



ATmega128로 배우는  
마이크로컨트롤러 프로그래밍

ATmega128 Programming

허경용 지음

#### 코드 연습 문제 4-1

```
int with_while(int sum)
{
    int _sum = 0, i = 1;

    while(_sum < sum){
        _sum += i++;
    }

    printf("with   while : %d\r\n", i - 1);
}

int with_do_while(int sum)
{
    int _sum = 0, i = 1;

    do{
        _sum += i++;
    }while(_sum < sum);

    printf("with do-while : %d\r\n", i - 1);
}

int with_for(int sum)
{
    int _sum = 0, i;

    for(i = 1; _sum < sum; i++){
        _sum += i;
    }

    printf("with   for : %d\r\n", i - 1);
}
```

#### 코드 연습 문제 4-2

```
void print_bits(uint8_t no)
{
    for(int i = 7; i >= 0; i--){
        uint8_t bit = (no >> i) & 0x01;

        printf("%d th bit is %d.\r\n", i, bit);
    }
}
```

#### 코드 연습 문제 7-1

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

char circular_shift_right(char pattern, int n)
{
    char new_pattern = (pattern << (8 - n)) | (pattern >> n);
}
```

```

    return new_pattern;
}

char circular_shift_left(char pattern, int n)
{
    char new_pattern = (pattern << n) | (pattern >> (8 - n));

    return new_pattern;
}

int main(void)
{
    char pattern = 0xA0;
    DDRB = 0xFF;

    while (1)
    {
        PORTB = pattern;

        pattern = circular_shift_left(pattern, 1);

        _delay_ms(500);
    }

    return 0;
}

```

## 코드 연습 문제 7-2

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

char circular_shift_right(char pattern, int n)
{
    char new_pattern = (pattern << (8 - n)) | (pattern >> n);

    return new_pattern;
}

char circular_shift_left(char pattern, int n)
{
    char new_pattern = (pattern << n) | (pattern >> (8 - n));

    return new_pattern;
}

int main(void)
{
    char pattern_old, pattern_new;
    char pattern_left = 0x80, pattern_right = 0x01;

    DDRB = 0xFF;

    while (1)
    {
        pattern_new = pattern_left | pattern_right;

```

```

    pattern_old = pattern_new;

    PORTB = pattern_new;

    do{
        pattern_left = circular_shift_right(pattern_left, 1);
        pattern_right = circular_shift_left(pattern_right, 1);

        pattern_new = pattern_left | pattern_right;
    }while(pattern_new == pattern_old);

    _delay_ms(500);
}

return 0;
}

```

### 코드 연습 문제 8-1

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

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

    PORTB = 0x00;

    while(1)
    {
        PORTB = PINF;
    }

    return 0;
}

```

### 코드 연습 문제 9-2

```

#define F_CPU 16000000L

#include <avr/io.h>
#include <util/delay.h>
#include <string.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

int main(void)
{
    char buffer[100] = "";
}

```

```

stdout = &OUTPUT;
stdin = &INPUT;

UART1_init();

while(1)
{
    scanf("%s", buffer);

    for(int i = 0; buffer[i]; i++){
        if(buffer[i] >= 'a' && buffer[i] <= 'z'){
            buffer[i] = buffer[i] + 'A' - 'a';
        }
        else if(buffer[i] >= 'A' && buffer[i] <= 'Z'){
            buffer[i] = buffer[i] + 'a' - 'A';
        }
    }

    printf("\r\n%s\r\n", buffer);
}

return 0;
}

```

### 코드 연습 문제 10-1

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

void ADC_init(unsigned char channel)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADFR);

    ADMUX |= ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));

    return ADC;
}

int main(void)
{

```

```

int read;

stdout = &OUTPUT;
stdin = &INPUT;

UART1_init();
ADC_init(0);

DDRB = 0xFF;

while(1)
{
    read = read_ADC();

    int LED_position = (read >> 7);
    uint8_t pattern = (1 << LED_position);

    PORTB = pattern;

    printf("Read : %d,\t%dth LED is ON!\r\n", read, LED_position);

    _delay_ms(100);
}

return 0;
}

```

## 코드 연습 문제 10-2

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

void ADC_init(void)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
}

void ADC_select_channel(unsigned char channel)
{
    ADMUX = ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));
}

```

```

    return ADC;
}

int main(void)
{
    int read0, read1;

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();
    ADC_init();

    DDRB = 0xFF;

    while(1)
    {
        ADC_select_channel(0);
        read0 = read_ADC();
        ADC_select_channel(0);
        read0 = read_ADC();

        ADC_select_channel(1);
        read1 = read_ADC();
        ADC_select_channel(1);
        read1 = read_ADC();

        printf("(%4d, %4d)\r\n", read0, read1);

        _delay_ms(1000);
    }

    return 0;
}

```

### 코드 연습 문제 11-1

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

void ADC_init(void)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
}

void ADC_select_channel(unsigned char channel)
{

```

```

    ADMUX = ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));

    return ADC;
}

int main(void)
{
    int read0, read1;

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();
    ADC_init();

    while(1)
    {
        ADC_select_channel(0);
        read0 = read_ADC();
        ADC_select_channel(0);
        read0 = read_ADC();

        ADC_select_channel(1);
        read1 = read_ADC();
        ADC_select_channel(1);
        read1 = read_ADC();

        uint8_t mask = (1 << ACO);

        printf("(%4d, %4d) : %s\r\n", read0, read1, ((ACSR & mask) == mask ? "ON" : "OFF"));

        _delay_ms(1000);
    }

    return 0;
}

```

## 코드 연습 문제 11-2

```

#define F_CPU 16000000L
#include <avr/io.h>

int main(void)
{
    DDRB |= 0x01;

    SFIOR |= (1 << ACME);
    ADCSRA &= ~(1 << ADEN);
    ADMUX = (ADMUX & 0xE0) | 0x01;

    while(1)
    {

