Robotics and Embedded Systems

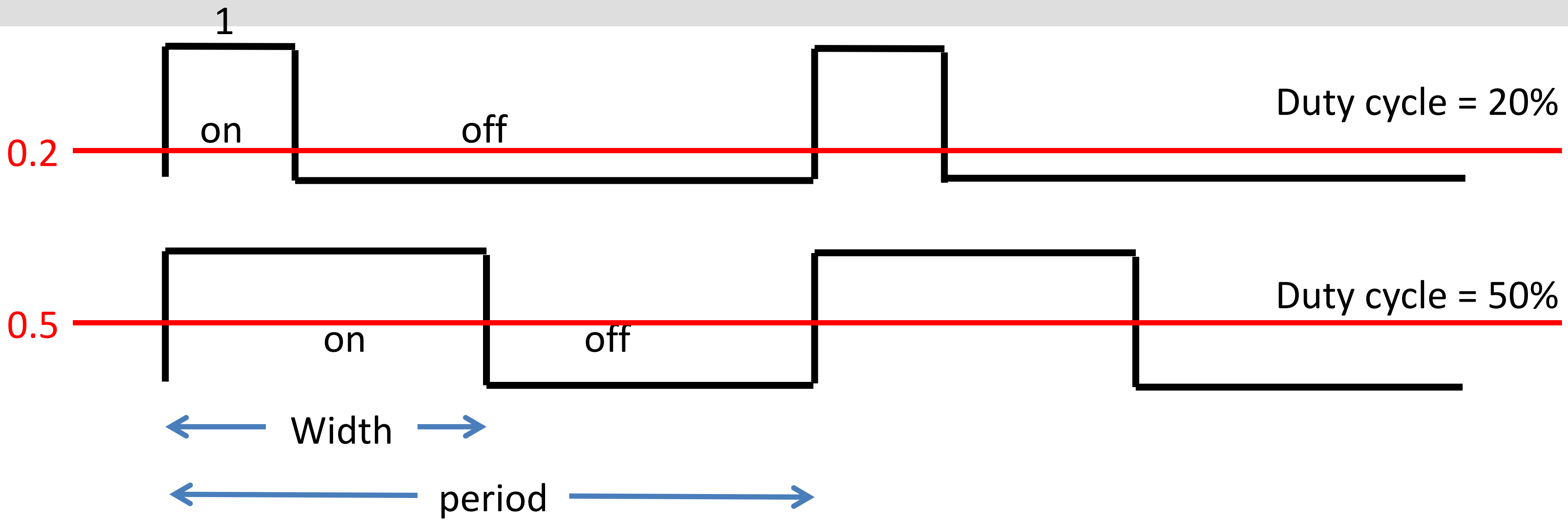
Technische Universität München

Introduction to Pulse Width Modulation (PWM)

What is PWM?

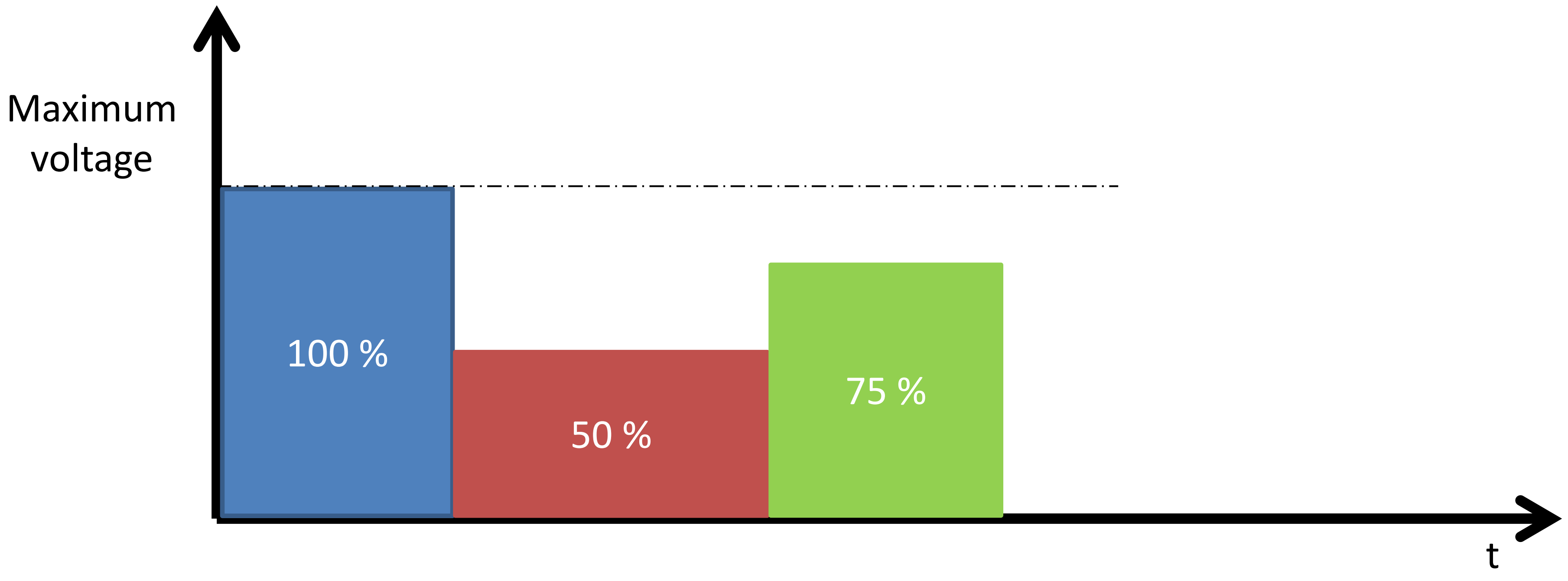
- Output signal alternates between on and off within specified period.
- Control the power received by a device.
- The voltage seen by the load is directly proportional to the source voltage.

