

APLIKAČNÍ POSTUP

Práce s vysokorychlostní sběrnici

SPEED-Bus – digitální signály



Práce s vysokorychlostní sběrnici SPEED-Bus – digitální signály

Abstrakt

SPEED-Bus je vysokorychlostní 32bitová paralelní sběrnice, která je implementovaná v některých typech CPU řady 300S od společnosti VIPA. Díky paralelnímu přenosu dat a celkovému pojetí sběrnice je dosahováno extrémně rychlého přenosu dat mezi CPU a moduly připojenými ke sběrnici SPEED-Bus. Moduly SPEED-Bus se na rozdíl od modulů pro standardní sériovou sběrnici připojují k CPU z levé strany a zasouvají se do speciální lišty SPEED-Bus.

Každý modul na sběrnici SPEED-Bus obsahuje čítač s rozlišením 1 μ s. Čítač se rozběhne po startu CPU a pro moduly na sběrnici SPEED-Bus představuje časovou základu s přesností $\pm 1 \mu$ s. Jde o 16bitový čítač, který cyklicky běží od 0 do W#16#FFFF, což odpovídá 0 až 65535 decimálně. Tohoto čítače využívá funkce ETS, která umožňuje uložit aktuální hodnotu čítače při změně signálu na vstupní svorce a tím tuto hodnotu připravit pro pozdější zpracování.

Tento aplikační postup ukazuje využití funkce ETS na příkladu měření doby cyklu OB1.

Aplikační postup navazuje na aplikační postup „Hardwarová konfigurace vysokorychlostní sběrnice SPEED-Bus“, který použijte jako referenci pro přípravu hardwarové konfigurace.

Přílohy

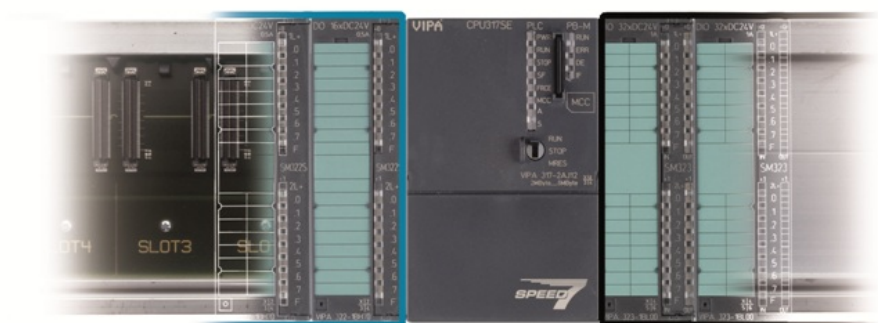
- Manuál k CPU 314-6CF02
- Manuál k modulům na sběrnici SPEED-Bus
- GSD soubory Cx000023_207.zip
- Projekt ve STEP7

HW komponenty

- CPU 314-6CF02
- Ethernetový kabel pro download projektu
- Lišta SPEED-Bus 391-1AF10
- Vysokorychlostní modul 16 digitálních vstupů/výstupů 323-1BH70
- Vysokorychlostní modul 8 analogových vstupů 331-7BF70 (pro tento aplikační postup není nutné)