```



```

uint8_t mask = (1 << ACO);
if((ACSR & mask) == mask){
    PORTB |= 0x01;
}
else{
    PORTB &= ~0x01;
}
}

return 0;
}

```

## 코드 연습 문제 12-1

```

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

volatile int state = 0;

ISR(INT0_vect)
{
    state = 0;
}

ISR(INT1_vect)
{
    state = 1;
}

void INIT_PORT(void)
{
    DDRB = 0xFF;
    PORTB = 0x00;

    DDRD = 0x00;
}

void INIT_INT01(void)
{
    EIMSK |= (1 << INT0) | (1 << INT1);
    EICRA |= (1 << ISC00) | (1 << ISC01) | (1 << ISC10) | (1 << ISC11);
    sei();
}

int main(void)
{
    INIT_PORT();
    INIT_INT01();

    while(1){
        if(state == 1){
            PORTB = 0xFF;
        }
        else{
            PORTB = 0x00;
        }
    }
}

```

```

    }

    return 0;
}

```

## 코드 연습 문제 12-2

```

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

volatile int state = 1;

ISR(INT0_vect)
{
    state *= -1;
}

void INIT_PORT(void)
{
    DDRB = 0xFF;
    PORTB = 0x00;

    DDRD = 0x00;
}

void INIT_INT0(void)
{
    EIMSK |= (1 << INT0);
    EICRA |= (1 << ISC00) | (1 << ISC01);
    sei();
}

int main(void)
{
    INIT_PORT();
    INIT_INT0();

    int index = 0;

    while(1){
        index += state;

        if(index >= 8) index = 0;
        if(index < 0) index = 7;

        uint8_t pattern = (1 << index);

        PORTB = pattern;

        _delay_ms(500);
    }

    return 0;
}

```

### 코드 연습 문제 13-1

```
#include <avr/io.h>
#include <avr/interrupt.h>

volatile int count = 0;
volatile int pattern = 0;

ISR(TIMER0_OVF_vect)
{
    count++;
    if(count == 32){
        count = 0;
        pattern = (pattern + 1) % 8;
    }
}

int main(void)
{
    DDRB = 0xFF;
    PORTB = (1 << pattern);

    TCCR0 |= (1 << CS02) | (1 << CS01) | (1 << CS00);

    TIMSK |= (1 << TOIE0);

    sei();

    while(1){
        PORTB = (1 << pattern);
    }

    return 0;
}
```

### 코드 연습 문제 13-2

```
#include <avr/io.h>
#include <avr/interrupt.h>

#define MILLIS_INCREMENT_PER_OVERFLOW    1
#define MICROS_INCREMENT_PER_OVERFLOW    24

volatile unsigned long timer0_millis = 0;
volatile int timer0_micros = 0;

ISR(TIMER0_OVF_vect)
{
    unsigned long m = timer0_millis;
    int f = timer0_micros;

    m += MILLIS_INCREMENT_PER_OVERFLOW;
    f += MICROS_INCREMENT_PER_OVERFLOW;

    m += (f / 1000);
    f = f % 1000;

    timer0_millis = m;
}
```

```

    timer0_micros = f;
}

unsigned long millis()
{
    unsigned long m;
    uint8_t oldSREG = SREG;

    cli();

    m = timer0_millis;

    SREG = oldSREG;

    return m;
}

void init_timer0()
{
    TCCR0 |= (1 << CS02);
    TIMSK |= (1 << TOIE0);

    sei();
}

int main(void)
{
    uint8_t pattern = 0;

    init_timer0();

    DDRB = 0xFF;
    PORTB = (1 << pattern);

    unsigned long time_previous, time_current;
    time_previous = millis();

    while(1)
    {
        time_current = millis();

        if((time_current - time_previous) > 500){
            time_previous = time_current;

            pattern = (pattern + 1) % 8;
            PORTB = (1 << pattern);
        }
    }

    return 1;
}

```

## 코드 연습 문제 14-1

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

#define TIME_1S      16383
#define TIME_0_1S   1638

int state = 0;

void ADC_init(unsigned char channel)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADFR);

    ADMUX = ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));

    return ADC;
}

ISR(TIMER1_COMPA_vect)
{
    state = !state;
    if(state) PORTB = 0xFF;
    else PORTB = 0x00;

    TCNT1 = 0;
}

int main(void)
{
    int read;

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();

    DDRB = 0xFF;
    PORTB = 0x00;
```

```

ADC_init(0);

OCR1A = TIME_1S;

TCCR1B |= (1 << CS12) | (1 << CS10);

TIMSK |= (1 << OCIE1A);

sei();

while(1)
{
    read = read_ADC();
    int new_compare_match = (int)((TIME_1S - TIME_0_1S) / 1023.0 * read + TIME_0_1S);

    printf("%d : %d\r\n", read, new_compare_match);

    OCR1A = new_compare_match;

    _delay_ms(100);

}

return 0;
}

```

#### 코드 연습 문제 14-2

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>

#define TIME_1S      16383
#define TIME_0_1S   1638

volatile int pattern = 0;

void ADC_init(unsigned char channel)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADFR);

    ADMUX = ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));

    return ADC;
}

```

```

ISR(TIMER1_COMPA_vect)
{
    pattern = (pattern + 1) % 8;

    TCNT1 = 0;
}

int main(void)
{
    int read;

    DDRB = 0xFF;
    PORTB = 0x00;

    ADC_init(0);

    OCR1A = TIME_1S;

    TCCR1B |= (1 << CS12) | (1 << CS10);

    TIMSK |= (1 << OCIE1A);

    sei();

    while(1)
    {
        read = read_ADC();
        int new_compare_match = (int)((TIME_1S - TIME_0_1S) / 1023.0 * read + TIME_0_1S);

        OCR1A = new_compare_match;

        PORTB = (1 << pattern);

        _delay_ms(100);
    }

    return 0;
}

