

# 마이크로프로세서 설계 실험

## 결과보고서

**7조**

2015115818 구병조

2015114501 이창훈

## <과제명>

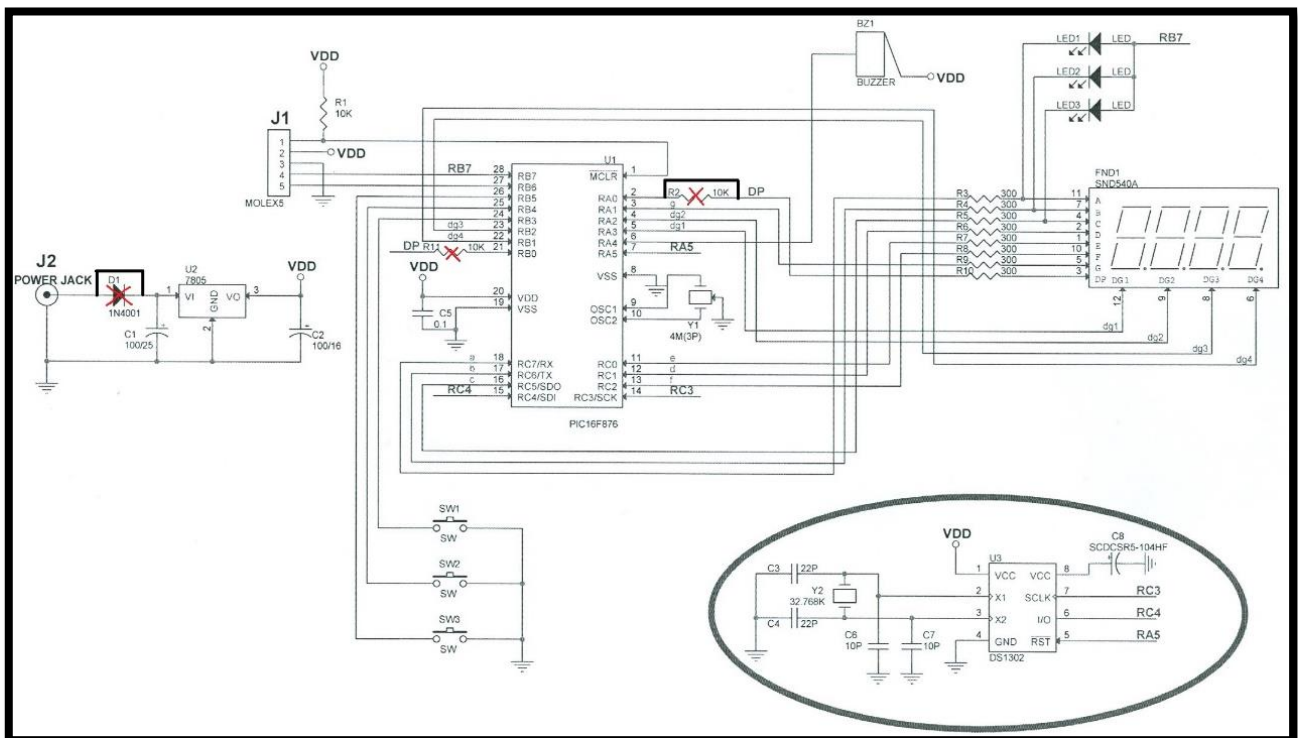
- 게임기 (피아노 타일)
- 스마트폰 게임 어플 '피아노 타일'을 구현해보았습니다.
- 관련 링크 <<https://www.youtube.com/watch?v=anImEkyTI7Q>>

