

**Регулирование температуры термоцикла по заданному закону**

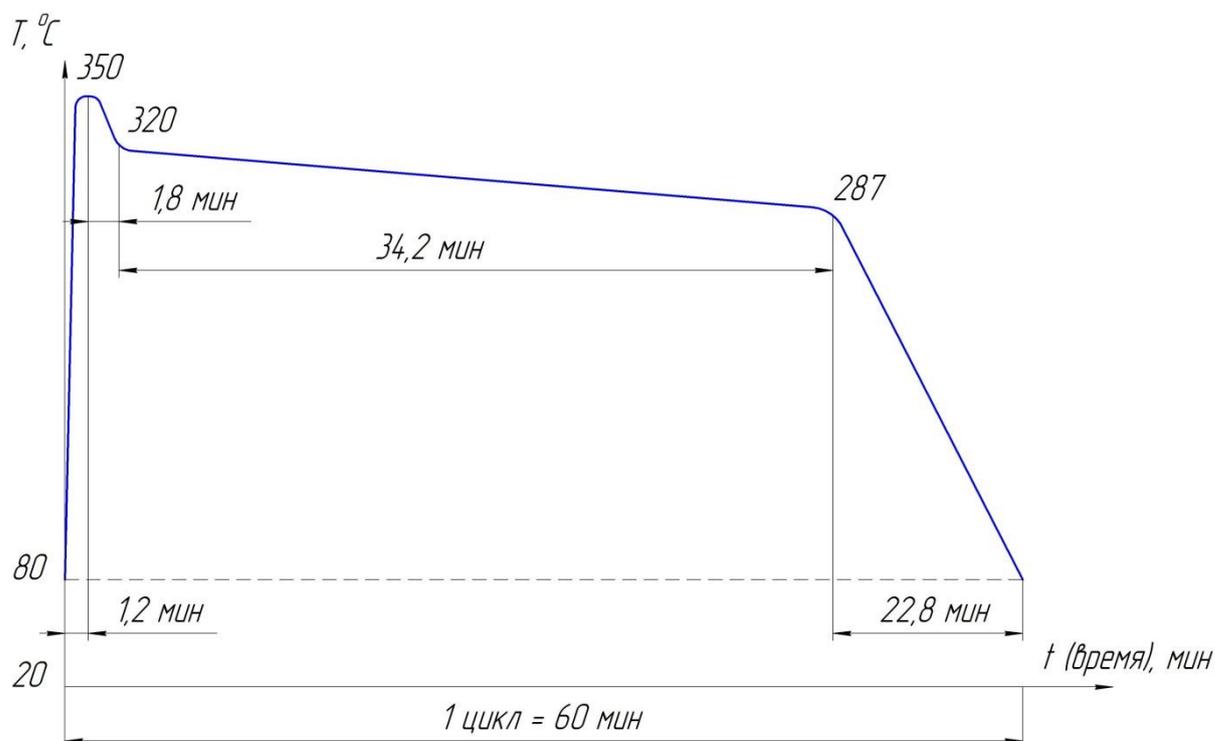
**Задание:** написать программу регулирования, обеспечивающую термоциклирование композитных образцов по заданному закону из температуры от времени цикла.

**Исходные данные:**

начальная температура по термопаре –  $80^{\circ}\text{C}$ ;

количество циклов для реализации – 3;

Требуемый график термоцикла приведен ниже:



**Примечание:** при учебной реализации процесса необходимо изменить время цикла, качественно сохранив график термоциклирования.