```

## 코드 연습 문제 15-2

```

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

#define MILLIS_INCREMENT_PER_OVERFLOW 1
#define MICROS_INCREMENT_PER_OVERFLOW 24

volatile unsigned long timer0_millis = 0;
volatile int timer0_micros = 0;

ISR(TIMER0_OVF_vect)
{
    unsigned long m = timer0_millis;
    int f = timer0_micros;
}

```

```

    m += MILLIS_INCREMENT_PER_OVERFLOW;
    f += MICROS_INCREMENT_PER_OVERFLOW;

    m += (f / 1000);
    f = f % 1000;

    timer0_millis = m;
    timer0_micros = f;
}

unsigned long millis()
{
    unsigned long m;
    uint8_t oldSREG = SREG;

    cli();

    m = timer0_millis;

    SREG = oldSREG;

    return m;
}

void init_timer0()
{
    TCCR0 |= (1 << CS02);
    TIMSK |= (1 << TOIE0);

    sei();
}

int main(void)
{
    init_timer0();

    DDRB |= (1 << PB5) | (1 << PB6);

    TCCR1A |= (1 << WGM10);

    TCCR1A |= (1 << COM1A1);
    TCCR1A |= (1 << COM1B1) | (1 << COM1B0);

    TCCR1B |= (1 << CS12);

    int dim = 0;
    int direction = 1;

    unsigned long time_previous, time_current;
    time_previous = millis();

    int interval = 10;

    OCR1A = dim;
    OCR1B = dim;

    while(1)
    {

```



```

    time_current = millis();

    if(time_current - time_previous >= interval){
        time_previous = time_current;

        dim += direction;

        OCR1A = dim;
        OCR1B = dim;

        if(dim == 0) direction = 1;
        if(dim == 255) direction = -1;
    }
}

return 0;
}

```

### 코드 연습 문제 15-3

```

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

#define MILLIS_INCREMENT_PER_OVERFLOW 1
#define MICROS_INCREMENT_PER_OVERFLOW 24

volatile unsigned long timer0_millis = 0;
volatile int timer0_micros = 0;

ISR(TIMER0_OVF_vect)
{
    unsigned long m = timer0_millis;
    int f = timer0_micros;

    m += MILLIS_INCREMENT_PER_OVERFLOW;
    f += MICROS_INCREMENT_PER_OVERFLOW;

    m += (f / 1000);
    f = f % 1000;

    timer0_millis = m;
    timer0_micros = f;
}

unsigned long millis()
{
    unsigned long m;
    uint8_t oldSREG = SREG;

    cli();

    m = timer0_millis;

    SREG = oldSREG;

    return m;
}

```

```

}

void init_timer0()
{
    TCCR0 |= (1 << CS02);
    TIMSK |= (1 << TOIE0);

    sei();
}

void ADC_init(unsigned char channel)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADFR);

    ADMUX = ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));

    return ADC;
}

int main(void)
{
    init_timer0();

    DDRB |= (1 << PB5) | (1 << PB6);

    TCCR1A |= (1 << WGM10);

    TCCR1A |= (1 << COM1A1);
    TCCR1A |= (1 << COM1B1) | (1 << COM1B0);

    TCCR1B |= (1 << CS12);

    int dim = 0;
    int direction = 1;

    unsigned long time_previous, time_current;
    time_previous = millis();

    int interval = 10;

    ADC_init(0);

    OCR1A = dim;
    OCR1B = dim;

    while(1)
    {
        time_current = millis();

```

```

int read = read_ADC();
interval = (int)(9 / 1023.0 * read + 1);

if(time_current - time_previous >= interval){
    time_previous = time_current;

    dim += direction;

    OCR1A = dim;
    OCR1B = dim;

    if(dim == 0) direction = 1;
    if(dim == 255) direction = -1;
}
}

return 0;
}

```

### 코드 연습 문제 16-1

```

void EEPROM_write_int(int address, int value)
{
    uint8_t *p = (uint8_t*)&value;

    EEPROM_writeByte(address, *p);
    EEPROM_writeByte(address + 1, *(p + 1));
}

int EEPROM_read_int(int address)
{
    int result = EEPROM_readByte(address) + (EEPROM_readByte(address + 1) << 8);

    return result;
}

```

### 코드 연습 문제 16-2

```

void EEPROM_write_int(int address, int value)
{
    EEPROMint temp;
    temp.value = value;

    EEPROM_writeByte(address, temp.bytes[0]);
    EEPROM_writeByte(address + 1, temp.bytes[1]);
}

int EEPROM_read_int(int address)
{
    EEPROMint temp;
    temp.bytes[0] = EEPROM_readByte(address);
    temp.bytes[1] = EEPROM_readByte(address + 1);

    return temp.value;
}

```

## 코드 연습 문제 17-1

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include <avr/interrupt.h>

#include "I2C_RTC.h"

int state = 0;

uint8_t bcd_to_decimal(uint8_t bcd)
{
    return (bcd >> 4) * 10 + (bcd & 0x0F);
}

uint8_t decimal_to_bcd(uint8_t decimal)
{
    return ( ((decimal / 10) << 4) | (decimal % 10) );
}

void INIT_INT2(void)
{
    EIMSK |= (1 << INT2);
    EICRA |= (1 << ISC21);
    sei();
}

ISR(INT2_vect)
{
    state = (state + 1) % 2;

    PORTB = 0xFF * state;
}

int main(void)
{
    I2C_init();

    INIT_INT2();

    DDRB = 0xFF;
    PORTB = 0x00;

    while(1){}
    return 0;
}
```

## 코드 연습 문제 17-2

```
#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>

#include "I2C_RTC.h"
```

```

uint8_t address = 0x68;

uint8_t bcd_to_decimal(uint8_t bcd)
{
    return (bcd >> 4) * 10 + (bcd & 0x0F);
}

uint8_t decimal_to_bcd(uint8_t decimal)
{
    return ( ((decimal / 10) << 4) | (decimal % 10) );
}

int get_seconds_from_RTC(void)
{
    int seconds;

    I2C_start();
    I2C_transmit(address << 1);

