KIT-Builder – Development Environment

Introduction

The KIT-Builder development environment provides a fast way to create and debug simple applications, it is designed for the KITV40, KIT386EX control units used with TERM01, TERM05, TERM06 terminals, for TERM10 industrial terminals, TERM10/VP visualisation panels or for the KOMPAKT control system, provided the task does not require extremely short time responses.

KIT-Builder

The KIT-Builder development environment allows writing programs in the KIT-Basic proprietary language, derived from a simplified structure of Pascal or Basic and complemented by elements from languages designed for programmable automated machines. KIT-Builder includes the following separate components:

- **KBDLCD** - Graphic design of the TERM10B or TERM01, TERM05, TERM06 screens, integrating the KIT-Builder environment into a single entity,
  PC platform, Windows95 operating system and higher
- **KBDComp**KBDComp – Compiler of the KIT-Basic language to the so called p-code,
  PC platform, Windows95 operating system and higher
- **KBDProc**KBDProc - Simulator-Interpreter of the p-code, including TERM10B a TERM01 keyboard and screen simulators,
  PC platform, MS-DOS (can run in a MS-Windows window)
- **KBDCON**– Loader of programs to the KITV40, KIT386EX, TERM10B, KOMPAKT configuration set, allows on-line monitoring of the program's progress, reading archives to PC
  PC platform, Windows95 operating system and higher
- **KBDProc**- Interpreter of p-code in the KITV40, KIT386EX, TERM10B, KOMPAKT configuration, SofCon HW platform

Program, its Processing and Components

The KB environment has approached programmable automat units by simplifying the program processing. The processing consists of a basic, periodically repeated scan, composed of the input read section, the MAIN program processing, the write of outputs and the system services. Having any time the option to use a special wait feature to interrupt the main program, the KB has retained the capacity of program processing known to the more sophisticated programming languages (PASCAL, C, BASIC). If the system fails to be served within 1 s, the program runs the WATCHDOG feature and resets the control unit.

When first run after a power failure, the program runs the INIT section (if specified in the program), allowing to define the service for initializations using ordinary program commands. For fast or regular responses, you can define the FAST10 procedure to be called regularly every 10ms. This procedure can also include reads or writes of fast inputs or outputs. A part of the program itself is a hardware configuration, performed very simply by specifying the name and the board's parameters in one of the program's lines.

Memory Processor

The KIT-BASIC language allows working with two data structures – the 2000 byte integer data structure, and the real number array in size of 250 real numbers (4 + 2 byte). In the integer data structure, the user can place any combination of overlapping variables of the bit, byte, word, integer type (2 bytes in size), longint (4 bytes), and string (by default, 20 bytes in size). Each variable allows an absolute or a symbolic definition. The program also allows using constants that can be defined in the CONSTANT section.

Standard Operators, Functions and Procedures

To describe its own algorithms, the language uses standard commands found in Pascal or Basic (if-then-else; case-of-else-end, repeat-until, while-do, for-to/downto-do), the procedure call command (without parameters). To save the new value in the register, the language uses the standard Pascal assign command (:=). For expressions, you can use any usual relational, logical and arithmetic operators (<, >, <=, >=, <>, =, and, or, xor, not, shl, shr, + , -, *, /, div, mod). The conversion between the bit, byte, word, integer, longint and real types is performed automatically. The usual real functions (sin, cos, sqrt atd.) are defined for the real type variables. In addition, you can also use special system functions and procedures, archive or terminal services that are described in the manual.

Terminals

If the control system includes a terminal, this device is operated parallel to the program processing using functional blocks – procedures describing each screen (each procedure corresponds to one of the screen). Screen definition commands (bitmap, font, position, print, edit, point line, rect, fill, circle, bar, onkey) allow to combine text, graphical objects, value edits on the screen, and to define the keyboard operation or tinge the screen using a bitmap. You can select the current screen (for example, the ALARM screen) directly from the program, or as a response to a key pressed on the terminal. You can perform graphic design of the screen using the KBDLCD program.

Timers, PID Controllers, Archives,...

Other predefined objects include the TIMER object with the basic 10ms period (on startup, the timer content is incremented, maximum counting period of the timer being approximately 248 days), the real time clock, the PID controller object allowing to run the regulation process on the background, and the ARCHIVE object. The archive object enables to simply archive values in the back-up memory. You can read each archive item using the communication line from the connected PC, or you can view the archive directly on the terminal's screen, provided it is included in the system.

Communications

The KB environment also allows to operate communication lines. You can use drivers for the COM general channel, (RS-232, RS-485) LECOM protocol, the PRT protocol (SofCon's proprietary protocol), modem communications, etc.

How to Create Applications

For an example of how to create an application, see KIT-BUILDER - Easy Way to Create Applications. To create his/her own visualisations on PC, the user can use the LnkSofMA.DLL Communication Library to communicate with the SofCon control system.

Conclusion

KIT-Builder is a comprehensive development tool, enabling a fast, simple and effective way of creating a number of applications, designed, above all, for system control in the area of heating, air-conditioning, simple machinery, etc.

Application creators and users who are not professional programmers tend to be more productive with this tool than Borland Pascal / os ReTOS.