

CAREL

базовый курс 1Tool

базовый курс по обучению *Itool*

2/02/2009 - 04/02/2009

Добро пожаловать

Семинар проводит:

А.Головин, United Elements



CAREL

1tool

Программные средства для всех контроллеров и терминалов CAREL *pCO sistema*



Цель семинара

➤ Получить базовые знания по ведению проекта в среде 1tool путем выполнения следующих упражнений:

- Создание, редактирование, компиляция и выгрузка проекта
Обзор компонентов 1tool
- Специальные функции
Обзор специальных функций: виртуальных клавиш, аварийных сигналов, назначений клавиш и проч.
- Обмен переменными с системой BMS
Сеть системы мониторинга
- Обмен переменными с другими контроллерами
сеть pLAN
- Конвертация проекта, созданного в EasyTools
- Тестирование и конфигурирование приложений онлайн при помощи Commissioning tool
- Поддержка
электронная почта, веб-сайт

Раздел 1/5

- Создание нового решения или проекта (*SolutionExplorer*)
Обзор редакторов
- Определение алгоритма управления (*StrategyEditor*)
- Определение пользовательского интерфейса (*MaskEditor*)
- Компиляция
- Моделирование (*Simulator*)
- Загрузка приложения в контроллер (*pCOLOAD*)

Раздел 2/5

- Создание параметра (например, уставки) ПЗУ (Т), значения по умолчанию, файл DEV
- Связь с текстом (*ResourceUsedOnBool/Int*)
Свойства объектов
- Создание новой маски
- Определение функции клавиши (*KeyFunctionEditor*)
- Цикл масок
- Разрешение маски (свойство *EnableOn*)
- Уровни: *Field* (поле), *Mask* (маска), *Global* (глобальный).
Поле *move*: позволяет отобразить предыдущую или следующую маску

Раздел 3/5

- Некоторые макроблоки: *GE, GT, LE, LT, GradCmp, Limit/_Low/_High*
- Управление статусом устройства (условный оператор IF)
- *Системные переменные (KEY_COUNTER)*
- *Виртуальные клавиши*
- Создание новой *StrategyPage*
- Стандартное управление аварийными сигналами
Макроблоки *Hyst_Dir_Set2Diff, Hyst_Rev_Set2Diff, Hyst_Dir_OnOff, Hyst_Rev_OnOff*.
- Маска аварийного сообщения: аварийная ситуация/
память аварийных сообщений
- Системные переменные: BUZZER, GLOBAL_ALARM,
RESET_BUZZER, RESET_ALARM

Раздел 4/5

- Создание макроблока
- Создание резервной копии *Solution*: скопировать папку *Soluzione*
- Определение маски (без поля *move*)
- Отображение специальных символов (например, °)
- Конвертация проекта, созданного в EasyTools (*Migration Wizard*)
- Содержимое флеш-памяти:
BOOT, BIOS, IUP, BLB/BIN, DEV
- Сервисная служба *Itool*., форма «задать вопрос специалисту» на сайте

Раздел 5/5

- Управление параметрами времени: системные переменные *CURRENT_HOUR*, *NEW_HOUR*, *SET_HOUR*
Код поля, макроблок *Schedule_Daily*
- BlockSequenceNumber:
основная программа, события (атом/ оператор *Do_Task*)
- Сеть pLAN (до 32 устройств)
 - Маска каждого контроллера может отображаться макс. на 3 терминалах (*SHared*, *PRivate*)
 - Обмен значениями переменных между контроллерами (до 2048 связей)
- Конфигурирование сети pLAN (адреса, конфигурации терминалов)
- BMS: отправка и получение переменных

Установка

- Запустить ОС Windows (версия XPpro или Vista)
 - ➔ Войти под учетной записью “Administrator”
- Вставить DVD 1tool
Открыть файл *D:\INDEX.HTM*
- Выбрать *English, Setup 1tool, First installation*
- Запустить файл *1tool_prerequisites_setup.exe*
Выбрать "Запуск" (Run). Принять все варианты.
- Запустить файл *1tool_setup.exe*
Выбрать "Запуск" (Run). Принять все варианты.
- Запустить 1tool: *Start/Programs/Carel/1tool/1tool*



