14주차

2015. 12. 2.

후반부 강의 계획

9	마이크로프로세서 개요1 C언어	C언어 강의	보드 납땜 (컨트롤러부, LED)
10	마이크로프로세서 개요2 GPIO 1	마이크로프로세서란? Atmega128의 구조와 기능 Firmware ? GPIO ? LED 제어	C언어 Quiz
11	Interrupt GPIO 2	Interrupt 이해 및 실습	C언어 Quiz
12	타이머/카운터	타이머/카운터 이해 및 실습 + LED	C언어 Quiz
13	주변장치 제어1	인터럽트와 4x4 keypad 제어	4x4 Keypad
14	주변장치 제어2	인터럽트와 타이머/카운터를 이용한 7-Segment 제어	7-Segment
15	주변장치 제어3	디지털 시계 / 스톱워치	
16	기말고사	기말고사 (이론 + 실기)	

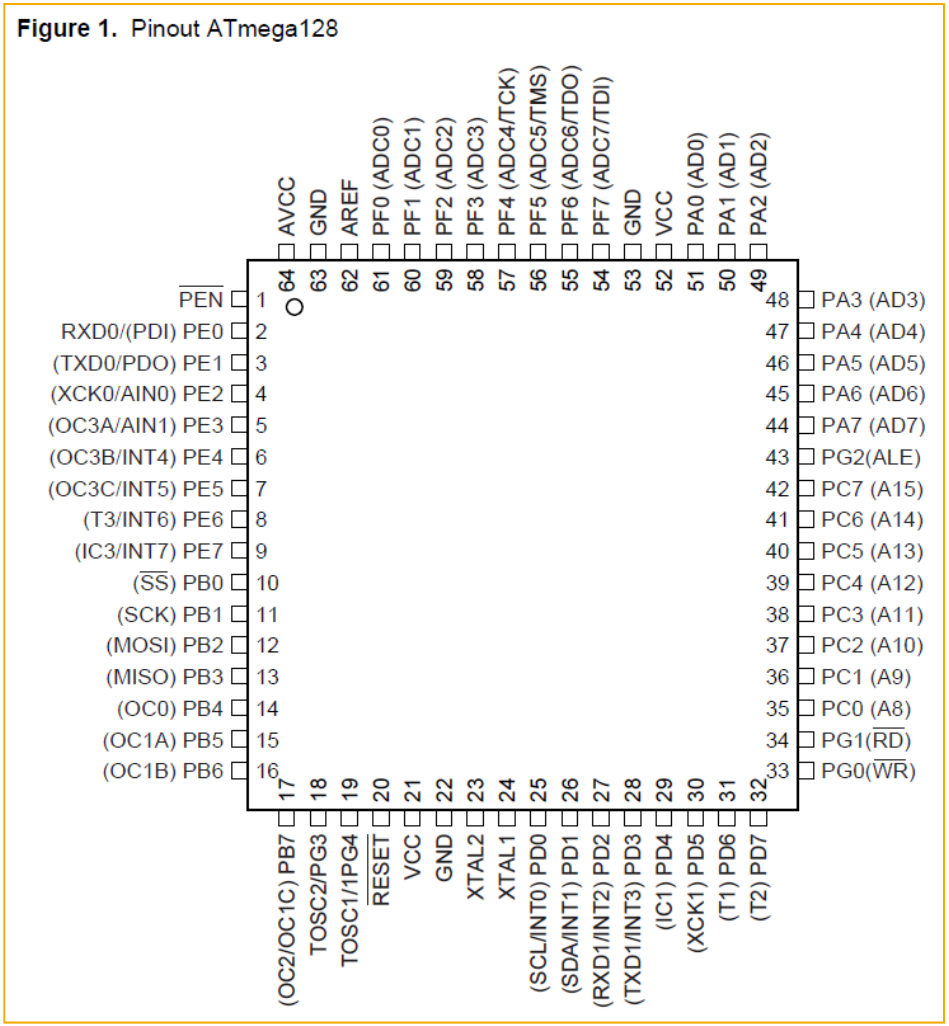
ATMEGA128의 주요 특징

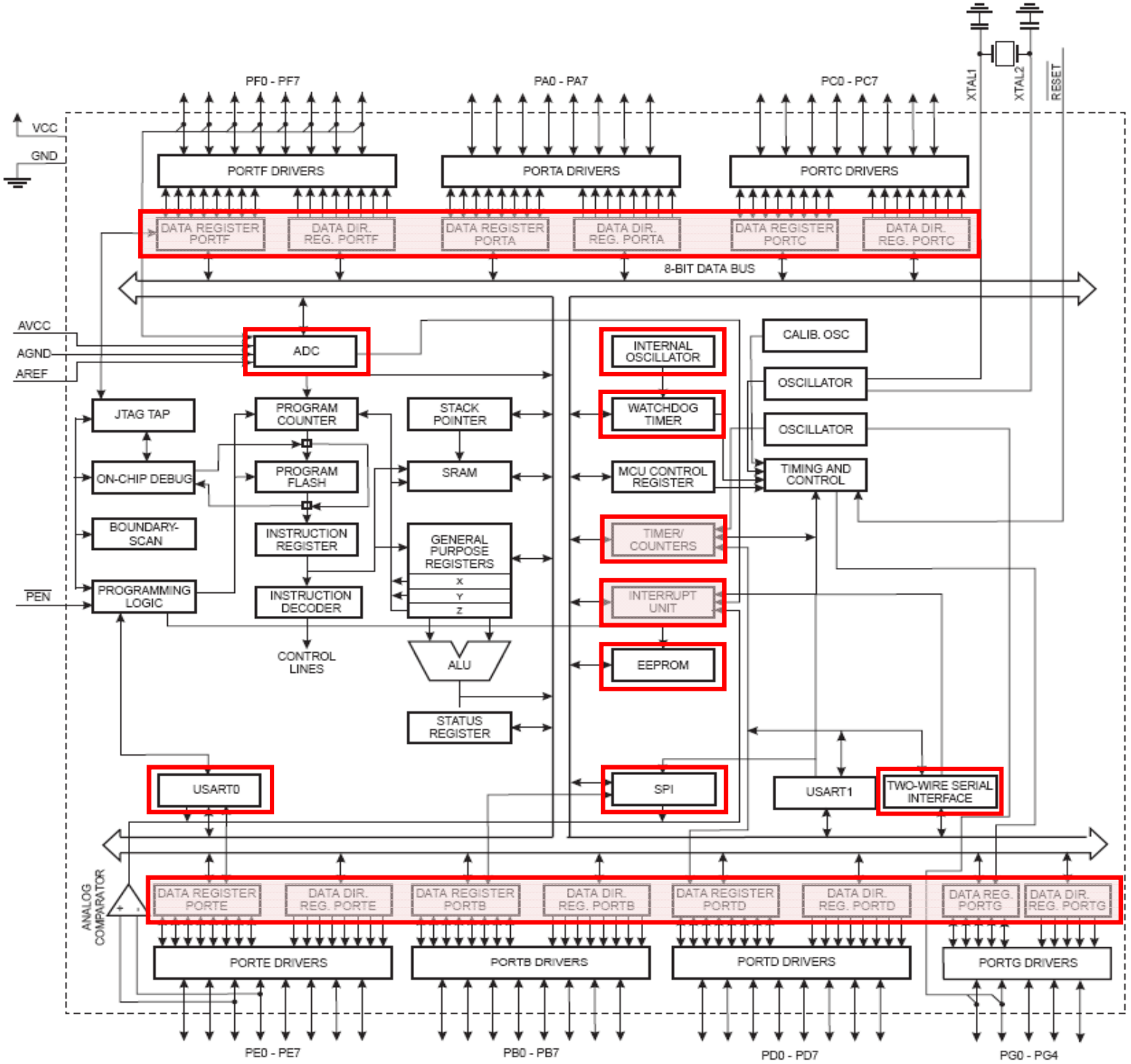
- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 128K Bytes of In-System Reprogrammable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 4K Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
  - Output Compare Modulator
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels
    2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with On-chip Oscillator
  - On-chip Analog Comparator

- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP and 64-pad MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega128L
  - 4.5 - 5.5V for ATmega128
- Speed Grades
  - 0 - 8 MHz for ATmega128L
  - 0 - 16 MHz for ATmega128

교재 15-20 페이지 참고

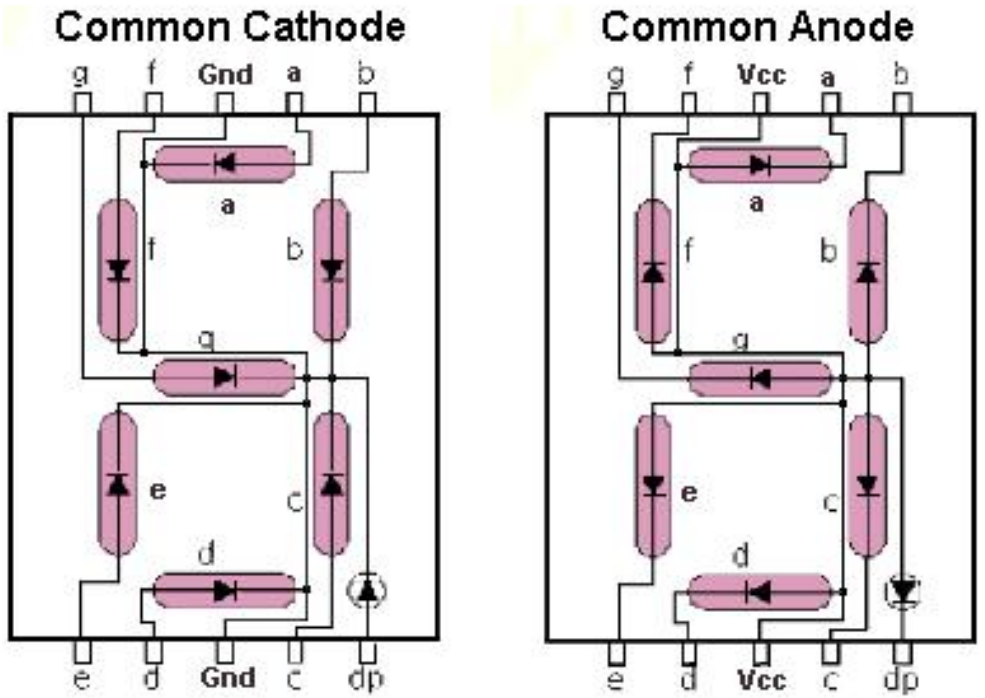
Figure 1. Pinout ATmega128

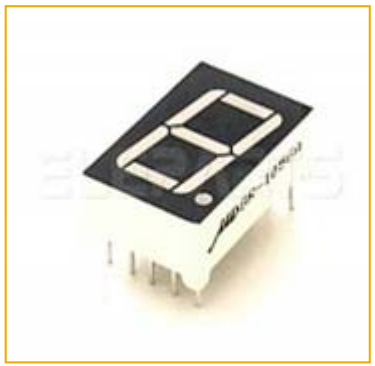



PF0 - PF7
PA0 - PA7
PC0 - PC7
XTAL1
XTAL2
RESET
VCC
GND
PORTF DRIVERS
PORTA DRIVERS
PORTC DRIVERS
DATA REGISTER PORTF
DATA DIR. REG. PORTF
DATA REGISTER PORTA
DATA DIR. REG. PORTA
DATA REGISTER PORTC
DATA DIR. REG. PORTC
8-BIT DATA BUS
AVCC
AGND
AREF
ADC
INTERNAL OSCILLATOR
CALIB. OSC
OSCILLATOR
OSCILLATOR
JTAG TAP
PROGRAM COUNTER
STACK POINTER
WATCHDOG TIMER
ON-CHIP DEBUG
PROGRAM FLASH
SRAM
MCU CONTROL REGISTER
TIMING AND CONTROL
BOUNDARY-SCAN
INSTRUCTION REGISTER
GENERAL PURPOSE REGISTERS
X
Y
Z
TIMER/ COUNTERS
PEN
PROGRAMMING LOGIC
INSTRUCTION DECODER
INTERRUPT UNIT
CONTROL LINES
ALU
EEPROM
STATUS REGISTER
USART0
SPI
USART1
TWO-WIRE SERIAL INTERFACE
ANALOG COMPARATOR
DATA REGISTER PORTE
DATA DIR. REG. PORTE
DATA REGISTER PORTB
DATA DIR. REG. PORTB
DATA REGISTER PORTD
DATA DIR. REG. PORTD
DATA REG. PORTG
DATA DIR. REG. PORTG
PORTE DRIVERS
PORTB DRIVERS
PORTD DRIVERS
PORTG DRIVERS
PE0 - PE7
PB0 - PB7
PD0 - PD7
PG0 - PG4

