How to Flash SimonK to ESCs

Introduction

Flashing your ESC involves some needed software & hardware. Also some knowledge is needed about the connections and specific settings. This page intends to cover the necessary hard & software and the flashing process.

Requirements

Hardware

Programming device.
You will need an additional programmer. The programmer will be connected to the specified points on the ESC and communicates with the computer. There are several versions of these programmers from different brands and in different price ranges. You should be able to find an adapter around 15$.

Commonly used USBasp adapters can be found for example here:
HobbyKing: *USBasp AVR*

Additionally, you need a way to connect the programmer to the ESC. This can be done with pins or one can just solder the wires onto the ESC.

Programming Device for ATMEL processors
GoodLuckBuy: *USB-ASP Atmel ISP Programmer Download Adapter*
ProtoStack: *USBASP AVR Programmer*
CuteDigi: AVRISP STK500 programmer

Drivers:

On Linux and MacOS X no kernel driver is needed. Windows requires a driver for USBasp:

USB Windows BASP driver. usbasp-windriver (Driver source: Fischl.de)

Interface Software

Depending on your platform you will need dedicated interface software installed on your computer to flash your ESC with the new firmware.

Interface software:

ESC Flash Tool - This interface program uses the tools from AVRDUDE and WINAVR. It automatically downloads the latest available SimonK firmware version for the ESC. (ESC Flash Tool requires Microsoft .NET framework to run.)

KKmulticopter Flash Tool - This tool was initially written to update the KKmulticopter firmware. It is, however, perfectly suited to flash ESCs. All information and manuals can be found on the author's web page. Windows, Mac and OS X and Linux are supported in different languages.

AVRDUDE - AVRDUDE is software for programming Atmel AVR Micro controllers.
eXtremeBurner - GUI Software for USBasp based USB AVR Programmers for Windows & Linux
AVR Studio - and toolchain from Atmel

Firmware

You will need to upgrade or flash the firmware on the ESC through the interface software. The firmware is developed by Simon Kirby.
There are different versions for various ESCs. Please refer to the ESC Spreadsheet below for the correct firmware version.

ESC Spreadsheet by TomSn0w

Brand / Model	Amp	MCU [1]	Pads	Ext. Osc. [2]	FETs [3]	FW f
TowerPro						
Mag8 TP_w12A	12	Atmega	no	no	P/N	tgy.hex
TowerPro	17	Atmega		yes	P/N	tp.hex
TowerPro	18	Atmega	yes	no	P/N	tgy.hex
TP18A_BESC	18	Atmega	no	no	N	tp_nfet.hex
TowerPro	25	Atmega		yes	P/N	tp.hex
TowerPro	25	Atmega		no	P/N	tgy.hex
TP_w25A	25	Atmega	no	no	N	tp_nfet.hex
TP W30A	30	Atmega	no	no	N	tp_nfet.hex

