

Trường Đại học Bách khoa thành phố Hồ Chí Minh
Khoa Điện-Điện tử
Ngành Điều khiển tự động



BÀI TẬP VI XỬ LÝ:
THIẾT KẾ MẠCH PHÁT ÂM THANH NHƯ PIANO
GV Giảng dạy: HỒ TRUNG MỸ

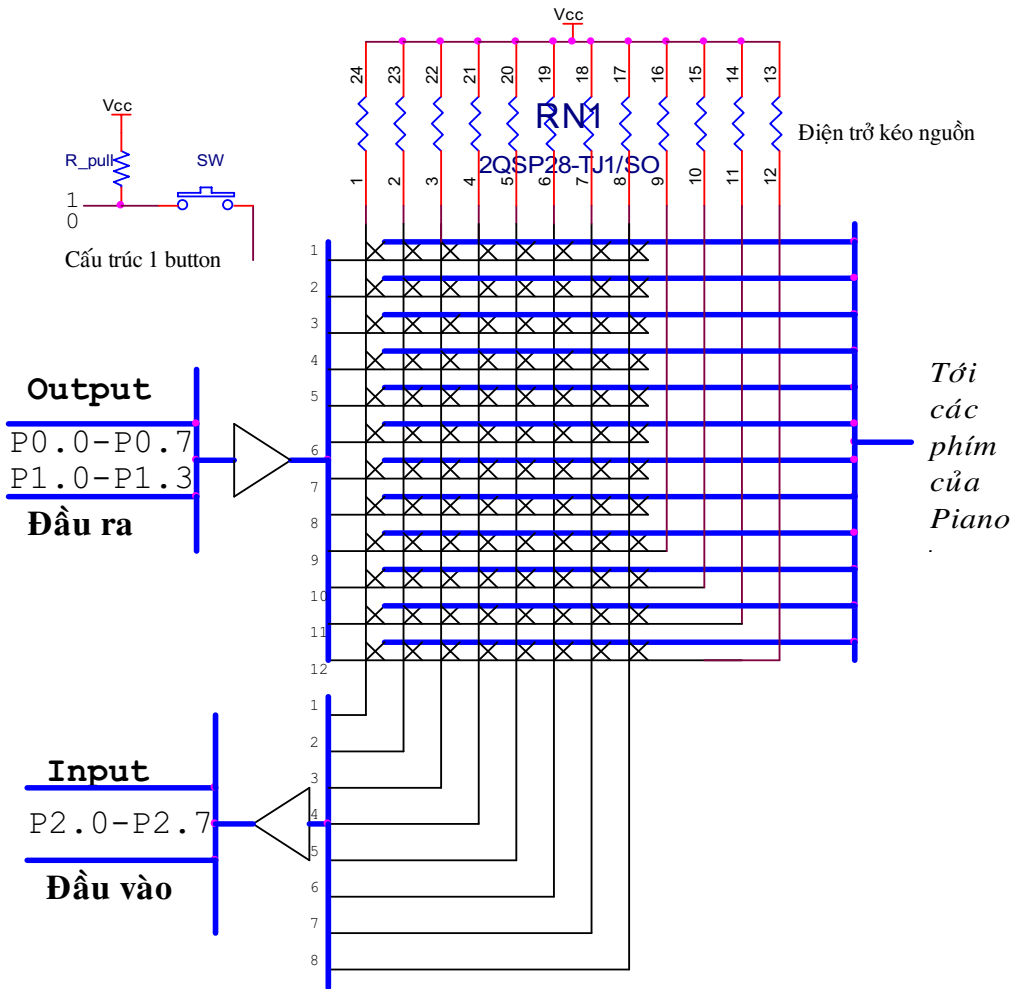
LỚP DD02KSTN

Nhóm 5:

✚ BÙI TRUNG HIẾU
✚ NGUYỄN ĐỨC HOÀNG
✚ NGUYỄN VĂN HUY

HỌC KÌ 2 NĂM HỌC 2004-2005
TP Hồ Chí Minh, ngày 9 tháng 6 năm 2005
Ж

Piano là một công cụ phát ra âm thanh ở tần số thấp (27.5Hz đến 4186Hz) < tần số của các nốt nhạc (note) ở hình bên¹, >.(tai người có thể cảm nhận được các tần số từ 20-20kHz) Dùng vi điều khiển 8051 (MCU851), ta có thể tạo được các tần số mong muốn khi nhấn các switch tương ứng. Giả sử nối ngoại vi (52 notes chuẩn và 36 notes thăng (#)) ta dùng 88 nodes với các kí hiệu tương ứng, nối vào các cổng từ P0 đến P2 theo như hình dưới đây:



