

More Motor Control

- Last time, we saw how to use an H-bridge to drive the motor forward and backward
- ↳ Unfortunately, we had no way to control speed
- ↳ we saw two lectures ago that we can't control motors with pulsed switching
- ↳ we can use a method called pulse width modulation to implement speed control of motors

- we could use analog inputs, but the Nucleo we are using only has 2 DACs, not great for this!

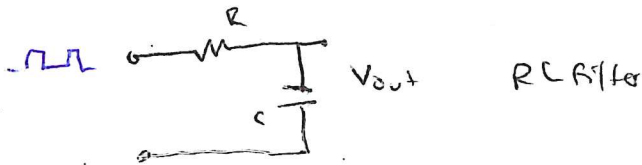
PWM signals



- by changing the period, the average value can be changed

$$\text{Duty cycle} = \frac{t_{on}}{T} \times 100\% = D$$

- average value can be found by getting a low pass filter (not 2 mgs)



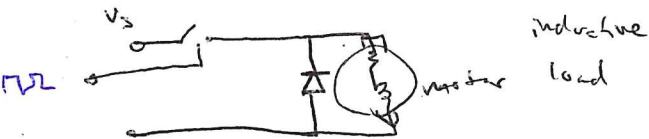
- Two primary benefits

- ↳ easier to send DC signals this way (port availability)

- ↳ signals are still digital, so you can send between devices easily

MBED API Functions for PWM

| Pwm object | pwm object |
|-------------|------------------------------|
| write (or=) | output duty cycle 0.0 to 1.0 |
| read | current settings of pwm out |
| period | [s, float] |
| pwm_dms | [ms, int] |
| period_us | [us, int] |



pulsewidth
 pulsewidth_ms
 pulsewidth_us
 pulsewidth_in_seconds float
 pwm in ms, int
 pwm in us, int

DEX: Key terms:

include "mbed.h"

Pwm out pwm(D2);

int main() {
while(1) {

- pwm.period(0.010); // 10ms
 - pwm = 0.2; // 20% duty cycle

}
 }

DEX: modify your program from last time to use PWM instead of digital outputs