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

R.V. Pushkarskiy[1,2], V.A. Belsky [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 a HART-compatible Colibri field bus. Within the framework of the Colibri project, the Colibri software is being developed to work with intelligent sensors and their electronic descriptions.

Keywords: HART, Qt Framework, EDDL, Colibri, CDD.

Introduction

Currently, there is a problem of lack of support for foreign software for intelligent sensors manufactured in Russia.

Aspect LLC has developed a Hart compatible field data bus "Colibri", which provides interaction of automated process control and monitoring systems with intelligent field devices. Aspect has developed prototypes of a system for end-to-end control and diagnostics of field equipment based on the Colibri protocol, as well as a field Hart compatible Colibri communicator [1].

Bench tests are being conducted for the compatibility of the Colibri protocol with instrumentation and control equipment using Hart technology. As part of the development of the Colibri software, the task is to create a dynamic user interface for all types and models of intelligent sensors in the industry, which will be described in this report. In addition, it is necessary that the software being developed runs on low-performance devices, such as industrial secure tablets, and for this reason it is important to monitor the use of PC resources [2].

Description of UI Creation in Colibri Software

Qt framework for C++ was chosen to develop the application, since the main goals are good performance, cross-platform compatibility and support for low-performance devices [3]. Target OS: Astra Linux, Astra Linux Mobile and other Linux based systems, as well as Windows. Qt 5.15.2 LTS and its standard libraries are used for development.

To work with intelligent sensor, a special file needs to be loaded in software. This file is called electronic device description (EDD) in HART and Colibri device description (CDD) in Colibri software. EDD or CDD contain different elements such as variables, methods, menus etc. [4]. These files are used to display menus where different parameters of intelligent sensor could be monitored or changed.

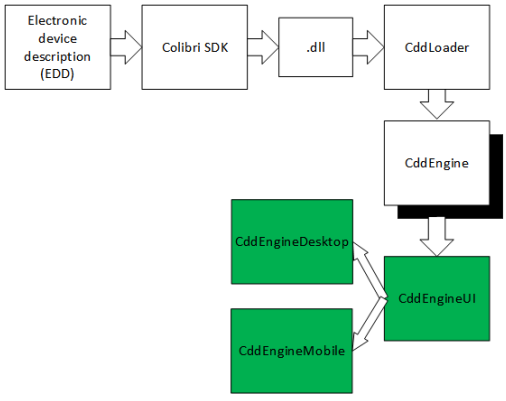

Fig. 1. The general scheme of UI creation in Colibri software

CDD is processed by a special package Colibri SDK, which generates a binary file. Then, through CddLoader, the binary file is loaded into the Colibri software, processed and transmitted to the CddEngine, which transmits it to CddEngineUI, which is responsible for creating the user interface. CddLoader, ColibriDD, CddEngine, CddEngineUI are the C++ classes of the Colibri project.