Компоненты и редакторы *ltool*

- **Solution Explorer**
Создание проектов, запуск редакторов и добавление языков
- **Strategy Editor**
Алгоритм управления
- **Mask Editor**
Пользовательский интерфейс (содержимое *масок*, функции клавиш)
- **Network Editor**
проекты в сетях pLAN (выбор устройств и проектов, определение связей между устройствами)
- **Simulator**
Моделирование на ПК
- **pCO Manager/ pCO Load**
Выгрузка файлов приложения на контроллер, определение журналов и проч.
- **Migration Wizard**
Конвертация проектов, созданных в ПО EasyTools.



Фазы проекта



Переменные

Типы данных (основные)

Булевы (0/1)
Целочисленные (-32768/+32767)
Аналоговые (-3276,8/+3276,7)

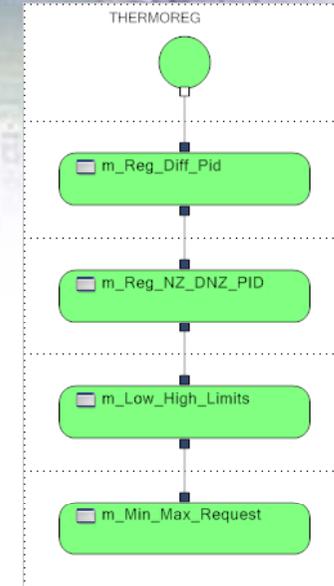
Типы памяти (основные)

Тип памяти	Идентификатор
ЭСППЗУ – permanent (флеш-память)	T
вспомогательное ЗУ – auxiliary (RAM) (в случае аварийного сбоя электропитания данные теряются)	X

Примечание: все атомы записываются в память типа X
(за исключением RA_IN_T / RI_IN_T / RD_IN_T, Move_BT_En, Move_IT_En)

Цикл масок

- Цикл масок представляет собой группу масок.
- Если курсор находится в верхнем левом углу (в поле *move*), то переход к следующей или предыдущей разрешенной маске цикла осуществляется клавишами "вверх-вниз".
- Поле *move* позволяет изменять маски клавишами "вверх-вниз".
- Поле *move* автоматически добавляется в новые маски. Если оно не требуется, его можно удалить. Для восстановления скопируйте его из другой маски.
- Объект *Loop* опционален.
- Оператор *Goto_loop()*;  (соответствует *Goto_Mask* (первая маска цикла))



```
B
High limit:  +AA.A°C
Low limit:   +AA.A°C
Limit diff.: AA.A°C
En.double action: B
```

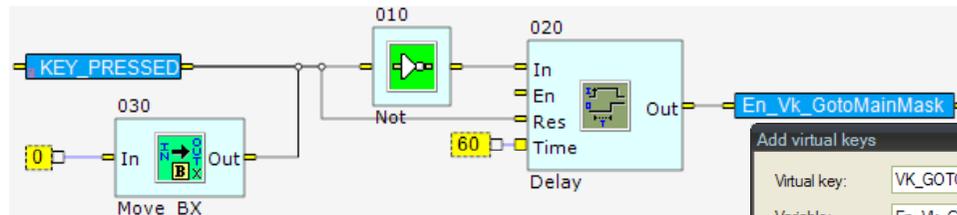
```
4 KEY(UP)
5 {
6   Goto_previous_mask();
7 }
8
9 KEY(DOWN)
10 {
11   Goto_next_mask();
12 }
```

Виртуальные клавиши

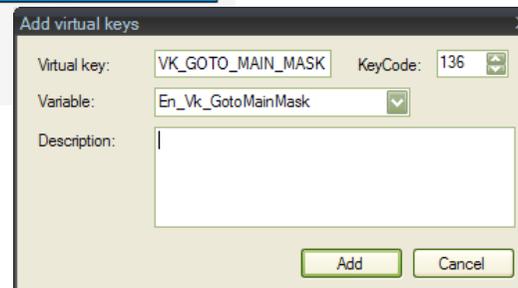
- Обычно при нажатии физической клавиши выполняется оператор *KEY* .
- Иногда требуется создать клавишу в соответствии со значениями переменных. Это клавиша будет *виртуальной*. При изменении переменной с 0 на 1 происходит "нажатие" *виртуальной клавиши*.