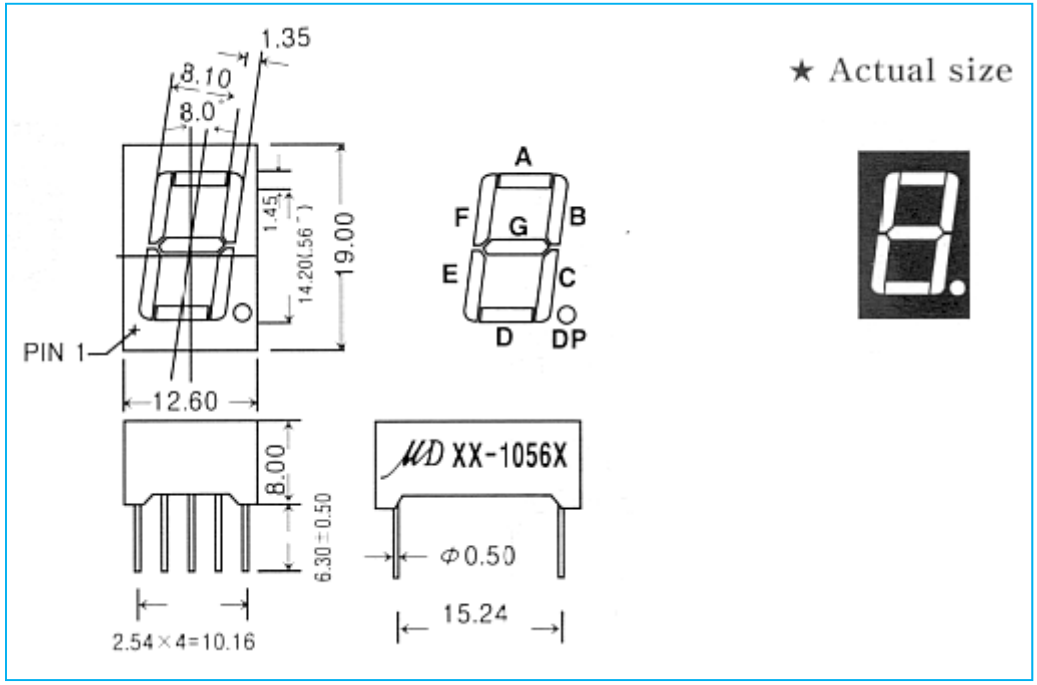
7-Segment

- FND(Flexible Numeric Display)
  = 7-segment

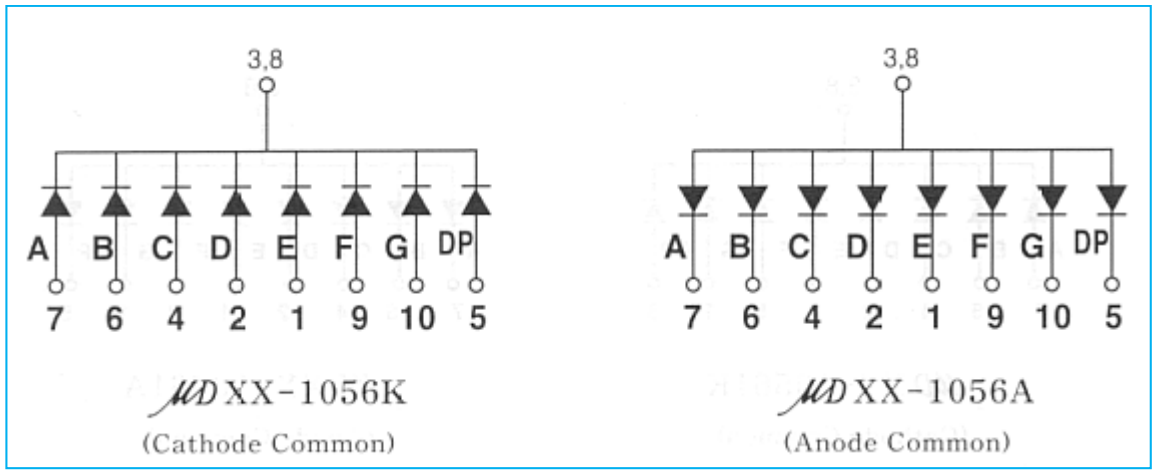




외형


★ Actual size
1.35
8.10
8.0
1.45
14.20(1.56 *)
19.00
PIN 1
12.60
8.00
6.30±0.50
2.54×4=10.16
A
F
G
B
E
C
D
DP
XX-1056X
Φ0.50
15.24

외형


3,8
A B C D E F G DP
7 6 4 2 1 9 10 5
XX-1056K
(Cathode Common)
3,8
A B C D E F G DP
7 6 4 2 1 9 10 5
XX-1056A
(Anode Common)

