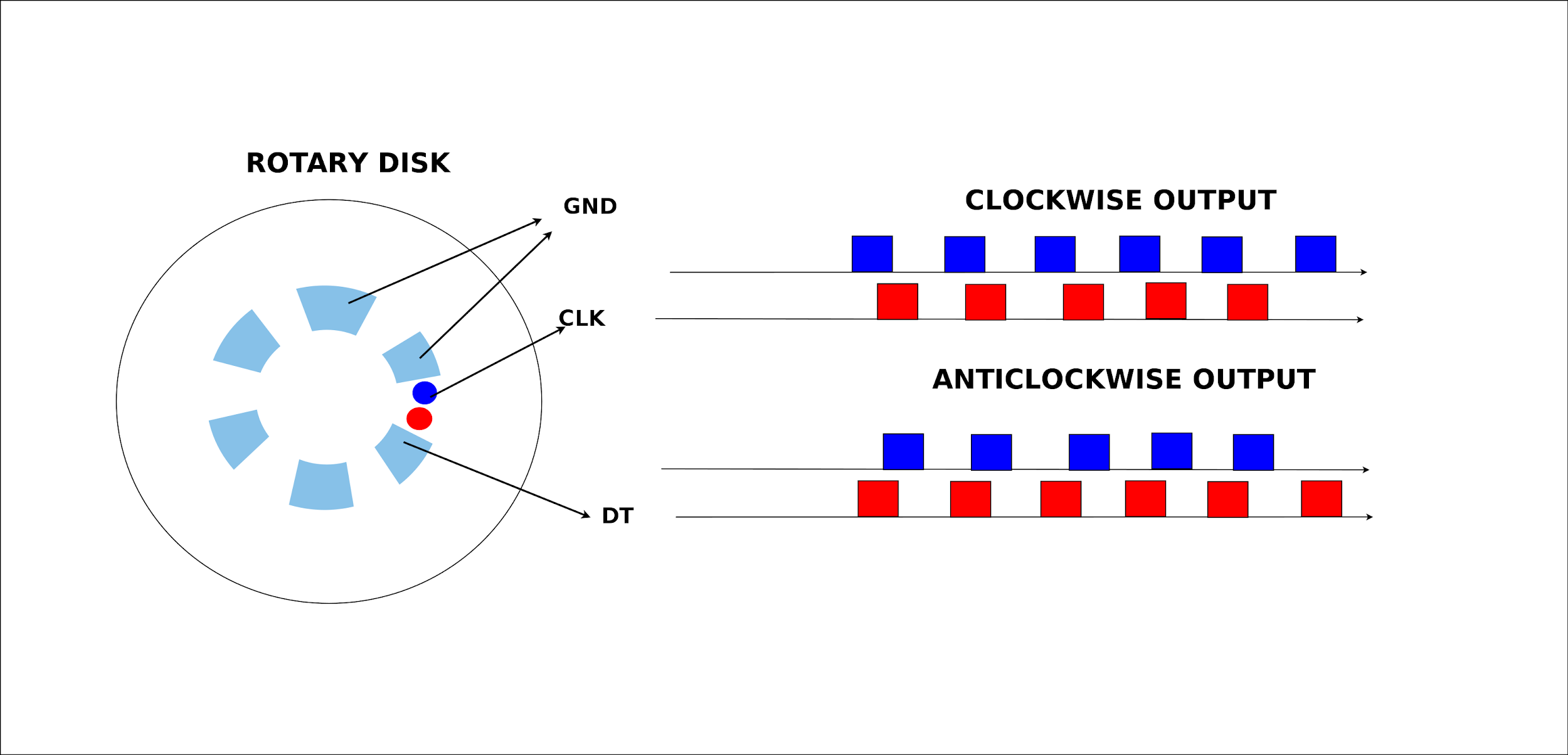
Encoder basics: <http://codelectron.com/rotary-encoder-with-raspberry-pi/>

Speed measurement:<https://www.raspberrypi.org/forums/viewtopic.php?t=151465>

PID basics: <https://medium.com/luosrobotics/an-introduction-to-pid-control-with-dc-motor-1fa3b26ec661>



Currently:

Use speed.py to measure how fast the wheel is rotating in terms of falling edges per second

The accuracy/sensitivity changes according to the bouncetime parameter in the gpio.add\_event\_detect call

We can enforce speed controlling on each wheel individually to ensure straight movement

<https://robotics.stackexchange.com/questions/1711/approach-to-using-pid-to-get-a-differential-robot-driving-straight>

Why it may not work: I am running the full GUI version. The motors and encoders that I am using are 11 pole quadrature encoders on the motor axle with a 31:1 gear box to the wheels. So each full revolution of the wheel gets 11x4x31=1364 signals. So at even at a medium robot speed of 60RPM the signals are already sub 1ms, then I need 2 motors to be able to drive the robot and so double the amount of signals then the fact that both motors can output a signal at exact the same time. This tends to mean it misses counting some signals and when it misses a signal then it meas you miss counting 2 signals as it becomes out of pattern so you need to get another signal to know the pattern again.

Now the biggest reason that I want to be able to count encoders properly is to be able to drive the robot accurately. To get a robot to drive in a straight line is a very difficult thing to do without using stepper motors. If 1 wheel is slightly out of scyn with the speed of the other wheel then the robot will change direction a little and drive off at an angle. This happens if the counter misses signals or the PID is tunes finely

This is what I found comparing software encoder counters compared to hardware counters si when I print both to the screen it doesn't take a pulse rate very high before the software counter returns a lower vlause than the hardware counter.

Although ardunio counts fine without missing and pulses when I tried to use an ESP chip to count in the same way I found even the ESP microcontrollers missed counts becvause they were maintaining the WiFi that causes latency.