TP-W30A	30	Atmega	no	no	N	tp_nfet.hex
TowerPro	40	Atmega	no	no	N	tp_nfet.hex
Mag8 HXTBSC65	65	Atmega	no	yes	N	tp70a.hex
Hobbyking F-Series "UBEC"						
261000001	6	Atmega	row	yes	P/N	bs.hex (old @
261000002	10	Atmega	row	yes	P/N	bs.hex
F-20A	20	Atmega	row	yes	N	bs_nfet.hex
F-30A	30	Atmega	row	yes	N	bs_nfet.hex
F-40A	40	Atmega	row	yes	N	bs.hex
F-60A	60	Atmega	row	yes	N	bs.hex
F-80A	80	Atmega	row	yes	N	bs.hex
F-90A	90	Atmega	row	yes	N	bs.hex
Hobbyking BlueSeries						
Blueseries	12	Atmega	no	yes	N	bs_nfet.hex
Blueseries	20	Atmega	yes	yes	N	bs_nfet.hex
Blueseries	30	Atmega	no	yes	P/N	bs.hex
Blueseries	30	Atmega	yes	yes	N	bs_nfet.hex
Blueseries	40	Atmega	yes	yes	N	bs40a.hex
Blueseries	50	Atmega	yes	yes	N	bs.hex
Blueseries	60	Atmega	yes	yes	N	bs40a.hex
Hobbyking SuperSimple (HK-SS), also HiModel GX series						
Note: New (2012-04) HK-SS20A-HW boards, maybe others, ship with reset pin disabled (RSTDISBL						
Note: HK-SS18A, HK-SS20A, and maybe other sizes kill themselves (P-FET gate noise) when ran a						
HK-SS10A	10	Atmega	no	yes	P/N	tp.hex
HK-SS18A	15	Atmega	no	yes	P/N	tp_8khz.hex
HK-SS20A	18	Atmega	no	yes	P/N	tp_8khz.hex
HK-SS20A-HW	18	Atmega	row	no	P/N	tgy.hex
HK-SS30A	25	Atmega	no	yes	P/N	tp_8khz.hex
HK-SS30A-HW	25	Atmega	row	no	P/N	tgy.hex
HK-SS40A	35	Atmega	no	yes	N	rct50a.hex
HK-SS50A	50	Atmega	row	yes	N	rb50a.hex
HK-SS70A	60	Atmega	no	yes	N	
HK-SS200ALV	200	Atmega	no	yes	N	hk200a.hex
HobbyKing Multistar						
9351000065, 9351000066	6	Atmega	no	yes	N	kda_nfet_ni.he
9351000001	10	Atmega		yes	P/N	kda.hex
9351000002	15	Atmega	no	yes	P/N	kda.hex
9351000003, 9351000007	20	Atmega	yes	yes	N	kda_nfet.hex
9351000003, 9351000007	20	Atmega	yes	yes	N	kda.hex
9351000003, 9351000007	20	Atmega	no	yes	P/N	kda.hex
9351000004 after mid 2014	30	Atmega	yes	yes	N	kda_nfet.hex
9351000004 early 2014	30	Atmega	yes	yes	P/N	kda_8khz.hex
9351000004 older	30	Atmega	yes	yes	P/N	kda.hex
9351000005	45	Atmega	no	yes	N	dlu40a.hex
HobbyKing / HexTronik F3J						
HXT200A	200	Atmega	row	yes	N	hxt200a.hex
Mystery						
MY12ABEC	12	Atmega	no	yes	P/N	bs.hex
Mystery	20	Atmega	yes	yes	P/N	bs.hex
Mystery	20	Atmega	yes	yes	P/N	bs_nfet.hex
Cloud	20	Atmega	no	yes	P/N	
Cloud	30	Atmega	no	yes	P/N	tp.hex
Cloud	50	Atmega	no	yes	N	rb50a.hex
Pentium	30	Atmega	yes	yes		
Firedragon	30	Atmega	no	no	P/N	tgy.hex
Mystery	40	Atmega	yes	yes	N	bs.hex
Mystery	60	Atmega	no	yes	N	bs.hex
Turnigy dlux						
9192000026	20	Atmega	yes	yes	N	dlux.hex
Turnigy Plush / Basic / Hobbywing Pentium						
Plush / Pentium / Sentry	6	Atmega	row	no	P/N	tgy6a.hex
Plush / Pentium / Sentry	12	Atmega	row	no		tgy.hex
Plush / Pentium / Sentry	18	Atmega	row	no	P/N	tgy.hex
Plush / Pentium / Sentry	25	Atmega	row	no	P/N	tgy.hex
Plush / Pentium / Sentry	30	Atmega	row	no	P/N	tgy.hex ?
Plush / Pentium / Sentry	40	Atmega	row	no	N	tp_nfet.hex
Plush / Pentium / Sentry	60	Atmega	row		N	tp_nfet.hex

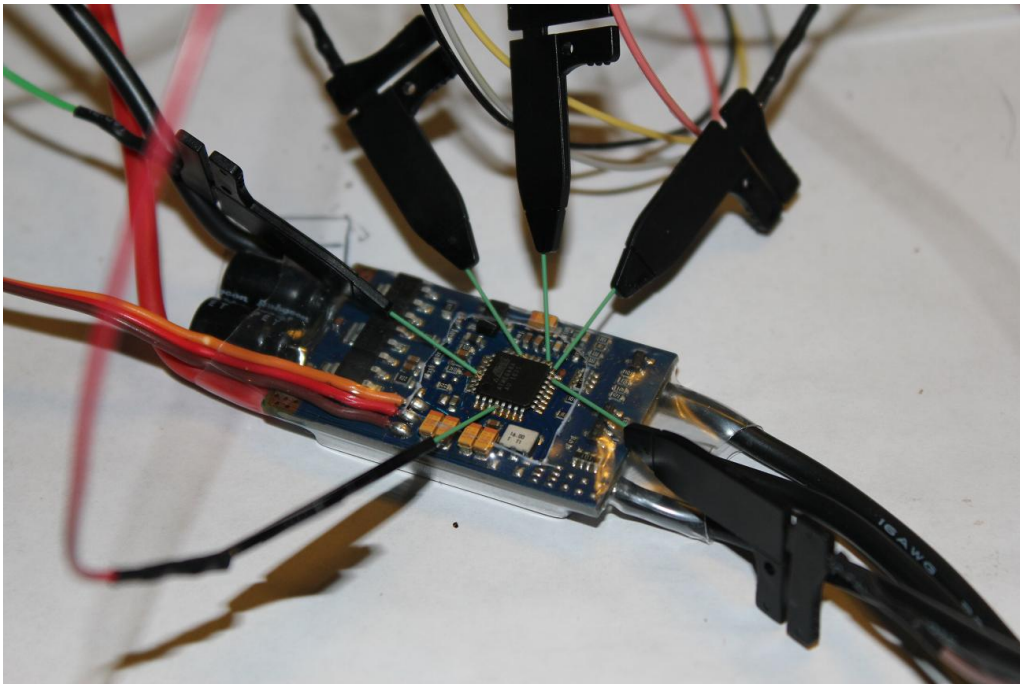
Plush / Pentium / Sentry	80	Atmega	row		N	tp_nfet.hex
Turnigy Plush / Hobbywing Flyfun / HiModel professional SiLabs						
Plush / Flyfun	6	SiLabs	row		P/N	TURNIGY_PLUSH
Plush / Flyfun	10	SiLabs	row		P/N	TURNIGY_PLUSH
Plush / Flyfun	12	SiLabs	row		P/N	TURNIGY_PLUSH
Plush / Flyfun	18	SiLabs	row		P/N	TURNIGY_PLUSH
Plush / Flyfun	25	SiLabs	row		P/N	TURNIGY_PLUSH
Plush / Flyfun	30	SiLabs	row		P/N	TURNIGY_PLUSH
Plush / Flyfun NFET	18	SiLabs	row		N	TURNIGY_PLUSH
Plush / Flyfun NFET	25	SiLabs	row		N	TURNIGY_PLUSH
Plush / Flyfun NFET	30	SiLabs	row		N	TURNIGY_PLUSH
Plush / Flyfun	40	SiLabs	row		N	TURNIGY_PLUSH
Hobbywing Flyfun 40A OPTO	40	SiLabs	row		N	TURNIGY_PLUSH
Plush / Flyfun	60	SiLabs	row		N	TURNIGY_PLUSH
Plush / Flyfun	80	SiLabs	row		N	TURNIGY_PLUSH
RC Timer						
RC Timer	6	SiLabs	row		P/N	RCTIMER_6A_[...]
RC Timer	10	Atmega	row	no	P/N	tgy.hex
RC Timer	18	Atmega	row	no	P/N	tgy.hex
RC Timer	20	Atmega	row	no	P/N	tgy.hex

