



# UNI-PRO

**СРЕДА РАЗРАБОТКИ ДЛЯ  
ПРОГРАММИРУЕМЫХ КОНТРОЛЛЕРОВ**



**РУКОВОДСТВО ПО СТАНДАРТНЫМ  
БИБЛИОТЕКАМ**

**CODE 114UPROSLE44**

**Важное замечание**

Перед использованием следует внимательно прочитать данное Руководство по эксплуатации и соблюдать все предупреждения, руководство следует сохранить для дальнейшего использования.

# Содержание

1 ОСНОВНЫЕ ПОЛОЖЕНИЯ .....	9
2 СИСТЕМНЫЕ БИБЛИОТЕКИ .....	10
2.1 Analogue Inputs .....	10
Humidity_4_20_mA .....	10
Pressure_4_20_mA_0_30_Bar .....	10
Pressure_4_20_mA_0_7_Bar .....	11
Pressure_4_20_mA_General .....	11
Temperature_RESISTANCE_NTC .....	12
Temperature_RESISTANCE_Ni1000 .....	13
SelectUniversalAIN .....	14
AI_TimeoutError .....	15
2.2 Analogue Outputs .....	16
SelectTypeAOn .....	16
PWM_FrequencyAOn .....	17
2.3 BACnet MSTP .....	18
BACnetMSTP_BaudRate .....	18
BACnetMSTP_DeviceInstance .....	19
BACnetMSTP_MacID .....	19
BACnetMSTP_MaxMaster .....	20
2.4 BACnet IP .....	21
BACnetIP_DeviceInstance .....	21
BACnetIP_PortNumber .....	21
BACnetIP_BBMD_IPAddress.....	22
BACnetIP_BBMD_PortNumber .....	23
BACnetIP_BBMD_TimeToLive .....	23
2.5 Browser .....	25
AddBrowserGraph_240_128_Type .....	25
AddBrowserColor_320_240_Type .....	26
AddBrowserGraph_240_140_Type .....	27
AddBrowserGraph_128_64_Type .....	27
AddBrowserGraph_120_32_Type .....	28
AddBrowserAlpha_4_20_Type .....	28
AddBrowserAlpha_4_16_Type .....	29
AddBrowserSevenSeg_1_3D_Type .....	29
AddBrowserSevenSeg_1_4D_Type .....	30
2.6 1 <sup>st</sup> CAN & 2 <sup>nd</sup> CAN.....	31
CAN1st_IsMaster.....	31
CAN1st_MyNode .....	32
CAN1st_BaudRate .....	33
CAN1st_GetStatus .....	34
CAN1st_GetNetworkStatus .....	34
CAN1st_NetworkNode_n .....	35
CAN1st_NodeGuardTimeout .....	35
2 <sup>nd</sup> CAN.....	36
2.7 CAN Master .....	37
CAN_Read10MultipleHoldingRegisters .....	37
CAN_Write10MultipleHoldingRegisters .....	38

2.8	History .....	40
	HistoryErase .....	40
	HistoryReadEvent .....	40
	HstoryWriteEvent .....	41
2.9	Data Log .....	43
	DataLog_RecordingTime .....	43
	DataLog_StartStop .....	43
	DataLog_Erase .....	44
	DataLog_ItemEnableLog .....	44
	DataLog_PrintDataToFileUSB .....	45
	DataLog_ExtractRecordData .....	46
	DataLog_GetMemoryStatus .....	49
	DataLog_GetErrors .....	49
	DataLog_GetStatus .....	50
2.10	Identity .....	51
2.11	IO Utils .....	52
	AI_to_DI .....	52
	AI_to_RESISTANCE .....	52
	DI_Frequency.....	53
	DI_PulseCounter .....	53
	AI_Config .....	54
	AO_Config .....	55
	DI_Config .....	56
	DO_Config .....	57
2.12	Ethernet TCP/IP .....	58
	Ethernet_IP_Address - Ethernet_Gateway - Ethernet_SubnetMask .....	58
2.13	WebServer TCP .....	58
	WebServer_PortNum .....	58
	WebServer_PasswordToWrite .....	59
	WebServer_PasswordLevel1 .. WebServer_PasswordLevel3 .....	60
2.14	Messages .....	61
	Send_Async_Mail .....	61
	Send_Async_SMS .....	62
	Send_Async_Ping .....	64
2.15	Modbus.....	65
	Modbus1_Address – Modbus2_Address .....	65
	Modbus1_Baud – Modbus2_Baud.....	65
	Modbus1_Parity – Modbus2_Parity .....	66
	Modbus1_StopBit – Modbus2_StopBit .....	67
	Modbus1_Protection_SetAccess – Modbus2_Protection_SetAccess .....	67
	Modbus1_Protection_SetLevel – Modbus2_Protection_SetLevel .....	68
	Modbus1_Protection_Get – Modbus2_Protection_Get .....	69
	Modbus1_Protection_Timeout – Modbus2_Protection_Timeout .....	69
	ModbusMaster_Timeout .....	70
	ModbusMaster_Interframe .....	71
	ModbusMaster_GetNetworkStatus .....	71
	ModbusMaster_UART_Polarization .....	72
	ModbusMaster_ReadSingleCoilStatus .....	73
	ModbusMaster_ReadSingleInputStatus .....	74
	ModbusMaster_ReadSingleInputRegister .....	75
	ModbusMaster_ReadSingleHoldingRegister .....	77

ModbusMaster_WriteSingleCoilStatus .....	78
ModbusMaster_WriteSingleHoldingRegister .....	79
2.16 Modbus TCP .....	81
ModbusTCP_Protection_SetAccess .....	81
ModbusTCP_Protection_SetLevel .....	81
ModbusTCP_Protection_Get .....	82
ModbusTCP_Protection_Timeout .....	82
ModbusSlave_TCP_PortNum .....	83
ModbusMaster_TCP_PortNum .....	84
ModbusMaster_TCP_ReadSingleCoilStatus .....	84
ModbusMaster_TCP_ReadSingleInputStatus .....	86
ModbusMaster_TCP_ReadSingleInputRegister .....	87
ModbusMaster_TCP_ReadSingleHoldingRegister .....	88
ModbusMaster_TCP_WriteSingleCoilStatus .....	90
ModbusMaster_TCP_WriteSingleHoldingRegister .....	91
2.17 Password .....	94
EnablePrgLevel .....	94
PasswordIndependent.....	95
PasswordTimeout .....	95
PasswordLevel1 – PasswordLevel5 .....	96
EnableLevel1 – EnableLevel5 .....	97
2.18 System .....	98
InfoMaskNum .....	98
InfoMaskRev .....	98
InfoMaskVer .....	98
InfoProjNum .....	99
InfoProjRev .....	99
InfoProjVer .....	99
E2_GetStatus .....	100
RCT_GetStatus .....	100
Language .....	101
LCDBacklight .....	101
LCDTouch_Buzzer .....	102
Reset .....	102
SW_Reboot .....	103
Param_GetStatus .....	103
Enable_Led_RUN e Enable_Led_COMM .....	104
3 СТАНДАРТНЫЕ БИБЛИОТЕКИ .....	105
3.1 Arithmetic .....	105
ADD .....	105
SUB .....	106
MUL .....	106
DIV.....	107
MOD .....	108
ABS .....	109
SQRT .....	110
POW .....	110
3.2 Bistable.....	112
SR .....	112
RS .....	113
FLIP-FLOP .....	114

MONOS .....	115
3.3 Bit Shift .....	116
SHL (Shift Left) .....	116
SHR (Shift Right).....	117
ROL (Rotary Left) .....	118
ROR (Rotary Right) .....	119
3.4 Comparators .....	120
EQ (Equal Comparator) .....	120
GE (Greater or Equal Comparator) .....	120
GT (Greater Comparator) .....	121
LE (Less or Equal Comparator) .....	121
LT (Less Comparator) .....	122
NE (Not Equal Comparator) .....	123
U_GE (Unsigned Greater or Equal Comparator) .....	123
U_GT (Unsigned Greater Comparator) .....	124
U_LE (Unsigned Less or Equal Comparator) .....	124
U_LT (Unsigned Less Comparator) .....	125
3.5 Conversion .....	126
ANALOG_GET_ERROR .....	126
ANALOG_GET_VALUE .....	126
TO_ANALOG .....	127
ANALOG_GET_FIELDS .....	127
Celsius2Fahrenheit .....	128
Fahrenheit2Celsius .....	128
DATETIME2FIELDS .....	128
FIELDS2DATETIME .....	129
DATETIME_GET_YEAR, DATETIME_GET_MONTH and DATETIME_GET_DAY ....	130
DATETIME_GET_WEEKDAY .....	130
DATETIME_GET_HOUR, DATETIME_GET_MIN and DATETIME_GET_SEC .....	131
DATETIME_GET_TIME .....	131
BYTE2BIT .....	132
BYTE2SBYTE.....	132
BYTE2LED .....	133
LED2BYTE .....	133
DWORD2LONG .....	134
LONG2DWORD .....	134
SBYTE2BYTE.....	135
SHORT2WORD .....	135
WORD2SHORT .....	136
BUTTON_GET_FIELDS .....	136
COMMAD_GET_FIELDS .....	137
16BITS2WORD .....	138
WORD216BITS .....	139
CondensationP/TConversion .....	139
3.6 Counters .....	141
CTD / MCTD (Down Counter) .....	141
CTU / MCTU (Up Counter).....	142
CTUD (Up Down Counter).....	144
3.7 Edge detection .....	146
F_TRIG .....	146
R_TRIG .....	146

FR_TRIG .....	147
3.8 Linear .....	148
A_LINEAR_CONVERTER .....	148
A_LOW_PASS_FILTER - S_LOW_PASS_FILTER .....	149
A_HIGH_PASS_FILTER - S_HIGH_PASS_FILTER .....	150
LINEAR_RAMP .....	151
3.9 Logic .....	152
AND (0-1 AND) .....	152
OR (0-1 OR).....	153
NOT (0-1 NOT) .....	153
NAND (0-1 NAND).....	154
NOR (0-1 NOR) .....	154
XOR (0-1 XOR) .....	155
AND16 (16 bit AND) .....	155
OR16 (16 bit OR) .....	156
NOT16 (16 bit NOT) .....	156
NAND16 (16 bit NAND) .....	157
NOR16 (16 bit NOR) .....	157
NOR16 (16 bit NOR) .....	158
AND32 (32 bit AND) .....	158
OR32 (32 bit OR) .....	159
NOT32 (32 bit NOT) .....	159
NAND32 (32 bit NAND) .....	160
NOR32 (32 bit NOR) .....	160
XOR32 (32 bit XOR) .....	161
3.10 Not Linear .....	162
DEAD_BAND .....	162
LIMIT.....	163
ZONE .....	164
PID .....	165
PI .....	167
PID_ADV.....	168
PID_AT .....	170
THERMO_HOT.....	173
THERMO_COLD .....	173
THERMO_HOT_COLD .....	174
3.11 Selection .....	176
MAX_BIT .....	176
MAX_BYTE .....	176
MAX_DWORD .....	177
MAX_LONG .....	177
MAX_SHORT .....	178
MAX_S_BYTE .....	178
MAX_WORD .....	179
MIN_BIT .....	179
MIN_BYTE .....	180
MIN_DWORD .....	180
MIN_LONG .....	181
MIN_SHORT .....	181
MIN_S_BYTE .....	182
MIN_WORD .....	182

SEL_BIT .....	183
SEL_BYTE .....	183
SEL_DWORD .....	184
SEL_LONG .....	184
SEL_SHORT .....	185
SEL_S_BYTE .....	185
SEL_WORD .....	186
3.12 Timers .....	187
TOF (Off Delay Timer).....	187
TON (On Delay Timer).....	188
TP (Pulse Timer) .....	189
3.13 Timing .....	191
TDELAY .....	191
TSHIFT .....	192
TSUSTAIN .....	193
PWM .....	194



## 1 ОСНОВНЫЕ ПОЛОЖЕНИЯ

В этом документе перечислены библиотеки, поставляемые с программой UNI-PRO Development Environment.

Рассматриваемые библиотеки можно разделить на две основные группы:

Standard libraries (Стандартные библиотеки)

System libraries (Системные библиотеки)

Стандартные библиотеки содержат все необходимые детали для реализации типового проекта. Они предоставляют функции для выполнения вычислений, преобразований и ряд стандартных функций, которые могут быть полезны в приложениях общего назначения.

Системные библиотеки содержат функции, относящиеся к проекту и используемому оборудованию. По сути, они имеют необходимые функции для управления сетевой связью (если есть), управления аналоговыми/цифровыми входами/выходами, памятью контроллера для управления любыми журналами и всеми особенностями, относящимися к среде UNI-PRO.

Чтобы помочь понять и прояснить методы использования, каждая библиотека снабжена примерным проектом. Пожалуйста, обратитесь к папке UNI-PRO \...\Samples\Libraries\ для просмотра этой информации.

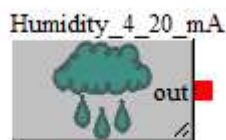
Для получения более подробной информации о программных свойствах библиотек обратитесь к руководству по программному обеспечению среды разработки.

Для получения подробной информации об оборудовании обратитесь к документации, относящейся к используемым приборам.

## 2 СИСТЕМНЫЕ БИБЛИОТЕКИ

### 2.1 Analogue Inputs

#### Humidity\_4\_20\_mA



Выходы	Тип	Границы	Описание
Value	CJ_ANALOG	Значение: 0..100 Ошибка: 0..2	Выдает значение относительной влажности.

#### Описание

Библиотека *Humidity\_4\_20\_mA* использует 4-20mA аналоговый вход и выдает аналоговое преобразование ток-влажность на выходе.

Имеется выход, выводящий значение между 0-100 (процентных точек), пропорционально току, измеренному датчиком.

Область *Ошибки* определена как:

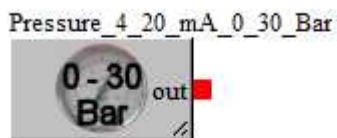
0. Нет ошибки
1. Короткое замыкание датчика
2. Повреждение или отсутствие датчика

#### Замечания

Шаблонный тип библиотеки - на картинке.

Выходное значение имеет точность в одну процентную единицу.

#### Pressure\_4\_20\_mA\_0\_30\_Bar



Выходы	Тип	Границы	Описание
Value	CJ_ANALOG	Значение: 0..30 Ошибка: 0..2	Выдает значение относительного давления.

#### Описание

Библиотека *Pressure\_4\_20\_mA\_0\_30\_Bar* использует 4-20mA аналоговый вход и выдает аналоговое преобразование ток-давление на выходе.

Имеется выход, выводящий значение между 0-30 Бар, пропорционально току, на датчике.

Область *Ошибки* определена как:

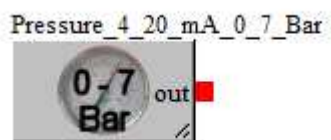
0. Нет ошибки
1. Короткое замыкание датчика
2. Повреждение или отсутствие датчика

#### Замечания

Шаблонный тип библиотеки - на картинке.

Выходное значение имеет точность в одну десятую (1/10) Бар.

#### Pressure\_4\_20\_mA\_0\_7\_Bar



Выходы	Тип	Границы	Описание
Value	CJ_ANALOG	Значение: 0..7 Ошибка: 0..2	Выдает значение относительного давления.

#### Описание

Библиотека *Pressure\_4\_20\_mA\_0\_7\_Bar* использует 4-20mA аналоговый вход и выдает аналоговое преобразование ток-давление на выходе.

Имеется выход, выводящий значение между 0-7 Бар, пропорционально току, на датчике.

Область *Ошибки* определена как:

0. Нет ошибки
1. Короткое замыкание датчика
2. Повреждение или отсутствие датчика

#### Замечания

Шаблонный тип библиотеки - на картинке.

Выходное значение имеет точность в одну десятую (1/10) Бар.

#### Pressure\_4\_20\_mA\_General

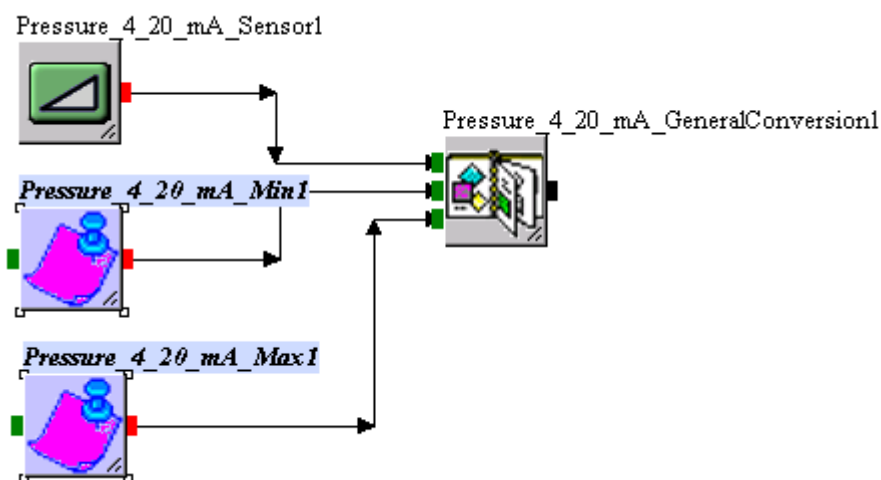


Выходы	Тип	Границы	Описание
Value	CJ_ANALOG	Значение: min..max Ошибка: 0..2	Выдает значение относительного давления.

### Описание

Библиотека *Pressure\_4\_20\_mA\_General* использует 4-20mA аналоговый вход и выдает аналоговое преобразование ток-давление на выходе.

Диапазон значений выхода устанавливается в пределах библиотеки. Для установки необходимого диапазона измените значение *value* показателей двух параметров, *Pressure\_4\_20\_mA\_Min1* и *Pressure\_4\_20\_mA\_Max1*.



### Замечания

Шаблонный тип библиотеки - в картинке.

Выходное значение имеет точность в одну десятую (1/10) Бар.

Чтобы изменить точность значения в одну десятую, установите показатель *точности представления данных* параметров в библиотеке.

### Temperature\_RESISTANCE\_NTC



Выход	Тип	Границы	Описание
out	CJ_ANALOG	-	Датчик «ТЕМПЕРАТУРА»

### Описание

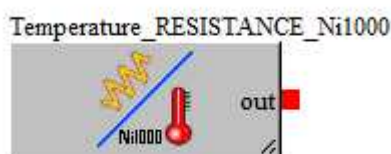
Библиотека позволяет преобразовать датчик «СОПРОТИВЛЕНИЕ» в датчик «ТЕМПЕРАТУРА». Стандартный датчик EVCO NTC определяется в соответствии со следующей диаграммой (-19,0°C / +42,0°C).

°C	Ом
-19.00	64400
-13.00	48500
-7.00	37000
-1.00	28400
5.00	22000
11.00	17200
18.00	13000
25.00	10000
32.00	7700
42.00	5400
49.00	4300

### Замечания

Шаблонный тип библиотеки - на картинке.

### Temperature\_RESISTANCE\_Ni1000



Выход	Тип	Границы	Описание
out	CJ_ANALOG	-	Преобразованный датчик «ТЕМПЕРАТУРА» Ni1000 DIN 43760

### Описание

Библиотека позволяет преобразовать датчик «СОПРОТИВЛЕНИЕ» в датчик «ТЕМПЕРАТУРА» типа Ni1000 DIN 43760.

Библиотека содержит сконфигурированный АНАЛОГОВЫЙ ВХОД, поэтому при перетаскивании этой библиотеки в проекте появится АНАЛОГОВЫЙ ВХОД, который будет объединен в инструменте Назначения связи программного обеспечения Uni-Pro.

**SelectUniversalAIIn**

<b>Вход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	in: 1..10	Устанавливает тип n-го аналогового входа.

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	out: 1..10	Выдает тип установки аналогового входа.

**Описание**

Библиотека *SelectUniversalAIIn* позволяет установить тип аналогового входа для n-го датчика контроллера. Диапазон допустимых значений выхода данной библиотеки 1-10, и каждое значение устанавливает определенный тип датчика. См. следующие соответствия:

- 1 : SENSOR\_PTC. Датчик PTC
- 2 : SENSOR\_NTC. Датчик NTC
- 3 : SENSOR\_0\_20mA. Датчик тока 0-20 mA
- 4 : SENSOR\_4\_20mA. Датчик тока 4-20 mA
- 5 : SENSOR\_0\_5V. Датчик напряжения 0-5 V
- 6 : SENSOR\_0\_10V. Датчик напряжения 0-10 V
- 7 : SENSOR\_PT1000. Датчик PT1000
- 8 : NTC 10K2. Зонд NTC 10K2
- 9 : NTC 10K3. Зонд NTC 10K3
- 10 : RESISTANCE. значение сопротивления, подключенного к входу (Ом / 10)

Чтобы установить значение по умолчанию, настройте свойство `value` параметра `Select_UniversalAIIn` в библиотеке.

**Замечания**

Шаблонный тип библиотеки - на картинке.

Обратитесь к документации по описанию прибора, чтобы установить максимальное количество универсальных входов контроллера.

После каждого изменения необходимо выключить и снова включить питание, чтобы изменения вступили в силу.

Необходимо, чтобы данные библиотеки были использованы только раз за проект.

**AI\_TimeoutError**

<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_BYTE	1..255	Проверка ошибки по времени задержки

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_BYTE	1..255	Проверка ошибки по времени задержки

**Описание**

Библиотека *AI\_TimeoutError* позволяет установить время задержки для обнаружения ошибок зонда. Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_ErrorProbeTimeout*, присутствующего в библиотеке.

**Замечания**

Шаблонный тип библиотеки - на картинке.

Очень важно, чтобы эти библиотеки использовались только один раз для каждого проекта.

## 2.2 Analogue Outputs

### SelectTypeAOn



Вход	Тип	Границы	Описание
In	CJ_BYTE	in: 0..6	Устанавливает тип аналогового выхода

Выход	Тип	Диапазон	Описание
Out	CJ_BYTE	out: 0..6	Возвращает тип аналогового выхода.

#### Описание

Библиотека *SelectTypeAOn* позволяет установить тип n-го аналогового выхода. Диапазон допустимых значений для этой библиотеки — от 0 до 6, и каждое значение определяет определенный тип вывода. Ассоциация чисел следующая:

- 0: FAN. Аналоговый выход фазорезки
- 1: 0-20mA. Токовый аналоговый выход 0-20 mA
- 2: 4-20mA. Токовый аналоговый выход 4-20 mA
- 3: 0-10V. Аналоговый выход напряжения 0-10 V
- 4: PWM. ШИМ аналоговый выход
- 5: BELIMO. Аналоговый выход для управления модулями Belimo/MPBus
- 6: EEV\_U. Аналоговый выход для управления однополярным ЭРВ

Чтобы установить значение по умолчанию, настройте свойство `value` параметра `Select_TypeAOn`, присутствующего в библиотеке.



#### Замечания

Шаблонный тип библиотеки - на картинке.

Обратитесь к документации по описанию прибора, чтобы установить максимальное количество универсальных входов контроллера.

После каждого изменения необходимо выключить и снова включить питание, чтобы изменения вступили в силу.

Необходимо, чтобы данные библиотеки были использованы только раз за проект.



**PWM\_FrequencyAOn**

<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>frequency</i>	CJ_WORD	10..2000	Устанавливает частоту (Гц) аналогового выхода, сконфигурированного как ШИМ.

**Описание**

Библиотека *PWM\_FrequencyAOn* позволяет установить тактовую частоту аналогового выхода, когда последний сконфигурирован как ШИМ.

Диапазон допустимых значений для этой библиотеки составляет 10 Гц... 2000 Гц; возможные значения за пределами этого диапазона будут отклонены.

**Замечания**

Посмотрите документацию по оборудованию, чтобы узнать максимум выходов, поддерживаемых контроллером. Библиотеки необходимо использовать один раз в проекте.

## 2.3 BACnet MSTP

### BACnetMSTP\_BaudRate



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_BYTE	0..3	Устанавливает скорость передачи

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_BYTE	0..3	Возвращает скорость передачи

#### **Описание**

Библиотека позволяет установить скорость передачи данных для связи по сети BACnet MSTP. Поддерживаются следующие скорости передачи:

- 0 : 9.6 Kbit
- 1 : 19.2 Kbit
- 2 : 38.4 Kbit
- 3 : 76.8 Kbit

Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_BACnetMSTP\_BaudRate* в библиотеке.

#### **Замечания**

Шаблонный тип библиотеки - на картинке.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**BACnetMSTP\_DeviceInstance**

BACnetMSTP\_DeviceInstance



Вход	Тип	Границы	Описание
in	CJ_LONG	0.. 4194303	Устанавливает идентификатор устройства

Выход	Тип	Границы	Описание
out	CJ_LONG	0.. 4194303	Возвращает идентификатор устройства

**Описание**

Библиотека позволяет установить однозначный идентификатор устройства в сети BACnet. Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_BACnetMSTP\_DeviceInstance* в библиотеке.

**Замечания**

Шаблонный тип библиотеки - на картинке.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**BACnetMSTP\_MacID**

BACnetMSTP\_MacID



Вход	Тип	Границы	Описание
in	CJ_BYTE	in: 1..127	Устанавливает адрес MacID

Выход	Тип	Границы	Описание
out	CJ_BYTE	Out: 1..127	Возвращает адрес MacID

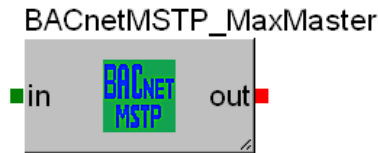
**Описание**

Библиотека позволяет установить физический адрес устройства RS485 в сети BACnet MSTP. Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_BACnetMSTP\_MacID* в библиотеке.

**Замечания**

Шаблонный тип библиотеки - на картинке.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**BACnetMSTP\_MaxMaster**



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	in: 1..127	Устанавливает максимальный адрес, на который проходит токен в сети.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	Out: 1..127	Возвращает максимальный адрес, по которому проходит токен в сети.

**Описание**

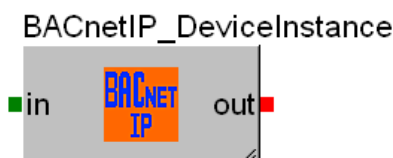
Библиотека позволяет установить максимальный адрес, на который проходит токен в связи BACnet, поиск нового устройства в сети.  
Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_BACnetMSTP\_MaxMaster* в библиотеке.

**Замечания**

Шаблонный тип библиотеки - на картинке.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

## 2.4 BACnet IP

### BACnetIP\_DeviceInstance



Вход	Тип	Границы	Описание
in	CJ_LONG	in: 0.. 4194303	Устанавливает идентификатор устройства

Выход	Тип	Границы	Описание
out	CJ_LONG	out: 0.. 4194303	Возвращает идентификатор устройства

#### Описание

Библиотека позволяет установить однозначный идентификатор устройства в сети BACnet. Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_BACnetIP\_DeviceInstance* в библиотеке.

#### Замечания

Библиотека носит шаблонный характер. Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

### BACnetIP\_PortNumber



Вход	Тип	Границы	Описание
in	CJ_WORD	47808 .. 47823 0xBAC0..0xBACF	Устанавливает номер IP-порта BACnet

Выход	Тип	Границы	Описание
out	CJ_WORD	47808 .. 47823 0xBAC0..0xBACF	Возвращает номер IP-порта BACnet.

**Описание**

Библиотека позволяет установить свойство «Номер порта», используемое контроллером в IP-сети ВАСnet.

Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_VASnetIP\_PortNumber* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**VASnetIP\_BBMD\_IPAddress**

<b>Вход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Устанавливает IP-адрес BBMD

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Возвращает IP-адрес BBMD

**Описание**

Библиотека позволяет установить IP-адрес ВАСnet Broadcast Management Device (BBMD) в сети ВАСnet.

Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_VASnetIP\_BBMDAddress* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**BACnetIP\_BBMD\_PortNumber**

<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	47808 .. 47823 0xBAC0..0xBACF	Устанавливает номер порта BBMD

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	47808 .. 47823 0xBAC0..0xBACF	Возвращает номер порта BBMD

**Описание**

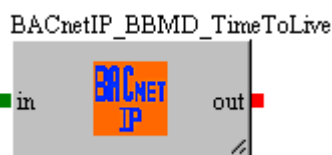
Библиотека позволяет установить свойство BBMD Port Number, на которое устройство может отправлять широковещательные сообщения в IP-сети BACnet.

Чтобы установить значение по умолчанию, настройте свойство value параметра *P\_BACnetIP\_BBMDPortNumber* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**BACnetIP\_BBMD\_TimeToLive**

<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	15 .. 65535	Устанавливает тайм-аут в секундах службы BBMD

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	15 .. 65535	Возвращает время ожидания в секундах службы BBMD.

**Описание**

Библиотека позволяет установить свойство `BBMD Time To Live`, время в секундах, в течение которого `BBMD` пересылает широковещательные сообщения.

Чтобы установить значение по умолчанию, настройте свойство `value` параметра `P_VASnetIP_BBMDTimeToLive` в библиотеке.

#### **Замечания**

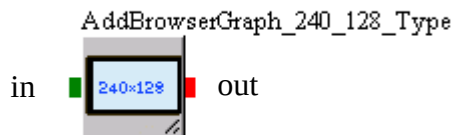
Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.



## 2.5 Browser

### AddBrowserGraph\_240\_128\_Type



Вход	Тип	Границы	Описание
In	CJ_WORD	1..240	Определяет ID страницы, с которой Вы хотите работать.

Выход	Тип	Границы	Описание
Out	CJ_WORD	0..65535	Возвращает необходимую информацию.

#### Описание

С помощью этой библиотеки можно добавить информацию на тип дисплея, необходимую для работы страницы с установленным ID. В действительности, в рамках проекта, могут быть страницы с одинаковыми ID, созданные для различных дисплеев.

С помощью библиотеки *AddBrowserGraph\_240\_128\_Type* можно выбрать для использования страницу с 240x128 пиксельной графикой.

#### Замечания

Пример использования данной библиотеки приведен ниже.

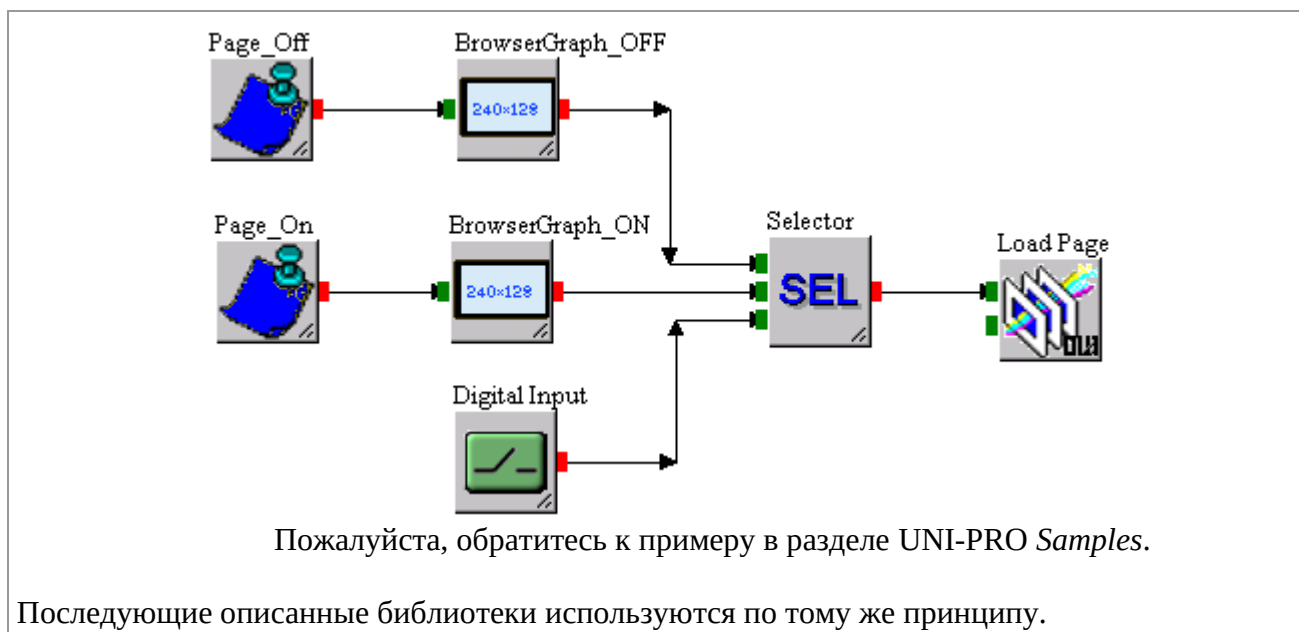
Настройте выбранные Цифровые Входы для просмотра страницы.

*Page\_On* содержит ID открытой страницы, в то время как *Page\_Off* содержит ID закрытой страницы.

Если Цифровой Вход = 1 select *Page\_On*.

Если Цифровой Вход = 0 select *Page\_Off*.

После получения сигнала триггер посылает команду *LoadPage* для загрузки страницы, выбранной с помощью *Selector*.



### AddBrowserColor\_320\_240\_Type



Вход	Тип	Границы	Описание
in	CJ_WORD	1..240	Указывает ID страницы, которую требуется обработать

Выход	Тип	Границы	Описание
out	CJ_WORD	0..65535	Возвращает информацию, для обработки

#### Описание

Через эту библиотеку можно добавить информацию о типе дисплея, необходимую для обработки страницы с идентификатором. Ведь внутри проекта могут быть страницы с одинаковым ID, созданные для разных дисплеев.

С помощью библиотеки *AddBrowserColor\_320\_240\_Type* можно выбрать графическую сенсорную цветную страницу размером 320 \* 240 пикселей и использовать ее в приложении.

