



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## MODBUS/TCP ROZHRANÍ PRO ŘÍZENÍ MODELU KŘIŽOVATKY

MODBUS / TCP INTERFACE FOR CROSSROAD MODEL CONTROL

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Pavel Horký

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Macho, Ph.D.

BRNO 2022

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Pavel Horký

**ID:** 211148

**Ročník:** 3

**Akademický rok:** 2021/22

**NÁZEV TÉMATU:**

## **Modbus/TCP rozhraní pro řízení modelu křižovatky**

### **POKYNY PRO VYPRACOVÁNÍ:**

1. Seznamte se s problematikou řízení křižovatek a modelem křižovatky.
2. Seznamte se se sběrnici Modbus/TCP a řídicím systémem Logix.
3. Definujte požadavky na rozhraní mezi modelem křižovatky a řídicím systémem Logix.
4. Navrhněte obvodové řešení rozhraní, zvolte vhodný mikrokontrolér a stanovte hodnoty součástek.
5. Navrhněte desky plošných spojů.
6. Rozhraní realizujte a oživte.
7. Vytvořte a odlaďte softwarové vybavení pro mikrokontrolér.
8. Vytvořte a odlaďte testovací aplikaci pro řídicím systémem Logix demonstrující funkčnost rozhraní.
9. Zhodnoťte dosažené výsledky.

### **DOPORUČENÁ LITERATURA:**

EtherNet/IP Socket Interface. Application Technique. Rockwell Automation. ENET-AT002C-EN-P. 2014.

Modbus Messaging Implementation Guide. Modbus Organization. 2006.

[https://modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf).

**Termín zadání:** 7.2.2022

**Termín odevzdání:** 23.5.2022

**Vedoucí práce:** Ing. Tomáš Macho, Ph.D.

**doc. Ing. Václav Jirsík, CSc.**  
předseda rady studijního programu

### **UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

V této práci naleznete shrnuté informace týkající se řízení křižovatek. Dále pak druhy průmyslových komunikací s upřesněním a podrobnějším rozbohem pro protokol Modbus TCP. Druhou částí je pak samotná tvorba finálního produktu. Je zde rozebrána problematika požadavků na dané zařízení a způsob jakým byli vyřešeny.

## **KLÍČOVÁ SLOVA**

Řízení křižovatek, průmyslová komunikace, protokol Modbus TCP, základní deska, vstupní a výstupní periferie, ATmega32

## **ABSTRACT**

In this work you will find summary of information regarding the management of intersections. Furthermore, the types of industrial communications with specification and more detailed analysis for the Modbus TCP protocol. The second part is the creation of the final product. It addresses the issue of requirements for the device and the way in which they were resolved.

## **KEYWORDS**

Junction control, industrial communication, Modbus TCP protocol, motherboard, input and output peripherals, ATmega32

HORKÝ, Pavel. *Modbus/TCP rozhraní pro řízení křižovatky*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2021, 56 s. Bakalářská práce. Vedoucí práce: Ing. Tomáš Macho, Ph.D.



## Prohlášení autora o původnosti díla

**Jméno a příjmení autora:** Pavel Horký  
**VUT ID autora:** 211148  
**Typ práce:** Bakalářská práce  
**Akademický rok:** 2021/22  
**Téma závěrečné práce:** Modbus/TCP rozhraní pro řízení křižovatky

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 20.5.2022

.....

podpis autora\*

---

\* Autor podepisuje pouze v tištěné verzi.

## PODĚKOVÁNÍ

Chci poděkovat vedoucímu bakalářské práce panu Ing. Tomáši Machovi, Ph.D. za pomoc při řešení problematiky dané práce, za rady, podmětné návrhy a trpělivost při častých konzultacích. Dále bych chtěl poděkovat mému otci Pavlu Horkému za častou pomoc při řešení problémů s hardwarem a celkově své rodině za velkou podporu.

# Obsah

Úvod	11
<b>1 Řízení světelných křižovatek</b>	<b>12</b>
1.1 Druhy křižovatek	12
1.2 Vybavení křižovatek	13
1.3 Řízení křižovatek	13
1.3.1 Principy pevného a dynamického řízení dopravy	13
<b>2 Komunikační možnosti systému Logix</b>	<b>15</b>
2.1 Průmyslová komunikace	15
2.2 Druhy průmyslové komunikace	16
2.2.1 Sériové protokoly	16
2.2.2 Protokoly založeny na ethernetu	16
2.3 MODBUS / TCP	17
<b>3 Návrh elektronických obvodů</b>	<b>19</b>
3.1 Volba procesoru	19
3.2 Základní deska	20
3.3 Výstupní periferie	21
3.4 Vstupní periferie	22
3.5 Ethernet modul	24
3.6 Step-down modul	24
3.7 ISP Programátor	25
<b>4 Návrh DPS</b>	<b>26</b>
4.1 Základní deska	26
4.2 Rozšíření pro základní desku	27
4.3 Výstup periferie	27
4.3.1 Parametry	28
4.4 Vstupní periferie	28
<b>5 Konstrukční uspořádání</b>	<b>29</b>
5.1 Základní deska	29
5.1.1 Propojení ethernetového modulu	30
5.1.2 Příprava na programování zařízení	31
5.2 Rozšiřující modul	32
5.3 Periferie	32
5.4 Finální produkt	33