Overview

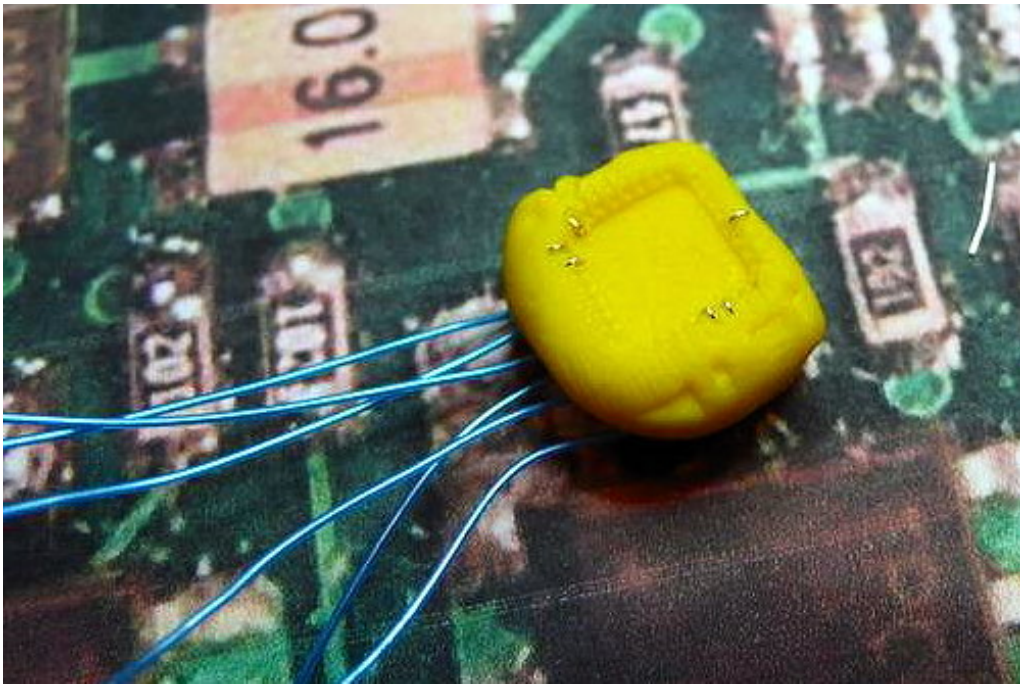
Connection to the MCU

If your ESC has the advantage of pads, then you can solder wires to them or make a non-permanent connection with pin headers:

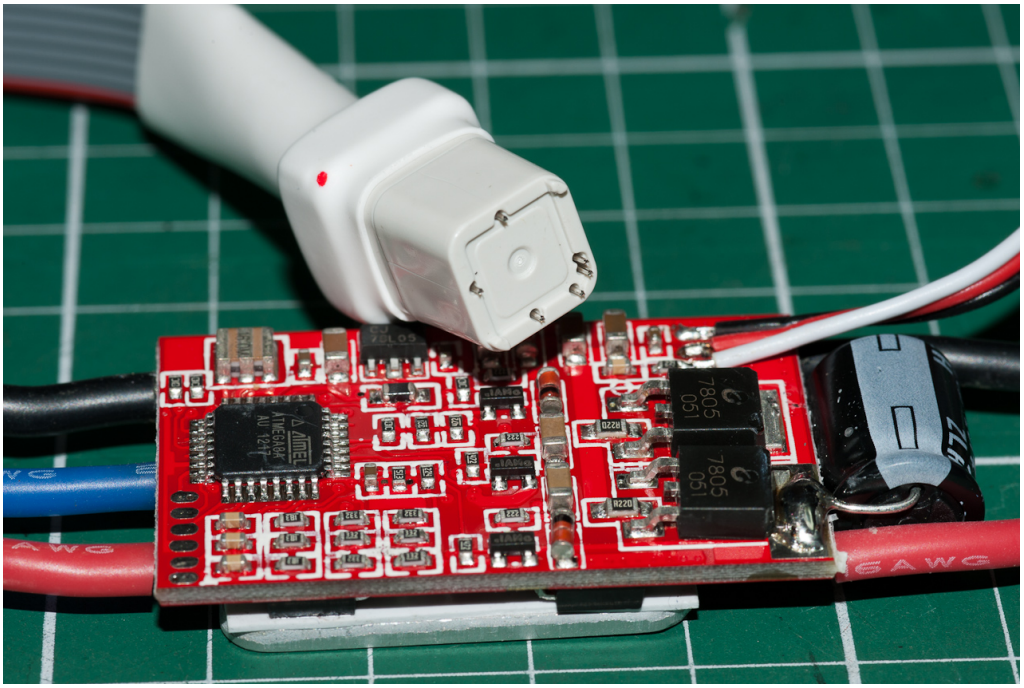
Or you have to grab the legs of the Atmel itself with micro clips:



Another method is to make a programming adapter from modelling clay. Please refer to this link for a detailed write-up of this adapter.



Another, similar adapter is now commercially made for flashing Amtel Atmega chips and is available through Hobby King:



Basic Knowledge

MOSI, MISO, SCK...

The communication between the ESC and the programmer is done over a Serial Peripheral Interface Bus, commonly referred to as an SPI. The necessary connections in the bus are called **MOSI, MISO, SCK, VCC, RST & GND**. These terms are the denominations for the connections between the programmer and the ESC, hence the reason why the connections on the ESC pictures are called like this. You need to connect the correct pins of the programmer to the correct pin on the ESC. See the "Connection" section below for more information.

Fuses

The term fuses has nothing to do with a regular protection you will find in ordinary electronics. The term fuse in this case applies to a small part of the MCU (micro controller). Depending on the fuse settings, the MCU knows how it should act. Incorrect fuse settings will lead to improper behavior, hence the reason they are specified. The latest Simonk firmware doesn't normally require the fuses to be changed or adapted.

Fuses can be calculated or determined with fuse calculators. E.g. Engbedded Atmel AVR® Fuse Calculator