A0	27.5	A0#	29.135
B0	30.868		
C1	32.703	C1#	34.648
D1	36.708	D1#	38.891
E1	41.203		
F1	43.654	F1#	46.249
G1	48.999	G1#	51.913
A1	55.000	A1#	58.270
B1	61.735		
C2	65.406	C2#	69.296
D2	73.416	D2#	77.782
E2	82.407		
F2	87.307	F2#	92.499
G2	97.999	G2#	103.83
A2	110.00	A2#	116.54
B2	123.47		
C3	130.81	C3#	138.59
D3	146.83	D3#	155.56
E3	164.81		
F3	174.61	F3#	185.00
G3	196.00	G3#	207.65
A3	220.00	A3#	233.08
B3	246.94		
C4	261.63	C4#	277.18
D4	293.66	D4#	311.13
E4	329.63		
F4	349.23	F4#	369.99
G4	392.00	G4#	415.30
A4	440.00	A4#	466.16
B4	493.88		
C5	523.25	C5#	554.37
D5	587.33	D5#	622.25
E5	659.25		
F5	698.46	F5#	739.99
G5	783.99	G5#	830.61
A5	880.00	A5#	932.33
B5	987.77		
C6	1046.5	C6#	1108.7
D6	1174.7	D6#	1244.5
E6	1318.5		
F6	1396.9	F6#	1480.0
G6	1568.0	G6#	1661.2
A6	1760.0	A6#	1864.7
B6	1979.5		
C7	2093.0	C7#	2217.5
D7	2349.3	D7#	2489.0
E7	2637.0		
F7	2793.8	F7#	2960.0
G7	3136.0	G7#	3322.4
A7	3520.0	A7#	3729.3
B7	3951.1		
C8	4186.0		

Kết nối ngoại vi của 8051 với các phím Piano

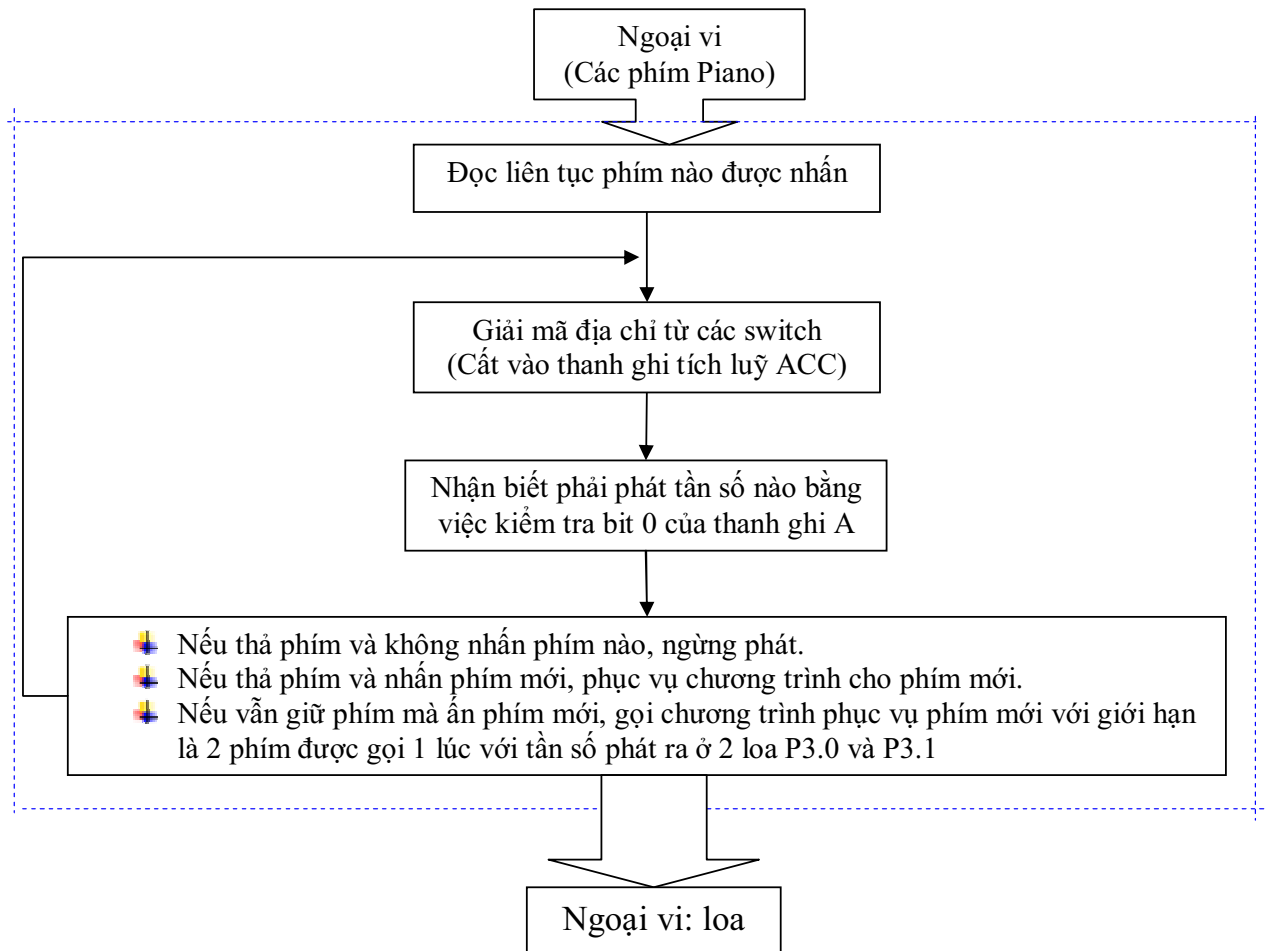
Với cách kết nối như trên, ta chỉ dùng 88 phím từ điểm có tọa độ 1.1 đến 12.7. Quy định kết nối theo thứ tự bảng 1.

