DEVELOPMENT OF A DYNAMIC USER INTERFACE FOR AN ELECTRONIC DEVICE DESCRIPTIONS INTERPRETER FOR INTELLIGENT SENSORS

Pushkarskiy R.V.[1,2], Belski V.A. [1,2]
[1]Tomsk Polytechnic University,
30 Lenin Avenue, Tomsk, 634050, Russia, group: A2-43, A3-38,
e-mail: rvp6@tpu.ru, vab59@tpu.ru
[2]ASPECT LLC,
37 Saltykova-Shchedrin str., office 18, Tomsk, 634021, Russia, software developer

Abstract

To solve the problem of supporting intelligent sensors manufactured in the Russian Federation by foreign software, ASPECT LLC has developed the Colibri Field Data Bus Server. This software allows to work with HART- and Colibri-based devices with support of their electronic descriptions (CDD/EDD). The software facilitates dynamic GUI based on CDD/EDD, offering advanced control and diagnostic capabilities for field equipment.

Keywords: HART, Colibri protocol, CDD/EDD, Qt Framework, dynamic GUI, industrial automation.

Introduction

The Colibri Field Data Bus Server (Colibri software) provides a unified interface for interaction with intelligent field devices, including both domestic (Colibri or HART protocol) and international (HART protocol) equipment [1, 2]. The software has undergone compatibility testing with most popular industrial software and hardware such as Kaspersky Industrial Cybersecurity, protected virtualization environment zVirt, backup software Cyberbackup, R7 Office and R7 Graphics, ABAK PLC, Regul PLC and others [3].

The development of Colibri software graphical user interface (GUI) posed two critical challenges:

- Developing a universal dynamic interface adaptable to diverse industrial devices.
- Developing a scalable enterprise structure visualization capable of managing thousands of field devices.

Dynamic GUI creation

Colibri software utilizes colibri device descriptions (CDDs), which are unique to each device model and define device-specific parameters and menu structures [2]. To provide universal device configuration, a dynamic GUI creation engine was developed, that interprets CDDs at real time, so UI is automatically generated at that exact moment. As shown in Figure 1, this engine parses CDD elements and maps them to interactive components:

- Variables → Input fields
- Methods → Command-invoking buttons
- Built-in functions → Context-aware dialog windows

This approach eliminates the need for static interface designs, ensuring compatibility with any CDD-compliant device. Engine also provides a CDD developer and Colibri software administrator extended interface customization capabilities. Key configurable parameters include:

- Access control: Variables can be set as read-only or writable based on user roles
- Validation rules: Value ranges (min/max), data types, and input masks
- Contextual guidance: Tooltips, pop-up explanations (QToolTip), and hierarchical help menus
- Function metadata: Descriptions for built-in methods visible during runtime execution

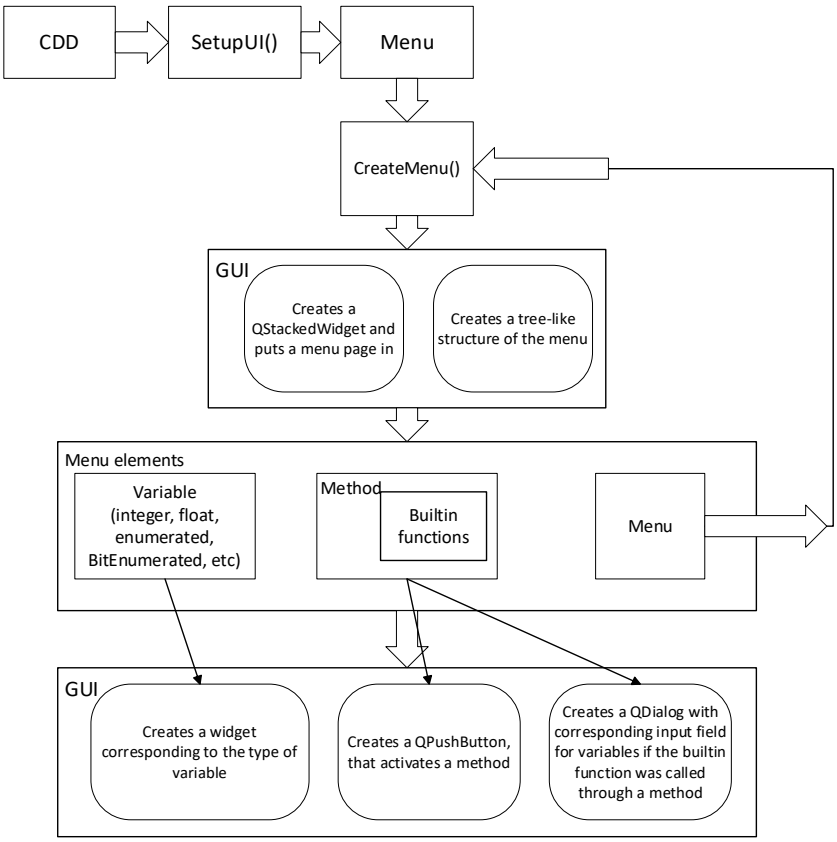

Fig. 1. A diagram of GUI creation in Colibri Software

Enterprise Structure Customization

To manage large-scale facility, the Colibri software integrates a hierarchical device tree (Figure 2) using Qt's QTreeWidget and JSON-based configurations. Key features include:

- Unlimited nested folder structures (e.g., "ABAK PLC," "New Devices" and others in the Figure 2)
- Drag-and-drop device organization
- User-defined grouping/sorting rules
- Multi-profile support for customized layouts

Current efforts focus on optimizing performance for facilities with 10,000+ devices, including load balancing and asynchronous data handling.