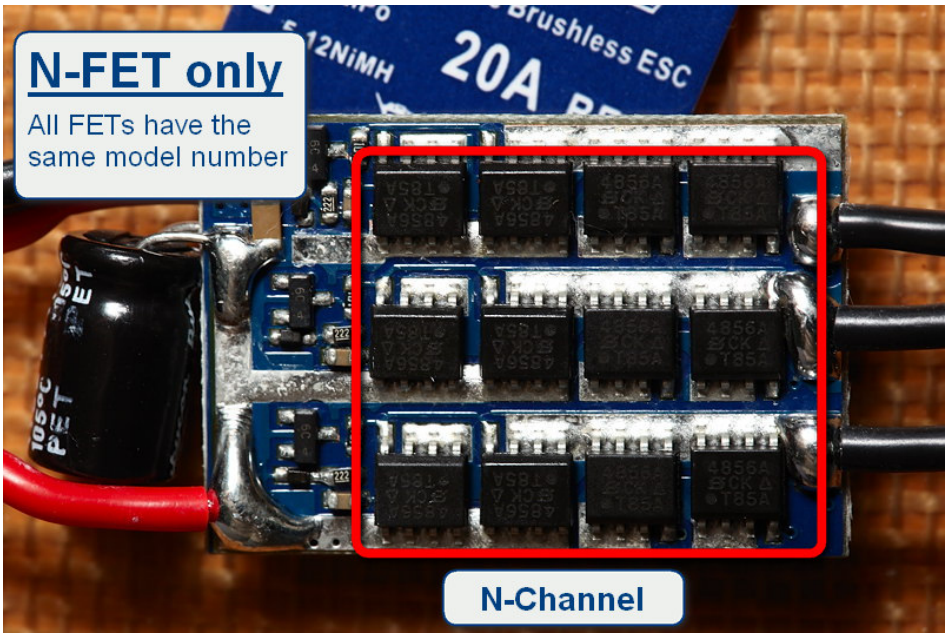
Hex files

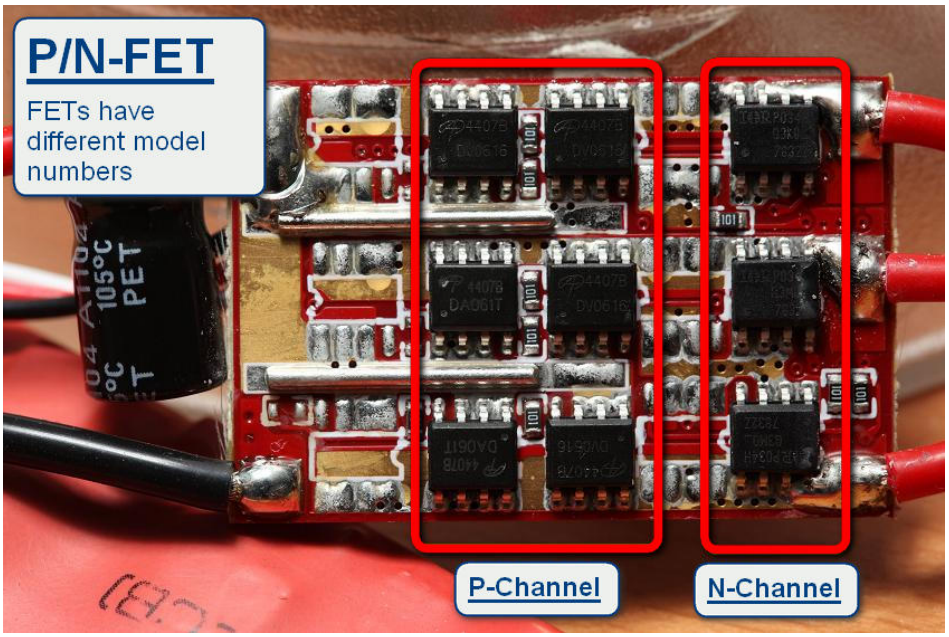
The firmware for the ESCs is available in a .hex file. One .hex file contains the complete firmware for one particular ESC. You can flash your ESC multiple times but only the last flashed firmware is active on the micro controller. If the wrong firmware is flashed, you may end up with a damaged ESC when it's powered by the battery supply. It's important to safely test your ESC after the firmware upgrade.

FET type

An ESC has either P-channel and N-channel FETs mixed or N-channel FETs only. The ESCs which use only N-channel FETs have the advantage of lower resistance, less losses and hence more efficiency. The matching .hex file depends on the type of FETs. It is also possible that the latest version of an ESC has N-FETs only whereas an older version used P/N FETs.

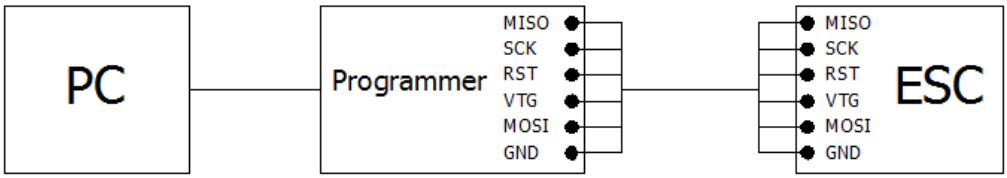
This is how you determine the FET type:



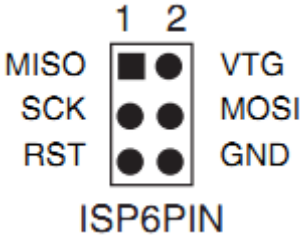


Connection

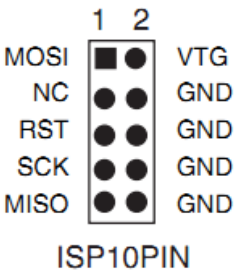
An Atmel ISP header comes typically in 2 versions, a 6-pin or 10-pin layout. You must connect your ESC connections (**MOSI, MISO, SCK, VCC, RST & GND**) to the same pins of the programmer.

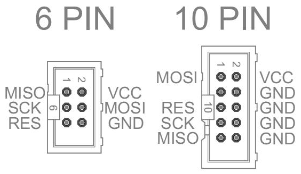


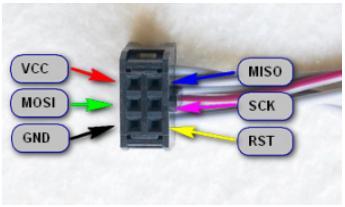
6-pin ISP socket pin-out

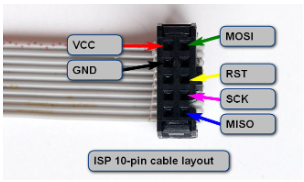


10-pin ISP socket & plug pin-out.

