

# Labrapport TTK4175

Group 22

Student Vebjørn Wøllo

Student Magne Angvik Hovdar

February 3, 2021

## Contents

<b>1</b>	<b>ABB - lab</b>	<b>1</b>
1.1	Del 1) . . . . .	1
1.2	Del 2) . . . . .	1
1.3	Del 3) . . . . .	1
1.4	Del 4) . . . . .	1
1.5	Bilder . . . . .	2

# 1 ABB - lab

## 1.1 Del 1)

En 'control module type' er en template for en 'control module'. En 'control module' er en instans av en 'control module type'. En 'singel control module' er en 'control module' som bare kan bli instansiert én gang.

## 1.2 Del 2)

Når vi skrur av og på lasten på varmekolben, så kan vi se at motoren trekker mer strøm under last.

Sammenhengen mellom avlest referanse for motorhastigheten og faktisk hastighet i rpm er gitt ved:

$$rpm = \frac{3}{40} motorspeed \quad (1)$$

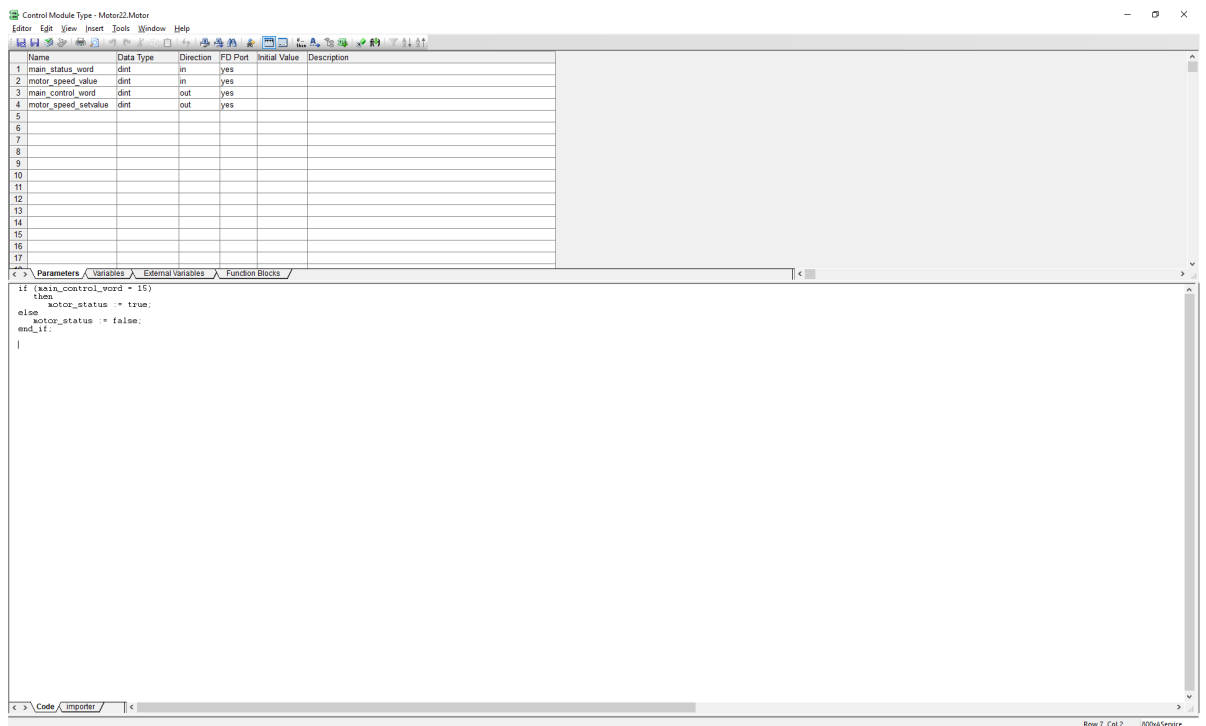
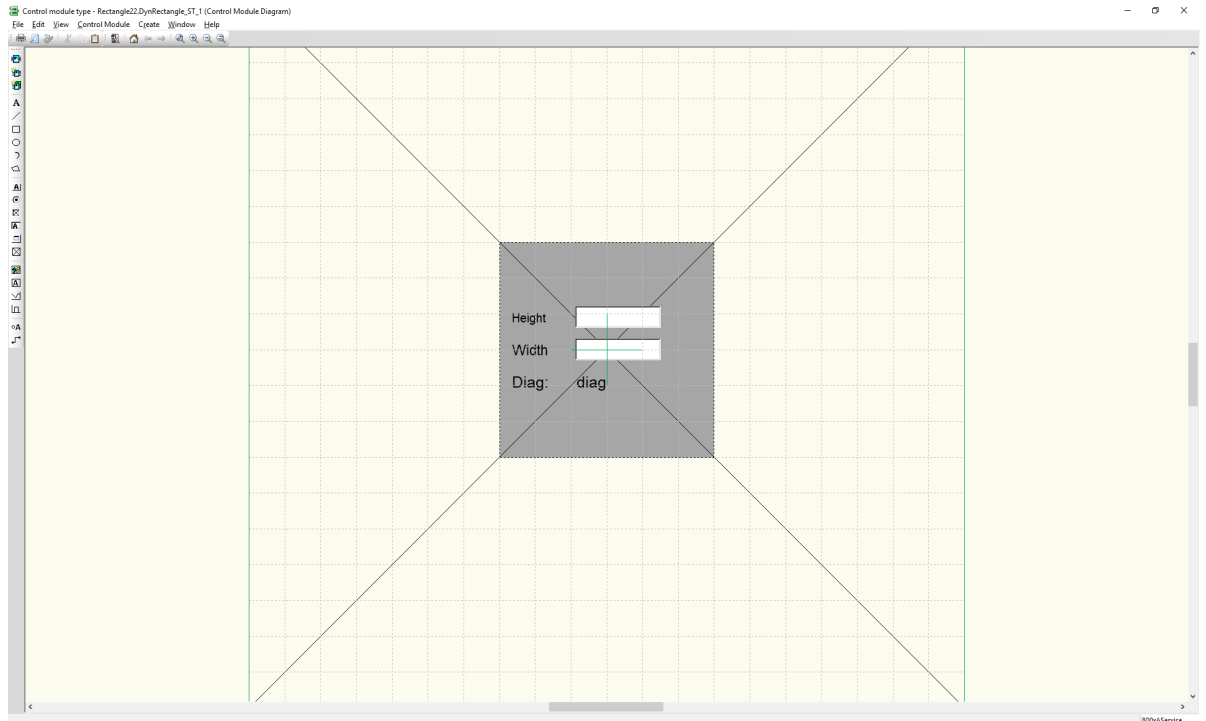
## 1.3 Del 3)

