



ОСНОВЫ ПРОГРАММИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ



Лекция №1

B&R (Bernecker + Rainer)

B&R Industrial Automation GmbH (B&R) — европейский производитель компонентов промышленной автоматизации.

4 апреля 2017 года Bernecker + Rainer был приобретён корпорацией ABB



Продукция

- Программируемые логические контроллеры (ПЛК)
- Системы распределённого ввода-вывода
- Промышленные компьютеры
- Панели оператора
- Сервоприводы
- Шаговые приводы
- Преобразователи частоты
- Синхронные двигатели
- Шаговые двигатели
- Системы Безопасности Safety
- Распределённая система управления APROL
- Мотор-редукторы

Automation Studio

Automation Studio представляет собой среду программирования для оборудования B&R.

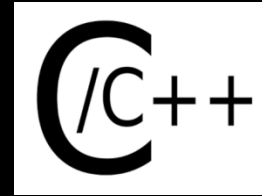
Основные особенности:

- Простота обучения
- Кросс-платформенная совместимость
- Независимость от поставщика
- Более быстрый старт серийного производства вследствие высокого уровня эффективности разработки
- Максимальная возможность повторного использования программ для других вариантов машин и серий
- Обслуживание удаленных (находящихся на объекте) систем с помощью Web

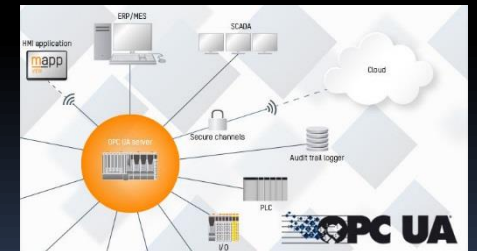
Особенности Automation studio.

Соответствие международным стандартам:

- МЭК 61131-3
- ANSI C и C++
- PLCopen
- OPC и OPC UA
- MATLAB® и Simulink®



PLCopen



Особенности Automation studio.

Интуитивное управление

- ▣ Графическое отображение аппаратной конфигурации с System Designer
- ▣ Простой ввод в эксплуатацию приводов
- ▣ Быстрое обучение

Независимость от аппаратного обеспечения

- ▣ Разработка ПО без необходимости привязки к конкретному оборудованию
- ▣ Поддержка нескольких конфигураций машин в одном проекте
- ▣ Запуск имитации одним кликом мышки

Управление

- ▣ Полная интеграция управления, визуализации, контроля движения и функций обеспечения безопасности
- ▣ Программная совместимость со всеми аппаратными платформами
- ▣ Динамическая замена компонентов
- ▣ Децентрализованное аппаратное обеспечение, централизованное управление данными

Безопасность

- ▣ Идентичный вид данных входов/выходов безопасности от стандартных приложений и приложений безопасности
- ▣ На приложение безопасности не влияют изменения, вносимые в стандартное приложение.
- ▣ Безопасность во время программирования обеспечивается использованием сертифицированных функциональных блоков PLCopen
- ▣ Не зависящие от полевой шины технологии безопасности с openSAFETY

Automation Runtime

Automation Runtime – модульная операционная система реального времени.

Automation Runtime - ядро программного обеспечения, которое позволяет запускать приложения на целевой системе.


- Предоставляет возможность быстрого повторного выполнения приложения в пределах выделенного интервала времени.
- Предоставляет пользователю детерминированный, аппаратно-независимый, многозадачный инструмент для создания приложений.



Задачи ОС Automation Runtime

ОС решает следующие основные задачи:

- ▣ управление аппаратными ресурсами
- ▣ управление программными ресурсами
- ▣ предоставление унифицированного интерфейса доступа к аппаратному обеспечению.



Automation Runtime полностью интегрирована в соответствующую целевую систему, что позволяет прикладным программам получать доступ к модулям ввода/вывода системы, интерфейсам, полевым шинам, сетям и устройствам хранения данных.



