

9

9. ŘÍZENÍ BINÁRNÍCH VSTUPŮ A VÝSTUPŮ

9.1 CAN-BUS periferie pro binární vstupy a výstupy

element CANChannel	Nastavení kanálu CAN-BUSu	
atribut No	číslo kanálu CAN-BUS	
	1	pro periferie CAN-BUS
	xx	
atribut Active	je CAN kanál aktivní ?	
	0	CAN kanál neaktivní (default)
	1	CAN kanál aktivní
atribut PhysicalCanChannel	číslo fyzického CAN kanálu	
	1	pro periferie CAN-BUS
	xx	
atribut CanSpeed	komunikační rychlost CAN-busového kanálu [bit/s]	
	1000000	komunikační rychlost 1 MBd (default)
	500000	komunikační rychlost 0.5 MBd
	250000	komunikační rychlost 0.25 MBd
	125000	komunikační rychlost 0.125 MBd
atribut ServicePeriod	perioda obsluhy CAN-busového kanálu v mikro sekundách	
	250	perioda obsluhy CAN-BUSu po ¼ ms (default)
	500	perioda obsluhy CAN-BUSu po ½ ms
	1000	perioda obsluhy CAN-BUSu po 1 ms
atribut SyncPeriod	perioda posílání SYNCu jako násobek základního taktu	
	1	perioda vysílání SYNC po 1 ms (default)
	1,2,...,15	perioda vysílání SYNC
atribut InOut08Cnt	Počet periferií INOUT08 připojených k danému kanálu	
	0	žádná periferie INOUT08 (default)
	1,2,...,32	počet periferií INOUT08 – maximálně 32
atribut Kla50Cnt	Počet periferií KLA50 (tlačítka panelu) připojených ke kanálu	
	0	žádná periferie KLA50 (default)
	1,2,3,4	počet periferií KLA50 – maximálně 4
atribut KnobCnt	Počet točítek	
	0	(default)
	1,2	počet točítek

Nastavení **NODE-ID** pro systémové CAN-BUS periferie:

skupina (group)		Pořadové číslo jednotky ve skupině (board)						
		1	2	3	4	5	6	...
INOUT08	1	21h	22h	23h	24h	25h	26h	20h+board
Panel, KLA50, EPU	2	41h	42h	43h	44h			40h+board
Točítko	3	45h	46h					44h+board
Ruční panel	4	47h	48h					46h+board

9.2 Binární vstupy a výstupy u jednotek INOUT08

PLC program snímá vstupy a vysílá výstupy pro externí periferie INOUT08 jen pomocí publikovaných (veřejných) bitových polí. PLC program má možnost si bitové pole přepsat do vlastního bitového pole, ve kterém jsou pomocí instrukce DFM definovány vlastní názvy bitových proměnných nebo použít složitější způsoby adresace bitu (viz Kapitola 3, Základní instrukce jazyka - syntax instrukce LDR).

PLC program může řídit vyhodnocování chyb výstupů, může například provést reakci na zkrat sledovaného výstupu apod.

Příklad pro snímání binárních vstupů a vysílání výstupů:

```
;V deklaraci dat
IP0:  DFM  ALFA1, ALFA2, , , ,ALFA6, ,      ;bity z portu P1IN
IP1:  DFM  BETA1, BETA2, , , , , ,          ;bity z portu P7IN
OP0:  DFM  GAMA1, GAMA2, , , , , ,          ;bity na port P4OUT
```

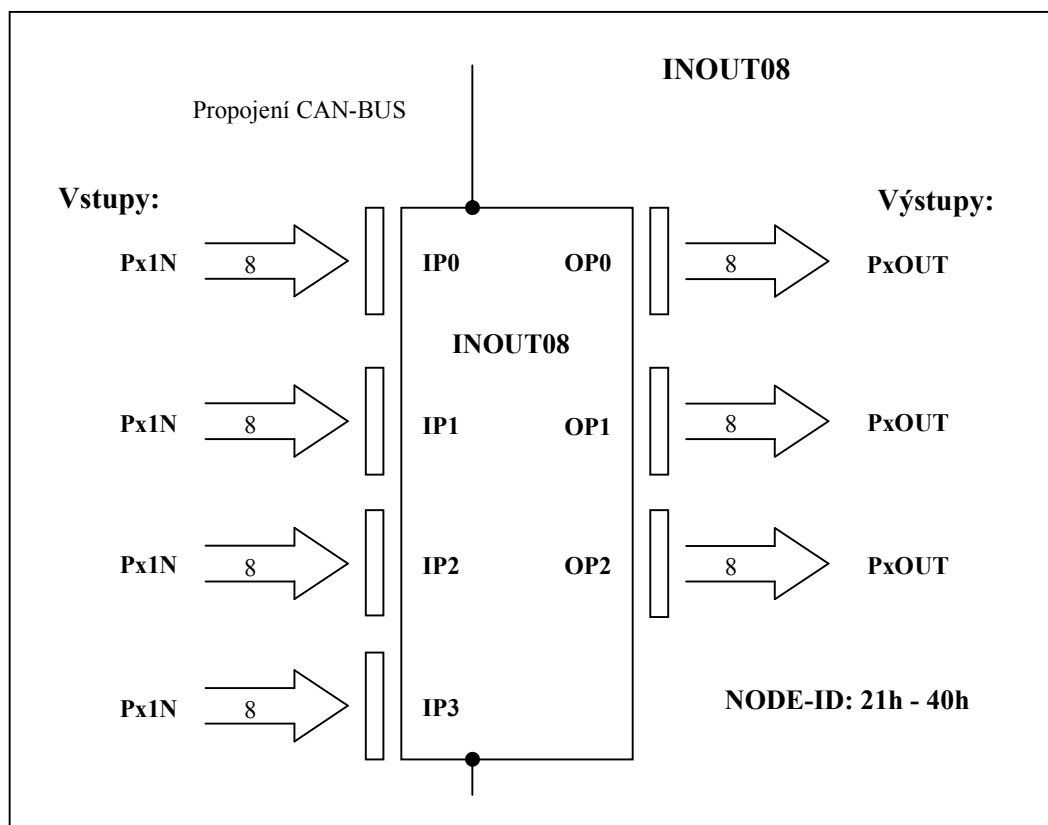
```
;V modulu MODULE_INPUT
    LOD    P1IN      ;přepis portu P1IN do buňky IP0
    STO    IP0
    LOD    P7IN      ;přepis portu P7IN do buňky IP1
    STO    IP1
```

```
;Na konci modulu MODULE_MAIN
    LOD    OP0        ;přepis buňky OP0 na port P4OUT
    STO    P4OUT
```

periferie 1.kanál	NODE-ID	vstup-popis	vstup PLC	výstup-popis	Výstup PLC
1. INOUT08 - CAN2	21h	IP0	P1IN	OP0	P0OUT
		IP1	P2IN	OP1	P1OUT
		IP2	P3IN	OP2	P2OUT
		IP3	P4IN		
2. INOUT08 - CAN2	22h	IP0	P5IN	OP0	P3OUT
		IP1	P6IN	OP1	P4OUT
		IP2	P7IN	OP2	P5OUT
		IP3	P8IN		
3. INOUT08 - CAN2	23h	IP0	P9IN	OP0	P6OUT
		IP1	P10IN	OP1	P7OUT
		IP2	P11IN	OP2	P8OUT
		IP3	P12IN		
4. INOUT08 - CAN2	24h	IP0	P13IN	OP0	P9OUT
		IP1	P14IN	OP1	P10OUT
		IP2	P15IN	OP2	P11OUT
		IP3	P16IN		
5. INOUT08 - CAN2	25h	IP0	P17IN	OP0	P12OUT
		IP1	P18IN	OP1	P13OUT
		IP2	P19IN	OP2	P14OUT
		IP3	P20IN		
6. INOUT08 - CAN2	26h	IP0	P21IN	OP0	P15OUT
		IP1	P22IN	OP1	P16OUT
		IP2	P23IN	OP2	P17OUT
		IP3	P24IN		
7. INOUT08 - CAN2	27h	IP0	P25IN	OP0	P18OUT
		IP1	P26IN	OP1	P19OUT
		IP2	P27IN	OP2	P20OUT
		IP3	P28IN		
8. INOUT08 - CAN2	28h	IP0	P29IN	OP0	P21OUT
		IP1	P30IN	OP1	P22OUT
		IP2	P31IN	OP2	P23OUT
		IP3	P32IN		
9. INOUT08 - CAN2	29h	IP0	P33IN	OP0	P24OUT
		IP1	P34IN	OP1	P25OUT
		IP2	P35IN	OP2	P26OUT
		IP3	P36IN		
10. INOUT08 - CAN2	2Ah	IP0	P37IN	OP0	P27OUT
		IP1	P38IN	OP1	P28OUT
		IP2	P39IN	OP2	P29OUT
		IP3	P40IN		
11. INOUT08 - CAN2	2Bh	IP0	P41IN	OP0	P30OUT
		IP1	P42IN	OP1	P31OUT
		IP2	P43IN	OP2	P32OUT
		IP3	P44IN		
12. INOUT08 - CAN2	2Ch	IP0	P45IN	OP0	P33OUT
		IP1	P46IN	OP1	P34OUT
		IP2	P47IN	OP2	P35OUT
		IP3	P48IN		
13. INOUT08 - CAN2	2Dh	IP0	P49IN	OP0	P36OUT
		IP1	P50IN	OP1	P37OUT
		IP2	P51IN	OP2	P38OUT
		IP3	P52IN		