Создание виртуальной клавиши:

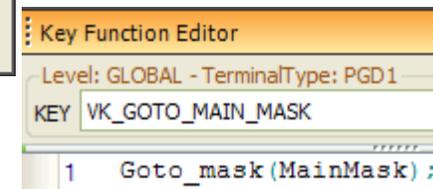
1. Создать переменные, которым присваивается значение 1 при необходимости "нажатия" виртуальной клавиши.



2. Создать виртуальную клавишу и привязать ее к переменной (код клавиши: 137-179, 200-254).



3. Определение функции новой виртуальной клавиши.



Упражнение: отобразить главную маску после неактивности клавиатуры в течение 60 с.



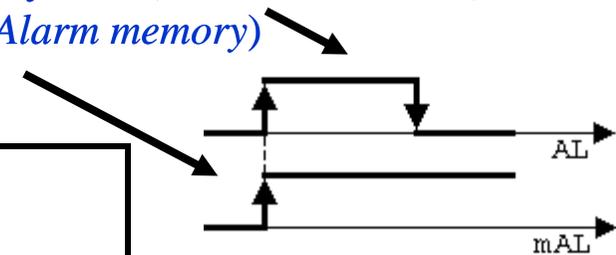
Управление аварийными сигналами (1/2)

- Среда 1tool предлагает набор системных переменных для включения и выключения звукового сигнала (MAN_BUZZER_EN, MAN_BUZZER_ON, MAN_BUZZER_OFF).

Встроенная система стандартного управления аварийными сигналами предусматривает следующее:

- Включение звукового сигнала при обнаружении аварийной ситуации (*Alarm status* 0→1)
- Для обработки переменной в памяти аварийных сообщений (*Alarm memory*)
- Набор системных переменных:

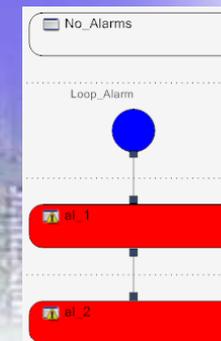
BUZZER 0: звуковой сигнал ВЫКЛ 1: звуковой сигнал ВКЛ	GLOBAL_ALARM 0: журнал аварийных ситуаций пуст 1: как минимум 1 запись об аварийной ситуации
RESET_BUZZER 1: запрос на отключение звукового сигнала (автоперезапуск)	RESET_ALARMS 1: запрос на перезапуск всех записей об аварийных ситуациях (автоперезапуск)



Данная функция осуществляется следующим образом:

- Маски аварийной сигнализации
- Свойства *AlarmStatus*, *AlarmMemory*
- AlarmMemory* (mAL) разрешает маску ALARM
- переменная *EnableOn* разрешает маску DATA

Управление аварийными сигналами (2/2)



При нажатии клавиши ALARM:

Если...	то...	язык ltool
Сигналы об аварийных ситуациях отсутствуют	Переход к маске "No_Alarms"	<pre>if (SysVarList.GLOBAL_ALARM==0) { Goto_loop(No_Alarms); }</pre>
Как минимум 1 запись об аварийной ситуации и задействуется звуковой сигнал	Отключить звуковой сигнал (запрос) Переход к циклу "Alarm"	<pre>if ((SysVarList.GLOBAL_ALARM==1) && (SysVarList.BUZZER==1)) { SysVarList.RESET_BUZZER=1; Goto_loop(Loop_Alarms); }</pre>
Как минимум 1 запись об аварийной ситуации и звуковой сигнал не задействуется	Сброс всех записей об аварийных ситуациях (запрос) Переход к циклу "Main"	<pre>if ((SysVarList.GLOBAL_ALARM==1) && (SysVarList.BUZZER==0)) { SysVarList.RESET_ALARMS=1; Goto_loop(Loop_Main); }</pre>

Макроблоки

Как открыть макроблок (модуль):

- Контекстное меню *Open block*.

