

VISONIK®
COLBAS

Function sheet



COLBAS is the Landis & Staefa programming language for the VISONIK BPS. The abbreviation COLBAS stands for:

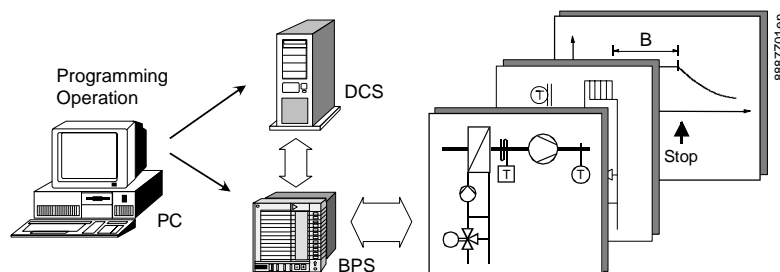
Control Oriented Language for Building Automation Systems.

COLBAS is based on the programming language BASIC, but contains additional, special functions for solving measuring and control tasks.

Use

Implementation

COLBAS is implemented in the Building Process Station (BPS) and in the Data and Communication Server (DCS). However, in the DCS, COLBAS is only used for special requirements and interfaces to other systems and is thus not covered in the following.



Tasks

The BPS is designed to control and monitor a wide variety of building services systems both simultaneously and in real time. Its functionality is often expanded through requirements such as: Optimized plant operation, reporting, fault handling and alarm reporting, data acquisition, etc.

Implementation

All the tasks given above are defined and implemented using COLBAS programs. COLBAS has the following additional functions compared to BASIC:

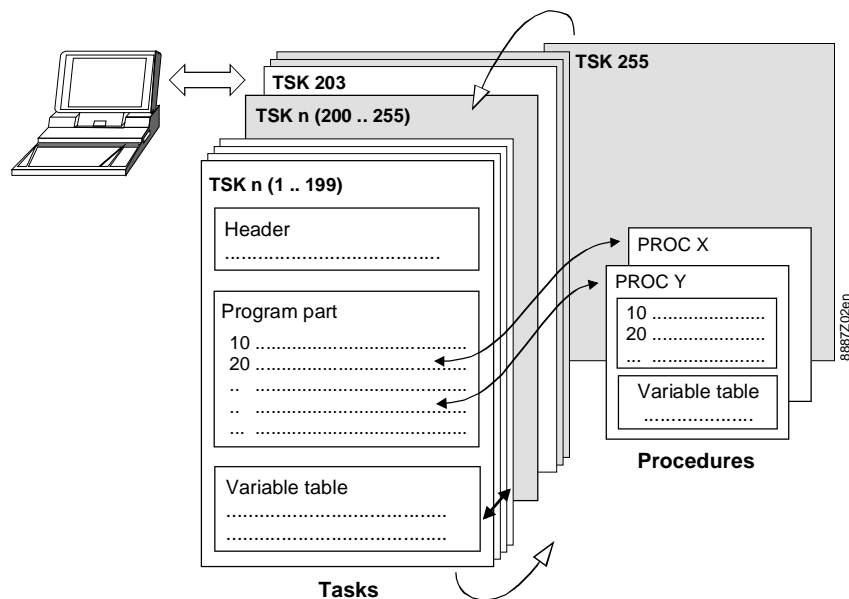
- Access to process values either manually or from the program (immediate mode or deferred mode).
- Parallel processing of programs (multitasking) either independently or synchronized mutually.
- Real-time behavior, i.e. programs react immediately to external events and can run synchronously to real time.

COLBAS programs are written with the support of PC software containing program libraries for the most common applications. Special requirements can be solved with individual COLBAS programs.

Program structure

Process and instrumentation diagram

The following diagram shows a simplified program structure in the BPS.



Program elements

The program elements shown above are:

Element	Explanation
General information on the task	<p>A task (TSK) is an independent program unit in a COLBAS program. Tasks can be:</p> <ul style="list-style-type: none"> – Individually generated, edited, and deleted. – Processed parallel and independently from one another, and started and stopped by other tasks. – Exchanged among each other and with the process data. <p>Each COLBAS task contains the following parts:</p> <ul style="list-style-type: none"> – Header with the general task specifications and the variable declaration. – Program part with the individual instructions. – Variable table with the task variables (current values).
TSK 1..199	<p>Tasks freely available for the plant operating program, whereby certain task numbers are intended for certain tasks such as:</p> <ul style="list-style-type: none"> – TSK 5 for plant coordination. – TSK 10, TSK 20, TSK 30, etc. for the individual plants. – TSK 160 for outputting local fault messages.
TSK 200..255	<p>Tasks reserved for: Data exchange and reports, time switch programs and catalogs, telephony, POP card operation, COLBAS procedures (TSK 255).</p> <p>These tasks are not accessible, with few exceptions such as: Basic tasks TSK 201 .. 203 to operate the BPS using COLBAS instructions in immediate mode via TTY1 .. TTY3.</p>
Procedure	<p>A COLBAS procedure is an independent program unit with its own table of variables. The COLBAS procedure library contains procedures for a large number of routine tasks such as:</p> <ul style="list-style-type: none"> – Fan control. – Control of recirculated air dampers. – Room temperature cascade control for temperature or humidity. – Fault message printouts. – etc. <p>Procedures must be called by the superposed task.</p>

Instructions

Instruction structure

A COLBAS instruction usually begins with a keyword followed by a supplementary element. The keyword tells which actions should be carried out; the supplementary element gives additional information.