**AddBrowserGraph\_240\_140\_Type**

AddBrowserGraph\_240\_140\_Type\_1



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	1..240	Определяет ID страницы, с которой Вы хотите работать.

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Возвращает информацию, для обработки

**Описание**

С помощью этой библиотеки можно добавить информацию на тип дисплея, необходимую для работы страницы с установленным ID. В действительности, в рамках проекта, могут быть страницы с одинаковыми ID, созданные для различных дисплеев.

С помощью библиотеки *AddBrowserGraph\_240\_140\_Type* можно выбрать графическую страницу размером 240\*140 пикселей и использовать ее в приложении.

**AddBrowserGraph\_128\_64\_Type**

AddBrowserGraph\_128\_64\_Type



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	1..240	Указывает ID страницы, которую требуется обработать

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Возвращает информацию, для обработки

**Описание**

Через эту библиотеку можно добавить информацию о типе дисплея, необходимую для обработки страницы с идентифицирующим идентификатором. Ведь внутри проекта могут быть страницы с одинаковым ID, созданные для разных дисплеев.

С помощью библиотеки *AddBrowserGraph\_128\_64\_Type* можно выбрать графическую страницу размером 128\*64 пикселей и использовать ее в приложении.

**AddBrowserGraph\_120\_32\_Type**

AddBrowserGraph\_120\_32\_Type\_1



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	1..240	Указывает ID страницы, которую требуется обработать

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Возвращает информацию для обработки

**Описание**

Через эту библиотеку можно добавить информацию о типе дисплея, необходимую для обработки страницы с идентификатором. Ведь внутри проекта могут быть страницы с одинаковым ID, созданные для разных дисплеев.

С помощью библиотеки *AddBrowserGraph\_120\_32\_Type* можно выбрать графическую страницу размером 120\*32 пикселя и использовать ее в приложении.

**AddBrowserAlpha\_4\_20\_Type**

AddBrowserAlpha\_4\_20\_Type



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	1..240	Указывает ID страницы, которую требуется обработать

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Возвращает информацию для обработки

**Описание**

Через эту библиотеку можно добавить информацию о типе дисплея, необходимую для обработки страницы с идентификатором. Ведь внутри проекта могут быть страницы с одинаковым ID, созданные для разных дисплеев.

С помощью библиотеки *AddBrowserAlpha\_4\_20\_Type* можно выбрать буквенно-цифровую страницу размером 4 \* 20 символов и использовать ее в приложении.

### AddBrowserAlpha\_4\_16\_Type



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	1..240	Указывает ID страницы, которую требуется обработать

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Возвращает информацию для обработки

#### *Описание*

Через эту библиотеку можно добавить информацию о типе дисплея, необходимую для обработки страницы с идентификатором. Ведь внутри проекта могут быть страницы с одинаковым ID, созданные для разных дисплеев.

С помощью библиотеки *AddBrowserAlpha\_4\_16\_Type* можно выбрать буквенно-цифровую страницу размером 4 \* 16 символов и использовать ее в приложении.

### AddBrowserSevenSeg\_1\_3D\_Type



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	1..240	Указывает ID страницы, которую требуется обработать

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Возвращает информацию для обработки

**Описание**

Через эту библиотеку можно добавить информацию о типе дисплея, необходимую для обработки страницы с идентификатором. Ведь внутри проекта могут быть страницы с одинаковым ID, созданные для разных дисплеев.

С помощью библиотеки *AddBrowserSevenSegAlpha\_1\_3D\_Type* можно выбрать 7 - сегментную трехзначную буквенно-цифровую страницу и использовать ее в приложении.

**AddBrowserSevenSeg\_1\_4D\_Type**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_WORD	1..240	Указывает ID страницы, которую требуется обработать

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_WORD	0..65535	Возвращает информацию для обработки

**Описание**

Через эту библиотеку можно добавить информацию о типе дисплея, необходимую для обработки страницы с идентификатором. Ведь внутри проекта могут быть страницы с одинаковым ID, созданные для разных дисплеев.

С помощью библиотеки *AddBrowserSevenSegAlpha\_1\_4D\_Type* можно выбрать 7 - сегментную четырехзначную буквенно-цифровую страницу и использовать ее в приложении.

## 2.6 1<sup>st</sup> CAN & 2<sup>nd</sup> CAN

Эти библиотеки позволяют настраивать и управлять протоколом связи CAN, используемым аппаратными средствами. Программное обеспечение UNI-PRO предусматривает использование до двух каналов CAN и, следовательно, делает доступными две библиотеки, 1stCAN и 2nd CAN, относящиеся к первому и второму каналам связи соответственно.

Следует отметить, что не все оборудование потенциально может использовать два канала связи CAN.

Пожалуйста, обратитесь к документации конкретного контроллера для получения дополнительной информации по этой теме.

В этом параграфе поясняется функциональность библиотек по отношению к первому каналу, характеристики второго канала связи такие же.

### CAN1st\_IsMaster



Вход	Тип	Границы	Описание
in	CJ_BIT	0-1	Устанавливает узел мастером сети

Выход	Тип	Границы	Описание
out	CJ_BIT	0-1	Возвращает статус мастера

#### Описание

Библиотека *CAN1st\_IsMaster* используется для определения того, является ли узел, на котором будет установлен проект, главным в сети.

Для установки стандартного значения, присвойте находящемуся в библиотеке показателю параметра *CAN1st\_IsMaster* значение *value*. Если параметр имеет значение '1', то узел будет главным узлом сети, если имеет значение '0', то узел будет являться подчиненным.

#### Замечания

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. После каждого изменения необходимо выключить и снова включить питание, чтобы изменения вступили в силу.

**CAN1st\_MyNode**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	1..127	Устанавливает текущий номер узла

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	1..127	Возвращает текущий номер узла

**Описание**

Библиотека *CAN1st\_MyNode* представляет собой физический адрес контроллера, на который загружена программа.

Чтобы выбрать значение по умолчанию, настройте свойство *value* параметра *CAN\_1st\_MyNode* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. После каждого изменения необходимо выключить и снова включить питание, чтобы изменения вступили в силу.



**CAN1st\_BaudRate**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	1..4	Устанавливает скорость передачи данных для первого канала связи CAN.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	1..4	Возвращает скорость передачи данных первого канала связи CAN.

**Описание**

Библиотека *CAN1st\_BaudRate* позволяет установить скорость передачи данных для связи по первому каналу CAN.

Поддерживаются следующие скорости передачи:

1. 20 Kbit
2. 50 Kbit
3. 125 Kbit
4. 500 Kbit

Установите значение по умолчанию, изменив свойство *value* параметра *CAN\_1st\_BaudRate*.

**Замечания**

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

После каждого изменения необходимо выключить и снова включить питание, чтобы изменения вступили в силу.

**CAN1st\_GetStatus**

Вход	Тип	Границы	Описание
out	CJ_SHORT	0..3	Возвращает статус канала связи

**Описание**

Библиотека *CAN1st\_GetStatus* возвращает статус первого канала CAN.

Допустимые выходные значения следующие:

- 0 : CJ\_CAN\_OK. Канал работает исправно.
- 1 : CJ\_CAN\_WARNING. Канал исправен, но требует внимания.
- 2 : CJ\_CAN\_ERROR. Канал не исправен и работает только на прием данных.
- 3 : CJ\_CAN\_BUS\_OFF. Канал имеет серьезную неисправность и отключены прием и передача данных.

**CAN1st\_GetNetworkStatus**

Вход	Тип	Границы	Описание
in	CJ_BYTE	1..32	Номер логического узла, статус которого необходимо проверить

Выход	Тип	Диапазон	Описание
out	CJ_BYTE	0..3	Возвращает статус запрошенного узла

**Описание**

Библиотека *CAN1st\_GetNetworkStatus* возвращает статус логического узла, запрошенного входом и подключенного к первому каналу CAN.

Допускаются следующие состояния выходов:

- 0 : CJ\_CAN\_OK. Узел работает исправно.
- 1 : CJ\_CAN\_WARNING. Узел исправен, но требует внимания.
- 2 : CJ\_CAN\_ERROR. Узел не исправен и работает только на прием данных.
- 3 : CJ\_CAN\_BUS\_OFF. Узел имеет серьезную неисправность и отключены прием и передача данных.

**Замечания**

Данная библиотека может быть использована для каждого из 32 потенциальных узлов сети, соединенных с первым каналом связи CAN.

**CAN1st\_NetworkNode\_n**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	0..127	Устанавливает адрес узла сети.
<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	0..127	Выдает адрес узла сети.

**Описание**

Библиотека *CAN1st\_NetworkNode\_n* представляет физический адрес n-го узла первого канала сети (n = 1...32).

Для установки стандартного значения, присвойте находящемуся в библиотеке показателю параметра *CAN1st\_NetworkNode\_n* значение *value*.

Адреса могут принимать значения между 1 и 127 включительно, и должны быть индивидуальными для каждого канала. Значение 0 определяет отсутствие узла.

**Замечания**

Библиотека носит шаблонный характер.

Необходимо, чтобы данная библиотека была использована только раз для каждого n-го узла.

**CAN1st\_NodeGuardTimeout**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	1..60	Устанавливает, как часто главный узел проверяет состояние подчиненных узлов.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	1..60	Возвращает, как часто главный узел проверяет состояние подчиненных узлов.

### **Описание**

Библиотека *CAN1st\_NodeGuardTimeout* используется для установки времени защиты протокола CAN. Для установки стандартного значения, присвойте находящемуся в библиотеке показателю параметра *CAN1st\_NodeGuardTimeout* значение *value*.

По истечении значения времени *value*, контроллер, если он является главным узлом, проверяет все подключенные к нему подчиненные узлы и их относительное состояние, используя значение *CAN1st\_NodeGuardTimeout*. Можно считывать состояние каждого узла.

Единица измерения значения параметра- секунды. Например, если установить значение 5, то все узлы сети будут запрашиваться главным узлом каждые 5 секунд.

### **Замечания**

Необходимо, чтобы данная библиотека была использована только раз за проект.

## **2<sup>nd</sup> CAN**

Библиотеки, относящиеся ко второму каналу связи те же что и относящиеся к первому, а именно:

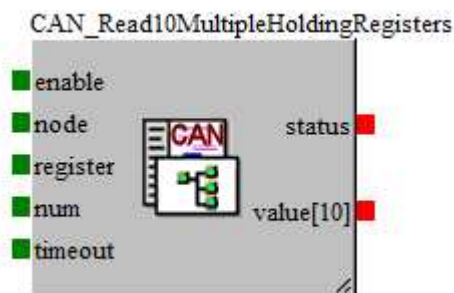
- CAN2nd\_GetStatus
- CAN2nd\_GetNetworkStatus
- CAN2nd\_MyNode
- CAN2nd\_IsMaster
- CAN2nd\_Networknode\_n
- CAN2nd\_BaudRate
- CAN2nd\_NodeGuardTimeout

Как уже было отмечено ранее, они выполняют те же функции, что и относящиеся к первому каналу CAN, следовательно, описываться далее не будут, а просто будут перечислены. Для методов использования, пожалуйста, обратитесь к похожим разделам, описанным ранее.

## 2.7 CAN Master

Эти библиотеки позволяют получить доступ через сеть CAN к регистрам, экспортированным по Modbus (через инструмент UNI-PRO Export Entities).

### CAN\_Read10MultipleHoldingRegisters



Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включить процесс чтения
<i>node</i>	CJ_BYTE	1-32	Логический адрес в сети CAN
<i>register</i>	CJ_WORD	1-65000	Начальный регистр Modbus
<i>num</i>	CJ_BYTE	1-10	Количество адресов для чтения
<i>timeout</i>	CJ_WORD	0-600	Время ожидания (мс)

Выход	Тип	Границы	Описание
<i>status</i>	CJ_BYTE	0-6	Возвращает статус чтения: 0 = Свободен 1 = Ожидание. Добавлен новый запрос 2 = В ходе выполнения. 3 = ОК. Чтение завершено правильно 4 = Ошибка. Чтение завершено со сбоем 5 = Тайм-аут. Чтение не завершено до тайм-аута. 6 = Отклонено. Чтение отклонено.
<i>value[10]</i>	CJ_WORD	0-65535	Прочитанные значения

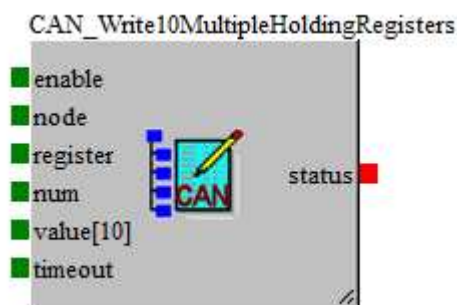
#### Описание

Библиотека *CAN\_Read10MultipleHoldingRegisters* позволяет считывать до 10 последовательных регистров хранения, экспортированных по Modbus, на узел сети CAN. Для запуска операции чтения необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен запрос на чтение *num* данных, присутствующих в *register* узла *node*. После добавления сообщение будет обработано, и ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ будет положительным, и значение будет в выводах *value[10]*, вместо этого, если вывод имеет значение 4 ответ был отрицательным, если выход имеет значение 5, ответ был отрицательным, потому что время ожидания истекло, если выход имеет значение 6, чтение для регистра недоступно. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи вход *enable* должен поддерживаться в состоянии 1.

### Замечания

Очень важно, чтобы эта библиотека использовалась только один раз в проекте.

### CAN\_Write10MultipleHoldingRegisters



Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включить процесс записи
<i>node</i>	CJ_BYTE	1-32	Логический адрес в сети CAN
<i>register</i>	CJ_WORD	1-65000	Начальный регистр Modbus
<i>num</i>	CJ_WORD	1-10	Количество адресов для записи
<i>value[10]</i>	CJ_WORD	0-65535	Значения для записи
<i>timeout</i>	CJ_WORD	0-600	Время ожидания (мс)

Выход	Тип	Границы	Описание
-------	-----	---------	----------

<i>status</i>	CJ_BYTE	0-6	Возвращает статус записи: 0 = Свободен 1 = Ожидание. Добавлен новый запрос 2 = В ходе выполнения. 3 = ОК. Запись завершена правильно 4 = Ошибка. Запись завершена со сбоем 5 = Тайм-аут. Запись не завершена до тайм-аута. 6 = Отклонено. Запись отклонена.
---------------	---------	-----	--

### Описание

Библиотека *CAN\_Write10MultipleHoldingRegisters* позволяет записывать до 10 последовательных регистров хранения, экспортированных по Modbus, на узел сети CAN. Для запуска операции записи необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен запрос на запись *num* данных, присутствующих в *register* узла *node*. После добавления сообщение будет обработано, и ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ будет положительным, вместо этого, если вывод имеет значение 4 ответ был отрицательным, если выход имеет значение 5, ответ был отрицательным, потому что время ожидания истекло, если выход имеет значение 6, запись для регистра недоступна. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи вход *enable* должен поддерживаться в состоянии 1.

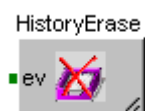
### Замечания

Библиотека используется в проекте только один раз.

## 2.8 History

Библиотеки, показанные ниже, должны использоваться для управления журналом истории (очистка, чтение и запись). Чтобы иметь возможность их использовать, необходимо включить использование журнала истории и определить его размеры в параметрах проекта. Более подробное описание этой темы см. в руководстве по программному обеспечению UNI-PRO.

### HistoryErase

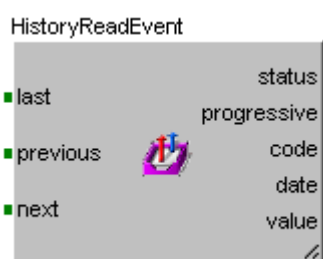


Вход	Tun	Границы	Описание
ev	CJ_BIT	0-1	Запрос на очистку записей журнала. Очистка запрашивается, когда значение входа меняется от 0 до 1.

#### Описание

Использование библиотеки *HistoryErase* позволяет сбросить журнал, удаляя все записи. Действие выполняется, когда вход *ev* изменяется от 0 до 1 (передний фронт).

### HistoryReadEvent



Вход	Tun	Границы	Описание
<i>last</i>	CJ_BIT	0-1	Запрос последней записи в журнале истории
<i>previous</i>	CJ_BIT	0-1	Запрос предпоследней записи в журнале истории
<i>next</i>	CJ_BIT	0-1	Запрос последующей записи в журнале истории



Выход	Тип	Границы	Описание
<i>status</i>	CJ_BYTE	1..4	Определяет состояние операции считывания.
<i>progressive</i>	CJ_LONG	-2147483648.. 2147483647	Порядковый номер события.
<i>code</i>	CJ_WORD	0..65535	Код события.
<i>date</i>	CJ_DATETIME	0.. 2147483647	Дата и время записи.
<i>value</i>	CJ_SHORT	-32768 .. 32767	Значение присоединенное к событию

### Описание

С помощью библиотеки *HistoryReadEvent* можно прочитать записи, записанные ранее в журнале событий с помощью библиотеки *HistoryWriteEvent*. Это асинхронная операция, поэтому необходимо воздействовать на входы, а затем проверять выход *value*, чтобы узнать статус операции. Используя вход *last*, можно запросить последнее событие, предыдущие события можно запросить с помощью *previous*, а последующие события можно запросить с помощью входа *next*.

После завершения чтения события (*value = HISTORY\_READ\_OK*) можно сохранить текущий номер, код, дату и значение для запрошенного события.

Допускаются следующие состояния выхода *status*:

- 1: *HISTORY\_READ\_OK*. Считывание успешно.
- 2: *HISTORY\_READ\_NONE*. Элементы не сохраняются.
- 3: *HISTORY\_READ\_DIM*. Ошибка 'переполнения' памяти.
- 4: *HISTORY\_READ\_FAIL*. Ошибка считывания.

### HistoryWriteEvent



Вход	Тип	Границы	Описание
<i>code</i>	CJ_WORD	0..65535	Код сохраняемого события
<i>event</i>	CJ_BIT	0-1	Запрос на сохранение события. По переднему фронту инициируется сохранение события.
<i>clock</i>	CJ_DATETIME	0..2147483647	Дата и время, когда произошло событие (обычно связаны с системными часами)
<i>value</i>	CJ_SHORT	-32768 .. 32767	Значение присоединенное к событию

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>status</i>	CJ_BYTE	0..2	Статус операции

### **Описание**

Использование библиотеки *HistoryWriteEvent* позволяет сохранять события в журнале. Присоедините код события с входом *input*, а часы системы с входом *clock* и поле *value* со значением, которое нужно присоединить к событию в записи. Когда произойдет изменение от '0' до '1', активируется вход *data* для сохранения процесса. Чтобы определить результат операции, проверьте значение выхода *status*.

Допускаются следующие состояния выхода:

0: *HISTORY\_WRITE\_OK*. Запись успешна.

1: *HISTORY\_QUEUE\_FULL*. Запись остановлена, так как очередь запросов переполнена.

2: *HISTORY\_WRONG\_DATETIME*. Ошибка записи из-за некорректных данных.

## 2.9 Data Log

Чтобы иметь возможность использовать эти библиотеки, необходимо включить использование журнала данных в параметрах проекта. Более подробное описание этой темы см. в руководстве по программному обеспечению UNI-PRO.

### DataLog\_RecordingTime



Вход	Tun	Границы	Описание
time	CJ_WORD	10-65535	Время записи (в секундах)

Выход	Tun	Границы	Описание
status	CJ_BIT	0..1	Возвращает статус операции: 0 = Обновление успешно. 1 = Обновление неудачно.

#### Описание

Библиотека *DataLog\_RecordingTime* позволяет установить частоту записи (в секундах) элементов TIME, выбранных в таблице журнала. Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_LOG\_RecordingTime* в библиотеке.

#### Замечания

Библиотека носит шаблонный характер.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

### DataLog\_StartStop



Вход	Tun	Границы	Описание
startSop	CJ_BIT	0-1	Запускает (1) или останавливает (0) выбранную запись данных.

#### Описание

Через библиотеку *DataLog\_StartStop* можно запускать или останавливать запись объектов в Журнал. Для включения ведения журнала установите *startSop=1*, а для остановки *startStop=0*.

### Замечания

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

### DataLog\_Erase



Вход	Tun	Границы	Описание
<i>evErase</i>	CJ_BIT	0-1	Запрашивает очистку журнала данных. Стирание выполняется, когда вход <i>evErase</i> изменяется с 0 на 1.

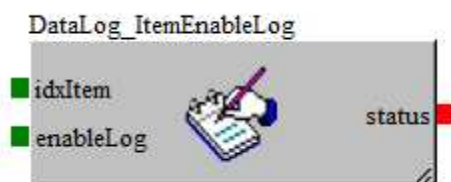
### Описание

С помощью библиотеки *DataLog\_Erase* вы можете удалить все записи журнала данных. Операция выполняется, когда вход *evErase* изменяется с 0 на 1 (передний фронт).

### Замечания

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

### DataLog\_ItemEnableLog



Вход	Tun	Границы	Описание
<i>idxItem</i>	CJ_WORD	1-64	Индекс элемента журнала .
<i>enableLog</i>	CJ_BIT	0-1	Включение элемента: 0 = Отключен 1 = Включен

Выход	Tun	Границы	Описание
-------	-----	---------	----------

<i>status</i>	CJ_WORD	0-4	Возвращает статус операции: 0 = Операция выполнена. 1 = Ошибка. Индекс за границами. 2 = Ошибка. Объект не найден 3 = Обновление не запрошено 4 = Общая ошибка.
---------------	---------	-----	--

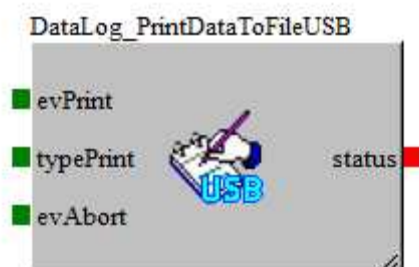
### Описание

Библиотека *DataLog\_ItemEnableLog* используется для включения записи элемента, представленного *idxItem*. Журнал данных записывает только значения включенных объектов (*enableLog=1*). Для проверки правильности работы контролировать состояние вывода *status*. Включение/выключение элемента должно происходить при статусе журнала STOP. Внесенные изменения включения/отключения элемента, если они выполняются в текущем журнале данных, влекут за собой форматирование всего журнала: очистка данных и мгновенная потеря связи с локальными и удаленными дисплеями.

### Замечания

Чтобы узнать индексы объектов, экспортируемых в журнал, обратитесь к таблице экспорта, созданной с помощью инструмента журнала данных в UNI-PRO.

### DataLog\_PrintDataToFileUSB



Вход	Тип	Границы	Описание
<i>evPrint</i>	CJ_BIT	0-1	Запрос экспорта CSV-файла через USB-порт. Экспорт выполняется, когда вход <i>evPrint</i> изменяется с 0 на 1.
<i>typePrint</i>	CJ_BIT	0-1	Тип экспорта: 0 = С приостановкой 1 = Не приостанавливать
<i>evAbort</i>	CJ_BIT	0-1	Прервать запрос на экспорт CSV-файла (эта процедура имеет смысл, только если <i>typePrint=1</i> ). Прерывание выполняется, когда вход <i>evAbort</i> изменяется с 0 на 1.

Выход	Тип	Границы	Описание
-------	-----	---------	----------

<i>status</i>	CJ_WORD	0-1	Возврат статуса проработки: 0 = Выполнено/экспорт не выполняется. 1 = Экспорт выполняется.
---------------	---------	-----	--

### Описание

Библиотека *DataLog\_PrintDataToFileUSB* используется для запроса экспорта файлов (с именем *ИмяПроекта\_YYYY\_MM\_DD.csv*), содержащих все записи журнала данных через USB-порт контроллера. Операция выполняется при изменении входа *evPrint* с 0 на 1 (передний фронт).

Вы можете выполнить эту процедуру в двух режимах (вход *typePrint*):

- 0 = С приостановкой. Выполнение алгоритмов приложения останавливается, а выходы/входы остаются замороженными в текущем состоянии до конца операции.
- 1 = Без приостановки. Работа алгоритмов приложения продолжается без перерыва. Этот режим медленнее предыдущего. Окончание операции может произойти путем проверки выхода *status*.

Через вход *evAbort* можно остановить обработку для экспорта записей и экспортировать файлы через USB-порт до точки прерывания (операция выполняется при изменении входа с 0 на 1).

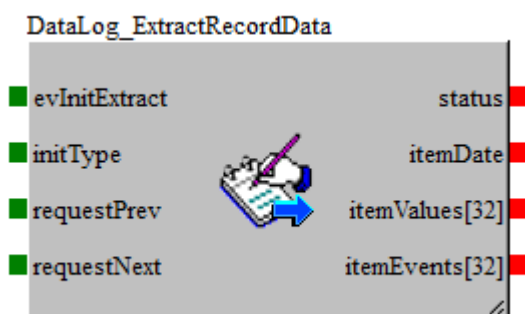
Через выход *status* можно следить за ходом операции.

Вход *evAbort* и выход *status* можно использовать только в режиме Без приостановки (*typePrint=1*).

### Замечания

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. Во время экспорта желательно остановить Журнал данных, чтобы не растягивать время обработки (используя библиотеку *DataLog\_StartStop*).

### DataLog\_ExtractRecordData



Вход	Тип	Границы	Описание
<i>evInitExtract</i>	CJ_BIT	0-1	Запрос на получение первой записи журнала данных. Это событие делает доступным первый элемент Журнала для вывода. Запрос является выполненным, если вход

			<i>evInitExtract</i> изменяется с 0 на 1.
<i>initType</i>	CJ_BYTE	1-2	Тип инициализации: 1 = Позиционирование на первую запись журнала (самую старую). 2 = Позиционирование на последнюю запись журнала (самую новую).
<i>requestPrev</i>	CJ_BIT	0-1	Запросить предыдущую запись.
<i>requestNext</i>	CJ_BIT	0-1	Запросить следующую запись.

Выход	Tun	Границы	Описание
<i>status</i>	CJ_BYTE	0-2	Статус операции: 0 = Нет обновления 1 = Выходные данные обновлены 2 = Достижение конца лога (вверх) 3 = Достижение конца лога (вниз) Если ошибка 2 или 3 на выходе, это недопустимые данные.
<i>itemDate</i>	CJ_DATETIME		Дата регистрации текущей записи.
<i>itemValues[32]</i>	CJ_LONG	-2147483648.. 2147483647	Значения элементов с типом Time в текущей записи.
<i>itemEvents[32]</i>	CJ_BIT	0-1	Значения элементов с типом Event в текущей записи.

### Описание

Библиотека *DataLog\_ExtractRecordData* позволяет читать записи, хранящиеся в журнале данных. Сначала нужно установить указатель на начальную запись, используя вход *evInitExtract*, это делается при изменении входа с 0 на 1 (передний фронт). Можно выбрать, где начать чтение, используя ввод *initType*:

- =1: Выберите первую зарегистрированную запись.
- =2: Выберите последнюю зарегистрированную запись.

Операция инициализации, а также подготовка структур заполняет выходные данные данными выбранной записи.

Используя входы *requestNext* и *requestPrev*, вы можете просмотреть все записи журнала. Для каждого запроса выходы обновляются данными выбранной записи:

- *itemDate* : запись даты записи данных.
- *itemValues[32]*: значение элементов Time в текущей записи (TIME/ALL).
- *itemEvents[32]*: значение элементов Event в текущей записи (EVENT/ALL).
- *status*: определяет статус запроса
  - 0: Нет запроса/ожидание данных
  - 1: Вывод данных (*itemDate*, *itemValues*, *itemEvents*) обновлен. Предупреждение о том, что это состояние остается в силе в течение одного цикла программы.
  - 2: После того, как действие *requestNext* превысило последний элемент журнала, выходные данные недействительны. Последние доступные данные соответствуют предыдущему запросу.

- 3: После того, как действие *requestPrev* превысило первый элемент журнала, выходные данные недействительны. Последние доступные данные соответствуют предыдущему запросу.

### **Интерпретация выходных данных *itemValues*[32] и *itemEvents*[32]**

Каждый индекс двух массивов является элементом таблицы журнала экспорта, созданной с помощью UNI-PRO, индексы расположены последовательно и упорядочены соответственно с таблицей журнала данных. Выходные данные представляют собой массивы из 32 элементов, поскольку максимальное количество сущностей типа TIME и EVENT равно 32 для каждого. Значения отключенных элементов в журнале и отсутствующих элементов всегда равны 0.

Пример.

Таблица экспорта журнала данных с 6 объектами TIME и 5 объектами EVENT.

Idx	Linked Entity	Item Description	MU	Type	Variation	Log
1	Var1	DescriptionVar1	°C	TIME	ALL	X
2	Var2	DescriptionVar2	°C	TIME	ALL	X
3	Event1	DescriptionEvent1		EVENT	0->1	X
4	Event2	DescriptionEvent2		EVENT	0->1	X
5	VarEvent1	DescriptionVarEvent1	°C	ALL	ALL	X
6	Event3	DescriptionEvent3		EVENT	ALL	X
7	Var3	DescriptionVar3	°C	TIME	ALL	X
8	VarEvent2	DescriptionVarEvent2	°C	ALL	1->0	X
9	Var5	DescriptionVar5	°C	TIME	ALL	X

Entity Log Options		Data Log Info	
Description	<input type="text"/>	Recording Time	10 (Sec.)
Measurement Unit	<input type="text"/>	Number TIME Entities	6
Type	TIME	Number EVENT Entities	5
Variation	ALL	Max Dim. Entity	2 Bytes
Enable Logging	<input type="checkbox"/>	Time (Enabled)	1month 1day
<input type="button" value="Set"/>		Time (All)	27days 21h

Для каждого запроса *requestNext* или *requestPrev*, выводимые *itemValues*[32] и *itemEvents*[32], необходимо интерпретировать как:

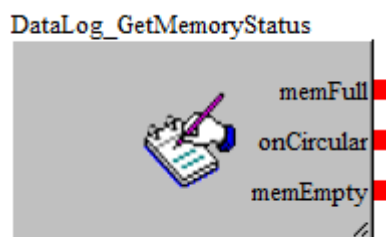
TIME		EVENT	
<i>itemValues</i> [0]	Значение of Var1	<i>itemEvents</i> [0]	Событие of Event1
<i>itemValues</i> [1]	Значение of Var2	<i>itemEvents</i> [1]	Событие of Event2
<i>itemValues</i> [2]	Значение of VarEvent1	<i>itemEvents</i> [2]	Событие of VarEvent1
<i>itemValues</i> [3]	Значение of Var3	<i>itemEvents</i> [3]	Событие of Event3
<i>itemValues</i> [4]	Значение of VarEvent2	<i>itemEvents</i> [4]	Событие of VarEvent2
<i>itemValues</i> [5]	Значение of Var5(Ноль так как отключен)	<i>itemEvents</i> [5]	0
<i>itemValues</i> [6]	0	<i>itemEvents</i> [6]	0
...		....	

Все значения в последовательных индексах, которые не используются, равны 0.

### **Замечания**

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.



**DataLog\_GetMemoryStatus**

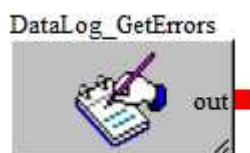
<b>Output</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>memFull</i>	CJ_BIT	0-1	Когда верно (1), память заполнена.
<i>onCircular</i>	CJ_BIT	0-1	Когда верно (1), память исчерпана и журнал сохраняется циклически.
<i>memEmpty</i>	CJ_BIT	0-1	Когда верно (1) память свободна, никакие записи не ведутся.

**Описание**

Библиотека *DataLog\_GetMemoryStatus* возвращает состояние памяти журнала данных, определяет состояния полной памяти, пустой памяти и циклической записи. При переходе записи в циклический режим (*onCircular*=1) старые данные в памяти перезаписываются.

**Замечания**

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**DataLog\_GetErrors**

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_WORD	0-6	Возвращает статус работы Журнала данных: 0 = Журнал данных не настроен. 1 = Журнал данных ОК, в статусе STOP. 2 = Журнал данных ОК, в статусе RUN. 3 = Ошибка, память несовместима. 4 = Ошибка RTC. 5 = Ошибка, повреждена память. 6 = Ошибка, недостаточно места.

**Описание**

Библиотека *DataLog\_GetErrors* возвращает состояние работы журнала данных.

### Замечания

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

## DataLog\_GetStatus



Выход	Тип	Границы	Описание
out	CJ_BYTE	0-3	Возвращает статус работы Журнала данных: 0=Бездействие/форматирование завершено успешно. 1=Выполняется форматирование/удаление. 2=Отформатировать/удалить по запросу. 3=Инициализация и тест памяти журнала данных.

### Описание

Библиотека *DataLog\_GetStatus* возвращает рабочее состояние журнала данных на этапах стирания и форматирования.

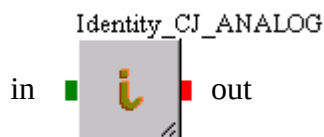
### Замечания

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

## 2.10 Identity

Эти библиотеки возвращают данные, вводимые через них. Например, они полезны внутри подсхемы, если желательно несколько раз использовать входные данные, экспортированные из самой подсхемы. Один пример проиллюстрирован для понимания.

### Identity\_CJ\_ANALOG



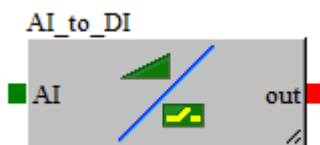
Выход *out* представляет вход *in* без каких-либо изменений. В этой библиотеке два «терминала» принимают структуру типа CJ\_ANALOG.