<b>6 Software pro rozhraní</b>	<b>34</b>
6.1 AVRDUDE . . . . .	34
6.1.1 Fuses . . . . .	35
6.2 Oživení základové desky a periférií programem . . . . .	37
6.3 Firmware pro ATMEGA32 . . . . .	38
6.3.1 Testování komunikace . . . . .	39
6.4 Modbus tool . . . . .	40
6.4.1 Nastavení pro jednotlivé funkce . . . . .	41
<b>7 Návrhy na zlepšení</b>	<b>42</b>
<b>Závěr</b>	<b>43</b>
<b>Literatura</b>	<b>44</b>
<b>Seznam symbolů a zkratk</b>	<b>47</b>
<b>A Packet Modbus funkcí</b>	<b>48</b>
A.1 Funkce 1 . . . . .	48
A.2 Funkce 2 . . . . .	49
A.3 Funkce 3 . . . . .	50
A.4 Funkce 4 . . . . .	51
A.5 Funkce 5 . . . . .	52
A.6 Funkce 6 . . . . .	53
A.7 Funkce 7 . . . . .	54
A.8 Chybové hlášení . . . . .	55
<b>B Elektronická příloha</b>	<b>56</b>
B.1 Software pro ATMEGA32 . . . . .	56
B.2 Návrhy DPS . . . . .	56

# Seznam obrázků

1.1	Druhy křižovatek [21]	12
2.1	Infrastruktura podniku [2]	15
2.2	ISO model protokolu Modbus/TCP [8]	17
2.3	syntaxe Modbus/TCP [9]	18
3.1	zapojení základní desky	20
3.2	zapojení výstupní periferie	21
3.3	zapojení vstupní periferie	22
3.4	Ethernet modul [18]	24
3.5	Step-down modul [11]	25
3.6	ISP programátor [12]	25
4.1	Návrh DPS základní desky	26
4.2	Návrh DPS rozšíření pro základní desku	27
4.3	Návrh DPS výstupní periferie	27
4.4	Návrh DPS vstupní periferie	28
5.1	Realizace základní desky	29
5.2	Propojení ISP komunikace	30
5.3	Konfigurace pro programátor	31
5.4	Zapojení programátoru	31
5.5	Realizace rozšiřující desky	32
5.6	Realizace periferií	33
5.7	Finální produkt	33
6.1	Nastavení programu AVRDUDE	34
6.2	Nastavení fuses	35
6.3	Výpis z příkazového řádku na příkaz ping	39
6.4	Výpis z Wiresharku na příkaz ping	39
6.5	Výpis z Wiresharku na Write Multiple Coils	39
6.6	Modbus master tool	40

# Seznam tabulek

1.1	Přehled způsobu řízení [20] . . . . .	14
4.1	Tabulka vlastností výstupní periferie 4.3 . . . . .	28
5.1	Tabulka propojení pro obrázek 5.2 . . . . .	30
5.2	Tabulka konfigurace pro programátor 5.3 . . . . .	31
5.3	Tabulka konfigurace pro Ethernet . . . . .	32
6.1	Výběr zařízení pro hodinový takt . . . . .	36
6.2	Nastavení CKOPT vůči frekvenci . . . . .	36
6.3	Nastavení CKSEL0 a SUT . . . . .	36

# Úvod

Tato práce je zaměřena na problematiku řízení modelu křižovatky pomocí průmyslové komunikace. V rámci práce je požadavek na návrh zařízení, které bude komunikovat s řídicím PLC pomocí protokolu Modbus/TCP. Zařízení bude v konfiguraci slave pro dané PLC a bude pracovat jako mezičlen pro řízení modelu soustavy křižovatek. Tento model byl pro tuto práci poskytnut, všechny informace týkající se jeho zapojení a funkčnosti jsou čerpány z [1].

Důležitost této práce je v radikální redukci potřebných vodičů pro řízení dané křižovatky. Tato důležitost se projevuje, když je zapotřebí umístit řídicí PLC do větší vzdálenosti od dané křižovatky. To znamená, že silný svazek signálových vodičů je pak nahrazen jedním UTP kabelem. Zařízení má za úkol sbírání dat o stavu křižovatky, jejich následné zaslání do PLC, ty jsou pak vyhodnoceny a je zpět zaslána potřebná řídicí akce. Vnitřní logika zapojení tohoto zařízení bude 5V, ale výstupní periferie bude mít logiku 24V.

Pro správnou funkci, bude zapotřebí navrhnout výstupní periferie, které budou dostatečně rychlé a zároveň robustní s možností zatížitelnosti, vstupní periferie, které vždy zajistí jasnou logickou 1 nebo 0, komunikační jednotkou a vše musí být řízeno vhodným mikrokontrolerem.

# 1 Řízení světelných křižovatek

V následující části se budu věnovat typům křižovatek, co se na nich nachází a jak jsou řízeny. Informace čerpám ze zdrojů [19], [20] a [21].

## 1.1 Druhy křižovatek

V realitě rozeznáváme 5 druhů křižovatek. Mezi nejčastější patří průsečná (ve tvaru kříže) a styková (ve tvaru T). Každá křižovatka pak může mít více pruhů, může být opatřena třeba odbočovacím pruhem. Všechny zmíněné druhy mohou obsahovat světelnou signalizaci a to ať pro auta nebo chodce. Chodci pak mohou mít možnost tlačítek u přechodu, které informují o tom, že je požadavek o přechod silnice. Čím více těchto prvků křižovatka obsahuje, tím více je pak náročná na HW i SW řešení, ale tím lépe se dá řídit její chod. Na modelu poskytnutém školou se nachází 6 křižovatek, z to jsou pouze 4 říditelné. Mezi říditelné patří 3 průsečné a jedna styková. Všechny obsahují přechody, kterou jsou osazeny světelnou signalizací i tlačítky.