Функции Automation Runtime

- работает на всех целевых системах B&R;
- обеспечивает независимость приложения от аппаратного обеспечения;
- обеспечивает детерминированное поведение благодаря циклическому алгоритму системы исполнения;
- позволяет настроить различное время цикла;
- поддерживает 8 различных классов задач;
- гарантирует реакцию на превышение времени цикла;
- обеспечивает настройку предельно допустимых отклонений для всех классов задач;
- поддерживает библиотеки функций в соответствии с МЭК 61131-3;
- обеспечивает доступ к любым сетям и шинам;
- включает в себя встроенные FTP-, Web- и VNC- сервера;
- обеспечивает комплексную диагностику системы (SDM).

Начальная загрузка Automation Runtime

При включении выполняются следующие задачи:

- Проверка оборудования
- Обновление оборудования / микропрограммного обеспечения (если это необходимо)
- Проверка BR модулей
- Копирование BR модулей из ПЗУ в динамическое ОЗУ (DRAM)
- Копирование переменных из статического ОЗУ (SRAM) в динамическое ОЗУ (DRAM)
- Инициализация области памяти для переменных
- Выполнение программы инициализации
- Активация циклических программ

Этапы начальной загрузки.

BOOT (НАЧАЛЬНАЯ ЗАГРУЗКА)

- Если не вставлена карта памяти CompactFlash
- Если на карте CF нет операционной системы
- Переключатель узла выставлен на "00"

DIAG (ДИАГНОСТИКА)

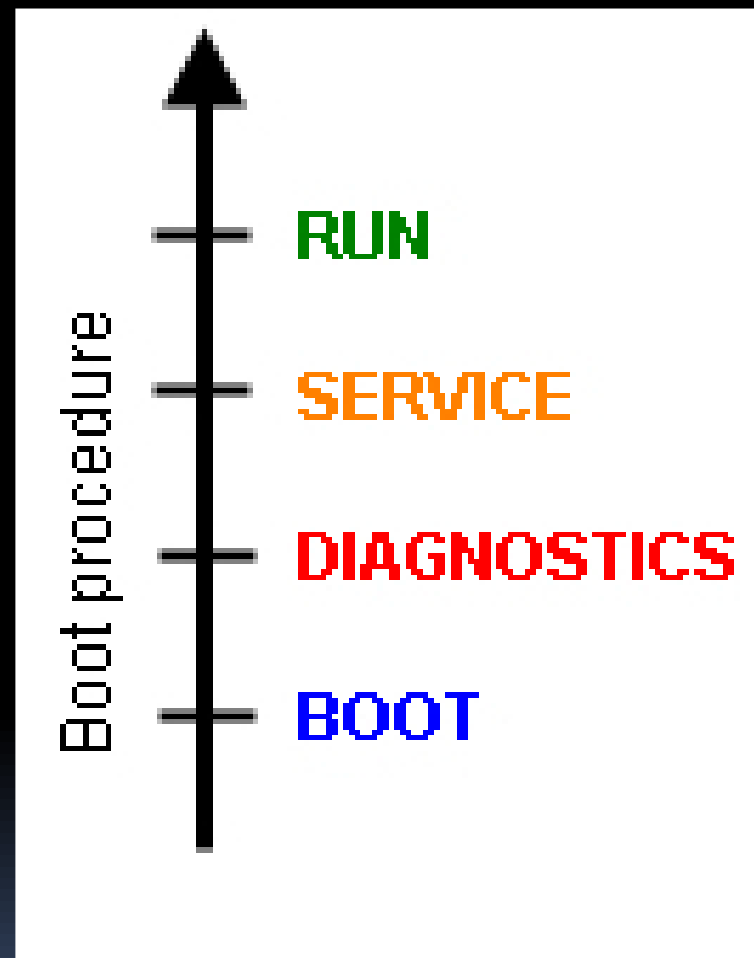
- Очистка памяти
- Неустраняемая системная ошибка
- Переключатель узла выставлен на "FF"

SERV (СЕРВИС)

- Деление на ноль
- Ошибка доступа к памяти (Page fault)
- Превышение времени цикла
- ЦПУ остановлен по команде из Automation Studio
- Прочие ошибки

RUN (РАБОТА)

- Ошибок нет



Tcpip/DAIP=10.0.0.2


CP1485 V3.00

RUN



Операционные системы.