Другие библиотеки, принадлежащие группе *Identity*, следующие:

- Identity\_CY\_BIT
- Identity\_CY\_BTN
- Identity\_CY\_BUZZ
- Identity\_CY\_BYTE
- Identity\_CY\_CMD
- Identity\_CY\_DATE
- Identity\_CY\_DATETIME
- Identity\_CY\_DWORD
- Identity\_CY\_LED
- Identity\_CY\_LONG
- Identity\_CY\_SHORT
- Identity\_CY\_S\_BYTE
- Identity\_CY\_TIME
- Identity\_CY\_WORD

## 2.11 IO Utils

### AI\_to\_DI



Вход	Tun	Границы	Описание
AI	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Вход датчика

Выход	Tun	Границы	Описание
out	CJ_BIT	0..1	Значение цифрового входа

#### Описание

Преобразует аналоговый вход в цифровой. Вывод AI должен быть напрямую подключен к реальному аналоговому входу (выход сущности ANALOGIN), а не к переменной CJ\_ANALOG, полученной в результате разработок.

Аналоговый вход должен быть сконфигурирован как NTC.

- DI = 1 для контактов с сопротивлением  $\leq 600$  ohms
- DI = 0 для контактов с сопротивлением  $\geq 60k$  ohms

### AI\_to\_RESISTANCE

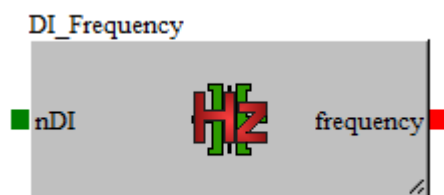


Вход	Tun	Границы	Описание
AI	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Вход датчика CJ_ANALOG «RESISTANCE»

Выход	Tun	Границы	Описание
out	CJ_LONG	0.. 2147483648	Значение сопротивления в Ом

#### Описание

Преобразует аналоговый входной сигнал типа RESISTANCE в значение в Ом. Вывод AI должен быть напрямую подключен к реальному аналоговому входу (RESISTANCE), а не к переменной CJ\_ANALOG, полученной в результате разработок.

**DI\_Frequency**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>nDI</i>	CJ_BYTE	0..255	Номер цифрового входа, который считывает частоту

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>frequency</i>	CJ_WORD	0.. 65535	Возвращает значение частоты (Гц)

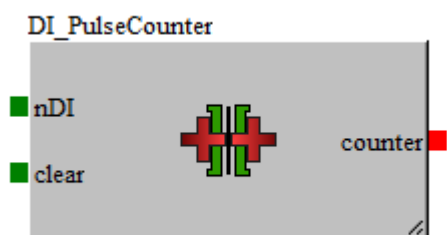
**Описание**

Возвращает частоту в Гц, обнаруженную цифровым входом, выбранным с помощью *nDI*. Значение *nDI* должно быть номером контролируемого цифрового входа.

**Замечания**

Цифровые входы контроллеров, частота считывания которых меняется. Чтобы использовать правильные входы, обратитесь к руководству по аппаратному обеспечению.

- Семейство Micro/Kilo: DI3 и DI4
- Семейство Node Kilo: DI1 и DI2

**DI\_PulseCounter**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>nDI</i>	CJ_BYTE	0..255	Номер цифрового входа, используемого счетчиком импульсов
<i>clear</i>	CJ_BIT	0-1	Очистка счетчика

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
--------------	------------	----------------	-----------------

<i>counter</i>	CJ_DWORD	0.. 4294967295	Счетчик импульсов, обнаруженных цифровым входом
----------------	----------	----------------	---

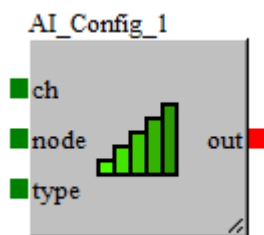
**Описание**

Возвращает число импульсов, обнаруженных цифровым входом, выбранным с помощью *nDI*. Значение *nDI* должно быть номером контролируемого цифрового входа. Для сброса счетчика установите входное значение *clear*=1.

**Замечания**

Цифровые входы, используемые в качестве счетчика импульсов контроллеров, различаются, чтобы использовать правильные входы, обратитесь к аппаратному руководству.

- Семейство Micro/Kilo: DI3 и DI4
- Семейство Node Kilo: DI1 и DI2

**AI\_Config**

Вход	Тип	Границы	Описание
<i>ch</i>	CJ_BYTE	1..N	Физический канал аналогового входа контроллера или расширения
<i>node</i>	CJ_BYTE	0..32	Логический адрес в сети CANbus контроллера (0) или расширения (1..32). Адрес соответствует установленному в Hardware Expert. Если расширений нет, по умолчанию это 0 (контроллер)
<i>type</i>	CJ_BYTE	1..N	Тип настраиваемого датчика. Только для контроллера. Если не подключен, датчик не настроен.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_ANALOG	-	Значение, считываемое аналоговым входом <i>ch</i> контроллера или расширением с адресом <i>node</i> . Если вы прочитали значение CJ_ERR_REF в поле Ошибка, конфигурация датчика, запрошенная со значением <i>type</i> , не была успешной.

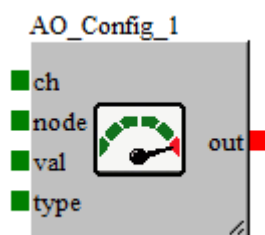
**Описание**

Позволяет считывать и настраивать аналоговый вход, указывая физический канал и логический узел в сети CANbus.

Диапазон допустимых значений на входе *type* для этой библиотеки составляет от 1 до 10, и каждое значение идентифицирует конкретный тип датчика. Ассоциация следующая:

- 1 : SENSOR\_PTC. PTC датчик
- 2 : SENSOR\_NTC. NTC датчик
- 3 : SENSOR\_0\_20mA. 0-20 mA токовый датчик
- 4 : SENSOR\_4\_20mA. 4-20 mA токовый датчик
- 5 : SENSOR\_0\_5V. 0-5 V вольтовый датчик
- 6 : SENSOR\_0\_10V. 0-10 V вольтовый датчик
- 7 : SENSOR\_PT1000. PT1000 датчик
- 8 : NTC 10K2. NTC 10K2 сенсор
- 9 : NTC 10K3. NTC 10K3 сенсор
- 10 : RESISTANCE. значение сопротивления, подключенного к входу (Ом / 10)

Для обеспечения корректного функционирования библиотеки все AI-сущности, с которыми используются эти библиотеки, должны быть объявлены в проекте и их не нужно связывать.

**AO\_Config**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>ch</i>	CJ_BYTE	1..N	Физический канал аналогового входа контроллера или расширения
<i>node</i>	CJ_BYTE	0..32	Логический адрес в сети CANbus контроллера (0) или расширения (1..32). Адрес соответствует установленному в Hardware Expert. Если расширений нет, по умолчанию это 0 (контроллер)
<i>val</i>	CJ_WORD	0..100.00	Значение для аналогового выхода.
<i>type</i>	CJ_BYTE	0..N	Тип настраиваемого выхода. Только для контроллера. Если не подключен, выход не настроен.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
--------------	------------	----------------	-----------------

<i>out</i>	CJ_WORD	0..100.00	Значение, подаваемое на аналоговый выход <i>ch</i> контроллера или расширения с адресом <i>node</i> . Если вы прочитали значение 0.01, конфигурация выхода со значением <i>type</i> не была успешной.
------------	---------	-----------	---

### Описание

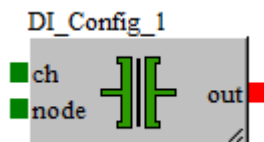
Позволяет управлять и настраивать аналоговый выход, указывая физический канал и логический узел в сети CANbus.

Диапазон допустимых значений на входе *type* для этой библиотеки от 0 до 6, каждое значение идентифицирует определенный тип выходных данных. Ассоциация следующая:

- 0: FAN. Выход обрыва фазы.
- 1: 0-20mA. 0-20 mA токовый выход
- 2: 4-20mA. 4-20 mA токовый выход
- 3: 0-10V. 0-10 V вольтовый выход
- 4: PWM. ШИМ выход
- 5: BELIMO. Выход управления Belimo/MPBus модулями
- 6: EEV\_U. Выход управления однополярными клапанами

Для корректного функционирования библиотеки все сущности АО, в которых используются эти библиотеки, должны быть объявлены в проекте (необходимо, чтобы они были подключены к сущности типа VAR, независимо от значения, так как оно будет перезаписано).

### DI\_Config



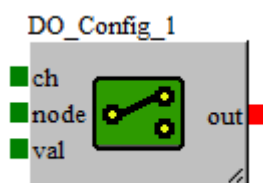
Вход	Тип	Границы	Описание
<i>ch</i>	CJ_BYTE	1..N	Физический канал цифрового входа контроллера или расширения
<i>node</i>	CJ_BYTE	0..32	Логический адрес в сети CANbus контроллера (0) или расширения (1..32). Адрес соответствует установленному в Hardware Expert. Если расширений нет, по умолчанию это 0 (контроллер)

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0..1	Значение, считанное цифровым входом <i>ch</i> контроллера или расширением с адресом <i>node</i> .



**Описание**

Позволяет прочитать цифровой вход через физический канал и логический узел в сети CANbus. Для корректного функционирования библиотеки все *DI*-сущности, с которыми используются эти библиотеки, должны быть объявлены в проекте и их не нужно связывать.

**DO\_Config**

Вход	Тип	Границы	Описание
<i>ch</i>	CJ_BYTE	1..N	Физический канал цифрового выхода контроллера или расширения
<i>node</i>	CJ_BYTE	0..32	Логический адрес в сети CANbus контроллера (0) или расширения (1..32). Адрес соответствует установленному в Hardware Expert. Если расширений нет, по умолчанию это 0 (контроллер)
<i>val</i>	CJ_BIT	0..1	Значение для цифрового выхода.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0..1	Значение, предоставляемое цифровым выходом <i>ch</i> контроллера или расширением с адресом <i>node</i> .

**Описание**

Позволяет управлять цифровым выходом, указывая физический канал и логический узел в сети CANbus.

Для корректного функционирования библиотеки все сущности *DO*, с которыми используются эти библиотеки, должны быть объявлены в проекте (необходимо, чтобы они были подключены к сущности типа VAR, независимо от значения, так как оно будет перезаписано).

## 2.12 Ethernet TCPIP

### Ethernet\_IP\_Address - Ethernet\_Gateway - Ethernet\_SubnetMask



Вход	Tun	Границы	Описание
in	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	IP-адрес, ШЛЮЗ и МАСКА ПОДСЕТИ для устройства

Выход	Tun	Границы	Описание
out	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	IP-адрес, ШЛЮЗ и МАСКА ПОДСЕТИ для устройства

#### Описание

Библиотеки позволяют установить IP-адрес контроллера, шлюза и маску подсети. Чтобы установить значение по умолчанию, настройте свойство *value* параметров в библиотеках.

#### Замечания

Библиотеки носят шаблонный характер. Очень важно, чтобы эти библиотеки использовались только один раз для каждого проекта. Библиотеки имеют смысл, только если для соответствующего порта выбран протокол Ethernet.

## 2.13 WebServer TCP

### WebServer\_PortNum



Вход	Tun	Границы	Описание
in	CJ_WORD	0..65535	Номер порта Web-сервера

Выход	Tun	Границы	Описание
out	CJ_WORD	0..65535	Номер порта Web-сервера

**Описание**

Библиотека позволяет установить номер порта для связи с Web-сервером по протоколу TCP-IP. Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_WebServer\_PortNum* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер. Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. Библиотека имеет смысл, только если на соответствующем порту выбран протокол Ethernet и включен Web-сервер.

**WebServer\_PasswordToWrite**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in[12]</i>	CJ_CHAR		Задаёт символы строки пароля (макс. 12 символов)

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out[12]</i>	CJ_CHAR		Возвращает символы строки пароля (макс. 12 символов)

**Описание**

Библиотека позволяет установить пароль для изменения статуса, экспортируемого на веб-страницу Web-сервера через TCP-IP.

Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_WebServer\_PasswordToWrite* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер. Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. Библиотека имеет смысл, только если на соответствующем порту выбран протокол Ethernet и включен Web-сервер.

**WebServer\_PasswordLevel1 .. WebServer\_PasswordLevel3**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in[12]</i>	CJ_CHAR		Задаёт символы строки пароля (макс. 12 символов)

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out[12]</i>	CJ_CHAR		Возвращает символы строки пароля (макс. 12 символов)

**Описание**

Библиотеки позволяют установить пароль для доступа к Web-странице, относящейся к уровню защиты Web-сервера по TCP-IP.

Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_WebServer\_PasswordLevel* в библиотеках.

**Замечания**

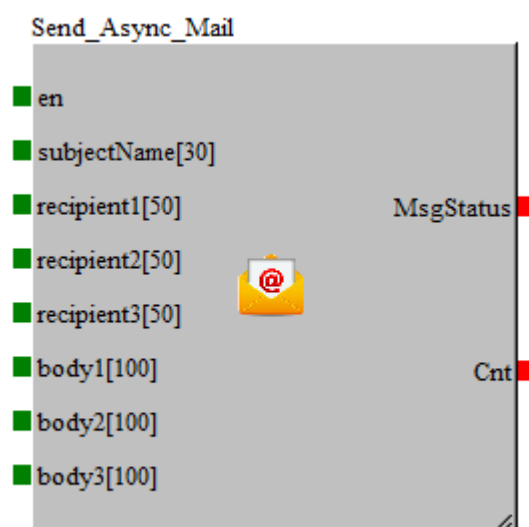
Библиотеки носят шаблонный характер.

Очень важно, чтобы эти библиотеки использовались только один раз для каждого проекта.

Библиотеки имеют смысл, только если на соответствующем порту выбран протокол Ethernet и включен Web-сервер.

## 2.14 Messages

### Send\_Async\_Mail



Вход	Тип	Границы	Описание
<i>en</i>	CJ_BIT	0..1	Включить передачу
<i>subjectName[30]</i>	CJ_CHAR		Тема письма (макс. 30 символов)
<i>recipient1[50]</i>	CJ_CHAR		Первый получатель (макс. 50 символов)
<i>recipient2[50]</i>	CJ_CHAR		Второй получатель (макс. 50 символов)
<i>recipient3[50]</i>	CJ_CHAR		Третий получатель (макс. 50 символов)
<i>body1[100]</i>	CJ_CHAR		Первая часть тела письма
<i>body2[100]</i>	CJ_CHAR		Вторая часть тела письма
<i>body3[100]</i>	CJ_CHAR		Третья часть тела письма

Выход	Тип	Границы	Описание
<i>MsgStatus</i>	CJ_BYTE	0..6	Статус передачи: 0 = Свободно 1 = Отправка (от контроллера на сервер) 2 = Доставка (от сервера к получателям) 3 = ОК (Сообщения дошли до получателей) 4 = Ошибка 5 = Прервано 6 = Ожидание (используемый канал уникален, если есть более одного последовательного запроса, следующий ставится в очередь)

<i>Cnt</i>	CJ_WORD	0..65535	Счетчик с момента начала передачи сообщения до тех пор, пока сервер не вернет OK или ERROR. Сообщение прибывающее через 300 секунд прерывает запрос по умолчанию.
------------	---------	----------	---

### Описание

Библиотека *Send\_Async\_Mail* позволяет отправлять электронную почту. Вход *en* должен быть установлен на 1, чтобы разрешить отправку и до тех пор, пока она не завершится успешно (OK) или с ошибкой (ERROR). Если он установлен в 0, то состояние перейдет в ABORT. По крайней мере, один из трех получателей (*recipientsN* входы) не должен быть нулевым, никаких проверок достоверности, если серверу не удастся выполнить переадресацию одному из получателей, повторите попытку для всех получателей. Каждое тело письма (входы *bodyN*) состоит из 100 символов, 99 из которых свободны, при этом в первом обязательно должен быть указан тип тела, а именно:

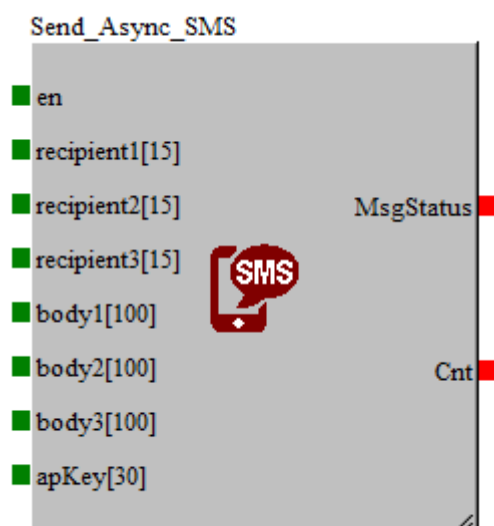
- 0 = HTML
- 1 = Text UTF8(\*)
- 2 = Table

(\*) В случае символа, превышающего 127 ASCII, необходимо кодировать с помощью нотации UTF8. Например. "2°C" надо писать "2\ **xc2**\ **xb0C**"

### Замечания

Библиотека имеет смысл, только если на соответствующем порту выбран протокол Ethernet и включена опция Mail/SMS.

### Send\_Async\_SMS



Вход	Тип	Границы	Описание
<i>en</i>	CJ_BIT	0..1	Включить передачу

<i>recipient1[15]</i>	CJ_CHAR		Номер первого получателя (начало с '+' за которым следует международный префикс)
<i>recipient2[15]</i>	CJ_CHAR		Номер второго получателя (начало с '+' за которым следует международный префикс)
<i>recipient3[15]</i>	CJ_CHAR		Номер третьего получателя (начало с '+' за которым следует международный префикс)
<i>body1[100]</i>	CJ_CHAR		Первая часть тела SMS
<i>body2[100]</i>	CJ_CHAR		Вторая часть тела SMS
<i>body3[100]</i>	CJ_CHAR		Третья часть тела SMS
<i>apKey[30]</i>	CJ_CHAR		API - ключ для отправки сообщений

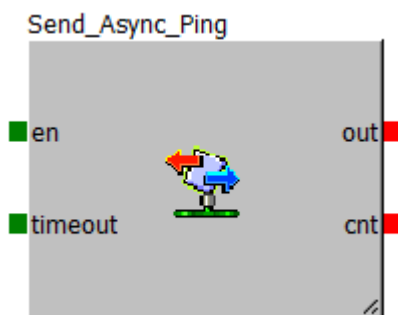
<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>MsgStatus</i>	CJ_BYTE	0..6	Статус передачи: 0 = Свободно 1 = Отправка (от контроллера на сервер) 2 = Доставка (от сервера к получателям) 3 = ОК (Сообщения дошли до получателей) 4 = Ошибка 5 = Прервано 6 = Ожидание (используемый канал уникален, если есть более одного последовательного запроса, следующий ставится в очередь)
<i>Cnt</i>	CJ_WORD	0..65535	Счетчик с момента начала передачи сообщения до тех пор, пока сервер не вернет ОК или ERROR. Сообщение прибывающее через 300 секунд прерывает запрос по умолчанию.

**Описание**

Библиотека *Send\_Async\_SMS* позволяет отправлять SMS. Вход *en* должен быть установлен на 1, чтобы разрешить отправку и до тех пор, пока она не завершится успешно (ОК) или с ошибкой (ERROR). Если он установлен в 0, то состояние перейдет в ABORT. По крайней мере, один из трех получателей (*recipientsN* входы) не должен быть нулевым, никаких проверок достоверности, если серверу не удастся выполнить переадресацию одному из получателей, повторите попытку для всех получателей.

**Замечания**

Библиотека имеет смысл, только если на соответствующем порту выбран протокол Ethernet и включена опция Mail/SMS.

**Send\_Async\_Ping**

Вход	Тип	Границы	Описание
<i>en</i>	CJ_BIT	0..1	Разрешить выполнение
<i>timeout</i>	CJ_BYTE	0..255	Таймаут выполнения (с)

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..5	Статус передачи: 0 = Свободно 1 = Отправка/Повтор 2 = Ожидание 3 = ОК (Отправка успешно завершена) 4 = Ошибка 5 = Прервано
<i>cnt</i>	CJ_WORD	0..255	Счетчик с момента начала отправки пакета до тех пор, пока выход «out» не достигнет состояния ОК или Ошибка.

**Описание**

Библиотека *Send\_Async\_Ping* позволяет отправить пакет в сеть для проверки его статуса. Вход *en* должен быть установлен на 1, чтобы разрешить отправку и до тех пор, пока она не завершится успешно (ОК) или с ошибкой (Ошибка).

Если значение выхода *cnt* превышает установленное на *timeout*, выход *out* переходит в состояние Ошибка.

**Замечания**

Библиотека имеет смысл, только если на соответствующем порту выбран протокол Ethernet и включена опция Mail/SMS.



## 2.15 Modbus

Эти библиотеки позволяют настраивать стандартный протокол MODBUS и управлять им. Программное обеспечение UNI-PRO предусматривает возможность назначения этого протокола одному или обоим доступным интерфейсам UART на контроллере. Библиотеки UART1 будут обозначаться префиксом Modbus1, библиотеки UART2 — префиксом Modbus2. Пожалуйста, обратитесь к документации по оборудованию контроллера, чтобы проверить совместимость с этим протоколом.

### Modbus1\_Address – Modbus2\_Address



Вход	Тип	Границы	Описание
in	CJ_BYTE	1..247	Устанавливает физический адрес контроллера в сети.

Выход	Тип	Границы	Описание
out	CJ_BYTE	1..247	Возвращает физический адрес контроллера в сети.

#### Описание

Библиотека *ModbusAddress* представляет физический адрес контроллера в сети MODBUS. Чтобы установить значение по умолчанию, настройте свойство *value* параметра *ModBus\_Address* в библиотеке.

#### Замечания

Библиотека носит шаблонный характер. Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. Библиотека имеет смысл только для режима Modbus Slave.

### Modbus1\_Baud – Modbus2\_Baud



Вход	Тип	Границы	Описание
in	CJ_BYTE	0..9	Устанавливает скорость передачи

Выход	Тип	Границы	Описание
-------	-----	---------	----------

<i>out</i>	CJ_BYTE	0..9	Возвращает скорость передачи
------------	---------	------	------------------------------

**Описание**

Библиотека *ModbusBaud* позволяет настроить скорость передачи данных.

Поддерживаются следующие скорости передачи:

- 0: 1200 Kbit
- 1: 2400 Kbit
- 2: 4800 Kbit
- 3: 9600 Kbit
- 4: 19200 Kbit
- 5: 28800 Kbit
- 6: 38400 Kbit
- 7: 57600 Kbit
- 8: 76800 Kbit
- 9: 115200 Kbit

Значение по умолчанию задается через *value* параметра *ModBus\_Baud* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

Библиотеку можно использовать для режимов Modbus Master и Modbus Slave.

**Modbus1\_Parity – Modbus2\_Parity**



Вход	Тип	Границы	Описание
<i>in</i>	CJ_BYTE	0..2	Устанавливает четность для передачи данных

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..2	Возвращает четность для передачи данных

**Описание**

Библиотека *ModbusParity* позволяет установить четность для передачи данных. Четность — это код, используемый для проверки целостности данных, бывает следующих типов:

- 0. NONE. Нет четности
- 1. ODD. Нечетный
- 2. EVEN. Четный

Значение по умолчанию задается через *value* параметра *ModBus\_Parity* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.  
 Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.  
 Библиотеку можно использовать для режимов Modbus Master и Modbus Slave.

**Modbus1\_StopBit – Modbus2\_StopBit**



Вход	Tun	Границы	Описание
in	CJ_BIT	0-1	Устанавливает тип «стопового бита»

Выход	Tun	Границы	Описание
out	CJ_BIT	0-1	Возвращает тип «стопового бита»

**Описание**

Библиотека *ModbusStopBit* позволяет установить тип «стопового бита» для передачи данных. Этот метод используется, чтобы позволить передающему/приемному устройству обнаружить конец кадра связи. Ассоциация значений следующая:

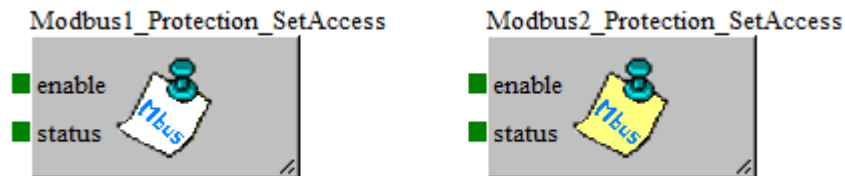
- 0 = 1 Стоповый бит.
- 1 = 2 Стоповых бита.

Значение по умолчанию задается через *value* параметра *ModBus\_StopBit* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.  
 Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.  
 Библиотеку можно использовать для режимов Modbus Master и Modbus Slave.

**Modbus1\_Protection\_SetAccess – Modbus2\_Protection\_SetAccess**



Вход	Tun	Границы	Описание
enable	CJ_BIT	0-1	Включить модификацию доступа для чтения

<i>status</i>	CJ_BIT	0-1	Статус доступа для чтения: 0=Не доступен 1=Доступен
---------------	--------	-----	---

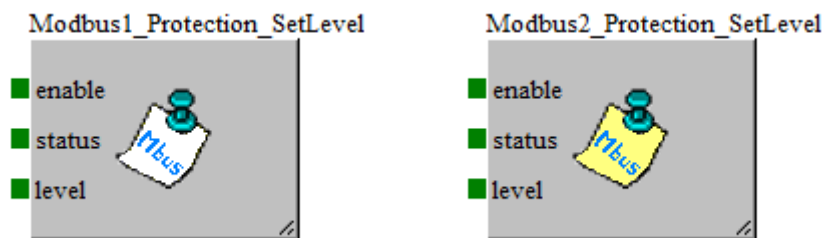
**Описание**

Библиотека *Modbus\_Protection\_SetAccess* позволяет установить защиту чтения для соответствующего UART Modbus; модификация выполняется с *enable=1*, в противном случае, если *enable=0*, никаких изменений не производится.

**Замечания**

Библиотеку можно использовать для Modbus Slave.

**Modbus1\_Protection\_SetLevel – Modbus2\_Protection\_SetLevel**



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>enable</i>	CJ_BIT	0-1	Включить модификацию уровня
<i>status</i>	CJ_BIT	0-1	Блокировка/разблокировка уровня: 0=Уровень заблокирован 1=Уровень разблокирован
<i>level</i>	CJ_BYTE	1-5	Уровень блокировки/разблокировки

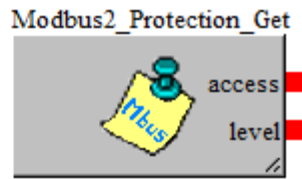
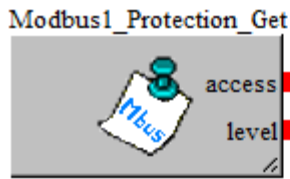
**Описание**

Библиотека *Modbus\_Protection\_SetLevel* позволяет установить блокировку/разблокировку (защиту от записи) уровня, передаваемого в качестве входных данных на соответствующем UART Modbus, модификация выполняется с *enable=1*, при *enable=0*, изменений нет.

**Замечания**

Библиотеку можно использовать для Modbus Slave.

### Modbus1\_Protection\_Get – Modbus2\_Protection\_Get



Выход	Tun	Границы	Описание
<i>access</i>	CJ_BIT	0-1	Статус защиты доступа на чтение: 0=Не доступен 1=Доступен
<i>level</i>	CJ_BYTE	0-5	Верхний разблокированный уровень

#### Описание

Библиотека *Modbus\_Protection\_Get* возвращает состояние защиты Modbus.

Выход *access* принимает следующие значения:

- 0= Защита доступа для чтения не активна.
- 1= Защита доступа для чтения активна.

Выход *level* возвращает разблокированный уровень (управление блокировкой уровня иерархическое, затем разблокируются предыдущие уровни).

#### Замечание

Библиотеку можно использовать для Modbus Slave.

### Modbus1\_Protection\_Timeout – Modbus2\_Protection\_Timeout



Вход	Tun	Границы	Описание
<i>in</i>	CJ_WORD	0-65535	Время ожидания уровня разблокировки (сек)

Выход	Tun	Границы	Описание
<i>in</i>	CJ_WORD	0-65535	Время ожидания уровня разблокировки (сек)

#### Описание

Библиотека *ModbusTCP\_Protection\_Timeout* позволяет установить продолжительность разблокировки уровней, счетчик пересчитывается при каждом доступе. Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_Modbus\_ProtectionTimeout* в библиотеке. Если значение равно 0, уровни всегда разблокированы. Для включения защиты необходимо использовать эту библиотеку в проекте, введя в качестве входных данных значение, отличное от 0.

### Замечания

Библиотека носит шаблонный характер. Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. Библиотеку можно использовать для Modbus Slave.

### ModbusMaster\_Timeout



Вход	Тип	Границы	Описание
<i>in</i>	CJ_BYTE	1..240	Устанавливает время ожидания ответа во время связи с ведомым устройством.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	1..240	Возвращает время ожидания ответа во время связи с ведомым устройством.

### Описание

Библиотека *ModbusMasterTimeout* представляет максимальное время, в течение которого ведомое устройство может ответить ведущему. По истечении этого тайм-аута, если от ведомого устройства нет ответа, ведущее устройство устанавливает ошибку связи и может начать другой процесс связи. Чтобы установить значение по умолчанию, настройте свойство *value* параметра *ModBusMaster\_Timeout* в библиотеке. Значение указывается в сотнях миллисекунд (например, значение 10 означает 1 секунду).

### Замечания

Библиотека носит шаблонный характер. Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. Библиотека имеет значение только для Modbus Master.

**ModbusMaster\_Interframe**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	0..240	Устанавливает межкадровое время в 100 мс.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	0..240	Возвращает межкадровое время в 100 мс.

**Описание**

Библиотека *ModbusMaster\_Interframe* позволяет установить минимальное время между пакетом данных и последующим в сообщении Modbus.

Чтобы установить значение по умолчанию, работайте со свойствами *value* параметра *P\_ModbusMaster\_Interframe* в библиотеке.

Значение находится в 100 x миллисекундах (это означает, что 10 равно 1 секунде).

**Замечания**

Библиотека представляет собой библиотеку шаблонов.

Эту библиотеку необходимо использовать один раз в проекте.

**ModbusMaster\_GetNetworkStatus**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>node</i>	CJ_BYTE	0..247	Логический узел для проверки.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
--------------	------------	----------------	-----------------

<i>out</i>	CJ_S_BYTE	-1..2	<p>Возвращает статус связи Modbus Master с <i>node</i> (логическим узлом сети).</p> <p>-1 = Устройство недоступно                      0 = Связь ОК                      1 = Устройство без связи                      2 = Устройство не выходит на связь</p> <ul style="list-style-type: none"> <li>- В начале, если ведомое устройство никогда не опрашивалось</li> <li>- Если было 0 (ОК) и прошло не менее одной минуты после последней связи с мастером</li> <li>- Если было 1 (NoLink) и остается в состоянии ошибки по истечении минуты</li> </ul>
------------	-----------	-------	---

**Описание**

Библиотека *ModbusMaster\_GetNetworkStatus* позволяет проверить состояние связи Modbus Master с логическим узлом, установленным входом *node*.  
 Библиотека применима только к логическим узлам устройств, сконфигурированных через инструмент ModbusMaster в проекте UNI-PRO.

**Замечания**

Эту библиотеку можно использовать в проекте несколько раз.  
 Библиотеку можно использовать только для Modbus Master.

**ModbusMaster\_UART\_Polarization**



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BIT	0-1	<p>Задаёт поляризацию порта ModbusMaster:</p> <p>0: UART Без поляризации                      1: UART С поляризацией</p>

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	<p>Возвращает поляризацию порта ModbusMaster.</p>

**Описание**



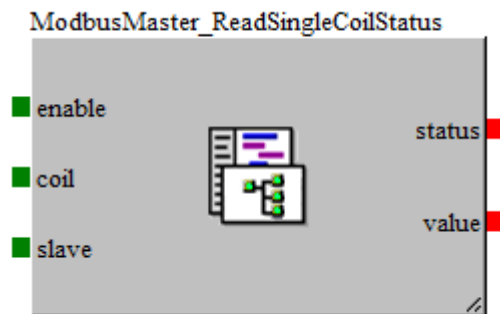
*ModbusMaster\_UART\_Polarization* позволяет установить поляризацию коммуникационного порта для протокола ModbusMaster.

Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_ModbusMaster\_UART\_Polarization* в библиотеке.

### Замечания

Библиотека носит шаблонный характер.  
Эту библиотеку необходимо использовать один раз в проекте.

### ModbusMaster\_ReadSingleCoilStatus



Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>coil</i>	CJ_WORD	1-65000	Идентификатор флага, к которому вы хотите получить доступ
<i>slave</i>	CJ_BYTE	0-247	Адрес подчиненного узла. 0 = Широковещательный режим.

Выход	Тип	Границы	Описание
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_BIT	0-1	Прочитанное значение

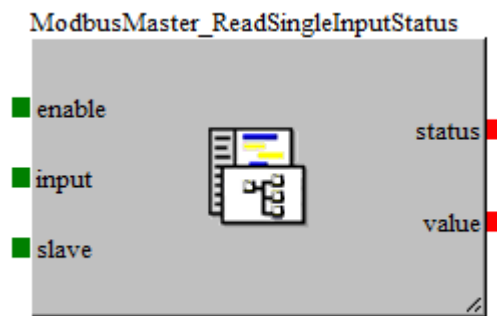
### Описание

Библиотека *ModbusMaster\_ReadSingleCoilStatus* позволяет считывать значение из Регистра флагов подчиненного сетевого узла.  
 Эта библиотека реализует отправку сообщения Modbus, реализующего код функции 0x01. Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.  
 После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

**Замечания**

Эту библиотеку можно использовать в проекте несколько раз.  
 Библиотеку можно использовать только для Modbus Master.

