

# Панели оператора СПЗхх-Р

## Синхронизация времени с ПЛК

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

Установить системное время можно на экране **60002**. Подразумевается, что на дисплее будет находиться кнопка перехода на этот экран, и оператор при необходимости вручную сможет скорректировать время часов панели.

В некоторых случаях такой вариант не подходит – требуется автоматически синхронизировать часы панели с часами ПЛК через заданные интервалы времени. Для этого используются системные регистры панели **PSW220** и **PSW221**. Регистр **PSW220** является буферным регистром, в который записывается значение одного из разрядов времени. В регистре **PSW221** хранятся биты записи:

- PSW220.0 – запись значения года;
- PSW220.1 – запись значения месяца;
- PSW220.2 – запись значения дня;
- PSW220.3 – запись значения часов;
- PSW220.4 – запись значения минут;
- PSW220.5 – запись значения секунд;
- PSW220.6 – запись номера дня.

Таким образом, требуется записать значение нужного разряда времени в регистр **PSW220** и сгенерировать импульс переднего фронта в нужном бите регистра **PSW221**. В данном примере для упрощения этого процесса используется макрос.

Макрос **Func1** производит запись значения регистров PSW300-305 в часы панели (где **PSW300** – значение года, **PSW301** – значение месяца и т.д.). Макрос использует вспомогательные функции, объявленные в **глобальном макросе** (поэтому пользователю следует также перенести их в свой проект). В частности, эти функции используются, чтобы преобразовать десятичные значения разрядов времени, получаемые с ПЛК, в [двоично-десятичный формат](#), используемый для хранения разрядов времени в панели.

Таким образом, пользователю необходимо сделать следующее:

1. Считать системное время в программе ПЛК и выделить из него отдельные разряды времени в десятично формате (например, в среде **CoDeSys 2.3** для этого используется библиотека **SysLibTime**, а в **CODESYS 3.5** – библиотека **CAA DTUtil**);
2. Организовать передачу этих значений в панель. В рамках примера подразумевается, что ПЛК выполняет функцию **Modbus Master**, а панель – **Modbus Slave**. ПЛК должен записать значения разрядов в регистры панели **PSW300-305** (где PSW300 – год, PSW301 – месяц и т.д.). Для этого следует использовать функцию **0x06** (Write Single Register) или **0x10** (Write Multiple Registers).

3. После этого ПЛК должен однократно записать значение **TRUE** в бит панели **PSB300** (с помощью функции **0x05** (Write Single Coil) или **0x0F** (Write Multiple Registers)).

4. Бит **PSB300** активирует функциональную область, в которой вызывается макрос **Func1**. **Обратите внимание** – макрос должен вызываться в **параллельном режиме**. После выполнения макроса панель переводит бит **PSB300** в состояние **FALSE**.

В рамках примера можно также ввести значения разрядов времени вручную, после чего записать их кнопкой **Запись в RTC**.

Системное время:  
FFFF / FF / FF  
FF / FF / FF

**PSB300**  
Функциональная область

ПЛК должен записать разряды системного времени в регистры PSW300-305, после чего однократно записать TRUE в бит PSB300.

Для проверки можно использовать кнопку "Записать в RTC".

Синхронизация выполняется в макросе. Вспомогательные функции объявлены в глобальном макросе.

Установка системного времени	
Год	<b>PSW300</b> 0000
Месяц	<b>PSW301</b> 00
День	<b>PSW302</b> 00
Часы	<b>PSW303</b> 00
Минуты	<b>PSW304</b> 00
Секунды	<b>PSW305</b> 00

**PSB300**  
Записать в RTC

Рис. 1. Внешний вид экрана визуализации примера

```

1 WORD uTemp = 0;
2 WORD i = 0;
3
4
5
6 //////////////// Function DEC_TO_BCD ////////////////
7 WORD DEC_TO_BCD(WORD DEC)
8 {
9     return (DEC) ? (( DEC_TO_BCD( DEC / 10 ) << 4) + (DEC % 10)) : 0;
10 }
11
12
13 //////////////// Function WriteTo ////////////////
14 void WriteTo(WORD Address, WORD Value, WORD DATA_TYPE)
15 {
16
17     if (DATA_TYPE==TYPE_PSB)
18     {
19         if (Value==TRUE)
20         {
21             SetPSB(Address);
22         }
23         else
24         {
25             ResetPSB(Address);
26         }
27     }
28
29     if (DATA_TYPE==TYPE_PSU)
30     {

```

```

1 For ( i = 0; i < 6; i++ )
2 {
3     PSW[220]=DEC_TO_BCD(PSW[300+i]);
4     WriteBitInReg(221, 0+i, TRUE, TYPE_PSW);
5     Delay(100);
6     WriteBitInReg(221, 0+i, FALSE, TYPE_PSW);
7 }
8

```

Рис. 2. Исполняемый код размещен в макросе **Func1**, а код доп. функций – в глобальном макросе