

TUTORIAL 2. - The Music Synthesizer, by Chuck Thomka

The synthesizer circuit, which is contained wholly within the 40 pin custom I/O chip, is a very versatile circuit which contains counters and amplifiers to give the programmer tremendous control of the three voice output along with a tremolo, vibrato, and even a noise generator. The output frequency range is very accurately adjustable from less than 14 hertz to ultrasonic frequencies. The upper limit may be set by the capacity of your tv sound system.

Toward the end of this article I'll give you some examples of figuring out an accurate oscillating frequency, and two programs. One to play the entire synthesizer via the &(16) through &(23) registers, and the other to simulate the Bell Telephone Touch Tone sound. By holding a telephone mouthpiece at the tv speaker, you can dial a telephone number. Refer to the logical diagram included to fix in your mind just what is happening.

The operation goes like this. The clock triggers the Master Counter which counts, starting from zero, the pulses up to the number set into the &(16) register. When the Master Counter reaches that number, it puts out a pulse and then the counter will either reset to zero or preset to the number from the &(20) or &(23) register (these will be explained later) and then starts the count up all over again.

The pulse coming from the Master Counter goes to the inputs of the identical A, B, and C Counters. While all three counters are identical, the number that each counts up to is individually adjustable by using the &(17), &(18), and &(19) registers.

Each of these counters, upon reaching its count number, will "toggle" its output. Toggle means to change its output once; if it was low, it will go high and if it was high it will go low. Since the counter has reached its assigned count level, it will reset to zero at the next pulse from the Master Counter and start counting up again so that it can once again toggle.

Each of the outputs of the A, B, and C Counters goes to its own amplifier. That amplifier is the volume control of that counter (or voice) and it has 16 levels of output. While the lowest level isn't absolutely zero, it is almost inaudibly low. The &(22) register controls Volume A (the four least significant bits (=LSB)) and Volume B (the four most significant bits (=MSB)) while half of register &(21) (the four LSB) controls Volume C.

There is one more amplifier. This is the Noise Volume. But it must be 'enabled', to work. (that is, turned on) Only one bit is the enable, bit 32 of the &(21) register. This bit must be high for the Noise Generator to be heard at the volume setting of the &(23) register (the four MSB).

&(20) is the Vibrato control register. What it will do to the final audio output is make the sound shift from one frequency to another frequency at a set rate (1 of 4). This makes it very useful for sound effects. How it does this is by not allowing the Master Counter to reset to zero once the Master Counter has reached the &(16) value. Instead, at that time, on the following clock the Master Counter will preset to the value set into the 6 LSB's of the &(20) register. This means instead of the next count cycle of the Master Counter starting at zero, it will start at some number from 0 to 63. This could greatly reduce the time required for the Master Counter to reach its &(16) value. For example, if the &(16) is set to 14, which would normally take 15 clocks (0 to 14 is 15), and the 6 LSB's of the &(20) were set to 10, then this would triple the normal output frequency. This is because now the Master Counter would only have to count from 10 to 14, a total of five clocks, or three times faster than normal.

I mentioned a rate selection of 1 out of 4 choices that the Vibrato will 'vibrate' at. This is selected by the remaining 2 bits of the &(20) register, the 2 MSB's. The four combinations of these 2 bits are 00, 01, 10, and 11, in increasing value. These rates are (approximately) 18.5 milliseconds, 37 milliseconds, 74 milliseconds, and 148 milliseconds, respectively. What happens during this rate time is that for rate 00 the Master Counter will work normally and be allowed to count up to its determined value and then reset to zero for a period of 18.5 milliseconds; then, for the next 18.5 milliseconds the counter will not reset to zero but instead preset to the value of the 6 LSB's in the &(20) register. Of course, if that value is also zero (000000) then there is no difference between reset and preset, hence no vibrato. Each of the other rate selections work in a similar manner except that the duration of normal count time and preset count time will be longer, which is a slower vibrato effect.

