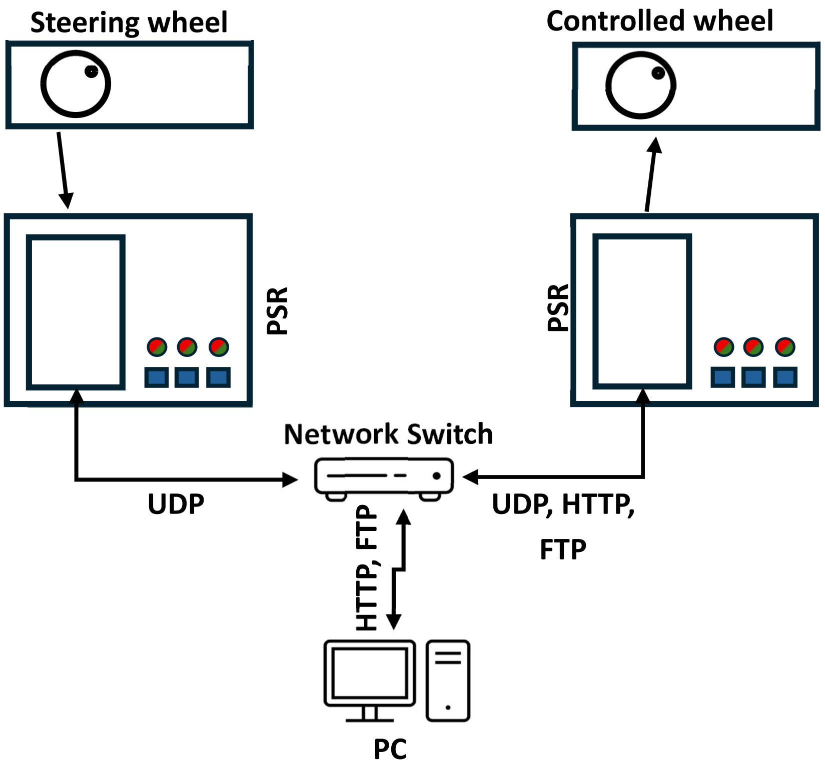
Semester Work Assignment

The goal of the semestral work is to create a digital motor controller. Your program will control the position of the motor according to the set-point given by the position of another motor, moved by hand (steer-by-wire). The set-point will be transferred between the two motor controllers using UDP messages. The actual state of the controller and its history will be published as live graphs over the HTTP protocol.



Instructions

- Work in teams of two students.
- Use version control system GIT. We expect commits from both of you in the repository. Use the FEL Gitlab repository.
- Implement a driver for rotary increment motor sensor. The output value or values should not make any discontinuities when transiting around zero angle position.
- Implement a driver for driving the motor.
- Implement a PID controller to control the shaft position of the Controlled wheel.
- Implement a UDP communication from Steering wheel board to the controlled board to transmit the shaft position setpoint.
- Implement a simple web server for displaying live graphs of the controlled motor. Live graphs served from the web server should display at least:
    - actual motor position (absolute value),
    - requested position (absolute value as received from the reference motor), and

  - current PWM duty cycle (in range –100%, +100%).
- The graphs should show at least 2 seconds of history with time resolution ≤ 5 ms. The graph should be refreshed at least every 100 ms.
- FTP server working.

Assessment

The Semester work assessment comprises the following independently assessed parts:

Motor control program (max. 18 points)

Item	Points
Correct implementation of drivers (sensor and motor)	2
P(ID) or better controller (no ad-hoc solution)	3
Quality of regulation (no oscillations, fast response, minimal steady state error, not affected by FTP file transfer and http webserver)	3
Setpoint specification (choose one of the following)	
- **From reference motor via UDP**	3
- **From webserver**	1
Controller "debugging" (choose one of the following):	
- **Web page with live graphs**	7
- **ASCII "bargraph" of action value (PWM width). Something like:** `-0.5 .....=====|..........`	1

Use of version control system (max. 4 points)

Item	Points
Project is maintained using Git	1
At least one commit every week	1
Commit messages accurately describe the content of the commit	1
No generated files or binary files (e.g. .o, .elf) are added in the repository	1

Programmer's documentation (max. 3 points)

Item	Points
Data-Flow Diagram of your application i.e. how individual components (e.g. IRQ handler, controller, web-server) exchange data.	1
Description of global functions and variables (created by you) including the description of function parameters and return values.	1
Short textual description of the application in the introduction (or on the main HTML page).	1

Hints

Clone the preset project

Clone the preset project from our git repository. Navigate to This PC >> Documents >> STM32CubeIDE. Hold Shift key while pressing right mouse button, pop-up menu will appear. Choose option Open Git Bash here. Then execute following command.

```
git clone https://gitlab.fel.cvut.cz/terenond/psr_threadx_blank.git -b
semesterWork C:/Users/ADMIN/Documents/STM32CubeIDE/PSR_ThreadX_blank
```

The project contains timers configurations for driving the motor and sensing the encoder position on a hardware level.

Reading the motor encoder

To read the value from the encoder read the register:

```
TIM1->CNT
```

Driving the motor

Write values from 0 up to 500 to the registers

```
TIM2->CCR3=0;
TIM2->CCR4=500;
```

To turn the motor one direction and

```
TIM2->CCR3=500;
TIM2->CCR4=0;
```

To turn the motor another direction.