    I2C_transmit(0);
    I2C_stop();

    I2C_start();
    I2C_transmit( (address << 1) + 1 );

    seconds = bcd_to_decimal(I2C_receive_NACK());

    I2C_stop();

    return seconds;
}

int main(void)
{
    I2C_init();

    int second_previous, second_current, state = 0;

    DDRB = 0xFF;
    PORTB = 0x00;

    second_previous = get_seconds_from_RTC();

    while(1){
        second_current = get_seconds_from_RTC();

        if(second_current != second_previous){
            second_previous = second_current;

            state = (state + 1) % 2;
            PORTB = 0xFF * state;
        }
    }

    return 0;
}

```

## 코드 연습 문제 18-1

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

#define SEGMENT_DELAY          5
#define MILLIS_PER_OVERFLOW    16
#define MICROS_PER_OVERFLOW    384

uint8_t numbers[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x27, 0x7F, 0x67};

volatile uint32_t millis0 = 0;
volatile uint16_t micros = 0;

ISR(TIMER0_OVF_vect)
{
    micros += MICROS_PER_OVERFLOW;
    millis0 += MILLIS_PER_OVERFLOW;

    millis0 += (micros / 1000);
    micros %= 1000;
}

void display_digit(int position, int number)
{
    PORTC |= 0x0F;
    PORTC &= ~(1 << (position - 1));

    PORTD = numbers[number];
}

unsigned long millis()
{
    unsigned long m;
    uint8_t oldSREG = SREG;

    cli();

    m = millis0;

    SREG = oldSREG;

    return m;
}

int main(void)
{
    DDRD = 0xFF;
    DDRC = 0x0F;

    DDRB = 0x00;

    TCCR0 |= (1 << CS02) | (1 << CS01) | (1 << CS00);
    TIMSK |= (1 << TOIE0);
    sei();

    int minutes = 0, seconds = 0;
```

```

unsigned long time_previous, time_current;
uint8_t start = 0;

time_previous = millis();

while(1)
{
    uint8_t button_state = PINB & 0x01;

    if(button_state){
        start = (start + 1) % 2;
    }

    time_current = millis();

    if(start == 0){
        time_previous = time_current;
    }

    if(time_current - time_previous >= 1000){
        time_previous = time_current;
        seconds++;
        if(seconds >= 60){
            minutes = (minutes + 1) % 60;
            seconds %= 60;
        }
    }

    display_digit(1, minutes / 10);
    _delay_ms(SEGMENT_DELAY);
    display_digit(2, minutes % 10);
    _delay_ms(SEGMENT_DELAY);

    display_digit(3, seconds / 10);
    _delay_ms(SEGMENT_DELAY);
    display_digit(4, seconds % 10);
    _delay_ms(SEGMENT_DELAY);
}

return 0;
}

```

## 코드 연습 문제 18-2

```

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

#define SEGMENT_DELAY      5
#define MILLIS_PER_OVERFLOW 16
#define MICROS_PER_OVERFLOW 384

uint8_t patterns[] = {0x3E, 0x3D, 0x3B, 0x37, 0x2F, 0x1F};

volatile uint32_t millis0 = 0;
volatile uint16_t micros = 0;

```

```

ISR(TIMERO_OVF_vect)
{
    micros += MICROS_PER_OVERFLOW;
    millis0 += MILLIS_PER_OVERFLOW;

    millis0 += (micros / 1000);
    micros %= 1000;
}

void display_pattern(int position, int number)
{
    PORTC |= 0x0F;
    PORTC &= ~(1 << (position - 1));

    PORTD = patterns[number];
}

unsigned long millis()
{
    unsigned long m;
    uint8_t oldSREG = SREG;

    cli();

    m = millis0;

    SREG = oldSREG;

    return m;
}

int main(void)
{
    DDRD = 0xFF;
    DDRC = 0x0F;

    DDRB = 0x00;

    TCCR0 |= (1 << CS02) | (1 << CS01) | (1 << CS00);
    TIMSK |= (1 << TOIE0);
    sei();

    unsigned long time_previous, time_current;

    time_previous = millis();

    int index = 0;

    while(1)
    {
        time_current = millis();

        if(time_current - time_previous >= 1000){
            time_previous = time_current;

            index = (index + 1) % 6;
        }
    }
}

```



```

    display_pattern(1, index);
    _delay_ms(SEGMENT_DELAY);
    display_pattern(2, (index + 1) % 6);
    _delay_ms(SEGMENT_DELAY);
    display_pattern(3, (index + 2) % 6);
    _delay_ms(SEGMENT_DELAY);
    display_pattern(4, (index + 3) % 6);
    _delay_ms(SEGMENT_DELAY);
}

return 0;
}

```

### 코드 연습 문제 19-1

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

#define set_bit(bit)      ( PORTB |= _BV(bit) )
#define clear_bit(bit)   ( PORTB &= ~_BV(bit) )

#define SHIFT_CLOCK      0
#define LATCH_CLOCK      1
#define DATA             2

uint8_t numbers[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x27, 0x7F, 0x67};

void ShiftClock(void)
{
    set_bit(SHIFT_CLOCK);
    clear_bit(SHIFT_CLOCK);
}

void LatchClock(void)
{
    set_bit(LATCH_CLOCK);
    clear_bit(LATCH_CLOCK);
}

void ByteDataWrite(uint8_t data)
{
    for(uint8_t i = 0; i < 8; i++){
        if(data & 0b10000000)
            set_bit(DATA);
        else
            clear_bit(DATA);

        ShiftClock();
        data = data << 1;
    }

    LatchClock();
}

int main(void)
{
    DDRB |= _BV(SHIFT_CLOCK) | _BV(LATCH_CLOCK) | _BV(DATA);
}

```

```

uint8_t index = 0;
while (1)
{
    uint8_t pattern = numbers[index];
    index = (index + 1) % 10;

    ByteDataWrite(pattern);