Под ОС понимают комплекс управляющих и обрабатывающих программ, который, с одной стороны, выступает как интерфейс между аппаратурой компьютера и пользователем с его задачами, а с другой - предназначен для наиболее эффективного использования ресурсов вычислительной системы и организации надежных вычислений.





ОС реального времени.

Операционная система реального времени (ОСРВ, англ. real-time operating system, RTOS) — тип операционной системы, основное назначение которой — предоставление необходимого и достаточного набора функций для работы систем реального времени на конкретном аппаратном оборудовании.




Реальное время в операционных системах — это способность операционной системы обеспечить требуемый уровень сервиса в определённый промежуток времени.



Система реального времени.

Система реального времени (СРВ) — это система, если правильность ее функционирования зависит не только от логической корректности вычислений, но и от времени, за которое эти вычисления производятся.

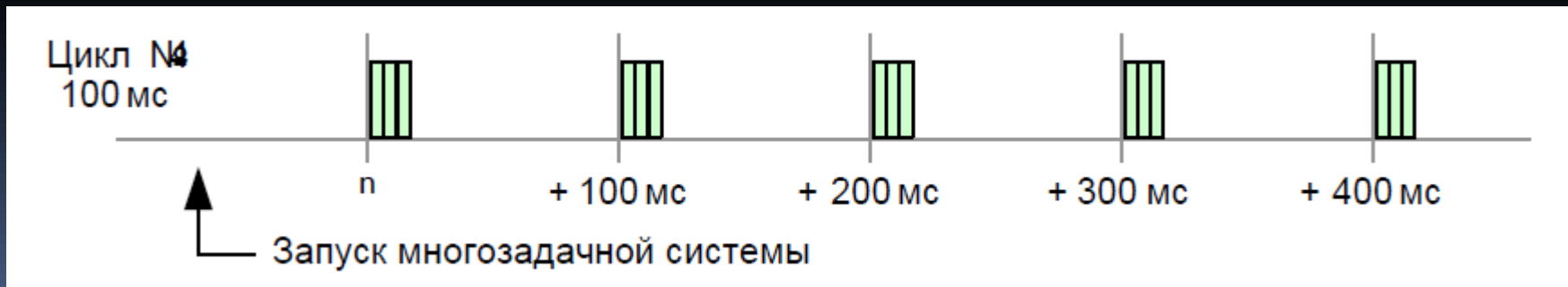


Говорят, что система работает в реальном времени, если ее быстродействие обработки данных и генерирования управляющих сигналов адекватно скорости протекания физических процессов на объектах контроля или управления.

Automation runtime.

Automation Runtime является **детерминированной многозадачной операционной системой реального времени.**