## <구현내용 요약>

- 첫 번째 7-segment에서부터 노트가 랜덤으로 아래로 떨어집니다. (일정한 속도로 노트가 떨어집니다.)
- 마지막 7-segment에 노트가 왔을 때, 해당 위치에 있는 스위치를 누르면 노트가 눌린 것으로 판별되고 버저에서 짧게 소리가 납니다.
- 만약 스위치를 누르는 것에 실패했다면, 7-segment 소수점 자리에 불이 하나씩 들어옵니다. (틀린 횟수 표시 기능 : 총 4번)
- 5번 실패하면 FAIL이 뜨고 게임이 종료되며 버저에서 일정 간격으로 소리가 납니다.
- RESTART 기능으로 다시 게임을 실행할 수 있습니다.
- 노트가 내려오는 속도를 수동으로 조절할 수 있습니다. (보통, 빠르게)

## <인터페이스 회로도 및 입출력 할당 테이블>

- 프로젝트에 쓰인 기판의 회로도에서 입출력은 다음과 같이 고정적으로 결선되어 있습니다.

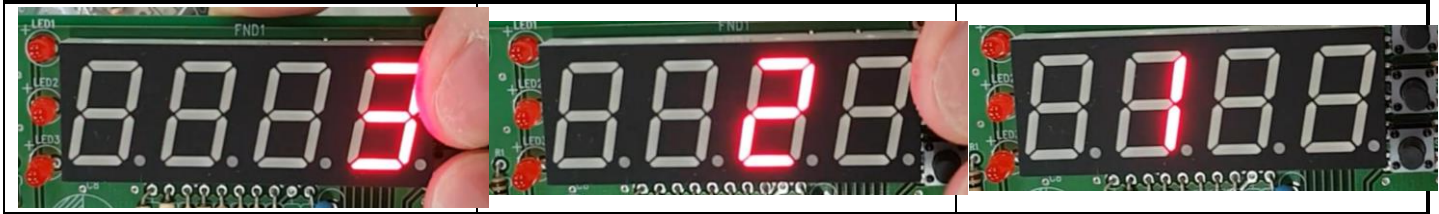


PORTA	<table><tr><td></td><td>BUZZER</td><td>dg1</td><td>dg2</td><td>g</td><td colspan="2">DP(소수점)</td></tr></table>								BUZZER	dg1	dg2	g	DP(소수점)		
	BUZZER	dg1	dg2	g	DP(소수점)										
PORTB	<table><tr><td></td><td></td><td>sw3</td><td>sw2</td><td>sw1</td><td>dg3</td><td>dg4</td><td></td></tr></table>									sw3	sw2	sw1	dg3	dg4	
		sw3	sw2	sw1	dg3	dg4									
PORTC	<table><tr><td>a</td><td>b</td><td>c</td><td></td><td></td><td>f</td><td>d</td><td>e</td></tr></table>							a	b	c			f	d	e
a	b	c			f	d	e								

<구현기능 상세설명>

- 전원을 연결했을 때(다시 시작할 때마다)
  - 약간의 간격이 주어지고 게임이 시작됩니다.

( 3 -> 2 -> 1 ) 카운트다운을 한 뒤, 게임이 시작됩니다.


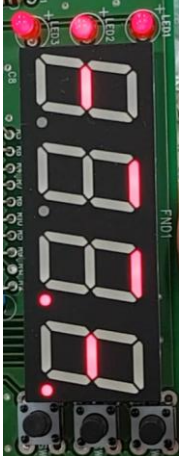
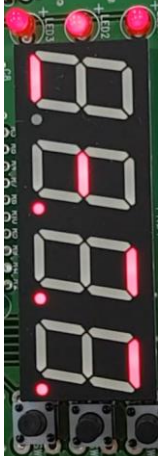
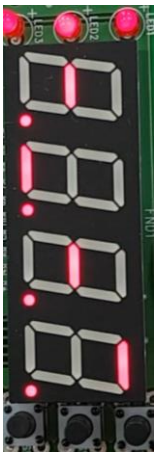


2. 게임실행

- 첫 번째 segment에서 마지막 segment까지 노트가 하나씩 떨어집니다.
- 떨어지는 노트는 일정한 규칙으로 떨어지는 것이 아니라 랜덤으로 떨어집니다.
- 마지막 segment에 도착한 노트를 해당 위치의 스위치를 눌러 없앨 수 있습니다.
- 떨어지는 노트를 누르지 못했을 경우 실패로 간주되어 버저가 짧게 울립니다.

