**Part 1:**

Code in the encoder.cpp file in method getSpeed().

**Part 2:**

|  |  |
| --- | --- |
| Measures | Time to get to 63% max Speed in ms |
| 1 | 45.0 |
| 2 | 44.7 |
| 3 | 43.6 |
| 4 | 43.2 |
| 5 | 44.3 |
| 6 | 43.7 |
| 7 | 43.2 |
| 8 | 44.6 |
| 9 | 44.6 |
| 10 | 44.6 |
| Average / τ | 44.15 |

**Part 3:**

minimum update rate =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Timing requirements** | **method to meet the time-critical task** | **Resources (timer, interrupts….)** | **Validation** |
| Counting encoder pulses | 664.45 μ𝑠 (encoder pulse period (see Project 1)) | Encoder.position() and counter variable | Interrupt activates the encoder | Counter is printed |
| Computing speed using a  stable time base | 664.45 μ𝑠 (encoder pulse period (see Project 1)) | Measured time between to highs on the encoder | Interrupt activates the encoder | getspeed() is showing the same value as set in setspeed() from Part 4 |
| Updating the control output at a required rate | Minimum update rate ca. 227 Hz | Measuring with 500 Hz | delay(2) from Ardunio.h | Not needed |
| Storing and transmitting the response to the laptop. | Response need to be as fast as the Updating rate | Output on the monitor | serial.print() | Output on the monitor |
| Stable PWM output at a suitable rate, observing | 2.27 kHz (10\* update frequency) | Method Analog.out.init | Timer module of microcontroller | Oscilloscop |

**Part 4:**

Reference value (w\_ref), actual value (w) and PWM value (duty) with a Kp of 7.  
Kp is 7 because Kp of 7.5 leads to stocking in the motor rotation.  
Here w is equivalent to the rotational speed .

Monitor output:

w\_ref: 8.00, w: 7.48, duty: 0.06  
w\_ref: 8.00, w: 8.16, duty: 0.01  
w\_ref: 8.00, w: 6.60, duty: 0.83  
w\_ref: 8.00, w: 6.64, duty: 0.71  
w\_ref: 8.00, w: 7.12, duty: 0.50  
w\_ref: 8.00, w: 6.88, duty: 0.67  
w\_ref: 8.00, w: 6.56, duty: 0.74  
w\_ref: 8.00, w: 6.78, duty: 0.67  
w\_ref: 8.00, w: 6.95, duty: 0.63  
w\_ref: 8.00, w: 6.84, duty: 0.72  
w\_ref: 8.00, w: 6.90, duty: 0.64  
w\_ref: 8.00, w: 6.80, duty: 0.68  
w\_ref: 8.00, w: 6.64, duty: 0.68

Step Response:

Response to motor load:

**Increasing load**

**Increasing load**

After shortly increasing friction on the tire at 58 ms, the PWM firstly goes down and then tries to correct the value to the reference speed. Therefore, the PWM turns to 1. After exceeding the reference speed, the PWM decreases. Same on the second load at 72 ms. After exceeding the reference speed in 76 ms the controller tries to stay at the reference speed.

Github Link: <https://github.com/Marlenexyz/EMBE-Group>

YouTube Link: <https://www.youtube.com/watch?v=DSs6kUWA5Ww>