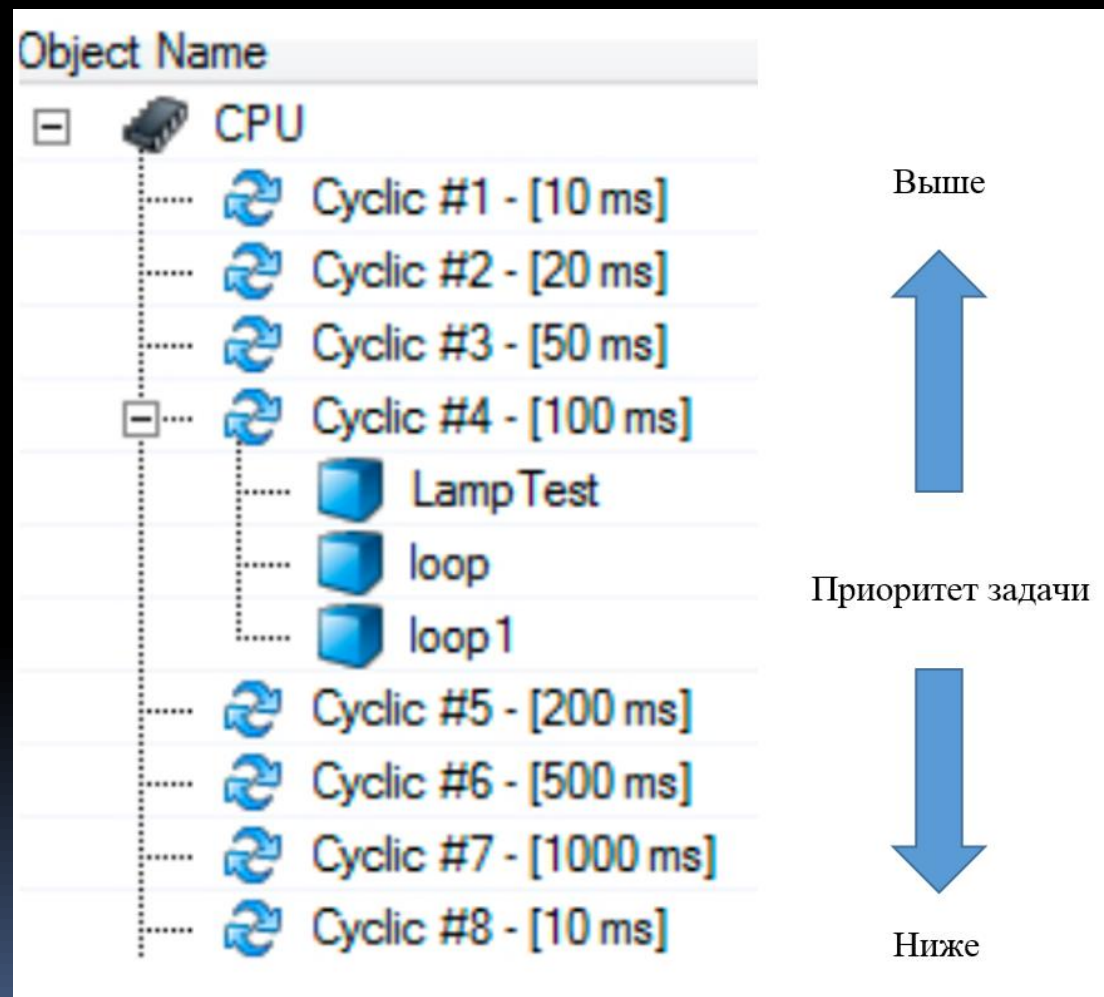
Задачам выделяются определенные ресурсы, а выполнение их осуществляется последовательно в определенные временные интервалы (**time slices**).



