

AixOCAT – Introducing a Library for Building Automation and Control Systems



Nowadays, the building sector faces an increasing complexity in the design and operation of building energy systems. Hence, a proper implementation of the building automation and control systems (BACS) for such systems is time consuming and expensive. In order to reduce the implementation effort, a structured control development with reusable program code is desirable. Although current commercial control libraries are available, they raise considerable costs and provide only limited interoperability with further software. Hence, and because of the complexity, the restricted amount of time and little proper support, especially small- and medium-sized companies are limited in their capabilities of implementing BACS. In order to support the implementation of control algorithms, we propose an open source library for building automation software based on the OSCAT library. This article shows the library's general usability and the workflow for the implementation of control algorithms in BACS using the example of a control loop in a recently constructed building. The library, the examples and the workflow could become a major contribution to the construction of intelligent and energy efficient buildings.

Keywords: building automation and control systems, programmable logic controller (PLC), control algorithm, open-source library, object-oriented programming

Building automation and control systems are considered a vital ingredient of healthy, energy-efficient and comfortable buildings. In this field, there is a large variety of use cases with different requirements [1]. The common engineering process of BACS, however, is often inefficient and results in building energy systems that have a high potential for performance improvement [1,2]. These deficiencies result from a lack of a commonly accepted concepts for the creation of building automation software, leading to a very heterogeneous building stock [1,3]. Due to the large differences between individual buildings, it is challenging to apply intelligent algorithms such as automatic fault detection and diagnosis or model predictive control [3], which are considered vital concepts for high-performance buildings.

A freely available library containing modules and best practice examples for future automation software projects could greatly support engineers to improve BACS and building operation. In this work, we present an open-source library for automation software compliant with the IEC 61131 standard to enable efficient development of automation code that is both reliable and enables the application of advanced control and operation analysis.

The AixOCAT library

In order to increase the quality of building automation projects and at the same time reduce the engineering effort, we identified the goals listed below and designed the AixOCAT library [4] accordingly. It provides both code templates and best practice examples that follow the principles.

We aim to

1. achieve high re-usability of code and thus reduce engineering effort
2. increase reliability of technical building equipment and test benches
3. support extendibility of automation systems
4. enable interoperability, especially with cloud-based applications

Re-usability and reliability

The AixOCAT library contains code templates that provide typical functionality for building energy systems and test benches. We have integrated most of the open-source code by the Open Source Community for Automation Technology (OSCAT) [5]. The OSCAT library contains many useful modules for building automation that could be reused in a variety of projects. Moreover, AixOCAT is an open-source library, meaning that various contributors write and review code, report software bugs and request further developments.

Therefore, the automation community contributing to the library will be able to bundle expertise and thus ensure reliable automation projects as required in goal 2.

Extendibility and Interoperability

We follow a modular design approach to ensure extendibility and interoperability as described in goal 3 and 4: The library mainly contains Programming Organization Units (POUs) implemented in Structured Text (ST) according to IEC 61131-3 standard. These POUs are mostly Function Blocks (FB), which transform several inputs into several outputs. In order to allow for an object-oriented programming style and thus extend the FB functionality with properties and methods. The software application TwinCAT [6] is one example we tested that allows to use these extended FBs for programming real-time controllers.

Using FBs, we structure the programs mainly according to physical subsystems in the energy system such as “air handling unit” (AHU) or “concrete core activation” (CCA). For further structuring, we propose an operation mode-based approach. In this context, an operation mode is a collection of pre-defined actuator commands or set-points for field devices. The transitions from one operation mode to another are based on pre-defined conditions. These can be error states or temperature thresholds [7].

In order to support extended interoperability, one operation mode can be dedicated to external control applications. It can thus allow connection to control algorithms that may be hosted outside the actual automation system, e.g. on a virtual machine, to send control commands to the controller. In this case, the actuator commands and set points are not pre-defined but generated in the external application. In the event of a communication loss, a transition to a local operation mode always ensures a secure system operation.

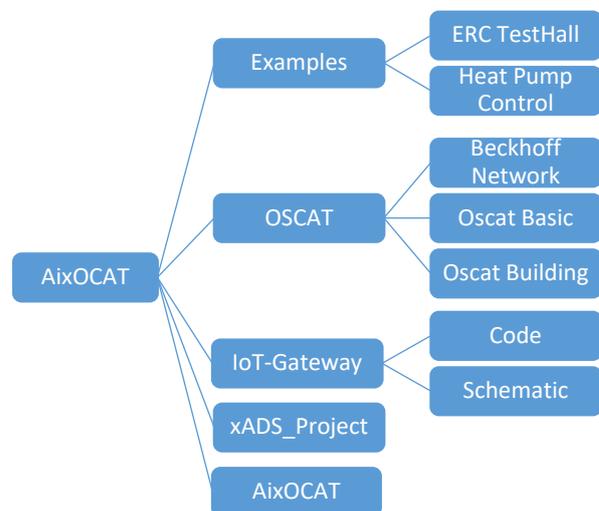


Figure 1. Current structure of the library.