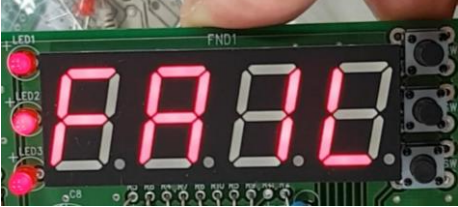
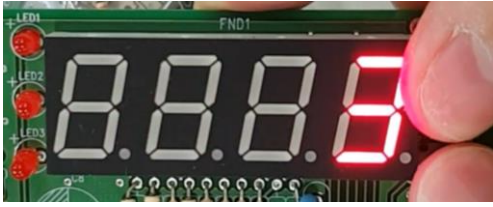
3. 노트를 누르지 못했을 경우

- 실패할 때마다 segment의 소수점에 불이 들어옵니다.
- 4개의 소수점에 불이 들어온 상태에서 실패했을 경우 게임이 종료됩니다.

1회 실패	2회 실패	3회 실패	4회 실패
			

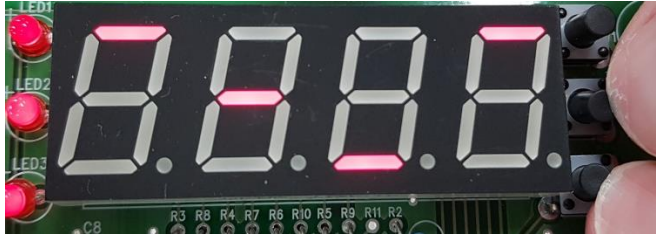
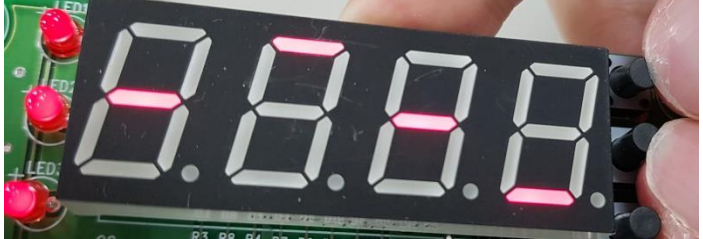
#### 4. 게임 종료 & RESTART

- 게임이 종료가 되면, segment에 FAIL이라 표시되고 버저가 일정 간격으로 울리게 됩니다.
- 모든 스위치를 한 번에 눌러 게임을 다시 시작할 수 있습니다.

5회 실패(FAIL)	RESTART
	

#### 5. 노트 속도 조절

- 보통 속도와 빠른 속도로 게임을 실행할 수 있습니다.
- 기본 속도는 보통이고 2번째, 3번째 스위치를 같이 눌렀을 때, 빠른 속도가 됩니다.
- 반대로 빠른 속도일 때, 1번째, 2번째 스위치를 같이 누르면 보통 속도가 됩니다.

보통 -> 빠름	빠름 -> 보통
	

#### 6. 일시정지

노트가 내려오는 것을 멈출 수 있습니다.

1번째, 3번째 스위치를 같이 누르면 일시정지가 됩니다.