    _delay_ms(500);
}

return 0;
}

```

## 코드 연습 문제 19-2

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "I2C_RTC.h"
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

#define MCP23017_ADDRESS      0x20
#define MCP23017_IODIRA      0x00
#define MCP23017_IODIRB      0x01
#define MCP23017_GPPUA       0x0C
#define MCP23017_GPPUB       0x0D
#define MCP23017_GPIOA        0x12
#define MCP23017_GPIOB        0x13

int main(void)
{
    I2C_init();
    UART1_init();

    stdout = &OUTPUT;
    stdin = &INPUT;

    I2C_start();
    I2C_transmit(MCP23017_ADDRESS << 1);
    I2C_transmit(MCP23017_IODIRA);
    I2C_transmit(0x00);
    I2C_transmit(0xFF);
    I2C_stop();

    I2C_start();
    I2C_transmit(MCP23017_ADDRESS << 1);
    I2C_transmit(MCP23017_GPPUB);
    I2C_transmit(0);
    I2C_stop();
}

```

```

while(1)
{
    I2C_start();
    I2C_transmit(MCP23017_ADDRESS << 1);
    I2C_transmit(MCP23017_GPIOB);
    I2C_stop();

    I2C_start();
    I2C_transmit((MCP23017_ADDRESS << 1) + 1);
    uint8_t button_state = I2C_receive_NACK();
    I2C_stop();

    if(button_state != 0){
        I2C_start();
        I2C_transmit(MCP23017_ADDRESS << 1);
        I2C_transmit(MCP23017_GPIOA);
        I2C_transmit(button_state);
        I2C_stop();

        printf("%d\r\n", button_state);
    }

    _delay_ms(500);
}

return 0;
}

```

### 코드 연습 문제 19-3

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "I2C_RTC.h"
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

#define MCP23017_ADDRESS      0x20
#define MCP23017_IODIRA      0x00
#define MCP23017_IODIRB      0x01
#define MCP23017_GPPUA       0x0C
#define MCP23017_GPPUB       0x0D
#define MCP23017_GPIOA       0x12
#define MCP23017_GPIOB       0x13

int main(void)
{
    I2C_init();
    UART1_init();

    stdout = &OUTPUT;
    stdin = &INPUT;
}

```

```

I2C_start();
I2C_transmit(MCP23017_ADDRESS << 1);
I2C_transmit(MCP23017_IODIRA);
I2C_transmit(0x00);
I2C_transmit(0xFF);
I2C_stop();

I2C_start();
I2C_transmit(MCP23017_ADDRESS << 1);
I2C_transmit(MCP23017_GPPUB);
I2C_transmit(0);
I2C_stop();

while(1)
{
    I2C_start();
    I2C_transmit(MCP23017_ADDRESS << 1);
    I2C_transmit(MCP23017_GPIOB);
    I2C_stop();

    I2C_start();
    I2C_transmit((MCP23017_ADDRESS << 1) + 1);
    uint8_t button_state = I2C_receive_NACK();
    I2C_stop();

    if(button_state != 0){
        uint8_t pos, state = 0;

        for(int i = 7; i >= 0; i--){
            if(button_state & (1 << i)){
                pos = i;
                break;
            }
        }

        for(int i = 0; i <= pos; i++){
            state |= (1 << i);
        }

        I2C_start();
        I2C_transmit(MCP23017_ADDRESS << 1);
        I2C_transmit(MCP23017_GPIOA);
        I2C_transmit(state);
        I2C_stop();

        printf("%d\r\n", button_state);
    }

    _delay_ms(500);
}

return 0;
}

```

## 코드 연습 문제 20-1

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

#define REGISTER_DDR    DDRD
#define REGISTER_PORT   PORTD

#define set_bit(bit)    ( REGISTER_PORT |= _BV(bit) )
#define clear_bit(bit) ( REGISTER_PORT &= ~_BV(bit) )

#define SHIFT_CLOCK     2
#define LATCH_CLOCK     1
#define DATA           0

#define COL_ON          0
#define COL_OFF         1
#define ROW_ON          1
#define ROW_OFF         0

uint8_t numbers[][8] = {
    { 0b00000000, 0b01111110, 0b10000001, 0b10000001, 0b10000001, 0b10000001, 0b01111110, 0b00000000 },
    { 0b00000000, 0b00000000, 0b00000000, 0b00000010, 0b11111111, 0b00000000, 0b00000000, 0b00000000 },
    { 0b00000000, 0b10000010, 0b11000001, 0b10100001, 0b10010001, 0b10001001, 0b10000110, 0b00000000 },
    { 0b00000000, 0b01000010, 0b10000001, 0b10001001, 0b10001001, 0b10001001, 0b01110110, 0b00000000 },
    { 0b00000000, 0b00011000, 0b00010100, 0b00010010, 0b00010001, 0b11111111, 0b00010000, 0b00000000 },
    { 0b00000000, 0b10001111, 0b10001001, 0b10001001, 0b10001001, 0b10001001, 0b01110001, 0b00000000 },
    { 0b00000000, 0b01111110, 0b10001001, 0b10001001, 0b10001001, 0b10001001, 0b01110010, 0b00000000 },
    { 0b00000000, 0b10000001, 0b01000001, 0b00100001, 0b00010001, 0b00001001, 0b00000111, 0b00000000 },
    { 0b00000000, 0b01110110, 0b10001001, 0b10001001, 0b10001001, 0b10001001, 0b01110110, 0b00000000 },
    { 0b00000000, 0b00000110, 0b10001001, 0b01001001, 0b00101001, 0b00011001, 0b00000110, 0b00000000 }
};

volatile int pattern_index = 0;
volatile int count = 0;

ISR(TIMER0_OVF_vect)
{
    count++;
    if(count == 64){
        count = 0;
        pattern_index = (pattern_index + 1) % 10;
    }
}

void init_interrupt(void)
{
    TCCR0 |= (1 << CS02) | (1 << CS01) | (1 << CS00);

    TIMSK |= (1 << TOIE0);

    sei();
}

void ShiftClock(void)
{
    set_bit(SHIFT_CLOCK);
}
```

```

    clear_bit(SHIFT_CLOCK);
}

void LatchClock(void)
{
    set_bit(LATCH_CLOCK);
    clear_bit(LATCH_CLOCK);
}

void ByteDataWrite(uint8_t data)
{
    for(uint8_t i = 0; i < 8; i++){
        if(data & 0b10000000)
            set_bit(DATA);
        else
            clear_bit(DATA);

        ShiftClock();
        data = data << 1;
    }

    LatchClock();
}

void init_port()
{
    REGISTER_DDR |= _BV(SHIFT_CLOCK) | _BV(LATCH_CLOCK) | _BV(DATA);
}

int main(void)
{
    init_interrupt();
    init_port();

    while(1)
    {
        for(int i = 0; i < 8; i++){
            uint8_t col_data = ~(1 << i);

            ByteDataWrite(col_data);
            ByteDataWrite(numbers[pattern_index][i]);

            _delay_ms(2);
        }
    }
}

