실험 6. 전자 자물쇠 구현하기

이번에는 잠시 쉬어 가는(?) 작업을 하여보도록 하겠습니다. 앞에서 keypad와 display를 동시에 사용하는 방법을 실험하여 보았는데, 이를 이용하는 응용의 한 예로 전자 자물쇠를 만들어 보기로 하겠습니다. 요즘 지문 인식이니 동체 인식이니 하는 인식기가 많이 이야기되고 있는데, 가장 기본적인 기능이 암호를 인식하여 동작하는 것이 아닐까요. 암호의 인식은 크게 암호가 저장된 카드를 이용하여 사람이 직접적인 숫자 조작 없이 암호를 전송하여 인식하는 형태(신분증 형태로 만들어져 출입 통제 장치에 사용)와 암호인식기에 부착된 keypad를 이용하여 직접 사람이 암호 숫자를 넣어서 사용하는 2가지 방법(주로 소규모의 출입 통제나 집 등에서 사용하고 있음)이 있지요. 이중 후자를 한번 만들어 보도록 합시다. 방법은 크게 두 가지로 생각해 볼 수 있습니다. 먼저 표시기 1개와 숫자 key 10개로 하여 암호를 바로 숫자 key로 넣어 주는 것입니다. 생각은 단순하나 어떻게 보면 회로도 복잡하고 키 인식 프로그램도 복잡하지요. 앞 실험을 그대로 이용하면 구현이 가능합니다. 다음으로는 표시기 1개와 key 1개만을 사용하여도 구현이 가능하지요. key 1개로 어떻게 암호를 입력할 수 있는가? key를 누르는 시간차로 인식되는 숫자를 다르게 나타내도록 하는 것입니다. 즉 key를 계속 누르고 있으면 숫자가 0에서 9까지 순서대로 증가되도록 하고 key를 놓는 순간 그 값이 입력 key 값이 되도록 하면 됩니다. 어느 것이 더 구현하기가 쉽겠지요? 본 실험에서는 가능하면 다른 형태의 프로그램과 주변 회로를 최소로 할 수 있는 후자를 선택하도록 합시다.

1. 전자 자물쇠를 구현하기 위한 회로 설계

암호를 입력하기 위해서는 최소 회로가 1개의 key와 1개의 display가 사용됩니다. 그러나 실제로 사용 가능한 구성이 되려면 대문 안쪽에서 수동으로 문을 여닫는 key가 필요하며, 또 부저 소리 및 문을 여닫는 출력이 필요합니다. 즉 key 입력으로 3개, 표시기로 7개 그리고 출력으로 2개가 필요하므로 총 12개가 사용됩니다. 이는 PIC16F876으로 직접 구현 가능한 범위이며, 이것이 후자를 선택한 이유이지요.

앞의 표시기 프로그램의 기본 기능을 그대로 사용하기 위하여 7-segment에 연결되는 방법은 그대로 하면, RC7, 6, 5, 4, 3, 2, 1, 0를 7-segment의 A, B, C, D, E, F, G, DP 단자에 연결하고 DG4(COM4) 는 V_{CC}(5V)에 연결하면 되지요. 그리고 부져는 RA5, 자물쇠 출력은 RA3으로 하여 LED1에 연결합시다. 또 암호 key 입력은 RA0에 수동 열림은 RA1, 수동 닫힘은 RA2로 하면 모든 I/O의 배정 끝나지요.

2. 전자 자물쇠를 구현하기 위한 flow chart 설계

실제로 사용 가능한 프로그램이 되려면 기능이 복잡해지고 프로그램이 길어지므로 그냥 프로그램 하려고 하지 말고 먼저 어떻게 동작시킬 것인가를 설계하라고 강조했지요. 다시 강조 하지만 마이크로프로세서는 하드웨어를 프로그램으로 동작시키는 것이므로 하드웨어와 프로그램을 동시에 알아야 합니다. 명령어를 명령어 자체의 의미만으로 이해하지 말고 프로그램을 보기 전에, 작성하기 전에 내가 하려는 일이 무엇인가를 하드웨어를 그려서 손으로 따라 가면서 이해해야 합니다. 그리고 회로도를 완벽하게 그리지 않는 이유도 여기에 있습니다. Pin 번호까지 기록된 회로도는 여러분을 data sheet에서 멀어지게 하며, 아무런 생각 없이 회로도대로 선을 연결하는 습관이 생기지요. 이제는 여러분이 design 하는 것입니다. 절대 복잡한 것이 아닙니다. 그래서 최소한의 회로소자로 구현 가능한 것을 실험하고 있는 것입니다.

