A Note on ESC Protocols
Aerial Robotics Lab

Ali Grivani
400165627

August 27, 2019

1 ESC Communication Protocols

While calibrating and setting up the configurations on the flight controller, there are several option for ESC protocols and one may not know what to choose. For instance, in Cleanflight, in configuration tab, there is a section called "ESC/Motor Features". In this section, we will try to provide some useful information and further detail about ESC protocols and hopefully, the reader would be able to choose the best one for their application.

1.1 Definition

The flight controller generates the proper commands (control input signals) for a vehicle to follow the desired path and send them to a board called ESC (Electronic Speed Controller). As its name suggests, an ESC tries to control the rotation speed of the motors based on the command given by the controller. In this procedure, the controller and ESC must be able to understand and communicate appropriately and therefore, they need a language or a common ground to interpret their sentences. This language is regarded as a communication protocol.

1.2 Most Commonly Used Protocols

1.2.1 PWM

PWM is an abbreviation for Pulse Width Modulation which is an analog type of communication. This protocol utilizes electrical pulse to determine the value of the signal and these pulses are continous with a defined duration (period). There are two essential properties for each PWM signal and that is frequency and duty cycle. *The duty cycle describes the amount of time the signal is in a high (on) state as a percentage of the total time of it takes to complete one cycle. The frequency determines how fast the PWM completes a cycle (i.e. 1000 Hz would be 1000 cycles per second), and therefore how fast it switches between high and low states*[1].

Based on the duty cycle, the value of the signal is interpreted. For instance, in the old days, the minimum and maximum values for a PWM signal were 1 and 2 ms, respectively. In other words, if a 1-ms pulse was received, the pulse corresponded to the minimum value of throttle (zero). Theoretically, a PWM pulse could reach a frequency of 500 Hz (based on that 2-ms pulse length) and in practice, the frequency of 490 Hz can be achieved. *Nowadays, for a standard PWM, it will take 1000 to 2000 microseconds to send one data packet.*

Recently, a great number of ESCs in the market support PWM with higher speeds up to 600 Hz frequency. Although this protocol has had a promising performance over the past decade, the refresh rate gradually fell behind the current advancements in computational power. In particular, for aerial vehicles, faster rates of update and responses were required to stabilize and dampen the effects of undesirable disturbances. Hence, developers tried to solve the issues such as latency, delays, and interference by offering new protocols such as Oneshot, Multishot and Dshot.

[1]From here.

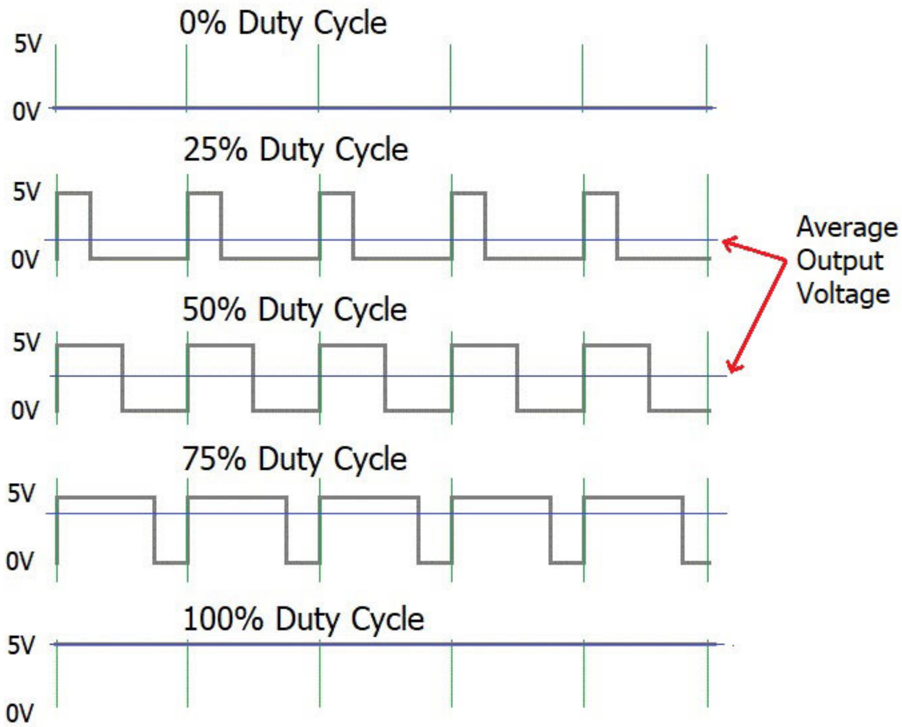

Figure 1: Pulse Width Modulation Signal. Source: circuitdigest.com

1.2.2 Oneshot

In terms of structure, Oneshot protocol is similar to PWM with some significant improvements. It was firstly introduced by BLHeli firmware developers and their major contribution was to alleviate synchronization and latency problems, both at the same time. Analogous to PWM, this protocol falls into the realm of analog communication protocols and the main difference with its ancestors (PWM) is in higher refresh rate and different pulse modulations. Three major improvements can be listed as follows:

1. Synchronization of the signal with the controller control loop.
2. Shorter duty cycle which reduces the latency between the controller and the ESC.
3. Higher rates of updates for control inputs.

Recently, the newest controllers are capable of performing at frequencies up to 32kHz and high-rate protocols such as Oneshot give us the chance to adjust rapidly the control commands in case of disturbances and thus, the stability of the drone can be guaranteed.

Three main versions of this protocol is presented by state-of-the-art firmwares:

1. Oneshot125: 125 - 250 μs pulse length
2. Oneshot42: 42 - 84 μs pulse length
3. Multishot: 5 - 25 μs pulse length

A visual comparison between PWM and its successor, Oneshot is depicted in the following figure[2].

[2] Borrowed from here.

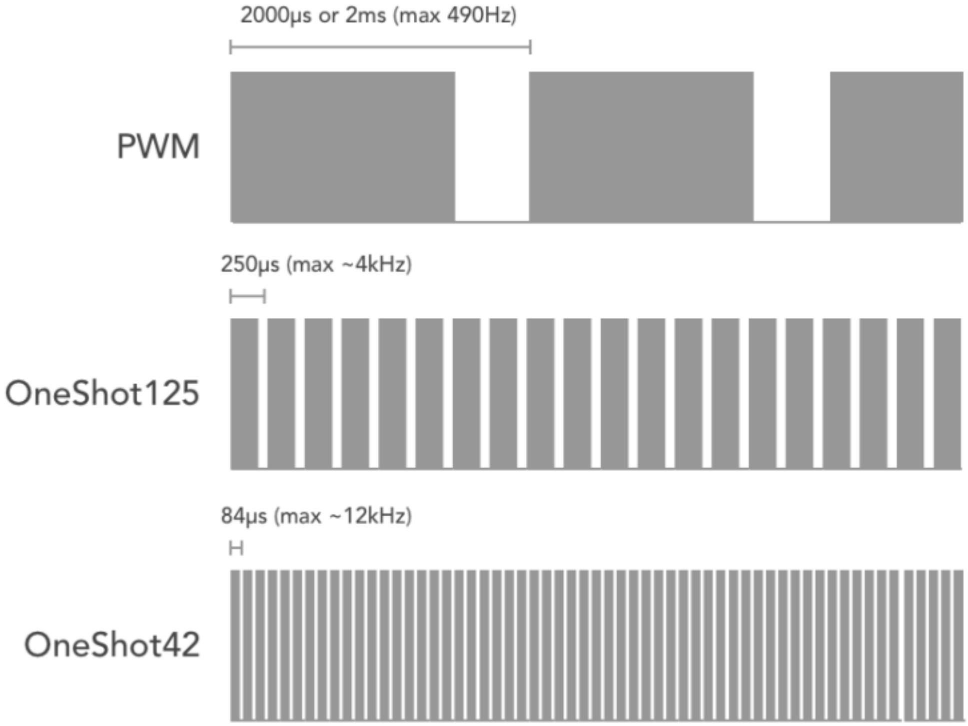

Figure 2: A comparison between Oneshot and PWM

1.2.3 Dshot

The two protocols introduced above had a common and crucial characteristic. They were both analog. In contrast, a new generation of ESC protocols has been introduced in the past few years and their structure is digital. They are fundamentally different from the previous versions, known as DShot. It was introduced by BLHeli_S firmware and is able to perform at a different range of speeds. Rather than using an electric pulse to determine the command value, DShot uses a sequence of binary values (which can be described as a sequence of pulses).

DShot data packages consist of 16 bits: 11 bits for throttle value, one bit for telemetry request and 4 bits for checksum. *The telemetry request serves to request data back from the ESC on models that support it and transmission of telemetry data is done in a different line*[3]. The following figure provides a good visual clue about the differences between OneShot and DShot.

[3] Borrowed from here.

