THE WEST RAND AMATEUR RADIO CLUB

June 2001
Volume 1, Issue 12

ANODE

Inside this issue:

Special points of interest:

-
-
- Contact details on back page

From the Editor

There will be a Boot Sale on July 28th in the afternoon. Please note that this will be opening in the afternoon to allow for the bring and braai to take place after the Boot Sale.

Subscriptions are going up to R75 per year. Even so this has to be the cheapest club in SA.

Don't forget to attend the AGM on the 23rd June. If you cannot make it, please use the proxy form supplied with your notification.

IBM has announced a small 2.5" disk drive with a new media surface that will have a capacity of 400GB.

When did you last do a backup? Yet another OM this week came to me with another hard disk that had failed. You know Murphy's law better than most, you know he will strike at the worst possible time. Consider doing a backup soon, ok?

Its been fun doing the Anode for the last year. Thank you all for your support.

JB.

Computer Controlled Frequency Synthesizer pt2

Construction

Printed-circuit construction is recommended for this project, though it's possible to assemble it on perforated board using suitable Wire Wrap or soldering hardware. You can make your double-sided pc board from the actual-size artwork given in Fig. 7, but you need special hardware to be able to solder IC sockets (recommended) and P1 into place. Alternatively, you can purchase a ready-to-wire board from the source given in the note at the end of the Parts List.

Assuming pc construction, place the board in front of you oriented as shown in Fig. 8. Begin populating it by installing and soldering into place the sockets for all DIP ICs. Do not plug the ICs into the sockets until after you've conducted preliminary voltage checks and are certain that everything is okay.

Next, install and solder into place the resistors, capacitors, transistors and crystal. Make certain that the electrolytic capacitors are properly oriented and that the transistors are properly based before soldering their leads into place. Then install and solder into place the 25-pin right-angle DB-25 connector in the P1 location.

The only off-the-board components in this project are BNC connector J1, the POWER LED that connects between R13 and ground and the optional jack for the external plug-in power supply.

(Continued on page 2)

Computer Controlled Frequency Synthesizer pt2

(Continued from page 1)

Component problems you might encounter are few. Make sure you use the -2 version of the 82C54 chip. Use zero-temperature-coefficient capacitors for C2 and C4 to maximize temperature stability of the circuit, and use a low-leakage ± 10% tantalum capacitor for C5.

When you have the circuit completely assembled, mount it in any enclosure that will accommodate it preferably plastic for easy machining bf the slot required for P1. Use a hot knife or other suitable tools to make the slot for the DB-25 connector, and drill suitable size holes for mounting the LED, BNC connector and plug-in power-supply connector. Then drill smaller holes for mounting the board to the floor of the enclosure.

If you prefer, you can eliminate the connector for the power supply. In this case, simply route the output cable from the supply through its own hole in the enclosure and wire it directly to the appropriate points on the board. Then use Y,' spacers to mount the circuit-board assembly inside the enclosure.

Mount the LED in its hole in the enclosure. If necessary, use a drop of fast-setting epoxy cement to secure it in place. Then wire it to the circuit board assembly, lengthening its leads with stranded hookup wire as needed and insulating all exposed wiring to prevent short circuiting. The anode lead goes to the free end of R13 the cathode lead to circuit ground.

Control Word	Numbering System		
	Binary	Hex	Decimal
0	0011,0110	36	54
1	0111,0110	76	118
2	1011,0110	B6	182

Finish up by mounting the BNC connector in its hole and wiring it to the circuit-board assembly. Use coaxial cable for the connections from the BNC connector to the OUTPUT and nearby ground holes in the circuit board.

With no ICs, except regulator U5 installed on the circuit-board assembly, power up the circuit. Use a dc voltmeter or a multimeter set to the dc volts

(Continued on page 3)

Computing Coils

To the average amateur, the winding of coils for home-built equipment is really a bit of a hit-or-miss affair. Many old timers, however, seem to have the ability to look at a coil, sniff twice in the air and say: 'I reckon that will just about tune eighty' - and be right.

This, though, is a result of many years constructional experience and even then, if presented with a coil of unusual diameter or wound with a much thicker or thinner gauge wire than usual, the OT will quite likely be hopelessly wrong.

As a consequence, over the years many constructors have preferred to purchase ready wound coils - from the Eddystone and Wearite ranges available just after the war and the Denco and Electroniques coils of the fifties to the Toko range today.

