

[Get started](#)[Open in app](#)

To make Medium work, we log user data.

By using Medium, you agree to our

[Privacy Policy](#), including cookie policy.

## Cyrus Beh

13 Followers

[About](#)[Follow](#)

# Donkey Car — Part One: Understanding RC Car Control Protocol

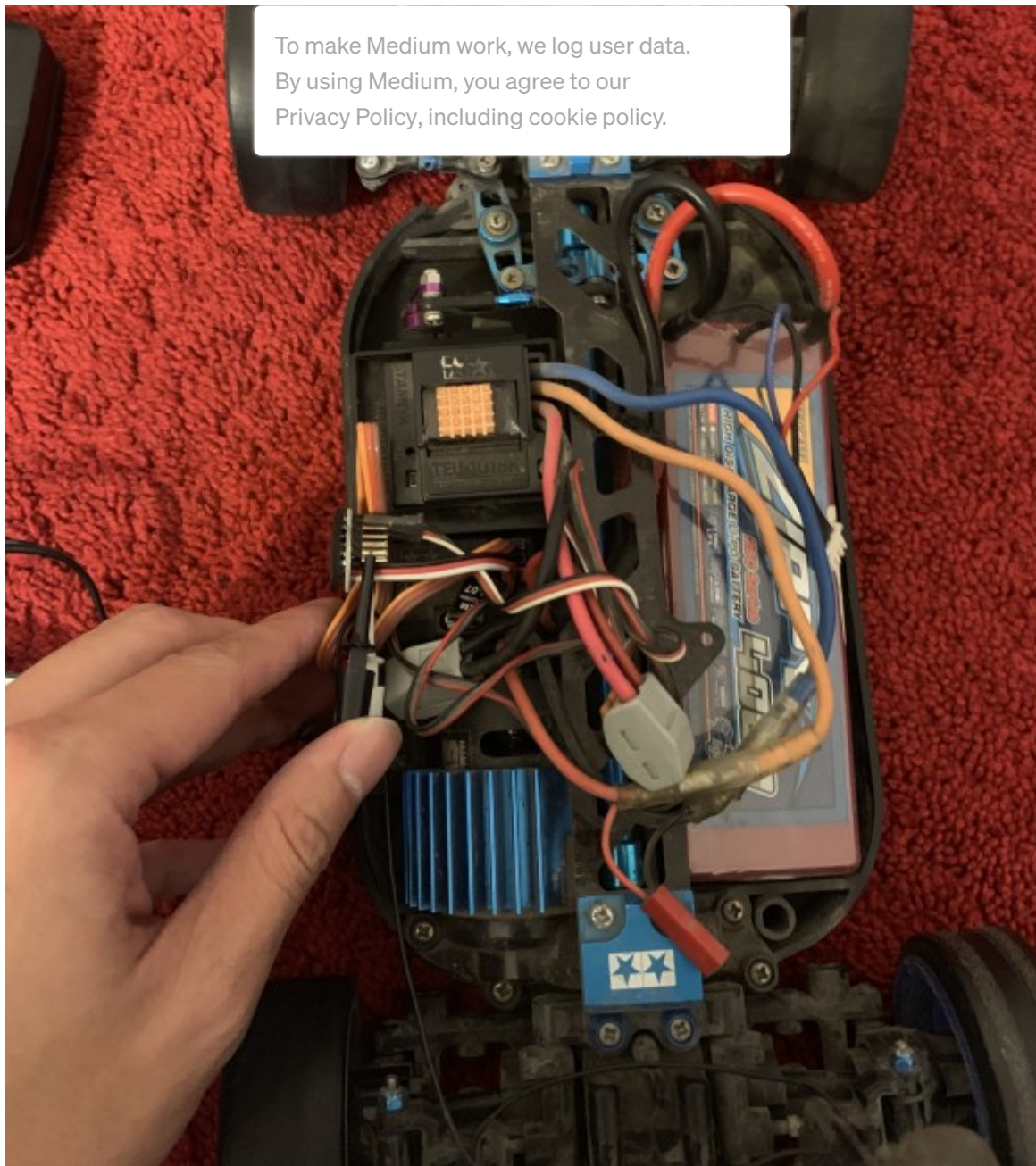
[Cyrus Beh](#) May 20, 2019 · 2 min read

As I mentioned in the [Prelude](#), I am planning to build a Donkey Car using an RC car (in this case, a 1/10th scale Tamiya TT-01D) and the Nvidia Jetson Nano. In addition, instead of controlling the car using the Nano, I would like to use the original control since the car is much more responsive this way. One little caveat straight off the bat is that the TT-01D is a *drift* car, the very antithesis of good control. It remains to be seen how wild the images from the camera will be, and how well it will train. Worst case scenario? I'll just drive it extremely slowly.

## Understanding Control Protocol

A key reason why the RC cars such as the Tamiya are such ideal vehicles for this work is that they use 3-pin, PWM connectors to control the throttle and steering servos. These are the same types of servo motors used in robotics, which is an obvious and early application for machine learning. The RC transmitter sends signals through a receiver, that connects to the electronic speed control (ESC) and the steering servo. To determine the settings needed to drive the car, I used a servo splitter to hijack the signal on its way to the ESC and steering servo.





The signal is then analyzed on my DSO Nano Oscilloscope. You can see in the video that the frequency of the signal is 66.6Hz, and the voltage of the signal pin is about 5–5.2V. In addition, the duty cycle was determined to range from 7.2% for full reverse, to 10.2% at neutral, and 12.6% at full throttle. These values are shifted with the throttle trim. This is known as the Pulse Width Modulation (PWM), where the width ranges from around 1.1ms to 1.9ms.

To make Medium work, we log user data.  
By using Medium, you agree to our  
Privacy Policy, including cookie policy.

One thing to note is that the battery was almost flat, and this has a big impact on the measured values (frequency, voltage, *etc* which you can see at the tail end of the video) and is presumably the reason why RC cars with batteries on their last legs tend to sputter along — the lowered voltage results in spontaneous and sporadic triggering of the ESC. I'm currently recharging the battery and will provide an update on the stability of the signal with a fresh battery.

## Conclusion

This little exercise was useful for understanding the ranges of values that I will expect to see from the receiver. This will in turn help with the logging of the data for steering and throttle measurements during the data collection process. Next up, I will start work on the Jetson Nano installation. Stay tuned!

[Remote Control](#)   [Autonomous Vehicle](#)   [Donkey Car](#)   [Jetson](#)   [Tamiya](#)

[About](#)   [Write](#)   [Help](#)   [Legal](#)

Get the Medium app