14. INOUT08 - CAN2	2Eh	IP0	P53IN	OP0	P39OUT
		IP1	P54IN	OP1	P40OUT
		IP2	P55IN	OP2	P41OUT
		IP3	P56IN		
15. INOUT08 - CAN2	2Fh	IP0	P57IN	OP0	P42OUT
		IP1	P58IN	OP1	P43OUT
		IP2	P59IN	OP2	P44OUT
		IP3	P60IN		
16. INOUT08 - CAN2	30h	IP0	P61IN	OP0	P45OUT
		IP1	P62IN	OP1	P46OUT
		IP2	P63IN	OP2	P47OUT
		IP3	P64IN		
17. INOUT08 - CAN2	31h	IP0	P65IN	OP0	P48OUT
		IP1	P66IN	OP1	P49OUT
		IP2	P67IN	OP2	P50OUT
		IP3	P68IN		
18. INOUT08 - CAN2	32h	IP0	P69IN	OP0	P51OUT
		IP1	P70IN	OP1	P52OUT
		IP2	P71IN	OP2	P53OUT
		IP3	P72IN		
19. INOUT08 - CAN2	33h	IP0	P73IN	OP0	P54OUT
		IP1	P74IN	OP1	P55OUT
		IP2	P75IN	OP2	P56OUT
		IP3	P76IN		
20. INOUT08 - CAN2	34h	IP0	P77IN	OP0	P57OUT
		IP1	P78IN	OP1	P58OUT
		IP2	P79IN	OP2	P59OUT
		IP3	P80IN		
21. INOUT08 - CAN2	35h	IP0	P81IN	OP0	P60OUT
		IP1	P82IN	OP1	P61OUT
		IP2	P83IN	OP2	P62OUT
		IP3	P84IN		
22. INOUT08 - CAN2	36h	IP0	P85IN	OP0	P63OUT
		IP1	P86IN	OP1	P64OUT
		IP2	P87IN	OP2	P65OUT
		IP3	P88IN		
23. INOUT08 - CAN2	37h	IP0	P89IN	OP0	P66OUT
		IP1	P90IN	OP1	P67OUT
		IP2	P91IN	OP2	P68OUT
		IP3	P92IN		
24. INOUT08 - CAN2	38h	IP0	P93IN	OP0	P69OUT
		IP1	P94IN	OP1	P70OUT
		IP2	P95IN	OP2	P71OUT
		IP3	P96IN		
25. INOUT08 - CAN2	39h	IP0	P97IN	OP0	P72OUT
		IP1	P98IN	OP1	P73OUT
		IP2	P99IN	OP2	P74OUT
		IP3	P100IN		
26. INOUT08 - CAN2	3Ah	IP0	P101IN	OP0	P75OUT
		IP1	P102IN	OP1	P76OUT
		IP2	P103IN	OP2	P77OUT
		IP3	P104IN		
27. INOUT08 - CAN2	3Bh	IP0	P105IN	OP0	P78OUT
		IP1	P106IN	OP1	P79OUT
		IP2	P107IN	OP2	P80OUT
		IP3	P108IN		

28. INOUT08 - CAN2	3Ch	IP0	P109IN	OP0	P81OUT
		IP1	P110IN	OP1	P82OUT
		IP2	P111IN	OP2	P83OUT
		IP3	P112IN		
29. INOUT08 - CAN2	3Dh	IP0	P113IN	OP0	P84OUT
		IP1	P114IN	OP1	P85OUT
		IP2	P115IN	OP2	P86OUT
		IP3	P116IN		
30. INOUT08 - CAN2	3Eh	IP0	P117IN	OP0	P87OUT
		IP1	P118IN	OP1	P88OUT
		IP2	P119IN	OP2	P89OUT
		IP3	P120IN		
31. INOUT08 - CAN2	3Fh	IP0	P121IN	OP0	P90OUT
		IP1	P122IN	OP1	P91OUT
		IP2	P123IN	OP2	P92OUT
		IP3	P124IN		
32. INOUT08 - CAN2	40h	IP0	P125IN	OP0	P93OUT
		IP1	P126IN	OP1	P94OUT
		IP2	P127IN	OP2	P95OUT
		IP3	P128IN		



9.3 Řízení a vyhodnocování chyb externích periférií

Pro každou externí periférii INOUT08 je možno řadit nebo potlačit vyhodnocování chyb pro zobrazování na obrazovce systému, pro informaci PLC programu a pro časovou kontrolu funkčnosti externí periferie.

Každá periferie má přidělenou jednu systémovou strukturu **INOITS** (viz dále). PLC program má možnost pracovat se systémovými strukturami pomocí instrukcí **CAN_SYSIO_READ** a **CAN_SYSIO_WRITE**.

Struktura INOITS obsahuje také řídicí bitový záznam **CONTROL**

Popis bitů pro záznam CONTROL ve struktuře INOITS pro vyhodnocování chyb

CONTROL_INOUT	název bitu pro PLC	funkce
bit 0	IO_DIS_ERR	Hodnota log.0 povoluje a hodnota log.1 blokuje výpis chyb výstupů na obrazovce systému. Blokování nezpůsobí jiný efekt, takže PLC program si může chybu zjistit. Také dojde k odpojení zkratovaného výstupu. (předvolba řízení po zapnutí systému je v 1.dekádách strojních konstant R330-337)
bit 1	IO_DIS_ERPIS	Hodnota log.0 povoluje a hodnota log.1 blokuje zprávu o chybě výstupů pro PLC program. Pokud je v tomto bite nastavena hodnota log.0, systém bude trvale nulovat ty výstupy, které měly chybu (zkrat), až PLC program chybu nepotvrdí. Bitové pole pro PLC program je ERRLI_PxOUT a bude popsáno dále. (Předvolba řízení po zapnutí systému je v 2.dekádách strojních konstant R330-337.)
bit 2	IO_DIS_TMOUT	Hodnota log.0 povoluje a hodnota log.1 blokuje časovou kontrolu externí periferie. Blokování způsobí, že systém nezahlásí chybu například při odpojení nebo při ztrátě napájení periferie. Při obnovení činnosti bude periferie normálně funkční. (Předvolba řízení po zapnutí systému je v 3.dekádách strojních konstant R330-337.)
bit 6	IO_DIS_NULL	Hodnota log.0 povoluje a hodnota log.1 blokuje nulování všech periférií při vážné chybě této periferie.

CHYBA NÍZKÉ IMPEDANCE (ZKRAT)

Tato chyba je předávána do PLC programu (pokud není pro periférii nastaven bit IO_DIS_ERPIS na hodnotu log.1) v bitovém poli **ERRLI_PxOUT**. Pokud není zvláštní požadavek na ošetření chyb, PLC program nemusí na bity bitového pole ERRLI_PxOUT reagovat.

Bitové pole ERRLI_PxOUT má stejné označení s portami PxOUT (například ERRLI_P4OUT je chybové pole k portu P4OUT). Pro orientaci v chybovém poli se s výhodou může použít „složitější adresace bitu“ a přitom se využije označení odpovídajícího bitu z pole PxOUT.