Obr. 1.1: Druhy křižovatek [21]



## 1.2 Vybavení křižovatek

Pro potřeby této práce se v této části budu věnovat pouze prvkům, které se mohou podílet na řízení za pomoci programovatelného automatu. Ty pak můžeme rozdělit do dvou skupin:

- Světelná signalizace
  - návěstidlo pro auta (obsahující zelenou, žlutou a červenou barvu)
  - návěstidlo pro auta (obsahující zelenou šipku pro opuštění křižovatky)
  - návěstidlo pro chodce (obsahující zelenou a červenou barvu)
- Snímače
  - snímač pro snímání přítomnosti vozidla
  - tlačítko pro snímání přítomnosti chodce

## 1.3 Řízení křižovatek

Následující kapitola a její podsekcce jsou citací zdroje [20].

Způsob, jak je řízení provozu z provozně technického hlediska realizováno světelným signalizačním zařízením, je popsáno systémem řízení. Různé systémy řízení se liší ovlivnitelností nebo proměnlivostí prvků signálního plánu. Jaký systém řízení má být zvolen, závisí na zadaných cílech, a ty lze uskutečnit použitím provozně technických veličin. S jejich pomocí může být kvalifikován systém řízení na konkrétní lokalitu. Řízení světelným signalizačním zařízením lze pokládat za část regulačního obvodu, jímž má být průběh provozu ovlivňován a optimalizován vyhodnocováním určitých parametrů.

### 1.3.1 Principy pevného a dynamického řízení dopravy

Způsoby řízení dopravy světelným signalizačním zařízením lze rozdělit na:

- řízení pevné
- řízení dynamické

Principem pevného řízení provozu je neměnné řízení v daném časovém období. To předpokládá ustálenou intenzitu provozu vozidel s pouze dlouhodobými změnami, neboť při tomto způsobu řízení není reagováno na krátkodobé výkyvy intenzity provozu.

Principem dynamického řízení je přizpůsobování se provozu tak, že i krátkodobé výkyvy intenzity provozu ovlivňují řízení. Dynamický způsob řízení provozu vyžaduje o něco vyšší technické náklady než v případě řízení pevného.

Tab. 1.1: Přehled způsobu řízení [20]

<b>A</b>		
ROZHODOVÁNÍ PŘI ŘÍZENÍ MIMO PRŮBĚH SIGNÁLNÍHO PLÁNU v delších časových intervalech v krocích řádově desítek minut až hodin		
volba signálních plánů a režimů řízení	A1 ČASOVĚ ZÁVISLÁ podle předem zadaného časového nastavení programů	
	A2 DOPRAVNĚ ZÁVISLÁ podle aktuálních dopravních nároků v reálném čase	
<b>B</b>		
ROZHODOVÁNÍ PŘI ŘÍZENÍ V PRŮBĚHU SIGNÁLNÍHO PLÁNU v krátkých časových intervalech v krocích řádově několika sekund		
pevné řízení	pevný signální plán	B1 ŽÁDNÁ MOŽNOST ZMĚN podle aktuálních dopravních nároků
dopravně závislé (dynamické) řízení	modifikace signálního plánu	B2 PROMĚNNÁ DÉLKA VOLNA B3 ZMĚNA POŘADÍ FÁZÍ B4 VKLÁDÁNÍ FÁZE NA VÝZVU B5 OKAMŽITÉ DOPLNĚNÍ NEKOLIZNÍHO VOLNA DO PROBÍHAJÍCÍ FÁZE
	plánu	B6 VOLNÁ MĚNITELNOST PRVKŮ Podle aktuálních dopravních nároků

**Způsoby A** – rozhodování při řízení mimo průběh signálního plánu slouží především k zohlednění dlouhodobých změn zatížení v komunikační síti i na jednotlivých křižovatkách. Zvolené režimy řízení a signální plány z předem připravených (částe) signálních plánů jsou zapínány v závislosti na čase (A1) nebo v závislosti na dopravě (A2) a zůstávají aktivní delší dobu.

**Způsoby B** – rozhodování při řízení v průběhu signálního plánu zohledňuje krátkodobé změny stavu dopravy na křižovatce, pokud nejsou používány pevné signální plány. Všechny způsoby předpokládají předem vypočtené signální plány nebo alespoň části signálních plánů.

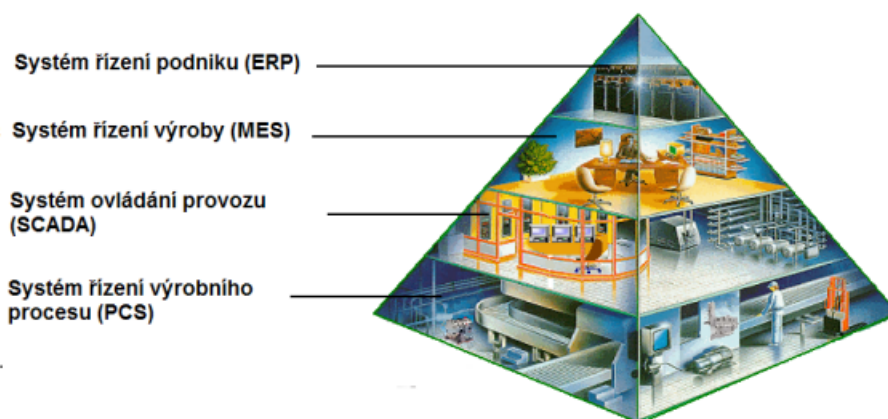
U žádného způsobu řízení nesmí být z důvodů bezpečnosti měněny mezičasy a minimální doby signálů volno.

