2015104124
전자공학과
진우빈

IoT 소프트웨어 과제9

예제 1)

Motor 모듈을 이용하여 모터의 방향을 왼쪽, 오른쪽으로 달리하여 회전하는 예제입니다. 하드웨어 구성은 Raspberry Pi Adapter와 Motor모듈을 10핀을 이용해서 연결하였습니다.

코드에서는 먼저 모터의 PWM제어를 하기 위해서 WiringPi에 포함된 softPwm헤더를 사용하였습니다. 또한 왼쪽으로 회전하게 되는 모터 핀은 MOTOR_NPIN, 오른쪽으로 회전하게 되는 모터 핀은 MOTOR_PPIN으로 define하고 해당 GPIO핀 번호를 설정하였습니다.
MotorStop함수는 함수 Pulse Width를 0으로 write하여 Duty Cycle이 0이 되도록 만드는 함수입니다. 즉 모터는 작동하지 않게됩니다.
MotorControl은 parameter로 회전방향의 값을 받게됩니다. 만약 1을 입력받으면 MOTOR_NPIN은 50만큼의 Pulse Width를 출력하여 작동하게 되고 MOTOR_PPIN은 출력하지 않습니다. 즉 왼쪽으로 회전할 때입니다. 2를 입력받으면 반대로 셋팅이되고 오른쪽으로 회전하게 됩니다.

main문에서 Motor모듈에 연결된 핀들을 모두 출력핀으로 바꿔주었으며 softPwmCreate 함수를 사용하여 주기를 100으로 설정하여 pwm을 생성하였습니다. MotorContorl함수에서 softPwmWrite를 사용하여 50의 값을 입력받았으므로 모터가 동작할 때의 Duty Cycle = 0.5 * 100 으로 50프로의 출력을 생성하게 됩니다. while문 안에서는 delay를 사용하여 2초간 모터를 돌리고 1초간 멈추는 것을 반복하였습니다.
컴파일 후 실행한 결과 모터가 한번은 왼쪽 그 다음은 오른쪽으로 번갈아가면서 정해진 시간동안 돌아가고 멈추는 것을 확인할 수 있었습니다.

lab2-5_1.c

```c
#include <wiringPi.h>
#include <softPwm.h>

#define MOTOR_NPIN 17
#define MOTOR_PPIN 4

#define LEFT 1
#define RIGHT 2

void MotorStop() {
    softPwmWrite(MOTOR_NPIN, 0);
    softPwmWrite(MOTOR_PPIN, 0);
}

void MotorControl(int rotate) {
    if(rotate==LEFT) {
        digitalWrite(MOTOR_PPIN, LOW);
        softPwmWrite(MOTOR_NPIN, 50);
    }

    else if(rotate==RIGHT) {
        digitalWrite(MOTOR_NPIN, LOW);
        softPwmWrite(MOTOR_PPIN, 50);
    }
}

int main(void) {
    if(wiringPiSetupGpio()==-1) return 1;

    pinMode(MOTOR_NPIN, OUTPUT);
    pinMode(MOTOR_PPIN, OUTPUT);

    softPwmCreate(MOTOR_NPIN, 0, 100);
    softPwmCreate(MOTOR_PPIN, 0, 100);

    while(1) {

        MotorControl(LEFT);
        delay(2000);
        MotorStop();
        delay(1000);

        MotorControl(RIGHT);
        delay(2000);
        MotorStop();
        delay(1000);
    }

    return 0;
}
```

<lab2-5_1.c 코드>

예제2)

모터의 방향이 아니라 속도를 주기적으로 변경하는 예제입니다. Motor를 정지하는 MotorStop함수는 예제1과 같으나 MotorControl함수에서는 위와 다르게 parameter값이 speed입니다. 매개변수로 전달받은 speed값에 의해 PulseWidth가 바뀌게 되는 것입니다.

main문 안에서 Motor핀을 모두 출력으로 하고, softPwmCreate함수를 써서 주기를 100으로 설정하였습니다.
while문 안에서 처음엔 25 그 다음엔 50 마지막으로 75의 값으로 MotorControl 함수를 2초동안 실행합니다. 즉 주기가 100이므로 각각 1/4, 1/2, 3/4의 출력으로 모터가 돌아가게 되는 것입니다. 방향은 MotorControl에서 오른쪽 회전출력은 하지 않도록 하기 때문에 왼쪽으로 모터가 돌아가게 됩니다.
컴파일 후 실행 결과 3번에 걸쳐서 모터가 점점 빠르게 돌아감을 반복하는 것을 확인할 수 있었습니다.

lab2-5_1.c lab2-5_2.c

```c
#include <wiringPi.h>
#include <softPwm.h>

#define MOTOR_NPIN 17
#define MOTOR_PPIN 4

void MotorStop() {
    softPwmWrite(MOTOR_NPIN, 0);
    softPwmWrite(MOTOR_PPIN, 0);
}

void MotorControl(int speed) {
    digitalWrite(MOTOR_PPIN, LOW);
    softPwmWrite(MOTOR_NPIN, speed);
}

int main(void) {
    if(wiringPiSetupGpio()==-1) return 1;

    pinMode(MOTOR_NPIN, OUTPUT);
    pinMode(MOTOR_PPIN, OUTPUT);

    softPwmCreate(MOTOR_NPIN, 0, 100);
    softPwmCreate(MOTOR_PPIN, 0, 100);

    while(1) {

        MotorControl(25);
        delay(2000);
        MotorStop();
        delay(2000);

        MotorControl(50);
        delay(2000);
        MotorStop();
        delay(2000);

        MotorControl(75);
        delay(2000);
        MotorStop();
        delay(2000);
    }

    return 0;
}
```

<lab2-4_2.c 코드>

과제 1)

예제1, 예제2에서 했던 것을 동시에 하는 실습입니다. 즉, 모터의 회전방향과 속도제어를 같이 하는 것입니다. 따라서 MotorControl함수에 두 개의 인자를 받도록 하였습니다. 첫 번째 변수인 rotate로 회전의 방향을 제어하고 두 번째 변수인 speed로 PulseWidth를 생성합니다. 예제1, 예제2의 MotorControl을 합친 것입니다.