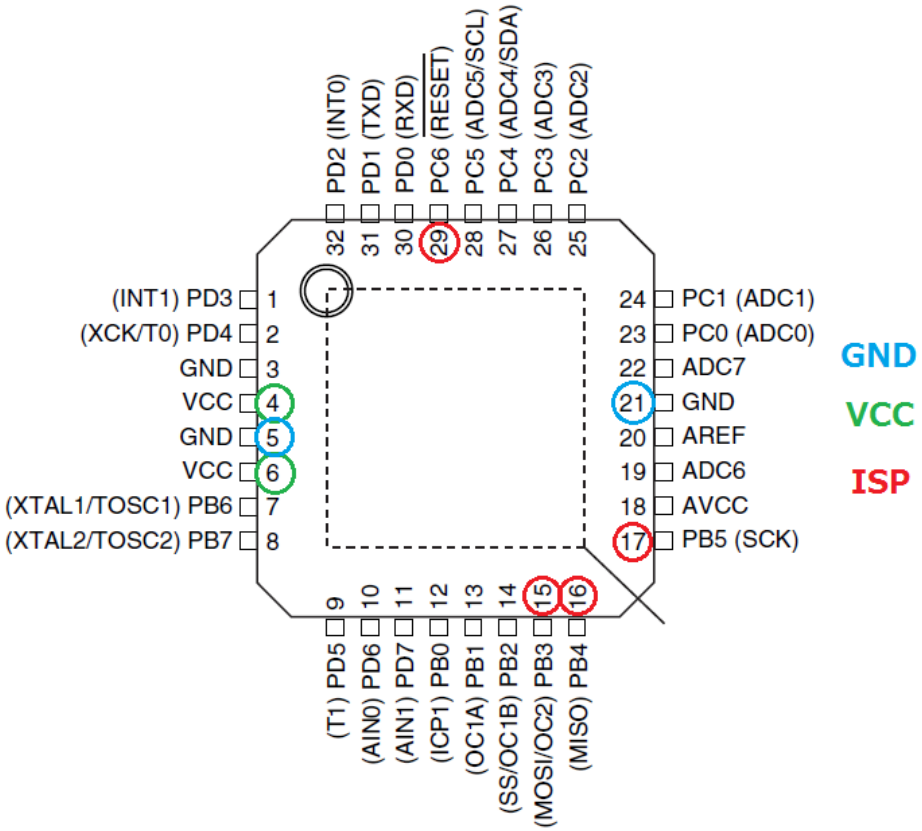








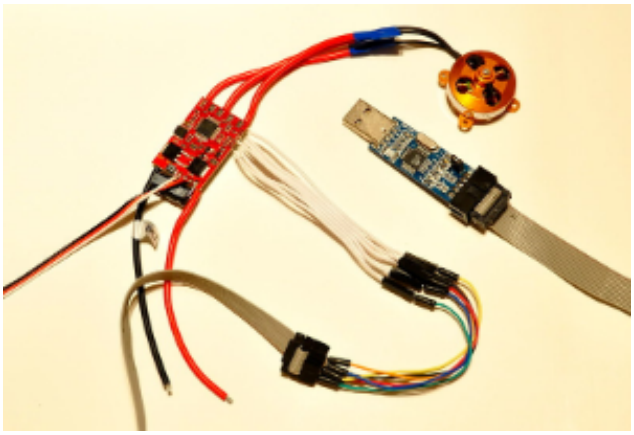
Atmega8 pinout



Flashing step by step examples.

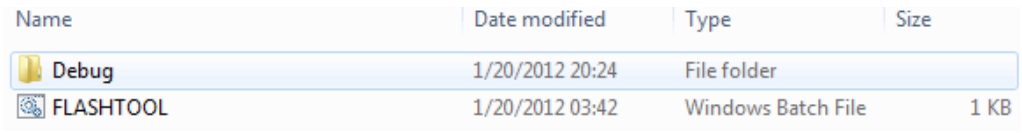
Example: Turnigy Plush 18A

In this example, an older Turnigy Plush 18A ESC will be flashed on a Windows system using the ESC Flash Tool with a USBasp adapter.
You should have connected the programmer to the computer and your ESC to the programmer. In this example, the connections are soldered onto the ESC.



The ESC Flash Tool requires Microsoft .NET Framework to run. If you do not have .NET framework installed, you can download and install it from the Microsoft website (free of charge).

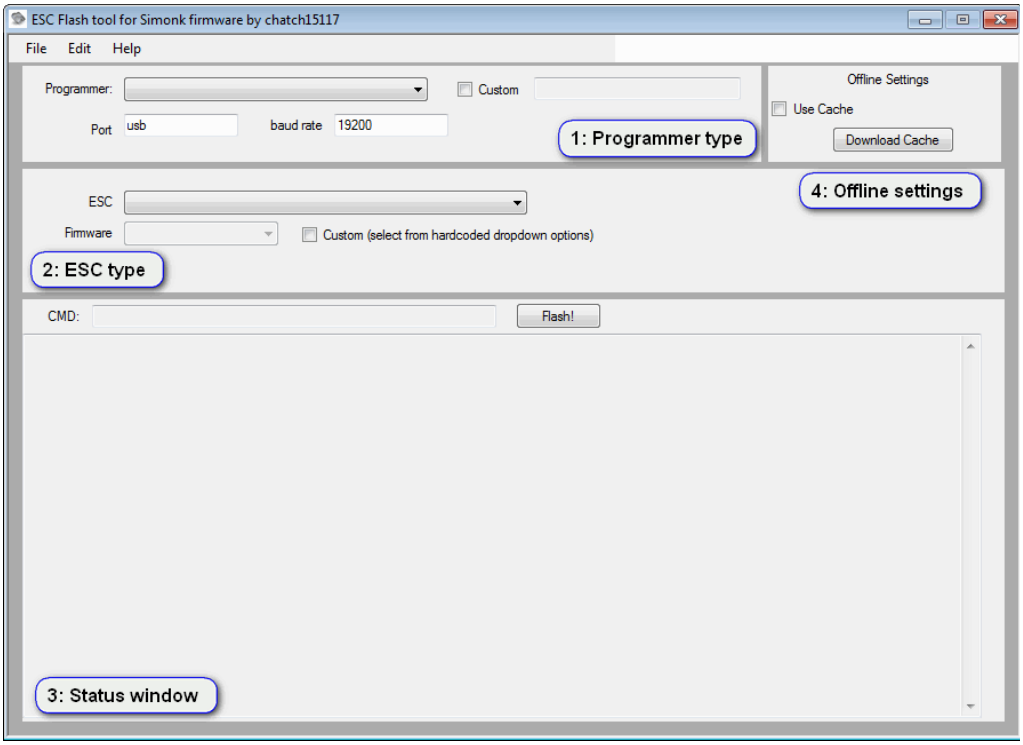
The ESC Flash Tool is packed in a *flashtool.zip* file. If you haven't done so, you can download the latest version here. Extract the contents of the *flashtool.zip* file in a directory of your choice. When the tool is extracted, a directory called Debug and a FLASHTOOL.bat file should be available.



