

2018



СПК

Системное время

Руководство для начинающих пользователей

Версия: 1.1
Дата: 20.02.2018



Оглавление

1. Цель и структура документа	3
2. Установка системного времени в Конфигураторе СПК	4
3. Отображение системного времени в визуализации.....	6
4. Типы переменных хранения времени	8
5. Описание библиотеки САА DTUtil	9
5.1. Добавление библиотеки в проект CODESYS.....	9
5.2. ФБ DTU.GetDateAndTime	10
5.3. ФБ DTU.SetDateAndTime.....	11
5.4. Функция DTU.DTSplit.....	12
5.5. Функция DTU.DTConcat	13
5.6. Функция DTU.GetDayOfWeek	14
6. Пример работы с системным временем.....	15
7. Синхронизация времени с NTP-сервером.....	25
8. Системный таймер ПЛК	25

1. Цель и структура документа

Данный документ посвящен вопросам работы с **системным временем** сенсорных панельных контроллеров **СПК** – отображению его в визуализации, считыванию в программу и изменению из программы. Для работы с системным временем в программе необходимо воспользоваться одной из системных библиотек. Наиболее простой и функциональной из них является **CAA DTUtil**. Документ включает пример работы с данной библиотекой.

Для однократной установки системного времени можно воспользоваться **Конфигуратором СПК** (см. [п. 2](#)).

Самый простой способ отобразить системное время в визуализации – использовать спецификатор формата вывода **%t[формат времени]** с заполнителями. Работа с ним описана в [п. 3](#).

Список типов переменных, используемых для хранения времени, приведен в [п. 4](#). Описание библиотеки приведено в [п. 5](#). Пример работы с библиотекой приведен в [п. 6](#).

В [п. 7](#) приведена информация о синхронизации времени по протоколу **NTP**.

В [п. 8](#) приведена информация о системном таймере контроллера.

2. Установка системного времени в Конфигураторе СПК

Для однократной установки системного времени можно воспользоваться **конфигуратором СПК**, который запускается из **сервисного меню**. Чтобы попасть в сервисное меню, необходимо после включения питания **до начала загрузки проекта** контроллера коснуться экрана три раза. Подробное описание сервисного меню и конфигуратора приведено в документе **СПК.FAQ**.

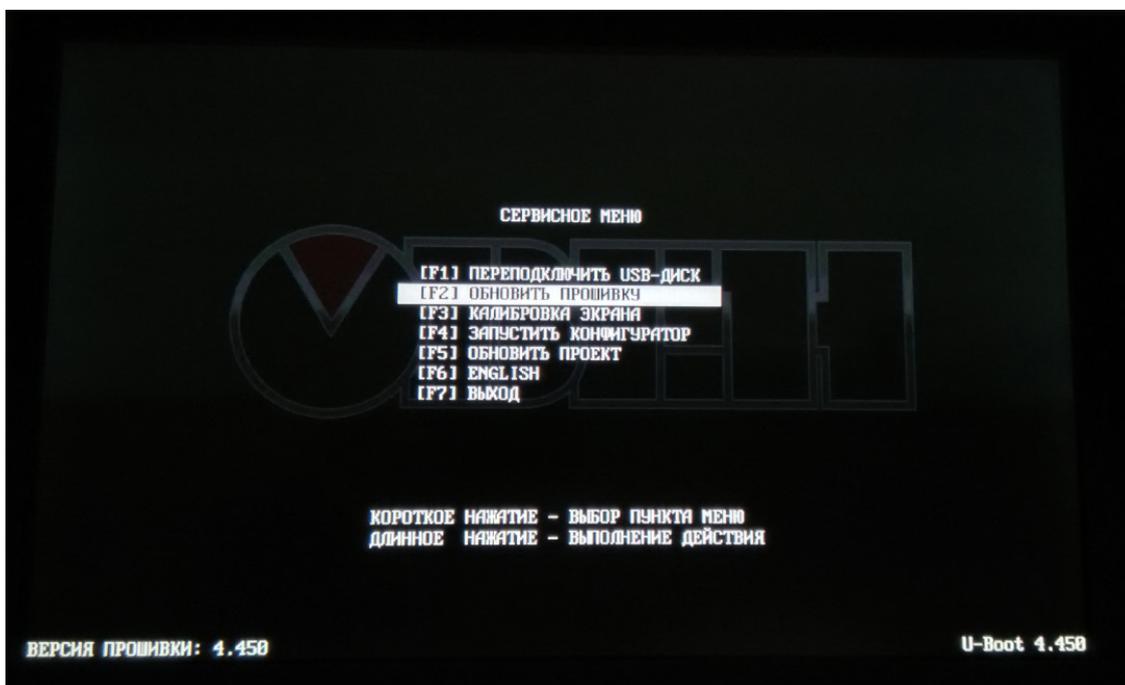


Рис. 1. Внешний вид сервисного меню

В конфигураторе СПК во вкладке **Дата и время** нажмите кнопку **Настроить**.

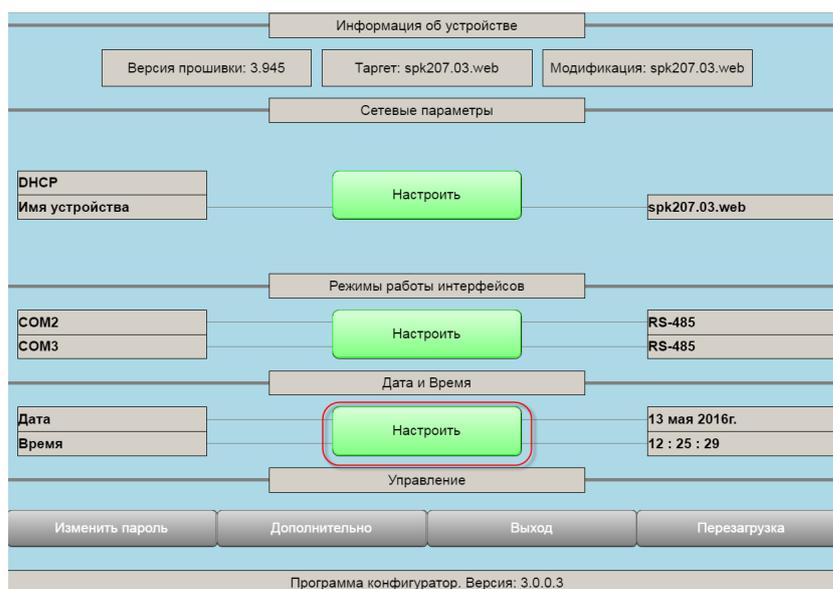


Рис. 2. Внешний вид конфигуратора СПК

В появившемся диалоговом окне измените системное время с помощью кнопок **-/+** или экранной клавиатуры (появляется после нажатия на любой из разрядов времени).

