I built a coil winder

http://www.youtube.com/watch?v=sHzdH1Agz8g&list=UUicUEyjIwyyoGVPbiSGQnFw&index=19&feature=plcp

It worked well except for the difficulty of setting the wire size. It took a lot of fiddling to get it close even. Well why not put it under micro processor control, I thought. Shouldn't be that hard? So I bought an Arduino, lost cause.

Then I came across the Maximite. This then was the start of my learning curve. After a week and some help I had a blinking led. Another month and I had an operating stepper motor.

I needed a counter

How to count the turns? My first Maximite was an EV. I carefully removed the resistor from the sound pin, as recommended and made up a gate driver and fet from the junk box and without testing connected it up to the sound pin. Woops, the gate driver was blown and put 12 volts on the sound pin. This was not the thing to do as the sound pin no longer responds to commands. The other pins all work so it is not a throw away just yet.

So this prompted me to buy a CGMMStick1 and be more careful of my connections to the pins. I will use opto couplers from now on. I bought the CGVGAKBD1 also and will detail this later.

Time to get serious here I thought and started in on my study of PWM. Two kind souls from Australia came to my aid and before long we had a DC motor under control with the following code, not only that, but with the result in a box:

```
10 ' PWM1.bas using Sound output pin
11 '
15 CLS
20 LINE(100, 100) - (300, 150), 1, B ' draws a box
30 Duty = 30 ' init to 30/70 PWM
40 Sound 200, 2000000, Duty ' 200 Hz duration about 33 minutes 50 % duty
50 LOCATE 115, 120 ' where to print in box
55 FONT #2 ' big print
60 I$=Inkey$ : if I$="" then goto 60 ' Test for keyboard input until key press
70 If I$="z" Then Duty = 15 : Print"Speed  15 %" : GoTo 120
80 If I$="x" Then Duty = 25 : Print"Speed  25 %" : GoTo 120
90 If I$="c" Then Duty = 40 : Print"Speed  40 %" : GoTo 120
91 If I$="v" Then Duty = 50 : Print"Speed  50 %" : GoTo 120
92 If I$="b" Then Duty = 60 : Print"Speed  60 %" : GoTo 120
93 If I$="n" Then Duty = 75 : Print"Speed  75 %" : GoTo 120
```

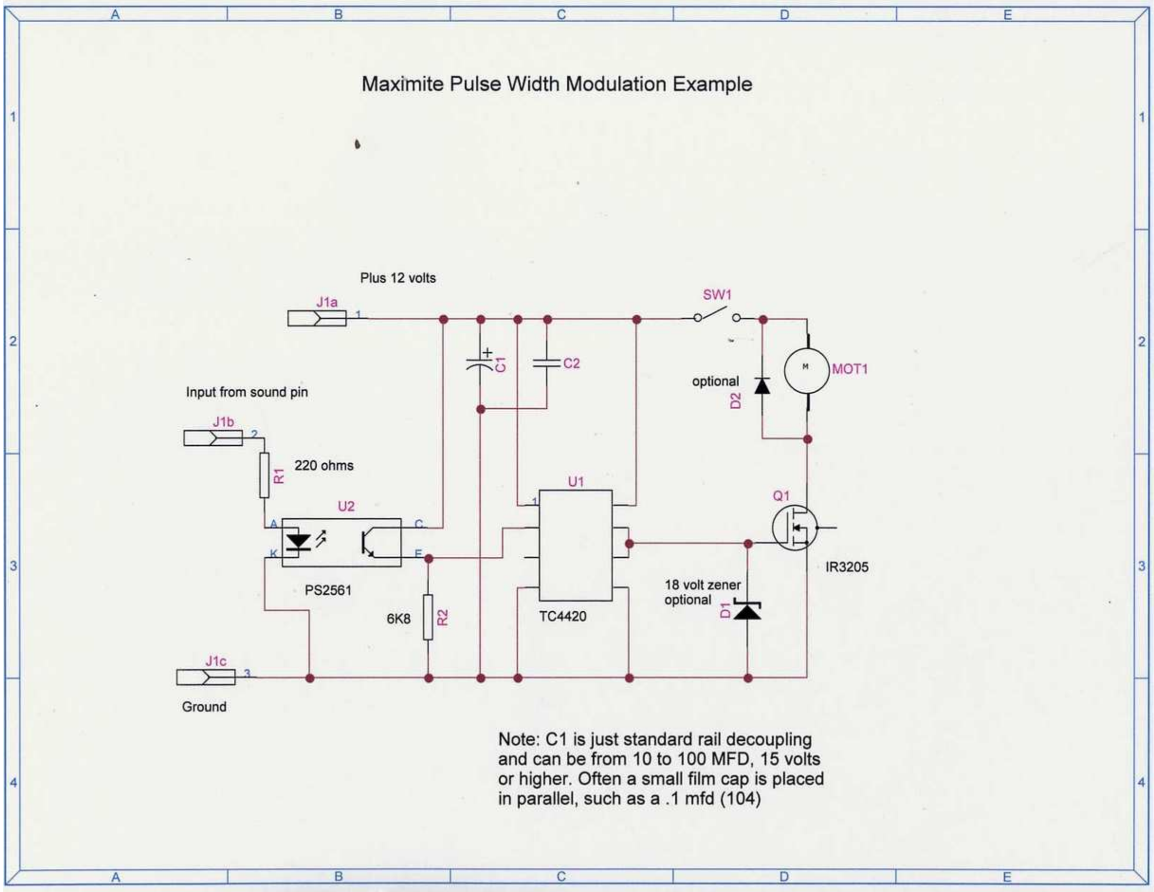
```
95 If I$="m" Then Duty = 100 : Print "Speed 100 %" : GoTo 120
100 GoTo 60
110 '
120 Sound 200, 2000000, Duty
130 GoTo 50
```

