**eval**

*Задание интерпретации текстовых выражений в процессе выполнения программы.*

**Синтаксис:**

eval(<строковая переменная>);

**Описание:**

При помощи ключевого слова eval язык программирования позволяет компилировать текстовые выражения непосредственно при выполнении программы.

**Пример 2:**

|  |  |
| --- | --- |
|  | **beforecompile**    //Эта секция выполняется ДО компиляции остального скрипта    **if** **signalexist**(Name1+"\_L1\_SET") **then eval**("define ISST1") **else eval**("undefine ISST1");  **if** **signalexist**(Name1+"\_H1\_SET") **then eval**("define ISST4") **else eval**("undefine ISST4");    //определяем флаги наличия уставок    //если какой-то уставки нет, то выключаем флаг !!! иначе - //включаем  **if signalexist**("AS\_"+Name1+"\_L1\_ON") **then eval**("define LAS11") **else eval**("undefine LAS11");  **if signalexist**("AS\_"+Name1+"\_H1\_ON") **then eval**("define HAS11") **else eval**("undefine HAS11");    **if signalexist**("PS\_"+Name1+"\_L1\_ON") **then eval**("define LPS11") **else eval**("undefine LPS11");    if signalexist("PS\_"+Name1+"\_H1\_ON") **then eval**("define HPS11") **else eval**("undefine HPS11");  //теперь, если флаг есть - то уставка есть, иначе - не делаем // ничего с этой уставкой  **end**; |

В примере определена секция скрипта, выполняемая до компиляции основного скрипта. В секции определяется наличия уставок для сигналов, если какой-то уставки нет, то выключаем флаг наличия уставки, иначе – включаем. При помощи ключевого слова **eval** определяются текстовые выражения, которые интерпретируются в момент выполнения скрипта.