Name	Date modified	Type	Size
Debug	1/20/2012 20:24	File folder	
FLASHTOOL	1/20/2012 03:42	Windows Batch File	1 KB

Start the tool.

Execute FLASHTOOL.bat or start ESCFlashTool.exe in the Debug directory.

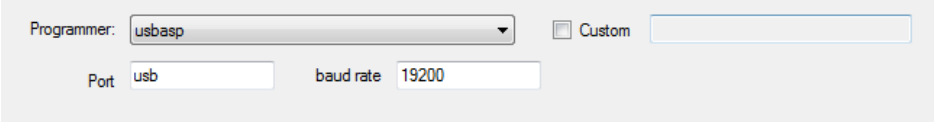


The tool has different interface sections:

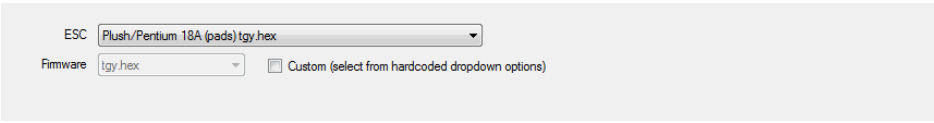
Section 1 identifies the programmer type.
Section 2 specifies the correct ESC type.
Section 3 indicates the status of the flashing process.
Section 4 is used when the tool is used in offline mode without an internet connection.

Selections

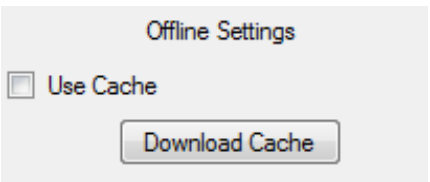
1. Select your interface programmer. In this example, a USBasp USB interface programmer is used to flash the ESC.



2. Select your ESC from the list.
   Refer to the ESC database to check the correct ESC and .hex file combination.

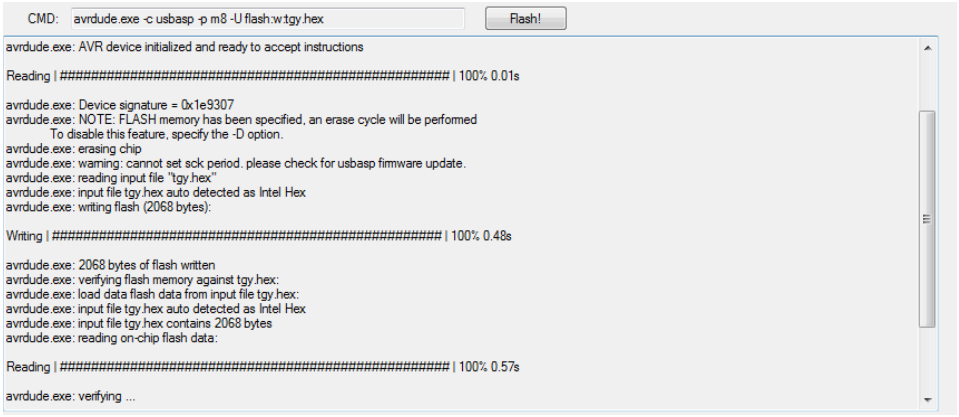


3. If you do not have an internet connection, you can select to use the firmware files on your computer (use cache). Normally you won't touch this setting. The program will automatically download the latest firmware files from the internet.