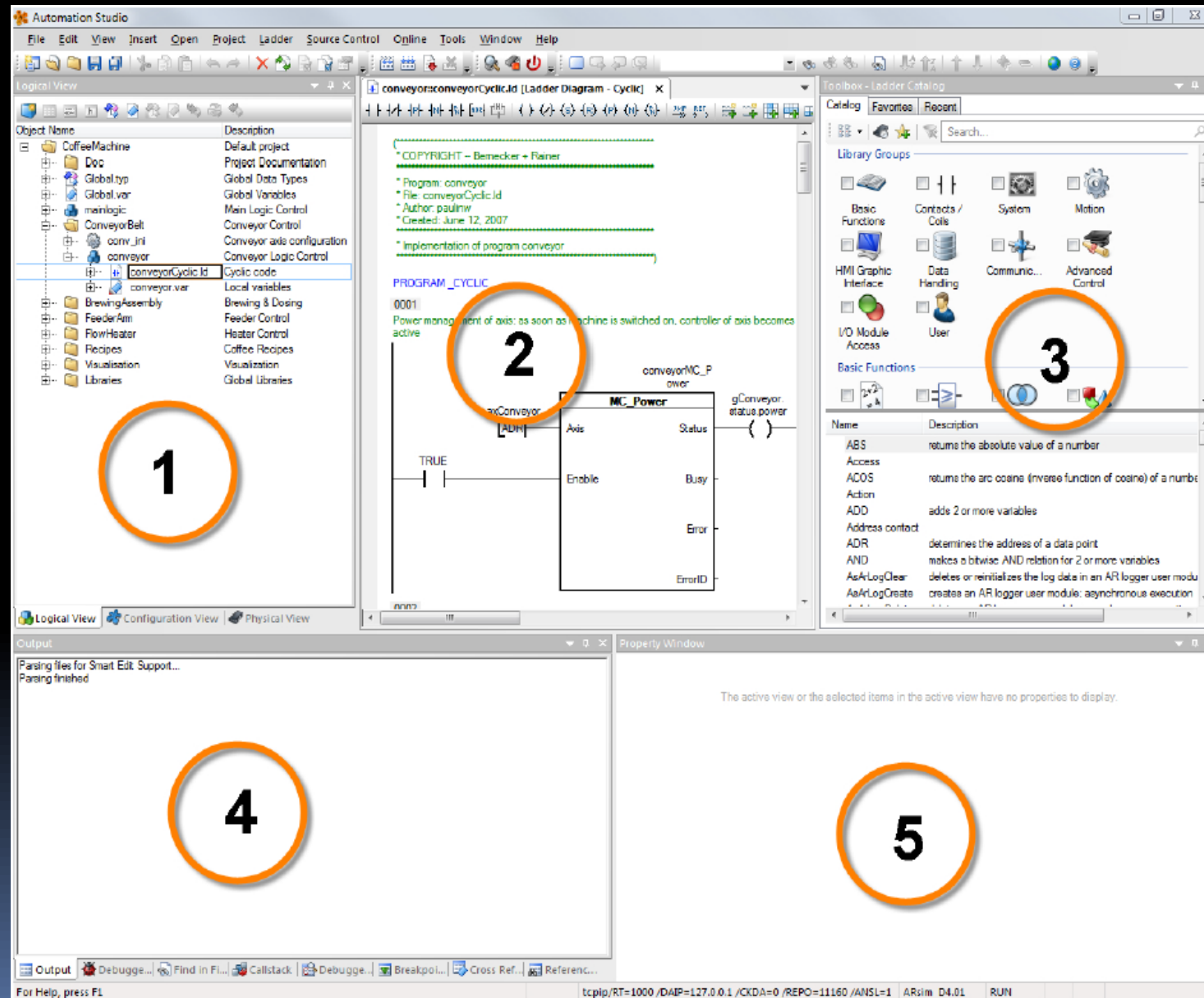
Классы задач.

Задача выполняется циклически в течение времени, определенного для её класса (task class), которое также называют временем цикла (cycle time).

Приоритет класса задач определяется его номером. Чем меньше номер, тем выше приоритет класса задач.