SW komponenty

- STEP7 od společnosti Siemens



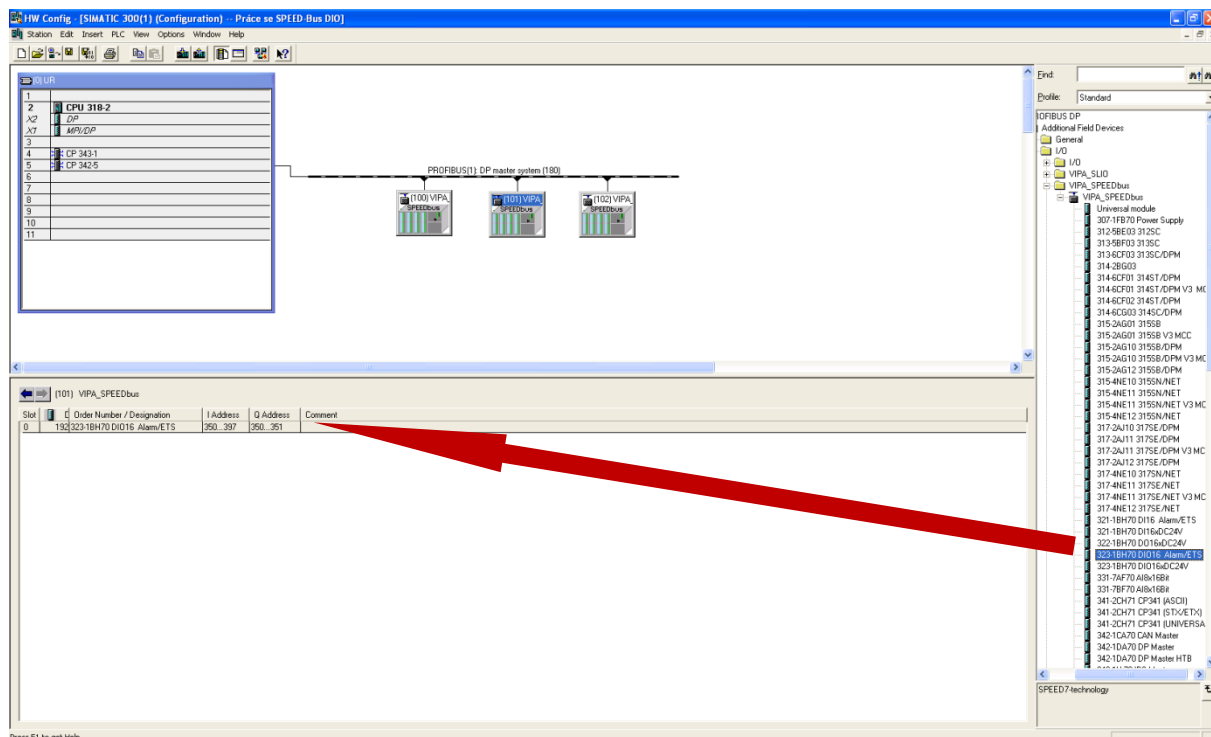
Důležitá poznámka


Aplikační postupy demonstrují typické úkony na konkrétních případech. Nekladou si za cíl kompletnost a v žádném případě nenahrazují návod k obsluze! Změna aplikačních postupů je vyhrazena.