Flashing

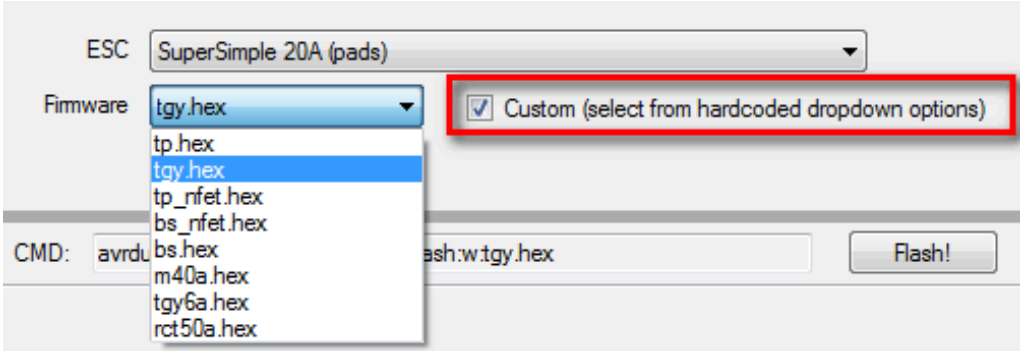
Press the Flash! button. The tool will download the latest firmware files, or use the offline files if that option was checked.



Your ESC is flashed with the new firmware. These are the only steps needed for most ESCs. You should test your ESC now.

Advanced flashing

If the ESC in the list is not associated with the correct .hex file, you can flash an alternative firmware using the *Custom* check-box.



If a special firmware was designed for your ESC and it's not available through the regular selection, then choose the advanced option in the *Edit* menu. This option allows you to select an alternative firmware file.

Testing your ESC.

If you have flashed your ESC and something is wrong, then the chance is real that the ESC will get damaged immediately when it's powered up.

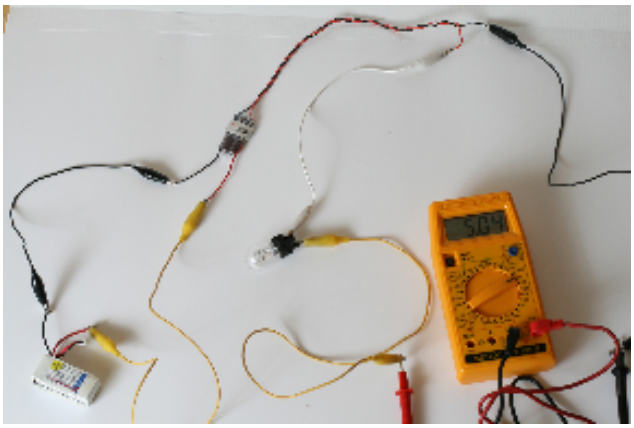
Power the ESC for the first time

The safest arrangement is to use a current-limited bench power supply, set to a low voltage (6V-7V), for both flashing and initial testing. If the pin-out is wrong and causes a short, the current-limiting causes the input voltage to drop below the brown-out detection voltage of the MCU, causing all output pins to go high-impedance in hardware and an automatic reset when the voltage comes back.

If you do not have a current-limited supply, you can improvise by using 4 AA

batteries or an old NiCad pack, or even a LiPo with a 12V light bulb in series to act as a current limiter. Also, be careful when touching the board, since it can be quite easy to turn on a FET gate with just your finger. This should be okay if you have a current-limited supply, since it should just reset.

Sample current-limited supply using 3s Lipo, 5v ubec & lightbulb.



ESC start-up tones

To confirm that your ESC has been successfully flashed, it should make a different start-up sound compared to the original firmware.
In the following video, the start-up tones from a copter with unflashed ESCs is compared to one flashed with SimonK firmware.

0:03

00:37

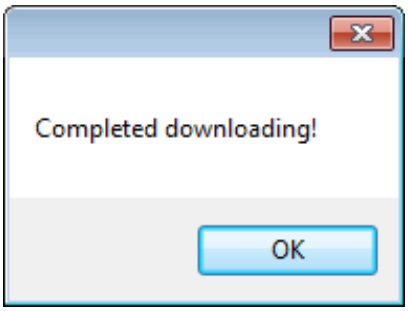
FAQ's

My ESC doesn't arm.

A flashed ESC should generate 3 increasing tones at start-up. If the signal pulse is below ± 1030µs then a fourth beep is emitted. In the below movie, the ESC is started and then the signal is lowered slowly below 1030µs which generates the 4th beep. If you connect the flashed ESC to your receiver, you may need to lower your end points to "arm" the ESC.

The firmware is not available offline.

You should download all the firmware files with the *Download Cache* option before you can use this function. The tool will then download all files and notify the user when the files are downloaded and available for offline use.



Can I program the ESC like I normally would, setting the recommended settings?

No. Once the ESC is flashed, you cannot program the settings like brake on/off, Lipo/NiMh etc. etc. All the best settings have been embedded in the upgrade. However, the throttle range is still programmed like most standard ESCs. **Remove your props,** and disconnect the battery. Push your radio throttle to max, connect the flight battery, wait for the unique beeps, then immediately lower the throttle to 0.

You will then hear another unique beep, indicating low throttle set. Disconnect the battery.

References

- Turnigy/TowerPro/AfroESC/BlueSeries AVR ESC software from Simon Kirby (SimonK).
- Comprehensive writeup of the ESC firmware including flashing & testing information. (must read).
- RCTimer/Turnigy/Hobbywing ESC DIY Firmware Flashing RCG thread. (A lot of information has been gathered and grouped from that huge thread.)