

26

26. PERIFERIE PŘIPOJENÉ NA ETHERCAT

26.1 EtherCAT – základy

Systém umožňuje připojit různé periferie (pohony, vstupy a výstupy...) pomocí sběrnice EtherCAT. Na připojení se využívá samostatný vhodný port ETHERNET. Všechny připojené periferie pracují v reálném čase s periodou obsluhy například 200 μ s.

Sběrnici EtherCAT obsluhuje program „KPA EtherCAT Master“. Jedná se o úlohu reálného času nad REAL-TIME jádrem „RTX“. Mezi jeho základní vlastnosti patří: „Distribuovaný Clock“ (DC), vstupy a výstupy přes „Process Image“ (PI), čtení a zápis parametrů „MailBox“ typu „CoE“ nebo „SoE“.

Kompletní obsluha EtherCAT periferií je zpřístupněna v PLC programu.

Konfigurace, prvotní inicializace, konfigurace při změně stavu a definice cyklické výměny dat (Process image) je definovaná v souboru typu XML. Tento soubor je generovaný externím programem „EtherCAT Studio“ například od firmy „Koenig“. Tak konfigurace EtherCAT periferií ovlivní jen PLC program a není závislá na CNC systému. Konfigurační XML soubor má název „MASTER.XML“ a je uložený v adresáři „CNC Machine Files\Config“. Do tohoto adresáře se musí překopírovat prostřednictvím SETUPu PLC programu (soubor „PLC.NSI“).

Pro aktivaci sběrnice EtherCAT je nutno nastavit registr Windows. Nastavení se provede automaticky už při instalaci WinCNC (musí se zvolit typ EtherCAT). Další nastavení slouží jen pro speciální požadavky a lze je provést v SETUPu PLC programu v souboru PLC.NSI:

```
WriteRegDWORD HKLM "Software\MEFI\WinCNC\Machine\Common"
    .. "RtxEtherCAT" 1
    .. "EtherCATSynchMode" 2
    .. "EtherCATDebug" 0 (7)
    .. "EtherCATPhaseAdjust" 1
    .. "EtherCATDelayOP" 500
    .. "EtherCATMasterConfigFile" Master.xml
    .. "EtherCATMasterCycleTime" 1000
    .. "FastestPeriod" 1000
    .. "HALTimerPeriod" 1000
    .. "EtherCATStatistic" 0
    .. "EtherCATWatchMaster" 1
    .. "EtherCATWatchSlaves" 0
    .. "EtherCATOPViaPLC" 0
```

Význam všech možností pro nastavení EtherCATu v registru Windows bude vysvětlen dále.

CNC systém musí mít zakoupené Run-Time licence pro „RTX“ a pro „KPA EtherCAT Master – RTX“.

26.2 Cyklická výměna dat

26.2.1 Obraz „Process Image“ v PLC programu

V PLC programu je „Process Image“ definován jako proměnné, které jsou definovány jako logické vstupy a výstupy pomocí instrukcí **DEF_IN** a **DEF_OUT** (viz návod „Logické vstupy a výstupy modulů systému“).

Systém automaticky definuje logické vstupy a výstupy, které souvisí s EtherCATem na základě konfiguračního XML souboru, který je vytvořen pomocí EtherCAT studia.

Definice „Process Image“ formou logických vstupů a výstupů umožňuje jejich propojování jen na základě konfigurace a bez nutnosti změny PLC programu. Při definici logických prvků pro „virtuální spoje“ v PLC programu se proměnné automaticky deklarují – vymezí se jim požadovaný paměťový prostor.

Datové prvky, které definují logický spoj, mohou být definovány jako pole s ohledem na typ proměnné. V tomto případě logický spoj k danému prvku určuje jeho jméno a koncový index.

Pro každý logický spoj možno definovat konverzi, která se uplatní při propojení prvků.

Všechny logické vstupy a výstupy mají automaticky nastaveno „**přímé sdílení**“ a tak se značně zjednoduší přístup uživatelských obrazovek a dialogů k jejím hodnotám. Na logické vstupy a výstupy není potřeba používat instrukce **SHARE_VAR** a **SHARE_BIT**.

Příklad pro definici prvků „Process Image“:

```
;Drive status word, servo 0
DEF_IN      wEcatDriveStatus0, 'PIInDriveStatus0', TYPE_UN_16, -, FAST

;Position feedback, servo 0
DEF_IN      dwEcatDrivePos0, 'PIInDrivePos0', TYPE_INT_32, -, FAST

;Master control word, servo 0
DEF_OUT     wEcatDriveControl0, 'PIOutDriveControl0', TYPE_UN_16, -, FAST

;Velocity demand value, servo 0
DEF_OUT     dwEcatDriveVeloReq0, 'PIOutDriveVeloReq0', TYPE_INT_32, -, FAST
```

Pro práci s „ControlWordem“ a „StatusWordem“ je možno deklarovat formální definice bitů a využít složitější adresaci bitu.

Příklad:

```
;Formalni definice bitu pro Controlword a Statusword pro CoE a SoE

;Statusword CoE
CoEStat0: DFM StatCoE_RDY, StatCoE_ON, StatCoE_ENABLE, StatCoE_FAULT,
          StatCoE_VOLT, StatCoE_QSTOP, StatCoE_DISABLE, StatCoE_WARNING
;StatCoE_RDY          Ready to switch on
;StatCoE_ON           Switched on
;StatCoE_ENABLE       Operation enable
;StatCoE_FAULT        Fault
;StatCoE_VOLT         Disable voltage
;StatCoE_QSTOP        Quick stop
;StatCoE_DISABLE      Switch on disabled
;StatCoE_WARNING      Warning

;Statusword SoE
SoEStat0: DFM , , , StatSoE_COMMAND, , , ,
SoEStat1: DFM StatSoE_M0, StatSoE_M1, StatSoE_M2, , , , StatSoE_READY,
          StatSoE_POWER
;StatSoE_COMMAND      Status command value processing
;StatSoE_M0           Operation mode 0
;StatSoE_M1           Operation mode 1
;StatSoE_M2           Operation mode 2
;StatSoE_READY        ready to operate 0    drive ready
;StatSoE_POWER        ready to operate 1    main power applied

;Controlword CoE
CoECnt0: DFM CntCoE_ON, CntCoE_VOLT, CntCoE_QSTOP, CntCoE_ENABLE,
          CntCoE_M0, CntCoE_M1, CntCoE_M2, CntCoE_FAULT
;CntCoE_ON            Switch ON (1)
;CntCoE_VOLT          Disable Voltage (0)
;CntCoE_QSTOP         Quick Stop (0)
;CntCoE_ENABLE        Enable Operation
;CntCoE_M0            Operation mode specific
;CntCoE_M1            Operation mode specific
;CntCoE_M2            Operation mode specific
;CntCoE_FAULT         Reset Fault

;Controlword SoE
SoECnt0: DFM , , , , , , , ,
SoECnt1: DFM CntSoE_OP0, CntSoE_OP1, CntSoE_SYNCH, CntSoE_OP2,
          , CntSoE_HALT, CntSoE_ENABLE, CntSoE_ON
;CntSoE_OP0           Operation Mode 0
;CntSoE_OP1           Operation Mode 1
;CntSoE_SYNCH         Control unit synchronisation bit
;CntSoE_OP2           Operation Mode 2
;CntSoE_HALT          Halt/Restart drive
;CntSoE_ENABLE        Enable drive
;CntSoE_ON            Drive ON/OFF
```

Příklad mechanismu pro ENABLE pohonu:

Příklad:

```
; Mechanizmus provádí "ENABLE" pohonu pro servo 0 (typ SoE).
MECH_BEGIN M_ECAT_DRIVE0_ENABLE
    ; Jako žádanou polohu nastavit aktuální polohu hlášenou pohonem
    LOD      dwEcatDrivePos0
    STO      dwEcatDrivePosReq0
    ;V ControlWordu daného pohonu nastavit příslušné bity
    FL      1,wEcatDriveControl0+1.CntSoE_ON
    FL      1,wEcatDriveControl0+1.CntSoE_ENABLE
    FL      1,wEcatDriveControl0+1.CntSoE_HALT
    FL      1,wEcatDriveControl0+1.CntSoE_SYNCH
    EX
    ;Počkat, až pohon oznámí splnění požadavku
    LDR     wEcatDriveStatus0.StatSoE_COMMAND
    LA      wEcatDriveStatus0+1.StatSoE_READY
    LA      wEcatDriveStatus0+1.StatSoE_POWER
    TEX0    -, 350, LBL_ECAT_DRIVE0_ENABLE_ERR
    ...
```

26.2.2 Propojení „Process Image“ s obrazem v PLC

Propojování jednotlivých prvků se provede jen formou konfigurace, bez nutnosti změny PLC programu.

Logické virtuální spoje jsou definovány dvěma textovými identifikátory. Identifikátor logického výstupu a identifikátor logického vstupu. Textové identifikátory používají „tečkovou konvenci“ podle zásad popsaných v návodu „Logické vstupy a výstupy modulů systému“.

Identifikátor pro připojení k logickým vstupům a výstupům definovaným v „Process image“ pro EtherCAT komunikaci: **ECAT.<jmeno>[xx]**. Typ prvku je definován v „Process image“ (vytvořeno v EtherCAT studiu a zaznamenáno v XML souboru). Možnost periody obsluhy: FAST.

Konfigurace pro logické spoje jsou v souboru typu „ChannelConfig“ v elementu „Connections“.

Příklad:

```
<Connections>

  <Connection Source="PLC.Output.PIOutDriveControl0"
    Destination="ECAT.Drive 0.Outputs.Control word"
    Connected="1"></Connection>

  <Connection Source="PLC.Output.PIOutDriveVeloReq0"
    Destination="ECAT.Drive 0.Outputs.Target velocity"
    Connected="1"></Connection>

  <Connection Source="ECAT.Drive 0.Inputs.Status word"
    Destination="PLC.Input.PIInDriveStatus0"
    Connected="1"></Connection>

  <Connection Source="ECAT.Drive 0.Inputs.Position actual value"
    Destination="PLC.Input.PIInDrivePos0"
    Connected="1"></Connection>

</Connections>
```

26.2.3 Start cyklické výměny dat

Cyklická výměna dat se odstartuje automaticky po přechodu Mustru do stavu OPERATIONAL. Jen v případě, že se požaduje zpožděný start cyklické výměny, který řídí PLC program a je nastaven klíč EtherCATOPViaPLC = 1, tak cyklickou výměnu dat může odstartovat PLC program prostřednictvím bitu:

flEcatStartCyclicOperation.

Nastavením bitu „flEcatStartCyclicOperation“ na hodnotu log.1 se spustí cyklická výměna dat mezi EtherCAT Mastrem a všemi EtherCAT Slave. Současně se bude měnit stav Mustru:

INIT -> PRE-OPERATIONAL -> SAFE-OPERATIONAL -> OPERATIONAL

Každá změna stavu Mustru je doprovázena vysláním příkazů podle definic v souboru „MASTER.XML“. Tím dojde k inicializaci a k nastavení všech potřebných parametrů v EtherCAT periferiích.

Po úspěšném dosažení stavu „OPERATIONAL“ systém dá zprávu PLC programu nastavením bitu „flEcatOperational“. PLC program potom může pokračovat v inicializaci periferií, například přechod do stavu „ENABLE“

flEcatStartCyclicOperation	Žádost z PLC o start cyklické výměny dat
flEcatOperational	Zpráva pro PLC o dosažení stavu „OPERATIONAL“

26.3 Stavy EtherCAT Mastru

EtherCAT Master se může nacházet ve stavech:

Stav	Hodnota	Symbol pro PLC
BOOTSTRAP	3	ECAT_BOOTSTRAP
INIT	1	ECAT_INIT
PRE-OPERATIONAL	2	ECAT_PREOPERATIONAL
SAFE-OPERATIONAL	4	ECAT_SAFEOPERATIONAL
OPERATIONAL	8	ECAT_OPERATIONAL

Po správné inicializaci se EtherCAT MASTER nachází ve stavu OPERATIONAL. (je nahozen bit `flEcatOperational`). PLC program má možnost žádat o změnu stavu MASTRA. Tato změna musí být uvážena, protože se týká všech EtherCAT periférií. Například pohony musí být ve stavu DISABLE. Při opětovném přechodu do stavu OPERATIONAL, je potřeba zrušit referenci a znovu načíst polohu motoru.

Žádosti z PLC o změnu stavu EtherCAT MASTRu:

flEcatBootStrapReq	Žádost z PLC o přechod MASTRu do stavu „BOOTSTRAP“
flEcatInitReq	Žádost z PLC o přechod MASTRu do stavu „INIT“
flEcatPreOperationalReq	Žádost z PLC o přechod MASTRu do stavu „PRE-OPERATIONAL“
flEcatSafeOperationalReq	Žádost z PLC o přechod MASTRu do stavu „SAFE-OPERATIONAL“
flEcatOperationalReq	Žádost z PLC o přechod MASTRu do stavu „OPERATIONAL“

Jako potvrzení změny stavu MASTRu se příslušný bit žádosti sám vynuluje.

Příklad:

;Mechanismus pro přechod do stavu PRE-OPERATIONAL:

```
MECH_BEGIN M_ECAT_PREOPERATIONAL
    FL    1, flEcatPreOperationalReq
    EX
    LDR   flEcatPreOperationalReq
    TEX1  -, 500, LBL_ECANT_PREOP_ERR
    ...
MECH_END   M_ECANT_PREOPERATIONAL
```

26.4 Mailbox typu CoE a SoE, příkazy pro Mastra

Modul „Mailbox“, který je součástí EtherCAT Mastru slouží pro čtení a nastavování parametrů jednotlivých periférií. Jedná se vždy o jednorázové – neperiodické přístupy k parametrům jednotlivých „Slave“.

26.4.1 Čtení a zápis parametru typu CoE

Pro komunikaci s EtherCAT periférií se využívá komunikační profil „CANopen“ (CoE = „CANopen over EtherCAT“) podle normy: „CiA Standard 402 (DSP402, Device profile for drives and motion control)“. Z toho se využívají hlavně komunikační objekty „SDO (Service Data Object)“ a pro adresaci se využívá „INDEX“ a „SUBINDEX“ objektu podle normy CANopen.

Pro SDO komunikaci byly zavedeny instrukce pro čtení a zápis. Jedná se o instrukce, které mohou být použity jen v mechanismech, protože se vždy čeká na výsledek komunikace.

instrukce	ECAT_COE_READ ECAT_COE_WRITE
------------------	---

funkce	ECAT_COE_READ	Čtení registru z EtherCAT periferie typu CoE
	ECAT_COE_WRITE	Zápis do registru EtherCAT periferie typu CoE

syntax	ECAT_COE_READ	Master, Slave, Index, Subindex, Val, Len
	ECAT_COE_WRITE	Master, Slave, Index, Subindex, Val, Len

1. parametr	„Master“	ID Mastra
2. parametr	„Slave“	ID Slave
3. parametr	„Index“	Index pro SDO (WORD)
4. parametr	„Subindex“	Subindex pro SDO (BYTE)
5. parametr	„Val“	pointer nebo název proměnné pro data
6. parametr	„Len“	název proměnné pro uložení délky dat (DWORD)

parametr	název	význam	typ
1.	Master	ID EtherCAT Mastra (0,1,..). Pokud je použit jen jeden, tak hodnota 0.	Byte
2.	Slave	ID EtherCAT Slave (0,1,..). Adresa „Slave“ je v pořadí jak jsou připojeny na ETHERNET.	Word
3.	Index	Index pro adresaci registru v „Slave“ typu SDO	Word
4.	Subindex	SubIndex SDO registru v „Slave“	Byte
5.	Val	Název datové proměnné nebo pole pro významová data. Parametr se předává instrukci odkazem (ne hodnotou).	Data
6.	Len	Název datové proměnné pro uložení délky dat typu DWORD. Parametr se předává instrukci odkazem (ne hodnotou).	Data

Návratové hodnoty instrukcí:

Instrukce se musí používat v mechanismech a jsou typu „EX“.

Vrácené datové hodnoty se zapisují přímo do datové proměnné „Val“ a „Len“ .

Datová proměnná „Val“ musí mít rezervovaný dostatečný prostor pro zápis požadovaných hodnot.

Datová proměnná „Len“ pro délku slouží jak pro vstup požadované délky dat, tak jako návratovou hodnotu skutečné délky načtených dat.

Všechny instrukce se mohou volat průchodově a mají návratové hodnoty v registrech:

RLO=0, DR=0 stav čekání na dokončení operace
RLO=1, DR=0 operace dokončena bez chyb
RLO=1, DR<>0 operace dokončena, ale při výkonu vznikla chyba

Příklad:

Čtení statusu CoE ze Slave 0:

```
bEcatData:      DS    4           ;data
dwEcatDataLen:  DS    4           ;pro uložení délky

MECH_BEGIN M_COE_READ
  LOD           CNST.2
  STO          dwEcatDataLen
  ECAT_COE_READ 0,0, 6041h, 0, bEcatData, dwEcatDataLen
  EX0
  EQ           CNST.0
  JLO          LBL_COE_ERR       ;Error
  ...
MECH_END M_COE_READ
```


26.4.2 Čtení a zápis parametru typu SoE

Pro komunikaci s EtherCAT periferií se využívá komunikační profil „SERCOS“ (SoE = „Servo Profile over EtherCAT“) podle normy: „IEC 61491“. Využívají se komunikační objekty „IDN“ typu „S“ a „P“.

Pro SERCOS komunikaci byly zavedeny instrukce pro čtení a zápis. Jedná se o instrukce, které mohou být použity jen v mechanismech, protože se vždy čeká na výsledek komunikace.

instrukce	ECAT_SOE_READ ECAT_SOE_WRITE
------------------	---

funkce	ECAT_SOE_READ ECAT_SOE_WRITE	Čtení registru z EtherCAT periferie typu SoE Zápis do registru EtherCAT periferie typu SoE
syntax	ECAT_SOE_READ ECAT_SOE_WRITE	Master, Slave, DrNo, ElemFl, IDN, Val, Len Master, Slave, DrNo, ElemFl, IDN, Val, Len
1. parametr	„ Master “	ID Mastra
2. parametr	„ Slave “	ID Slave
3. parametr	„ DrNo “	DriveNo (BYTE)
4. parametr	„ ElemFl “	název proměnné pro uložení ElementFlags (BYTE)
5. parametr	„ IDN “	IDN (WORD)
6. parametr	„ Val “	pointer nebo název proměnné pro data
7. parametr	„ Len “	název proměnné pro uložení délky dat (DWORD)

parametr	název	význam	typ
1.	Master	ID EtherCAT Mastra (0,1,..). Pokud je použit jen jeden, tak hodnota 0.	Byte
2.	Slave	ID EtherCAT Slave (0,1,..). Adresa „Slave“ je v pořadí jak jsou připojeny Slave na ETHERNET.	Word
3.	DrNo	DriveNo (číslo zařízení)	Byte
4.	ElemFl	Název datové proměnné pro uložení ElementFlags (příznaky elementu). Parametr se předává instrukci odkazem (ne hodnotou).	Byte
5.	IDN	IDN registru	Word
6.	Val	Název datové proměnné nebo pole pro významová data. Parametr se předává instrukci odkazem (ne hodnotou).	Data
7.	Len	Název datové proměnné pro uložení délky dat typu DWORD. Parametr se předává instrukci odkazem (ne hodnotou).	Data

Návratové hodnoty instrukcí:

Instrukce se musí používat v mechanismech a jsou typu „EX“.

Vrácené datové hodnoty se zapisují přímo do datové proměnné „**Val**“ a „**Len**“ .

Datová proměnná „**ElemFl**“, slouží jak pro vstup, tak pro výstup ElementFlags příslušného registru.

Datová proměnná „**Val**“ musí mít rezervovaný dostatečný prostor pro zápis požadovaných hodnot.

Datová proměnná „**Len**“ pro délku slouží jak pro vstup požadované délky dat, tak jako návratová hodnota skutečné délky načtených dat.

Všechny instrukce se mohou volat průchodově a mají návratové hodnoty v registrech:

RLO=0, DR=0 stav čekání na dokončení operace
RLO=1, DR=0 operace dokončena bez chyb
RLO=1, DR<>0 operace dokončena, ale při výkonu vznikla chyba

Příklad:

Nulování chyb SoE v Slave 0, na IDN S-0-0099

bEcatData: DS 4 ;data
dwEcatDataLen: DS 4 ;pro uložení délky
bEcatElem: DS 1 ;pro ElementFlags

```
MECH_BEGIN M_SOE_CLEAR_ERR
  LOD      cnst.40h
  STO      bEcatElem      ;ElementFlags = „VAL“
  LOD      CNST.3
  STO      DWRD.bEcatData ;data = 3
  LOD      CNST.2
  STO      dwEcatDataLen ;délka
  ECAT_SOE_WRITE 0,0,0,bEcatElem,99,bEcatData,dwEcatDataLen
  EX0
  JL1      LBL_SOE_ERR    ;Error
MECH_END M_SOE_CLEAR_ERR
```

26.4.3 Zasilání příkazů pro EtherCAT Mastra

Pro zasilání příkazů pro EtherCAT Mastra z PLC programu slouží instrukce `ECAT_COMMAND`. Jedná se o instrukci, která může být použita jen v mechanismech, protože se vždy čeká na výsledek komunikace.

instrukce	<code>ECAT_COMMAND</code>
-----------	---------------------------

funkce	<code>ECAT_COMMAND</code>	Příkaz pro EtherCAT Mastra
syntax	<code>ECAT_COMMAND</code> <code>ECAT_COMMAND</code>	<code>Master, Slave, Cmd</code> <code>Master, Slave, Cmd [, Val, Len]</code>
1. parametr	<code>„Master“</code>	ID Mastra
2. parametr	<code>„Slave“</code>	ID Slave
3. parametr	<code>„Cmd“</code>	Příkaz podle výčtu <code>ECATCOMMAND</code> (WORD)
4. parametr	<code>„Val“</code>	pointer nebo název proměnné pro data
5. parametr	<code>„Len“</code>	název proměnné pro uložení délky dat (DWORD)

parametr	název	význam	typ
1.	Master	ID EtherCAT Mastra (0,1,..). Pokud je použit jen jeden, tak hodnota 0.	Byte
2.	Slave	ID EtherCAT Slave (0,1,..). Adresa „Slave“ je v pořadí jak jsou připojeny na ETHERNET.	Word
3.	Cmd	Požadovaný příkaz podle výčtu <code>ECATCOMMAND</code>	Word
4.	Val	Zadáva se jen u příkazů, které jej vyžadují. Název datové proměnné nebo pole pro významová data. Parametr se předává instrukci odkazem (ne hodnotou).	Data
5.	Len	Zadáva se jen u příkazů, které jej vyžadují Název datové proměnné pro uložení délky dat typu DWORD. Parametr se předává instrukci odkazem (ne hodnotou).	Data

Návratové hodnoty instrukcí:

Instrukce se musí používat v mechanismech a jsou typu „EX“.

Vrácené datové hodnoty se zapisují přímo do datové proměnné „**Val**“ a „**Len**“ .

Datová proměnná „**Val**“ musí mít rezervovaný dostatečný prostor pro zápis požadovaných hodnot.

Datová proměnná „**Len**“ pro délku slouží jak pro vstup požadované délky dat, tak jako návratovou hodnotu skutečné délky načtených dat.

Všechny instrukce se mohou volat průchodově a mají návratové hodnoty v registrech:

<code>RLO=0,</code>	<code>DR=0</code> stav čekání na dokončení operace
<code>RLO=1,</code>	<code>DR=0</code> operace dokončena bez chyb
<code>RLO=1,</code>	<code>DR<>0</code> operace dokončena, ale při výkonu vznikla chyba

Přehled příkazů ECATCOMMAND

Symbol příkazu	Hodnota	Popis
ECATCMD_MASTER_BOOTSTRAP	3	Žádost o přechod Mastra do stavu BOOTSTRAP
ECATCMD_MASTER_INIT	1	Žádost o přechod Mastra do stavu INIT
ECATCMD_MASTER_PREOPERATIONAL	2	Žádost o přechod Mastra do stavu PRE-OPERATIONAL
ECATCMD_MASTER_SAFEOPERATIONAL	4	Žádost o přechod Mastra do stavu SAFE-OPERATIONAL
ECATCMD_MASTER_OPERATIONAL	8	Žádost o přechod Mastra do stavu OPERATIONAL
ECATCMD_SLAVE_BOOTSTRAP	13	Žádost o přechod jednoho Slave do stavu BOOTSTRAP
ECATCMD_SLAVE_INIT	11	Žádost o přechod jednoho Slave do stavu INIT
ECATCMD_SLAVE_PREOPERATIONAL	12	Žádost o přechod jednoho Slave do stavu PRE-OPERATIONAL
ECATCMD_SLAVE_SAFEOPERATIONAL	14	Žádost o přechod jednoho Slave do stavu SAFE-OPERATIONAL
ECATCMD_SLAVE_OPERATIONAL	18	Žádost o přechod jednoho Slave do stavu OPERATIONAL
ECATCMD_MASTER_STATE	20	Žádost o zjištění aktuálního stavu Mastru. Příkaz musí mít parametry pro návratovou hodnotu (Val, Len).
ECATCMD_SLAVE_STATE	21	Žádost o zjištění aktuálního stavu jednoho Slave. Příkaz musí mít parametry pro návratovou hodnotu (Val, Len).

Příklad:

Přechod 2.EtherCAT Slave (ID=1) do stavu SAFE-OPERATIONAL a zjištění jeho aktuálního stavu

```
bEcatData:      DS    8           ;data
dwEcatDataLen:  DS    4           ;pro uložení délky
dwCmdResult:    DS    4           ;pro test chyby
```

```
MECH_BEGIN M_ECAT_SLAVE
    ECAT_COMMAND      0,1,ECATCMD_SLAVE_SAFEOPERATIONAL
    EX0
    STO               dwCmdResult
    ECAT_COMMAND      0,1,ECATCMD_SLAVE_STATE,bEcatData,dwEcatDataLen
    EX0
    STO               dwCmdResult
    LOD               BYTE.bEcatData
    EQ                ECAT_SAFEOPERATIONAL
    ...
MECH_END M_ECAT_SLAVE
```