Motor Position Control

System Description

Bonden i Nol
hakanbrolin@hotmail.com
Rev PA1, 2017-12-30, Håkan Brolin

Table of Contents

1 Introduction....3
1.1 Purpose....3
1.2 Platform....3
1.3 Resources....3
2 PID controller....4
2.1 Theory....4
2.2 References....4
2.3 Motor position controller....5
3 Development tools....6
4 Software....7
5 Components....8
5.1 DC motor....8
5.2 DC motor driver....8
6 MCU....9
6.1 SAM9-L9260....9
6.2 External connector EXT....9
6.3 External connector AEXT....9
7 PID controller test#1....10
7.1 Conditions....10
7.2 PID controller test#1.1....11
7.3 PID controller test#1.2....12
8 PID controller test#2....13
8.1 Conditions....13
8.2 PID controller test#2.1....14
8.3 PID controller test#2.2....15

1 Introduction

1.1 Purpose

This project investigates how to control the position of a DC motor shaft using PID controller.

1.2 Platform

Controller application is implemented in C and executes on a ARM9 32bit micro-controller.

A “bare metal” approach was selected to get low latency and overhead, and to achieve high real-time performance. This decision made the application more complex then when using an operating system (Linux). The learning curve is steeper and the lack of support of system resources makes it harder to get up and running in a small amount of time.

But the “bare metal” approach makes it possible to customize the application with a minimum use of system resources. It's also an interesting way to get to know the hardware and increases the knowledge about various aspects of the system, like start-up sequence, interrupt handling and more.

1.3 Resources

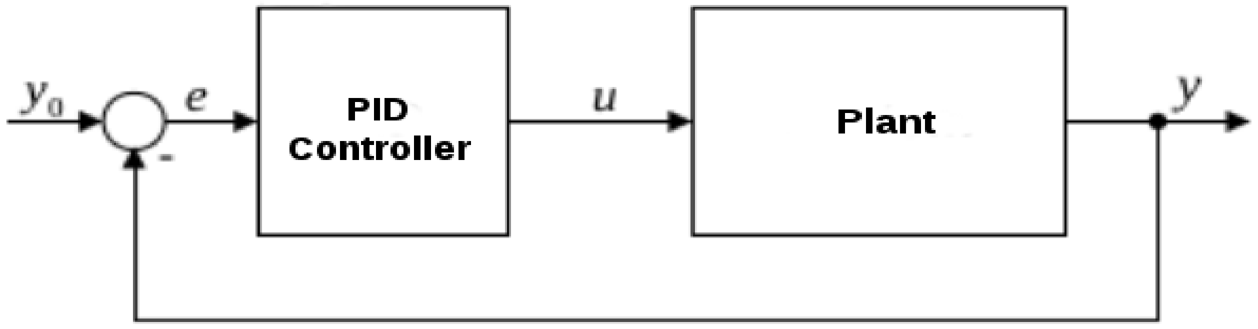
All resources for this project, including source code and documentation can be found at:

https://github.com/emwhbr/dcmotor

2 PID controller

2.1 Theory

A closed-loop embedded PID control system:



The PID controller function is described as:

$$u(t) = \mathrm{MV}(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{de(t)}{dt}$$

The discrete form of the algorithm is:

$$u(n) = K_p e(n) + K_i \sum_{k=0}^{n} e(k) + K_d \left(e(n) - e(n-1)\right) \qquad K_i = \frac{K_p T}{T_i} \qquad K_d = \frac{K_p T_d}{T}$$

The derivate on error *e(t)* is identical to the negative of the derivate on process variable *y(t)*, except when set point changes. When set point changes, derivate on *e(t)* results in an undesirable action on controller output which is called the “derivate kick”. Basing the derivate term on the process variable only results in an improved PID controller.

This controller lacks the impact of the “derivate kick” when the set point changes:

$$u(n) = K_p e(n) + K_i \sum_{k=0}^{n} e(k) - K_d \left(y(n) - y(n-1)\right)$$

This formula is used in the motor position controller software.

