RCSL No: 42-i1780

Edition: July 1981

Author: Erik Jeppesen

Title:

A Survey of COMAL80
for the RC700 Microcomputer
- Piccolo -

A/S REGNECENTRALEN
af 1979

Keywords:

COMAL80, RC700, survey.

Abstract:

This document contains a survey of the COMAL80 system implemented to the RC700 microcomputer.

(20 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

FOREWORD

First edition: RCSL No 42-i1714.
This document contains a survey of COMAL80 on the RC700 microcomputer, Piccolo.

COMAL80 was proposed by a working group including manufacturers, schoolteachers and university people in Denmark during 1979/80.

The language implemented by RC Computer is a superset of the COMAL80 which again can be seen as an extension to BASIC. COMAL80 however contains facilities which makes it more comparable to PASCAL than to BASIC.

There will be facilities in COMAL80 which are not described in this document. For instance there will be statements/functions for accessing the RS232-port on the Piccolo which makes it possible to use devices connected by the user (such as plotters, digitizers, etc.) and to communicate with other Piccolo-systems in a transparent mode or by means of a protocol for transmission of binary data.

It should be mentioned that: Variable-names may consist of up to 16 characters; that arithmetic operations are carried out to a significans of 13 decimal digits, and that numbers are in the range 10E-128 < n < 10E127.

The proposal published by the COMAL80 working group contains more details about the language. It may be ordered from RC Computer (RCSL No 42-i1516).

Erik Jeppesen
A/S REGNECENTRALEN af 1979, March 1981

Second edition: RCSL No 42-i1780.
Some minor changes and extensions concerning the file system.

Erik Jeppesen
A/S REGNECENTRALEN af 1979, July 1981

TABLE OF CONTENTS PAGE

1. COMAL80 STATEMENTS 1.

Format/Description Use

```
⎧ <var>  ⎫          ⎧ <var>  ⎫
⎨        ⎬:= <expr> ;⎨        ⎬ := <expr>   ...
⎩ <svar> ⎭          ⎩ <svar> ⎭
```

Assigns the value of an expression to a variable.

STATEMENT or COMMAND

```
<name>:
```

A label that can be referenced by a goto-statement.

STATEMENT

```
  CASE <expr> OF
  WHEN <expr> [,<expr>] ...
    <statements-1>

⎡⎡  .                           ⎤⎤
⎢⎢  .                           ⎥⎥
⎢⎢  .                           ⎥⎥
⎢⎢WHEN <expr> [,<expr>] ...     ⎥⎥
⎢⎢  <statements-n>              ⎥⎥
⎢⎣                              ⎦⎥
⎢⎡OTHERWISE                     ⎤⎥
⎣⎣  <statements>                ⎦⎦
  ENDCASE
```

The expression following CASE is evaluated and compared with the expressions following WHEN. If there is a match in the i'th WHEN statement, statements-i are executed. If no match is found the statements following OTHERWISE are executed if specified. If OTHERWISE is not specified and no match is found then an error occurs.

STATEMENT

CHAIN <filename>

Runs the SAVE'd program referred to by the filename when the statement is encountered in the user's program. STATEMENT

$$\text{DATA}\begin{Bmatrix}\text{<val>}\\ \text{<slit>}\end{Bmatrix}\left[,\begin{Bmatrix}\text{<val>}\\ \text{<slit>}\end{Bmatrix}\right]\ \ldots$$

Provides values to be read into variables appearing in READ statements. STATEMENT

$$\text{DIM}\ \begin{Bmatrix}\text{<array>(m)}\\ \text{<array>(<row>,<col>)}\\ \text{<svar> [(m)] OF <expr>}\end{Bmatrix}\left[,\begin{Bmatrix}\text{<array>(m)}\\ \text{<array>(<row>,<col>)}\\ \text{<svar>[(m)] OF <expr>}\end{Bmatrix}\right]\ldots$$

Defines the size of numeric arrays, string variables and string arrays. STATEMENT or COMMAND