일시정지


#### <조원 별 분담 내용>

- 구병조 - 납땜, 보조 코드 작성, 테스트 및 피드백, 사진 동영상 촬영 및 보고서 작성
- 이창훈 - 납땜, 주요 코드 작성, 테스트 및 피드백, 사진 동영상 촬영 및 보고서 작성

#### <결론 및 고찰>

평소 게임에 관심이 많았던 저희 조는 자유주제의 팀 프로젝트 과제를 두고 장난삼아 '게임 하나 만들어 볼까?'라는 장난처럼 나온 말로 팀 프로젝트를 시작하게 되었습니다. 쓸 수 있는 부품들이 스위치, 7-segment, 버저, LED로 제한이 많았기 때문에, 어려운 게임보다는 누구나 쉽게 가능한 게임이 좋겠다고 생각했고 피아노 타일이라는 모바일 게임을 떠올리게 되었습니다. 원래 음악에 맞춰 떨어지는 노트를 클릭해서 없애고 음악이 끝날 때까지 틀리지 않아야 CLEAR 되는 게임이지만 저희는 음악을 넣는 것은 불가능하다고 생각하여 단순히 그 메커니즘만 구현해 보기로 했습니다.

주제를 정하고 제일 먼저 한 일은 받은 데모 보드와 부품들을 납땜하는 작업이었습니다. 빨리 끝낼 수 있을 거라 생각했지만 생각 외로 시간이 많이 소요되었습니다. 특히 PIC 같은 경우는 핀 사이의 간격이 엄청 좁아 더 신중히 납땜해야 했고, 핀 수도 많아 오래 걸렸던 작업이었습니다. 납땜을 다 하고 실험 코드를 받아 확인했을 때 정상 동작을 하는 것을 보고 다행이라 생각했지만, 코드를 짜는 과정에 갑자기 7-segment에서 불이 비정상적으로 켜지는 현상이 발생해 남은 데모 보드와 부품들을 다시 납땜해야 했습니다. 하지만 이번에 납땜한 것은 7-segment의 d번째 자리가 꺼지는 경우가 발생하여 결국 앞서 선배님들이 했던 보드를 받아 과제를 수행했습니다. 예상외로 많은 시간을 납땜에 소요해서 힘들었고 정상적으로 동작하지 않아 많이 아쉽게 느끼고 있습니다.

저의 조의 팀 프로젝트 목표는 매주 한, 두 가지 기능을 생각해서 코드를 짜고 구현해 시험 전주에 다 끝내는 것이었습니다. 첫 주, 저희 조는 위에서 노트가 떨어지고 눌렀을 때 노트가 사라지는 것과 실패했을 때 소수점에 불이 들어오는 것을 구현했습니다. 배운 내용들을 응용하다 보니 어렵지 않게 구현해 낼 수 있었습니다.

처음에 저희가 짠 코드는 떨어지는 노트는 저희가 정한 규칙으로 떨어지게 했는데, 이렇게 구현하면 코드 길이가 엄청나게 길어져 비효율적이고 사용자가 5번 실패를 했을 때만 게임이 종료되도록 하는 것을 바라고 있었기 때문에 2번째 주는 랜덤으로 떨어지는 노트를 구현하는 것으로 했습니다. 난수를 발생시키는 명령어가 없기 때문에 주기적으로 발생하는 인터럽트를 이용해 구현해 봤습니다. 인터럽트가 발생할 때마다 1씩 증가하는 TIME의 값에 0번째 비트를 확인해서 특정 루틴으로 빠질 수 있게 코드를 짜니 랜덤으로 노트가 떨어질 수 있게 구현할 수 있었습니다. 팀 프로젝트를 하면서 가장 뿌듯하면서 성취감을 느낄 수 있었습니다.

마지막 주는 재시작 기능과 속도 조절 기능을 추가했습니다. 2~3개 버튼의 누를 때만 기능이 실행되도록 설계해서 게임 중 갑자기 속도가 증가하거나 재시작되는 경우를 막아보려고 했습니다. 그리고 전원이 들어왔을 때 게임이 바로 시작되어 실수하는 경우가 많았기 때문에 3초 카운트다운을 넣어 그 점을 보완했습니다.