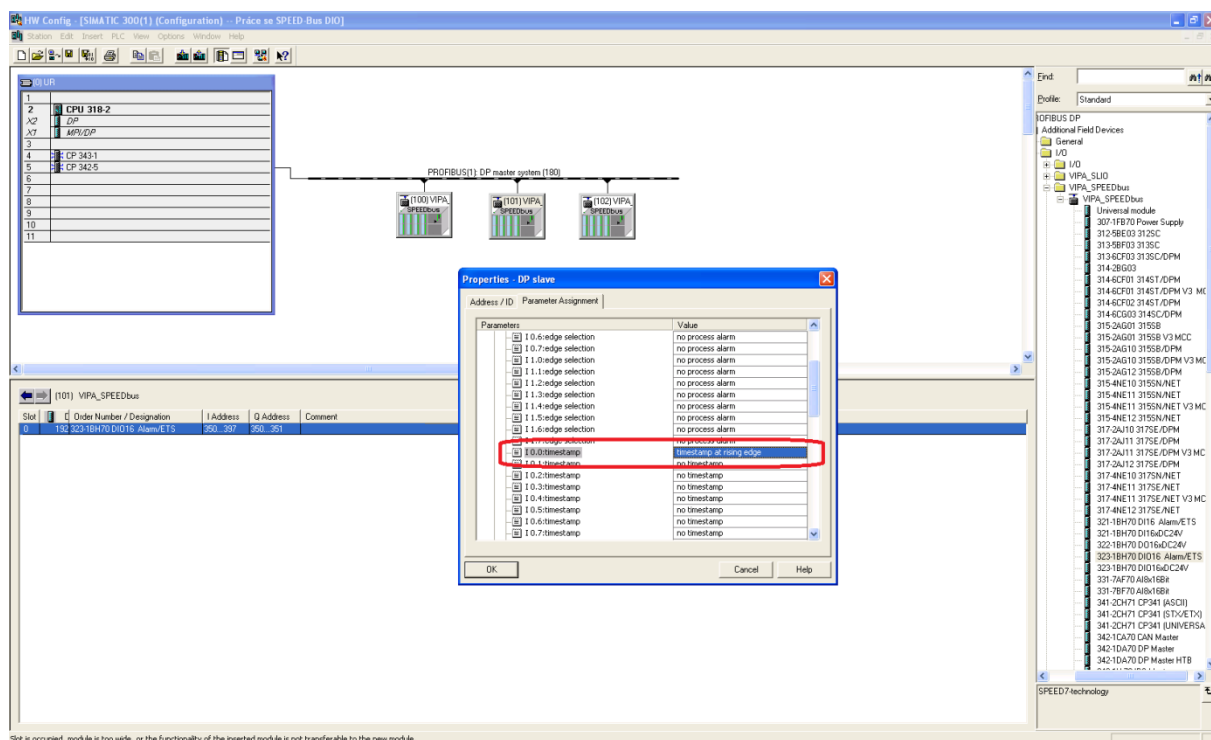
Postup

Předpokládáme, že máte vytvořenou hardwarovou konfiguraci podle aplikačního postupu „Hardwarová konfigurace vysokorychlostní sběrnice SPEED-Bus“ s CPU 314-6CF02, digitálním vstupně/výstupním modulem 323-1BH70 a analogovým modulem 331-7BF70. Analogový modul 331-7BF70 v tomto aplikačním listu není potřeba, ale pro snadnou návaznost na předchozí aplikační list jej budeme uvažovat.

Ubezpečte se, že používáte modul s funkcí ETS, tedy 323-1BH70 DIO16 Alarm/ETS, pokud ne, přetáhněte jej z katalogu do příslušného Slave modulu virtuální sítě Profibus.



Ve vlastnostech modulu 323-1BH70 DIO16 Alarm/ETS, které se zobrazí po dvojkliku na tento modul zvolte v záložce „Parameter Assignment“ pro I0.0: timestamp „timestamp at rising edge“. Hardwarovou konfiguraci uložte pomocí tlačítka  a nahrajte do CPU.



Modul 323-1BH70 DIO16 Alarm/ETS obsadí vstupní adresovou oblast CPU o velikosti 48 WORD a výstupní paměťovou oblast o velikosti 2 WORD. V hardwarové konfiguraci jsme modul umístili od adresy 350, a to jak ve vstupní, tak ve výstupní paměťové oblasti, tedy k adresám popsaným v textu níže je nutno v tomto případě přičíst hodnotu 350.

Pro detekci nové hodnoty „time stamp“ pro prvních 8 vstupů slouží 4. WORD ve vstupní paměťové oblasti. Vždy při uložení hodnoty „time stamp“ je v tomto registru nastaven bit pro příslušný vstup. Hodnota tohoto registru se automaticky nuluje po jejím vyčtení. Protože v tomto příkladu budeme používat pouze první vstup a ostatní předpokládáme, že nejsou zapojeny, stačí detekovat novou hodnotu „time stamp“ prostým porovnáním s nulou – viz programový příklad níže. Uloženou hodnotu „time stamp“ nalezneme pro první vstup na adrese 12. Vstupní adresová oblast je popsána v následujícím výňatku z manuálu:

**48bytes in the
process image**

The module occupies 48byte in the input address range of the CPU that have the following meaning:

Input address range

Byte	Bit 7 ... Bit 0
0	State of the channels (1 = set, 0 = not set) Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
1	State of the channels (1 = set, 0 = not set) Bit 8: Status I+1.0 ... Bit 15: Status I+1.7
2 3	reserved
4	1. edge evaluation (1=edge detected, 0=no edge detected) Here the last presence of an edge since the last read access to the register is stored. After a read access to this register (in the module) it is set back. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
5	Edge evaluation Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
6 ... 7	reserved
8	Edge lost (1=edge lost, 0=no edge lost) Here is noted if an edge change has been lost, i.e. if there has been more than one edge change since the last read access. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
9	Edge lost Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
10...11	reserved

continued ...