내부 구조

타이머 이용

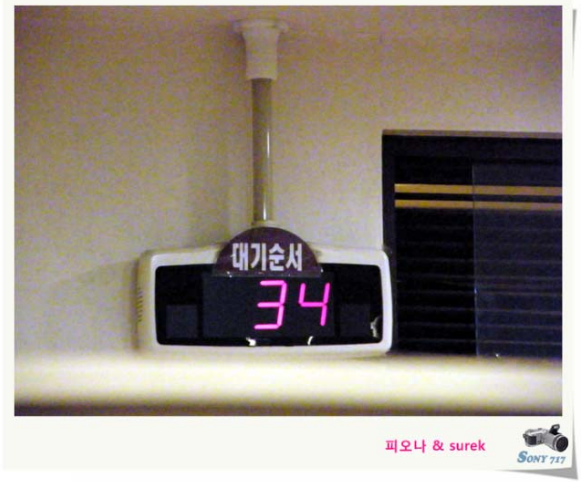
ATMEGA128

카운터 이용

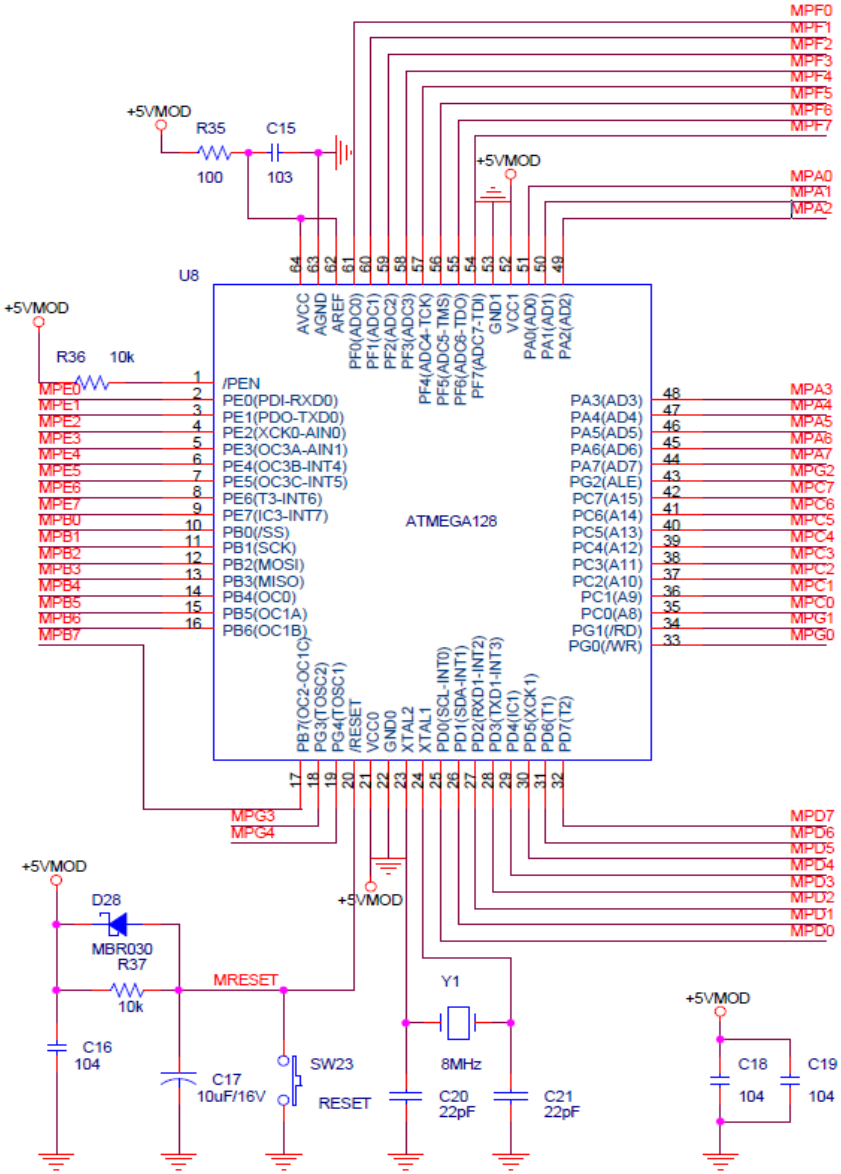
Function Generator

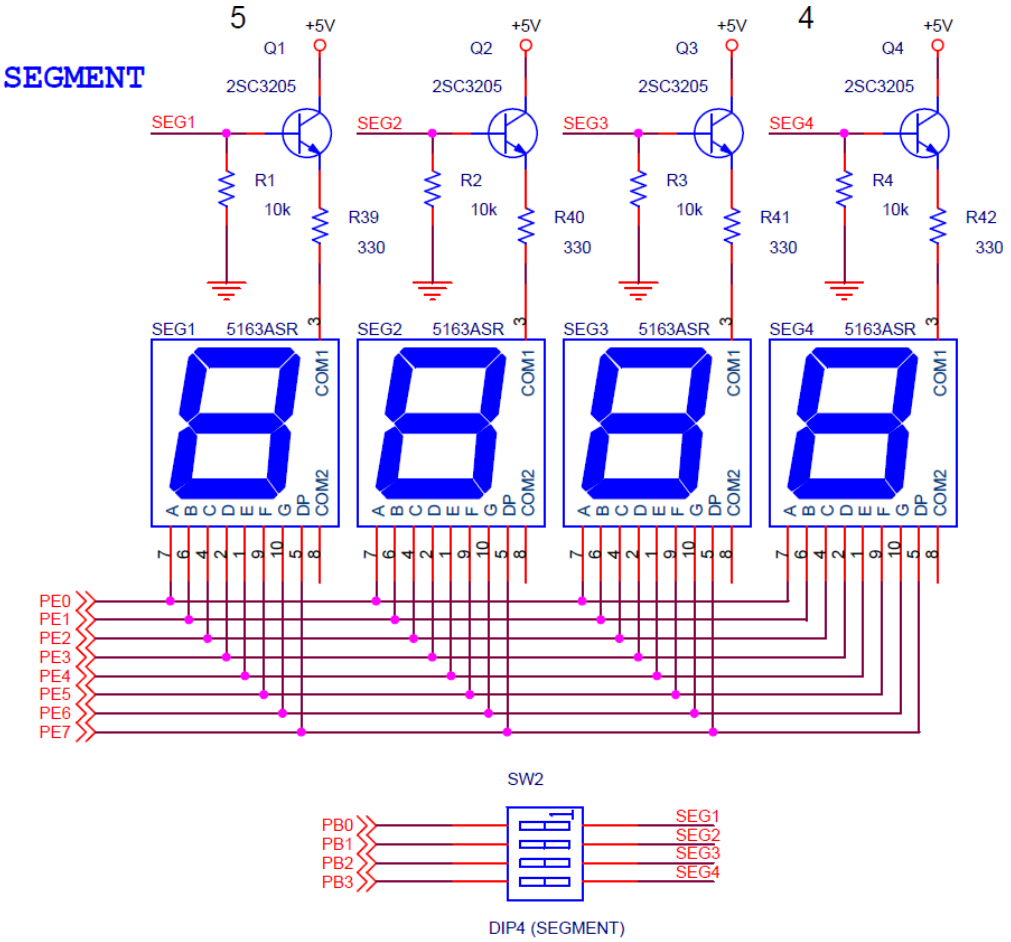
TTL

ATMEGA8



7-Segment 인터페이스





7-Segment - 예제 1

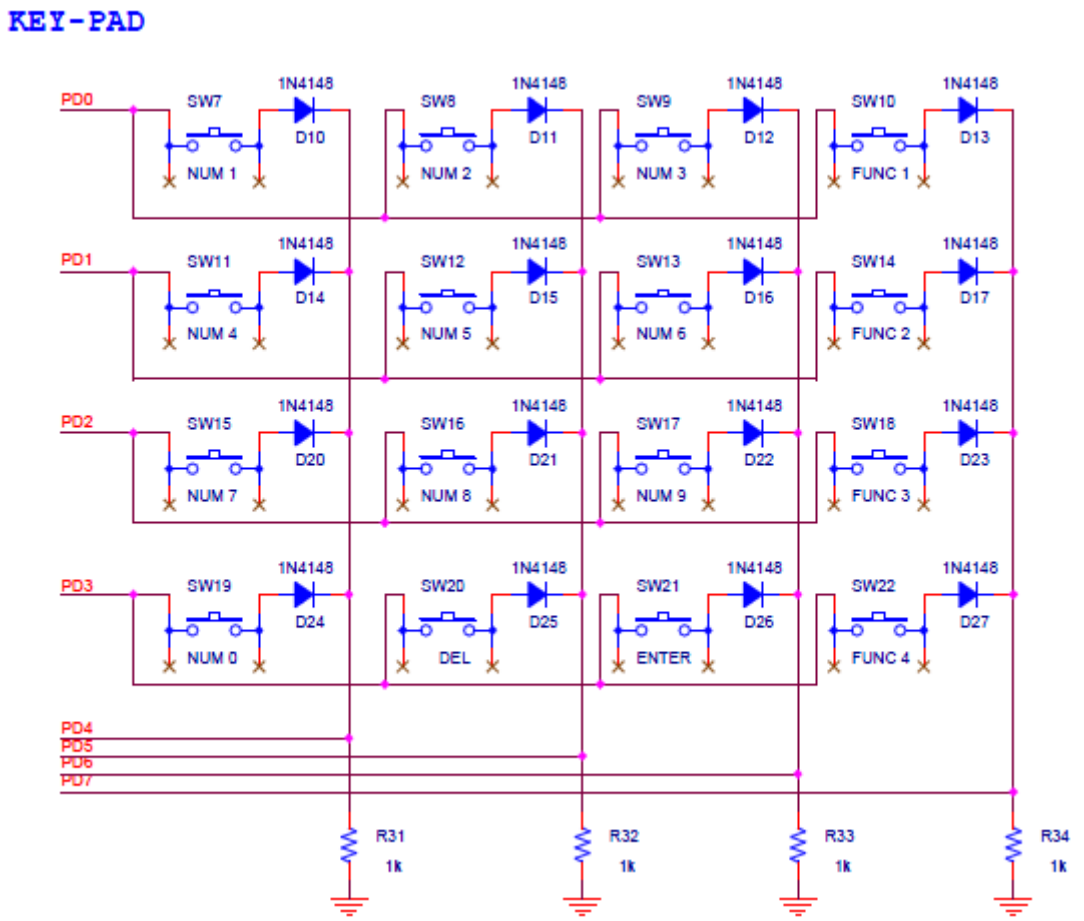
```
#include <iom128.h>
#include <intrinsics.h>

unsigned char segment_decoder[] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xd8, 0x80, 0x98};
unsigned char num_count = 0;

void main(void)
{

  DDRB=0x0F;PORTB=0x0F;  // Segment Control
  DDRE=0xFF;PORTE=0x00;  // Segment LED

  while(1)
  {
   PORTE = segment_decoder[num_count];

   __delay_cycles(800000);

   num_count++;
   if(num_count>9)
    num_count = 0;

  }
}
```

