Protocol PR2014-12-02
Software Design of "Interface"

Author: M. Siegl, G. Damianitsch
Modified: 2014-12-07
File: PR2014-12-02
Version: 1.1

1 Background

The first activity of implementation was, to create the APIdoc with public methods between the "Interface" and the Gbot. Until now, no correction according the API-doc arrived. We assume, the API-doc is accepted.

To see some details about the native interface to the RS422/Modbus and the CAN, two G2021 are requested. Until now, the requested Hardware is not available.

DSG requested git for code-sharing during development. We can use it too, but we doesn't provide the internet service GitHub.

During internal discussion, some details about the implementation of the Software independent of the Hardware are defined. In this protocol shows the result of the discussion.

2 Results of Discussion

2.1 DataPoint

1. The "Interface" provide some public methods, which are called by the Gbot. This public methods are the front-end of the DataPoint, which will be created by calling of the Gbot. As a rule, a Gbot creates its own DataPoint.
2. The provided methods are specified by the API-doc.

2.2 Buffer

1. The Buffer is feed with sequence of bits from the gatherer. The gatherer realises the native interface to the sensors – connected in the RS422 or CAN. Details to this feed must be defined ...
2. A sensor gives a sequence of bits, which represet the physical value, measured by the sensor.
3. The sequence of bits must be corrected – shifted, swaped, etc. The result after correction is a number (an instance of java.lang.Number). This number cannot be interpreted, it is only a number which is input for further calculation. Each sensor need a specific definition. This correction mechanism is defined in a generic manner. So the implementation of new sensors is possible without new coding.
4. Next the number passes the Adapter-Chain. The Adapter-Chain consists of functions and filters. It is similar to the pipe "|" in the unix command shell. The number of the elementes of the adapter chain is not limited. It has to be configured in the buffer configuration.

The following listed adapters will be implemented:

- A Scaling Adapter instance will calculate an output value y based on the current input value x using the equation $y = a x^2 + b x + c$. The factors a, b and c have to be configurated.
- The Triggering Adapter instance will toggle between two output values. The upper and the lower input threshold value and the output values are configuration items.
- The Lowpass Adapter instances will act as lowpass filter. In this case, the time constant is configurable. Lowpass Adapter instances will be triggered by an internal 1 sec time ticker and will calculate the output values every second.
- A Filtering Adapter will suppress marginal input value changes. The maximum difference between the last sent output value and the current input value is a configuration item.

5. The output value of the Adapter Chain is not a meaningless number anymore, this is in conjunction with the configurated buffer Meta-Data the wellformed sensor value.
6. Each time, the value changes, the new value will be sent to the DataPoints.

2.3 Communication

1. The transfer of information between Buffer and DataPoint goes over a socket. The socket is defined for communication via the IP-Network. Therefore a distributed "Interface" can be created later.
2. The communication is coded in XML. The associated definition (XSD) must be defined.

2.4 Testing Points

For testing purposes some code is usefull.

1. A peace of software showing the usage of the public methods of a DataPoint by a Gbot. The SW will print some status-messages to the command-line window.
2. The transfered information through the socket can be logged (logging level "fine").
3. A peace of software generate some dummy sensor values (bit sequence). The adapter chain will be feed by this.

2.5 Java

1. As defined at the beginning, the implementation will be done based in Java 1.8. A downgrade to Java 1.7 is not possible. So, some gain in performance will be achieved.
2. The Java based Software will be developed with the path:

   `at.ac.tuwien.infosys.g2021.intf`

3 Next Activities

3.1 Definitions

Some open definitions must be done – internal task of Citem.

3.2 Native Interface

Waiting for the requested Hardware goes on.

The gathering of input values from the device drivers is one of the most risky and time-consuming tasks of the Interface development. Due to missing specifications and unmade decisions, it is not possible to do anything for this complex task.

Nevertheless, in case of ongoing missing the requested G2021, some alternatives should be found. Maybe we use a Raspberry Pi or some other hardware – clearance between DSG and Citem.

3.3 Document and Code Sharing

A easy way for document and code sharing between DSG and Citem should be defined. If DSG is running a git-hub, it should be investigated to use this hub. The simple way to send some files and documents via email is possible too – clearance between DSG and Citem.

3.4 Testing Points

The peace of software, which shows the usage of the public methods, has to be coded and provided to DSG. The appropriate code inside the Gbot can be written accordingly – task of Citem.