Справка:

- Выделить блок, перейти к контекстному меню “?”

Как открыть макроблок:

- Создать *New project, Module/Macroblock*.
- Удалить *Terminal node* (в макроблоках маски отсутствуют).
- Определить *Strategy*.
- Создать точки подключения (выбор свойства *Block pin*).
- Определить контур при помощи *Shape editor*.
- Сохранить блок в библиотеке через контекстное меню “*Save to library As*”.

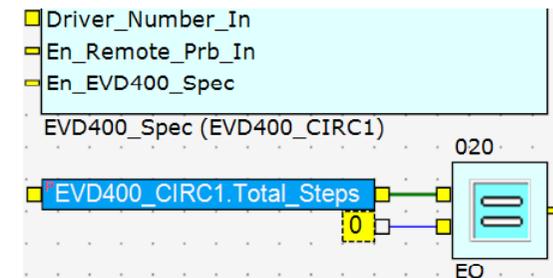
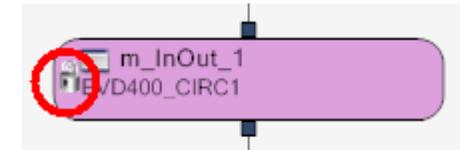
Модули (1/2)

Типы блоков:

- Блок "Атом"
- Блок "Макроблок" (только алгоритм)
- Блок "Модуль" (макроблок + маски)

Вставка масок:

- *Связанные (Linked):*
 - Маски поддерживают связь с экземпляром блока.
 - Если блок удалить, его маски также будут удалены
 - Также возможно добавлять TEXT/IMAGES/ANIM/LINE/RECT и перемещать маски.
- *Импортированные (Imported):*
 - Маски импортируются и становятся масками проекта.
 - Допускаются любые модификации.
- *Только алгоритм (Strategy only):*
 - Вставка масок не осуществляется.
- У модулей имеются метки (LABEL) - например, EVD400_CIRC1.
- Переменные, особые поля, сенсорные схемы будут переименованы (пример: *Fan_1 Set*).
- У данного приложения имеется доступ к переменным модуля: *LABEL.VarName*.
Только тех входов-выходов, которые не отображаются в контуре (*Visible in shape*).



Модули (2/2)

Модуль можно разработать для разных типов дисплея и разных языков.

Модуль	Проект	Действие в зависимости от терминала
PGD0	PGD0	Вставляется
PGD1		Игнорируется
	PGD3	Маски не вставляются

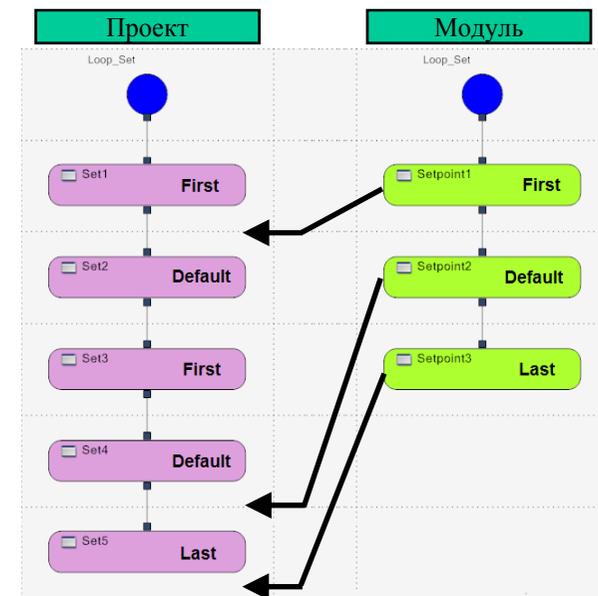
Модуль	Проект	Действие в зависимости от языка
EN	EN	Вставляется
IT	IT	Вставляется
FR		Игнорируется
	DE	Используется первый язык (EN)