7-Segment - 예제 2

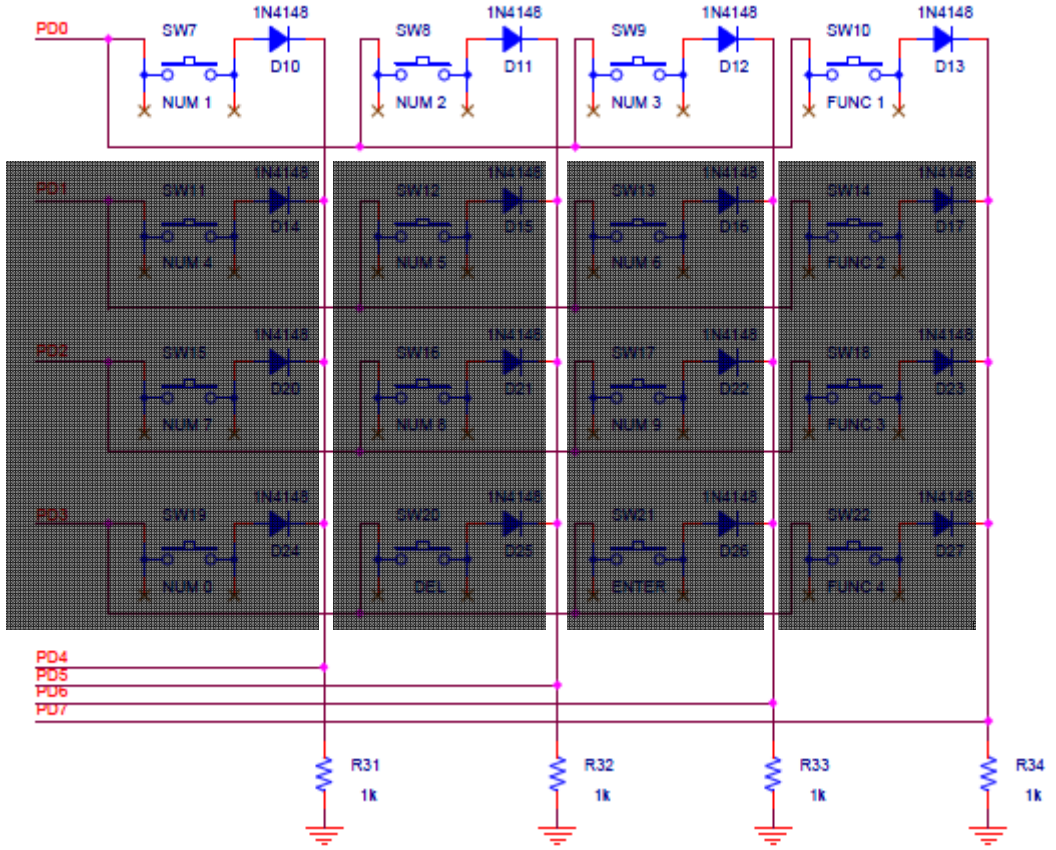
```
#include <iom128.h>
#include <intrinsics.h>

unsigned char segment_decoder[] = {0xc0, 0xf9,
0xa4, 0xb0, 0x99, 0x92, 0x82, 0xd8, 0x80, 0x98};
unsigned char num_count = 0, num_count_1 = 0,
num_count_2 = 0;
unsigned char t_sharing = 0;


#pragma vector=TIMER0_OVF_vect
__interrupt void TIMER0_OVF_interrupt(void)
{
  if(t_sharing > 6)
      t_sharing = 0;  //  1us x 255 x 5 = 1.275ms
  else
  t_sharing++;
}

#pragma vector=TIMER1_OVF_vect
__interrupt void TIMER1_OVF_interrupt(void)
{
  num_count++;
  if(num_count > 99)
    num_count = 0;
}
```

```
void main(void)
{
   DDRB=0x0F;PORTB=0x0F;
   DDRE=0xFF;PORTE=0x00;

   TCCR0=0x02;
   TCNT0=0x00;

   TCCR1A=0x03;
   TCCR1B=0x1D;
   TCNT1H=0x00;TCNT1L=0x00;
   OCR1AH=0x03;OCR1AL=0x0C;

   TIMSK=0x05;

   PORTE = segment_decoder[0];

   __enable_interrupt();

while(1)
{
      t_sharing = 0;
      while(t_sharing < 5) {
       PORTB = 0x04;
       num_count_2 = num_count/10;
       PORTE = segment_decoder[num_count_2];
       }

       t_sharing = 0;
       while(t_sharing < 5) {
       PORTB = 0x08;
       num_count_1 = num_count - (num_count_2*10);
       PORTE = segment_decoder[num_count_1];
      }
  }
}
```