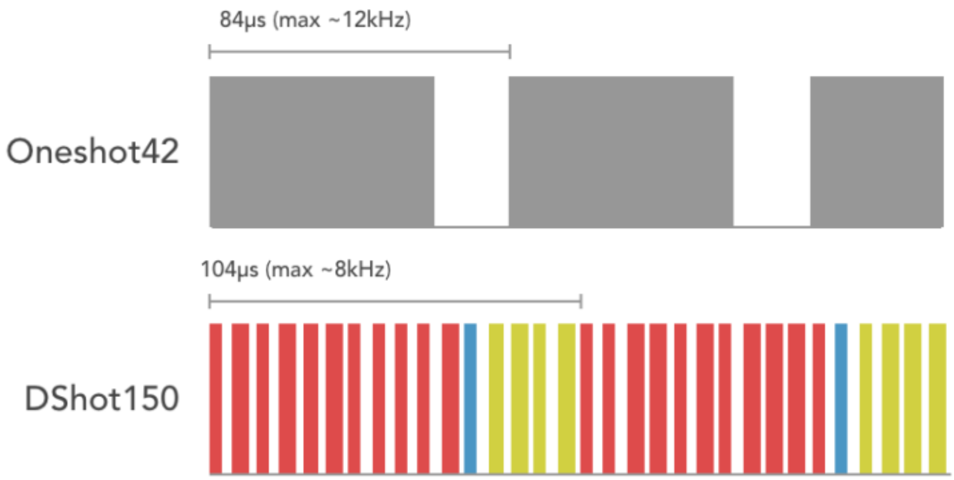

Figure 3: Comparison Between OneShot and Dshot Protocol

Based on the speed of data transfer, four versions of DShot has been reported as following:

1. DShot150: 150 kilo-bits per second, 9375 Hz update frequency.
2. DShot300: 300 kilo-bits per second, 18750 Hz update frequency.
3. DShot600: 600 kilo-bits per second, 37500 Hz update frequency.
4. DShot1200: 300 kilo-bits per second, 75000 Hz update frequency.

1.3 Challenges, Disadvantages and Advantages

Analog protocols are prone to electrical interference, which can come from different sources. The electrical devices such as brushless DC motors, ESC board, flight controller and other stuff, they all increase the possibility of electrical issues such as interference in the circuit. With this point of view, the protocols with a shorter pulse width (newer ones) are more vulnerable to be corrupted compared to old-fashioned PWM protocols. Nonetheless, this problem has been tried to overcome by sending multiple pulses within one single loop of the controller and averaging over all the pulses. That is one of the reasons why some experienced people suggest to run the controller at higher rates and claim that this would help to improve the performance.

While using analog protocols, PWM for instance, the ESC needs to be calibrated before using. The communication between ESC and the flight controller depends on the speed of their clock generators. If they are not consistent, the measurements and interpretations of ESC might have some errors, leading to serious inaccuracies. Moreover, both the sender and the receiver must know the maximum and minimum level of PWM pulse and also there is a fact that different brands of ESC have different default values.

Fortunately, using digital protocols such as DShot150, one may not expect these problems to occur. ESC calibration will not be necessary anymore. Digital signals are more resistant to electrical interference and noise since they are essentially one's and zero's. In theory, by using DShot protocol, the values sent to ESC from the flight controller are supposed to be precise and flawless, hence, it is not required to average over multiple pulses. Furthermore, the checksum mechanism will allow us to know if a package is correct and reliable. Consequently, the noise in the system loses its importance in comparison to what we have in analog protocols.

Meanwhile, the rise of DShot protocol has increased the requirements for hardware and electrical boards. Therefore, not all flight controllers are able to support them. In addition, one of the other drawbacks shows itself when LC filters are used to eliminate noise from signals (such as video signals). The same problems may happen in some specific structures, for example in large UAVs with long wires, the electrical properties of the involved components can culminate in acting as a low-pass filter. Besides, other deficiencies such as poor-quality soldering make things more complicated and worse.

1.4 Final Recommendation

According to the above explanations, one might easily come to this conclusion that newer protocols, e.g. DShot150, are better than the conventional PWM protocol. Generally, it is an accepted idea among the users and can be easily found in all the related discussion forums. All in all, it is suggested to use DShot protocols which are faster and safer. However, if you do not need super-fast communication, the recommendation is to stay with DShot 150 or even MultiShot protocols, since they are supposed to work fine and flawlessly.

Note: This suggestion is a general purpose notion and it was evaluated based upon technical judgments and online developers comments. In our lab, we have been using PWM protocol based on our specific application and programming.