TTK4175 - Instrumentation Systems
Foundation Fieldbus

Håkon Espeland – 772703
Ali Al-Jumaili – 772170

Group 05

09. February 2017

NTNU – Trondheim
Norwegian University of
Science and Technology

Contents

1 Part 1: Configuration of fieldbus using Syscon

1.1 Preparations

The PC is connected by a CAT-5 ethernet cable to a FI302 unit in the field, consisting of power supply (PS302), terminator (PSI302) and processor (FI302).

1.2 Communication

The network is terminated in both ends, and BT is connected to OUT+.

The network is now powered, but does not have any nodes/units connected. In this assignment we will first do the physical set-up, before we do the logic.

1.3 Syscon Start-up

Syscon started, and file created with the name **Gruppe 05.**

1.4 Adding network and initializing communication

myBus created as **New Fieldbus...** with the following **Communication Settings:**

- Server ID: Smar.DFIOLEServer.0

- Server Context: All.

- Node Name: myBus

→ Communication initialized.

1.5 Adding new bridge for ethernet/fieldbus

New bridge created.

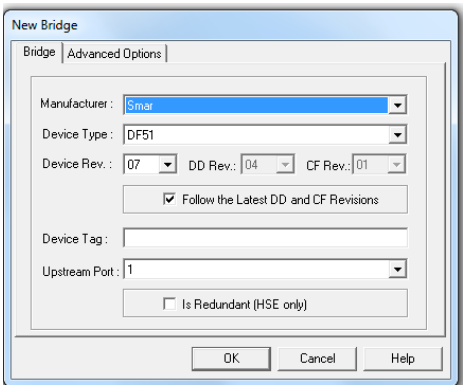

Figure 1: New bridge

Commissioning of the new bridge was applied to check the **Device Configuration** against the **Physical Device**.

Device Configuration:

Device Rev. → 07

DD Rev. → 04

CF Rev. → 01

Physical Device:

Device Rev. → 07

DD Rev. → 01

CF Rev. → 01

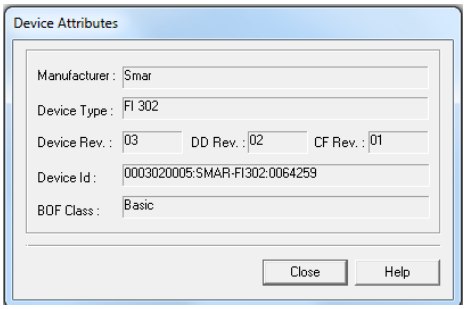

Figure 2: Device Attributes FI302

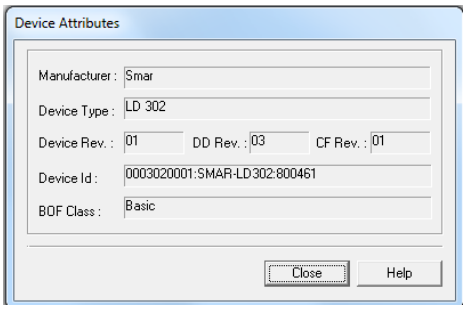

Figure 3: Device Attributes LD302

The **Exchange** function was used to apply the **Physical Device Rev**. for both (07, 01, 01).

The Rev. values from the **Livelist** was noted for further tasks.

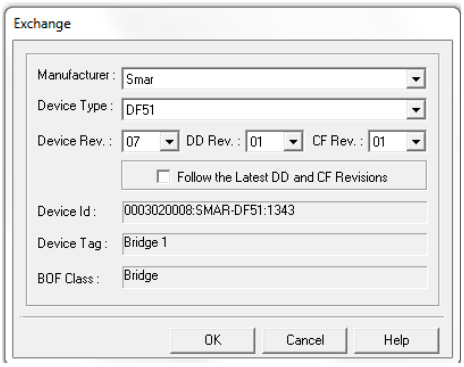

Figure 4: Exchange

1.6 FI302 and LD302

FI302 and LD302 were added as **New Device** with the same Rev. values found in the Livelist. Then we used **Commission** to establish contact with the nodes.