Byte	+3	+2	+1	+0
	The following bytes contain the values of the μ s ticker for a channel at the time of an edge change. Only the lower 16Bit are taken over. An overflow after 65ms has to be accordingly processed in the user application.			
12	Time stamp I+0.1		Time stamp I+0.0	
16	Time stamp I+0.3		Time stamp I+0.2	
20	Time stamp I+0.5		Time stamp I+0.4	
24	Time stamp I+0.7		Time stamp I+0.6	
28	Time stamp I+1.1		Time stamp I+1.0	
32	Time stamp I+1.3		Time stamp I+1.2	
36	Time stamp I+1.5		Time stamp I+1.4	
40	Time stamp I+1.7		Time stamp I+1.6	

Byte	Bit 7 ... Bit 0
44	2. edge evaluation (1=edge detected, 0=no edge detected) Here the last presence of an edge since the last read access to the register is stored. After a read access to this register (in the module) the register is not reset. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
45	Edge evaluation Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
46...47	reserved

Ukázkový program níže je kompletně napsán v OB1. Při každém cyklu OB1 se mění stav výstupu Q1.0, což je v tomto případě první výstup na svorkách vestavěných IO na CPU. Tento výstup je nutno na svorkách propojit s prvním vstupem modulu 323-1BH70. Modul 323-1BH70 uloží hodnotu „timestamp“ vždy při náběžné hraně signálu. Odečtením 2 po sobě jdoucích hodnot „timestamp“ měříme dobu dvojice cyklu OB1. V situaci, kdy čítač dosáhne hodnoty W#16#FFFF a začne čítat znovu od 0, dojde k přetečení. Rozdíl 2 po sobě jdoucích hodnot by byl záporný, proto je tato situace v ukázkovém programu řešena v obsluze přetečení, kde je zjišťován rozdíl od hodnoty W#16#FFFF – dekadicky 65535 a ten je přičítán ke 2. hodnotě z měření. S čísly je v tomto případě nutno pracovat ve formátu Long. V závěru ukázkového programu je volána funkce WAIT, která dokáže dobu cyklu uměle prodloužit.

Výpis programu v OB1

```
// při každém cyklu OB1 změnit stav výstupy
    AN    Q      1.0
    =     Q      1.0

// kontrola hrany (zápis hodnoty ETS)
    L     PIB    354
    L     0
    ==I
    JC     end

// aktuální hodnota
```

```

        L      MD      10
// stará hodnota
        T      MD      20
// uložení hodnoty timestamp
        L      PIW     362
        T      MD      10

        L      MD      10
        L      MD      20

// test na přetečení
        <D
        JC      OVF

        -D
// měřeny jsou vždy 2 cykly OB1
        T      MD      30
        L      2
        /D
// aktuální doba cyklu OB1

        T      MD      40

        JU      end

// obsluha přetečení
OVF:    L      L#65535
        L      MD      20
        -D
        L      MD      10
        +D
// měřeny jsou vždy 2 cykly OB1
        T      MD      30
        L      2
        /D
// aktuální hodnota cyklu

        T      MD      40

end:    NOP      0

// úprava délky cyklu pro test funkce

        CALL   "WAIT"
        WT:=MW50

```


Program nahrajte do PLC a ve VAT tabulce můžete sledovat následující hodnoty:

- PIW 362 – aktuální hodnota timestamp
- MD 10 – předchozí hodnota timestamp
- MD 40 – aktuální doba cyklu
- MW 50 – tato proměnná je navázána na funkci WAIT a prodlužuje čas cyklu o určitý počet μ s, které specifikujete. Všimněte si, že funkce WAIT neprodukuje hodnotu cyklu o přesný počet μ s, ale skokově.

	Address	Symbol	Display format	Status value	Modify value
1	// hodnota timestamp				
2	PIW 362		DEC	24598	
3	MD 10		DEC	L#24598	
4	// úprava délky cyklu v μ s				
5	MW 50		DEC	0	0
6	// aktuální doba cyklu				
7	MD 40		DEC	L#503	
8					

Protože odečítáme 2 hodnoty měřené stejným způsobem, můžeme uvažovat výslednou dobu cyklu OB1 měřenou s přesností v μ s. V případě, že bychom se zajímali o časovou hodnotu hrany signálu na svorkách modulu, bylo by nutno z hodnoty „timestamp“ odečíst dopravní zpoždění způsobené zpracováním signálu 1 μ s a časový filtr, který je parametrizovatelný v hardwarové konfiguraci ve vlastnostech modulu. Přednastavená hodnota je 2,56 μ s.