What is PWM?



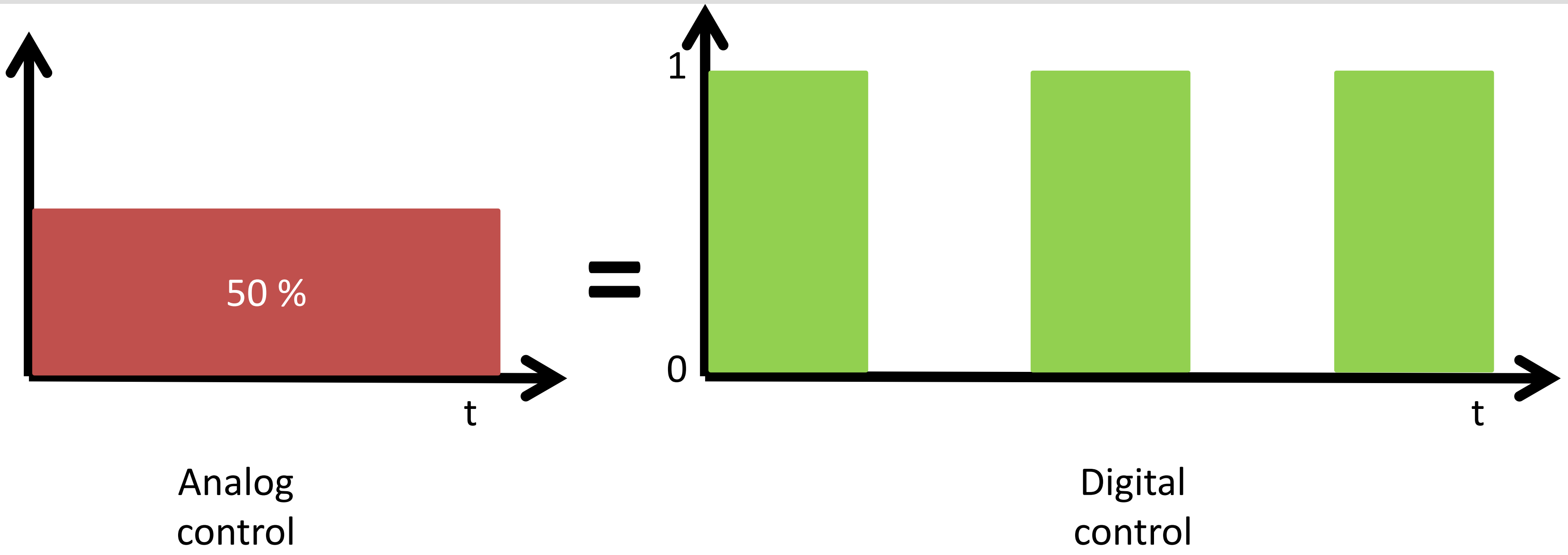
- Depending on the requirement the width of the pulse is modulated (adjusted).
- Duty cycle = $t_{on} / (t_{on} + t_{off})$.

Why PWM?



- Analog voltage control:
  - Voltage can be changed to control the motor speed
  - Can NIOS change voltage ?

Why PWM?