When I submitted this to the Library, Hugh suggested a few changes.
This is the recommended version, very simple:

```
' Demonstration of Pulse Width Modulation
' PWM1.bas using Sound output pin
' Hugh Buckle, Ron Pugh
Cls
Line(100, 100) - (300, 150), 1, B ' draws a box
Font #2 ' big print
        ' init to 30/70 PWM
Soundit 30 ' 200 Hz duration about 33 minutes 50 % duty

Do   ' loop for ever

  Do  ' wait for keyboard input
    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 @(115,120) "Speed";Duty;" % "
End Sub
```

To interface from the maxi to the motor I used the following circuit:



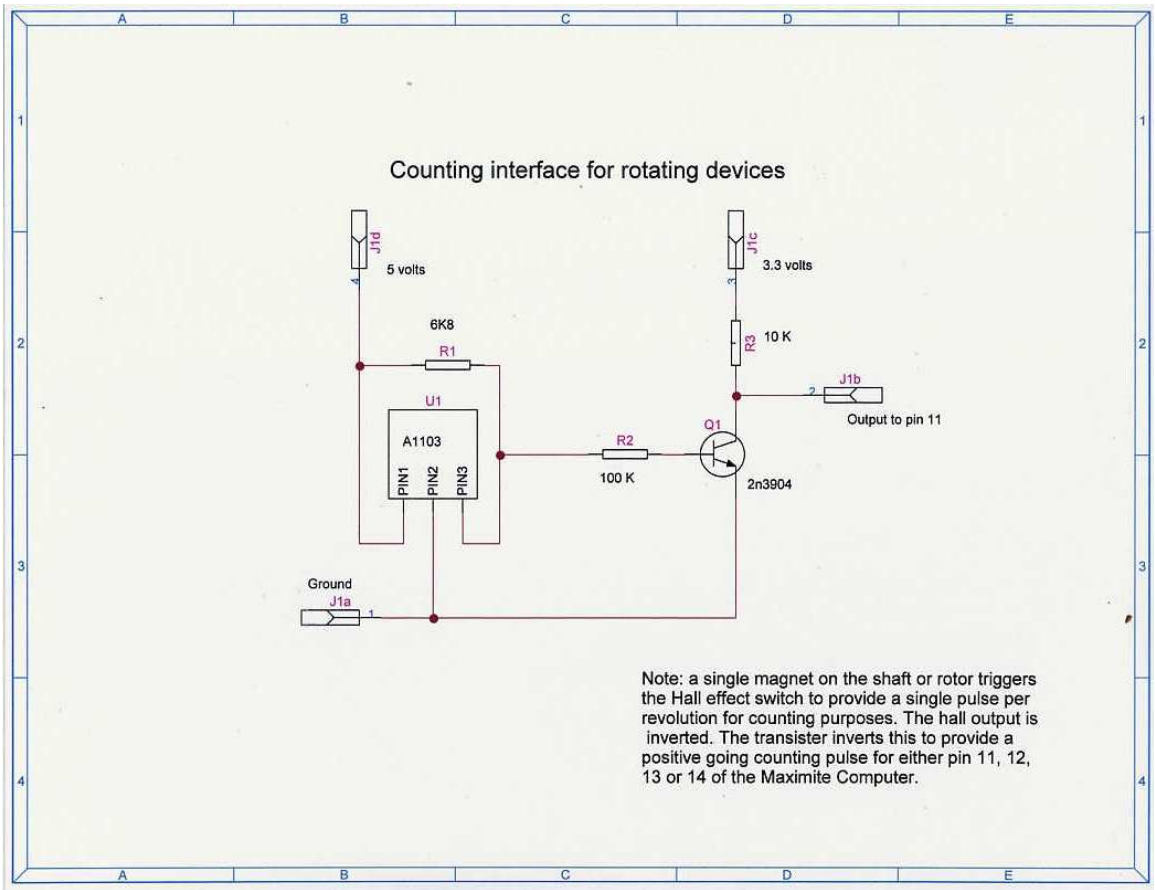
Maximite Pulse Width Modulation Example

Note: C1 is just standard rail decoupling and can be from 10 to 100 MFD, 15 volts or higher. Often a small film cap is placed in parallel, such as a .1 mfd (104)

