07/30/2023
Wesley Cooth

Blue Robotics T200 Underwater thruster

The T200 is a 3phase brushless dc motor. It operates in a similar manner to a stepper motor, but is constructed very differently. Based on the way current is applied to the three phases, the motor will spin. However, it ~~is much f~~ can achieve much greater speeds.

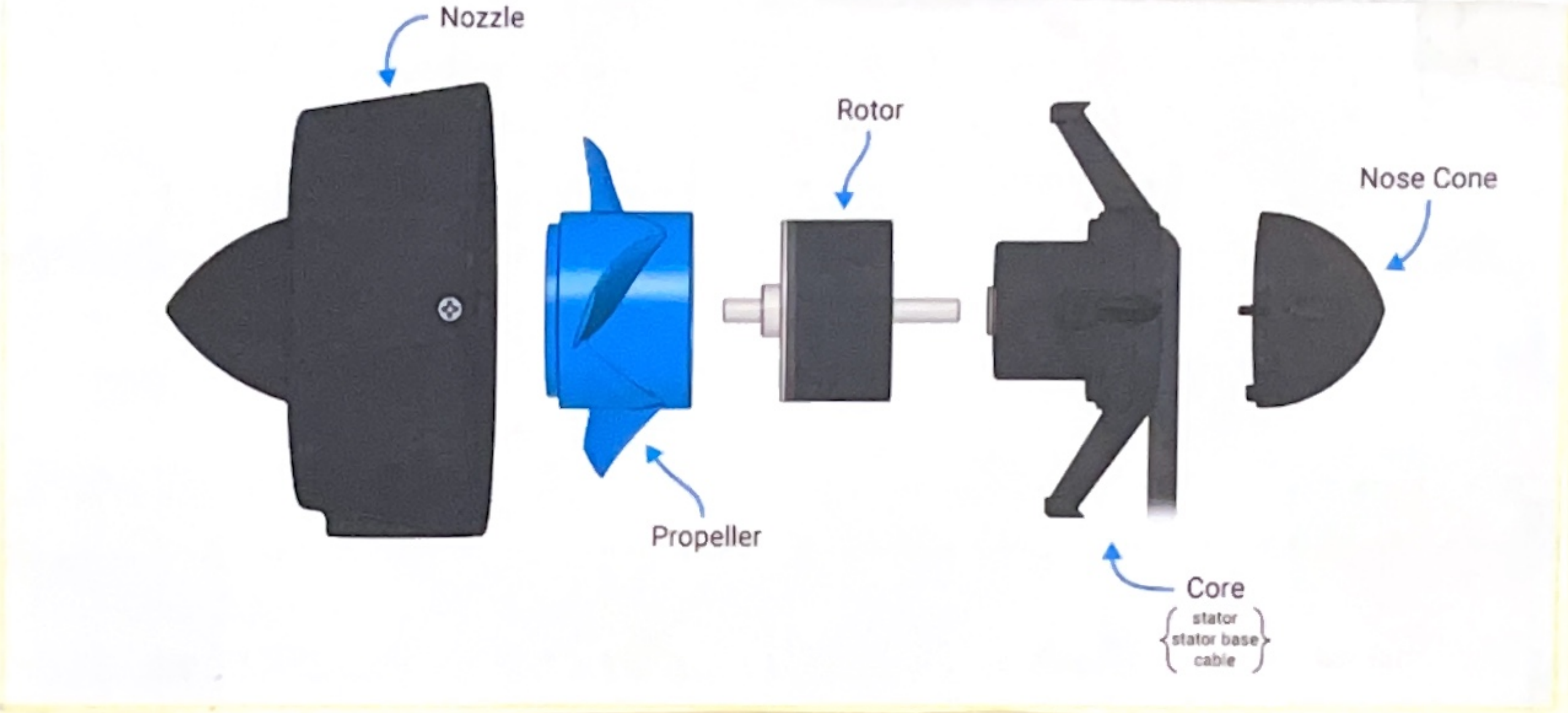
These motors require an electronic speed controller.* The speed controller will handle sending current to the correct phases at the appropriate times. The ESC requires a pulse of 20 ms with an on period of between 1000us and 2000us.