- Маски вставляются в исходный цикл согласно свойству *MaskDefaultPosition*:
Default, First mask, Last mask (по умолчанию, первая маска, последняя маска)

- Автоматическое соединение

Примечание

- Модуль не может включать в себя другие модули (с масками) - исключительно *атомы* и *макроблоки*. Маски будут игнорироваться.
- Модуль не может включать в себя *GlobalCode/Events*



Системы управления зданием (BMS)

- Внимание!
Сети BMS и pLAN являются независимыми.
- Цель сети BMS - обмен переменными с *системой мониторинга/ BMS* (PlantVisor, pCOWeb, WebGate, шлюзом, модемом, Lon, Trend, DLL и т.д.)
- Отправка и получение переменных (для любой архитектуры)
 1. Адрес в сети (системная переменная BMS_ADDRESS (W,T))
 2. Скорость передачи данных (системная переменная COM_BAUDRATE_BMS (W,X))
 3. Протокол передачи данных (системная переменная COM_PROTOCOL_BMS (W,X))
 4. Выбор переменных (2 варианта):
 1. Свойства переменных, свойства BMS
 2. Удаленные атомы
 - RD_In_T, RI_In_T, RA_In_T (переменные в памяти типа "Т")
 - RD_In_X, RI_In_X, RA_In_X (переменные в памяти типа "X")
 - RD_Out, RI_Out, RA_Out (переменные в памяти типа "Т" и "X")
 - RD_Multiser, RI_Multiser, RA_Multiser

До 207 переменных для каждого типа.



Архитектура контроллеров

■ Автономная (локальная)

- Только 1 контроллер (со встраиваемым терминалом или без него)
Адрес обязательно должен быть 0.
- Только 1 опциональный внешний терминал (всегда индивидуальный – *Private*)
Адрес обязательно должен быть 0.



■ Сетевая (pLAN)

- До 31 контроллера (со встраиваемым терминалом или без него)
Адрес не должен быть 0.
- До 31 внешнего терминала (*Индивидуальный/ Private* или *Общий/ Shared*)
Адрес не должен быть 0.



Устройства для сети pLAN



pCO3/2



pCO1



pCOXS



pCOC



pCOB

LCD
4x20



PGD
0/1/2/3



Графический
дисплей



ARIA/PAD



FCM



EVD – электронный ТРВ



Сеть pLAN

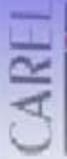
Внимание!

Сети BMS и pLAN являются независимыми.

- До 32 устройств (устройство с номером 32 может быть только терминалом)
- Цель создания сети pLAN:
 - Маска каждого контроллера может отображаться макс. на 3 дисплеях (как на общих, так и на индивидуальных)
 - Обмен переменными между контроллерами (до 2048 исходящих связей для каждого контроллера)
- Предварительная настройка приложения:
 - У переменных, которые используются в связи, должен стоять флажок pLAN.
- NetworkEditor, двоичные файлы
- Настройка pLAN
 - Компиляция
 - Выгрузка
 - Адресация устройств
 - Электроподключение
 - Конфигурирование дисплея



Конфигурирование терминала pLAN



- Удерживать клавиши ВВЕРХ + ВНИЗ + ENTER нажатыми в течение 5 секунд

- Повторить для всех контроллеров

- Удерживать клавиши ВВЕРХ + ВНИЗ + ENTER нажатыми в течение 10 секунд

Display address setting.....:03	←	Адрес терминала (только для PGD)
I/O Board address:01	←	ADDRESS конфиг. контроллера
Terminal Config Press ENTER to continue		Нажать ENTER (не обращая внимания на звуковой сигнал)
P01: Adr Priv/Shared	←	Адрес контроллера
Trm1 03 Sh	←	Терминал 1/3 ADDRESS PR/SH
Trm2 None --	←	Терминал 2/3 ADDRESS PR/SH
Trm3 None -- OK?No	←	Терминал 3/3 ADDRESS PR/SH
NetSTAT 10001111118		Подключенные устройства: ☒ : контроллер (pCOx, ARIA, привод TPВ...) ☒ : терминал (текстовый, PGD, СИД) ┐: отсутствуют
T:03 9111111116		
Enter 1711111124		
To quit 2511111132		