My wording earlier was that this present value could greatly reduce the time required for the Master Counter to reach its $\&(16)$ value. It is also possible that the preset value could greatly increase the time, thus making the vibrato shift in audio frequency go lower instead of higher. This is possible when the vibrato value (range 0 to 63) is set to a higher value than the number in the $\&(16)$ register. For example, in my previous example I had $\&(16)$ set to 14 (or 15 counts) and the 6 LSB's of $\&(20)$ set to 10, showing that the resultant vibrato shift would be three times greater than normal. Now in this new example lets have $\&(16)$ still at 14 and put $\&(20)$ (the 6 LSB's) to 15. During the time of vibrato preset operation, when the Master Counter finally gets to 14, its determined value, at the next clock the counter will preset to 15. The counter has no way of knowing that it is now at a count greater than its determined value, it is still looking for the value 14, and 15 is not 14 - so as the counter receives more clock pulses it will continue to count up. And up to count 255 (binary 11111111) it continues. The next clock will overflow the counter to zero (no provisions for carryover are made) and still continue - 1, 2, 3, etc. And it finally reaches count 14 to output the Master Counter Pulse. The next clock repeats the reset to 15 and another long count, "round the horn", to 14. So how many clock pulses does this take? Well, counting 15 to 255 is 241, plus 0 to 14 is an additional 15 for a total of 256. Instead of the normal reset time of 15 clocks, the preset time will take 256 clocks in this example, or over 17 times longer. The longer time for the output pulse from The Master Counter results in a much slower rate of toggling of the A, B, and C counters, which results in a much lower output (voice) frequency.

A Tremolo would be a warble effect in the tone of a note. This is controlled by $\&(23)$, but it must first be enabled by $\&(21)$ bit 16. This bit acts like a switch. When it is off, $\&(20)$ -the Vibrato Register- can be engaged. If $\&(21)$ bit 16 is on, the Vibrato will not work and the Tremolo register is engaged. The Tremolo and Vibrato cannot be engaged at the same time, but as you become more familiar with the sounds out of the Sound Synthesizer, and their ranges, you'll see it is not really necessary anyway.

The operation of the Tremolo is similar to the Vibrato in that it also presets the Master Counter to some number before it starts its count. But, whereas the Vibrato had a rate control (the period of which the counter would alternately reset or preset), the Tremolo always presets. The number to which it is preset is randomly selected and only limited to a maximum number as set into $\&(23)$. All eight bits of $\&(23)$ are used for the Tremolo preset. But remember that the four highest bits are also used for the Noise Volume, in case you try to use the Noise along with the Tremolo. Also remember that the Noise is switched in by $\&(21)$ bit 32.

The preset of a number into the Master Counter, if it is less than the $\&(16)$ determined value of the Master Counter, will shorten the time involved in reaching that $\&(16)$ value. The result will be an upward shift of the resultant A, B, and/or C frequency or pitch, and the next preset will be another randomly selected value (up to the maximum of $\&(23)$). The audible result of all this will be a tremble or quivering effect. If the $\&(23)$ value is greater than the Master Oscillator's $\&(16)$, the resultant may be a random long count 'round the horn' for the same reason that the Vibrato could have a long count. The audible effect will be a wildly varying Tremolo sound. One last remark about the Tremolo is that if $\&(23)$ is set to zero, there will be no Tremolo effect at all since a random preset of up to zero can only be zero, and that is the same as a reset of the Master Counter.

Now lets get into how to figure out a simple set of frequencies. First, the key to the whole thing is to know what the main clock frequency is. I am using a frequency of 1,777,940 hertz for all my figures, measured on pins 16 and 20 of the I/O chip after some warmup. The reciprocal of this frequency is 0.562448 microseconds. This is the unit of time that the Master Counter counts. This should be 1/8 of the crystal oscillator frequency (supposed to be 14,318,180 hertz) but the error in my unit is -.0018%, a quite acceptable figure.

For some practice, here are some parameters:

$\&(16)$ = 60	Master Counter
$\&(17)$ = 99	Counter A
$\&(18)$ = 74	Counter B
$\&(19)$ = 49	Counter C

Forgetting for the moment the Vibrato, Tremolo, Noise and Volume controls, I want to show you what the resultant A, B, and C voices would be. First remember that all of these counters start at count zero, which is one less than you or I count on our fingers. Second, since the period of total time it takes for the A, B, or C counter to toggle is only one change of state, it will take yet another toggle to equal one cycle of some resultant frequency. This is why you will find a 2 in the figures below.