## 2 Komunikační možnosti systému Logix

Řídicí systém Logix je navržen pro průmyslové programovatelné automaty z rodiny Allen-Bradley. Protože se jedná o průmyslové automaty, využívá tento systém standardních průmyslových komunikací. Cílem průmyslové komunikace je propojit celou infrastrukturu podniku. Všechny informace pro tuto kapitolu čerpám ze zdroje [2].

### 2.1 Průmyslová komunikace

Jak jsem již zmínil, jedná se o komunikaci, která zaštiťuje celou firmu. Řadí se do čtyř kategorií:



Obr. 2.1: Infrastruktura podniku [2]

**ERP (Enterprise Resource Planning)** je systém, který má za úkol v reálném čase sbírat data o všem v podniku. Mezi tyto data pak patří množství výrobních surovin, výrobků na skladě nebo také hotovost, kterou může určitý úsek disponovat.

**MES (Manufacturing Execution System)** sleduje proces výroby produktu. Tyto data jsou poskytovány v reálném čase a to od počátku, tedy výrobního materiálu až po konec, tedy finální podoby výrobku.

**SCADA (Supervisory Control And Data Acquisition)** je rozhraní mezi operátorem a strojem. Poskytuje možnost sledovat technologický provoz a také do něj zasahovat a manuálně řídit běh stroje.

**PCS (Process Control Systems)** jsou myšleny pak fyzické komponenty výrobního stroje. Zaštiťuje tak všechny snímače, motory, pneumatické či hydraulické písty atd.

## 2.2 Druhy průmyslové komunikace

V dnešní době máme na trhu velké množství programovatelných automatů v rámci průmyslové automatizace. Proto, abychom nebyli nuceni používat pouze jednu značku, bylo zapotřebí navrhnout standardizované komunikační protokoly. Ty zajišťují bezpečnou a spolehlivou komunikaci bez ohledu na výrobce. Každý protokol má pak své vlastnosti, jako jsou rychlosti přenosu, počet adresových uzlů apod. Mezi nejpožívanější pak řadíme tyto:

### 2.2.1 Sériové protokoly

Všechny informace pro tuto část, byli čerpány ze zdroje [3]

**PROFIBUS** je sériový protokol, který zajišťuje komunikaci procesních a pomocných dat až do rychlosti 12 Mb/s při podpoře až 126 síťových uzlů.

**sběrnice CAN** dosahuje rychlosti komunikace je až 1 Mb/s. Existují dva protokoly, CANopen a DeviceNet, které jsou standardizovány na sběrnici CAN. CANopen podporuje 127 síťových uzlů, zatímco DeviceNet podporuje 64 síťových uzlů.

**MODBUS** je jednoduchý open-source protokol, který dokáže připojit až 247 síťových uzlů. Má přenosovou rychlost až 115 kb/s a je snadno implementován na linkách RS-232 nebo RS-485.

**CC-Link** byl vyvinutý firmou Mitsubishi, používaný hlavně v asii. Tento protokol je na RS-485 schopen připojit až 64 síťových uzlů až do rychlosti 10 Mb/s.

### 2.2.2 Protokoly založeny na ethernetu

Všechny informace pro tuto část, byli čerpány ze zdroje [2] [3] [4] [5] [6] a [7]

**EtherCAT** umožňuje použití klasické topologie zapojení v různých kombinacích do hvězdy či stromu se standardními síťovými komponentami. Vyznačuje se přenosovými rychlostmi 100 Mb/s a více. EtherCAT umožňuje propojit až 65 535 síťových uzlů.

**EtherNet/IP** může mít neomezený počet uzlů systému, protože používá standardní ethernet a switche. Přestože je kompatibilní s řadou standardních internetových a ethernetových protokolů, má omezené schopnosti v reálném čase. Jeho přenosová rychlost se potom pohybuje mezi 10 Mb/s až 10 Gb/s.

**PROFINET** je velmi robustní a spolehlivou komunikací, která umožňuje vytvářet složité komunikační topologie. Dosahuje se rychlosti přenosu až 100 Mb/s.

**POWERLINK** je rychlý a deterministický protokol, který dokáže řídit až 254 slave zařízení.

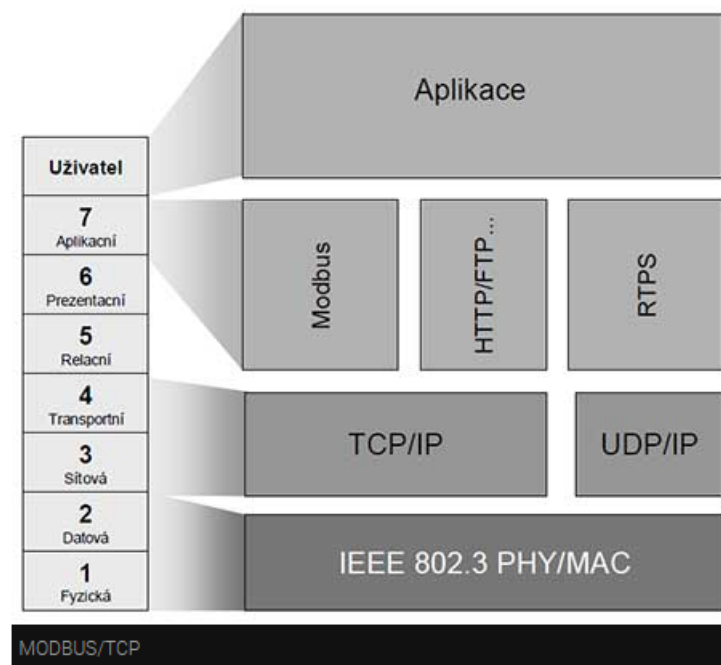
**Sercos III** je třetí generací tohoto protokolu a disponuje velmi nízkou odezvou. Délka jednoho cyklu je 31,25  $\mu$ s a může mít až 511 síťových uzlů.

