

## PWM,OneShot, and DShot Protocol ESCs

The most common electric motor used with ArduPilot vehicles is brushless and requires a brushless ESC for control. The main protocols for controlling these ESCs are PWM, OneShot, and DShot. In addition, some include the ability to be configured via an external PC application called BLHeli Suite or BLHeli32 Suite. See [BLHeli32/ BLHeli-S Pass-Through Support](#) for more information on ESCs with that capability, but the following sections apply to ArduPilot's setup for those ESCs also.

### PWM

These are the most common ESCs for non-copter applications and were historically the first brushless ESCs.



PWM ESCs use a periodic input pulse of width typically between 1000uS and 2000uS for zero to full power, respectively. The frame rate of these pulses is usually between 50Hz to 490Hz. The faster frame rates allow quicker control reactions to be sent to the motor, if the ESC has capability for those frame rates.

revers, and `SERVO_125` for PWM ESCs for normal forward motors only. `Q_125` controls this for PWM protocol VTOL esc/motors in QuadPlane.

### ⓘ Note

be sure of the capabilities of your ESC before selecting a higher frame rate to avoid damage to the ESC.

## OneShot 125

An even faster PWM protocol is OneShot125 (sometimes shortened to just OneShot). If the ESC has this capability, then the pulse widths are divided by a factor of 8 for even faster communication from the autopilot to the ESC since the commands get to the ESC 8 times faster at any given frame rate. In addition, the capability to increase the frame rate up to 490Hz is allowed.

### ⓘ Note

OneShot (vs OneShot125) is an older protocol that uses the same pulse widths as Normal PWM, but has a fixed frame rate equal to the autopilot main loop rate. It has been replaced by OneShot125.

## DShot

DShot is a digital ESC protocol. In contrast to traditional servo-type PWM it allows fast, high resolution digital communication. This opens the door for more precise vehicle control. This is especially useful in multirotor and quadplane applications.

DShot is available in various digital communication baud rates, as well as a version supporting ESC telemetry via the control input connection (some ESCs offer telemetry reporting via a separate serial connection):

- DShot150 at 150kbaud (recommended for larger aircraft with long signal lead runs)
- DShot300 at 300kbaud
- DShot600 at 600kbaud (recommended)
- DShot1200 at 1200kbaud
- Bi-directional DShot at 150, 300, 600 and 1200kbaud on supported firmware (includes telemetry)

### ⓘ Note

capability. See [BLHeli32/ BLHeli-S Pass-Through Support](#) for more information on ESCs with that capability.

### ! Note

Only try DShot on ESCs that are known to support it or you will get unpredictable results. Reverse thrust is supported in ArduPilot 4.0 and later firmware versions.

The DShot ESC protocol's key advantages are:

- all values sent to the ESC are checksum-protected
- clock differences between the ESC and autopilot don't affect flight performance
- no need to do any ESC throttle range calibration
- very high protocol frame rates are supported

While DShot is available on many ESCs, those with BLHeli32 and BLHeli-S setup capability, offer the most performance and often, additional features such as ESC telemetry. In addition, ArduPilot provides special programming capability for BLHeli32 and BLHeli-S ESCs via its passthrough feature (See [BLHeli32/ BLHeli-S Pass-Through Support](#) section.)

### ! Note

Recently there is a growing number of proprietary and non-proprietary 16 / 32 bit ESCs with firmware that support DShot and other digital ESC protocols, but not BLHeli\_32-specific features like passthrough and telemetry. See your ESC's manual for further detail on supported features.

For larger aircraft with longer cable runs using DShot ESC protocol, we recommend using the lowest baud rate, DShot150, as it is the most reliable protocol (lower baudrates are less susceptible to noise on cables). Higher rates can be more susceptible to noise but also tie up the allocated DMA channel for a shorter period so can be beneficial on flight controllers with a lot of DMA sharing.

For smaller craft, DShot600 is by far the most widely used and can therefore be a more suitable choice simply because of the amount of testing that it has had, rather than the newer DShot1200 protocol.

Bi-directional DShot involves a longer pulse width since it has to wait for a response from the ESC before it can send another pulse and thus DShot300 and DShot600 are to be preferred. Bi-directional DShot does not share DMA channels and so there is no impact on other peripherals.

### ! Note

output will always be ignored since DShot does not use these parameters. The trim value used will be 1500 if it's a reversible output, or 1000 if normal output setup in DShot, and the output range always be 1000-2000. No ESC calibration step is required.

## Configuration

### Protocol Selection

The ESC control protocol is selected by the [MOT\\_PWM\\_TYPE](#) parameter on Copter and Rover, or [Q\\_M\\_PWM\\_TYPE](#) on QuadPlanes for its copter function motors.

#### ! Note

The autopilot should be re-booted after changing the protocol type.

On Plane, all other motors use Normal (PWM) protocol. However, in Plane, any motor, like the traditional fixed wing's main motor or Dual Motor Tailsitters ([SERVO<sub>n</sub>\\_FUNCTION](#) = 70 throttle, 73 throttle left and / or 74 throttle right), can be changed to a protocol other than PWM using the [SERVO\\_B<sub>LH</sub>\\_MASK](#) parameter to specify the output number of the motor together with the [SERVO\\_B<sub>LH</sub>\\_OTYPE](#) parameter to select the protocol of these motors independently of that selected for the copter function motors.

#### ! Note

All mask-based configuration can only be changed at a PWM group level, please consult the documentation for your flight controller to ascertain which outputs are on different groups. See [Mixing ESC Protocols](#) section below.

#### ! Note

In Plane, Oneshot (not Oneshot125) loop synchronization can be added to any motor running an normal PWM ESC using the [ONESHOT\\_MASK](#) bitmask. It will trigger pulses at the [SCH<sub>ED</sub>\\_LOOP\\_RATE](#), but no lower than at 250Hz. Be sure the ESCs can handle this rate.

#### ! Warning

Be sure your ESC can support the configuration you select for it. Damage can occur otherwise. This includes frame rates discussed below. Also be careful when switching between digital and analogue output types without re-calibrating ESCs as this can lead to uncommanded motor output.

## DShot Update Rates

The frequency at which DShot pulses are sent can be configured through [SERVO\\_DSHOT\\_RATE](#). By default ArduPilot will output a DShot data pulse every time a new IMU sample is ready *and* at a fixed 1Khz interval. On a copter with the standard 400Hz scheduler loop rate this works out at about 1.4Khz. However, the output is quite irregular - in order to get more regular output [SERVO\\_DSHOT\\_RATE](#) can be configured to send pulses at multiples of the scheduler loop rate. Thus if set to 2 the pulses will be sent at 800Hz, set to 3 at 1.2Khz and so on. The difference being very, very even output which can benefit copters needing tighter motor control (for instance smaller racers). It is not recommended to send pulses at less than 1Khz due to reports of BLHeli32 occasionally missing frames on some flight controllers, similarly sending at higher rates can result in increased reliability and faster recovery from missed pulses where needed at the cost of some CPU. Very high rates can only be used for faster DShot speeds since otherwise pulses might overlap - for instance the fastest rate that DShot150 can theoretically support is 4Khz.

## DShot Commands

On certain ESCs DShot commands are supported. These allow functions such as ESC LEDs, beeps and motor direction to be manipulated by the flight controller. In order to use DShot commands [SERVO\\_DSHOT\\_ESC](#) must be set to the type of ESC that is in use. Notify functions (e.g. LEDs [NTF\\_LED\\_TYPES](#) and Buzzer [NTF\\_BUZZ\\_TYPES](#)) can then be configured to include DShot as an output type.

The current commands supported are:

- Dshot LEDs on/off
- Dshot Buzzer on/off
- Reverse motor direction
- Enable 3D operation

### Warning

Currently, ArduPilot supports the command set ([SERVO\\_DSHOT\\_ESC](#) =1) that is commonly used, however, others are appearing and may not be compatible, resulting in undefined operation. Use caution (remove blades!) until correct operation using type=1 is verified for your ESC

## Mixing ESC Protocols

While all the servo/motor outputs of an ArduPilot autopilot are capable of Normal PWM operation at 50Hz frame rates, not all are capable of other ESC protocol configurations. And, usually, these configurations must apply to pre-designated groups of outputs, even if they are

1. The 8 “MAIN” outputs of autopilots using an IOMCU (like PixHawk and Cube), cannot be used for protocols other than Normal PWM and OneShot. On these autopilots, only the additional “AUX” outputs can properly support OneShot125 and DShot protocols. If you attempt to set a “MAIN” output to DShot, then normal PWM output will occur, even though it has been set to a DShot protocol.
2. For [Pixracer](#) and [other boards](#) without a separate IOMCU coprocessor, all servo/motor outputs can be used.
3. Groups of outputs sharing a common timer, MUST have the same advanced configuration. Usually, these are specified in the autopilot’s hardware description linked from the [Autopilot Hardware Options](#) page. For example, if an output is configured for DShot in a group, then you cannot use another output in that group for Normal PWM ESC or normal PWM servo operation.

#### Note

Everytime the autopilot initializes, it sends a log message to the ground control station, showing which outputs are PWM/Oneshot/or DShot. The remaining higher numbered outputs are assigned as GPIOs.