Příklad:

```
OP4:  DFM    ALFA, , , BETA, , , ,
      LOD    OP4
      STO    P4OUT                ;výstup OP4 s bitem BETA

      LDR    ERRLI_P4OUT.BETA    ;test výstupu BETA na zkrat
      JLI    Error_Beta
```

NEZÁVISLÉ NASTAVOVÁNÍ VSTUPŮ A VÝSTUPŮ

Pro nezávislé nastavování vstupů a výstupů slouží bitová pole **SETL_PxxIN**, **SETH_PxxIN**, **SETL_PxxOUT** a **SETH_PxxOUT**:

Ruční řízení výstupů je nezávislé na nastavované hodnotě výstupu z PLC programu a řízení vstupů je nezávislé na fyzické sejmuté hodnotě vstupu. Najednou je možno ručně ovládat libovolný počet vstupů a výstupů.

bitové pole	význam
SETL_PxxIN	nastavení vstupů pro PLC na hodnotu log.0
SETH_PxxIN	nastavení vstupů pro PLC na hodnotu log.1
SETL_PxxOUT	nastavení výstupů z PLC na hodnotu log.0
SETH_PxxOUT	nastavení výstupů z PLC na hodnotu log.1

Pro orientaci v bitovém poli se s výhodou může použít „složitější adresace bitu“ a přitom se využije označení odpovídajícího bitu z pole PxOUT.

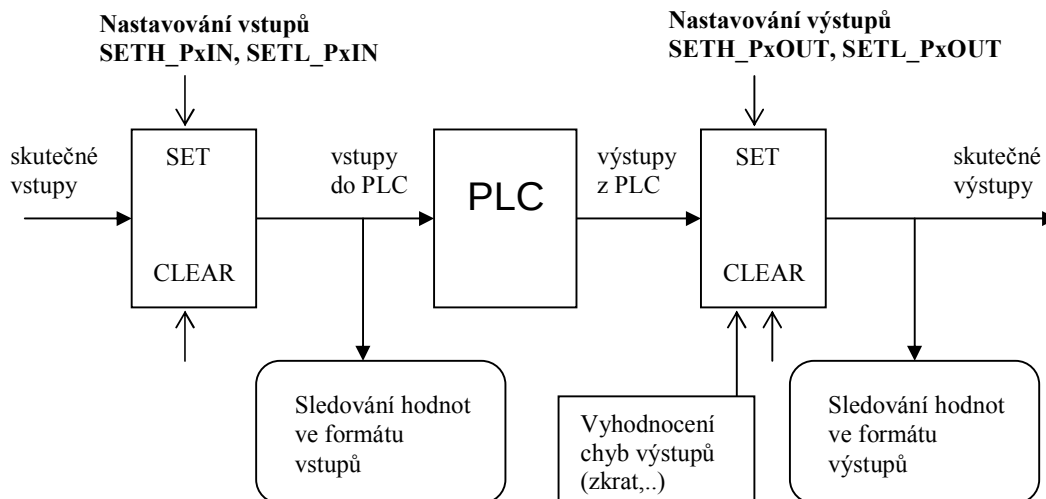
Nezávislé nastavování vstupů a výstupů se může použít provádět ladicím programem Wintechnol.

Pro zobrazování stavu vstupů a výstupů s ohledem na ruční ovládání slouží standardní formáty pro sledování vstupů, výstupů a matice panelu stroje. Pomocí těchto formátů můžeme získat přehled o všech vstupech a výstupech, které byly přednastaveny pomocí ručního ovládání.

Pokud má zobrazovaný bit políčko podbarveno **zeleně**, je tento bit právě ručně ovládán a jeho hodnota by měla odpovídat našemu ručnímu nastavení. Pokud je bit podbarven zeleně, ale jeho hodnota neodpovídá požadovanému ručnímu nastavení, pravděpodobně není příslušná periferní jednotka nakonfigurovaná (R231). Když se příslušný bit uvolní, zelené podbarvení se musí ztratit.

Pokud má zobrazovaný bit políčko podbarveno **červeně**, vykazuje tento bit právě chybu. Z největší pravděpodobnosti se jedná o chybu zkratu. Pokud chyba nebude potvrzena v příslušném chybovém poli od PLC programu, zůstane bit v chybě až do vypnutí systému.

Blokové schéma pro nezávislé nastavování vstupů a výstupů:



HLÁŠENÍ VÁŽNÝCH CHYB PRO PLC

Každá periferie má přidělenou jednu systémovou strukturu INOUTS (viz dále), v které se nachází také bajtové položky ERROR a ERR_CODE. V buňce ERROR je okamžitý stav „error registru 1001“ z jednotky a v buňce ERR_CODE je paměť poslední vzniknuté chyby.

PLC program má možnost číst struktury INOUTS pomocí instrukce CAN_SYSIO_READ (viz dále), nebo testovat společně bajtové buňky CAN_IO_ERROR_CODE a CAN_IO_ERROR_NUM:

Skupina 1. (INOUT08)

buňka (BYTE)	hodn.	význam
CAN_IO_ERROR_CODE	0, xx	bitové pole „error registru 1001“
CAN_IO_ERROR_NUM	1, 2,..	Pořadové číslo jednotky INOUT08
CAN_IO_TIMEOUT_CODE	0, 1	1 = time-out pro INOUT08
CAN_IO_TIMEOUT_NUM	1, 2,..	Pořadové číslo jednotky INOUT08

Skupina 2. (KLA50)

buňka (BYTE)	hodn.	význam
CAN_MAT_ERROR_CODE	0, xx	bitové pole „error registru 1001“
CAN_MAT_ERROR_NUM	1, 2,..	Pořadové číslo jednotky KLA50
CAN_MAT_TIMEOUT_CODE	0, 1	1 = time-out pro KLA50
CAN_MAT_TIMEOUT_NUM	1, 2,..	Pořadové číslo jednotky KLA50

Skupina 3. (TOC)

buňka (BYTE)	hodn.	význam
CAN_TOC_ERROR_CODE	0, xx	bitové pole „error registru 1001“
CAN_TOC_ERROR_NUM	1, 2,..	Pořadové číslo jednotky TOC
CAN_TOC_TIMEOUT_CODE	0, 1	1 = time-out pro TOC
CAN_TOC_TIMEOUT_NUM	1, 2,..	Pořadové číslo jednotky TOC

Definice chyb v „error registru“ pro jednotky INOUT08, KLA50 a TOC

bit	označení pro PLC	význam
bit 0.	ERR_IO8_ERROR	Bit je nastaven při chybě vždy
bit 1.	ERR_IO8_SHORT	Chyba výstupů – zkrat *)
bit 2.	ERR_IO8_LOW_SUPPLY	Malé napájecí napětí
bit 3.	ERR_IO8_TEMP_OVER	Překročena teplota
bit 4.	ERR_IO8_COMM	Chyba komunikace (systém zjistí sám, například „time-out“)
bit 5.	-	
bit 6.	-	
bit 7.	ERR_IO8_INPUT	Chyba vstupů

*)

Chyba zkratu výstupů se nepočítá mezi vážné chyby periferie (nenulují se výstupy na všech periferiích). Systém při chybě zkratu hlásí chybu s číslem periferie. Automaticky si vyžádá informaci o zkratu z periferie a nastaví ve struktuře INOUTS bitové pole **SHORT_OUT0**, **SHORT_OUT1** a **SHORT_OUT2**. Na obrazovce systému možno zjistit zkratované výstupy ve formátu sledování výstupů. Výstupy, které jsou ve zkratu budou podbarveny červeně.

Příklad:

Test funkčnosti jednotek INOUT08:

```
LDR    CAN_IO_ERROR_CODE.ERR_IO8_ERROR    ;společný bit pro errorry
JL1    ErrorIO
```

9.4 Sdílený přístup pro CAN kanál

Sdílené použití CAN kanálu v PLC programu znamená, že PLC program používá CAN-BUS kanál společně se systémem, nebo alespoň pro jeho řízení používá systémové prostředky. Inicializaci CAN kontroléru a řízení vysílání a příjmu paketů také provádí systém. PLC program jen žádá a zařazení svých paketů do fronty pro vyslání a čte frontu příjmutých paketů.

Masky bitů pro vyhodnocování		
Symbol	Maska	Význam
CAN_ERR_OK	00h	Vše v pořádku
CAN_ERR_XMTFULL	01h	Plná fronta pro vysílání
CAN_ERR_RCVEMPTY	02h	Nepřijala se nová zpráva

INSTRUKCE PRO PRÁCI S CAN-BUSEM SE SDÍLENÝM PŘÍSTUPEM

Funkce pro sdílenou obsluhu CAN-BUSu jsou do PLC programu implementovány jako 2 speciální instrukce.