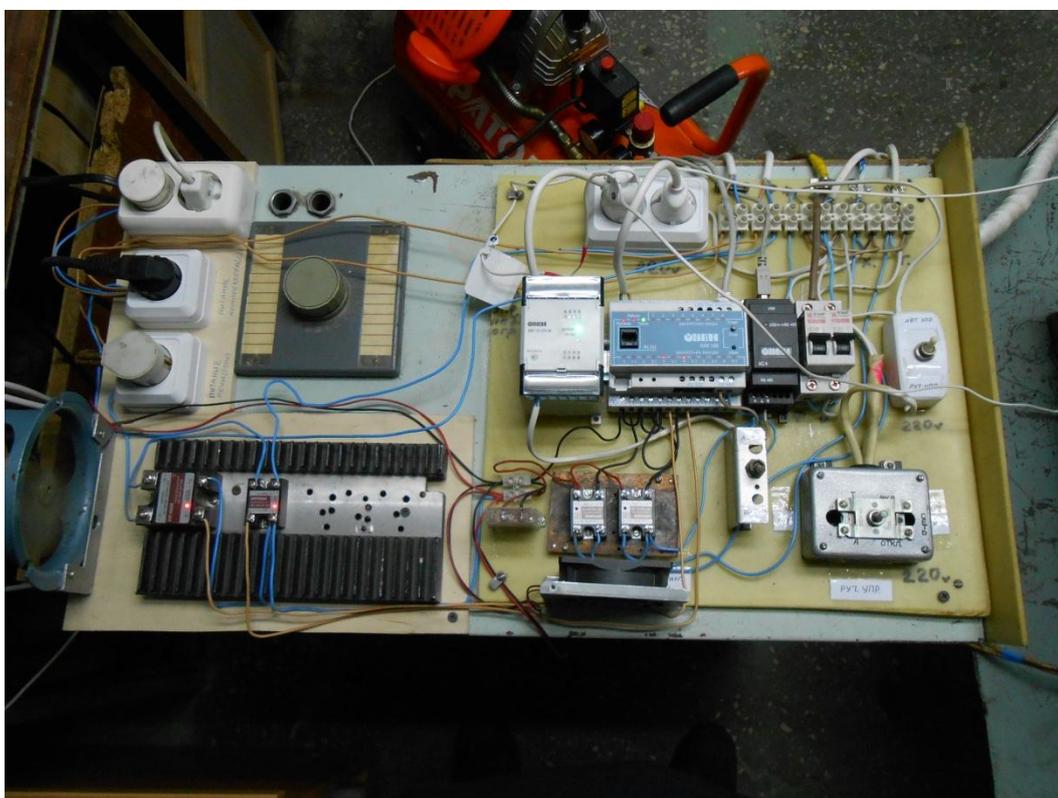
Этап 1 – нагрев с  $80$  до  $350^{\circ}\text{C}$  – может осуществляться с максимальной скоростью  $10^{\circ}\text{C}/\text{сек}$ ;

Этап 2 – быстрое охлаждение с  $350$  до  $320^{\circ}\text{C}$  в течение 20 секунд – максимальная скорость охлаждения в этом случае может составлять  $5^{\circ}\text{C}/\text{сек}$ ;

Этап 3 – плавное охлаждение с  $320$  до  $287^{\circ}\text{C}$  в течение 120 секунд – максимальная скорость охлаждения в этом случае может составлять  $5^{\circ}\text{C}/\text{сек}$ ;

Этап 4 – плавное охлаждение с  $287$  до  $80^{\circ}\text{C}$  в течение 60 секунд – максимальная скорость охлаждения в этом случае может составлять  $5^{\circ}\text{C}/\text{сек}$ ;

Далее приведены фотографии реальной установки для термоциклирования и график реализованного с ее помощью термоцикла.



