

Controlling The Real World With Computers

::: Control And Embedded Systems :::

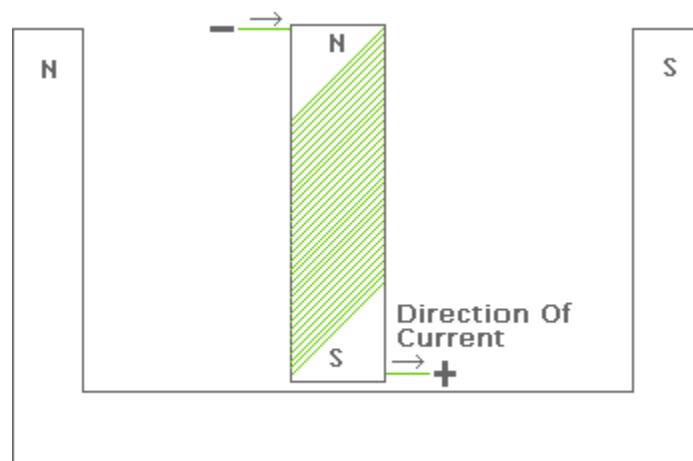
[Table Of Contents](#)

[Previous: Programming Experiment 1](#) [Next: A High Power Driver For Motors And Lights](#)

Putting It All Together - Controlling The Hardware With The Software Programming Experiment 2

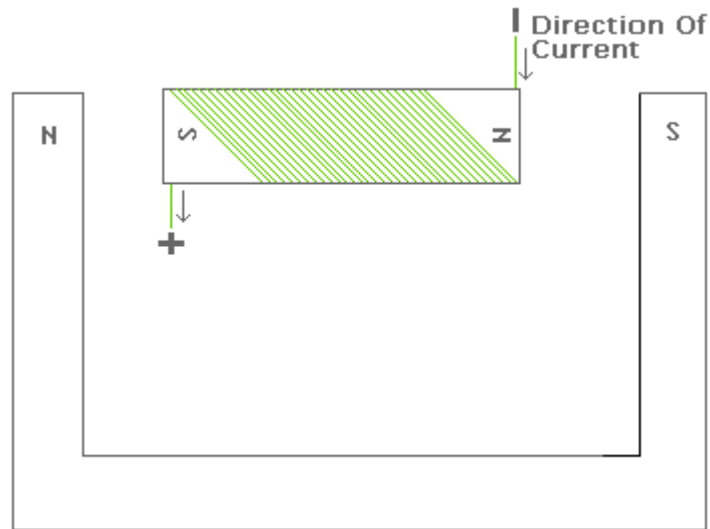
Just about everybody played with magnets in grade school. From that experience it was discovered that opposite poles of magnets attract and like poles repel. We were told later that an electromagnet could be formed by running electrical current through a wire wrapped around a piece of iron. By using a direct current, such as that from a battery or a DC power supply powered from the wall socket, the piece of iron will have fixed North and South poles just like a permanent magnet. It will be attracted to and repelled from a permanent magnet in the same manner as another permanent magnet. A motor can be constructed using that knowledge.

I will cover only the basic ideas of motor operation here. Notice what will happen when an electromagnet is placed inside a permanent magnet (wrap the fingers of your left hand around the electromagnet in the direction of the current and your thumb will point to North). The electromagnet is mounted on an imaginary shaft to allow it to rotate:

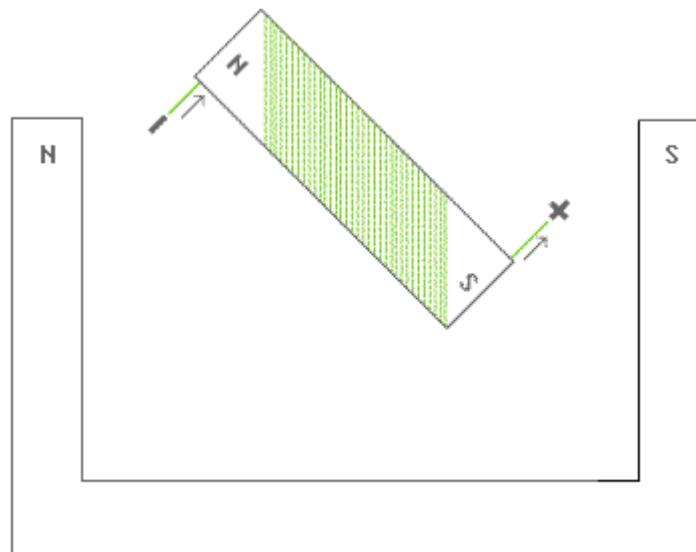


2

The opposite poles will attract, and the like poles will repel, so the electromagnet will rotate:



