

박기홍_07_26_과제보고서

MCU 2일차

작성자 : 박기홍.

업데이트 : 2024.08.01.

과제 1

데이터 시트 ADC 파트 번역하여 공부하고 공부한 내용 보고서로 제출

ADC란 무엇인가?

아날로그 디지털 변환을 수행하는 기계 장치를 아날로그 디지털 변환기입니다.

Analog/Digital Convertor

ADC Clock – clk_{ADC}

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

Note

"ATmega128 Datasheet p.36"

ADC에는 전용 클럭 도메인이 제공됩니다. 이를 통해 CPU 및 I/O 클럭을 중지하여 디지털 회로에서 발생하는 노이즈를 줄일 수 있습니다. 이를 통해 보다 정확한 ADC 변환 결과를 제공합니다.

Question

ADC가 아날로그 신호를 디지털 신호로 변환해 준다고 했는데, 그럼 표본화, 양자화, 부호화를 거쳐서 진행 되는가? 예전에 표본화, 양자화, 부호화 관련 개념을 배운 적이 있는데, 다시 리마인드 시킬 필요성이 있음.

ADC 레지스터

ATmega128를 원활하고 효율적으로 사용하기 위해서는 각 기능별 레지스터를 파악하여 공부할 필요가 있습니다. LED 센서를 다루기 위해 DDRx, PORTx, PINx 레지스터에 대하여 공부하였습니다. 가변저항 센서를 원활하게 다루기 위해 ADC의 세부 레지스터에 대하여 공부한 내용을 정리하였습니다.

ADMUX

ADMUX(ADC Multiplexer Selection Register)는 AD 변환을 위한 기준 전압을 설정하고 입력 채널을 선택하기 위한 레지스터입니다.

ADCSRx

ADCSRA(ADC Control and Status Register A)는 ADC 모듈의 동작을 설정하고, 상태를 표시합니다.

ADCH/L

ADCH/L ((ADC Data Register)는 ADC의 반환 결과를 저장합니다.

[참고 자료]

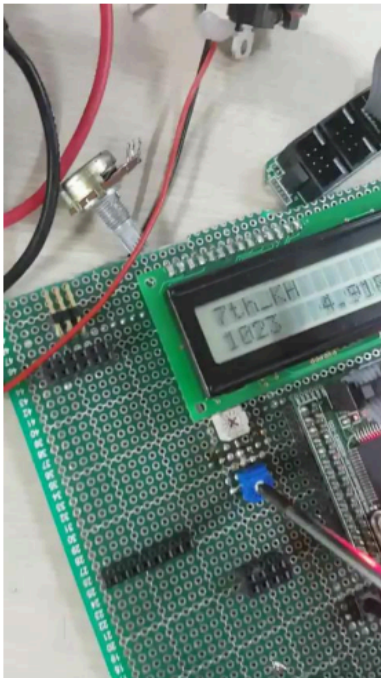
- [ATmega128A Data Sheet - ADMUX](#)
- [ATmega328 : 아날로그-디지털 변환](#)
- [atmega128_A/D 변화 레지스터](#)

과제 2

HW_002

 과제 설명

| 과제 2



1. 가변 저항 값에 따라 LED를 이동시킨다.
 2. ADC값을 LCD에 띄우기.
 3. ADC값을 계산해서 현재 가변저항의 전압 값을 LCD에 표시.
- 이름 이니셜 표기할 것

[과제 풀이]

Github repo : [바로가기](#)

과제 시연 영상 : MCU-2일차-과제2-시연영상.mp4

기능 모음

<1번 기능>

가변 저항 값에 따라 LED를 이동시킨다.

<2번 기능>

ADC값을 LCD에 띄우기.

<3번 기능>

ADC값을 계산해서 현재 가변저항의 전압 값을 LCD에 표시.

<4번 기능>

이름 이니셜 표기할 것.

전체 소스코드

```
/*
 * HW_002.c
 *
 * Created: 2024-08-01 오후 2:50:31
 * Author : lordk
 * SRS:
 * 모든 SRS(명세서)는 과제보고서와 작업일지(개인)에 기록함.
 * 1. 가변 저항 값에 따라 LED를 이동시킨다.
 * 2. ADC값을 LCD에 띄우기.
 * 3. ADC값을 계산해서 현재 가변저항의 전압 값을 LCD에 표시.
 * 4. 이름 이니셜 표기할 것.
 */

#define F_CPU 16000000
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "LCD_Text.h"

int main(void){
    DDRA = 0xFF;
    DDRF = 0x00;

    PORTA = 0xFF;

    ADMUX = 0x40;
```

```
ADCSRA = 0x87;  
//SREG = 0x80; // -> 에러 남.  
sei(); // SREG 대신 sei() 함수 사용함.
```

```
lcdInit();  
lcdClear();
```

```
int cnt = 0xFF;
```

```
while(1){
```

```
    unsigned int adcValue = 0;  
    unsigned char channel = 0x00;
```

```
    ADMUX = 0x40 | channel;  
    ADCSRA |= 0x40;
```

```
    while((ADCSRA & 0x10) == 0){  
        adcValue = ADC;  
        int adcValueVoltage = (5 / 1024) * adcValue;  
        // <2번 기능>  
        lcdNumber(1, 0, adcValue);  
        // <3번 기능>
```

lcdNumber(1, 6, adcValueVoltage); // adcVaulue는 네 칸을 확보
하므로, 가독성을 위해 한 칸 공백 후 6칸부터 전압을 표시함.