A0 - A7	A0# - A7#	B0 - B7	C1 - C8	C1# - C7#	D1 - D7	D1# - D7#	E1 - E7
1.1-1.8	2.1-2.8	3.1-3.8	4.1-4.8	5.1-5.7	6.1-6.7	7.1-7.7	8.1-8.7
F1 - F7	F1# - F7#	G1 - G7	G1# - G7#	P0.0-P0.7	P1.0-P1.1	P2.0-P2.7	P1.4-P1.5
9.1-9.7	10.1-10.7	11.1-11.7	12.1-12.7	1-8	9-10	1-8	9-10
				Đầu ra 8051		Đầu vào 8051	

QUY ĐỊNH KẾT NỐI PHÍM PIANO-MCU8051 VỚI CHUẨN TỌA ĐỘ

¹Được lấy từ trang: <http://www.vibrationdata.com/piano.htm>

Công việc của 8051 có thể phát thảo lại như sau: (Phần có đóng khung xanh)



Với cấu trúc chương trình như trên, ta viết hàm cho MCU8051:

1. Hàm nhận phím từ bàn phím Piano
2. Giải mã địa chỉ, tạo ra 88 giá trị ở thanh ghi tích lũy với các giá trị tương ứng của ma trận phím.
3. Tạo bảng tần số cho các giá trị ACC tương ứng.
4. Hàm tạo tần số cho các chân ở cổng P3. Với 88 notes ta có 88 hàm tạo tần số!

Ghi lại bảng tần số ở dạng sau:

A0	A1	A2	A3	A4	A5	A6	A7
27.5	55	110	220	440	880	1760	3520
A0#	A1#	A2#	A3#	A4#	A5#	A6#	A7#
29.135	58.27	116.54	233.08	466.16	932.33	1864.7	3729.3
B0	B1	B2	B3	B4	B5	B6	B7
30.868	61.735	123.47	246.94	493.88	987.77	1979.5	3951.1
C1	C2	C3	C4	C5	C6	C7	C8
32.703	65.406	130.81	261.63	523.25	1046.5	2093	4186
C1#	C2#	C3#	C4#	C5#	C6#	C7#	
34.648	69.296	138.59	277.18	554.37	1108.7	2217.5	
D1	D2	D3	D4	D5	D6	D7	
36.708	73.416	146.83	293.66	587.33	1174.7	2349.3	
D1#	D2#	D3#	D4#	D5#	D6#	D7#	
38.891	77.782	155.56	311.13	622.25	1244.5	2489	

E1	E2	E3	E4	E5	E6	E7
41.203	82.407	164.81	329.63	659.25	1318.5	2634.0
F1	F2	F3	F4	F5	F6	F7
43.654	87.307	174.61	349.23	698.46	1396.91	2793.8
F1#	F2#	F3#	F4#	F5#	F6#	F7#
46.249	92.499	185	369.99	739.99	1480	2960
G1	G2	G3	G4	G5	G6	G7
48.999	97.999	196.00	392.00	783.99	1568	3136
G1#	G2#	G3#	G4#	G5#	G6#	G7#
51.913	103.83	207.65	415.3	830.61	1661.2	3322.4

Bảng2 : TẦN SỐ CỦA CÁC NOTE TƯƠNG ỨNG

Ta thấy rằng chỉ có 12 tần số cơ bản cần tạo ra, các tần số khác được suy ra bằng cách tăng một số nguyên lần (2^n) tần số cơ bản.

