

논리회로 실습 보고서

2019.11.12

2조 이학민 고예은 이현호

15. FND

1) 기본문제

```
int a=7; //FND의 a를 아두이노 7번과 연결
int b=6; //FND의 b를 아두이노 6번과 연결
int c=5; //FND의 c를 아두이노 5번과 연결
int d=11; //FND의 d를 아두이노 11번과 연결
int e=10; //FND의 e를 아두이노 10번과 연결
int f=8; //FND의 f를 아두이노 8번과 연결
int g=9; //FND의 g를 아두이노 9번과 연결
int dp=4; //FND의 dp를 아두이노 4번과 연결
void digital_1(void) //숫자1 //1은 b, c가 HIGH
{
    unsigned char j; //문자형 변수 j
    digitalWrite(c,HIGH); //c HIGH
    digitalWrite(b,HIGH); //b HIGH
    for( j=7;j<=11;j++) //a,f,g,e,d에 대해
        digitalWrite( j,LOW); //'0' 출력
    digitalWrite(dp,LOW); //dp에도 '0' 출력
}
void digital_2(void) //숫자2 //2는 a, b, g, e, d가 HIGH
{
    unsigned char j; //문자형 변수 j
    digitalWrite(b,HIGH); //b HIGH
    digitalWrite(a,HIGH); //a HIGH
    for( j=9;j<=11;j++) //g, e, d에 대해
        digitalWrite( j,HIGH); //'1' 출력
    digitalWrite(dp,LOW); //dp에는 '0' 출력
    digitalWrite(c,LOW); //c는 '0'
    digitalWrite(f,LOW); //f는 '0'
}
void digital_3(void) //숫자3 //3은 a, b, c, d, g, f가 HIGH
{
    unsigned char j; //문자형 변수 j
    digitalWrite(g,HIGH); //g HIGH
    digitalWrite(d,HIGH); //d HIGH
    for( j=5;j<=7;j++) //a, b, c, d에 대해
        digitalWrite( j,HIGH); //'1' 출력
```

```
for( j=5;j<=7;j++) //a, b, c, d에 대해
digitalWrite( j,HIGH); //'1' 출력
digitalWrite(dp,LOW); //dp에는 '0' digitalWrite(f,LOW); //f는 '0'
digitalWrite(e,LOW); //e는 '0'
}
void digital_4(void) //숫자4 //4는 f, g, b, c가 HIGH
{
digitalWrite(c,HIGH); //c HIGH
digitalWrite(b,HIGH); //b HIGH
digitalWrite(f,HIGH); //f HIGH
digitalWrite(g,HIGH); //g HIGH
digitalWrite(dp,LOW); // dp LOW
digitalWrite(a,LOW); //a LOW
digitalWrite(e,LOW); //e LOW
digitalWrite(d,LOW); //d LOW
}
void digital_5(void) //숫자5 //5는 a, f, g, c, d가 HIGH
{
unsigned char j; //문자형 변수 j
for( j=7;j<=9;j++) //a, f, g에 대해
digitalWrite( j,HIGH); //'1' 출력
digitalWrite(c,HIGH); //c HIGH
digitalWrite(d,HIGH); //d HIGH
digitalWrite(dp,LOW); //dp LOW
digitalWrite(b,LOW); //b LOW
digitalWrite(e,LOW); //e LOW
}
void digital_6(void) //숫자6 //6은 a, f, e, d, c, g가 HIGH
{
unsigned char j; //문자형 변수 j
for( j=7;j<=11;j++) //a, f, e, d, g 에 대해
digitalWrite( j,HIGH); //'1' 출력
digitalWrite(c,HIGH); //c HIGH
digitalWrite(dp,LOW); //dp LOW
digitalWrite(b,LOW); //b LOW
}
}
```

```

digitalWrite(dp,LOW); //dp LOW |
digitalWrite(b,LOW); //b LOW
}
void digital_7(void) //숫자7 //7은 a, b, c, f가 HIGH
{
    unsigned char j; //문자형 변수 J
    for( j=5;j<=7;j++) //a, b, c에 대해
        digitalWrite( j,HIGH); //HIGH digitalWrite(dp,LOW); //dp는 LOW
    for( j=8;j<=11;j++) //f, g, e, d에 대해
        digitalWrite( j,LOW); //'0' 출력
}
void digital_8(void) //숫자8 //8은 a, b, c, d, e, f, g가 HIGH
{
    unsigned char j; //문자형 변수 J;
    for( j=5;j<=11;j++) //a, b, c, d, e, f, g 에 대해
        digitalWrite( j,HIGH); //'1'을 출력
    digitalWrite(dp,LOW); //dp는 LOW
}
void digital_9(void) //숫자9 //9는 a, b, c, d, f, g가 HIGH
{
    digitalWrite(a,HIGH); //a HIGH
    digitalWrite(b,HIGH); //b HIGH
    digitalWrite(c,HIGH); //c HIGH
    digitalWrite(d,HIGH); //d HIGH
    digitalWrite(e,LOW); //e LOW
    digitalWrite(f,HIGH); //f HIGH
    digitalWrite(g,HIGH); //g HIGH
    digitalWrite(dp,HIGH); //dp HIGH
}
void setup()
{
    int i; //변수 i
    for(i=4;i<=11;i++) //4~11에 대해
        pinMode(i,OUTPUT); //pinMode는 출력
}
void loop()