An important research application is the hardware-in-the-loop method, in which a real-life controller is coupled to virtual technical building equipment represented by a simulation model. For that end, AixOCAT provides interfaces for Modelica, an object-oriented multi-physics modelling language. The application uses the Automation Device Specification (ADS) to exchange data with the PLC.

Collaborative development

We aim for a flexible, open and well-organised development of the library, which is available to the public on GitHub [8]. We propose a work flow for non-restricted but efficient contribution to the library. This work flow defines basic requirements such as programming style, documentation and issue tracking. Moreover, new developments and code enhancements are done in separate branches, ensuring that the master branch remains stable and usable by the community. A so-called “pull request” requires another contributor to review the newly added code and to merge the reviewed code into the master branch.

Use Case - ERC Test Hall

For demonstrating the use of the library and its benefits, we present a real-life project, namely the ERC test hall in Aachen, Germany, as a representative example for laboratory and machine halls. The facility consists of two hall parts with an area of 630 m² and 360 m², respectively, for test beds and an additional area with five office rooms and a total area of 90 m². The test hall is equipped with an AHU, a CCA and ceiling panel heaters (CPH) as well as chilled beams within the office rooms. Moreover, the energy system contains external after heaters and coolers. Using the example of the CCA, we demonstrate the functionality of the AixOCAT library.

Control Strategy

The required hot water for AHU, CCA and the decentralized after heaters is supplied by a heat distributor. As described in the previous section, the library especially supports PLC programming by providing object-oriented programs for BAS. For instance, the CCA comprises a conventional admixing circuit consisting of a pump, a mixing valve as well as four temperature sensors and a control unit. **Figure 2** shows the developed object-oriented representation of the CCA as a class diagram in the unified modelling language (UML).

The components are designed as interfaces, which are implemented and mapped according to the communication interfaces of the physical components. By using this structure, basically any admixing circuit is

described in the same way while following our objectives of re-usability, extendibility and interoperability.

Details and Results

As described above, the modules of the hydraulic circuit are interfaces, which can be implemented in various ways: In the test hall, we implemented

1. the temperature sensors as PT100 (including conversion of integer to temperature),
2. the valve as (0)2-10 V device for both actual and set position (including conversion between integer value and opening percentage),
3. the pump as BACnet device,
4. the control unit as a mode-based control that regulates the mixing supply temperature via proportional-integral-derivative (PID) control.

In the auto-mode of the control unit, the set temperature is determined in an ambient air heating curve. This value is compared to the actual temperature and a PID controller generates the actuating signal to the mixing valve, whereas the pump generates a constant pressure increase.

The entire code for the described modules of the admixing circuit can be found in the library. **Figure 3** shows a plot of the supply temperature for the CCA as well as the temperature at the distributor. The supply temperature was set to 22 °C. The plot shows that the oscillating distributor temperature is compensated quite well by the admixing circuit.

Beside the CCA, the induction units, the preheater, reheater and the cooler of the AHU are also realized by admixing circuits, hence the above described code for the hydraulic circuit only needs to be linked to their physical sensors, pumps and valves.

Conclusions

We introduced an open-source library that we published to support the engineering of building automation and control systems. We present the library structure and demonstrate its usability in the control system of a real-life building.

As many deficiencies in the building stock result from a lack of a commonly accepted concepts for the creation of building automation software, we identified requirements for the library to enable three main ideas:

- Re-usability and reliability,
- Extendibility and Interoperability and
- Collaborative development.