Commercially wound coils, however, can be expensive and if the equipment is being constructed 'from the junk-box', may add considerably to the final cost. What alternative options are therefore available?

The first possibility is that the equipment under construction is being made to a published design, in which case the coil data should be supplied. But what if the prescribed diameter coil formers or the correct gauge wire is not available?

The second possibility is to search through old magazines or handbooks and use the coils described for a broadly similar circuit. For this, however, a large and comprehensive library is necessary to ensure even a moderate chance of success.

The third method is to delve into the text books and calculate the values required. This again gives the choice of three

(Continued on page 3)

Computer Controlled Frequency Synthesizer pt2

(Continued from page 2)
function to first check the polarity of the power-supply connections. If it's correct, clip the common meter lead to any circuit-ground point and use the "hot" lead to probe the Vcc contacts of the IC sockets (pin 14 of U1 and U2; pin 16 of U3, U4, U6 and U7., and pin 24 of U5). If you fail to obtain a reading of +5 volts at any of these points, power down and correct the problem before proceeding.

Once you're sure your wiring is correct, power down and plug the ICs into their respective sockets. Make sure each IC is properly oriented and that no IC pins overhang the sockets or fold under between ICs and sockets.

Programming

To use the Frequency Synthesizer, you must create a program that asks for the desired output frequency and then automatically sets up the counter chip and latch to generate the specified frequency. You must perform a few preliminary procedures before you can get into actual setting of the counter. The counter chip provides six operating modes for each counting element. Two modes are repetitive; once set, they keep on running. The other four are one-shot modes. Since you're dealing with frequency dividers, you want the repetitive modes.

In the rate-generator mode, the counter is preset to some number between 2 and 216. When the counter indexes to 0, a single clock pulse appears at the output, and countdown starts again. The second repetitive mode, called square-wave mode, works with the N, M and output divider counter of the Synthesizer. After priming the system, your first programming task is to set each of the three counters in U5 to the square-wave mode.

You can then set up each counter to count in binary or BCD. Keeping things simple, I chose the binary mode. Next is the read/write least significant-byte-first, most-significant-byte-second mode. The control word format for the various modes is

(Continued on page 4)

Computing Coils

(Continued from page 2)
options: pencil and paper, a nomogram or a suitable computer program.

A standard formula for the number of turns necessary for a coil of given inductance is:

```
N = SQR( L * (457.2 * d
+ 1016 * I) / MU ) / d
```

where: N is the number of turns; L is the inductance required; MU is the permeability of the core; d is the internal diameter; I is the length in mm.

On inspection of this formula, a mathematical difficulty is immediately apparent. To calculate the number of turns, the length of the coil must be known, but how can this be known before the number of turns has been calculated?

The usual subterfuge is to make an intelligent guess for a value of length, and then do the calculation. The number of turns calculated, a suitable wire gauge and turns spacing can be selected. If for any reason a 'silly' result is obtained (such as a 1kWatt tank coil using 42-gauge wire), another calculation using an alternative value for 'L' will have to be made. In all, this method can tend to be both time consuming and laborious.

The second method is to use a nomogram. This can be quite quick and efficient but, like most graphical methods, requires a certain experience to give an accurate result.

The third method in this modern age is to use the digital dexterity of the home microcomputer equipped with a suitable program.

The program

The program described in this article is designed to calculate the number of turns

(Continued on page 5)

Computer Controlled Frequency Synthesizer pt2

(Continued from page 3)
illustrated in Fig. 6. From Fig. 6, you can see that bits through DS of each control word are the same. Bits D6 and D7 differ because they specify different counters. Therefore, the control words are summarized in table 1.

Before you can feed the control words to the counter chip, you must prime the system, make sure the control lines and latch output lines are at the correct levels. There are only two lines with you must concern yourself -/CS-/WT to the counter chip and enable to U7. These must be in the high and low states, respectively. Both lines come from the printer control register and can be, set to the required levels by outputting an appropriate bit pattern to the printer control port. The bit pattern is as follows:

C3 C2 C1 C0
0 1 X X
Latch Strobe
Write Strobe

The Xs are don't-care states. Keep in mind that outputs C3, C1 and C0 of the control port are inverted. Arbitrarily setting the Xs to 1s, the nibble pattern becomes 0111 binary, which is 7 hex and decimal.