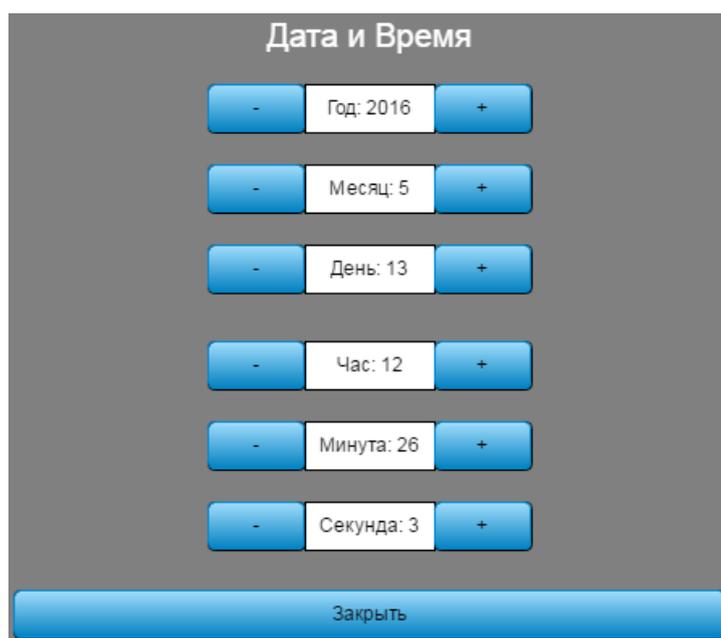


Рис. 3. Настройка системного времени в конфигураторе СПК

3. Отображение системного времени в визуализации

Для отображения системного времени в визуализации проще всего использовать один из базовых элементов (например, **Прямоугольник**) со спецификатором формата вывода **%t[формат времени]**. Этот спецификатор должен быть указан на вкладке **Тексты** в параметре **Текст** (см. рис. 4). **Обратите внимание**, что использование спецификатора **%t** без форматирования бесполезно.

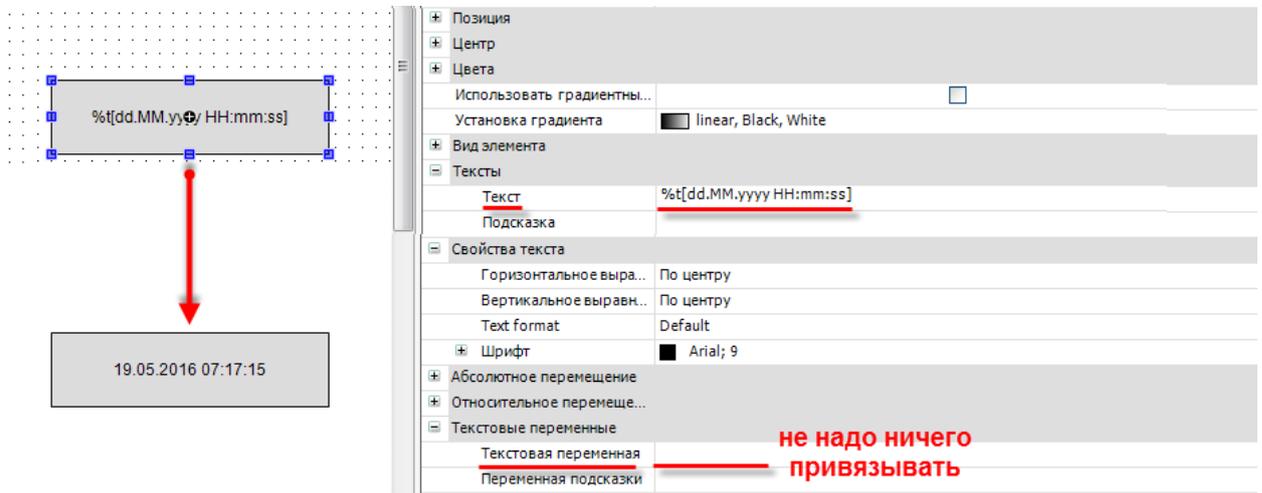


Рис. 4. Вывод системного времени с помощью спецификатора **%t[формат времени]**

Элемент будет отображать системное время только в том случае, если во вкладке **Тексты** значение параметра **Текстовая переменная** оставлено пустым. Если к элементу привязана какая-то переменная, то отображаться будет уже она. Например, это может быть использовано при необходимости отобразить время таймера типа **TON** и т.д.

Список заполнителей формата времени приведен в табл. 1:

Табл. 1. Заполнители формата времени

Заполнитель	Отображаемое значение	Пример отображения
ddd	Сокращенное название дня недели	Fri (пятница)
dddd	Полное имя дня недели	Monday (понедельник)
dddddd	День недели в виде числа	0 (воскресение), 1 (понедельник)
MMM	Сокращенное название месяца	Feb (февраль)
MMMM	Полное название месяца	February (февраль)
d	День в виде числа (1 – 31)	8
dd	День с ведущим нулем (01 – 31)	08
M	Месяц в виде числа (1 – 12)	8
MM	Месяц с ведущим нулем (01 – 12)	08
jjj	День в году с ведущим нулем (001-366)	253
y	Год века (0 – 99)	8
yy	Год века с ведущим нулем (00 – 99)	08
yyyy	Год	2008
HH	Час в 24-часовом формате (01– 24)	08
hh	Час в 12-часовом формате (01 – 12)	08 (и для 8-00, и для 20-00)
m	Минуты (0 – 59)	8
mm	Минуты с ведущим нулем (00 – 59)	08
s	Секунды (0 – 59)	8
ss	Секунды с ведущим нулем (00 – 59)	08
ms	Миллисекунды (0 – 999)	888
t	Идентификатор для 12-часового формата: A (часы <12) и P (часы >12)	A (8 часов)
tt	Идентификатор для 12-часового формата: AM (часы <12) и PM (часы >12)	PM (15 часов)

Если помимо времени необходимо отображать сопроводительный текст, то следует заключать этот текст в одиночные кавычки (например, %t[dd 'дни' hh 'часы']).

4. Типы переменных хранения времени

CODESYS содержит несколько типов данных, предназначенных для хранения времени. Для хранения **системного времени** обычно используется тип **DATE_AND_TIME**. Список доступных типов приведен в табл. 2:

Табл. 2. Типы данных хранения времени

Тип переменной	Размер	Описание	Пример объявления
TIME	32 бита	Время в формате <день-час-минута-секунда-миллисекунда> . Нет необходимости обязательно определять все составляющие, но присутствующие разряды должны сохранять логическую последовательность (т.е. формат <секунда-минута-час> неприемлем).	tVar:TIME:= t#10ms; tVar:TIME:=t#12h10m; tVar:TIME:=t#2d5h15m10s15ms;
DATE	32 бита	Дата в формате <год-месяц-день> . Значения типа DATE обрабатываются как DWORD и содержат время в секундах, начиная с 0 часов 1 января 1970 года .	dVar:DATE:= d#2016-05-13; dVar:DT:= d#2014-02-07;
TIME_OF_DAY (TOD)	32 бита	Время суток в формате <час:минута:секунда.миллисекунда> . Значения типа TIME_OF_DAY обрабатываются как DWORD и содержат время в миллисекундах, начиная с 0:0:0.000 .	todVar:TIME_OF_DAY:= tod#02:10:14.455; todVar:TOD:=tod#13:04:34.001;
DATE_AND_TIME (DT)	32 бита	Дата и время в формате <год-месяц-день-час:минута:секунда> . Значения типа DATE_AND_TIME обрабатываются как DWORD и содержат время в секундах, начиная с 0 часов 1 января 1970 года .	dtVar:DATE_AND_TIME:=dt#2015-05-06-15:36:30; dtVar:DT:=dt#2016-01-03-04:25:55;
LTIME	64 бита	Расширение типа TIME с поддержкой точности до наносекунд.	ltVar:LTIME:=1000d15h23m12s34ms 2us44ns;

5. Описание библиотеки CAA DTUtil

5.1. Добавление библиотеки в проект CODESYS

Библиотека **CAA DTUtil** используется для считывания в программу значения системного времени контроллера и изменения его из программы. Пример работы с библиотекой приведен в [п. 6.](#)

Для добавления библиотеки в проект **CODESYS** в **Менеджере библиотек** нажмите кнопку **Добавить** и выберите библиотеку **CAA DTUtil Extern**, расположенную в папке **Intern/CAA/System**.

