

1tool

CAREL

Guidelines for developing application programs in 1Tool



GB Guidelines

→ **LEGGI E CONSERVA
QUESTE ISTRUZIONI** ←
**READ AND SAVE
THESE INSTRUCTIONS**

T e c h n o l o g y & E v o l u t i o n

Content

1. General.....	5
2. Programming style on Strategy	5
3. Layout of the objects on the page	5
4. Variable names	6
5. References	6
6. Pages	7
7. Migrated project.....	7
8. Programming style in User interface	7
9. Miscellaneous.....	10

INTRODUCTION

The purpose of this document is to provide guidelines to be followed during the development of an application program in 1Tool.

1. GENERAL

- 1.1 Observe all the rules described in the application program development procedure: archiving, structure of the sources on the PC, code-version: see *codifica_applicativi.doc*
- 1.2 Add the properties of project like description, author, version, date
- 1.3 Where possible, use standard Carel macroblocks.
- 1.4 Use the same style in the entire application. When editing an existing application, copy the style used. If this is completely outside of these guidelines, evaluate whether it is worth reviewing the entire application.
- 1.5 A value must be given to the Copy_Password system variable in all applications. If no value is designated in the development specifications, assign the value 0 (zero).
- 1.6 Use the "Template_Project" to start a new Carel application.
- 1.7 All the mean variables (use into user interface, manage via supervisory, public variable for the log...) must have the description. That description will be use for the documentation (parameters list) and for the PI@ntVisor.

STRATEGY

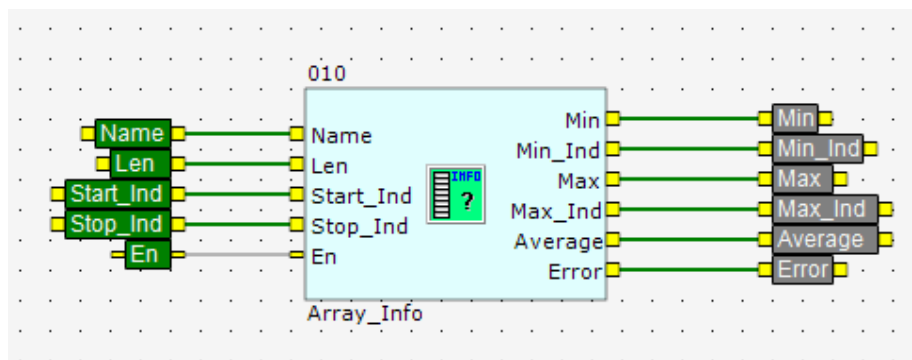
2. PROGRAMMING STYLE ON STRATEGY

When developing an application, always remember that such application may be designed, modified or debugged by any other 1Tool developer, and therefore it must be easy to understand and read.

- 2.1 Add remarks wherever possible for the algorithms, in particular the more complex ones.
- 2.2 Add remarks for the main variables.
- 2.3 No absolute timers of any kind may be used to control the sequence of operations executed. When changing to different hardware this causes incompatibility errors due to differences in the execution speed of the controllers.

3. LAYOUT OF THE OBJECTS ON THE PAGE

- 3.1 The atoms, macroblocks and variables should be aligned in columns (see Figure 3.1).
- 3.2 It is suggested to align the variable around the atoms/Mbk because the autorouting working better.



- 3.3 Similar controls should always be designed in the same way, otherwise they will appear to run different functions.
- 3.4 The paths of the wires must be "clean", avoid contorted paths (see Figure 3.1).
- 3.5 Where possible, if a variable is created and used inside the same page, the variable should not be repeated in more than one place, but rather a direct connection with a wire can be used. If this is not possible, indicate where the variable is created
- 3.6 The variables and the constants must be connected to the pins with wires of the same length for reasons of visibility (see Figure 3.1).
- 3.7 Use lines to divide the functions (see Figure 3.1)