Now turn off the power long enough to let the electromagnet coast a little, then turn it back on, but with the polarity reversed. The whole process starts over:

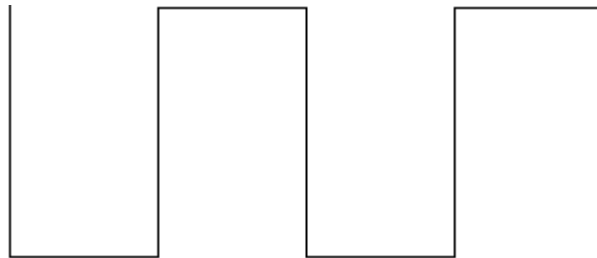


The rotating electromagnet is called the armature. The switch used to reverse polarity is formed by brushes which connect to the coil by sliding over curved pieces of metal. Together the curved pieces of metal are called the commutator. They are bonded to an insulator on the shaft going through the armature.

In addition to producing mechanical movement, magnetism can also produce voltage. If a wire moves at right angles through a magnetic field or a magnetic field moves across a wire, a voltage will be produced in the wire. The magnitude of the voltage depends on the velocity of the wire or field, among other things. A magnet is formed when voltage is applied to the motor. A magnetic field exists around the armature just as it would around a permanent magnet. When the voltage is turned off, the magnetic field rapidly collapses. Since the field collapses in the opposite direction from which it was initiated, the voltage is reversed. It can be quite high compared to the powering voltage; enough to damage components connected to it.

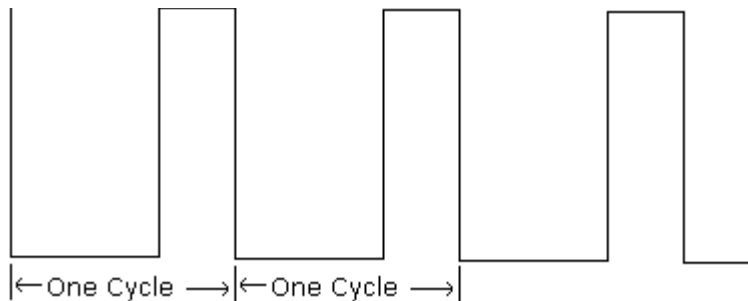
Back to Pulse Width Modulation (PWM) a little, since it's so important in, for example, the control of motors for robotics. Let's go back to childhood for a little while. Did you ever turn a bicycle upside-down and spin its wheels with your hand?

The more slaps per unit of time, the faster it turns. Also, the longer the hand is in contact with the tire, the faster it turns. This is how Pulse Width Modulation works. Pulsing the motor with the full voltage and current produces the same magnetic force in the armature, but for a shorter time. Thus, the force provided by the motor is about the same. Hit the tire or pulse the motor only half the time however, and the average voltage is cut in half. The motor will turn at half its full-voltage speed but still supply the same force:



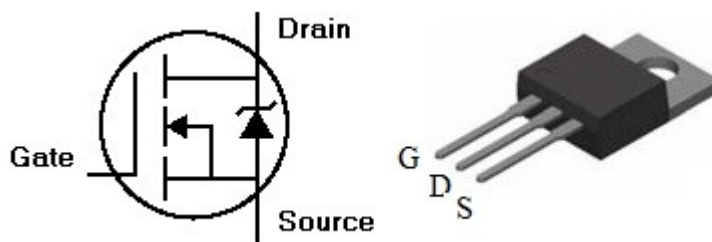
The motor will run at half its normal speed, providing the pulses are fast enough to take advantage of the motor's moving inertia. If you try to slow the shaft down with your fingers however, you will find that the motor has about the same rotational force (torque) as it does at full speed.

Here's another one. If we use a transistor to provide the increased current needed for a motor, and it's on for 30% of a cycle, the duty cycle is 30% and the motor will run at about 30% of its full speed:



The little UNO works fine with LEDs, but it is not capable of running a motor. The UNO can supply a maximum of 40ma at its outputs. Since the LED needs only about 20ma, that's not a problem. But since a motor can require a thousand times that much or more, we need a little help.

A MOSFET (Metal Oxide Semiconductor Field Effect Transistor) will provide the extra power for us in this experiment. The MOSFET comes in many types and configurations. We will be using an Enhancement mode N-Channel device here. A schematic representation and a picture showing the pin designations is below. The case, a TO-220 package, is illustrated. The Drain is also connected to the metal tab, making it necessary to insulate the MOSFET when using a heatsink, a metal device used to dissipate heat from the transistor:



An **enhancement mode** N channel MOSFET is turned on by making the gate more positive than the source. What is enhanced is the conductivity between source and drain. The unit of conductivity is the **mho**, which is simply ohm spelled backwards. The IRFZ48N used here will turn on if the gate is about 2 volts more positive than the source. The gate is not physically connected to the rest of the device. It influences the device by the fact that it is on one side of a very thin insulator, and works by means of the imposed field effect. In fact, the insulator is so thin that it can be ruptured by static voltages as weak as that from a piece of plastic. **Great care** should be taken when working with a MOSFET. The leads are sometimes shorted with a piece of wire or conductive foam when shipped. The shorting material should not be removed until the last moment when installing the device. Also, make sure you are grounded or have at least touched ground before touching a MOSFET.