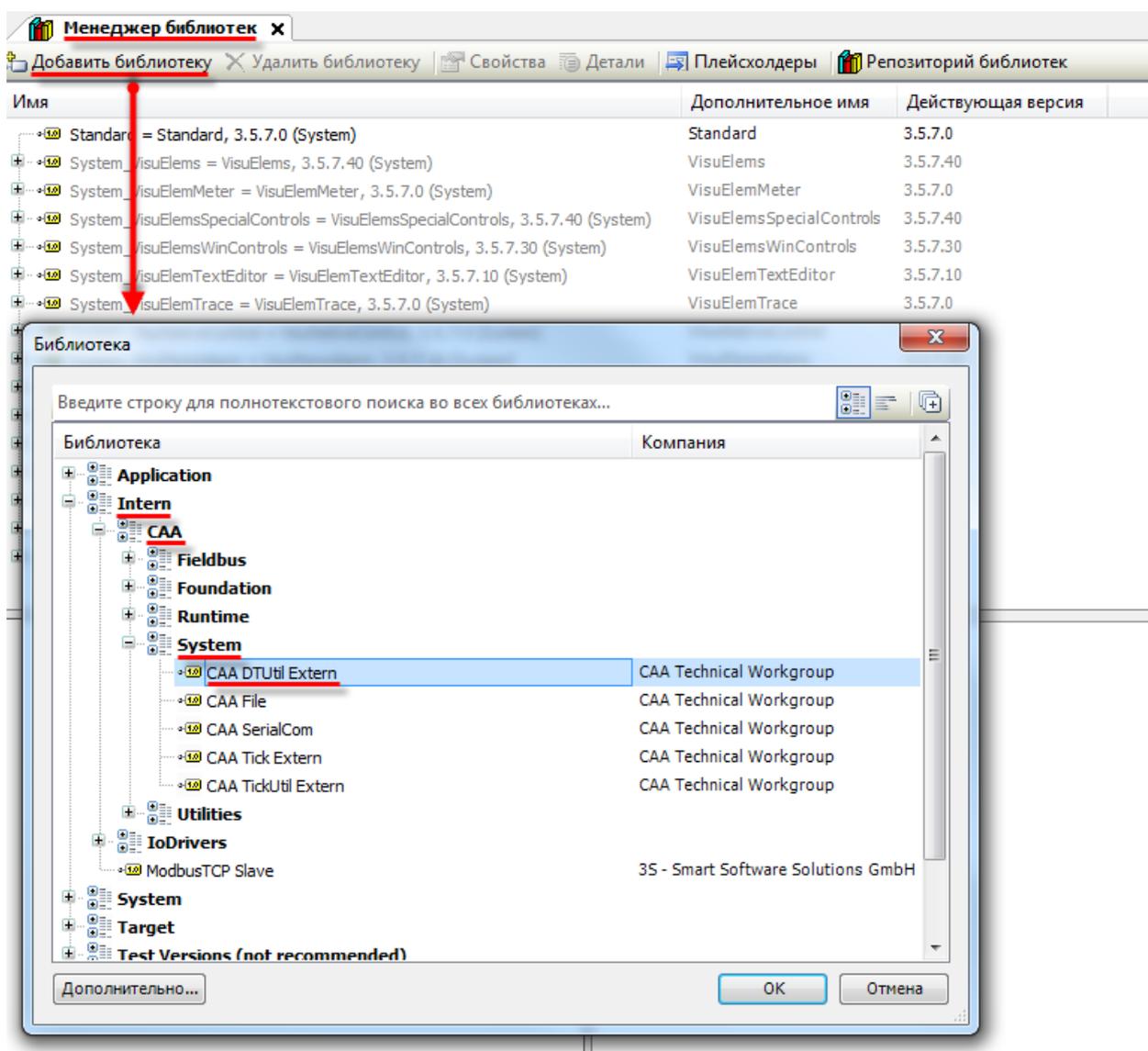


Рис. 5. Добавление библиотеки **CAA DTUtil** в проект **CODESYS**

Обратите внимание, что при обращении к ФБ и функциям библиотеки необходимо перед их названием указывать префикс **DTU**. (пример: **DTU.DTSplit**).

5.2. ФБ DTU.GetDateAndTime

Функциональный блок **DTU.GetDateAndTime** используется для считывания значения **системного времени Linux** в переменные программы. По переднему фронту на входе **xExecute** происходит однократное считывание системного времени. После появления на выходе **xDone** значения **TRUE**, можно забрать считанное время в формате [DATE AND TIME](#) с выхода **dtDateAndTime**.

Название	Тип данных	Описание
Входные переменные		
xExecute	BOOL	Переменная активации блока. Считывание системного времени происходит по <u>переднему фронту</u> переменной. Если переменная принимает значение FALSE , то значения всех выходов блока обнуляются.
Выходные переменные		
xDone	BOOL	Флаг успешного завершения работы блока. При значении TRUE можно забрать считанное значение системного времени с выхода dtDateAndTime .
xBusy	BOOL	Флаг «ФБ в процессе работы».
xError	BOOL	Флаг ошибки. Принимает значение TRUE при возникновении ошибки.
eError	DTU.ERROR	Статус работы ФБ (или имя ошибки).
dtDateAndTime	DATE_AND_TIME	Текущее системное время.
ePeriode	RTCLK.PERIODE	Временной период. Возможные значения: 0 – неизвестен; 1 – зимнее время; 2 – летнее время.

Обратите внимание, что если при старте проекта системное время сбрасывается на **00:00:00 01-01-1970**, то с подавляющей вероятностью причиной является разряжение батарейки аппаратных часов реального времени. Во всех СПК используется батарейка **CR2032 Li-Ion**.

5.3. ФБ DTU.SetDateAndTime

Функциональный блок **DTU.SetDateAndTime** используется для установки системного времени. По переднему фронту на входе **xExecute** происходит запись значения времени входа **dtDateAndTime** в **системные часы Linux**.

Название	Тип данных	Описание
<i>Входные переменные</i>		
xExecute	BOOL	Переменная активации блока. Установка системного времени происходит по <u>переднему фронту</u> переменной.
dtDateAndTime	DATE_AND_TIME	Время, которое будет записано в системные часы Linux .
<i>Выходные переменные</i>		
xDone	BOOL	Флаг успешного завершения работа блока.
xBusy	BOOL	Флаг «ФБ в процессе работы».
xError	BOOL	Флаг ошибки. Принимает значение TRUE при возникновении ошибки.
eError	DTU.ERROR	Статус работы ФБ (или имя ошибки).

5.4. Функция DTU.DTSplit

Функция **DTU.DTSplit** используется для выделения значений отдельных разрядов времени из переменной типа [DATE_AND_TIME](#). Обычно применяется совместно с ФБ [DTU.GetDateAndTime](#).