**CC-Link IE** má dva verze, CC-Link IE Control a CC-Link IE Field. Verze Control je určena pro komunikaci typu kontroler-kontroler a může mít až 120 síťových uzlů. Verze Field může mít síťových uzlů až 254. Přenosová rychlost pak dosahuje až 1 Gb/s.

**MODBUS / TCP** je rozšířením k protokolu Modbus. Toto rozšíření umožňuje zaslání zpráv Modbus přes TCP / IP na Ethernetu a to rychlostí 10 Mb/s až 100 Mb/s.

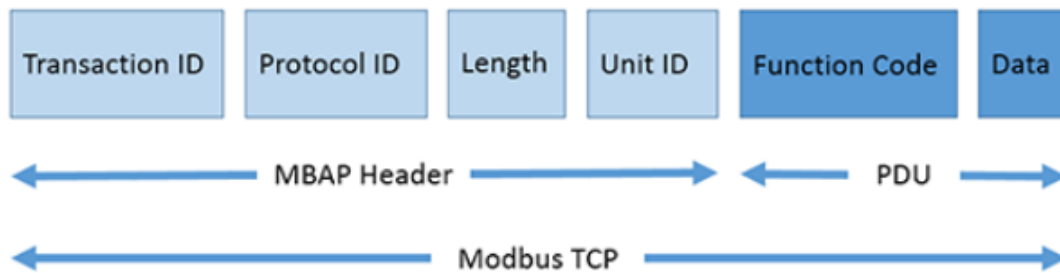
## 2.3 MODBUS / TCP

Informace použité v této části jsou použity z [8] [9] a [10]. Modbus protokol byl vytvořen z roce 1979 firmou Modicon, která spadá pod Schneider Electric. Modifikace Modbus/TCP založená na Ethernetu pak vznikla v roce 1999.



Obr. 2.2: ISO model protokolu Modbus/TCP [8]

Od roku 1999 se tento protokol stal standardizovanou metodou pro přenos diskrétních i analogových I/O informací mezi průmyslovým řízením a monitorovacími zařízeními. Zařízení používající tento protokol využívají master-slave techniku komunikace. To znamená, že jakoukoliv komunikaci začíná hlavní jednotka "master" a "slave" pouze odpovídá. Slave je pak jakékoliv periferní zařízení jako například: I/O, ventil, jednotka pro měření atd. Zpráva odesílána pomocí tohoto protokolu má složení dle následujícího obrázku 2.3.



Obr. 2.3: syntaxe Modbus/TCP [9]

**Transaction ID (Identifikátor transakce)** - Zařízení Master nastaví 2 bajty, aby každý požadavek jednoznačně identifikoval. Mohou být libovolné. Tyto bajty jsou v odezvě opakovány zařízením Slave, protože odezvy zařízení Slave nemusí být vždy přijímány ve stejném pořadí jako požadavky [9].

**Protocol ID (Identifikátor protokolu)** - Zařízení Master nastaví 2 bajty, vždy to budou 00 00, což odpovídá protokolu Modbus [9].

**Length (Délka)** - Zařízení Master nastaví 2 bajty a identifikuje počet bajtů v následující zprávě. Počítá se od Unit Identifier do konce zprávy. [9].

**Unit Identifier (Identifikátor jednotky nebo adresa zařízení)** - Zařízení Master nastaví 1 bajt. Opakuje se zařízením Slave, aby bylo možné zařízení Slave jednoznačně identifikovat. [9].

## 3 Návrh elektronických obvodů

Tato část bude zaměřena na cíle, které byly stanoveny pro každou z daných částí, tak aby finální zařízení bylo schopno bezpečně a spolehlivě pracovat. Mezi důležité parametry patří např. modulárnost a nízké proudové zatížení procesoru.

### 3.1 Volba procesoru

Volba procesoru pro desku byla podmíněna hned několika požadavky. Prvním požadavkem byl počet vstupů a výstupů procesoru. Na modelu, který bude řízen, byla stanovena křížovatka s největším počtem prvků a zjistilo se tak, že bude potřeba 26 vstupů a výstupů, bez komunikačního modulu. Komunikační modul poté potřebuje 4. Dalším požadavkem byla jednoduchost, neboli procesor, do kterého bude jednoduché program nahrát a nebude požadovat operační systém pro chod. Pro požadavek modularity základní desky je nutností, aby pouzdro procesoru bylo THT (vývodové). Následně byla provedena analýza dostupnosti na trhu.