```

## 코드 연습 문제 20-2

```

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

#define REGISTER_DDR    DDRD
#define REGISTER_PORT   PORTD

#define set_bit(bit)    ( REGISTER_PORT |= _BV(bit) )

```

```

#define clear_bit(bit)      ( REGISTER_PORT &= ~_BV(bit) )

#define SHIFT_CLOCK        2
#define LATCH_CLOCK        1
#define DATA                0

#define COL_ON              0
#define COL_OFF             1
#define ROW_ON              1
#define ROW_OFF             0

volatile int pattern_index = 0;
volatile int count = 0;

ISR(TIMER0_OVF_vect)
{
    count++;
    if(count == 64){
        count = 0;
        pattern_index = (pattern_index + 1) % 16;
    }
}

void init_interrupt(void)
{
    TCCR0 |= (1 << CS02) | (1 << CS01) | (1 << CS00);

    TIMSK |= (1 << TOIE0);

    sei();
}

void ShiftClock(void)
{
    set_bit(SHIFT_CLOCK);
    clear_bit(SHIFT_CLOCK);
}

void LatchClock(void)
{
    set_bit(LATCH_CLOCK);
    clear_bit(LATCH_CLOCK);
}

void ByteDataWrite(uint8_t data)
{
    for(uint8_t i = 0; i < 8; i++){
        if(data & 0b10000000)
            set_bit(DATA);
        else
            clear_bit(DATA);

        ShiftClock();
        data = data << 1;
    }

    LatchClock();
}

```

```

void init_port()
{
    REGISTER_DDR |= _BV(SHIFT_CLOCK) | _BV(LATCH_CLOCK) | _BV(DATA);
}

int main(void)
{
    init_interrupt();
    init_port();

    uint8_t smile[] = {
        0b00111100, 0b01000010, 0b10010101, 0b10100001, 0b10100001, 0b10010101, 0b01000010, 0b00111100,
        0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000 };

    while(1)
    {
        for(int i = 0; i < 8; i++){
            uint8_t col_data = ~(1 << i);

            ByteDataWrite(col_data);
            ByteDataWrite(smile[(pattern_index + i) % 16]);

            _delay_ms(2);
        }
    }
}

```

## 코드 연습 문제 22-1

```

#define F_CPU 16000000
#include "Text_LCD.h"
#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>

uint8_t MODE = 4;

int main(void)
{
    LCD_init();

    srand(0);

    while(1)
    {
        int x = random() % 16;
        int y = random() % 2;
        char ch = random() % 30 + 'A';
        if(ch > 'Z') ch = ' ';

        LCD_goto_XY(y, x);
        LCD_write_data(ch);

        _delay_ms(200);
    };

    return 0;
}

```



## 코드 연습 문제 23-1

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

#define ROTATION_DELAY    1000
#define PULSE_MIN        1300
#define PULSE_MID        3000
#define PULSE_MAX        4700

void ADC_init(unsigned char channel)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADFR);

    ADMUX = ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));

    return ADC;
}

void InitializeTimer1(void)
{
    TCCR1A |= (1 << WGM11);
    TCCR1B |= (1 << WGM12) | (1 << WGM13);

    TCCR1A |= (1 << COM1A1);

    TCCR1B |= (1 << CS11);

    ICR1 = 39999;
}

int main(void)
{
    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();
    ADC_init(0);

    DDRB |= (1 << PB5);
```

```

InitializeTimer1();

while(1)
{
    int read = read_ADC();
    int position = (int)((PULSE_MAX - PULSE_MIN) / 1023.0 * read) + PULSE_MIN;

    printf("%d\t\t: %d\r\n", read, position);

    OCR1A = position;
    _delay_ms(100);
}
}

```

### 코드 연습 문제 23-2

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

uint8_t step_data[] = {0x01, 0x03, 0x02, 0x06, 0x04, 0x0C, 0x08, 0x09};
int step_index = -1;

uint8_t stepForward(void)
{
    step_index++;
    if(step_index >= 8) step_index = 0;

    return step_data[step_index];
}

uint8_t stepBackward(void)
{
    step_index--;
    if(step_index < 0) step_index = 7;

    return step_data[step_index];
}

int main(void)
{
    DDRC = 0x0F;

    while(1)
    {
        for(int i = 0; i < 400; i++){
            PORTC = stepForward();

            _delay_ms(5);
        }
        _delay_ms(1000);

        for(int i = 0; i < 400; i++){
            PORTC = stepBackward();

            _delay_ms(5);
        }
        _delay_ms(1000);
    }
}

```

```

    }

    return 0;
}

```

### 코드 연습 문제 24-1

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdio.h>
#include <util/delay.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

#define MILLIS_INCREMENT_PER_OVERFLOW    1
#define MICROS_INCREMENT_PER_OVERFLOW    24

volatile unsigned long timer0_millis = 0;
volatile int timer0_micros = 0;

unsigned long T = 5000;