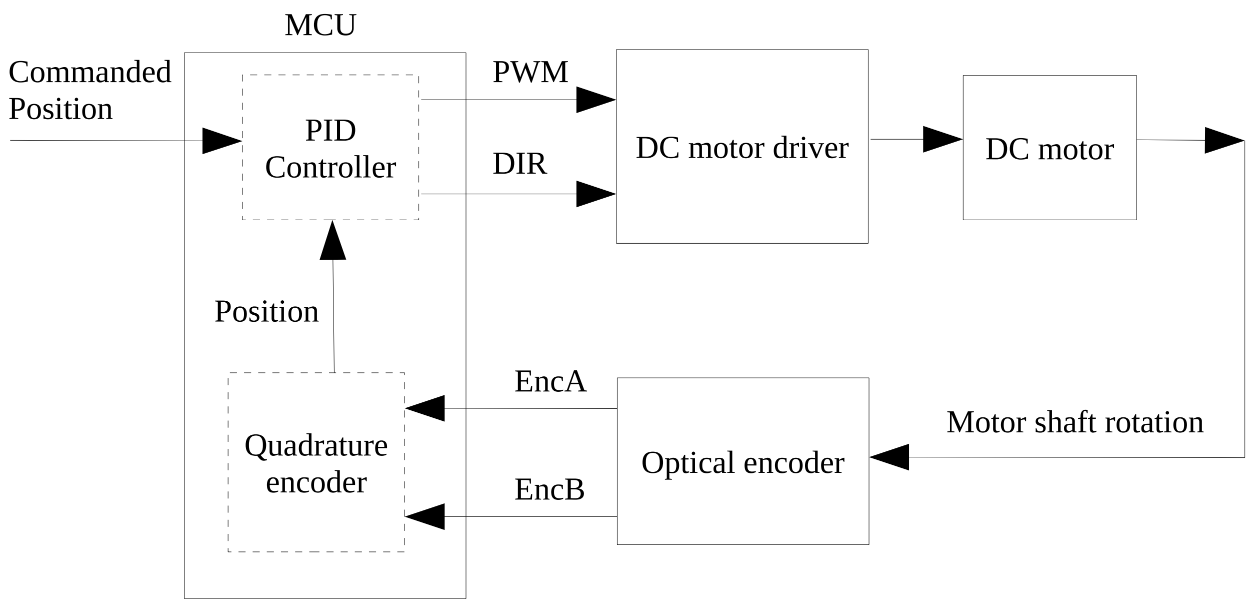
2.2 References

References on PID controller theory and how to apply them in an embedded system:

- PID control and derivate on measurements.
  http://controlguru.com/pid-control-and-derivative-on-measurement
- PID without a PhD, Tim Wescott.
  http://www.wescottdesign.com/articles/pid/pidWithoutAPhd.pdf
  http://m.eet.com/media/1112634/f-wescot.pdf

2.3 Motor position controller

The motor position controller system is implemented according to figure below:



The DC motor driver hardware controls the direction and speed of the DC motor.

The output from the PID controller software is a PWM duty cycle value ranging from -65000 .. 65000. The absolute value of this output is feed as a PWM signal to the motor driver hardware. Positive values results in a clockwise direction signal and negative values results in an anti-clockwise signal being generated.

Motor shaft rotation is transformed by the optical encoder hardware into two output channels that indicates both position and direction of the rotation. The quadrature encoder software transforms these signals back to an absolute position. This is an interrupt driven process where both encoder outputs generates an interrupt to the MCU.

The PID controller software executes at a constant frequency by a timer interrupt. This means that the sampling of the actual position is done at a consistent interval with a minimum of variations.