```

```
}  
void loop()  
{  
  while(1) //무한반복  
  {  
    digital_1(); //숫자1  
    delay(1000); //대기 1초  
    digital_2(); //숫자2  
    delay(1000); //대기1초  
    digital_3(); //숫자3  
    delay(1000); //대기 1초  
    digital_4(); //숫자4 |  
    delay(1000); //대기 1초  
    digital_5(); //숫자5  
    delay(1000); //대기 1초  
    digital_6(); //숫자6  
    delay(1000); //대기 1초  
    digital_7(); //숫자7  
    delay(1000); //대기 1초  
    digital_8(); //숫자8  
    delay(1000); //대기 1초  
    digital_9(); //숫자9  
    delay(1000); //대기 1초  
  }  
}
```

2) 심화학습

- (1) 스위치가 눌러졌을 때, 확실하게 0을 검출하기 위해서입니다.
- (2) FND를 모두 연결한 뒤, LED에 불이 들어오는지를 통해 구별할 수 있다.
- (3) 괄호 안의 조건이 1(TRUE)이 되면 while문 안에 있는 문장들을 계속 수행하고 괄호 안의 조건이 0(FALSE)가 되면 작동을 정지합니다.
- (4) while문과 달리 do를 무조건 한번 수행한 후 값을 비교합니다.
- (5) 1은 TRUE를 의미하며 while루프 안의 문장을 무한으로 수행합니다.
- (6) 홀수 번만 나오게 코딩

```
void loop()
{
  while(1) //무한반복
  {
    digital_1(); //숫자1
    delay(1000); //대기 1초
    digital_3(); //숫자3
    delay(1000); //대기 1초
    digital_5(); //숫자5 |
    delay(1000); //대기 1초
    digital_7(); //숫자7
    delay(1000); //대기 1초
    digital_9(); //숫자9
    delay(1000); //대기 1초
  }
}
```

3) 창의문제

- 9부터 1까지 숫자가 줄어들도록 코드를 수정한다.

```
void loop()
{
    while (1) //무한반복
    {
        digital_9(); //숫자9
        delay(1000); //대기 1초
        digital_8(); //숫자8
        delay(1000); //대기 1초
        digital_7(); //숫자7
        delay(1000); //대기 1초
        digital_6(); //숫자6
        delay(1000); //대기 1초
        digital_5(); //숫자5
        delay(1000); //대기 1초
        digital_4(); //숫자4
        delay(1000); //대기 1초
        digital_3(); //숫자3
        delay(1000); //대기 1초
        digital_2(); //숫자2
        delay(1000); //대기1초
        digital_1(); //숫자1
        delay(1000); //대기 1초
    }
}
```