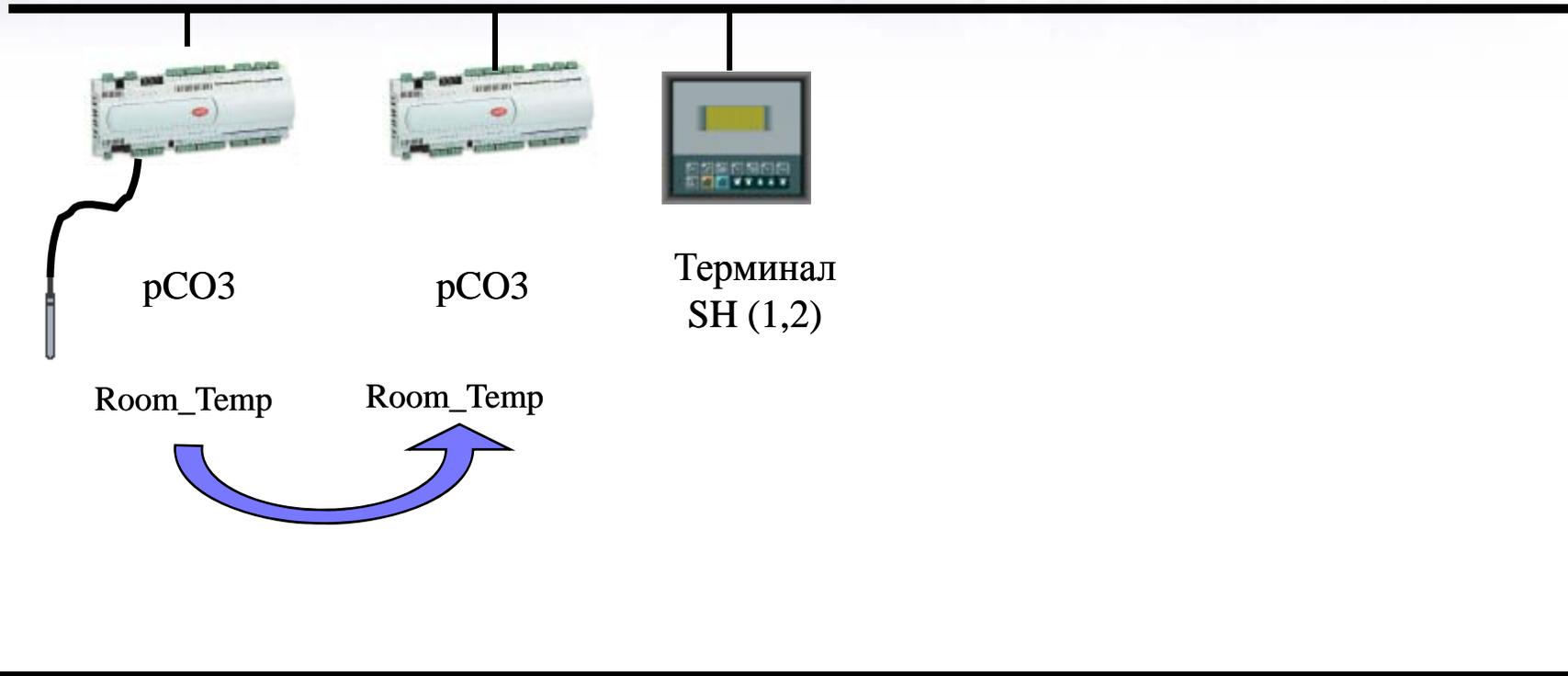


Упражнение по rLAN

Устройство 1

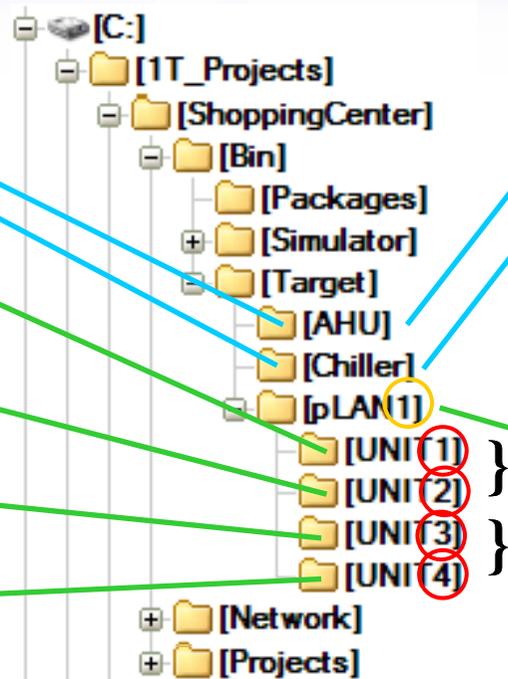
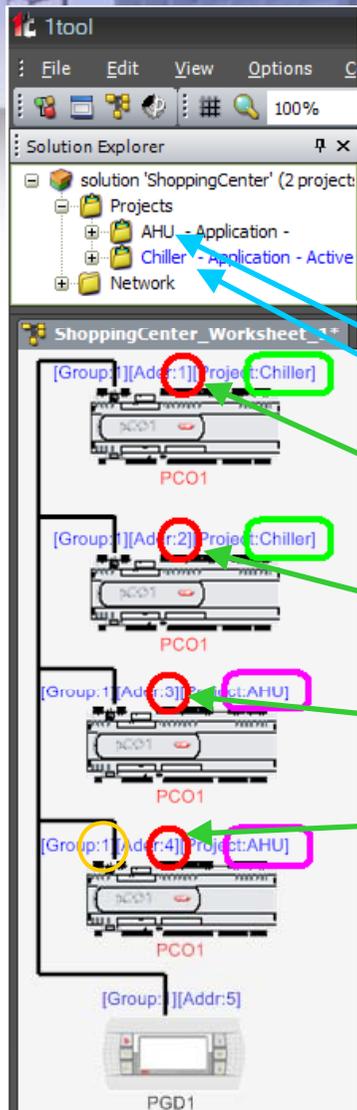
Устройство 2

Устройство 3



Source Unit	Source Variable	Target Unit	Target Variable
1	Room_Temp	2	Room_Temp

Компиляция и двоичные файлы



Автономный

..\Имя_решения\Bin\Target\Имя_приложения

pLAN

..\Имя_решения\Bin\Target\PLANn\Адрес

В папках UNIT1-2 содержатся файлы для чиллеров (папка CHILLER)

В папках UNIT3-4 содержатся файлы для центральных кондиционеров (папка AHU)

Содержимое флеш-памяти

Загрузка (boot) + резерв 32 КБ	Загрузчик для прочих КОМПОНЕНТОВ Обычно обновление не требуется.
BIOS 512 КБ (текущая BIOS)	Операционная система Требуется обновление (новые функции и отладка). Обновления см. на сайте KSA
Память типа “Т” + резерв 32 КБ	ПЗУ
Файлы приложения Для флеш-памяти 1 МБ: 448 КБ Для флеш-памяти 2 МБ: 1472 КБ	IUP+BLB/BIN

Сервисная служба 1tool

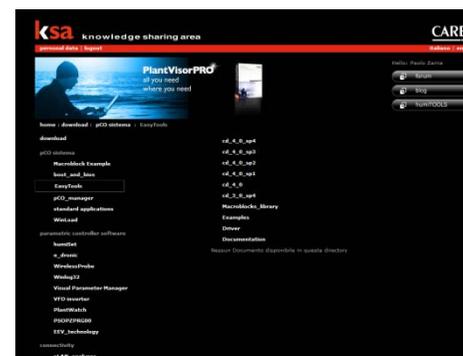
Электронная почта: . Для русскоязычных пользователей рекомендуется использовать форму «Задать вопрос специалисту» на



➤ Сайт KSA

- Новые выпуски *1tool*
- Новые версии BIOS/BOOT
- Новости
- Форум

Запросите логин и пароль



Семинар по 1tool

Спасибо

