

TCL/TK Based API for OMRON'S RS485 Interface Network

D. Batchvarov, V. Geortchev, A. Boneva, R. Krasteva

*Central Laboratory of Mechatronics and Instrumentation-BAS, 1113 Sofia
E-mail: veso@bgcict.acad.bg*

1. Introduction

RS485 is commonly used interface for electronic devices of OMRON. Hardware realization of the protocol corresponds to IEC Standard (Compo Way/F Communication Protocol) [1]. Software logical protocol is based on Omron's internal standard and includes large number of application and special commands. There exist seriously difficulties about the system functionality testing of a group of devices, cooperating into a network. The activation of the final system adds new real time depending on errors sources, such as signal breaking, devices errors, slow processing of data, deviations of electrical communication line parameters. The errorless system is not exist, but the minimizing of numbers of errors is possible if the operator has ability for detect, watch and analyze the communication traffic.

The API makes odd, static and dynamic diagnostics. And more, it includes a library, consisting a function for each application command, methods for it's processing and "logging" of it's status and results. All record information is accessed "on line" by scrollable message box.

2. OMRON RS485 Specifications

OMRON connects the micro controllers, joining into LAN (Local Area Network), using interface RS485. Each one of these devices acts as "slave" and one computer-IBM PC is included as "master". The network is electrically built as 2-wire

differential line having ability for a connection to 32 units with a length to 1000 m [1, 3]. From the device side view, the interface is RS232C and using adapters the signals are converted to the necessary levels. The synchronization of the traffic is possible only if the communication is working in half-duplex mode, that is and the case. The OMRON logical solution is based on packages exchange between the “master” and the “slaves”. These “packages” have fixed length and are included into predefined frames. The data-integrity and validity are controlled by “check sum bytes”. All data are digital and are present in ASSCII coded BCD (Binary Converted to Decimal) format.

The packages structures are depended by the direction of the transfer (from “master” or to “master”) [1]:

- from master @AADCCYYXXXX ”CS” ”CR”;
- to master @AADCCYYXXXX ”CS” ”CR” or @AADYY ”CS” ”CR”.

The first case in “to master” is when the device answers correctly and the second by received error.

The chars denotes different fields in the frame and are described in the Table 1.

Table 1

Symbols	Range	Functions
@	0x40	start char
AA	00 to 31	unit address
D	0.1 or 2	communication mode
CC	00–99	code of operation
YY	00–99	error code
XXXX	0–1999	data field
“F”XXX	–1XXX	data field
“E”XXX	–0XXX	data field
CS	00–FF	check sum
CR	0x0D	terminator char

The data and error fields are context dependent.

There exist a large number of predefined application commands:

- for reading and writing into PV/SP fields;
- for setting process control constants;
- for defining of the limits;
- for receiving of the over values.

There exists also a special class of commands (for direct control and communication processing).

3. Typical distributed control system based on OMRON RS495

The typical distributed control system includes IBM PC (master, Operation station) and other devices (slaves, digital regulators), connected to LAN [2].

Fig. 1 shows block diagram on distributed control system.

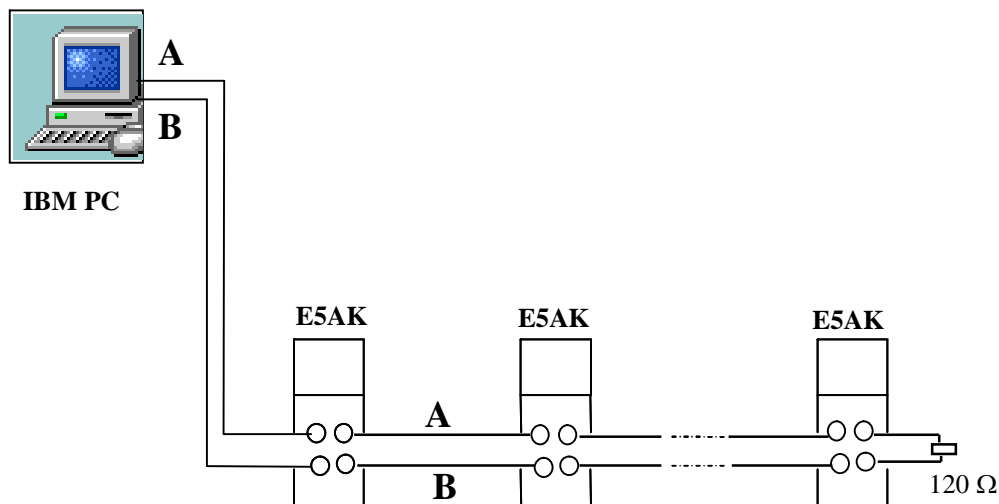


Fig. 1. Distributed control system (E5AK-OMRON digital process controller; A, B – half duplex communication lines; 120 Ω – terminate resistor)

The typical time cycle consists of two phases [4]:

- Request from “master” to any “slave” (this can be instruction for setting, reading or special command);
- Answer to the request from the selected “slave” to the “master” (this consists answer and status information from asked device).