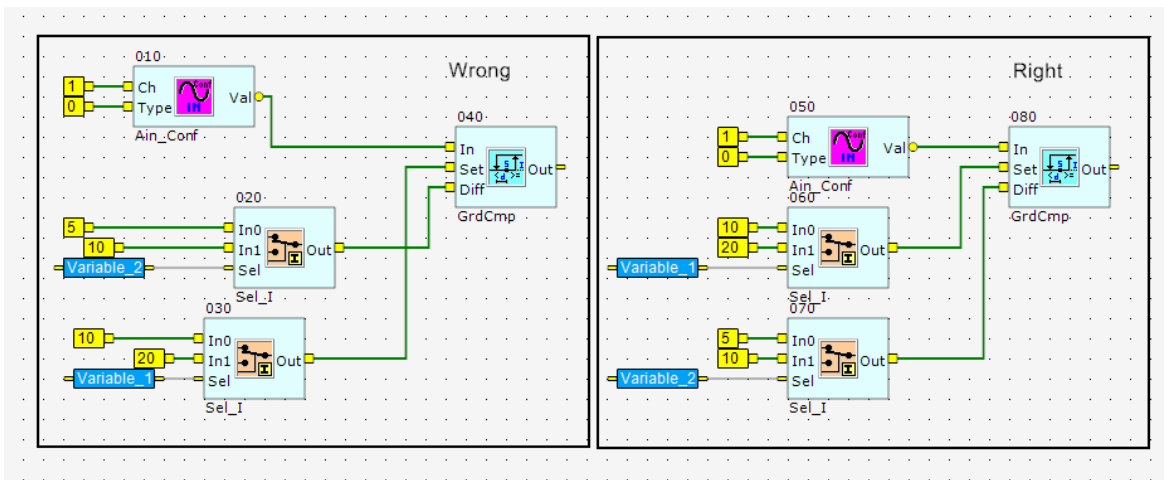


Figure 3.1

4. VARIABLE NAMES

- 4.1 The names must be meaningful (self-explanatory) and in English.
- 4.2 The names of the lists must have the following format: "Lst_" + LISTNAME.
- 4.3 For variables deriving from the pCO* inputs, avoid using the same names as the inputs, e.g. B1, B2 etc. Always assign names that are coherent with the function of the input / output.
One exception may be applications that feature different I/O configurations (multi-platform).
- 4.4 Characters used for the variable names: see paragraph 1.5.
- It is recommended to use the same notation as the atom pins, that is, the first letter of each word in upper case and the rest lower case, with an underscore between one word and the next: Xxxxx_Yyyyyy
- 4.5 When names are abbreviated, maintain the plural, for example En_Compressors should become En_Cmps.

5. REFERENCES

Pay careful attention to the order of the references used in a routine. Failure to observe the correct order means creating delays of a number of program cycles in the results

- 5.1 The references must not be consecutive, use step at least of 10
This makes it simple to enter any atoms/macroblock in the middle of a function.
- 5.2 Add remarks to any inverted references, that is, when the sequence is not the natural left-to-right
- 5.3 Jump for each page, avoiding the use of cumulative Jumps. The exception is the "Initial Jump".
- 5.4 When logging the alarms from the application, use DO_EVENT A and B in the routine.
- 5.5 Do not put atoms belonging to different DO_EVENT statements on the same page. The exception is the alarm log from application.
- 5.6 Pay attention to use atoms to manage time inside the DO_EVENT: i.e the output variable of a R_TRIG atom used inside the DO_EVENT does not give an impulse but it became 1 only.

6. PAGES

- 6.1 The names must be meaningful (self-explanatory) and in English.
- 6.2 Avoid pages with too many atoms (to make them clearer and for future modifications). Also see paragraph 3.8. One page should be created for each routine, or alternatively, if the function is complex. it can be divided into consecutive pages.
- 6.3 Equal routines must be on the same page or on consecutive pages (see paragraph 6.4), i.e:
 - Special routines, for example: management of the application version and automatic installation of default values, automatic return to the menu, key_switch management, check the hardware used, En_Builtin_Keyboard management, etc....
 - Analogue / digital inputs
 - Analogue / digital outputs
 - Device management
 - Supervisor
 - Log
- 6.4 In order to have a right flow of the different managements, the pages must be arranged in logical order. If this is not respect the application are slow to process the data and the devices are not correctly manage and protect. i.e:
 - inputs
 - alarms
 - devices
 - other functions (supervisor, log, etc....)
 - outputs
- 6.5 Use titles that identify the groups of pages by functions, alarms, master, slave, supervisor etc..

7. MIGRATED PROJECT

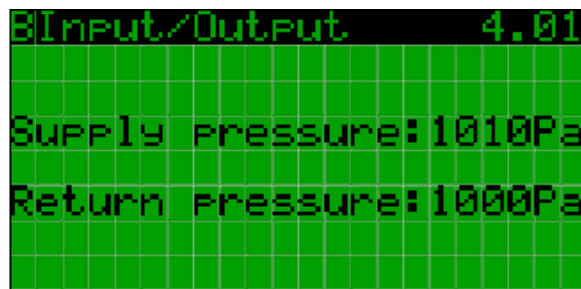
- 7.1 Always add the new 1T atoms and macroblock, if it is possible replace the other.
- 7.2 Delete, if it is present, the first page (usually sheet 0) that contains link to other page. Pay attention, from the sheet 0, it is possible to recover the sheet description.

