

A technical explanation of Electronic Speed Controllers (ESCs)

This is an attempt to explain how ESCs work rather than describe what they do.

As you saw in article three on brushless motors, all electric motors rely on one magnet pulling or pushing on another so the motor shaft goes round. The motors we use in model aircraft have two types of magnet. One is a permanent magnet which can move round. The other is a coil of wire that is fixed. When you make an electric current go through a coil of wire it makes it into a magnet. Our motors have fixed coils that can be switched on to make them into magnets and then off again so they stop.

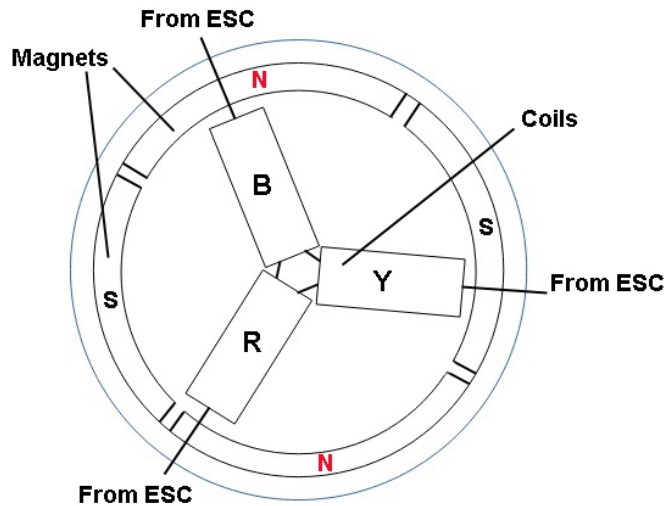


Figure 1

The coils are fixed so are called the stator (static). In figure 1 they are labelled R (red), B (black) and Y (yellow). These are often the colours of the wires on the motor and ESC.

Look at the circular bit labelled **NSNS**. This is the outer case of the motor onto which permanent magnets are fixed. The N and S are the poles that face inwards. They are made using a rare earth metal, usually neodymium (Nd). The case rotates and drives the propeller so it is called the 'rotor' (turning). If the red R coil is switched on so it becomes a south pole it will pull the magnet labelled N (north) round to line up with it.

If at the same time the black coil B is made a north pole it will push the nearby N magnet.

The rotor has started to turn. If B stays north it will pull the south magnet. If Y is made south it will pull the north magnet.

Why not use all three coils? The ESC needs to know the position of the rotor. The turning rotor creates (induces) a voltage in the now unpowered R coil and the ESC can read it to work out the position and speed.

The ESC sends constantly changing currents through pair of coils. It moves on from one pair to the next. The effect is that the magnetic field effectively rotates and drags and pushes the permanent magnets with it. No electric current has to be fed to the rotor through brushes. This makes our motors simpler and more reliable and is why they are called 'brushless'. Remember all those sparks you used to get from brushed motors? Our

motors use tens of amps or even over a hundred. You can imagine the sparks you would get.

How is all this switching done?

We have to use a device called an Electronic Speed Controller (ESC). These have electronic switches called Field Effect Transistors (FETs) that can switch on and off in very short times and a circuit board that makes them switch.

Have a look at the following circuit diagram Figure 2, even if you are not used to reading them. On the left are the sets of U, V and W coils. These are the standard letters used for the coils I labelled red, black and yellow. On the right are the FET switches in the Power Circuit. They are labelled Tr1 to Tr6 because they are transistors.

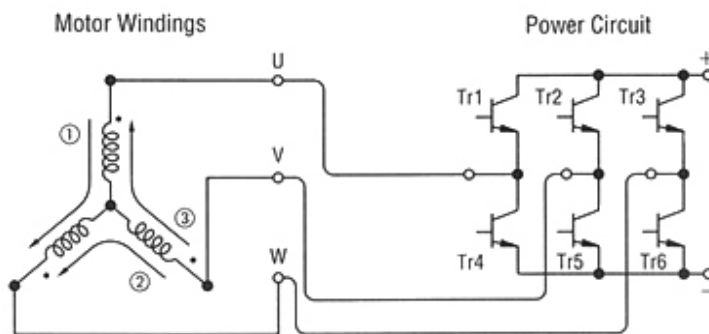


Figure 2

Figure 3 shows one of the transistors. As you see there is some variation in the symbols used. In Figure 3 I have drawn the most common one.