**ModbusMaster\_ReadSingleInputStatus**



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>input</i>	CJ_WORD	1-65000	Идентификатор входа, к которому вы хотите получить доступ
<i>slave</i>	CJ_BYTE	0-247	Адрес подчиненного узла. 0 = Широковещательный режим.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
--------------	------------	----------------	-----------------

<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_BIT	0-1	Прочитанное значение

### Описание

Библиотека *ModbusMaster\_ReadSingleInputStatus* позволяет считывать значение состояния входа подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus, реализующего код функции 0x02.

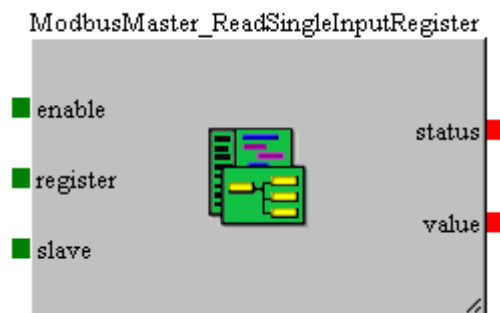
Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

### Замечания

Эту библиотеку можно использовать в проекте несколько раз.  
Библиотеку можно использовать только для Modbus Master.

### ModbusMaster\_ReadSingleInputRegister



Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>register</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ

<i>slave</i>	CJ_BYTE	0-247	Адрес подчиненного узла: 0 = Широковещательный режим.
--------------	---------	-------	--

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_WORD	0-65000	Прочитанное значение

### Описание

Библиотека *ModbusMaster\_ReadSingleInputRegister* позволяет читать значение из регистра входов подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus, реализующего код функции 0x04.

Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

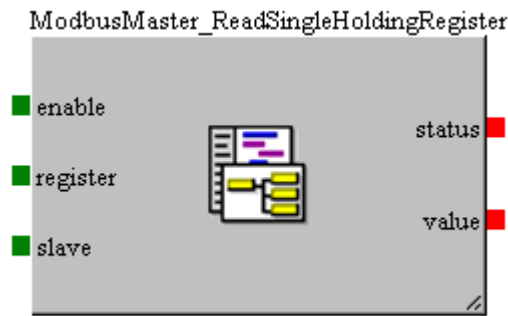
### Замечания

Эту библиотеку можно использовать в проекте несколько раз.  
Библиотеку можно использовать только для Modbus Master.

С помощью библиотек можно также последовательно читать больше регистров:

- *ModbusMaster\_Read10MultipleInputRegisters*
- *ModbusMaster\_Read100MultipleInputRegisters*

Они позволяют читать соответственно до 10 и 100 регистров хранения подряд, выбирая на входе *num* количество регистров для чтения.

**ModbusMaster\_ReadSingleHoldingRegister**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>register</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_BYTE	0-247	Адрес подчиненного узла. 0 = Широковещательный режим.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_WORD	0-65000	Прочитанное значение

**Описание**

Библиотека *ModbusMaster\_ReadSingleHoldingRegister* позволяет считывать значение из регистра хранения подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus, реализующего код функции 0x03.

Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

**Заявления**

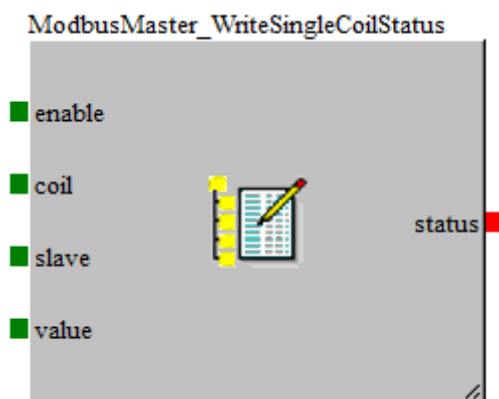
Эту библиотеку можно использовать в проекте несколько раз.

Библиотеку можно использовать только для Modbus Master.

Также возможно прочитать больше регистров последовательно через библиотеки:

- *ModbusMaster\_Read10MultipleHoldingRegisters*
- *ModbusMaster\_Read100MultipleHoldingRegisters*

Они позволяют читать соответственно до 10 и 100 регистров хранения подряд, выбирая на входе *num* количество регистров для чтения.

**ModbusMaster\_WriteSingleCoilStatus**

Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включает процесс записи кадра ведущий/ведомый
<i>coil</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_BYTE	0-247	Адрес подчиненного узла. 0 = Широковещательный режим.
<i>value</i>	CJ_BIT	0-1	Значение, которое необходимо записать

Выход	Тип	Границы	Описание
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster

**Описание**

Библиотека *ModbusMaster\_WriteSingleCoilStatus* позволяет записывать значение в Регистр флагов подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus, реализующего код функции 0x05.

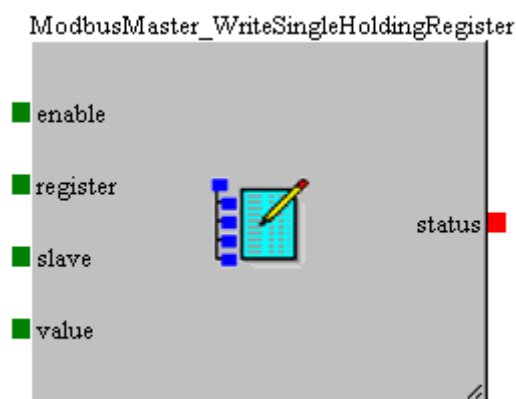
Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на запись данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

**Замечания**

Эту библиотеку можно использовать в проекте несколько раз.

Библиотеку можно использовать только для Modbus Master.

**ModbusMaster\_WriteSingleHoldingRegister**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>enable</i>	CJ_BIT	0-1	Включает процесс записи кадра ведущий/ведомый
<i>register</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_BYTE	0-247	Адрес подчиненного узла. 0 = Широковещательный режим.
<i>value</i>	CJ_WORD	0-65000	Значение, которое необходимо записать

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
--------------	------------	----------------	-----------------

<i>status</i>	CJ_BYTE	0-5	<p>Возвращает состояние связи ведущий/ведомый:</p> <p>0 = свободен для соединения</p> <p>1 = добавлен запрос на соединение</p> <p>2 = соединение в процессе</p> <p>3 = соединение корректно завершилось</p> <p>4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут)</p> <p>5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster</p>
---------------	---------	-----	---

### Описание

Библиотека *ModbusMaster\_WriteSingleHoldingRegister* позволяет записывать значение в регистр хранения подчиненного узла сети. Эта библиотека реализует отправку сообщения Modbus, реализующего код функции 0x06.

Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на запись данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

### Замечания

Эту библиотеку можно использовать в проекте несколько раз.

Библиотеку можно использовать только для Modbus Master.

С помощью библиотек можно также последовательно записать больше регистров:

- *ModbusMaster\_Write10MultipleHoldingRegisters*
- *ModbusMaster\_Write100MultipleHoldingRegisters*

Они позволяют записывать соответственно до 10 и 100 регистров хранения подряд, выбирая на входе *num* количество регистров для чтения. Эти библиотеки реализуют отправку сообщения Modbus, реализующего код функции 0x10.



## 2.16 Modbus TCP

Эти библиотеки позволяют настраивать стандартный протокол MODBUS и управлять им через TCP. Пожалуйста, обратитесь к документации по оборудованию контроллера, чтобы проверить совместимость с этим протоколом.

### ModbusTCP\_Protection\_SetAccess



Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включить модификацию доступности чтения
<i>status</i>	CJ_BIT	0-1	Устанавливает доступность чтения: 0=Не доступно 1=Доступно

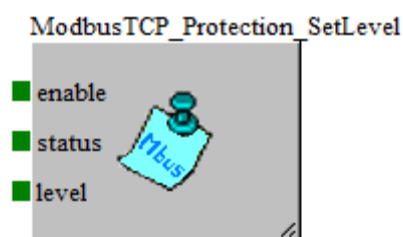
#### Описание

Библиотека *ModbusTCP\_Protection\_SetAccess* позволяет установить защиту от чтения для соответствующего UART Modbus, модификация выполняется с *enable*=1, в противном случае, если *enable*=0, никаких изменений не производится.

#### Замечания

Библиотеку можно использовать для Modbus Master и Modbus Slave.

### ModbusTCP\_Protection\_SetLevel



Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включить модификацию уровня
<i>status</i>	CJ_BIT	0-1	Задаёт блок/разблокировку уровня: 0=Уровень заблокирован 1=Уровень разблокирован
<i>level</i>	CJ_BYTE	1-5	Уровень для блокировки/разблокировки

**Описание**

Библиотека *ModbusTCP\_Protection\_SetLevel* позволяет установить блокировку/разблокировку (защиту от записи) уровня, передаваемого в качестве входных данных на соответствующем UART Modbus, модификация выполняется с *enable=1*, иначе, если *enable=0*, изменений нет.

**Замечания**

Библиотеку можно использовать для Modbus Master и Modbus Slave.

**ModbusTCP\_Protection\_Get**



Выход	Тип	Границы	Описание
<i>access</i>	CJ_BIT	0-1	Статус защиты доступа на чтение: 0=Недоступно 1=Доступно
<i>level</i>	CJ_BYTE	0-5	Макс. разблокированный уровень

**Описание**

Библиотека *ModbusTCP\_Protection\_Get* возвращает состояние защиты Modbus.

Выход *access* принимает следующие значения:

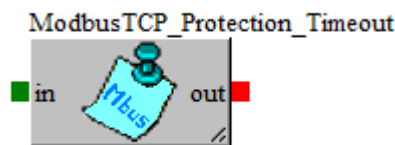
- 0= Защита доступа чтения не активна.
- 1= Защита доступа чтения активна.

Выход *level* возвращает разблокированный уровень (управление блокировкой уровня иерархическое, затем разблокируются предыдущие уровни).

**Замечания**

Библиотеку можно использовать для Modbus Master и Modbus Slave.

**ModbusTCP\_Protection\_Timeout**



Вход	Тип	Границы	Описание
<i>in</i>	CJ_WORD	0-65535	Таймаут уровня разблокировки (сек)

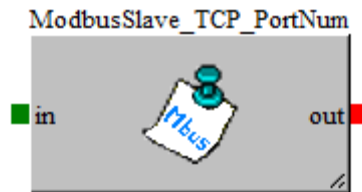
Выход	Tun	Границы	Описание
in	CJ_WORD	0-65535	Таймаут уровня блокировки (сек)

**Описание**

Библиотека *ModbusTCP\_Protection\_Timeout* позволяет установить продолжительность разблокировки уровней, счетчик пересчитывается при каждом доступе. Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_Modbus\_ProtectionTimeout* в библиотеке. Если значение равно 0, уровни всегда разблокированы.

**Замечания**

Библиотека носит шаблонный характер.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.  
Библиотеку можно использовать для Modbus Master и Modbus Slave.

**ModbusSlave\_TCP\_PortNum**

Вход	Tun	Границы	Описание
in	CJ_WORD	0..65535	Номер порта связи Modbus Slave через TCP.

Выход	Tun	Границы	Описание
out	CJ_WORD	0..65535	Номер порта связи Modbus Slave через TCP.

**Описание**

Библиотека позволяет установить номер порта для связи с протоколом Modbus Slave через TCP-IP.

Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_MbSlaveTCPIP\_PortNum* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.  
Библиотека имеет смысл только в том случае, если на соответствующем порту выбран протокол Ethernet и включен протокол Modbus Slave через TCP-IP.

**ModbusMaster\_TCP\_PortNum**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_WORD	0..65535	Номер порта связи Modbus master

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Modbus master номер порта связи

**Описание**

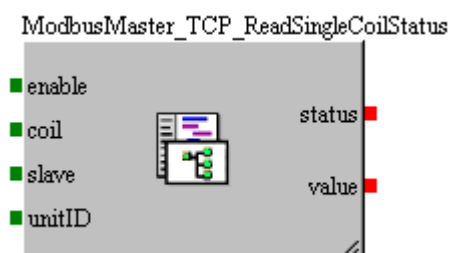
Библиотека позволяет установить номер порта для связи с протоколом Modbus Master через TCP-IP.

Чтобы установить значение по умолчанию, настройте свойство *value* параметра *P\_MbMasterTCPIP\_PortNum* в библиотеке.

**Замечания**

Библиотека носит шаблонный характер.

Библиотека имеет смысл только в том случае, если на соответствующем порту выбран протокол Ethernet и включен протокол Modbus Master over TCP-IP.

**ModbusMaster\_TCP\_ReadSingleCoilStatus**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>coil</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Адрес подчиненного узла

<i>unitID</i>	CJ_BYTE	1-247	Удаленный адрес ведомого устройства в сети RS485, доступной через шлюз (Примечание: этот идентификатор 255 обычно используется для адресации самого шлюза)
---------------	---------	-------	--

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_BIT	0-1	Прочитанное значение

#### **Описание**

Библиотека *ModbusMaster\_TCP\_ReadSingleCoilStatus* позволяет читать значения из Регистров флагов подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus TCP, которое реализует код функции 0x01.

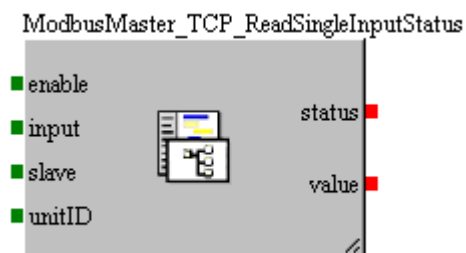
Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

#### **Замечания**

Эту библиотеку можно использовать в проекте несколько раз.

Библиотека может использоваться только для Modbus Master через TCP в контроллерах с включенным соединением Ethernet.

**ModbusMaster\_TCP\_ReadSingleInputStatus**

<b>Вход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>inpu</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Адрес подчиненного узла
<i>unitID</i>	CJ_BYTE	1-247	Удаленный адрес ведомого устройства в сети RS485, доступной через шлюз (Примечание: этот идентификатор 255 обычно используется для адресации самого шлюза)

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_BIT	0-1	Прочитанное значение

**Описание**

Библиотека *ModbusMaster\_TCP\_ReadSingleInputStatus* позволяет считывать значение состояния Регистра входа подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus TCP, которое реализует код функции 0x02.

Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.

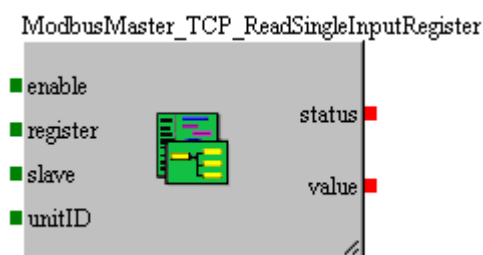
После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

### Замечания

Эту библиотеку можно использовать в проекте несколько раз.

Библиотека может использоваться только для Modbus Master через TCP в контроллерах с включенным соединением Ethernet.

### ModbusMaster\_TCP\_ReadSingleInputRegister



Вход	Tun	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>register</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Адрес подчиненного узла
<i>unitID</i>	CJ_BYTE	1-247	Удаленный адрес ведомого устройства в сети RS485, доступной через шлюз (Примечание: этот идентификатор 255 обычно используется для адресации самого шлюза)

Выход	Tun	Границы	Описание
-------	-----	---------	----------

<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_WORD	0-65000	Прочитанное значение

### Описание

Библиотека *ModbusMaster\_TCP\_ReadSingleInputRegister* позволяет читать значение из Регистра входов подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus TCP, которое реализует код функции 0x04.

Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

### Замечания

Эту библиотеку можно использовать в проекте несколько раз.

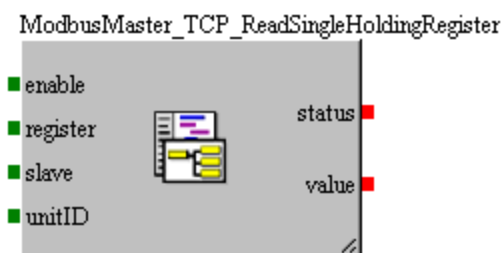
Библиотека может использоваться только для Modbus Master через TCP в контроллерах с включенным соединением Ethernet.

С помощью библиотек можно также последовательно читать больше регистров:

- *ModbusMaster\_TCP\_Read10MultipleInputRegisters*
- *ModbusMaster\_TCP\_Read100MultipleInputRegisters*

Они позволяют читать соответственно до 10 и 100 регистров подряд, выбирая на входе *num* количество регистров для чтения.

### ModbusMaster\_TCP\_ReadSingleHoldingRegister





<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>register</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Адрес подчиненного узла
<i>unitID</i>	CJ_BYTE	1-247	Удаленный адрес ведомого устройства в сети RS485, доступной через шлюз (Примечание: этот идентификатор 255 обычно используется для адресации самого шлюза)

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
<i>value</i>	CJ_WORD	0-65000	Прочитанное значение

### **Описание**

Библиотека ModbusMaster\_TCP\_ReadSingleHoldingRegister позволяет считывать значение в регистре хранения подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus TCP, которое реализует код функции 0x03.

Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на чтение данных.

После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

### **Замечания**

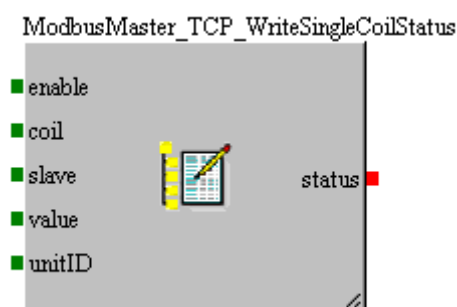
Эту библиотеку можно использовать в проекте несколько раз.  
Библиотека может использоваться только для Modbus Master через TCP в контроллерах с включенным соединением Ethernet.

С помощью библиотек можно также последовательно читать больше регистров:

- *ModbusMaster\_TCP\_Read10MultipleHoldingRegisters*
- *ModbusMaster\_TCP\_Read100MultipleHoldingRegisters*

Они позволяют читать соответственно до 10 и 100 регистров подряд, выбирая на входе *num* количество регистров для чтения.

### ModbusMaster\_TCP\_WriteSingleCoilStatus



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>enable</i>	CJ_BIT	0-1	Включает процесс считывания кадра ведущий/ведомый
<i>coil</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ
<i>slave</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Адрес подчиненного узла
<i>value</i>	CJ_BIT	0-1	Значение, которое необходимо записать
<i>unitID</i>	CJ_BYTE	1-247	Удаленный адрес ведомого устройства в сети RS485, доступной через шлюз (Примечание: этот идентификатор 255 обычно используется для адресации самого шлюза)

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
--------------	------------	----------------	-----------------

<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster
---------------	---------	-----	--

### Описание

Библиотека ModbusMaster\_TCP\_WriteSingleCoilStatus позволяет записывать значение в Регистр флагов подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus TCP, которое реализует код функции 0x05.

Чтобы начать операцию, необходимо установить переход от «0» к «1» на входе *enable*: будет добавлен в очередь запрос на запись данных.

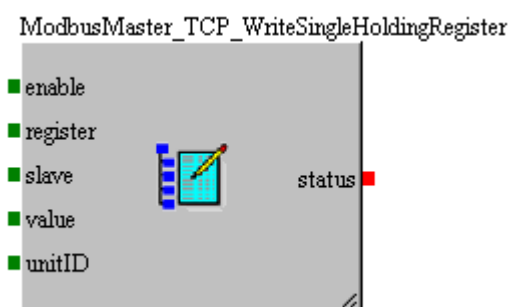
После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

### Замечания

Эту библиотеку можно использовать в проекте несколько раз.

Библиотека может использоваться только для Modbus Master через TCP в контроллерах с включенным соединением Ethernet.

### ModbusMaster\_TCP\_WriteSingleHoldingRegister



Вход	Тип	Границы	Описание
<i>enable</i>	CJ_BIT	0-1	Включает процесс записи кадра ведущий/ведомый
<i>register</i>	CJ_WORD	1-65000	Идентификатор регистра, к которому вы хотите получить доступ

<i>slave</i>	CJ_IP_ADDRESS	0.0.0.0 255.255.255.255	Адрес подчиненного узла
<i>value</i>	CJ_WORD	0-65000	Значение, которое необходимо записать
<i>unitID</i>	CJ_BYTE	1-247	Удаленный адрес ведомого устройства в сети RS485, доступной через шлюз (Примечание: этот идентификатор 255 обычно используется для адресации самого шлюза)

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>status</i>	CJ_BYTE	0-5	Возвращает состояние связи ведущий/ведомый: 0 = свободен для соединения 1 = добавлен запрос на соединение 2 = соединение в процессе 3 = соединение корректно завершилось 4 = соединение не завершилось корректно (т.е. ответ NAK или таймаут) 5 = соединение не корректно завершено из-за переполнения очереди запросов ModbusMaster

### Описание

Библиотека ModbusMaster\_TCP\_WriteSingleHoldingRegister позволяет записывать значение в регистр хранения подчиненного сетевого узла. Эта библиотека реализует отправку сообщения Modbus TCP, которое реализует код функции 0x06.

Для запуска записи необходимо установить переход с «0» на «1» на вход *enable*, в очередь будет добавлен запрос на запись данных, присутствующих в этот момент на входе *value*, для отправки в *register* ведомого *node*. После добавления в очередь сообщение будет обработано, а ответ будет проанализирован, контролируя выход *status*: когда выход имеет значение 3, ответ ведомого будет положительным, вместо этого, если выход имеет значение 4, ответ был отрицательным или ведомый не ответил. Установка *enable* на 0 освобождает очередь для следующей передачи. В течение всего периода связи должен поддерживаться вход *enable* = 1.

### Замечания

Эту библиотеку можно использовать в проекте несколько раз.  
Библиотека может использоваться только для Modbus Master через TCP в контроллерах с включенным соединением Ethernet.

С помощью библиотек можно также последовательно записать больше регистров:

- *ModbusMaster\_TCP\_Write10MultipleHoldingRegisters*
- *ModbusMaster\_TCP\_Write100MultipleHoldingRegisters*

Они позволяют читать соответственно до 10 и 100 регистров хранения подряд, выбирая на входе *num* количество регистров для чтения. Эти библиотеки реализуют отправку сообщения Modbus, реализующего код функции 0x10.

## 2.17 Password

### EnablePrgLevel



Вход	Tun	Границы	Описание
in	CJ_BIT	0-1	Устанавливает значение параметра управления 'EnablePrglevel'.

Выход	Tun	Границы	Описание
out	CJ_BIT	0-1	Считывает значение параметра управления 'EnablePrglevel'

#### Описание

Библиотека *EnablePrglevel* позволяет автоматически подключать запрос на отправление первой страницы уровней 1-5, всякий раз, когда соответствующая команда прерывается (см. список команд в инструкции по программному обеспечению UNI-PRO).

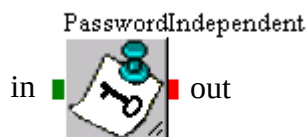
Некоторые из этих команд соответствуют конкретной комбинации кнопок терминала:  
 нажмите и удерживайте кнопку "PRG" около 3 секунд: *PrgLevel 1 command*;  
 нажмите и удерживайте одновременно кнопки "ENTER+ESC" около 3 секунд:  
*PrgLevel 2 command*;  
 нажмите и удерживайте одновременно кнопки "LEFT+RIGHT" около 3 секунд:  
*PrgLevel 3 command*;

Если '1', то при использовании комбинации вышеотмеченных клавиш, можно получить доступ к первой странице уровней 1-3. Если '0', тогда этот режим деактивируется. Для установки стандартного значения, присвойте находящемуся в библиотеке параметру *Enable\_Prglevel* значение *value*.

#### Замечания

Библиотека носит шаблонный характер.

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**PasswordIndependent**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_BIT	0-1	Устанавливает значение параметра управления 'PasswordIndependent'.

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_BIT	0-1	Считывает значение параметра управления 'PasswordIndependent'.

**Описание**

Библиотека *PasswordIndependent* позволяет распределять пароли на уровне страницы, как независимо, так и иерархически (пароли высшего уровня также имеют доступ к нижним уровням). Если данный параметр принимает значение '1', тогда уровни будут определяться как независимые. Для установки стандартного значения, присвойте находящемуся в библиотеке показателю параметра *Password\_Independent* значение *value*.

**Замечания**

Библиотека носит шаблонный характер.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**PasswordTimeout**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>in</i>	CJ_BYTE	0..240	Устанавливает количество секунд для настраивания задержки времени ожидания.

Выход	Тип	Границы	Описание
out	CJ_BYTE	0..240	Выдает количество секунд для настраивания задержки времени ожидания.

**Описание**

Библиотека *PasswordTimeout* позволяет устанавливать/выдавать количество секунд бездействия клавиатуры, после которого потребуется ввести заново пароль, чтобы получить доступ к различным уровням системы.

**Замечания**

Библиотека носит шаблонный характер. Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта. Дополнительные сведения о настройке уровней и паролей см. в руководстве по программному обеспечению UNI-PRO.

**PasswordLevel1 – PasswordLevel5**

Вход	Тип	Границы	Описание
in	CJ_SHORT	-32768..32767	Устанавливает пароль различных уровней защиты.

Выход	Тип	Границы	Описание
out	CJ_SHORT	-32768..32767	Считывает значения паролей для различных уровней защиты.

**Описание**

Библиотеки *PasswordLeveln* позволяют считывать и записывать пароли, относящиеся к различным уровням защиты. Каждый пароль может принимать значения в пределах диапазона, установленного типом данных CJ\_SHORT (от -32768 до 32767). Значение '0' означает, что этот уровень не имеет защиты. Для установки стандартного значения пароля для каждого уровня, присвойте находящемуся в библиотеке показателю параметра *Password\_Leveln* значение *value*.



**Замечания**

Библиотека носит шаблонный характер.  
 Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.  
 Дополнительные сведения о настройке уровней и паролей см. в руководстве по ПО.

**EnableLevel1 – EnableLevel5**

<b>Вход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BIT	0-1	Устанавливает подключения уровня n.

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	Выдает подключение уровня n.

**Описание**

Библиотеки *EnableLeveln* позволяют устанавливать состояние подключения относительно различных уровней защиты.  
 Для установки стандартного значения пароля для каждого уровня, присвойте находящемуся в библиотеке показателю параметра *Enable\_Leveln* значение *value*, считывать и записывать пароли, относящиеся к различным уровням защиты. Если '1', тогда уровень доступен (при желании может быть защищен паролем). Если '0', тогда уровень не доступен.

**Замечания**

Библиотека носит шаблонный характер.  
 Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.  
 Дополнительные сведения о настройке уровней и паролей см. в руководстве по ПО.

## 2.18 System

### InfoMaskNum



Выход	Tun	Границы	Описание
out	CJ_WORD	0..65535	Выдает номер встроенного ПО.

#### Описание

Библиотека *InfoMaskNum* имеет назначение возвращать код встроенного программного обеспечения использованного прибора.

### InfoMaskRev



Выход	Tun	Границы	Описание
out	CJ_WORD	0..65535	Выдает номер издания встроенного ПО.

#### Описание

Библиотека *InfoMaskRev* имеет назначение возвращать номер издания встроенного программного обеспечения использованного прибора.

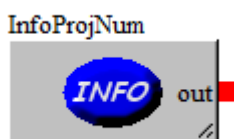
### InfoMaskVer



Выход	Tun	Границы	Описание
out	CJ_WORD	0..65535	Выдает номер издания встроенного ПО.

#### Описание

Библиотека *InfoMaskVer* имеет назначение выдавать версию встроенного программного обеспечения использованного прибора.

**InfoProjNum**

<i>Выход</i>	<i>Тип</i>	<i>Граниы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Выдает номер текущего проекта.

**Описание**

Библиотека *InfoProjNum* возвращает номер, присвоенный текущему проекту.

**InfoProjRev**

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Выдает номер редакции текущего проекта.

**Описание**

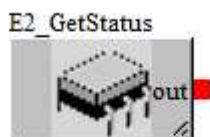
Библиотека *InfoProjRev* имеет назначение возвращать номер редакции текущего проекта.

**InfoProjVer**

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Выдает версию текущего проекта.

**Описание**

Библиотека *InfoProjVer* имеет назначение выдавать версию текущего проекта.

**E2\_GetStatus**

Выход	Тип	Границы	Описание
out	CJ_SHORT	0..3	Выдает код состояния для E2 памяти.

**Описание**

Библиотека *E2\_GetStatus* имеет назначение выдавать код, отражающий состояние контроллера EEPROM (E2) в момент, когда был сделан запрос.

Представлены следующие выдаваемые коды:

0 – *CJ\_E2\_OK*. Работает исправно.

1 – *CJ\_E2\_READ\_ERROR*. Была обнаружена ошибка доступа E2.

2 – *CJ\_E2\_WRITE\_ERROR*. Была обнаружена ошибка записи E2.

3 – *CJ\_E2\_CRC\_ERROR*. Несогласованные данные в памяти.

**RTC\_GetStatus**

Выход	Тип	Границы	Описание
out	CJ_SHORT	0..2	Выдает код состояния RTC.

**Описание**

Библиотека *RTC\_GetStatus* имеет назначение выдавать код, отражающий состояние контроллера RTC в момент, когда был сделан запрос.

Представлены следующие выдаваемые коды:

0 : *CJ\_RCT\_OK*. Работает исправно.

1 : *CJ\_RCT\_READ\_ERROR*. Была обнаружена ошибка доступа считывания RTC.

2 : *CJ\_RCT\_LOW\_VOLTAGE*. Чип RTC перешел минимальный порог необходимого напряжения для сохранения информации. Представленные данные не могут быть верными.

## Language



Вход	Тип	Границы	Описание
<i>in</i>	CJ_BYTE	0..255	Идентификатор языка, который необходимо установить.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..255	Идентификатор языка, который необходимо установить.

**Описание**

Библиотека *Language* позволяет изменять язык системы.

Идентификаторы языка имеют следующие соответствия:

- 0: Английский
- 1: Итальянский
- 2: Французский
- 3: Испанский
- 4: Немецкий
- 5: Русский
- 6: Португальский
- 7..255 : Для последующего задания.

## LCDBacklight



Вход	Тип	Границы	Описание
<i>in</i>	CJ_BYTE	0..2	Устанавливает режим подсветки LCD

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..2	Возвращает режим подсветки LCD

**Описание**

Библиотека *LCDBacklight* позволяет изменить режим подсветки LCD:

- 0: OFF. Подсветка всегда выключена
- 1: ON. Подсветка всегда включена
- 2: TIME. Подсветка включается в течение времени (по умолчанию 60 секунд)

**Замечания**

Библиотека носит шаблонный характер.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**LCDTouch\_Buzzer**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_BYTE	0..2	Устанавливает режим «бип» для сенсорного дисплея

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BYTE	0..2	Возвращает режим «бип» для сенсорного дисплея

**Описание**

Библиотека *LCDTouch\_Buzzer* позволяет изменять режим «бип» при использовании касания:

- 0: Нет звука
- 1: Звуковой сигнал в любой зоне
- 2: Звуковой сигнал только в области прикосновения

**Замечания**

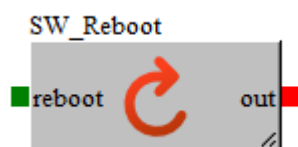
Библиотека носит шаблонный характер.  
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**Reset**

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>Reset_Q</i>	CJ_BIT	0-1	Выдает состояние Сброса. Выдает '1' если происходит Сброс, иначе выход принимает значение '0'.

**Описание**

Библиотека сброса позволяет выполнять действия по состоянию сброса системы. Это является начальной стадией, которую система обрабатывает перед запуском всей программы.

**SW\_Reboot**

Вход	Тип	Границы	Описание
reboot	CJ_BIT	0..1	Когда =1, начинается перезагрузка.

Выход	Тип	Границы	Описание
out	CJ_BIT	0..1	Вернуть статус перезагрузки.

**Описание**

Библиотека *SW\_Reboot* позволяет выполнить программный сброс. Программный сброс выполняется при изменении входа *reboot* с «0» на «1», при сбросе выход *out* становится равным «1» для данного основного цикла.

**Замечания**

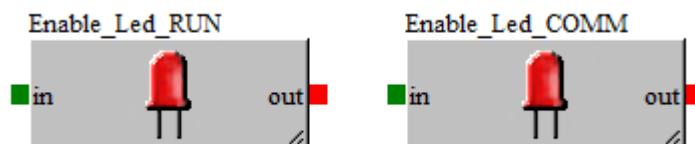
Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.

**Param\_GetStatus**

Выход	Тип	Границы	Описание
out	CJ_Byte	0-1	Возвращает статус области параметров. 0 = свободна или загрузка завершена ОК 1 = одно из этих условий: Инициализация и проверка памяти Лога Идет загрузка Ожидание сброса

**Описание**

Библиотека *Param\_GetStatus* позволяет проверять состояние области памяти параметров, при первой перезагрузке, после компиляции или при использовании команды сброса по умолчанию, это занимает много времени, особенно если параметров много.

**Enable\_Led\_RUN и Enable\_Led\_COMM**

Вход	Tun	Границы	Описание
<i>in</i>	CJ_BIT	0..1	Включить/выключить управление светодиодом диагностики контроллера.

Выход	Tun	Границы	Описание
<i>out</i>	CJ_BIT	0..1	Возвращает статус включения/отключения диагностического светодиода.

**Описание**

Библиотеки *Enable\_Led\_RUN* и *Enable\_Led\_COMM* позволяют включать/отключать управление диагностическим светодиодом RUN и COMMUNICATION (CAN).

**Замечания**

Очень важно, чтобы эта библиотека использовалась только один раз для каждого проекта.