**Рис. 1.** Фотография установки для испытаний КМ при термоциклических воздействиях

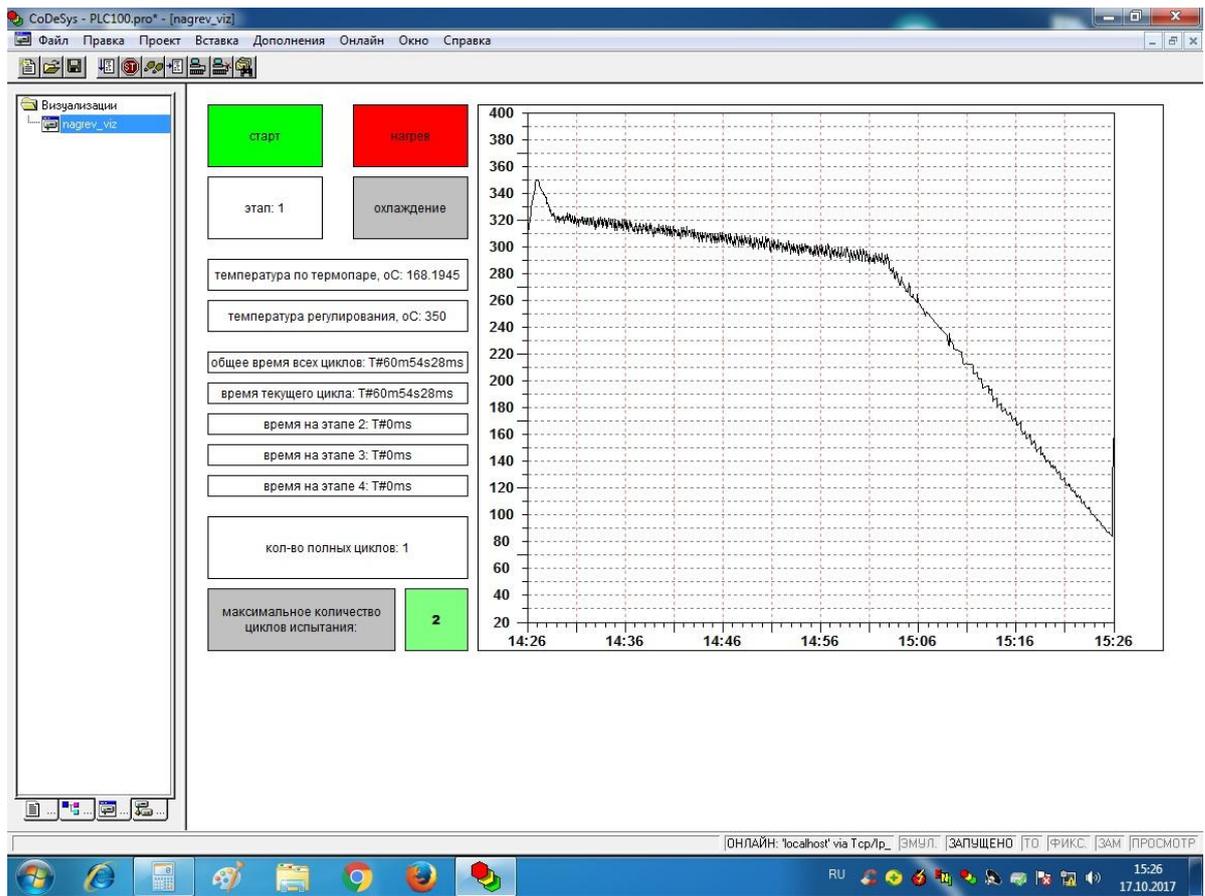
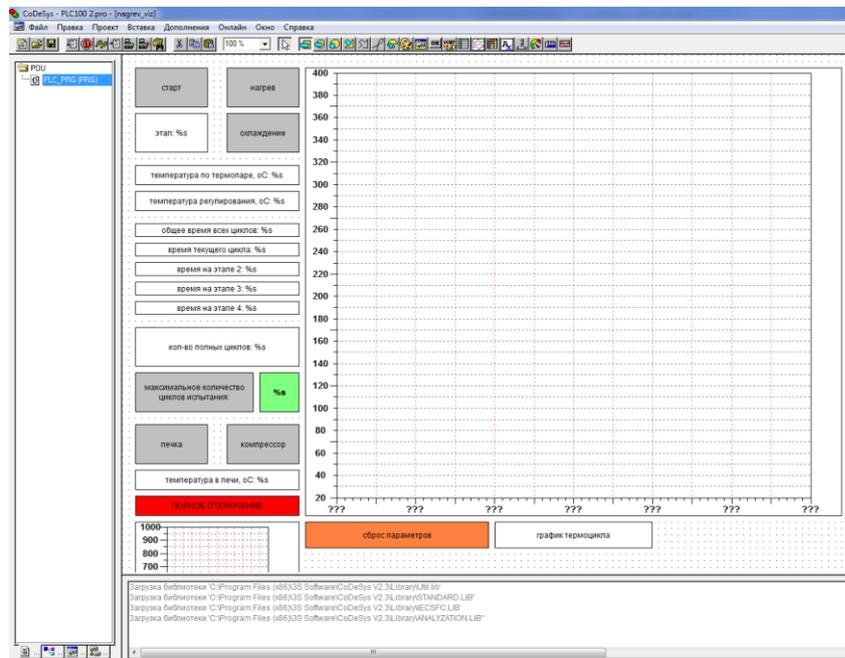
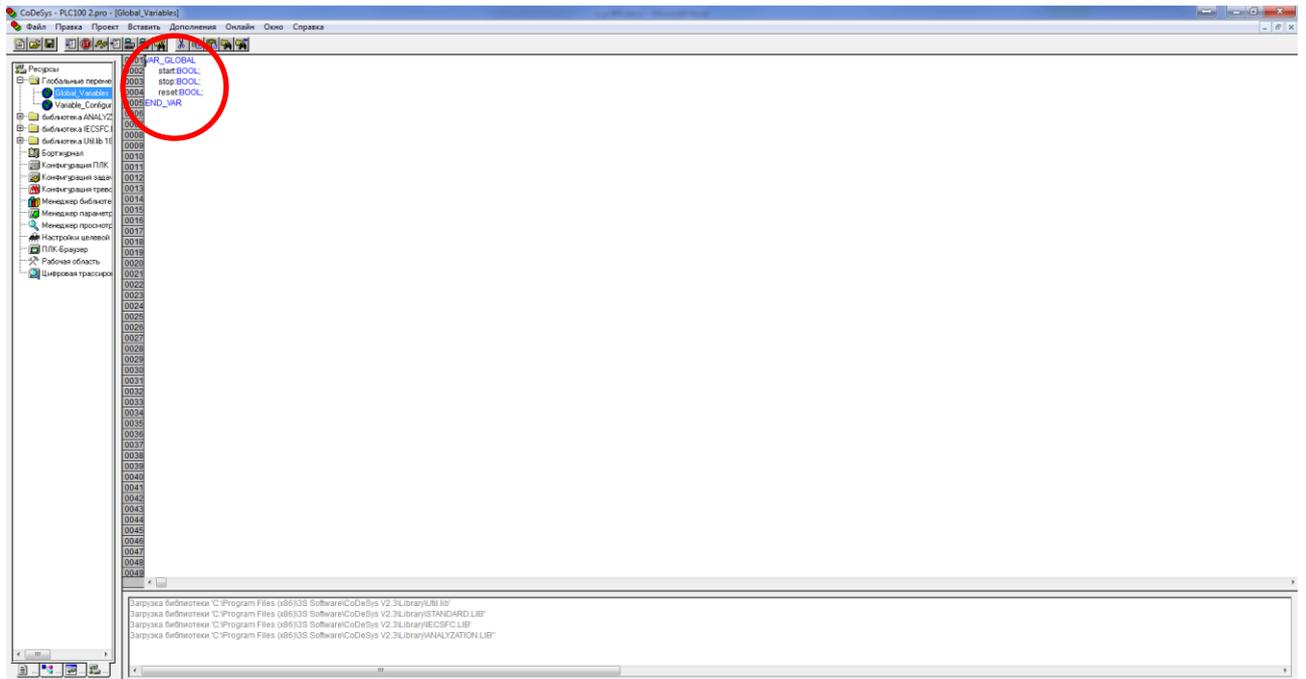