3 Development tools

TBD

4 Software

TBD

5 Components

5.1 DC motor

Brushed DC (12V) motor with gearbox and quadrature encoder: Pololu, part#3240.

https://www.pololu.com/product/3240

Specifications:

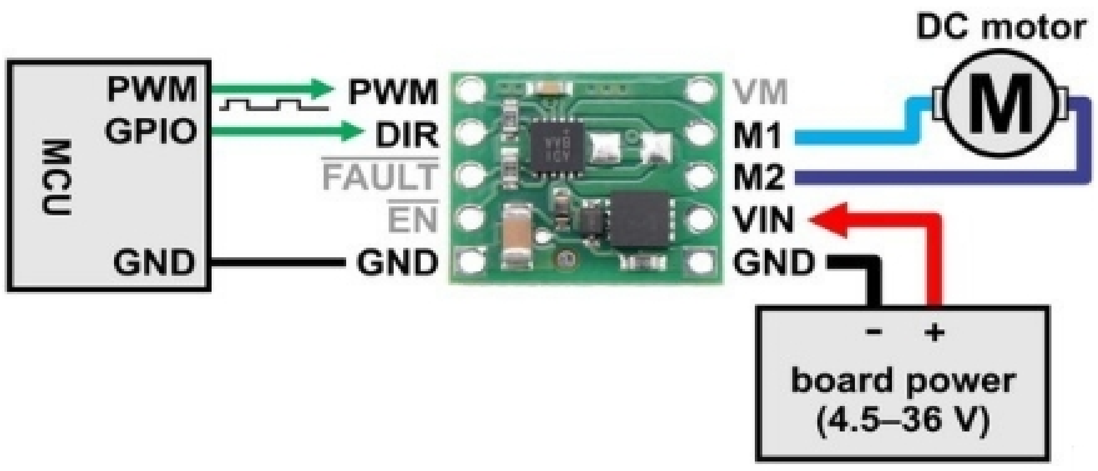
Free-run current:	200 mA (stall current: 2100 mA)
Free-run speed:	220 rpm (3.67 rev/s, 273ms/rev)
Gear ratio:	34.014:1
Encoder:	48 CPR, quadrature (34.014 x 48 = 1632,672 counts per revolution)

5.2 DC motor driver

Motor driver carrier board for MAX14870: Pololu, part#2961.

https://www.pololu.com/product/2961

Application circuit:



6 MCU

6.1 SAM9-L9260

SAM9-L9260 development board (Olimex).

6.2 External connector EXT

Pin	MCU pin	MCU function	Application
1, 2	-	-	3.3V
4	-	-	5V
39, 40	-	-	GND
5	PB0	PIO Controller B Peripheral B TC3, TIOA3	PWM output to MAX14780 PWM input.
7	PB1	PIO Controller B PIO, output	Motor direction.
9	PB2	PIO Controller B PIO, output	Debug pin#1.
11	PB3	PIO Controller B PIO, output	Debug pin#2.
13		PIO Controller B PIO, input	Encoder A input. Interrupt enabled.
15		PIO Controller B PIO, input	Encoder B input. Interrupt enabled.

6.3 External connector AEXT

Pin	MCU pin	MCU function	Application
1	GNDANA	Analog ground	GND
2	-	-	-
3	PC0	PIO Controller C Peripheral A, AD0	Analog input channel.
4	-	-	-
5	ADVREF	Analog reference	3.3V
6	VDDANA	Analog power supply	3.3V