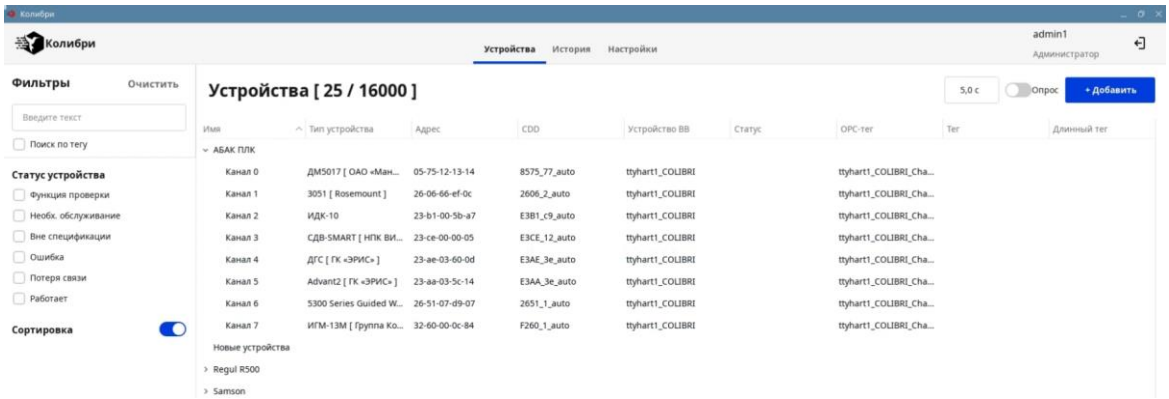

Fig. 2. Enterprise structure

Scalability Validation via Virtual Unit Software

A HART/Colibri Virtual Emulator (Figure 3) was developed to stress-test the device tree of Colibri software under heavy loads. The emulator:

- Simulates configurable response delays (0-5000 ms)
- Generates up to 64 devices per emulated I/O module
- Operates as a distributed server cluster with HART-IP support
- Leverages multi-core CPU processing for parallel task execution, which allowed to use a single OS machine for testing

In a 72-hour stress test, the Colibri software successfully managed 8,192 concurrent devices without performance degradation. Planned upgrades aim to validate scalability for 16,000+ devices.

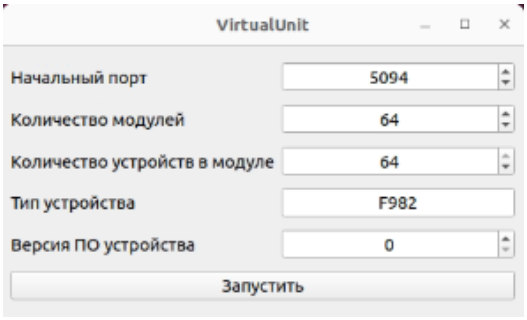

Fig 3. Virtual Unit UI

Conclusion

Colibri software is a field device monitoring and diagnostics system, that was developed by ASPECT LLC. In the scope of this research, as a part of Colibri software development, three key aspects were developed:

1. A dynamic GUI engine for CDD interpretation was created, enabling automatic interface adaptation for any type of field device.
2. A customizable facility structure visualization system, allowing for convenient visualization, grouping and sorting of field device assets.
3. A virtual emulation framework for hardware-free load testing, which allows to conduct stress tests without need to connect large amounts of field devices.

References

1. Продукты и решения // ООО "Аспект": Разработка цифровых измерительных систем – URL: https://digitalmetrolog.com/ru/products (accessed: 20.02.2025).

2. ОТЗ-06.01.15.2024 Полевая шина передачи данных Колибри // ИНТИ. – URL: inti.expert/docs/industrial%20automation/otz-06-01-15-2024/ (accessed: 20.02.2025)

3. Продукты и решения // ООО "Аспект": Разработка цифровых измерительных систем – URL: digitalmetrolog.com/documents (accessed: 20.02.2025)

4. Будников А.И. Сравнительный анализ производительности реализаций инструментария Qt для языков C++ и Python // Стратегия устойчивого развития регионов России. – 2014. – № 21. – URL: cyberleninka.ru/article/n/sravnitelnyy-analiz-proizvoditelnosti-realizatsiy-instrumentariya-qt-dlya-yazykov-c-i-python (accessed: 10.03.2025).

5. Qt Documentation | Home // Qt Documentation. – URL: doc.qt.io (accessed: 10.03.2024).

6. Будников А.И. Сравнительный анализ производительности реализаций инструментария Qt для языков C++ и Python // Стратегия устойчивого развития регионов России. – 2014. – № 21. – URL: cyberleninka.ru/article/n/sravnitelnyy-analiz-proizvoditelnosti-realizatsiy-instrumentariya-qt-dlya-yazykov-c-i-python (accessed: 10.03.2025).

7. FCG_TS61804-4 Edition 2.1 EDD Interpretation // FieldComm Group Library. (n.d.). library.fieldcommgroup.org/61804/TS61804-4/2.1/#page=1 (accessed: 20.02.2025)