Example: RUN 3 (Start task 3).

RUN = Keyword. 3 = Supplementary element, here the task number.

Available instructions

COLBAS recognizes the following three groups of instructions:

Group	Definition / examples
Statements	Keyword for deferred mode (entry with line number): <ul style="list-style-type: none">– Assign variables and I/O point values: LET– Output statements: PRINT, PLOT.– Entry statements: INPUT, INKEY.– Branchings: GOTO, IF .. THEN .. ELSE.– Program loops: FOR/NEXT, GOSUB/RETURN– Access to data lists: DATA, READ.– Real-time instructions: WAIT, WAIT UNTIL, WAIT FOR.
Functions	Rules for calculating a new value from predefined values: <ul style="list-style-type: none">– Arithmetic functions: MOD, EXP, LOG.– Trigonometric functions: SIN, COS.– Logical and Boolean functions: AND, OR, x < y– String functions: A\$ + B\$– Control functions (DDC): CONTROL.– Monitoring functions: IF MW > 20 THEN.– Calendar functions: DAY, DDAY, YEAR.
Operating command	Keyword for immediate mode (entry without line number): In addition to the BASIC commands such as COPY, DEL, DIR, END, LIST, RUN, STOP, etc., COLBAS implements further commands: They are used chiefly for system configuration, editing tasks and procedures, and for access to the process variables.

COLBAS programs can be edited (created, modified, deleted) online, i.e. directly from the BPS or from the DCS. The programs in the BPS are saved in battery-buffered RAM memories.

Task example

A task consists of a maximum of 350 lines. Each line begins with a line number (1 .. 32767), followed by one or more COLBAS instructions. The following problem is solved using the task given in the example below:

- The measured value at I/O point \$022 is acquired at intervals of 15 minutes during a 24-hour period.
- The time and date of the acquisition are also saved.

```
TSK 33
  5  REM "ACQUIRE TEMPERATURE AT $022.MW"
 10  DIM M(97)
 20  DIM T(97); DIM D(97)

1000 FOR J=1 TO 97
1100 M(J) := $022.MW
1200 T(J) := TIME; D(J) := DATE
1300 WAIT DURING 00:15
1400 NEXT J

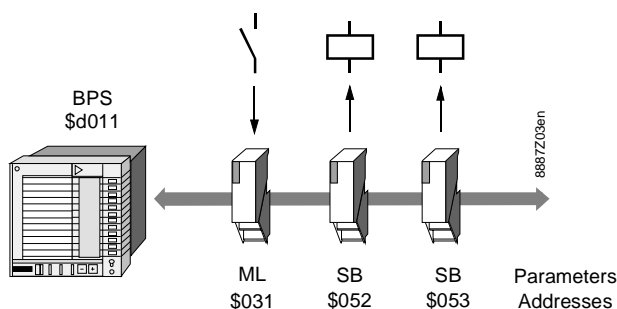
2000 END
```

The acquired data can now be evaluated using another program.

Process access

I/O point addresses and parameters

An important requirement to control building services systems is that the process variables and/or I/O points can be accessed either by the operator or via the program. In COLBAS this data is accessed via addresses and parameters.



Example

This example illustrates how the I/O points shown above are accessed by the program:

```
.....
3000 Mz := $031.ML
4000 IF Mz=1 THEN GOTO 8000 ELSE GOTO 9000
.....
8000 $052.SB:=1
.....
9000 $053.SB:=0
.....
```

Explanations of the examples

The following instructions are given in the individual lines:

Line	Explanation
3000	The I/O point with address \$031 (signaling point) reads the state of parameter ML and assigns the variable Mz.
4000	If variable Mz=1, then go to line 8000, else go to line 9000.
8000	The I/O point with address \$052 (switching command) is assigned value 1, i.e. switching command = ON.
9000	The I/O point with address \$053 (switching command) is assigned value 0, i.e. switching command = OFF.

Note

Note the difference between an allocation with "==" (lines 3000, 8000 and 9000) and the logical comparison for equivalence with "=" (line 4000).

INPUT and OUTPUT points

When accessing the data described above, distinguish between INPUT points and OUTPUT points.

- Signaling points, measured value points, and counter value points are INPUT points. The main value for these points is read-only. This main value cannot be overwritten either manually or via the program unless the point in question has been set to "fictitious" for testing purposes.
- Switching command and positioning command points are OUTPUT points. The operator and the program can **both** read and write the main value for these points.

Access to the whole system

COLBAS allows all the states and parameter values to be read for all the data points configured in the VISONIK system. The address entry for I/O points outside the system's "own" BPS must contain both the point address and the BPS number, for example \$d011'031 to call up the signaling point shown above.

Additional information

Refer to the following documents for more information on this topic:

Document No.	Title
CM2Z8387en	Language Description COLBAS
CM2T8567E	VISONIK System Basics

© 2001 Siemens Building Technologies Ltd.