Название	Тип данных	Описание
<i>Входные переменные</i>		
dtDateAndTime	DATE_AND_TIME	Время, которое будет записано в системные часы Linux.
puiYear	POINTER TO UINT	Указатель на переменную типа UINT , в которую будет записан год, выделенный из переменной dtDateAndTime .
puiMonth	POINTER TO UINT	Указатель на переменную типа UINT , в которую будет записан месяц, выделенный из переменной dtDateAndTime .
puiDay	POINTER TO UINT	Указатель на переменную типа UINT , в которую будет записан день, выделенный из переменной dtDateAndTime .
puiHour	POINTER TO UINT	Указатель на переменную типа UINT , в которую будут записаны часы, выделенные из переменной dtDateAndTime .
puiMinute	POINTER TO UINT	Указатель на переменную типа UINT , в которую будут записаны минуты, выделенные из переменной dtDateAndTime .
puiSecond	POINTER TO UINT	Указатель на переменную типа UINT , в которую будут записаны секунды, выделенные из переменной dtDateAndTime .
<i>Выходные переменные</i>		
DTSplit	DTU.ERROR	Статус работы функции (или имя ошибки).

Функции **DateSplit** и **TODSplit** работают аналогичным образом, выделяя разряды времени из переменных типа [DATE](#) и [TIME_OF_DAY](#) соответственно.

5.5. Функция DTU.DTConcat

Функция **DTU.DTConcat** используется для «склеивания» значений отдельных разрядов времени в переменную типа [DATE AND TIME](#). Обычно применяется совместно с ФБ [DTU.SetDateAndTime](#).

Название	Тип данных	Описание
<i>Входные переменные</i>		
uiYear	UINT	Год.
uiMonth	UINT	Месяц.
uiDay	UINT	День.
uiHour	UINT	Часы.
uiMinute	UINT	Минуты.
uiSecond	UINT	Секунды.
peError	POINTER TO DTU.ERROR	Указатель на переменную типа DTU.ERROR , в которую будет записываться статус работы функции (или имя ошибки).
<i>Выходные переменные</i>		
DTConcat	DATE_AND_TIME	Время в формате DATE AND TIME , «склеенное» из значений разрядов времени, записанных на входы функции.

Функции **DateConcat** и **TODConcat** работают аналогичным образом, «склеивая» значения отдельных разрядов времени в переменные типа [DATE](#) и [TIME OF DAY](#) соответственно.

5.6. Функция DTU.GetDayOfWeek

Функция **DTU.GetDayOfWeek** используется для получения номера дня недели из переменной типа [DATE](#).

Название	Тип данных	Описание
<i>Входные переменные</i>		
dtDate	DATE	Дата.
peError	POINTER TO DTU.ERROR	Указатель на переменную типа DTU.ERROR, в которую будет записываться статус работы функции (или имя ошибки).
<i>Выходные переменные</i>		
GetDayOfWeek	RTCLK.WEEKDAY	Номер дня недели (0 – воскресенье, 1 – понедельник ... 6 – суббота).

6. Пример работы с системным временем

Рассмотрим пример работы с системным временем, включающий в себя:

- отображение системного времени в визуализации;
- считывания системного времени в программу с помощью библиотеки [CAA DTUtil](#);
- изменения системного времени из программы с помощью библиотеки [CAA DTUtil](#).

Вывод системного времени через заполнители: %t[dd.MM.yyyy HH:mm:ss]	
Чтение системного времени из кода программы	Изменение системного времени из кода программы
Год: %d	Год: %d
Месяц: %d	Месяц: %d
День: %d	День: %d
Час: %d	Час: %d
Минута: %d	Минута: %d
Секунда: %d	Секунда: %d
	Записать
Вывод системного времени через заполнители: %t[yyyy / MMM / dddd HH-mm-ss]	

Рис. 6. Внешний вид примера **Example_SystemTime**

Пример создан в среде **CODESYS 3.5 SP7 Patch 4** и подразумевает запуск на контроллере СПК207 с target-файлом **3.5.4.20 (023)**.

Пример доступен для скачивания: [Example_SystemTime.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с программой **PLC_PRG** на языке **CFC**.
2. В **Менеджере библиотек** [добавим библиотеку CAA DTUtil](#) версии **3.5.1.0**.
3. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     GetSysDT:DTU.GetDateAndTime; // ФБ считывания системного времени
4
5     // считанное системное время:
6
7     uiReadYear: UINT; // год
8     uiReadMonth: UINT; // месяц
9     uiReadDay: UINT; // день
10    uiReadHour: UINT; // часы
11    uiReadMinute: UINT; // минуты
12    uiReadSecond: UINT; // секунды
13
14
15    SetSysDT:DTU.SetDateAndTime; // ФБ установки системного времени
16    xWriteDT: BOOL; // триггер записи системного времени
17
18    // устанавливаемое системное время
19
20    uiWriteYear: UINT; // год
21    uiWriteMonth: UINT:=1; // месяц
22    uiWriteDay: UINT:=1; // день
23    uiWriteHour: UINT; // часы
24    uiWriteMinute: UINT; // минуты
25    uiWriteSecond: UINT; // секунды
26
27    eWriteDTerror: DTU.ERROR; // статус работы функции DTConcat
28 END_VAR
```

Рис. 7. Объявление переменных программы **PLC_PRG**

4. Код программы PLC_PRG будет выглядеть следующим образом:

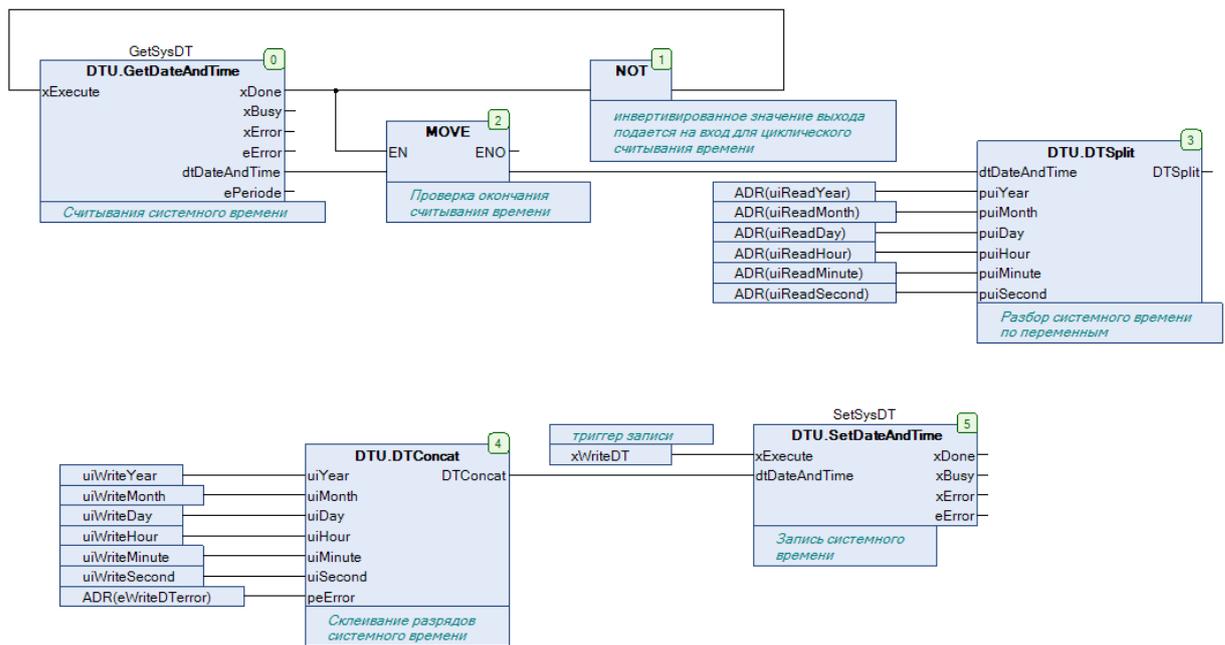


Рис. 8. Код программы PLC_PRG