이렇게 저희 조는 차근차근 기능을 추가해 가며 프로젝트를 진행했습니다. 아쉬운 점도 스

스로 칭찬해 줄 부분도 있었습니다. 단순히 시계를 만드는 것이 아니라 자율주제로 우리가 원하는 것을 스스로 고민해보고 문제를 파악하고 해결해 보는 과정에서 얻을 수 있었던 것이라 생각합니다. 끝으로, 완벽하진 않지만 한정된 부품으로 우리가 생각했던 기능을 대부분 구현했다는 점을 가장 자랑스럽게 생각하고 있습니다.

## <소스 코드 전문>

PROCESSOR 16F876A

INCLUDE <P16F876A.INC>

STATUS\_TEMP EQU 20H

W\_TEMPEQU 21H

INC EQU 22H

INT\_CNTEQU 23H

NOTEA EQU 24H ;7-SEGMENT 상단(a)에 내려올 노트. 1번 bit에서 4번 bit까지 7-SEG에 표시

NOTEB EQU 25H ; // 중단(g)

NOTEC EQU 26H ; // 하단(d)

NOTEAS EQU 35H ;NOTEA를 8비트 이상으로 연장

NOTEBS EQU 36H ;NOTEB를 //

NOTECS EQU 37H ;NOTEC를 //

DN EQU 27H ;눌린 버튼을 확인

ER EQU 28H ;실패 횟수 기록

BT EQU 29H ;LED 조절, 본 게임에서는 게임이 시작되면 항상 ON

TIME EQU 31H ;난수 생성을 위한 시간에 따른 변수

DBUF1 EQU 33H

DBUF2 EQU 34H

SPEED EQU 38H ;속도 조절을 위한 변수

ORG 00H

REGAME

GOTO START\_UP ;재시작 지점

ORG 04H

;ISR 시작 번지

MOVWF W\_TEMP ;현재 사용되고 있는 W REG 저장

SWAPF STATUS, W

MOVWF STATUS\_TEMP

CALL DISP ;DISPLAY 부 프로그램

SWAPF STATUS\_TEMP, W

MOVWF STATUS

SWAPF W\_TEMP, F

SWAPF W\_TEMP, W

BCF INTCON, 2

RETFIE

;DISPLAY ROUTINE

DISP

INCF INC, F

BTSS INC, 2 ;8번으로 나뉘는 분기점

GOTO DISPA

GOTO DISPB

DISPA

```
BTFSS INC, 1
GOTO DISPAA
GOTO DISPAB
```

DISPB

```
BTFSS INC, 1
GOTO DISPBA
GOTO DISPBB
```

DISPAA

```
BTFSS INC, 0
GOTO DISP1
GOTO DISPLED
```

DISPAB

```
BTFSS INC, 0
GOTO DISP2
GOTO DISPLED
```

DISPBA

```
BTFSS INC, 0
GOTO DISP3
GOTO DISPLED
```

DISPBB

```
BTFSS INC, 0
GOTO DISP4
GOTO DISPLED
```

ST

;노트 표시 초기화

```
BCF PORTC, 7
BCF PORTA, 1
BCF PORTC, 1
BCF PORTA, 0
RETURN
```

DISPLED

;LED. 본게임에서는 7-SEGMENT와 별개로 항상 ON

```
BSF PORTB, 1
BSF PORTA, 3
BSF PORTA, 2
BSF PORTB, 2
BSF PORTB, 7
BTFSC BT, 0
CALL LED1
BTFSC BT, 1
CALL LED2
BTFSC BT, 2
CALL LED3
BCF PORTC, 5
BCF PORTC, 6
```



```
BCF    PORTC, 7
RETURN
```

DISP1

```
BCF    PORTB, 7      ;LED 끄기
CALL   ST             ;노트 초기화
BTFSC  NOTEA, 4       ;상단에 표시해야 할 노트 있는가?
BSF    PORTC, 7
BTFSC  NOTEB, 4       ;중단      //
BSF    PORTA, 1
BTFSC  NOTEC, 4       ;하단      //
BSF    PORTC, 1
BTFSC  ER, 3          ;현재 자리에 에러 표시해야 할 만큼 실패 횟수 쌓였는가?
BSF    PORTA, 0
BTFSC  ER, 4          ;실패 5번 했는가
CALL   FF
BCF    PORTA, 3       ;digit1번 표시
RETURN
```