GPIO Control – 4x4 key matrix

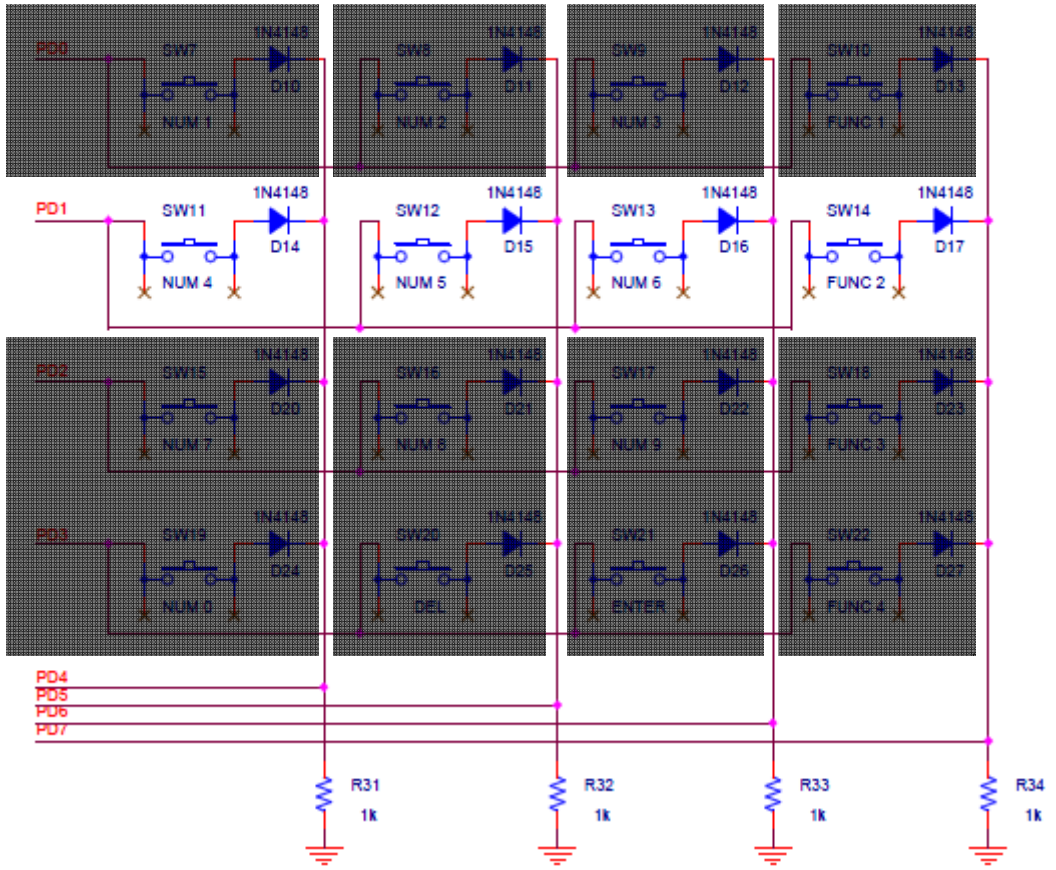


KEY-PAD



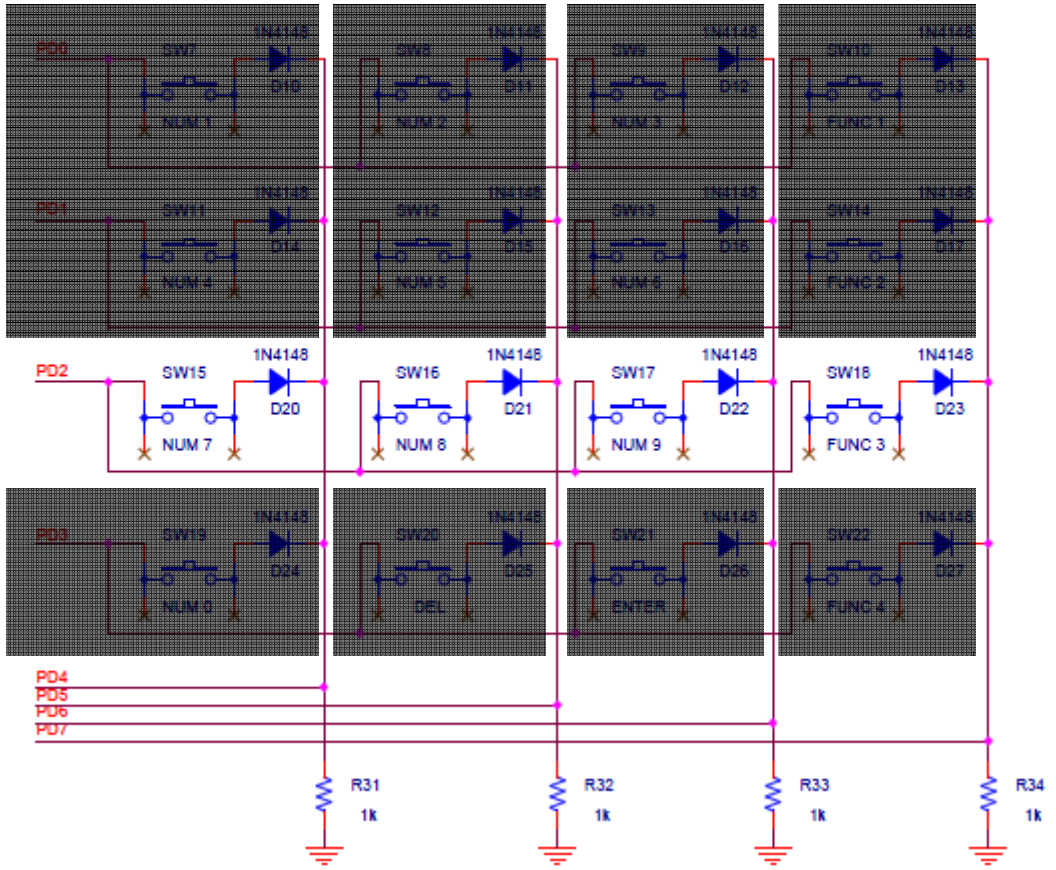

```
PORTD = 0x01;
Key_info = PIND & 0xF0;
```

KEY-PAD



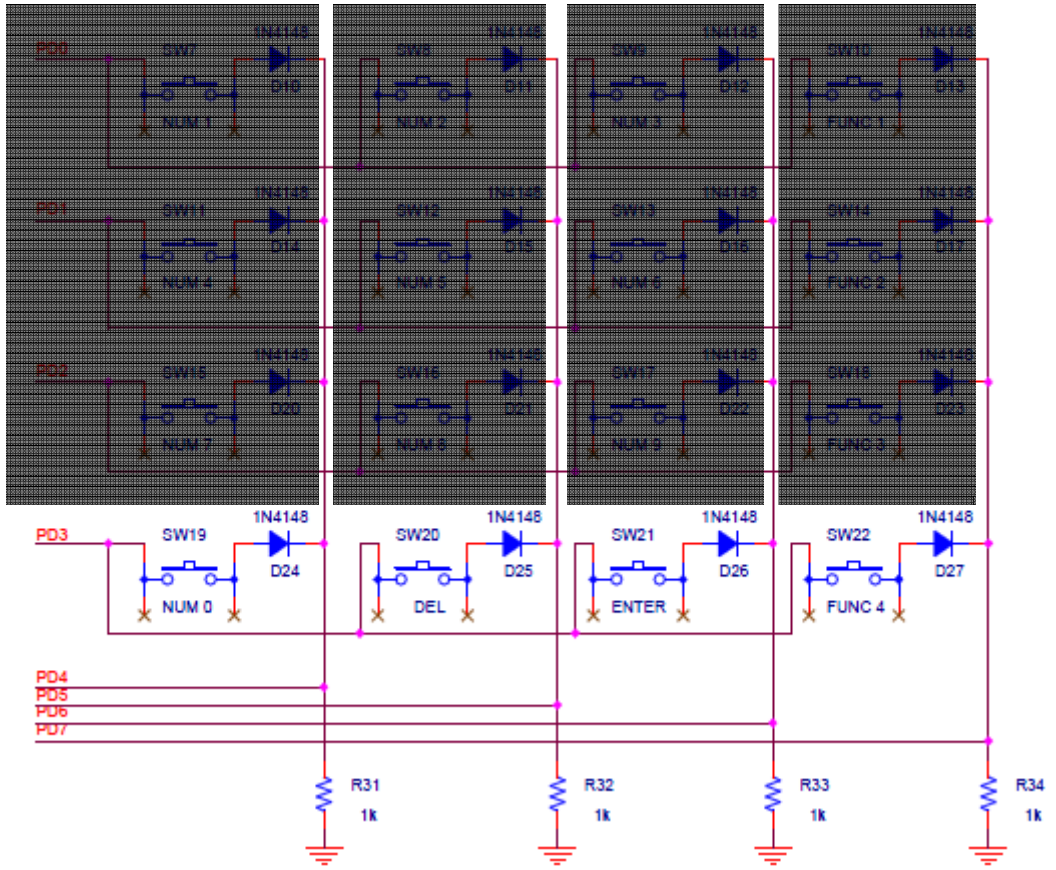

```
PORTD = 0x02;
Key_info = PIND & 0xF0;
```

KEY-PAD



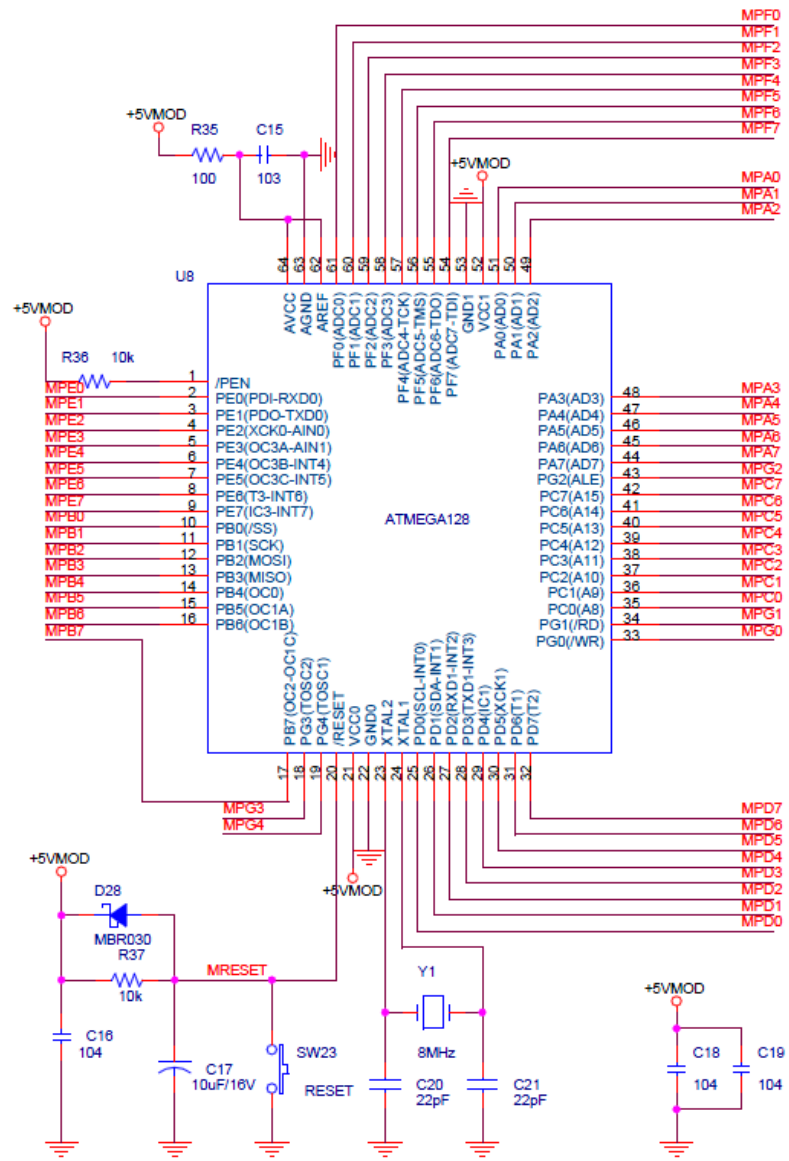

```
PORTD = 0x04;
Key_info = PIND & 0xF0;
```