먼저 하려고 하는 것을 구체화 시켜보도록 하겠습니다. key가 1개이므로 1개로 여러 값을 입력하는 방법을 찾아보아야지요. 기본 방법은 한가지이지요. key를 누르는 시간차를 이용하는 것입니다. key를 짧게 누르면 누를 때마다 숫자를 증가시키고, 특정시간 누르지 않으면 그 숫자를 입력 key로 인식하는 것입니다. 또는 key를 계속 누르고 있으면 숫자가 증가하도록 하고 key를 놓으면 그때 값을 입력 숫자로 인식하게 하는 것입니다. 어느 것이 쉽지요. 일단은 후자를 선택해 보도록 합시다.

그럼 동작 단계를 말로 써보겠습니다.

① 전원이 들어가면 잠긴 상태가 되도록 한다.
- 부저 off, LED off, 숫자 표시 off

② key가 눌러짐을 확인하며, 눌러지지 않으면 2)를 반복한다.
- key가 눌러지면 key 종류를 확인하여 각자 기능으로 분지한다.

③ 수동 key 일 때
- 수동 잠김일 때는 잠시 동안 부저 on 후 LED off 후 2)로 이동
- 수동 열림일 때는 잠시 동안 부저 on 후 LED on 후 2)로 이동

④ 암호 key 일 때
- 암호 입력 key 이면 눌려진 시간에 따라서 숫자를 0에서 증가시켜 계속 순환되도록 한다. 당연히 표시가 되어야죠!
- key 눌러짐이 끝나면 현재 표시값이 입력 숫자가 되어 저장되도록 함.
- 주어진 암호 개수만큼 암호가 들어와 저장되면 내부에 기록된 암호와 들어온 암호를 비교하여
  - 다르면 부저 on 후 LED off 후 2)로 이동
  - 같으면 부저 on 후 LED on 후 2)로 이동

⑤ ??변경 기능이 더 필요합니다.

위 흐름을 보면 동작이 머리에 그려집니까? 혹시 추가해야 될 내용이 없는지 확인해 봅시다. ??변경 기능은 무엇일까요? 먼저 생각해 보세요. (최소한 5분간!!!)

앞의 말을 그림으로 그린 것이 flow chart이지요.

우리가 어떤 프로그램을 작성할 때는 위의 흐름을 여러 각도에서 검토해야만 발생될 수 있는 문제점을 최소화시킬 수 있습니다. 그리고 가능하면 단위 기능으로 나누어 작성해야만 다음에 수정하기가 편리합니다. 위 프로그램에서는 display 기능과 delay 기능은 단위 기능으로 나눌 수 있고, 따라서 부 프로그램으로 작성하는 것이 유리하지요. 부 프로그램으로 작성된 것은 data를 변수로 주고받도록 하고 main에서는 변수에 data를 넣어 주면됩니다. 앞의 실험에서 이렇게 사용하였습니다.

앞의 일 중에서 가장 복잡한 것이 단계 4)입니다. 암호를 받아들이는 부분이지요. 이 부분을 자세히 나누어 설명하면 아래와 같은 단계로 구현하는 것이 가능합니다. 여기서 설명한 단계는 물론, 하나의 예이고 작성하는 방법은 여러 가지가 있지요.

