

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

**Komunikační modul pro
průmyslovou sběrnici
FOUNDATION Fieldbus**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Pavel BÖHM**
Osobní číslo: **A10N0025P**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Číslicové systémy**
Název tématu: **Komunikační modul pro průmyslovou sběrnici Foundation
Fieldbus**
Zadávací katedra: **Katedra informatiky a výpočetní techniky**

Z á s a d y p r o v y p r a c o v á n í :

1. Prostudujte základní vlastnosti a metody přenosu průmyslových sériových sběrnic, zejména pak sběrnici Foundation Fieldbus.
2. Specifikujte požadavky na komunikační modul včetně uživatelského interface.
3. Proveďte návrh komunikačního modulu.
4. Navržený modul implementujte ve formě funkčního modulu.
5. Proveďte základní funkční testy komunikačního modulu.

Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **min. 40 stran původního textu**
Forma zpracování diplomové práce: **tištěná**
Seznam odborné literatury:
dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Doc. Ing. Vlastimil Vavříčka, CSc.**
Katedra informatiky a výpočetní techniky

Datum zadání diplomové práce: **31. srpna 2011**
Termín odevzdání diplomové práce: **17. května 2012**



Doc. Ing. František Vávra, CSc.
děkan



Prof. Ing. Jiří Šafařík, CSc.
vedoucí katedry

V Plzni dne 15. září 2011

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 15. 5. 2012

.....

Pavel Böhm

Abstract

FOUNDATION Fieldbus communication module

This thesis describes the FOUNDATION Fieldbus system. It gives a summary of a communication model of this fieldbus from data transmission using a physical medium to create a control loops in the user layer.

The main objective of the diploma thesis is development of a FOUNDATION Fieldbus communication module. The hardware and software design and implementation of the module is described.

The central component of the module is 32-bit microcontroller EFM32G880 based on ARM Cortex-M3. The module contains a fieldbus controller UFC100-F1, media access unit AMIS-49200 and circuits for serial communication RS232, USB to serial UART interface and RS485.

The function of the communication module was tested and verified on a fieldbus experimental segment.

Obsah

1	Úvod	8
2	Teoretická část	9
2.1	Průmyslové sběrnice	9
2.1.1	Historie	9
2.1.2	Stručný popis nejrozšířenějších sběrnic	10
2.2	FOUNDATION Fieldbus	11
2.2.1	Foundation H1 a High Speed Ethernet	12
2.2.2	Typy zařízení	13
2.2.3	Komunikační model	14
2.2.4	Fyzická vrstva	15
2.2.5	Linková vrstva	17
2.2.6	Aplikační vrstva	18
2.2.7	Uživatelská vrstva	19
2.2.8	Komunikace na sběrnici a struktura vybraných rámců	21
3	Praktická část	25
3.1	Požadavky na komunikační modul	25
3.2	Návrh komunikačního modulu	26
3.3	Vlastnosti a parametry použitých obvodů	28
3.3.1	Mikrokontrolér EFM32G880F128	28
3.3.2	Vývojová deska Olimex EFM32G880F128-H	31
3.3.3	Unified Fieldbus Controller UFC100-F1	32
3.3.4	Media Access Unit AMIS-49200	34
3.3.5	FT232RL	35
3.3.6	MAX3232	36
3.3.7	ADM-3078E	36
3.4	Realizace komunikačního modulu	37

3.5	Softwarová realizace komunikačního modulu	38
3.5.1	Inicializace mikrokontroléru	38
3.5.2	UART	39
3.5.3	Rozhraní externí sběrnice	41
3.5.4	Nastavení kontroléru UFC100	43
3.5.5	Možnosti přístupu k implementaci řídicího algoritmu	48
3.5.6	Implementace úloh ve FreeRTOS	49
3.6	Testování na segmentu průmyslové sběrnice FF-H1	51
3.7	Naměřené parametry komunikačního obvodu	53
3.8	Programová realizace monitorovacího programu	54
4	Závěr	55
	Přehled zkratk	56
	Literatura	57
	Přílohy	60

1 Úvod

Průmyslové sběrnice jsou důležitou součástí při monitorování, sběru, distribuci a vyhodnocování dat různého druhu, v nejrůznějších technických systémech a odvětví. Můžeme se s nimi setkat například v automatizovaném výrobním prostředí, řízení výroby a distribuce energie, telekomunikaci nebo i v dopravních prostředcích či v moderní domácnosti. Digitální průmyslové sběrnice prochází dynamickým vývojem a postupně nahrazují starší, na provoz i porízení náročnější, analogové rozvody. Vzhledem k jejich širokému spektru použití a potřeb jednotlivých firem, které v této oblasti působily, existuje několik rozdílných typů průmyslových sběrnic. Každá je přizpůsobena právě daným specifikám pro nasazení v konkrétním prostředí. Naopak najdeme i sběrnice, které jsou si dosti podobné, pouze vznikaly na jiném kontinentu, což bylo dáno politickou situací v době jejich vývoje.

Diplomová práce se soustřeďuje na problematiku několika moderních sběrnic, zejména na sběrnice FOUNDATION Fieldbus. Tyto získané znalosti jsou pak využity k návrhu a vývoji komunikačního modulu.

Cílem diplomové práce je vytvořit funkční experimentální modul pro průmyslovou sběrnici FOUNDATION Fieldbus. Modul je určen pro vývoj zařízení, jehož funkce bude závislá na programovém vybavení. Například jako převodník mezi sběrnicemi, konfigurátor nebo jiné reálné zařízení této sběrnice.

Výslednou realizaci tvoří výběr součástkové základny, osazení navržené a vyrobené desky plošných spojů a vytvoření programového vybavení modulu.

Základem je návrh struktury modulu tak, aby centrální mikrokontrolér řídil přídavné obvody, určené přímo pro implementaci nižších vrstev sběrnice. Stěžejní částí práce je ovládání těchto obvodů zvoleným mikrokontrolérem a vytvoření aplikační logiky zařízení. Aplikační logikou jsou myšleny vyšší vrstvy protokolu průmyslové sběrnice.

Pro testy sběrnice je sestaven experimentální segment, na němž je ověřena funkčnost realizovaného komunikačního modulu.

2 Teoretická část

2.1 Průmyslové sběrnice

Průmyslové sběrnice (fieldbus) slouží ke komunikaci mezi inteligentními zařízeními, určených k řízení technologických procesů a automatizovaných výrobních systémů. Nahrazují starší klasické analogové rozvody přenášející procesní veličiny. Tím umožňují snížit požadavky na množství kabeláže. Dovolují spravovat zařízení připojené ke sběrnici a vzdáleně jej diagnostikovat. Dále mohou snižovat nároky na řídicí systém. Chytrá zařízení na jednom segmentu spolu mohou komunikovat bez nutnosti jejich centrálního řízení a bez závislosti na funkčnosti ostatních segmentů. [5]

2.1.1 Historie

S nástupem mikrokontrolérů v 90. letech přestává vyhovovat klasická, pomalá a ve větších systémech složitě realizovatelná analogová technologie. Chytřejší zařízení umožňují přenášet data digitálně. Začalo vznikat několik, mezi sebou nekompatibilních, průmyslových sběrnic. Pro sjednocení do společného standardu organizace The International Society of Automation (ISA), International Electrotechnical Commission (IEC), Profibus (Německý národní standard) a FIP (Francouzský národní standard) vytvořily výbor nesoucí název IEC/ISA SP50 Fieldbus committee. ISA SP50 zformuloval komunikační model průmyslových sběrnic do těchto čtyř vrstev:

- Fyzická - specifikuje přenosové médium a přenos dat po něm.
- Linková - řídí přístup ke sběrnici a umožňuje přenos dat mezi adresovatelnými uzly.
- Aplikační - zapouzdřuje uživatelská data do zpráv srozumitelných pro všechna zařízení na sběrnici.
- Uživatelská - specifikuje funkční bloky řídicí aplikace.

Ale až na přelomu konce 20. století se ujednotil standard Mezinárodní elektrotechnické komise (International Electrotechnical Commission – IEC) za podpory organizací a firem

Fieldbus Foundation, ControlNETu, Profibusu a Siemensu pod označením IEC 61158. Ukázalo se, že nelze vytvořit jeden protokol vhodný pro všechny průmyslové oblasti působení, proto v něm bylo definováno 8 typů různých protokolů: FOUNDATION Fieldbus H1 a HSE, ControlNet, Profibus, P-Net, SwiftNet, WorldFIP a Interbus-S.

V současnosti se začínají prosazovat průmyslové sběrnice na bázi ethernetu. Mezi některé protokoly patří EtherNET/IP, FOUNDATION Fieldbus High Speed Ethernet nebo PROFINet.

Podrobněji historii vývoje popisují zdroje [3] a [5].

2.1.2 Stručný popis nejrozšířenějších sběrnic

Seznámení s několika nejpoužívanějšími sběrnici, které popisuje [5].

- **FOUNDATION Fieldbus:** Podrobně popsána v následující kapitole.
- **Profibus:** Německá organizace Profibus sdružuje přes tisíc členů, zabývajících se zařízeními pro průmyslové řízení. Profibus je jedna z prvních světově rozšířených sběrnic. Je primárně určena pro řízení procesů výroby. Používá metodu přenosu master/slave. Profibus existuje ve třech variantách FMS (FieldBus Message Specification), DP (Decentralized Periphery) a PA (Process Automation).

Profibus FMS je nejstarší varianta a v dnešní době se již příliš nepoužívá.

DP je nejpoužívanější varianta.

Profibus PA je upravená verze DP pro prostředí s nebezpečím výbuchu.

- **ControlNET:** V dnešní době je součástí sdružení ODVA, v rodině protokolů Common Industrial Protocol (CIP), mezi které patří také EtherNet/IP, DeviceNet a CompoNet. Nasazuje se pro komunikaci zařízení v průmyslové automatizaci. Umožňuje vysokorychlostní sériovou komunikaci v reálném čase. Používá metodu přenosu master/slave.

- **HART** (Highway Addressable Remote Transducer): Mezi další významný protokol pro průmyslovou sběrnici patří HART, který je také jeden z nejrozšířenějších, avšak se jedná o de facto standard. Dovoluje přenášet digitální data po analogové proudové smyčce 4 - 20 mA, takže obě technologie mohou být použity zároveň.
- **CAN** (Controller Area Network): Protokol byl původně navržen firmou Bosch pro použití v dopravních prostředcích, zejména v automobilech. Jedná se o sériovou komunikaci mezi zařízeními s decentralizovaným řízením v reálném čase. Je zde kladen důraz na vysokou spolehlivost.

2.2 FOUNDATION Fieldbus

FOUNDATION Fieldbus (FF) je průmyslová sběrnice vyvinutá stejnojmennou nezávislou organizací, kladoucí si za cíl stabilizovat oblast průmyslových sběrnic v automatizačním a výrobním procesu. Počátky vývoje sahají až do roku 1970, avšak přijata jako standard, významnými standardizačními organizacemi, byla až o třicet let později. Organizace sdružuje významné společnosti zabývající se vývojem zařízení pro automatizační a regulační techniku. Tato zařízení ověřuje a registruje jako kompatibilní s FF. Specifikace sběrnice je kompatibilní s projektem ISA SP50 a je součástí standardu IEC 61158. [4]

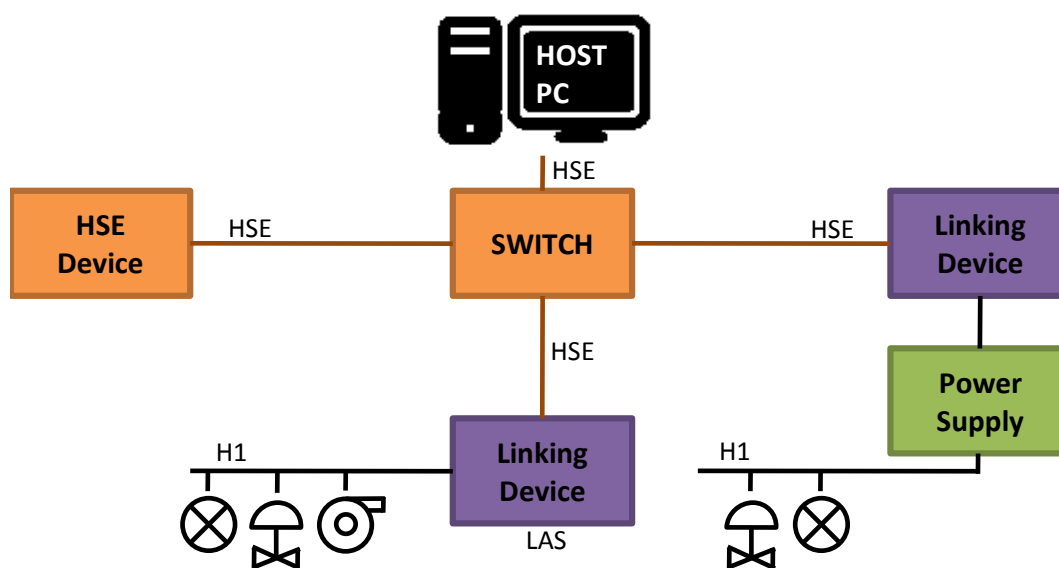
Sběrnice umožňuje vytvořit průmyslovou síť propojující řídicí a monitorovací střediska s akčními členy, ventily, senzory a kontrolními zařízeními.

Mezi hlavní znaky sběrnice patří:

- plně digitální poloduplexní sériová komunikace,
- schopnost vypořádat se s výpadky několika zařízení,
- na jeden pár vodičů může být připojeno více jednotek,
- napájení zařízení po jednom páru vodičů společně s daty,
- ochranné bariéry umožňující fungovat ve výbušném prostředí,
- možnost propojení vysokorychlostní páteřní sítě High Speed Ethernet (HSE).

2.2.1 Foundation H1 a High Speed Ethernet

FOUNDATION Fieldbus (FF) definuje dva komunikační protokoly. Protokol H1 je určen pro propojení průmyslových prvků, mezi které patří například senzory, převodníky a akční členy. Protokol HSE (High Speed Ethernet) slouží převážně pro vysokorychlostní propojení pracovních stanic, serverů a H1 podsystémů. Příklad struktury sběrnice FF je na obrázku 1. [4] [6]



Obrázek 1: Příklad sběrnice FF H1.

Foundation H1 používá kódování Manchester s rychlostí přenosu 31,25 kbps. Umožňuje napájet zařízení po sběrnici na jednom páru vodičů společně s daty pomocí připojeného zdroje na tento segment sběrnice. Odběr proudu zařízení na segmentu je samozřejmě omezen. Zařízení s větším odběrem (akční členy) musí mít tedy navíc svůj externí zdroj. Topologie je většinou sběrnicevá. Může být i hvězdicová za pomoci rozbočovače a nebo kombinovaná. Zařízení lze připojovat, konfigurovat a odpojovat za běhu (plug and play). Maximální počet zařízení na segment je 32 a jeho délka je omezena na 1900 metrů. Jejich konkrétní počet a délka spoje jsou závislé na napájení, oblasti použití a podobně. Jsou to tedy spíše teoretické hraniční hodnoty. Sběrnice je zakončena terminátory. Kabeláž je

doporučená stíněná, není to však podmínka. Pro komunikaci ve výbušném prostředí je potřeba sběrnici rozdělit bezpečnostní bariérou, která omezuje napětí a proud na bezpečné hodnoty. [2] [4] [6]

High Speed Ethernet je založen na klasické technologii *ethernet* a je kompatibilní s klasickými prvky sítě Ethernet. Jedná se o aplikační nadstavbu technologie ethernet a zaručuje stejnou funkčnost inteligentních zařízení jako H1. Přenosová rychlost je 100/1000 Mbps. Vzhledem k řízení kolizí na sběrnici metodou CSMA/CD, je pro real-time aplikace vhodné připojit jen několik málo zařízení na segment nebo použít switche. Množství kolizí je také minimalizováno plánováním vysílacích časových oken každého zařízení FF. Pro větší spolehlivost je možné síť i zařízení zdvojit, což podporuje samotný ethernet. Používá standardní ethernetovou kabeláž s možností použití i optických vláken pro ještě větší odolnost proti rušení. [2] [4] [6]

Pro komunikaci mezi oběma protokoly slouží H1-HSE mosty (H1-HSE Bridges). Mosty konvertují datové pakety.

Přehledné srovnání základních vlastností H1 a HSE je v následující tabulce 1.

	Foundation H1	HSE
Rychlost	31,25 kbps	100/1000 Mbps
Délka (segmentu)	1900 m	100 m
2-vodičové vedení	Ano	Ne
Napájení s daty	Ano	Ne
Práce ve výbušném prostředí	Ano	Ne

Tabulka 1: Základní vlastnosti H1 a HSE.

2.2.2 Typy zařízení

Z hlediska funkce, kterou zařízení na sběrnici FOUNDATION Fieldbus (FF) zaujímají, je lze rozdělit do těchto několika skupin [1]:

- **Basic Device** je základní zařízení, které dokáže komunikovat po sběrnici H1. Jedná se především o průmyslové prvky řídicích systémů.

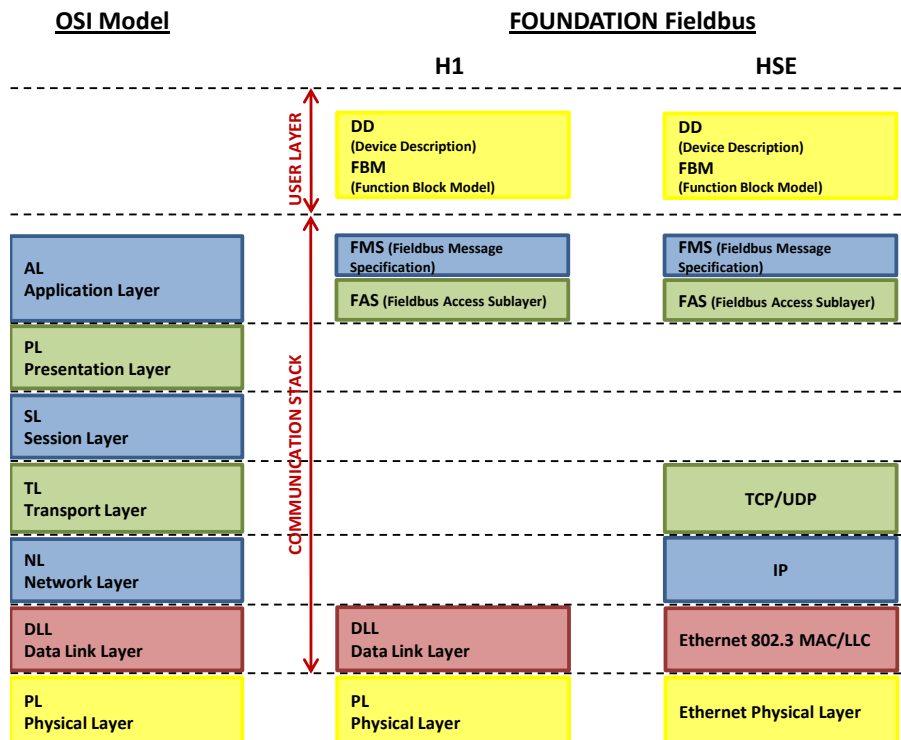
- **Link Masters** jsou navíc schopné řídit a plánovat provoz na H1 segmentu. Na každém segmentu musí být nejméně jedno takové zařízení.
- **H1 bridges** umožňují vytvářet stromovou topologii na sběrnici H1. Rozdělují segment sběrnice na menší a snižují tak možnost kolizí při přístupu ke sběrnici a poruch.
- **Linking Device** jsou prvky mezi H1 a HSE segmenty. Provádí konverzi datových paketů mezi těmito protokoly.
- **Ethernet Device** je průmyslové zařízení, které místo komunikačního zásobníku H1 používá HSE. Většinou se jedná o sofistikovanější zařízení, které využije větší datovou propustnost.
- **Gateway Device** je obecné zařízení, které propojuje FF s jiným typem sběrnice nebo s obecným vstupně/výstupním systémem.
- **Host Device** je pracovní stanice pro konfiguraci a správu zařízení a celé sběrnice. Obvykle osobní počítač s potřebným softwarem.

2.2.3 Komunikační model

Komunikační model FOUNDATION Fieldbus (FF) tvoří *Uživatelská Vrstva (User Layer)*, *Komunikační Zásobník (Communication Stack)* a *Fyzická Vrstva (Physical Layer)*.

V porovnání s referenčním modelem OSI (Open Systems Interconnect model) jsou u typu H1 nevyužity vrstvy 3 - 6, které souvisí se směrováním, navazováním spojení, udržováním sezení mezi dvěma uzly a správnou reprezentací dat. Naopak byla přidána uživatelská vrstva, která obsahuje bloky pro řízení průmyslové aplikace. Oba modely s referenčním OSI modelem jsou na obrázku 2. [1] [4] [6]

Pod pojmem komunikační zásobník je zapouzdřeno několik vrstev a záleží na konkrétním typu sběrnice (H1/HSE).



Obrázek 2: Komunikační modely OSI, H1 a HSE.

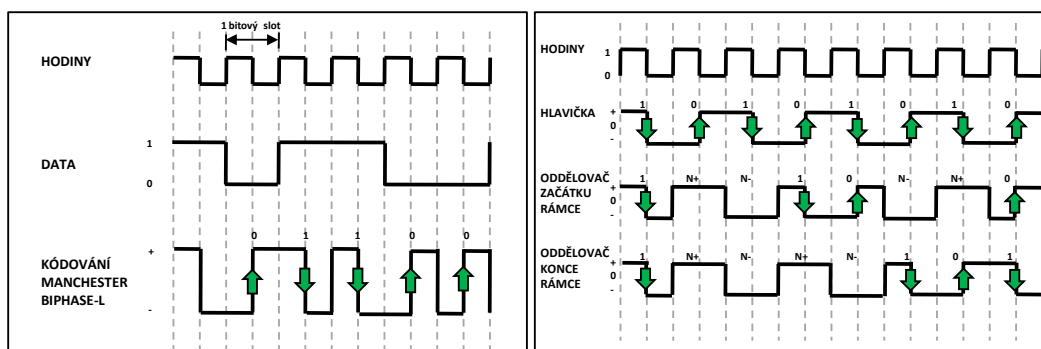
2.2.4 Fyzická vrstva

Stejně jako u OSI modelu připravuje data z vyšších vrstev pro přenos po komunikačním médiu a odděluje tak jejich funkce od fyzické reprezentace signálu. Přidává k datům synchronizační znaky: synchronizační hlavičku (Preamble), začátek (Start Frame) a ukončení rámce (End Frame). Specifikuje parametry pro kódování, napět'ové úrovně, napájení, kabeláž a podobně.

Pro H1 je použito synchronní kódování Manchester, který slučuje hodinový a datový signál do jednoho. Jednotlivé logické hodnoty jsou vyjádřeny změnou uprostřed bitového intervalu, která zároveň slouží k synchronizaci. Logická 1 je vyjádřena změnou z vysoké úrovně na nízkou, logická 0 pak opačnou hranou.

Hlavička rámce slouží k synchronizaci přijímače. Pro začátek a konec rámce slouží speciální znaky $N+$ a $N-$, které nejsou interpretovány změnou signálu uprostřed bitového intervalu.

Na obrázku 3 je ilustrován Manchester kód a zmiňované speciální synchronizační značky.

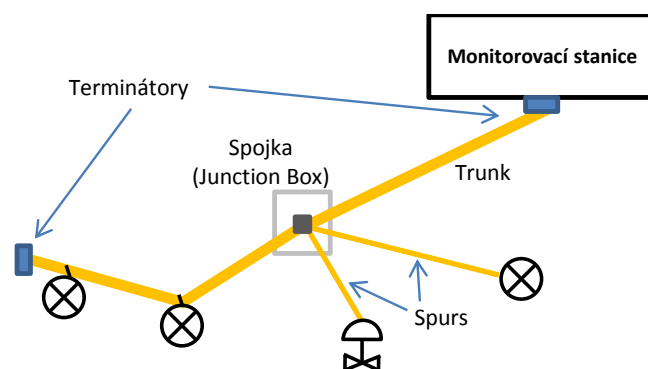


Obrázek 3: Manchester kódování a synchronizační značky.

Rychlost přenosu je 31,25 kbps u H1 a 100/1000 Mbit/s pro HSE.

Napětí na vodičích se pohybuje v rozmezí 9 - 32 V. Konkrétní hodnota je závislá na bezpečnosti prostředí.

Fyzická topologie může být stromová, hvězdicová nebo kombinací obou. Páteř sběrnice (trunk) je obvykle zakončena terminátory na obou koncích. Délky jednotlivých propojek (spurs) jsou závislé na počtu paralelních zařízení, typu kabeláže a prostředí. Doporučená se pohybuje v rozmezí od 1 metru pro dvaatřicet zařízení až do 120 metrů do dvanácti zařízení. Celková délka by neměla přesáhnout 1900 metrů. Viz obrázek 4. [2] [3] [6]



Obrázek 4: Příklad topologie sběrnice FF.

2.2.5 Linková vrstva

Linková vrstva (Data Link Layer) umožňuje poslat data jinému uzlu na sběrnici. Rozděluje data do rámců a předává je fyzické vrstvě k odeslání na médium, provádí kontrolu chyb rámců a řídí přístup ke sběrnici.

Přístup ke sběrnici je přesně plánován aktivním zařízením typu Link Master (LM, Link Active Scheduler - LAS) na každém segmentu. Jedná se o komunikaci na způsobu předávání tokenů. LAS je zvolen z LM zařízení, které to první ohlásí a má vždy adresu $0x04$ nebo nižší.

Na sběrnici jsou uzly adresovány *DL-address*, která se skládá ze tří položek [2].

Link	Node	Selector
(16 bit)	(8 bit)	(8 bit)

Link je adresa segmentu. Je používána při komunikaci mimo aktuální segment skrz mosty, v případě komunikace na jednom segmentu je často vynechávána. *Selector* je používán pro identifikaci spojení v rámci zařízení DLCEP (Data Link Connection End Point). *Node* je adresa uzlu. Adresní rozsah je rozdělen podle typu zařízení, viz následující tabulka 2.

0x10 až V(FUN)	rozsah pro Link Master zařízení
V(FUN) + V(NUN) až 0xF7	rozsah pro Basic zařízení
0xF8 až 0xFC	defaultní adresy pro nová zařízení
0xFD až 0xFF	adresy pro dočasné zařízení

Tabulka 2: Adresní rozsahy

$V(FUN)$ a $V(NUN)$ jsou parametry pro adresní mezeru, označující adresy, se kterými se uzel nemůže připojit.

Na linkové úrovni se rozeznávají následující typy rámců (Data Link Protocol Data Units - DL PDUs) (tabulka 3) [2] [6]:

DLPDU	Název	Funkce
EC	Establish Connection	připojení DLCEP
DC	Disconnect Connection	odpojení
CD	Compel Data	výzva publisheru
DT	Data Transfer	odeslání datové jednotky
PT	Pass Token	poskytnutí tokenu
RT	Return Token	vrácení tokenu
RI	Request Interval	žádost o více tokenů
PN	Probe Node	vyhledávání nových uzlů
PR	Probe Response	odpověď nového uzlu
TD	Time Distribution	synchronizace času
CT	Compel Time	vynucení synchronizace času
RQ	Round-trip Time Query	měření času CT
RR	Round-trip Time Response	měření času CT
CL	Claim LAS	ohlášení LAS
TL	Transfer LAS	žádost o LAS
IDLE	Idle	neaktivita

Tabulka 3: Typy rámců FF.

2.2.6 Aplikační vrstva

Aplikační vrstva je rozdělena na dvě podvrstvy *Fieldbus Access Sublayer* a *Fieldbus Message Specification*.

Fieldbus Access Sublayer (FAS) poskytuje 3 typy virtuálních spojení (Virtual Communication Relationship, VCR). Přehled VCR je v tabulce 4. *Client/Server VCR* spojení slouží pro nastavování zařízení operátorem. *Report Distribution VCR* slouží k rozeslání událostí a alarmů. *Publisher/Subscriber VCR* je komunikace pro rozesílání dat. [2] [3] [6]

Client/Server	Report Distribution	Publisher/Subscriber
Zprávy operátora	Zprávy s událostmi a hlášeními	Distribuce dat
Změny požadovaných hodnot Změny módů Ladění Upload/Download Nastavení alarmů Nastavení pohledů Vzdálená diagnostika	Zasílání alarmových hlášení Zasílání dat pro vytváření histogramů	Zasílání regulačních údajů jiným zařízením nebo operátorovi

Tabulka 4: 3 typy VCR.

Fieldbus Message Specification (FMS) vytváří strukturu zprávy pro přenos po sběrnici z dat od uživatelské vrstvy a poskytuje služby pro síťovou komunikaci. Data připravuje ke správné interpretaci, jak mají být použita v jiném zařízení.

V jednom fyzickém zařízení je jeden nebo více virtuálních zařízení *Virtual Field Device*, které obsahují jeho určité funkce. Umožňují tak oddělit více funkcí jednoho zařízení, které spolu ale nesouvisí. Obvykle je jedno VFD pro správu (Network Management a System Management) a další VFD vykonávají nějakou funkci zařízení.

System management spravuje adresy a tagy zařízení, řídí distribuci času a plánování komunikace. Řídí systém v nenormálních situacích (spouštění, špatná konfigurace, selhání zařízení). *Network management* umožňuje prostřednictvím sběrnice FF vzdáleně číst a zapisovat objekty komunikačního zásobníku. [2] [4] [6]

FMS spravuje objekty, pomocí kterých jsou popsány jednotlivé bloky a parametry zařízení ostatním zařízení na sběrnici. Tyto popisy objektů (*Object Description*) jsou uloženy v seznamu objektů (*Object Dictionary* - OD).

V OD jsou jednotlivé popisky identifikovány indexy. Tyto indexy jsou unikátní v rámci jednoho VFD. Index 0 odkazuje na hlavičku OD. Další indexy do 255 jsou rezervovány pro datové typy a pro popis složitějších datových struktur. Nad 255 jsou indexovány uživatelské služby.

Pokud VFD komunikuje s jiným zařízením, správnou interpretaci přenesených hodnot se dozví z jeho OD. [2] [4] [6]

2.2.7 Uživatelská vrstva

Používá bloky a objekty, které zapouzdřují data, parametry a funkce jednotlivých zařízení. Pomocí těchto bloků vytváří operátor řídicí aplikaci.

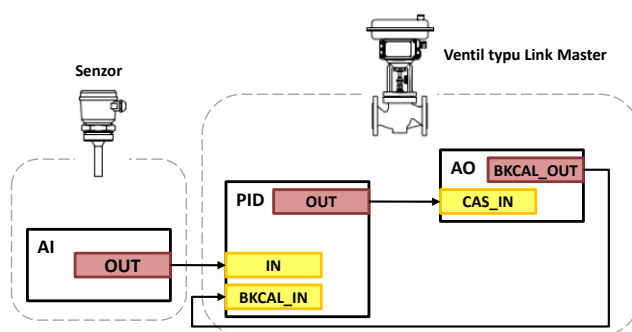
Device Description (DD) obsahuje soubor všech funkcí, které zařízení umožňuje. Jsou zde uvedeny parametry, jak interpretovat hodnoty, měřítka, jednotky a podobné. Pokud zařízení umožňuje nestandardní funkci, musí být k její plnému využití zde popsána. K tomuto

popisu slouží Device Description Language.

Uživatelská vrstva dále definuje **3 typy bloků**. *Resource Block* je hlavní blok uchovávající informace o zařízení, jako je typ zařízení, výrobní číslo, sériové číslo a stav ostatních bloků. V zařízení je vždy právě jeden tento blok. *Transducer Block* odděluje funkční bloky od fyzického rozhraní akčních členů nebo sensorů. Funkční bloky (*Function Block*) umožňují vytvořit monitorovací či řídicí aplikaci, kterou zařízení na sběrnici vykonává. Standardní funkční bloky jsou [2] [3]:

AI: Analog Input	AO: Analog Output
BG: Bias/Gain	CS: Control Selector
DI: Discrete Input	DO: Discrete Output
ML: Manual Loader	PD: Proportional/Derivate
PID: Proportional/Integral/Derivate	RA: Ratio

Propojením těchto bloků a nastavením parametrů v příslušném programu lze jednoduše sestavit řídicí aplikaci (*Function Block Application*). Příklad aplikace řídicí smyčky sestavené ze dvou fyzických zařízení a třech funkčních bloků se nachází na obrázku 5. Hodnota ze senzoru je přivedena na PID člen ventilu. Působení ventilu je zpětně odečítáno do PID členu v rámci jednoho zařízení, takže se tato hodnota neúčastní přenosu po sběrnici. [3]



Obrázek 5: Příklad sestavení řídicí smyčky z funkčních bloků.

Funkční bloky jsou identifikovány pomocí názvu (*Tagu*) a indexem v seznamu objektů (Object Dictionary).

Kromě těchto tří bloků jsou definovány další pomocné **objekty**. *Link objects* definují spojení mezi funkčními bloky. *View objects* jsou skupiny uživatelem nadefinovaných parametrů, které chce zobrazit. *Alert objects* umožňují zařízení vytvořit a odeslat alarmovou zprávu. *Trend objects* uchovávají hodnoty s časovou značkou pro reprezentaci stavu v grafu. [2]

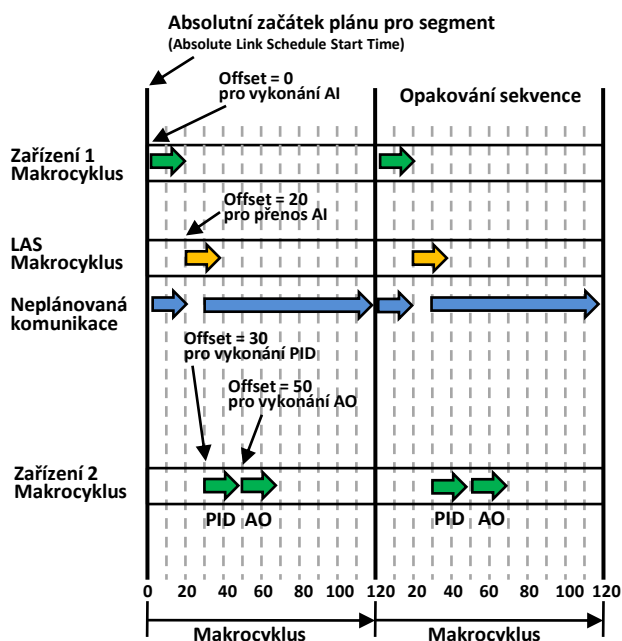
2.2.8 Komunikace na sběrnici a struktura vybraných rámců

Komunikace na sběrnici FOUNDATION Fieldbus (FF) je rozdělena na plánovanou (**Scheduled**) a neplánovanou (**Unscheduled**). Plánovaný přenos dat slouží pro cyklické přenosy řídicích smyček mezi zařízeními na sběrnici. Pokud není v danou chvíli naplánována žádná komunikace po segmentu, lze ji použít pro neplánovanou komunikaci. Do neplánované komunikace spadá nastavování parametrů a diagnostika zařízení. [2] [4]

Vykonání funkčních bloků a přenos dat je striktně plánován. Toto načasování provádí operátor při konfiguraci FF systému. O plánování na segmentu se poté stará *Link Active Scheduler* (LAS), který řídí komunikaci na segmentu. LAS se také stará o synchronizaci času rozesláním speciálního paketu Time Distribution (TD). Krom toho také rozesílá Compel Data (CD), Pass Token (PT) a Probe Node (PN). CD slouží jako výzva pro zařízení (publisher), aby neprodleně po doručení odeslalo požadovaná data daným příjemcům (subscriber). Ve volném časovém slotu LAS vyšle jednomu vybranému zařízení v seznamu (*Live List*) PT, který dovoluje zařízení využít sběrnici pro neplánovanou komunikaci. Zařízení poté může komunikovat, jak dlouho potřebuje a dokud nevyprší timeout. Pokud zařízení na PT několikrát za sebou neodpoví, je vyřazeno z Live Listu. PN token je broadcastován všem zařízením, které nejsou v Live Listu a slouží pro přidání nového zařízení. Pokud některé zařízení odpoví paketem Probe Response (PR), LAS ho přidá do seznamu. [2] [3] [4]

Jeden cyklus iterace plánování se nazývá makrocyklus (macrocycle). Vykonávání kaž-

dého funkčního bloku a distribuce dat jsou určena offsetem od začátku makrocyklu a o jejich nastavování se stará operátor. V momentě doručení CD, musí být data s potřebným předstihem připravena, aby mohla být ihned odeslána. Na obrázku 6 je příklad načasování operací a komunikace sběrnice FF. [4]

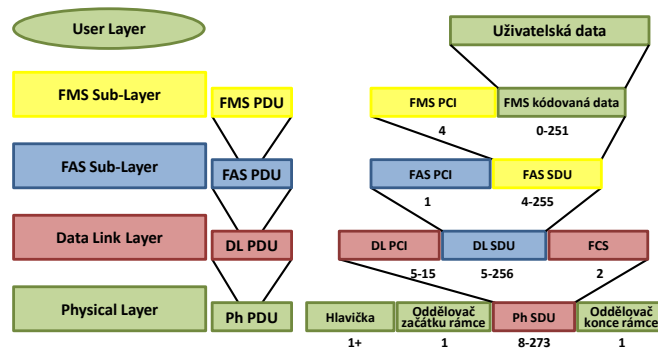


Obrázek 6: Příklad plánování aplikace FF.

Průchodem uživatelských dat jednotlivými vrstvami komunikačního modelu jsou data obalena přidáním informací každé vrstvy, jak ukazuje obrázek 7. [5]

Datová jednotka (Protocol Data Unit - PDU) je označení pro data, která jsou mezi stanicemi na stejné vrstvě. V každé vrstvě je do PDU přidána řídicí informace vrstvy (Protocol Control Information - PCI) a zapouzdřená data vyšší vrstvy (Service Data Unit - SDU).

Linková vrstva ještě připojuje vypočtený kontrolní součet celého rámce (Frame Check Sequence - FCS). Na úrovni fyzické vrstvy se k rámci přidávají synchronizační značky.



Obrázek 7: Průchod uživatelských dat jednotlivými vrstvami komunikačního modelu.

Struktura rámců a jejich délka závisí na typu rámce. Jako příklady jsou zde uvedeny rámce CD a DT pro plánovanou komunikaci a pro neplánovanou komunikaci rámce PT a DT.

Rámec CD od LAS slouží pro výzvu publishera k odeslání dat v **plánované komunikaci**. *Frame Control* obsahuje kód CD rámce s informací o délce adresy a prioritou. Následuje adresa příjemce, kontrolní součet, oddělovač a mezera, kdy na médiu nesmí nikdo vysílat (Post TX Gab).

Typická struktura rámce Compel Data (CD) Data Link Protocol Data Unit (DLPDU).

Preamble	Start	Frame	Publisher	Frame Check	End	Post TX
(1 B)	Delimiter	Control	Address	Sequence	Delimiter	Gab
(1 B)	(1 B)	(1 B)	(2 B)	(2 B)	(1 B)	(1 B)

Odpovědí na CD je DT od publishera, který byl vyzván k odeslání dat. Obsahuje adresu zdroje, která může být vynechána, následují řídicí informace a data.

Struktura rámce Data Token (DT) DLPDU.

Preamble	Start	Frame	Source	Control	FAS	FMS	Data	FCS	End	Post TX
(1 B)	Delimiter	Control	Address	(1 B)	(1 B)	(7 - 8 B)	(5 B)	(2 B)	Delimiter	Gab
(1 B)	(1 B)	(1 B)	(2 B)	(1 B)	(1 B)	(7 - 8 B)	(5 B)	(2 B)	(1 B)	(1 B)

Ve volném časovém slotu pro **neplánovanou komunikaci** je zde uveden příklad PT, který dovoluje příjemci použít sběrnici. Tento token obsahuje tedy adresu příjemce a dobu,

po kterou může využívat sběrnici.

Struktura rámce Pass Token (PT) DLPDU.

Preamble	Start Delimiter	Frame Control	Sender Address	Duration	Frame Check Sequence	End Delimiter	Post TX Gab
(1 B)	(1 B)	(1 B)	(1 B)	(2 B)	(2 B)	(1 B)	(1 B)

Po obdržení tokenu může zařízení označené jako klient vyslat datový token, který obsahuje adresu klienta (*Source Address*), adresu serveru (*Destination Address*) a informaci o požadované činnosti a datech.

Struktura klientského rámce Data Token (DT) DLPDU.

Preamble	Start Delimiter	Frame Control	Dest Addr	Src Addr	Control	FAS	FMS	Index	FCS	End Delimiter	Post TX Gab
(1 B)	(1 B)	(1 B)	(2 B)	(2 B)	(1 B)	(1 B)	(4 B)	(2 B)	(2 B)	(1 B)	(1 B)

Zařízení na straně serveru musí počkat na povolení k vysílání PT od LAS. Poté může vyslat odpověď s požadovanými daty klientovi.

Struktura rámce Data Token (DT) DLPDU od serveru.

Preamble	SD	Frame Control	Dest Addr	Src Addr	Control	FAS	FMS	Data	FCS	End Delimiter	Post TX Gab
(1 B)	(1 B)	(1 B)	(2 B)	(2 B)	(1 B)	(1 B)	(4 B)	(1 - 251 B)	(2 B)	(1 B)	(1 B)

3 Praktická část

V této části je popsán vývoj komunikačního modulu pro průmyslovou sběrnici FOUNDATION Fieldbus (FF) - H1.

3.1 Požadavky na komunikační modul

Pro komunikační modul jsou v současné době možná následující uplatnění:

- Součást libovolného zařízení typu *Basic Device*, pracujícího na sběrnici FF.
- Most mezi sběrnici MODBUS a FF. Podobné zařízení (FINT T610) je k dispozici pro vytvoření experimentálního segmentu.
- Konfigurátor sběrnice FF, který vytváří rozhraní pro sběrnici FF v osobním počítači přes USB. Podobné zařízení (USB-8486) je dostupné pro vytvoření experimentálního segmentu sběrnice.
- Pasivní zařízení, které by monitorovalo komunikaci na segmentu sběrnice. Zachycenou komunikaci bude odesílat přes sériové rozhraní do osobního počítače, kde jsou data zobrazována.

Fyzická a linková vrstva průmyslové sběrnice je implementována kontrolérem UFC100-F1 a obvodem pro přístup na sběrnici AMIS-49200. Aplikační a uživatelskou vrstvu realizuje zvolený mikrokontrolér. Jelikož je použit kontrolér UFC100-F1, který také implementuje nižší vrstvy sběrnice Profibus PA, lze komunikační modul po patřičných úpravách využít i pro tento typ sběrnice.

Komunikační modul musí mít tedy standardní sériovou linku RS232 pro komunikaci s osobním počítačem. Dále rozhraní RS485 pro případnou komunikaci na sběrnici MODBUS.

Kontrolér UFC100 je řízen jednočipovým mikrokontrolérem, který musí pro tento účel splňovat většinu níže zvolených kritérií.

Požadavky na mikrokontrolér:

- Rozhraní s externí sběrnici pro komunikaci s UFC100-F1 (min 8 b data, 7 b adresa - je vycházeno ze specifikace UFC100).
- Periferie alespoň pro jednu sériovou linku RS232 (UART) a volné porty pro linku RS485.
- Dostatečný výpočetní výkon. Paměti RAM alespoň 64 kB a ROM 128 kB pro program. S podporou pro případné použití operačního systému reálného času.
- Cenově dostupný vývojový kit s vyvedenými potřebnými piny. Vývojový kit urychlí realizaci a umožňuje snadnou opravu při výskytu možných chyb.
- Snadné programování a ladění ve volně dostupném vývojovém prostředí.
- Nízký odběr.

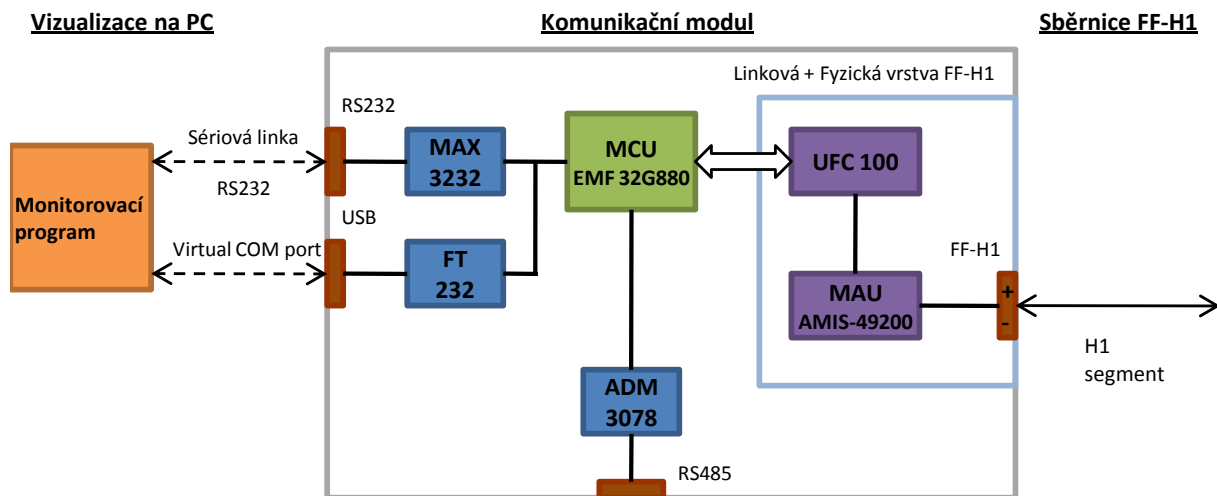
Modul také musí umožňovat monitorování komunikace na sběrnici, což ulehčí odladění komunikačního protokolu FF. Monitorovací aplikace na osobním počítači komunikuje s modulem sériovou linkou.

3.2 Návrh komunikačního modulu

Podle výše specifikovaných požadavků je navrženo následující blokové schéma komunikačního modulu (obr. 8).

První funkční vzorek komunikačního modulu je navržen s ohledem na snadné ladění a snadné měření významných signálů přímo na desce plošných spojů.

Jako mikrokontrolér (MCU) je zvolen EFM32G880F128, který splňuje kladené nároky na rozhraní externí sběrnice, sériová rozhraní, výkon a další. Pro první verzi je vybrán tento MCU také hlavně kvůli cenově dostupné vývojové desce od společnosti Olimex, která má vyvedeny všechny potřebné piny MCU v konektorových lištách.



Obrázek 8: Blokové schéma navrženého komunikačního modulu.

Návrh obsahuje sériová rozhraní RS232 (popř. i alternativu virtuálního COM portu přes USB) a RS485, ovládané tímto MCU a upravované na patřičné napěťové úrovni. K úpravě napěťových úrovní, pro použitá rozhraní, slouží k tomu určené převodníky.

Na straně sběrnice FF je kontrolér UFC100-F1 řízený po externí sběrnici MCU. Kontrolér implementuje první dvě komunikační vrstvy sběrnice FF. Přímý přístup ke sběrnici pak obstarává jednotka MAU AMIS-49200.

Programová realizace spočívá v obsluze použitých periferií. Zejména pak vybírání rámců z FIFO fronty kontroléru UFC100, jejich zpracování, odesílání přijatých rámců na sériovou linku pro monitoring, generování odpovědi na jednotlivé rámce a jejich odeslání opět přes kontrolér UFC100.

Podle aplikace použití jsou programově implementovány zbylé dvě vrstvy sběrnice FF, aplikační a uživatelská.

Konfigurace komunikačního modulu, tedy jeho uživatelské rozhraní, je realizována standardně přes sběrnici FF jako většina zařízení, která jsou určena pro tuto sběrnici.

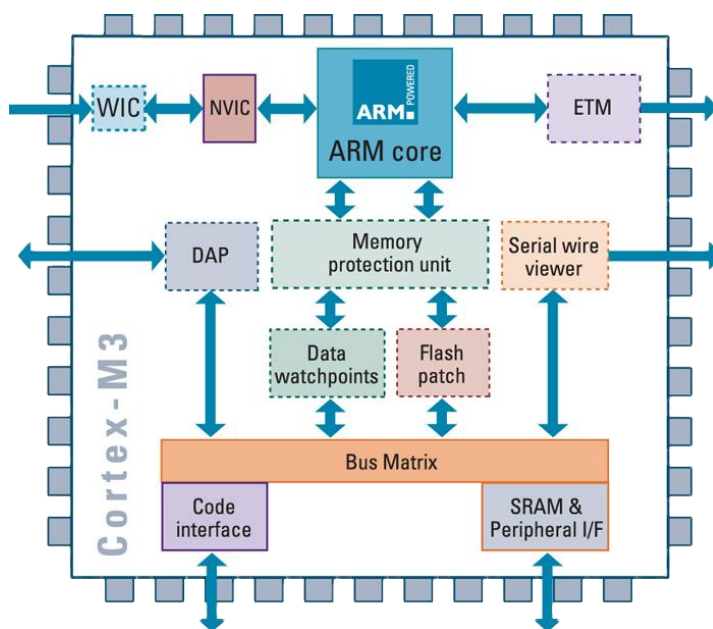
3.3 Vlastnosti a parametry použitých obvodů

Následuje přehled základních vlastností a parametrů použitých obvodů v komunikačním modulu.

3.3.1 Mikrokontrolér EFM32G880F128

Jako centrální prvek komunikačního modulu je použit mikrokontrolér EFM32G880F128 od norské společnosti Energy Micro. Jedná se o 32 bitový, nízkoenergetický mikrokontrolér s jádrem založeném na architektuře ARM Cortex-M3.

Cortex-M3 (obr. 9) je založen na Harvardské architektuře, má tedy oddělenou datovou a instrukční sběrnici. Instrukce a data mohou být načítány zároveň. Tím zvyšuje výkon zpracování instrukcí. Používá 3 stupňovou pipeline a sadu instrukcí označovanou jako Thumb-2, která dokáže dekódovat jak 32, tak i 16 bitové instrukce. Celý procesor je postaven na nízké spotřebě při vysokém výkonu. Jako většina dnešních procesorů podporuje hardwarové ladění standardem JTAG/SWD.



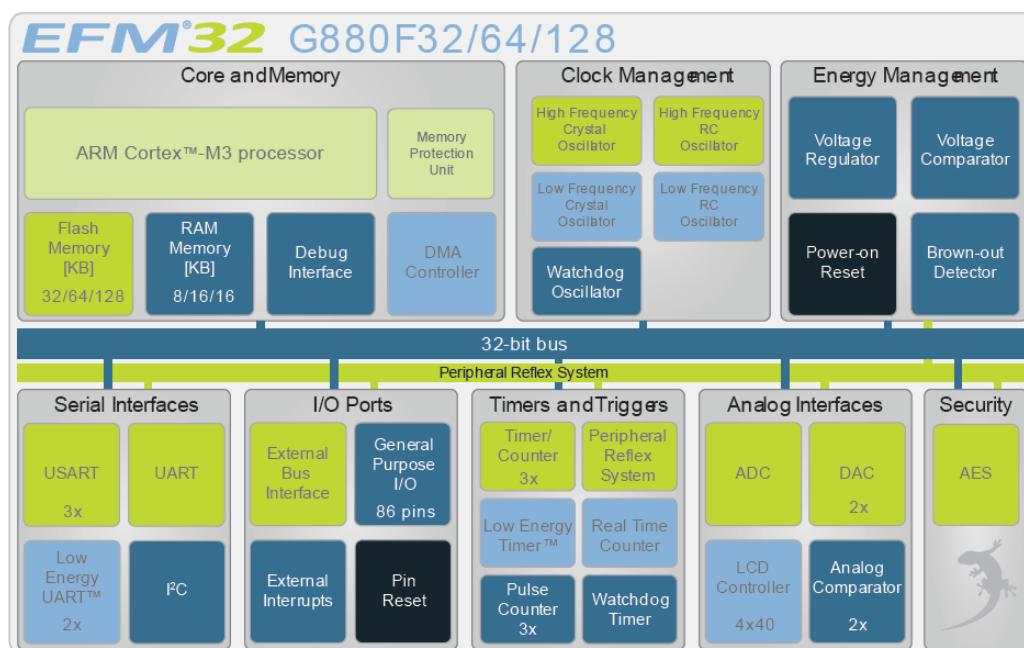
Obrázek 9: Blokové schéma Cortex-M3 [7].

Wakeup Interrupt Controller (WIC) a Nested Vectored Interrupt Controller (NVIC) je systém pro řízení přerušení. NVIC navíc obsahuje systémový časovač, který usnadňuje časování a běh operačních systémů reálného času.

Memory Protection Unit (MPU) je jednotka pro práci s paměťovými banky. Řídí přístup do paměti pomocí přístupových práv. Umožňuje chránit paměť před přepisem nebo nepovoleným čtením.

Debug Access Port (DAP) slouží pro ladění protokolem JTAG/SWD a jednotka Flash Patch s jednotkou Data Watchpoint umožňují použití až 6-ti programových breakpointů. Pro ladění slouží také jednotka Embedded Trace Macrocell (ETM). [7]

Mikrokontrolér **EFM32G880F128** je popsán několika funkčními bloky (obr. 10).



Obrázek 10: Blokový diagram mikrokontroléru [8].

Core and Memory blok je popsán výše v odstavci Cortex-M3. MCU obsahuje 128 kB flash paměti a 16 kB RAM.

V *Clock Management* je zapouzdřeno ovládání časování mikrokontroléru. MCU umožňuje použít 4 zdroje hodin, podle napájecího módu a rychlosti zotavení. Lze tedy využít rychle rozbíhané RC oscilátory, tak i na náběh pomalejší, ale přesnější krystalové oscilátory. Od obou typů podporuje jeden rychlejší (1 - 32 MHz) a jeden pomalejší (32,768 kHz).

MCU rozlišuje pět napěťových režimů (*Energy Management*). Podle jednotlivých odstínů na obrázku 11 lze určit energetickou náročnost dané jednotky z obrázku 10. Od nejtmaší, energeticky nejméně náročné, až po nejsvětlejší, energeticky nejnáročnější.



Obrázek 11: Příkonové módy mikrokontroléru [8].

- **Energy Mode 0 (EM0)** - V tomto režimu lze aktivovat všechny jednotky a MCU odebírá při výkonu programu z flash paměti do $180\mu\text{A}/\text{MHz}$.
- **Energy Mode 1 (EM1)** - V EM1 je CPU v režimu spánku a odběr je do $45\mu\text{A}/\text{MHz}$. Ostatní periferie jako například UART, EBI nebo ADC mohou být stále používány.
- **Energy Mode 2 (EM2)** - EM2 je hlubší spánek CPU, kdy je spuštěn jen oscilátor $32,768\text{kHz}$ a jsou zachovány hodnoty v registrech a RAM. Používat lze jen nízkopříkonové periferie jako LEUART, LETIMER nebo RTC. Odběr je v jednotkách μA . Aktivování CPU je možné přerušením periferie nebo asynchronním přerušením externím pinem.
- **Energy Mode 3 (EM3)** - V tomto režimu jsou periferní jednotky a všechny oscilátory vypnuty. Hodnoty CPU registrů a RAM jsou stále uchovány. Vzbuzení je možné

asynchronním přerušením a zabere nepatrný čas nutný pro aktivování oscilátoru.

- **Energy Mode 4 (EM4)** - Pro tento režim je odběr v řádu jednotek nA a celý čip je vypnutý. Vzbuzení je možné pouze resetem.

Posledním blokem jsou periferní jednotky.

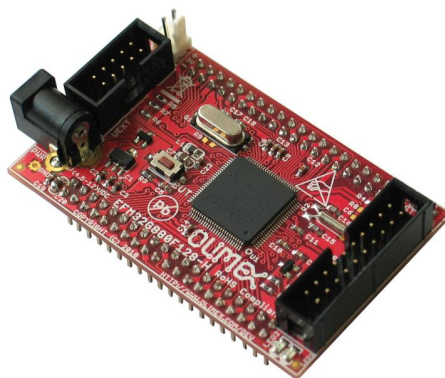
Ty se dělí na sériové rozhraní, mezi které patří standard UART (Universal Asynchronous Receiver/Transmitter), 3x USART (Universal Synchronous/Asynchronous Receiver/Transmitter) a I2C. Navíc je tu nízkopříkonová verze 2x Low Energy UART.

Dále tu jsou klasické vstupně/výstupní porty. Mezi ně patří i 16-ti bitové rozhraní externí sběrnice (External Bus Interface - EBI), kterou je možné ovládat externí zařízení napávané do adresního prostoru MCU.

Samozřejmostí jsou čítače/časovače. MCU má i 19-ti kanálový A/D převodník a D/A převodník. MCU obsahuje i kontrolér pro řízení LCD displeje 4x40 segmentů a AES (Advanced Encryption Standard Accelerator) jednotku pro hardwarové kryptování 128 - 256 bitovým klíčem. [8]

3.3.2 Vývojová deska Olimex EFM32G880F128-H

K vytvoření prototypu komunikačního modulu je použita vývojová deska EFM32G880F128-H od společnosti Olimex (obr. 12).

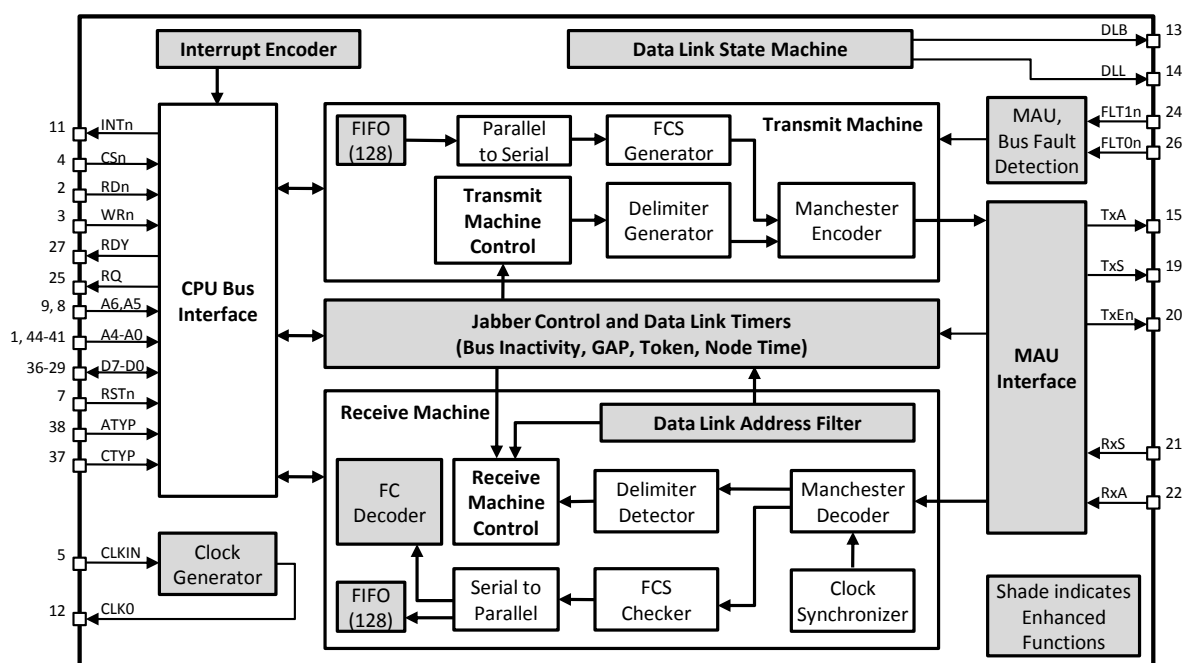


Obrázek 12: Vývojová deska s MCU EFM32G880F128 [9].

Deska je velmi jednoduchá a pro tyto účely plně dostačující. Obsahuje MCU, obvody pro napájení, krystaly a vývody pinů MCU pomocí lišt, které se dají vsadit do desky vyvíjeného komunikačního modulu.

3.3.3 Unified Fieldbus Controller UFC100-F1

UFC100-F1 implementuje část fyzické a linkové vrstvy FF-H1 a Profibus-PA. Jeho blokové schéma je na obrázku 13.



Obrázek 13: Blokové schéma kontroléru podle [10].

Kontrolér je nastavován a řízen několika registry. Registry jsou dostupné podle nastaveného módu. Po HW resetu se nachází v kompatibilním módu (*Compatible Mode*), avšak pro maximální využití lze kontrolér přepnout do rozšířeného módu (*Enhanced Mode*).

Registry jsou přístupné po 8-bitové paralelní sběrnici, která může pracovat v synchronním i asynchronním režimu. Adresování registrů je 7 / 5 bitové podle použitého módu. Kontrolér obsahuje prioritní dekodér přerušení a signalizuje ho do MCU pinem *INTn*. Pro zápis nebo čtení transportních dat lze použít režim DMA.

Pro interní časování potřebuje vstup hodinového signálu na pinu *CLKIN* (1 - 40 MHz), který musí být upraven vnitřními, uživatelsky nastavovatelnými, dělicími poměry na 500 kHz. Z těchto hodin je také odvozena rychlost přenosu na sběrnici 31,25 kbps.

Kontrolér má implementovanou jen část fyzické vrstvy (*MAU interface*). Pro navázání signálu na sběrnici a jeho odfiltrování ze sběrnice je potřeba další obvod, pro který má vyvedené signály *RxA* (*received signal activity*), *RxS* (*received signal*), *TxE* (*transmission signal enable*), *TxS* (*transmission signal*), *FLT0n* a *FLT1n* (*MAU fault*).

Pro vysílání disponuje 128 bytovou FIFO pamětí, kterou plní MCU po bytech na adrese příslušného registru. Celý paket musí být připraven MCU, kontrolér k němu pouze přidá synchronizační hlavičku, záhlaví a zápatí rámce a vypočítaný kontrolní součet. Takto sestavený rámec při pokynu odeslání kóduje Manchester kódem a odesílá přes MAU. Veškerá konfigurace vysílání, indikace nejrůznějších chyb a generování přerušení při dokončení přenosu je v příslušných registrech.

Přijímač stejně jako vysílací část obsahuje 128 bytovou FIFO paměť, Manchester enkodér a odděluje linkové záhlaví a zápatí od dat vyšších vrstev. Krom toho lze nastavit filtrování rámců podle adresy a módu zařízení (Basic Device / Link Master). Přijímací jednotka dokáže také rozeznat typ paketu, který se dá přečíst z registru.

Dále obsahuje několik HW čítačů a časovačů, které mohou být automaticky nastavovány při přijetí či nepřijetí daného rámce. Časovače mohou také generovat přerušení signalizující povel vykonání operace od plánovače FF pro přípravu dat.

V kontroléru lze nastavit i zpětnou smyčku pro příjem odeslaných dat a nastavit určité registry pro generování nějaké chyby, což umožňuje snadnější ladění a testování.

Data Link State Machine slouží pro vizuální signalizaci stavu kontroléru a lze na jeho piny připojit LED diody. Jejich funkci lze opět nastavovat v registrech.

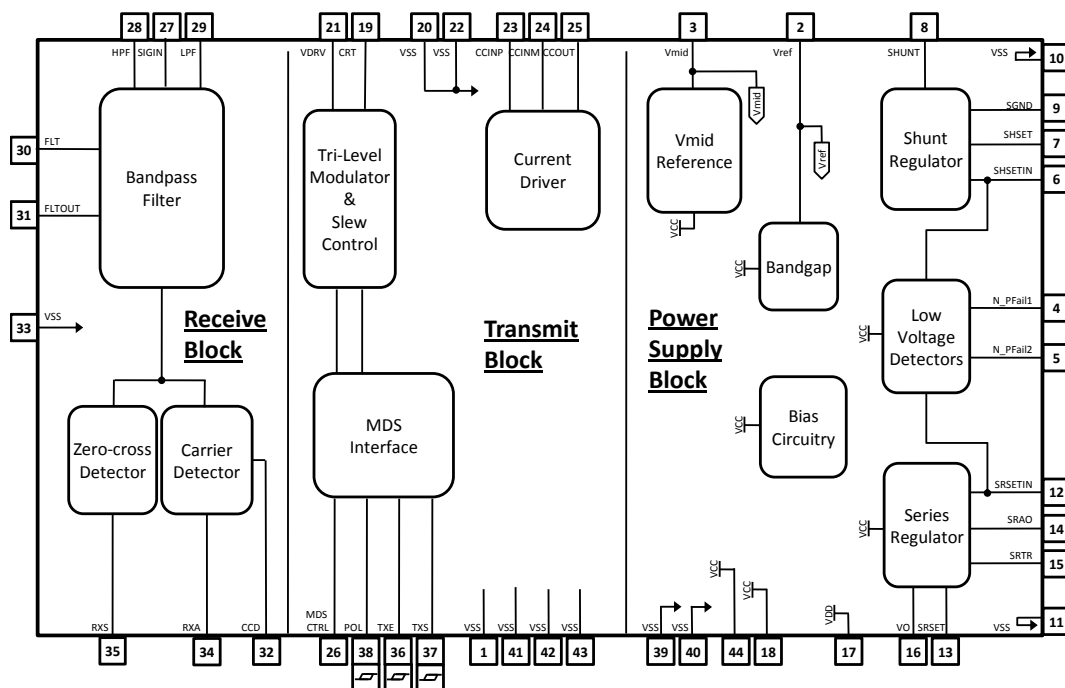
Podrobnější informace lze získat z uživatelského manuálu [10].

3.3.4 Media Access Unit AMIS-49200

Jednotka pro přístup na sběrnici (Media Access Unit - MAU) AMIS-49200 od společnosti ON Semiconductor implementuje část fyzické vrstvy průmyslových sběrnic Foundation Fieldbus-H1 a Profibus-PA. Obvod doplňuje rozhraní přístupu ke sběrnici kontroléru UFC100-F1, se kterým je propojen signály RxA (received signal activity), RxS (received signal), TxE (transmission signal enable) a TxS (transmission signal).

Pomocí analogových externích obvodů moduluje datový signál na sběrnici. Zároveň také filtruje datový signál ze sběrnice a detekuje jeho příjem signalizováním RxA.

Blokové schéma MAU je na obrázku 14.



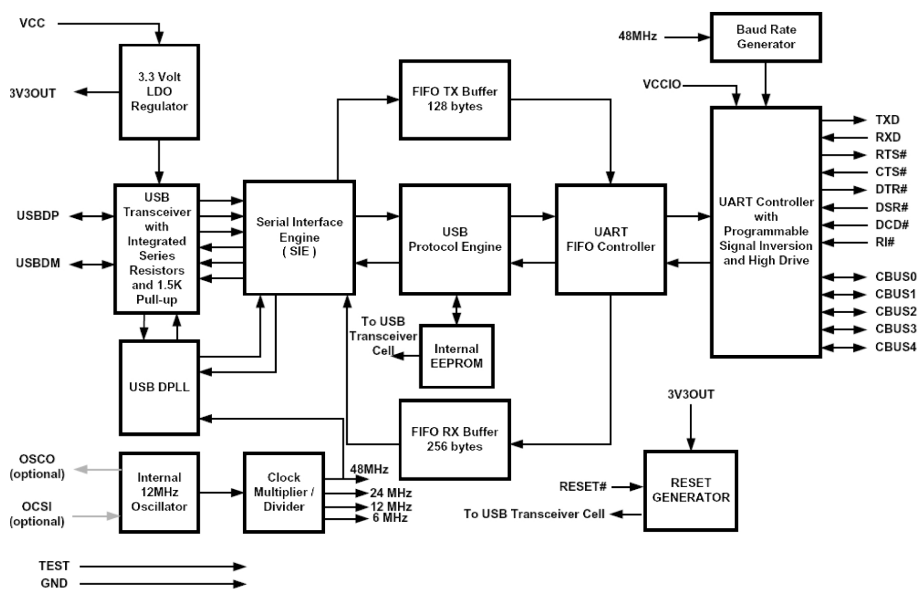
Obrázek 14: Blokové schéma MAU AMIS-49200 podle [11].

Obvod umožňuje odebírat napájení ze sběrnice vnitřními regulátory pro analogovou i digitální část. Také dokáže sledovat a signalizovat pokles napětí na sběrnici.

Doporučené zapojení analogové části s diskrétními součástkami je v katalogového listu výrobce [11], kde jsou zároveň podrobnější informace o tomto obvodu.

3.3.5 FT232RL

Tento obvod od firmy Future Technology Devices International Ltd. (FTDI) slouží pro virtualizování sériového portu v počítači přes USB. Jeho blokové schéma je na obrázku 15.



Obrázek 15: Blokové schéma obvodu FT232R [12].

Obvod implementuje kompletní USB protokol verze 1.1 a 2.0. Piny VCC, USBDP, USBDM a GND lze přímo připojit ke konektoru USB bez dalších externích součástek. Napájení z USB je interním regulátorem upravováno na 3,3 V pro napájení bloků USB vysílače/přijímače.

Řadič UART protokolu je napájen pinem VCCIO. Disponuje vyrovnávacími FIFO pamětmi a automatickým nastavením rychlosti komunikace. Standardní UART je rozšířen o ostatní řídicí piny protokolů RS232 a RS485.

Pro vnitřní časování obsahuje integrovaný generátor hodin, který je možné nahradit externím oscilátorem.

Paměť EEPROM slouží pro uložení USB Vendor ID (VID), Product ID (PID) a ostatních parametrů včetně výrobního označení. V této paměti je také uloženo volitelné nastavení signalizačních CBUS pinů [12].

Podrobnější informace o obvodu lze nalézt v příloženém katalogovém listu [12].

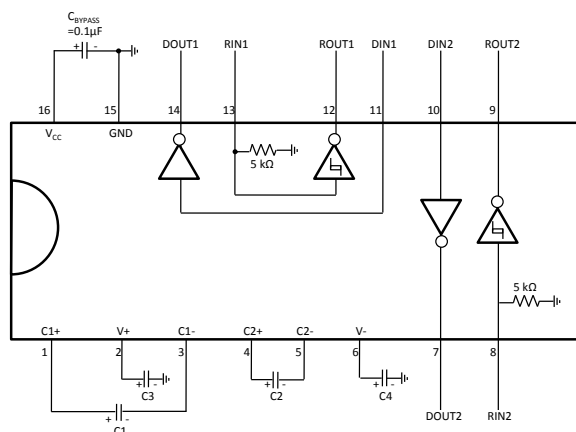
Po zapojení do počítače je nutné nainstalovat příslušné ovladače pro virtuální sériovou linku. Ty jsou volně dostupné na webových stránkách výrobce <http://www.ftdichip.com/>.

3.3.6 MAX3232

Obvod MAX3232 slouží k převodu napětí mezi TTL logikou a RS232, který používá záporné a kladné napětí pro vyjádření logické úrovně.

Vnitřní zapojení obvodu je na obrázku 16. Rozsah napájecího napětí je od 3 do 5,5 V s typickým odběrem kolem $300 \mu A$. Obvod obsahuje dva převodníky pro vysílání (DIN1 a DIN2) a dva pro přijímání (RIN1 a RIN2). Pokud je vstupní pin přijímače ve stavu vysoké impedance, na jeho výstupu je log. 1, což značí klidový stav. Teoretická rychlost jednoho kanálu je 250 kbps.

Pro vyšší výstupní napětí využívá obvod kondenzátory $C_1..C_4$ pro vytvoření nábojové pumpy [13].



Obrázek 16: Blokové schéma vnitřního zapojení obvodu MAX3232 podle [13].

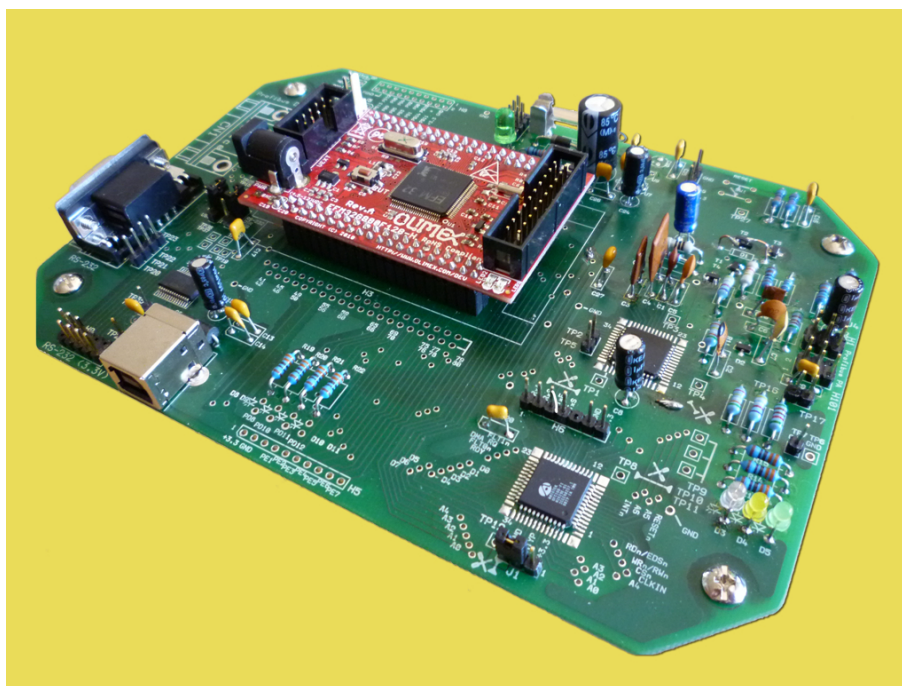
3.3.7 ADM-3078E

Obvod ADM-3078E od Analog Devices slouží pro převod obousměrné komunikace od MCU na poloduplexní podle standardu RS485. Umožňuje tedy aktivovat směr pro příjem nebo naopak směr pro vysílání na poloduplexní sběrnici RS485 [14].

3.4 Realizace komunikačního modulu

Po provedení návrhu elektroniky modulu byl proveden návrh desky plošných spojů (DPS). Samotný návrh DPS není součástí této práce. Soubory z návrhového systému Formica jsou pro úplnost přiloženy k práci. Je přidán obvod FT232 pro virtualizaci sériového portu na PC přes USB. Dále jsou přidány potřebné napěťové stabilizátory a jsou vyvedeny některé důležité měřící body pro snazší ladění komunikačního obvodu. Dále je z mikrokontroléru vyvedeno rozhraní pro LCD displej.

DPS byla následně osazena (obrázek 17) a oživena. Další ukázky modulu jsou v příloze A.



Obrázek 17: První verze osazené DPS komunikačního modulu.

Digitální část je napájena externím zdrojem. Napájecí napětí je regulováno na 3,3 V. Analogová část je podle doporučeného zapojení napájena ze sběrnice.

Programová realizace je popsána samostatně v další kapitole.

3.5 Softwarová realizace komunikačního modulu

Mikrokontrolér EFM32G880F128 na desce od *Olimexu* je programován *JTAG/SWD Emulátorem* s USB rozhraním od firmy *Segger*. Jeho *EDU* verze pro nekomerční projekty umožňuje užívat program *J-Link*, kterým lze programovat flash paměť mikrokontroléru a ladit program vkládáním breakpointů.

CPU je při programování přepnut do ladícího módu. Inicializuje stahování programu přes sériové rozhraní SWD pomocí bootladeru a zapisuje ho do paměti flash. K přepnutí do debug módu je potřeba, aby CPU byl několik vteřin po resetu aktivní. Jinak nelze tímto USB emulátorem flash paměť přeprogramovat (například při programovém přepnutí do EM4 nebo přepnutí na nerozběhnutý oscilátor krátký čas po resetu). Pokud dojde k takové situaci, lze flash paměť vymazat připojením mikrokontroléru na některý vývojový kit přímo od společnosti *Energy Micro* a použít jejich program *Simplicity Studio*.

Na webových stránkách *Energy Micro* (<http://www.energymicro.com/>) jsou také k dispozici soubory s definovanými adresami periferií a registrů mikrokontroléru. Dále jsou zde ke stažení "knihovny" s funkcemi pro obsluhu periferií a příklady pro práci s nimi. Knihovny obsahují i soubory pro standardní programové rozhraní *CMSIS* (*Cortex Microcontroller Software Interface Standard*) k procesorům *ARM-CortexM*.

Program je vyvíjen v prostředí *IAR Embedded Workbench*. Konkrétně ve verzi pro procesory ARM v6.30 s limitovanou velikostí kódu na 32 KB, která je volně ke stažení. Tento nástroj je kompatibilní s debuggerem *J-Link*, takže umožňuje přímo ve vývojovém prostředí programovat flash paměť nebo vkládat breakpointy. K tomu je nejprve nutné v nastavení projektu zvolit ovladač pro debugger *J-Link/J-Trace*. Dále musí být při každém přístupu k MCU přes *J-Link* vedle IAR také spuštěn program *J-Link Commander*.

3.5.1 Inicializace mikrokontroléru

Pro korektní práci mikrokontroléru by měla být první spuštěna funkce `CHIP_Init()`. Ta je dána výrobcem a nachází se v balíčku knihoven. Funkce slouží pro odstranění programových odlišností z výroby. Po této funkci následuje uživatelský program.

K maximálnímu výkonu mikrokontroléru je pro další výpočty přepnut zdroj hodinového signálu z interního RC oscilátoru s frekvencí kolem 14 MHz na 32 MHz externí krystalový oscilátor. Ten se nachází přímo na vývojovém kitu. Krystalový oscilátor je se svou vysokou přesností vhodný také pro asynchronní sériovou komunikaci, kde při větších rychlostech udržuje stále velmi přesný bit time.

Přepnutí zdroje hodin se provádí v několika fázích nastavováním jednotky *CMU (Clock Management Unit)*. V první fázi je potřeba zapnout externí oscilátor v příslušném řídicím registru. MCU obsahuje obvody pro kontrolu stability oscilátoru, takže je možné programově čekat na indikaci připravenosti oscilátoru. Poté je teprve možné přepnout zdroj hodin a je vhodné nepoužívaný RC oscilátor vypnout.

Dále jsou inicializovány ostatní používané periferní jednotky. Jejich konfigurace se nachází v následujících kapitolách.

Shrnutí kroků z této části:

1. `CHIP_init()` - Spuštění na doporučení výrobce pro odstranění výrobních odlišností.
2. Přepnutí zdroje hodin na externí krystalový oscilátor. Zapnutí oscilátoru, čekání na stabilizování krystalu, přepnutí zdroje hodin, vypnutí nepoužívaného oscilátoru.
3. Inicializace ostatních periférií.

3.5.2 UART

Pro sériovou asynchronní linku je použit *UART0* v lokalizaci 0, tzn. na pinech 6 a 7 *portu F*.

Do každé periferní jednotky je potřeba přivést hodinový signál pro jejich aktivaci. K tomu slouží funkce `CMU_ClockEnable(cmuClock_UART0, true)`. Ta má dva parametry, první je jednotka, která má být nastavena, a druhý je zda má být jednotka aktivována či deaktivována.

Následuje aktivace vstupního pinu *Rx* a výstupního pinu *Tx*. Opět k tomu připravenou

funkcí `GPIO_PinModeSet(gpioPortF, 6, gpioModePushPull, 0)` např. pro *Tx*. Vstupními parametry této funkce jsou označení portu, číslo pinu, mód a inicializační logická hodnota na pinu. Obdobně je nastaven signál *Rx* s tím, že pin je ve vstupním módu.

Dále je nutné nastavit rychlost komunikace. K tomu slouží registr *CLKDIV*, kde je odvozená rychlost dána výpočtem (odvození vzorce v dokumentaci mikrokontroléru):

$$256 * ((UART_PERCLK_FREQUENCY / (16 * UART_BAUDRATE)) - 1),$$

kde *UART_PERCLK_FREQUENCY* je rychlost hodin periferních jednotek (v tomto případě stejná jako frekvence CPU) a *UART_BAUDRATE* je potřebná rychlost sériového kanálu v bitech za vteřinu.

Pro bezchybnou komunikaci je zapotřebí nastavit strukturu rámce sériové linky. Ta je zvolena na 8 datových bitů, jeden stop bit a žádný paritní bit. Konfigurace sériového kanálu pro komunikující stranu musí odpovídat nastavení udávané tabulkou 5.

Port	název portu, ke kterému je kom. modul připojen
Rychlost	115 200 baudů
Počet datových bitů	8
Parita	žádná
Počet stop bitů	1

Tabulka 5: Parametry sériové linky komunikačního modulu

Nakonec je potřeba v řídicím registru *CMD* jednotky *UART0* povolit příjem a vysílání.

V jeho příslušném souboru jsou funkce pro odesílání i přijímání jak v blokujícím formátu, kdy CPU čeká na odeslání/přijmutí v blokující smyčce, tak i ve formátu řízení pomocí přerušování, kdy procesor může vykonávat jiné výpočty.

Ke komunikaci po sériové lince je upraven příklad z webových stránek *Energy Micro*. Podrobnější nastavení jednotky a implementace jednotlivých funkcí lze získat nahlédnutím do komentovaných zdrojových kódů.

Přehled a shrnutí kroků pro nastavení jednotky UART0:

1. `CMU_ClockEnable(cmuClock_UART0, true)` - Přivedení zdroje hodin do jednotky UART0.
2. `GPIO_PinModeSet(gpioPortF, 6, gpioModePushPull, 0)` - Nastavení výstupního signálu pro Tx.
3. `GPIO_PinModeSet(gpioPortF, 7, gpioModeInput, 0)` - Nastavení vstupního signálu pro Rx.
4. Nastavení rychlosti přenosu v registru *CLKDIV*.
5. Nastavení struktury rámce, konfigurace parametrů seriového kanálu.
6. Povolení příjmu a vysílání v registru *CMD*.
7. Jednotka UART0 je připravena pro příjem nebo vysílání.

3.5.3 Rozhraní externí sběrnice

Rozhraní externí sběrnice (*Extern Bus Interface - EBI*) umožňuje efektivně komunikovat s externím zařízením po paralelní sběrnici, které je namapováno do adresního prostoru MCU.

Ke konfiguraci je potřeba, stejně jak je uvedeno výše, aktivovat hodinový signál do jednotky *EBI*.

Dále pak nastavit jednotlivé piny. Adresní piny *A0..A7* na *portu A*. Datové piny *D0..D7* na *portu E*. Výběr obvodu signálem *chip select - CS* je aktivován automaticky podle adresy, na kterou se přistupuje. V našem případě je zvolen úplně první segment adresního prostoru *EBI* a to od adresy `0x80000000`, což odpovídá banku 0 a je tedy aktivován vždy *CS0*.

Zbývají nastavit signály *ARDY* pro signalizaci připravenosti zařízení na další příkaz. Signál *Wen* pro zápisový cyklus a signál *Ren* pro čtecí cyklus.

V inicializační části *EBI* jsou nastavovány všechny potřebné piny pro komunikaci s mikrokontrolérem, jejich význam je vysvětlen v následující kapitole, protože přímo nesouvisí s jednotkou *EBI*.

Po nakonfigurování vstupně/výstupních pinů zbývá nastavit samotný režim sběrnice. Protože rozhraní řadiče *UFC100* používá 8 bitovou sběrnici pro data a maximálně 7 bitovou sběrnici pro adresaci, je použit 8 bitový režim jednotky. Dále, jak už bylo zmíněno, je aktivován *bank 0* a *CS0*, čímž je zvolena bázová adresa *EBI*.

Následuje nastavení časových intervalů pro sled jednotlivých signálů. Jedná se o časy měřené v počtu taktů pro přednastavení adresy před a po aktivaci cyklu. Dále přednastavení adresy a dat, doba aktivního signálu pro čtení *Wen* a podržení stabilní adresy a dat po skončení signálu *Wen* u zápisového cyklu. Obdobně i u čtecího cyklu. MCU dokáže měnit signál při 32 MHz zhruba každých 30 ns. Jelikož časové intervaly řadiče *UFC100* jsou v jednotkách nanosekund, můžou být ponechány minimální hodnoty.

Pro bezchybnou komunikaci mezi kontroléry je aktivováno čekání na připravenost kontroléru *UFC100* signálem *ARDY*. Ten je opět součástí jednotky *EBI*, takže jeho povolením se s ním jednotka vypořádá automaticky.

Tímto je externí sběrnice nakonfigurována. Nutno ještě poznamenat, že je zvolen 8 bitový mód, takže při pokusu zapsání delších dat reaguje procesor chybou *HardFault*.

Pro přehlednější program je vytvořeno makro

```
#define UFC(ADDR) (*(uint8_t*)(UFC_BASE_ADDRESS + ADDR))
```

Takto lze provádět čtení nebo zápis na libovolnou adresu řadiče *UFC100* dosazením této adresy za proměnnou *ADDR*. Adresa je poté promítnuta na adresní vodiče *EBI*.

Tímto lze komunikovat s řadičem *UFC100* rychlostí, odpovídající přesunu dat v adresním prostoru MCU.

Přesnější popis implementace, včetně aktivních logických hodnot jednotlivých signálů opět poskytnou komentované zdrojové kódy.

Shrnutí kroků pro nastavení externí sběrnice:

1. `CMU_ClockEnable(cmuClock_EBI, true)` - Přivedení zdroje hodin do jednotky EBI.
2. Nastavení adresních a datových pinů. Nastavení řídicích pinů CS0, ARDY, Wen a Ren funkcí `GPIO_PinModeSet`.
3. Konfigurace 8 bitové sběrnice (8 bitů data + 8 bitů adresa).
4. Povolení adresního banku 0 a CS0.
5. Nastavení časování signálů.
6. Povolení čekání na signál ARDY.
7. Jednotka je připravena pro čtení a zápis od adresy `0x80000000`.

3.5.4 Nastavení kontroléru UFC100

Kromě signálů pro paralelní komunikaci jsou MCU a UFC100 propojeny dalšími signály.

Prvním z nich je přerušení (*Int*). Tímto signálem si kontrolér UFC100 vynucuje aktivitu od MCU. Kontrolér signalizuje přerušení pro několik zdrojů (interní časovač, příjem rámce a další). Všechna ale signalizuje jedním signálem, o jaký zdroj přerušení se jedná, musí být přečteno z příslušného registru. Po nastavení pinu na vstupní, musí být nastaveno, že se jedná o signál přerušení. To se provede těmito funkcemi:

```
NVIC_EnableIRQ(GPIO_EVENT_IRQn);  
GPIO_IntConfig(gpioPortA, 10, false, true, true);
```

Obě funkce jsou opět knihovní. První funkce povoluje přerušení pro IO porty. Druhou funkcí je nastavení přerušení u příslušného pinu. Parametry jsou port, číslo pinu, reakce na náběžnou hranu, reakce na sestupnou hranu a povolení tohoto přerušení. Je tedy vidět to, že přerušení reaguje na sestupnou hranu, protože je aktivní v logické 0. Nutno ještě

zmínit, že všechny IO piny sdílejí jednu přerušovací rutinu (*IRQHandler*), takže pokud by bylo takto nastaveno více pinů, musí být v přerušovací rutině zjištěno, o který pin se jedná.

K důležitým signálům patří i zdroj hodinového signálu. Příslušný pin je nastaven jako výstupní. Jednotka *CMU* umožňuje také nasměrovat některý zdroj hodin na výstupní pin, čehož bylo využito. Přesměrován je tedy rychlý krystalový oscilátor s 32 MHz, který taktuje i celý MCU.

Zbývá ještě nastavit signál pro reset. Signály *FLTx* nebyly použity a měly být propojeny s obvodem *MAU*, kde signalizují jeho chyby. Žádost o přenos DMA nebyl také použit, protože se nepodařilo při testování jeho aktivitu vyvolat.

Shrnutí kroků pro konfiguraci rozhraní s kontrolérem UFC100:

1. Konfigurace sběrnice EBI (viz předchozí kapitola).
2. Nastavení dalších pinů jako jsou: Int, DMAreq, Reset a zdroj hodin. K tomu slouží funkce `GPIO_PinModeSet`.
3. Nastavení přerušovacího signálu:
`NVIC_EnableIRQ(GPIO_EVEN_IRQn)` - Povolení přerušení pro vstupně výstupní piny.
`GPIO_IntConfig(gpioPortA, 10, false, true, true)` - Nastavení přerušovacího pinu na sestupnou hranu.
4. Přesměrování krystalového oscilátoru jako zdroje hodin pro UFC100.
5. Nyní je připravena komunikace s kontrolérem UFC100, který je namapován do adresního prostoru MCU.

Konfigurace UFC100

Po nakonfigurování potřebných portů je možné nastavit samotný řadič UFC100. Řadič obsahuje několik desítek 8 bitových registrů. Ty se dělí na řídicí, informační a datové (FIFO). Řídicími je řadič nastavován, informační slouží pro indikování stavu nějaké

jednotky řadiče a jsou nastavovány samotnou jednotkou. Pro přehlednější kód jsou tyto registry a jejich políčka nadefinovány pod jejich názvem do hlavičkového souboru.

Příklad definice registru UFC100 pro programový reset a verzi řadiče na adrese *0x00*:

```
/** Register base address */
#define    UfcVer    0x00
/** Bit fields for UfcVer */
#define    RstSoft    (0x01 << 7)
#define    Ver_MASK    0x3F
#define    Ver_F1    0x10
```

Použití pak na příkladu softwarového resetu vypadá takto:

```
UFC(UfcVer) = RstSoft;
```

Vzhledem k minimální dokumentaci řadiče UFC100 pro základní mód, je zvolen rozšiřující mód *Enhanced Mode (EM)*, který také dovoluje nejlépe využít všech funkcí tohoto řadiče.

Pro **přepnutí do EM** je zapotřebí do řadiče zapsat speciální odemykající sekvenci, kterou je chráněn před neoprávněnou změnou módu. Mód je nastavován v registru *UfcType*.

1. Hardwarový nebo programový reset.
2. Zapsání hodnoty 0x13 do registru UfcVer na adrese (0x00).
3. Zapsání hodnoty 0x2C do registru UfcVer (0x00).
4. Zapsání hodnoty 0x80 (nejvyšší bit značí rozšiřující mód, na ostatních v tuto chvíli nezáleží) do registru UfcType (0x0F).
5. Zápis ostatních řídicích bitů do registru UfcType (0x0F), v našem případě volba Basic Device pro FF.

Po této sekvenci příkazů se řadič nachází v módu EM. Protože je v tomto módu potřeba zapisovat na adresy delší než 5 bitů, je v řadiči automaticky zapnut režim stránkování, který umožňuje používat 5 bitovou adresní sběrnici a zbylé dva bity zapisovat do speciálního registru. Jelikož se používá 7 bitovou sběrnice, je tento režim vypnut v registru *UfcOptn*, vynulováním bitu *PageEn*. Zároveň jsou v tomto registru přepnuty *MAU Control* a *Bus Monitor* také do módu EM.

Nastavení rozhraní s MAU

Následuje nastavení rozhraní s obvodem MAU. Zde je v registru *MauCntl* nastaven full-duplexní mód, který umožňuje zpětně přijímat vlastní odeslané rámce, což je důležité pro monitorování komunikace na sběrnici. Režim vysílače a přijímače je ponechán v základním módu, který se shoduje s aktivitou signálů s fyzickým obvodem MAU. V registru *PhlPara* lze také nastavit počet bytů hlavičky (preamble) při vysílání a mezeru mezi rámci v počtu bitů. Jejich nastavení ale závisí na konfiguraci sběrnice, ke které je zařízení připojené.

Nastavení interního generátoru hodin

Pro generování a odvozování hodin pro Manchester kód je potřeba nastavit jejich správnou frekvenci. Signálu *ClkUfcInt* předchází dvě děličky ze vstupního hodinového signálu *CLKIN* od MCU. Tento vnitřní signál má mít frekvenci 500 kHz a jsou od něj odvozeny hodiny pro ostatní jednotky řadiče. Této frekvence docílíme nastavením správných dělicích poměrů v registru *ClkGenU*. Pokud je vstupní frekvence 32 MHz, lze například nastavit první dělicí poměr ku 16 (nastavením na 15, protože se zde automaticky přičítá jedna), a druhou děličku na 4. Zároveň musí být generátor hodin povolen bitem *UfcClkEn*. Na výstupech *CLKOx* lze ověřit správné nastavení frekvence vyvedením vnitřních hodin na některý tento pin v registru *ClkOutSel*.

Odesílání rámců

Nastavení módu vysílače v registru *TxmMode* lze nechat v základním nastavení, ve kterém jsou vypnuty kontroly časových limitů a není aktivní žádné vyvolávání přerušení při

dokončení odesílání.

Před odesláním každého rámce musí být provedeny tyto kroky:

1. V registru *TxmCmd* musejí být vymazány příznaky posledního odesílání a musí být zvolen režim přenosu dat z MCU do FIFO fronty řadiče. Zároveň může být rovnou nastaven příznak pro start odesílání.
2. Do registrů *TxmCount* je nutné před plněním FIFO zapsat velikost dat.
3. Do FIFO je zapisován znak po znaku přes registr *TxmFIFO*, dokud není *TxmCount* nulový.

Nastavením příznaku pro start odesílání před naplněním fronty lze odeslat větší blok dat, než je velikost FIFO fronty (128 znaků). Každým zapsáním znaku do FIFO se dekrementuje registr *TxmCount*. V registru *TxmStatA* je pak indikováno vynulování čítacího registru. Ten také obsahuje další informace o stavu fronty, proto je vhodné ho průběžně během plnění fronty číst.

Mód přenosu do FIFO fronty je zvolen základní, připravenost přijmutí dalšího znaku se řídí přímo *EBI* jednotkou, signálem *ARDY*. Další módy jsou přerušením a DMA přenosem, který by byl vhodnější z hlediska vytíženosti CPU. Bohužel se DMA přenos nepodařilo zprovoznit.

Po odeslání se automaticky nastaví v několika registrech příznaky o úspěšnosti vyslání, které je možné kontrolovat.

Příjem rámců

Nastavením přijímače v registru *RcvMode* je aktivováno generování přerušení při dokončení příjmu rámce. Generování přerušení je nutné ještě globálně povolit v registru *IntCmd*.

Pro experimentální účely je také vypnuto veškeré filtrování rámců v registru *RcvFlt-Mode*. To zaručuje, že bude akceptován každý rámeček, který bude řadičem identifikován znaky začátku a konce rámce.

Kroky pro nastavení přijímače jsou obdobné jako u vysílače:

1. Pro aktivaci přijímače je potřeba nastavit registr *RcvCmd*. Zde musejí být vynulovány příznaky z posledního příjmu, lze zde povolit i zápis přijatého kontrolního součtu do FIFO, následuje volba přenosu z FIFO jako u vysílače a povolení jednotky přijímače.
2. Nastavení velikosti dostupné fronty MCU pro příjem (kolik znaků je schopen přijmout) v registrech *RcvCount*.
3. Po obdržení přerušování od řadiče, signalizující příjem znaku, může být započato čtení přijatých dat ve FIFO frontě z registru *RcvFIFO*. Opět je zde registr *RcvStatA*, který obsahuje informace o stavu fronty. Například, že je prázdná a přečtená data nejsou platná.
4. Po přečtení dat se musí opět vynulovat příznaky o proběhlém příjmu (registry *RcvCmd* a *RcvCount*). V opačném případě další rámeček nemusí být přijat.

Přesnější informace o nastavení lze vyčíst z dokumentace k řadiči UFC100 a ze zdrojových kódů.

3.5.5 Možnosti přístupu k implementaci řídicího algoritmu

Aplikace na mikrokontroléru (MCU) vykonává funkci základního zařízení pro průmyslovou sběrnici FOUNDATION Fieldbus (FF). Pro snazší ladění umožňuje monitorování komunikace na FF a tuto komunikaci lze zobrazovat na terminálu sériového portu osobního počítače (PC). Řídicí úloha se dá dekomponovat na:

- Příjem rámce ze sběrnice FF.
- Zpracování rámce a vygenerování případné odpovědi.
- Odeslání kopie rámce do PC pro zobrazení.
- Odeslání případné odpovědi na sběrnici FF.

K celé aplikaci lze přistupovat několika způsoby. Prvním je klasický přístup programování, kdy jsou jednotlivé části obsluhy prováděny v jedné hlavní nekonečné smyčce. Tento způsob nejlépe využívá výpočetní zdroje. Ale vzhledem k rozsáhlosti kódu, jeho přehlednější správu, popřípadě jeho pozdější rozšiřitelnost je zvolen druhý přístup, a to realizace jednotlivých úloh v operačním systému reálného času. Tato implementace je obsahem následující kapitoly.

3.5.6 Implementace úloh ve FreeRTOS

FreeRTOS je volně dostupný operační systém reálného času (Real Time Operating System - RTOS) šiřitelný pod licencí GNU GPL (General Public License).

Aktuální verze je V7.1.0 a je ke stažení na webových stránkách projektu FreeRTOS (www.freertos.org). Součástí balíčku jsou zdrojové soubory, příklady pro řadu podporovaných MCU a licenční informace.

FreeRTOS implementuje preemptivní plánovač, prostředky pro synchronizaci a komunikaci mezi jednotlivými úlohami. Je také postaven na kooperaci s přerušovacími rutinami.

Mezi dostupnými portacemi je i port pro jádro Cortex-M3 za použití překladače IAR. Složka obsahuje tyto 3 soubory: *port.c*, *portasm.s*, *portmacro.h*. V knihovním souboru *startup_efm32.s* od Energy Micro poté stačí upravit namapování příslušných rutin přerušování (*vPortSVCHandler*, *xPortPendSVHandler*, *xPortSysTickHandler*) a FreeRTOS je připraven k použití.

Aplikace ve FreeRTOS se definuje jednotlivými úlohami. Těm jsou konfigurovatelnou strategií plánovače poskytovány výpočetní zdroje MCU. Úlohy jsou nezávislé úseky kódu, které jsou v rámci každé úlohy vykonávány v nekonečné smyčce. Úlohy se můžou nacházet ve stavech:

- Running - úloha má aktuálně přidělený procesor a běží.
- Ready - úloha je připravena ke spuštění. Na CPU je ale vykonávána jiná úloha s větší nebo stejnou prioritou.

- Blocked - úloha je v blokujícím stavu a čeká na vnější událost.
- Suspended - úloha je vyjmuta z plánovacího mechanismu.

FreeRTOS je možné konfigurovat pomocí definic obvykle v souboru *FreeRTOSConfig.h*. Mezi nejdůležitější definice patří:

```
#define configCPU_CLOCK_HZ 32000000UL
#define configTICK_RATE_HZ ((portTickType) 1000)
#define configUSE_PREEMPTION 1
```

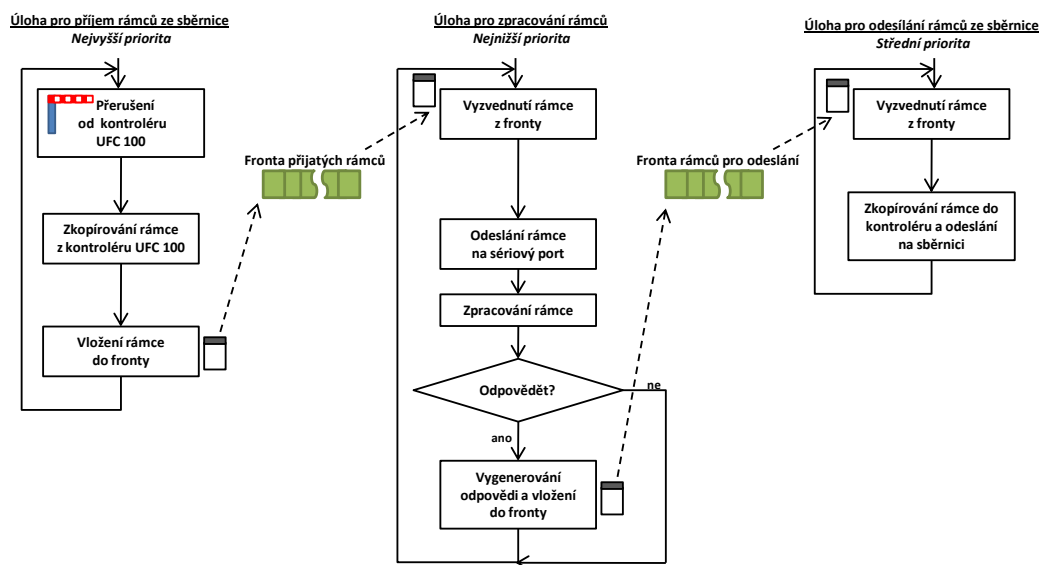
Definice *configCPU_CLOCK_HZ* slouží pro informaci o rychlosti taktu a spolu s *configTICK_RATE_HZ* určují periodu pro aktivování plánovače. V tomto případě je kontrola přepřánování aktivována každou 1 ms. Tento čas je dostatečně krátký pro co nejrychlejší obsluhu přijatého rámce (při rychlosti 31,25 kHz odpovídá jeden byte na sběrnici kolem 250 μs) a zároveň dostatečně dlouhý, aby časté spouštění plánovače neplýtvalo systémovými prostředky. Definice *configUSE_PREEMPTION* umožňuje volit mezi preemptivním a nepreemptivním plánováním, kdy se běžící úloha musí vzdát CPU sama.

Řídící aplikace komunikačního modulu

Pro řídicí aplikaci jsou vytvořeny 3 úlohy. Jednotlivé úlohy jsou ilustrovány diagramem na obrázku 18. Všechny úlohy jsou implementovány v nekonečné smyčce.

První úloha s nejvyšší prioritou se stará o příjem rámce z radiče UFC100. Pokud radič přijme rámeček, signalizuje to přerušáním do MCU, v jehož obsluze odblokuje synchronizační semafor. Ten povolí pokračování v této úloze, kde jsou přijatá data zkopírována z radiče do MCU a vložena do fronty přijatých rámečků. Pokud je tedy vyvoláno přerušování, vždy minimálně se zpožděním 1 ms tato úloha provede jednu otočku.

Druhá úloha s nejnižší prioritou vybírá z fronty přijaté rámce. Tyto rámce přeposílá na rozhraní UART a dále zpracovává. Podle typu zprávy zde může vytvořit odpověď, kterou vloží do jiné fronty pro odeslání rámečků.



Obrázek 18: Úlohy pro řízení komunikačního modulu.

V poslední úloze, která má střední prioritu, se vybírají rámce z fronty pro odeslání a kopírují se do řadiče UFC100, který je odešle na sběrnici.

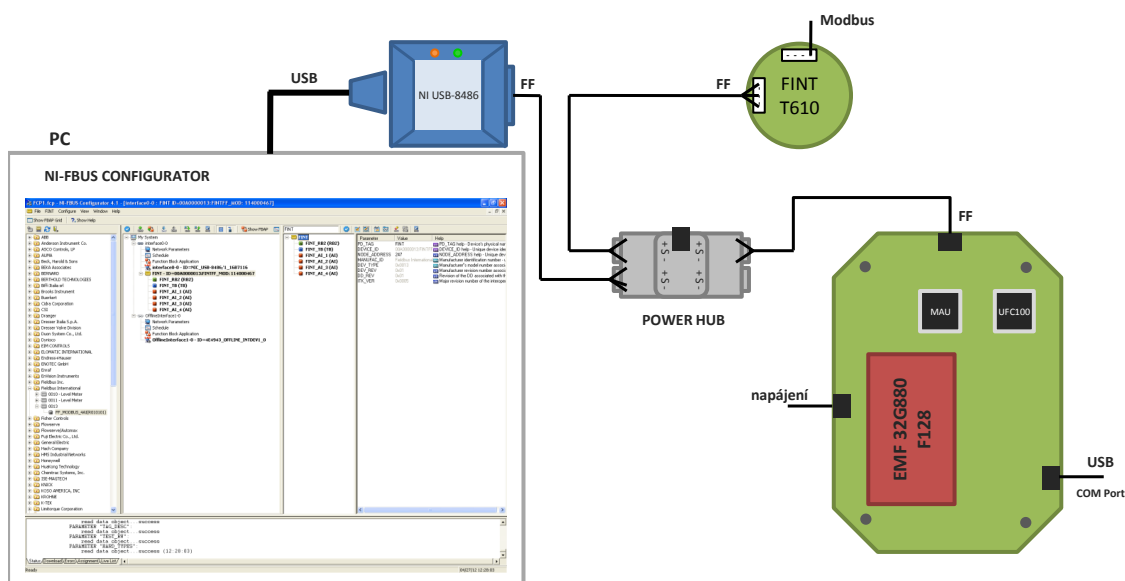
Priority jsou tedy postaveny tak, aby vždy byl přijat rámec z UFC100. Následuje zpracování přijatého rámce a případné vygenerování odpovědi. Tato odpověď je vždy maximálně s prodlevou 1 ms neprodleně odeslána na sběrnici.

3.6 Testování na segmentu průmyslové sběrnice FF-H1

Pro testování komunikačního modulu s reálnými zařízeními sběrnice FOUNDATION Fieldbus (FF) H1 bylo k dispozici:

- *National Instruments (NI) USB-8486*: Konfigurátor segmentu sběrnice FF-H1 přes rozhraní USB v programu NI-FBUS Communications Manager.
- *Fieldbus International (FINT) T610 Modbus to Foundation*: Převodník FF-H1 na modbus se čtyřmi AI bloky.
- *Realcom Inc. Power Hub FCS-PH-PL*: Napájení sběrnice 19 V se čtyřmi konektory s celkovým maximálním odběrem 200 mA.

Zapojení celého segmentu ilustruje obrázek 19 a příloha B. U konfigurátoru USB-8486 je nutné propojit zemnicí vodič se záporným pólem sběrnice, jinak je signál na sběrnici zarušený a konfigurátor vysílá s poloviční amplitudou. Konfigurátor je v základu nastaven jako LAS, takže řídí přenos na sběrnici a rozesílá tokeny. Při správném propojení je v programu NI-FBUS automaticky vyhledán převodník FINT T610 a jsou načteny jeho uživatelské bloky.



Obrázek 19: Zapojení segmentu FF-H1 pro testování komunikačního modulu.

Komunikační modul zachytává veškerou komunikaci mezi konfigurátorem USB-8486 a převodníkem FINT T610.

Pro možnost odpovědi komunikačního modulu na některý typ rámce od konfigurátoru USB-8486 je potřeba přenastavit parametry segmentu sběrnice FF-H1 v programu NI-FBUS u položky *Network Parameters*. USB-8486 vysílá při neplánované komunikaci opakovaně Pass Token (PT) pro zařízení v Live Listu a Probe Node (PN) se střídajícími se adresami pro vyhledávání nových uzlů. Rychlost tohoto opakování závisí na parametru *Slot Time*. Aby mohl komunikační modul odpovědět, je třeba nastavit větší *Slot Time* (15 ms, testováno pro 2 zařízení na segmentu). Tím se vytvoří na sběrnici širší mezera,

kdy není obsazena sběrnice. V tomto nastavení je komunikační model schopen vyslat na sběrnici odpověď do 1 ms od výzvy. Tento čas odpovědi je podobný i u komunikace mezi konfigurátorem a převodníkem FINT T610.

Ukázka programu NI-FBUS je v příloze D. Podrobnější informace o zapojení konfigurátoru USB-8486 naleznete v [15]. Uživatelská příručka k programu NI-FBUS je k nalezení v [16]. Stručná uživatelská příručka k převodníku T610 je v [17].

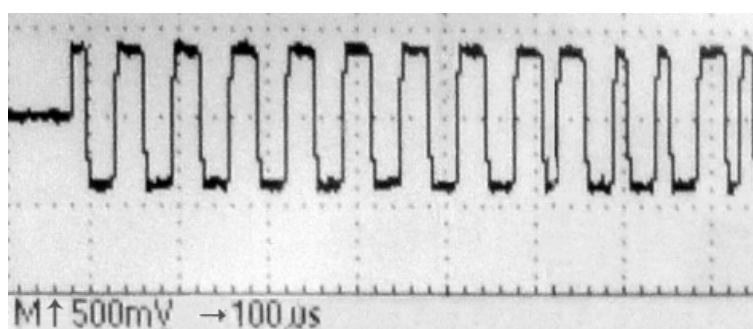
3.7 Naměřené parametry komunikačního obvodu

V tabulce 6 jsou zjištěné a nastavené parametry komunikačního modulu.

Frekvence CPU	32 MHz
Odběr digitální části	typ. 42 mA
Odběr analogové části	typ. 10 mA
Opakovatelnost vyslání rámce (pro FF)	přibližně 25 ms
Amplituda signálu (pro FF)	1 V _{pp}

Tabulka 6: Parametry komunikačního modulu

Na obrázku 20 je ukázka tvaru signálu s kódováním Manchester vysílaným komunikačním modulem.



Obrázek 20: Ukázka vysílaného signálu komunikačního modulu.

3.8 Programová realizace monitorovacího programu

Pro zobrazení rámců přenášených na segmentu sběrnice je vytvořen program *Packet Analyzer* v programovacím jazyce *C#*. Program přijímá a zobrazuje typ a obsah zachycených rámců komunikačním modulem.

Uživatelská příručka programu se nachází v příloze C.

Vygenerovaná programová dokumentace v nástroji *SandCastle* a okomentované zdrojové kódy se nachází na datovém médiu přiloženém k diplomové práci.

4 Závěr

Práce se soustředí na problematiku vybraných průmyslových sběrnic, zejména pak na sběrnici FOUNDATION Fieldbus. Cílem práce je návrh komunikačního modulu pro tuto sběrnici a kompletní realizace funkčního zařízení, které je schopné komunikace po sběrnici. Tento modul byl navržen a implementován.

Programová realizace komunikačního modulu se dá rozdělit na dvě části. První část zahrnuje ovládání periférií a řadičů. Druhá část programově realizuje komunikační protokol sběrnice.

Po odstranění několika drobných návrhových chyb, byly úspěšně zprovozněny veškeré potřebné periferní jednotky, včetně řadiče pro vysílání a příjem po průmyslové sběrnici.

Realizace komunikačního protokolu na mikrokontroléru EFM32G880 je rozpracována s využitím operačního systému reálného času FreeRTOS. Mikrokontrolér dokáže zpracovat přijatý rámec a vygenerovat na něj patřičnou odpověď. Celý projekt je koncipován tak, aby poskytoval přehledné rozhraní pro rozšíření reakce na všechny typy rámců.

Připojením komunikačního modulu na vytvořený segment sběrnice byla ověřena jeho funkčnost. Pro experimentální segment jsou k dispozici konfigurátor NI USB-8486 s programem NI-FBUS a převodník FINT T610. Na segmentu byla ověřena schopnost modulu reagovat na přijatý rámec a odposlouchávat komunikaci mezi ostatními zařízeními sdílející tento segment. Takto odposlechnutá komunikace je vizualizována na osobním počítači, který je propojen s modulem sériovou linkou. K vizualizaci je vytvořen program, který dekoduje typ rámce a umožňuje zobrazit jeho obsah. Analyzováním komunikace může být usnadněna následná realizace komunikačního protokolu v modulu.

Takto realizovaný komunikační modul bude použit pro vývoj jmenovaných aplikací a zařízení, která jsou použita pro experimentální segment.

Přehled zkratek

ADC	Analog Digital Converter
AES	Advanced Encryption Standard
CAN	Controller Area Network
CIP	Common Industrial Protocol
CPU	Central Processing Unit
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DAP	Debug Access Port
DD	Device Description
DMA	Direct Memory Access
DP	Decentralized Peripherals
EBI	External Bus Interface
ETM	Embedded Trace Macrocell
FAS	Fieldbus Access Sublayer
FCS	Frame Check Sequence
FF	Foundation Fieldbus
FIP	Factory Instrumentation Protocol
FMS	Fieldbus Message Specification
HART	Highway Addressable Remote Transducer
HSE	High Speed Ethernet
IEC	International Electrotechnical Commission
ISA	The International Society of Automation
JTAG/SWD	Joint Test Action Group/Serial Wire Debug
LAS	Link Active Scheduler
LM	Link Master

MAU	Media Access Unit
MCU	Microcontroller
MPU	Memory Protection Unit
NVIC	Nested Vectored Interrupt Controller
OD	Object Description Object Dictionary
ODVA	Open DeviceNet Vendors Association
OSI	Open Systems Interconnection
PA	Process Automation
PCI	Protocol Control Information
PDU	Protocol Data Unit
PID	Proportional–Integral–Derivative
RAM	Random Access Memory
RTC	Real Time Clock
SDU	Service Data Unit
UART	Universal Asynchronous Receiver/Transmitter
VCR	Virtual Communications Relationship
VFD	Virtual Fieldbus Device
WIC	Wakeup Interrupt Controller

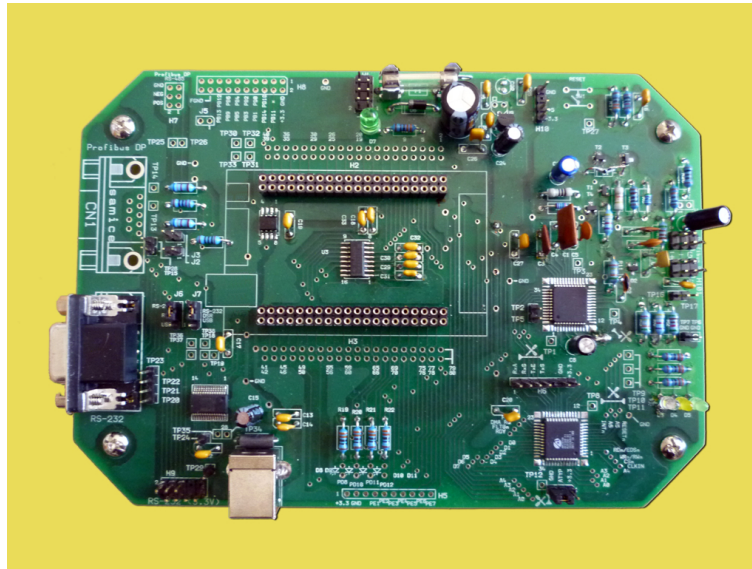
Literatura

- [1] Foundation Fieldbus Overview, National Instruments, 2007
- [2] FOUNDATION Fieldbus Book - A Tutorial, Yokogawa Electric Corporation, Japan, 2nd Edition, 2003
- [3] FOUNDATION Fieldbus, Part 4 Communication, SAMSON AG, 2000. [online] [cit. 27. 2. 2012]
<http://www.samson.de/pdf_en/1454en.pdf>
- [4] FOUNDATION Fieldbus Technical Overview, David A. Glanzer, FD-043 Revision 3.0
- [5] Dr. N. P. Mahalik: Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control, 2003, ISBN: 3-540-40183-0
- [6] Ian Verhappen, Augusto Pereira: Foundation Fieldbus, 3rd Revised edition, 2009, ISBN: 978-3540401834
- [7] An Introduction to the ARM Cortex-M3 Processor [online] [cit. 15. 3. 2012]
<http://cdn.energymicro.com/dl/pdf/arm_introtocortex-m3.pdf>
- [8] EFM32G Reference Manual [online] [cit. 15. 3. 2012]
<http://cdn.energymicro.com/dl/devices/pdf/d0001_efm32g_reference_manual.pdf>
- [9] EM-32G210F128-H development board, Users Manual [online] [cit. 15. 3. 2012]
<<http://www.olimex.com/dev/ARM/EM/EM-32G210F128/EM-32G210F128-H.pdf>>
- [10] Unified Fieldbus Controller, UFC100 User's Manual
- [11] Datasheet AMIS-492x0 Fieldbus MAU [online] [cit. 29. 3. 2012]
<http://www.onsemi.com/pub_link/Collateral/492X0-D.PDF>

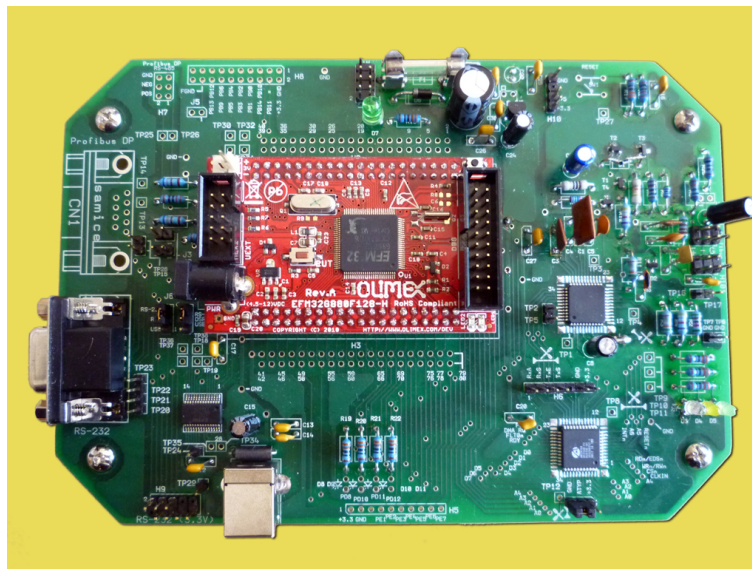
- [12] Datasheet FT232RL, FTDI Chip [online] [cit. 29. 3. 2012]
<http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf>
- [13] Datasheet MAX3232 [online] [cit. 29. 3. 2012]
<<http://datasheets.maxim-ic.com/en/ds/MAX3222-MAX3241.pdf>>
- [14] Datasheet ADM-3078E [online] [cit. 29. 3. 2012]
<http://www.analog.com/static/imported-files/data_sheets/ADM3070E_3071E_3072E_3073E_3074E_3075E_3076E_3077E_3078E.pdf>
- [15] FOUNDATION Fieldbus, NI-FBUS Hardware and Software User Manual
- [16] FOUNDATION Fieldbus, NI-FBUS Configurator User Manual
- [17] J. Norendal: The MODbus to FF build-in, the T610, Reference Manual

Přílohy

Příloha A - Fotodokumentace komunikačního modulu

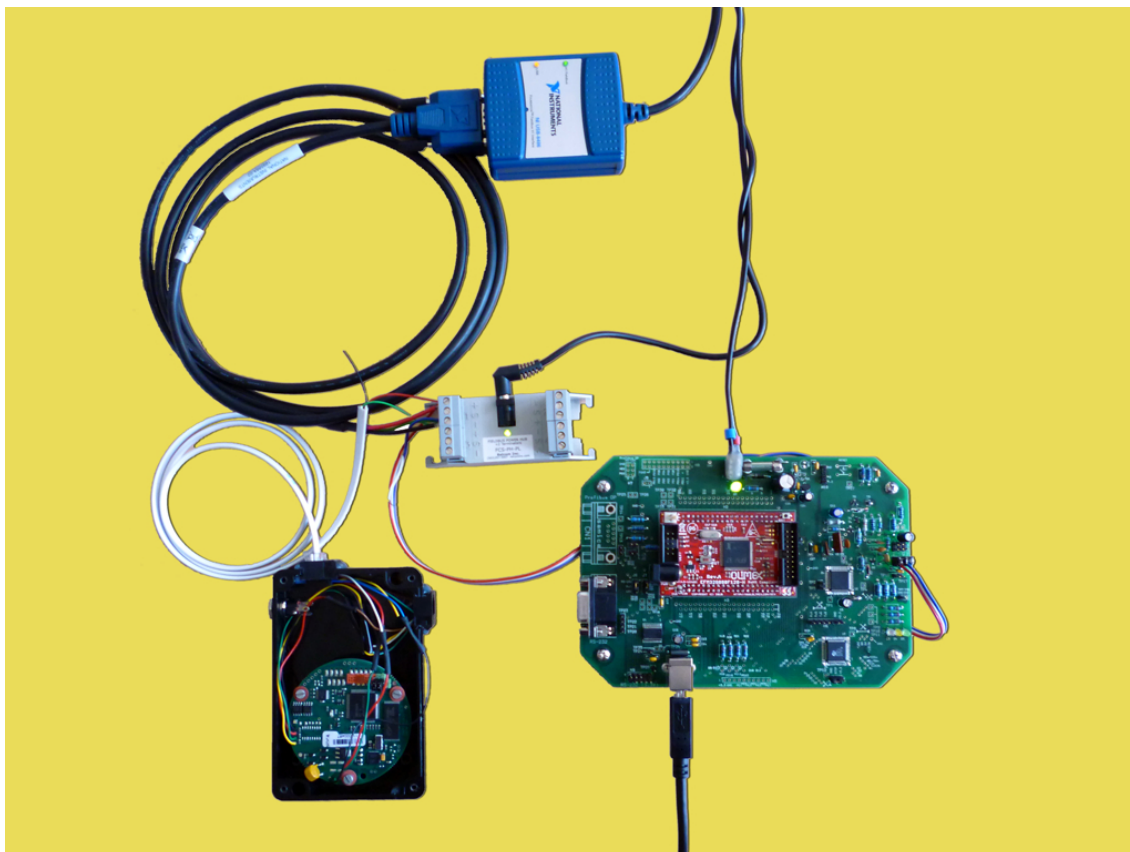


Obrázek 21: Komunikační modul.



Obrázek 22: Komunikační modul s vsazeným mikrokontrolérem.

Příloha B - Fotodokumentace zapojení zařízení segmentu sběrnice FF

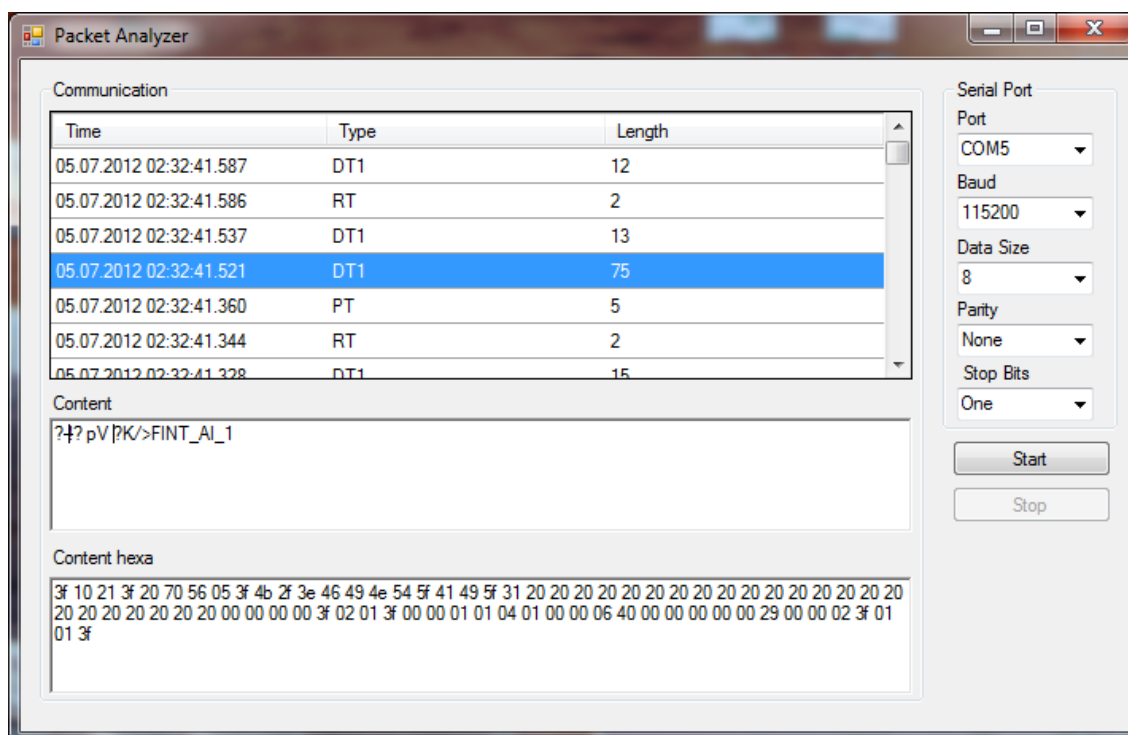


Obrázek 23: Zařízení použité pro experimentální segment sběrnice FF.

Příloha C - Uživatelská příručka monitorovacího programu Packet Analyzer

Program je určen k zobrazení zachycené komunikace realizovaným komunikačním modulem. Je kompilován a testován na platformě Windows.

Uživatelské rozhraní programu *PacketAnalyzer* ukazuje obrázek 24.



Obrázek 24: Okno programu Packet Analyzer.

Oblast *Serial Port* obsahuje nastavení sériové linky. Ta má konfiguraci podle komunikačního modulu, která je v tabulce 7.

Port	název portu, ke kterému je kom. modul připojen
Baud	115 200 baudů
Data Size	8 datových bitů
Parity	žádná (none)
Stop Bits	jeden (one)

Tabulka 7: Parametry sériové linky komunikačního modulu

Po nastavení sériového portu může být spuštěno zachytávání rámců tlačítkem *Start*.

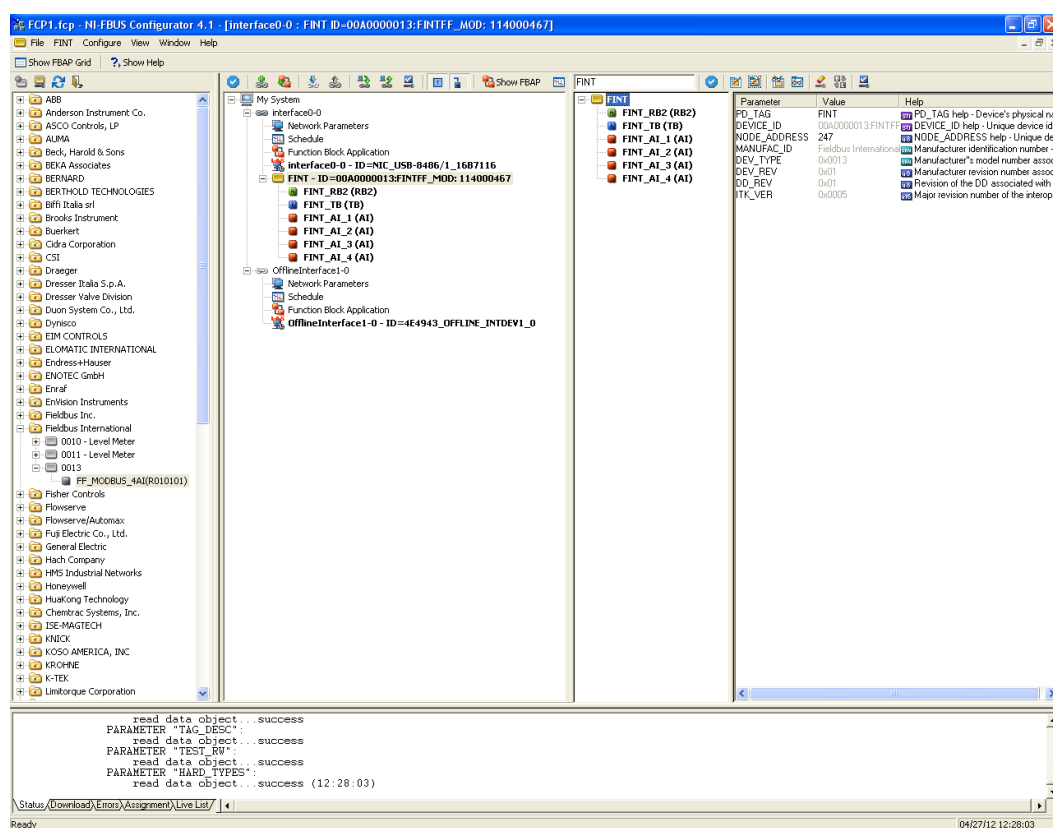
O jejich interpretaci se stará oblast *Communication*. V tabulce jsou zachycené rámce vypisovány. *Time* je čas, kdy byl přijatý rámeček zpracován, *Type* je typ rámce podle řadiče UFC100 a *Length* je délka celého rámce.

Po kliknutí na řádku tabulky se zobrazí v okně *Content* obsah rámce v textové podobě. Okno *Content hexa* zobrazuje obsah rámce v hexadecimálním tvaru jednotlivých bytů.

Pro ukončení zachytávání slouží tlačítko *Stop*.

Příloha D - Ukázka programu NI-FBUS pro konfiguraci sběrnice FF-H1

Obrázek 25 ilustruje prostředí programu NI-FBUS. V levém sloupci je seznam všech podporovaných zařízení. Další sloupec obsahuje segment sběrnice s konfigurátorem USB-8486 a převodníkem FINT T610 mezi FF-H1 a sběrnici Modbus. V prostředním sloupci je aktivováno okno s výběrem z konfiguračního stromu druhého sloupce. Sloupec vpravo obsahuje nápovědu.



Obrázek 25: Ukázka programu NI-FBUS s konfigurátorem USB-8486 a převodníkem FINT T610.

Příloha E - Errata

Zde jsou popsány zjištěné chyby při realizaci komunikačního modulu.

- 5 V napájení analogové části MAU není třeba, v tomto zapojení je napájena ze sběrnice.
- Zenerova dioda 5,1 V BZX84 (ve schématu ozn. D1) je vyráběna v SMD pouzdře, což se neshoduje s pouzdrem ve schématu.
- SMD tranzistory v analogové části mají opačné vyvedení báze a emitoru z pouzdra, než je ve schématu.
- Sériově zapojené diody v pouzdře SMD (ve schématu ozn. D2) v analogové části mají prohozené vývody anody a katody.