Use this formula: (Substitute counter B or C when calculating those frequencies)

$$\text{Freq. A} = \frac{1}{(\text{Master Counter} + 1) \times (\text{counter A} + 1) \times \text{Time Unit} \times 2}$$

This converts to:

$$\text{Freq. A} = \frac{1}{61 \times 100 \times 0.562448 \times 10^{-6} \times 2} = 145.7 \text{ hertz}$$

$$\text{Freq. B} = \frac{1}{61 \times 75 \times 0.562448 \times 10^{-6} \times 2} = 194.3 \text{ hertz}$$

$$\text{Freq. C} = \frac{1}{61 \times 50 \times 0.562448 \times 10^{-6} \times 2} = 291.4 \text{ hertz}$$

So there are three different frequencies, all available at one time, by simply turning on the A, B, and C volumes. Notice that while Counter A + 1 is twice the value of Counter C + 1, the resultant frequencies have Freq. C twice the value of Freq. A. This is because Counter A has to count twice as many Master Pulses, before toggling, as Counter C does.

Now all that is useful if you are trying to find the resultant frequency for some known values, but we usually want to go the other way. As an example, we will see the steps to find the values need to generate the Bell System Touch Tones, the number "0". Bell Systems require two frequencies outputted whenever one key is pushed on their regular pad. When pushing "0" the two frequencies are supposed to be 1336 and 941 hertz, with an acceptable tolerance of ± 5 hertz. Calculate Freq. A values for 1336 hertz and Freq. B values for 941 hertz:

Step 1 Find the reciprocal of the frequency

Step 2 Divide the reciprocal by the Time Period

Step 3 Divide by 2, and use the nearest whole number. This answer is the total number of time periods that must be counted before having an output toggle for the frequency's half cycle.

For Freq. A, 1 divided by 1336 divided by 0.562448×10^{-6} divided by 2 = 665.39 = 665

For Freq. B, 1 divided by 941 divided by 0.562448×10^{-6} divided by 2 = 944.7 = 945

The values of 665 and 945 are the total number of clock periods as mentioned above. But recall that the Master Counter must count some of those clocks, and it sends a pulse to both A and B Counters, so it must count a number of clocks which have a common denominator to 665 and 945. For ease, I have set the Master Counter to count 5 clock periods before sending out that pulse, so $\&(16) = 4$ (0, 1, 2, 3, 4 = 5) Now that means Counter A must be set to $665/5=133$, therefore $\&(17)=132$. And Counter B must be set at $945/5=189$, therefore $\&(18)=188$.

If you run these results back through the first formulas, you will find that Freq A = 1336.8 hertz and Freq B = 940.7 hertz, well within the tolerance band.

To have both voices come from the tv speaker at the same time only requires that Volume A and Volume B be brought up to sufficient numbers. Both would be maximum if $\&(22)$ is set to 255.

I have found two exceptions that may be unique to my own computer... One, when $\&(16)$ is set to 0, it recognizes it as 2; and the other is when $\&(16)$ is set to 1, it recognizes it as 3. I can't explain why, it may have to do with the high speed at which you are trying to make the Master Counter work.

Load the "SOUND GRAPH" program and get some 'hands on' experience, and try out different ideas. Also, there is an optional modification to the program that will display all eight binary bits of the value that the knob is adjusted to at that time. It does slow down the program, but it is a good teaching aid.

Line # Statements

```

4 .RETURN
5 SOUND GRAPH
6 CHUCK THOMKA 28AP79
10 CLEAR; NT=0
20 FOR A=16 TO 23
30 PRINT #1, A; NEXT A
40 M=KN(1)+128
50 IF M=Y GOTO 90
60 V=M
80 Y=-32; GOSUB 200
90 L=L-JX(1)
100 IF L<11=1
110 IF L>84=8
120 Y=48-8*L
130 BOX -74, Y, 12, 7, 2
140 CX=-77; CY=Y
150 PRINT #1, L+L,
160 IF TR(1) GOSUB 200; R(1
5+L)=Y
170 GOTO 40
200 BOX 6, Y, 148, 7, 2
210 IF Y+2=0 GOTO 230
220 BOX Y+4-68, Y, Y+2, 3, 1
230 CX=64; CY=Y
240 PRINT #1, Y; RETURN
THIS IS OPTIONAL
20 GOSUB 300
300 CY=-40; B=Y
310 FOR F=10 TO 3 STEP -1
320 CX=-77
330 B=B+2
340 PRINT #F, RM,
350 NEXT F
360 RETURN
    
```