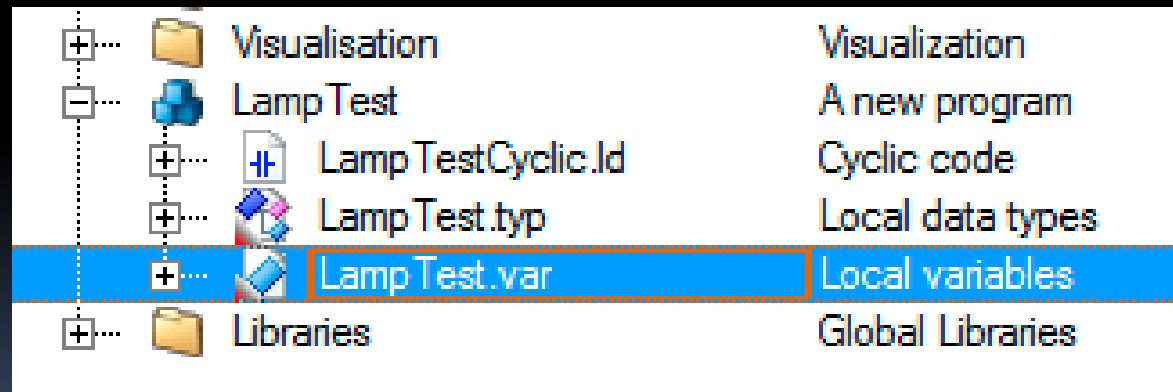


Automation studio.



Переменные.


Переменные представляют собой участки памяти, предназначенные для хранения данных определенного приложения. В среде Automation Studio переменные объявляются в файле .var.





Типы данных.

Типы данных описывают такие свойства переменной, как например, диапазон или точность числа, содержащегося в переменной, или какие операции могут выполняться с ней.



Типы данных.

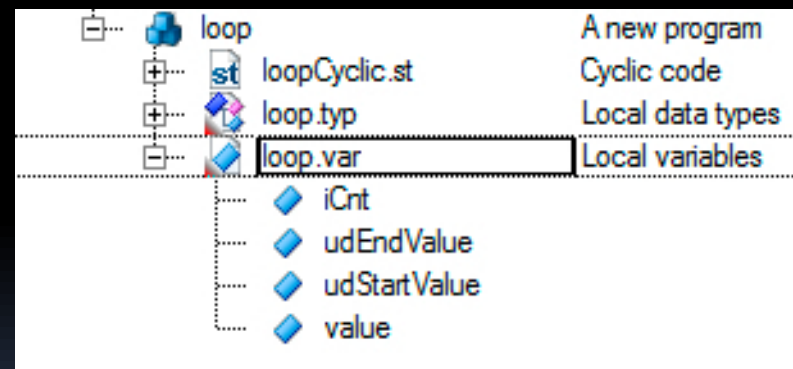
Type	Channel width (bytes)	Value range
BOOL	1	TRUE (1) and FALSE (0)
SINT	1	-128 ... +127
INT	2	-32768 ... +32767
DINT	4	-2147483648 ... +2147483647
USINT	1	0 ... 255
UINT	2	0 ... 65535
UDINT	4	0 ... 4294967295
REAL	4	-3.4E38 ... +3.4E38
LREAL	8	-1.79769313486231E308 ... +1.79769313486231E308
TIME	4	T#-24d_20h_31m_23s_648ms ... T#24d_20h_31m_23s_647ms
DATE	4	D#1970-01-01 ... D#2106-02-07
TIME_OF_DAY or TOD	4	TOD#00:00:00.000 ... TOD#23:59:59.999
DATE_AND_TIME or DT	4	DT#1970-01-01-00:00:00 ... DT#2106-02-07-06:28:15
STRING	1 per character	
BYTE	1	Bit sequence with a length of 8
WORD	2	Bit sequence with a length of 16
DWORD	4	Bit sequence with a length of 32
WSTRING	2 per character	

Глобальные и локальные переменные.

Область видимости и свойства переменной определяют ее поведение во время запуска и исполнения.

Локальные переменные:

Локальные переменные имеют локальную область видимости в пределах определенной программы и не могут использоваться в других программах.






Локальная переменная описывается файлом .var на том же уровне, что и программа.

Глобальные переменные.

Глобальные переменные отображаются на верхнем уровне панели Logical View и могут использоваться в любом месте проекта Automation Studio.

Глобальная переменная описывается на самом высоком уровне в файле Global.var. Для улучшения структурированности проекта могут быть созданы дополнительные файлы .var.

Object Name	Description
 CoffeeMachine	Default project
 Global.var	Global Variables
 gMain	

Глобальные переменные уровня пакета, которые объявлены в пакете, видны только в пределах этого пакета и всех подчиненных пакетов.

