RuggedBoard PWM HAL Library (Dual Channel) Documentation

This document describes the usage of the RuggedBoard dual-channel PWM HAL (Hardware Abstraction Layer) library.

Library Name: rb_pwm

Header File:

#include "rb_pwm.h"

Channel Usage:

- Channel 0: Servo (high voltage PWM signal)
- Channel 1: LED (low voltage PWM signal)

Available Functions:

1. int rb_pwm_export(int channel);
   - Exports the given PWM channel.
   - Returns 0 on success, -1 on failure.

2. int rb_pwm_unexport(int channel);
   - Unexports the given PWM channel.
   - Returns 0 on success, -1 on failure.

3. int rb_pwm_enable(int channel);
   - Enables PWM output on the specified channel.

- Returns 0 on success, -1 on failure.

4. int rb_pwm_disable(int channel);

- Disables PWM output on the specified channel.

- Returns 0 on success, -1 on failure.

5. int rb_pwm_set_period_ns(int channel, int period_ns);

- Sets the PWM signal period in nanoseconds.

6. int rb_pwm_set_frequency(int channel, int frequency_hz);

- Sets the PWM frequency.

- Internally computes period = 1e9 / frequency.

7. int rb_pwm_set_duty_percent(int channel, float percent);

- Sets duty cycle as a percentage of the period.

8. int rb_pwm_set_duty_ns(int channel, int duty_ns);

- Sets duty cycle directly in nanoseconds.

Usage Example: LED Fading (Channel 1)

```
#include "rb_pwm.h"
#include <unistd.h>

int main() {
    rb_pwm_export(1);
    rb_pwm_enable(1);
```

```
    rb_pwm_set_frequency(1, 1000); // 1kHz

    for (int i = 0; i <= 100; i++) {
        rb_pwm_set_duty_percent(1, i);
        usleep(20000);
    }
    for (int i = 100; i >= 0; i--) {
        rb_pwm_set_duty_percent(1, i);
        usleep(20000);
    }

    rb_pwm_disable(1);
    rb_pwm_unexport(1);
    return 0;
}
```

Usage Example: Servo Control (Channel 0)

```
#include "rb_pwm.h"
#include <unistd.h>
#include <stdio.h>

int main() {
    rb_pwm_export(0);
    rb_pwm_enable(0);
    rb_pwm_set_frequency(0, 50); // 50Hz = 20ms period
```

```
    printf("Rotating to 0 degrees...
");
    rb_pwm_set_duty_percent(0, 5.0); // 1ms
    sleep(1);

    printf("Rotating to 90 degrees...
");
    rb_pwm_set_duty_percent(0, 7.5); // 1.5ms
    sleep(1);

    printf("Rotating to 180 degrees...
");
    rb_pwm_set_duty_percent(0, 10.0); // 2ms
    sleep(1);

    rb_pwm_disable(0);
    rb_pwm_unexport(0);
    return 0;
}
```

Notes:

- This library supports multiple channels (e.g., PWM0 for servo, PWM1 for LED).
- Period and duty cycle are handled via /sys/class/pwm/pwmchip0/pwmX/
- Ensure proper pin mapping in the device tree.

Compile with Poky Toolchain:

```
$ source /opt/poky-tiny/2.5.2/environment-setup-cortexa5hf-neon-poky-linux-musleabi
```

$ $CC -I./include -L./build -lpwm test.c -o pwm_test

Makefile Targets:

$ make clean

$ make

Outputs:

- build/libpwm.a: static library (reusable across projects)