Before you can send the control nibble to the control port, you must XOR it with hex B. So, would do the trick if you knew the address of the control register. You do know it. Assuming you're using LPTI, the control-port address is 03BE hex.

```
CTW = &H7 XOR &HB
OUT (control-register ad-
dress), CTW
```

You must now set up the latch to turn off the output signal. Since the MUX on/off control is connected to the L3 output of the latch and is active, you must set it high to turn off the output. To do so, first set the data port to 08 hex or decimal and then strobe printer control port C3 high and then low. The data on the data line then gets

Listing 1

```
OUT &H3BC, 8   'DATA TO
DATA PORT
CTW = &HF XOR &HB  'CONTROL
WORD FOR LOGIC 1 TO LATCH
OUT &H3BA, CTW'ENABLES
LATCH
CTW = &H7 XOR &HB  'CONTROL
WORD FOR LOGIC 0 TO LATCH
OUT &H3BE, CTW'DISABLES
LATCH
```

Listing 2

```
DT   =   &H3BC    'Printer
     data port address
CR   =   &H3BE    'Printer
     control port address
CT   = n               '
Counter # (n = 0, 1, 2)
CW(0)    =    &H38
     'Control word counter
zero
CW(1)    =    &N78
     'Control word counter
one
CW(2)    =    &HBB
     'Control word counter
two
```

Listing 3

```
FOR I = 1 TO  3
     OUT DT,    CW(I)
     'Control word to data
register
     CTW  =  &H3  XOR  &HB
     'Set  "Write"  to  a
zero
     OUT CR, CTW
     'Output  to  control
port
     CTW  =  &H7  XOR  &HB
     'Set "Write" to a one
     OUT CR, CTW
     'Output  to  control
port
```

Listing 4

```
N = 6,000
UPPER  BYTE  =  NU  =  INT
(N/256)
LOWER  BYTE  =  NL  =  N  -
256*NU
```

Listing 5

```
OUT DT, NL
     'Lower  byte  to  data
port
CTW =     &H0 XOR &HB
     'Write bit to zero
OUT  CR, CTW
     'Output  to  control
port
CTW =     &H7 XOR &HB
     'Write it to one
OUT  CR, CTW
     'Output  to  control
port
OUT DT, NU
     'Lower  byte  to  data
port
CTW =     &H0     XOR  &HB
     Write bit to zero
OUT  CR, CTW
     'Output  to  control
port
CTW =     &H7 XOR &HB
     'Write it to one
OUT  CR, CTW
     'Output  to  control
```

(Continued on page 5)

Computer Controlled Frequency Synthesizer pt2

(Continued from page 4)

port

Listing 6

```
SUB CTSET
    IF CT = 0 THEN S1 = 11: S2 = 15
    IF CT = 1 THEN S1 = 10: S2 = 14
    IF CT = 2 THEN S1 = 9: S2 = 13
    OUT DT, NL : OUT CR, S2: OUT CR,S1: OUT CR,S2
    OUT DT, NU: OUT CR, S2: OUT CR,S1: OUT CR,S2
END SUB.
```

Listing 7

```
A:
'** ASKS FOR OUTPUT FREQUENCY
INPUT "Enter Frequency in Hz, (1.000 to 10,000,000)";Fo
'** CHECKS RANGE OF INPUT
    IF Fo<1 OR IF Fo>11^7 THEN
        PRINT "OUT OF RANGE": BEEP
        SLEEP (4): CLS: GOTO A
    END IF:
'**COMPUTES EXPONENT SCALING FACTOR
EP = 6 - INT(LOG(Fo)/LOG(10))). 'Scaling constant
IF EP <0 THEN EP = 0
Fx = Fo * (10^EP)  'Scale input to 1-10MHz
H = INT(Fx/1000)  'Counter 1 divisor
NU = INT(N/256) 'Upper byte counter 1
NL = H - 256 * NU  'Lower byte counter 1
CT = 1  'Selects counter 1
CALL CTSET  'Sets counter 1
'** SETS THE VCO RANGE
    IF Fo<3000000 THEN VCO = 0 ELSE VCO = 1
'** SETS DIVIDER SELECTION. NOTE, DCO AND DC1 ARE USED BY
'** THE LATCH TO SET THE DECODER INPUTS TO THE 74HC151
SELECT CASE EP
    CASE 0
        DCO = O: DC1 = 0: NU = O: NL = 0
    CASE 1, 2, 3, 4
        DCO = 1: DC1 = 0
        IF EP = 1 THEN NU = 0 NL = 10
        IF EP = 2 THEN NU = 0 NL = 100
        IF EP = 3 THEN NU=INT( 1000/256): NL=1000-(NU*256)
        IF EP = 4 THEN NU=INT(10000/256): NL=10000-(NU*256)
```

(Continued on page 6)

Computing Coils

(Continued from page 3)

necessary for a coil of given inductance, or if the value of the required inductance is not known, derive this from the resonant frequency required, and the parallel capacity.