Рис. 2. График реализованного на испытательной установке термоцикла

Окончательный вид визуализации программы регулирования представлен на рис. далее.



В качестве глобальных переменных объявлены следующие переменные

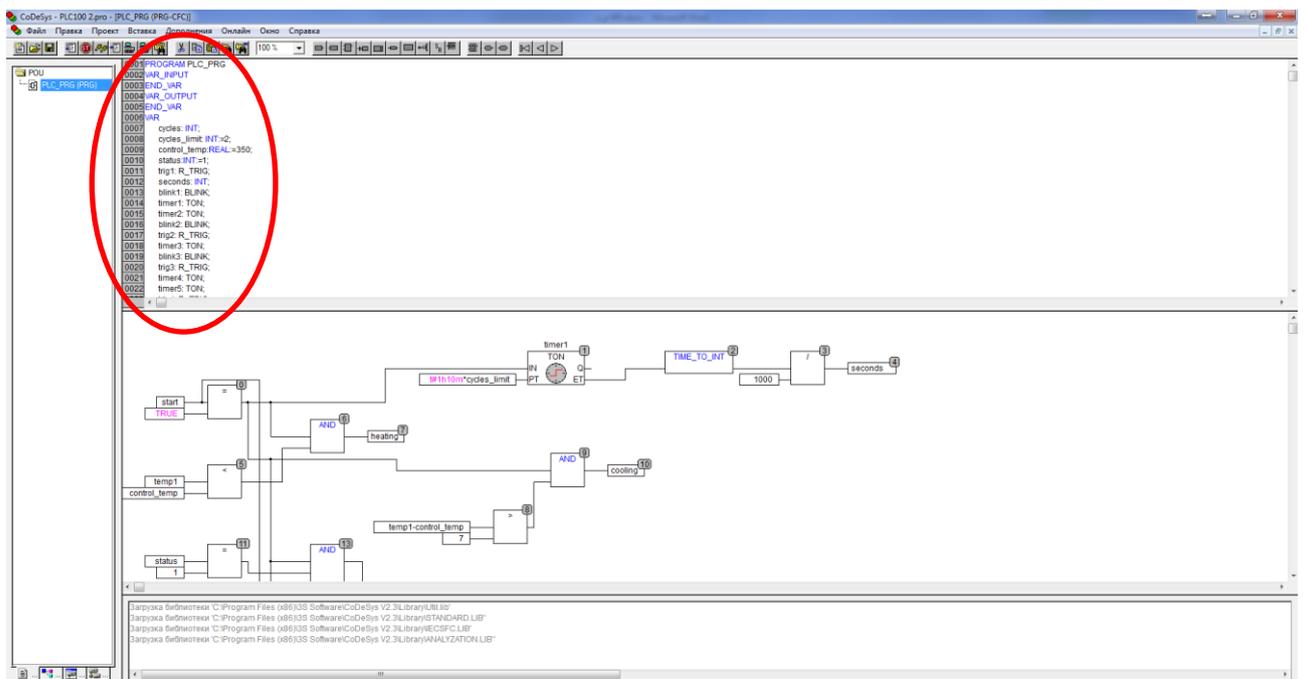


К данным переменным необходимо добавить также булевы переменные **heating** и **cooling**.