Comments

ADJUSTS FOR KNOB RANGE
IF π NOT CHANGED JUMP

ERASES (FLASHES) & COMMAND

REPRINTS & COMMAND #
IF TRIGGER PULLED CHANGE
& COMMAND VALUE

ERASES OLD LINE & NUMBER

DRAWNS NEW LINE

PRINTS NEW NUMBER

IF YOU WANT TO SEE A
BINARY REPRESENTATION
OF THE NUMBER IN
THE VALUE REGISTER

Line # Statements

```

4 .RETURN
5 TOUCH TONE SIMULATE
6 CHUCK THOMKA 28AP79
10 CLEAR
20 A=0
30 NT=3; N=KP
40 IF N=116 GOTO 110
50 IF N=105 GOTO 100
60 IF N=31A=A-1; GOTO 90
70 A=A+1
80 Q(A)=N
90 TV=N,
100 GOTO 30
110 NT=0; PRINT
120 FOR P=1 TO A
130 TV=Q(P)
140 GOSUB 252+Q(P)
150 R(17)=C; R(18)=R
160 R(16)=4; R(22)=255
170 FOR T=1 TO 75
180 NEXT T
190 R(22)=0
200 NEXT P
210 GOTO 30
300 C=132; R=188; RETURN
301 C=146; R=254; RETURN
302 C=132; R=254; RETURN
303 C=119; R=254; RETURN
304 C=146; R=230; RETURN
305 C=132; R=230; RETURN
306 C=119; R=230; RETURN
307 C=146; R=208; RETURN
308 C=132; R=208; RETURN
309 C=119; R=208; RETURN
    
```

Comments

COMMANDS ARE "PRINT", "CLEAR"
AND "ERASE". THESE WILL
PLAY NUMBERS IN MEMORY
CLEAR MEMORY
ERASE ONE NUMBER
PUT NUMBER IN MEMORY

GET FREQ FM TABLE
SET IN NEW FREQS.
SET MASTER OSC. & VOLUME
PLAY NUMBER LOOP

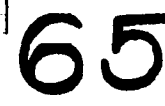
SHUT OFF SOUND
NEXT NUMBER TO DIAL

0= 1336.8 HZ & 940.7 HZ
1= 1209.5 HZ & 697.2 HZ
2= 1336.8 HZ & 697.2 HZ
3= 1481.6 HZ & 697.2 HZ
4= 1209.5 HZ & 769.7 HZ
5= 1336.8 HZ & 769.7 HZ
6= 1481.6 HZ & 769.7 HZ
7= 1209.5 HZ & 850.7 HZ
8= 1336.8 HZ & 850.7 HZ
9= 1481.6 HZ & 850.7 HZ