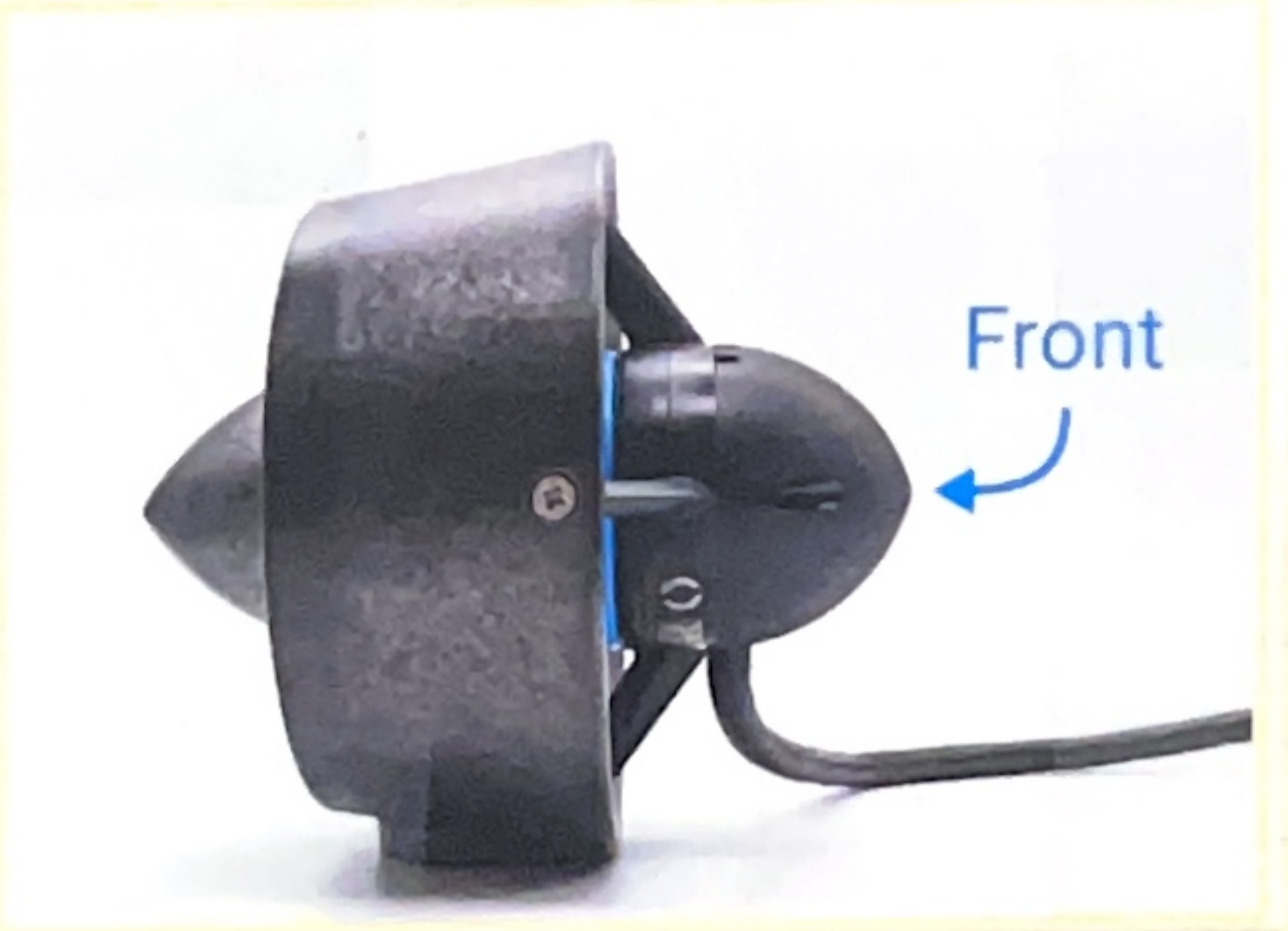
*(ESC)