7 PID controller test#1

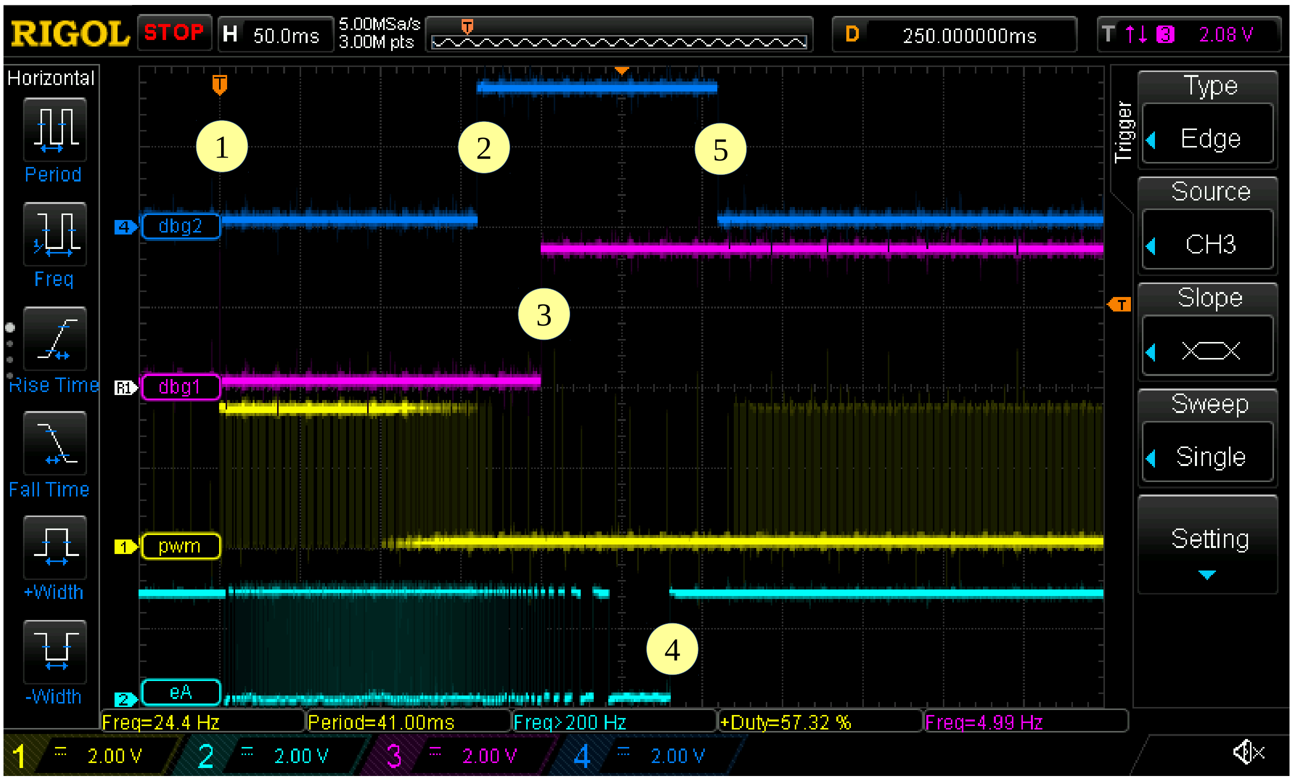
7.1 Conditions

- PID controller frequency: 100Hz (T=10ms)
- PID parameters: P=250, I=0, D=700

7.2 PID controller test#1.1

Conditions:

- Motor shaft position set point: 0x330 = 816 (180°)