END

Terminates the execution of a program. STATEMENT

$$\text{EXEC <name>}\ \ \text{[(}\begin{Bmatrix}\text{<var>}\\ \text{<svar>}\\ \text{<array>}\\ \text{<sarray>}\\ \text{<expr>}\end{Bmatrix}\left[,\ \begin{Bmatrix}\text{<var>}\\ \text{<svar>}\\ \text{<array>}\\ \text{<sarray>}\\ \text{<expr>}\end{Bmatrix}\right]\text{...)]}$$

Executes a procedure defined by PROC-ENDPROC. STATEMENT

FOR <control var>:= <expr1> TO <expr2>
[STEP <expr3>] DO <statement>

Executes a single statement a number of times. STATEMENT

```
FOR <control var>:= <expr1> TO <expr2> [STEP <expr3>] DO
  <statements>
NEXT <control var>
```

FOR begins a FOR-NEXT loop and defines the number of times a block of statements is to be executed. NEXT is the last statement in the loop and changes the value of the control variable. STATEMENT

$$\text{GLOBAL} \left\{ \begin{array}{l} \text{<var>} \\ \text{<svar>} \\ \text{<array>} \\ \text{<sarray>} \\ \text{<proc name>} \end{array} \right\} \left[, \left\{ \begin{array}{l} \text{<var>} \\ \text{<svar>} \\ \text{<array>} \\ \text{<sarray>} \\ \text{<proc name>} \end{array} \right\} \right] \ldots$$

Makes variables in a closed procedure global. STATEMENT

```
GOTO <name>
```

Transfers control unconditionally to the statement following the label-statement specified by <name>. STATEMENT

```
IF <expr> THEN <statement>
```

Executes a single statement depending on whether an expression is true or false. STATEMENT

```
IF <expr> THEN
  <statements>
ENDIF
```

Executes a block of statements depending on whether an expression is true or false. STATEMENT

```
IF <expr> THEN
  <statements-1>
ELSE
  <statements-2>
ENDIF
```

Executes statements-1 if an expression is true, otherwise statements-2. STATEMENT

$$\text{INPUT [<slit>:]}\ \left\{\begin{matrix}\text{<var>}\\\text{<svar>}\end{matrix}\right\}\left[\,,\left\{\begin{matrix}\text{<var>}\\\text{<svar>}\end{matrix}\right\}\right]\ldots\left[\left\{\begin{matrix},\\;\end{matrix}\right\}\right]$$

Assigns values entered from the keyboard to numeric or string variables.

STATEMENT

MARGIN <expr>

Defines the number of characters to be output before a new-line is automatically generated.

STATEMENT
or COMMAND

$$\text{PRINT}\quad\text{<expr>}\left[\left\{\begin{matrix},\\;\end{matrix}\right\}\ \text{<expr>}\ \right]\ldots\ \left[\left\{\begin{matrix},\\;\end{matrix}\right\}\right]$$

Prints specified items on the selected output device.

STATEMENT
or COMMAND

PRINT USING <format>: <expr>[,<expr>] ... [,]

Outputs the items in the argument list using a specified format.

STATEMENT

$$\text{PROC <name>}\quad(\left\{\begin{matrix}\text{[REF|<var>}\\\text{[REF]<svar>}\\\text{REF <array>([,])}\\\text{REF <sarray>()}\end{matrix}\right\}\left[\,,\left\{\begin{matrix}\text{[REF]<var>}\\\text{[REF]<svar>}\\\text{REF <array>([,])}\\\text{REF<sarray>()}\end{matrix}\right\}\right]\ldots)\quad\text{[CLOSED]}$$

<statements>

ENDPROC <name>

Defines a procedure or a function. When the procedure or function is called, control is transferred to the first statement following PROC. ENDPROC is the last statement in a procedure or function and returns control to the first statement following the EXEC statement (procedure call) or to the statement containing the function call.