- Digital voltage control:
  - Can only control ‘1’ and ‘0’
  - X% of maximum analog voltage = X% of duty cycle

PWM Control Example

- Disco gate:
  - 100 % open gate = 10 persons per second
  - 50% open gate = 5 persons per second
- Analog control:
  - Open 50 % gate
  - Total how many people can go in 10 seconds?
- Digital control:
  - Open 100 % gate on every odd second (1,3,5,7,9, ..)
  - Total how many people can go in 10 seconds?

Usage of PWM

- Motor Control

- Intensity of LED

How to generate PWM signal?

- Software method
  - Using counter
    - Count to 100 in a loop
    - Set the output value to 1 in the beginning of the loop
    - Set the output value to 0 as soon as the counter reaches the value of required duty cycle.
    - Continue the process
  - Using interrupt
    - Home work
    - Think about the concept

Your tasks

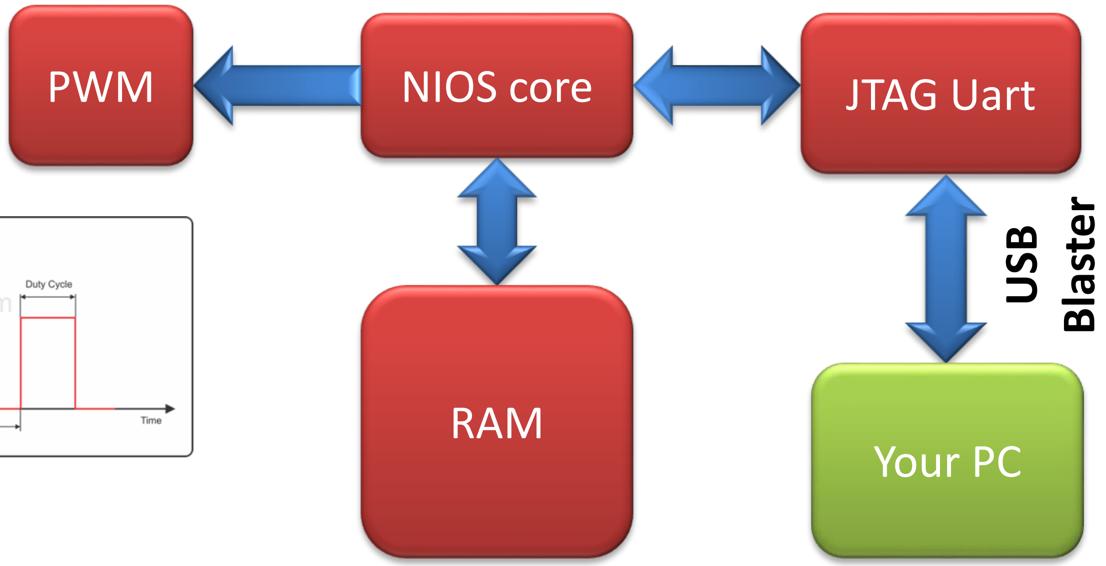
- Create projects in a usual way using provided SOPCINFO file.
- Type the code in your application project.
- Change duty cycle variable and observe the effect on oscilloscope or LED.
- Using oscilloscope, verify the duty cycle.
  - Is it precise?
  - Is it efficient?

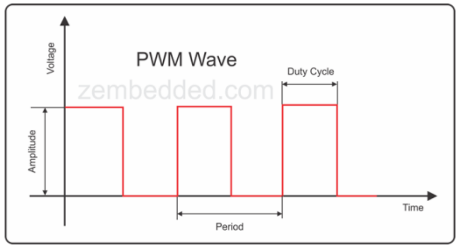
Software PWM

- Output pin:
  - **GPIO_0[0]** ==> *find this pin in the user manual*
  - Using the manual find out the correct pin and observe the resulting PWM on the oscilloscope
  - In C program, use the following instructions to change the output
    - IOWR(**PIO_0_BASE**, 0, 0); // set output 0
    - IOWR(**PIO_0_BASE**, 0, 1); // set output 1
- Control LED (optional):
  - Apply the PWM signal to LED, observe the intensity

Questions

Hardware PWM IP





- Programmable/configurable
- Precise

Hardware PWM IP

- Follow the simple use sequence
  - Enable
  - Configure
  - Modify
  - Disable
- Only one function for controlling two PWM signals

```
motor_setting(phase1, duty1,
              phase2, duty2,
              period, enable);
```

Hardware PWM IP

```
motor_setting(phase1, duty1,
              phase2, duty2,
              period, enable);
```

Common period

Enable mask= 1
Disable = 0

Questions