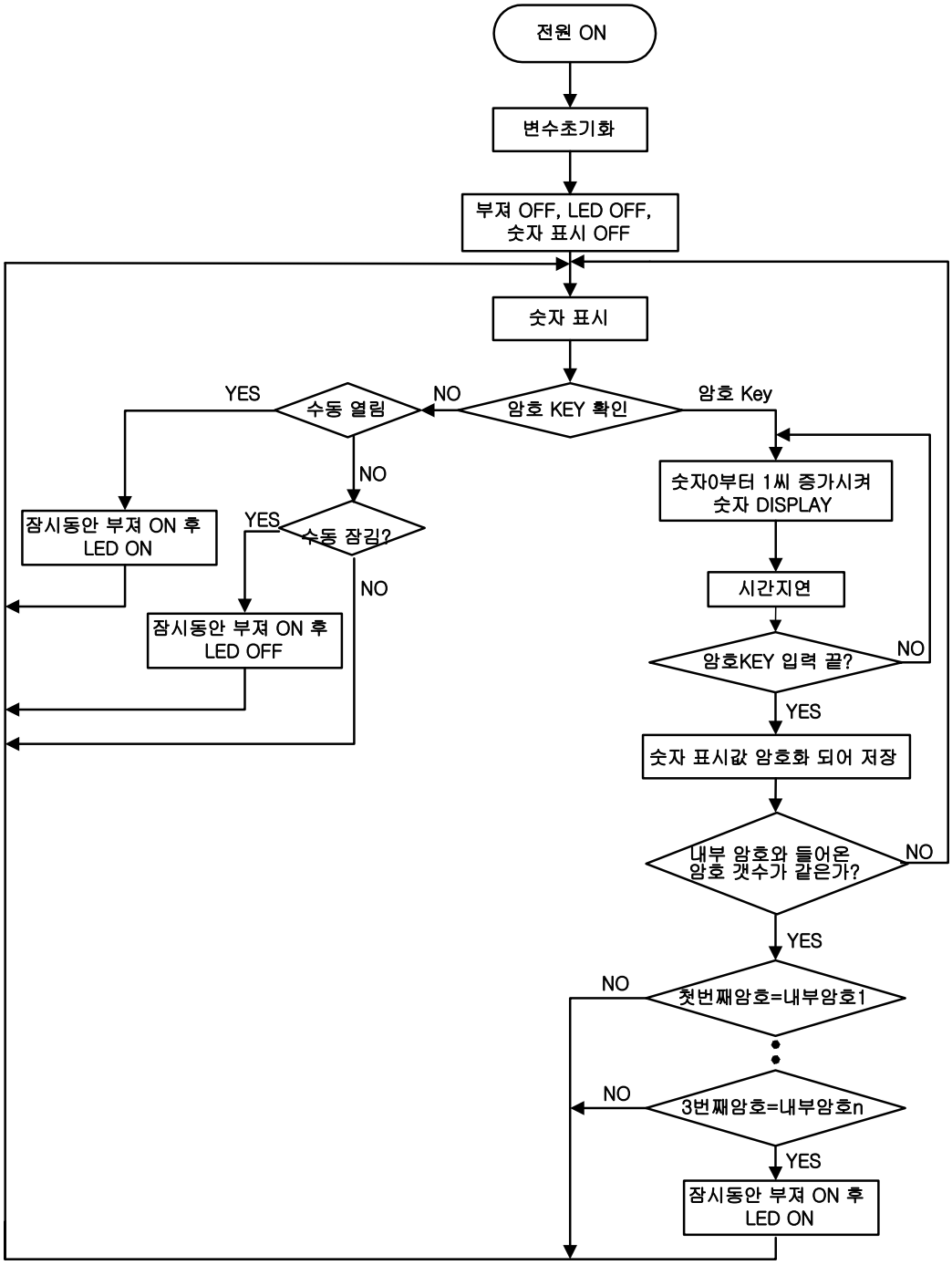

그림 E6-1. 전자 자물쇠 flow chart

① 암호 입력 key가 눌러지면 초기 표시값으로 '0'를 표시한다.

② 시간 지연을 시킨다.

③ 암호 입력 key가 계속 눌러져 있으면, 표시값을 증가시킨다.
그리고 ②로 반복.