Instrukce pro sdílený přístup	Popis
CAN_SEND	Zařazení paketu do fronty pro vysílání na periférii
CAN_RECV	Čtení paketu z příjmové fronty

instrukce	CAN_SEND
-----------	----------

```
funkce    Zařazení paketu do výstupní fronty

syntax    CAN_SEND    can, pmsg

1.parametr „can“      číslo CAN kanálu
2.parametr „pmsg“     pointer na paket
```

Instrukce slouží pro zařazení paketu do výstupní fronty paketů, které jsou vysílány přes CAN kanál na periférii. Instrukce po vykonání vrátí v datovém registru výsledek, který je složen z masek chyb (popsaných dříve). Pokud při řazení nevznikla chyba, vrácená hodnota bude CAN_ERR_OK (00h). pokud se paket do fronty nepodařilo zařadit (fronta je přeplněna), bude vrácená hodnota CAN_ERR_XMTFULL (01h)

Význam parametrů:

Parametr	Hodnota (příklad)	Význam
can	1, 2	Logický CAN kanál (CAN1, CAN2)
pmsg	CANBUFF_OUT	Pointer, který ukazuje na vyhrazené místo 12 bajtů. Návěští u instrukce STR, kde je místo pro paket

Příklad:

```

CANBUFF_OUT:      STR    12                ;výstupní paket

                  CAN_SEND  2, CANBUFF_OUT    ;Zařazení paketu
                  EQ        CAN_ERR_OK        ;Bylo zařazeno ?
                  JL1       OK                ;zařazeno
                                           ;nezařazeno

```

Podrobnější příklad bude uveden v závěru této kapitoly

instrukce	CAN_RECV
-----------	----------

funkce Čtení paketu z příjmové fronty (Nedoporučuje se používat)

syntax **CAN_RECV** can, pmsg

1.parametr „can“ číslo CAN kanálu
2.parametr „pmsg“ pointer na paket

Instrukce je uvedena jen pro kompatibilitu a nedoporučuje se používat. Místo ní je zavedena instrukce **CAN_RECV_REQ**, které popis je dále.

Instrukce slouží pro čtení paketu z fronty paketů, které byly přijaty z periferii přes CAN kanál. Instrukce po vykonání vrátí v datovém registru výsledek, který je složen z masek chyb (popsaných dříve). Pokud při čtení nevznikla chyba, vrácená hodnota bude CAN_ERR_OK (00h). pokud se žádný paket pro PLC nepřijal, bude vrácená hodnota CAN_ERR_RCVEMPTY (02h)

Význam parametrů:

Parametr	Hodnota (příklad)	Význam
can	2	Logický CAN kanál (CAN2)
pmsg	CANBUFF_IN	Pointer, který ukazuje na vyhrazené místo 12 bajtů. Návěští u instrukce STR, kde je místo pro paket

Příklad:

```

CANBUFF_IN:      STR    12                ;vstupní paket

                  CAN_RECV  2, CANBUFF_IN    ;Příjem paketu
                  EQ        CAN_ERR_OK        ;Bylo přijato ?
                  JL1       OK                ;přijato
                                           ;nic se nepřijalo

```

Podrobnější příklad bude uveden v závěru této kapitoly

instrukce	CAN_RECV_REQ
------------------	---------------------

funkce **Požadavek pro volání událostní procedury po příjmu paketu**

syntax **CAN_RECV_REQ** **can, objl, objh, idl, idh, proc, pmsg**

1.parametr	„can“	číslo CAN kanálu
2.parametr	„objl“	minimální ID objektu zprávy
3.parametr	„objh“	maximální ID objektu zprávy
4.parametr	„idl“	minimální NODE-ID
5.parametr	„idh“	maximální NODE-ID
6.parametr	„proc“	název událostní procedury
7.parametr	„pmsg“	pointer na paket

Instrukce má návratové hodnoty:

RLO=1 ... operace dokončena bez chyb
RLO=0 ... operace dokončena, ale požadavek nebyl akceptován
DR (DWRD) ... handle žádosti o volání událostní procedury

Instrukce slouží pro nastavení žádosti o volání událostní procedury v PLC po příjmu určitých „objednaných“ paketů z CAN kanálu (nastavení „CALLBACK“ volání). Když instrukce proběhne bez chyb (RLO=1), tak si PLC program může uchovat „handle žádosti“ do vlastní proměnné typu DWORD. Tuto proměnnou může PLC program využít v případě, že by později chtěl žádost o volání procedury zrušit.

Instrukce může být použita v modulu MODULE_INIT.

Význam parametrů:

Parametr	Hodnota (příklad)	Význam
can	1, 2	Logický CAN kanál (CAN1, CAN2)
objl	například 580h	Minimální ID objektu zprávy, které příjem se očekává. Jedná se o horní 4 bity z 11-bitové adresy COB-ID (podle normy CanOpen)
objh	-	Maximální ID objektu zprávy, které příjem se očekává. Parametr je nepovinný. Musí se uvést, když je požadován příjem intervalu zpráv. Pokud parametr nebude uveden (bude nahrazen znakem „-“) tak se bude očekávat příjem zpráv jednoho objektu.
idl	například 21h	Minimální NODE-ID objektu zprávy, které příjem se očekává. Jedná se o 7 dolních bitů z 11-bitové adresy COB-ID.
idh	-	Maximální NODE-ID objektu zprávy, které příjem se očekává. Parametr je nepovinný. Musí se uvést, když je požadován příjem intervalu zpráv. Pokud parametr nebude uveden (bude nahrazen znakem „-“) tak se bude očekávat příjem zpráv jednoho NODE-ID.
proc	například P_ON_CANRECV	Název událostní procedury, která se spustí po příjmu požadovaného paketu. Jedná se o standardní proceduru napsanou v PLC programu. Procedura po zavolání bude mít v DR registru nastaveno číslo logického CAN kanálu (1, 2).
pmsg	například sCANRECV	Pointer, který ukazuje na vyhrazené místo 12 bajtů. Návěští u instrukce STR (kde je místo pro paket) do kterého se naplní celý přijatý paket, který odpovídá požadovanému COB-ID.

Příklad bude uveden později.

Struktura COB-ID (Communication **O**bject **I**dentifier) podle normy CANopen :

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	x	x	x	x	x	Function (0-0Fh)				NODE-ID (0-7Fh)						

instrukce	CAN_RECV_DEL	
funkce	Zrušení požadavku pro volání událostní procedury příjmu paketu	
syntax	CAN_RECV_END	handle
1.parametr	„handle“	handle žádosti o volání událostní procedury

Instrukce má návratové hodnoty:

RLO=1 ... operace dokončena bez chyb
RLO=0 ... operace dokončena, ale požadavek nebyl akceptován

Instrukce slouží pro zrušení žádosti volání událostní procedury v PLC po příjmu paketů z CAN kanálu. Jako parametr instrukce se musí zadat název proměnné typu DWORD, kde je uloženo „handle žádosti“, které vrátila instrukce CAN_RECV_REQ.

Příklad:

Žádost o zachytávání odpovědi na SDO pakety (objekt 580h) pro jednotku INOUT08 s NODE-ID=21h.

```
;Buňka DWORD pro uložení Handle
dwHANDLE:   ds      4
```

```
;Místo pro přijatý paket:
sCANRECV:   STR     12
```

```
;Událostní procedura pro zpracování odpovědi na SDO paket
PROC_BEGIN  P_ON_CANRECV
    LOD      word.sCANRECV.CAN_DATA+1      ;Index zprávy
    EQ       cnst.1008h
    ;....
    ;....
PROC_END     P_ON_CANRECV
```

```
;Žádost o volání procedury P_ON_CANRECV
CAN_RECV_REQ 2, 580h, -, 21h, -, P_ON_CANRECV, sCANRECV
STO          dwHANDLE
```

```
;Zrušení žádosti o volání procedury
CAN_RECV_END dwHANDLE
```

PŘÍKLAD PRO VSTUPY A VÝSTUPY NA NESYSTÉMOVOU CAN PERIFERII SE SDÍLENÝM PŘÍSTUPEM

Příklad:

ID adresa periferie je nastavena na 10h.

