TTK4175 - Instrumenteringssystemer
Profibus

Håkon Espeland – 772703
Ali Al-Jumaili – 772170

Group 05

20. March 2017

NTNU – Trondheim
Norwegian University of
Science and Technology

Contents

1 Part 1

In this section the task was to create a new project and configure the PLC along with the Profibus network. The instructions were followed as clearly mentioned in the assignment text, afterwards we were able to load our program onto the PLC.

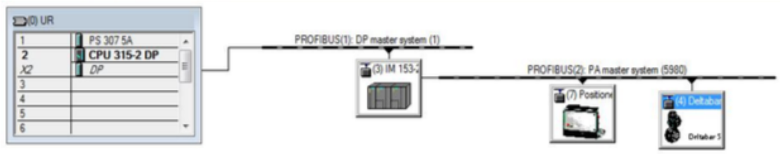

Figure 1: Configuration view

2 Part 2

In this section, it was desirable to implement a PID-regulator to keep the level of the tank as desired. The PROFIBUS network contains two slave-nodes contributing us with necessary information to use the master (PLC) to regulate the water level. The valve is used as gain, where the valve opening changes along with the set value (SV). The DP-cell contributed with actual differensial pressure, and after calibration, the actual level of the tank.

In Simatic Manager we used the following blocks; PID (FB41), write to valve (SFC15), read from DP (SFC14) and different data- and function-blocks.

To implement the different blocks we needed the correct addresses. These were found using the programming calculator on the Desktop PC, and as a verification we checked them against old projects.

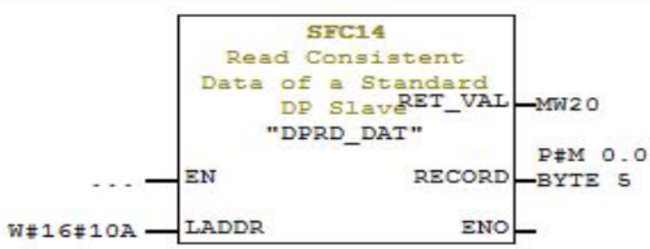

Figure 2: SFC14 block

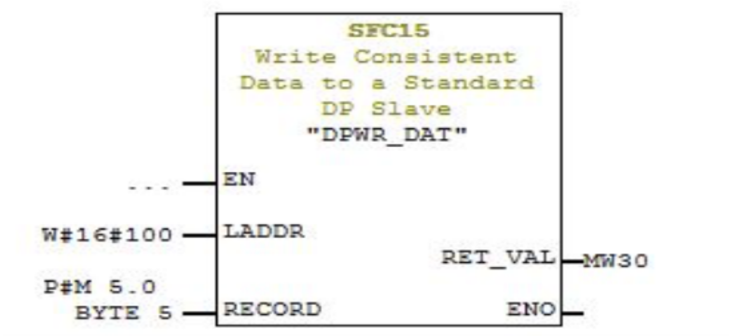

Figure 3: SFC15 block

3 Part 3

Measuring the level of water in the tank is done by a DP-cell, measuring the differential pressure of the bottom and the top of the tank. In PDM we were able to modify this sensor by setting the range to 0-500 mBar, 0-100 %. Then we did a calibration procedure by manually modifying the actual value.

The first step towards calibration, performing a zero-adjust. We then manually filled the tank so that the level glass indicated 0%. We then entered "zero-like" values such as 0, 0.001, 0.0001, 0, until we were satisfied.

The second step, performing a top-adjust, manually filling the tank so that the level glass indicated 100%. We then entered values such as 99.999, 100, 100.01, 99.999, 100, until we were satisfied.

Both steps were repeated until it corresponded to the entire scale, 0-100 %.

4 Part 4

In this section, the implementation of the PID-regulator is done.

We followed the instructions given in the assignment text, and after some documentation reading on the address handling in the Siemens software, we finally got the PID to respond to our set-value. Making new variables required space in memory, hence RECORD taking up 5 bytes from M5.0, we made our variables from M10.0 $\rightarrow$

	Address	Symbol	Display format	Status value	Modify value
1	MB 0		HEX	B#16#43	
2	MB 1		HEX	B#16#10	
3	MB 2		HEX	B#16#72	
4	MB 3		HEX	B#16#ED	
5	MB 4		HEX	B#16#80	
6	MD 0	"PV_IN"	FLOATING_POINT	144.4489	
7	MW 20		HEX	W#16#0000	
8	MB 5		HEX	B#16#00	
9	MD 5		HEX	DW#16#00000000	
10					
11					
12					
13	MW 30		HEX	W#16#0000	
14					
15	M 10.1	"MAN_BOOL"	BOOL	false	false
16					
17	M 10.2	"P_BOOL"	BOOL	true	true
18	M 10.3	"I_BOOL"	BOOL	true	true
19					
20	M 10.4	"D_BOOL"	BOOL	true	true
21	MD 40	"REF"	FLOATING_POINT	50.0	50.0
22	MD 45	"GAIN"	FLOATING_POINT	1.0	1.0
23	M 10.5	"PID_BOOL"	BOOL	true	true
24					

Figure 4: VAT_1

Using PID, requires some values to be found. In a normal situation we would used Ziegler-Nichols methods to find suitable gain, T_i and T_d, but due to the valve not working optimal this was rather hard. When the valve exceeded 80% opening, it got locked. This could maybe be issued by having tightened the seals of the valve too much. Fixing this problem requires mechanical maintenance on the valve, something that one of the lab-responsibles were notified about.

Below, figure 6 is showing a trend over the process, but it is not optimal due to the "locking-valve". To make the valve react after being locked, we needed to send it a 0 signal, release the water pressure and wait approximately 10-15 seconds.