## 3 STANDARD LIBRARIES

### 3.1 Arithmetic

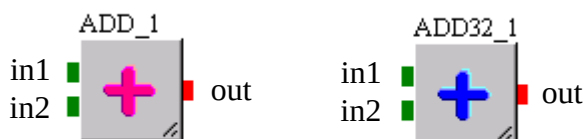
Эти библиотеки позволяют выполнять арифметические операции между 16- и 32-битными операндами.

Для каждой арифметической библиотеки проверяется возможное состояние переполнения; в зависимости от типа ошибки определяются соответствующие флаги, которые можно прочитать с помощью специальных функций:

- *CJ\_BIT CJ\_Math\_Error\_Read (void)*
- *CJ\_BIT CJ\_DivByZero\_Error\_Read (void)*
- *CJ\_BIT CJ\_Overflow\_Error\_Read (void)*
- *CJ\_BIT CJ\_Underflow\_Error\_Read (void)*
- *CJ\_BIT CJ\_NaN\_Error\_Read (void)*

Если функция возвращает «1», соответствующая ошибка активна.

#### ADD



Вход	Тип	Границы	Описание
<i>in1</i>	CJ_SHORT	-32768..32767	16-битный вход
<i>in1</i>	CJ_LONG	-2147483648..2147483647	32-битный вход
<i>in2</i>	CJ_SHORT	-32768..32767	16-битный вход
<i>in2</i>	CJ_LONG	-2147483648..2147483647	32-битный вход

Выход	Тип	Границы	Описание
<i>out</i>	CJ_SHORT	-32768..32767	Результат 16-битного сложения.
<i>out</i>	CJ_LONG	-2147483648.. 2147483647	Результат 32-битного сложения.

#### Описание

Библиотеки *ADD* и *ADD32* позволяют выполнение 16 и 32-битных операций.

#### Замечания

Для упрощения операций, библиотеки были объединены.

*CJ\_SHORT* тип входов/выходов относится к 16-битным операциям.

*CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

## SUB



Вход	Тип	Границы	Описание
in1	CJ_SHORT	-32768..32767	16-битный вход
in1	CJ_LONG	-2147483648..2147483647	32-битный вход
in2	CJ_SHORT	-32768..32767	16-битный вход
in2	CJ_LONG	-2147483648..2147483647	32-битный вход

Выход	Тип	Границы	Описание
out	CJ_SHORT	-32768..32767	Результат 16-битного сложения.
out	CJ_LONG	-2147483648.. 2147483647	Результат 32-битного сложения.

**Описание**

Библиотеки *SUB* и *SUB32* позволяют выполнение 16 и 32-битных операций вычитания.

**Замечания**

Операции выполняются с использованием следующей синтаксической структуры: in1-in2.  
 Для упрощения операций, библиотеки были объединены.  
*CJ\_SHORT* тип входов/выходов относится к 16-битным операциям.  
*CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

## MUL



Вход	Тип	Границы	Описание
in1	CJ_SHORT	-32768..32767	16-битный вход
in1	CJ_LONG	-2147483648..2147483647	32-битный вход
in2	CJ_SHORT	-32768..32767	16-битный вход
in2	CJ_LONG	-2147483648..2147483647	32-битный вход

Выход	Тип	Границы	Описание
out	CJ_SHORT	-32768..32767	Результат 16-битного умножения.
out	CJ_LONG	-2147483648.. 2147483647	Результат 32-битного умножения.

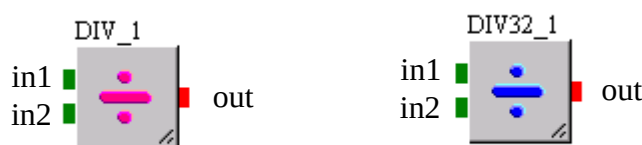
### Описание

Библиотеки *MUL* и *MUL32* позволяют выполнение 16 и 32-битных операций умножения соответственно.

### Замечания

Для упрощения операций, библиотеки были объединены.  
*CJ\_SHORT* тип входов/выходов относится к 16-битным операциям.  
*CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

## DIV



Вход	Тип	Границы	Описание
in1	CJ_SHORT	-32768..32767	16-битный вход
in1	CJ_LONG	-2147483648..2147483647	32-битный вход
in2	CJ_SHORT	-32768..32767	16-битный вход
in2	CJ_LONG	-2147483648..2147483647	32-битный вход

Выход	Тип	Границы	Описание
out	CJ_SHORT	-32768..32767	Результат 16-битного деления
out	CJ_LONG	-2147483648.. 2147483647	Результат 32-битного деления

### Описание

Библиотеки DIV и DIV32 позволяют выполнять 16- и 32-битные операции деления.

### Замечания

Операции выполняются с использованием следующей синтаксической структуры: in1/in2. Для упрощения операций, библиотеки были объединены. *CJ\_SHORT* тип входов/выходов относится к 16-битным операциям. *CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

## MOD



Вход	Тип	Границы	Описание
in1	CJ_SHORT	-32768..32767	16-битный вход
in1	CJ_LONG	-2147483648..2147483647	32-битный вход
in2	CJ_SHORT	-32768..32767	16-битный вход
in2	CJ_LONG	-2147483648..2147483647	32-битный вход

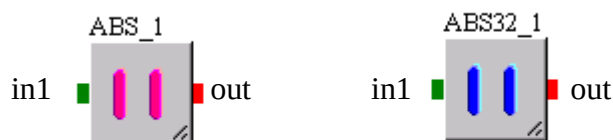
Выход	Тип	Границы	Описание
out	CJ_SHORT	-32768..32767	Остаток от деления между 16-битными входами
out	CJ_LONG	-2147483648.. 2147483647	Остаток от деления между 32-битными входами

**Описание**

Библиотеки *MOD* и *MOD32* позволяют получить остаток от деления между двумя выходами 16-битным и 32-битным соответственно.

**Замечания**

Операции выполняются с использованием синтаксической структуры: *in1 MOD in2*.  
 Для упрощения операций, библиотеки были объединены.  
*CJ\_SHORT* тип входов/выходов относится к 16-битным операциям.  
*CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

**ABS**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_SHORT	-32768..32767	16-битный вход
<i>in1</i>	CJ_LONG	-2147483648..2147483647	32-битный вход

Выход	Тип	Границы	Описание
<i>out</i>	CJ_SHORT	0..32767	Абсолютное 16-битное значение
<i>out</i>	CJ_LONG	0.. 2147483647	Абсолютное 32-битное значение

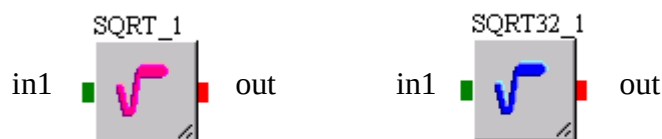
**Описание**

Библиотеки *ABS* и *ABS32* позволяют выполнять операции ввода 16- и 32-битных абсолютных значений соответственно.

**Замечания**

Для упрощения операций, библиотеки были объединены.  
*CJ\_SHORT* тип входов/выходов относится к 16-битным операциям.  
*CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

## SQRT



Вход	Тип	Границы	Описание
in1	CJ_SHORT	0..32767	16-битный вход
in1	CJ_LONG	0..2147483647	32-битный вход

Выход	Тип	Границы	Описание
out	CJ_SHORT	0..181	16-битный квадратный корень
out	CJ_LONG	0.. 46340	32-битный квадратный корень

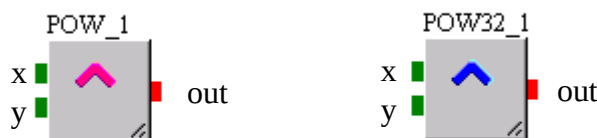
**Описание**

Библиотеки *SQRT* и *SQRT32* позволяют выполнять операции по извлечению 16 и 32-битного квадратного корня на основе данных входа *in*.

**Замечания**

Для упрощения операций, библиотеки были объединены.  
*CJ\_SHORT* тип входов/выходов относится к 16-битным операциям.  
*CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

## POW



Вход	Тип	Границы	Описание
x	CJ_SHORT	-32768..32767	16-битный вход основания
x	CJ_LONG	-2147483648..2147483647	32-битный вход основания

<i>y</i>	CJ_SHORT	-32768..32767	16-битный ввод степени
<i>y</i>	CJ_LONG	-2147483648..2147483647	32-битный ввод степени

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_SHORT	-32768..32767	16-битный результат.
<i>out</i>	CJ_LONG	-2147483648.. 2147483647	32-битный результат.

**Описание**

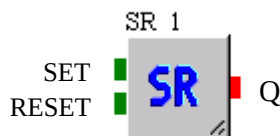
Библиотеки *POW* и *POW32* позволяют выполнять 16 и 32-битные операции возведения в степень соответственно.

**Замечания**

Операции выполняются с использованием следующей синтаксической структуры:  $x^y$ .  
 Для упрощения операций, библиотеки были объединены.  
*CJ\_SHORT* тип входов/выходов относится к 16-битным операциям.  
*CJ\_LONG* тип входов/выходов относится к 32-битным операциям.

## 3.2 Bistable

### SR



Вход	Tun	Границы	Описание
SET	CJ_BIT	0-1	Вход сигнала установки.
RESET	CJ_BIT	0-1	Вход сигнала сброса.

Выход	Tun	Границы	Описание
Q	CJ_BIT	0-1	Значение выхода.

#### Описание

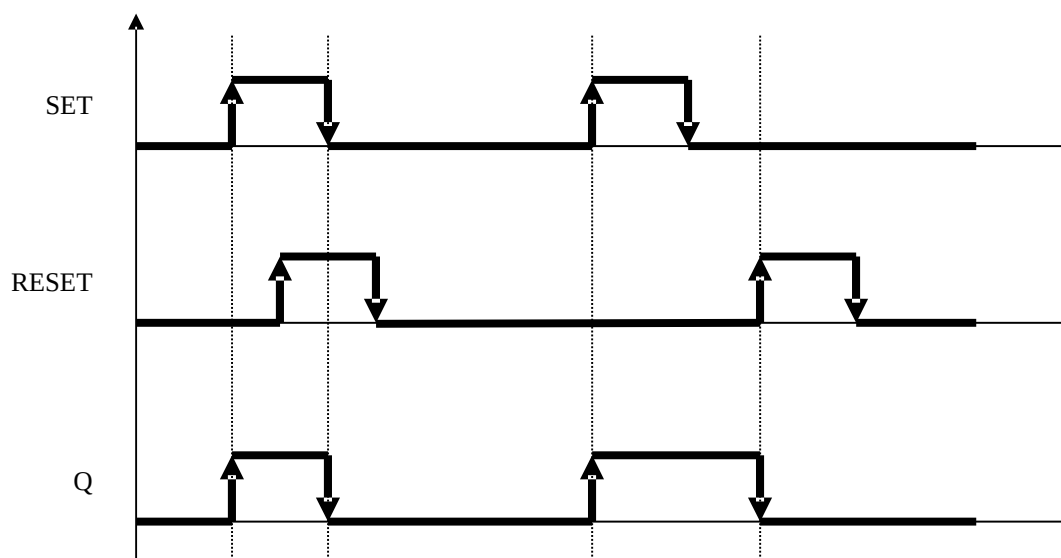
Библиотека SR выполняет функции SR, т.е. устанавливает выходное значение по логике:

- Q устанавливает значение '1' каждый раз, когда SET равняется '1'.
- Q устанавливает значение '0' каждый раз, когда RESET равняется '1', за исключением случаев, когда значение SET '1'.

Практически, выход настроен на отдачу предпочтения значению SET.

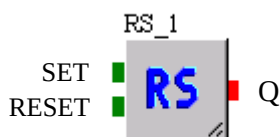
#### Замечания

Пожалуйста, обратитесь к следующему графику.





## RS



Вход	Тип	Границы	Описание
SET	CJ_BIT	0-1	Вход сигнала установки.
RESET	CJ_BIT	0-1	Вход сигнала сброса.

Выход	Тип	Границы	Описание
Q	CJ_BIT	0-1	Значение выхода.

**Описание**

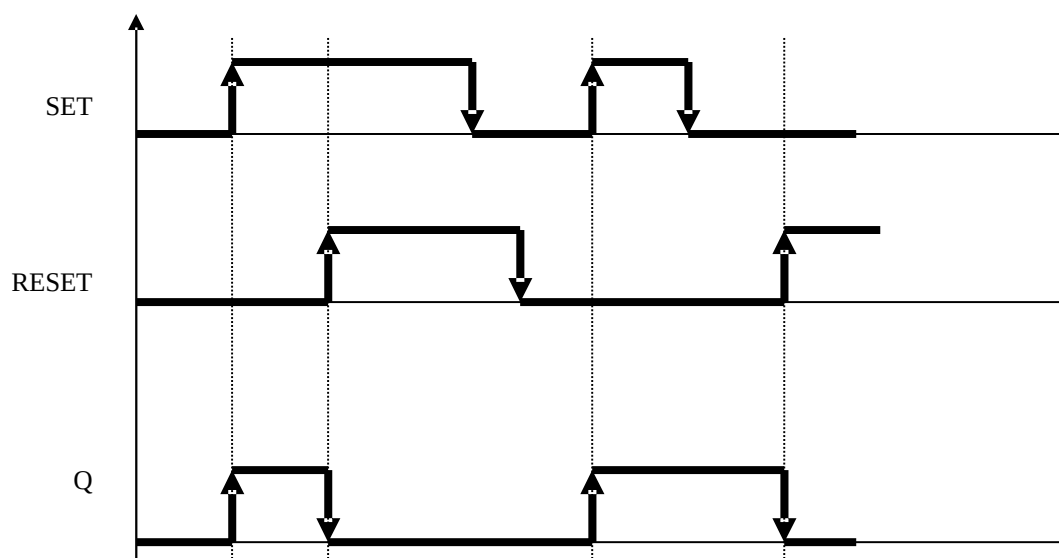
Библиотека RS выполняет RS функции, т.е. устанавливает значение выхода по логике:

- Q устанавливает значение '1' каждый раз, когда RESET равняется '1'.
- Q устанавливает значение '0' каждый раз, когда SET равняется '1', за исключением случаев, когда значение RESET '1'.

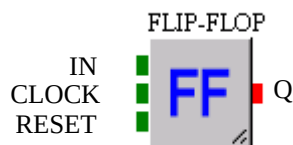
Практически, выход настроен на отдачу предпочтения значению RESET.

**Замечания**

Пожалуйста, обратитесь к следующему графику.



## FLIP-FLOP



Вход	$T_{up}$	Границы	Описание
<i>IN</i>	CJ_BIT	0-1	Вход сигнала установки.
<i>CLOCK</i>	CJ_BIT	0-1	Часы
<i>RESET</i>	CJ_BIT	0-1	Вход сигнала сброса.

Выход	$T_{up}$	Границы	Описание
<i>Q</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

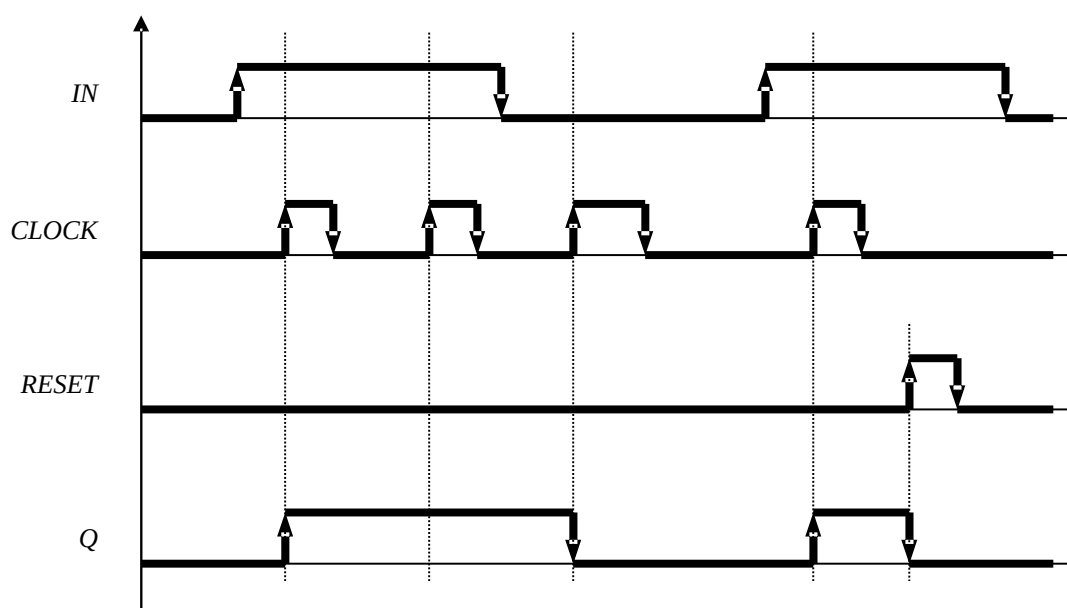
Библиотека *FLIP\_FLOP* выполняет функции SR триггера, т.е. для каждого переднего фронта сигнала она сохраняет значение входа и выдает его на выходе.

- Q устанавливается на значение *IN* для каждого переднего фронта входа *CLOCK*.
- Q устанавливается на значение «0» для каждого переднего фронта входа *RESET*.
- Во всех остальных случаях Q возвращает предыдущее значение.

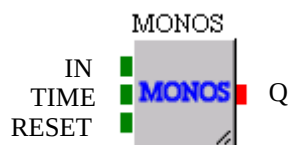
Практически, эта функция сохраняет вход и поддерживает его, пока условия не изменятся.

**Замечания**

Пожалуйста, обратитесь к следующему графику.



MONOS



Вход	<i>T<sub>up</sub></i>	Границы	Описание
<i>IN</i>	CJ_BIT	0-1	Вход сигнала установки.
<i>TIME</i>	CJ_WORD	0..65535	Время в 100мс в течение, которого выход сохранит значение '1'.
<i>RESET</i>	CJ_BIT	0-1	Вход сигнала сброса.

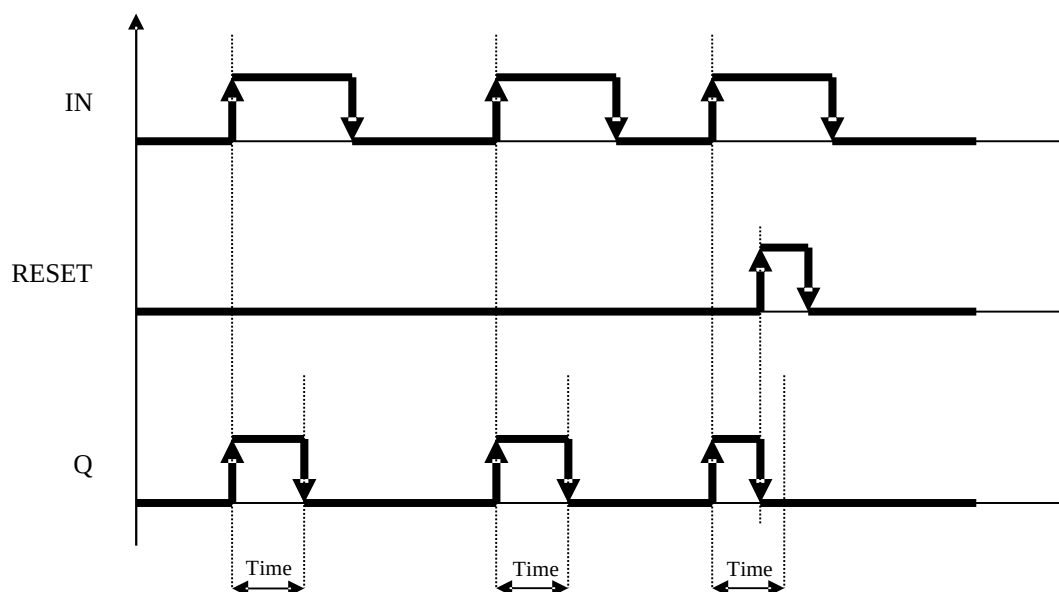
Выход	<i>T<sub>up</sub></i>	Границы	Описание
<i>Q</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Библиотека MONOS выполняет моностабильную функцию, для каждого входного переднего фронта выход остается «1» в течение периода времени, установленного входом TIME. Если во время этой фазы должен произойти СБРОС, то выход Q немедленно возвращается к «0», даже если установленное время не истекло. По истечении установленного времени, выход возвращается к значению «0». СБРОС, когда вход IN равен «0», не влияет на выход Q.

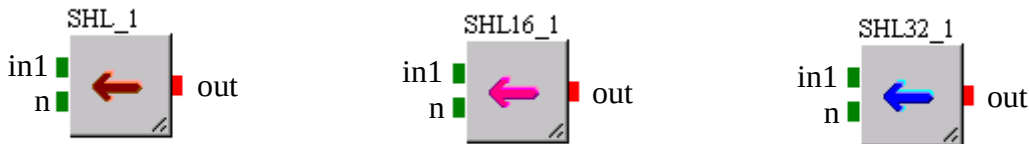
**Замечания**

Пожалуйста, обратитесь к следующему графику.



### 3.3 Bit Shift

#### SHL (Shift Left)



Вход	Тип	Границы	Описание
<i>in1</i>	CJ_BYTE	0..255	8-битный вход.
<i>in1</i>	CJ_WORD	0..65535	16-битный вход.
<i>in1</i>	CJ_DWORD	0..4294967295	32-битный вход
<i>n</i> (одинаковый для трех типов)	CJ_BYTE	0..255	Число разрядов, на которое был выполнен сдвиг.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..255	8-битный результат
<i>out</i>	CJ_WORD	0..65535	16-битный результат
<i>out</i>	CJ_DWORD	0..4294967295	32-битный результат

#### Описание

Библиотеки *SHL*, *SHL16* и *SHL32* позволяют выполнять операции сдвига влево 8, 16 и 32-битных входов соответственно.

Регулируя *n*, можно осуществить ввод *n* бит справа. Результатом будет то, что менее значимые биты бинарной строки будут переведены влево на *n* позиций.

Пример для 8 бит:

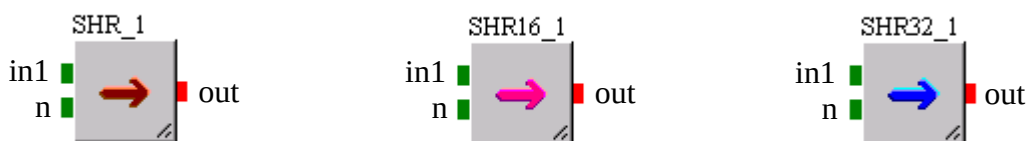
*in1* = 10011101 (157 или 0x9D)

*n* = 2

SHL = 01110100 (116 или 0x74)

#### Замечания

Чтобы сделать операции более унифицированными, библиотеки были объединены.

**SHR (Shift Right)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_BYTE	0..255	8-битный вход.
<i>in1</i>	CJ_WORD	0..65535	16-битный вход.
<i>in1</i>	CJ_DWORD	0..4294967295	32-битный вход.
<i>n</i> (одинаковый для трех типов)	CJ_BYTE	0..255	Число разрядов, на которое был выполнен сдвиг.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..255	8-битный результат
<i>out</i>	CJ_WORD	0..65535	16-битный результат
<i>out</i>	CJ_DWORD	0..4294967295	32-битный результат

**Описание**

Библиотеки SHR, SHR16 и SHR32 позволяют выполнять операции сдвига вправо на 8-, 16- и 32-битных входах соответственно.

Регулируя *n*, можно сделать так, чтобы *n* битов вводились слева. В результате старшие биты двоичной строки будут сдвинуты вправо на *n* позиций.

Пример для 8 бит:

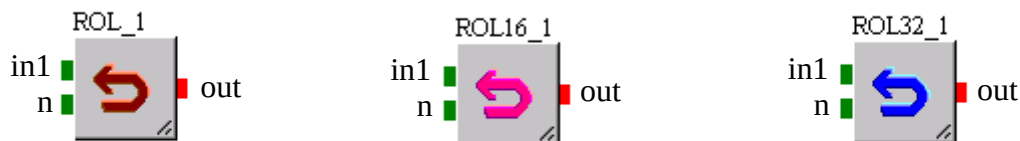
*in1* = 10011101 (157 или 0x9D)

*n* = 2

SHR = **00**100111 (39 или 0x27)

**Замечания**

Чтобы сделать операции более унифицированными, библиотеки были объединены.

**ROL (Rotary Left)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_BYTE	0..255	8-битный вход.
<i>in1</i>	CJ_WORD	0..65535	16-битный вход.
<i>in1</i>	CJ_DWORD	0..4294967295	32-битный вход.
<i>n</i> (одинаковый для трех типов)	CJ_BYTE	0..255	Количество битов, на которых должно быть выполнено вращение

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..255	8-битный результат
<i>out</i>	CJ_WORD	0..65535	16-битный результат
<i>out</i>	CJ_DWORD	0..4294967295	32-битный результат

**Описание**

Библиотеки ROL, ROL16 и ROL32 позволяют выполнять операции поворота влево на 8-, 16- и 32-битных входах соответственно.

Регулируя *n*, можно заставить вращаться последние *n* битов. В результате в начало двоичной строки будут вставлены младшие значащие *n* битов, а оставшиеся биты сдвинуты вправо на *n* позиций.

Пример для 8 бит:

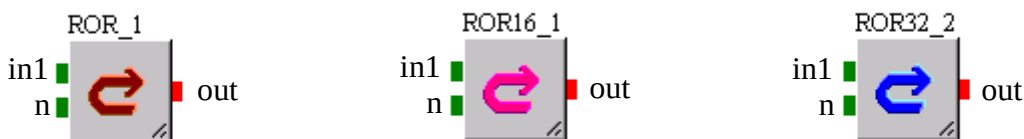
*in1* = **10011101** (157 или 0x9D)

*n* = 2

ROL = **01110110** (118 или 0x76)

**Замечания**

Чтобы сделать операции более унифицированными, библиотеки были объединены.

**ROR (Rotary Right)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_BYTE	0..255	8-битный вход.
<i>in1</i>	CJ_WORD	0..65535	16-битный вход.
<i>in1</i>	CJ_DWORD	0..4294967295	32-битный вход.
<i>n</i> (одинаковый для трех типов)	CJ_BYTE	0..255	Количество битов, на которых должно быть выполнено вращение

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BYTE	0..255	8-битный результат
<i>out</i>	CJ_WORD	0..65535	16-битный результат
<i>out</i>	CJ_DWORD	0..4294967295	32-битный результат

**Описание**

Библиотеки ROR, ROR16 и ROR32 позволяют выполнять операции вращения вправо на 8-, 16- и 32-битных входах соответственно.

Регулируя *n*, можно заставить первые *n* бит вращаться.

В результате старшие *n* битов будут вставлены в конец двоичной строки, а оставшиеся биты сдвинуты влево на *n* позиций.

Пример для 8 бит:

*in1* = 10011101 (157 или 0x9D)

*n* = 2

ROR = 01100111 (103 или 0x67)

**Замечания**

Чтобы сделать операции более унифицированными, библиотеки были объединены.

### 3.4 Comparators

#### EQ (Сравнение на равенство)



Вход	Тип	Границы	Описание
<i>in1</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.
<i>in2</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

#### Описание

Сравниваются входы *in1* и *in2*: если оба значения равны, тогда выход принимает значение '1', в другом случае, значение '0'.

#### Замечания

Данные входа преобразовываются в 32-битные значения со знаками.

#### GE (Сравнение на больше или равно)



Вход	Тип	Границы	Описание
<i>in1</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.
<i>in2</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

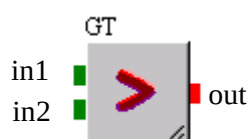


**Описание**

Сравниваются входы *in1* и *in2*: если значение *in1* больше или равно значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

**Замечания**

Данные входа преобразовываются в 32-битные значения со знаками.

**GT (Сравнение на больше)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.
<i>in2</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.

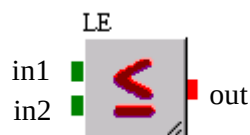
Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Сравниваются входы *in1* и *in2*: если значение *in1* больше значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

**Замечания**

Данные входа преобразовываются в 32-битные значения со знаками.

**LE (Сравнение на меньше или равно)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.

<i>in2</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.
------------	---------	-------------------------	---------------------

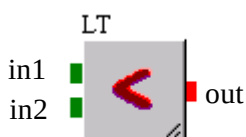
<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Сравниваются входы *in1* и *in2*: если значение *in1* меньше или равно значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

**Замечания**

Данные входа преобразовываются в 32-битные значения со знаками.

**LT (Сравнение на меньше)**

<b>Вход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.
<i>in2</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.

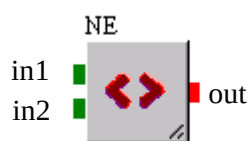
<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Сравниваются входы *in1* и *in2*: если значение *in1* меньше значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

**Замечания**

Данные входа преобразовываются в 32-битные значения со знаками.

**NE (Сравнение на неравенство)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.
<i>in2</i>	CJ_LONG	-2147483648..2147483647	Вход для сравнения.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Сравниваются входы *in1* и *in2*: если значение *in1* отличается от значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

**Замечания**

Данные входа преобразовываются в 32-битные значения со знаками.

**U\_GE (Сравнение без знака на больше или равно)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_DWORD	0..4294967295	Вход для сравнения.
<i>in2</i>	CJ_DWORD	0..4294967295	Вход для сравнения.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Сравниваются входы *in1* и *in2*: если значение *in1* больше или равно значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

**Замечания**

Данные входа преобразовываются в 32-битные значения без знаков.

**U\_GT (Сравнение без знака на больше)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_DWORD	0..4294967295	Вход для сравнения.
<i>in2</i>	CJ_DWORD	0..4294967295	Вход для сравнения.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Сравниваются входы *in1* и *in2*: если значение *in1* больше значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

**Замечания**

Данные входа преобразовываются в 32-битные значения без знаков.

**U\_LE (Сравнение без знака на меньше или равно)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_DWORD	0..4294967295	Input to be compared
<i>in2</i>	CJ_DWORD	0..4294967295	Input to be compared

Выход	Тип	Границы	Описание
-------	-----	---------	----------

<i>out</i>	CJ_BIT	0-1	Значение выхода.
------------	--------	-----	------------------

### Описание

Сравниваются входы *in1* и *in2*: если значение *in1* меньше или равно значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

### Замечания

Данные входа преобразовываются в 32-битные значения без знаков.

## U\_LT (Сравнение без знака на меньше)



Вход	Тип	Границы	Описание
<i>in1</i>	CJ_DWORD	0..4294967295	Вход для сравнения.
<i>in2</i>	CJ_DWORD	0..4294967295	Вход для сравнения.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

### Описание

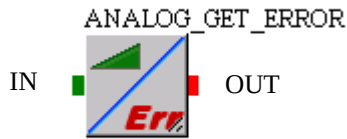
Сравниваются входы *in1* и *in2*: если значение *in1* меньше значения *in2*, тогда выход принимает значение '1', в другом случае, значение '0'.

### Замечания

Данные входа преобразовываются в 32-битные значения без знаков.

### 3.5 Conversion

#### ANALOG\_GET\_ERROR



Вход	Тип	Границы	Описание
IN	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Аналоговый вход.

Выход	Тип	Границы	Описание
OUT	CJ_BYTE	0..2	Тип ошибки.

#### Описание

Выдает тип ошибки аналогового входа, а именно:

0. Ошибки нет.
1. Короткое замыкание датчика.
2. Повреждение или отсутствие датчика.

#### ANALOG\_GET\_VALUE



Вход	Тип	Границы	Описание
IN	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Аналоговый вход.

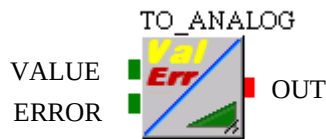
Выход	Тип	Границы	Описание
OUT	CJ_SHORT	-32768..32767	Значение датчика

#### Описание

Выдает значение, измеренное датчиком.

Если датчик не поврежден, то выдаваемое значение принимает максимум (32767).

## TO\_ANALOG



Вход	Тип	Границы	Описание
VALUE	CJ_SHORT	-32768..32767	Измеренное значение.
ERROR	CJ_BYTE	0..2	Ошибка

Выход	Тип	Границы	Описание
OUT	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Значение датчика.

### Описание

Создает CJ\_ANALOG структуру, используя значения, введенные на входе.

## ANALOG\_GET\_FIELDS

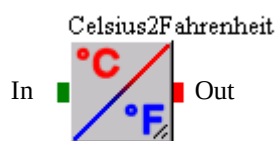


Вход	Тип	Границы	Описание
analogue	CJ_ANALOG	value: -32768..32767 error: 0..2	Вход датчика.

Выход	Тип	Границы	Описание
value:	CJ_SHORT	-32768..32767	Измеренное значение.
error:	CJ_BYTE	0..2	Ошибка

### Описание

Выдает поля ввода CJ\_ANALOG структуры.

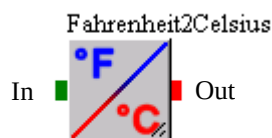
**Celsius2Fahrenheit**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>In</i>	CJ_SHORT	-32768..32767	Температура в °C

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>Out</i>	CJ_SHORT	-32768..32767	Температура в °F

**Описание**

Выполняет преобразование °C - °F.

**Fahrenheit2Celsius**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>In</i>	CJ_SHORT	-32768..32767	Температура в °F

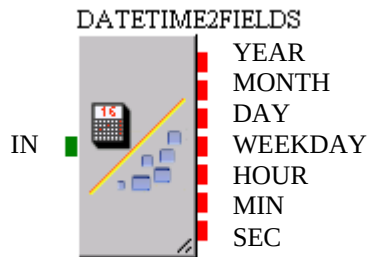
<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>Out</i>	CJ_SHORT	-32768..32767	Температура в °C

**Описание**

Выполняет преобразование °F - °C.