V deklaraci dat:

```
CANBUFF_IN:      STR 12      ;misto na prijem paketu
CANBUFF_OUT:     STR 12      ;misto pro vysilani paketu
```

V modulu MODULE_MAIN:

```

CAN_RECV 2,CANBUFF_IN          ;Cteni paketu
EQ      CAN_ERR_OK             ;Je novy paket ?
JL0     NIC_NEPRIJATO

;Test na prijem paketu ID=10h+1, delka=2, Byte1=Opc=1, Byte2=Input
LOD     WORD.CANBUFF_IN.CAN_ID ;Test ID
ANDB    CNST.7FFh              ;11 bitove ID
EQ      CNST.10h+1             ;Odpoved od ID=10h ?
JL0     CAN_OTH
LOD     BYTE.CANBUFF_IN.CAN_LEN ;Delka
EQ      CNST.2                 ;Je delka = 2 ?
JL0     CAN_OTH
LOD     BYTE.CANBUFF_IN.CAN_DATA+0 ;Opcode
EQ      CNST.1                 ;Opcode 1 pro vstupy ?
JL0     CAN_OTH
LOD     BYTE.CANBUFF_IN.CAN_DATA+1 ;Sejmuti vstupnich dat !
STO     IP_CAN                 ;Misto pro nova data
CAN_OTH:                                ;Jiny paket, opakujeme

;**** Obsluha CAN-BUSu ... vysilani vystupu ****
LOD     CNST.10h               ;Vyslani na ID=10h
STO     WORD.CANBUFF_OUT.CAN_ID
LOD     CNST.0h                ;Neni "remote request"
STO     BYTE.CANBUFF_OUT.CAN_RTR
LOD     CNST.2                 ;delka paketu
STO     BYTE.CANBUFF_OUT.CAN_LEN
LOD     CNST.2                 ;Opcode = 2
STO     BYTE.CANBUFF_OUT.CAN_DATA+0
LOD     IP0                    ;data pro vyslani
STO     BYTE.CANBUFF_OUT.CAN_DATA+1

CAN_SEND 2,CANBUFF_OUT        ;Zarazeni paketu do fronty
```

9.5 Ovládání systémových CAN-BUS periférií

PLC program má možnost číst a nastavovat struktury pro systémové CAN-BUS periferie, jako jsou INOUT08 a KLA50. Kromě toho zůstává možnost použít instrukce CAN_SEND a CAN_RECV pro sdílený přístup.

Funkce pro přístup k systémovým strukturám CAN-BUS periférií jsou do PLC programu implementovány jako 2 speciální instrukce.

Instrukce pro sdílený přístup	Popis
CAN_SYSIO_READ	Čtení prvku z pole systémových struktur pro CAN-BUS periferie
CAN_SYSIO_WRITE	Zápis prvku do pole systémových struktur pro CAN-BUS periferie

Každá periferie má přidělenou jednu strukturu INOUTS. Prvky struktury je možno sledovat také pomocí diagnostické obrazovky „CAN-BUS peripherals diagnostic“. Pomocí této obrazovky je umožněno také vysílat a přijímat SDO pakety.

```

INOUTS  STRUC
        SIZE          DW      0      ;velkost
        PRESENT        DB      0      ;Je jednotka pritomna ?
        MODE           DB      0      ;mod (0=standard, 1=analog, 2=matice)
        VENDORID       DD      0      ;SDO 1018
        DEVICENAME     DD      0      ;SDO 1008
        HWVERSION      DD      0      ;SDO 1009
        SWVERSION      DD      0      ;SDO 100A

        IN0            DB      0      ;vstupy
        IN1            DB      0
        IN2            DB      0
        IN3            DB      0

        OUT0           DB      0      ;vystupy
        OUT1           DB      0
        OUT2           DB      0
                   DB      0

        ANL0           DB      0      ;analog
        ANL1           DB      0
        ANL2           DB      0
        ANL3           DB      0
        ANL4           DB      0
        ANL5           DB      0
        ANL6           DB      0
        ANL7           DB      0

        ERROR          DB      0      ;error registr 1001 - okamzity
        ERR_CODE       DB      0      ;kod chyby ... zapise pri vzniku chyby

        ERR_STAT       DB      0      ;hlaseni chyb - record E_IOR
                                ;bit0(IO_ER_HLI_ACT) = chyba (INOUT_ERR_CODE)
                                ;bit1 (IO_ER_TM_ACT)= time-out
                                ;bit2 (IO_ER_HLI) = chyba (INOUT_ERR_CODE)
                                ;bit3 (IO_ER_TMOUT) = time-out pamet
                                ;bit4 (IO_ER_NULL) = zadost o nulovani periferii

        CONTROL        DB      0      ;rizeni - record C_IOR
                                ;bit0 (IO_DIS_ERR) = blokovani vypisu chyb
                                ;bit1 (IO_DIS_ERPIS) = blokovani chyb pro PLC
                                ;bit2 (IO_DIS_TMOUT) = blokovani chyb time-out
                                ;bit6 (IO_DIS_NULL) = blokovani nulovani periferii

        COUNT          DW      0      ;citac pro time-out

        SEND_REQ       DB      0      ;zadost o vyslani SDO paketu INOUT_SEND
        RESP_COUNT     DB      0      ;citac prijmu
        SHORT_SEND_REQ DB      0      ;zadost o informaci zkratu

        SHORT_OUT0     DB      0      ;informace o zkratu
        SHORT_OUT1     DB      0

```

```

SHORT_OUT2      DB      0

SEND             TCANMSG2 <0> ;paket pro vyslani
RESP            TCANMSG2 <0> ;response paket

RPDO_COUNT      DW      0      ;dgn cistac zarazenych RPDO paketu pro vyslani
TPDO_COUNT      DW      0      ;dgn cistac prijmutych TPDO paketu

PxIN             DW      0      ;cislo vstupniho PLC portu 1=PlIN
PxOUT           DW      0      ;cislo vystupniho PLC portu 1=PlOUT

lpINOUT_PxIN     DW 2 DUP (0)  ;pointry na PLC oblasti
lpINOUT_PxOUT    DW 2 DUP (0)
lpINOUT_ERRHI_PxOUT DW 2 DUP (0)
lpINOUT_SETH_PxOUT DW 2 DUP (0)
lpINOUT_SETH_PxIN DW 2 DUP (0)

MODE_ACT        DB      0      ;mod aktual
MODE_FAZE       DB      0

TL0             DB      0      ;tlacitka matice
TL1             DB      0
TL2             DB      0
TL3             DB      0

IRC0            DW      0      ;irc
IRC1            DW      0

INOUTS ENDS

```

Definice chyb v „error registru“ pro jednotky INOUT08, KLA50 a TOC

bit	označení pro PLC	význam
bit 0.	ERR IO8 ERROR	Bit je nastaven při chybě vždy
bit 1.	ERR IO8 SHORT	Chyba výstupů – zkrat *)
bit 2.	ERR IO8 LOW SUPPLY	Malé napájecí napětí
bit 3.	ERR IO8 TEMP OVER	Překročena teplota
bit 4.	ERR IO8 COMM	Chyba komunikace (systém zjistí sám, například „time-out“)
bit 5.	-	
bit 6.	-	
bit 7.	ERR IO8 INPUT	Chyba vstupů

Definice bitů ve „statusu“ jednotek INOUT08, KLA50 a TOC:

bit	označení pro PLC	význam
bit 0.	IO ER HLI ACT	Chyba - kód je v error-registru (okamžitý stav)
bit 1.	IO ER TM ACT	Chyba Time-Out (okamžitý stav)
bit 2.	IO ER HLI	Chyba - kód je v error-registru (paměť)
bit 3.	IO ER TMOUT	Chyba Time-Out (paměť)
bit 4.	IO ER NULL	Žádost o nulování všech periférií

Definice bitů v „control-registru“ jednotek INOUT08, KLA50 a TOC:

bit	označení pro PLC	význam
bit 0.	IO DIS ERR	Blokování výpisu chyb výstupů
bit 1.	IO DIS ERPIS	Blokování hlášení chyb pro PLC
bit 2.	IO DIS TMOUT	Blokování chyb Time-Out
bit 6.	IO DIS NULL	Blokování nulování všech periférií při vážné chybě

instrukce	CAN_SYSIO_READ
------------------	-----------------------

funkce Čtení prvku z pole systémových struktur pro CAN-BUS periferie

syntax CAN_SYSIO_READ can, group, board, item

1.parametr „can“ číslo CAN kanálu
 2.parametr „group“ skupina systémových CAN-BUS periférií
 3.parametr „board“ pořadové číslo jednotky ve skupině
 4.parametr „item“ identifikátor prvku systémové struktury INOUTS

Instrukce **CAN_SYSIO_READ** slouží pro načtení jednoho prvku z pole struktur INOUTS. Instrukce načte do DR registru bajt, word nebo double-word podle typu zadaného prvku. Pokud se instrukce provede bez chyb (prvek ve struktuře se najde), tak vrací RLO registr nastaven na hodnotu 0. Pokud RLO registr je nastaven na hodnotu 1, potom instrukce skončila s chybou a data v DR registru nejsou platná.