### BOOL константы:

BOOL константы могут иметь значение TRUE или FALSE.

В качестве локальных переменных используем переменные, представленные на рисунке далее.



### REAL/LREAL константы:

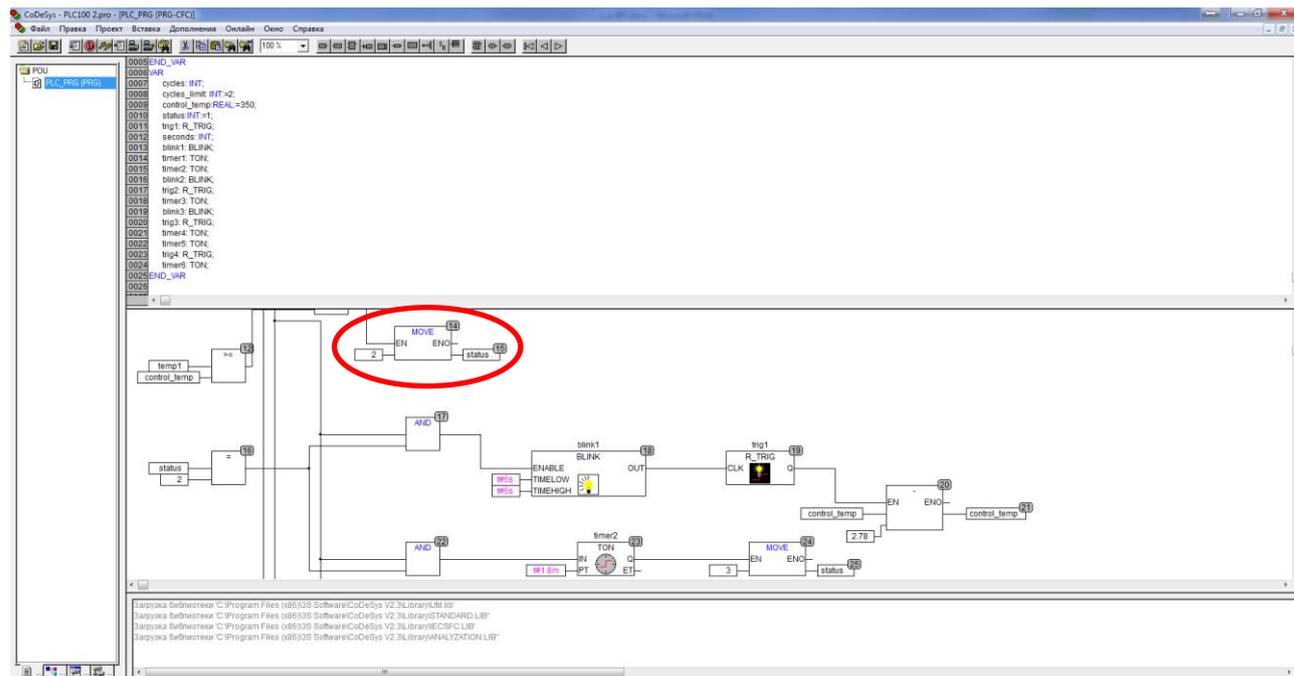
REAL и LREAL константы представляются в формате с десятичной точкой либо в экспоненциальном формате. Запятая вместо точки не допускается.

## Примеры:

7.4 но не 7,4

1.64e+009 но не 1,64e+009

Для перехода от этапа к этапу используется переменная **status**.



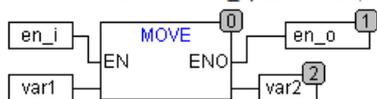
## Команда MOVE

### MOVE

Присвоение значения одной переменной другой соответствующего типа. В графических редакторах CFC и LD существует возможность управлять разрешением работы блока (разрешать или запрещать операцию) с помощью входов EN/ENO. В FBD этого делать нельзя.

**Пример применения EN/ENO в CFC:**

Только если значение *en\_i* равно TRUE, значение переменной *var1* будет присвоено *var2*.