Dùng 8051 có tần số thạch anh là 11.0592MHz, ta có thể tạo các tần số sau:

Ta tính số chu kì máy theo công thức: $MC = \frac{T_{note}}{T_{osc}} = \frac{11.0592MHz}{12} \times \frac{1}{f_{note}}$. Cũng từ công thức trên, ta

có bảng giá trị các tần số:

Phím	A0	A0#	B0	C1	C1#	D1	D1#	E1	F1	F1#	G1	G1#
Tần số	27.5	29.135	30.868	32.703	34.648	36.708	38.891	41.203	43.654	46.249	48.999	51.913
MC	33513	31632	29856	28181	26599	25106	23697	22367	21111	19927	18809	17753
MC_on	16756	15816	14928	14090	13299	12553	11848	11183	10555	9963	9404	8876
Phím	A7	A7#	B7	C7	C7#	D7	D7#	E7	F7	F7#	G7	G7#
Tần số	3520	3729	3951.1	4186	2217.5	2349.3	2489	2634	2793.8	2960	3136	3322.4
MC	261	247	233	220	414	391	369	349	329	311	293	277
MC_on	131	123	116	110	207	196	185	174	164	155	147	138

Bảng 3: CÁC GIÁ TRỊ CỦA TỔNG SỐ CHU KÌ MÁY CÁC TẦN SỐ CƠ BẢN

Ta thử phát thảo chương trình 8051:

Đọc lần lượt giá trị của các Notes A, A#, B,C,C#,D,D#,E,E#,F,F#,G,G# vào thanh ghi tích lũy, nếu phát hiện bit 0 ở vị trí nào thì phục vụ chương trình tạo sóng cho vị trí đó với điều kiện TR=0, tất nhiên sẽ có 2 Timer cùng chạy, nên sẽ có 2 tần số cùng được tạo ở các Port thích hợp.

Thứ tự ưu tiên phục vụ tạo âm thanh từ trên xuống theo trật tự trên

Chương trình chính sẽ là:

ORG 0000H ;Điểm vào từ chương trình Reset

LJMP MAIN

ORG 000BH

SJMP T0ISR

ORG 001BH

SJMP T1ISR

<Chỗ này sẽ đặt các chương trình phục vụ ngắt Timer, người viết có tình đặt dưới cho dễ theo dõi>

ORG 50H

MAIN:

;Khởi tạo các TIMER hoạt động ở MODE2, auto reload

MOV TMOD,#22H

;Khởi tạo cho phép ngắt Timer

MOV IE,#10001010B

;Chương trình đọc bàn phím và thực thi lệnh

NOTEA:

```

SETB      P1.3
CLR        P0.0
MOV        P2,#0FFH
MOV        A,P2
CJNE       A,#0FFH,NOTE_AA
SJMP       NOTEAT
NOTE_AA:   LJMP  NOTE_A           ;Do quá địa chỉ tương đối.
NOTEAT:
MOV        P2,#0FFH
SETB       P0.0
CLR        P0.1
MOV        A,P2
CJNE       A,#0FFH,NOTE_ATT
SJMP       NOTEB
NOTE_AAT:  LJMP  NOTE_AT         ;Do quá địa chỉ tương đối.
NOTEB:
MOV        P2,#0FFH
SETB       P0.1
CLR        P0.2
MOV        A,P2
CJNE       A,#0FFH,NOTE_BB
SJMP       NOTEC
NOTE_BB:   LJMP  NOTE_B         ;Do quá địa chỉ tương đối.
NOTEC:
MOV        P2,#0FFH
SETB       P0.2
CLR        P0.3
MOV        A,P2
CJNE       A,#0FFH,NOTE_CC
SJMP       NOTECT
NOTE_CC:   LJMP  NOTE_C         ;Do quá địa chỉ tương đối.
NOTECT:
MOV        P2,#0FFH
SETB       P0.3
CLR        P0.4
MOV        A,P2
CJNE       A,#0FFH,NOTE_CTT
SJMP       NOTED
NOTE_CTT:  LJMP  NOTE_CT        ;Do quá địa chỉ tương đối.
NOTED:
MOV        P2,#0FFH
SETB       P0.4
CLR        P0.5
MOV        A,P2
CJNE       A,#0FFH,NOTE_DD
SJMP       NOTEDT
NOTE_DD:   LJMP  NOTE_D         ;Do quá địa chỉ tương đối.
NOTEDT:
MOV        P2,#0FFH
SETB       P0.5