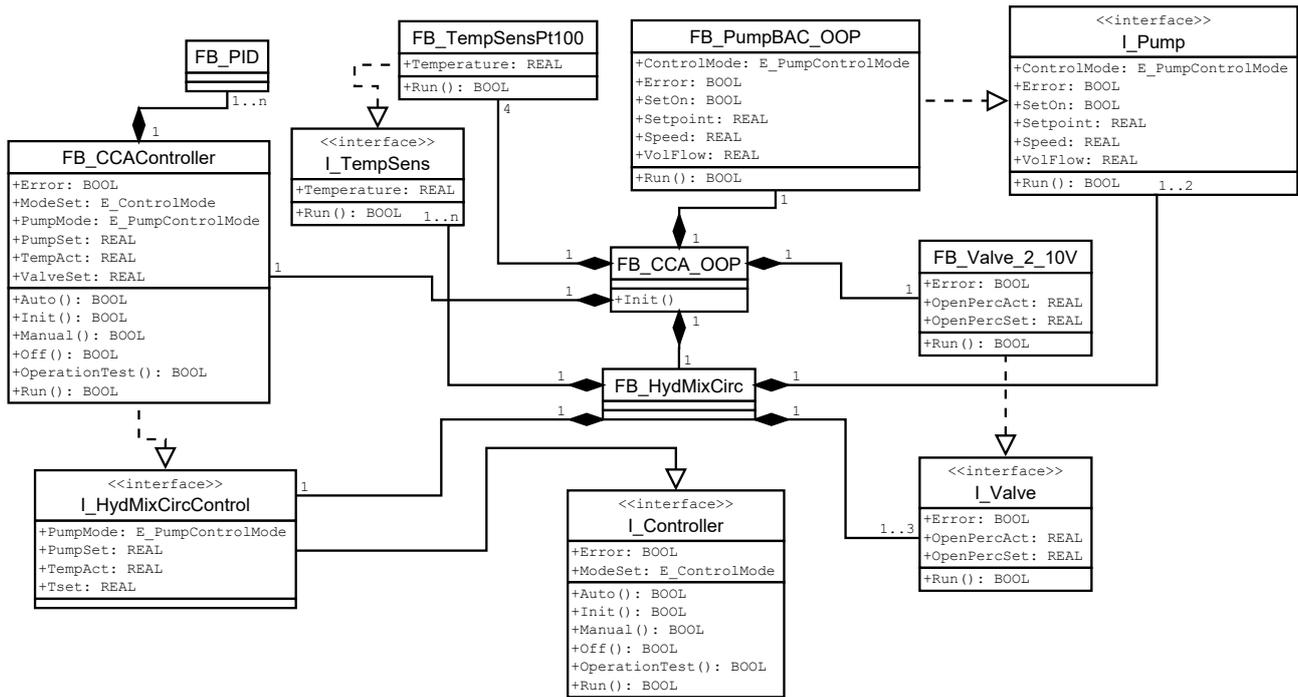


Figure 2. UML diagram of the CCA automation system; standard property gets and set methods are omitted for clarity.

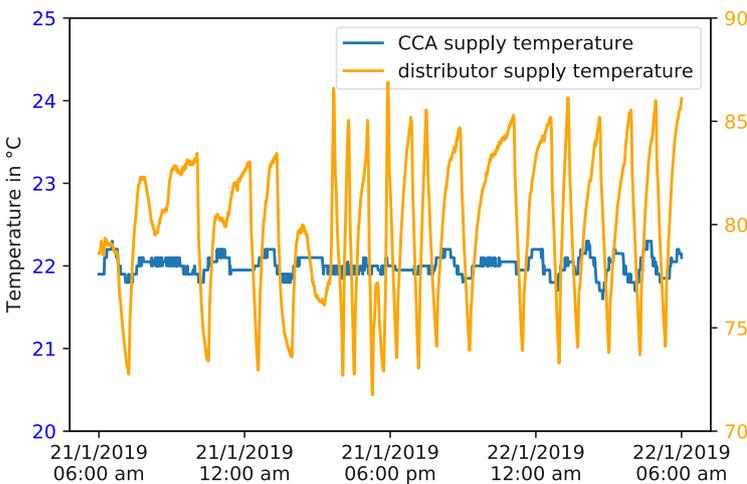


Figure 3. Plot of the CCA supply temperature and the distributor temperature in the considered hall for a duration of 24 h; the considerable oscillation of the distribution temperature is sufficiently compensated by the simple PID control.

The library is based on these ideas and, due to the open-source development, enables the building automation community to collaborate on the standardization of the engineering process.

The library will constantly be improved and is available on GitHub [4]. ■

Acknowledgement

We thank the ERDF for their financial support, promotional reference EFRE-0800537, and gratefully acknowledge the financial support provided by the Federal Ministry for Economic Affairs and Energy (BMWi), promotional reference 03ET1485A.



EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung

References

- [1] Domingues, Pedro; Carreira, Paulo; Vieira, Renato; Kastner, Wolfgang (2016): Building automation systems: Concepts and technology review. In: Computer Standards & Interfaces 45, S. 1–12. DOI: 10.1016/j.csi.2015.11.005.
- [2] Royapoor, Mohammad; Antony, Anu; Roskilly, Tony (2018): A review of building climate and plant controls, and a survey of industry perspectives. In: Energy and Buildings 158, S. 453–465. DOI: 10.1016/j.enbuild.2017.10.022.
- [3] Burak Gunay, H.; Shen, Weiming; Newsham, Guy (2019): Data analytics to improve building performance: A critical review. In: Automation in Construction 97, S. 96–109. DOI: 10.1016/j.autcon.2018.10.020.
- [4] <https://github.com/RWTH-EBC/AixOCAT>
- [5] <http://www.oscat.de/>
- [6] <https://www.beckhoff.de/twincat3/>
- [7] Schild, T. P.; Baranski, M. A.; Müller, D. (2018): Modus-basierte Steuerungsentwicklung für modulare Energieversorgungssysteme. In: VDI Verlag GmbH
- [8] <https://github.com>