**DATETIME2FIELDS**





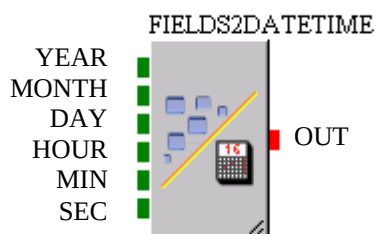
<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
IN	CJ_DATETIME	0..2147483647	Дата и время.

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
YEAR, MONTH, DAY, WEEKDAY, HOUR, MIN, SEC	CJ_BYTE	00..68 1..12 1..31 0..6 0..23 0..59	Год Месяц День День недели Час Минуты и секунды

**Описание**

Выдает отдельные поля введения даты и времени.

**FIELDS2DATETIME**



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
YEAR, MONTH, DAY, HOUR, MIN, SEC	CJ_BYTE	00..68 1..12 1..31 0..23 0..59	Год Месяц День Час Минуты и секунды
<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>

<i>OUT</i>	CJ_DATETIME	0..2147483647	Дата и время.
------------	-------------	---------------	---------------

**Описание**

Создает дату и время из полей ввода.

**DATETIME\_GET\_YEAR, DATETIME\_GET\_MONTH and DATETIME\_GET\_DAY**



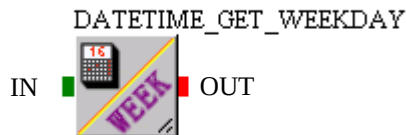
<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_DATETIME	0..2147483647	Дата и время.

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_BYTE	0..68 1..12 1..31	Год Месяц День

**Описание**

Выдает год, месяц и день, в отдельности от даты ввода.

**DATETIME\_GET\_WEEKDAY**

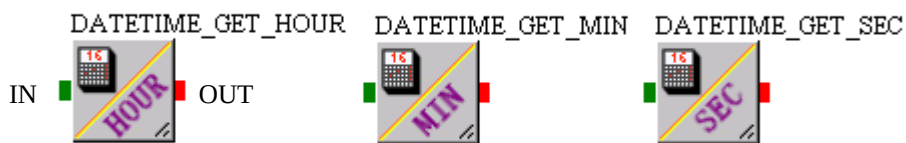


<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_DATETIME	0..2147483647	Дата и время.

Выход	Тип	Границы	Описание
OUT	CJ_BYTE	0..6	День недели.

Описание
Выдает цифру дня недели. [0=Воскресенье, ..., 6=Суббота]

### DATETIME\_GET\_HOUR, DATETIME\_GET\_MIN and DATETIME\_GET\_SEC



Вход	Тип	Границы	Описание
IN	CJ_DATETIME	0..2147483647	Дата и время.

Выход	Тип	Границы	Описание
OUT	CJ_BYTE	0..23 0..59 0..59	Часы Минуты Секунды

Описание
Выдает час, минуту и секунду, в отдельности от времени ввода.

### DATETIME\_GET\_TIME



Вход	Тип	Границы	Описание
IN	CJ_DATETIME	0..2147483647	Дата и время.

Выход	Тип	Границы	Описание
OUT	CJ_TIME	0..86400	Выдает время (час, минуту, секунду)

**Описание**

Данная библиотека выдает время (час, минуту, секунду) с выхода типа CJ\_DATETIME. Значение 86400 означает число секунд в одном дне.

**BYTE2BIT**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_BYTE	0..255	8-битный вход.

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_BIT	0-1	Binary output

**Описание**

Выдает '1', если IN принимает отличное от нуля значение, иначе выдает '0'.

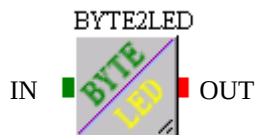
**BYTE2SBYTE**

<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_BYTE	0..255	8-битный вход (без знака).

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_S_BYTE	-128..127	8-битный выход (со знаком).

**Описание**

Выполняет преобразование 8-битных данных без знака в 8-битные данные со знаком.

**BYTE2LED**

<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_BYTE	0..3	Значение типа светодиода.

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_LED	0..3	Тип светодиода.

**Описание**

Устанавливает тип работы светодиода, в частности:

0. LED выключен
1. LED включен непрерывно
2. LED включен с низкой частотой
3. LED включен с высокой частотой

**LED2BYTE**

<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_LED	0..3	Тип светодиода.

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_BYTE	0..3	Значение типа светодиода.

**Описание**

Выдает вводимый тип действия светодиода, а именно:

4. LED выключен
5. LED включен непрерывно
6. LED включен с низкой частотой
7. LED включен с высокой частотой

## DWORD2LONG



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_DWORD	0..4294967295	32-битный вход (со знаком).

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_LONG	-2147483647.. 2147483648	32-битный выход (без знака).

### *Описание*

Выполняет преобразование 32-битных данных без знака в 32-битные данные со знаком.

## LONG2DWORD



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_LONG	-2147483647.. 2147483648	32-битный вход (без знака).

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
--------------	------------	----------------	-----------------

<i>OUT</i>	CJ_DWORD	0..4294967295	32-битный выход (со знаком).
------------	----------	---------------	------------------------------

**Описание**

Выполняет преобразование 32-битных данных со знаком в 32-битные данные без знака.

**SBYTE2BYTE**



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_S_BYTE	-128..127	8-битный вход (со знаком).

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_BYTE	0..255	8-битный выход (без знака).

**Описание**

Выполняет преобразование 8-битных данных со знаком в 8-битные данные без знака.

**SHORT2WORD**



<i>Вход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_SHORT	-32768..32767	16-битный вход (со знаком).

<i>Выход</i>	<i>Тип</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_WORD	0..65535	16-битный выход (без знака).

**Описание**

Performs the conversion of 16 bit data with sign to 16 bit data without sign

**WORD2SHORT**



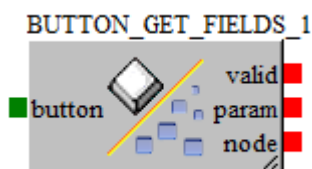
<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>IN</i>	CJ_WORD	0..65535	16-битный вход (без знака).

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>OUT</i>	CJ_SHORT	-32768..32767	16-битный выход (со знаком).

**Описание**

Выполняет преобразование 16-битных данных без знака в 16-битные данные со знаком.

**BUTTON\_GET\_FIELDS**



<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>button</i>	CJ_BTN	Valid: 0-1 Node: 0..127 Param: -32768..32767	Работа кнопки.

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>valid</i>	CJ_BIT	0-1	Поле действия.
<i>param</i>	CJ_WORD	0..65535	Поле параметра.
<i>node</i>	CJ_BYTE	0..127	Поле узла.

**Описание**



Выдает структурные поля CJ\_BTN.

*valid*. Устанавливает, было ли выполнено действие *button*. Если '1' – действие выполнено.  
*param*. Представляет количество секунда удерживания соответствующей кнопки.  
*node*. Логический узел, в котором действие было проверено.

## COMMAD\_GET\_FIELDS



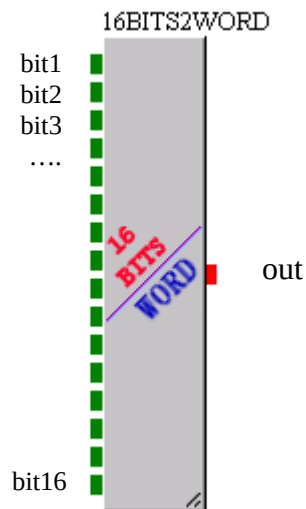
Вход	Tun	Границы	Описание
<i>command</i>	CJ_CMD	Valid: 0-1 Node: 0..127 Param: -32768..32767	Исполнение команды.

Выход	Tun	Границы	Описание
<i>valid</i>	CJ_BIT	0-1	Поле действия.
<i>param</i>	CJ_WORD	0..65535	Поле параметра.
<i>node</i>	CJ_BYTE	0..127	Поле узла.

### Описание

Выдает структурные поля CJ\_CMD.

*valid*. Устанавливает, было ли выполнено действие *command*. Если '1' – действие выполнено.  
*param*. Представляет параметр команды.  
*node*. Логический узел, в котором действие было проверено.

**16BITS2WORD**

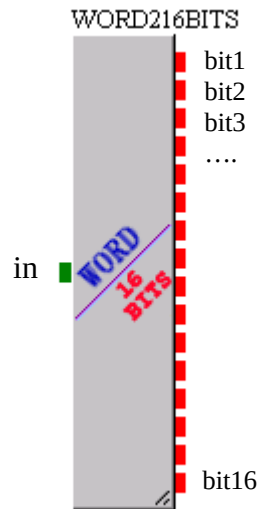
<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>bit1</i>	CJ_BIT	0-1	Первый бит.
<i>bit2</i>	CJ_BIT	0-1	Второй бит.
<i>bitN</i>	CJ_BIT	0-1	п-ый бит.
<i>bit16</i>	CJ_BIT	0-1	Шестнадцатый бит.

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..65535	Слова, извлеченные из строки разрядов, выделенных на входе.

**Описание**

Создает значение в Словах, начиная с 16 бит, выделенных на входе.

**WORD216BITS**

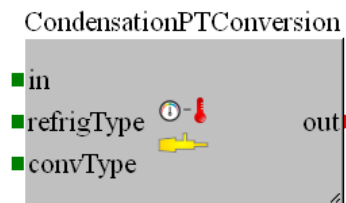


Вход	Tun	Границы	Описание
in	CJ_WORD	0..65535	Значение входа.

Выход	Tun	Границы	Описание
bit1	CJ_BIT	0-1	Первый бит.
bit2	CJ_BIT	0-1	Второй бит.
bitN	CJ_BIT	0-1	n-ый бит.
bit16	CJ_BIT	0-1	Шестнадцатый бит.

Описание
Выдает 16 одиночных разрядов, вычисленных из значения входа Слова.

**CondensationPTConversion**



Вход	Tun	Границы	Описание
in	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Преобразуемое значение в соответствии с параметром <i>convType</i> : -давление испарения в барах изб. с двумя десятичными знаками -Температура насыщения в °С с десятичной точкой

refrType	CJ_BYTE	0..19	Used GAS: 0: R-22 1: R-134A 2: R-402A 3: R-404A 4: R-407A 5: R-407C 6: R-410A 7: R-417A 8: R-422A 9: R-422D 10: R-507A 11: R-744 12: R-438A 13: R-401B 14: R-290 15: R-717 16: R-1270 17: R-32 18: R-407F 19:R-1234ZE
convType	CJ_BIT	0-1	Тип преобразования: 0: P →T преобразование 1: T →P преобразование

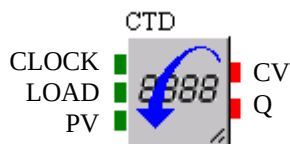
<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
out	CJ_ANALOG	Value:-32768..32767 Error: 0..2	Преобразованное значение, согласно параметру convType: -давление испарения в барах изб. с двумя десятичными знаками -Температура насыщения в °C с десятичной точкой

**Описание**

Вычисляет температуру насыщения [°C] по давлению [бар изб.] или давление насыщения [бар изб.] по температуре в [°C] для конкретного газа.

### 3.6 Counters

#### CTD / MCTD (Вычитающий счетчик)



Вход	Тип	Границы	Описание
<i>CLOCK</i>	CJ_BIT	0-1	Режим вычитания. При появлении переднего фронта на входе <i>CLOCK</i> , текущее значение <i>CV</i> уменьшается, за исключением случаев, когда вход <i>LOAD</i> активирован или когда <i>CV</i> принял значение 0.
<i>LOAD</i>	CJ_BIT	0-1	Когда вход <i>LOAD</i> будет установлен на '1', состояние счетчика загрузит значение <i>PV</i> .
<i>PV</i>	CJ_WORD	0..65535	Предустановленное значение. Начальное значение счетчика, которое загружается каждый раз, когда вход <i>LOAD</i> устанавливается на значение «1».

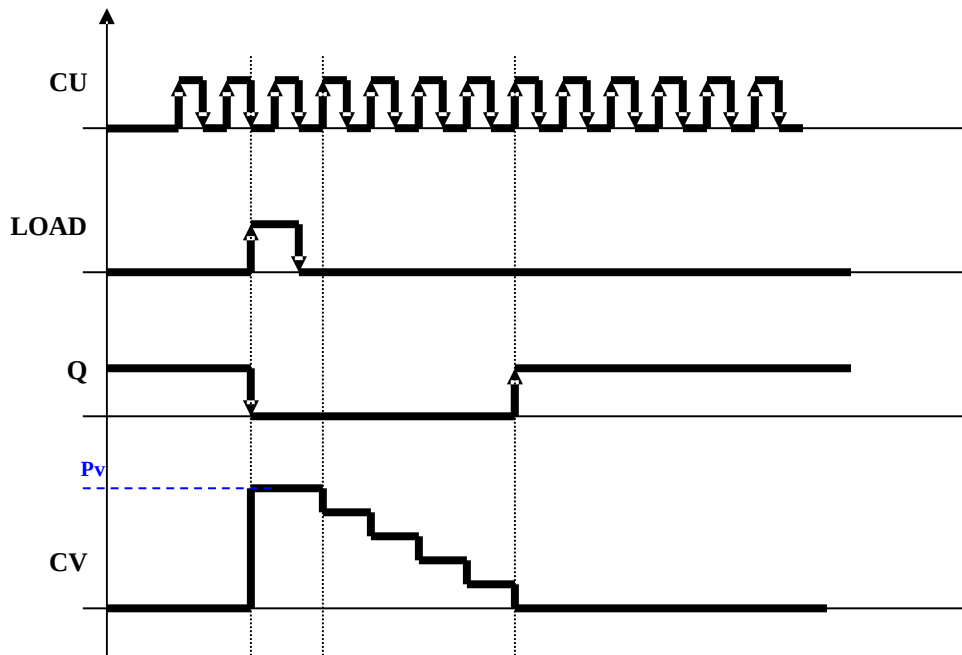
Выход	Тип	Границы	Описание
<i>CV</i>	CJ_WORD	0..65535	Текущее значение. Выдает текущее значение счетчика.
<i>Q</i>	CJ_BIT	0-1	Представляет завершение отсчета. Устанавливает '1', когда состояние счетчика, установлено на ноль ( <i>CV</i> =0).

#### Описание

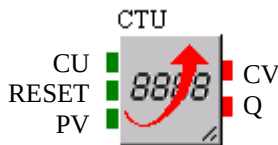
Запускает обратный отсчет события от установленного значения *PV*.  
MCTD сохраняет текущее значение и запускается с этого значения при подаче питания.

#### Замечания

Пожалуйста, обратитесь к следующему графику.



**СТУ / МСТУ (Суммирующий счетчик)**



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>CU</i>	CJ_BIT	0-1	Режим прибавления. При появлении переднего фронта на входе <i>CLOCK</i> , текущее значение <i>CV</i> увеличивается, за исключением случаев, когда вход <i>LOAD</i> активирован или когда <i>CV</i> равен 65535.
<i>RESET</i>	CJ_BIT	0-1	Когда вход <i>RESET</i> будет установлен на '1', состояние счетчика <i>CV</i> установится на ноль.
<i>PV</i>	CJ_WORD	0..65535	Предустановленное значение. Когда внутреннее состояние <i>CV</i> счетчика превысит <i>PV</i> , выход <i>Q</i> установится на '1'.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>CV</i>	CJ_WORD	0..65535	Текущее значение. Выдает текущее значение счетчика.

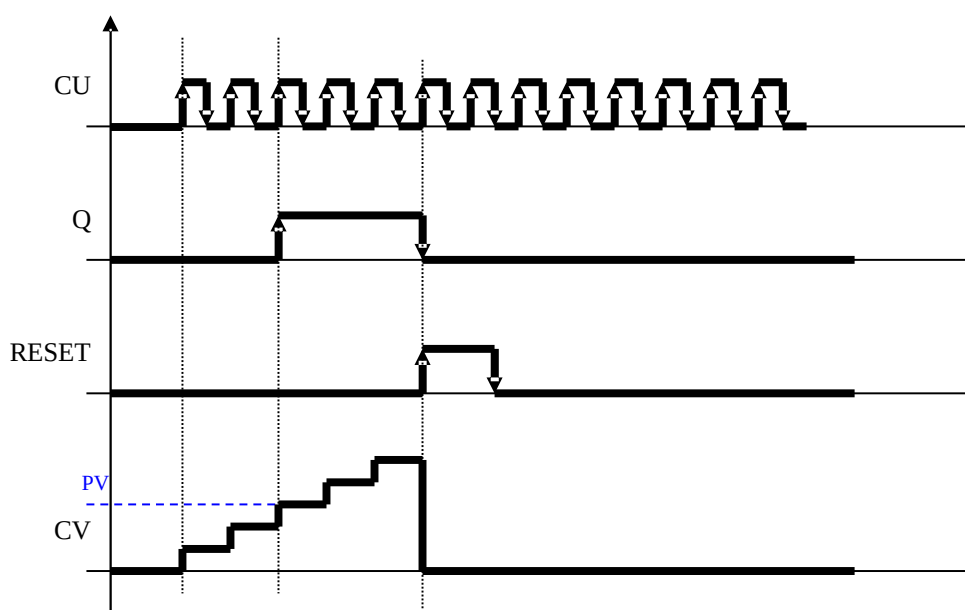
Q	CJ_BIT	0-1	Представляет завершение счета. Устанавливает '1', когда внутреннее состояние счетчика больше или равно состоянию входа PV.
---	--------	-----	--

**Описание**

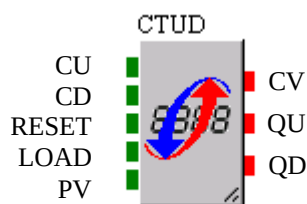
Запускает счетчик событий начинать отсчет со значения 0.  
MSTU сохраняет текущее значение и запускается с этого значения после подачи питания.

**Замечания**

Пожалуйста, обратитесь к следующему графику.



## CTUD (Up Down Counter)



Вход	Тип	Границы	Описание
CU	CJ_BIT	0-1	Режим счета вверх. При появлении переднего фронта на входе CU, текущее значение CV увеличивается, за исключением случаев, когда хотя бы один из входов LOAD и RESET активирован, до тех пор, пока не достигнет значения 65535. Если CU и CD работают одновременно, то счетчик увеличивает.
CD	CJ_BIT	0-1	Режим счета вниз. При появлении переднего фронта на входе на входе CLOCK, текущее значение CV уменьшается, за исключением случаев, когда активен хотя бы один из входов LOAD и RESET или когда CV достигает значения 0. Если CU и CD активны одновременно, счетчик увеличивает.
RESET	CJ_BIT	0-1	Когда вход RESET будет установлен на '1', внутреннее состояние счетчика CV установится на ноль.
LOAD	CJ_BIT	0-1	Когда вход LOAD будет установлен на '1', состояние счетчика загрузит значение PV. Если вход RESET в это время активирован, он игнорируется.
PV	CJ_WORD	0..65535	Заданное значение. Когда внутреннее состояние CV счетчика превысит PV, выход QD установится на '1'.

Выход	Тип	Границы	Описание
CV	CJ_WORD	0..65535	Текущее значение. Выдает текущее значение счетчика.
QU	CJ_BIT	0-1	Устанавливает '1', когда CV больше или равно PV.
QD	CJ_BIT	0-1	Устанавливает '1', когда CV равно нулю.

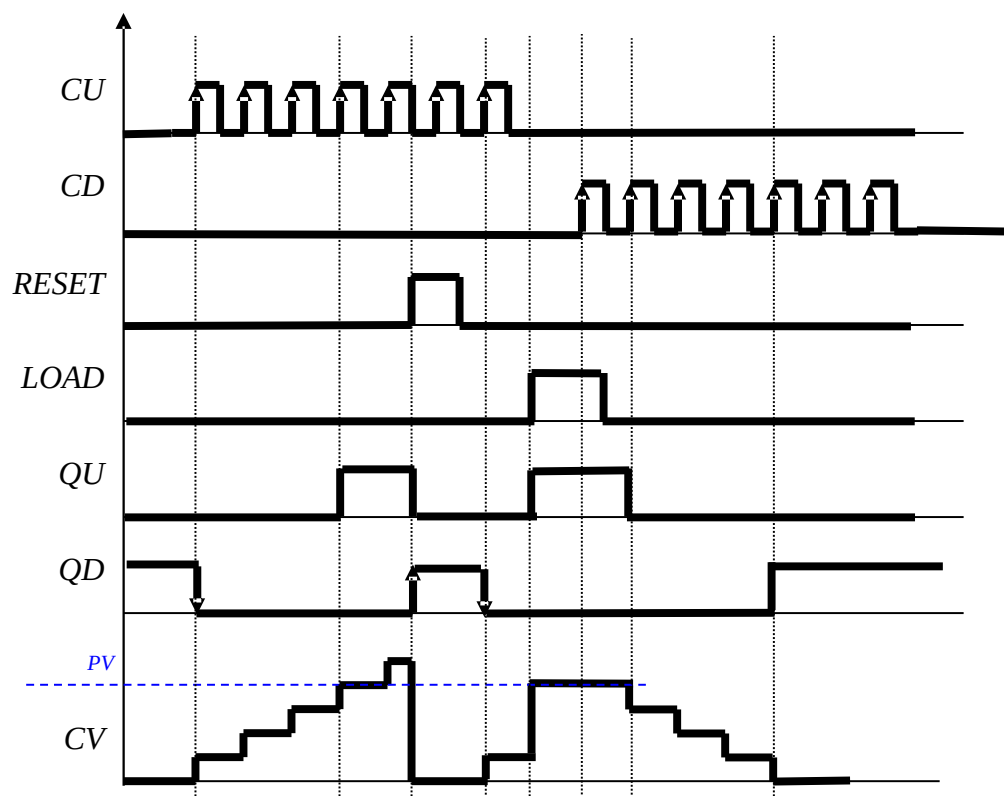


**Описание**

Заставляет счетчик событий подсчитывать в прямом и обратном направлении.

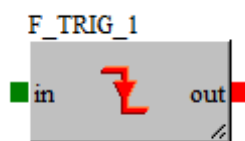
**Замечания**

Пожалуйста, обратитесь к следующему графику.



### 3.7 Edge detection

#### F\_TRIG



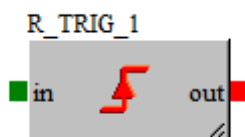
Вход	Tun	Границы	Описание
in	CJ_BIT	0-1	Такт

Выход	Tun	Границы	Описание
out	CJ_BIT	0-1	Когда задний фронт будет обнаружен (значение изменится от '1' до '0'), на выходе устанавливается '1' на время одного рабочего цикла (в основном с выдержкой времени в 5мс или 100мс).

#### Описание

Используйте эту библиотеку в случае, когда необходимо обнаружить изменение цифрового параметра с высокого уровня состояния на низкий.

#### R\_TRIG



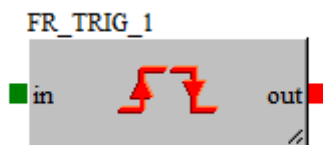
Вход	Tun	Границы	Описание
in	CJ_BIT	0-1	Такт

Выход	Tun	Границы	Описание
out	CJ_BIT	0-1	Когда передний фронт будет обнаружен (значение изменится от '0' до '1'), на выходе устанавливается '1' на время одного рабочего цикла (в основном с выдержкой времени в 5мс или 100мс).

#### Описание

Используйте эту библиотеку в случае, когда необходимо обнаружить переход цифрового параметра с низкого уровня состояния на высокий.

## FR\_TRIG



<i><b>Вход</b></i>	<i><b>Tun</b></i>	<i><b>Границы</b></i>	<i><b>Описание</b></i>
<i>in</i>	CJ_BIT	0-1	Такт

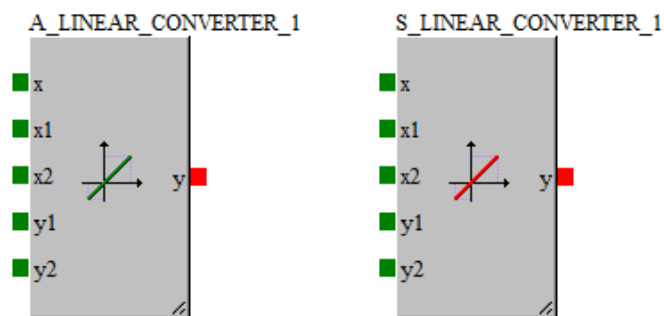
<i><b>Выход</b></i>	<i><b>Tun</b></i>	<i><b>Границы</b></i>	<i><b>Описание</b></i>
<i>out</i>	CJ_BIT	0-1	Когда на входе обнаруживается задний или передний фронт (значение меняется с «1» на «0» или наоборот), выход устанавливается на «1» на время одного цикла обработки (основного или 100 мс)

### ***Описание***

Используйте эту библиотеку в случае, когда требуется обнаружить изменение цифровой величины от высокого к низкому состоянию или наоборот.

### 3.8 Linear

#### A\_LINEAR\_CONVERTER



Вход	Тип	Границы	Описание
x	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Сигнал аналогового входа.
x	CJ_SHORT	-32768..32767	Числовое значение
x1	CJ_SHORT	-32768..32767	Координата x1.
x2	CJ_SHORT	-32768..32767	Координата x2.
y1	CJ_SHORT	-32768..32767	Координата y1.
y2	CJ_SHORT	-32768..32767	Координата y2.

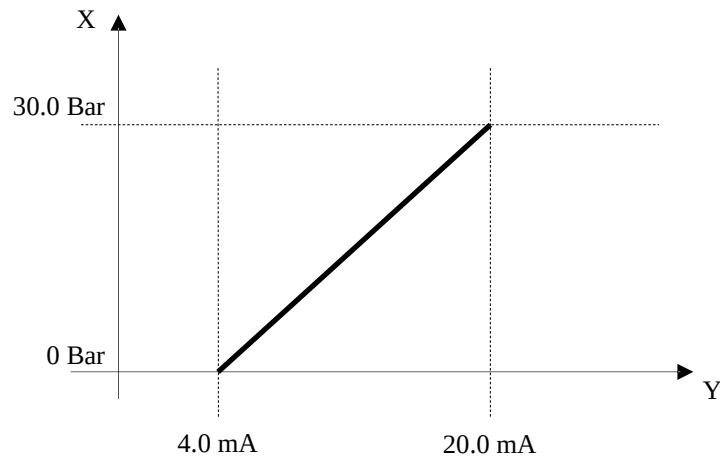
Выход	Тип	Границы	Описание
y	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Аналоговый сигнал линейризуется в соответствии со входами
y	CJ_SHORT	-32768..32767	Числовое значение линейризуется в соответствии со входами

#### Описание

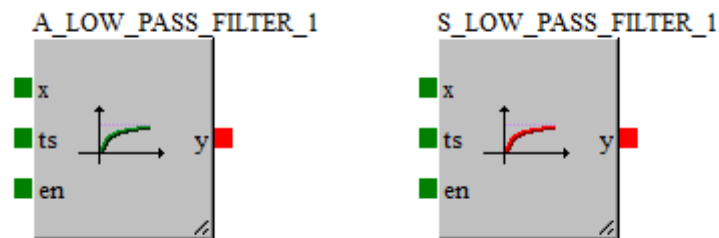
Библиотека *A\_Linear\_Converter* выполняет линейное преобразование входного аналогового сигнала/числового значения. Параметры функции задаются пользователем через входы x1, x2, y1 и y2. На выход функция предает линейризованный аналоговый сигнал/числовое значение.

#### Замечания

Если вход находится в состоянии ошибки, выход также будет в состоянии ошибки (библиотека *A\_Linear\_Converter*). Пожалуйста, обратитесь к следующему графику. Линейное преобразование от 4-20 мА до 0-30 бар. По оси ординат отложено давление (бар), а по оси абсцисс ток (мА).



### A\_LOW\_PASS\_FILTER - S\_LOW\_PASS\_FILTER



Вход	Тип	Границы	Описание
x	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Аналоговый входной сигнал
x	CJ_SHORT	-32768..32767	Числовое значение
ts	CJ_WORD	0..65535	Время установления, в 100 мс
en	CJ_BIT	0-1	Если '1', то фильтрация работает. Если '0', то выход соответствует входу.

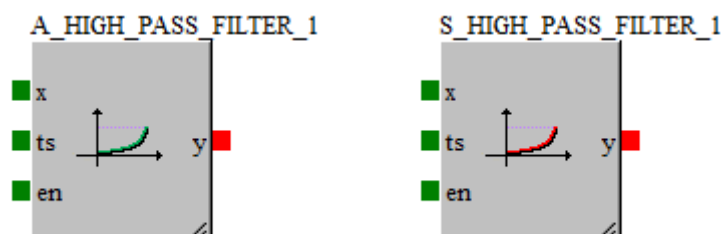
Выход	Тип	Границы	Описание
y	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Аналоговый сигнал устанавливается в соответствии с настройками входа.
y	CJ_SHORT	-32768..32767	Числовое значение устанавливается в соответствии с настройками входа.

#### Описание

Библиотека *LOW\_PASS\_FILTER* позволяет отфильтровывать входные колебания с помощью фильтра нижних частот. В случае каких-либо ступенчатых изменений на входе эта библиотека позволяет включить время установления до того, как выходное значение будет приведено в соответствие с входным значением *x*. Если функция активна (*en* = '1'), выходной сигнал достигнет значения входного сигнала через период времени, равный  $ts * 100$  мс.

**Замечания**

Для библиотеки *LOW\_PASS\_FILTER*, если вход находится в состоянии ошибки, выход также будет в состоянии ошибки.

**A\_HIGH\_PASS\_FILTER - S\_HIGH\_PASS\_FILTER**

Вход	Тип	Границы	Описание
x	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Аналоговый входной сигнал
x	CJ_SHORT	-32768..32767	Числовой входной сигнал
ts	CJ_WORD	0..65535	Время установления, в 100 мс
en	CJ_BIT	0-1	Если '1', то фильтрация работает. Если '0', то выход соответствует входу.

Выход	Тип	Границы	Описание
y	CJ_ANALOG	Value: -32768..32767 Error: 0..2	Аналоговый сигнал устанавливается в соответствии с настройками входа.
y	CJ_SHORT	-32768..32767	Числовое значение устанавливается в соответствии с настройками входа.

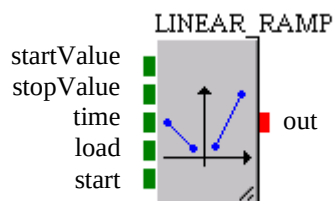
**Описание**

Библиотека *HIGH\_PASS\_FILTER* позволяет отфильтровывать входные колебания с помощью фильтра верхних частот. В случае каких-либо ступенчатых изменений на входе эта библиотека позволяет включить время установления до того, как выходное значение будет приведено в соответствие с входным значением x. Если функция активна (en = '1'), выходной сигнал достигнет значения входного сигнала через период времени, равный  $ts * 100$  мс.

**Замечания**

Для библиотеки *HIGH\_PASS\_FILTER*, если вход находится в состоянии ошибки, выход также будет в состоянии ошибки.

## LINEAR\_RAMP



Вход	Tun	Границы	Описание
<i>startValue</i>	CJ_SHORT	-32768..32767	Начальное значение.
<i>stopValue</i>	CJ_SHORT	-32768..32767	Конечное значение.
<i>time</i>	CJ_WORD	0..65535	Время, в секундах, необходимое для перехода от <i>startValue</i> до <i>stopValue</i> .
<i>load</i>	CJ_BIT	0-1	Загружает начальное значение и перенастраивает выход на это значение.
<i>start</i>	CJ_BIT	0-1	Включает функцию.

Выход	Tun	Границы	Описание
<i>out</i>	CJ_SHORT	-32768..32767	Выход функции.

**Описание**

Библиотека *LINEAR\_RAMP* выполняет линейно-нарастающую функцию  $y=kx+c$ , где угловой коэффициент  $k$  является функцией параметров *stopValue*, *startValue* и *time*.

Параметр *time* определяет количество секунда, необходимое функции, чтобы достичь значения *stopValue*, начиная с любой заданного начального значения *startValue*. В соответствии с вышеизложенными соображениями, угловой коэффициент линейно-нарастающей функции определяется следующим образом:  $stopValue - startValue / time$ .