H 50.0ms | 5.00MSa/s 3.00M pts | D 250.000000ms | T 3 2.08 V

Type: Edge | Source: CH3 | Slope | Sweep: Single | Setting

Freq=24.4 Hz | Period=41.00ms | Freq>200 Hz | +Duty=57.32 % | Freq=4.99 Hz

1 2.00 V | 2 2.00 V | 3 2.00 V | 4 2.00 V

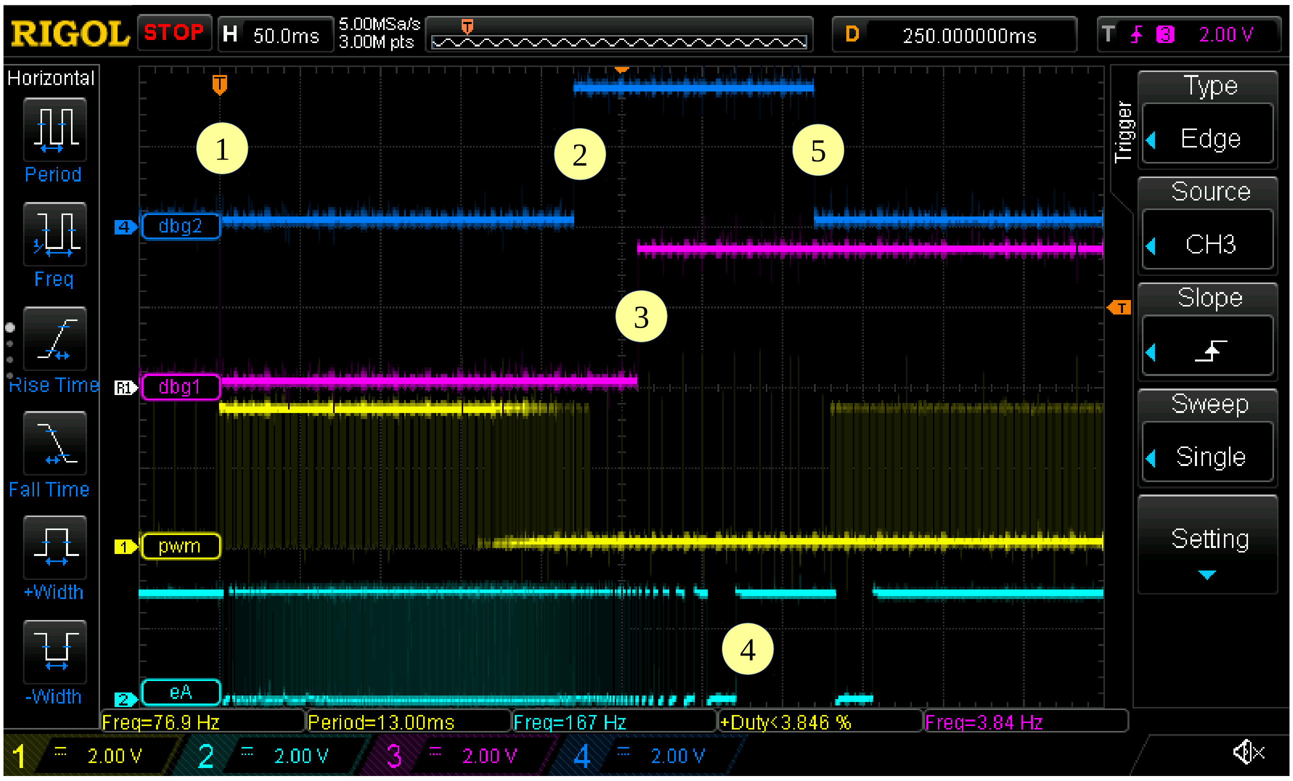
Description:

1. T=0ms
   First PID controller cycle after calibration.
   Motor shaft starts moving 3.2ms after PWM activation.
2. T=160ms
   Entering stop state due to request to change motor direction.
   Duration of stop state is 150ms.
3. T=200ms
   Motor shaft position is within set point ±1.5%.
   This position keeps stable within limits.
4. T=280ms
   Motor shaft completely stopped 120ms after entering stop state.
5. T=310ms
   Leaving stop state with very low PWM duty, not enough to move motor shaft.

7.3 PID controller test#1.2

Conditions:

- Motor shaft position set point: 0x4c8 = 1224 (270°)



Description:

1. T=0ms
   First PID controller cycle after calibration.
   Motor shaft starts moving 4ms after PWM activation.
2. T=220ms
   Entering stop state due to request to change motor direction.
   Duration of stop state is 150ms.
3. T=260ms
   Motor shaft position is within set point ±1.5%.
   This position keeps stable within limits.
4. T=320ms
   Motor shaft completely stopped 100ms after entering stop state.
5. T=370ms
   Leaving stop state with low PWM duty, just enough to move motor shaft.