Initially, the “master” configures all “slaves”. After, it has ability to change the configuration personally and so to change the algorithm (control law) of each device. Normally, it changes the SP (Set Point) only.

There exist three problems of the Control System:

1. The first one – what has to do when any device is damaged?
2. The second one – about real time processing and validity control of incoming data from the “slaves”.
3. The third one – about the approach for realization of the transfer in real time by the “master”.

In this paper an attempt is made to decide these problems by building API on the “master”, as program extension with ability for integration.

The API, named “Control panel”, is designed using script language Tcl/Tk [5]. It is platform independent, because Tcl/Tk have such properties. There exists realizations for Windows /95/98/NT, UNIX, LINUX and others.

Fig. 2 shows a view of the control panel.

The API includes DTBS (Data Base) and dedicated LIBRARY for control program design process .

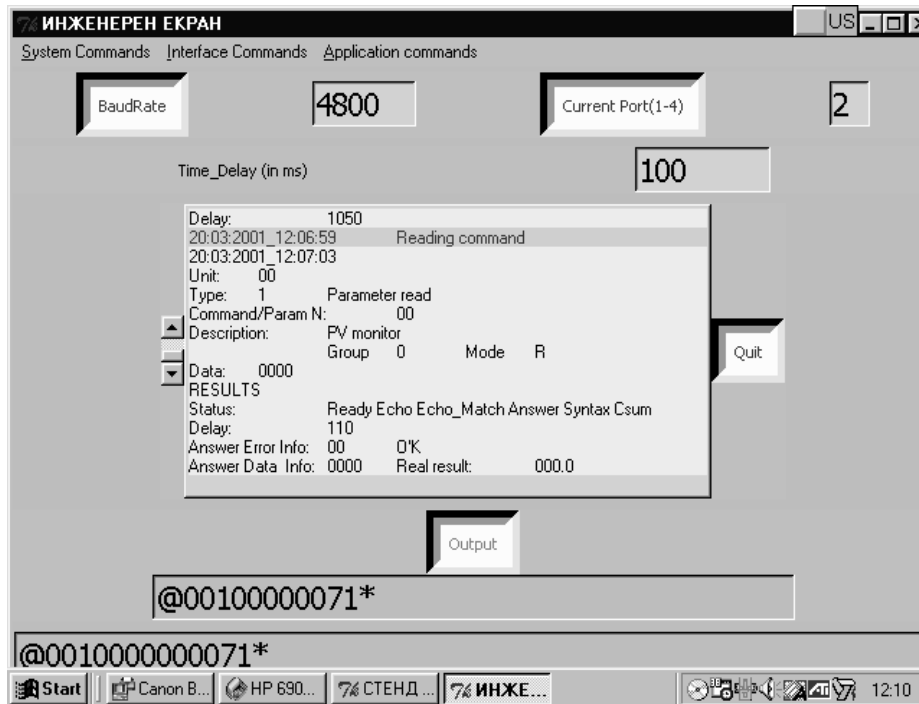


Fig. 2. Panel view

The operator uses suitable Graphical interface including the Pull-Up menu, Text Scroll Box, Active Buttons and other View Get objects.

All operator and program actions and status information are recording into the Text Scroll Box and are accessed any time from the operator.

The API works in Run and Test modes. At the last, it generates local commands to the slaves and return results. At the RUN mode it executes control and registration functions according to the main control program.

All control and test tasks use a common DTBS and Library. As result, application programs generate automatically history data files [5].

4. Tcl/Tk and using program technique

Tcl/Tk is a script language allowing the user access to some commonly platform and custom designed libraries [5]. The language have flexible syntax and good debug utilities. The access to different platform objects (View Gets) and interpreter model

of executing of the programs makes it very useful for building of real time graphical applications, as API and testing software. Post-processing utility MKTCLAPP, creating final *.exe or *.dll final files, equalize Tcl/Tk power to C⁺⁺. And more, this program utility allows to user mix of Tcl/Tk and C⁺⁺ program files, libraries and DLL.