The program will calculate to an accuracy of better than 5% within the limits of the formula, and give indication when these limits have been exceeded.

The language used is standard Microsoft Basic and the program has been successfully run on BBC B, Oric and Sanyo computers. Furthermore, it may be easily modified to 'Sinclair' Basic for Spectrum machines. Doubtless many other machines will be equally suitable.

Finally, being fully aware that the longer the listing the greater the possibility of inadvertent error, two measures have been taken to guard against this. Firstly, the program has been 'pruned' to minimum length, even at the expense of slightly greater complexity and, secondly, a series of test calculations have been devised which will thoroughly check all aspects of the program. If, after keying in the listing, the test calculations can be successfully performed, it can be confidently assumed that the program is error-free and is ready for use.

Using the program

If, after the program is loaded, the 'RUN' instruction is given, the screen will clear and the user will be asked to input certain parameters of the coil required such as inside diameter, wire gauge, wire spacing and permeability of the core.

This complete, the user is

(Continued on page 6)

Computer Controlled Frequency Synthesizer pt2

(Continued from page 5)

```
      CASE 5
                DCO = 0: DC1=1:
   NU=INT(10000/256):
   NL=10000-(NU*256)
   CASE  6
                DCO = 1: DC1=1:
NU=INT(10000/256):
NL=10000-(NU*256)
END SELECT
CT = 2: CALL CTSET
LTCH =8 + 4 * DC1 + 2 * DC2
+ 1 * VCO: CALL LATCH
'** SLEEP DELAY LETS PLL
LOCK AND SETTLE DOWN
SLEEP (1)
'** TURNS OUTOUT SIGNAL ON
LTCH = LTCH -8
CALL LATCH
GOTO A

SUB LATCH
      SHARED DT, CR, LTCH
      OUT DT, LTCH
                         OUT  CR,12:
OUT CR, 4: OUT CR,12
END SUB
```

Listing 8

```
SUB REFSET
      Fcy = 6002000: N =
Fcy / 1000: NU = INT(N /
256)
      NL = INT(N - (NU *
256)): CT = 0: CALL CTSET
END SUB

SUB SCRN
PS = (80 - LEN(QT$)) / 2:
LOCATE 22, PS: PRINT Qr$:
LOCATE 1, 1: PRINT "
      ENTER FREQUENCY"
      PRINT
      PRINT " (1.000Hz <=
Fe <= 10,999,000MHz, 4
SIGNIFICANT DIGITS)"
      LOCATE 7, 35: PRINT
END SUB

SUB FREQCOMP
      LTCH = 0: CALL
LATCH
      'TURNS OFF OUTPUT
      EP = 6 - INT((LOG
(Fo)) (LOG(10))) '
COMPUTES
      IF EP < 0 THEN EP =
0      'HIGH & LOW
BYTES
      Fx = Fe * (10 ^
EP): N      INT(FX
   1000)           'FOR
COUNTER 1
      NU = INT(N / 256):
NL  H  (NU  256)
      CT = 1: CALL CTSET
'SETS PLL DIVIDER
      IF Fx > 300 0000
```

(Continued on page 10)

Computing Coils

then asked if the required inductance is known. If so, it is entered. If not, the frequency of operation and the parallel capacity are requested.

During this sequence each figure is compared with predetermined limits to ensure that the validity of the formula will not be compromised. Despite this, it is possible for an answer to be derived which is, in itself, beyond the limit of accuracy of the calculation.

In such a case, although the answer will be printed, this will be accompanied by an appropriate warning such as "TOO LONG" or "TOO SHORT".

Program check calculations

Coil inside dia (mm) = 25
Wire gauge (SWG) = 34
Wire spacing (O=none) = 0
Ferrite mu (air=1) = 5
Inductance (uH) = 20.6776
Frequency (MHz) = 3.5
Capacity (pF) = 100 Coil turns = 9.51218
Length (mm) = 2.36637
!! TOO SHORT!!