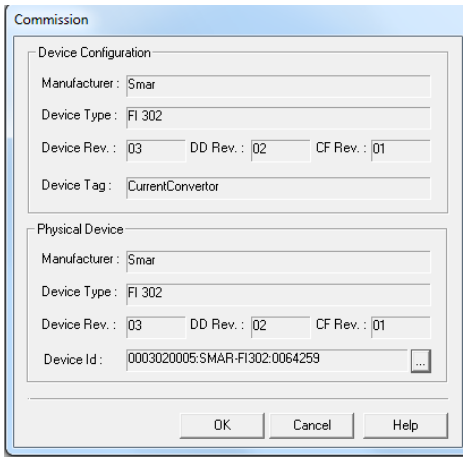

Figure 5: Commission FI

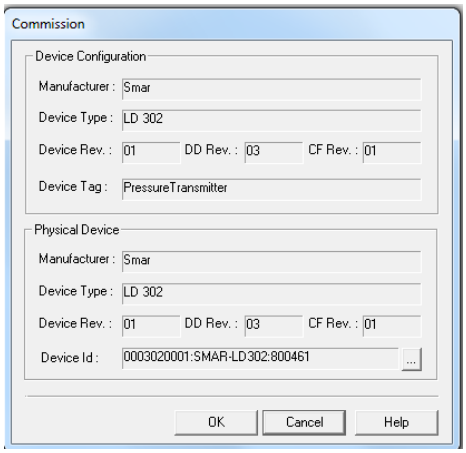

Figure 6: Commission LD

1.7 Implementing function blocks for the field units

Transducer block: Insulates the function block from the specific I/O hardware, such as sensors, actuators. Transducer block controls access to I/O through manufacturer specific implementation. This permits the transducer block to execute as frequently as necessary to obtain good data from sensors without burdening the function blocks that use the data. It also insulates the function block from the manufacturer specific characteristics of certain hardware. By accessing the hardware, the transducer block can get data from I/O or passing control data to it. The connection between Transducer block and Function block is called channel. These blocks can exchange data from its interface. Normally, transducer blocks perform functions, such as linearization, characterization, temperature compensation, control and exchange data to hardware.

Resource block: Are used to define hardware specific characteristics of function block applications. Similar to transducer blocks, they insulate function blocks from the physical hardware by containing a set of implementation independent hardware parameters.

We added the function blocks as:

AI → Pressure Transmitter.

AO → Current Converter.

1.8 Function block parameters

MODE_BLK: has 4 elements.

- Target - This is the mode requested by the operator. Only one mode from those allowed by the permitted mode parameter may be requested, that check will be done by the device.
- Actual - This is the current mode of the block, which may differ from the target based on operating conditions and block configuration, as input parameter status and bypass configuration, for example. Its value is always calculated as part of block execution, therefore the user can not write in this attribute.
- Permitted – It defines the modes that are allowed for an instance of the block. The permitted mode is configured based on the application requirement. For example, if a PID block does not have link for CAS_IN, the Cas mode should not be permitted for that block. It is like a list of mode types selected from the supported modes.
- Normal - This is the mode which the block should be set to during normal operating conditions. The normal attribute is used as a reminder. It does not affect the algorithm calculation.

XD_SCALE: The AI block has the XD_SCALE parameter to define the engineering units expected from the transducer.

OUT_SCALE: When the OUT value exceeds the OUT_SCALE range and no worse condition exists in the block then the OUT status will be "uncertain, EU Range Violation".

PV_SCALE: Converts the error to percentage (PID).

L_TYPE: The L_TYPE parameter determines how the values passed by the transducer block will be used into the block. The options are:

- Direct - the transducer value is passed directly to the PV. Therefore OUT_SCALE is useless.
- Indirect - the PV value is the FIELD_VAL value converted to the OUT_SCALE.
- Indirect with Square Root - the PV value is square root of the FIELD_VAL converted to the OUT_SCALE.

CHANNEL: The CHANNEL parameter configuration depends on the device features as it follows:

- Fixed I/O Device: This type of device has a fixed number of I/O. All Smar field devices belong to this class. The channel is numbered from 1 to the maximum number of I/O.
- Configurable I/O Device: The user may configure the number of I/O modules as well the I/O type (input, output, discrete, analog, pulse,...). The DFI302 is the only device classified as a configurable I/O decice. All I/O modules have the I/O points arranged as Point (P), Group (G), Slot (S) and Rack (R).

TERMINAL_NUMBER: Is used to assign inputs to belonging transducer.

2 Part 2: Testing hardware, configuration and logic

2.1 Hardware test by downloading

- Configuration downloaded
- OPC Server updated
- Tags exported
- Valve tested in Online-mode (0-100%)

The valve functions as intended, but the pressure transmitter (PT) needs some calibration according to the level glas (LG).

Old parameters:

$EU\_100 = 690$

$EU\_0 = 240$

New parameters:

$EU\_100 = 435$

$EU\_0 = 170$

2.2 Logic Configuration

We created an Analog Out (AO) block in the CurrentConverter controlling the valvle, and an Analog Input (AI) block in the PressureTransmitter giving input to the PID on present value (PV).

