

|  |  |
| --- | --- |
| VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ BRNO UNIVERSITY OF TECHNOLOGY   FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION   ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY DEPARTMENT OF CONTROL AND INSTRUMENTATION       SLAVE MODUL VYUŽÍVAJÍCÍ KOMUNIKAČNÍ PROTOKOL ETHERCAT SLAVE ETHERCAT MODULE | |
| BAKALÁŘSKÁ PRÁCE BACHELOR'S THESIS | |
| AUTOR PRÁCE AUTHOR | Jakub Hadámek |
| VEDOUCÍ PRÁCE SUPERVISOR | Ing. Soběslav Valach |
| BRNO 2018 |  |

zadání Bakalářské/Diplomové práce – vloží se na tuto pozici

Zadání bakalářské/diplomové práce se generuje v IS. Je možné vložit i naskenované zadání, které student převzal od vedoucího práce. Na zadání musí být nové logo VUT.

Poznámka:

**Červeným písmem je uvedeno, co má být napsáno resp. Aktualizováno!!**

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém jazyce.

Klíčová slova

Sem budou zapsána jednotlivá klíčová slova v českém jazyce, oddělená čárkami.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

Bibliografická citace:

HADÁMEK, J. *Slave modul využívající komunikační protokol EtherCAT*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 85s. Vedoucí bakalářské práce byl Ing. Soběslav Valach.

Prohlášení

„Prohlašuji, že svou diplomovou (*bakalářskou*) práci na téma Slave modul využívající protokol EtherCAT jsem vypracoval samostatně pod vedením vedoucího diplomové (*bakalářské*) práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové (*bakalářské*) práce dále prohlašuji, že v souvislosti s vytvořením této diplomové (*bakalářské*) práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **10. května 2018** …………………………

podpis autora

Poděkování (nepovinné)

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc

(externí zadavatel, konzultant, apod.).

Děkuji vedoucímu diplomové (*bakalářské*) práce Prof. Ing. Jiřímu Novotnému, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **10. května 2018**  …………………………

podpis autora

OBSAH

[1 Úvod 8](#_Toc507713091)

[2 Ethercat 3](#_Toc507713092)

[2.1 Popis 3](#_Toc507713093)

[2.1.1 Field memory management unit 3](#_Toc507713094)

[2.1.2 SyncManager 3](#_Toc507713095)

[2.1.3 Odolnost 5](#_Toc507713096)

[2.2 Protokol 6](#_Toc507713097)

[2.3 Adresování 8](#_Toc507713098)

[2.3.1 Inkrementální 8](#_Toc507713099)

[2.3.2 Pevné 8](#_Toc507713100)

[2.3.3 Logické 9](#_Toc507713101)

[2.4 Synchronizace 9](#_Toc507713102)

[3 hardware 10](#_Toc507713103)

[4 master 10](#_Toc507713104)

[5 slave 10](#_Toc507713105)

[6 Závěr 10](#_Toc507713106)

# Úvod

V dnešním průmyslovém světě je kladen velký důraz na rychlost komunikace a co nejlepší synchronizaci aplikací. Tyto požadavky splňuje právě sběrnice EtherCAT, vyvinutá pro potřeby automatizace. Mnoho výrobců již tuto technologii využívá ve svých aplikacích a zařízeních.

Výhoda EtherCATu je mimo jiné i to, že je bezplatná a každý výrobce ji může využít pro svůj vývoj či aplikaci. Tato práce se bude zabývat implementací EtherCAT protokolu na vývojovou desku od firmy Infineon. K tomuto účelu je využito open source řešení Simple Open EtherCAT Slave (SOES) od Open EtherCAT Society.

Za účelem testování je potřeba v práci použít take EtherCAT mastera. Zvoleným řešením je Simple Open EtherCAT Master, který je rovněž open source od Open EtherCAT Society.

# Ethercat

EtherCAT (Ethernet for Control Automation) je jedním z mnoha typů průmylového Ethernetu. Byl vyvinut firmou Beckhoff Automation a poprvé představen v roce 2003. Později byl v roce 2007 uznán jako mezinárodní standard. Za jeho další vývoj je zodpovědná skupina ETG (EtherCAT techology group).

## Popis

EtherCAT byl vyvinut přednostně pro použití v automatizaci, z čehož vyplývají také jeho primární vlastnosti jako je velmi krátký komunikační cyklus, lepší využití Ethernetového rámce, absence přepínačů a nízká cena. Využívá stejnou fyzickou vrstvu jako Ethernet, avšak oproti němu funguje real-time, neboť veškeré zpoždění sběrnice je díky zpracování rámců za „letu“ („data on the fly“) dáno pouze zpožděním hardwaru.

EtherCAT funguje na principu master-slave, přičemž master je jediný, kdo v síti vysílá nové Ethernetové rámce. Ty se po průchodu posledním slavem vracejí zpět a uzavírají tak logický kruh.

K implementaci mastera není potřeba žádný speciální hardware, stačí Ethernet kontrolér nebo obyčejná síťová karta. Masterem se tak může stát jakékoliv zařízení s Ethernet portem. Celá jeho funkce spočívá v softwaru, který může být buď placený od firmy Beckhoff Automation nebo open source určený pro operační systémy založené na linuxu. Mezi tyto softwary patří EtherLab a SOEM (Simple Open EtherCat Master). Slave naopak vyžaduje speciální hardware, u kterého jsou kladeny velké nároky na rychlost průchodu signálu.

### Field memory management unit

FMMU se používají pro mapování procesních dat z logického obrazu v masteru do fyzické paměti ve slave zařízení. Procesní data jsou v masteru řazena podle úkolů. Master pomocí FMMU nastavuje slave zařízení, která se smí mapovat ve stejném datagramu k automatickému seskupení procesních dat.

### SyncManager

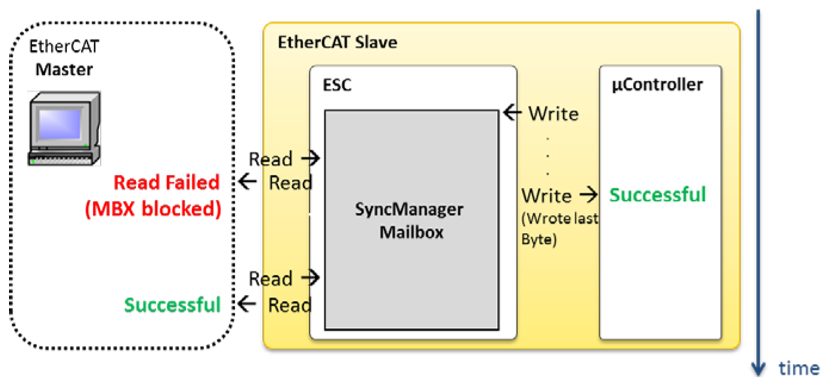
SyncManager je mechanismus chránící data v DPRAM před současným přístupem mastera (prostřednictvím EtherCAT sítě) a MCU (prostřednictvím PDI). Pokud slave používá FMMU, je SyncManager umístěn mezi FMMU a DPRAM. Může pracovat ve dvou módech:

**Mailbox mód**

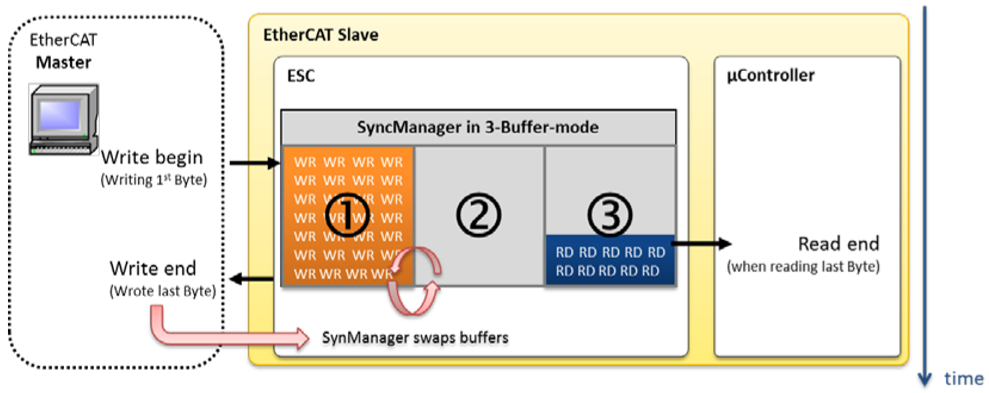
Pro výměnu dat mezi ESC a MCU se vzužívá handshake. Ethercat master a MCU může ke schránce přistupovat pouze tehdy, pokud druhý z nich dokončil svůj přístup. Pokud do ní jedna strana zapsala data, je uzamčena dokud ji druhá strana nevyprázdní. Tento mód se používá pro necyklické odesílání dat, při kterém o žádná nechceme přijít.

**Buffer mód**

Používá se pro cyklickou výměnu. V tomto režimu je potřeba třikrát větší paměťový prostor, než je velikost procesních dat, neboť jsou použity tři buffery. Master i MCU smí k datům přistupovat kdykoliv bez omezení. V případě, že jsou nová data k dispozici dříve, než se stará stihla odeslat nebo přečíst, je buffer aktualizován a stará jsou zahozena.



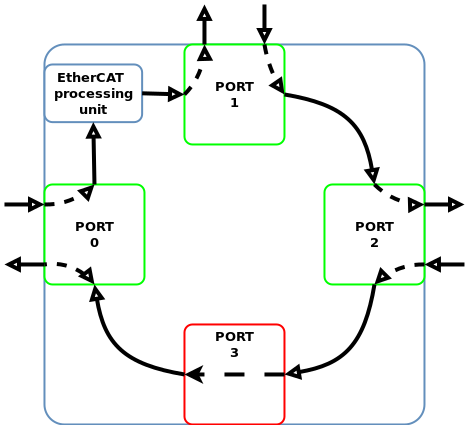
Obrázek 1 SyncManager v Mailbox módu



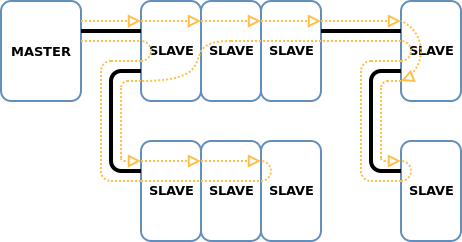
Obrázek 2 SyncManager v 3 buffer módu

Topologie

EtherCAT může být fyzicky zapojen do různých topologiích jako například strom, hvězda, kruh, bus nebo line. Kterákoliv z nich ve výsledku uzavírá logický kruh, přičemž je možno připojit až 65 535 zařízení. Do vzdálenosti 100 m se jako fyzická vrstva používá fast Ethernet (100BASE‑TX), na větší vzdálenosti pak optické vlákno (100BASE-FX). EtherCAT je vhodný pro centralizované i decentralizované systémy, umožňuje komunikaci master-master, master-slave i slave-slave. Jeho použití je jednodušší než použití Ethernetu, protože není zapotřebí přepínačů, MAC, ani IP adres.



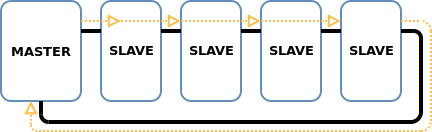
Obrázek 3 Průchod packetu přes ESC



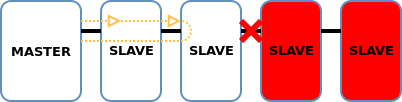
Obrázek 4 Topologie EtherCAT sítě

### Odolnost

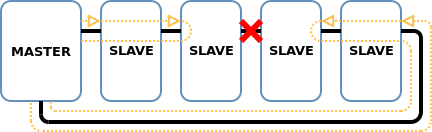
Při přerušení řetězce z důvodu porušení média nebo selhání slave zařízení jsou další zařízení odstřižena od sítě. K neutralizaci tohoto nedostatku se používá zapojení, v němž se poslední slave v síti propojí s masterem, čímž se uzavře smyčka. V případě výpadku tak dojde ke ztrátě pouze jednoho komunikačního cyklu, neboť master detekuje chybu a komunikace bude dále pokračovat. K použití tohoto je potřeba pouze, aby měl master dva Ethernetové porty.



Obrázek 5 EtherCAT síť při přerušení média



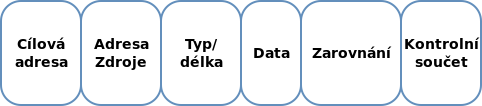
Obrázek 6 EtherCAT síť v odolném zapojení



Obrázek 7 EtherCAT v odolném zapojení při poruše

## Protokol

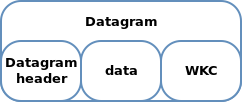
Ethercat používá standardní Ethernetový rámec, který je identifikován hodnotou 0x88A4 na místě type. Samotný rámec EtherCATu je vložen do místa pro přenášená data a skládá se z hlavičky následované jednotlivými datagramy, nazývanými také PDO. Každý datagram se zkládá z hlavičky, určující jaká operace se má vykonat (čtení/zápis), dat a pracovního čítače (WKC = WorKing Counter). Ten je inkrementován každým zařízením, které s danými daty pracovalo. Poté master vyhodnotí, zda se rovná předem vypočtané hodnotě, jíž by se měl WKC rovnat. Pokud se nerovná, vyhodnotí to jako chybu přenosu.



Obrázek 8 Ethernet rámec podle normy IEEE 802.3



Obrázek 9 EtherCAT rámec zapouzdřený v Ethernet rámci



Obrázek 10 Jeden EtherCAT datagram

**CAN over EtherCAT**

S COE EtherCAT nabízí stejný komunikační mechanismus, jako v CANopen standardu EN50325‑4.

**Servo over EtherCAT**

Umožňuje řízení servomotorů, využívající speciální protokoly k tomuto určené.

**Ethernet over EtherCAT**

Ethernetové rámce mohou být přepravena přes EtherCAT segment, neboť mají společné fyzické médium. Ethernet zařízení může být připojeno k EtherCAT síti pomocí tzv. switchportu. Rámce jsou tak tunelovány skrz EtherCAT síť, což ji dělá pro Ethernetová zařízení kompletně transparentní.

**File access over EtherCAT**

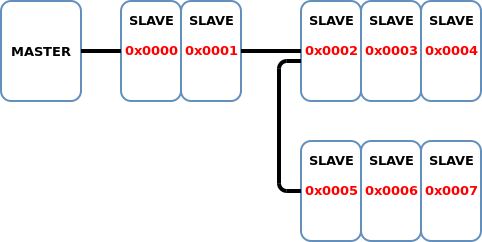
Jednoduchý protokol podobný TFTP umožňuje souborový přístup v zařízení a nahrání firmwaru přes EtherCAT síť.

## Adresování

Slave zařízení mohou být síti adresována inkrementálně, pevně, nebo logicky. Všechny tři způsoby jsou popsány níže.

### Inkrementální

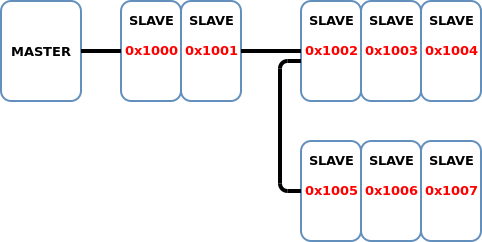
Zařízení jsou adresována postupně v pořadí, ve kterém jsou zapojena. První slave za masterem dostane adresu 0, uloží ji, inkrementuje a pošle dalšímu. Délka adresy je 2 byty.



Obrázek 11 Inkrementální adresování

### Pevné

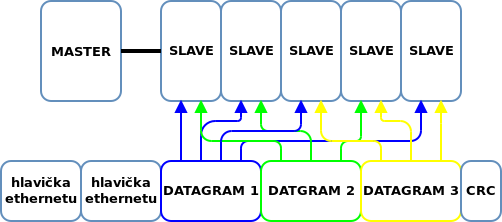
Zařízení má přidělenou adresu, které neodpovídá jeho pořadí v síti. Adresa je ztracena v případě výpadku napětí. Délka adresy je 2 byty.



Obrázek 12 Pevné adresování

### Logické

Slave čte/zapisuje data z/do logického adresového prostoru o velikosti 4GB. Délka adresy je v tomto případě 32 bitů.



Obrázek 13 Logické adresování

## Synchronizace

V aplikacích, kde je potřeba provádět více operací současně je kladen velký důraz na sychronizaci jednotlivých prvků. Jako příklad lze uvést více servomotorů provádějící koordinovaný pohyb. Ethercat nabízí možnost distribuce času, díky čemuž lze procesy synchronizovat s přesností na méně než 1 μs.

Master je schopen vyrovnat zpoždění na sběrnici tak, že zjistí čas potřebný pro průchod mezi jednotlivými slave zařízeními. Toho se dosáhne vysláním synchronizačního datagramu, do něž každý slave uloží čas průchodu a čas návratu. Poté vypočte zpoždění každého zařízení a počítá s ním v následujících operacích. Referenční čas je běžně brán z prvního slave zařízení, které má schopnost distribuovat čas.

## EtherCAT state machine

# masteR

Jak už bylo zmíněno v ?????????, k implementaci mastera není zapotřebí žádný speciální hardware. Jedinou podmínkou k jeho vytvoření je zařízení disponující ethernetovým kontrolérem. Masterem se tedy může stát vše od průmyslových PC, přes obyčejná PC až k zařízením jako je raspberry pi.

V tomto projektu byl EtherCAT master implementován do notebooku a raspberry pi. V závěru došlo ke srovnání jejich vlastností, jako jsou zpoždění způsobená výkonem zařízení a jitter.

## Porovnání

K dispozici je několik placených i open source řešení EtherCAT mastera. Těmi jsou TwinCAT od firmy Beckhoff, která je zodpovědná za vymyšlení EtherCATu. Tento software je založený na vývojovém prostředí Visual Studio od společnosti Microsoft. Lze jej získat a vyzkoušet ve formě dema, které je kromě časového omezení plně funkční.

Dalšími možnostmi jsou open source řešení EtherLAB a Simple Open EtherCAT Master, obě vytvořená pro operační systém Linux.

První zmíněné, EtherLAB, je napsán pro starší verze Kernelu a je použitelný ve formě modulu běžící v kernel space operačního systému. Jeho výhodou je tedy větší determinističnost. SOEM je oproti tomu dodáván ve formě knihoven a spustitelný z User space operačního systému, což jej činí nedeterministickým, avšak jednodužším na implementaci. Po zvážení pozitiv a negativ obou řešení jsem se rozhodl pro SOEM, neboť v tomto případě nepotřebuji mít aplikaci deterministickou, ale jednoduše implementovatelou. SOEM dodává ke zdrojovým kódům také demo aplikaci, na které jsem stavěl a uprovaval ji dle potřeb práce.

## Implementace

## Aplikace

# slave

Slave zařízení použité v práci je XMC4800 Relax kit od firmy Infineon.

## Porovnání

## Implementace

## Information file

## Aplikace

### Procesní data

### Firmware update

# výkon

# Porovnání s 1gbit

# Závěr

Literatura

Seznam příloh

Příloha 1. Manuál ...

Příloha 2. Zdrojové texty ...

Příloha 3. CD/DVD ...