Parts of a Thruster

This diagram shows the main components of a T200 or T500 thruster.

https://bluerobotics.com/learn/thruster-usage-guide/





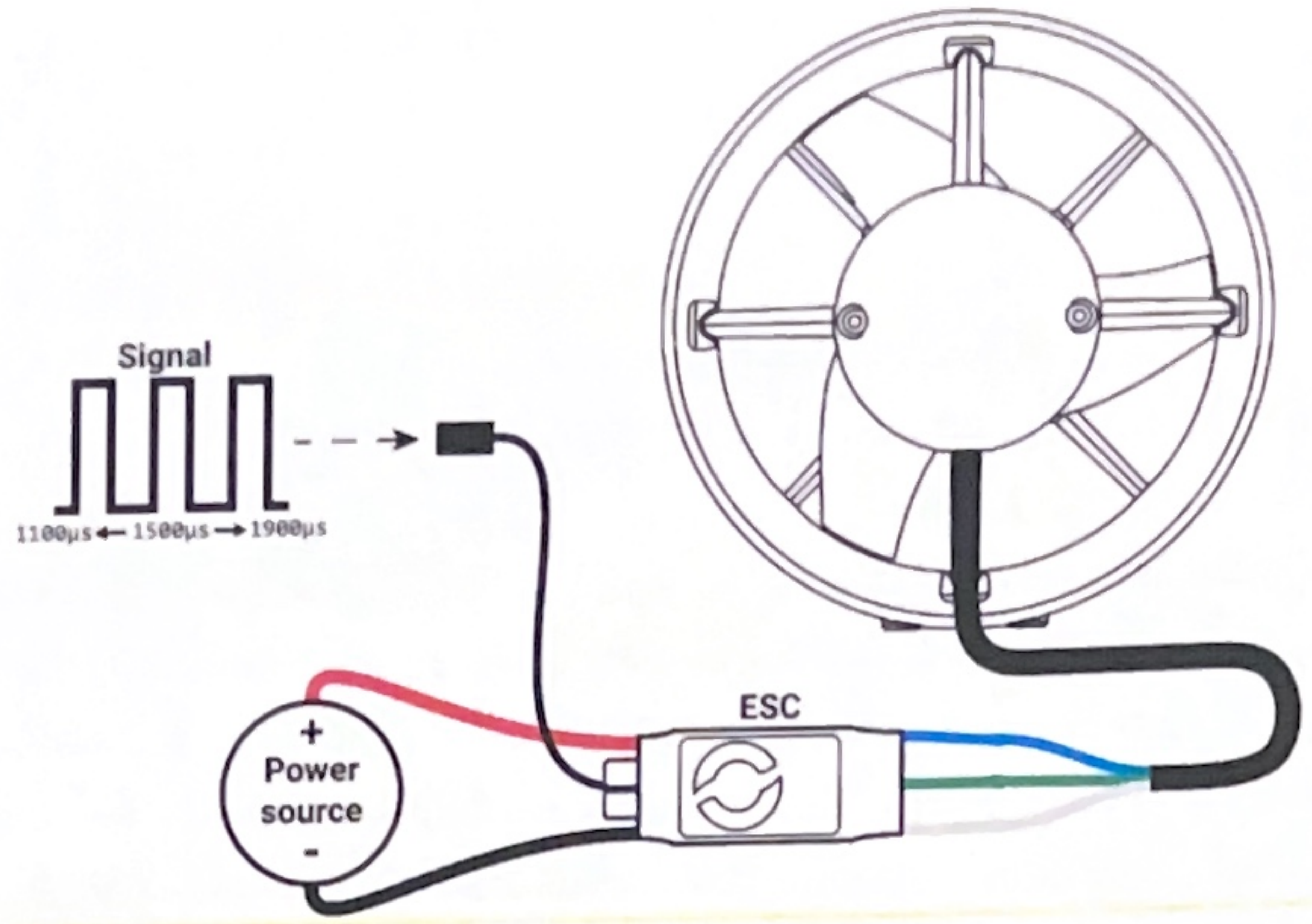
Figures 1 and 2

Basic components of the T200 thruster.