Na stránce TME [16] lze vyčíst, že se na českém trhu nachází tyto rodiny procesorů:

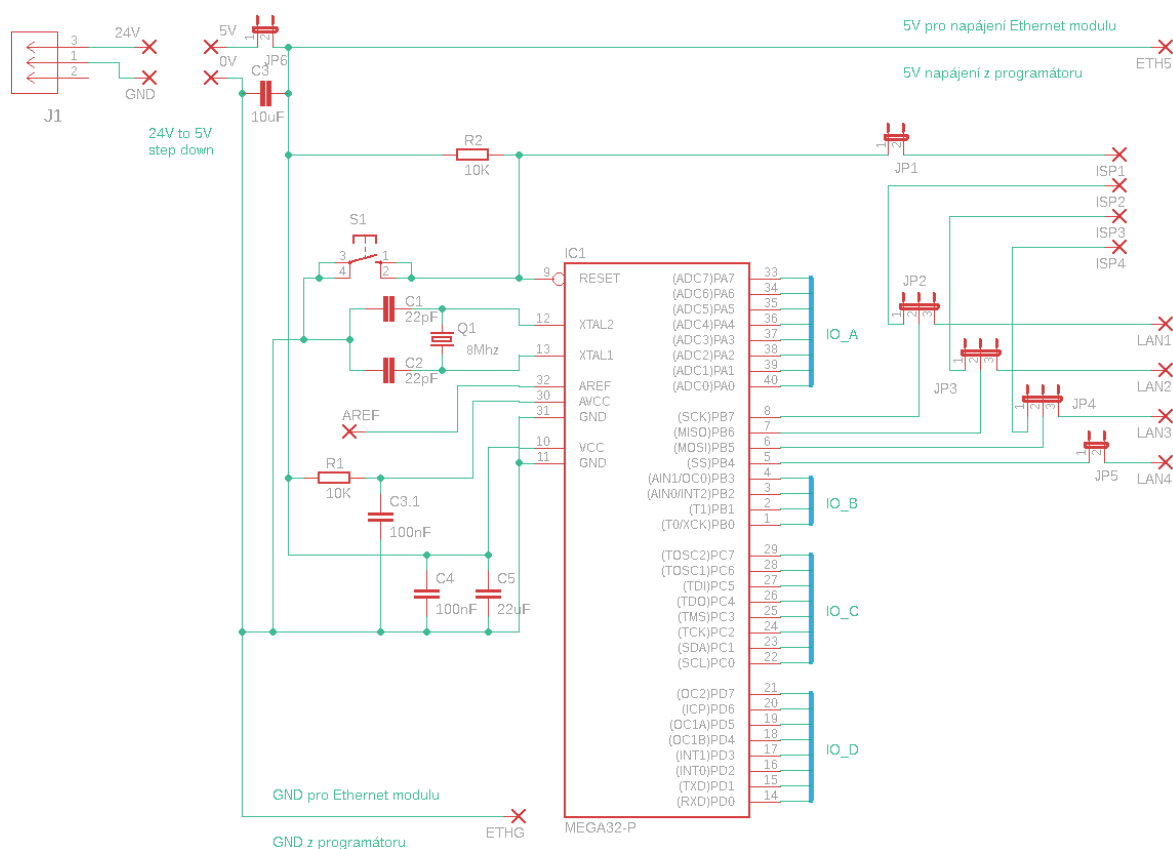
- Microchip
  - 8051 (8 bit)
  - AVR (8/32 bit)
  - PIC (8/16/32 bit)
  - dsPIC (16 bit)
  - ARM (32 bit)
- Infineon Technologies
- NXP
- Texas Instruments
- ST

Po zadání požadavku na THT pouzdro, bylo možno vyřadit rodinu Infineon Technologies a NXP. Dále požadavek na alespoň 28 vstupů / výstupů + 4 piny pro komunikaci. Tímto byla vyloučena většina rodin a zbyli pouze procesory typu AVR (8bit), 8051 (8 bit) a PIC (8 bit).

Pro tyto typy procesorů následně zbyli rodiny AT80, AT89, ATMEGA a PIC16. Všechny tyto rodiny splňovali požadavky, byl tedy vybrána rodina ATMEGA, procesor typu AVR [17].

## 3.2 Základní deska

Účelem základní desky je, obsluha mikrokontroleru. To znamená, že obsahuje zapojení součástek nutných pro funkčnost mikrokontroleru, viz. obrázek 3.1. Dále je potřeba stabilního napájení. Pro napájení byl vybrán step-down modul, který dokáže udržet stabilních 5 V na výstupu, při standardních napětích od 5 V do 28 V. Na schématu se nachází mezi kontakty 24V, 5V, GND a 0V. Důležitou částí step-down je přidání jumper portu JP6, který má za úkol odpojit tento step-down modul v případě napájení z programátoru. Nakonec byli přidány filtrační kondenzátory pro vyhlazení napětí pro mikrokontroler. Fyzické uložení step-down modulu lze vidět na obrázku 5.1.



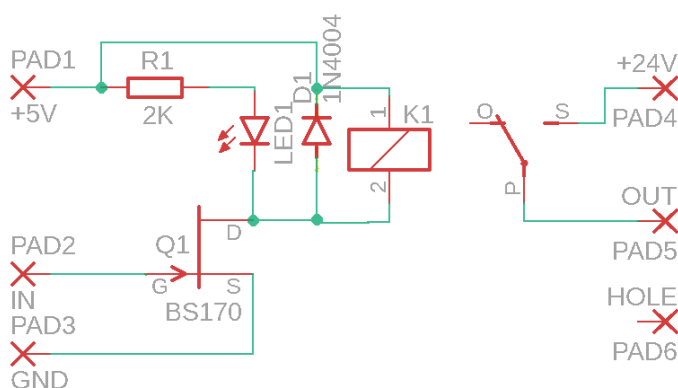
Obr. 3.1: zapojení základní desky

Schéma na obrázku 3.1 je převzato z [22], jedná se o zapojení vývojového kitu od firmy AND-TECH. Toto schéma bylo následně upraveno podle diplomové práce [14] na kterou tato práce navazuje.