주기가 128ms이므로 1/4씩 PulseWidth를 늘려가고 줄일거기 때문에 WIDTH값을 32로 정의하였습니다. while문 안에서는 변수 s를 받아 4번 씩 for문을 돌아가면서 s값을 32씩 증가시킵니다. 따라서 왼쪽으로 모터가 돌아갈 때는 25%-50%-75%-100%로 Duty Cycle이 증가하면서 모터가 2초간 회전하고 2초간 정지하게 됩니다. 반대로 오른쪽으로 돌아갈 때는 s값을 32씩 빼주어 반대로 Duty Cycle이 줄어들면서 모터가 작동하게 됩니다.

컴파일 후 실행결과 약 32초 동안의 주기로 반복문이 작동하는 것을 확인하였습니다. 4초간 8번 실행되기 때문입니다. 또한 Cycle이 증가하다가 감소하면서 회전의 방향이 바뀌는 것을 확인할 수 있었습니다.

```c
void MotorControl(int rotate, int speed) {
    if(rotate==LEFT) {
        digitalWrite(MOTOR_PPIN, LOW);
        softPwmWrite(MOTOR_NPIN, speed);
    }

    else if(rotate==RIGHT) {
        digitalWrite(MOTOR_NPIN, LOW);
        softPwmWrite(MOTOR_PPIN, speed);
    }
}

int main(void) {
    if(wiringPiSetupGpio()==-1) return 1;

    pinMode(MOTOR_NPIN, OUTPUT);
    pinMode(MOTOR_PPIN, OUTPUT);

    softPwmCreate(MOTOR_NPIN, 0, 128);
    softPwmCreate(MOTOR_PPIN, 0, 128);

    while(1) {
        int s = 0;
        int i;
        for(i=0; i<4; i++) {
            s+=WIDTH;
            MotorControl(LEFT, s);
            delay(2000);
            MotorStop();
            delay(2000);
        }

        for(i=0; i<4; i++) {
            MotorControl(RIGHT, s);
            delay(2000);
            MotorStop();
            delay(2000);
            s-=WIDTH;
        }
    }

    return 0;
}
```

<ass2-4_1.c의 의 main문과 MotorControl 함수>

과제 2)

모터가 아니라 Led로 Pwm을 발생시키는 실습입니다. Motor 모듈대신 Led 모듈을 연결하였습니다. 또한 Led 모듈의 첫 번째 Led가 연결된 GPIO 4번 핀을 출력핀으로 설정하였습니다. LedControl이라는 함수를 만들어서 모터에서 속도를 매개변수로 받은 것처럼 밝기 값을 매개변수로 받게 하였습니다.

주기를 500ms로 설정하였고 while문 안에서는 점차적으로 밝기가 증가하는 것을 확인하기 위해서 for문에서 500번 돌면서 PulseWidth값을 점차적으로 증가시켰습니다. 초기의 Pulsewidth값은 변수 ledlight값으로 0이지만 for문이 반복되면서 Pulsewidth값이 5씩 증가하게 됩니다. 즉 500*5의 2.5초동안 점차적으로 PulseWidth값이 증가하면서 Duty Cycle이 100%까지 가게됩니다. 전체 출력의 100%가 되면 불을 1초 동안 키고 이후 반대로 Pulsewidth 값을 줄여나가게 됩니다. Pulsewidth값이 0이되면 마찬가지로 1초 동안 불을 끄게 설정하였습니다.

컴파일 후 실행결과 불이 2.5초간 서서히 켜지고 이후 서서히 꺼지는 것을 반복함을 확인할 수 있었습니다.

lab2-5_1.c | lab2-5_2.c | ass2-5_1.c | ass2-5_2.c

```c
#include <wiringPi.h>
#include <softPwm.h>

#define LED_PIN 4

const int Led[15]={17,18,27,22,23,24,25,6,12,13,16,19,20,26,21};

void LedControl(int light) {
    softPwmWrite(LED_PIN, light);
}

int main(void) {
    if(wiringPiSetupGpio()==-1) return 1;

    pinMode(LED_PIN, OUTPUT);
    softPwmCreate(LED_PIN, 0, 500);

    int i;
    for(i=0; i<15; i++) {
        digitalWrite(Led[i], LOW);
    }

    while(1) {

        int ledlight = 0;

        for(i=0; i<500; i++) {
            LedControl(ledlight);
            delay(5);
            ledlight+=1;
        }

        LedControl(ledlight); // ledlight 100%
        delay(1000);

        for(i=0; i<500; i++) {
            LedControl(ledlight);
            delay(5);
            ledlight-=1;
        }

        LedControl(ledlight);
        delay(1000);
    }

    return 0;
}
```

<ass2-4_2.c 의 main문>

각 예제 및 과제의 코드와 실행결과 영상은 폴더에 첨부하였습니다.