USER INTERFACE

8. PROGRAMMING STYLE IN USER INTERFACE

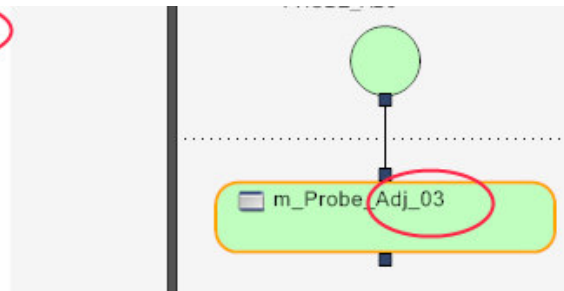
If no specific requirement from the customer, to use the following rules.

- 8.1 The interface must be respected the loop, the layout described in the *User_Interface.xls*
- 8.2 If the application is standard Carel, add an index for each screen (or mask). The index is positioned in the top right corner of the screen.

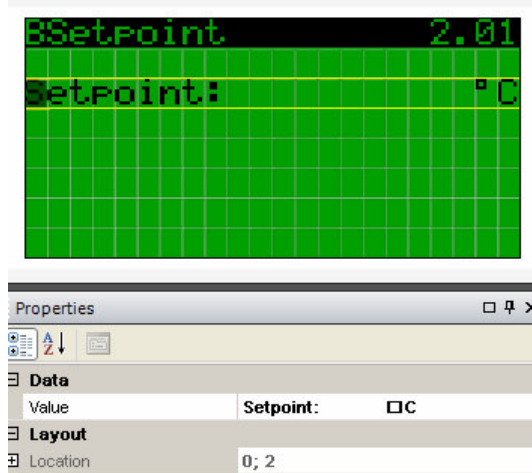


- 8.3 The index and the parameters this identifies must never be changed, once described in the manual.
- 8.4 The index must respect be shown in the name of the screen. The suffix must also be shown in the name of the screen.

```
3Probe adjust. 7 fb03  
Supply pressure:-010Pa  
Return pressure:-010Pa
```



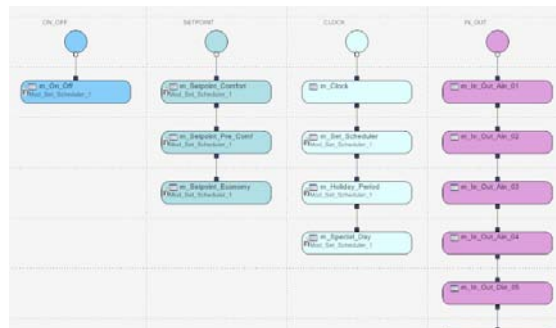
- 8.5 The name of the screens must be in English and must have the following format: "m_" + screen_name + screen_index, if present.
- 8.6 The name of the screens that the "refer_to" statements refer to must have the following format: "m_ref_" + screen_name.
- 8.7 There must be only 1 level of "refer_to"
- 8.8 Each loop of screens must have an identifying colour (see User_Interface.xls).
- 8.9 The first letter, and only the first, of the parameter description must be in upper case.
- 8.10 The wording must be aligned to the left.
- 8.11 The fields must be aligned to the right.
- 8.12 Reach the length of the line by adding spaces between the description and the unit of measure. For example:



Note that "Temperature °C" is just one string. (Not applicable for PGD23)

This setting is useful when managing multi-language applications.