16. 4개의 FND

1) 기본문제

```
#define SEG_A 2 //Arduino Pin2--->SegLed Pin11
#define SEG_B 3 //Arduino Pin3--->SegLed Pin7
#define SEG_C 4 //Arduino Pin4--->SegLed Pin4
#define SEG_D 5 //Arduino Pin5--->SegLed Pin2
#define SEG_E 6 //Arduino Pin6--->SegLed Pin1
#define SEG_F 7 //Arduino Pin7--->SegLed Pin10
#define SEG_G 8 //Arduino Pin8--->SegLed Pin5
#define SEG_H 9 //Arduino Pin9--->SegLed Pin3
#define COM1 10 //Arduino Pin10--->SegLed Pin12
#define COM2 11 //Arduino Pin11--->SegLed Pin9
#define COM3 12 //Arduino Pin12--->SegLed Pin8
#define COM4 13 //Arduino Pin13--->SegLed Pin6
unsigned char table[10][8] = //10개의 숫자, 8개ledpin
{
    {0, 1, 1, 1, 1, 1, 1, 1},
    //0
    {0, 0, 0, 0, 0, 1, 1, 0},
    //1
    {0, 1, 0, 1, 1, 0, 1, 1},
    //2
    {0, 1, 0, 0, 1, 1, 1, 1},
    //3
    {0, 1, 1, 0, 0, 1, 1, 0},
    //4
    {0, 1, 1, 0, 1, 1, 0, 1},
    //5
    {0, 1, 1, 1, 1, 1, 0, 1},
    //6
    {0, 0, 0, 0, 0, 1, 1, 1},
    //7
    {0, 1, 1, 1, 1, 1, 1, 1},
    //8
    {0, 1, 1, 0, 1, 1, 1, 1}
    //9
};
```

```
void setup()
{ //pinMode OUTPUT 설정
pinMode(SEG_A, OUTPUT);
pinMode(SEG_B, OUTPUT);
pinMode(SEG_C, OUTPUT); pinMode(SEG_D, OUTPUT);
pinMode(SEG_E, OUTPUT);
pinMode(SEG_F, OUTPUT);
pinMode(SEG_G, OUTPUT);
pinMode(SEG_H, OUTPUT);
pinMode(COM1, OUTPUT);
pinMode(COM2, OUTPUT);
pinMode(COM3, OUTPUT);
pinMode(COM4, OUTPUT);
}
void loop()
{
  Display(1,1); //첫 번째 FND, 1출력
  delay(500);
  Display(2,2); //두 번째 FND, 2출력
  delay(500);
  Display(3,3); //세 번째 FND, 3출력
  delay(500);
  Display(4,4); //네 번째 FND, 4출력
  delay(500);
}
void Display(unsigned char com, unsigned char num)
{
  digitalWrite(SEG_A, LOW);
  digitalWrite(SEG_B, LOW);
  digitalWrite(SEG_C, LOW);
  digitalWrite(SEG_D, LOW);
  digitalWrite(SEG_E, LOW);
  digitalWrite(SEG_F, LOW);
  digitalWrite(SEG_G, LOW);
}
```

```

switch (com)
{
case 1: //첫 번째 case
digitalWrite(COM1,LOW); //COM1이 '0'이면 digitalWrite(COM2,HIGH);
digitalWrite(COM3,HIGH);
digitalWrite(COM4,HIGH);
break;
case 2: //두 번째 case
digitalWrite(COM1,HIGH);
digitalWrite(COM2,LOW); //COM2가 '0'이면
digitalWrite(COM3,HIGH);
digitalWrite(COM4,HIGH);
break;
case 3: //세 번째 case
digitalWrite(COM1,HIGH);
digitalWrite(COM2,HIGH);
digitalWrite(COM3,LOW); //COM3이 '0'이면
digitalWrite(COM4,HIGH);
break;
case 4: //네 번째 case
digitalWrite(COM1,HIGH);
digitalWrite(COM2,HIGH);
digitalWrite(COM3,HIGH);
digitalWrite(COM4,LOW); //COM4가 '0'이면
break;
default:break;
}
digitalWrite(SEG_A,table[num][7]);
digitalWrite(SEG_B,table[num][6]);
digitalWrite(SEG_C,table[num][5]);
digitalWrite(SEG_D,table[num][4]);
digitalWrite(SEG_E,table[num][3]);
digitalWrite(SEG_F,table[num][2]);
digitalWrite(SEG_G,table[num][1]);
digitalWrite(SEG_H,table[num][0]);
}

