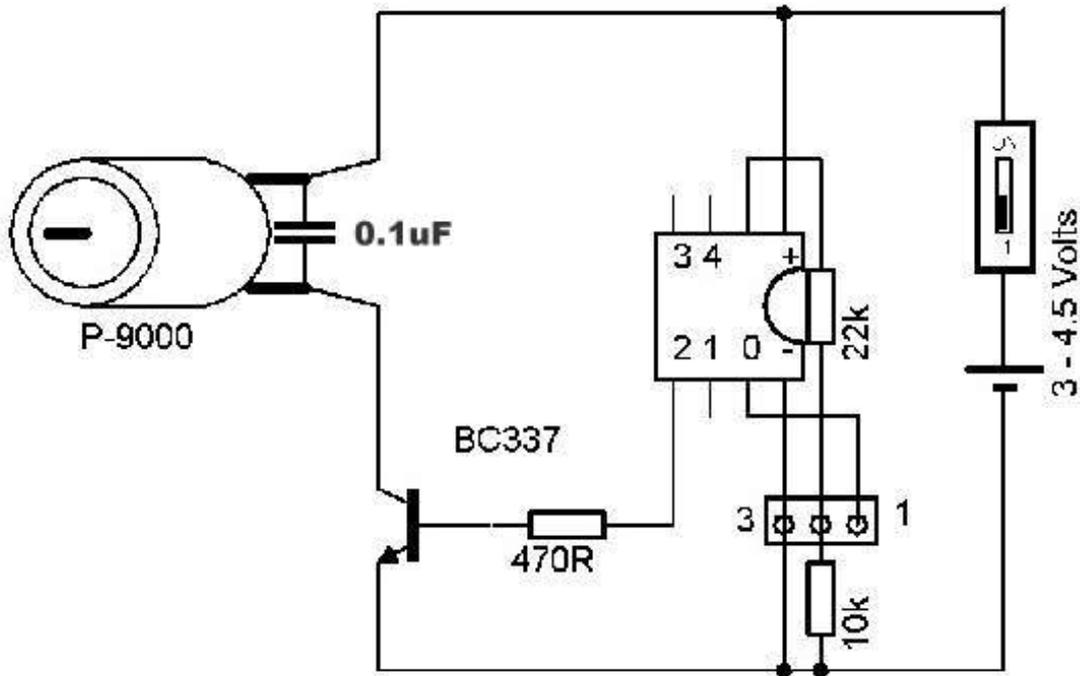


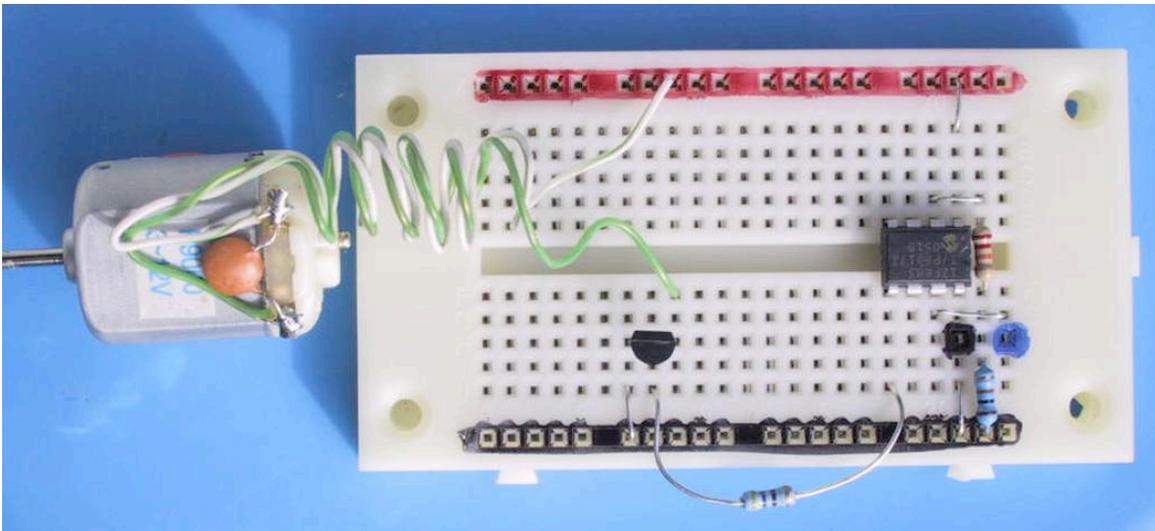
PWMOut Control



Loopa:

```

for b0 = 0 to 255 step 16 'For each value of b0 from 0 to 255 = 0 to 100% Power
pwmout 2, 64, b0 'PWMOut modulate pin 2 at freq time base of 64
debug b0 'Send debug value of b0 to the F6 debug screen
next b0 'Next power level
goto loopa
  
```



- **PWMOUT (continuous PWM) only works from pin 2. M2 chips have more options**
- **A Lamp or other load (peltier device e.g. for cooling) can be controlled by this circuit**
- **The 100nF ceramic capacitor soldered DIRECTLY to the motor brushes is ESSENTIAL and to prevent sparks and interference affecting and resetting the PICAXE**
- **PICAXE > Wizards > pwout is available to choose a time base feq**
- **Look up PWMOut in the Help ? Manual II section for more tips, info and help**
- **The pwmout command signal from pin 2 is continuous as the program loops. Once the pwmout command has been updated the power level remains set regardless of what the program loop is doing or up to until pwmout is updated**
- **A choice of 64 for the time base allows a 'power level' bn variable range of exactly 0 to 255 that gives a full range of 0 (fully OFF) to 100% (fully ON) power levels**
- **The programmer wizard can help choose alternative PWMOut time bases and ranges**

Hi Nigel
>>>We are going to have to slow the robot down as it moves ahead too quickly to be able to sense and then react to line changes using the PWM command and there will be an explanation of this as well.

