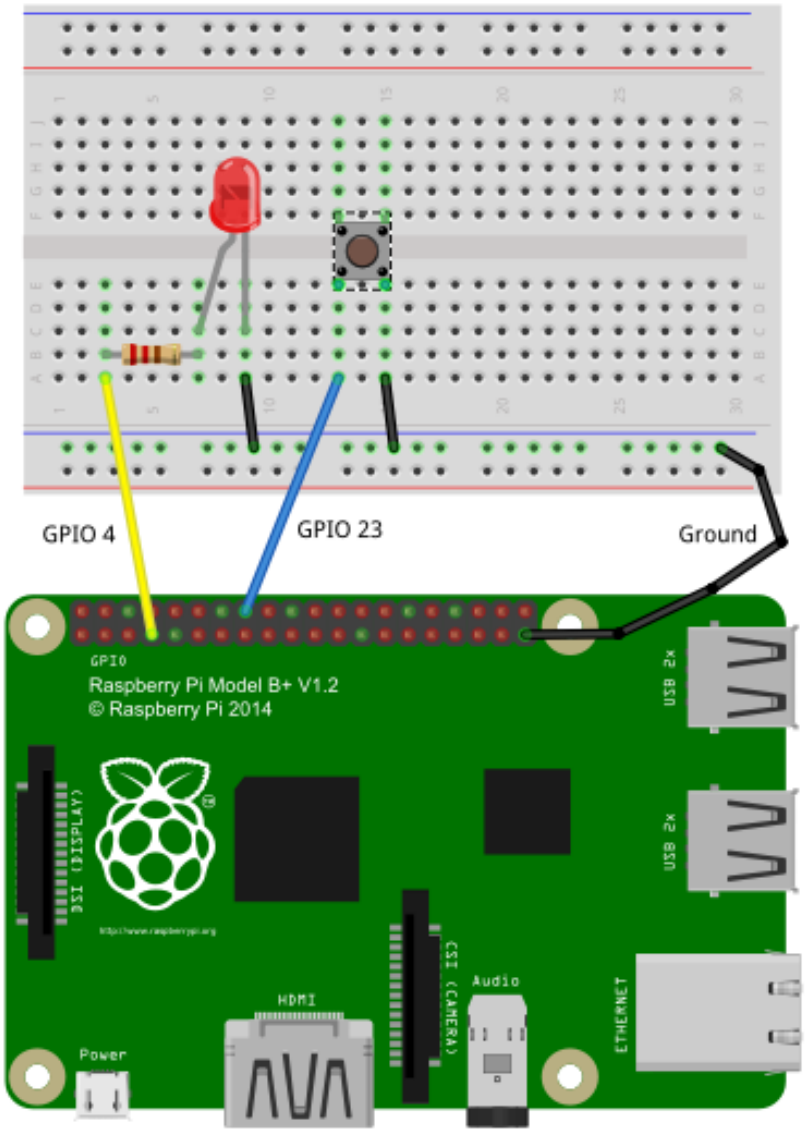
```
void *
thr_fn1(void *arg)
{
    while(1){
        for(pwm_i = 0 ; pwm_i < 500 ; pwm_i++){
            pwmWrite(LED1, pwm_i);
            usleep(2000);
        }
        for(; pwm_i > 0 ; pwm_i--){
            pwmWrite(LED1,pwm_i);
            usleep(2000);
        }
    }
    return((void *)0);
}

void *
thr_fn2(void *arg)
{
    while(1){
        for(pwm_j = 0 ; pwm_j < 500 ; pwm_j++){
            pwmWrite(LED2, pwm_j);
            usleep(3000);
        }
        for(; pwm_j > 0 ; pwm_j--){
            pwmWrite(LED2, pwm_j);
            usleep(3000);
        }
    }

    return((void *)0);
}
```

실습8-5


```
pi@robotcode ~ $ gcc -Wall -W -lpthread -lwiringPi threadEx5.c -o threadEx5
threadEx5.c: In function 'main':
threadEx5.c:25:13: warning: unused variable 'tmp' [-Wunused-variable]
threadEx5.c: In function 'thr_fn1':
threadEx5.c:58:15: warning: unused parameter 'arg' [-Wunused-parameter]
threadEx5.c: In function 'thr_fn2':
threadEx5.c:74:15: warning: unused parameter 'arg' [-Wunused-parameter]
pi@robotcode ~ $ sudo ./threadEx5
the resolution of the clock(tv_nsec):1
pwm_i: 59, pwm_j:389
```

실습9-1 : 스위치로 스레드 시작

- 구성
  - **스위치**(GPIO 23)
    - **내부 풀업**(pull up) **저항 사용**
  - LED(GPIO 4)



실습9-2

파일명 : threadEx6.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <wiringPi.h>

#define LED     4
#define SW      23

void *thr_fn(void *arg);

struct timespec diffTimespec(struct timespec t1, struct timespec t2);

int
main(void)
{
    int err;
    int isPushed = 0;
    pthread_t tid;

    if(wiringPiSetupGpio() == -1) {          // wiringPi 초기화
        printf("wiringPiSetupGpio error\n");
        return 1;
    }
    pinMode(LED, OUTPUT);
    pinMode(SW, INPUT);
    pullUpDnControl(SW, PUD_UP);    // 내부 pull UP 저항 ON
```

실습9-3

파일명 : threadEx6.c

```
    while(1){
        if(digitalRead(SW) == 0 && isPushed == 0){
            err = pthread_create(&tid, NULL, thr_fn, NULL);
            if (err != 0){
                fprintf(stderr, "can't create thread 1");
                exit(1);
            }
            isPushed = 1;
        }else if(digitalRead(SW) == 1){
            isPushed = 0;
        }

    }
    return 0;
}
```

실습9-4

파일명 : threadEx6.c

```
void *
thr_fn(void *arg)
{
    int i = 0;
    pthread_t tid;

    tid = pthread_self();
    printf("(0x%lx) thread start!\n", (unsigned long)tid);

    for(i = 0 ; i < 10 ; i++){
        digitalWrite(LED, HIGH);
        printf("(0x%lx) LED ON!\n", (unsigned long)tid);
        usleep(100000);
        digitalWrite(LED, LOW);
        printf("(0x%lx) LED OFF!\n", (unsigned long)tid);
        usleep(100000);
    }
    printf("(0x%lx) thread end!\n", (unsigned long)tid);
    return((void *)0);
}
```

실습9-5

```
pi@robotcode ~ $ gcc -Wall -W -lpthread -lwiringPi threadEx6.c -o threadEx6
threadEx6.c: In function 'thr_fn':
threadEx6.c:46:14: warning: unused parameter 'arg' [-Wunused-parameter]
pi@robotcode ~ $ sudo ./threadEx6
(0xb6d7b470) thread start!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) LED ON!
(0xb6d7b470) LED OFF!
(0xb6d7b470) thread end!
```

미션2

- 스위치 2개를 이용해 스톱 워치를 구현하시오.