Practical assignment

Topic: Controlling a device over USB CDC using text commands

Goal

Implement the following on an STM32 microcontroller:

- data transfer over USB CDC (virtual COM port);
- a parser for text commands (with syntax similar to SCPI);
- handling a set of text commands and generating appropriate responses;
- control of the on-board user LED (ON/OFF + brightness via PWM).

Parser

1. General

- Commands are received as text strings (ASCII).
- Commands and arguments are processed **case-insensitively**:
  `LED ON`, `led on`, `Led On` are treated as the same command.
- The end of a command can be `\r`, `\n` or `\r\n`.
  The device must correctly handle all of these variants.

Command set

2.1. Identification

- Command:
  `*IDN?`
- Example response:
  `Eltesta,STM3240G,1234,1.0`

2.2. LED state

- Commands:
  - `LED ON` / `LED 1` - turn the LED on;
  - `LED OFF` / `LED 0` - turn the LED off;
  - `LED?` - return the current state: `On` or `Off`.

2.3. LED brightness (PWM)

- Set command:
  `LED:BRIGHT <0..100>` - set brightness in percent (0..100%).

- Query command:
  `LED:BRIGHT?` - return the current brightness as an integer value (0..100).

Notes

(optional)

- To exchange commands with the microcontroller, you can use the **Termite** terminal emulator.
- To analyze/debug USB traffic, you can use **Wireshark**.