DISP2

;위의 작업 반복

```
BCF    PORTB, 7
CALL   ST
BTFSC  NOTEA, 3
BSF    PORTC, 7
BTFSC  NOTEB, 3
BSF    PORTA, 1
BTFSC  NOTEC, 3
BSF    PORTC, 1
BTFSC  ER, 2
BSF    PORTA, 0
BTFSC  ER, 4
CALL   A
BCF    PORTA, 2
RETURN
```

DISP3

```
BCF    PORTB, 7
CALL   ST
BTFSC  NOTEA, 2
BSF    PORTC, 7
BTFSC  NOTEB, 2
BSF    PORTA, 1
BTFSC  NOTEC, 2
BSF    PORTC, 1
BTFSC  ER, 1
BSF    PORTA, 0
BTFSC  ER, 4
```

```
CALL    I
BCF     PORTB, 2
RETURN
```

#### DISP4

```
BCF     PORTB, 7
CALL    ST
BTFSC   NOTEA, 1
BSF     PORTC, 7
BTFSC   NOTEB, 1
BSF     PORTA, 1
BTFSC   NOTEC, 1
BSF     PORTC, 1
BTFSC   ER, 0
BSF     PORTA, 0
BTFSC   ER, 4
CALL    RR
BCF     PORTB, 1
BSF     PORTA, 4
CALL    BTN
INCF    INT_CNT
RETURN
```

#### START\_UP

```
BSF     STATUS, RP0    ;RAM BANK 1 선택
MOVLW   B'00000000'    ;PORT I/O 선택
MOVWF   TRISA
MOVLW   B'00111000'    ;PORT I/O 선택
MOVWF   TRISB
MOVLW   B'00000000'    ;PORT I/O 선택
MOVWF   TRISC
MOVLW   B'00000111'
MOVWF   ADCON1
MOVLW   B'00000000'    ;INTERRUPT 시간 설정 --- 2.048msec
MOVWF   OPTION_REG
BCF     STATUS, RP0    ;RAM BANK 0 선택
CALL    START
BSF     INTCON, 5       ;TIMER INTERRUPT ENABLE
BSF     INTCON, 7       ;GLOBAL INTERRUPT ENABLE
GOTO    MAIN_ST
```

#### MAIN\_ST

```
;
CLRF    INT_CNT
CLRF    ER
```

```

CLRF    TIME
CLRF    PORTA
CLRF    PORTC
CLRF    PORTB
CLRF    NOTEA
CLRF    NOTEB
CLRF    NOTEC
CLRF    DN
CLRF    BT
CLRF    SPEED
BSF      PORTA, 4
MOVLW B'10001001'
MOVWF NOTEAS
MOVLW B'01010100'
MOVWF NOTEBS
MOVLW B'00100010'
MOVWF NOTECS
CLRF    INT_CNT

M_LOOP                                     ;메인 루프. 속도 설정에 따른 분기
    BTFSS    SPEED, 0
    GOTO     SPEEDA
    GOTO     SPEEDB

SPEEDA                                     ;느린 속도. 인터럽트가 120번 일어날 때마다 노트 이동
    MOVLW    .120
    SUBWF    INT_CNT, W
    BTFSS    STATUS, Z
    GOTO     XLOOP
    GOTO     CK_LOOP

SPEEDB                                     ;빠른 속도. 인터럽트가 60번 일어날 때마다 XLOOP로 이동
    MOVLW    .60
    SUBWF    INT_CNT, W
    BTFSS    STATUS, Z
    GOTO     XLOOP
    GOTO     CK_LOOP

CK_LOOP
    CLRF     INT_CNT
    ADDLW    0
    RRF      NOTEAS, F                     ;8비트 이상을 하나의 주소처럼 사용하기 위해 동시에 변경
    RRF      NOTEA, F
    ADDLW    0                             ;혹시 넘어갈지 모르는 Carry 제거
    RRF      NOTEBS, F
    RRF      NOTEB, F
    ADDLW    0
    RRF      NOTECS, F