8 PID controller test#2

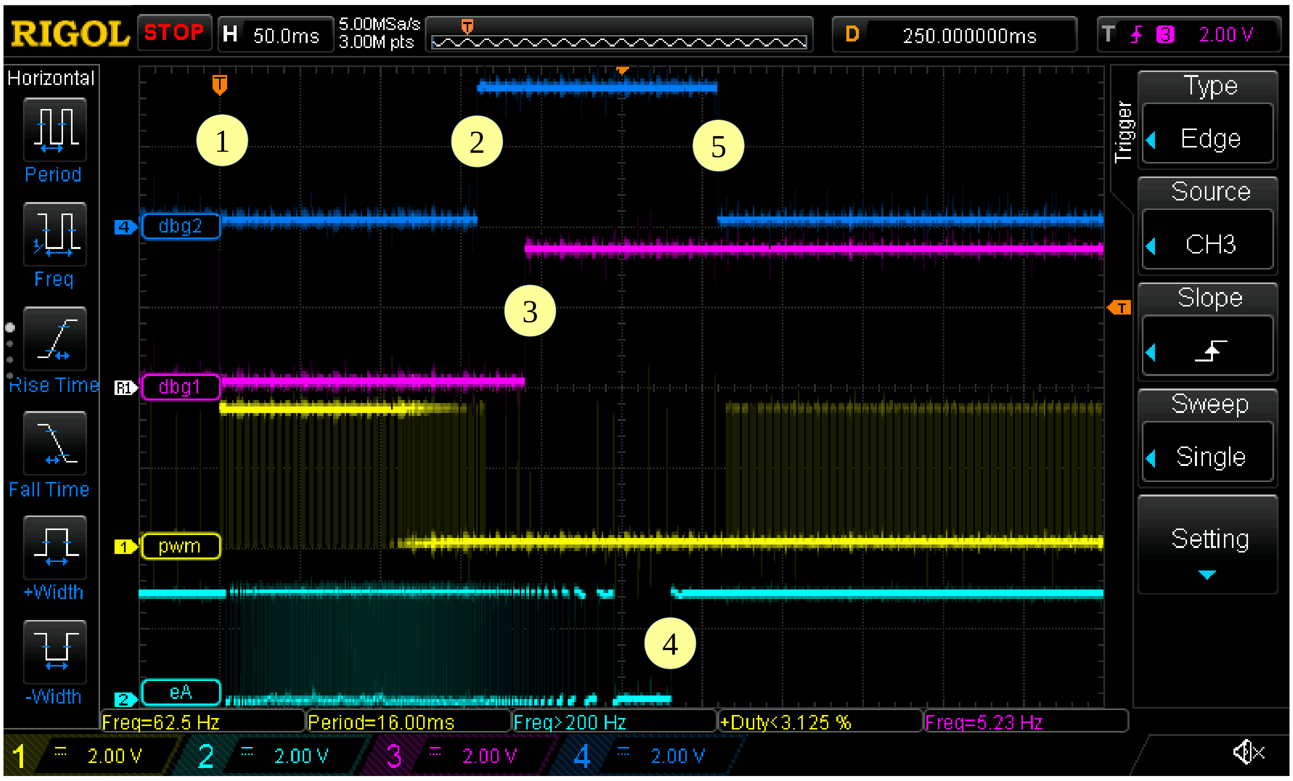
8.1 Conditions

- PID controller frequency: 200Hz (T=5ms)
- PID parameters: P=300, I=0, D=1650

8.2 PID controller test#2.1

Conditions:

- Motor shaft position set point: 0x330 = 816 (180°)