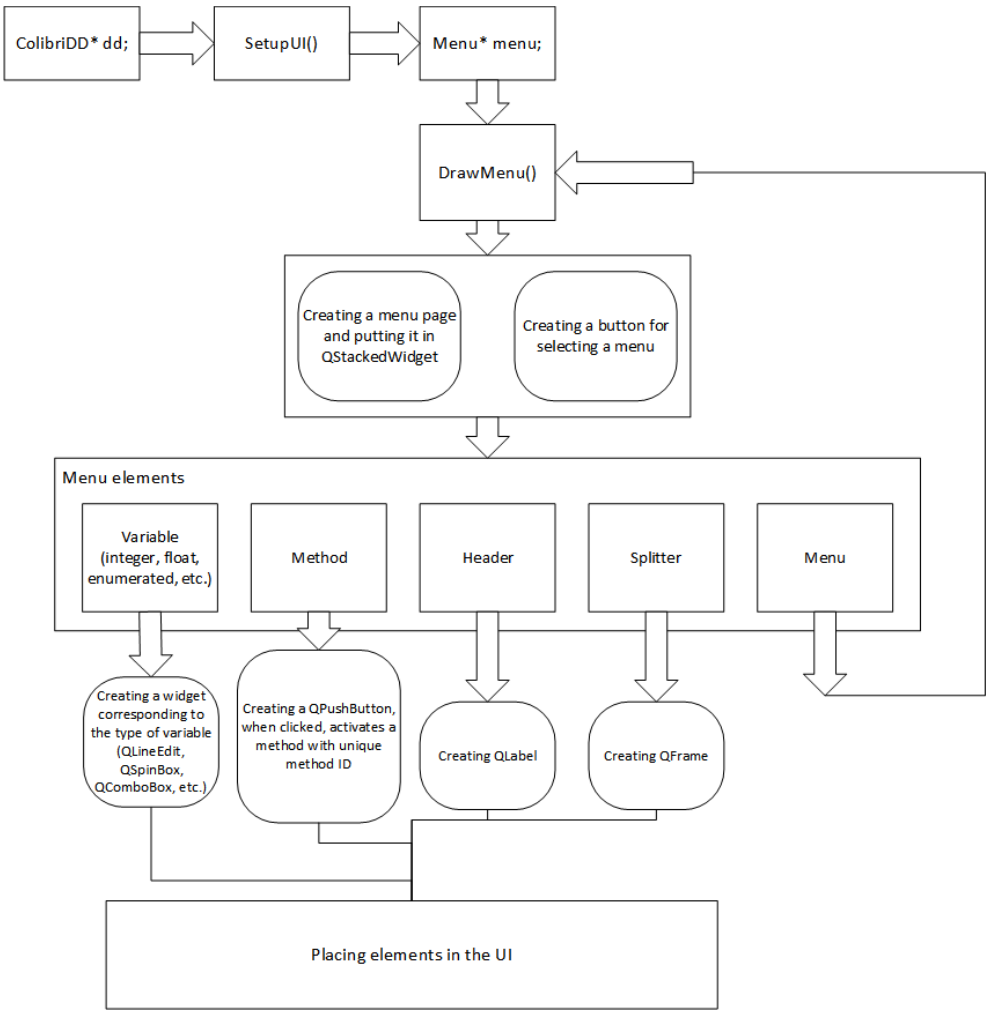

Fig. 2. A detailed scheme for creating a UI

Figure 2 shows a detailed diagram of creating a user interface from CDD files in CddEngineUI. An object of the ColibriDD class (which is created in CddEngine) is passed to the SetupUI method of the CddEngineUI class. This object contains a set of menus that can contain different types of variables, their captions, methods, headers, splitters and menus. For each menu, the DrawMenu method is called, which creates a page for this menu in the UI and a button to select this menu in the menu list. Next, the method goes through all the menu elements and performs certain actions for each type of element: for different types of variables, a corresponding widget is created that displays the value of the variable and allows to change it; a button widget is created for the method, which will call the corresponding method when clicked; for the header and a splitter QLabel with the text and a QFrame styled as a line is created, respectively; finally, if the menu item is also a menu, the DrawMenu method will be recursively called. After going through all the menus in the electronic description of the devices, all the created elements are placed in the UI and displayed to the user.

Experiment

The same CDD was tested on different operating systems with the same hardware configuration, which was achieved using Oracle VM VirtualBox. The table below shows the configuration of the Host OS and the configuration of the virtual machines.

Table 1

Test bench configuration for Host OS and VM

OS Type	**Configuration**
Host OS	Windows 11 Pro x64, Intel Core i5-1135G7 @ 4 x 2.4Ghz, 40 Gb RAM.
VM (Astra Linux SPE, Ubuntu, Debian, Win 11 Pro)	Intel Core i5-1135G7 @ 2 x 2.4 Ghz, 8 Gb RAM, Nested Paging, PAE/NX, Video memory 128 Mb

According to the test results, the following data on Colibri software memory usage was obtained when opening the device settings, which are formed from its electronic description.

Table 2

RAM usage by different operating systems when using the same CDD

OS Type	**Astra Linux SPE**	**Debian**	**Ubuntu**	**Win 11**
Experiment number	**RAM usage, MiB**			
1	2,2	1,0	2,4	5,6
2	2,3	1,5	2,1	4,8
3	2,2	0,7	2,4	4,6
4	2,3	1,9	2,4	4,6
5	2,2	1,4	2,4	4,6
6	2,0	1,5	2,5	4,7
7	2,2	1,5	2,5	4,7
8	2,2	1,3	2,6	4,5
9	2,3	2,3	2,6	4,7
10	2,0	1,6	2,5	4,6
Average value	2,2	1,5	2,4	4,7

Colibri Generic was also tested, which is identical to PACTware Generic HART DTM 1.2.0.0 and three CDD (Colibri Device Description) containing 2, 4 and 8 Colibri Generics respectively.