```

2) 심화학습

(1) 숫자 4568을 출력한다.

```
void loop()
{
    Display(1,4); //첫 번째 FND, 4출력
    delay(500);
    Display(2,5); //두 번째 FND, 5출력
    delay(500);
    Display(4,6); //세 번째 FND, 6출력
    delay(500);
    Display(3,8); //네 번째 FND, 8출력
    delay(500);
}
```

(2) 각각의 케이스는 고유한 값을 가지고 있는데, 전달 인자로 들어오는 같은 케이스문만 실행되고 나머지는 무시되는 형식이다. 이때, case 값으로는 실수와 문자열만 올 수 있다.

3) 창의문제

- FND의 숫자 출력 순서를 거꾸로 바꿔본다. (오른쪽부터)

```
void loop()
{
    Display(4,1); //첫 번째 FND, 1출력
    delay(500);
    Display(3,2); //두 번째 FND, 2출력
    delay(500);
    Display(2,3); //세 번째 FND, 3출력
    delay(500);
    Display(1,4); //네 번째 FND, 4출력
    delay(500);
}
```

17. 74HC595

1) 기본문제

```
_17_74HC595$  
//connect 74hc595 pin10:MR--->VCC; Pin13:OE--->GND  
int latchPin = 5; //to 595 pin12  
int clockPin = 4; //to 595 pin11  
int dataPin = 2; //to 595 pin14  
void setup ()  
{  
  pinMode(latchPin,OUTPUT);  
  pinMode(clockPin,OUTPUT);  
  pinMode(dataPin,OUTPUT); //pinMode 출력  
}  
void loop()  
{  
  for(int a=0; a<256; a++)  
  {  
    digitalWrite(latchPin,LOW); // latchPin '0'  
    shiftOut(dataPin, clockPin, LSBFIRST, a); //(data, clock, LSB/MSB, 실제 쓰여질 data)  
    digitalWrite(latchPin,HIGH); //latchPin '1'  
    delay(1000); //1초 대기  
  }  
}
```

2) 심화학습

(1) 시프트 레지스터가 8비트이며 2의 8제곱은 256이기 때문이다.

(2) 2의 3제곱은 8이므로 LED 3개에서만 ON/OFF 하게 된다.

```
void loop()  
{  
  for(int a=0; a<8; a++)  
  {  
    digitalWrite(latchPin,LOW); // latchPin '0'  
    shiftOut(dataPin, clockPin, LSBFIRST, a); //(data, clock, LSB/MSB, 실제 쓰여질 data)  
    digitalWrite(latchPin,HIGH); //latchPin '1'  
    delay(1000); //1초 대기  
  }  
}
```

(3) MSB: Most Significant Bit : 가장 큰 자릿수의 비트, 즉 가장 왼쪽 비트

LSB: Least Significant Bit: 가장 작은 자릿수의 비트, 즉 가장 오른쪽 비트

(4) shiftOut함수

shiftOut(dataPin, clockPin, LSBFIRST, leds)에서 각각의 의미

- dataPin : DS 핀 넘버
- clockPin : Clock 핀 넘버
- LSBFIRST : LSB비트로부터 전송하는 것으로 1byte 데이터 중 가장 오른쪽의 데이터로부터 전송한다는 뜻을 가지고 있다.
- leds : 8개의 비트 상태값을 담은 1byte 데이터