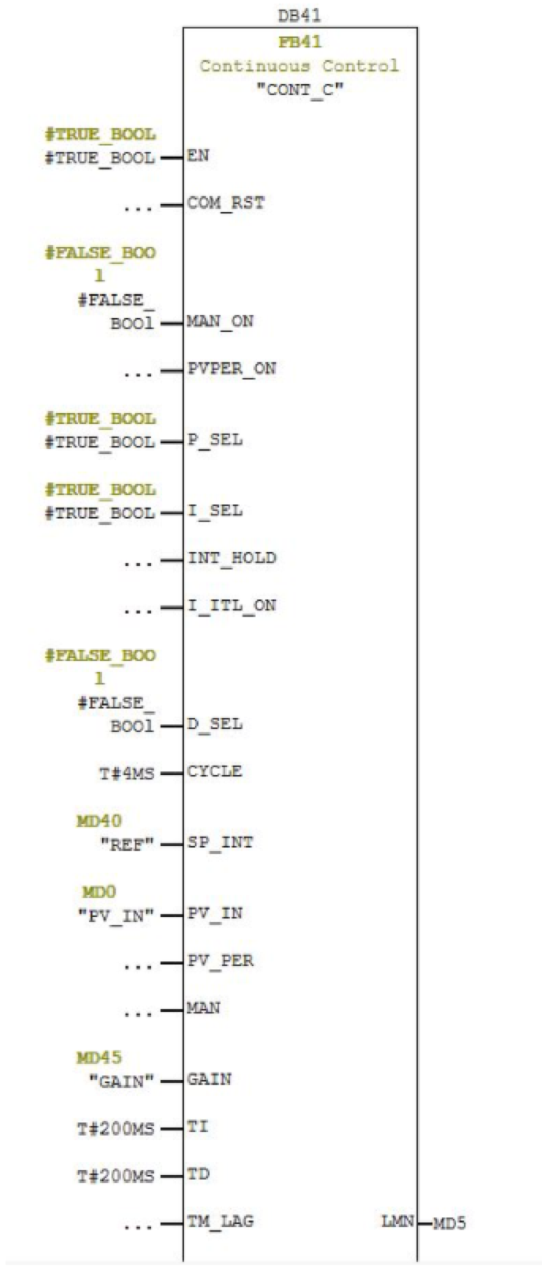
DB41
FB41
Continuous Control
"CONT_C"
#TRUE_BOOL
#TRUE_BOOL
EN
...
COM_RST
#FALSE_BOO
l
#FALSE_
Bool
MAN_ON
...
PVPER_ON
#TRUE_BOOL
#TRUE_BOOL
P_SEL
#TRUE_BOOL
#TRUE_BOOL
I_SEL
...
INT_HOLD
...
I_ITL_ON
#FALSE_BOO
l
#FALSE_
Bool
D_SEL
T#4MS
CYCLE
MD40
"REF"
SP_INT
MD0
"PV_IN"
PV_IN
...
PV_PER
...
MAN
MD45
"GAIN"
GAIN
T#200MS
TI
T#200MS
TD
...
TM_LAG
LMN
MD5

Figure 5: PID

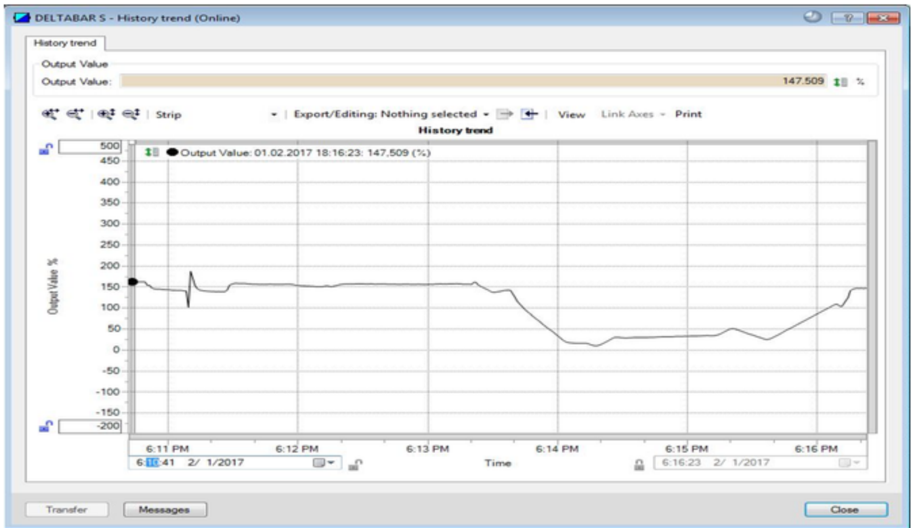

Figure 6: Trend

5 Theory

Profibus consists of two methods, PA (Process Automation) and DP (Decentralised Peripherals). PA is used for monitoring of measuring and regulating systems. DP is used to operate sensors and actuators. Profibus contains the first, second and seventh layer of the OSI model. Profibus transmits in three different ways. Electric with transfer speeds from 9.6 kbit/s to 12 Mbit/s and a range of 100 to 1200 meters. Optical transmission with a high speed and range. MBP where both data and fieldbus power is transmitted in the same cable, long range and transfer rate of 31.25 kbit/s.

The benefits of using Profibus vs. 4-20mA appears in this system by not having to use two cables to the devices in the periphery, and connects them instead to the same bus.

6 Evaluation

Assignment text: The theory for this lab was very detailed and good, making it easy to teach the basics of the software, though we did prepare us by some additional reading on Profibus.

The lab: Throughout this lab we learned a lot about Profibus and its benefits, along with the Siemens software. Though not having an optimal valve for the PID-task, we felt like we got the most right.

We used approximately 8 hours on this lab assignment.