With this under control I returned to the counting and we came up with this code:

```
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
```

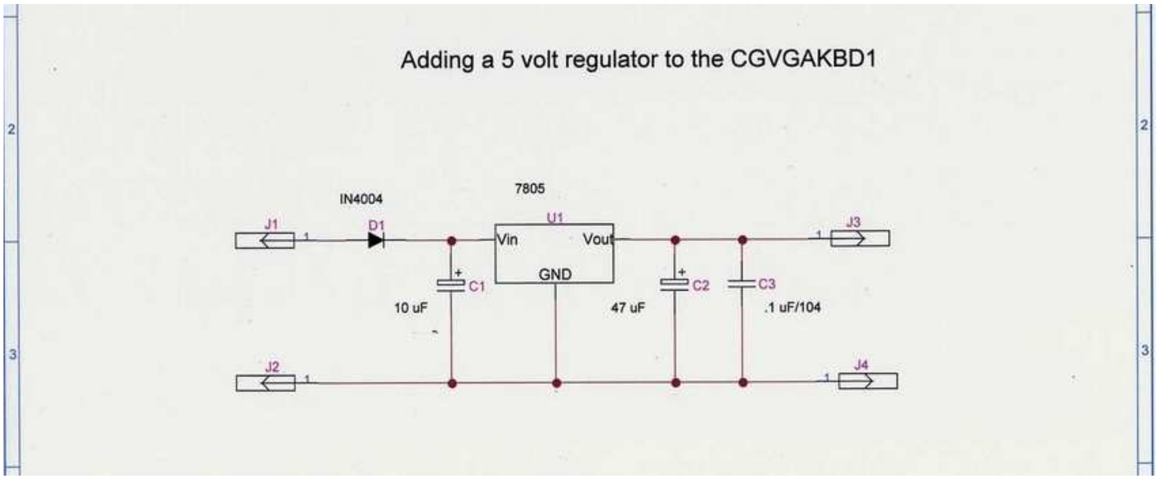
The circuit used to interface:



Counting interface for rotating devices

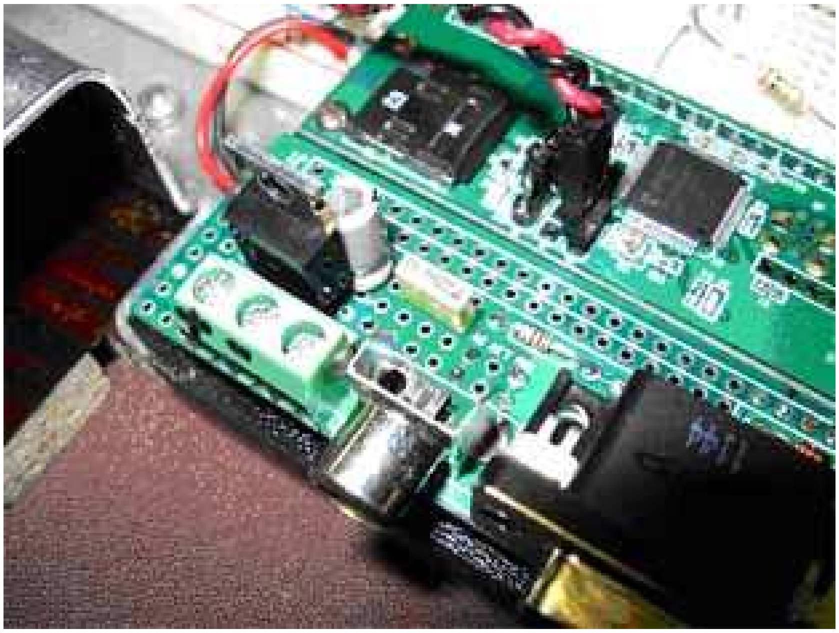
Note: a single magnet on the shaft or rotor triggers the Hall effect switch to provide a single pulse per revolution for counting purposes. The hall output is inverted. The transister inverts this to provide a positive going counting pulse for either pin 11, 12, 13 or 14 of the Maximite Computer.