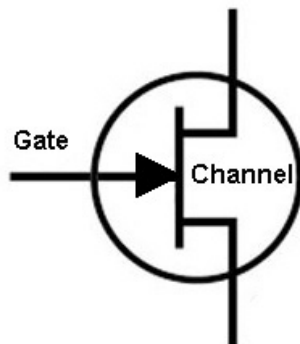


Figure 3

Electric current can travel through the transistor channel from top to bottom but only when the correct voltage is connected to the small spike sticking out to the left, which is called a 'gate'. The electric field that the voltage creates allows the current to flow or not. That is why it is called a Field Effect Transistor (FET). It acts like the switch you use on your house lights and indeed FETs are used in light dimmers. They switch the bulbs on and off rapidly with different amounts of on and off, so changing the brightness.

Why are there six FETs, not three? After all there are only three coils to switch on. Look at the connection to coil U. If TR1 switches on, the coil is connected to the plus +. If TR4 is

on, U connects to the minus -. So depending on which is switched the coil can be made north or south.

Step	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬
Tr1	ON					ON	ON					ON	ON
Tr2		ON	ON					ON	ON				
Tr3				ON	ON					ON	ON		
Tr4			ON	ON					ON	ON			
Tr5					ON	ON					ON	ON	
Tr6	ON	ON					ON	ON					ON
Phase U	N	—	S	S	—	N	N	—	S	S	—	N	N
Phase V	—	N	N	—	S	S	—	N	N	—	S	S	—
Phase W	S	S	—	N	N	—	S	S	—	N	N	—	S

Table 1

Figures 2 and 4 and Table 1 are from Orientalmotor.com

If you have followed this so far you now know how the motors work. As the coils magnetise in turn they produce a magnetic field that turns round continuously pulling the rotor with it. Almost all electric motors used in cars and industrial machines work like this. They are also called ‘synchronous’ motors.

Choosing an ESC

The compactness, rotational speed and power of brushless motors is impressive. When I started electric flying it took me a while to realise that they were small versions of the three-phase motors I had come across in industry. I was puzzled where the driving waves came from, so that sparked my interest in electronic speed controllers. You now know that they produce the three phases at large currents, but they can be programmed for braking the motor, reversing the motor, throttle response speed and other things. You can even change the sounds they make.

My favourite simple ESCs are Turnigy Plush. They are quite cheap, compact and very robust and they can be programmed using a cheap card rather than struggling with beeps from a transmitter. Usually the defaults are fine.

My favourite ESCs of all are the FrSky Neuron range, so-called no doubt because they are ‘brainy’. They are available in 40, 60 and 80 A versions with a +50% burst capacity. I use the word brainy because they provide a range of telemetry data for FrSky radio equipment, including rpm, temperature, current and lipo voltage and mAh used and from which you can also derive power. They produce up to 7 A for the receiver and servos and as a final bonus they are small. The only downside is that they are expensive, starting at around £45 (\$60).

How do you choose an ESC?

There are several factors

- Current capacity. It is best to choose one that can produce a current at least 20% more than the maximum for the motor.
- Number of cells in the lipo. Most ESCs can handle lipos of between two and six cells. For larger numbers of cells, for example when you use two lipos in series, you need to find a suitable ESC, which is likely to be expensive.

- Whether the ESC can produce the voltage for the receiver and servos. This part of the ESC is called a Battery Equivalent Circuit (BEC). Make sure it can pass enough current for the receiver and servos. If not, fit a separate receiver battery or find out about power boxes. I will cover them in article eight on servos.
- Size.
- Weight.
- Whether it will switch off power to the motor if the lipo voltage drops low. This ensures that it can still supply power to the receiver and servos so you can safely land deadstick.
- Whether there is a programming card available. (Yes for Plush. No for Neurons)

The more I learn about ESCs the more clever I think they are, or at least the people who design them. I like to know how things work. By that I don't mean what they do but the underlying principles. These will be a combination of science, engineering and computer code.

Let us suppose that the throttle lever is at maximum. The ESC starts to energise the coils and the motor turns. What determines the speed at which it turns? For a given voltage and motor there is a speed determined by the propellor. If the propellor match is good the motor will turn at roughly its kV rating multiplied by the voltage. For a 3S battery of 12.6 volts and a motor of 1000 kV this will be 12600 rpm. In practice it will be lower as the battery voltage will be lower than its maximum value.

So how do they work?

We have covered the physical arrangement. This does not explain how the ESC sends the correct pulse of current to the correct coil at the correct moment. This diagram shows the control loop. Note that there are boxes for detection of position and speed. The speed setting is the signal from the throttle stick.