### 3.3 Výstupní periferie

Na výstupní periferie je kladeno nejvíce požadavků. Velkou důležitostí zde hraje galvanické oddělení mezi mikrokontrolerem a výstupem zařízení. Tento požadavek zaručuje, že když se nachází zařízení připojené na výstup v poruše, tak nemůže dojít k poškození samotného mikrokontroleru. Další podmínkou je, aby spínání tohoto prvku nebylo proudově náročné na mikrokontroler a bylo tak možné sepnutí maximálního počtu výstupu, bez přetížení. Zároveň je potřeba, aby na výstupu bylo stabilní napětí, tedy aby nedocházelo k úbytku při zatížení. V neposlední řadě by spínání nemělo být pomalé. Všechny požadavky jsou pak potřeba splnit na DPS, která bude velikostně a pinově odpovídat vstupní periferii.



Obr. 3.2: zapojení výstupní periferie

Původní myšlenkou bylo použití optočlenu pro galvanické oddělení a zároveň pro spínání výstupu. Zde se však objevil problém s úbytkem napětí na přechodu tranzistoru, který se zvětšoval se vzrůstající zátěží. Bylo tedy potřeba použít relé, pro možnost proudového zatížení až do 0,5 A. Tato možnost však přinesla problém času potřebného ke sepnutí, protože polovodičový přechod, je mnohem rychlejší než mechanický spínač. Tento čas však i se zákmitem činí 0,5 ms, což je dostačující. Následný problém bylo napájení cívky daného relé. Pro sepnutí použitého relé je nutné dodat 10 mA proudu, což by vedlo k znemožnění sepnutí maximálního počtu výstupů. Tento problém byl vyřešen přidáním unipolárního tranzistoru Q1 jako spínače. Nakonec bylo relé vybaveno usměrňovací diodou v závěrném směru a LED signalizující sepnutý stav.

Hodnota rezistoru R1 ve schématu na obrázku 3.2 byla stanovena na 2000  $\Omega$  v závislosti na požadavcích LED1 a to: napětí 1,8 V a proud 1,6 mA. Dané informace lze vyčíst z datasheetu [15].

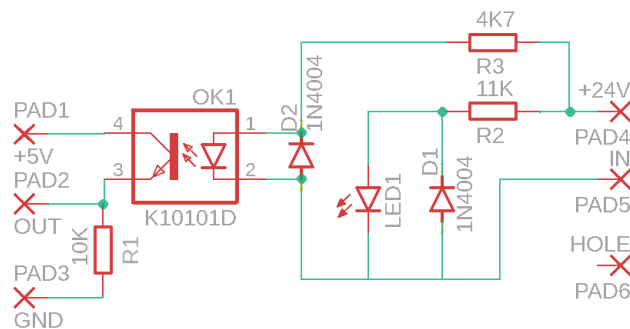
$$R_1 = \frac{U_{cc} - U_{LED1}}{I_{LED1}} = \frac{5 - 1,8}{0,0016} = 2000 \Omega \quad (3.1)$$

Výkon na tomto rezistoru pak bude 5 mW

$$P_{R1} = (U_{cc} - U_{LED1}) \cdot I_{LED1} = 3,2 \cdot 0,0016 = 0,005 \text{ W} \quad (3.2)$$

### 3.4 Vstupní periferie

Vstupní periferie obsahují mnoho stejných požadavků jako periferie výstupní. Nej důležitějším je také galvanické oddělení, které zde ale zastává optočlen. Optočlen na rozdíl od výstupní periferie zde není proudově zatížen a je na něm tedy velmi malý úbytek napětí. Na schématu na obrázku 3.3 lze dále pozorovat usměrňovací diody D1 a D2, které zde zastávají funkci ochrany proti přepólování. Toho je docíleno tak, že v případě přepólování udržují na LED1 a vstupu optočlenu 0,7 V. Toto napětí je pak menší než napětí v závěrném směru, které by bylo potřebné na zničení daných prvků.



Obr. 3.3: zapojení vstupní periferie

Poslední požadavek byl, aby toto zapojení rozměrově odpovídalo velikosti výstupní periferii a ctilo rozložení pinů, tak jak již bylo zmíněno v kapitole 3.3.