Colibri Generic contains 128 unique elements. These elements are identical to those of Generic HART DTM mentioned above. The detailed composition of Colibri Generic and the Qt interface elements (widgets) of the Colibri application used for display in the UI are shown in Table 3.

Table 3

Composition of Colibri Generic

Quantity	**Type of CDD element**	**Type of variable**	**Qt widget used for displaying element in the UI**
16	Variable	Integer	QSpinBox
24		Float	
50		Enumerated	QComboBox
19		BitEnumerated	QListView
3		Date/Time	QDateEdit/QTimeEdit
4		Packed, Latin_1	QLineEdit
12	Menu		QWidget

The testing was carried out on a virtual machine with Astra Linux Special Edition., the characteristics of the VM are shown in Table 1. Results are presented in the table below and in the graph (Figure 3).

Table 4

Number of elements in CDD and Colibri RAM usage

CDD type	Number of elements	RAM usage, MiB
Colibri Generic (CG)	128	4,46
2x CG	256	7,67
4x CG	512	13,86
8x GC	1024	26,47

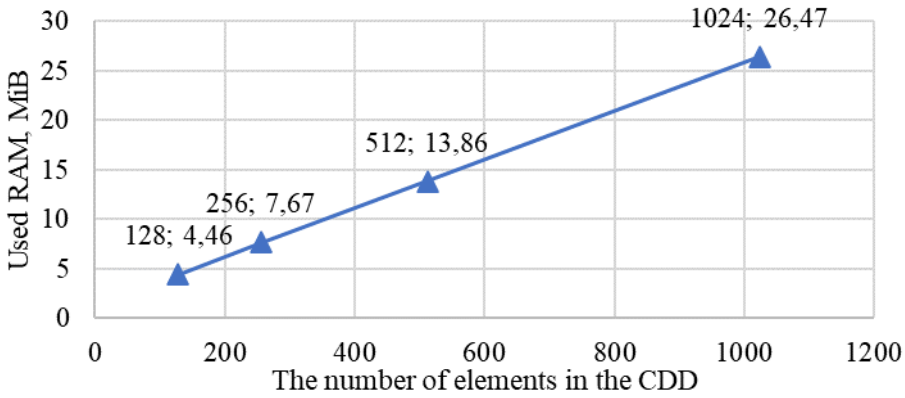

Fig. 3. RAM usage on the number of CDD elements

The linear dependence between number of CDD elements and RAM usage can be observed. In addition, 8 times Colibri Generic is almost equivalent by size to Rosemount 5300 electronic device description, so this data could help in calculation system requirement for the computers installed in the facility based on the fleet of intelligent sensors used.

Conclusion

The usage of RAM by different operating systems installed on the same hardware when working with the same electronic description was investigated.

The usage of RAM by Colibri software installed on the Astra Linux Special Edition when working with Colibri generic and CDDs twice, four times and eight times larger than Colibri generic was investigated.

A linear dependence of the amount of RAM used on the number of elements in the electronic description of the device was revealed.

Data was obtained that allows to estimate the amount of RAM required to work with a certain fleet of intelligent sensors before deploying software at a certain facility.

References

1. Продукты и решения // ООО "Аспект": Разработка цифровых измерительных систем. – URL: https://digitalmetrolog.com/ru/products (accessed: 20.02.2024).

2. Каталог оборудования – Промышленные планшеты // Mobile Inform Group. – URL: https://m-infogroup.ru/oborudovanie/planshetnye-kompjutery/ (accessed: 04.03.2024).

3. Будников А.И. Сравнительный анализ производительности реализаций инструментария Qt для языков C++ и Python // Стратегия устойчивого развития регионов России. – 2014. – №21. – URL: https://cyberleninka.ru/article/n/sravnitelnyy-analiz-proizvoditelnosti-realizatsiy-instrumentariya-qt-dlya-yazykov-c-i-python (accessed: 10.03.2024).

4. FCG_TS61804-4 Edition 2.1 EDD Interpretation // FieldComm Group Library. (n.d.). – URL: https://library.fieldcommgroup.org/61804/TS61804-4/2.1/#page=1 (accessed: 20.02.2024)

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