```

GetSysDT (xExecute:=NOT (GetSysDT.xDone) );

IF GetSysDT.xDone THEN

    DTU.DTSplit
    (
        dtDateAndTime := GetSysDT.dtDateAndTime,
        puiYear       := ADR (uiReadYear),
        puiMonth      := ADR (uiReadMonth),
        puiDay        := ADR (uiReadDay),
        puiHour       := ADR (uiReadHour),
        puiMinute     := ADR (uiReadMinute),
        puiSecond     := ADR (uiReadSecond)
    );

END_IF

SetSysDT
(
    xExecute      := xWriteDT,
    dtDateAndTime := DTU.DTConcat
    (
        uiYear      := uiWriteYear,
        uiMonth     := uiWriteMonth,
        uiDay       := uiWriteDay,
        uiHour      := uiWriteHour,
        uiMinute    := uiWriteMinute,
        uiSecond    := uiWriteSecond,
        peError     := ADR (eWriteDTerror)
    ),
);

```

Рис. 9. Пример семантически эквивалентного кода на языке ST

Программа работает следующим образом:

Блоки 0-1. Реализация считывания системного времени в цикле программы. Поскольку ФБ [DTU.GetDateAndTime](#) работает по переднему фронту, то для циклического выполнения на вход **xExecute** подается инвертированное значение выхода **xDone**.

Блоки 2-3. По переднему фронту выхода **xDone** считанное значение системного времени в формате [DATE AND TIME](#) подается на вход функции [DTU.DTSplit](#), в результате чего в переменные **uiReadYear**, **uiReadMonth** и т.д. записываются значения отдельных разрядов системного времени. **Обратите внимание**, что на входах функции указываются не переменные, а их адреса.

Блоки 4-5. Подготовленные для записи значения разрядов системного времени (**uiWriteYear**, **uiWriteMonth** и т.д.), «склеиваются» в переменную формата [DATE AND TIME](#) с помощью функции [DTU.DTConcat](#), которая поступает на вход ФБ [DTU.SetDateAndTime](#). По переднему фронту переменной **xWriteDT** происходит запись этого значения в системные часы Linux.

5. Добавим в проект экран визуализации. Нажмем **ПКМ** на его название, выберем в контекстном меню пункт **Свойства**, после чего на вкладке **Визуализация** укажем **Заданный размер визуализации**: ширина – **800**, высота – **480** (эти значения соответствуют дисплею СПК207).

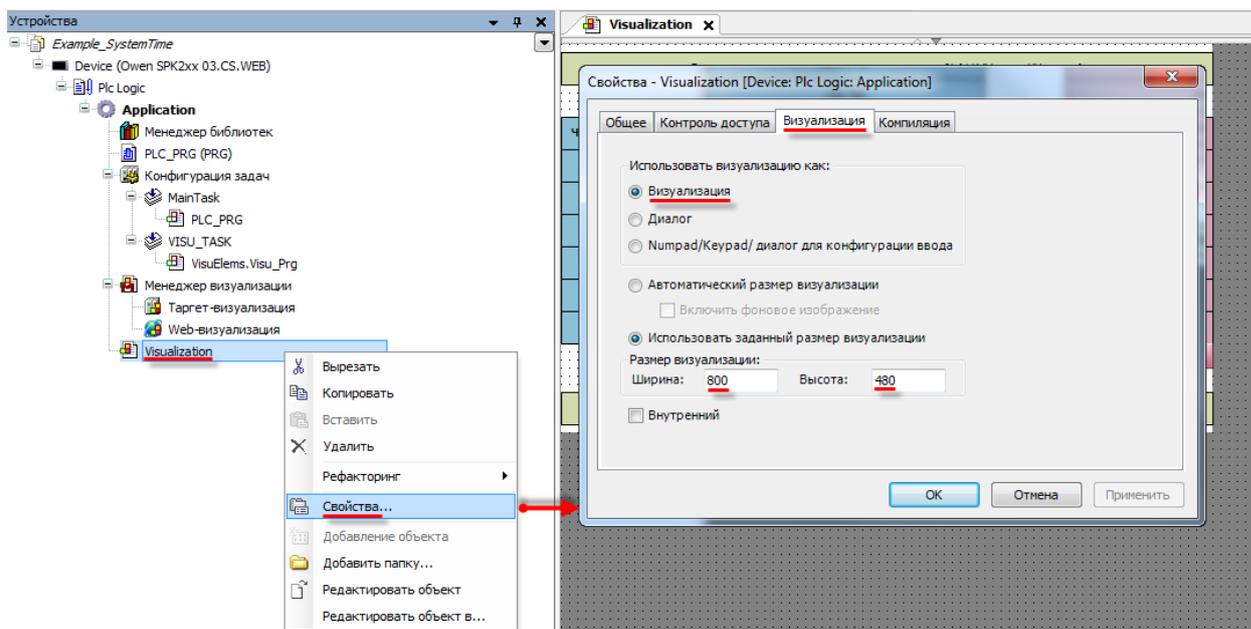


Рис. 10. Настройки размера экрана визуализации

6. Настройки target- и web-визуализации будут выглядеть следующим образом:

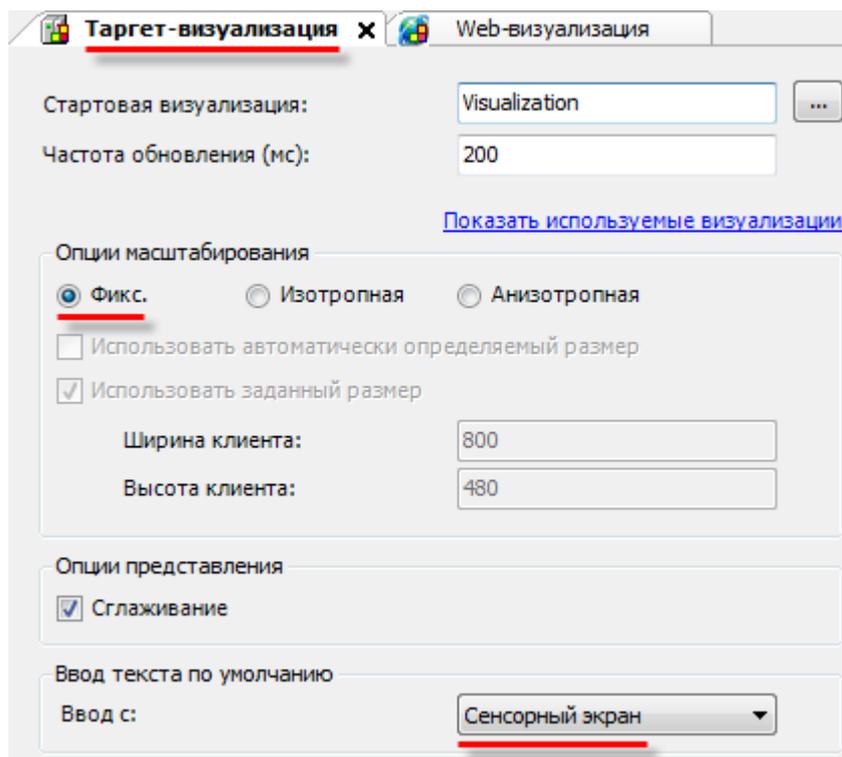


Рис. 11. Настройки target-визуализации

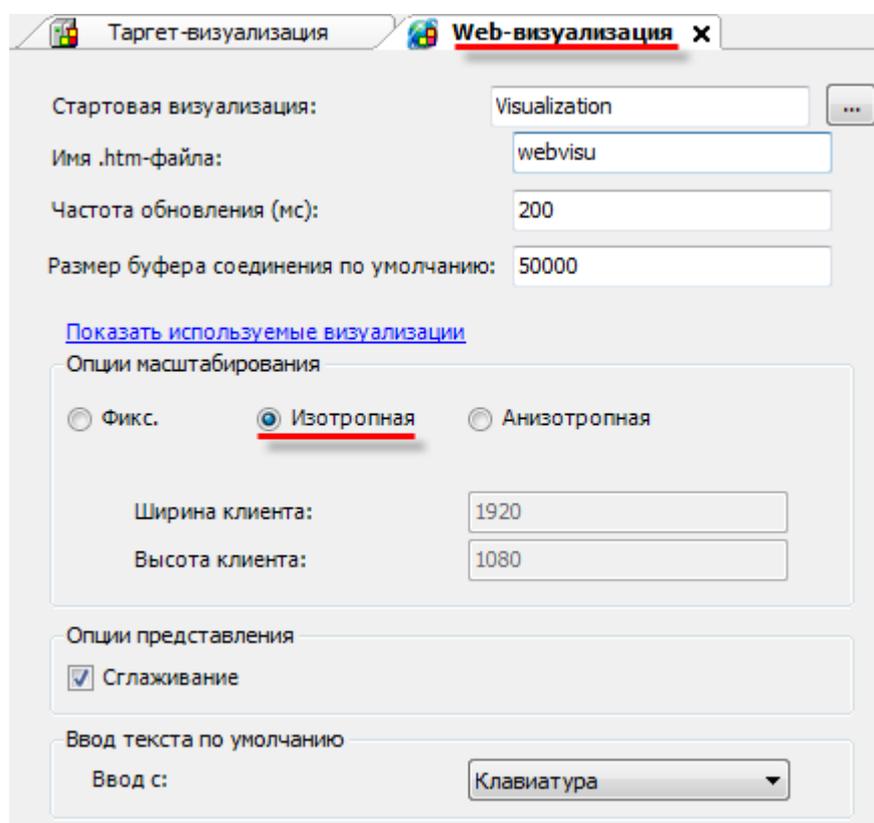


Рис. 12. Настройки web-визуализации

7. Экран визуализации будет содержать 16 элементов **Прямоугольник** и один элемент **Кнопка**.

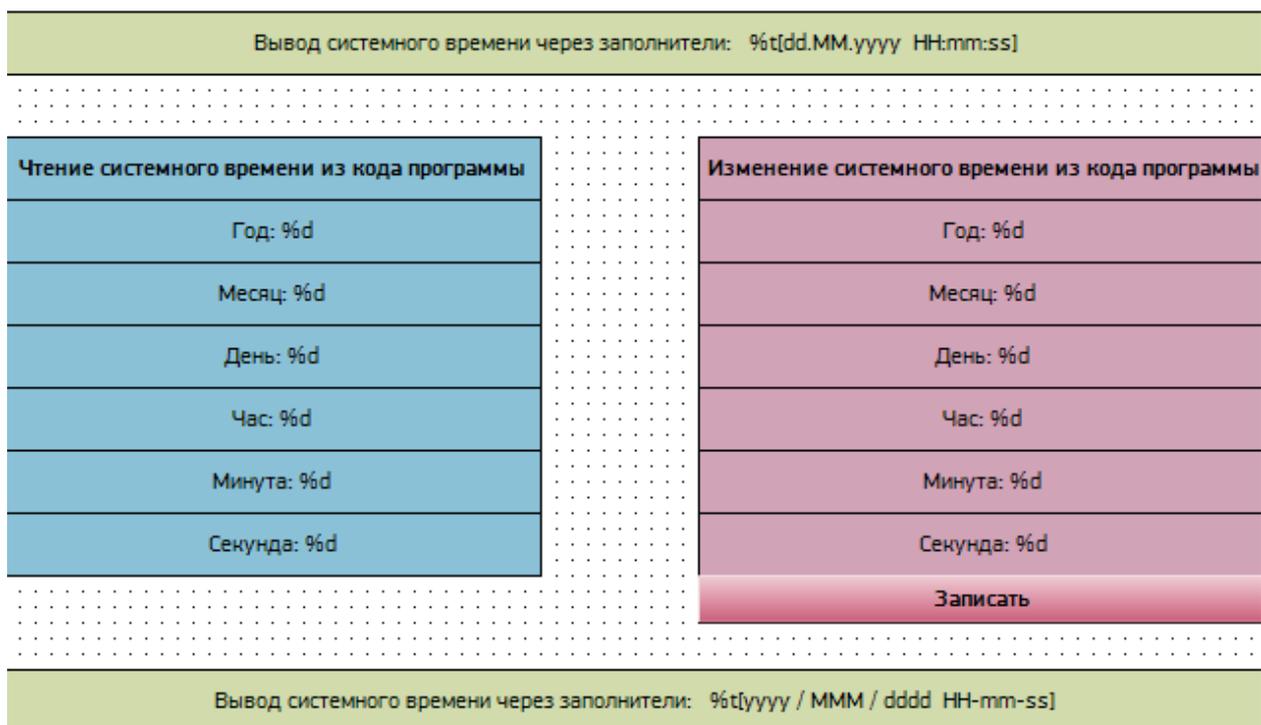


Рис. 13. Внешний вид экрана визуализации

Верхний и нижний зеленый прямоугольники будут использоваться для отображения системного времени через [заполнители](#). Для этого достаточно в параметре **Текст** указать соответствующую строку со спецификатором формата вывода **%t[формат времени]**:

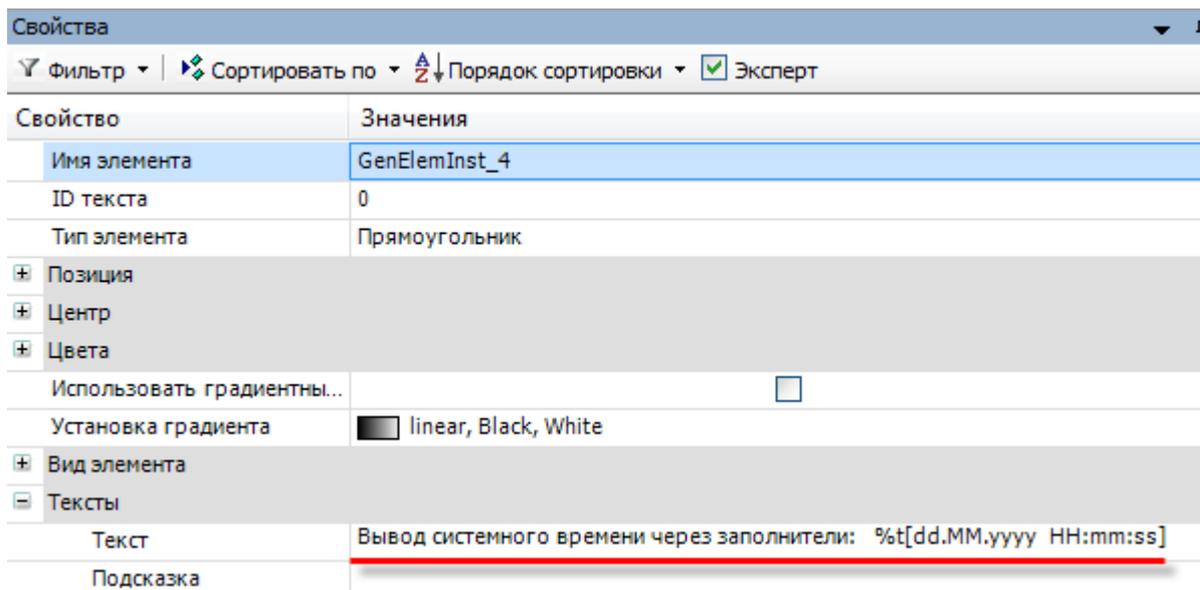


Рис. 14. Отображение системного времени через спецификатор формата вывода **%t[формат времени]**

Левая группа прямоугольников будет использоваться для отображения разрядов системного времени, считанного в программе **PLC_PRG**. В параметре **Текст** укажите название разряда и спецификатор формата вывода **%d** (используется для целочисленных значений). К параметру **Текстовая переменная** привяжите соответствующую переменную программы (**uiReadYear**, **uiReadMonth** и т.д.)

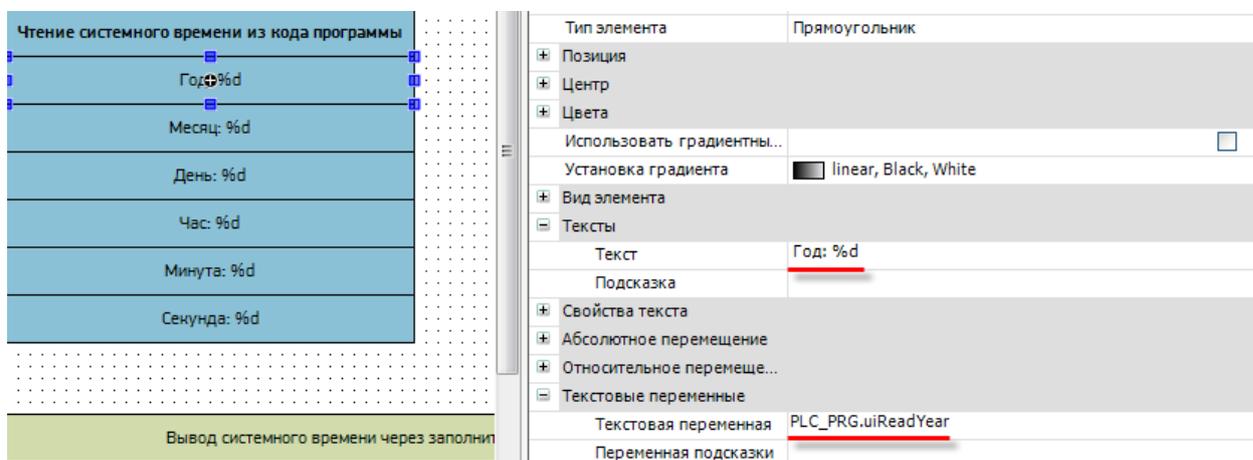


Рис. 15. Настройка элементов отображения считанного в программе системного времени

Правая группа прямоугольников будет использоваться для ввода и отображения значений, которые будут записаны в разряды системного времени после нажатия на кнопку **Записать**. В параметре **Текст** укажите название разряда и спецификатор формата вывода **%d** (используется для целочисленных значений). К параметру **Текстовая переменная** привяжите соответствующую переменную программы (**uiWriteYear**, **uiWriteMonth** и т.д.). Во вкладке **Конфигурация ввода** к параметру **OnMouseClicked** привяжите действие **Записать переменную** с типом ввода **Numpad**. Укажите нижний и верхний предел вводимых значений (см. табл. 3).

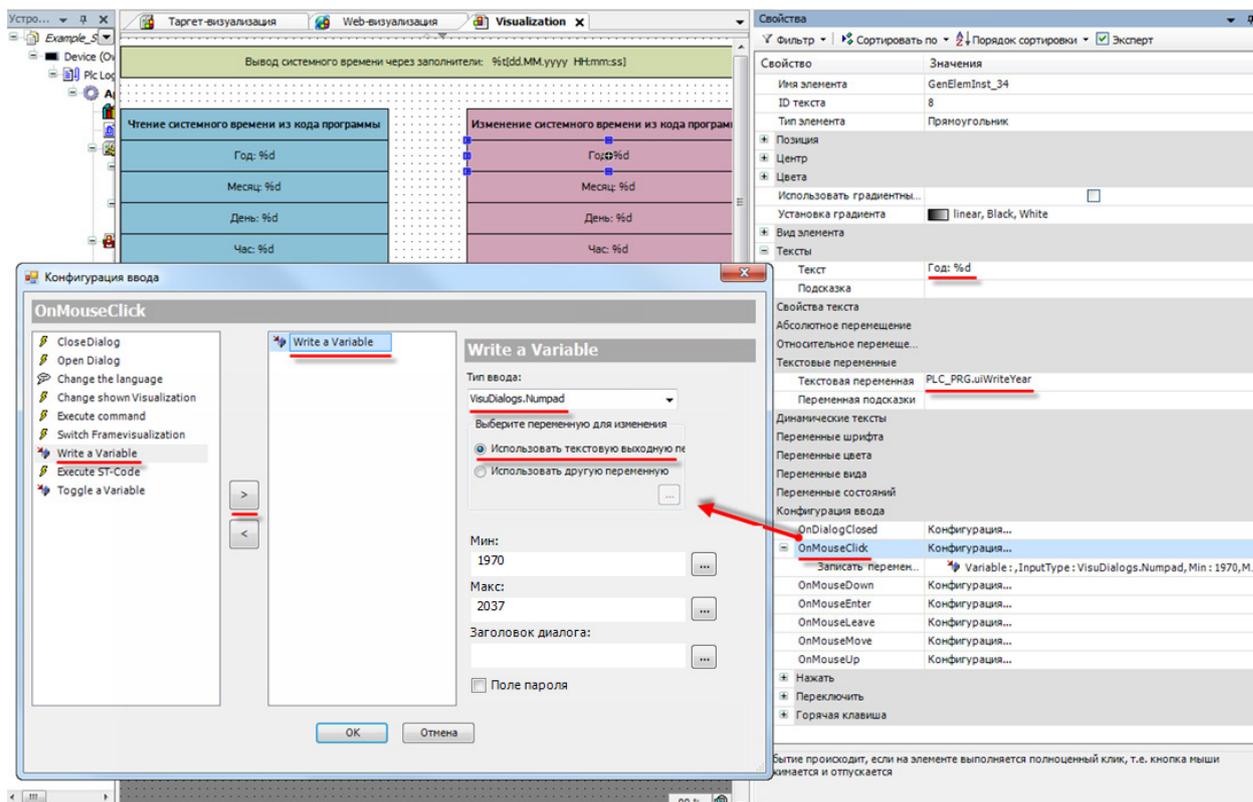


Рис. 16. Настройка элементов установки разрядов системного времени

Табл. 3. Ограничения на значения разрядов времени в CODESYS

Разряд	Нижний предел	Верхний предел
Год	1970	2037
Месяц	1	12
День	1	31
Часы	0	23
Минуты	0	59
Секунды	0	59

Для кнопки **Записать** в **Конфигурации ввода** во вкладке **Нажать** привяжите переменную **xWriteDT**. По нажатию кнопки будет происходить запись подготовленного значения времени в системные часы Linux.

