

## COILWIND

[i\\_built\\_a\\_coil\\_winder.pdf](#)

*This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.*

Coil Winder by Ron Pugh ronee@telus.net

**Please read the attached PDF for a detailed explanation of this project.**

The coil winder created by Ron uses a Maximite for computer control in conjunction with a pulse width modulation circuit to control bobin motor speed and a Sparkfun stepper motor controller.

The current video is available at <http://youtu.be/IXV4aA0fCws>

### Count1.bas:

```
10 ' Count Demo, Mick Gulovsen, Ron Pugh
20 '
30 Setpin 11,5
40 A=pin(11)
50 Print A
60 B=pin(11)
70 IF B>A then A=B: goto 50
80 goto 60
```

### PWM1.bas:

```
10 ' PWM1.bas using Sound output pin
20 ' Keith Williams, Ron Pugh
30 Cls
40 Line(100, 100) - (300, 150), 1, B ' draws a box
50 Font #2 ' big print
60 Duty = 30 ' init to 30/70 PWM
70 ' 200 Hz duration about 33 minutes 50 % duty
80 '
90 GoSub 200
100 I$=Inkey$ : If I$="" Then GoTo 100 ' Test for keyboard input until key
press
110 If I$="z" Then Duty = 15
120 If I$="x" Then Duty = 25
130 If I$="c" Then Duty = 40
140 If I$="v" Then Duty = 50
150 If I$="b" Then Duty = 60
160 If I$="n" Then Duty = 75
170 If I$="m" Then Duty = 100
180 GoTo 80
```

```
190 '  
200 ' Subroutine to sound and print the duty cycle  
210 Sound 200, 2000000, Duty  
220 Print @(125,120) "Speed ";Duty;"% "  
230 Return
```

### PWM2a.bas:

```
10 ' PWM2ac.bas using Sound output pin  
12 ' Plus nuber of turns (total count)  
15 ' Mick Gulovsen, Ron Pugh  
20 CLS  
30 Line(100, 100) - (250, 150), 1, B1  
40 Font 2 : Print@(120, 120) "Speed %"  
50 Line(300, 100) - (400, 150), 1, B2  
60 Line(100, 250) - (250, 300), 1, B3  
70 Font 2 : Print@(105, 270) "Turns Count"  
80 Line(300, 250) - (400, 300), 1, B4  
90 Setpin 11,5  
100 Duty = 30 ' init to 30/70 PWM  
110 Sound 200, 2000000, Duty ' 200 Hz duration about 33 minutes 30 % duty  
130 FONT #2 ' big print  
140 I$=Inkey$ : if I$="" then goto 270 'Test for keyboard input until key  
press  
150 If I$="z" Then Duty = 15 : Print @(320, 120) " 15" : GoTo 230  
160 If I$="x" Then Duty = 25 : Print @(320, 120) " 25" : GoTo 230  
170 If I$="c" Then Duty = 40 : Print @(320, 120) " 40" : GoTo 230  
180 If I$="v" Then Duty = 50 : Print @(320, 120) " 50" : GoTo 230  
190 If I$="b" Then Duty = 60 : Print @(320, 120) " 60" : GoTo 230  
200 If I$="n" Then Duty = 75 : Print @(320, 120) " 75" : GoTo 230  
210 If I$="m" Then Duty = 100 : Print @(320, 120) "100" : GoTo 230  
220 GoTo 140  
230 Sound 200, 2000000, Duty  
250 GoTo 130  
260 A=pin(11)  
270 if pin(11) <>A then A=Pin(11): Font 2 : Print @(320, 270) A  
280 goto 140 ' go back to check for key press
```

### PWM2ac.bas:

```
' PWM2ac.bas using Sound output pin  
' Plus number of turns (total count)  
' Mick Gulovsen, Ron Pugh  
' This is the same as PWM2A.BAS but without line numbers and  
' in a more structured programming form.
```

```
Cls  
Line(100, 100) - (300, 150), 1, B ' draws a box  
Font 2 : Print@(120, 120) "Speed %"  
Line(300, 100) - (400, 150), 1, B  
Line(100, 250) - (250, 300), 1, B
```

```

Font 2 : Print@(105, 270) "Turns Count"
Line(300, 250) - (400, 300), 1, B

SetPin 11,5

Font 2      ' big print
Soundit 30 ' init to 30/70 PWM
           ' 200 Hz duration about 33 minutes 50 % duty

Do      ' loop for ever

    Do ' Print count while waiting for change of Duty
        If Pin(11) <> A Then
            A=Pin(11)
            Font 2
            Print @(320,270) A
        EndIf
        I$=Inkey$
    Loop Until I$<>""

    If I$="z" Then
        Soundit 15
    ElseIf I$="x" Then
        Soundit 20
    ElseIf I$="x" Then
        Soundit 25
    ElseIf I$="c" Then
        Soundit 40
    ElseIf I$="v" Then
        Soundit 50
    ElseIf I$="b" Then
        Soundit 60
    ElseIf I$="n" Then
        Soundit 75
    ElseIf I$="m" Then
        Soundit 100
    EndIf

Loop

Sub Soundit Duty
    Sound 200, 2000000, Duty
    Print @(320,120) Duty;" "
End Sub