Инициализация области памяти для хранения переменной

По умолчанию во время инициализации переменным присваиваются значения «0».

При необходимости в файле объявления переменной могут быть указаны иные значения, которые должны быть присвоены при инициализации.

◆ iCnt	UDINT	<input type="checkbox"/>	0
◆ udStartValue	UDINT	<input type="checkbox"/>	344
◆ udEndValue	UDINT	<input type="checkbox"/>	120
◆ value	UDINT	<input type="checkbox"/>	0

Инициализация программы.

В каждой программе может содержаться блок инициализации.

В нем решаются следующие задачи:

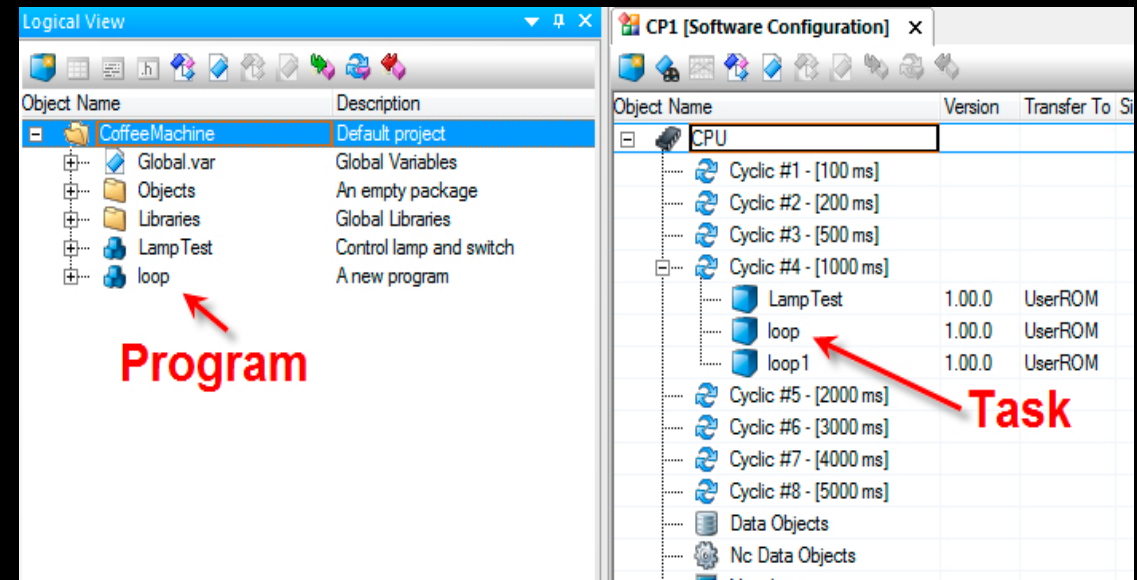
- инициализация переменных,
- считывание данных с ПЛК,
- считывание данных о текущей системной конфигурации.

Перед запуском первой циклической программы **однократно** выполняются подпрограммы инициализации всех задач в порядке, указанном в конфигурации программного обеспечения.

Циклические программы.

В конфигурации программного обеспечения программе назначается определенное время выполнения или класс задач.

Программы в конфигурации программного обеспечения называются задачами.