ISR(TIMEROV_vect)
{
    unsigned long m = timer0_millis;
    int f = timer0_micros;

    m += MILLIS_INCREMENT_PER_OVERFLOW;
    f += MICROS_INCREMENT_PER_OVERFLOW;

    m += (f / 1000);
    f = f % 1000;

    timer0_millis = m;
    timer0_micros = f;
}

unsigned long millis()
{
    unsigned long m;
    uint8_t oldSREG = SREG;

    cli();

    m = timer0_millis;

    SREG = oldSREG;

    return m;
}

void init_timer0()
{

```

```

TCCR0 |= (1 << CS02);
TIMSK |= (1 << TOIE0);

sei();
}

int main(void)
{
    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();
    init_timer0();

    DDRB |= 0x01;
    DDRC &= 0xFC;

    unsigned long time_start, time_current;
    uint8_t state = 0;

    while (1)
    {
        uint8_t buttons = PINC;

        if(buttons & 0x01){
            time_start = millis();
            state = 1;
            printf("** Switched ON !!\r\n");
            PORTB |= 0x01;
            _delay_ms(200);
        }
        else if(buttons & 0x02){
            state = 0;
            printf("** Switched OFF !!\r\n");
            PORTB &= 0xFE;
            _delay_ms(200);
        }

        if(state == 1){
            time_current = millis();
            if(time_current - time_start >= T){
                state = 0;
                printf("** Automatically switched OFF after %ld ms !!\r\n",
                                                                    time_current - time_start);
                PORTB &= 0xFE;
            }
        }
    }

    return 0;
}

```

## 코드 연습 문제 25-1

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

#define ADC_MIN 100
#define ADC_MAX 700

void ADC_init(unsigned char channel)
{
    ADMUX |= (1 << REFS0);

    ADCSRA |= 0x07;
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADFR);

    ADMUX |= ((ADMUX & 0xE0) | channel);
    ADCSRA |= (1 << ADSC);
}

int read_ADC(void)
{
    while(!(ADCSRA & (1 << ADIF)));

    return ADC;
}

int main(void)
{
    int read;

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();
    ADC_init(5);

    DDRB = 0xFF;
    PORTB = 0x00;

    while(1)
    {
        read = read_ADC();

        printf("%d\r\n", read);

        int LED_count = 8 - (read - ADC_MIN) * 8 / (ADC_MAX - ADC_MIN);
        uint8_t LED_state = 0;
        for(int i = 0; i < LED_count; i++){
            LED_state = (LED_state << 1) + 1;
        }
    }
}
```

```

        PORTB = LED_state;

        _delay_ms(1000);
    }

    return 0;
}

```

## 코드 연습 문제 25-2

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

#define PRESCALER 1024

#define MILLIS_INCREMENT_PER_OVERFLOW    1
#define MICROS_INCREMENT_PER_OVERFLOW    24

volatile unsigned long timer0_millis = 0;
volatile int timer0_micros = 0;

int THRESHOLD = 10;

ISR(TIMER0_OVF_vect)
{
    unsigned long m = timer0_millis;
    int f = timer0_micros;

    m += MILLIS_INCREMENT_PER_OVERFLOW;
    f += MICROS_INCREMENT_PER_OVERFLOW;

    m += (f / 1000);
    f = f % 1000;

    timer0_millis = m;
    timer0_micros = f;
}

unsigned long millis()
{
    unsigned long m;
    uint8_t oldSREG = SREG;

    cli();

    m = timer0_millis;

    SREG = oldSREG;
}

```

```

    return m;
}

void init_timer0()
{
    TCCR0 |= (1 << CS02);
    TIMSK |= (1 << TOIE0);

    sei();
}

void Timer_init(void)
{
    TCCR1B |= (1 << CS12) | (1 << CS10);
}

uint8_t measure_distance(void)
{
    PORTB &= ~(1 << PB1);
    _delay_us(1);
    PORTB |= (1 << PB1);
    _delay_us(10);
    PORTB &= ~(1 << PB1);

    TCNT1 = 0;
    while(!(PINB & 0x01))
        if(TCNT1 > 65000) return 0;

    TCNT1 = 0;
    while(PINB & 0x01){
        if (TCNT1 > 650000){
            TCNT1 = 0;
            break;
        }
    }

    double pulse_width = 1000000.0 * TCNT1 * PRESCALER / F_CPU;

    return pulse_width / 58;
}

int main(void)
{
    uint8_t distance;

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();

    DDRB |= 0x02;
    DDRB &= 0xFE;

    uint8_t LED_state = 0;
    DDRC = 0xFF;
    PORTC = 0xFF * LED_state;

    Timer_init();
}

```

```

init_timer0();

unsigned long time1 = millis(), time2;
uint8_t state = 0;

while(1)
{
    distance = measure_distance();

    if(distance < THRESHOLD){
        if(state == 0){
            time1 = millis();
            state = 1;
        }
    }
    else{
        state = 0;
    }

    if(state == 1){
        time2 = millis();
        if(time2 - time1 > 500){
            time1 = time2;
            LED_state = (LED_state + 1) % 2;
            PORTC = 0xFF * LED_state;
        }
    }
    else{
        PORTC = 0x00;
    }
}

return 0;
}

```

### 코드 연습 문제 26-1

```

#include <avr/io.h>
#include <stdio.h>
#include <stdlib.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

int main(void)
{
    char buffer[4] = "";
    int index = 0;

    DDRB = 0xFF;
    PORTB = 0x00;

    uint8_t data;

    stdout = &OUTPUT;

```



```

stdin = &INPUT;

UART1_init();

while (1)
{
    scanf("%c", &data);

    buffer[index++] = data;

    if(index == 3){
        index = 0;
        buffer[3] = 0;
        int num = atoi(buffer);

        PORTB = num;
    }
}

return 0;
}

```

## 코드 연습 문제 26-2

```

#include <avr/io.h>
#include <stdio.h>
#include <stdlib.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

int main(void)
{
    char buffer[10] = "";
    int index = 0;