_delay_ms(100); // lcd 초기화를 진행하여 값의 변화에 따라 ldc
에 출력함.

```
    lcdClear();
```

```
    // <1번 기능>
```

```
    if(adcValue < (1023 / 8) * 1){  
        PORTA = 0b11111110;  
    }else if(adcValue < (1023 / 8) * 2){  
        PORTA = 0b11111101;  
    }else if(adcValue < (1023 / 8) * 3){  
        PORTA = 0b11111101;  
    }else if(adcValue < (1023 / 8) * 4){  
        PORTA = 0b11110111;  
    }else if(adcValue < (1023 / 8) * 5){  
        PORTA = 0b11101111;  
    }else if(adcValue < (1023 / 8) * 6){  
        PORTA = 0b11011111;  
    }else if(adcValue < (1023 / 8) * 7){  
        PORTA = 0b10111111;  
    }else if(adcValue < (1023 / 8) * 8){  
        PORTA = 0b01111111;  
    }  
}
```

```

    }

    lcdString(0, 0, "19th_KH");

    _delay_ms(50);
}

}

```

코드 리뷰하기

[1. 전처리문]

```

#define F_CPU 16000000
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "LCD_Text.h"

```

- <util/delay.h> 헤더파일 내부에 있는 delay() 함수를 사용하기 위해서, 클럭을 지정해 주어야 하므로 F_CPU를 사용하고 있는 MCU인 ATmega128의 클럭인 16Mhz에 알맞는 스펙으로 정의하였습니다.
- avr의 input, output이 가능하게 하기 위해 관련 헤더파일인 <avr/io.h>를 선언하였습니다.
- delay() 함수를 사용하기 위해 관련 헤더파일인 <util/delay.h>를 선언하였습니다.
- Interrupt 관련 처리를 하기 위해 <avr/interrupt.h>를 선언하였습니다.
- LCD 출력 관련 처리를 위해 "LCD_Text" 헤더파일을 선언하였습니다.

[2. main() 함수 내부]

```

int main(void){
    DDRA = 0xFF;
    DDRF = 0x00;

    PORTA = 0xFF;

    ADMUX = 0x40;
    ADCSRA = 0x87;
    //SREG = 0x80; // -> 에러 남.
    sei(); // SREG 대신 sei() 함수 사용함.

    lcdInit();
    lcdClear();

    int cnt = 0xFF;
}

```

```

while(1){

    unsigned int adcValue = 0;
    unsigned char channel = 0x00;

    ADMUX = 0x40 | channel;
    ADCSRA |= 0x40;

    while((ADCSRA&0x10) == 0){
        adcValue = ADC;
        int adcValueVoltage = (5 / 1024) * adcValue;
        // <2번 기능>
        lcdNumber(1, 0, adcValue);
        // <3번 기능>
        lcdNumber(1, 6, adcValueVoltage); // adcVaulue는 네 칸을 확보
        하므로, 가독성을 위해 한 칸 공백 후 6칸부터 전압을 표시함.
        _delay_ms(100); // lcd 초기화를 진행하여 값의 변화에 따라 ldc
        에 출력함.

        lcdClear();

        // <1번 기능>
        if(adcValue < (1023 / 8) * 1){
            PORTA = 0b11111110;
        }else if(adcValue < (1023 / 8) * 2){
            PORTA = 0b11111101;
        }else if(adcValue < (1023 / 8) * 3){
            PORTA = 0b11111011;
        }else if(adcValue < (1023 / 8) * 4){
            PORTA = 0b11110111;
        }else if(adcValue < (1023 / 8) * 5){
            PORTA = 0b11101111;
        }else if(adcValue < (1023 / 8) * 6){
            PORTA = 0b11011111;
        }else if(adcValue < (1023 / 8) * 7){
            PORTA = 0b10111111;
        }else if(adcValue < (1023 / 8) * 8){
            PORTA = 0b01111111;
        }

    }

    lcdString(0, 0, "19th_KH");

    _delay_ms(50);
}

```

```
}
```

[2-1. 레지스터 선언하기]

```
DDRA = 0xFF;
DDRF = 0x00;

PORTA = 0xFF;

ADMUX = 0x40;
ADCSRA = 0x87;
//SREG = 0x80; // -> 에러 남.
sei(); // SREG 대신 sei() 함수 사용함.

lcdInit();
lcdClear();
```

- LED PIN의 켜기/끄기를 위해 DDRA로 선언한 후, 출력모드로 초기화 하였습니다.
- 가변저항 센서를 사용하기 위해 DDRF로 선언한 후, 입력모드로 초기화 하였습니다.
- LED PIN이 연결된 전체 PORTA를 꺼짐 상태로 초기화 하였습니다.
- ADMUX를 0x40(0100 000)번째에 있는 입력 채널로 초기화 하였습니다.
- ADCSRA를 0x87(1000 0111)로 초기화 하였습니다.
- 왜 해당 값으로 초기화 하였는지는 더 자세히 알아본 후에 보고서를 업데이트하도록 하겠습니다.
- SREG 에러가 발생하여 두 시간동안 관련 사례를 찾아보다가 sei() 함수를 사용해야 한다는 것을 동기들로부터 전해 들은 후, sei()함수를 사용하여 interrupt 관련 초기화를 진행하였습니다.
- "LDC_Text.h" 내부에 있는 lcdInit() 함수와 lcdClear() 함수를 호출하여 lcd를 초기화 하였습니다.

과제 3

기존 과제를 진행하느라 과제 진행이 늦어졌습니다. 그럼에도 불구하고, 포기하지 않고 과제 2를 수정한 후에 과제 3을 끝까지 진행하여 보고서 작성 후, 재제출 드리겠습니다.

전체 과제 리뷰

회로 작업이 늦게 끝나 과제를 늦게 완수하여 제출드립니다. 과제 2번 같은 경우에는 전압 출력 관련하여 수정한 후에, 보고서 추가 작성과 함께 재제출 드리겠습니다. 과제 3번 같은 경우에도 완수한 후에 제출드리겠습니다.