COLUMN FREQS

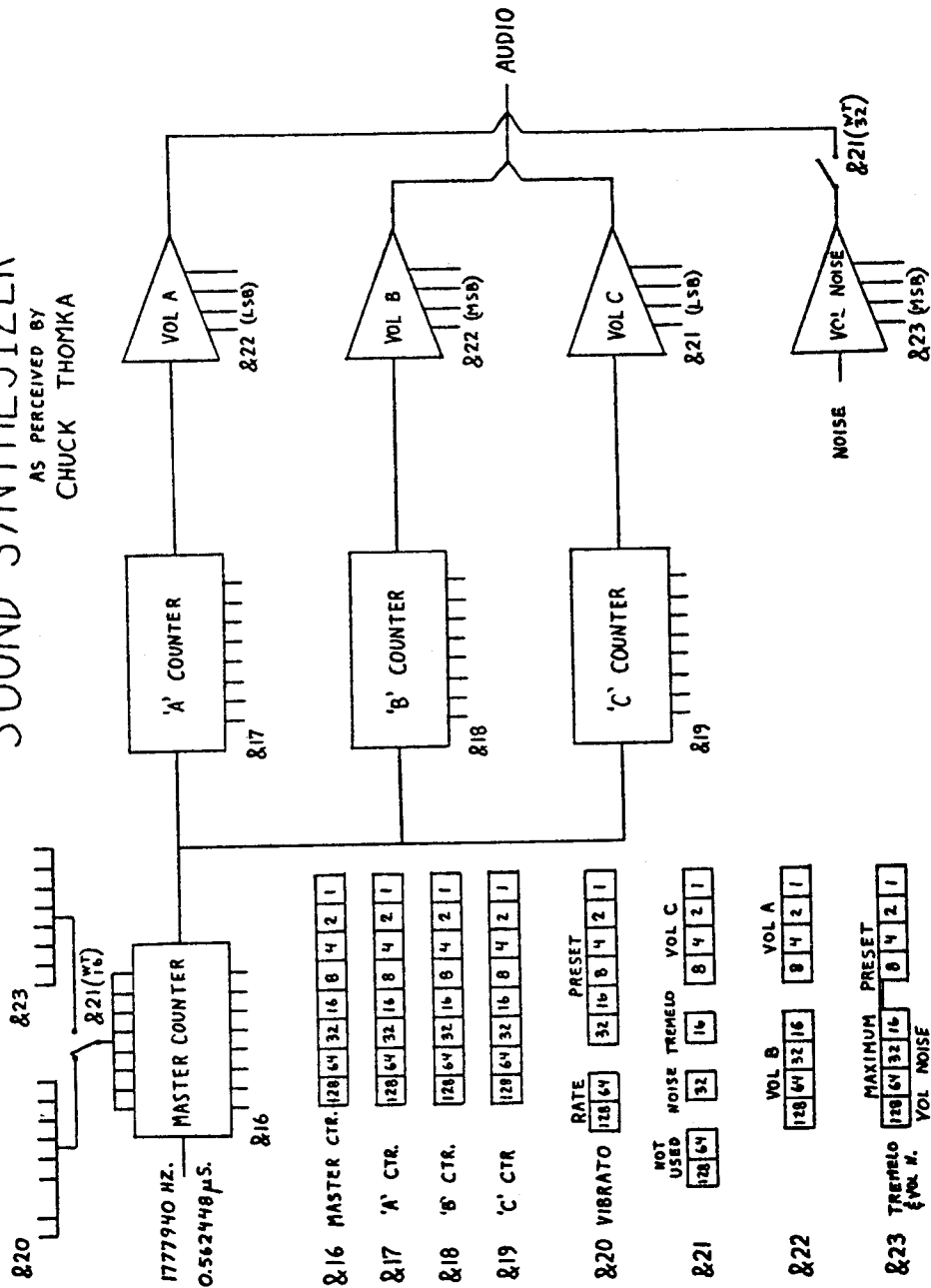
BELL	1209	1336	1477	1633
SYSTEMS	1	2	3	4
REQUIRED	7	5	6	8
FREQ.	7	8	9	852
GRID	*	0	#	941

ONE COLUMN FREQ AND ONE
ROW FREQ PER NUMBER
PUSHED. ALSO I'VE ADDED
A TO D IF YOU HAVE THE
16 KEY KEY-PAD PHONE



THE SOUND SYNTHESIZER

AS PERCEIVED BY
CHUCK THOMKA



TUTORIAL - SOUND SYNTHESIZER. Part 2 by Chuck Thomka

Whenever RESET is pushed, the &(16) to &(23) registers are set to fixed values. (This sort of thing is called DEFAULT) This also happens at POWER TURN ON. &(16) is set to 71, and &(17) through &(23) are set to zero.

Since pushing most keys on the keypad will generate a sound, one of the voices must be used. This means that since most keys have unique tones when pushed, they must be loading unique values into one or more of the registers. The voice used is the 'A' counter. Each key, when pushed, puts a value into the &(17) register that the 'A' counter will count up to. It will also put value 15 into the &(22) register, that will adjust the 'A' volume to its maximum so that the resultant frequency can be heard. At the end of the time of outputting the tone, the &(17) and &(22) are both put back to zero.

At anytime that the computer is stopped, the &(16) register will be set to 71, and &(17) and &(22) will be set to zero. This may affect some results of sound effects in programs where you want those registers to be left at some other values. All the other registers will be as they were last adjusted to, so remember this if you still have a tone or noise remaining after the computer has stopped.