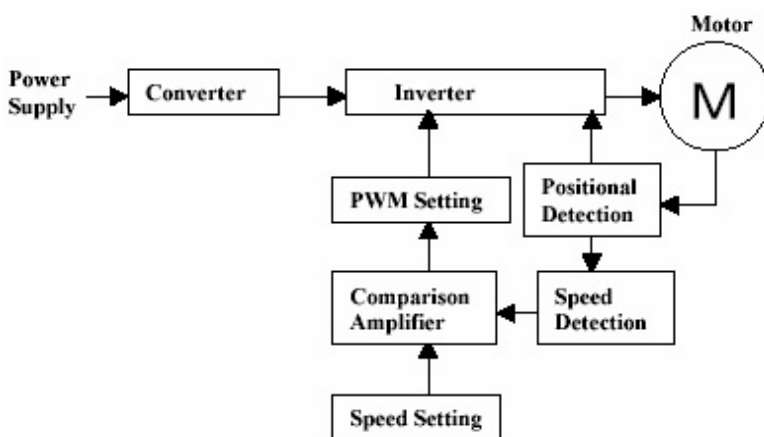


Figure 4

The Inverter produces and switches the power to the coils on and off as decided by the PWM (Throttle: Pulse Width Modulation) Setting. This varies the power as you will see later.

Detecting position and speed

As we saw earlier the ESC powers two coils at a time. That leaves the third coil and wire unused. To see how it is used we need to understand a scientific principle called induction.

Coils are made of windings of wires. If a wire moves through a magnetic field, or a stationary wire is in a varying field, a voltage is created in the wire. The rate at which the magnetic field 'lines' are cut by the wire decides the voltage. As a rotor magnet passes the third coil it creates a voltage in it by induction. The ESC circuitry notes this voltage by reading the third wire and uses it to calculate the rotor position and hence how fast it is turning. It uses this data to vary the speed at which the coils are switched.

Throttle positions less than maximum

To send reduced power the FETs that drive the coils are switched on and off during each power pulse, called Pulse Width Modulation (PWM) or the 'duty cycle'. At full throttle both of the coils are on all or most of the time. They are most efficient at full throttle as each switching operation causes a small energy loss.

Some ESCs have to be calibrated the first time they are used

Many ESCs do not need this. If yours does, this is how to do it when connecting up for the first time or when changing transmitters.

- Ideally remove the propellor or at least tie the model down.
- Switch your transmitter on and set the throttle stick to maximum position.
- Power up your receiver.
- Connect the battery pack to the ESC.
- In a short while the ESC will beep.
- Then pull the throttle stick to the zero power position.
- The ESC will beep again, which indicates that your ESC has got the signal range of the throttle from your transmitter.
- The throttle is now calibrated and your ESC is ready for operation.

What programs do they run?

Hard as I looked I could not find any information on how the ESC reads and processes the data. This is not surprising. Companies making ESCs will be jealous of the commercial value of the program code. This does mean that some of what I say is speculation from first scientific principles.

So I thought it might be an idea to work out how I would set about writing the program code for an ESC and it might help you understand how they work. I might not get it right of course and I hope someone will correct me if I don't. I ain't proud! Or maybe my method is correct but there are other ways of doing it.

We start with the data. Of the three wires to the motor, two will be carrying current depending on throttle setting. What will we 'see' on the third wire? As it is pushed at speed past a magnet, a pulse of voltage will be induced in the coil to which it is connected. The height of the pulse will tell us the speed and the centre of the pulse tells us the position.

How does that work? Now the science bit.

The strength of a magnetic field is called flux density Φ (phi - 'fi'). When you played with iron filings and magnets they were tighter packed where Φ was greatest, near to the magnet. When a wire moves through magnetic flux a voltage is created (induced) in it. This works with a fixed magnet and a moving wire, a moving magnet and a fixed wire or

where wire and magnet are both moving. Just for completeness, though not relevant here, a fixed wire in a magnetic field that is changing, such as a field made by a coil of wire with a varying current, will also have a voltage induced in it. That is how transformers work.

So induced voltage = rate of change of the magnetic flux linked with the wire.
Physicists write that $d\Phi/dt$ pronounced 'defy by detee'.
So the induced voltage tells us the speed of the coils of wire and hence rotor.