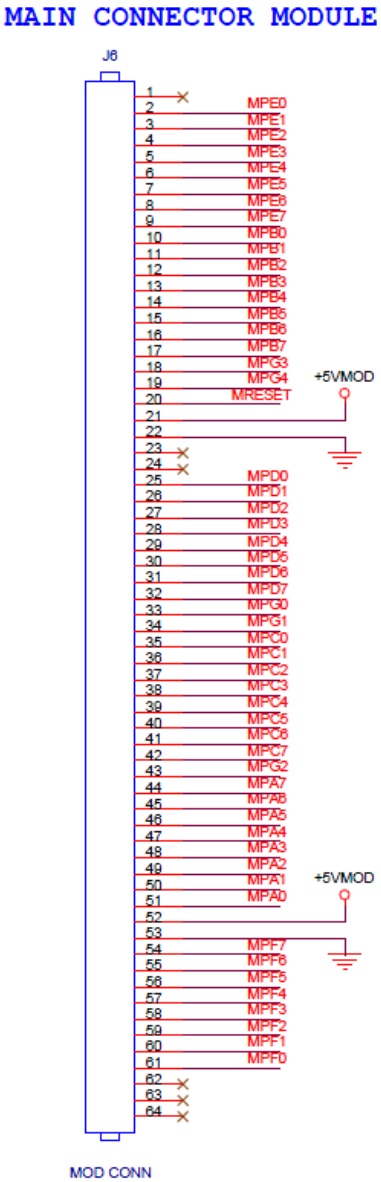
KEY-PAD

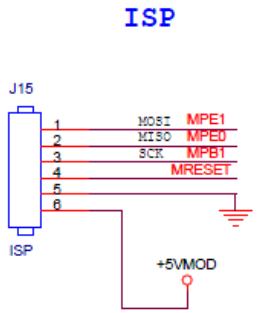



```
PORTD = 0x08;
Key_info = PIND & 0xF0;
```