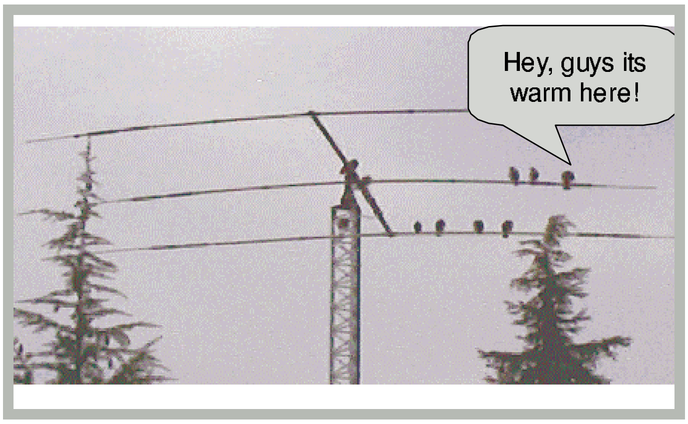
Coil inside dia (mm) = 10
Wire gauge (SWG) = 22
Wire spacing (O=none) = 0
Ferrite mu (air=l) = 1
Inductance (uH) = 1
Frequency (MHz) = 0
Capacity (pF) = 0
Coil turns = 10.765
Length (mm) = 8.40534

Coil inside dia (mm) = 12.5
Wire gauge (SWG) = 25
Wire spacing (O=none) = 1
Ferrite mu (air=l) = 1
Inductance (uH) = 1.1
Frequency (MHz) = 0
Capacity (pF) = 0
Coil turns = 10.8593
Length (mm) = 12.1152

Coil inside dia (mm) = 12.5
Wire gauge (SWG) = 12
Wire spacing (O=none) = 1
Ferrite mu (air=l) = 1
Inductance (,uH) = .46
Frequency (MHz) = 0
Capacity (pF) = 0
Coil turns = 12.0153
Length (mm) = 64.3721
!!TOO LONG!!

The operator is then asked whether a further calculation is required. If it is, the program is re-run. In such circumstances certain parame-

(Continued on page 7)



Computing Coils

(Continued from page 6)

ters, such as wire gauge or coil diameter, may remain constant. If, when these are requested, the RETURN key is pressed, the computer will assume the previous value, thus obviating re-entry.

Note: If, after a calculation in which the required inductance has been derived from frequency and capacity, the induct ance is specified in the succeeding calculation, the frequency and capacity previously specified will be displayed in the print out. These figures, however, will be meaningless and will in no way affect the accuracy of the computation.

Operation of the program

Although a detailed knowledge of the program is not necessary for its use, a few notes about its operation may be of interest.

The first action of the program is at line 30 where a 'workspace' of 9 variables (X (1) to X(9)) is defined by use of the DIMENSION instruction. Lines 40 to 70 then cause the heading to be printed on the screen.

Line 80 (B$ = is inserted to remove any value for B$ which may be remaining from a previous calculation, but has no effect during the initial input phase of the program.

Lines 100-120 enable the program to request the first four lines of data (lines 460 to 490) by means of the input/output sub-routine, which starts at line 570. This sub-routine also ensures that the figures entered are within predetermined parameters.

The computation mode of the program is defined at lines 130 to 150. This either reads the required inductance directly or calculates it from the resonant frequency and capacity. In the former case, line 160 is merely a continuation of the sequence in lines 100 to 120.

Before the frequency and capacity can be entered (at lines 190 to 210), the program must issue a dummy read instruc-

(Continued on page 9)

Printer Port Basics

PC printer ports consist of three registers: data, status and control. Port addresses depend on the parallel printer port used and are summarized as follows:

Port	Data	Status	Control
LPT1	03BC	03BD	03BE
LPT2	0378	0379	037A
LPT3	0278	0279	027A

The addresses here are in hexadecimal notation.

To send a byte to the port, use the OUT instruction. For example, OUT &H03BC (any number between 0 and 255) sets data-port lines to the states specified by the output number.

Therefore, if you want to send a byte to the Frequency Synthesizer, send it to the data register.

Note that all program segments are compatible with QBASIC. If you use TBASIC, change the SLEEP instruction to the DELAY function. Otherwise, the instruction listings are okay for TBASIC.

Since we don't have to "read back" from the Synthesizer, we can ignore the status port. On the other hand the control port makes it possible to transfer data between the Synthesizer and PC with which it's used.

The status port isn't a full eight-bit port. Rather, it's a four bit (nibble) port. Just as with the data port, you can write to the status port with an OUT instruction, with appropriate change in address, naturally.