På motoren la vi alt av informasjon og styring inn i "motor info". Vi brukte trikset med string-variabler med initialverdi lik det det skulle stå for å få gjort main-status-word koder om til leselig tekst. Vi importerte deretter og koblet sammen variable fra "info" og "motoricon" inn til motor. Til slutt kunne da motor-modulen vår importeres inn i applikasjonen i sin helhet. Vi fulgte tilsvarende prosedyre med heater. Her var det ingenting som skulle styres fra brukeren så det var bare å importere. På generator måtte vi se på dokumentasjonen for å finne ut at IT-01 målte load. Da hentet vi denne ut herifra og fulgte tilsvarende prosedyre som ved de foregående.

## 1.4 Del 4)

Regulatoren trenger en referanse og en feedback-verdi. Vi brukte real2CC-blokker for å sende inn en brukerdefinert referanse og en feedbacktemp. Vi brukte graphical connections istedet for å forbinde variablene "direkte". Vi måtte importere basiclib og endre noen typer fra dint til real, så fikk vi det error-fritt. Programmet kjørte fint.

## 1.5 Bilder



Control Module Type - Motor22.Motor\_info

Editor Edit View Insert Tools Window Help

Name	Data Type	Start Attribute	Attributes	Initial Value	Description
1 motor_speed_value_rpm	dint	retain		0	
2 motor_speed_setvalue_rpm	dint	retain		0	
3 off_button	bool	retain		false	
4 ready_button	bool	retain		false	
5 run_button	bool	retain		false	
6 off_string	string	retain		off	
7 ready_string	string	retain		ready	
8 run_string	string	retain		run	
9 undefined_string	string	retain		undefined	
10 motor_readable_status	string	retain		undefined	
11 rdy_on	bool	retain		false	
12 rdy_running	bool	retain		false	
13 running	bool	retain		false	
14					
15					
16					
17					

Parameters Variables External Variables Function Blocks

```

if (ready_button = true)
then
  main_control_word := 7;
  ready_button := false;
end_if;

if (run_button = true)
then
  main_control_word := 15;
  run_button := false;
end_if;

if (off_button = true)
then
  main_control_word := 6;
  off_button := false;
end_if;

rdy_on := TestBit(dint_to_word(main_status_word), 0);
rdy_running := TestBit(dint_to_word(main_status_word), 1);
running := TestBit(dint_to_word(main_status_word), 2);

if (running = true)
then
  motor_readable_status := run_string;
elseif (rdy_running = true)
then
  motor_readable_status := ready_string;
elseif (rdy_on = true)
then
  motor_readable_status := off_string;
else
  motor_readable_status := undefined_string;
end_if;

motor_speed_value_rpm := motor_speed_value * (3/40);
motor_speed_setvalue := motor_speed_setvalue_rpm * (40/3);
  
```

Code

Row 1, Col 1 800xService

Control Module Type - Motor22.Motor\_info

Editor Edit View Insert Tools Window Help

Name	Data Type	Direction	FD Port	Initial Value	Description
1 main_status_word	dint	in	yes		
2 motor_speed_value	dint	in	yes		
3 main_control_word	dint	out	yes		
4 motor_speed_setvalue	dint	out	yes		
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					

Parameters Variables External Variables Function Blocks

```

if (ready_button = true)
then
  main_control_word := 7;
  ready_button := false;
end_if;

if (run_button = true)
then
  main_control_word := 15;
  run_button := false;
end_if;

if (off_button = true)
then
  main_control_word := 6;
  off_button := false;
end_if;

rdy_on := TestBit(dint_to_word(main_status_word), 0);
rdy_running := TestBit(dint_to_word(main_status_word), 1);
running := TestBit(dint_to_word(main_status_word), 2);

if (running = true)
then
  motor_readable_status := run_string;
elseif (rdy_running = true)
then
  motor_readable_status := ready_string;
elseif (rdy_on = true)
then
  motor_readable_status := off_string;
else
  motor_readable_status := undefined_string;
end_if;

motor_speed_value_rpm := motor_speed_value * (3/40);
motor_speed_setvalue := motor_speed_setvalue_rpm * (40/3);
  
```

Code

Row 17, Col 8 800xService