```

### PWMs.bas:

```

10 ' -----
20 ' Coil Winder by Ron Pugh
30 ' Version 2 2012
40 ' With programming assistance from

```

```
50 ' Mick Gulovsen and Hugh Buckle
60 ' -----
70 ' PWMs.bas using Sound output pin
80 ' Plus number of turns (total count)
90 Cls
100 Font #2
110 Input"Enter Wire Dia (in Thou)";dia
120 Input"Enter Bobbin Length (in Thou)";bobbin
130 ' steps per wire thickness calculation
140 ' assumptions
150 ' 200 pulses per rotation on stepper
160 ' 10 rotations per inch stepped
170 ' Direction starts with 0 then goes to 1
180 No_Puls=Cint(dia*2/10) ' No of Pulses per 1/10th rotation
190 Puls_Fr = dia*2/10 - No_Puls ' saves the fractional part of an extra
rotation
200 Max_puls=bobbin*2 : Puls_Cnt=0
210 SetPin 1,8 : Pin(1)=0 ' set pin 1 for Stepper pulses and init as LOW
220 SetPin 2,8 : Pin(2)=0 : Dir_stat=0 ' set pin 2 for Direction and init as
0
230 Cls
240 Line(100, 100) - (250, 150), 1, B1
250 Font 2 : Print@(120, 120) "Speed %"
260 Line(300, 100) - (400, 150), 1, B2
270 Line(100, 250) - (250, 300), 1, B3
280 Font 2 : Print@(105, 270) "Turns Count"
290 Line(300, 250) - (400, 300), 1, B4
300 GoSub 810
310 SetPin 11,5
320 Duty = 30 ' init to 30/70 PWM
330 Sound 200, 2000000, Duty ' 200 Hz duration about 33 minutes 30 % duty
340 Font #2 ' big print
350 I$=Inkey$ : If I$="" Then GoTo 470 'Test for keyboard input until key
press
360 If I$="z" Then Duty = 20 : Print @(320, 120) " 20" : GoTo 440
370 If I$="x" Then Duty = 30 : Print @(320, 120) " 30" : GoTo 440
380 If I$="c" Then Duty = 40 : Print @(320, 120) " 40" : GoTo 440
390 If I$="v" Then Duty = 50 : Print @(320, 120) " 50" : GoTo 440
400 If I$="b" Then Duty = 60 : Print @(320, 120) " 60" : GoTo 440
410 If I$="n" Then Duty = 75 : Print @(320, 120) " 75" : GoTo 440
420 If I$="m" Then Duty = 100 : Print @(320, 120) "100" : GoTo 440
430 GoTo 350
440 Sound 200, 2000000, Duty
450 GoTo 340
460 'A=PIN(11)
470 If Pin(11)=A Then
480   GoTo 350
490 Else
500 ' As the calc Cint(dia*2/10) will usually result in a remainder,
510 ' here we accumulate the error and add an extra turn when the
520 ' accumulated error is greater than or equal to 1
```

```
530 Accum_Puls_Fr = Accum_Puls_Fr + Puls_Fr
540 If Accum_Puls_Fr >= 1 Then
550     Pulses = No_Puls + 1
560     Accum_Puls_Fr = Accum_Puls_Fr - 1
570 Else
580     Pulses = No_Puls
590 EndIf
600 For X=1 To Pulses
610     Pin(1)=1
620     Pause .2
630     Pin(1)=0
640     Pause .7
650     GoSub 720
660 Next x
670 A=Pin(11)
680 EndIf
690 Font 2
700 Print @(320,270)Int(A/10)
710 GoTo 350 ' go back to check for key press
720 '
730 puls_Cnt=Puls_Cnt+1
740 If Puls_Cnt>Max_Puls Then
750     Dir_stat=Not(Dir_stat)
760     Pin(2)=Dir_stat
770     GoSub 810
780     Puls_cnt=Puls_cnt-Max_puls
790 EndIf
800 Return
810 '
820 Print@(150, 375) " ";
830 If Dir_stat=0 Then Print "RIGHT ----->"; Else Print"<----- LEFT ";
840 Return
```

From:

<https://fruitoftheshed.com/wiki/> - FotS

Permanent link:

[https://fruitoftheshed.com/wiki/doku.php?id=mmbasic\\_original:coilwind](https://fruitoftheshed.com/wiki/doku.php?id=mmbasic_original:coilwind)

Last update: **2024/01/19 09:39**