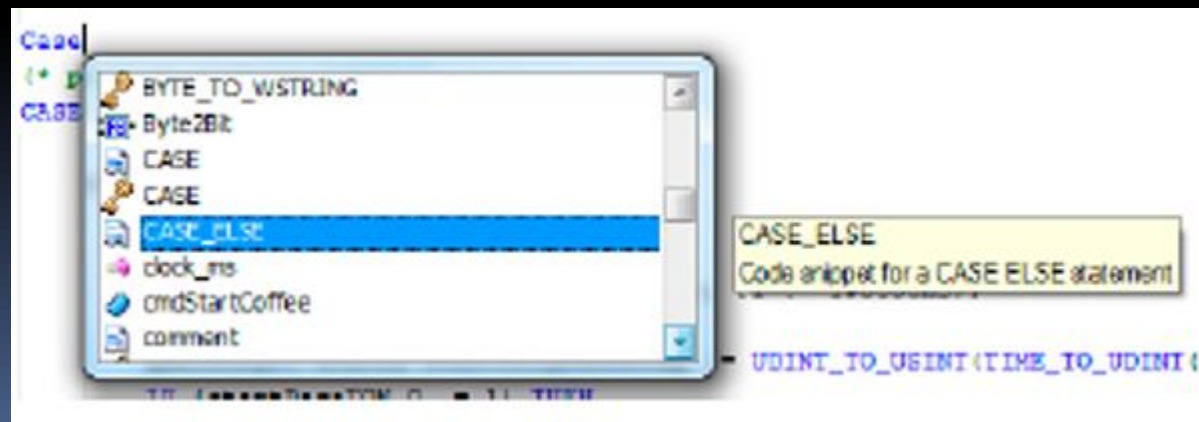
Smart Edit.

Автозаполнение

Для автоматического завершения кода используется клавиша <TAB> .

Данная функция поддерживается следующими элементами:

- Имена переменных и элементы структур
- Имя функции
- Языковые конструкции (IF THEN, CASE, FOR)



1111



The image shows two parts of the Visual Studio interface. The top part is the 'Insert Snippet' menu, which is open and shows a list of snippets: ACTION, CASE, CASE_ELSE (highlighted), comment, FOR, FUNCTION, FUNCTION_BLOCK, and IF. A yellow callout box points to 'CASE_ELSE' with the text: 'Code snippet for a CASE ELSE statement. Shortcut: CASE_ELSE'. The bottom part is the 'Code Snippet Manager' dialog. It shows the 'Language' set to 'ANSI C Snippets'. Under 'Existing Snippets:', there is a list of snippets including #if, #if#elif#else, #if#else, #ifdef, #ifdef#elif#else, #ifdef#else, #ifndef, #ifndef#elif#else, #ifndef#else, _EX, _INI, com, do, and else. The #ifdef snippet is selected. Below this, the 'Header' section shows the Title as '#ifdef', the Shortcut as '#ifdef', and the Description as 'Code snippet for #if'. The 'Snippet Type' section has checkboxes for 'Expansion' (checked) and 'Surrounds/With' (checked). The 'Code' section shows the snippet code: '#ifdef \$expression\$'. On the right, the 'Literals' table is visible, showing a table with columns 'ID' and 'Default', and a row with 'expression' in both columns.

Insert Snippet:

- ACTION
- CASE
- CASE_ELSE
- comment
- FOR
- FUNCTION
- FUNCTION_BLOCK
- IF

Code snippet for a CASE ELSE statement
Shortcut: CASE_ELSE

Code Snippet Manager

Language: ANSI C Snippets

Existing Snippets:

#if	#ifdef#else	_EX
#if#elif#else	#ifndef	_INI
#if#else	#ifndef#elif#else	com
#ifdef	#ifndef#else	do
#ifdef#elif#else	_CYCLIC	else

Header

Title: #ifdef

Shortcut: #ifdef

Description: Code snippet for #if

Snippet Type: ☒ Expansion ☒ Surrounds/With

Code:

```
#ifdef $expression$
```

Literals:

ID	Default
expression	expression

Добавление фрагментов кода

Горячие клавиши.

Функция	Сочетание клавиш
Заккрытие фрагментов кода	<TAB>
Автозаполнение имен переменных и функций	<CTRL> + <SPACE>
Активация окна выбора фрагмента кода	<CTRL> + <q>, <k>
Включение и отключение выделения всех парных скобок	<CTRL> + <q>, <l>
Включение и отключение выделения текущих парных скобок	<CTRL> + <q>, <m>
Переход к объявлению переменной	<CTRL> + <d>
Переход к указанию типа данных для переменной	<CTRL> + <t>
Переход к реализации функции или функционального блока	<CTRL> + <i>