Zephyr: Pulse Width Modulation (PWM)

Medical Electrical Equipment (BME590L)

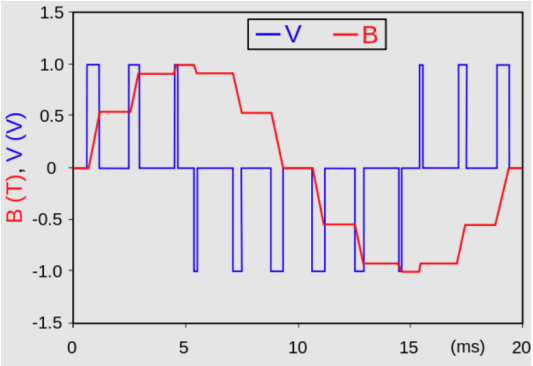
Mark L. Palmeri, M.D., Ph.D.

March 20, 2023

Duke | BME

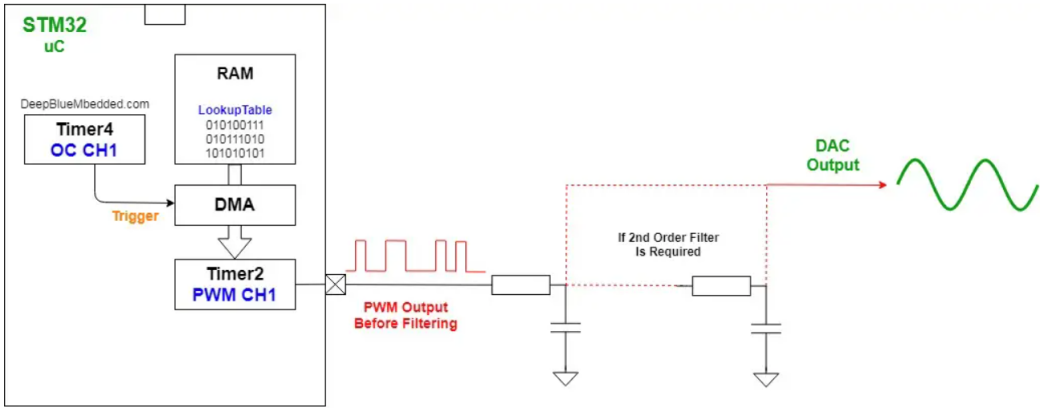
Pulse Width Modulation (PWM)

- Most MCUs don't have a native DAC.
- Much easier to output digital `LOW`/`HIGH`.
- Modulate timing of pulses and combine with LPF to create "smooth" analog output.



Duke | BME

Sine from PWM Block Diagram



https://deepbluembedded.com/stm32-change-pwm-duty-cycle-with-dma-for-sine-wave-generation/

Duke | BME

PWM: Common Uses

- Adjust LED brightness
- Control motor speed
- Control servo motor direction
- Analog output
- Data encoding

prj.conf

```
CONFIG_PWM=y
```

Duke | BME

DT: Root

```
aliases = {
    drv1 = &drv1;
    drv2 = &drv2;
};
pwm_motor {
    compatible = "pwm-leds";
    drv1: drv_1 {
        pwms = < &pwm0 0 PWM_MSEC(1) PWM_POLARITY_NORMAL>;  // 0 - channel
        label = "MOTOR DRIVE 1";
    };

    drv2: drv_2 {
        pwms = < &pwm0 1 PWM_MSEC(1) PWM_POLARITY_NORMAL>; // 1 - channel
        label = "MOTOR DRIVE 2";
    };
};
```

DT: PWM0

```
&pwm0 {
    compatible = "nordic,nrf-pwm";
    reg = <0x4001c000 0x1000>;
    interrupts = <28 NRF_DEFAULT_IRQ_PRIORITY>;
    status = "okay";
    #pwm-cells = <3>;
    pinctrl-0 = <&pwm0_default>;
    pinctrl-1 = <&pwm0_sleep>;
    pinctrl-names = "default", "sleep";
};
```

Duke | BME

DT:Pinctrl (used for all non-GPIO IO)

```
&pinctrl {
    compatible = "nordic,nrf-pinctrl";
    status = "okay";
    pwm0_default: pwm0_default {
        group1 {
            psels = <NRF_PSEL(PWM_OUT0, 0, 28)>, // P0.28, channel 0
                    <NRF_PSEL(PWM_OUT1, 0, 30)>; // P0.30, channel 1
            nordic,invert;
        };
    };
    pwm0_sleep: pwm0_sleep {
        group1 {
            psels = <NRF_PSEL(PWM_OUT0, 0, 28)>, // P0.28, channel 0
                    <NRF_PSEL(PWM_OUT1, 0, 30)>; // P0.30, channel 1
            low-power-enable;
        };
    };
};
```

main: preamble

```
// load in the Zephyr library
#include <zephyr/drivers/pwm.h>

// define structs based on DT aliases
static const struct pwm_dt_spec mtr_drv1 = PWM_DT_SPEC_GET(DT_ALIAS(drv1));
static const struct pwm_dt_spec mtr_drv2 = PWM_DT_SPEC_GET(DT_ALIAS(drv2));
```

Duke | BME

main: check PWM device ready

```
if (!device_is_ready(mtr_drv1.dev))  {
 LOG_ERR("PWM device %s is not ready.", mtr_drv1.dev->name);
 return -1;
}
```

Duke | BME

main

```
err = pwm_set_pulse_dt(mtr_drv1, mtr_drv1->period/2); // 50% duty cycle
if (err) {
    LOG_ERR("Could not set motor driver 1 (PWM0)");
}
```

The pulse length (duty cycle) can be changed "on the fly", but only changes at the next period.

https://docs.zephyrproject.org/latest/hardware/peripherals/pwm.html#c.pwm_set_pulse_dt

Duke | BME

References

- Zephyr Docs: PWM
- PWM: Pulse Width Modulation: What is it and how does it work?
- Zephyr Samples: PWM Blinky
- Nordic: PWM