Modding the CGMMStick: The CGMMStick1 runs just fine off of USB power. However to use the full features of the Maximite it needs the CGVGAKBD1. This is a 13 dollar extra which has the keyboard and VGA connectors, also a composite video connector and a prototyping area. But no 5 volt input or regulator. This is easy enough to add though. Just a standard circuit:



Adding a 5 volt regulator to the CGVGAKBD1

On the CGVGAKBD1 there are 6 pads immediately behind the Composite video connector labeled 3.3v, 5v, and ground. These are the connectors you want the output of the regulator to connect to. Forget the 3.3 volt connectors for now, this is the 3.3 volts coming back from the CGMMStick regulator,present also on pin 1.



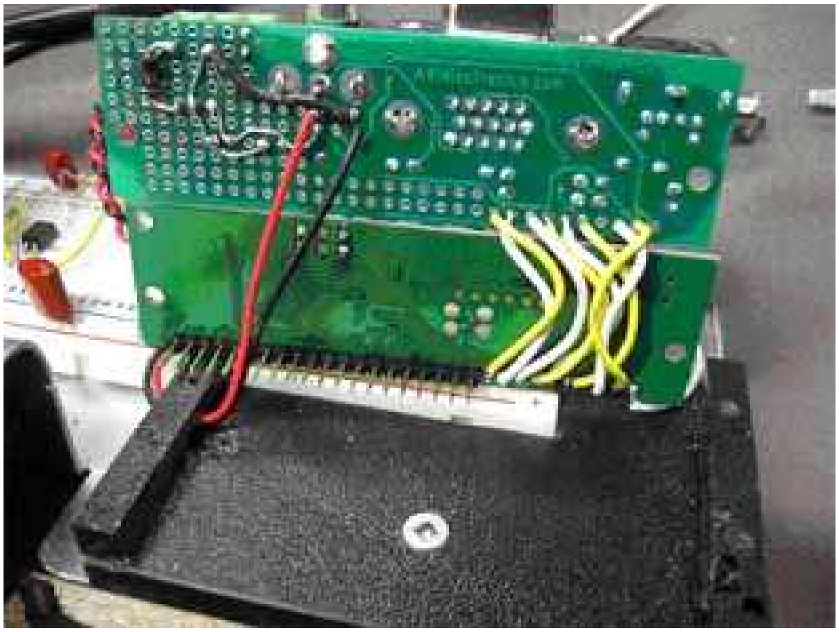
I have the regulator connected here and also two flying leads (red and black) from the 5 volt regulator out to the top rail on the bread board.

The 3.3 volts on pin 1, CGMMStick, goes to the first column on the bread board

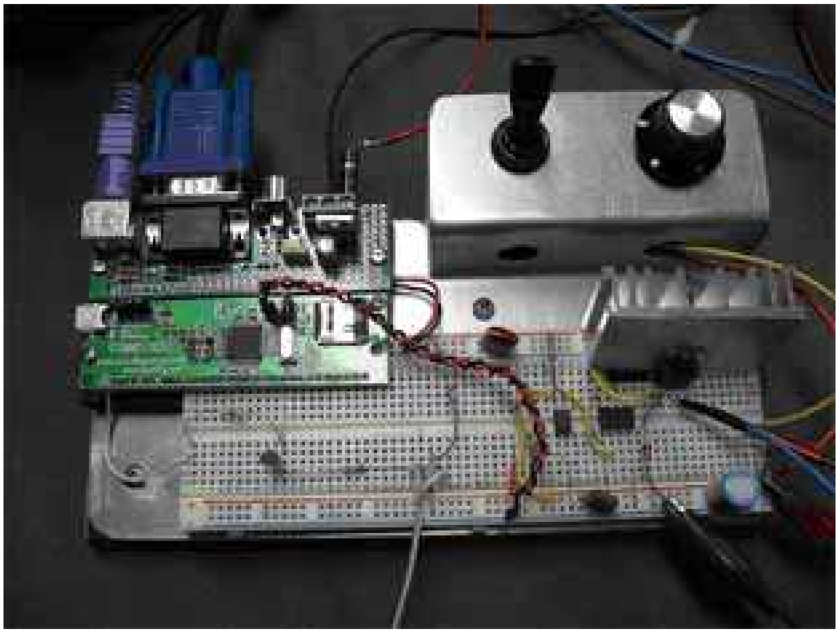
Pins 2 to 10 are connected to the corresponding 2 to 10

Only 11 to 30 (maximite 1 to 20) plug into the breadboard in this setup

I glued a couple of 6 mm by 6mm ABS onto a 6 mm base plate, which brought the 'Sticks' up to the bread board height
The messy hole on the right allow the board pins to sit down flush"



All up it looks like this:



With the ducks in a row it was now time to write code for four boxes and merge the two programs.

The video shows the results:

http://youtu.be/pjDlpbPddCA