④ 암호 입력 key가 눌러짐이 해제되면, 현재 표시되는 값을 암호 저장 buffer에 저장시킨다. 그리고 저장 개수가 암호 길이와 같은가를 확인한다.
- 길이가 다르면 단계 ②로 이동
- 길이가 같으면 단계 ⑤로 이동

⑤ 길이가 같으면
- 미리 저장된 암호 숫자와 들어온 암호를 비교하여
- 전부 같으면 부저 on 후 LED on 후 2)로 이동
- 다르면 부저 on 후 LED off 후 2)로 이동

이렇게 세부적으로 나누어 보면 프로그램의 흐름이 눈에 보이지요. 그러나 문제점은 아직도 많이 남아 있을 것 같지 않나요? 생각해 보세요. (1분간)

Flow chart를 그린다는 것은 꼭 그림을 그린다는 것이 아닙니다. 이렇게 흐름을 만들어 보라는 것이며, 말로 써도 됩니다. 이것을 조금 어려운 말로는 algorithm(?) 이라고 하지요. 문제점을 찾으셨나요! 암호를 저장하는 buffer를 관리하는 부분이 없습니다. 즉 초기에 암호를 넣는 buffer를 당연히 설정해야 하며, 위치 counter도 초기화 해야 하지요. 이것은 상식이며, 지금 빠진 부분은 암호 key가 일정 시간 동안 눌러지지 않으면 다시 초기화를 해야 한다는 것입니다. 왜냐하면 누군가 암호 key를 한번 눌러 놓으면, 다음 들어온 암호는 2번째인가요? 잘못 눌렀을 때에는 잠시 기다렸다가 처음부터 다시 시작되어야 하겠지요. 따라서 암호 key가 눌러지지 않은 시간을 감지하는 기능이 추가되어야 하지요.

아이고 머리야 !! 지구가 돌아가니 머리가 아픈 것이지 어려워서는 아니지요. 여러분 지구가 돌고 있는 것을 느낄 수 있습니까?

이 부분까지 전부 처음부터 고려한다는 것은 너무 따져야 할 것이 많으므로 여러분에게 남겨 두고 구체적인 프로그램 설명에서는 가장 단순한 기능만 따라 가 보도록 하겠습니다.

☞ 프로그램을 작성할 때 처음부터 모든 것을 다 고려하여 작성하면, 너무 복잡해져서 coding 상의 문제점을 볼 수가 없습니다. 따라서 구체적인 프

로그램 작업은 가장 단순한 기능부터 하나씩 구현하여 개별적으로 확인한 다음 집합하는 것이 훨씬 좋은 방법입니다. 즉 부 프로그램을 많이 활용하는 것이 유리하지요.

3. 전자 자물쇠를 구현하기 위한 기본 program 작성

프로그램에서 사용하는 변수를 정의합시다. 암호 위치를 지정하는 counter 변수로 PASS_C, 암호를 저장하는 buffer로 PASS_1,2,3,4를 정의한다.(현재는 암호를 4자리 숫자로 정의하였으므로 4개를 지정한 것이다.) 그리고 들어온 암호 숫자를 임시로 저장하고 표시하는 변수로 DISP_B를 사용한다.