```

```

CLR      P0.6
MOV      A,P2
CJNE     A,#0FFH,NOTE_DTT
SJMP     NOTEE
NOTE_DTT: LJMP  NOTE_DT      ;Do quá địa chỉ tương đối.
NOTEE:
MOV      P2,#0FFH
SETB     P0.6
CLR      P0.7
MOV      A,P2
CJNE     A,#0FFH,NOTE_EE
SJMP     NOTEF
NOTE_EE: LJMP  NOTE_E      ;Do quá địa chỉ tương đối.
NOTEF:
MOV      P2,#0FFH
SETB     P0.7
CLR      P1.0
MOV      A,P2
CJNE     A,#0FFH,NOTE_FF
SJMP     NOTEFT
NOTE_FF: LJMP  NOTE_F      ;Do quá địa chỉ tương đối.
NOTEFT:
MOV      P2,#0FFH
SETB     P1.0
CLR      P1.1
MOV      A,P2
CJNE     A,#0FFH,NOTE_FTT
SJMP     NOTEG
NOTE_FTT: LJMP  NOTE_FT    ;Do quá địa chỉ tương đối.
NOTEG:
MOV      P2,#0FFH
SETB     P1.1
CLR      P1.2
MOV      A,P2
CJNE     A,#0FFH,NOTE_GG
SJMP     NOTEGT
NOTE_GG: LJMP  NOTE_G      ;Do quá địa chỉ tương đối.
NOTEGT:
MOV      P2,#0FFH
SETB     P1.2
CLR      P1.3
MOV      A,P2
CJNE     A,#0FFH,NOTE_GTT
LJMP     NOTEA
NOTE_GTT: LJMP  NOTEG_T

```

Trong đoạn code vừa viết tên, người viết chủ đích chịu tốn bộ nhớ chương trình để tiết kiệm thời gian hồi vòng và phục vụ ngắt cho các phím nhấn.

Đoạn code sau đây phục vụ chương trình cho các note nhạc khi được chọn:

(*)

NOTE_A:

```
TESTA0:  JNB      ACC.0,NOTEA0
TESTA1:  JNB      ACC.1,NOTEA1
TESTA2:  JNB      ACC.2,NOTEA2
TESTA3:  JNB      ACC.3,NOTEA3
TESTA4:  JNB      ACC.4,NOTEA4
TESTA5:  JNB      ACC.5,NOTEA5
TESTA6:  JNB      ACC.6,NOTEA6
TESTA7:  JNB      ACC.7,NOTEA7
LJMP     NOTEB
```

;Các hàm con trong NOTE_A

NOTEA0:

```
MOV      R0,#1
CALL     TIMEA
SJMP     TESTA1
```

NOTEA1:

```
MOV      R0,#2
CALL     TIMEA
SJMP     TESTA2
```

NOTEA2:

```
MOV      R0,#4
CALL     TIMEA
SJMP     TESTA3
```

NOTEA3:

```
MOV      R0,#8
CALL     TIMEA
SJMP     TESTA4
```

NOTEA4:

```
MOV      R0,#16
CALL     TIMEA
SJMP     TESTA5
```

NOTEA5:

```
MOV      R0,#32
CALL     TIMEA
SJMP     TESTA6
```

NOTEA6:

```
MOV      R0,#64
CALL     TIMEA
SJMP     TESTA7
```

NOTEA7:

```
MOV      R0,#128
```

```
CALL    TIMEA
LJMP    NOTEAT
```

 Kết thúc chuỗi lệnh này, quay về kiểm tra phím A# có được nhấn hay không?