A minor problem with the four bits from the control port is that three of the bits are inverted to meet the requirements of printer interfaces in which the lines had to be logic 1s.

Shown in Fig. 3a are the pin-out details for a standard DB-25 printer-end parallel printer port. The four control-port lines are on pins 1, 14, 16 and 17 with only the one on pin 16 not inverted. Thus, when you send 0 to the control port,, out comes a hex B instead of 0. This can easily be corrected by using the XOR (exclusive-OR) function on the nibble to be sent to the control port prior to executing the OUT instruction.

To send hex C to the control port, for example, first XOR hex C with hex B. Then OUT the results to the control port. The example shown in Fig. B should clarify things.

It's easy to control peripheral hardware through a printer port. Just use a port instruction and send data to the data port. Reset the four control lines emanating from the control register with OUT instructions to take care of the hardware handshakes.

Suppose you have a D/A converter on the data lines of a printer port. Assume that the D/A needs a negative-going strobe pulse to write data into its internal register. Here's how you'd accomplish this:

(1) Set the control port to hex F:
CP = &HF XOR %HB Prime D/A OUT Control Port Address, CP " "
(2) Load the D/A byte into the data register with an OUT instruction:
OUT Data Port Address, Data Word
(3) Assuming bit 0, pin 1, of the DB-25 connector is used to strobe the D/A register ;

```
CP   x   &HE   XOR   &HB
'Pin 1 to 0V
OUT Control Port Address,
CP       'Pin 1 to 0V
CP   =   &FH   XOR   &HB
'Pin 1 to +V
OUT Control Port Address,
CP      'Pin 1 to + 5V
```

So the name of the game is: send the byte to the data port, send the control word or pulse to the control port. That's it. The only catch is that the control "pulse" line(s) to the output hardware must be clean (no noise or extraneous pulses).

Taken from the Magazine Computer Craft.

Computing Coils

(Continued from page 7)

tion (line 180) which makes no use of the information but merely serves to move the internal data pointer forward one line and allows the next input action to use the next two data statements at lines 510 and 520.

With the frequency required and parallel capacity now defined, the necessary inductance is calculated at line 220.

On lines 230 and 240 the wire gauge is converted to 'close wound turns per millimeter' using an empirical formula which is accurate to better than 5%, whilst on line 260 the mean diameter of the coil is determined from the inside diameter and the diameter of the wire.

The calculation of the number of turns of wire necessary to achieve the required inductance is on lines 270 to 300, with the answer being associated with the appropriate data statement at line 310 and the length of the coil calculated at line 320.

Line 330 then sets the internal data pointer back to the top of the data list (i.e. line 460).

The complete list of data, which has either been provided by the user or calculated within the program. is then printed by the action of lines 340 to 380.

The ratio of length to diameter is next examined in lines 390 and 400 and if the result is outside the limits of accuracy of the formula used in the calculation, an appropriate warning is given.

In conclusion, the user is given the option of a further calculation at lines 420 to 440 and if this is not taken up, the program stops at line 450.

Future programs

This is the second in a series of engineering' programs which will appear in R&EW over the next few months. These will all be written in standard Microsoft Basic so as to be compatible with the majority of home computers.

It is hoped that the use of these programs will considerably simplify the design of home built equipment.

Unfortunately Radio & Electronics World ceased publication in the early 1980's.

Computing Coils - the program

```
10 REM COIL TURNS CALCULATOR
20 REM J.M.HOWELL JUNE 1984
30 DIM X(9)
40 CLS
50 PRINT
60 PRINT TAB(5); "COIL TURNS CALCULATOR"
70 PRINT
80 B$ = ""
90 RESTORE
100 FOR I = 1 TO 4
110 GOSUB 570
120 NEXT I
130 PRINT "INDUCTANCE KNOWN? (Y/N)      "
140 INPUT C$
150 IF C$ = "N" THEN GOTO 180
160 GOSUB 570
170 GOTO 230
180 READ A$, LO, HI
190 FOR I = 6 TO 7
200    GOSUB 570
210 NEXT I
220 X(5) = ((25330 / X(6)) / X(6)) / X(7)
```

