

# ARM매물형처리기에 의한 걸음전동기조종에 대한 연구

최영훈

ARM11계열의 매물형처리기 S3C6410의 경우에 GPIO의 기능에 대해서는 소개되어있지만 리용방법에 대해서는 구체적으로 소개된것이 없다.[1-3]

본문에서는 ARM11계열의 매물형처리기 S3C6410에 기초한 매물형체계 FriendlyARM을 리용하여 걸음전동기를 조종하기 위한 GPIO기능의 응용에 대하여 서술하였다.

## 1. 걸음전동기조종회로의 구성

걸음전동기조종요소인 UCN4202를 리용한 걸음전동기의 조종회로는 그림 1과 같다.

걸음전동기의 회전방향조종신호와 걸음입력신호는 각각 처리기의 GPIO E포구인 GPE의 GPE4와 GPE5에 의해서 조종된다.

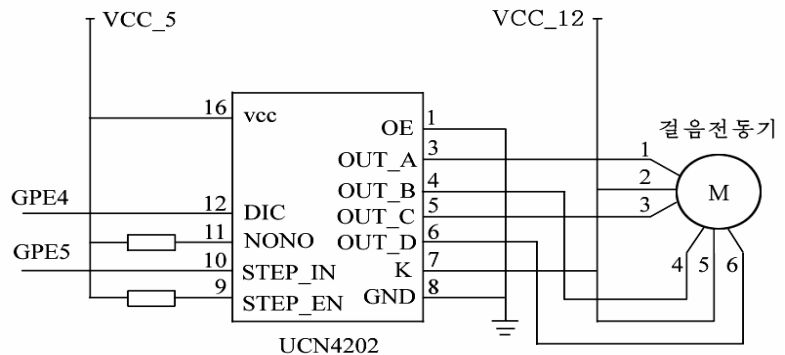


그림 1. 걸음전동기조종회로

## 2. 걸음전동기조종알고리즘

S3C6410은 187개의 다기능입출력포구단자를 가지고있는데 이것들은 17개의 포구그룹을 이루고있다.

GPIO는 2개의 부분 즉 능동부분과 차단부분으로 구성되어있다.

S3C6410의 GPIO기능에 기초한 걸음전동기조종알고리즘은 그림 2와 같다.

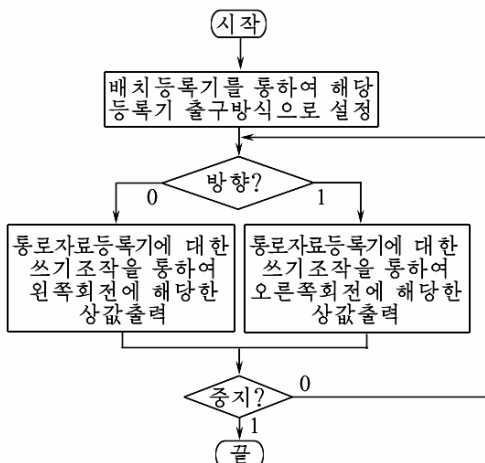


그림 2. 걸음전동기조종알고리즘

## 3. 걸음전동기조종프로그램

여기에서는 그림 2의 알고리즘에 기초하여 작성된 장치구동프로그램과 조종프로그램의 주요부분들에 대하여 보기로 한다.