3) 창의문제

- LED점멸 속도를 빠르게 바꾸고 LSBFIRST를 MSBFIRST로 바꾸어 가장 왼쪽 데이터부터 데이터를 전송하여 오른쪽 LED부터 불이 들어오도록 한다.

```
void loop()
{
  for(int a=0; a<256; a++)
  {
    digitalWrite(latchPin,LOW); // latchPin '0'
    shiftOut(dataPin, clockPin, MSBFIRST, a); //(data, clock, LSB/MSB, 실제 쓰여질 data)
    digitalWrite(latchPin,HIGH); //latchPin '1'
    delay(100); //1초 대기
  }
}
```

18. 서보모터제어

1) 기본문제

```
_18_servomotor$
#include<Servo.h>
//UART send 1~9==>20~180 degree
int servopin=9; //servopin을 D9로 초기화
int myangle; //변수 myangle 선언
int pulsewidth; //변수 pulsewidth(신호폭) 선언
int val=0; //센서 값 저장
void setup(){
  pinMode(servopin,OUTPUT); //servopin's pinMode : OUTPUT
  Serial.begin(9600); //시리얼통신 전송속도 9600
  Serial.println("servo=0_seral_simple ready" ); //시리얼모니터에 띄움
}
void servopulse(int servopin,int myangle)
{
  pulsewidth=(myangle*11)+500; //myangle : 0((0*11)+500)~180((180*11)+500) degree
  digitalWrite(servopin,HIGH); //servopin HIGH
  delayMicroseconds(pulsewidth); //microsecond delay
  digitalWrite(servopin,LOW); //servopin LOW
  delay(20-pulsewidth/1000); //20ms-millisecond =low value
} void loop()
{
  val=Serial.read();
  if(val>'0'&&val<='9') //시리얼모니터에 0~9까지 입력
  {
    val=val-'0';// =0x39-0x30=9 |
    val=val*(180/9); //1마다 20degree씩 변화(20~180degree)
    Serial.print("moving servo to ");
    Serial.print(val,DEC);
    Serial.println();
    for(int i=0;i<=180;i++)
    {
      servopulse(servopin,val); //i = val
    }
  }
}
```


2) 심화학습

(1) 1마다 30도씩 변화하도록 변경

```
if (val > '0' && val <= '6')
{
    val = val - '0';
    val = val * (180 / 6);
    Serial.print("moving servo to ");
    Serial.print(val, DEC);
    Serial.println();
    for (int i = 0; i <= 180; i++)
    {
        servopulse(servopin, val);
    }
}
```

(2) 서보모터로 제어할 수 있는 RC카의 바퀴

- 자동차의 회전, 즉 앞바퀴의 좌/우 이동을 담당하는 부품임.

(3) 서보모터로 제어할 수 있는 다른 예시

- 선풍기 같은 회전운동을 하는 것.

(4) Val = val-'0'을 해주는 이유에 대해 생각해봅시다.

- 만약 Val에 9를 넣는다면 Val의 아스키코드는 Val = 0x39이다. 그리고 0의 아스키코드는 0x30이므로 9의 값을 받기 위해서는 0x39 - 0x30 을 해야 한다. 하지만 실제로 서보모터 각도를 처리하기 위해서는 아스키코드가 아닌 숫자 9가 필요하기때문에 0을 빼주는 것이다.

3) 창의문제

- 회전각이 180도로 제한되어있었던 것을 270도까지 확장시키고 숫자 1당 40도를 돌 수 있도록 코드를 수정한다.

```
val=Serial.read();  
if(val>'0'&&val<='9') //시리얼모니터에 0~9까지 입력  
{  
  
    val=val-'0';// =0x39-0x30=9  
    val=val*(270/9); //1마다 20degree씩 변화(40~270degree)  
    Serial.print("moving servo to ");  
    Serial.print(val,DEC);  
    Serial.println(); |  
    for(int i=0;i<=270;i++)  
    {  
        servopulse(servopin,val); //i = val  
    }  
}}
```