```

```

RRF    NOTEC, F
ADDLW  0

XLOOP
INCF    TIME, F      ;난수를 위함
BTFSC   NOTEA, 0     ;상단의 노트가 표시부를 지나왔는가(놓쳤는가)
CALL    ERR          ;실패 횟수 추가
BTFSC   NOTEB, 0     ;중단의          / /
CALL    ERR
BTFSC   NOTEC, 0;    ;하단의          / /
CALL    ERR
GOTO    M_LOOP

```

```

ERR
CALL    RANDOMERR     ;하나의 노트가 사라졌으므로 새로운 노트 생성
BCF     NOTEA, 0
BCF     NOTEB, 0
BCF     NOTEC, 0
BTFSC   ER, 3         ;이미 쌓인 실패 횟수 확인
BSF     ER, 4         ;쌓인 실패 횟수에 따라 추가 기록
BTFSC   ER, 4
BCF     PORTA, 4
BTFSC   ER, 2
BSF     ER, 3
BTFSC   ER, 1
BSF     ER, 2
BTFSC   ER, 0
BSF     ER, 1
BSF     ER, 0
RETURN

```

```

BTN
MOVF    PORTB, W      ;3개의 버튼 입력을 3~5번 비트에 저장
MOVWF   DN
ADDLW   0
RRF     DN, F
RRF     DN, F
RRF     DN, W
ANDLW   B'00000111'
CALL    LU            ;Look Up Table
RETURN

```

```

LU
ADDWF   PCL, F        ;눌린 버튼 조합에 따라 실행 분기
GOTO    S0
GOTO    S1
GOTO    S2

```

	GOTO S3	
	GOTO S4	
	GOTO S5	
	GOTO S6	
	GOTO S7	
S0		;3개의 버튼 동시에 눌림. 재시작
	CALL REGAME	
	RETURN	
S1		;왼쪽 2개 버튼 동시에 눌림. 속도 감소
	CALL DOWN	
	RETURN	
S2		;양쪽 2개 버튼 동시에 눌림. 일시정지
	CLRF INT_CNT	
	RETURN	
S3		;왼쪽 끝 버튼 눌림
	BTFSC NOTEC, 1	
	CALL RANDOM3	
	BCF NOTEC, 1	
	RETURN	
S4		;오른쪽 2개 버튼 동시에 눌림. 속도 증가
	CALL UP	
	RETURN	
S5		;가운데 버튼 눌림
	BTFSC NOTEB, 1	
	CALL RANDOM2	
	BCF NOTEB, 1	
	RETURN	
S6		;오른쪽 버튼 눌림
	BTFSC NOTEA, 1	
	CALL RANDOM1	
	BCF NOTEA, 1	
	RETURN	
S7		;아무것도 눌리지 않음
	RETURN	
FF		;실패했을 경우 노트 대신 FAIL이라는 글자 7-SEGMENT에 표시
	MOVLW B'10000101'	
	MOVWF PORTC	
	BCF PORTA, 0	
	BSF PORTA, 1	
	RETURN	
A		
	MOVLW B'11100101'	
	MOVWF PORTC	
	BCF PORTA, 0	