걸음전동기조종을 위한 장치구동프로그램은 본질적으로는 장치구동프로그램의 구조에 맞게

GPIO기능을 리용하는 과정을 반영하고있는데 주요함수들을 보면 다음과 같다.

```
void s3c6410_gpio_cfgpin(unsigned int pin, unsigned int function)
{ /*어떤 단자들에 대하여 배치(입력/출력/기타기능) */
    unsigned long base=s3c6410_GPIO_BASE(pin); /*통로그룹의 기준주소얻기*/
    unsigned long shift=1;
    unsigned long mask=0x03; /*배치등록기의 두 비트를 리용하여 하나의 통로를 표시*/
    unsigned long con;
    unsigned long flags;
    if (pin < s3c6410_GPIO_BANKB)
    {
        shift =0;
        mask = 0x01; /* GPE통로에서 배치등록기의 한비트를 리용하여 하나의 통로를 표시*/
    }
    mask <<= (S3C6410_GPIO_OFFSET(pin) << shift);
    local_irq_save(flags); /*상태보관, 다음번의것이 원자조작임을 담보*/
    con = __raw_readl(base+0x00);
    con &= ~mask;
    con |= function;
    __raw_writel(con,base + 0x00); /*배치 등록기에 새로운 배치자료쓰기*/
    local_irq_restore(flags); /*상태회복*/
}

void s3c6410_gpio_pullup(unsigned int pin, unsigned int to)
{ /* 끌어올림기능배치*/
    unsigned long base = S3C6410_GPIO_BASE(pin); /*통로그룹의 기준주소 얻기*/
    unsigned long offs = S3C6410_GPIO_OFFSET9pin);/*통로그룹안의 편이주소 얻기*/
    unsigned long flags;
    unsigned long up;
    if (pin < S3C6410_GPIO_BANKB)
    {
        return;
    }
    local_irq_save(flags);
    up = __raw_readl(base + 0x08);
    up &= ~(1 << offs);
    up |= to << offs;
    __raw_writel(up, base+ 0x08); /*끌어올림기능등록기를 향하여 새 배치자료쓰기*/
    local_irq_restore(flags);
}

void s3c6410_gpio_setpin(unsigned int pin, unsigned int to)
{ /*어떤 단자로 출력 */
    unsigned long base = S3C6410_GPIO_BASE(pin);
    unsigned long offs = S3C6410_GPIO_OFFSET(pin);
    unsigned long flags;
    unsigned long dat;
    local_irq_save(flags);
    dat = __raw_readl(base + 0x04);
```

```

    dat  &= ~(1 << offs);
    dat |= to  << offs);
    __raw_writel(dat, base + 0x04); /*자료등록기에 새 자료쓰기*/
    local_irq_restor(flags);
}

```

우에서 s3c6410\_gpio\_cfgpin( ) 함수와 s3c6410\_gpio\_pullup( ) 함수는 매 통로에 대한 배치를 진행하며 s3c6410\_gpio\_setpin( ) 함수는 자료등록기의 어떤 통로로의 출력을 실현한다.

한편 걸음전동기조종프로그램은 GPIO의 걸음전동기조종시간선도에 따르는 GPE그룹에 대한 쓰기조작이므로 작성된 장치구동프로그램을 리용하는 경우 다음과 같이 간단히 조종프로그램을 작성할수 있다.

```

/* gpio_motor.c*/
...
while(1)
{
    i = (i+1) % 4;
    motor_timer(dev_fd, i, 2);/* 2박자 출력, 2박자 휴식*/
}

```

우에서 본 걸음전동기조종알고리즘과 장치구동프로그램에 의한 조종방법은 매물형 Linux체계환경에서 동작시킬수 있으므로 선행한 방법[3]에서 제기한 직접구동에 의한 조종방법보다 조종정확도가 1.2배 개선되고 유연성이 높다는것이 실험적으로 입증되었다.

## 맺 는 말

ARM의 GPIO기능을 리용하여 장치들을 조종하기 위한 한가지 방법으로서 걸음전동기를 조종하기 위한 회로구성과 장치구동프로그램, 장치조종프로그램을 설계하고 그 동작의 유연성과 정확도가 개선되었음을 실험적으로 입증하였다.

## 참 고 문 헌

- [1] 최영훈; 매물형체계와 응용, 김일성종합대학출판사, 112~125, 주체101(2012).
- [2] Arvind Krishnaswamy; LCTES'02-SCOPES'02, 6, 19, 2012.
- [3] 王飞 等; 电子技术应用, 120, 27, 2009.

주체106(2017)년 8월 5일 원고접수

## Research on the Stepping Motor Control by ARM Embedded Processor

*Choe Yong Hun*

In this paper we described on the application of GPIO function for stepping motor control using embedded system FriendlyARM by ARM embedded processor S3C6410. We confirmed action accuracy of stepping motor control circuit and program running by experiment.

Key words: embedded processor, GPIO, stepping motor