Technical Note: Encoder Speed Calculation and Filtering

This technical note explains the steps for calculating DC motor speed (RPM) using pulses obtained from an optical encoder on an STM32F446RE microcontroller and for filtering measurement noise. The described method aims to ensure that the wheel speed data published via micro-ROS is stable and reliable.

1. Encoder Pulse Counting

Signals obtained from a 20-slot optical encoder disk mounted on the DC motor shaft are counted using the STM32's TIM2 timer, which has 32-bit resolution. The timer is configured in External Clock Mode 1, so the counter is incremented on each rising edge of the encoder signal. Thanks to the use of a 32-bit counter, reliable pulse counting can be achieved during long-term operation without errors caused by counter overflow.

2. Pulse Difference Calculation

Speed measurement is performed with a 100 ms sampling period. At each measurement instant, the current TIM2 counter value is read and the difference from the counter value recorded in the previous measurement is calculated. Since the counter variables are of type uint32_t, the difference calculation remains correct even if the counter wraps around from its maximum value back to zero. This approach eliminates the need for additional overflow handling.

3. RPM Calculation

The calculated pulse difference is converted into mechanical speed using the number of slots on the encoder disk. Considering the 100 ms sampling period, the number of revolutions per second is determined and then scaled to obtain the raw RPM value. At this stage, the calculated speed is not yet filtered and may contain instantaneous fluctuations.

4. Moving Average Filtering

To reduce sudden spikes and measurement noise in the raw RPM data, a moving average filter is applied. The most recent five measurements are stored in an array using a FIFO mechanism. With each new measurement, the oldest value is updated, and the arithmetic mean of all values in the array is calculated to obtain the filtered RPM value. This method ensures that the speed data is smoother, more stable, and physically meaningful.

5. Output Data

The filtered speed value is published using micro-ROS as a Float32 message on the wheel_speed topic. This allows the encoder-based speed information to be directly utilized by ROS 2–based control and odometry algorithms.