Computer Controlled Frequency Synthesizer pt2

(Continued from page 6)

```
THEN VCO = 1 ELSE VCO = 0
'SETS VCO RANGE
    Fd = (INT(FX / 1000)) * 10 ^ (3 EP)
    LOCATE 15, 22:
    PRINT "OUTPUT FREQUENCY IS USING Hz"; Fd

SELECT CASE EP
    CASE 0
    DCO = 0: DC1 = 0: NU = 0: HL = 0
    CASE 1, 2, 3, 4
    DCO = 1: DC1 = 0
    IF EP = 1 THEN NU = 0: NL = 10
    IF EP = 2 THEN NU = 0: NL = 100
    IF EP = 3 THEN NU = INT(1000 * 256): NL = 1000 (NU 256)
    IF EP = 4 THEN NU INT (10000 / 256): HL = 10000 (NU * 256)
    CASE 5
    DCO = 0: DC1 = 1: NU = INT(10000 / 256):
    HL = 10000 (NU * 258)
    CASE 6
    DCO = 1 : DC1 = 1: NU = INT(10000 / 256):
    NL = 10000 (NU * 256)

END SELECT
CT = 2: CALL CTSET 'SET OUTPUT DIVIDE
LTCH = 8 + 4: DC1 + 2 * DCO + 1 VCO: CALL LATCH
    'SETS LATCH
SLEEP (1) 'ONE SECOND DELAY
LTCH = LTCH 8: CALL LATCH 'TURNS ON OUTPUT
END SUB
```

The previous program segment for setting counter 0 can now be done with the following two lines:

```
CT = 0: NU = INT(WW/256):
NL = Wffl 256*NU
CALL CTSET
```

The next thing to do is set counter 1. If you want an output between 1MHz and 10 MHz, this is easy to do. Just divide the frequency by 1,000 and then break the results down to NU and NL and send them off to the Synthesizer via the CTSET subroutine.

For the 1Hz to 1MHz range you must do a bit of fancy footwork. You have to scale the desired frequency up to the 1MHz to 10MHz range, set counter 2 and then decide which divider output to use to get the desired frequency. Listing 7 shows one way of doing this. In looking over this Listing, you'll notice that one new subroutine (shown shaded) has been added.

A complete QBASIC listing for a *simple* Frequency Synthesizer program is given in Listing 8. Feel free to change, add to or rewrite it any way you choose. If you want to, you can write a pop-up program that activates when an unusual combination of keys is pressed and include a software switch to turn on and off the output signal. How about a sweep mode where the frequency changes by, say, 1,000 Hz every 5 seconds up to some predetermined frequency, after which the sweep reverses. As you can see, how you use the Frequency Synthesizer presented here is limited only by your imagination and ability to write routines to effect what you want.

Parts List

Semiconductors

Q1-2N4401 silicon npn transistor
U1, U2 - 74HC04 hex inverter
U3 - 74HC390 dual decade counter
U4 - 74HC151 eight-input multiplexer
U5 - 82C54-2 triple 10-MHz 16-bit counter
U6 - 74HC4046 phase-locked loop and phase comparator
U7 - 74HC75 four-bit latch
U8 - LM7805 fixed + 5-volt regulator

Capacitors

C1 - 8-to-50pF single-turn trimmer capacitor with leads on 0.2' centres
C2 - 20pF npo ± % ceramic disc
C3,C6 100-pF ceramic disc
C4 150pF npo ± % ceramic disc
C5,C8 10uF, 16-volt ± % tantalum
C9 thru C12 0.1pF ceramic disc

Resistors (0.25watt, 5% tolerance carbon-film)

R1, R5, R11 2,200 ohms
R2 220,000 ohms
R3 47 ohms
R4, R12 100 ohms
R6 10,000 ohms
R7,R10 22,000 ohms
R8 3,000 ohms
R9 8,200 ohms
R13 680 ohms

Miscellaneous

(Continued on page 11)

Computer Controlled Frequency Synthesizer pt2

J1 -Panel-mount coaxial BNC connector
P1 -Right-angle, pc-mount DB-25 connector
Y1-6-MHz crystal (see text)

Printed-circuit board (see text); 9- to 12-volt dc, 200-mA plug-in wall mount power supply; suitable 3-to 4-foot cable terminated at both ends in DB-25 connectors; suitable enclosure (see text); red panel-mount light emitting diode; power jack (optional; see text); machine hardware; solder; etc.