**Пример IL:**

```
LD ivar1
```

```
MOVE ivar2
```

```
ST ivar2 (* Результат: ivar2 принимает значение ivar1 *)
```

(! Аналогичный результат дает:

```
LD ivar1
```

```
ST ivar2 )
```

**Пример ST:**

```
ivar2 := MOVE(ivar1);
```

(! Аналогичный результат дает: ivar2 := ivar1;)



## R\_TRIG

Входит в *standard.lib*.

Детекторы импульсов

```
FUNCTION_BLOCK R_TRIG
VAR_INPUT
  CLK : BOOL;
END_VAR
VAR_OUTPUT
  Q : BOOL;
END_VAR
VAR
  M : BOOL := FALSE;
END_VAR
Q := CLK AND NOT M;
M := CLK;
```

Функциональный блок R\_TRIG генерирует импульс по переднему фронту входного сигнала.

Выход Q равен FALSE до тех пор, пока вход CLK равен FALSE. Как только CLK получает значение TRUE, Q устанавливается в TRUE. При следующем вызове функционального блока выход сбрасывается в FALSE. Таким образом, блок выдает единственный импульс при каждом переходе CLK из FALSE в TRUE.

Пример объявления:

```
RTRIGInst : R_TRIG ;
```

Пример IL:

```
CAL RTRIGInst (CLK := VarBOOL1)
LD RTRIGInst.Q
ST VarBOOL2
```

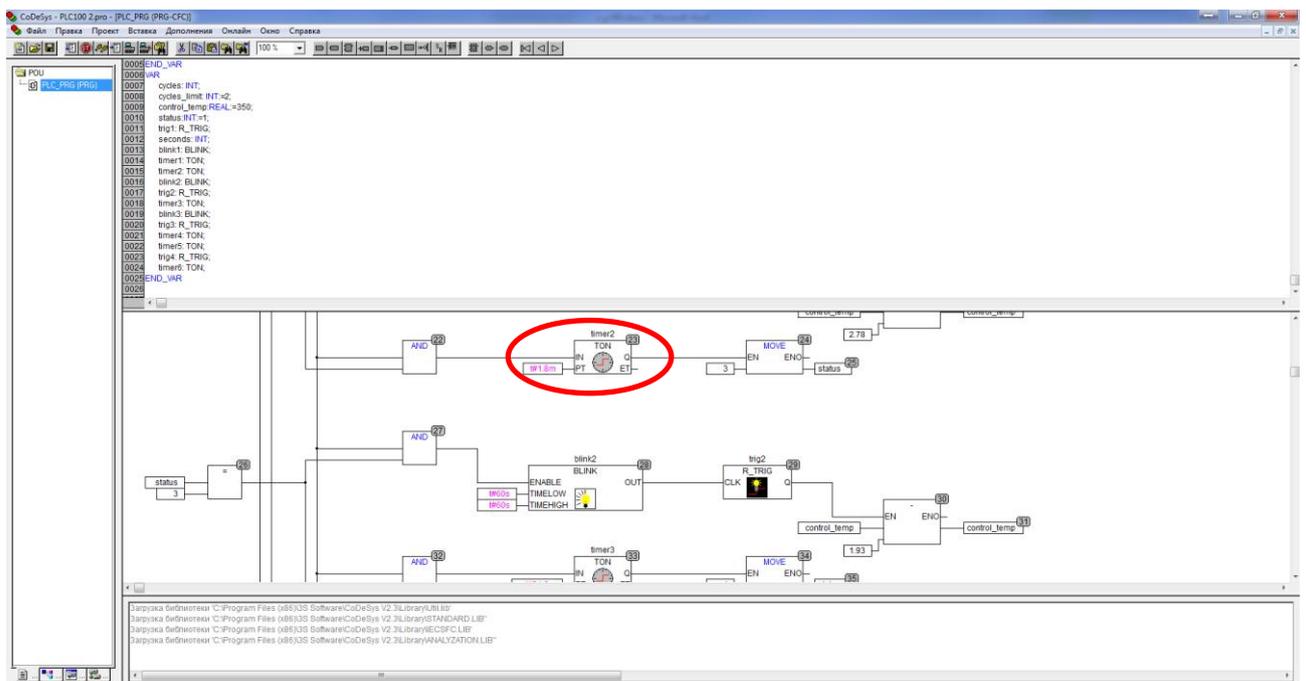
Пример FBD:



Пример ST:

```
RTRIGInst (CLK:= VarBOOL1);
VarBOOL2 := RTRIGInst.Q;
```

Для перехода с этапа, на котором требуется обеспечить определенное время выдержки, на следующий этап, используется команда **TON**.