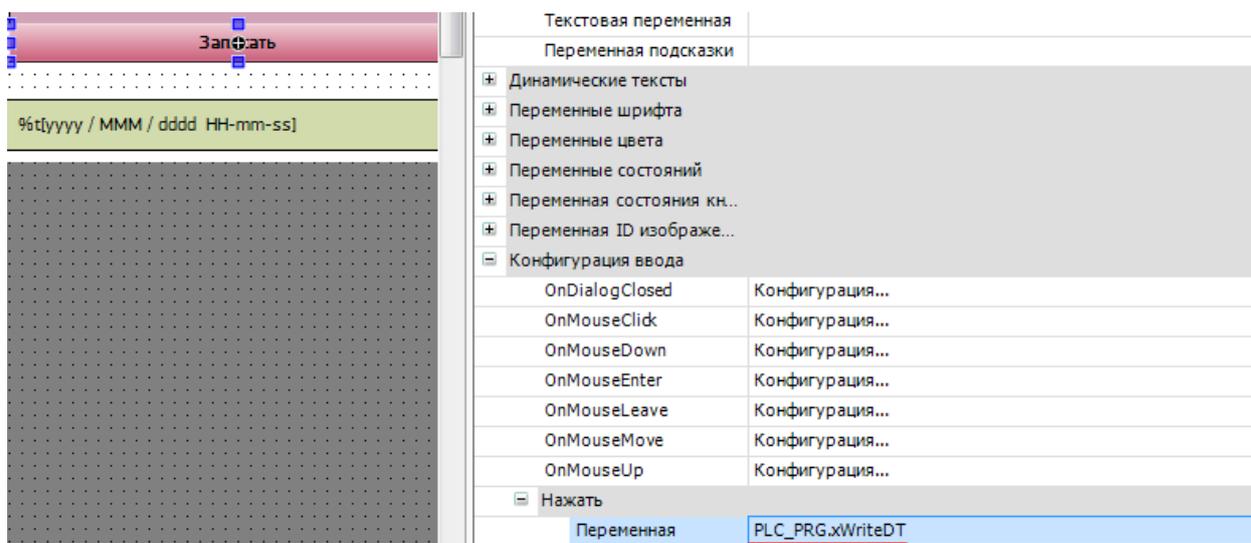


Рис. 17. Настройка кнопки установки системного времени

На этом создание экрана визуализации завершено.

8. Загрузите пример в СПК и запустите его.

Вывод системного времени через заполнители: 19.05.2016 10:00:27																
<table border="1"><thead><tr><th>Чтение системного времени из кода программы</th></tr></thead><tbody><tr><td>Год: 2016</td></tr><tr><td>Месяц: 5</td></tr><tr><td>День: 19</td></tr><tr><td>Час: 10</td></tr><tr><td>Минута: 0</td></tr><tr><td>Секунда: 27</td></tr></tbody></table>	Чтение системного времени из кода программы	Год: 2016	Месяц: 5	День: 19	Час: 10	Минута: 0	Секунда: 27	<table border="1"><thead><tr><th>Изменение системного времени из кода программы</th></tr></thead><tbody><tr><td>Год: 2016</td></tr><tr><td>Месяц: 1</td></tr><tr><td>День: 1</td></tr><tr><td>Час: 0</td></tr><tr><td>Минута: 0</td></tr><tr><td>Секунда: 0</td></tr><tr><td>Записать</td></tr></tbody></table>	Изменение системного времени из кода программы	Год: 2016	Месяц: 1	День: 1	Час: 0	Минута: 0	Секунда: 0	Записать
Чтение системного времени из кода программы																
Год: 2016																
Месяц: 5																
День: 19																
Час: 10																
Минута: 0																
Секунда: 27																
Изменение системного времени из кода программы																
Год: 2016																
Месяц: 1																
День: 1																
Час: 0																
Минута: 0																
Секунда: 0																
Записать																
Вывод системного времени через заполнители: 2016 / May / Thursday 10-00-27																

Рис. 18. Внешний вид примера **Example_SystemTime**

В верхнем и нижнем прямоугольниках будет отображаться текущее системное время, выведенное через спецификатор формата вывода **%t[формат времени]**.

В левой группе прямоугольников будут отображаться разряды системного времени, считанные в программе.

Нажмите на один из прямоугольников правой группы, чтобы установить новое значение одного из разрядов системного времени.