Вход *start* имеет следующие функции:

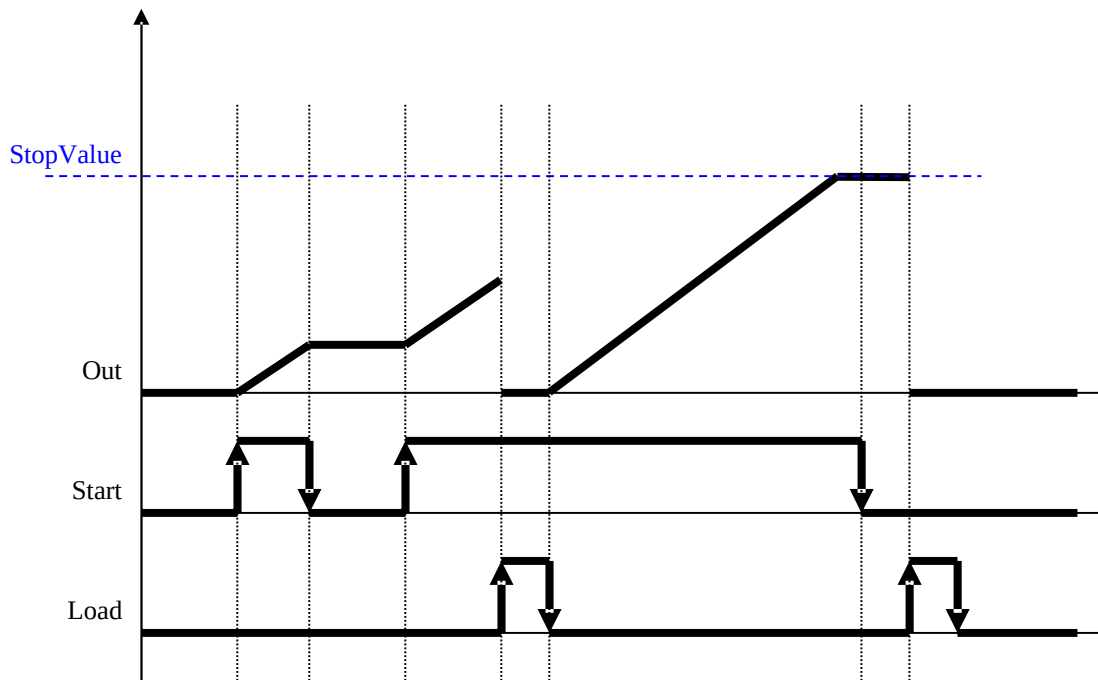
если установлено значение '1', то он запускает функцию;

если установлено значение '0', то он останавливает функцию, сохраняя значение на выходе в памяти. Если значение на входе *start* меняется на '1', то выход запускается снова с того места, где он был остановлен.

Вход *load* служит для переустановки выхода на начальное значение. Если *load*=1 линейно-нарастающая функция блокируется и выход принимает значение *startValue*.

**Замечания**

Ниже приведен пример графика для значения *startValue* равняющегося 0.



### 3.9 Logic

#### AND (0-1 AND)



Вход	Тип	Границы	Описание
<i>in1</i>	CJ_BIT	0-1	Цифровой вход.
<i>in2</i>	CJ_BIT	0-1	Цифровой вход.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

#### Описание

Выдает логическое И входов *in1* и *in2*.  
Если оба значения являются '1', выход принимает значение '1', иначе принимает значение '0'.



**OR (0-1 OR)**

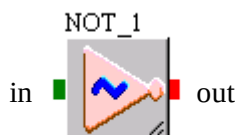
Вход	Тип	Границы	Описание
<i>in1</i>	CJ_BIT	0-1	Цифровой вход.
<i>in2</i>	CJ_BIT	0-1	Цифровой вход.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Выдает логическое ИЛИ входов *in1* и *in2*.

Если одно из двух значений является '1', выход принимает значение '1', иначе принимает значение '0'.

**NOT (0-1 NOT)**

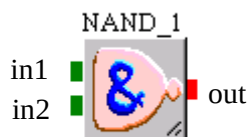
Вход	Тип	Границы	Описание
<i>in</i>	CJ_BIT	0-1	Цифровой вход.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Выдает отрицание цифрового входа *in*.

Если вход имеет значение '0', выход принимает значение '1', иначе принимает значение '0'.

**NAND (0-1 NAND)**

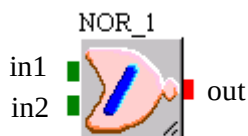
<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_BIT	0-1	Цифровой вход.
<i>in2</i>	CJ_BIT	0-1	Цифровой вход.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Выдает логическое И входов *in1* и *in2*.

Если оба значения являются '1', выход принимает значение '1', иначе принимает значение '0'.

**NOR (0-1 NOR)**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_BIT	0-1	Цифровой вход.
<i>in2</i>	CJ_BIT	0-1	Цифровой вход.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Выдает логическое ИЛИ входов *in1* и *in2*.

Если одно из двух значений является '1', выход принимает значение '1', иначе принимает '0'.

**XOR (0-1 XOR)**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_BIT	0-1	Цифровой вход.
<i>in2</i>	CJ_BIT	0-1	Цифровой вход.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	Значение выхода.

**Описание**

Выдает исключающее ИЛИ входов *in1* и *in2*.

Если значения различны, выход принимает значение '1', иначе принимает значение '0'.

**AND16 (16 bit AND)**

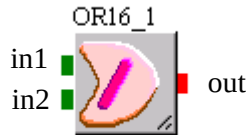
<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_WORD	0..65535	Цифровой вход.
<i>in2</i>	CJ_WORD	0..65535	Цифровой вход.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_WORD	0..65535	Значение выхода.

**Описание**

Выдает 16-битное И входов *in1* и *in2*.

Для каждого бита, если оба значения являются '1', соответствующий бит выхода принимает значение '1', иначе принимает значение '0'.

**OR16 (16 bit OR)**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_WORD	0..65535	Цифровой вход.
<i>in2</i>	CJ_WORD	0..65535	Цифровой вход.

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_WORD	0..65535	Значение выхода.

**Описание**

Выдает 16-битное ИЛИ входов *in1* и *in2*.

Для каждого бита, если, по крайней мере, одно из двух значений является '1', соответствующий бит выхода принимает значение '1', иначе, принимает значение '0'.

**NOT16 (16 bit NOT)**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_WORD	0..65535	Вход

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_WORD	0..65535	Значение выхода.

**Описание**

Выдает 16-битное отрицание цифрового входа *in*.  
 Для каждого бита, если значение являются '0', соответствующий бит выхода принимает значение '1', иначе принимает значение '0'.

### NAND16 (16 bit NAND)



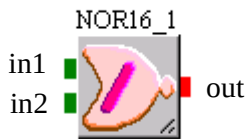
Вход	Tun	Границы	Описание
<i>in1</i>	CJ_WORD	0..65535	Вход
<i>in2</i>	CJ_WORD	0..65535	Вход

Выход	Tun	Границы	Описание
<i>out</i>	CJ_WORD	0..65535	Значение выхода.

#### Описание

Выдает 16-битное НЕ-И входов *in1* и *in2*.  
 Для каждого бита, если оба значения являются '1', соответствующий бит выхода принимает значение '0', иначе принимает значение '1'.

### NOR16 (16 bit NOR)



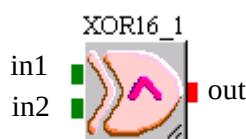
Вход	Tun	Границы	Описание
<i>in1</i>	CJ_WORD	0..65535	Вход
<i>in2</i>	CJ_WORD	0..65535	Вход

Выход	Tun	Границы	Описание
<i>out</i>	CJ_WORD	0..65535	Значение выхода.

**Описание**

Выдает 16-битное НЕ-ИЛИ входов *in1* и *in2*.

Для каждого бита, если, по крайней мере, одно из двух значений является '1', соответствующий бит выхода принимает значение '0', иначе принимает значение '1'.

**NOR16 (16 bit NOR)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_WORD	0..65535	Вход
<i>in2</i>	CJ_WORD	0..65535	Вход

Выход	Тип	Границы	Описание
<i>out</i>	CJ_WORD	0..65535	Значение выхода.

**Описание**

Выдает 16-битное исключающее ИЛИ входов *in1* и *in2*.

Для каждого бита, если значения входа различны, соответствующий бит выхода принимает значение '1', иначе принимает значение '0'.

**AND32 (32 bit AND)**

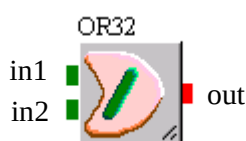
Вход	Тип	Границы	Описание
<i>in1</i>	CJ_DWORD	0..4294967295	Вход
<i>in2</i>	CJ_DWORD	0..4294967295	Вход

Выход	Тип	Границы	Описание
out	CJ_DWORD	0..4294967295	Значение выхода.

**Описание**

Выдает 32-битное И входов *in1* и *in2*.

Для каждого бита, если оба значения являются '1', соответствующий бит выхода принимает значение '1', иначе принимает значение '0'.

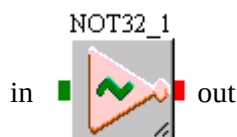
**OR32 (32 bit OR)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_DWORD	0..4294967295	Вход
<i>in2</i>	CJ_DWORD	0..4294967295	Вход
Выход	Тип	Границы	Описание
out	CJ_DWORD	0..4294967295	Значение выхода.

**Описание**

Выдает 32-битное ИЛИ входов *in1* и *in2*.

Для каждого бита, если, по крайней мере, одно из двух значений является '1', соответствующий бит выхода принимает значение '1', иначе, принимает значение '0'.

**NOT32 (32 bit NOT)**

Вход	Тип	Границы	Описание
<i>in1</i>	CJ_DWORD	0..4294967295	Вход
<i>in2</i>	CJ_DWORD	0..4294967295	Вход

Выход	Тип	Границы	Описание
out	CJ_DWORD	0..4294967295	Значение выхода.

**Описание**

Выдает 32-битное отрицание цифрового входа *in*.  
Для каждого бита, если значение являются '0', соответствующий бит выхода принимает значение '1', иначе значение '0'.

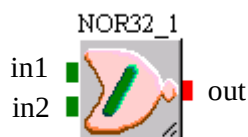
**NAND32 (32 bit NAND)**

Вход	Тип	Границы	Описание
in1	CJ_DWORD	0..4294967295	Вход
in2	CJ_DWORD	0..4294967295	Вход

Выход	Тип	Границы	Описание
out	CJ_DWORD	0..4294967295	Значение выхода.

**Описание**

Выдает 32-битное НЕ-И входов *in1* и *in2*.  
Для каждого бита, если оба значения являются '1', соответствующий бит выхода принимает значение '0', иначе значение '1'.

**NOR32 (32 bit NOR)**



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_DWORD	0..4294967295	Вход
<i>in2</i>	CJ_DWORD	0..4294967295	Вход

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_DWORD	0..4294967295	Значение выхода.

**Описание**

Выдает 32-битное НЕ-ИЛИ входов *in1* и *in2*.  
 Для каждого бита, если, по крайней мере, одно из двух значений является '1', соответствующий бит выхода принимает значение '0', иначе принимает значение '1'.

**XOR32 (32 bit XOR)**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>in1</i>	CJ_DWORD	0..4294967295	Вход
<i>in2</i>	CJ_DWORD	0..4294967295	Вход

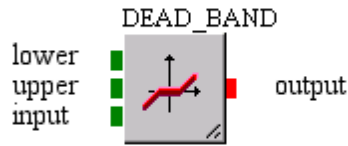
<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_DWORD	0..4294967295	Значение выхода.

**Описание**

Выдает 32-битное исключающее ИЛИ входов *in1* и *in2*.  
 Для каждого бита, если значения входа различны, соответствующий бит выхода принимает значение '1', иначе принимает значение '0'.

### 3.10 Not Linear

#### DEAD\_BAND



Вход	Тип	Границы	Описание
<i>lower</i>	CJ_SHORT	-32768.. <i>upper</i>	Граница нижнего диапазона.
<i>upper</i>	CJ_SHORT	<i>lower</i> ..32767	Граница верхнего диапазона.
<i>input</i>	CJ_SHORT	-32768..32767	Вход для проверки

Выход	Тип	Границы	Описание
<i>output</i>	CJ_SHORT	-32768..32767	Регулируемый выход.

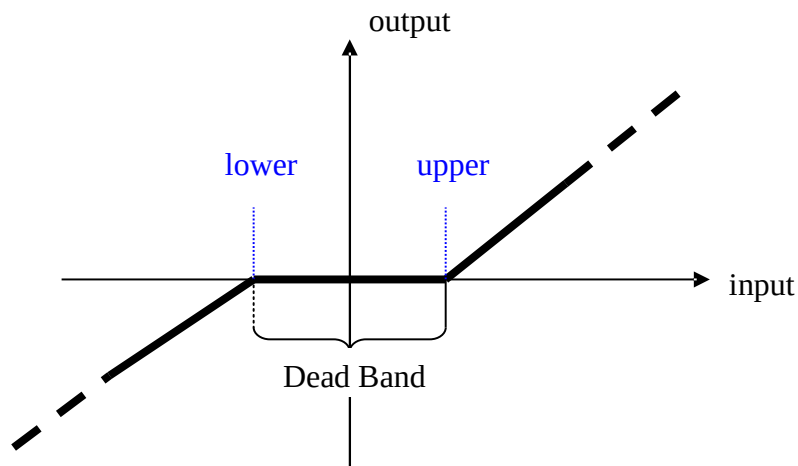
#### Описание

Библиотека *DEAD\_BAND* сравнивает значение на входе с мертвой зоной, которая установлена входами *lower* и *upper*. Результат функции выдается на выходе в соответствии со следующими критериями:

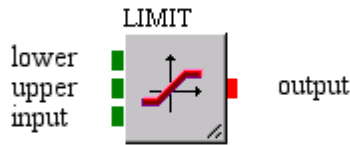
Если входное значение входа *input* меньше границы нижнего диапазона, то выход будет отвечать тому входу, из которого значение *lower* будет вычтено.

Если входное значение входа *input* больше границы верхнего диапазона, то выход будет отвечать тому входу, к которому значение *upper* будет добавлено.

Если входное значение входа *input* находится в пределах заданного диапазона, тогда выход принимает значение ноль.



**LIMIT**



Вход	Tun	Границы	Описание
<i>lower</i>	CJ_SHORT	-32768.. <i>upper</i>	Нижний предел
<i>upper</i>	CJ_SHORT	<i>lower</i> ..32767	Верхний предел
<i>input</i>	CJ_SHORT	-32768..32767	Вход для проверки

Выход	Tun	Границы	Описание
<i>output</i>	CJ_SHORT	-32768..32767	Регулируемый выход.

**Описание**

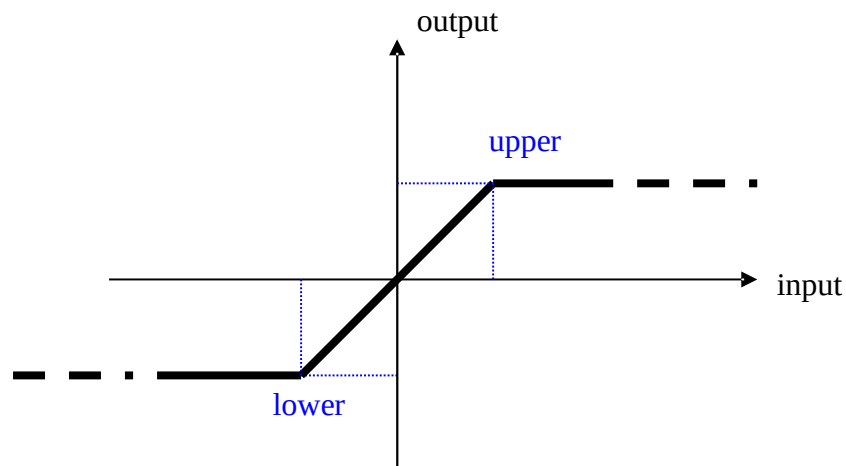
Библиотека *LIMIT* сравнивает значение на входе с верхней и нижней границами. Нижняя граница задается входом *lower*, верхняя граница задается входом *upper*. Результат функции выдается на выходе в соответствии со следующими критериями:

Если входное значение меньше нижней границы входа *lower*, тогда выход выведет значение нижней границы.

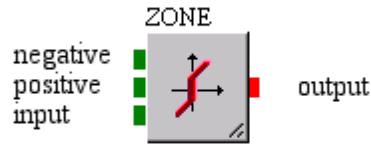
Если входное значение больше верхней границы входа *upper*, тогда выход выведет значение верхней границы.

Если значение находится в пределах *lower* и *upper* границ, тогда на выходе будет не измененное значение входа.

(См. следующий график).



ZONE



Вход	Tun	Границы	Описание
<i>negative</i>	CJ_SHORT	-32768..positive	Отрицательное ответвление.
<i>positive</i>	CJ_SHORT	negative..32767	Положительное ответвление.
<i>input</i>	CJ_SHORT	-32768..32767	Вход для проверки

Выход	Tun	Границы	Описание
<i>output</i>	CJ_SHORT	-32768..32767	Регулируемый выход.

**Описание**

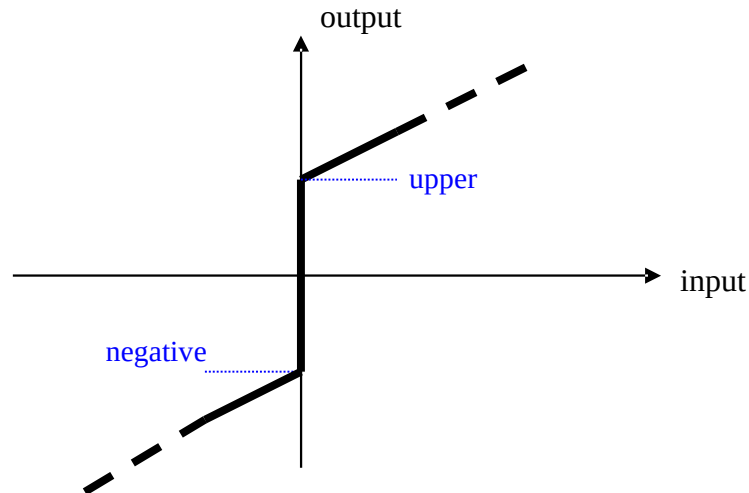
Библиотека *ZONE* добавляет значение ответвления к значению входа *input*. Отрицательные и положительные значения ответвления назначаются входами *negative* и *positive* соответственно. Результат функции выдается на выходе в соответствии со следующими критериями:

Если входное значение меньше нуля, тогда входное значение, к которому было прибавлено отрицательное ответвление, будет выведено как выход.

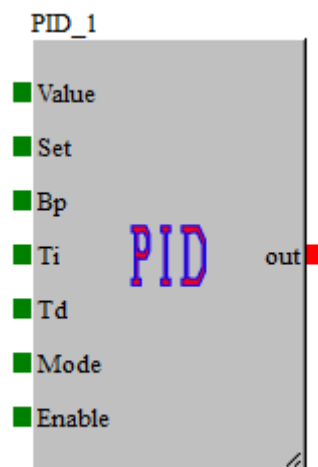
Если входное значение равняется нулю, тогда ноль будет выведен на выходе.

Если входное значение больше нуля, тогда входное значение, к которому было прибавлено положительное ответвление, будет выведено как выход.

(См. следующий график).



## PID

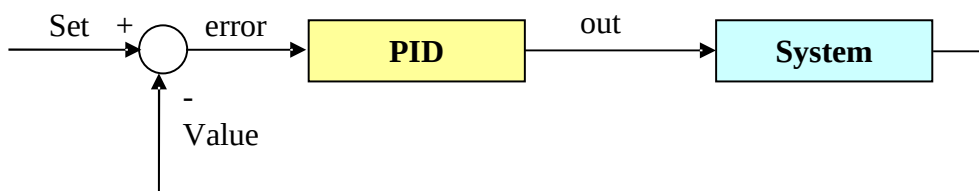


<i>Вход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>Value</i>	CJ_SHORT	-32768..32767	Регулируемое значение.
<i>Set</i>	CJ_SHORT	-32768..32767	Заданное значение.
<i>Bp</i>	CJ_WORD	0..65535	Зона пропорционального регулирования.
<i>Ti</i>	CJ_WORD	0..999	Время интегрирования (в секундах).
<i>Td</i>	CJ_WORD	0..999	Время дифференцирования (в секундах).
<i>Mode</i>	CJ_BIT	0-1	Режим работы: 0 = прямое действие; 1 = обратное действие.
<i>Enable</i>	CJ_BIT	0-1	Включить ПИД-регулирование

<i>Выход</i>	<i>Tun</i>	<i>Границы</i>	<i>Описание</i>
<i>out</i>	CJ_WORD	0..10000	Выход в процентах с сотыми долями.

**Описание**

Эта библиотека выполняет функцию ПИД-регулятора, который создает управляющий сигнал, представляющий собой сумму слагаемых: пропорциональное ошибке (часть P), пропорциональное интегралу (часть I) и производное ошибки (часть D). Логика, обеспечиваемая регулятором, следует идее, определенной на следующем рисунке:



В зависимости от установок времени интегрирования и дифференцирования, общий ПИД-регулятор может быть упрощен до трех типов дифференциальных регуляторов:

- Если параметр  $T_i$  равен нулю, вычисления интегральной части не выполняются, и регулятор будет работать, как пропорционально-дифференциальный регулятор (ПД)
- Если параметр  $T_d$  равен нулю, вычисления дифференциальной части не выполняются, и регулятор будет работать, как пропорционально-интегральный (ПИ)
- Если оба параметра  $T_i$  и  $T_d$  равны нулю, регулятор будет работать, как пропорциональный регулятор (П-регулятор).

В случае, когда все параметры установлены на значения, отличные от нуля, итоговая функция будет суммой трех воздействий, каждое из которых имеет разное действие.

*Пропорциональное регулирование* выдает на выход значение, пропорциональное измеренной ошибке. Это функция отклонения, чем больше коэффициент усиления, тем меньше ошибка.

*Интегральное регулирование* используется для приведения выходного сигнала регулятора к заданному значению и уменьшает отклонение только за счет пропорционального действия.

*Дифференциальное регулирование* уменьшает отклонение регулируемой величины и возвращает выход на заданное значение наиболее быстро. В зависимости от отклонения регулируемой величины, регулирование будет иметь тормозящее или ускоряющее действие на значение на выходе. Управляющий сигнал в процентах с сотыми долями будет соответствовать величине управления для того, чтобы итоговый сигнал был построен 'ближе' к установленному значению (параметр  $Set$ ).

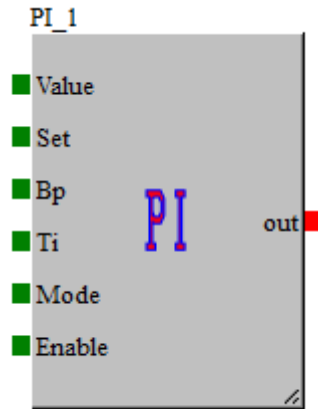
Параметр  $Mode$  разделяет регулятор на прямое ( $Mode=0$ ) или обратное ( $Mode=1$ ) регулирование системы контроля. Например:

- $Mode = 1$ , если выбрано управление процессом нагрева;
- $Mode = 0$ , если выбрано управления процессом охлаждения.

**Замечания**

Для исправной работы библиотеки, параметры  $V_r$ ,  $T_i$  и  $T_b$  установлены на базис динамических характеристик контролируемой системы (например, при использовании одного из различных ПИД-методов авто настройки).

PI

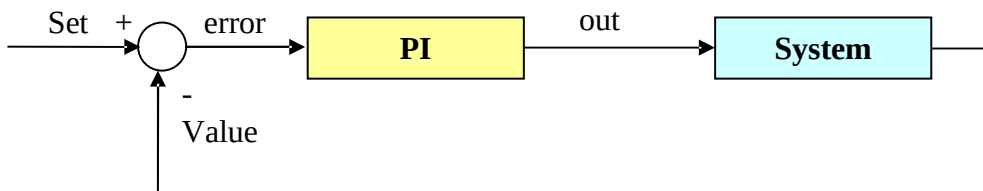


Вход	Тип	Границы	Описание
Value	CJ_SHORT	-32768..32767	Регулируемое значение.
Set	CJ_SHORT	-32768..32767	Заданное значение.
Bp	CJ_WORD	0..65535	Зона пропорционального регулирования.
Ti	CJ_WORD	0..999	Время интегрирования (в секундах).
Mode	CJ_BIT	0-1	Режим работы: 0 = прямое действие; 1 = обратное действие.
Enable	CJ_BIT	0-1	Включить ПИ-регулирование

Выход	Тип	Границы	Описание
out	CJ_WORD	0..10000	Выход в процентах с сотыми долями.

**Описание**

Библиотека выполняет функцию ПИ-регулятора, который выдает управляющий сигнал, являющийся суммой двух слагаемых, первое пропорционально отклонению, второе пропорционально интегралу отклонения. Логика работы соответствует следующему рисунку:



Если параметр *Ti* равен нулю, вычисления интегральной части не выполняются, и регулятор будет работать, как пропорционально-дифференциальный регулятор (ПД-регулятор). Если установленные значения всех параметров отличны от нуля, конечная функция будет представлять сумму двух воздействий, каждое из которых имеет различное регулирование. *Пропорциональное регулирование* выдает на выход значение, пропорциональное измеренной ошибке. Это функция отклонения, чем больше коэффициент усиления, тем меньше ошибка.

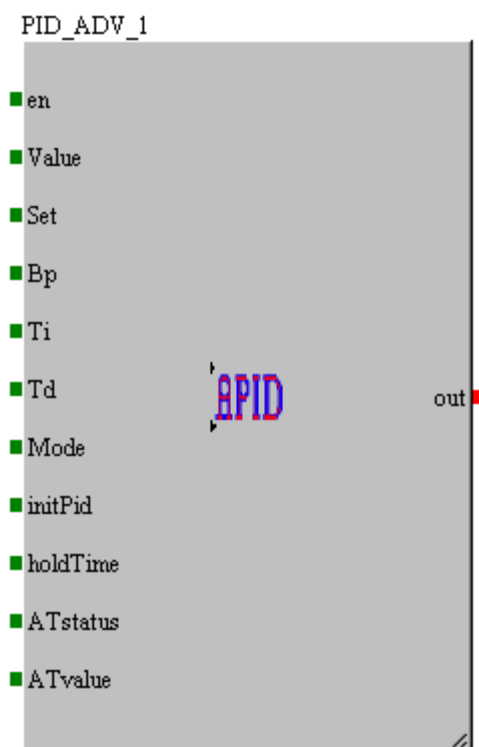
*Интегральное регулирование* используется для выведения на выходе регулятора заданного значения и уменьшения отклонения, вызванного пропорциональным регулированием. Выходной параметр двух компонентов создаст условие, в процентных точках с двумя десятичными знаками, которое будет отвечать величине сигнала, выработанного в качестве гарантии того, что выход возвращен ‘ближе’ к установленному значению (параметр *Set*). Параметр *Mode* разделяет регулятор на прямое (*Mode=0*) или обратное (*Mode=1*) регулирование системы контроля. Например:

- Mode = 1*, если выбрано управление процессом нагрева;
- Mode = 0*, если выбрано управления процессом охлаждения.

### Замечания

Для исправной работы библиотеки, параметры *Bp* и *Ti* установлены на базис динамических характеристик контролируемой системы (например, при использовании одного из различных ПИ-методов авто настройки).

### PID\_ADV



Вход	Тип	Границы	Описание
<i>en</i>	CJ_BIT	0-1	Активировать ПИД-регулятор
<i>Value</i>	CJ_SHORT	-32768..32767	Регулируемое значение.
<i>Set</i>	CJ_SHORT	-32768..32767	Заданное значение.
<i>Bp</i>	CJ_WORD	0..65535	Зона пропорционального регулирования.
<i>Ti</i>	CJ_WORD	0..999	Время интегрирования (в секундах).



<i>Td</i>	CJ_WORD	0..999	Время дифференцирования (в секундах).
<i>Mode</i>	CJ_BIT	0-1	Режим работы: 0 = Прямое действие (охлаждение) 1 = Обратное действие (нагрев)
<i>initPid</i>	CJ_WORD	0..10000 [0.00 .. 100.00]	Процентное выходное начальное значение, до двух знаков после запятой
<i>holdTime</i>	CJ_WORD	0..65535	Тайм-аут в секундах для поддержания выходного сигнала на начальном значении, до включения регулирования.
<i>ATstatus</i>	CJ_BYTE	0-3	Этот вывод должен быть подключен к выходу ATstatus в библиотеке PID_AT. Возможные значения: 0 = Автонастройка отключена 1 = Автонастройка выполняется 2 = Автонастройка ОК 3 = Автонастройка не выполнена
<i>ATvalue</i>	CJ_WORD	0..10000	Этот вывод должен быть подключен к выводу ATvalue в библиотеке PID_AT.

Выход	Тип	Границы	Описание
<i>out</i>	CJ_WORD	0..10000 [0.00 .. 100.00]	Вывод в процентах, до двух знаков после запятой

### Описание

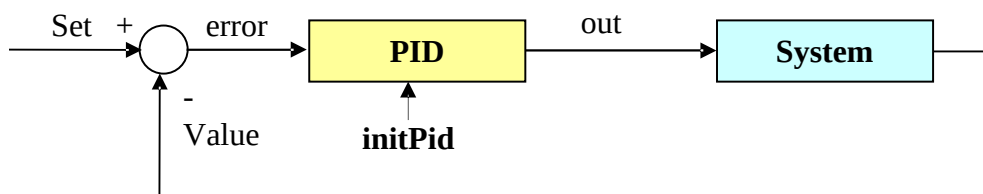
Эта библиотека реализует классический ПИД-регулятор, но с некоторыми доп. функциями:

Пока ПИД-регулятор не активирован (вход *en* = 0), выход остается равным 0.

Когда ПИД-регулятор включен (вход *en* = 1), выходное значение равно значению, представленному на входе *initPid*.

Если вход *HoldTime* не подключен или его значение равно 0, то ПИД-регулятор начинает работу с этого значения, в противном случае он ожидает истечения времени *HoldTime*.

Логика регулятора соответствует следующему рисунку:



Для получения пояснений по ПИД-регулированию и их параметрам обратитесь к документации библиотеки ПИД-регуляторов.

Вход *en* необязательный: если он не подключен, функциональный блок всегда включен.

Входы *initPid* и *holdTime* необязательные: если они не подключены, их значения равны 0.

Параметр *Mode* различает регулятор на прямое (*Mode* = 0) или обратное (*Mode* = 1) управление системой. Например:

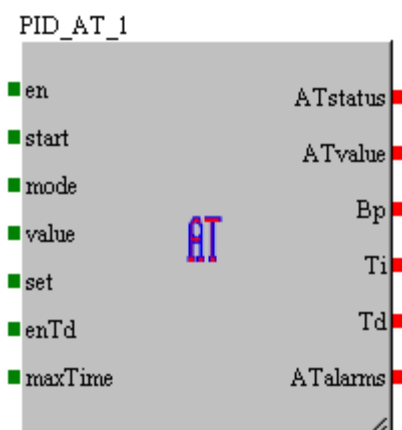
*Mode* = 1, если нужно контролировать процесс нагрева  
*Mode* = 0, если требуется контролировать процесс охлаждения

### Замечания

Для корректной работы библиотеки параметры *Bp* и *Ti* задаются на основе динамических характеристик управляемой системы (например, с помощью одного из различных методов автонастройки PI).

Эту библиотеку можно дополнить функцией *Autotuning*, соединив ее каскадом с библиотекой PID\_AT.

### PID\_AT



Вход	Tun	Границы	Описание
<i>en</i>	CJ_BIT	0-1	Включить процедуру автонастройки
<i>start</i>	CJ_BIT	0-1	Запуск процедуры (триггерный вход)
<i>mode</i>	CJ_BIT	0-1	Режим работы 0 = Прямое действие (охлаждение) 1 = Обратное действие (нагрев)
<i>value</i>	CJ_SHORT	-32768..32767	Контролируемое значение
<i>set</i>	CJ_SHORT	-32768..32767	Значение задания
<i>enTd</i>	CJ_BIT	0-1	Включить производное действие
<i>maxTime</i>	CJ_WORD	2-1440	Максимальное время (в минутах) расчета параметров ПИД-регулятора, после которого автонастройка останавливается с ошибкой. По умолчанию 30 минут.

Выход	Tun	Границы	Описание
-------	-----	---------	----------

<i>ATstatus</i>	CJ_BYTE	0-3	Этот вывод должен быть подключен к ATstatus библиотеки PID_ADV. Возможные значения: 0 = Автонастройка отключена 1 = Автонастройка выполняется 2 = Автонастройка успешно завершена 3 = Автонастройка не удалась
<i>ATvalue</i>	CJ_WORD	0..10000	Выходная величина автонастройки. Этот вывод должен быть подключен к входу ATvalue библиотеки PID_ADV.
<i>Pp</i>	CJ_WORD	0..30000	Рассчитанная зона пропорциональности
<i>Ti</i>	CJ_WORD	0..999	Рассчитанное интегральное время (сек)
<i>Td</i>	CJ_WORD	0..999	Рассчитанное дифференциал. время (сек)
<i>ATalarms</i>	CJ_WORD	0-65535	Дается описание возможной причины сбоя операции автонастройки. <i>бит0</i> : время с момента запуска превысило <i>maxTime</i> , не удалось рассчитать новые параметры <i>бит1</i> : время цикла слишком велико (2 с) <i>бит2</i> : изменение режима или установка параметров во время автонастройки <i>бит3</i> : были измерены недопустимые значения времени для расчета параметров <i>бит4</i> : расчетные значения параметров <i>Pp</i> , <i>Ti</i> и/или <i>Td</i> находятся за пределами допустимого диапазона

### Описание

Эта библиотека измеряет динамический отклик управляемой системы и автоматически рассчитывает параметры ПИД-регулятора: зону пропорциональности *Pp*, время интегрирования *Ti* и, если разрешено, время дифференцирования *Td*.

Эта библиотека должна быть подключена каскадом к библиотеке PID\_ADV.

Автонастройка основана на реализации вынужденных колебаний (предельных циклов) вокруг заданного значения. В конце 3 предельных циклов, если условия расчета достаточны, рассчитываются новые значения *Pp*, *Ti* и *Td*, и автонастройка проходит успешно. В противном случае диагностируется тип ошибки, помешавшей расчету.