## TON

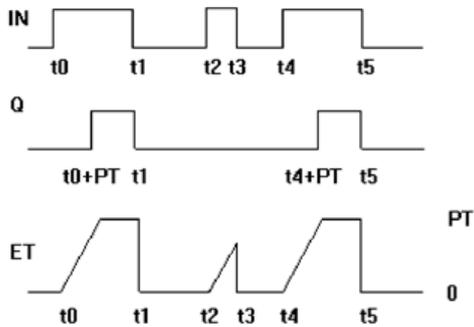
Входит в *standard.lib*.

Функциональный блок 'таймер с задержкой включения'.

**TON**(IN, PT, Q, ET) Входы IN и PT типов BOOL и TIME соответственно. Выходы Q и ET аналогично типов BOOL и TIME.

Пока IN равен FALSE, выход Q = FALSE, выход ET = 0. Как только IN становится TRUE, начинается отсчет времени (в миллисекундах) на выходе ET до значения, равного PT. Далее счетчик не увеличивается. Q равен TRUE, когда IN равен TRUE и ET равен PT, иначе FALSE. Таким образом, выход Q устанавливается с задержкой PT от фронта входа IN.

Временная диаграмма работы **TON**:



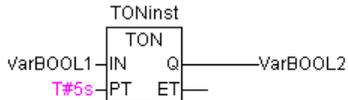
Пример объявления:

```
TONInst : TON ;
```

Пример IL:

```
CAL TONInst(IN := VarBOOL1, PT := T#5s)
LD TONInst.Q
ST VarBOOL2
```

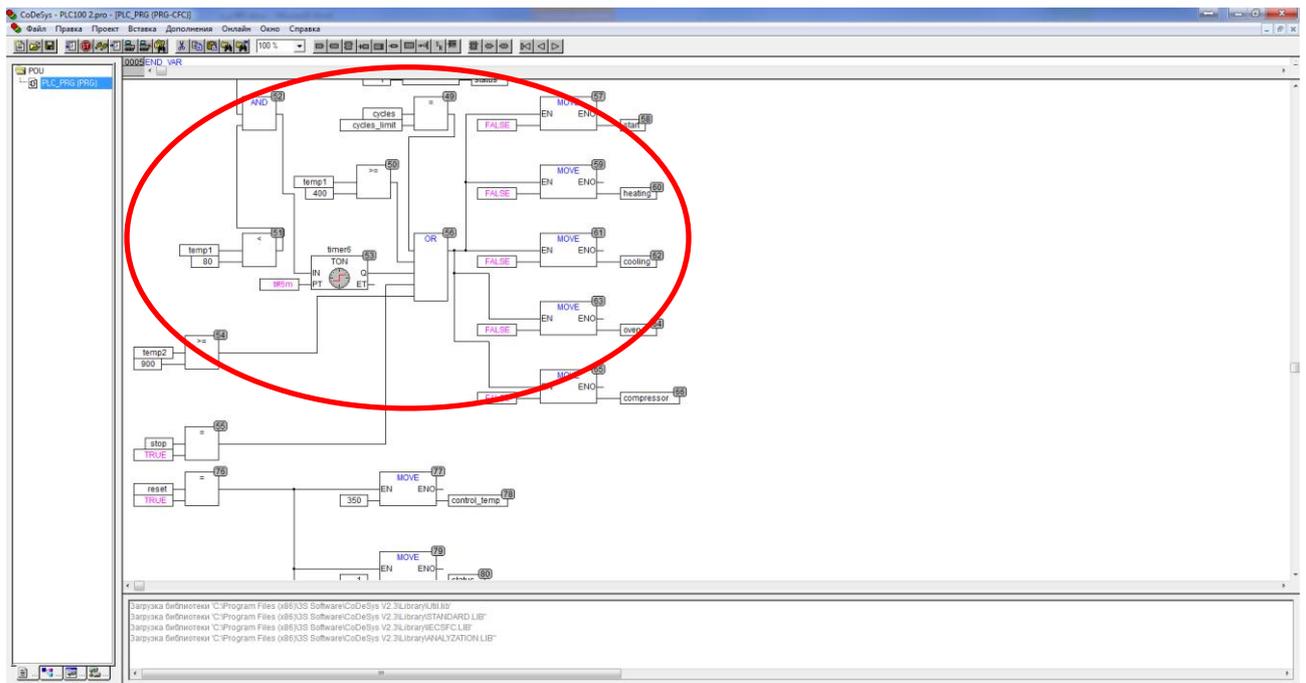
Пример FBD:



Пример ST:

```
TONInst(IN := VarBOOL1, PT:= T#5s);
```

Автоматическое отключение после достижения необходимого числа циклов реализовано комплексом условий с операторами **EQ** (=), **AND** и **OR**.