According to the assignment text, we should have chosen one of the nodes to include the PID controller. After a little bit of experimenting, we reasoned with that it could be implemented in both modules, and also directly into the application if the module attachments were done properly.

The PID parameters was setup as described in the assignment text.

When it comes to the control strategy, we connected the blocks accordingly to:

- AI.OUT to PID.IN
- PID.OUT to AO.CAS-IN
- AO.BKCAL_OUT til PID.BKCAL_IN

2.3 Logical testing and regulator tuning

This section resulted in a lot of testing and troubleshooting.

The Pressure Transmitter provided an output signal of 0 to -100 compared to the level glas, which made us want to invert that output. After some reading of documentation we found the invert-function, though it did not work as documented. We also experimented with arithmetic blocks to multiply the signal with a factor $K = -1$, but the result was unsuccesfull. Could this problem be caused by inverted polarity?

The software did neither accept the limits of 0 to -100 %, which again made this final task very difficult.

If the signal had been positive, it would have been a simple task to tune the regulator with Ziegler-Nichols method, but due to this issue, it did not work as intended.

If the level was approximately 50%, the transmitter provided an output signal of -50. When the set-value was defined positive from 0-100%, the PID would just open the valve more and more due to increasing deviation as the level output rised towards -100%.

After almost 10 hours at the lab, experimenting with different parameters in all modules, we concluded that there must be some modification to be done, to make the signal positive instead of negative.

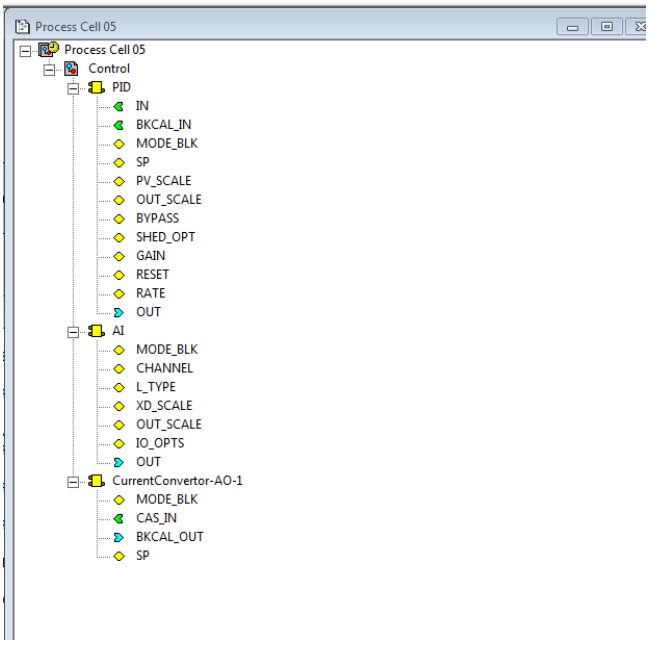

Figure 7: Change of Rev.

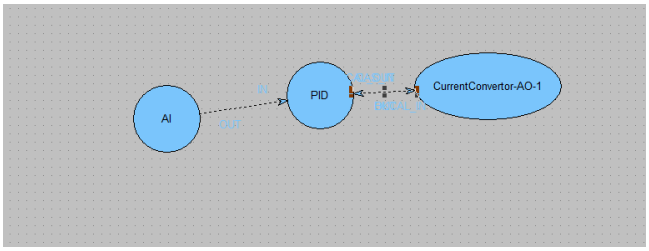

Figure 8: Change of Rev.

3 Part 3: Evaluation

a) Evaluation of the assignment text:

The theory in the assignment text was good and precise, and the Foundation Fieldbus documentation was also very good, as it aloud us to search for keywords in the PDF instead of going through piles of paper.

b) Evalution of the assignment:

We have learnt the basics of fieldbus, and how this can save costs regarding to multiple signals in one cable along with less equipment in the field.

If one disregards the signal-issue from the pressure transmitter, it was a very interesting and practical good assignment.

We used approximately 10 hours, though half of the time were used troubleshooting the signal from the pressure transmitter.

3.1 Important note

An important note to mention which can be dangerous to many groups, is the way the water drain hose is setup. If the drain valve is opened rapidly it will cause the hose to jump out from the drain, splashing all of the electronic components in the surrounding area. This has happened to us, but luckily no electronic were splashed, just a couple of jackets and chairs. We would recommend a hose strap or similar to keep it in place, or mentioning it as a warning in the assignment text.