예1)

```
        ; 앞의 HEADER에 해당하는 부분

        ; 사용자 변수 선언
        ;       --> 추가로 해야되지요.

        ; I/O를 설정하는 부분
        ;       --> 회로의 의미에 맞게 PORTA, PORTC를 선언

        ;  MAIN PROGRAM
        ;  ① 단계 부분
              BCF       PORTA,5     ; 부져 off 설정
              BCF       PORTA,3     ; 출력 LED1 off
              CALL      CLEARP      ; 암호 초기화 부분
        ;  ② 단계 부분
        LP    MOVF      DISP_B,W    ; 표시할 값을 W로
              CALL      DISP        ; W 값을 표시함
              BTFSS     PORTA,0     ; 암호 입력 스위치 눌러짐 확
                                      인
              GOTO      PS0
              BTFSS     PORTA,1     ; 수동 열림 스위치 눌러짐 확
```

```
                                    인
        GOTO        PS1
        BTFSS       PORTA,2     ; 수동 닫힘 스위치 눌러짐
                                    확인
        GOTO        PS2
        GOTO        LP
; ③ 단계 부분
PS1     BSF         PORTA,3     ; 자물쇠 on
        BSF         PORTA,5     ; 부져 on
        CALL        DELAY_1
        BCF         PORTA,5     ; 부져 off
        GOTO        LP1
PS2     BCF         PORTA,3     ; 자물쇠 off
        BSF         PORTA,5     ; 부져 on
        CALL        DELAY_1
        BCF         PORTA,5     ; 부져 off
        GOTO        LP1
;  암호 위치 초기화
LP1     CALL        CLEARP      ; 암호 초기화 부분
        GOTO        LP
;  ④ 단계 부분 --  암호 입력 부분
PS0     INCF        DISP_B      ; 초기값을 '0' 으로 만듬
        MOVF        DISP_B,W
        CALL        DISP
        CALL        DELAY
        BTFSC       PORTA,0     ; 암호 입력 스위치 눌러짐 확
                                    인
        GOTO        PSOFF
; 암호 입력 스위치가 계속 눌러짐
        GOTO        PS0
; 암호 입력 스위치가 놓여짐
PSOFF
; 저장 위치 구분
```

```
        MOVF    PASS_C,W
        ANDLW   03H         ; 4 자리로 제한
        ADDWF   PCL,F
        GOTO    SS_1
        GOTO    SS_2
        GOTO    SS_3
        GOTO    SS_4
SS_1    MOVF    DISP_B,W
        MOVWF   PASS_1      ; 1번째 들어온 암호 저장
        GOTO    LP3
SS_2    MOVF    DISP_B,W
        MOVWF   PASS_2      ; 2번째 들어온 암호 저장
        GOTO    LP3
SS_3    MOVF    DISP_B,W
        MOVWF   PASS_3      ; 3번째 들어온 암호 저장
        GOTO    LP3
; 한 개의 암호가 들어옴 --> 다음 위치 지정 및 표시값 초기화
LP3     INCF    PASS_C,F
        MOVLW   0FFH
        MOVWF   DISP_B
        GOTO    LP
; 마지막 암호가 들어옴
SS_4    MOVF    DISP_B,W
        MOVWF   PASS_4      ; 4번째 들어온 암호 저장
; 암호가 다 들어왔으므로 암호 검사
        MOVLW   1           ; 1번째 내부 암호 값
        SUBWF   PASS_1,W
        BTFSS   STATUS,ZF
        GOTO    LP5
        MOVLW   2           ; 2번째 내부 암호 값
        SUBWF   PASS_2,W
        BTFSS   STATUS,ZF
        GOTO    LP5
```

```
        MOVLW   3               ; 3번째 내부 암호 값
        SUBWF   PASS_3,W
        BTFSS   STATUS,ZF
        GOTO    LP5
        MOVLW   4               ; 4번째 내부 암호 값
        SUBWF   PASS_4,W
        BTFSS   STATUS,ZF
        GOTO    LP5
; 모든 암호가 맞으므로 자물쇠 on
        GOTO    PS1
; 암호가 다르므로 자물쇠 off
LP5     GOTO    PS2
; ------------ MAIN END --------------------

; 부프로그램 영역
        CLEARP                  ; 암호 초기화 부분
        CLRF    PASS_C          ; 암호 위치 counter 초기화
        MOVLW   0FFH            ; 암호 들어오지 않았을 때의
                                  초기 : 기억 값
        MOVWF   PASS_1          ; 1번째 들어온 암호 저장 위
                                  치
        MOVWF   PASS_2
        MOVWF   PASS_3
        MOVWF   PASS_4
        MOVWF   DISP_B          ; 들어온 암호 숫자를 임시로
                                  저장하는 변수
; 표시버퍼의 초기치로 0FFH를 사용한 것은 표시기에서 blank 상태로
  만들 수 있게 하기 위함
        RETURN
; W값을 1자리 7-SEGMENT에 표시하기
DISP    CALL    CONV            ; W를 7-SEGMENT 값으로
                                  변경
        MOVWF   PORTC           ; 숫자 값 출력
```

```
        RETURN
; DELAY 프로그램 부분
DELAY_1
;       예전에 사용한 것을 적당히 변경하시오
DELAY
;       예전에 사용한 것을 적당히 변경하시오

        END
```

프로그램을 따라 갈 수 있지요. 말로 설명하는 것보다 프로그램이 오히려 쉽게 느껴질 때가 되면 성공한 것입니다. 이해를 했는가는 예비 문제로 검토해 봅시다.

■ 예비문제 6

예1)의 프로그램에 대하여...