## EQ

Равно

Двоичный оператор возвращает TRUE, если значение первого параметра равно второму.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME\_OF\_DAY, DATE\_AND\_TIME и STRING.

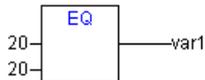
Пример IL:

```
LD 40
EQ 40
ST Var1 (*Результат - ИСТИНА*)
```

Пример ST:

```
VAR1 := 40 = 40;
```

Пример FBD:



## AND

Побитное И. Операция применима к типам BOOL, BYTE, WORD или DWORD.

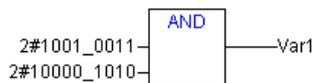
Пример IL:

```
Var1 BYTE
LD 2#1001_0011
AND 2#1000_1010
ST Var1 (* Результат 2#1000_0010 *)
```

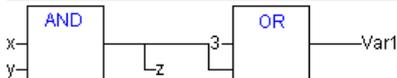
Пример ST:

```
var1 := 2#1001_0011 AND 2#1000_1010
```

Пример FBD:



**Внимание:** В логических выражениях нельзя гарантировать присваивание промежуточных результатов. Например, если условие SFS перехода выглядит так:



то, значение переменной z может быть не присвоено. Это происходит по причине оптимизации вычислений компилятором. Если значения x и y FALSE, то конечный результат очевиден и остаток выражения вычислять не нужно.

## OR

Побитное ИЛИ. Операция применима к типам BOOL, BYTE, WORD или DWORD.

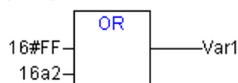
Пример IL:

```
var1 :BYTE;
LD 2#1001_0011
OR 2#1000_1010
ST var1 (* Результат 2#1001_1011 *)
```

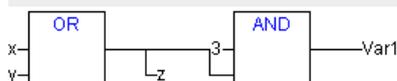
Пример ST:

```
Var1 := 2#1001_0011 OR 2#1000_1010
```

Пример FBD:



**Внимание:** См. примечание к AND.



## Дополнительные команды, используемые в проекте:

### Команда «+» эквивалентна команде **ADD**

#### **ADD**

Сложение переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL.

Две переменных типа TIME можно складывать (напр.  $t\#45s + t\#50s = t\#1m35s$ ). Результат имеет тип TIME.

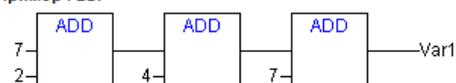
Пример IL:

```
LD 7
ADD 2,4,7
ST Var1
```

Пример ST:

```
var1 := 7+2+4+7;
```

Пример FBD:



### Команда «-» эквивалентна команде **SUB**

#### **SUB**

Вычитание значений переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL.

Переменной TIME можно присвоить результат вычитания двух других переменных типа TIME. Отрицательное время не определено.

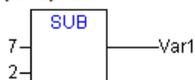
Пример IL:

```
LD 7
SUB 2
ST Var1
```

Пример ST:

```
var1 := 7-2;
```

Пример FBD:



### Команда «\*» эквивалентна команде **MUL**

#### **MUL**

Перемножение значений переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL.

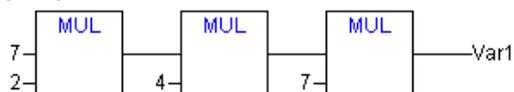
Пример IL:

```
LD 7
MUL 2,4,7
ST Var1
```

Пример ST:

```
var1 := 7*2*4*7;
```

Пример FBD:



### Команда «>» эквивалентна команде **GE**

## GE

Больше или равно

Двоичный оператор возвращает TRUE, если значение первого параметра больше или равно второму.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME\_OF\_DAY, DATE\_AND\_TIME и STRING.

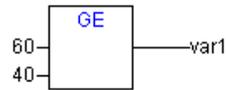
**Пример IL:**

```
LD 60
GE 40
ST Var1 (*Результат - ИСТИНА*)
```

**Пример ST:**

```
VAR1 := 60 >= 40;
```

**Пример FBD:**



Команда «<>» эквивалентна команде **LT**

## LT

Меньше

Двоичный оператор возвращает TRUE, если значение первого параметра меньше второго.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME\_OF\_DAY, DATE\_AND\_TIME и STRING.

**Пример IL:**

```
LD 20
LT 30
ST Var1 (* Результат ИСТИНА* )
```

**Пример ST:**

```
VAR1 := 20 < 30;
```

**Пример FBD:**