	BSF	PORTA, 1	
	RETURN		
I			
	MOVLW	B'01100000'	
	MOVWF	PORTC	
	BCF	PORTA, 0	
	BCF	PORTA, 1	
	RETURN		
RR			
	MOVLW	B'00000111'	
	MOVWF	PORTC	
	BCF	PORTA, 0	
	BCF	PORTA, 1	
	RETURN		
LED1			;조건에 따라 LED 표시. 본 게임에서는 시작되면 항상 ON
	BCF	PORTC, 7	
	BSF	PORTC, 6	
	BSF	PORTC, 5	
	RETURN		
LED2			
	BSF	PORTC, 7	
	BCF	PORTC, 6	
	BSF	PORTC, 5	
	RETURN		
LED3			
	BSF	PORTC, 7	
	BSF	PORTC, 6	
	BCF	PORTC, 5	
	RETURN		
RANDOM1			;난수를 통한 새로운 노트 생성
	BCF	PORTA, 4	
	BCF	NOTEA, 6	;3종류의 난수를 생성하는 것은 $2^n$ 종류 난수를 생성하는 것보다
	BCF	NOTEB, 6	;어렵고, 부정확하기 때문에
	BCF	NOTEA, 6	;A 버튼을 누르면 B C 둘 중 하나를 고를 수 있도록 2종류 난수로 대체
	BTFSS	TIME, 0	;B 버튼을 누르면 C A, C 버튼을 누르면 A B 중 하나가 나온다.
	GOTO	SB	
	GOTO	SC	
RANDOM2			
	BCF	PORTA, 4	
	BCF	NOTEA, 6	
	BCF	NOTEB, 6	
	BCF	NOTEA, 6	
	BTFSS	TIME,0	

	GOTO	SC	
	GOTO	SA	
RANDOM3			
	BCF	PORTA, 4	
	BCF	NOTEA, 6	
	BCF	NOTEB, 6	
	BCF	NOTEC, 6	
	BTFSS	TIME, 0	
	GOTO	SA	
	GOTO	SB	
RANDOMERR			;노트를 놓쳤을 때에도 맞췄을 때와 같이 새로운 노트를 생성해야한다.
	BCF	NOTEA, 6	
	BCF	NOTEB, 6	
	BCF	NOTEC, 6	
	BTFSS	TIME, 0	
	GOTO	SB	
	GOTO	SC	
SA			;새로운 노트
	BSF	NOTEA, 6	
	BSF	NOTEB, 7	
	RETURN		
SB			
	BSF	NOTEB, 6	
	BSF	NOTEC, 7	
	RETURN		
SC			
	BSF	NOTEC, 6	
	BSF	NOTEA, 7	
	RETURN		
DELAY			;딜레이
	MOVLW	.255	
	MOVWF	DBUF1	
LP1			
	MOVLW	.255	
	MOVWF	DBUF2	
LP2			
	DECFSZ	DBUF2, F	
	GOTO	LP2	
	DECFSZ	DBUF1, F	
	GOTO	LP1	
	RETURN		
UP			;속도 증가
	BTFSS	SPEED, 0	

```
INCF    SPEED, F
CLRF    INT_CNT
RETURN
```

DOWN

;속도 감소

```
BTFSC   SPEED, 0
DECF    SPEED, F
CLRF    INT_CNT
RETURN
```

START

;게임을 시작하면 가장 먼저 3 2 1을 표시한 후 시작한다.

```
BSF     PORTB, 1
BSF     PORTA, 3
BSF     PORTA, 2
BSF     PORTB, 2
MOVLW   B'11100010'
MOVWF   PORTC
BCF     PORTA, 0
BSF     PORTA, 1
BCF     PORTB, 1
CALL    DELAY
BSF     PORTB, 1
MOVLW   B'11000011'
MOVWF   PORTC
BCF     PORTA, 0
BSF     PORTA, 1
BCF     PORTB, 2
CALL    DELAY
BSF     PORTB, 2
MOVLW   B'01100000'
MOVWF   PORTC
BCF     PORTA, 0
BCF     PORTA, 1
BCF     PORTA, 2
CALL    DELAY
RETURN
END
```