However we can also find the speed from the time between this pulse and the one from when the next set of coils is switched on. We could use either, or perhaps both. The midpoint of the pulse will show when the magnet and coil are exactly lined up, so will show us the position of the magnet.

All computers (and an ESC is one) have a 'timebase', which is effectively a ticking clock that keeps everything in step.

A first try at the program code

Pseudo code is English-like text that describe how the program works. Here is my pseudo code that can be converted into any computer language. Note that it produces speed data in the two different ways. This code is for a situation where brake is on at zero throttle as it would be for folding glider props.

Repeat the following

 If throttle is more than zero

 Then do the following lines of code (start loop)

 Move to the next set of coils

 Identify the third wire by checking which has no signal going to it

 Read its voltage

 Read the clock time as a voltage appears above a threshold

 Sample the voltage repeatedly and rapidly and record values

 Read the clock time when it drops to the threshold value again

 Calculate the mid-point time

 Select the maximum value for the voltage

 Convert the maximum voltage into a speed value

 Convert the mid-point time into an angular position value

 Calculate a suitable delay

 Calculate the lengths of the drive current signals for wires 1 and 2 from throttle setting

 After the delay send the drive signals to the FETs for the relevant coils

 Calculate the speed from the current and last angular positions and the times

 Compare the mid-point time with the mid-point time of the drive signals

 If it is lagging or leading change the delay timing for the drive signals

 End loop

 Else (i.e. throttle zero) short out all coils to put brake on

 Read the throttle setting again ready for next loop

End of repeat (i.e. jump back to the start)

In a Neuron ESC there will be additional line of code for sending rpm, time, current and lipo voltage from other internal sensors, as telemetry data to the receiver.

In a twelve coil motor the above code would be repeated twelve times for each revolution. If the motor was turning at 10 000 rpm that would mean 120 000 loops each second. Is that even possible? To see why it is, you need to understand two things – machine code and clock speed.

First - machine code

Computer programmers (coders) usually write the programs in a language that is a bit like English. For example
`println("Hello");`

That makes it easier to write and understand the program. Then a program, called a compiler, reads each line and checks that it makes sense, a process called 'parsing'. If it does then it turns it into machine code instructions. These are compact and efficient being just a string of binary digits (1 or 0). It stores them all in a file. You can spot these files on a computer as they usually have the file extension .exe, standing for executable code. They are what a computer - the ESC in this case - runs as a program (application).

In the case of the ESC, and similar things like receivers, phones and washing machines, the machine code is stored permanently in a chip inside, called firmware, so does not have to be loaded and started like a normal application. It is usually erasable so the programs can be changed by updates. Firmware binary code does not have a file name. It starts automatically as soon as the device is switched on.

Some programmers write directly in machine code. I have occasionally. A good programmer might write machine code better than a compiler can, but it is a lengthy and brain-boiling process. Early computers could not even start themselves up. The startup code had to be put in line by line by setting a row of toggle switches to one or zero and then pressing a button. This was called 'toggling it in' and allowed the computer to start, in effect by pulling itself up by its own bootstraps. Hence the term that we still use for starting up, to 'boot' the computer. It isn't the urge to apply a fast and heavy foot to it even though we often feel that way. We keep that for vending machines.

Clock

The clock in a computer processor ticks at a speed that we find difficult to imagine. Even the simple Arduino runs at 8 or 16 MHz and it's possible that those in ESCs run faster than that. There are roughly 30 million seconds in a year. At 16 MHz a processor divides a second as a second divides six months. On average a machine code instruction takes two machine clock cycles (ticks). That's plenty of room for lots of instructions that each take a few microseconds to run. Incidentally the speeds at which modern computers work are measured in gigahertz (GHz or a thousand million ticks per second). For example a 3 GHz computer processor divides a second as a second divides 100 years and most computers have several processors sharing the work.

Is that it?

Er, no. ESCs can do all kinds of clever things such as braking the motor on zero throttle by shorting the ESC wires, reversing the motor for getting out of reeds when flying on lakes (or playing silly beggars in the air when flying a Durafly Tundra), changing the pulse timings for different type of motor, deciding what noises the ESC makes, calibrating for maximum and minimum throttle values, changing how rapidly the motor speed changes as

the throttle changes, and so on. This would need another article. You have suffered enough for now.

Useful sources for this article

<https://www.orientalmotor.com/brushless-dc-motors-gear-motors/technology/speed-control-methods-of-speed-control-motors.html>

<https://www.modelaviation.com/inside-esc>

Peter Scott © 2021

Last edit 27 February 2022