STATEMENT

RANDOMIZE

Causes the random generator to start at a random point in the sequence of random numbers generated by the RND function. STATEMENT

READ $\left\{\begin{array}{l}\text{<var>}\\ \text{<svar>}\end{array}\right\}\left[,\left\{\begin{array}{l}\text{<var>}\\ \text{<svar>}\end{array}\right\}\right]\ldots$

Reads in values from DATA statements and assigns the values to the variables listed in the statement. STATEMENT

REPEAT
  <statements>
UNTIL <expr>

Executes a block of statements repetitively until an expression is true. The block of statements is always executed at least once. STATEMENT

RESTORE [<name>]

Resets the data element pointer to the beginning of the data list or to a labelled DATA statement. STATEMENT

SELECT OUTPUT <device>

Selects the device or file to which output from a program (PRINT or PRINT USING statement) shall be directed. STATEMENT or COMMAND

STOP

Terminates execution of the current program. STATEMENT

WHILE <expr> DO <statement>

Executes a single statement repetitively while an expression is true. If the expression is false the first time the WHILE statement is executed then the statement following DO is not executed. STATEMENT

WHILE <expr> DO
  <statements>
ENDWHILE

Executes a block of statements repetitively while an expression is true. If the expression is false the first time WHILE is encountered, the block of statements is not executed even once.

STATEMENT

ZONE <expr>

Sets the zone spacing between elements output by PRINT statements.

STATEMENT
or COMMAND

2. COMAL80 FUNCTIONS 2.

ABS(<expr>)

Returns the absolute (positive) value of the argument.

AT(<expr>,<expr>)

Used in PRINT statements to position the cursor to the XY-position evaluated from the arguments.

ATN(<expr>)

Calculates the angle, in radians, whose tangent is the argument.

CHR$(<expr>)

Returns the character corresponding to the number found as the argument modulo 256.

COS(<expr>)

Calculates the cosine of an angle which is expressed in radians.

EOD

A Boolean function that assumes the value true when the last item in the DATA element list has been read, otherwise false.

EOF(<expr>)

A Boolean function that assumes the value true when the last record in a file has been read, otherwise false.

EXP(<expr>)

Calculates the value of e (2.71828) to the power of the argument.

INT(<expr>)

Returns the value of the nearest integer not greater than the argument.

LEN($\left\{ \begin{matrix} \text{<svar>} \\ \text{<slit>} \end{matrix} \right\}$)

Returns the current number of characters in a string.

LOG(<expr>)

Calculates the natural logarithm of the argument.

ORD($\left\{ \begin{matrix} \text{<svar>} \\ \text{<slit>} \end{matrix} \right\}$)

Returns the ordinal number of the first character of a string.

RND

Produces a pseudo random number in the range 0 to 1.

SGN(<expr>)

Returns the algebraic sign of the argument.

SIN(<expr>)

Calculates the sine of an angle which is expressed in radians.

SQR(<expr>)

Computes the square root of the argument.

TAB(<expr>)

Used in PRINT statements to tabulate the printing position to the column number evaluated from the argument.

TAN(<expr>)

Calculates the tangent of an angle which is expressed in radians.

3. COMAL80 FILE STATEMENTS 3.

CLOSE [<expr>]

Dissociates a filename and a file number (see OPEN) so that the file can no longer be referenced. The CLOSE form of the statement closes all open files.

STATEMENT or COMMAND

COPY <filename1>,<filename2>

Copies the contents of <filename1> to <filename2>.

STATEMENT or COMMAND

CREATE <filename>,<size>[,<recl>]

Creates a file in a directory.

STATEMENT or COMMAND

DELETE <filename>

Deletes a file in a directory.

STATEMENT or COMMAND

DIR <unitno>

Lists the names of all files in the directory to the selected output device.

STATEMENT or COMMAND

DISMOUNT <unitno>

Dissociates the diskette in the specified drive from the file system by closing the directory, so that the diskette may be removed.

STATEMENT or COMMAND

INPUT FILE <expr>: $\left\{ \begin{array}{l} \langle \text{var} \rangle \\ \langle \text{svar} \rangle \end{array} \right\} \left[, \left\{ \begin{array}{l} \langle \text{var} \rangle \\ \langle \text{svar} \rangle \end{array} \right\} \right] \ldots$

Reads data in ASCII form from a sequential access file to the variables in the argument list.

STATEMENT or COMMAND

MOUNT <unitno>

Associates the file system with the diskette in the specified drive by opening the directory.