H 50.0ms 5.00MSa/s 3.00M pts

D 250.000000ms

T 3 2.00 V

Type: Edge

Source: CH3

Slope

Sweep: Single

Freq=62.5 Hz Period=16.00ms Freq>200 Hz +Duty<3.125 % Freq=5.23 Hz

1 2.00 V 2 2.00 V 3 2.00 V 4 2.00 V

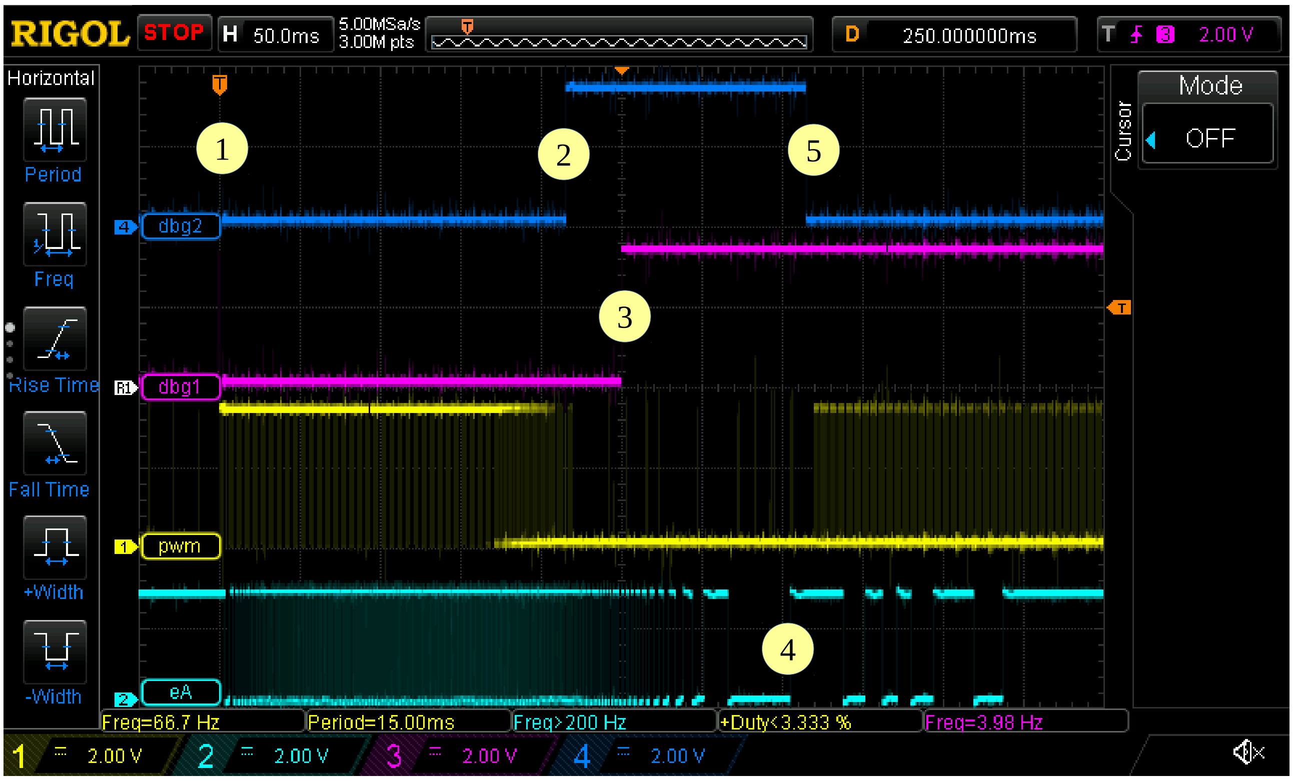
Description:

1. T=0ms
   First PID controller cycle after calibration.
   Motor shaft starts moving 4ms after PWM activation.
2. T=160ms
   Entering stop state due to request to change motor direction.
   Duration of stop state is 150ms.
3. T=190ms
   Motor shaft position is within set point ±1.5%.
   This position keeps stable within limits.
4. T=280ms
   Motor shaft completely stopped 120ms after entering stop state.
5. T=310ms
   Leaving stop state with very low PWM duty, not enough to move motor shaft.

8.3 PID controller test#2.2

Conditions:

- Motor shaft position set point: 0x4c8 = 1224 (270°)



Description:

1. T=0ms
   First PID controller cycle after calibration.
   Motor shaft starts moving 4ms after PWM activation.
2. T=215ms
   Entering stop state due to request to change motor direction.
   Duration of stop state is 150ms.
3. T=250ms
   Motor shaft position is within set point ±1.5%.
   This position keeps stable within limits.
4. T=355ms
   Motor shaft completely stopped 100ms after entering stop state.
5. T=365ms
   Leaving stop state with low PWM duty, just enough to move motor shaft.