Blue Robotics T200 Underwater Thruster

07/30/2023
Wesley Cooper

The following diagram shows how everything is connected:

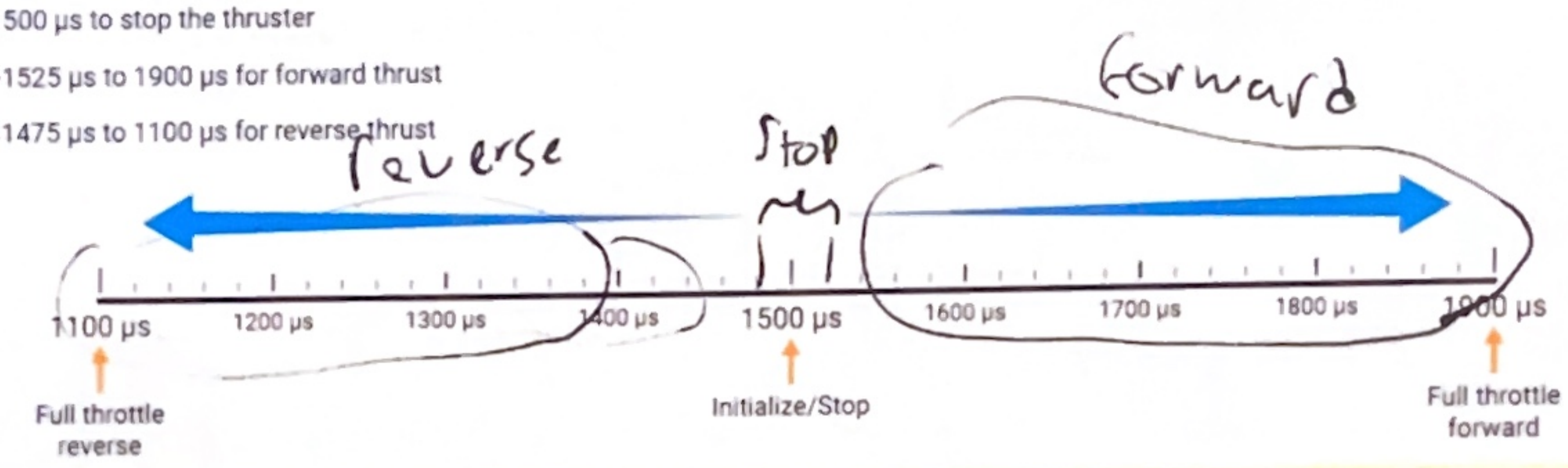


3. Sending the Signal

Connect the white ESC signal wire to the signal source output. Connect the black (ground) ESC wire to a ground pin on the signal source device. Send a PWM signal of 1500 microseconds (µs) to initialize the ESC. The ESC must be initialized before it can accept any other throttle signals. When the ESC is initialized you should hear two more beeps. The first beep indicates the ESC detects a throttle signal and the second beep indicates that the correct 1500 µs signal is detected and the ESC is now fully initialized.

After the ESC is initialized, sending a PWM signal from 1100 µs to 1900 µs will control the throttle of the thruster:

- * 1500 µs to stop the thruster
- >1525 µs to 1900 µs for forward thrust
- <1475 µs to 1100 µs for reverse thrust