Význam parametrů:

Parametr	Hodnota (příklad)	Význam
can	2	Logický CAN kanál (CAN2)
group	1, 2, 3	Skupina CAN-BUS periférií. 1 = jednotky vstupů a výstupů INOUT08 2 = jednotky maticových vstupů KLA50 3 = panálek s točátkem a displejem CAN-BUS
board	1, 2, ..., 32	Pořadové číslo jednotky ve skupině. Každá skupina má maximálně 32 jednotek. NODE-ID jednotky musí být nastaveno s ohledem na pořadové číslo jednotky a s ohledem na skupinu.
item	(MODE, IN0, ...)	Identifikátor prvku struktury podle definice struktury INOUTS

Příklad:

Načtení analogového napětí z 3.jednotky INOUT08 (NODE-ID = 23h):

```
CAN_SYSIO_READ      2, 1, 3, ANL0      ;1.skupina,3.jednotka,prvek ANL0
JL1                      Error
```


instrukce	CAN_SYSIO_WRITE
------------------	------------------------

funkce **Zápis do prvku v poli systémových struktur pro CAN-BUS periferie**

syntax **CAN_SYSIO_WRITE can, group, board, item**

1.parametr	„can“	číslo CAN kanálu
2.parametr	„group“	skupina systémových CAN-BUS periferií
3.parametr	„board“	pořadové číslo jednotky ve skupině
4.parametr	„item“	identifikátor prvku systémové struktury INOUTS

Instrukce **CAN_SYSIO_WRITE** slouží pro zápis jednoho prvku do pole struktur INOUTS. Instrukce запиše obsah DR registru do struktury jako bajt, word nebo double-word podle typu zadaného prvku. Pokud se instrukce provede bez chyb (prvek ve struktuře se najde), tak vrací RLO registr nastaven na hodnotu 0. Pokud RLO registr je nastaven na hodnotu 1, potom instrukce skončila s chybou a data se do struktury nezapsala.

Význam parametrů:

Parametr	Hodnota (příklad)	Význam
can	2	Logický CAN kanál (CAN2)
group	1, 2, 3	Skupina CAN-BUS periferií. 1 = jednotky vstupů a výstupů INOUT08 2 = jednotky maticových vstupů KLA50 3 = panálek s točítkem a displejem CAN-BUS
board	1, 2, ..., 32	Pořadové číslo jednotky ve skupině. Každá skupina má maximálně 32 jednotek. NODE-ID jednotky musí být nastaveno s ohledem na pořadové číslo jednotky a s ohledem na skupinu.
item	(MODE, IN0, ...)	Identifikátor prvku struktury podle definice struktury INOUTS

Příklad:

Namódování 4.jednotky INOUT08 pro snímání analogových vstupů:

LOD	CNST.1	;mode=1 pro analogové vstupy
CAN_SYSIO_WRITE	2, 1, 4, MODE	;1.skupina,4.jednotka,prvek MODE
JL1	Error	

9.6 Přímá komunikace z PLC pomocí SDO paketů

PLC program má možnost pomocí sdíleného použití CAN kanálu, přímo ovládat nebo zjistit libovolné informace z periférií INOUT08. PLC program používá CAN-BUS kanál společně se systémem a pro jeho řízení používá systémové prostředky. PLC program jen žádá a zařazení svých paketů do fronty pro vyslání a čte frontu přijmutých paketů.

Funkce pro sdílenou obsluhu CAN-BUSu jsou (už byly popsány):

Instrukce pro sdílený přístup	Popis
CAN_SEND	Zařazení paketu do fronty pro vyslání na periférii
CAN_RECV	Čtení paketu z příjmové fronty

SDO pakety pro komunikaci s perifériemi jsou odvozeny z normy CAN-OPEN Cíla DS301 a Cíla DS401.

Použité COB-ID pro PDO pakety:

jednotka	paket	COB-ID
INOUT08	RPDO1	220h + board, 221h, ...
	TPDO1	1A0h + board, 1A1h, ...
INOUT08 + analog.vstupy:	TPDO2	2A0h + board, 2A1h, ...
	RPDO1	240h + board, 241h, ...
KLA50	TPDO1	1C0h + board, 1C1h, ...
	RPDO1	244h + board, 245h, ...
TOC	TPDO1	1C4h + board, 1C5h, ...
	RPDO1	244h + board, 245h, ...

Komunikovat pomocí SDO paketů je umožněno i pomocí diagnostické obrazovky pro CAN-BUS periférie. Tato komunikace slouží jen pro diagnostické účely.

Dále uvádíme úplný přehled implementovaných SDO paketů u jednotky INOUT08:

```

INOUT08 software v1.00:
-----

Implemented SDO's:
-----

unsigned long DeviceType[1];      /* 1000h */
unsigned char ErrorRegister[1];   /* 1001h */
unsigned char Devname[20];        /* 1008h - Device name          */
unsigned char HWVer[20];          /* 1009h - Hardware version      */
unsigned char SWVer[20];          /* 100ah Software version        */
unsigned long ConsHeartbeat[1];   /* 1016h Consumer Heartbeat t.  */
unsigned int HeartbeatTime[1];   /* 1017h Producer Heartbeat t.  */
unsigned long VendorID[1];        /* 1018h Identity - VendorID     */
unsigned char ErrBeh[3];          /* 1029h Error behaviour         */
unsigned long RPDOCommPar[2];     /* 1400h */
PDO_MAP_STRUC RPDOMapPar[8];      /* 1600h */
unsigned long TPDOCommPar[2];     /* 1800h */
unsigned long TPDO2CommPar[2];    /* 1801h */
PDO_MAP_STRUC TPDOMapPar[8];      /* 1a00h */
PDO_MAP_STRUC TPDO2MapPar[4];     /* 1a01h */
unsigned char StopPgm[1];         /* 1f51h Stop program cmd       */
unsigned long Filler[1];          /* Only for align                */
unsigned char RealOutput[4];      /* 2100h */
unsigned char ShortedOutput[4];   /* 2101h */
unsigned char EnableFast[4];      /* 2102h */
unsigned char FastTrap[4];        /* 2103h */
unsigned long InpErrCnt[1];       /* 2104h */
unsigned int ADCOut[28];          /* 2105h */
unsigned int TimeCInp[1];         /* 2106h */

```

```

unsigned int TimeCShort[1];      /* 2107h */
unsigned char MatrixInp[4];      /* 2108h */
unsigned int MatOutCnt[1];       /* 2109h max 8 */
unsigned int SyncGuard[1];       /* 210ah Max. sync time */
unsigned int CapVoltage[1];      /* 210bh */
unsigned int UnderVoltage[1];    /* 210ch */
unsigned char ReadInput[4];      /* 6000h */
unsigned char PolarInput[4];     /* 6002h */
unsigned char FilterInput[4];    /* 6003h */
unsigned long GlobalIntEDig[1];  /* 6005h - boolean! */
unsigned char IntMANYChInput[4]; /* 6006h */
unsigned char IntMLtHInput[4];   /* 6007h */
unsigned char IntMhtLInput[4];   /* 6008h */
unsigned char WriteOutput[4];    /* 6200h */
unsigned char PolarOutput[4];    /* 6202h */
unsigned char ErrModeOutput[4];  /* 6206h */
unsigned char ErrValueOutput[4]; /* 6207h */
unsigned char FilterOutput[4];   /* 6208h */
unsigned int AnalogInput[4];     /* 6401h */
unsigned char AnalogIMfSp[8];    /* 6404h */

```

SDO's 1000h - 1f51, 6000h - 6404 are described in
Cia DS-301 and Cia DS-401. SDO's 2100h - 2109h
are device specific and are described below:

Device specific SDO's:

RealOutput[4] 2100h -
Real state on the digital output pins.
1 - switched on
0 - switched off
(Read only)

ShortedOutput[4] 2101h -
Digital outputs, which are switched off due to
detected short circuit.
1 - Output is switched off due to the short circuit.
0 - No short circuit detected.
(Read only)

EnableFast[4]; 2102h -
Enables autonomous fast reaction of the output pin
to the digital input pin:

RealOutput = (WriteOutput & ~FastTrap) ^ PolarOutput;
FastTrap = (FastTrap | ReadInput) & ~WriteOutput;

1 - Autonomous fast reaction to the input pin enabled
0 - Autonomous fast reaction disabled
(Read/Write)

FastTrap[4]; 2103h -
Input latch for the autonomous output reaction to the
input pin. Set by ReadInput. Cleared by WriteOutput.
(Read only)

InpErrCnt[1]; 2104h -
Count of detected input errors.
(Read only)

ADCOut[28]; 2105h -
Raw input from A/D converter.
Subindex 0 : Length of the array - 28
Subindex 1..24 : Voltage on the output pins:
0xffff - Voltage equal to the 24V supply voltage.
(Read only)

TimeCInp[1]; 2106h -
Time constant [in uS] for digital input filter.
Used for the inputs which have set 1 in
FilterInput (6003h). Minimal time constant is
approx. 1000uS.
(Read/Write)

TimeCShort[1]; 2107h -
Time constant [in uS] for output short circuit

detection. If short circuit condition lasts longer then this time constant, the output is switched off. (Read/Write)

```
MatrixInp[4];          2108h -
Up to 4 switched on inputs conected in matrix:
Subindex 0 : Length of the array - 4.
Subindex 1..4 : Switched on inputs.
0xff - No input switched on.
Other value - Code of the matrix input switched on.
For matrix input are used inputs IP0/0..IP0/7 and
IP1/0..IP1/7, and oututs from OP2/0.
(Read only)
```

```
MatOutCnt[1];          2109h -
Count of digital outputs, which are used for the matrix
input scan. Maximum value = 8.
0 - No output used for matrix input scan.
1 - OP2/0 used.
2 - OP2/0, OP2/1 used.
...
8 - OP2/0 .. OP2/7 used.
(Read/Write)
```

SyncGuard[1]; 210ah -
 Checking of the time periode between sync messages.
 0 - No checking of sync messages.
 Other value - maximum allowable time between
 sync messages in 1.024 milliseconds .
 If there is too long time without sync message,
 communication error in the status is set,
 and outputs are set to the state defined by SDO 6207h.
 (Read/Write)

CapVoltage[1]; 210bh -
Voltage on the supply capacitor in [0.1V].
Voltage may vary with current consumption of the board.
(Read only)

UnderVoltage[1]; 210ch -
If CapVoltage is less then this value, low
supply voltage error is reported.
(Read/Write)

```
Encoder[3];          210dh -
Value of encoder counter:
Subindex 0 : Length of the array - 3.
Subindex 1 : Encoder counter 1 (unsigned int).
Subindex 2 : Encoder counter 2 (unsigned int).
Subindex 3 : Encoder in wheel mode (unsigned int).
Counter is cleared (0000) at reset.
Overflows from 0xFFFF to 0000 and
underflows from 0000 to 0xFFFF.
(Read only)
```

```
VOvolt[1];          210eh -
Control of VO output voltage (PWM D/A converter - Display Brightness):
0xff - approx. 5V
0x00 - approx. 0V
(Read/Write)
```

```

Display[19];          210fh -
  Output to the graphic display:
Subindex 0 : Length of the array - 19.
Subindex 1 : 1 - initialise, 2 - display test.
Subindex 2 : Display enable/disable. 0 - disable, 1 enable.
Subindex 3 : Display RAM start. 0 - 63 (location of 1.displayed byte).
Subindex 4 : Display back light.          0 - off 1 - on.
Subindex 5 : reserved
# Subindex 6 : Row - in pixels, from upper left corner, 0-63.
# Subindex 7 : Column - in pixels, from upper left corner, 0-127.
# Subindex 8 : Mask(8 bits): 0 - keep previous pixel state, 1 - write
# Subindex 9 : Inversion(8 bits): 1 - pixel from character generator
                        inverted.
# Subindex 10 :Text size/Graphic 0 - single size text (6x8 dots/char)
                        1 - double size text (12x16)
                        8 - graphic

```

Subindex 11 : Count of bytes (or chars) in subindex 12 - 19 to be written.
 (Not active for SDO transfer - every byte will be written)

Subindex 12 - 19 : Up to 8 byte data to be written on the display.
 If graphic is selected (subindex 11), bit 0 of the 1.byte
 is written in the position specified at subindex 6 and 7.
 Bits 1-7 are written below it at the same column.
 Next byte is written at the right side.
 If alphanumeric is selected, upper left corner of 1.char is
 written to the position specified at subindex 6 and 7.
 Next char is written to the right side.
 At the SDO transfer, only setting of subindex 12 sets column
 as is at subindex 7. Setting of subindex 13 - 19 writes to
 the next position at the right side.

#-subindex can be mapped in RPDO packet.
 (Read/Write)

WheelMode[1]; 2110h -
 If nonzero, manual wheel mode is set - display can be initialised and
 wheel mode keyboard at subindex 2111h is active.
 (Read/Write)

WheelKeyb[2]; 2111h -
 Wheel mode keyboard. Every bit represents one key. 1 - Key is pressed.
 Subindex 0 : Length of the array - 2.
 Subindex 1 : Manual wheel keyboard bits 0 - 7.
 Subindex 2 : Manual wheel keyboard bits 8 - 15.
 (Read only)

SDO's implemented by different way than
 described in Cia DS-301 and Cia DS-401:

HeartBeat parameters (1016h and 1017h):
 Time is specified in 1.024ms interval.
 (Instead of 1ms)
 For Consumer Heartbeat time interval 1 cannot
 be specified. (Immediate error after first Heartbeat).

PDO communications parameter: (1400h, 1800h, 1801h)
 Supported transmission type (subindex 2) -
 only type 1-240 supported, for TPDO also type 252.
 Default value for transmission type is 1.
 For types 1-240 RTR (remote frame) not supported.

RPDO, TPDO mapping parameter: (1600h, 1a00h)
 Only fields with 8 bit length supported.
 Count of mapped fields (Subindex 0) fixed to 8.
 Unused fields must be mapped with dummy mapping.

TPDO2 mapping parameter: (1a01h)
 TPDO2 mapping fixed - set to analog inputs
 (Index 6401h Subindex 1 - 2).
 Length of bit fields fixed to 16.
 Count of mapped fields fixed to 2.

AnalogIMfSp[5]; 6404h -
 Up to 4 potentiometers can be connected.
 Subindex 1 - state of the first potentiometer,
 Subindex 2 - state of the second potentiometer....
 0 - rotated to the left end position
 0xff - rotated to the right end position
 Subindex 5 - fixed value 33h.
 (Read only)

SDO's implemented by different way than
 described in Cia DS-301 and Cia DS-401:

SDO Error Register (1001h):
 Manufacturer specific error (bit 80h)
 represents Input error (set when bad
 connection with the input subboard detected).

HeartBeat parameters (1016h and 1017h):
 Time is specified in 1.024ms interval.
 (Instead of 1ms)

For Consumer Heartbeat time interval 1 cannot be specified. (Immediate error after first Heartbeat).

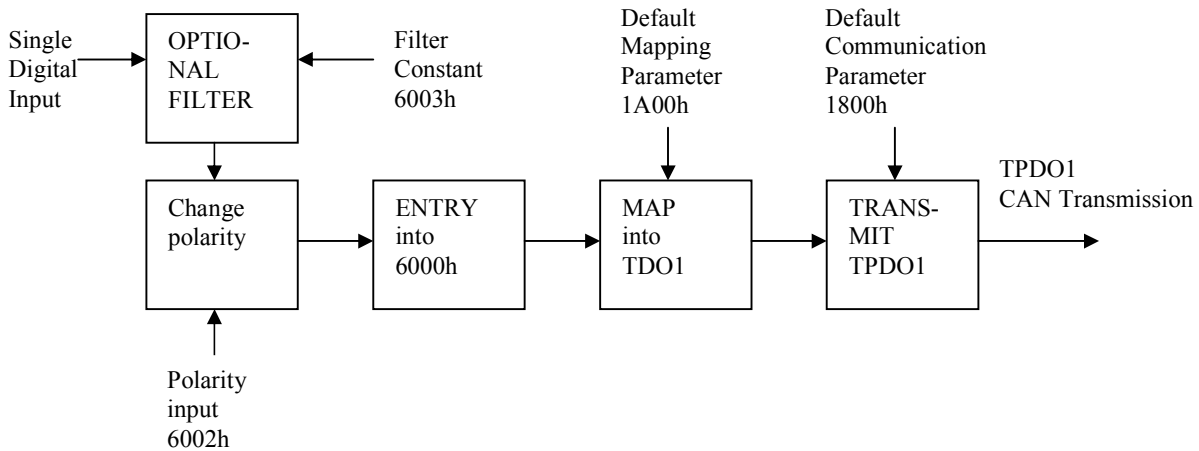
PDO communications parameter: (1400h, 1800h, 1801h)
 Supported transmission type (subindex 2) - only type 1-240 supported, for TPDO also type 252.
 Default value for transmission type is 1.
 For types 1-240 RTR (remote frame) not supported.

RPDO, TPDO mapping parameter: (1600h, 1a00h)
 Only fields with 8 bit length supported.
 Count of mapped fields (Subindex 0) fixed to 8.
 Unused fields must be mapped with dummy mapping.

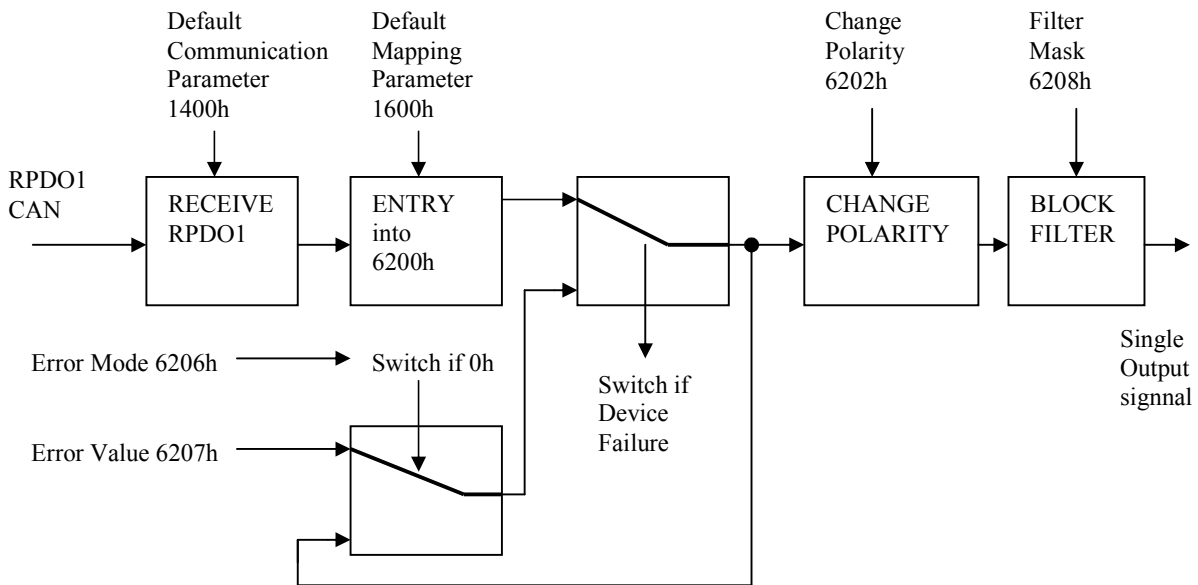
TPDO2 mapping parameter: (1a01h)
 TPDO2 mapping fixed - set to analog inputs (Index 6401h Subindex 1 - 4).
 Length of bit fields fixed to 16.
 Count of mapped fields fixed to 4.

AnalogIMfSp[8]; 6404h -
 Up to 7 potentiometers can be connected to output port OP1: common ref. input to OP1/0, rider of the first pot. to OP1/1,... rider of the seventh pot. to OP1/7.
 Subindex 1 - state of the first potentiometer,
 Subindex 2 - state of the second potentiometer....
 0 - rotated to the left end position
 0xff - rotated to the right end position
 (Read only)

Blokové schéma pro číslicové vstupy:



Blokové schéma pro číslicové výstupy:



9.7 Popis jednotky INOUT08

NAPÁJENÍ:

Karta INOUT07 je napájena z napájecího napětí vstupů a výstupů. Od systému je galvanicky oddělena na straně komunikační linky - max. 2kV. Vstupy a výstupy nejsou navzájem galvanicky odděleny.

Napájecí napětí:

Jmenovité napájecí napětí vstupů a výstupů: 24V ss.

Maximální napájecí napětí vstupů a výstupů: 36V šš.

Minimální napájecí napětí vstupů a výstupů: 16V min.

VSTUPY:

Všechny vstupy (32 na kartě) mají společný záporný konec spojený se zápornou svorkou napájecího kondensátoru. Napájecí kondensátor je spojen se zápornou svorkou napájení přes diodu, s kladnou přímo. Vstupy systému se chovají jako odpor 10kOhm $\pm 5\%$ proti záporné svorce napájecího kondensátoru desky. (Funkční schéma desky viz výkres INOUT08S.pdf)

Maximální napětí na vstupu: 36V šp.

Minimální napětí na vstupu: -36V šp.

Doba, za kterou se vstupy dostanou do systému na vstup programovatelného automatu: 1.3ms (při cyklu automatu 1ms).

Napětí pro log. 1 : > 50% napájecího napětí +3V (15V pro napájecí napětí 24V).

Napětí pro log. 0: < 33% napájecího napětí (8V pro napájecí napětí 24V)

Toto napětí se mění souhlasně se změnou napájecího napětí desky. Logické úrovně vstupů lze zjistit na panelu systému po stisknutí tlačítka WIN a navolení obrazovky diagnostika externích vstupů. Pokud je napájecí napětí

příliš nízké, karta o tom podá zprávu programovatelnému automatu. Pokud jsou příводы vedeny v prostorech se zvýšeným elektromagnetickým rušením, je třeba použít buď vodiče opatřené stíněním spojeným se záporným pólem napájecího napětí, nebo přidavné zatěžovací odpory nebo kondensátory, připojené mezi vstupní svorku a záporný pól napájecího napětí. Vstupní svorky jsou označeny IP0(0..7), IP1(0..7), IP2(0..7) a IP3(0..7) souhlasně se vstupy automatu.

VÝSTUPY:

Všechny výstupy (24 na kartě) mají společný kladný konec spojený s kladnou svorkou napájecího napětí. Mají jmenovitý výstupní proud 0.1A a jsou chráněné proti zkratu. Při zkratu dojde k vypnutí daného výstupu, a porucha je nahlášena do systému. K obnovení činnosti dojde po uvedení desky do počátečního stavu – nové spuštění systému nebo nové zapnutí.

Doba, za kterou se výstupy dostanou z programovatelného automatu na výstup: 1.3ms. (při cyklu automatu 1ms).

Doba, za kterou je schopna karta autonomně reagovat na podnět: 0.7ms. (Princip použití autonomního řízení výstupů viz výkres INOUT08F.pdf)

Po ztrátě komunikace se systémem, poruše snímání vstupů na desce nebo poklesu napájecího napětí se všechny výstupy nastaví do stavu, který je dán parametrem ErrValueOutput - Index 6207h (Při počátečním nastavení vypnuty)

Při přepólování napájecího napětí se karta nepoškodí, avšak všechny výstupy se budou chovat jako sepnuté ke kladné svorce napájecího napětí, a ochrana proti zkratu nebude funkční. Výstupy mohou být zatíženy jmenovitým proudem trvale a všechny. Všechny výstupy mohou být přetíženy až na max. 0.12A po dobu max 1 minuty. Napěťové špičky při spínání indukční zátěže jsou omezeny na hodnotu 58V+5% mezi výstupem a kladnou svorkou napájecího napětí ochrannými obvody na desce. Unikající proud při vypnutém výstupu a max. napájecím napětí: max. 0.25mA.

Pokles napětí mezi napájecím napětím a zapnutým výstupem: max. 1.5V.

Logické úrovně výstupů a případné vypnutí zkratové ochrany lze zjistit na panelu systému po stisknutí tlačítka WIN a navolení obrazovky diagnostika externích vstupů.

Výstupní svorky jsou označeny OP0(0..7), OP1(0..7) a OP2(0..7) souhlasně se vstupy automatu.

KOMUNIKACE:

Deska komunikuje s řídicím systémem po sběrnici CAN protokolem CANOPEN rychlostí 1MBaud, 800kBaud, 500kBaud, 250kBaud nebo 125kBaud. Rychlost komunikace se nastaví automaticky po spuštění řídicího systému. Než je navázána komunikace s řídicím systémem, deska indikuje blikáním identifikační číslo (deviceID) v hexadecimální soustavě. Pokud má deska základní nastavení (je systému jediná a není třeba identifikační číslo měnit, indikuje číslo 21h – blikne 2x, krátká pauza, blikne 1x, dlouhá pauza. Po úspěšném navázání komunikace s řídicím systémem začne svítit trvale.

Pokud je třeba identifikační číslo změnit, postupujeme takto:

Stiskneme pomocí vhodného nástroje (např. šroubovák) skrz otvory v horních dvou deskách tlačítko na spodní desce po dobu 3s. Bezprostředně poté (do 1s) stiskneme tlačítko krátce tolikrát, kolikrát deska INOUT08 to bude.

Příklad.:

budou-li v systému 4 desky a nastavovaná deska má být poslední, stiskneme tlačítko na 3s, krátce uvolníme a rychle stiskneme 4x po sobě. Pokud se to podařilo, bude od té doby deska před zahájením komunikace s řídicím systémem indikovat blikáním číslo 24h – blikne 2x, krátká pauza, blikne 4x, dlouhá pauza.

Poslední zařízení připojené ke sběrnici CAN by mělo zakončovat komunikační linku odporem. Pokud je tím posledním zařízením deska INOUT08, lze připojit k lince zakončovací odpor připravený na desce: provede se to tak, že se propojí vodičem svorky T a L komunikačního konektoru CAN. (Příklad připojení viz výkres INOUT08E.pdf)