Подготовка системы перед запуском автонастройки:

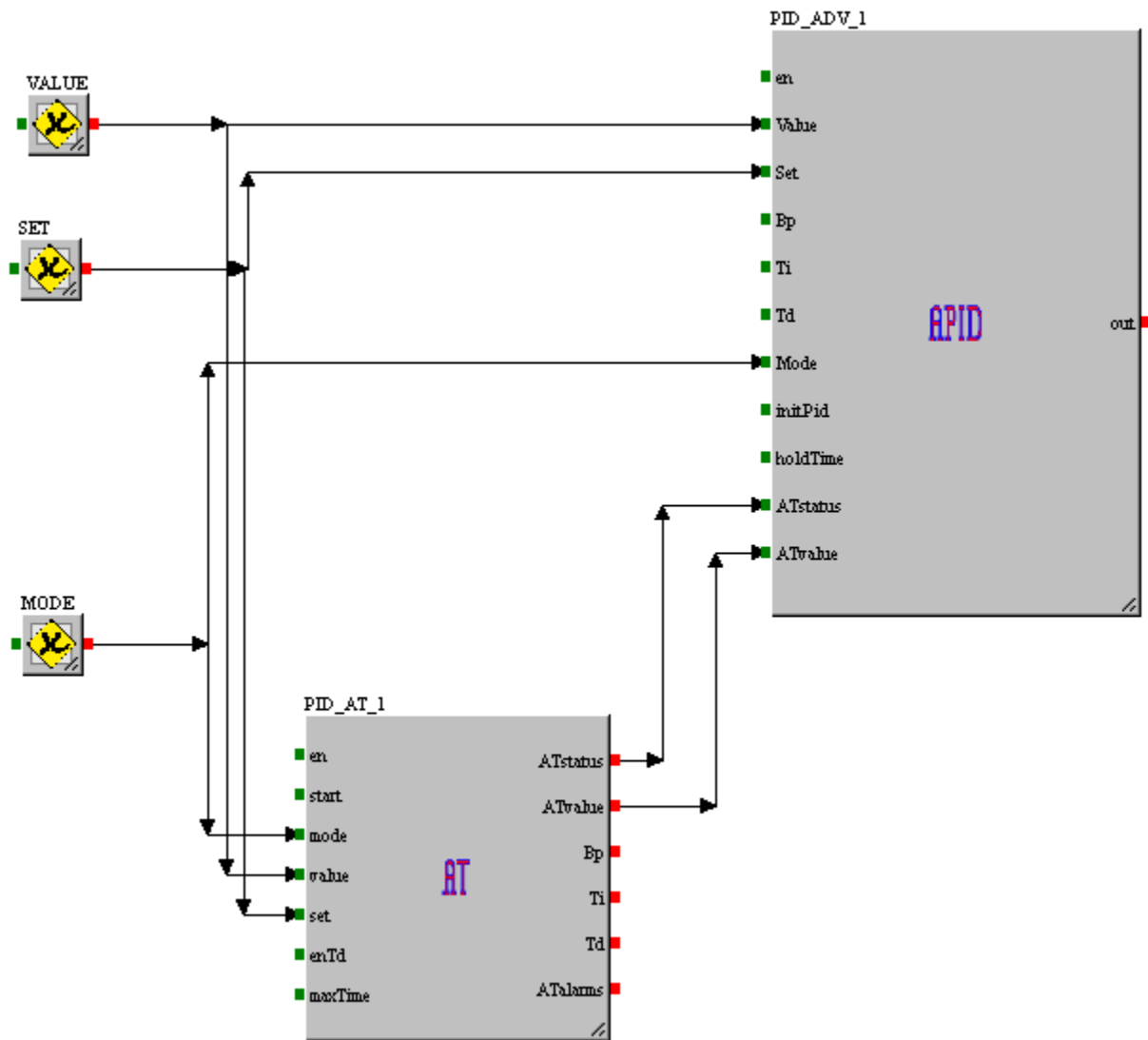
Подготовьте соединения между PID\_ADV и PID\_AT, чтобы они использовали один набор значений и режим. Кроме того, ATstatus и ATvalue PID\_AT должны быть подключены к соответствующим входам ATstatus и ATvalue PID\_ADV (см. рис. ниже).

Убедитесь, что система, управляемая ПИД-регулятором, нестабильна или значение сигнала не слишком сильно колеблется. Возможны фильтрация или опосредование сигнала. Обратите внимание на уставку, с которой вы хотите выполнить автонастройку, в процессе настройки может быть опасно превышать минимальное или максимальное значение уставки, что может привести к непредвиденным последствиям.

Выполняйте эту процедуру настройки в условиях максимальной безопасности.

Только в случае успеха (ATstatus = ОК) скопируйте рассчитанные значения *Pp*, *Ti* и *Td* в соответствующие параметры управления PID\_ADV.

Полный пример см. в проекте PID\_AT test.ucjp, поставляемом с UniPro.

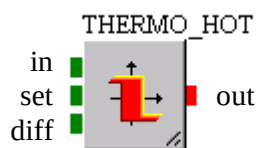


Для объяснения ПИД-управления и параметров есть документация библиотеки ПИД. Параметр *en* является необязательным: если он не подключен, блок всегда активен. Параметр *maxTime* является необязательным: если он не подключен, его максимальное время по умолчанию составляет 30 минут. Параметр *Mode* определяет прямое (*Mode* = 0) или обратное (*Mode* = 1) управление.  
*Mode* = 1, если нужно контролировать процесс нагрева  
*Mode* = 0, если требуется контролировать процесс охлаждения

### Замечания

Функция автонастройки из в этой библиотеки, связана только с библиотекой PID\_ADV.

## THERMO\_HOT



Вход	Tun	Границы	Описание
<i>in</i>	CJ_SHORT	-32768..32767	Входное значение
<i>set</i>	CJ_SHORT	-32768..32767	Заданное значение
<i>diff</i>	CJ_SHORT	-32768..32767	Дифференциал

Выход	Tun	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Состояние регулятора

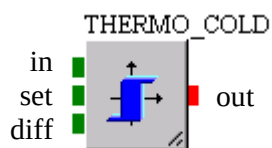
**Описание**

Библиотека выполняет функцию двухпозиционного регулятора обратного действия, т.е.:  
 Если значение входа *in*, меньше значения установки *set* с вычтенным дифференциалом *diff*, выход (*out*=1) активируется.

Если значение входа *in* больше или равно значению установки *set*, выход (*out*=1) деактивируется.

Например, эта библиотека может быть использована для регулирования процесса нагрева, а именно, термостата с гистерезисом ниже значения установки.

## THERMO\_COLD



Вход	Tun	Границы	Описание
<i>in</i>	CJ_SHORT	-32768..32767	Входное значение
<i>set</i>	CJ_SHORT	-32768..32767	Заданное значение
<i>diff</i>	CJ_SHORT	-32768..32767	Дифференциал

Выход	Tun	Границы	Описание
<i>out</i>	CJ_BIT	0-1	Состояние регулятора

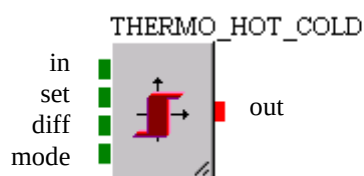
**Описание**

Библиотека выполняет функцию двухпозиционного регулятора прямого действия, т.е.:  
 Если значение входа *in* больше, чем сумма значения установки *set* и дифференциала *diff*, выход (*out=1*) активируется.

Если значение входа *in* меньше или равно значению установки *set*, выход (*out=0*) деактивируется.

Например, эта библиотека может быть использована для регулирования процесса охлаждения, а именно, термостата с гистерезисом выше значения установки.

**THERMO\_HOT\_COLD**



<b>Вход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>in</i>	CJ_SHORT	-32768..32767	Входное значение
<i>set</i>	CJ_SHORT	-32768..32767	Заданное значение
<i>diff</i>	CJ_SHORT	-32768..32767	Дифференциал
<i>mode</i>	CJ_BIT	0-1	Режим работы 0 = Прямое действие 1 = Обратное действие

<b>Выход</b>	<b>Tun</b>	<b>Границы</b>	<b>Описание</b>
<i>out</i>	CJ_BIT	0-1	Состояние регулятора

**Описание**

Библиотека разрешает выполнение как прямого, так и обратного регулирования, с помощью изменения входа *mode*. Прямое действие с *mode = 0*.

Если значение входа *in* больше, чем сумма значения установки *set* и дифференциала *diff*, выход (*out=1*) активируется.

Если значение входа *in* меньше или равно значению установки *set*, выход (*out=0*) деактивируется.

Регулирование прямого действия необходимо в случае термостата с гистерезисом, выше значения установки, например для процесса охлаждения.

Обратное действие с *mode=1*.

Если значение входа *in*, меньше значения установки *set* с вычтенным дифференциалом *diff*, выход (*out=1*) активируется.

Если значение входа *in* больше или равно значению установки *set*, выход (*out=1*) деактивируется.

Регулирование обратного действия необходимо в случае термостата с гистерезисом, ниже значения установки, например для процесса нагрева.

### 3.11 Selection

#### MAX\_BIT



Вход	Тип	Границы	Описание
IN1	CJ_BIT	0-1	Вход данных 1
IN2	CJ_BIT	0-1	Вход данных 2

Выход	Тип	Границы	Описание
OUT	CJ_BIT	0-1	Выход значения

#### Описание

Сравнивает входы *IN1* и *IN2* и выдает наибольшее значение.

#### MAX\_BYTE



Вход	Тип	Границы	Описание
IN1	CJ_BYTE	0..255	Вход данных 1
IN2	CJ_BYTE	0..255	Вход данных 2

Выход	Тип	Границы	Описание
OUT	CJ_BYTE	0..255	Выход значения

#### Описание

Сравнивает входы *IN1* и *IN2* и выдает наибольшее значение.



**MAX\_DWORD**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_DWORD	0..4294967295	Вход данных 1
<i>IN2</i>	CJ_DWORD	0..4294967295	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_DWORD	0..4294967295	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наибольшее значение.

**MAX\_LONG**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_LONG	-2147483648.. 2147483647	Вход данных 1
<i>IN2</i>	CJ_LONG	-2147483648.. 2147483647	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_LONG	-2147483648.. 2147483647	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наибольшее значение.

**MAX\_SHORT**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_SHORT	-32768..32767	Вход данных 1
<i>IN2</i>	CJ_SHORT	-32768..32767	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_SHORT	-32768..32767	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наибольшее значение.

**MAX\_S\_BYTE**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_S_BYTE	-128..127	Вход данных 1
<i>IN2</i>	CJ_S_BYTE	-128..127	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_S_BYTE	-128..127	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наибольшее значение.

**MAX\_WORD**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_WORD	0..65535	Вход данных 1
<i>IN2</i>	CJ_WORD	0..65535	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_WORD	0..65535	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наибольшее значение.

**MIN\_BIT**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_BIT	0-1	Вход данных 1
<i>IN2</i>	CJ_BIT	0-1	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_BIT	0-1	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наименьшее значение.

**MIN\_BYTE**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_BYTE	0..255	Вход данных 1
<i>IN2</i>	CJ_BYTE	0..255	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_BYTE	0..255	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наименьшее значение.

**MIN\_DWORD**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_DWORD	0..4294967295	Вход данных 1
<i>IN2</i>	CJ_DWORD	0..4294967295	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_DWORD	0..4294967295	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наименьшее значение.

**MIN\_LONG**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_LONG	-2147483648.. -2147483647	Вход данных 1
<i>IN2</i>	CJ_LONG	-2147483648.. -2147483647	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_LONG	-2147483648.. -2147483647	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наименьшее значение.

**MIN\_SHORT**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_SHORT	-32768..32767	Вход данных 1
<i>IN2</i>	CJ_SHORT	-32768..32767	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_SHORT	-32768..32767	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наименьшее значение.

**MIN\_S\_BYTE**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_S_BYTE	-128..127	Вход данных 1
<i>IN2</i>	CJ_S_BYTE	-128..127	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_S_BYTE	-128..127	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наименьшее значение.

**MIN\_WORD**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_WORD	0..65535	Вход данных 1
<i>IN2</i>	CJ_WORD	0..65535	Вход данных 2

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_WORD	0..65535	Выход значения

**Описание**

Сравнивает входы *IN1* и *IN2* и выдает наименьшее значение.

## SEL\_BIT



Вход	Тип	Границы	Описание
IN1	CJ_BIT	0-1	Вход данных 1
IN2	CJ_BIT	0-1	Вход данных 2
G	CJ_BIT	0-1	Вход выбора

Выход	Тип	Границы	Описание
OUT	CJ_BIT	0-1	Выход значения

**Описание**

Если цифровой вход *G* установлен на “0”, то выход *OUT* примет значение *IN1*, иначе примет значение *IN2*.

## SEL\_BYTE



Вход	Тип	Границы	Описание
IN1	CJ_BYTE	0..255	Вход данных 1
IN2	CJ_BYTE	0..255	Вход данных 2
dd	CJ_BIT	0-1	Вход выбора

Выход	Тип	Границы	Описание
OUT	CJ_BYTE	0..255	Выход значения

**Описание**

Если цифровой вход *G* установлен на “0”, то выход *OUT* примет значение *IN1*, иначе примет значение *IN2*.

**SEL\_DWORD**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_DWORD	0..4294967295	Вход данных 1
<i>IN2</i>	CJ_DWORD	0..4294967295	Вход данных 2
<i>dd</i>	CJ_BIT	0-1	Вход выбора

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_DWORD	0..4294967295	Выход значения

**Описание**

Если цифровой вход *G* установлен на “0”, то выход *OUT* примет значение *IN1*, иначе примет значение *IN2*.

**SEL\_LONG**

<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_LONG	-2147483648..2147483647	Вход данных 1
<i>IN2</i>	CJ_LONG	-2147483648..2147483647	Вход данных 2
<i>dd</i>	CJ_BIT	0-1	Вход выбора

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_LONG	-2147483648..2147483647	Выход значения

**Описание**

Если цифровой вход *G* установлен на “0”, то выход *OUT* примет значение *IN1*, иначе примет значение *IN2*.



## SEL\_SHORT



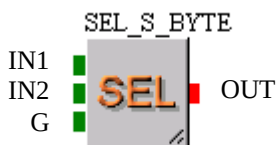
Вход	Тип	Границы	Описание
IN1	CJ_SHORT	-32768..32767	Вход данных 1
IN2	CJ_SHORT	-32768..32767	Вход данных 2
dd	CJ_BIT	0-1	Вход выбора

Выход	Тип	Границы	Описание
OUT	CJ_SHORT	-32768..32767	Выход значения

**Описание**

Если цифровой вход *G* установлен на “0”, то выход *OUT* примет значение *IN1*, иначе примет значение *IN2*.

## SEL\_S\_BYTE



Вход	Тип	Границы	Описание
IN1	CJ_S_BYTE	-128..127	Вход данных 1
IN2	CJ_S_BYTE	-128..127	Вход данных 2
dd	CJ_BIT	0-1	Вход выбора

Выход	Тип	Границы	Описание
OUT	CJ_S_BYTE	-128..127	Выход значения

**Описание**

Если цифровой вход *G* установлен на “0”, то выход *OUT* примет значение *IN1*, иначе примет значение *IN2*.

## SEL\_WORD



<b>Вход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>IN1</i>	CJ_WORD	0..65535	Вход данных 1
<i>IN2</i>	CJ_WORD	0..65535	Вход данных 2
<i>dd</i>	CJ_BIT	0-1	Вход выбора

<b>Выход</b>	<b>Тип</b>	<b>Границы</b>	<b>Описание</b>
<i>OUT</i>	CJ_WORD	0..65535	Выход значения

**Описание**

Если цифровой вход *G* установлен на “0”, то выход *OUT* примет значение *IN1*, иначе примет значение *IN2*.

### 3.12 Timers

#### TOF (Off Delay Timer)



Вход	Тип	Границы	Описание
PT	CJ_WORD	0..65535	Текущее время. 16-битное значение - задержка времени с выключением сигнала (100 мс для TOFmsec и сек для TOFsec).
IN	CJ_BIT	0-1	Включение таймера. Когда задний фронт будет обнаружен, таймер включается. Если передний фронт будет обнаружен раньше, чем внутренний таймер достигнет значения PT, тогда выход Q выключен не будет.

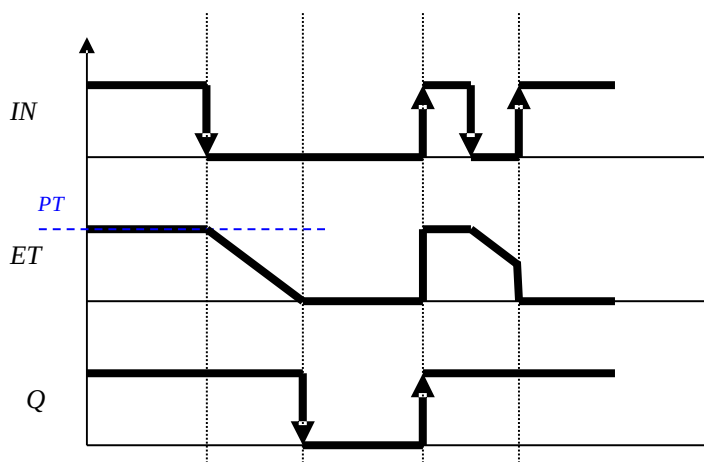
Выход	Тип	Границы	Описание
ET	CJ_WORD	0..65535	Время работы. Показывает текущее значение времени работы (сотни мс для TOFmsec и секунды для TOFsec).
Q	CJ_BIT	0-1	Выход сигнала. Показывает достижение заданного времени. Принимает значение '0', если PT=ET, иначе принимает значение '1'.

#### Описание

Библиотека TOF устанавливает задержку времени с выключением сигнала, например, выключение вентилятора устройства по истечению времени от выключения устройства.

#### Замечания

Пожалуйста, обратитесь к следующему графику.



### TON (On Delay Timer)



Вход	Tun	Границы	Описание
PT	CJ_WORD	0..65535	Текущее время. 16-битное значение - задержка времени с включением сигнала (100 мс для TONmsec и секунды для TONsec).
IN	CJ_BIT	0-1	Включение таймера. Когда передний фронт будет обнаружен, внутренний таймер включается.

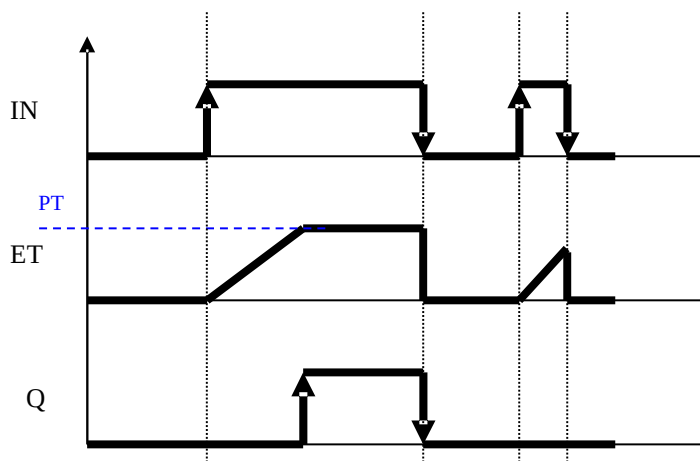
Выход	Tun	Границы	Описание
ET	CJ_WORD	0..65535	Время работы. Показывает текущее значение времени работы (100 мс для TONmsec и секунды для TONsec).
Q	CJ_BIT	0-1	Выход сигнала. Показывает достижение заданного времени. Принимает значение '1', если $PT=ET$ , иначе принимает значение '0'.

**Описание**

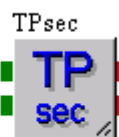
Библиотека TON устанавливает задержку времени с включением сигнала, например, включение компрессора по истечению времени с последующим сигналом включения.

**Замечания**

Пожалуйста, обратитесь к следующему графику.



**TP (Pulse Timer)**



Вход	Tun	Границы	Описание
PT	CJ_WORD	0..65535	Текущее время. 16-битное значение, которое определяет время, подсчитанное счетчиком (сотни мс для TPmsec и секунды для TPsec).
IN	CJ_BIT	0-1	Импульсный таймер. При каждом передним фронте на входе IN, таймер загружает значение PT. Если передний фронт обнаружен за время подсчета, таймер не загружает значение.

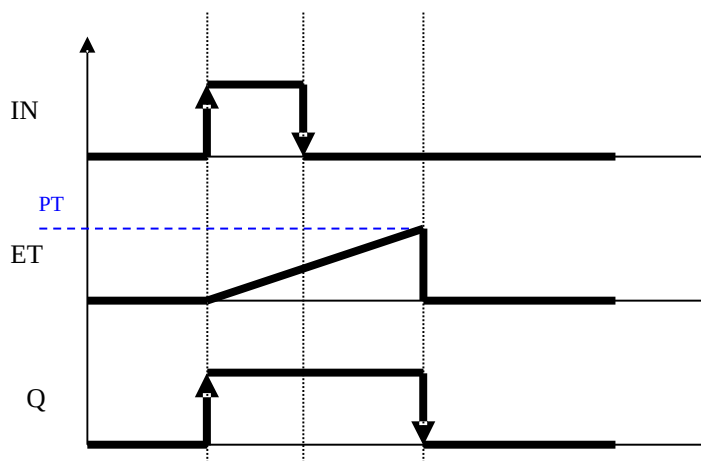
Выход	Тип	Границы	Описание
ET	CJ_WORD	0-1	Время работы. Показывает текущее значение времени работы (сотни мс для TPmsec и секунды для TPsec).
Q	CJ_BIT	0-1	Выход сигнала. Выход Q принимает значение '1' на время TP, пока не будет обнаружен новый передний фронт на входе IN.

### Описание

Библиотека TP позволяет устанавливать время импульса, например, время работы индикатора.

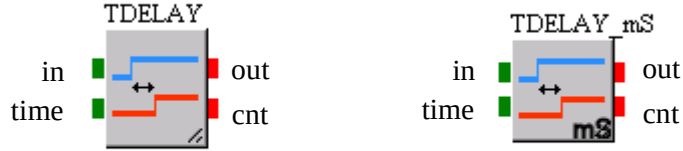
### Замечания

Пожалуйста, обратитесь к следующему графику.



### 3.13 Timing

#### TDELAY



Вход	<i>Tun</i>	Границы	Описание
<i>in</i>	CJ_BIT	0-1	Входной сигнал.
<i>time</i>	CJ_WORD	0..65535	Время задержки (стандартное в секундах для TDELAY и в 100 мс для TDELAY_mS).

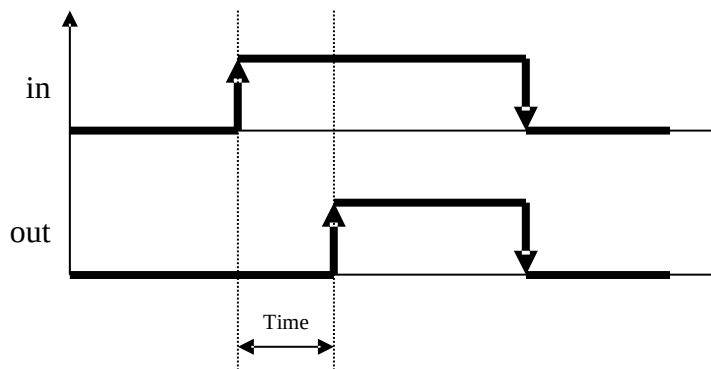
Выход	<i>Tun</i>	Границы	Описание
<i>cnt</i>	CJ_WORD	0..65535	Отсчет времени после активации задержки.
<i>out</i>	CJ_BIT	0-1	Выходной сигнал.

#### Описание

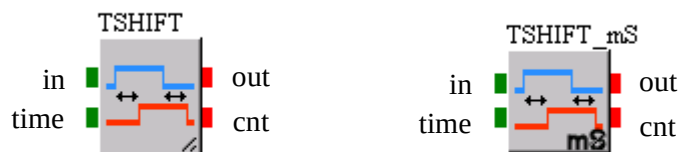
Библиотеки TDELAY и TDELAY\_mS позволяют установить задержку входного сигнала. Передний фронт выходного сигнала ожидает время *time* относительно входного сигнала.

#### Замечания

Пожалуйста, обратитесь к следующему графику.



## TSHIFT



Вход	<i>Tun</i>	Границы	Описание
<i>in</i>	CJ_BIT	0-1	Входной сигнал.
<i>time</i>	CJ_WORD	0..65535	Время задержки (стандартное в секундах для TSHIFT и в 100 мс для TSHIFT_mS).

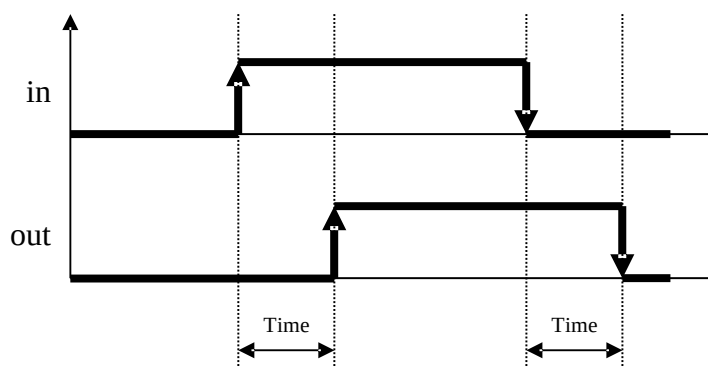
Выход	<i>Tun</i>	Границы	Описание
<i>cnt</i>	CJ_WORD	0..65535	Отсчет времени после вкл. задержки .
<i>out</i>	CJ_BIT	0-1	Выходной сигнал.

**Описание**

Библиотеки TSHIFT и TSHIFT\_mS позволяют установить задержку входного сигнала. Оба фронта (передний и задний) выходного сигнала будут переведены в продолжительность времени *time* относительно входного сигнала.

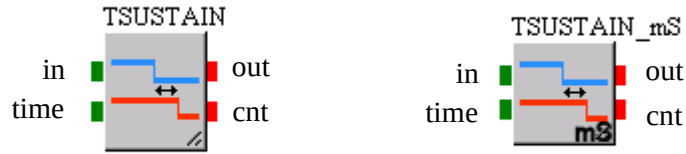
**Замечания**

Пожалуйста, обратитесь к следующему графику.





**TSUSTAIN**



Вход	<i>Tun</i>	Границы	Описание
<i>in</i>	CJ_BIT	0-1	Входной сигнал.
<i>time</i>	CJ_WORD	0..65535	Время задержки (стандартное в секундах для TSUSTAIN и в 100 мс для TSUSTAIN_mS).

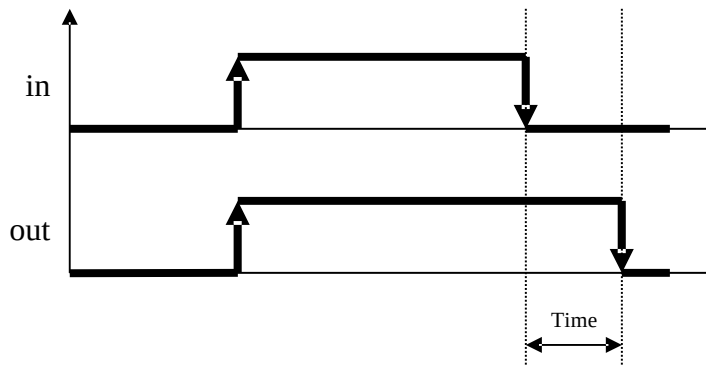
Выход	<i>Tun</i>	Границы	Описание
<i>cnt</i>	CJ_WORD	0..65535	Отсчет времени после вкл. задержки .
<i>out</i>	CJ_BIT	0-1	Выходной сигнал.

**Описание**

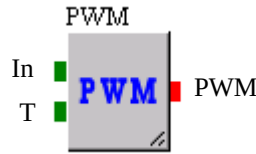
Библиотеки TSUSTAIN и TSUSTAIN\_mS позволяют установить задержку входного сигнала. Задний фронт выходного сигнала ожидает время *time* относительно входного сигнала.

**Замечания**

Пожалуйста, обратитесь к следующему графику.



## PWM



Вход	<i>Tup</i>	Границы	Описание
<i>In</i>	CJ_WORD	0..10000	Вход в % с двумя знаками после точки.
<i>T</i>	CJ_BYTE	0..255	Период времени функции (в секундах).

Выход	<i>Tup</i>	Границы	Описание
<i>PWM</i>	CJ_BIT	0-1	Выход функции.

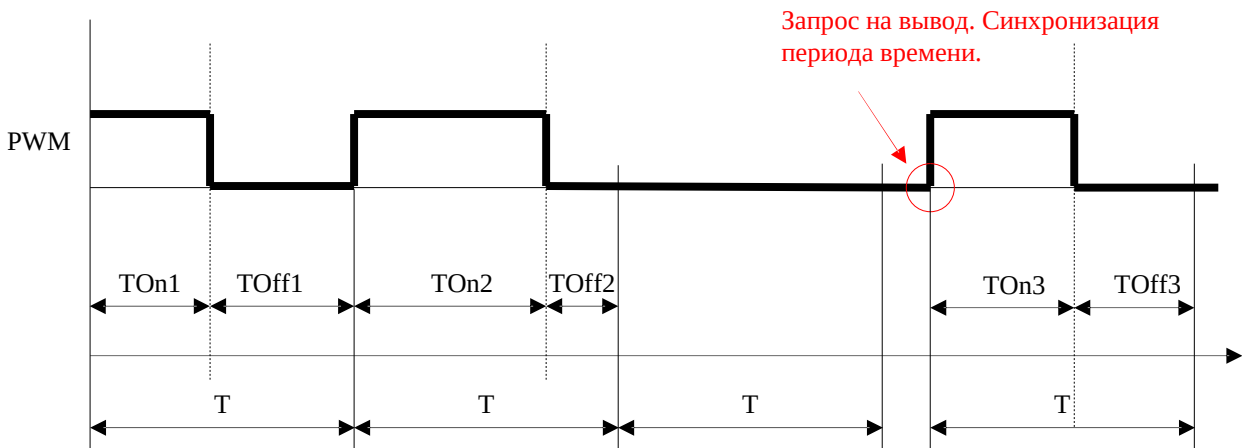
**Описание**

Библиотека PWM (ШИМ) использует два входа для вычисления двоичного выходного сигнала (вкл/выкл). Эта периодическая функция с периодом времени  $T$ . Она обеспечивает модулированный по времени выход, основанный на значении входа  $In$ . Этот вход определяет процент периода времени  $T$ , который выход PWM будет с логической частью '1' (вкл.). Оставшийся период времени выход будет с логической частью '0' ( $T_{off}=T-T_{on}$ ). Выход может принимать логическое значение '1' только один раз в течение каждого периода. Если значение  $In$  изменится с нуля (выход временно отключен) на какое-либо другое значение, начало периода синхронизируется заново и выход будет включен немедленно.

**Замечания**

Внутренние счетчика управляются с 100мс прерыванием. Рекомендуется установить логический показатель *timed*, связанный с выходом PWM библиотеки (например, DIGITALOUT) на "Timed 100mS".

Пожалуйста, обратитесь к следующему графику.



## UNI-PRO РУКОВОДСТВО ПО СТАНДАРТНЫМ БИБЛИОТЕКАМ

UNI-PRO — Руководство по стандартным библиотекам.

Версия 4.4 - Апрель 2022.

Код 114UPROSLE44.

Файл 114UPROSLE44.pdf.

Данная публикация является исключительной собственностью Evco. Копирование и воспроизведение материалов в любой форме без предварительного разрешения Evco запрещено. Evco не несёт ответственности за характеристики, техническую информацию и другие ошибки, представленные в данной публикации, а также за последствия их использования. Evco не несёт ответственности за нарушения, вызванные несоблюдением мер предосторожности. Компания сохраняет за собой право вносить изменения в ходе технических разработок в любое время без предварительного уведомления при отсутствии значительных изменений, касающихся функционала или безопасности.



#### **ГЛАВНЫЙ ОФИС**

##### **Evco**

Via Mezzaterra 6, 32036 Sedico Belluno ITALY  
Tel. +39 0437-852468  
Fax +39 0437-83648  
info@evco.it  
www.evco.it

#### **ПРЕДСТАВИТЕЛЬСТВА КОМПАНИИ**

##### **Control France**

155 Rue Roger Salengro, 92370 Chaville Paris FRANCE  
Tel. 0033-1-41159740  
Fax 0033-1-41159739  
control.france@wanadoo.fr

##### **Evco Latina**

Larrea, 390 San Isidoro, 1609 Buenos Aires ARGENTINA  
Tel. 0054-11-47351031  
Fax 0054-11-47351031  
evcolatina@anykasrl.com.ar

##### **Evco Pacific**

59 Premier Drive Campbellfield, 3061, Victoria Melbourne, AUSTRALIA  
Tel. 0061-3-9357-0788  
Fax 0061-3-9357-7638  
everycontrol@pacific.com.au

##### **Evco Russia**

111141 Russia Moscow 2-oy Proezd Perova Polya 9  
Tel. 007-495-3055884  
Fax 007-495-3055884  
info@evco.ru

##### **Every Control do Brasil**

Rua Marino Félix 256, 02515-030 Casa Verde São Paulo SÃO PAULO BRAZIL  
Tel. 0055-11-38588732  
Fax 0055-11-39659890  
info@everycontrol.com.br

##### **Every Control Norden**

Cementvägen 8, 136 50 Haninge SWEDEN  
Tel. 0046-8-940470  
Fax 0046-8-6053148  
mail2@unilec.se

##### **Every Control Shangai**

B 302, Yin Hai Building, 250 Cao Xi Road, 200235 Shangai CHINA  
Tel. 0086-21-64824650  
Fax 0086-21-64824649  
evcosh@online.sh.cn

##### **Every Control United Kingdom**

Unit 19, Monument Business Park, OX44 7RW Chalgrove, Oxford, UNITED KINGDOM  
Tel. 0044-1865-400514  
Fax 0044-1865-400419  
info@everycontrol.co.uk