1) 표시버퍼의 초기치로 0FFH를 사용한 것은 초기에 7-segment에 아무 것도 표시되지 않는 blank 상태로 만들 수 있게 하기 위함입니다. 이렇게 동작하기 위해서는 프로그램을 어떻게 작성해야 합니까?

2)
```
        MOVF    PASS_C,W
        ANDLW   03H         ; 4 자리로 제한
        ADDWF   PCL,F
```

에서 03H 값이 4자리로 제한하기 위함은 어떤 의미인가 자세히 설명하시오.

3) 위 프로그램에서 구체적인 4자리 암호는 얼마인가?
4) DELAY_1 과 DELAY를 서로 달리 만든 이유는?
5) 위 프로그램을 범용으로 사용할 수 없다. 그 이유를 설명해 보시오.

■ 실 험 6

1) 예1)를 작성하여 수행시켜 보시오.
그냥 가만히 있으면 문이 자동으로 열리나요? 암호와 암호가 아닌 것을 입력시켜 봐야 되지요.

☞ 암호 key를 누리지 않으면 7-segment에 'F'라는 글자가 표시되고, 암호 key를 누르면 숫자가 0에서부터 증가되어 표시됩니다.

2) DELAY와 DELAY_1의 시간을 조정하여 숫자가 증가하는 속도가 여러분이 조작하기 쉬운 적당한 속도가 되도록 하여 보시오.

3) 암호입력이 잘못될 경우는 프로그램을 처음부터 다시 시작하거나, 수동 닫힘 key를 한번 조작하고 나서 다시 입력하면 되지요. 이유를 프로그램 순서에 따라서 설명해 보시오.

☞ 어떻게 동작하느냐는 프로그램에 의해서 결정되는 것이지, 앞의 설명은 우리 실험의 경우이고, 모든 경우의 정답은 아닙니다.

4) 암호를 다른 값으로 변경하여 보시오.

5) 초기에 'F'가 표시되지 않고 blank가 표시되도록 하시오
(기존 프로그램에서 변경하면 됨.)

6) 암호 숫자 증가가 0에서 9까지만 되도록 프로그램을 변경하여 보시오.

☞ 프로그램 중 예 1)의 단계 4부분의 'IINCF DISP_B'를 변경하여 무작정 변수 'DISP_B'를 증가시키지 않고 '9'이상이 되면 다시 강제로 '0'으로 만들면 되지요.

7) 이 프로그램은 암호가 고정되는 형태이므로 사용자가 암호를 변경 설정할 수 없습니다. 암호를 변경할 수 있도록 프로그램을 변경하시오.

☞ 당연히 사용자 설정 암호가 저장되는 buffer가 최소 4개 필요하지요. 그리고 앞에서 생각해보라는 본문의 5) ??변경 기능에 대한 부분을 추가해야지요. 이때 암호 변경을 위한 새로운 key를 꼭 사용해야 할까요? 하드웨어가 복잡해지는 것보다 프로그램이 복잡해지는 것이 유리하지요. 각자 방법을 생각해 보세요.
이 기능에서는 최소한으로 수동스위치 2개를 동시에 누르지는 않습니다.

이 점을 사용하면..... 아시겠지요!

8) 위의 실험을 완벽히 한 사람은 암호입력이 잘못될 경우 처음부터 다시 암호를 넣을 수 있도록 변경하시오.

☞ 방법1 : 눌렀다 놓여있는 시간의 길이로 구분하여, 길면 초기화하도록.
방법2 : 암호숫자 증가를 0에서 10(A)까지로 하고 10이 되면 초기화되도록 하는 것도 가능 함

.

여러 가지 방법이 있을 수 있으나 사용자 입장에서는 방법**1**이 좋겠지요.

■ HOME WORK 6

1) 예1)의 프로그램은 아직 한가지 문제점이 남아있습니다. 무엇일까요?

☞ 문 안쪽과 바깥쪽에서 동시에 조작할 경우임

2) 1의 이유를 설명하고 간단히 해결방법을 쓰시오.

3) 실험 7)과 8)의 기능을 구현하기 위한 flow-chart를 자세히 쓰시오.

4) keypad를 이용하는 경우, 전자자물쇠를 구현하는 방법을 순서대로 쓰시오.