 Hàm TIMEA sẽ kiểm tra TIMER nào rảnh, sẽ thực hiện tạo sóng, nếu không, bỏ qua và quay về lại chương trình kiểm tra phím; dưới đây là chương trình:

```
-----
TIMEA:
JNB     TR0,FREET0A
JNB     TR1,FREET1A
RETURN:
RET
FREET0A:
MOV     TH0,#-131
MOV     TL0,#-131
MOV     R1,00H      ;R0
SETB    P1.4
SETB    TR0
SJMP    RETURNA
FREET1A:
MOV     TH1,#-131
MOV     TL1,#-131
MOV     R2,00H      ;R0
MOV     R3,00H      ;R0
MOV     R0,#0
SETB    P1.4
SETB    TR1
SJMP    RETURNA
```

-----(**)

Ta cho thanh ghi R0 và R1 thực hiện trong Timer0, R2 và R3 thực hiện trong Timer1 để dễ kiểm tra trong các chương trình ngắt.

 Sau đây là các chương trình ngắt: Ta không quan tâm đó là ngắt từ Note nào, ta chỉ quan tâm đến R0 và R1(ở Timer0) và (R2,R3 ở Timer1) có bằng 0 hay chưa để thực hiện các chương trình mong muốn.

```
-----
ORG     20H
T0ISR:
CJNE    R0,#1,NEXT_T0ISR      ;Khi R0 vẫn giữ mức khác 1, giảm R0 và RETI
CLR     P1.4
SETB    P1.5
DJNZ    R1,CONT_T0ISR
CLR     P1.5
SJMP    CONT_T0ISR
NEXT_T0ISR:
DJNZ    R0,CONT_T0ISR
CLR     TR0
CONT_T0ISR:
RETI
```



```

T1ISR:
CJNE      R2,#1,NEXT_T1ISR      ;Khi R2 vẫn giữ mức khác 1, giảm R2 và RETI
CLR       P1.6
SETB      P1.7
DJNZ      R2,CONT_T1ISR
CLR       P1.7
SJMP      CONT_T1ISR
NEXT_T1ISR:
DJNZ      R3,CONT_T0ISR
CLR       TR1
CONT_T1ISR:
RETI

```

Thủ vị là đoạn code trên vừa đúng 32 Bytes (Từ 20H đến 3FH)

Đoạn code này sẽ được đặt trước dòng 'ORG 50H' như đã nói từ đầu.

Việc còn lại, ta sẽ xử lý tương tự cho các Note khác, quá trình chỉ là copy lại đoạn code trên(từ (*) đến (**)) và chỉnh sửa chút ít về các giá trị cần nạp cho Timer.

Về cơ bản, chương trình đã hoàn thành, thử phân tích một số kết quả thu được.

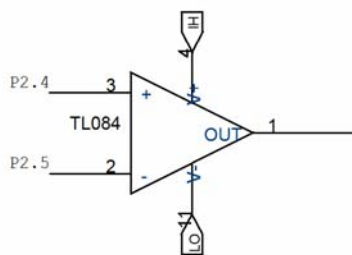
Khi viết hoàn chỉnh chương trình, ta thấy tổng cộng bộ nhớ chương trình cần sử dụng là từ vùng địa chỉ 0000H đến 06E7H

Mỗi phần kiểm tra 1 note nhạc mất 7MC

Nếu không có phím nhấn, mất $7 \times 12 = 84MC = 91\mu s$.

Trường hợp xấu nhất, phím G7# được nhấn kể từ reset vào chương trình chính, mất thêm 16MC nữa, tức tổng cộng sẽ mất : $84 + 16 = 100MC = 108.5\mu s$ khoảng gần 0.1ms, khoảng thời gian đó không đáng kể lắm so với tốc độ nhấn phím của người.

Kết quả sóng ra ở chân : P1.4 và P1.5-Timer0 (P1.6 và P1.7-Timer1) sẽ được nối vào OP-AMP để tạo xung vuông lưỡng cực, kết nối như sau:



Ta có thể phát triển chương trình để thoả mãn 12 phím được nhấn cùng lúc. Lúc ấy, ta dùng kỹ thuật chụp giá trị của thanh ghi Timer TH và TL, hoạch định những khoảng thời gian nhỏ. Tuy nhiên, bài này chỉ dừng lại ở chỗ dùng 2 Timer, không phải vì người viết không thể phát triển, mà đơn giản vì khi dùng các thuật toán đó có thể gây sai số lớn trong việc tạo tần số.

Bài toán cũng dừng lại ở việc tạo sóng vuông lưỡng cực và không phát triển thêm, việc phân tích các hài không mong muốn ở mỗi tần số không nằm trong phạm vi môn vi xử lý. Khi ước muốn ở mỗi tần số nhất định tạo ra khi phân tích qua miền tần số thực chỉ có một xung, điều đó chỉ có thể đúng với sóng sin.