    DDRB = 0xFF;
    PORTB = 0x00;

    uint8_t data;

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();

    while (1)
    {
        scanf("%c", &data);

        if(data != '\r'){
            if(data == '\n'){
                buffer[index++] = 0;
                int num = atoi(buffer);
            }
        }
    }
}

```

```

        PORTB = num;

        index = 0;
    }
    else{
        buffer[index++] = data;
    }

}

}

return 0;
}

```

### 코드 연습 문제 27-1

```

#include <avr/io.h>
#include <stdio.h>
#include "UART1.h"
#include <string.h>

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

int main(void)
{
    char strings[5][5] = {"abcd", "abce", "bcde", "aa", "1234"};
    char buffer[5];

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();

    for(int i = 0; i < 4; i++){
        for(int j = i + 1; j < 5; j++){
            if(strcmp(strings[i], strings[j]) > 0){
                strcpy(buffer, strings[i]);
                strcpy(strings[i], strings[j]);
                strcpy(strings[j], buffer);
            }
        }
    }

    for(int i = 0; i < 5; i++){
        printf("%s\r\n", strings[i]);
    }

    while (1){}
    return 0;
}

```

## 코드 연습 문제 27-2

```
#include <avr/io.h>
#include <stdio.h>
#include "UART1.h"
#include <string.h>

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

int main(void)
{
    char *address = "http://www.atmel.com/products/microcontrollers/avr/default.aspx";
    char *delimiter = "/.";

    stdout = &OUTPUT;
    stdin = &INPUT;

    UART1_init();

    address = strtok(address, delimiter);
    while(address != NULL){
        printf("%s\r\n", address);
        address = strtok(NULL, delimiter);
    }

    while (1){}
    return 0;
}
```

## 코드 연습 문제 29-1

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdio.h>
#include "UART1.h"

FILE OUTPUT \
= FDEV_SETUP_STREAM(UART1_transmit, NULL, _FDEV_SETUP_WRITE);
FILE INPUT \
= FDEV_SETUP_STREAM(NULL, UART1_receive, _FDEV_SETUP_READ);

volatile int bitCount;
volatile unsigned long receivedData;
int LED_count = 4;

uint8_t get_LED_pattern(int count)
{
    uint8_t value = 0;

    for(int i = 0; i < count; i++){
        value = (value << 1) + 1;
    }

    return value;
}
```

```

void print_received_data(int repeat)
{
    if(repeat == 1)    return;

    if(receivedData == 0xFF02FD){
        LED_count++;
        if(LED_count > 8) LED_count = 8;
    }
    else if(receivedData == 0xFF9867){
        LED_count--;
        if(LED_count < 0) LED_count = 0;
    }

    PORTB = get_LED_pattern(LED_count);;
}

ISR(INT0_vect)
{
    int time = TCNT0;
    int overflow = TIFR & (1 << TOV0);

    if(bitCount == 32){
        if ((time > 201) && (time < 221) && (overflow == 0)) {
            receivedData = 0;
            bitCount = 0;
        }
        else if((time > 166) && (time < 186) && (overflow == 0)){
            print_received_data(1);
        }
        else{
            bitCount = 32;
        }
    }
    else{
        if((time > 40) || (overflow != 0))
            bitCount = 32;
        else{
            if(time > 26)
                receivedData = (receivedData << 1) + 1;
            else
                receivedData = (receivedData << 1);

            if(bitCount == 31)
                print_received_data(0);

            bitCount++;
        }
    }

    TCNT0 = 0;
    TIFR = TIFR | (1 << TOV0);
}

int main(void)
{
    stdout = &OUTPUT;
    stdin = &INPUT;
}

```

```

UART1_init();

TCCR0 |= (1 << CS00) | (1 << CS01) | (1 << CS02);

EIMSK |= (1 << INT0);
EICRA |= (1 << ISC01);
sei();

bitCount = 32;

DDRB = 0xFF;
PORTB = get_LED_pattern(LED_count);

printf("** Initialization Completed... \r\n");

while(1);
return 0;
}

```

### 코드 연습 문제 30-1

```

#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

#define C4      262 // 261.63Hz
#define D4      294 // 293.66Hz
#define E4      330 // 329.63Hz
#define F4      349 // 349.23Hz
#define G4      392 // 392.00Hz
#define A4      440 // 440.00Hz
#define B4      494 // 493.88Hz
#define C5      523 // 523.25Hz
#define PAUSE  0

#define PRESCALER 8

const int notes[] = {C4, D4, E4, F4, G4, A4, B4, C5};

void init_music()
{
    DDRB |= (1 << PB6);

    TCCR1A |= _BV(COM1B0);

    TCCR1B |= _BV(WGM12) | _BV(CS11);
}

void play_note(int freq)
{
    if(freq == PAUSE){
        OCR1A = 0;
    }
    else{
        int ocr_value = F_CPU / 2 / PRESCALER / freq;

        OCR1A = ocr_value;
    }
}

```

```

    }

    _delay_ms(100);

    OCR1A = 0;
}

int main()
{
    init_music();

    DDRC = 0x00;

    while(1)
    {
        uint8_t buttons = PINC;

        if(buttons == 0){
            play_note(0);
        }
        else{
            for(int i = 0; i < 8; i++){
                if(buttons & (1 << i)){
                    play_note(tones[i]);
                    break;
                }
            }
        }
    }

    return 0;
}

```

### 코드 연습 문제 34-1

```

int pins[] = {16, 17, 18, 19, 20, 21, 22, 23};

void setup() {
    for(int i = 0; i < 8; i++){
        pinMode(pins[i], OUTPUT);
    }
}

void loop() {
    for(int i = 0; i < 8; i++){
        for(int j = 0; j < 8; j++){
            if(i == j) digitalWrite(pins[j], HIGH);
            else digitalWrite(pins[j], LOW);
        }
        delay(500);
    }
}

```