This article was taken from Computer Craft Magazine. Unfortunately this magazine has disappeared so don't expect any support for the design. JB

Computing Coils - the program

(Continued from page 9)

```
230 T = (1.1312 ^ X(2)) / 11.76
240 IF X(2) > 24 THEN T = T * .967 ^ (X
(2) - 24)
250 S = X(3) + 1
260 D = X(1) + 1 / T
270 N = D * T / S
280 FOR I = 1 TO 10
290    N = SQR(X(5) / X(4) * (457.2 * D
+ 1016 * N * S / T)) / D
300 NEXT I
310 X(8) = N
320 X(9) = S * N / T
330 RESTORE
340 CLS
350 FOR I = 1 TO 9
360     GOSUB 550
370  NEXT I
380  PRINT
390  IF X(9) > D * 4 THEN PRINT "!! TOO
LONG !!"
400  IF X(9) * 4 < D THEN PRINT "!! TOO
SHORT !!"
410  PRINT
420  PRINT "STOP NOW? (Y/N)"
430  INPUT A$
440  IF A$ <> "Y" THEN GOTO 40
450  END

460  DATA  "COIL INSIDE DIA. (mm)",1,300
470  DATA  "WIRE GAUGE (SWG)",8,42
480  DATA  "WIRE SPACING (O=NONE)",0,5
490  DATA  "FERRITE MU (AIR=1)",1,100
500  DATA  "INDUCTANCE (uH)",0.01,10000
510  DATA  "FREQUENCY (MHz)",0.01,1000
520  DATA  "CAPACITY (pF)",0.1,100000
530  DATA  "COIL TURNS",0,0
540  DATA  "LENGTH (mm)",0,0
550  B$ = " = " + STR$(X(I))
560  PRINT
570  READ A$, LO, HI
580  PRINT A$; TAB(22); B$
590  IF B$ <> "" THEN RETURN
600  INPUT C$
610  IF LEN(C$) = 0 THEN GOTO 630
620  IF C$ = "0" OR VAL(C$) <> 0 THEN
X(I) = VAL(C$)
630  IF X(I) <= HI THEN GOTO 680
640  CLS
650  PRINT
660  PRINT "TOO HIGH - LIMIT =", HI
670  GOTO 580
680  IF X(I) >= LO THEN RETURN
690  CLS
700  PRINT
710  PRINT "TOO LOW - LIMIT =", LO
720  GOTO 580
```

The West Rand Amateur Radio Club
26.14122 South - 27.91870 East

P.O. Box 562
Roodepoort
1725

Phone: +27 11 726 6892
Email: john.brock@pixie.co.za

Bulletins (Sundays at …)
11h15 Start call in of stations
11h30 Main bulletin start

Frequencies
145,625 MHz (West Rand Repeater)
10,135 MHz (HF Relay)

Radio Amateurs do it with more frequency!

We are going on the Web
www.zs6wr.org.za

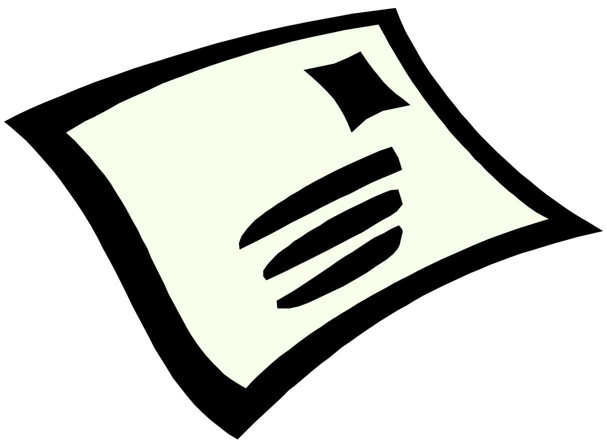
Please note this has been just been registered. Our site will be up in the new year.

Chairman	Bill	ZS6REV	726 6892	---
Vice-Chairman	John	ZS6BZF	768 1626 (A/H)	john.brock@pixie.co.za
Treasurer	Dave	ZR6AOC	475 0566	david.cloete@za.unisys.com
	Simon			simsny@global.co.za
	Anton			anton@xglobe.co.za
	Chris	ZR6AVA	673 2726	botham@global.co.za
	Keith	ZS6AGF	763 6929	mwbronie@iafrica.com
	John	ZS6WL		johnwhit@freemail.absa.co.za

West Rand members input - we need your input!

To make this the best ham radio magazine in South Africa we need your input. Please submit articles, comments, suggestions etc.

Please send plain text with no formatting to the email address



We need your input! Email us articles, comments and suggestions please.
john.brock@pixie.co.za