Later in this article is a table of all the sound generating keys, their &(17) values, the resultant frequencies, and any special notes about them. (p.73)

The keys that do not generate sounds are ♯, x, +, and -. These keys will modify the sounds created by the other keys if the modifying keys are used just prior to the normal sound keys.

The divide key (♯) will make the sound one octave lower in frequency than normal. This is done by temporarily making the master counter &(16) count twice as far. So while &(16) is normally at 71, for this one note it will be set to 143. As soon as the note has finished, &(16) will again return to 71 unless the next note is also preceded by a ♯.

The multiply key (x) will make the sound one octave higher in frequency. This is done by making &(16) equal 35 for the time the concerned note is sounding, at the end of which the &(16) will again return to 71.

The plus (+) and minus (-) keys are only used in conjunction with the numbers 1 through 7. This was arranged so that the plus and minus sign would be meaningful in playing musical sharps or flats in the Bally-mentioned 3 octave musical scale.

Another thing to mention is the "Note Timer" or NT. For each number of NT the notes played will be approximately 17 milliseconds long. An NT=0 results in no sound, while the maximum value of NT=255 results in about a 4.335 second note. ($0.017 \times 255 = 4.335$)

The 0 is used to extend the duration of a played note by taking the note timer and increasing it an additional NT quantity for each 0 following the note to be heard. For example, say we are to play a note while NT=10, and that this note is followed by 3 zeros, the resultant NT will be 40. After playing that modified NT, the NT will again return to normal (10 in this example) until called upon again.

A funny thing about this method of extending the duration of a played note is that you still cannot play any note longer than 4.335 seconds. This is because if you had a note timer extended by way of using zeros after a printed character, and it would result in an NT>255, the final result would probably be less than 255. To explain what I mean, you have to know about binary numbers and that the NT register is only 8 bits wide. If, for example, we had an NT of 50 and that some program that we are running is to print a character followed by 5 zeros, we would expect a temporary NT result of 300 ($1+5=6, 6 \times 50=300$) but an 8 bit register's maximum bit count is only 255 while a binary conversion of

decimal 300 requires 9 bits (1 0010 1100). The result is that only the least 8 bits (0010 1100) will be loaded into the NT register, so NT will temporarily be 44. This you see is a lot shorter than we had at first expected and even shorter than the normal NT of 50.

End of "Sound Synthesizer" Tutorial

RESULTANT FREQUENCY

NORMAL $f(16) = 71$

$f(16) = 143$

$f(16) = 35$

109.26 HZ

115.93

122.85

129.97

137.95

146.12

154.34

163.53

173.90

184.28

205.78

230.78

244.49

259.93

274.37

290.51

308.67

327.25

347.80

368.56

391.96

411.56

440.96

465.92

493.87

525.40

548.75

587.94

617.34

CHARACTER(S)

!

"

#

\$

%

&

' (APOST.)

(

)

*

, (COMMA)

• (PERIOD)

/

Y -1

Z 1

[+1 -2

\ 2

] +2 -3

↑ 3

← 4

↓ +4

→ 5

+5 -6

6

+6 -7

7

+7 8

9

:

;

$f(17)$ VALUE

225

212

200

189

178

168

159

150

141

133

119

106

100

94

89

84

79

74

70

66

62

59

55

52

49

46

44

41

39

RESULTANT FREQUENCY

NORMAL $f(16) = 71$

$f(16) = 143$

$f(16) = 35$

162.46 HZ

176.38

187.07

192.72

205.78

220.48

228.64

246.94

257.23

280.61

293.97

308.67

324.92

342.97

363.14

385.84

411.56

440.96

514.45

561.22

617.34

685.93

771.68

823.12

881.91

1028.90

1122.44

1234.68

1371.87

1543.35

1763.83

1927.66

2057.80

2469.36

3086.70

3527.66

4115.60

4938.72

6173.40

8231.20

12346.81

CHARACTER(S)

<

=

>

?

@

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

$f(17)$ VALUE

37

34

32

31

29

27

26

24

23

21

20

19

18

17

16

15

14

13

11

10

9

8

7

6

5

4

3

2

1