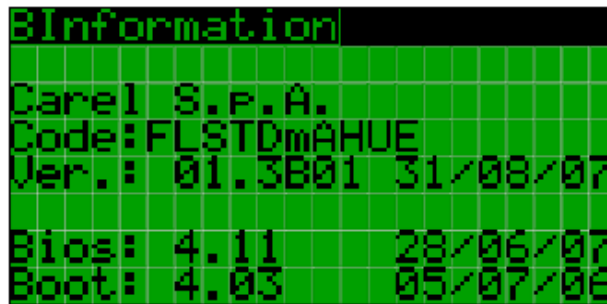
- 8.13 There must not be spaces between the unit of measure and the fields.
- 8.14 The texts used in the ASSBOOL and ASSINT statements for output fields must be: the first letter in upper-case and the other letter in lowercase.
- 8.15 The texts used in the ASSBOOL and ASSINT statements for the input - output fields must be in upper case.
- 8.16 If a text needs to be enabled only in certain conditions from a field, use USE_TXT with enable in the field and not Assbool.
- 8.17 The unit of measure must be written as follows: °C, °F, bar, PSI, kPa, s, min, h, etc...
- 8.18 For all integer and analogue input fields, the limits for the field must be entered.
- 8.19 There must not be any strange punctuation marks at the end of the parameter description, such as arrows "Temp.amb-->21.0°C" or dots "Temp.amb.....21.0°C" and so on
- 8.20 The abbreviations must be easy to read. Avoid abbreviations such as: temperature -> tmp - pressure -> prs
- 8.21 When abbreviating words, do not leave spaces between the full stop and the start of the next word. Example: "Amb.temp."
- Words must be abbreviated with care, truncating the word is not always effective and at times makes it hard to understand or read
- 8.22 The first line of an alarm screen must show the alarm code. Example: AL000
- 8.23 In the demo applications, where possible do not use the specific features of the terminal: "°C" compacted into one character, images etc....
Clearly this does not apply to demos for pGD1, pGD3.
- 8.24 Use multiple columns on the User interface worksheet, so as to make it easier to refer to the various loops.
Raccomended one loop for each column



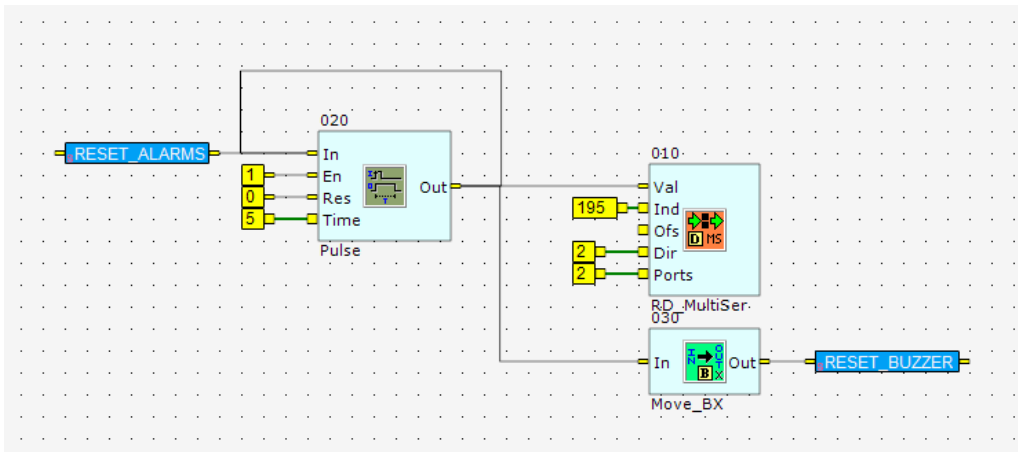
MISCELLANEOUS

9. MISCELLANEOUS

- 9.1 Each analogue/digital input/output must have a value/status in the inputs and outputs menu.
- 9.2 Each analogue input must be associated with the probe fault alarm.
- 9.3 Each analogue/digital input/output must be made available, with read-only access, to the supervisory system.
- 9.4 At least the following supervisor protocols must be implemented: Carel, Modbus, RS232, GSM, LonWorks.
- 9.5 The automatic return to the main menu must be included. Allow a delay time of 5 minutes unless otherwise indicated
- 9.6 On the PGD0 and PGD1 terminals the backlighting is common to the display + Esc, up, down, enter buttons. Unless otherwise indicated, activate it when pressing any button and deactivate it automatically 5 minutes after the last button was pressed (code LED 25).
- 9.7 Screens must be included to manage: application code, version and date, Boot and Bios version and date. Use the standard screens proposed by Carel:



- 9.8 Reset alarms from supervisor: use a 5 second Pulse



- 9.9 The names of the special fields in the demos must have the following format: "dsf_" + special_field_name
- 9.10 The mask name in the demos must have the following format: "d_" + nome_identificativo_della_funzione_del_demo.
- 9.11 The version of the application must be managed by STRATEGY and be visible from the supervisor.
- 9.12 For the PGD0 and PGD1 terminals, use a sliding menu to access the various submenus, compliant with the Carel style (see *Template_Solution and User_Interface.xls*).

CAREL reserves the right to modify the features of its products without prior notice.

CAREL

CAREL S.p.A.

Via dell'Industria, 11 - 35020 Brugine - Padova
(Italy)

Tel. (+39) 049.9716611 Fax (+39) 049.9716600