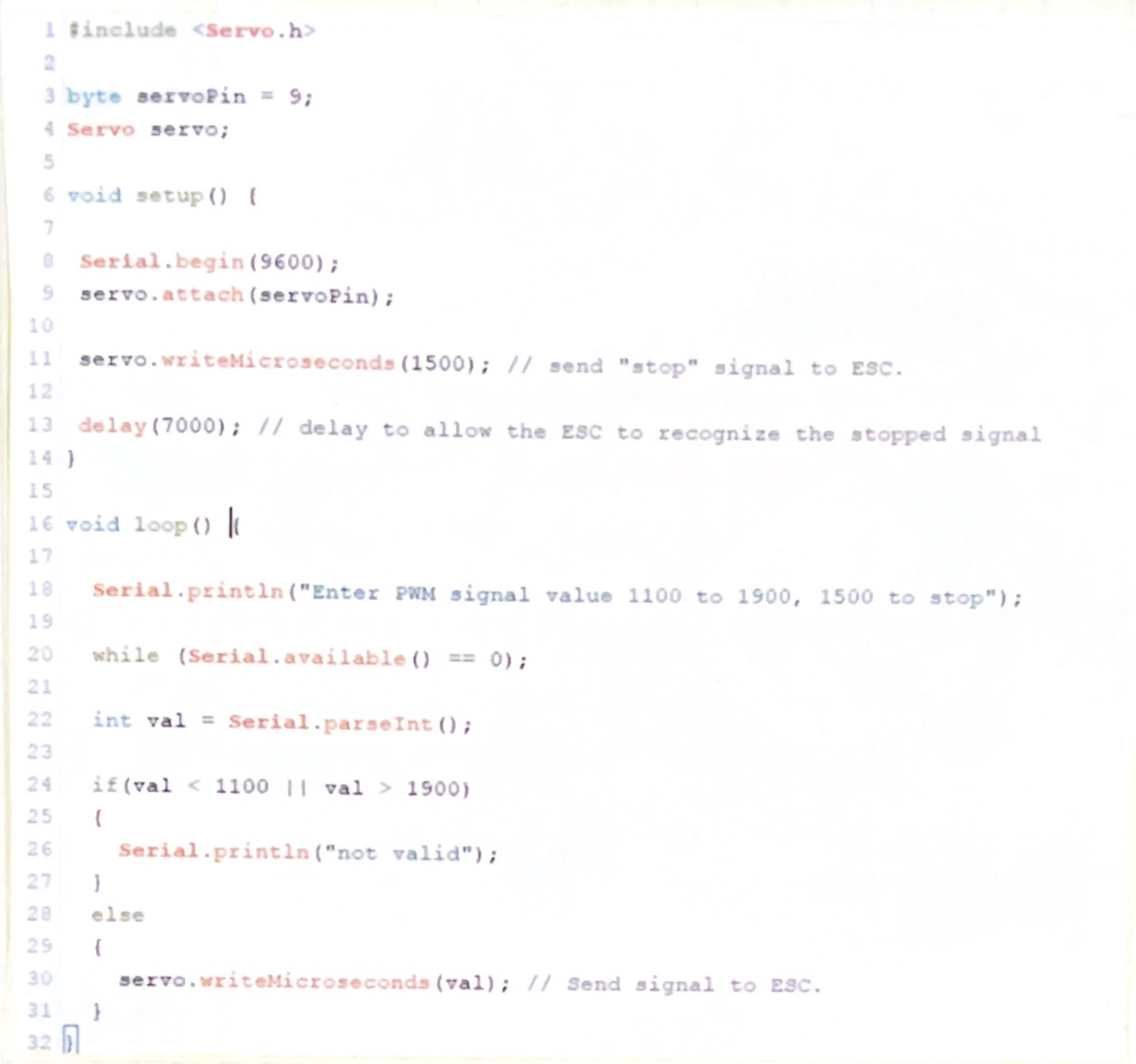


We used a 14.8V Li-Po battery as our power source.

To provide the signal to the ESC, we treat it as a servo object in code.

07/30/2023
Wesley
Cooke

Blue Robotics T200 Underwater Thruster



```
#include <Servo.h>

byte servoPin = 9;
Servo servo;

void setup() {

 Serial.begin(9600);
 servo.attach(servoPin);

 servo.writeMicroseconds(1500); // send "stop" signal to ESC.

 delay(7000); // delay to allow the ESC to recognize the stopped signal
}

void loop() {

  Serial.println("Enter PWM signal value 1100 to 1900, 1500 to stop");

  while (Serial.available() == 0);

  int val = Serial.parseInt();

  if(val < 1100 || val > 1900)
  {
    Serial.println("not valid");
  }
  else
  {
    servo.writeMicroseconds(val); // Send signal to ESC.
  }
}
```