Tcl/Tk consists of program tools for binding a large number of events, including in chars stream processing. There exists operator “FILE EVENT”, connecting receiving (in input communication buffer) of pre-determinate char for EndOfRecord to responding event processing procedure. The last tests data validity of the record (after reading it from the input buffer) and converts it to internal program format (if data are valid). The Tcl/Tk interpreter invokes this procedure in background mode. After it's end the foreground tasks continue normally. This routine exports some messages to the Windows manager, placing them into TextScrollBar.

The Tcl/Tk has power tools for char strings processing, based on REGULAR EXPRESSIONS. With them the string can be test for matching of correct sequence of sub strings, filter them and calculate final data at the time of receiving.

Really, the each input is logically connect to respond output. After the output from the master to any slave was ended the master starts for waiting of input. But other active processes, as active buttons, scrollbar, entries, and menus have to have ability for the getting processor time. Because of that all active processes are executed in background mode.

The Pull-Up menu is graphical tool for requesting from the operator to the master executing any application command in the Test mode. In the target window the operator enters responded parameters and starts the command. The result of commands are recording in the TextScrollBar.

Each master application can execute a application command too. For that it sends to procedure Execcommand as argument a list of data including cod of command and parameters. The action is similar to that of Test mode. The only exception is that the result is return to structural array named Command (global Tcl/Tk array). All Execcommand activities are processed in background mode, but it same is started in foreground mode. Tcl/Tk has abilities for excluding foreground for some time, while was not any global variable had set to new value. When the foreground tasks are waiting, the background left the active (that is the case).

5. Application of command data base

The LAN consists of different devices, each of them having a large number of application commands. These commands have similar structure, but their fields, arguments, actions, results are very specific and concrete processing reflects to Command Data Base.

Command Data Base is organized as an associative area, where Command name is index to other parameters of the command:

- code of operation,
- number of arguments,
- times (scan time, delay time, processing time),
- dimension data,
- limits,
- error messages responding to return error code,
- command structure information (for query and answer data).

Command Data Base directs communication software, that has ability for realization of common interface, common query and answer structures. Other possibility is the diagnostic function (realized by the power utilities building in Tcl/Tk, as Regular Expression mechanism). It allows the testing by prototype “match string” matching of the results. These “match strings” are recorded into Command Data Base and are accessed by command index.

6. Conclusion

The presented in the paper API allows LAN design, connecting different devices, using RS485. The used approach offers to users an ability for control, test, diagnostic and data log of functions on distributed process control system, realized in network structure.

The used language Tcl/Tk gives possibility to the system for working under different platforms.

R e f e r e n c e s

1. OMRON, Digital Process Controller Series E5K, Cat. No. GCTC12, OMRON, USA, 1999.
2. B u t c h v a r o v, D., G. S t a i n o v, A. B o n e v a, K. S t a n i s c h e v. Precise gas flow measurement system. – In: Symposium “MECHATRONICS and ROBOTICS’98”, Drjanovo, Bulgaria, 10–12 June, 1998. Proceedings “Scientific reports”, No 4 (26), Section 4: Automation, 4.1 - 4.8 (in Bulgarian).
3. OMRON, K3GN 1/32 DIN Digital Panel Meter, User’s Manual, Cat. No N102-E1-1, 1995.
4. Magic 485F Instruction Manual Part No 85993374 Issue A1, Prepared by James Chisholm, Amplicon Liveline Limited, 1995.
5. O u s t e r h o u t, J. Tcl/Tk Engineering Manual, Sun Microsystems Inc., 1994.

Прикладной интерфейс на базе TCL/TK для сети OMRON RS485

Д. Бачваров, В. Гьорчев, А. Бонева, Р. Крестева

Центральная лаборатория мехатроники и приборостроения, 1113 София

(Р е з ю м е)

Описывается применение языка TCL/TK для создания интерфейсной программы для OMRON RS485. Рассмотрены несколько тем: интерфейсную спецификацию, реализацию сетей при помощи этого интерфейса, построение софтвера как расширение Win95/98/NT и технику TCL/TK. Логические структуры баз данных тоже представлены.

Разработанный софтвер включен в разных проектах на основе RS485 интерфейса, реализующий распределенную систему управления.