STATEMENT or COMMAND

$$\text{OPEN <expr>,<filename>, }\left\{\begin{matrix}\text{READ}\\ \text{WRITE}\\ \text{APPEND}\\ \text{RANDOM <recl>}\end{matrix}\right\}$$

Associates a filename with a user file number so that the file can be referenced in other file statements. OPEN also specifies how the file is to be used.

STATEMENT
or COMMAND

PREFIX <sexpr>

Specifies a string by which the filename in a file-statement is prefixed, e.g. unitnumber.

STATEMENT
or COMMAND

$$\text{PRINT FILE <expr>: }\left\{\begin{matrix}\text{<expr>}\\ \text{<sexpr>}\end{matrix}\right\}\left[\left\{\begin{matrix},\\ ;\end{matrix}\right\}\left\{\begin{matrix}\text{<expr>}\\ \text{<sexpr>}\end{matrix}\right\}\right]\ldots\left[\left\{\begin{matrix},\\ ;\end{matrix}\right\}\right]$$

Outputs data in ASCII form to a sequential access file.

STATEMENT
or COMMAND

$$\text{PRINT FILE <expr> USING <format>: }\left\{\begin{matrix}\text{<expr>}\\ \text{<sexpr>}\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{<expr>}\\ \text{<sexpr>}\end{matrix}\right\}\right]\ldots\ [,]$$

Outputs data in ASCII form to a sequential file, using a specified format.

STATEMENT
or COMMAND

$$\text{READ FILE <expr>[,<recno>]: }\left\{\begin{matrix}\text{<var>}\\ \text{<svar>}\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{<var>}\\ \text{<svar>}\end{matrix}\right\}\right]\ldots$$

Reads data in binary format from a sequential access file or a record of a random access file to the variables in the argument list.

STATEMENT
or COMMAND

RENAME <filename1>,<filename2>

Renames a file in a directory.

STATEMENT
or COMMAND

$$\text{WRITE FILE <expr>[,<recno>]: }\left\{\begin{matrix}\text{<expr>}\\ \\ \text{<sexpr>}\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{<expr>}\\ \\ \text{<sexpr>}\end{matrix}\right\}\right]\ldots$$

Outputs data in binary format to a sequential access file or record of a random access file.

STATEMENT
or COMMAND

4. SYSTEM COMMANDS 4.

```
AUTO  ⎡⎧ <line1>[,]              ⎫⎤
      ⎣⎩ [<line1>]  ,<line2>     ⎭⎦
```

Provides automatic line numbering for program entering.

BYE

Leaves COMAL-80.

CON

Continues execution from where the program was stopped.

```
DEL   ⎡⎧ <line1>[,]              ⎫⎤
      ⎣⎩ [<line1>],<line2'       ⎭⎦
```

Deletes one or more statements in a program.

ENTER <filename>

Merges the statement lines from the file or device specified by a filename into the current program area.

```
LIST  ⎡⎧ <line1>[,]              ⎫⎤
      ⎣⎩ [<line1>],<line2>       ⎭⎦
```

Outputs part or all of the currently loaded program in ASCII format to the selected output device.

LOAD <filename>

Loads a previously SAVE'd program in binary format from the specified file or device into the program area.

NEW

Clears the program and data areas and closes any open files.

$$\text{RENUMBER}\left[\left\{\begin{array}{l}\texttt{<line1>[,]}\\ \texttt{[<line1>],<line2>}\end{array}\right\}\right]$$

Renumbers the statements in the current program.

RUN

Clears the data area and executes the current program from the lowest numbered statement.

SAVE <filename>

Writes the statements in the program area in binary format to the specified file or device.

SIZE

Returns the size of the program and data areas and the size of free memory space.

RETURN LETTER

Title: A Survey of COMAL80 for the RC700 Microcomputer - Piccolo -   RCSL No.: 42-i1780

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: ____________________ Title: ____________________

Company: ____________________

Address: ____________________

Date: __________

Thank you

. . . . Fold here . . . .

. . . . Do not tear - Fold here and staple . . . .

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark