Masarykova univerzita
Fakulta informatiky

Návrh a implementace nového protokolu sběrnice MTBbus

Diplomová práce

Jan Horáček

Brno, jaro 2021

Masarykova univerzita
Fakulta informatiky

Návrh a implementace nového protokolu sběrnice MTBbus

Diplomová práce

Jan Horáček

Brno, jaro 2021

Na tomto místě se v tištěné práci nachází oficiální podepsané zadání práce a prohlášení autora školního díla.

Prohlášení

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Jan Horáček

Vedoucí práce: RNDr. Zdeněk Matěj, Ph.D.

Poděkování

Děkuji především všem, kteří ve mně zažehli, podporovali a podporují můj velký koníček – řízení modelové železnice. Ať už se jedná o tátu Miroslava, přítelkyni Verču, členy brněnského modelářského klubu nebo vedoucího mé diplomové práce Zdeňka Matěje. Všichni ti mi umožnili psát diplomovou práci, jejíž náplň mi dává smysl, jejíž realizace mě baví a jejíž výsledky vidím v reálném světě. Moc si vaší podpory vážím. Děkuji svému vedoucímu Zdeňku Matějovi za korektury a podporu ve změně tématu. Děkuji konzultantovi Honzu Mrázkovi za odpovídání na mé dotazy a za rady. Děkuji všem korektorům této práce: Veronice Burgerové, Miroslavu Horáčkovi a Zdeňku Matějovi. Děkuji mé přítelkyni Veronice a tučňákovi Adalbertovi za mentální podporu.

Abstrakt

Se zaváděním počítačového řízení železniční dopravy vznikaly systémy umožňující centrálnímu řídicímu počítači interakci s venkovními prvky zabezpečovacího zařízení – výhybkami, návěstidly apod. Tato práce se zaměřuje na návrh a implementaci sběrnice, která takovou interakci umožní, ovšem na kolejištích modelových. Práce navrhuje a implementuje nový protokol sběrnice pro řízení modelových kolejišť *MTBbus*. Je popsáno, proč je současný systém řízení kolejiště nedostatečný, jsou formulovány požadavky na nový systém, tento systém je implementován. V rámci práce vznikl detailní návrh nového protokolu, nové hardwarové moduly pro řízení kolejiště, modul pro komunikaci s počítačem, obslužný počítačový software a knihovna integrující nový hardwarový systém se současným řídicím softwarem kolejiště. Nový řídicí systém byl otestován a nasazen na skutečná kolejiště, čímž umožnil jejich další rozšiřování a zprovoznění nových způsobů řízení dopravy. Vznikl otevřený a robustní systém s výhledem dlouhodobé udržitelnosti, který svými vlastnostmi převyšuje mnohé současné komerční systémy řízení modelových kolejišť. Vzniknuvší systém je obecný, takže jej lze použít i v mnoha jiných aplikacích.

Klíčová slova

sběrnice, plc, mtb, mtbbus, rs485, embedded, stm32, qt, usb, cdc, protokol, avr, arm, dcc

Obsah

1 Úvod **3**

2 Současné nasazení sběrnice MTBbus **5**
2.1 *Popis systému MTB* 5
2.2 *IR čidla* 9
2.3 *Systém MTB v kontextu řízení modelového kolejiště* 10
2.4 *Proč současný systém MTB nedostačuje* 12
2.4.1 Všeobecné požadavky na nový systém MTB . . 13

3 Existující řešení **14**
3.1 *Digital Command Control* 14
3.1.1 Sběrnice zpětného hlášení 19
3.2 *BiDiB* 22
3.3 *Závěr* 22

4 Návrh MTB v4 **24**
4.1 *Podrobná specifikace požadavků* 24
4.1.1 Periferie v kolejišti 24
4.1.2 Ovládání 26
4.1.3 Další požadavky 27
4.1.4 Shrnutí požadavků 28
4.2 *Návrh systému MTB v4* 29
4.2.1 Vysokoúrovňový návrh 29
4.2.2 MTBbus 31

5 Implementace MTB v4 **42**
5.1 *MTB-USB v4* 42
5.1.1 Komunikační protokol s počítačem 42
5.1.2 Hardware 45
5.1.3 Firmware 48
5.2 *MTB-UNI v4* 49
5.2.1 Základní parametry 49
5.2.2 Hardware 50
5.2.3 Deska plošných spojů 53
5.2.4 Firmware 53
5.3 *MTB-2-AVR* 55

5.4 *IRdet* 57
5.5 *MTB Daemon* 59
5.5.1 Protokol TCP serveru 59
5.5.2 Volba nástrojů 60
5.5.3 Implementace 61
5.6 *hJOP MTB Network RCS knihovna* 63

6 Závěr **64**
6.1 *Možná rozšíření* 65

Bibliografie **67**

A Přílohy **70**

Seznam použitých zkratek

API Application Interface, rozhraní aplikace. 12, 26, 31, 59, 61

BiDiB BiDirectional Bus, viz `http://bidib.org/`. 22, 23

CDC USB Comunications Device Class, třída USB protokolu implementující tunelování sériové linky skrze USB. 30, 31, 43, 46

DCC Digital Command Control, mezinárodně užívaný standardizovaný systém pro digitální řízení modelové železnice. 5, 11, 14, 15, 17–22, 51, 65, 66

DIL Dual in-line package, způsob provedení pouzdra integrovaných obvodů s dvěma řadami pinů pro osazovaní do děr (tzv. *THT* osazování). 52, 55

DPS Deska plošnýcn spojů. 46, 47, 49, 50, 52, 53, 55, 58

HID Human Interface Device, třída USB protokolu sloužící k připojení zařízení interagujících s uživatelem – myší, klávesnic apod. 43

IR Infra-red. 7–10, 16, 26, 29, 49, 56–58

KMŽ Brno I Klub modelářů železnic Brno I. 5, 10–12, 16, 22, 25–28, 63, 65

LED Light emitting diode, součástka vydávající světlo určité barvy. 27, 28, 36, 48, 55, 56

MTB Hardwarový systém pro řízení modelových kolejišť složený ze sběrnice MTBbus, MTB modulů, MTB-USB desky a počítačových knihoven. 4–7, 9–13, 17, 23–32, 34, 36–45, 47, 49, 56–62, 64–66

MTB-UNI Nejrozšířenější slave modul sběrnice MTBbus, má 16 digitálních vstupů a 16 digitálních výstupů. 7, 8, 10, 25, 26, 28, 29, 37, 38, 41, 49, 50, 52–57, 64, 72

MTB-USB Master modul sběrnice MTBbus, implementuje rozhraní mezi MTBbus a počítačem (USB). 5, 6, 9, 29–32, 34, 36, 38–40, 42–47, 59, 64, 65, 71

MTBbus Model Train Bus[1], sběrnice určená pro řízení modelových kolejišť. v, 4–7, 22, 24–26, 28–32, 34, 35, 37–45, 47, 53–56, 64

NMRA National Model Railroad Association. 14, 19

PLC Programmable Logic Controller, robustní zařízení průmyslové automatizace. 3, 4, 19, 24, 64

S-COM Jednoduchý jednosměrný komunikační protokol, který umožňuje přenášet návěsti návěstidel [1]. 7, 21, 24, 25, 28, 49

SMD Surface mount device, elektronická součástka určená pro osazení přímo na povrch plošného spoje. 46, 53, 55, 58

TCP Transmission Control Protocol, používaný protokol transportní vrstvy počítačových sítí garantující spolehlivé doručování dat. 31, 59, 62

TTL Transistor-transistor-logic, v kontextu této práce standard definující jaká napěťová úroveň odpovídá jaké logické úrovni. 8, 51

USB Universal Serial Bus, v současnosti nejpoužívanější sběrnice pro připojení periferií k počítači. 22, 30, 43, 44, 46–48

1. Expanze zkratky do jejího plného významu nedává smysl, i tak budeme používat označení MTBbus, protože tak je zkratka zaužívaná.

1 Úvod

Programmable Logic Controller (PLC) je vestavěný počítač, který zpracovává vstupy, provádí jejich vyhodnocování, komunikuje s dalšími PLC obvody a nastavuje výstupy [2]. Na rozdíl od běžných počítačů typu *PC* mají PLC větší množství vstupů a výstupů, běží na nich *realtime* operační systémy a jsou vysoce robustní.

PLC jsou základní jednotkou průmyslové automatizace. Používají se například pro řízení světel dopravních křižovatek, výrobních linek, eskálátorů, řízení elektráren, rozvoden apod. Všude tam PLC zpracovávají data z čidel a ovládají periferie – například motorické pohony, signalizační diody, robotické ruky, lasery apod.

Obrázek 1.1: Průmyslové PLC moduly. Převzato z `wikimedia.org`.

V této práci se zaměříme na podmnožinu PLC modulů – totiž na takové, jejichž hlavním úkolem je správně zpracovávat vstupní signály a ovládat výstupní periferie. Řízení logiky průmyslového procesu přenecháme počítači, s kterým PLC moduly komunikují.[1]

1. V praxi může PLC modul řídit logiku procesu přímo, my však tuto situaci zkoumat nebudeme.

Cílem této práce je navrhnout a implementovat novou verzi komunikační sběrnice PLC obvodů zvanou *MTBbus* (*Model Train Bus*[2]).

MTBbus je sběrnice, která se využívá pro řízení modelových kolejišť. Je součástí systému MTB. Skrze sběrnici lze číst signály z kolejiště (například polohy výhybek, obsazenost kolejových obvodů) a povelovat prvky v kolejišti (návěstidla, přestavníky výhybek, přejezdy apod.). Obdobným způsobem funguje zapezpečovací zařízení na skutečné železnici.

V této práci autor nejprve popíše současné nasazení systému MTB. Budou formulovány důvody vedoucí k nutnosti aktualizace sběrnice. Autor popíše, proč jsou dostupná komerční řešení nevhodná a navrhne vlastní nové (1) protokoly, (2) hardwarové moduly a (3) počítačové programy, které jím formulovaný problém řeší.

2. Expanze zkratky do jejího plného významu nedává smysl, i tak budeme používat označení MTBbus, protože tak je zkratka zaužívaná.

2 Současné nasazení sběrnice MTBbus

Systém MTB vznikl okolo roku 2000, kdy se hledal vhodný systém pro programovatelné počítačové řízení modelové železnice, spoluprací nadšenců z *Klubu modelářů železnic Brno I* (dále jen KMŽ Brno I) a *Klubu železničních modelářů Praha 3* [1] [3].

Digital Command Control (DCC), dnes pravděpodobně nejrozšířenější[2] mezinárodně standardizovaný systém pro digitální řízení modelové železnice [4], tehdy ještě nebyl v České republice rozšířený. Navíc přirozeně vyvstal požadavek na minimalizaci nákladů a maximalizaci nezávislosti na komerčních produktech rodiny jednoho výrobce. Vznikl tedy systém MTB, který se pro řízení modelových kolejišť v KMŽ Brno I používá doposud [5]. Podotkněme, že systém MTB se do dalších klubů po České Republice nerozšířil – nejspíš proto, že není podporovaný komerčními softwary pro řízení modelové železnice a že jej jeho původní autor v roce 2008 přestal udržovat.

2.1 Popis systému MTB

Systém MTB se skládá z:

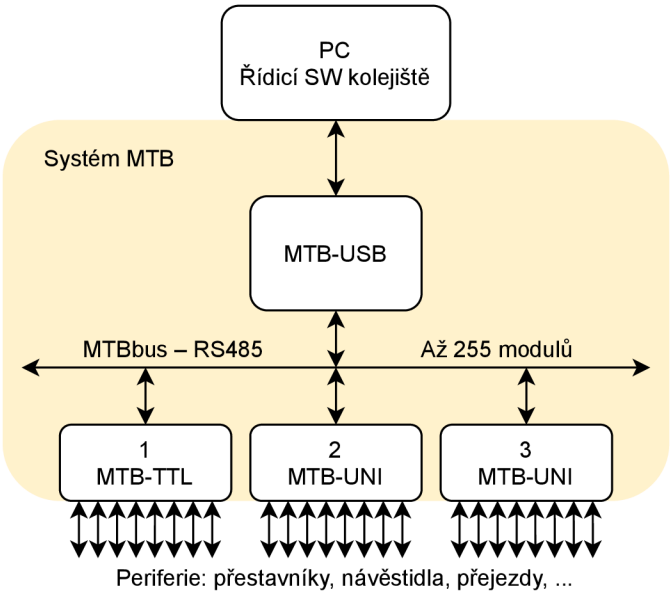
1. sběrnice MTBbus,
2. *MTB modulů*,
3. *MTB-USB* modulu,
4. počítačových knihoven pro přístup k MTB-USB.

Každé kolejiště má svou vlastní sběrnici MTBbus, ke které je připojený právě jeden MTB-USB modul a až 255 MTB modulů. MTB-USB řídí provoz sběrnice MTBbus a je připojen k počítači. MTB moduly jsou

---

1. Je překvapivé, že skupina nadšenců tehdy dokázala vytvořit systém, který by se mohl poměřovat se současnými komerčními systémy řízení modelových kolejišť, viz dále.
2. Autor této práce by rád odstranil slovo *pravděpodobně* a uvedl řádnou citaci. Bohužel neexistují studie, které by bylo možné citovat. Rozšířenost DCC se zakládá na autorových zkušenostech.

pevně instalovány v rámech kolejiště a komunikují po sběrnici MTBbus s MTB-USB modulem. Říkáme, že sběrnice je tzv. *single master, multiple slaves*. MTB-USB modul lze označit jako *master modul*, MTB moduly jsou tzv. *slave moduly*. Celou situaci přehledně ilustruje obrázek 2.1.



Obrázek 2.1: Topologie systému MTB.

Současná sběrnice MTBbus je založena na elektrickém standardu *RS485* [6]. Sběrnice RS485 je dvouvodičová (*R+, R-, GND*) vícebodová sériová poloduplexní komunikační průmyslová sběrnice vyvinutá s důrazem na odolnost vůči externímu rušení. Je vhodná pro přenos dat na větší vzdálenosti, řádově až stovky metrů [7]. Pro dosažení nejlepších elektrických vlastností by sběrnice měla být lineární. Sběrnice je na obou koncích terminována rezistorem *200 R* a pull-up a pull-down rezistory pro držení definované úrovně signálu.

RS485 je velice snadno implementovatelná do prakticky všech mikrokontrolérů – pro přístup ke sběrnici je třeba rozhraní *UART*, jeden pin pro řízení směru komunikace a driver k RS485.

Po sběrnici se komunikuje pevně definovaným protokolem. Protokol definuje příkazy pro moduly, odpovědi modulů, časování apod. [6]

Základní parametry MTBbus shrnuje tabulka 2.1.

Tabulka 2.1: Základní parametry sběrnice MTBbus [6]

Typ přenosu	RS485, formát dat UART
Komunikační rychlost	38400 Bd, 57600 Bd, 115200 Bd
Maximální počet modulů	255
Počet datových bitů	9
Stop bit	1
Parita	žádná
Maximální délka vedení	100 m

Zaměřme se nyní na MTB moduly. Každý MTB modul má:

1. 8bitovou adresu, která se konfiguruje *jumpery* přímo na modulu,
2. konfiguraci (typ vstupů/výstupů, rychlost sběrnice, ...),
3. vstupy (stav vstupů),
4. výstupy (stav výstupů).

MTB moduly mohou být různého typu:

- **MTB-UNI**

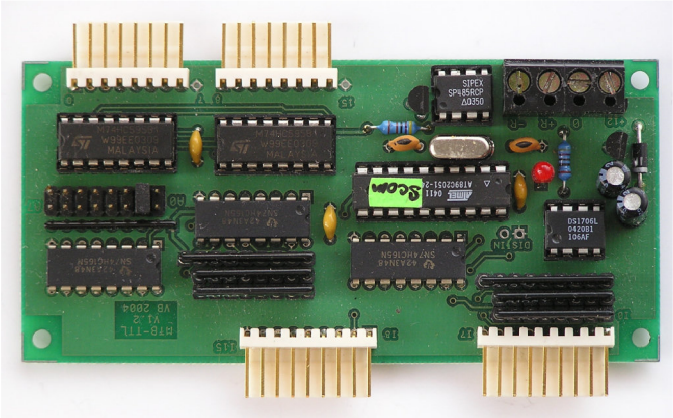
  MTB-UNI (*univerzální*) je nejpoužívanější typ modulu. Obsahuje 16 digitálních vstupů a 16 digitálních výstupů. Na výstupech 0–7 umožňuje kódovat návěst protokolem S-COM[3], což umožňuje připojení až 8 návěstidel k jednomu MTB-UNI. Modul dále umožňuje připojení IR čidel na vstupy[4]. Výstupy modulu jsou v režimu otevřeného kolektoru s maximální zátěží až 0.5 A / 8 výstupů.[5]

- **MTB-UNIm**

  MTB-UNIm modul je zjednodušením MTB-UNI modulu. Nemá podporu IR čidel, ale má výstupy v režimu otevřených kolektorů. Umožňuje připojení až 8 S-COM návěstidel.

3. S-COM je jednoduchý jednosměrný komunikační protokol, který umožňuje přenášet návěsti návěstidel [1].
4. IR čidla jsou bodová čidla detekující průjezd vlaku, viz 2.2.
5. Na modulu jsou osazeny 2 budiče výstupů, přičemž každý z budičů má celkový limit 0.5 A.

- **MTB-TTL**



Obrázek 2.2: Ukázka modulu MTB-TTL [3].

MTB-TTL je zjednodušením modulu MTB-UNI. Oproti MTB-UNI nemá podporu IR čidel a výstupy má v režimu TTL. Od tohoto typu modulu se v současnosti ustupuje, neboť TTL výstupy nejsou vhodné pro spínání vyšších napětí[6], a proto, že v případě neaktivního výstupu (neaktivní = +5 V; inverzní logika) aktivně napájí výstupní port, což způsobuje nechtěné chování, například pokud je na výstupním portu záměrně vypnutá periferie. V krajním případě může dojít až k přetížení TTL výstupu a jeho zničení. Na výstupy všech nasazených moduly typu MTB-TTL byly postupně instalovány dodatečné budiče s otevřenými kolektory.

- **MTB-REG**

  MTB-REG je modul umožňující generovat analogový výkonový výstup. Modul se připojí ke kolejím a řídí rychlost a směr lokomotivy v řízeném úseku kolejí.

  Tento způsob řízení jízdy (tzv. *analogový*) je dnes již překonaný. V současnosti se pro řízení jízdy lokomotiv na kolejišti používá tzv. systém *Digital Command Control* (viz 3.1), kde každá lokomotiva a mnohdy i vagón v sobě má mikroprocesor a celý systém je řízen digitálně [4].

6. Např. pokud periferie na výstupu obsahuje pull-up do +12 V.

Modul MTB-REG je tedy již překonaným modulem a autor jej uvádí především pro úplnost popisu systému MTB.

- **MTB-POT**

  Modul MTB-POT obsahuje 4 analogové vstupy a 4 digitální vstupy. Jeho původním účelem bylo, aby se připojil k potenciometru v pultu obsluhy kolejiště, kterým obsluha reguluje jízdu vlaku. Po sběrnici přepošle data modulu MTB-REG a tím dojde ke kýžené jízdě vlaku. Tento způsob řízení kolejišť se již nepoužívá, proto i modul MTB-POT na moderních kolejištích pozbývá svého smyslu.

Práce se sběrnicí z pohledu aplikace v počítači probíhá následovně:

1. Aplikace se připojí k MTB-USB modulu.
2. Aplikace provede sken aktivních modulů sběrnice.
3. Aplikace nahraje do všech aktivních modulů konfiguraci.
4. Aplikace přečte stav vstupů.
5. Aplikace zahájí provoz sběrnice – od této chvíle všechny moduly nahlašují změny stavů vstupů.
6. Aplikace čte vstupy, nastavuje výstupy a řídí provoz.
7. Aplikace ukončí provoz sběrnice a uvede výstupy MTB modulů do základního stavu.
8. Aplikace uzavře spojení s MTB-USB.

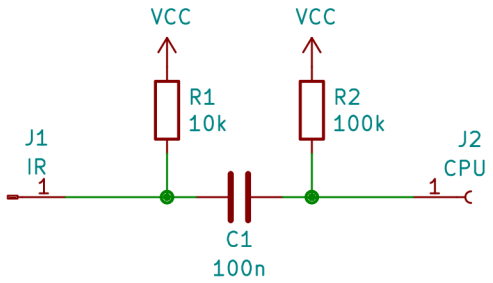
2.2 IR čidla

Nyní popíšeme princip fungování IR čidel v kolejišti, protože bude v dalších kapitolách stěžejní.

IR čidlo je bodový detektor průjezdu vlaku. Mezi kolejnice se vedle sebe umístí opticky oddělená vysílací dioda a fototranzistor, oba namířené vzhůru. Při průjezdu vlaku dojde k odrazu signálu od spodní části vozu nebo lokomotivy, čímž je detekován průjezd vlaku. Výhodou je nenápadná instalace mezi kolejové pražce, která neruší celkový dojem modelu.

Pro spolehlivou detekci odrazu od matných černých povrchů podvozků je třeba budit vysílací diodu vysokým proudem (až 200 mA), což vyžaduje použití diody v impulsním režimu. Současná MTB-UNI deska podporuje IR čidla na všech svých 16 vstupech, diody jsou buzeny tranzistory ve skupinách po 4 diodách.

Aby bylo zajištěno spolehlivé vyhodnocení odraženého signálu, je vstup z fototranzistoru zapojen na digitální vstup posuvného registru přes kapacitní vazbu (viz schéma 2.3). Kapacitní vazba odstraňuje stejnosměrnou složku signálu, která vzniká vlivem okolního osvětlení.



Obrázek 2.3: Zapojení vstupu kapacitní vazbou.

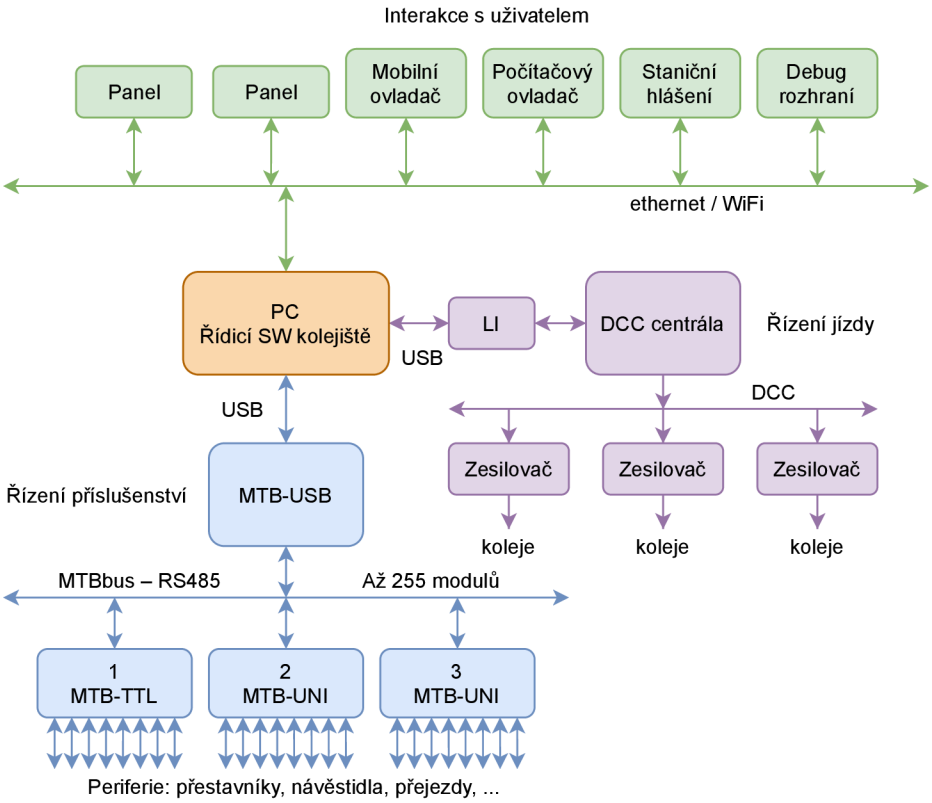
Vstupy MTB-UNI desky se konfigurují osazením rezistorů (běžný vstup) nebo kondenzátorů (vstup z IR čidla). Režim pinu musí být uložený v procesoru, aby se adekvátním způsobem zpracovával vstupní signál [8].

2.3 Systém MTB v kontextu řízení modelového kolejiště

V KMŽ Brno I je aktuálně systém MTB nasazen na dvou kolejištích, další nasazení je na modulovém kolejišti Mendelovy univerzity v Brně (MENDELU), s kterou klub spolupracuje. Na těchto kolejištích je aktuálně nasazeno dohromady 99 MTB modulů.

Pro kontext uveďme, jak systém MTB zapadá do celkové koncepce řízení kolejiště. Viz diagram 2.4.

Na první pohled je vidět, že kolejiště je řízeno dvěma různými systémy: systémem pro řízení příslušenství (modrá část diagramu) a systémem



Obrázek 2.4: Diagram komponent řízení kolejišť KMŽ Brno I.

pro řízení jízdy (fialová část diagramu). Vyvstává otázka, proč je nesloučit do systému jednoho. Integrovaná řešení nabízejí majoritní výrobci hardwaru pro digitální řízení modelové železnice. Řízení jízdy i příslušenství jedním systémem (DCC) je rozšířeným řešením jak v Česku tak v zahraničí. Všechna tato řešení jsou však o kompromisech, jak se dozvíme v 3.1.

U tvorby a nasazení systému MTB stály osoby, které pracují na zabezpečení skutečné železnice. Systém MTB je tak oproti komerčně dostupným řešením řízení modelových kolejišť navrhován s důrazem na spolehlivost a především bezpečnost.

Uveďme dva příklady za všechny: komerčně využívaná sběrnice RS pro snímání stavu kolejiště nedokáže rozpoznat, že nějaký z RS modulů (analogie MTB modulů) přestal pracovat [9]. Dále výstupní

moduly nijak nepotvrzují, že opravdu provedly akci, kterou počítač požaduje [10]. Tyto příklady ilustrují, že komerčně dostupná řešení jsou podle autorova názoru blíže hračce, než bezpečnému a spolehlivému systému pro řízení modelů.

Poslední část diagramu 2.4 je část zelená, která reprezentuje interakci s uživatelem. *Řídicí SW kolejiště* běží na samostatném serveru, ke kterému se připojují všichni uživatelé systému – dispečeři ze svých stanic, strojvedoucí ze svých ovladačů, ale také třeba technikové se svými diagnostickými nástroji nebo externí programy využívající API serveru. Server kontroluje identitu uživatelů a umožňuje jim řídit ty systémy, ke kterým mají uživatelé práva.

2.4 Proč současný systém MTB nedostačuje

Systém MTB s sebou nese řadu problémů.

1. **Licence, výrobní data**

   U současné implementace systému MTB bohužel nebyly řádně vyřešeny licenční podmínky mezi jeho autorem a Klubem. Souhrou událostí se KMŽ Brno I dostal do situace, kdy nemá zdrojová data schémat a layoutů desek plošných spojů ani zdrojové kódy firmwarů. Aktuální situace tak prakticky znemožňuje výrobu dalších MTB desek, o záměru chtít systém MTB nabízet mimo klub ani nemůže být řeč.

2. **Hardware**

   Systém MTB zaznamenal poslední aktualizaci hardwarových komponent v roce 2007. V současné době jsou některé použité součástky bohužel nedostupné, což znemožňuje výstavbu dalších částí kolejiště. To je zcela zásadní problém.

3. **Software**

   Současné MTB moduly využívají procesory *AT89C2051* (2 kB FLASH, 128 B SRAM). Tyto procesory svými parametry odpovídají době návrhu celého systému. Firmware v procesorech naráží

na jejich hardwarové limity – paměť je plná, takže nelze přidávat nové funkce. Procesorům navíc chybí některé klíčové periferie, například EEPROM paměť.

Současný systém MTB byl vyhodnocen jako celkově nedostatečný, je potřeba ho buď aktualizovat nebo nahradit systémem jiným.

2.4.1 Všeobecné požadavky na nový systém MTB

Na náhradu systému MTB autor práce stanovil především následující požadavky.

1. Systém musí být kompatibilní se současným řídicím softwarem kolejiště.
2. Systém musí být kompatibilní se současným hardwarem kolejiště.
3. Systém musí být udržitelný minimálně 20 let.
4. Finanční náklady a čas vložený do povýšení současného systému na nový systém by měly být minimalizovány.
5. Systém musí být schopen potvrzovat akce řídicího počítače a evidovat správnou funkčnost modulů.
6. Systém by měl být rozšiřitelný co se týče podporované funkcionality – nové požadavky by mělo být možné implementovat.

K naplnění těchto požadavků lze přistoupit dvěma různými způsoby: zakoupením komerčního produktu nebo vytvořením produktu vlastního. Prozkoumejme nejprve, jestli existují komerční řešení, která naplňují definované požadavky.

3 Existující řešení

Nyní představíme v současnosti nejrozšířenější standardy pro *digitální* řízení modelové železnice. Prozkoumáme zařízení, která tyto standardy podporují, a zhodnotíme jejich vlastnosti. Bude vyhodnoceno, jestli zařízení naplňují požadavky definované v předchozí kapitole.

3.1 Digital Command Control

Digital Command control (DCC) je celosvětově nejrozšířenější standard pro *digitální*[1] řízení modelové železnice [11]. K DCC existují alternativy, které však v Evropě nejsou rozšířené, proto je nebudeme uvažovat.[2]

DCC specifikuje, které komponenty se účastní řízení kolejiště a jak se tyto komponenty mají chovat – od elektrických standardů (např. doporučený způsob zapojení vodičů pod kolejištěm) až po komunikační protokoly. DCC je otevřený standard vytvořený organizací *National Model Railroad Association* (NMRA), která sdružuje modelářské kluby (a tedy i modeláře), jež jsou přímými uživateli DCC. Standard DCC je tedy navržen modeláři k naplnění potřeb modelářů.

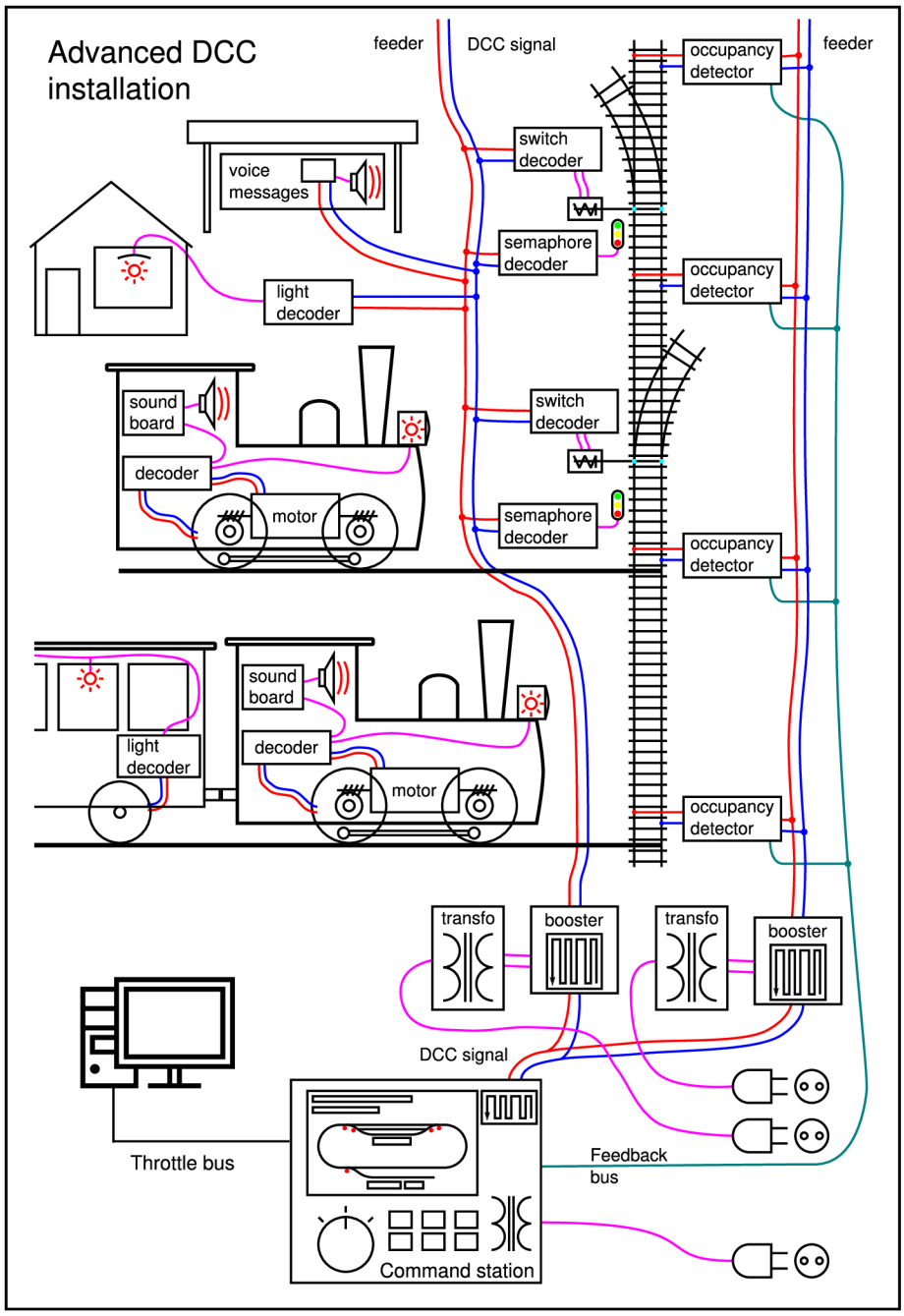
Jednotlivé prvky systému DCC jsou znázorněny v 3.1. Nyní tyto prvky stručně popíšeme.

Základní komponentou celého systému je *DCC centrála* (*Command Station*), která má 3 hlavní úkoly.

1. Číst stav *modulů zpětného hlášení* přes *sběrnici zpětného hlášení* (*feedback bus*).
2. Povelovat *dekodéry* (lokomotiv, přestavníků, návěstidel apod.).
3. Obousměrně komunikovat po *sběrnici ovladačů* (*throttle bus*).

Řízení dekodérů centrála provádí generováním *DCC signálu* (červený a modrý vodič ve schématu 3.1), do kterého centrála kóduje operace

1. *Digitální* znamená, že mezi jednotlivými prvky řízení proudí digitální data.
2. Toto tvrzení se zakládá na autorových zkušenostech, relevantní studie neexistují.



Obrázek 3.1: Schéma systému DCC. Převzato a upraveno z [12].

požadované po dekodérech – např. „dekodére číslo 541, zastav lokomotivu“, „dekodére číslo 4741, aktivuj čelní světla“.

Ke sběrnici ovladačů jsou připojené buď fyzické ovladače, nebo počítače. Operátor provozu zadává příkazy, například otočením kolečka ovladače nebo interakcí s příslušným softwarem, tyto příkazy jsou přeposlány do centrály, centrála je dále přeposílá dekodérům, které přímo vykonávají požadované akce.

Na *sběrnici zpětného hlášení* jsou připojeny moduly s digitálními vstupy, které jsou namontovány pevně v kolejišti a které čtou stav periferií. Centrála informuje uživatele skrze *sběrnici ovladačů* o změnách ve stavech vstupů *modulů zpětného hlášení*. Vstupy modulů zpětného hlášení jsou připojené například k signalizacím obsazení *kolejových obvodů*, takže uživatel nebo řídicí SW pozná, že se na koleji nachází nebo nenachází vlak, dále například ke koncovým spínačům poloh výhybek, takže řídicí software je schopen zabezpečit *vlakovou cestu*.

V KMŽ Brno I se používají následující výstupy a vstupy.

Výstupy:

- řízení jízdy lokomotiv,
- řízení osvětlení lokomotiv a vozů,
- řízení zvuků lokomotiv,
- nastavení poloh výhybek, výkolejek,
- rozpojovače,
- návěstidla,
- přejezdy,
- statické pouliční nebo domovní osvětlení,
- indikace v pultu obsluhy.

Vstupy:

- indikace obsazení kolejových obvodů,
- indikace bodového průjezdu vlaku daným místem skrze IR čidla (viz 2.2),
- indikace koncových poloh výhybek, výkolejek,
- indikace stavů přejezdů,
- tlačítka v pultu obsluhy.

Všimněme si klíčové vlastnosti systému: pro řízení dekodérů a pro sběr informací z kolejiště jsou použity 2 různé sběrnice: *dcc signál* a *sběrnice zpětného hlášení*. Platí totiž, že signál DCC je pouze jednosměrný. Signál putuje z DCC centrály přes *zesilovače* a přes koleje do lokomotiv, vozů a statických dekodérů, které poveluje. Existují jen velice omezené prostředky k poslání dat z lokomotivy zpět do DCC centrály[3]. Tyto prostředky navíc vznikly až v pozdějších verzích normy DCC, takže dnes nejsou všeobecně zaužívané [13]. Sběrnice DCC je tedy prakticky vzato jednosměrná – bez odpovědí dekodérů na příkazy.

To znamená, že centrála musí DCC příkazy posílat opakovaně, aby zaručila, že je dekodér skutečně přijme. Může se totiž snadno stát, že lokomotiva zrovna není schopna přijímat signál – například se pohybuje po znečištěných kolejích. Problém s přijímáním dat vlivem špatného kontaktu se netýká *dekodérů příslušenství* (na obrázku 3.1 *switch decoder* a *semaphore decoder*). Tyto dekodéry jsou připojeny pevným kabelem ke kolejím nebo přímo k centrále, takže mají nejlepší předpoklady dekódovat DCC signál korektně. Stále však platí, že dekodér neodpoví, že přijal a provedl povel. Vykonání povelu mohlo být narušeno čímkoliv od elektromagnetického rušení signálu až po nepřijetí povelu vlivem chyby ve firmwaru dekodéru.

Může se zdát, že samotný protokol DCC nesplňuje požadavky popsané v 2.4.1 a proto jej nemá smysl uvažovat jako nástupce současného systému MTB. Ačkoliv striktně vzato je tento závěr správný, je nutné si uvědomit širší souvislosti.

Při řízení provozu na modelovém kolejišti sledujeme mnohé cíle, z nichž na prvním místě stojí bezpečnost. Na skutečné železnici je bezpečnost samozřejmě mnohem důležitější než na železnici modelové. Přesto například materiální škody vzniklé projetím návěsti zakazující jízdu, vykolejením soupravy, srážky nebo následného pádu modelu na zem jsou tak markantní, že je třeba bezpečnost provozu řešit.

Bezpečnost řízení provozu železnice skutečné i modelové stojí na tom, že vlak nevjede do míst, kde by mu hrozila nehoda. K nehodě může dojít dvěma způsoby.

3. Takovému systému říkáme *RailCom* [13].

1. **Vlak stojí a je nesprávně rozjet do nebezpečného místa.**

   I přes to, že dekodéry (příslušenství) nepotvrzují přijetí signálu a provedení akce, této situace se jsme schopni vyvarovat. Všechny klíčové komponenty kolejiště, které řídí směr pohybu vlaku (výhybky, výkolejky) indikují svůj stav. To znamená, že když se nepovede provést například přestavení výhybky vlivem nedoručení příslušného *DCC paketu*, výhybka se nepřestaví, moduly zpětného hlášení správně indikují, že výhybka se nehýbe, řídicí software správně vyhodnotí, že nedošlo ke splnění podmínek pro rozjetí vlaku a vlak nerozjede.

   Z tohoto pohledu tedy nepotvrzování DCC příkazů nevede k nebezpečné situaci.

   Jiná situace nastane u dekodérů, které provádějí akce, aniž by indikovaly svůj stav. Takové dekodéry jsou například *dekodéry návěstidel* (*semaphore decoder* na obrázku 3.1). Tyto dekodéry však neovlivňují směr jízdy vlaku. Špatná návěst na návěstidle je nemodelová, ale ke hmotným ztrátám nedojde. [4]

2. **Vlak se nepovede zastavit.**

   Může nastat situace, kdy se vlak blíží k místu zastavení, řídicí software mu správně pošle příkaz k zastavení, ale lokomotiva přesto nezastaví. Například vlivem již popsaného špatného kontaktu lokomotivy s kolejemi a tudíž neschopnosti řádně dekódovat DCC signál. Norma DCC totiž specifikuje „pokud dekodér nedekóduje signál, zůstává ve stejném stavu“. Tedy například lokomotiva se pořád pohybuje vpřed.

   Ačkoliv popsaná situace v praxi nastává, není v záběru této práce s ní naložit. Tato práce se zaměřuje především na řízení příslušenství a snímání stavu kolejiště, nikoliv na systém řízení jízdy hnacích vozidel.

4. Jiná situace panuje na skutečné železnici, kde je návěst návěstidla důležitým komunikačním prvkem mezi dispečerem a strojvedoucím. Je nutné, aby strojvedoucí viděl správnou návěst, protože nesprávná návěst by mu mohla povolit jízdu, přestože se například proti němu blíží vlak. Nebo by například mohla povolit jízdu vyšší rychlostí, než jaká je v daném úseku předepsaná, a vlak by později nemusel stihnout zabrzdit. Proto se na skutečné železnici kontroluje, že světly návěstidel, která mají být rozsvícená, skutečně prochází proud.

Závěrem uveďme, že z dosavadního popisu systému DCC plyne, že tento systém je pro řízení příslušenství vhodný, i když s drobnými výhradami.

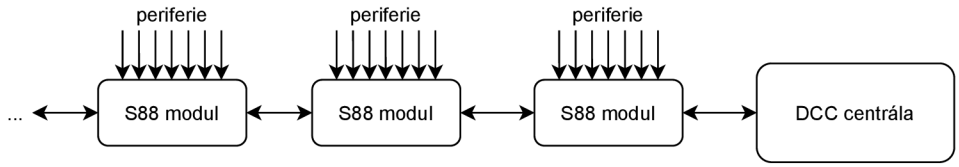
3.1.1 Sběrnice zpětného hlášení

V předchozí kapitole jsme při formulování závěrů vycházeli z předpokladu, že jsme schopni bezpečně indikovat stav všech prvků v kolejišti (pomocí *sběrnice zpětného hlášení – feedback bus* ve schématu 3.1). Prozkoumejme spolehlivost této sběrnice.

NMRA nedefinuje jednotný standard pro *sběrnici zpětného hlášení* [10], různí výrobci používají různé (vlastní) sběrnice [14]. Analyzujme tedy nejpoužívanější sběrnice.

S88

S88 je základní sběrnice zpětného hlášení. Její fungování lze v zásadě charakterizovat jako „vylepšené posuvné registry“ [15]. Jednotlivé moduly jsou řazeny za sebe, propojují se vždy sousední moduly. První modul je připojen do DCC centrály.



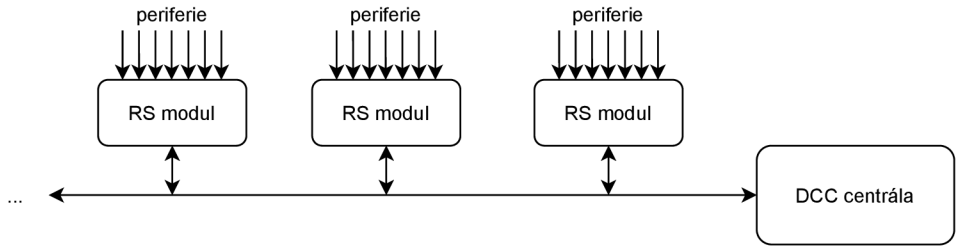
Obrázek 3.2: Topologie sběrnice S88.

S88 je nízkonákladová sběrnice určená pro malá kolejiště. Limitujícím faktorem je především napájení z 5 V, které není vhodné na delší rozvody, a malá odolnost proti rušení [15].

Jak je patrné z topologie sběrnice na obrázku 3.2, při výpadku (napájení) jednoho modulu jsou ztracena data i ze všech dalších modulů dále od centrály. Takové chování je pro profesionální PLC systém nevhodné.

RSbus

RSbus je sběrnice firmy *Lenz Elektronik*. Oproti *S88* je organizována do topologie sběrnice (viz obr. 3.3). Jednotlivé moduly jsou napájeny samostatně, sběrnice je dimenzovaná až na 128 modulů, každý s 8 digitálními vstupy [9] [16].



Obrázek 3.3: Topologie sběrnice RSbus.

Komunikaci po sběrnici řídí DCC centrála – posílá výzvy jednotlivým modulům a moduly odpovídají, pokud mají nějaká data k odeslání. Moduly odesílají data jen pokud došlo ke změně vstupů, což znamená, že nelze odlišit nefunkční modul od funkčního modulu se setrvalým stavem vstupů [16].

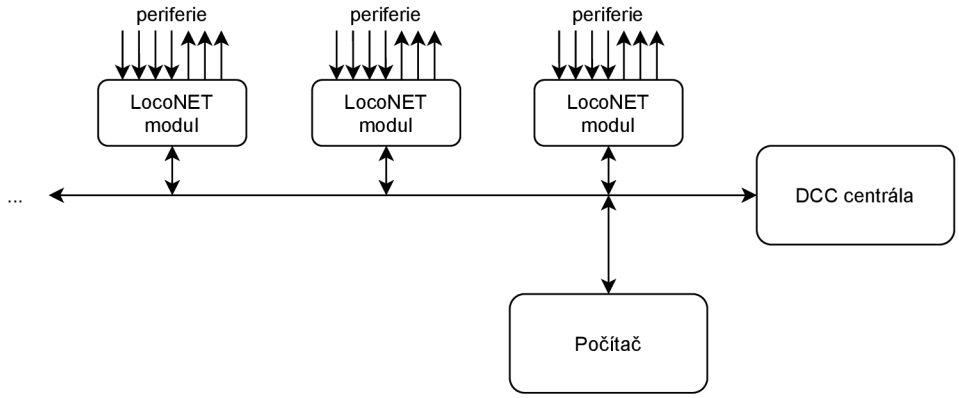
Takové chování bohužel nesplňuje požadavky definované v 2.4.1. Pokud by například došlo výpadku *RS modulu* a následnému obsazení kolejového obvodu, jehož stav tento modul indikuje, z pohledu řídicího softwaru by byl kolejový obvod stále volný. Řídicí software by pak povolil vjezd vlaku na obsazenou kolej, čímž by způsobil srážku.

K výpadku RS modulu může dojít snadno například ztrátou napájení vlivem chybné obsluhy, reakce pojistky na přetížení, poruchy apod.

LocoNET

Poslední z analyzovaných sběrnic je *LocoNET*. LocoNET je proprietární sběrnice společnosti *Digitrax* [17], jejíž plná specifikace není volně dostupná [18]. Plné použití této sběrnice podléhá licenčním poplatkům [18]. Zajímavou vlastností sběrnice je, že slučuje sběrnice *throttle bus* a *feedback bus* (viz schéma 3.1). Řídicí software v počítači tak může

přímo dotazovat moduly zpětného hlášení a úplně tak obejít DCC centrálu. Sběrnice LocoNET je navíc obousměrná, LocoNET moduly jsou vstupně-výstupní, takže počítač může přímo povelovat periferie [17], viz 3.4.



Obrázek 3.4: Topologie sběrnice LocoNET.

Sběrnice LocoNET umožňuje počítači zjišťovat stav modulů přímou komunikací bez zapojení centrály [19], takže počítač může rozpoznat, které moduly jsou na sběrnici přítomny, výpadek modulu a oživení modulu po výpadku.

Sběrnice LocoNET splňuje požadavky na bezpečné řízení periferií definované v 2.4.1. Při bližší inspekci však zjišťujeme, že nasazení této sběrnice by znamenalo nakoupit a vyměnit cca 100 modulů v kolejišti. Výměna by navíc musela být rozsáhlejší, LocoNET moduly přímo nepodporují například generování signálu *S-COM*. Pro nasazení sběrnice LocoNET by musely být instalovány přídavné moduly pro řízení návěstidel.

3.2 BiDiB

Zajímavou alternativou k systému DCC je sběrnice *BiDiB* [5] (*BiDirectional Bus*). BiDiB je komunitní sběrnice s otevřeným protokolem, která do velké míry splňuje všechny požadavky definované v 2.4.1. Systém BiDiB lze provozovat na více komunikačních médiích (RS485, Ethernet, sériový port), návrh systému je podobný návrhu USB – na sběrnici je jeden *master*, až 32 *slave* v každém segmentu, přičemž jedním ze *slave* může být *hub* vedoucí do dalšího segmentu.

Autoři BiDiB si dali za cíl, podobně jako autor této práce, vytvořit bezpečný systém pro řízení modelových kolejišť. Systém tak podporuje potvrzování příkazů, aktualizaci firmwaru modulů apod.

Pro nasazení BiDiB by bylo třeba provést netriviální úpravy aktuálních modulů kolejiště – BiDiB například definuje standardní konektory pro propojení modulů. Dále by bylo třeba vyřešit, jak naložit s limitem 32 desek v jednom segmentu sběrnice (protože MTBbus je navržen až pro 255 modulů) a s jiným způsobem adresování modulů sběrnice.

3.3 Závěr

V této kapitole byly popsány základní přístupy k řízení moderní digitální modelové železnice. Bylo zkoumáno, jestli komerční produkty splní požadavky definované v 2.4.1. Po vyloučení technicky nezpůsobilých řešení zbyly sběrnice LocoNET (s poznámkou, že její nasazení by bylo finančně náročné a pracné) a BiDiB.

Při uvažování sběrnice LocoNET vyvstal problém s integrací ovládání návěstidel. Tento problém je obecnějšího rázu – jakékoliv komerční řešení není schopné nabídnout takovou flexibilitu, jakou KMŽ Brno I potřebuje. Je nutné integrovat systém řízení kolejiště se současnými periferiemi, ale i s periferiemi budoucími. Je chtěné mít možnost pružně reagovat na nové periferie, které se objeví například za 10 let.

5. `http://bidib.org/`

Uvažme také aspekt zpětné kompatibility – vlastní systém umožní co možná nejrozumněji zachovat kompatibilitu se stávajícím systémem řízení kolejiště a minimalizovat tak finanční náklady a čas nutný pro aktualizaci systému. Nutnost rozumné zpětné kompatibility tak fakticky vylučuje i nasazení BiDiB.

Vnímejme tuto kapitolu tedy především jako přehled technologií, které se pro řízení současné digitální modelové železnice používají. Přistupme k návrhu nástupce současného systému MTB výrobou vlastních komponent.

4 Návrh MTB v4

V této kapitole představíme návrh vlastního PLC systému, který splňuje požadavky definované v 2.4.1. Návrh systému vychází z přirozeného požadavku na minimální změny v aktuálním hardwaru a softwaru kolejiště. Nový systém je navržen jako iterace současného systému MTB; jmenuje se *MTB v4*.

MTB v4 řeší všechny problémy aktuálního systému MTB popsané v kapitole 2.4. Při návrhu byly vzaty v potaz aktuální technologie v oblasti PLC obvodů, přičemž byla zvážena vhodnost jejich implementace do systému MTB v4. Tyto úvahy popíšeme. Detailně rozebereme současnou implementaci protokolu MTBbus a u každé položky budeme argumentovat, jestli ji zachovat.

4.1 Podrobná specifikace požadavků

Pro návrh systému MTB v4 je třeba formulovat přesnější specifikaci požadavků, než jsou požadavky definované v 2.4.1. Nyní tyto přesné požadavky formulujeme, přičemž budeme klást důraz na vysvětlení změn oproti současnému systému MTB.

Zaměřme se nejprve na požadavky, které na systém MTB kladou komponenty, které jej užívají.

4.1.1 Periferie v kolejišti

K MTB se na straně kolejiště připojují jednotlivé periferie popsané v 3.1. Všechny v současnosti na kolejišti používané vstupy jsou digitálního charakteru (binární vstupy), výstupy jsou buď digitální binární nebo S-COM. S-COM výstupy jsou ve skutečnosti digitální výstupy, do kterých MTB modul moduluje S-COM signál. S-COM signál je záměrně jednoduchý, pomalý signál, který lze s přehledem vytvářet běžným výstupním pinem procesoru nebo posuvného registru [1].

Oproti současnému MTB by nové MTB mělo podporovat tzv. *kmitavý výstup* s počítačově definovanou frekvencí kmitání a pevnou střídou. Frekvence kmitání postačí v řádu jednotek *Hz*. Kmitavý výstup se využije pro indikace v pultech a k ovládání rozpojovačů[1]. I kmitavý výstup je však z pohledu návrhu hardwaru jednoduchý digitální výstup – pro přidání kmitavého výstupu stačí upravit komunikační protokol MTBbus, software v počítači a firmware modulů.

Současné moduly MTB-UNI umožňují výstupy typu S-COM pouze na pinech 0–7, což je omezení dané malou pamětí procesoru současných MTB-UNI modulů. Toto omezení by mělo *MTB v4* relaxovat a umožnit tak, aby každý výstup mohl být v jednom z režimů

1. digitální,
2. S-COM,
3. kmitavý.

Co se týče vstupně-výstupních požadavků periferií kolejiště, MTB v4 by si vystačilo pouze s moduly typu MTB-UNI. Na kolejištích KMŽ Brno I v současnosti není žádný prvek, který by vyžadoval například analogový vstup, analogový výstup nebo *PWM* výstup. Na kolejištích se sice hojně používají servomotory, ale vždy pro řízení dvoupolohových prvků: výhybka, výkolejka, mechanické návěstidlo. Všechny tyto prvky k sobě mají řídicí elektroniku, která interaguje se systémem MTB pouze digitálními piny. Typicky stačí na každý takový prvek 2 digitální výstupy MTB-UNI modulu (pro nastavení požadované polohy) a případně 2 digitální vstupy (pro detekci koncové polohy).

Zmiňme na tomto místě, že rozhodnutí neimplementovat PWM výstupy do MTB-UNI v4 modulu je do velké míry koncepčním rozhodnutím autora této práce. Autor preferuje distribuovaný přístup, kdy „jedna komponenta dělá jednu věc a dělá ji dobře“. Redukuje se tak počet různých variant modulů, čímž se zjednodušuje sériová výroba.

Autor se rozhodl vydat cestou univerzálního kompaktního snadno sériově vyrobitelného modulu s prostými digitálními vstupy a výstupy.

1. Rozpojovač je mechanické zařízení pevně spojené s kolejištěm, jehož účelem je rozpojit modelové spřáhlo bez ručního zásahu obsluhy do kolejiště.

Připomeňme, že MTB-UNI moduly umožňují speciální typ vstupů – IR vstupy (viz 2.2). Tyto vstupy vyžadují další elektroniku navíc. V nových MTB-UNI v4 je přímá podpora pro IR čidla zrušena. Bude navržen samostatný přídavný modul pro buzení a vyhodnocování stavu IR čidel, který se bude připojovat přímo k digitálním vstupům MTB-UNI modulu. Ušetří se tak elektronika na MTB-UNI modulech, na kterých se IR vstupy nevyužívají, zjednoduší se návrh desky plošných spojů MTB-UNI v4 a umožní se využít IR čidla i s jinými moduly, než MTB (např. pro přímou indikaci v pultech).

V novém systému MTB tedy budou stačit pouze moduly typu MTB-UNI, z nichž bude navíc odstraněna přímá podpora IR čidel. I přes to by bylo neperspektivní si podporu jiných typů modulů zavřít volbou nevhodného komunikačního protokolu modulů. Nový protokol sběrnice MTBbus by měl být navržený univerzálně pro nejrůznější možné typy desek, které v budoucnu mohou přijít. Autor této práce má již v záměru takové desky desky vytvořit, viz 6.1.

4.1.2 Ovládání

Systém MTB interaguje s periferiemi v kolejišti a s řídicím softwarem (viz 2.1). Na kolejištích KMŽ Brno I se v současné době využívá řídicí software hJOP[2]. Tento software běží na operačním systému Windows a s hardwarem kolejiště komunikuje pomocí dynamicky linkované knihovny, která musí dodržet specifikované API [20]. Na straně řídicího softwaru jsou tedy pro návrh MTB v4 takřka neomezené možnosti.

Princip řízení kolejiště počítačem však definuje zásadní požadavek na organizaci sběrnice MTBbus v4: jednotlivé prvky jsou řízeny počítačem a výhradně počítačem.

Novým požadavkem je, aby v počítači mohlo existovat více programů, které sběrnici řídí. V praxi totiž nastává situace, kdy jsou na jednu sběrnici připojené systémy, které mají být z principu řízeny různými programy: například řízení provozu vlaků a řízení pouličního a domovního osvětlení. Je praktické mít oba systémy zapojené do jedné

2. `https://hjop.kmz-brno.cz/`

sběrnice (aby se minimalizoval počet MTB modulů). Dává však také smysl, aby software pro řízení provozu železnice neřídil osvětlení, protože to není jeho účel.

Dalším příkladem z KMŽ Brno I je jedna sběrnice, na které jsou připojeny periferie řízení tramvajové dopravy zároveň s periferiemi řízení silniční dopravy. Přestože oba systémy sdílí společnou sběrnici, na úrovni počítače by tyto systémy měly řídit dva různé programy.

Jednou z hlavních přidaných hodnot MTB v4 je, že umožní popsané *multi-master* řízení.

4.1.3 Další požadavky

Uveďme nyní některé méně důležité požadavky na MTB v4, jejichž naplnění nám však výrazně zpříjemní používání systému. Již teď předejměme, že všechny tyto požadavky se v implementaci podařilo naplnit.

1. **Bude možné nalézt MTB moduly i po startu sběrnice.**

   V kapitole 2.1 byl popsán striktní proces používání sběrnice – od vyhledání modulů až po ukončení práce se sběrnicí. Současná implementace tohoto procesu neumožňuje, aby počítač detekoval moduly, které na sběrnici přibudou za jejího chodu. Taková situace může nastat, například když je prvně zapnut řídicí počítač a až pak napájení sběrnice. Je nepříjemné muset v takovém případě manuálně spouštět proces skenování MTB modulů znovu.

   Obecně se celý postup práce se sběrnicí popsaný v kapitole 2.1 zdá být zbytečně striktním. MTB v4 by mělo tento proces zjednodušit.

2. **Z počítače bude možné na každém modulu zapnout identifikační LED.**

   Pro snadnou identifikaci konkrétního modulu v kolejišti, například při diagnostice závady, by operátor měl mít možnost z počítače na libovolném modulu zapnout indikační LED.

3. **Firmware MTB modulů by mělo být možné aktualizovat přímo po sběrnici MTBbus.**

   V kolejištích KMŽ Brno I je aktuálně nasazeno 99 MTB modulů. Aktualizace jejich firmwaru ručním programováním by byla časově náročná. MTB v4 by mělo umožňovat aktualizaci firmwaru přímo po MTBbus, nejlépe za současného plného chodu zbytku modulů.

4.1.4 Shrnutí požadavků

Shrňme stručně všechny požadavky na systém MTB v4.

1. MTB-UNI budou nově podporovat kmitavé výstupy.
2. MTB-UNI budou nově podporovat S-COM výstupy na všech výstupních pinech.
3. Systém MTB v4 musí být navržený univerzálně pro nejrůznější typy modulů. Jedinou společnou vlastností modulů je, že snímají vstupy a nastavují výstupy.
4. Je třeba vyřešit zpětnou kompatibilitu se stávajícími MTB moduly v kolejišti.
5. K MTB v4 se musí na straně počítače dát přistoupit skrze *hJOP RCS API*[3].
6. V počítači může běžet více softwarů řídících výstupy a snímajících vstupy modulů sběrnice MTBbus v4.
7. Moduly sběrnice MTBbus v4 by mělo být možné průběžně vyhledávat a detekovat nefunkční.
8. Z počítače bude možné na každém MTB modulu zapnout identifikační LED.
9. Firmware modulů bude možné aktualizovat přímo po MTBbus.
10. Implementace MTB v4 by měla využívat osvědčené komponenty, u nichž je dlouhodobý výhled dostupnosti.
11. MTB moduly potvrzují příkazy, lze monitorovat jejich správnou funkčnost.
12. Celé řešení by mělo být psáno opensource a openhardware.

3. `https://hjop.kmz-brno.cz/rcs`

4.2 Návrh systému MTB v4

Nyní vyjdeme z detailních požadavků definovaných v předchozí kapitole a provedeme návrh nového systému pro sběr dat a ovládání periferií kolejiště. Detailně zdůvodníme každý aspekt nově navrženého systému.

4.2.1 Vysokoúrovňový návrh

Základní koncept fungování systému MTB „single master, multiple slaves“ zůstane zachován. Bude vytvořena nová verze řídícího modulu MTB-USB, který se stane řídicím prvkem nové sběrnice MTBbus v4. Model *single master, multiple slaves* je osvědčeným modelem, který se snadno implementuje, jednoduše se v něm počítají nejlepší a nejhorší časy odezvy sběrnice a obecně se dobře mapuje na problém, který systém MTB řeší. Proto jej autor zachová.

Vytvořením nové verze MTB-USB autor sleduje především naplnění cíle existence svobodné implementace systému MTB. Nová deska umožní využití moderních procesorů a integrovaných obvodů, které usnadní implementaci. MTB-USB deska může být vytvořena jako úplně nová (bez nutnosti zpětné kompatibility se současným MTB-USB), protože je pouze jedna pro celé kolejiště a tedy je její výměna levná a snadná.

Bude navržen nový MTB-UNI modul, který splní všechny požadavky definované výše a který bude založen na moderních, ale dlouhodobě dostupných součástkách.

Nebude však nutné vyměnit všechny stávající MTB-UNI moduly v kolejišti. Stávajícím modulům se vymění pouze procesor, starým modulům tudíž bude umožněno fungovat na nové sběrnici.

Autor navrhne a popíše IR desku, která si využije při zapojení IR čidel k novému modulu MTB-UNI v4.

Zachování většiny stávajícího hardwaru současných MTB-UNI modulů a snaha o zachování kabeláže sběrnice s sebou nesou ponechání

sběrnice MTBbus na standardu RS485. RS485 nepřináší žádné zjevné nevýhody, takže není důvod ke změně hardwarové vrstvy sběrnice.

Nové procesory umožní trvalé uložení konfigurace v MTB modulech. Přesto bude autoritativním zdrojem konfigurace modulů i nadále počítač: po nalezení modulu na sběrnici jej počítač zkonfiguruje dle konfiguračního souboru programu. Tento přístup autor zvolil, aby bylo možné konfiguraci všech modulů kolejiště centralizovaně ukládat a verzovat.[4]

MTB moduly budou skrze MTB-USB komunikovat s počítačem po portu USB, kterým se tuneluje *USB CDC* protokol. V počítači tak bude MTB-USB modul dostupný jako sériový port. Počítač nemůže komunikovat po sběrnici RS485 přímo, protože není vybaven potřebnými hardwarovými periferiemi. Proto bude komunikovat s modulem MTB-USB, který řeší časově kritické operace sběrnice MTBbus[5].

Otázkou zůstává, kde naplnit požadavek na *multi-master* řízení celého systému. Existují 2 možné přístupy:

1. MTB-USB se bude k počítači připojovat rozhraním, které nativně umožňuje komunikaci pouze s jednou aplikací (napříkld USB), řízení více programy se vyřeší na úrovni softwaru v počítači.
2. MTB-USB se připojí k počítači rozhraním, které přirozeně podporuje více připojených zařízení, například *ethernet*.

Oba přístupy se v komerčních systémech pro digitální řízení modelové železnice běžně používají[6]. Ať bude podpora více řídicích systému implementována kdekoliv, oproti jednomu řídicímu systému vyžaduje tato podpora netriviální softwarovou logiku navíc. Autor práce se rozhodl přesunout co nejvíce složitější logiky do počítačových aplikací, protože ty se mnohem snáze vyvíjejí, ladí a aktualizují než firmware v embedded procesorech. V počítači je navíc možné využívat vysokoúrovňovější nástroje. Proto se autor vydal cestou, kdy se MTB-USB

4. Uložení konfigurace v modulech i tak má smysl – například aby modul mohl aplikovat uložený základní stav výstupů ihned po zapnutí napájení a nemusel čekat na řídicí SW v počítači.
5. Například kontrola odpovědi modulu v časovém limitu cca 250 *us*.
6. Viz např. rozhraní *Lenz LI-USB-Ethernet*, centrála *Digikeijs DR5000* aj.

připojuje k počítači pomocí virtuálního sériového portu, který umožňuje připojení jen jedné počítačové aplikace.

V počítači vznikne nová aplikace *MTB Daemon*, která se na jedné straně připojí k MTB-USB a na straně druhé vystaví JSON TCP server s jednoduchým API, které umožní povelovat MTB moduly z více počítačových programů. Získá se tím mj. přehledné programové API, které umožní snadné připojení z nejrůznějších programovacích jazyků. Tím se systém MTB zpřístupní širšímu spektru programátorů.

Třída CDC mezi počítačem a MTB-USB byla zvolena proto, že se jedná de facto o standard pro připojení specifických periferií k počítači, například v oblasti robotiky. Využijeme toho, že protokol CDC má nativní podporu ovladačů v majoritních operačních systémech a není tak nutné instalovat speciální ovladače. Tento aspekt umožní snadné nasazení dalším uživatelům.

4.2.2 MTBbus

Nyní podrobně rozebereme návrh nové sběrnice MTBbus v4. Autorem vytvořená plnohodnotná dokumentace nové sběrnice je k dispozici online[7].

Podrobné požadavky definované v 4.1.4 vynucují razantní úpravy komunikačního protokolu sběrnice[8]. Autor této práce se rozhodl sběrnici navrhnout od základů znovu. Sleduje tím především zajištění přehlednosti komunikačního protokolu a také vyřešení potíží s licencováním původního protokolu.

7. `https://github.com/kmzbrnoI/mtbbus-protocol`
8. Například umožnit odesílat delší zprávy (kvůli aktualizaci FW MTB modulů), navrhnout protokol univerzálně pro libovolné typy modulů apod.

Hardware

Jak již bylo zmíněno výše, po hardwarové stránce je zachován standard *RS485*. Počet komunikačních bitů je ponechán na 9[9]. Devátý bit ve zprávách z MTB-USB do MTB modulu indikuje, že se jedná o první byte zprávy. Ostatní byty zprávy mají tento bit nastavený na 0. Devítibitová komunikace je použita s cílem zavést na sběrnici klíčový pojem *zpráva*. Díky devátému bitu MTB modul dokáže poznat, kde zpráva začíná. Počet stop bitů je ponechán standardně na jednom bitu, kontrola konzistence dat paritou není použita.

Současná sběrnice MTBbus podporuje různé rychlosti – 38400 *Bd*, 57600 *Bd* a 115200 *Bd*. Komunikace vždy začíná na 38400 *Bd*, jedna z prvních zpráv je příkaz na změnu rychlosti. Při výpadku a opětovném oživení musí modul postupně vyzkoušet všechny 3 rychlosti a sledovat, kdy přijímá korektní data [6].

Možnost volby rychlosti by autor chtěl ponechat – některá kolejiště se spletitým zapojením sběrnice, dlouhou sběrnicí nebo sběrnicí se špatnými elektrickými vlastnostmi mohou vyžadovat nižší rychlosti. Postačí ponechat volbu mezi třemi rychlostmi. Rychlost nižší než 38400 *Bd* není třeba – tuto rychlost by měla zvládat každá sběrnice. Při rychlosti 115200 *Bd* dojde ke 20 skenům jednoho modulu za sekundu při počtu 50 modulů na sběrnici. Tato hodnota je pro účely zabezpečovacího softwaru již poměrně rozsáhlého kolejiště dostatečná, takže není třeba vyšších rychlosti.

Princip komunikace

Chod MTBbus řídí *master* deska – MTB-USB, která periodicky dotazuje všechny moduly (pomocí zprávy *Module Inquiry*). MTB modul na každý dotaz odpoví právě jednou zprávou[10]. Pokud MTB modul nemá žádná data k odeslání, odpoví zprávou *Acknowledgement*. Tím MTB-USB modulu potvrdí, že zprávu přijal a že komunikuje. MTB-USB tak

9. V dalším textu budeme o jednotlivých devítibitových blocích dat mluvit jako o bytech, přestože se fakticky jedná o devítibitová slova. Pro účely datového využití zprávy je však relevantních jen 8 bitů – byte.
10. Proto není nutné ve směru *slave* → *master* nastavovat devátý bit.

může monitorovat aktivní moduly sběrnice a dokonce detekovat moduly nové. Mechanismus pravidelného dotazování modulů nazýváme *polling*.

Příklad komunikace po sběrnici:

```
> Module 1 Inquiry
< Acknowledgement  # Modul 1 žije, žádná data k odeslání
> Module 2 Inquiry
# Timeout - modul 2 nežije
> Module 3 Inquiry
< State of inputs changed, new state: 0b00001010 0b11110000
> Module 6 Inquiry
< Acknowledgement
> Set Outputs of module 1 to 0b00101010 0b01000101
< Outputs Set to 0b00101010 0b01000101
> Module 10 Inquiry
< Acknowledgement
...
```

Všimněte si, že probíhá polling pouze některých adres – pro zmenšení latencí je vhodné často dotazovat aktivní moduly a neaktivní moduly dotazovat jen čas od času (aby šlo poznat, že moduly ožily). Dále si všimněte, že příkazy pro moduly (*Set Outputs*) se posílají nezávisle na tom, který modul je právě dotazován.

Komunikační protokol

Každá *zpráva* se skládá z následujících bytů:

1. **Adresa modulu.**

   Tento byte má jako jediný nastavený svůj 9. bit na hodnotu 1. Adresa modulu je 8bitové číslo. Adresa 0 znamená, že zpráva se posílá jako *broadcast*, tedy ji mají číst všechny moduly na sběrnici. Typickou zprávou typu *broadcast* je požadavek na reset výstupů všech modulů.

   Adresní prostor 255 modulů byl vyhodnocen jako dostatečně velký.

2. **Délka zprávy.**

   Tento byte obsahuje počet *datových bytů zprávy* plus jedna (pro *kód zprávy*).

   Pro univerzálnost protokolu bylo rozhodnuto jej navrhnout tak, aby délka zpráv nebyla součástí definice jednotlivých zpráv, ale aby byla ve zprávě obsažena explicitně. Toto rozhodnutí umožňuje například MTB-USB modulu přeposílat zprávy mezi MTBbus a počítačem, aniž by musel dekódovat obsah zpráv, případně mít uloženou tabulku *typ zprávy, délka*. Některé zprávy navíc mohou mít variabilní délku ze své podstaty.

   Maximální hodnota tohoto bytu je 121, což umožňuje zprávy bezpečně skládat například do bufferů o délce 128 bytů. Maximální délka zprávy byla zvolena s ohledem na nevelké kapacity *SRAM* mikrokontrolérů a časovou efektivitu případné retransmise.

   Současná verze protokolu MTBbus umožňuje zprávy o délce nejvýše 7 bytů [6]. Tato hodnota byla zásadně navýšena zejména proto, aby bylo možné novým protokolem posílat firmware pro aktualizaci MTB modulů.

3. **Kód zprávy.**

   Libovolné číslo v rozsahu 0–255, definuje význam zprávy. Význam jednotlivých kódů definuje specifikace protokolu.

4. **Data zprávy.**

   Až 120 bytů.

5. **Kontrolní součet** (2 byty).

   Kontrolní součet se ve zprávě uvádí, aby bylo možné ověřit integritu přijímané zprávy. Integrita zprávy může být narušena například elektromagnetickým rušením nebo špatnými elektrickými vlastnostmi přenosového média.

   Původní protokol MTBbus zabezpečuje integritu zprávy pomocí jednoduchého mechanismu *xor*: na jednotlivé byty zprávy je aplikován bitový *xor* a výsledná hodnota je připojena na konec zprávy. Tento způsob je vhodný pro malé zprávy.

Nová verze protokolu umožňuje delší zprávy, a tak byl navržen robustnější způsob zabezpečení. Sběrnice MTBbus v4 používá k zabezpečení integrity zprávy standardní mechanismus *CRC-16*. Tento mechanismus je založen na dělení polynomů se zbytkem. Konkrétně se používá verze `CRC-16-IBM`. Autor se inspiroval průmyslovou sběrnicí *ModBus*, která používá právě tento způsob [21].

Příklad zprávy: `0x105 0x001 0x001 0x0D0 0x051`.

1. `0x105`: zpráva pro modul s adresou 5.
2. `0x001`: následuje 1 byte zprávy.
3. `0x001`: kód zprávy `0x01`.
4. `0x0D0`, `0x051`: kontrolní součet *CRC-16*.

Zpráva neobsahuje žádná *data zprávy*.

Zprávy

Kompletní specifikace významů a kódů zpráv je dostupná online[11], zde uvedeme stručný přehled.

Zprávy master → slave spolu s jejich *kódy*:

- `0x01` *Module Inquiry*
- `0x02` *Module Information Request*
- `0x03` *Set Configuration*
- `0x04` *Get Configuration*
- `0x05` *Beacon*
- `0x10` *Get Input*
- `0x11` *Set Output*
- `0x12` *Reset Outputs*
- `0x20` *Change Address*
- `0xE0` *Change Speed*
- `0xF0` *Firmware Upgrade Request*
- `0xF1` *Firmware Write Flash*

11. `https://github.com/kmzbrnoI/mtbbus-protocol`

- `0xF2` *Firmware Write Flash Status Request*
- `0xFE` *Module-specific Command*
- `0xFF` *Reboot*

Zprávy slave → master spolu s jejich *kódy*:

- `0x01` *Acknowledgement*
- `0x02` *Error*
- `0x03` *Module Information*
- `0x04` *Module Configuration*
- `0x10` *Input Changed*
- `0x11` *Input State*
- `0x12` *Output Set*
- `0xF2` *Firmware Write Flash Status*
- `0xFE` *Module-specific Command*

Rozeberme nyní zprávy detailněji.

MTB-USB průběžně skenuje aktivní i neaktivní moduly (aktivní častěji) – posílá *Module Inquiry*. Moduly odpovídají buď *Acknowledgement*, nebo *Input Changed*.

Pomocí zpráv *Module Information Request* a *Module Information* lze zjistit základní informace o MTB modulu: jeho typ, verzi firmwaru, podporovanou verzi protokolu a jestli je modul v bootloaderu.

Zpráva *Set Output* slouží k nastavení stavu výstupů modulu, zpráva *Reset Outputs* slouží k navrácení výstupů do základního stavu. Tato zpráva je typicky odesílána MTB-USB modulem při ukončení komunikace s počítačem jako broadcast.

Počítač může zapisovat nebo číst konfiguraci modulů pomocí zpráv *Get Configuration*, *Set Configuration* a odpovědi *Module Configuration*.

Počítač může zapínat a vypínat indikační LED na MTB modulech pomocí zprávy *Beacon*.

Modulům, které si ukládají adresu softwarově, může operátor z počítače změnit adresu pomocí zprávy *Change Address*.

Plánovaná změna rychlosti sběrnice se modulům ohlašuje pomocí zprávy *Change Speed*, která je typicky odesílána jako broadcast. Každý

modul si ukládá rychlost sběrnice do nevolatilní paměti, uloženou rychlost použije při zapnutí modulu. Očekává se, že budoucí typy procesorů v modulech MTB budou schopny detekovat rychlost sběrnice automaticky.

V současnosti není na sběrnici MTBbus implementován žádný mechanismus pro automatickou detekci rychlosti. V situaci, kdy je například do sběrnice doplněn nový modul, který je ale z výroby nastaven na jinou rychlost, než aktuálně sběrnice používá, se očekává, že operátor přenastaví rychlost sběrnice na rychlost nového modulu, odešle mu příkaz na změnu rychlosti a vrátí rychlost sběrnice zpět. Tyto kroky lze provést za plného chodu sběrnice.

Popisu procesu aktualizace firmwaru modulů se budeme věnovat v samostatné kapitole.

Poslední zajímavou zprávou je *Module-specific command*. Tato zpráva umožňuje odeslat MTB modulu libovolná data (a modul může odpovědět libovolnými daty). *Module-specific command* se používá například pro řízení specializovaných operací modulů, viz 5.3.

Dvouvrstvost protokolu

Specifikace MTBbus v4 je tzv. *dvouvrstvá*. Protokol definuje typy zpráv, ale data některých zpráv specifikují až konkrétní typy MTB modulů. Typ modulu definuje:

1. **Formát dat vstupů, formát dat výstupů**

   Protokol je navržený tak, aby moduly mohly mít libovolné typy a počty vstupů i výstupů. Obsah datových bytů zpráv *Set Output*, *Input Changed* a *Input State* je proto specifikovaný až ve specifikaci modulů.

   Například modul MTB-UNI má 16 digitálních vstupů a 16 digitálních výstupů. Protože počet vstupů i výstupů je malý, posílá se vždy celý stav.

   Protokol počítá s tím, že jiné moduly mohou mít vstupy a výstupy například s větším rozsahem hodnot (analogové vstupy). V takovém případě protokol umožňuje například neposílat celý stav,

ale jen relevantní část. To je také důvodem, proč existují 2 různé zprávy *Input Changed* (posílá se jen stav změněných vstupů) a *Input State* (posílá se stav všech vstupů). [12]

2. **Formát konfigurace**

   Konfigurace různých typů modulů se mohou zásadně lišit v počtu konfigurovatelných hodnot.

   Například modul MTB-UNI ma 26 konfiguračních bytů. Proto se celá konfigurace posílá najednou. Protokol však umožňuje libovolný způsob zasílání konfigurace – počítač si například u modulů s větším množstvím konfiguračních bytů může vyžádat pouze podmnožinu konfiguračních hodnot. Nebo naopak poslat podmnožinu konfiguračních adres a hodnoty na těchto adresách k nastavení.

3. **Adresování a data paměti pro aktualizaci firmwaru**

   Různé typy modulů typicky obsahují různé procesory, které jinak adresují paměť. Liší se například šířka slova nebo velikost paměti. Význam datových bytů zpráv pracujících s aktualizací firmwaru procesoru je proto definován pro konkrétní typy modulů.

Typické fungování sběrnice

Následující kroky popisují typické fungování systému MTB v4 (srovnejte s postupem definovaným na konci 2.1).

1. Je zapnut řídicí počítač kolejiště.
2. Je zapnuto napájení MTB-USB modulu a všech MTB modulů.
3. MTB moduly z paměti načtou konfiguraci, aplikují základní stav výstupů, začnou poslouchat na sběrnici.
4. MTB-USB detekuje připojené moduly a zapamatuje si je.
5. Počítačová aplikace se připojí k MTB-USB.
6. Počítačová aplikace vyčte z MTB-USB seznam aktivních modulů.

12. Všimněte si, že tento mechanismus umožňuje provádět polling vstupů místo událostního hlášení změn stavů vstupů. To je zcela záměrný prvek MTBbus v4. Využije se například u budoucích modulů s analogovými vstupy.

7. Aplikace vyčte typ připojených modulů, zkonfiguruje je.
8. Řídicí SW kolejiště používá sběrnici – čte vstupy, nastavuje výstupy.
9. Při odpojení řídicího SW kolejiště je MTB modulům poslán příkaz k resetu stavu výstupů, čímž je zajištěno, že při odpojení nebo pádu řídicího softwaru se budou nacházet výstupy v definovaném stavu.

MTB v4 však nevyžaduje striktní pořadí těchto kroků. Napájení komponent může být zapnuto v libovolném pořadí, systém dokáže detekovat výpadky jednotlivých prvků a obnovovat je. Pro ilustraci uveďme několik příkladů situací, které v systému MTB v4 mohou nastat.

- **Jiné pořadí zapnutí prvků.**

  Pokud je MTB-USB zapnut až po zapnutí počítače, počítač tento stav korektně detekuje (chybí zařízení) a po připojení MTB-USB jej korektně inicializuje.

  Sken všech adres sběrnice MTBbus (255 adres) probíhá neustále, takže pokud jsou moduly sběrnice zapnuty až po zapnutí MTB-USB, jsou korektně detekovány.

- **Výpadek MTB modulu.**

  Pokud MTB modul neodpovídá na zprávy, MTB-USB situaci detekuje, nahlásí ji do počítače a počítač zareaguje.

  Pokud MTB-USB detekuje nový MTB modul, opět o tom zpraví počítač, ten vyčte typ modulu a zkonfiguruje jej.

- **Odpojení a připojení modulu za chodu sběrnice (např. výměna).**

  Vyňatý modul je při odpojení napájení nahlášen jako neaktivní, po výměně je aktivován nový modul.

- **Změna adresy modulu za chodu sběrnice.**

  Původní adresa se stane neaktivní (neodpovídá na požadavky), nová adresa je nalezena a zkonfigurována.

Aktualizace firmwaru MTB modulů

Sběrnice MTBbus je navržena tak, aby skrze ni mohl být aktualizován firmware MTB modulů. Při návrhu mechanismu aktualizace firmwaru bylo zvažováno, jestli aktualizaci nevést úplně jiným protokolem. MTBbus je však dostatečně obecný a navíc je třeba, aby při aktualizaci jednoho modulu ostatní moduly sběrnice aktualizační příkazy ignorovaly. Tyto argumenty vedly k implementaci mechanismu aktualizace firmwaru přímo do sběrnice MTBbus. Výhodou tohoto řešení je, že ostatní moduly sběrnice mohou při změně firmwaru jednoho modulu nadále komunikovat, aktualizace může probíhat za plného provozu.

Protokol počítá s tím, že přepis flash paměti procesorů MTB modulů může probíhat pouze z bootloaderu. Aktualizační protokol proto podporuje reboot procesoru do bootloaderu. Procedura aktualizace firmwaru probíhá následovně [13].

1. MTB-USB vyšle *Firmware Upgrade Request*. Tím informuje MTB modul, že má rebootovat do bootloaderu.
2. MTB odpoví *Acknowldgement* a rebootuje do bootloaderu.
3. MTB-USB pošle MTB modulu žádost o jeho obecné informace.
4. MTB odpoví obecnými informacemi, ve kterých je mj. zaznačeno, že je v bootloaderu.
5. MTB-USB pošle příkaz *Firmware Write Flash*, ve kterém pošle blok flash paměti k zapsání a její adresu. MTB zapíše data do své flash paměti.
6. MTB-USB se zeptá MTB modulu příkazem *Firmware Write Flash Status Request*, jestli již byl zápis do flash paměti dokončen. Jakmile je zápis dokončen, přesouvá se na zápis dalšího bloku paměti (krok 5).
7. Po zapsání celého firmwaru MTB-USB pošle příkaz *Reboot*. MTB modul provede kontrolu konzistence paměti a naběhne do hlavního programu.

13. Procedura je podrobně popsaná v dokumentaci protokolu MTBbus na `https://github.com/kmzbrnoI/mtbbus-protocol/blob/master/workflows.md`.

Je doporučeno (a nový firmware MTB-UNI modulů toto doporučení dodržuje), aby při každém startu procesoru MTB modulu proběhla kontrola konzistence paměti programu.

Kontrola konzistence v MTB-UNI v4 modulech je implementována pomocí uloženého kontrolního součtu (CRC-16) na konkrétních adresách paměti. Protokol MTBbus obsahuje atribut, kterým může MTB modul oznámit, že sice poslouchá, ale je v bootloaderu a odmítá nabootovat, protože nesedí kontrolní součet hlavního programu.

Aktualizace bootloaderu přes sběrnici MTBbus není možná, protože vlivem výpadku napájení při aktualizaci by mohlo dojít k situaci, kdy se modul stane nepoužitelným. Bootloader by měl být malý a odzkoušený kus firmwaru, který se nahraje jednou a nikdy ho nebude třeba přehrávat. Přehrání bootloaderu vyžaduje přímé programování modulu.

5 Implementace MTB v4

5.1 MTB-USB v4

Obrázek 5.1: Prototyp modulu MTB-USB.

MTB-USB modul vzniknuvší v rámci této práce byl navržen od základu znovu, inspirace současným MTB-USB modulem je minimální. Vznikly hardware, firmware a popisy komunikačních protokolů.

Hlavním úkolem MTB-USB modulu je přeposílat data mezi sběrnicí MTBbus a počítačem. MTB-USB provádí časově kritické operace sběrnice – například počítání timeoutu odpovědi na zprávu od MTB modulů nebo pravidelné dotazování MTB modulů. Počítač je o dění na MTBbus informován formou událostí.

5.1.1 Komunikační protokol s počítačem

Před implementací MTB-USB modulu je třeba navrhnout komunikační protokol s počítačem. Tento komunikační protokol musí být přirozeně jiný, než nový komunikační protokol sběrnice MTBbus, který jsme popsali v předchozí kapitole, protože zahrnuje jiné aktéry a funguje na jiném hardwaru.

Komunikační protokol MTB-USB a počítače je navržen tak, aby byl do velké míry nezávislý na protokolu sběrnice MTBbus. Nejdůležitější zprávou mezi počítačem a MTB-USB je požadavek o přeposlání zprávy mezi USB a MTBbus. Tato zpráva umožňuje počítačovému programu poslat libovolnou zprávu libovolnému MTB modulu (a naopak), aniž by MTB-USB deska musela znát sémantiku této zprávy.

Na MTB-USB desku můžeme tedy pohlížet v zásadě jako na tenkého přeposílatele mezi dvěma různými sběrnicemi – tzv. *gateway*.

Plnohodnotná specifikace protokolu mezi počítačem a MTB-USB je k dispozici online[1]. Popišme nyní stručně návrh protokolu.

Mezi počítačem a MTB-USB se komunikuje po virtuálním sériovém portu (CDC) tunelovaným skrze USB. Toto řešení bylo vybráno, protože je prakticky standardem pro připojení speciálních periferií k počítači (zvažován byl také *HID*). USB nezná pojem zprávy tak, jak bychom vyžadovali[2]. Protože USB podporuje pouze 8bitový sériový port, začátek zprávy je třeba označit jiným způsobem než pomocí 9. bitu. V navrženém protokolu je začátek zprávy označen speciální sekvencí dvou bytů `0x2A` a `0x42`. Tato sekvence se sice ve zprávě může objevit, ale pravděpodobnost, že dojde k tolika chybám, aby tato sekvence uprostřed zprávy byla považována za začátek zprávy, je malá.[3]

Každá zpráva mezi počítačem a MTB-USB má následující strukturu.

1. `0x2A`,
2. `0x42`,
3. počet následujících bytů,
4. kód zprávy,
5. data zprávy (až 122 bytů).

---

1. `https://github.com/kmzbrnoI/mtbbus-protocol/tree/master/pc`
2. Při užití třídy CDC se data posílají nejčastěji každou milisekundu a to v bloku o nejvýše 64 bytech. Počítačová aplikace díky bufferování operačního systému není schopna tyto bloky spolehlivě rozlišit.
3. Rozdělování dat do zpráv na straně přijímače se nemusí řídit jen detekcí sekvence `0x2A 0x42`. Pokud zprávy nechodí moc často, lze po delší době nepřijímání dat (řádově jednotky milisekund) vyprázdnit vstupní buffer a očekávat tedy, že další příchozí data budou novou zprávou. Odesílatel zprávy nesmí posílat jednotlivé části zprávy s velkou prodlevou, to je ale splnitelný požadavek.

Struktura je podobná struktuře zprávy protokolu MTBbus, viz 4.2.2. Zpráva neobsahuje kontrolní součet, protože integritu zprávy řeší přímo USB. Upozorňujeme, že *kód zprávy* není kód zprávy MTBbus, ale kód zprávy protokolu PC – MTB-USB.

Zprávy z počítače pro MTB-USB jsou:

- **Forward packet to MTBbus**

  V *datech zprávy* následují adresa a zpráva pro MTB modul.

- **MTB-USB Information Request**

- **Change Speed**

  Požadavek na změnu komunikační rychlosti MTB-USB modulu. Změnu rychlosti jednotlivých MTB modulů je třeba provést předchozím příkazem, typicky broadcastem všem MTB modulům.

- **Active modules request**

  Odpovědí na tento příkaz je seznam aktivních MTB modulů.

- **Ping**

Zprávy z MTB-USB pro počítač jsou:

- **Acknowledgement**

- **Error**

- **Packet from MTBbus**

  Zpráva je odeslána počítači při odpovědi MTB modulu na zprávu z počítače nebo na pravidelný sken MTB modulů. Z pohledu počítače tak přichází jak odpovědi na příkazy pro MTB modul, které poslal počítač, tak asynchronní události – informace o změně stavu vstupů.

- **MTB-USB Information**

- **Active modules list**

- **New module discovered event**
- **Module failed event**

MTB-USB si udržuje seznam aktivních MTB modulů. Polling modulů probíhá v iteracích. V každé iteraci jsou osloveny všechny aktivní moduly a 10 neaktivních modulů. Tím je zaručeno, že aktivní moduly jsou skenovány často a zároveň jsou detekovány moduly nové.

Pokud MTB modul neodpoví na *Module Inquiry* (4.2.2), počítači je odeslána zpráva *Module failed event*. Modul je považován za ztracený, jakmile neodpoví na *Module Inquiry* ve třech po sobě jdoucích iteracích. Zpráva *Module failed event* je při úplném výpadku modulu odeslána celkem třikrát proto, aby bylo možné z počítače monitorovat chod sběrnice. Občasné neodpovídání modulů na výzvy je dobrým indikátorem problémů sběrnice.

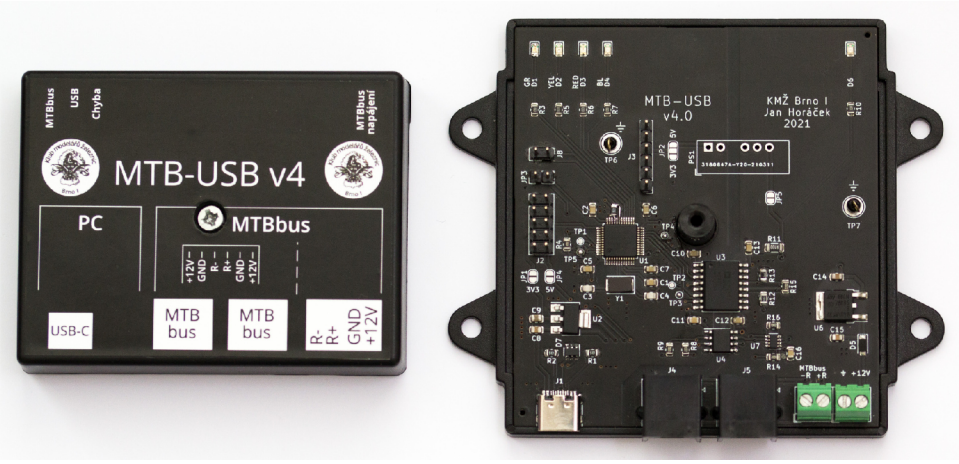
Vlastností MTBbus je, že každý MTB modul musí vždy odpovědět na každou zprávu, kterou přijme[4]. Protože MTBbus je potenciálně nespolehlivé médium, na kterém integritu příkazů zajišťujeme vlastními mechanismy, provádí modul MTB-USB retransmisi zpráv MTB modulům v případě, že na zprávu nepřijde žádná odpověď, a to až třikrát. Proto je součástí zprávy *Packet from MTBbus* také počítadlo, které říká, na kolikátý pokus byla odpověď přijata (u asynchronních událostí 0). Toto číslo bylo do protokolu vloženo opět se záměrem, aby bylo možné v počítači monitorovat chod sběrnice.

Na dalších příkazech protokolu není vcelku nic zajímavého, pokud je čtenář chce prostudovat, je mu k dispozici plná specifikace protokolu[5].

5.1.2 Hardware

Srdcem MTB-USB modulu je procesor `STM32F103`, což je moderní mikrokontrolér architektury *ARM*. Autor jej zvolil z několika důvodů.

4. Výjimkou jsou pouze *broadcast* zprávy.
5. `https://github.com/kmzbrnoI/mtbbus-protocol/tree/master/pc`



Obrázek 5.2: Krabička a deska plošných spojů MTB-USB.

1. Procesor `STM32F103` má hardwarovou podporu USB.
2. Procesory `STM32` nabízí velké velikosti pamětí a výpočetní výkon.
3. Procesory `STM32` se stávají standardy v embedded systémech.
4. Procesory `STM32` se dají velice snadno debugovat.
5. Procesor `STM32F103` je jako jeden z mála `STM` součástí *basic* součástek na `https://jlcpcb.com/`.
6. Autor se chtěl naučit pracovat s novou architekturou procesorů.

Hardwarová podpora USB umožňuje výrazně vyšší flexibilitu komunikace s počítačem než použití historicky zaužívaných převodních obvodů mezi USB a sériovou linkou. V programu procesoru je tak například možné definovat, že procesor má více CDC linek – druhá se využije například pro ladění programu. Procesor může v budoucnu používat libovolnou třídu USB. Celé řešení se tak stává mnohem lépe upravitelné pouze změnou softwaru, což je méně pracné, než změna hardwaru.

Všechny desky plošných spojů (*DPS*) navržené v rámci této práce jsou vytvořeny tak, aby se daly automaticky osazovat (firmou *JLCPCB*[6]). *JLCPCB* nabízí automatické osazování malých sérií DPS za dostupnou cenu, což výrazně zjednodušuje výrobu nových komponent. *JLCPCB* rozlišuje tzv. *basic* a *extended* součástky k osazení, přičemž za *basic* se

6. Aktuálně lze osazovat pouze jednu stranu desky a pouze SMD součástky.

neplatí režijní poplatek při osazování. *Basic* součástky jsou ty, které se používají skutečně často (typicky diskrétní součástky – rezistory, kondenzátory apod.) a u kterých se očekává, že se budou vyrábět i za desítky let. Procesor `STM32F103` byl zvolen mj. proto, že je *basic* součástkou.

Schéma a výkres DPS byly vytvořeny v nástroji *KiCad*, který autor této práce používal poprvé. Jedním z hlavních přínosů celé této práce pro něj je, že se naučil pracovat s programem *KiCad*. Schéma a výkres DPS jsou k dispozici online[7], schéma je přiloženo také jako příloha této práce (A.1).

Popišme nyní stručné schéma. Schéma je na první pohled rozděleno na 2 části, které jsou na DPS galvanicky oddělené – USB část (vlevo) a MTBbus část (vpravo). USB část obsahuje procesor a je napájena přímo z USB. MTBbus část obsahuje rozhraní sběrnice MTBbus a je napájena buď z externího zdroje (ze stejného jako zbytek sběrnice MTBbus), nebo přes galvanicky oddělený měnič `PS1`. Osazením nebo neosazením součástek se zvolí, která varianta se bude používat.

Hlavním prvkem USB části schématu je procesor `STM32F103` (vlevo nahoře). K procesoru jsou připojena diagnostická rozhraní (`J2`, `J3`). Celá USB část je napájena z USB portu, přičemž je využito moderního konektoru USB-C.

Rozhraní USB a MTBbus části schématu tvoří galvanicky oddělený budič sběrnice RS485 typu `ADM2483`. `ADM2483` je osvědčený budič, který zvládá proudy sběrnice až do 250 mA [22] a je tedy vhodný pro komunikaci s větším počtem MTB modulů.

Specialitou MTB-USB modulu je měřící obvod `INA219` ve sběrnicové části desky, který měří napětí a proud do sběrnicové části obvodu `ADM2483` a posílá naměřenou hodnotu přes galvanický oddělovač do procesoru. Procesor je tak schopen detekovat nestandardní chování sběrnice, například zkrat.[8]

Při návrhu desky plošných spojů bylo třeba vyřešit především to, do jaké krabičky desku vložit. MTB-USB modul totiž typicky není pevnou

7. `https://github.com/kmzbrnoI/mtb-usb-4-pcb`
8. Pro měření zatím není podpora v komunikačním protokolu s počítačem a ve firmwaru. Hardware je odzkoušený. Podporu je v plánu doplnit v další verzi.

součástí kolejiště, je umístěn vedle kolejiště u řídicího serveru. Vyvstal tak požadavek modul zavřít do krabičky, ideálně s možností uchycení na zeď. Po rešerši byla zvolena krabička firmy *Digikeijs*, jejíž zásadní výhodou je, že pro vyvedení konektorů a indikačních LED není třeba do krabičky frézovat. Navíc lze na krabičku velice elegantně nalepit potisk, který vysvětluje použití jednotlivých konektorů a význam LED. Celé řešení je zobrazeno na obrázcích 5.1 a 5.2.

5.1.3 Firmware

Firmware je psaný v jazyce `C`, který byl zvolen především proto, že ve stejném jazyce je psána základní knihovna pro interakci s periferiemi procesoru *STM32 HAL* [23], kterou firmware využívá.

Při práci s periferiemi je využito *Direct Memory Access* (*DMA*). Pro komunikaci přes USB byla využita komunitní knihovna `libusb_stm32`[9].

Celý firmware je k dispozici pod opensource licencí online[10].

9. `https://github.com/dmitrystu/libusb_stm32`
10. `https://github.com/kmzbrnoI/mtb-usb-4-fw`

5.2 MTB-UNI v4

Další komponentou nového systému MTB je nový modul typu MTB-UNI. Tento modul vznikl, aby bylo možné provádět nové instalace systému MTB a rozšiřovat současné instalace o nové moduly.

5.2.1 Základní parametry

Tabulka 5.1: Základní parametry modulu MTB-UNI v4

Vstupy	16 digitálních vstupů
Výstupy	16 výstupů v režimu otevřeného kolektoru, všechny umožňují kmitání a generování S-COM signálu
Napájení	7–17 V DC
Adresování	pomocí jumperů
Procesor	ATmega128

Aktuální modul MTB-UNI má 16 digitálních vstupů a 16 digitálních výstupů. Tento počet se autor práce rozhodl zachovat, protože vhodně škáluje pro malé i velké železniční stanice na modelových kolejištích. Současné MTB-UNI a MTB-TTL moduly používají různé konektory pro připojení periferií: svorkovnice nebo nasouvací konektory typu `PSH`[11]. Modul MTB-UNI v4 umožňuje variantní osazení jak svorkovnic, tak konektorů `PSH`. Tím modul slučuje současné moduly MTB-UNI, MTB-UNIm a MTB-TTL do jediné DPS, což zjednodušuje údržbu a vývoj.

Hardwarové řešení výstupů je ponecháno: výstupy jsou v režimu otevřených kolektorů, jsou řízeny skrze obvody `ULN2803`, které umožňují dostatečný výstupní proud až 0.5 A / 8 výstupů. Všechny výstupy umožňují kmitání i vysílání signálu S-COM.

Modul obsahuje 16 digitálních vstupů. Podporu IR čidel řeší samostatná deska, viz 5.4.

11. Např. `https://www.gme.cz/konektor-se-zamkem-psh02-04pg`.

Deska plošných spojů je navržena tak, aby byly zachovány rozměry a umístění upevňovacích otvorů se současným nejmenším modulem – MTB-TTL.

Návrh této desky vznikal ve spolupráci s MENDELU. Schéma vytvořil Robert Čížek, DPS, firmware a koncepční návrh modulu jsou dílem autora této práce.

5.2.2 Hardware

Při návrhu modulu připadají v úvahu dva přístupy:

1. malý procesor, posuvné registry na vstupy a výstupy;
2. velký procesor, vstupy a výstupy připojené přímo na piny procesoru.

Autor této práce zvolil přístup (2), protože chtěl minimalizovat počet součástek na desce a protože v dnešní době jsou malé procesory s větším počtem IO pinů již běžně dostupné.

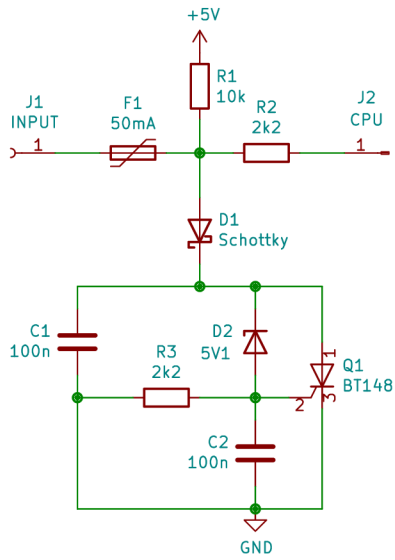
Schéma i DPS jsou vytvořeny v programu *Eagle* jako openhardware projekt, dostupné online[12], schéma také v příloze A.2. Okomentujme nyní zajímavé prvky schématu a DPS.

Jádrem modulu je procesor *ATmega128*. Autor této práce si zvolil procesor architektury *AVR*, protože s používáním těchto procesorů má dlouholeté zkušenosti. Model *ATmega128* byl pak zvolen proto, že je to nejmenší model rodiny *ATmega*, který obsahuje požadované množství pinů.

Vstupy

MTB-UNI v4 modul oproti modulu současnému obsahuje řádnou ochranu vstupů. Ochranný obvod vznikl ve spolupráci Jana Horáčka, Michala Petrilaka a Roberta Čížka. Zapojení jednoho vstupu je demonstrováno na obrázku 5.3.

12. `https://github.com/kmzbrnoI/mtb-uni-4-ele`



Obrázek 5.3: Zapojení vstupu modulu MTB-UNI.

Schéma 5.3 je rozděleno na 2 části Schottkyho diodou D1. Horní část včetně diody D1 je pro každý z 16 vstupních pinů zvlášť, spodní je vždy pro osmici pinů společná.

Očekává se, že periferie připojená k pinu (INPUT) bude tento pin uzemňovat, případně že bude pin v režimu TTL. Každý vstup obsahuje pull-up rezistor R1, aby měl vstupní signál vždy definovanou napěťovou úroveň. Pull-up rezistor je relativně tvrdý, aby se zabránilo ovlivňování vstupů okolním rušením, které generuje signál DCC při přenášení větších proudů.

K procesoru vede signál přes ochrannou pojistku F1 a ochranný rezistor R2. Tento rezistor zabraňuje nadproudu do pinu procesoru a tím jeho zničení.

Funkcí celého obvodu od D1 níže je vyzkratovat vstup na *GND* v momentě, kdy je na něj přivedené vyšší než dovolené napětí. Pin procesoru povoluje napětí nejvýše 5.5 V [24]. Jakmile se vstupní napětí začne blížit 5.4 V (5.1 V Zenerova dioda D2 + 0.3 V Schottkyho dioda D1), začne se dioda D2 otevírat a tím začne spínat i tyristor Q1.[13]

13. V praxi tato situace nastává už při trochu nižším napětí, cca 5.3 V, protože pro sepnutí Q1 stačí, aby se D2 otevřela jen nepatrně.

Vstupní proud proteče přes `Q1` do *GND* a tím je vstup zkratován. Proud je limitován pojistkou `F1`. Tyristor `Q1` je dimenzován jako výkonový, protože přes něj může téct zkratový proud všemi 8 piny, navíc chování polymerové pojistky umožňuje krátkodobé řádově větší proudy, které musí tyristor zvládnout.

V praxi se často používá zapojení *vstup – ochranný rezistor – pull-up – pin procesoru*. V zapojení MTB-UNI desky se oproti tomu používá zapojení *vstup – pull-up – ochranný rezistor – pin procesoru*. Výhodou zapojení, které bylo implementováno do MTB-UNI modulu, je, že na vstupu procesoru není napěťový dělič. V případě, že vstup má například napětí 0.7 V (což se snadno stane například tehdy, když periferie používá výstupy typu otevřené kolektory), je toto napětí přivedeno přímo na pin procesoru a tedy spolehlivě detekováno jako logická jednička[14]. Při zapojení prvního typu by na pinu procesoru vznikl napěťový dělič, který by mohl ohrozit spolehlivost čtení logického stavu pinu.

Výstupy

Na výstupech se používají již zmíněné obvody `ULN2803`. Za obvody je vřazena polymerová pojistka. Protože obvod `ULN2803` specifikuje proudové omezení 0.5 A / všechny výstupy [25], nemohou samostatné pojistky na jednotlivých výstupech zajistit nepřetížení obvodu a zároveň umožnit využití maximálního proudu. Proto jsou na DPS MTB-UNI v4 obvody `ULN2803` jako jedny z mála osazeny v paticích v *DIL* provedení.

Výstupy obsahují ochranu proti přivedení vysokého napětí, kterou realizuje stejný obvod, jako na obrázku 5.3 (obvod pod diodou `D1`). Obdobný obvod realizuje také ochranu proti vysokému napájecímu napětí celého modulu. Viz kompletní schéma A.2.

14. V MTB-UNI je detekováno jako logická jednička, protože modul používá inverzní logiku.

Adresování

Adresa modulu MTB-UNI v4 je určena jumpery na modulu. Při návrhu modulu vznikla diskuze, jaký je nejlepší způsob adresování modulů, přičemž soupeřily přístupy:

1. modul má adresu uloženou v EEPROM, adresu je možné změnit přes MTBbus,
2. adresa se konfiguruje přímo jumpery na modulu.

Druhý přístup, který zvolil autor této práce, si vysloužil mnohou kritiku. Autor si však stojí za tím, že snadná a především vždy pravdivá identifikace adresy modulu prostým pohledem stojí za 8 vstupních pinů procesoru navíc.

Pro elegantní řešení přístupu (1) je třeba mít na desce tlačítko, kterým se spouští změna adresy. Tlačítko se na desce nachází, takže pokud by byla v budoucnu snaha přejít na přístup (1), je změna otázkou změny firmwaru.

5.2.3 Deska plošných spojů

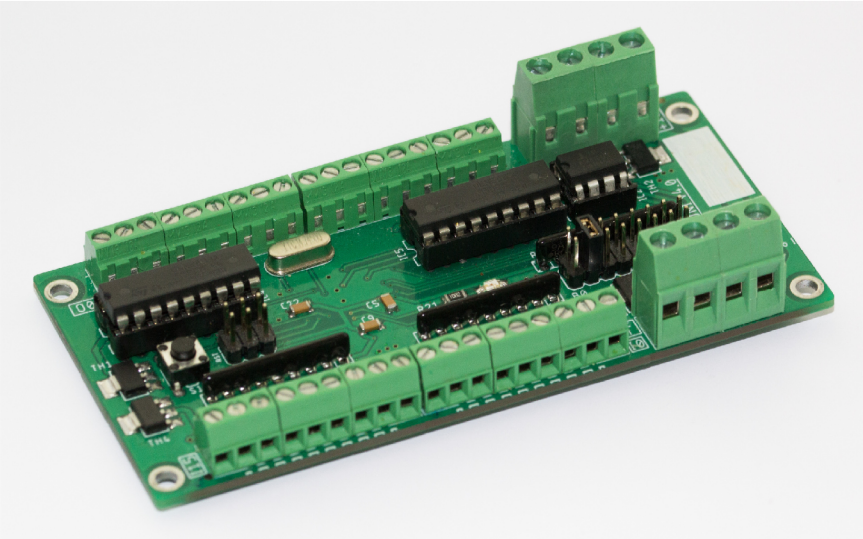
DPS je navržená tak, aby byly zachovány rozměry a umístění upevňovacích otvorů se současným nejmenším modulem – MTB-TTL. Deska je dvouvrstvá, používají se primárně SMD součástky velikosti *0805* umístěné na spodní straně desky (aby bylo možné DPS automaticky osazovat). Na obrázku 5.4 lze vidět prototyp desky MTB-UNI v4.0. Poslední verzí je v4.2, která oproti 4.0 obsahuje více SMD součástek a opravuje chyby prototypu.

5.2.4 Firmware

Firmware pro procesor *ATmega128* je psán v jazyce `C` a je dostupný pod opensource licencí online[15].

Zajímavým prvkem firmwaru je podpora jeho aktualizace přímo přes MTBbus. Firmware obsahuje dvě části:

15. `https://github.com/kmzbrnoI/mtb-uni-4-fw`



Obrázek 5.4: MTB-UNI v4.0.

1. hlavní program,
2. bootloader.

Bootloader je ve speciální části paměti (na konci) a zajišťuje aktualizaci hlavního programu. Bootloader je měnitelný pouze přímým programováním procesoru, nikoliv přes MTBbus. Firmware tak v zásadě obsahuje 2 samostatné programy, které se po zkompilování slinkují dohromady.

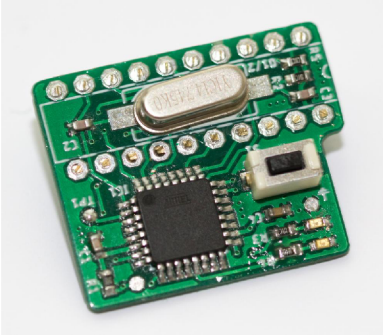
Bootloader umí pouze aktualizovat firmware a kontrolovat jeho konzistenci. Kontrola konzistence firmwaru je implementována pomocí uložení CRC-16 kontrolního součtu a počtu aktivních stránek paměti na konkrétních adresách paměti (na konci těsně před bootloaderem).

5.3 MTB-2-AVR

Aby nebylo nutné všechny současné MTB-UNI moduly vyměnit za moduly nové, je třeba současné moduly povýšit tak, aby podporovaly nový protokol MTBbus v4. Kvůli omezenosti procesoru v současných MTB-UNI není možné tuto úpravu provést pouze změnou firmwaru, viz 2.4.

Procesory v současných MTB-UNI modulech jsou v DIL patici. Nabízí se tedy přirozená cesta najít nový procesor odpovídajícího rozložení pinů. Procesor, který by měl vyhovující rozložení pinů a požadované parametry, bohužel neexistuje (uvažován byl například *ATtiny4313*). Proto se autor této práce vydal cestou výroby nástavné desky, která se zasune do DIL patice současné MTB-UNI desky a která na sobě bude mít SMD procesor.

Popsanou myšlenku realizuje deska *MTB-2-AVR*, viz obrázek 5.5.

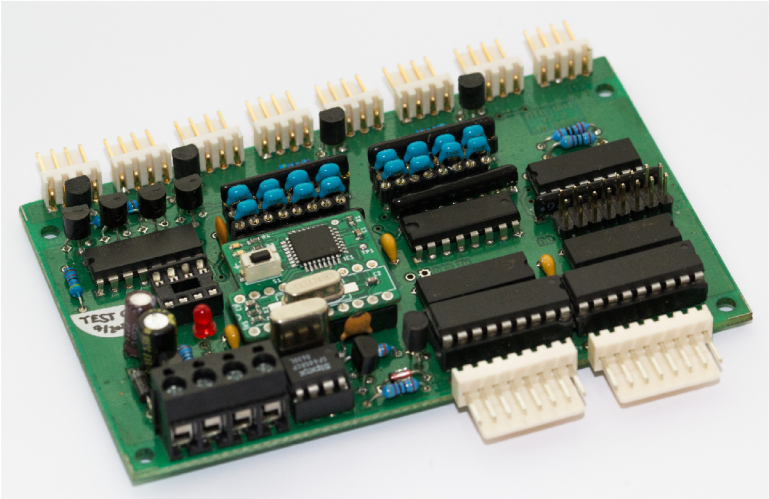


Obrázek 5.5: Nástavná deska *MTB-2-AVR*.

Deska je navržena tak, aby se dala umístit místo procesoru libovolného z modulů MTB-UNI, MTB-UNIm nebo MTB-TTL. Srdcem desky je procesor *ATmega328p* autorovy oblíbené architektury *AVR*. Konkrétně tento model byl zvolen, protože obsahuje dostatečnou kapacitu paměti pro bootloader a protože jej *JLCPCB* má mezi *basic* součástkami (viz 5.1.2).

Schéma a DPS jsou vyvinuty jako openhardware projekt, dostupné online[16] a v příloze A.3. Nástavná DPS obsahuje také LED, aby každá

16. `https://github.com/kmzbrnoI/mtb-2-avr-pcb`



Obrázek 5.6: Nástavná deska *MTB-2-AVR* v modulu MTB-UNI v2.

MTB deska měla zelenou, modrou a červenou LED. Dále obsahuje krystal, protože původní krystal na MTB-UNI má kmitočet mimo rozsah dovolených hodnot pro procesor *ATmega328p*. Nástavná deska dále obsahuje tlačítko (pro unifikaci hardwaru všech MTB-UNI modulů).

Jak již bylo zmíněno, nástavná deska se osazuje do patic současných MTB-UNI, MTB-UNIm i MTB-TTL desek (viz 5.6). Přitom u MTB-UNI modulů by měl procesor podporovat IR čidla na vstupech, u ostatních modulů ne. Deska *MTB-2-AVR* a firmware jsou proto navrženy tak, aby uměly detekovat, v jakém modulu se nachází (byl identifikován a využit HW rozdíl modulů). Podle detekovaného typu modulu procesor buď zapne, nebo vypne podporu IR čidel. Do počítače pak přes MTBbus nahlásí, jestli modul IR čidla podporuje nebo ne.

Zde mimochodem využijeme *Module specific command* (4.2.2). Modulu *MTB-2-AVR* lze poslat příkaz, aby znovu provedl autodetekci typu modulu, do kterého je vložen, případně nastavit typ ručně.

Firmware je pod opensource licencí dostupný online[17]. Firmware se opět skládá z bootloaderu a hlavního programu.

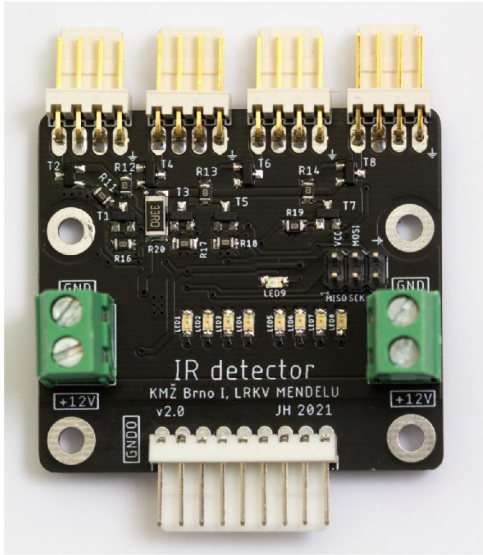
17. `https://github.com/kmzbrnoI/mtb-2-avr-fw`

5.4 IRdet

V rámci této práce vznikla jednoduchá deska, která umožňuje připojení až 8 IR čidel (2.2). Deska hlásí stav IR čidel pomocí opticky oddělených 8 digitálních výstupů. Výstupy lze připojit na vstupy MTB-UNI modulu nebo například k signalizačním diodám v pultu řízení. Deska IRdet není závislá na systému MTB. V této práci ji však uvádíme, protože vznikla především proto, aby bylo možné plnohodnotně nahradit současné MTB-UNI moduly (s přímou podporou IR čidel) použitím nových modulů MTB-UNI v4 a desek IRdet.

Tabulka 5.2: Základní parametry desky *IRdet*

Vstupy	Až 8 IR čidel
Výstupy	8 digitálních opticky oddělených výstupů
Napájení	10–17 V DC
Proud IR diodami	nastavitelný osazením příslušného rezistoru



Obrázek 5.7: Produkční verze desky IRdet

Schéma zapojení a DPS jsou k dispozici jako openhardware projekt online[18], firmware je psán jako opensource projekt a je k dispozici také online[19]. Schéma je dostupné v příloze A.4.

Hlavní komponentou návrhu je procesor řady *Atmega8**. Lze osadit libovolný z procesorů *ATmega8a*, *ATmega88*, *ATmega328* nebo jiný kompatibilní procesor. Signály z fototranzistorů jsou zapojeny v režimu kapacitní vazby, viz 2.3. Vysílací IR diody jsou buzeny přes tranzistory. Proud do vysílacích diod je řízen omezovacím rezistorem na stabilizovaném napětí.

Deska je navržena především z SMD součástek (velikosti *0805*) jako dvouvrstvá s cílem osazovat součástky automaticky. Napájení desky se očekává z napájecího rozvodu MTB modulů (ačkoliv to díky optickému oddělení výstupů není nutné).

18. `https://github.com/kmzbrnoI/irdet-ele`
19. `https://github.com/kmzbrnoI/irdet-fw`

5.5 MTB Daemon

MTB Daemon je první ze dvou počítačových aplikací, která v rámci této práce vznikla. Jejím hlavním úkolem je připojit se k MTB-USB modulu a umožnit jeho ovládání více různým aplikacím. K tomuto účelu vystavuje MTB Daemon TCP server, ke kterému se řídicí programy mohou připojit.

5.5.1 Protokol TCP serveru

TCP server musí umožňovat oboustrannou komunikaci – jak od klienta k serveru (*požadavek, odpověď*), tak od serveru ke klientovi (*asynchronní události*). Například protokol *http* v kombinaci s *REST API* tedy není vhodné použít. Z běžně používaných technologií byly zvažovány *websockets*, ty ale kvůli složitější inicializaci nebyly využity.

Autor zvolil snad nejjednodušší možný komunikační protokol: TCP spojení je udržováno trvale otevřené, posílají se textová data, každý řádek je jedna zpráva. Textová data jsou ve formátu *json*.

Textový protokol byl zvolen, aby bylo možné data číst lidským okem (snadná diagnostika). Očekává se, že řídicí aplikace se bude připojovat převážně ze stejného počítače, na kterém běží MTB Daemon, nemá tedy smysl šetřit „kapacitu linky“ binárním protokolem. Formát *json* byl zvolen proto, že je dnes moderním standardem pro výměnu dat mezi aplikacemi. Snad všechny majoritní programovací jazyky obsahují ve svých standardních knihovnách podporu pro práci s *json*.

Zprávy se posílají v kódování *UTF-8*. Nyní uvedeme příklad dvou zpráv (pro přehlednost je každá zpráva rozdělena na více řádků, ve skutečnosti se posílá na jednom řádku):

```
{
    "command": "mtbusb",
    "type": "request",
    "id": 42
}
```

```
{
    "command": "mtbusb",
    "type": "response",
    "id": 42,
    "status": "ok",
    "mtbusb": {
        "connected": true,
        "type": 1,
        "speed": 115200,
        "firmware_version": "1.0",
        "protocol_version": "1.0",
        "active_modules": [1, 5, 2, 121]
    }
}
```

Zprávy jsou tří typů (`type`):

1. `request` (zpráva od klienta serveru),
2. `response` (odpověď na `request`),
3. `event` (asynchronní událost – od serveru klientovi).

Aby klient věděl, která odpověď patří kterému požadavku (klient může posílat více požadavků rychle za sebou), lze u každého požadavku uvést `id` (číslo). Příslušná odpověď pak obsahuje stejné `id`.

Klient se může zaregistrovat ke konkrétním modulům příkazem `module_subscribe` (a odregistrovat příkazem `module_unsubscribe`). Pokud je klient zaregistrovaný k nějakému modulu, chodí mu asynchronní události o změně stavu modulu (např. změna stavu vstupu). Klienti si tak mohou volit, o data kterých MTB modulů mají zájem.

Kompletní specifikace protokolu je k dispozici online[20].

5.5.2 Volba nástrojů

MTB Daemon je klíčová aplikace infrastruktury řízení kolejiště. Proto se autor rozhodl ji programovat v programovacím jazyce se silným statickým typovým systémem.

20. `https://github.com/kmzbrnoI/mtb-daemon/tree/master/tcp-protocol`

Jako programovací jazyk bylo zvoleno `C++` (`C++17`) s frameworkem `Qt`, který nabízí přímočaré knihovny pro komunikaci se sériovým portem a pro síťovou komunikaci. U knihoven zejména není třeba složitěji řešit vlákna. `C++` a `Qt` umožňují multiplatformní řešení aplikace – MTB Daemon lze zkompilovat a spustit jak na OS Linux, tak OS Windows.

5.5.3 Implementace

Aplikace MTB Daemon je psána jako jednovláknová konzolová aplikace. Při návrhu byla zvažována i možnost implementovat aplikaci jako okenní (operátor obsluhy by přes aplikaci konfiguroval MTB moduly a měl by přehled o stavu sběrnice), nakonec však byl zvolen přístup, kdy aplikace pouze vystavuje API. Konfigurace se načítá ze souboru. V případě zájmu o vytvoření grafického nástroje pro interakci se systémem MTB bude tento systém vytvořen jako další z klientů aplikace MTB Daemon nebo jako nástroj pro přímou úpravu konfiguračního souboru.

Při implementaci bylo třeba vyřešit, jak zaručit, aby sběrnici nemohli ovládat neautorizovaní klienti. V současnosti je implementován pouze velice hrubý autorizační mechanismus – server poslouchá na specifických rozhraních, typicky jenom *localhost*. Očekává se, že lokální stroj je server kolejiště, na kterém by měly běžet výhradně bezpečné aplikace. V budoucnu možná autor přistoupí k implementaci autentizace založené na povolení konkrétních IP adres klientů. Očekává se, že klienti aplikace MTB Daemon jsou trvale připojená, nemigrující síťová zařízení, která řídí nějaký konkrétní systém kolejiště, proto není nutné implementovat žádný složitější autentizační mechanismus.

Jednotliví klienti mohou ovládat libovolné výstupy libovolných MTB modulů. Při návrhu aplikace vyvstala otázka, zda by v konfiguraci nemělo být specifikováno, který klient může ovládat který výstup (kterého modulu). Po uvážení však bylo od tohoto nápadu upuštěno, protože by vyžadoval netriviální (ručně psanou) konfiguraci navíc a identifikaci klientů. Očekává se, že klienti připojující se k MTB Daemon jsou buď autonomní aplikace, nebo aplikace, které řízení přístupu svých uživatelů implementují na své vlastní úrovni. Pokud by nastala problematická situace, kdy dva klienti chtějí ovládat stejný výstup,

jedná se o chybu v klientské aplikaci nebo její konfiguraci, nikoliv o nekalý záměr, kterému by měl systém vší silou zabránit. MTB Daemon nastavení jednoho výstupu více klienty umožní, ale současně zaloguje varování. To umožňuje operátorovi takovou situaci rozpoznat a konfiguraci klientských aplikací opravit.

MTB Daemon si pamatuje, který klient nastavil který výstup a při odpojení klienta provede reset jím nastavených výstupů do základního stavu.

Aplikace obsahuje hierarchii tříd (ve vztahu dědičnosti) reprezentujících jednotlivé typy MTB modulů. TCP protokol také počítá s různými typy modulů. Aplikace je tak připravena na přidání budoucích nových typů MTB modulů.

MTB Daemon je pod opensource licencí dostupný online[21].

21. `https://github.com/kmzbrnoI/mtb-daemon`

5.6 hJOP MTB Network RCS knihovna

Knihovna *hJOP MTB Network RCS Library* je posledním softwarem implementovaným v rámci této diplomové práce. Jeho účelem je propojit stávající systém řízení kolejiště *hJOP*[22] a MTB Daemon. hJOP pro přístup k hardwaru řízení příslušenství používá tzv. *RCS API*[23]. Systém pro řízení příslušenství se připojuje k hJOP jako dynamicky linkovaná knihovna, která implementuje *RCS API*.

Byla tedy implementována jednoduchá knihovna, která se přilinkuje do hJOP a připojí se k MTB Daemon. Pro implementaci bylo opět zvoleno `C++` a framework `Qt`. K této volbě vedly důvody popsané v kapitole 5.5.2.

Na samotné implementaci knihovny toho není mnoho zajímavého, proto ji uvádíme jen stručně. Knihovna je pod opensource licencí dostupná online[24].

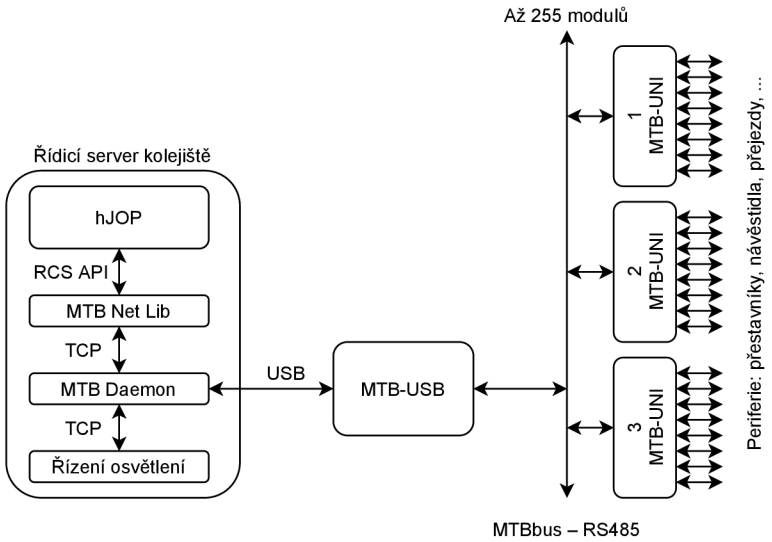
22. `https://hjop.kmz-brno.cz/`, používá se pro řízení všech kolejišť v KMŽ Brno I. Implementovaný na platformě OS Windows.
23. *Railroad Control System API*, viz `https://hjop.kmz-brno.cz/rcs`.
24. `https://github.com/kmzbrnoI/mtb-net-lib`

6 Závěr

Tato diplomová práce demonstrovala, že pro vytvoření plnohodnotného PLC systému je třeba zvládnout širokou škálu dílčích kroků. Od návrhu komunikačních protokolů, přes návrh hardwaru, programování firmwaru, až po programování počítačových aplikací a knihoven. Autorovi se povedlo všechny tyto kroky úspěšně provést. Podařilo se mu vytvořit

- nový protokol MTBbus,
- nový master modul sběrnice MTBbus (MTB-USB v4),
- nový univerzální *slave* modul sběrnice MTBbus (MTB-UNI v4),
- nástavný modul do starších modulů MTB-UNI (MTB-2-AVR),
- počítačovou aplikaci pro přístup k systému MTB (MTB Daemon),
- knihovnu pro integraci nového MTB do současného řídicího systému kolejiště (hJOP MTB Network RCS Library)

Komponenty vytvořené v rámci této práce a jejich interakce jsou přehledně zobrazeny v 6.1.



Obrázek 6.1: Komponenty vytvořené v rámci diplomové práce (mimo *hJOP* a *Řízení osvětlení*).

V rámci práce byly formulovány požadavky (4.1.4), které má nový systém MTB splnit. Všechny tyto požadavky byly naplněny. Podařilo se vytvořit systém, který za poměrně malé finanční a časové náklady značně povyšuje současný systém MTB. Především však vznikl otevřený a rozšiřitelný systém s výhledem udržitelnosti po desítky let. Nový systém MTB umožní stavbu dalších kolejišť, udržování a rozšiřování kolejišť současných. Nový systém MTB si může vyrobit každý, kdo bude chtít bezpečně řídit své modelové kolejiště.

Pro KMŽ Brno I znamená MTB v4 možnost nasadit více řídicích systémů kolejiště, a konečně tak zprovoznit řízení osvětlení, řízení tramvajové a autobusové dopravy. MTB v4 umožní nasazení pultů obsluhy, vytvoření nových chytrých zesilovačů DCC signálu. Umožní výstavbu nového depa kolejových vozidel, kde se budou používat speciální MTB moduly. Pro KMŽ Brno I je MTB v4 velkým krokem vpřed.

MTB v4 bylo nasazeno na testovacím kolejišti o 20 modulech, kde úspešně funguje.

6.1 Možná rozšíření

Při návrhu tak velkého systému, jako je MTB, se přirozeně ukázaly mnohé cesty, které by bylo možné v budoucnu prozkoumat.

Systém MTB by v budoucnu mohl podporovat vyšší rychlosti komunikace nebo automatickou detekci rychlosti sběrnice. MTB by mohlo být rozšířeno o jednotky, které provádějí retranslaci dat sběrnice bezdrátově (pro vzdálené připojení modulů).

Jednou z otázek týkajících se budoucího rozšíření je, jak umožnit, aby systém MTB mohl interagovat s komerčními softwary pro řízení modelových kolejišť. Dalším možným rozšířením je tak implementovat do majoritních komerčních softwarů pro řízení modelových kolejišť podporu pro MTB. Některé softwary jsou však uzavřené. Pro takové softwary by bylo možné napsat alternativní firmware do MTB-USB, který by zajistil, že MTB-USB modul se pro počítač bude tvářit jako některý z komerčních systémů pro řízení kolejišť.

Dalším přirozeným způsobem rozšíření, který má autor této práce v plánu v nejbližším roce uskutečnit, je přidání dalších typů modulů. V budoucnu vzniknou MTB moduly pro chytré zesilovače DCC signálu, *RailCom* detektory a jiné. Systém MTB je na tato rozšíření připraven.

Bibliografie

1. MTB MODEL. *Signalizace návěstidel S-com – popis* [online]. 2021 [cit. 2021-04-28]. Dostupné z: https://www.mtb-model.com/elektro/scom_popis.pdf.
2. UNITRONICS. *PLC: Programmable Logic Controller* [online]. 2021 [cit. 2021-04-25]. Dostupné z: https://www.unitronicsplc.com/what-is-plc-programmable-logic-controller/.
3. KLUB MODELÁŘŮ ŽELEZNIC BRNO I. *MTB: IO sběrnice pro řízení modelových kolejišť* [online]. 2021 [cit. 2021-04-25]. Dostupné z: https://mtb.kmz-brno.cz/.
4. DCC WIKI. *Introduction to DCC* [online]. 2021 [cit. 2021-04-25]. Dostupné z: https://dccwiki.com/Introduction_to_DCC.
5. KLUB MODELÁŘŮ ŽELEZNIC BRNO I. *Systém řízení kolejiště H0* [online]. 2021 [cit. 2021-04-25]. Dostupné z: https://www.kmz-brno.cz/rizeni-kolejiste-kmz-brno/.
6. VÍTĚZSLAV BÁŇA, Petr Trávník. *Komunikační protokol modulů MTB* [online]. 2021 [cit. 2021-04-28]. Dostupné z: https://mtb.kmz-brno.cz/assets/pdf/mtb-protok20.pdf.
7. RS485.COM. *REFERENCE FOR RS485, RS422, RS232 AND RS423* [online]. 2021 [cit. 2021-04-28]. Dostupné z: http://www.rs485.com/rs485spec.html.
8. VÍTĚZSLAV BÁŇA, Petr Trávník. *Zapojení a konfigurace modulu MTB-UNI v2.2* [online]. 2021 [cit. 2021-04-28]. Dostupné z: https://mtb.kmz-brno.cz/assets/pdf/mtb-uni22.pdf.
9. FUČÍK, Jindřich. *Zpětný ohlad RS bus* [online]. 2021 [cit. 2021-04-26]. Dostupné z: http://www.fucik.name/masinky/RS-Bus/.
10. NATIONAL MODEL RAILROAD ASSOCIATION. *Standards and Recommended Practices* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://www.nmra.org/index-nmra-standards-and-recommended-practices.

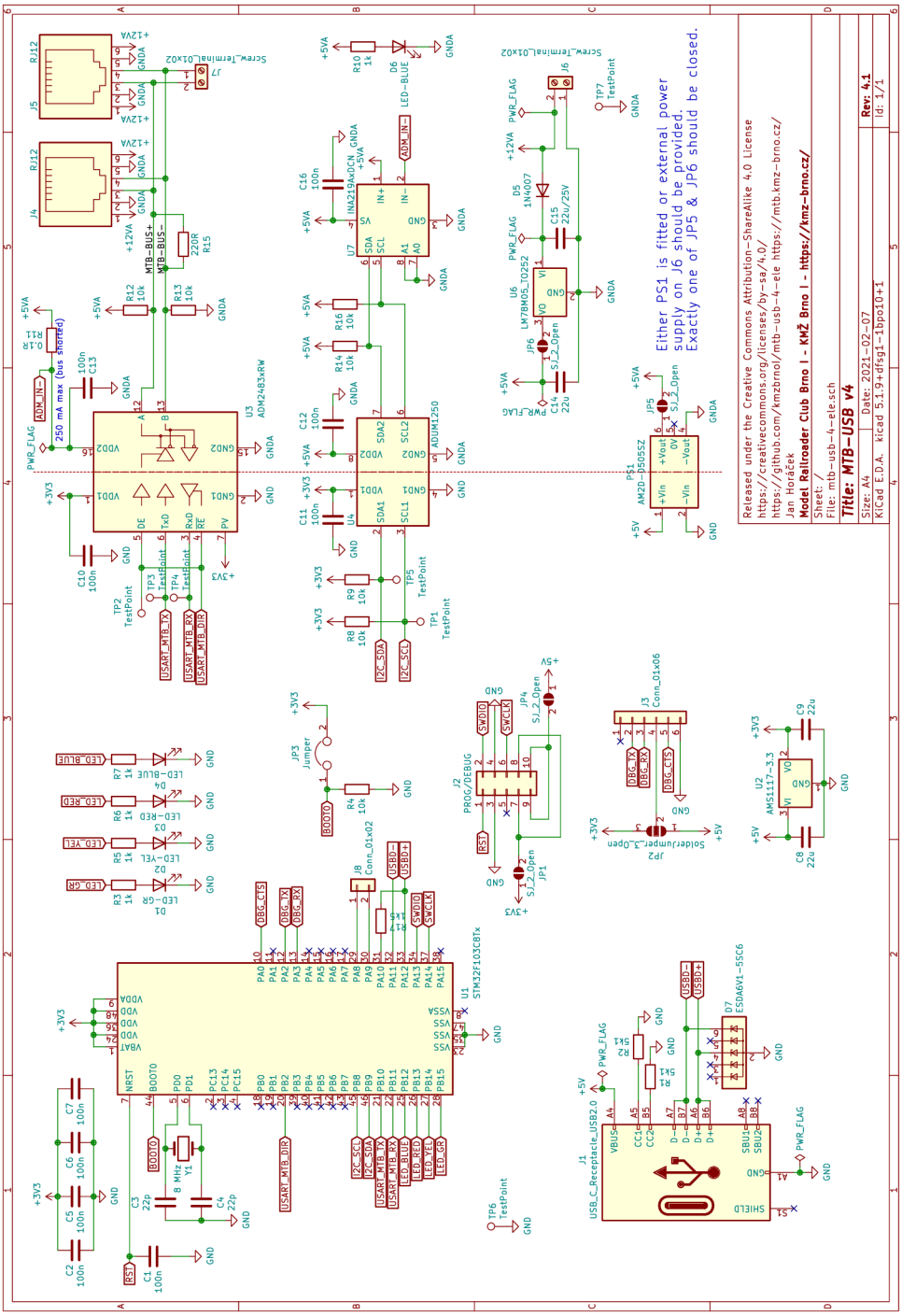
11. WIKIPEDIA. *Digital model railway control systems: Systems* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://en.wikipedia.org/wiki/Digital_model_railway_control_systems#Digital_Command_Control.
12. WIKIPEDIA. *Digital Command Control* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://en.wikipedia.org/wiki/Digital_Command_Control.
13. DCC WIKI. *RailCom and RailComPlus* [online]. 2021 [cit. 2021-05-03]. Dostupné z: https://dccwiki.com/Term:RailCom.
14. VENEMAN, Mark. *Overview of NMRA DCC systems* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://raicho.home.xs4all.nl/model/control/overview.htm.
15. OPENDCC. *Transmission of S88 data over network cable* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://www.opendcc.de/s88/s88_n/s88-n_e.html.
16. PRAS, Aiko. *RSbus Arduino Library* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://github.com/aikopras/RSbus.
17. DCC WIKI. *LocoNet overview* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://dccwiki.com/LocoNet.
18. DIGITRAX INC. *LocoNet System Architecture-Complete Train Control* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://www.digitrax.com/support/loconet/home/.
19. DIGITRAX INC. *LocoNet Personal Use Edition 1.0 Specification* [online]. 2021 [cit. 2021-04-26]. Dostupné z: https://www.digitrax.com/support/loconet/loconetpersonaledition.pdf.
20. HORÁČEK, Jan. *hJOP: systém RCS* [online]. 2021 [cit. 2021-05-03]. Dostupné z: https://hjop.kmz-brno.cz/rcs.
21. MODBUS ORGANIZATION INC. *ModBus Application Protocol Specification* [online]. 2021 [cit. 2021-05-04]. Dostupné z: https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.
22. ANALOG DEVICES. *ADM2483 Datasheet* [online]. 2021 [cit. 2021-05-04]. Dostupné z: https://www.analog.com/media/en/technical-documentation/data-sheets/adm2483.pdf.

23. STMICROELECTRONICS. *Description of STM32WL HAL and low-layer drivers – User Manual* [online]. 2021 [cit. 2021-05-04]. Dostupné z: https://www.st.com/resource/en/user_manual/dm00660673-description-of-stm32wl-hal-and-lowlayer-drivers-stmicroelectronics.pdf.

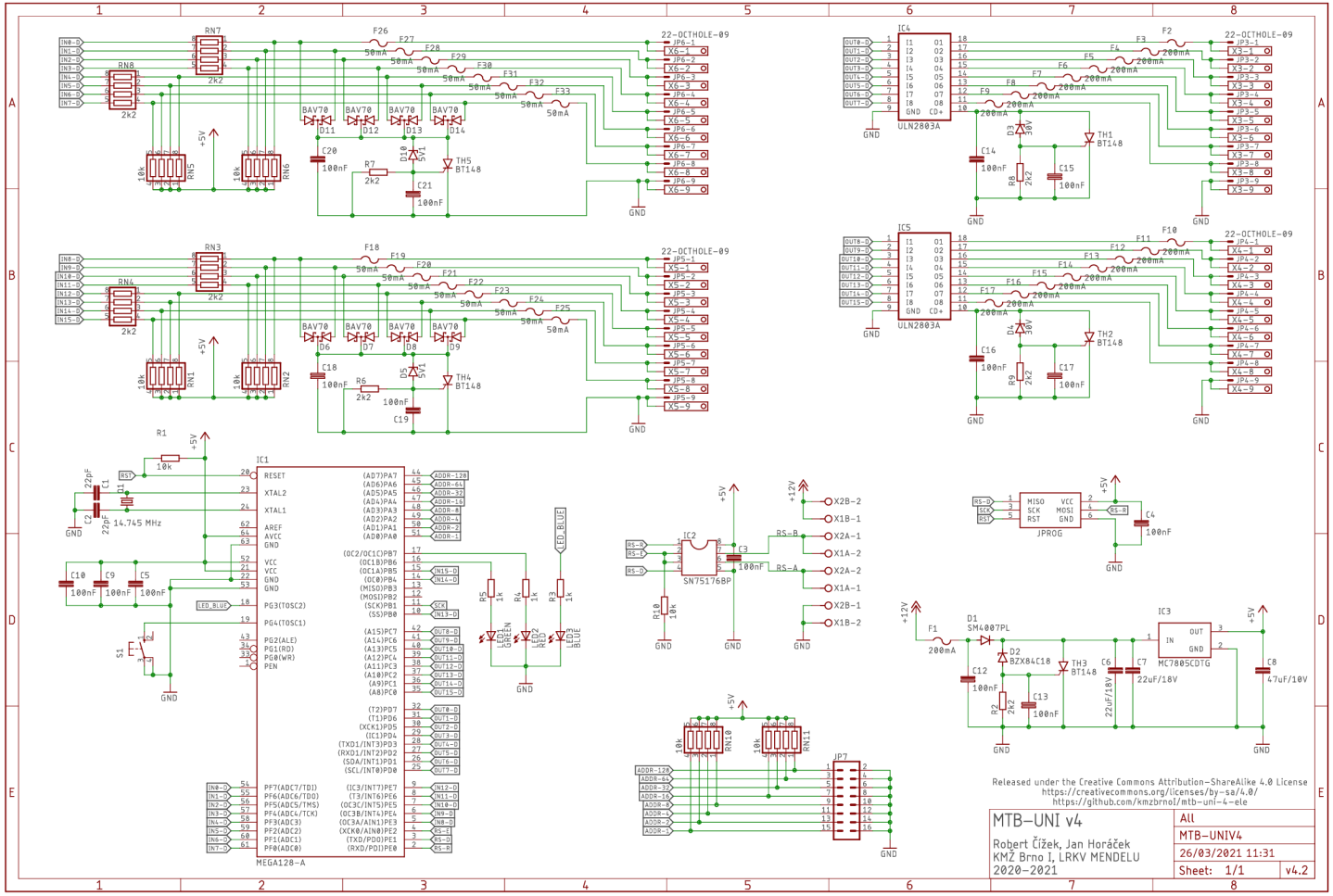
24. ATMEL. *ATmega128A Complete Datasheet* [online]. 2021 [cit. 2021-05-04]. Dostupné z: http://ww1.microchip.com/downloads/en/devicedoc/atmel-8151-8-bit-avr-atmega128a_datasheet.pdf.

25. TEXAS INSTRUMENTS. *ULN2803A Darlington Transistor Arrays* [online]. 2021 [cit. 2021-05-04]. Dostupné z: https://www.ti.com/lit/ds/symlink/uln2803a.pdf.

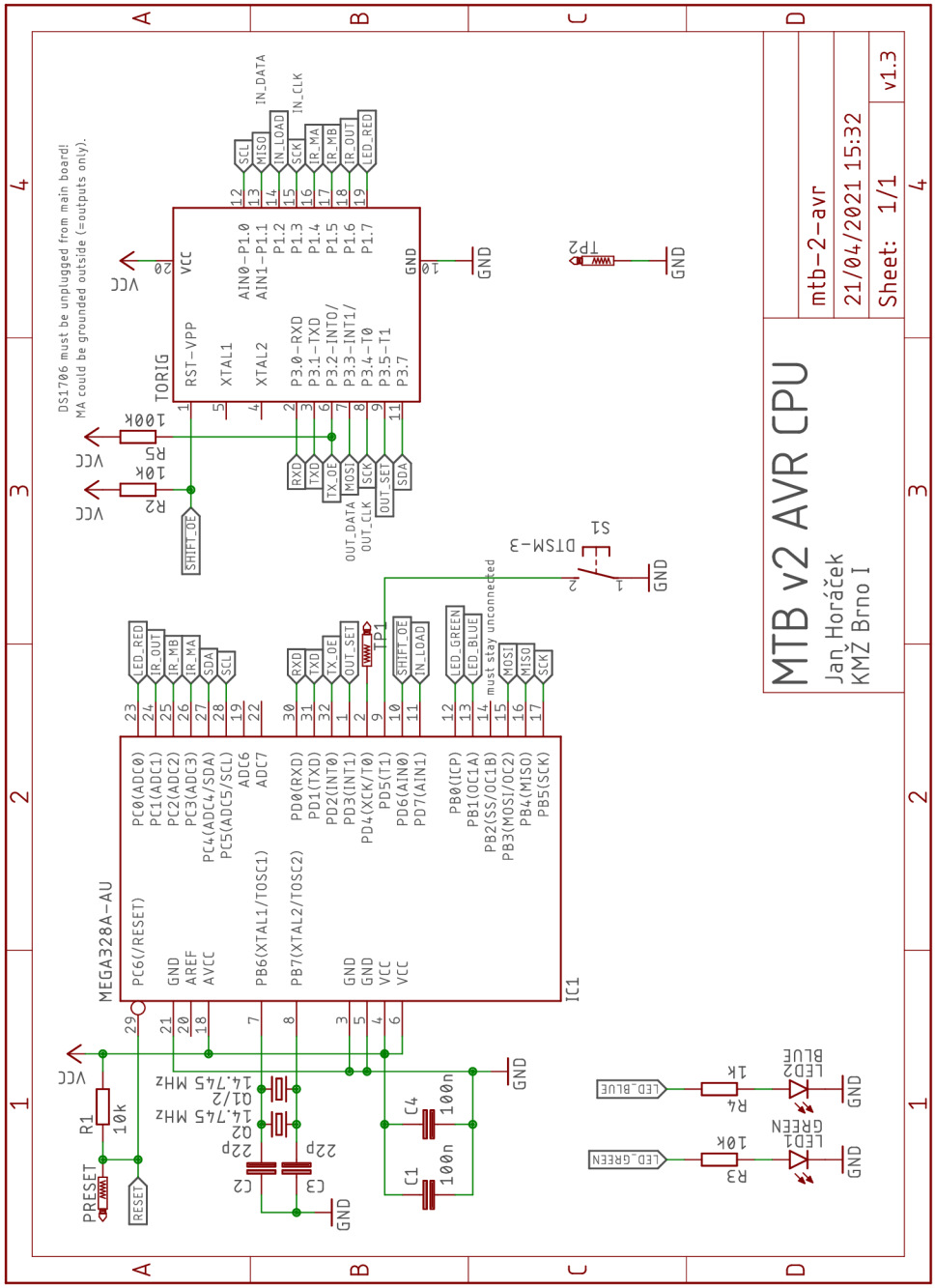
A Přílohy



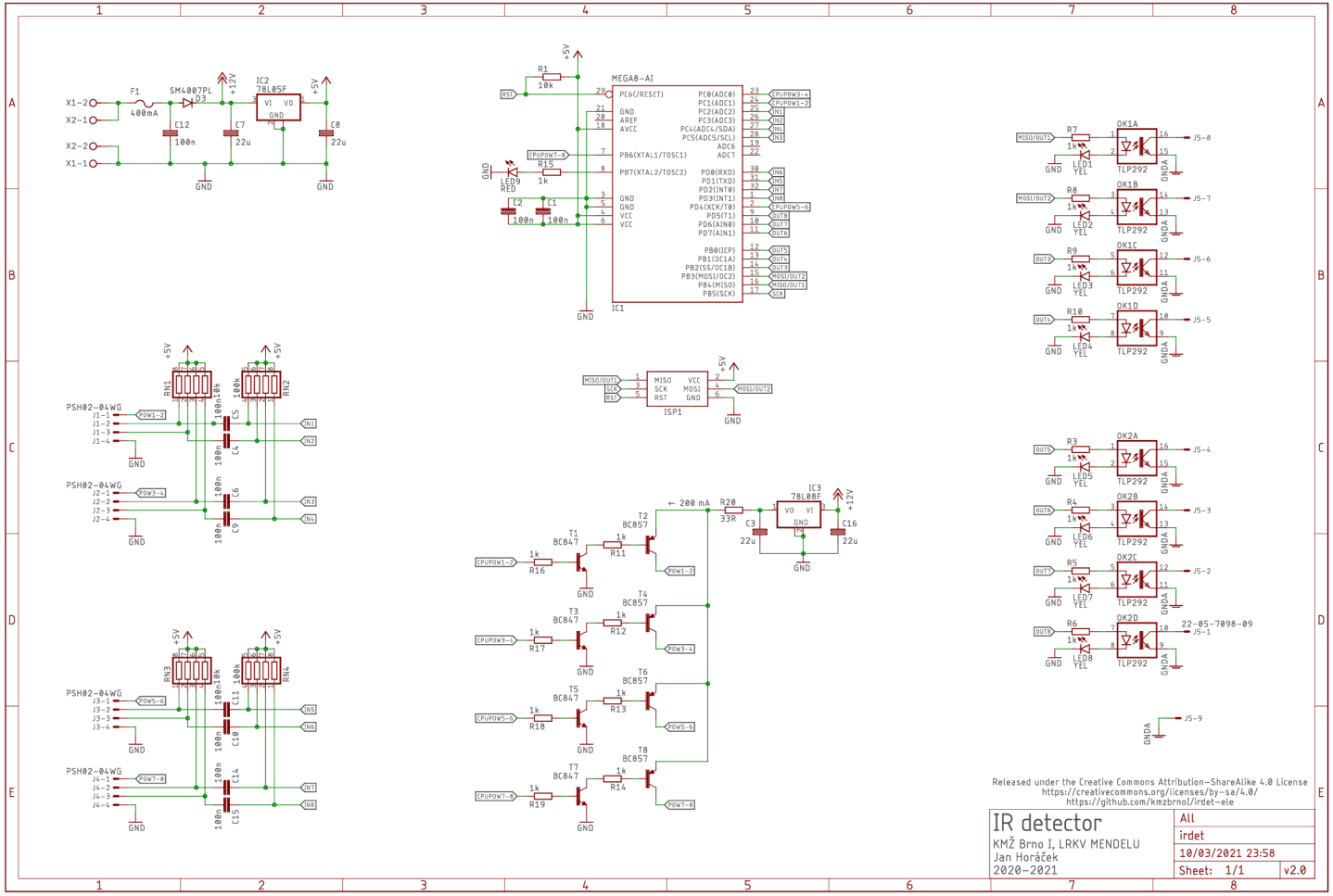
Obrázek A.1: Schéma MTB-USB v4 modulu.



Obrázek A.2: Schéma MTB-UNI v4 modulu.



Obrázek A.3: Schéma nástavné desky *MTB-2-AVR*.



Obrázek A.4: Schéma desky *IRdet*.