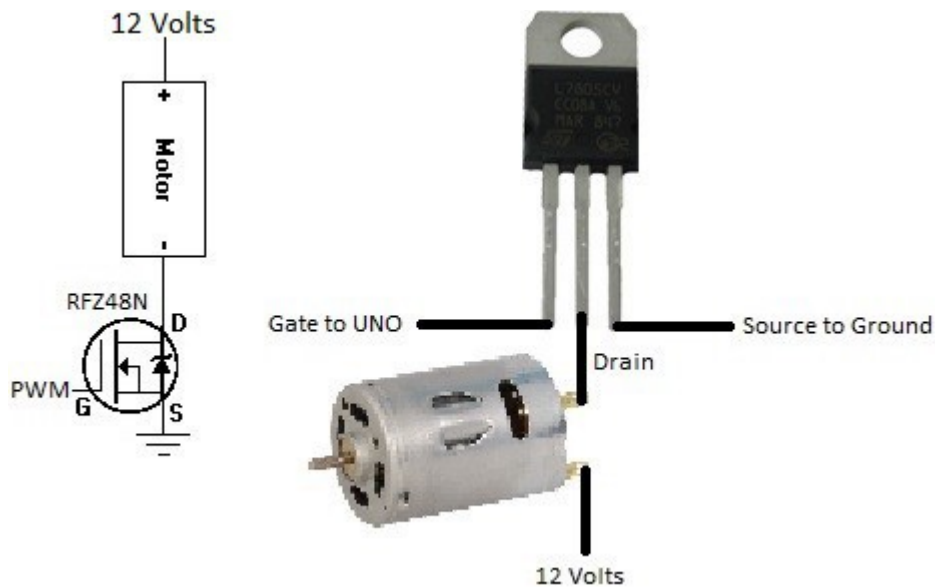
6

The formula for power is $P = IV$, with P expressed in watts. The resistance from Source to Drain of the [IRFZ48N](#), as an example, is very high when it is off, but less than one ohm with a gate to source voltage of 4 volts. Recall from [How To Read A Schematic](#) that $V = I * R$. That means we can replace the V in the power equation above with $I * R$ since the two are equal. Thus, $P = I * I * R$.

Let's say the motor needs 1 amp. So we get:

$P = 1 \text{ amp} * 1 \text{ amp} * 1 \text{ ohm} = 1 \text{ watt}$, which is very little power.

PWM from the UNO can be used to control a motor:



ColorChanger was used as a base to build software for the UNO that will permit changing the motor speed with a potentiometer. Wire a 12 volt motor as shown above, with the gate connected to 9~, then load [BasicMotorControl here](#) and run it to see what happens. It's a good practice to study the specifications for the transistor and not let the power (see above) go beyond about 1/4th of the transistor's capability without a heatsink.

The 12 volts can be obtained from a single 12 volt battery, a power supply or by wiring four 3 volt battery holders in series.

Notice the following in the program:

```
for(; MotorAnalogValue > 100; MotorAnalogValue--)  
{  
    // XOR not needed for the motor since the MOSFET inverts for us  
    analogWrite(Motor_PWM_Pin_9, MotorAnalogValue);  
    delay(10);  
}
```

This is the second for loop in the program. Notice that there is no initialization of MotorAnalogValue in the second for loop. It's not needed since it was declared in the first for loop.

The problem with the circuit is that it doesn't drive the MOSFET with enough voltage. That takes a driver, which we will cover later.

[Table Of Contents](#)

[Previous: Programming Experiment 1](#) [Product: A High Power Driver For Motors And Lights](#)

Other files; some used in this tutorial and some elsewhere, mainly experiments with The Listening Tree and The Listening Car:

[BasicMotorControl.c](#)

[BrightTestForListeningCar.c](#)

[Chase2.c](#)

[Chase4.c](#)

[Chase6.c](#)

[exmpl1.c](#)

[MultipleForLoop.c](#)

[ThreeSweep.c](#)

[BrightTest.c](#)

[Chase1.c](#)

[Chase3.c](#)

[Chase5.c](#)

[ColorChanger.c](#)

[MultiFadeUpDown.c](#)

[StrobeSlowFast.c](#)