Development of a Telegram Bot for Receiving Data from OPC Servers

Oleg Khmelik  
department Design and Operation of Electronic Devices  
Kharkiv National University of Radioelectronics  
Kharkov, Ukraine  
oleh.khmelyk@nure.ua

Pavlo Galkin  
ORCID 0000-0002-0558-6448  
department Design and Operation of Electronic Devices  
Kharkiv National University of Radioelectronics  
Kharkov, Ukraine  
pavlo.halkin@nure.ua

Abstract—It is important to develop programming of the embedded control system which allows the real-time system to change and actually program the built-in control system based on the PLC. Chatbot is a computer program developed on the basis of neural networks and machine learning technologies, which can be used to communicate in audio or text format, but chatbot can also be used to program embedded systems without changing the PLC program itself. The purpose of the work is to develop a chatbot in the Telegram service, which can be used to obtain information from embedded systems based on the OPC server and send the necessary information at the request of the user.

Keywords—OPC, PLC, Telegram bot, Chatbot, embedded control system, data, servers, IoT, IIoT, machine learning technologies.

I. INTRODUCTION

Embedded system - a specialized computer system or computing device designed to perform a limited number of functions, often with real-time constraints. The purpose of the work is to develop a chatbot in the Telegram service, which can be used to obtain information from embedded systems based on the OPC server [1] and send the necessary information [2] at the request of the user. An analysis of platforms and technologies was conducted [3-9]. Among other representatives of chatbots this development is unique, since in the Internet in free access there was no similar implementation of the chatbots programmed for this purpose which transferred information from the PLC [3] through the OPC server [4].

Development of telegram-bot for OPC-technology of programming of the embedded control system on the basis of the PLC. Thus, it is advisable to develop a programming system for built-in control system based on PLC.

II. DEVELOPMENT OF THE SCHEME OF DATA EXCHANGE THROUGH THE OPC SERVER

Data exchange of embedded systems based on the OPC server is as follows (fig. 1).

- the configurator itself is configured;
- information from the controllers is sent to the OPC server;
- SCADA-system receives data from controllers by reading data via OPC.

III. BOT DEVELOPMENT

The first step in programming a chatbot is to obtain its personal token. The personal token is a unique number that the chatbot works on, without it the program will not know through whom requests will be processed. The first step in programming a chatbot is to obtain its personal token. The personal token is a unique number that the chatbot works on, without it the program will not know through whom requests will be processed (Fig. 2).

The token is registered with the command "bot = telebot.TeleBot ('*')", where "*" is your token. This command sets the token for our telegram bot, without it it will not work.

When we need something from the chatbot, we prescribe a command in the chat and the result is displayed to us. '@Bot.message_handler (commands = ['start'])' with this command we set the necessary condition to start working with the bot, namely, when the user types in the chat '/start'.
Next, we specify what will be executed when the user enters the required command (Fig 3).

To read our variables from an embedded system based on OPC, there is a function "opc.properties (*)", where "*" is the name of our variable, which we learn thanks to the OPC Client utility. To output data from the OPC, we use the binding of the variable to the command. After receiving the data at the end of the function it is necessary to prescribe "opc.close ()" to optimize the use of the connection to the OPC server, without this command after performing the function of connection to the embedded system based on the OPC server does not stop, and re-execution of the command will add more 1 connection without disabling the past. Therefore, it is better to close the communication channel (Fig 4).

The basis for testing our development of a chatbot for embedded systems based on OPC will be a project in PC WORX with emulation thanks to PC WORX SRT software (Fig. 5).

IV. CONCLUSION

This development Telegram bot is a unique way of user interaction on programming by embedded control systems and exchange of information between them. In the future, the development of this technology can provide remote work for both users and developers of embedded systems for their programming, monitoring and debugging.

REFERENCES