Motors and PWM

+++++

Motors will only start to work at about a PWM level of > 150 so do not expect anything to happen below that. It is often best to start from 255 and ramp DOWN to about 100 when most motors will stop

PWM is different to PWMOut and there are always GOOD explanations and Examples in the programming reference that all students should be taught about and need to know how to use. It is there and to help you and them out. Help >> Manual II Basic Commands etc The F1 key gives you a very short helpful quick lookup but Manual II describes the command and how it works with each type of chip. Take care on reading there.. what type of chip ! The picaxe08M and the M2 behave quite differently and the M2 commands are generally a heck of a lot smarter.

PWM is the old 'privative' pwm method.

+++++

It does work in real actual programming time for short bursts while the command is executing so it can only work on any one OP pin at a time. It IS useful if used carefully. E.g. if two motors are being driven at full speed using a simple switch on or Hi command. If I wanted to decelerate one motor for a 'soft steer' I would give it a burst of PWM e.g. PWM 4, 200,100 This gives it 100 cycles (5ms per cycle) I.e. 500mS or half a second of reduced power. At the end of this however when the PWM finished the pin drops to a LOW state. So you need to set that pin back again to high speed / full on or just realise that that pin is now OFF so what are you going to do next. In short, you can only PWM one pin at a time and only for short durations

Ramping motor speed (of LED brightness etc)

+++++

Use a for next loop to gradually build up or decrease the motor speed

Speed_Up:

for b0 = 0 to 255

PWM 4, b0,10

next b0

wait 5

Slow_Down:

for b0 = 255 to 0 step -1

PWM 4, b0,10

next b0

PWMOut

+++++

This command is continuous and supersedes the PWM command. If you start pin 2 (only on an 08M) or any one other OP pin on a picaxe08M2 (still only one pin at a time) then the power level is adjusted and is maintained at a constant at all times. * The syntax has Nothing to do with the old PWM command

* If you use 64 as a time base then the power level is easily adjusted by a single byte e.g.

PWMOut 2,64,0 'Zero or no power

PWMOut 2,64,255 '100% power

PWMOut 2,64,10 'Low power

PWMOut 2,64,128 'Half power

* PWMOut power level is adjust instantaneously and the command takes zilch to execute.

* The power level is updated and there is NO hanging around. So to see anything happen you need to sticvk around for a bit hence the pause command

Speed_Up:

for b0 = 0 to 255

PWMOut 2, 64,b0

pause 10

next b0

I strongly recommend taking a step sideways and playing with this command as it takes a bit to get the feeling of motor mechanics and effects of frictions, inertia and just program cycle times as well as just having a chance to get it working by itself first.

Enjoy as they say

Perhaps try these word templates from the booster course on this command. They still do NOT replace reading the manual !

Good work and nice if you can get the PWM OR PWMOut command incorporated into the steering ! Both these commands are fun. I use the PWM command in LED artworks a lot and leave the OP either full on with a high command after ramping UP or Low after raming down etc. That way it can look like there are three PWM going on at the same time !

~ Andrew

You might need more than 2 minutes to get your head around this but....

What in detail is the speed adjustment for ?

* To fine tune handling (slow on corners etc or steering ?)

* To adjust the overall speed up and down ?

* Increase the performance of say a line follower so it gradually steers back to the line rather than jerky ON:OFF wobbling control ?

>>>In essence can I only run 1 motor at a time on pwmout (and which pins will it run on?)

You can use this on any one pin at any one time. Same goes for PWM

>>>Is there a way to make 2 motors slow down at the same time

Is this to 'decelerate' I.e. slow down over a fixed period of time or is it to make the motors slower ALL the time ?

You could start PWMOut and then use PWM on the other pin at the same time if this was the case

PWMOut 1, 64,200

PWM 4, 200, 100

PWMOut 1, 64,0

This would give both motors a power level of 200 for the duration of the PWM command (5ms x 100)

If you want long term / general speed control signal to slow things down then PWMOut could control one BC337 transistor that modulates the common power on the emitter of the other two transistors. Connect the collector of the one PWM transistor to the emitters of the two motor control transistors. This will act as a common 'chopper / pwm system' that sets a power level for one or the other or both the motor control transistors

Another note with PWM is that it creates (as far as the inertia and momentum of the motor armature is concerned) a voltage regulated supply to control the speed of the motor. The average voltage = the voltage coming from a regulated DC power supply. Bottom line: you get Very Good motor speed regulation with PWM. This because the RPM of a motor is a fixed ratio of the (average) DC voltage applied. The motor tries to match the back emf with the applied DC voltage at all times. Explains why current and power goes up when load comes on. Under revs the motor draws more current until the motor speed catches up. PWM does not limit current. It provides an average DC voltage to the (highly inductive) motor armature.

If you use a resistor to try to 'throttle' the motor speed then things get very bad as the voltage drop is proportional to the load (i.e. current) through the motor. This = boggy starting and overspeed under light load conditions. It is like driving a car with a ballon between your foot and the pedal... Also... Any variation in friction between the two motors, wheels, shafts, gearboxes, motor efficiency will be amplified by this effect. Resistors I.e. are bad news

If you want a fixed constant different speed then Tamiya gearboxes come in 4 different ratios. Use a better motor ? (P-9000) are the best all round unless you have a high power system. Generally the motors that come in kits draw way too much current and squeal and thrash and over rev and flatten batteries in minutes !!! P-9000 last all day by comparison.

~ Andrew