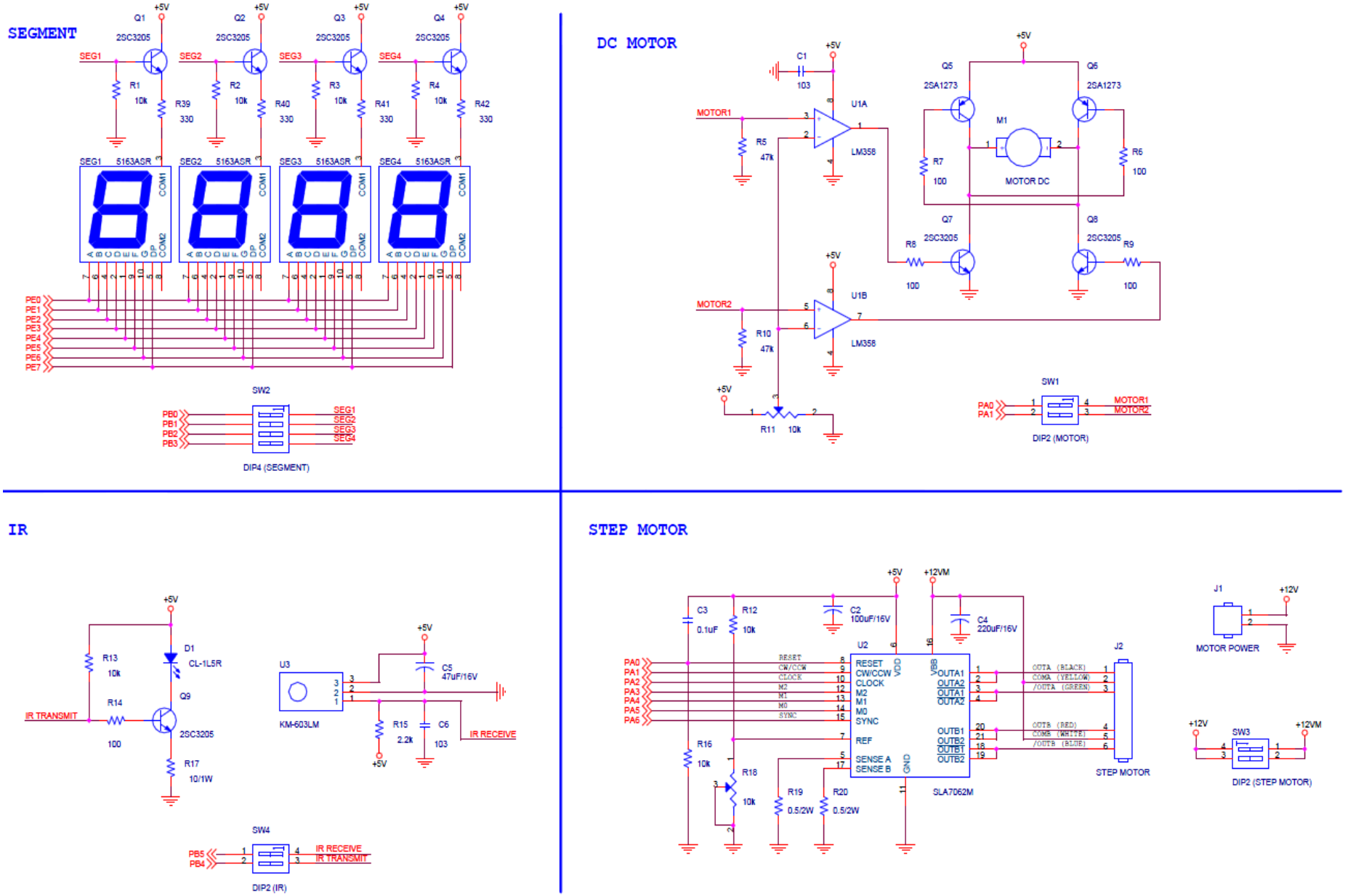
실험 Control Board



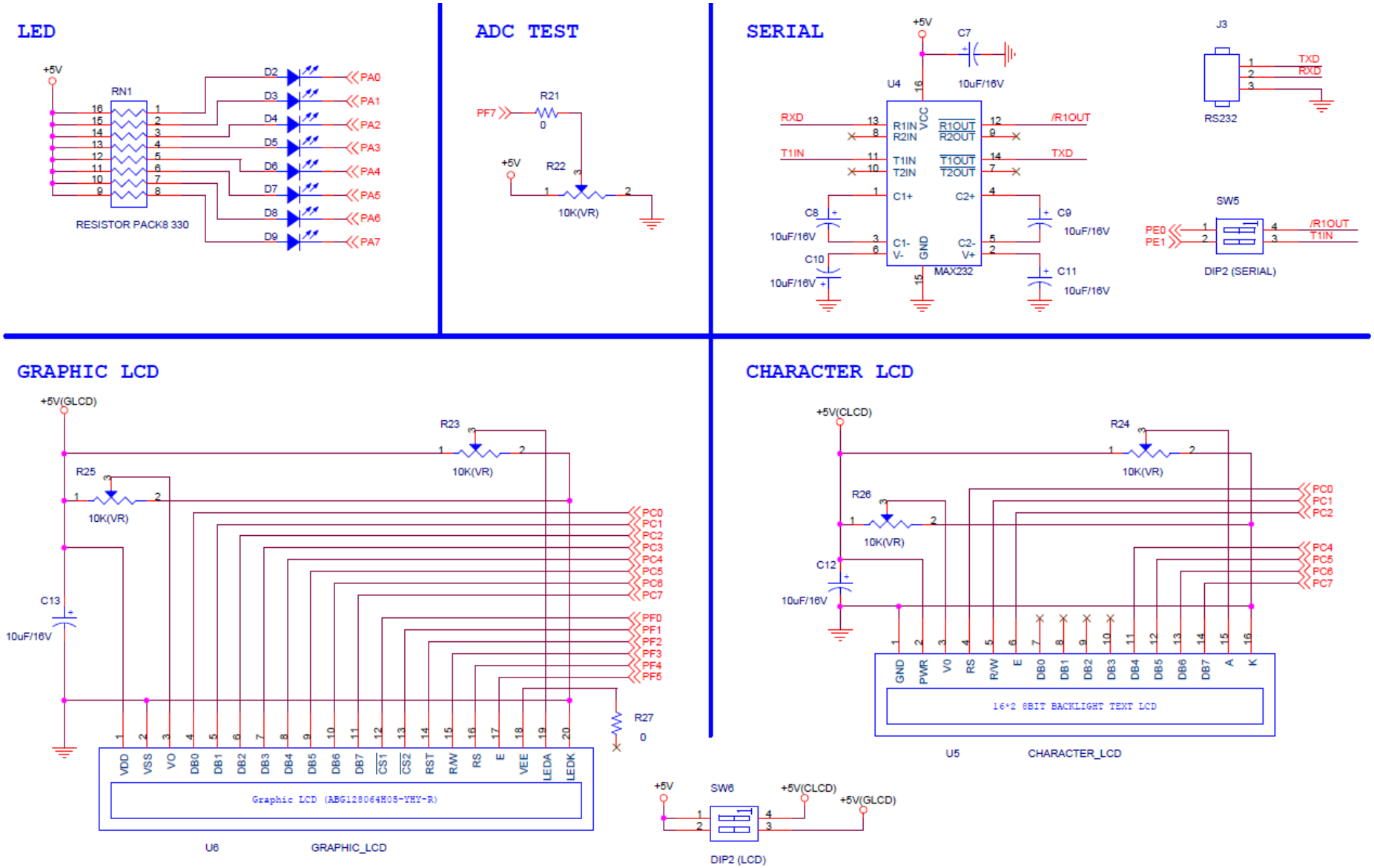




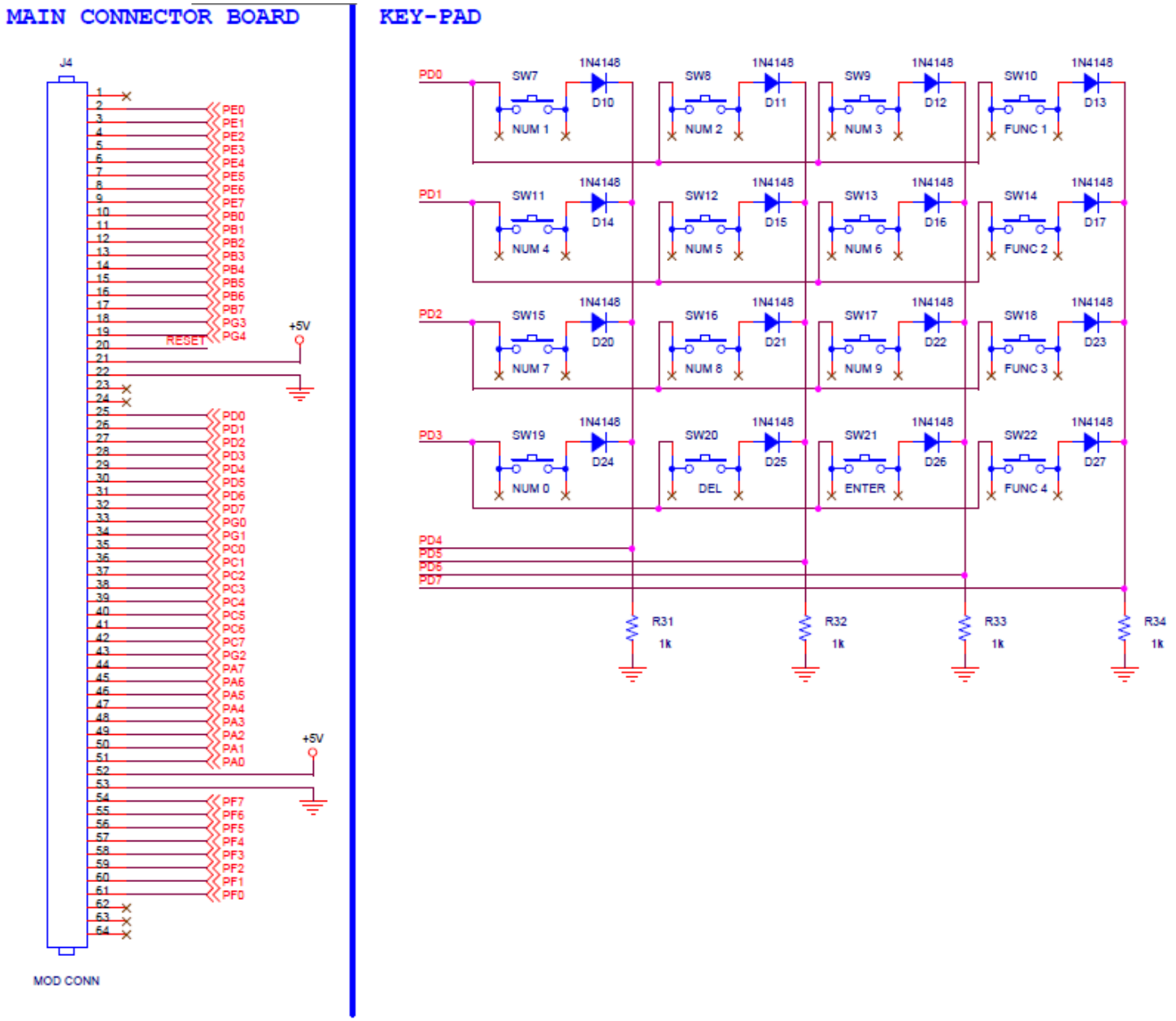
실험 Main Board 1



실험 Main Board 2



실험 Main Board 3



레포트

1. 실험 완료 할 것

반드시 동작을 이해할 것!!!

2. Mission

2-1. 7-Segment 2번째 예제를 기반으로 임의의 외부의 스위치 1개를 이용하여,
시간의 증가를 ON/OFF 하시오

2-2. 4x4 keypad 구현

2-3. (두 개의 7-segment 와 4x4 keypad 이용) 키패드를 누른 숫자 값이 segment 에 나타나도록 하시오.