<table border="1"><thead><tr><th>Чтение системного времени из кода программы</th></tr></thead><tbody><tr><td>Год: 2016</td></tr><tr><td>Месяц: 5</td></tr><tr><td>День: 19</td></tr><tr><td>Час: 10</td></tr><tr><td>Минута: 0</td></tr><tr><td>Секунда: 27</td></tr></tbody></table>	Чтение системного времени из кода программы	Год: 2016	Месяц: 5	День: 19	Час: 10	Минута: 0	Секунда: 27	<table border="1"><thead><tr><th>Изменение системного времени из кода программы</th></tr></thead><tbody><tr><td>Год: 2016</td></tr><tr><td>Месяц: 1</td></tr><tr><td>День: 1</td></tr><tr><td>Час: 0</td></tr><tr><td>Минута: 0</td></tr><tr><td>Секунда: 0</td></tr><tr><td>Записать</td></tr></tbody></table>	Изменение системного времени из кода программы	Год: 2016	Месяц: 1	День: 1	Час: 0	Минута: 0	Секунда: 0	Записать
Чтение системного времени из кода программы																
Год: 2016																
Месяц: 5																
День: 19																
Час: 10																
Минута: 0																
Секунда: 27																
Изменение системного времени из кода программы																
Год: 2016																
Месяц: 1																
День: 1																
Час: 0																
Минута: 0																
Секунда: 0																
Записать																

Рис. 19. Установка нового значения года

Выставите новые значения для требуемых разрядов и нажмите кнопку **Записать**. Новое значение времени будет записано в системные часы Linux. Это значение отобразится в верхнем и нижнем прямоугольнике, а также будет считано в программу и отображено в левой группе прямоугольников.

7. Синхронизация времени с NTP-сервером

Часы контроллера отсчитывают время с определенной точностью, и на длительном интервале эксплуатации из-за накапливаемой погрешности системное время начинает либо отставать, либо опережать реальное.

Если контроллер имеет интерфейс **Ethernet**, то можно организовать синхронизацию его часов с сервером точного времени по протоколу [NTP](#).

Этот функционал рассмотрен в примере [Example_SNTP.projectarchive](#)

Пример создан в среде **CODESYS 3.5 SP7 Patch 4**.

Теоретическая информация по примеру приведена в документе **СПК. Реализация обмена через сокеты**.

8. Системный таймер ПЛК

Системный таймер контроллера содержит время, прошедшее с момента последнего включения устройства.

Для чтения времени системного таймера используется функция **TIME()**, которая возвращает значение типа **TIME**:

```
// время, прошедшее с момента включения контроллера  
tSystemTimer := TIME();
```

На основе этой функции, в частности, реализованы таймеры из библиотеки **Standard**.