Pro zapojení na obrázku 3.3 bylo potřeba stanovit 3 hodnoty rezistorů. Hodnota rezistoru R1 byla stanovena na 10 kΩ. Proud pro tuto hodnotu činí 0,5 mA viz. vzorec 3.3. V následujícím vzorci 3.4 je pak výpočet pro zjištění výkonu na daném rezistoru.

$$I_{R1} = \frac{U_{cc}}{R1} = \frac{5}{10000} = 0,5 \text{ mA} \quad (3.3)$$

$$P_{R1} = U_{cc} \cdot I_{R1} = 5 \cdot 0.0005 = 0.0025 \text{ W} \quad (3.4)$$

Proud pro R3 musí být nastaven na 5 mA, aby tranzistor dosáhl 100 % proudového přenosu. Při této hodnotě proudu je úbytek napětí na led v optočlenu 1,3 V. Hodnota R3 byla tedy stanovena vzorcem 3.5. Dané informace lze vyčíst z datasheetu [24].

$$R3 = \frac{U_{vst} - U_{21}}{I_{21}} = \frac{24 - 1,3}{0,005} = 4540 \Rightarrow 4k7 \Omega \quad (3.5)$$

$$P_{R3} = (U_{vst} - U_{21}) \cdot I_{21} = 22,8 \cdot 0,005 = 0.114 \text{ W} \quad (3.6)$$

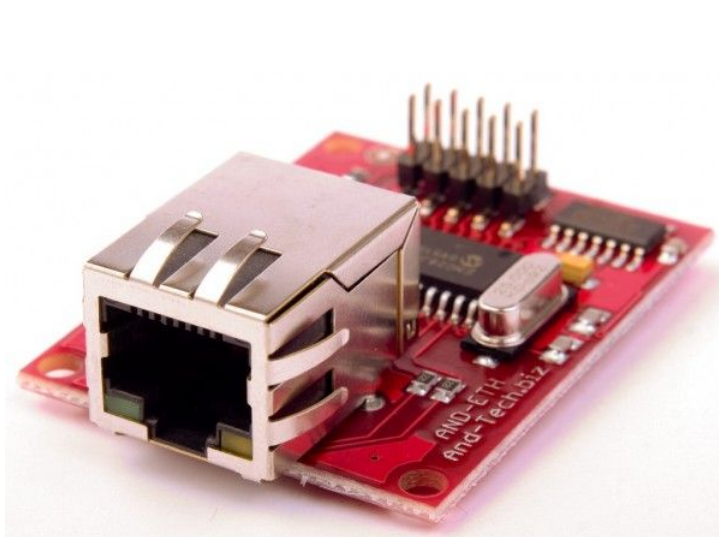
Proud tekoucí diodou LED1 je 2 mA, úbytek napětí při tomto proudu činí 1.9 V. Hodnota rezistoru R2 je pak dána ze vztahu 3.7. Dané informace lze vyčíst z datasheetu [15].

$$R2 = \frac{U_{vst} - U_{LED1}}{I_{LED1}} = \frac{24 - 1,9}{0,002} = 11000 \Omega \quad (3.7)$$

$$P_{R2} = (U_{vst} - U_{LED1}) \cdot I_{LED1} = 22,2 \cdot 0,002 = 0.0473 \text{ W} \quad (3.8)$$

### 3.5 Ethernet modul

Pro tuto část byl vybrán modul Ethernet AND-ETH v2 [18] osazen čipem ENC28J60. Tento čip má vlastní přidělenou MAC adresu. Modul obsahuje také stabilizátor, který umožňuje napájení 5 V místo 3,3 V. Dále využívá integrovanou síťovou zásuvku RJ45 s oddělovacím transformátorem a podporuje přenos až 10 Mb/s. K základní desce je poté připojen přes rozhraní SPI.



Obr. 3.4: Ethernet modul [18]

### 3.6 Step-down modul

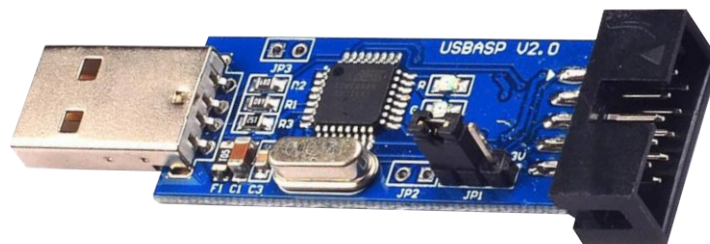
Pro napájení desky byl vybrán step-down modul pro jeho pozitivní vlastnosti. Mezi tyto vlastnosti patří možnost nastavení stálého napětí na výstupu s možností změny síťového adaptéru. Lze tedy použít adaptér poskytující 5-28 V. Poté dokáže dodat proud o velikosti 2 A, což je vyhovující, protože při maximálním odběru budu zařízení potřebovat 336 mA. Tento proud je stanoven jako, maximální počet výstupních periférií vynásobený jejich napájecím proudem. K tomu je následovně připočítán maximální odběr ethernetového modulu. Tedy 28 periférií krát 12 mA plus 180 mA (příkon mikrokontroleru byl zanedbán). Celkově tedy 516 mA. Další výhodou je, že zde nedochází k nadměrnému přehřívání a není tak potřeba přidávat chladič jako je tomu třeba u regulátorů s pouzdem TO-220. Nevýhodou je však zvlnění průběh výstupního napětí, to je ale vyřešeno přidáním filtračního kondenzátoru.



Obr. 3.5: Step-down modul [11]

### 3.7 ISP Programátor

Pro nahrávání programů do mikrokontroleru je potřeba obvodu disponujícím možností nahrávání pomocí ISP. První myšlenkou bylo vybavit samotnou základní desku tímto obvodem. Tato myšlenka však nebyla uskutečněna, protože součástky na výrobu tohoto obvodu byli dražší, než samotný produkt a samotný obvod nebyl potřebný pro chod zařízení. Byl zvolen externí programátor USBASP v2.0. Výhodou je, že v případě výroby více zařízení je možno použít jednoho programátoru. Nevýhodou tohoto programátoru je pouze nutnost manuální instalace potřebných driverů od výrobce a nemožnost debugování.



Obr. 3.6: ISP programátor [12]

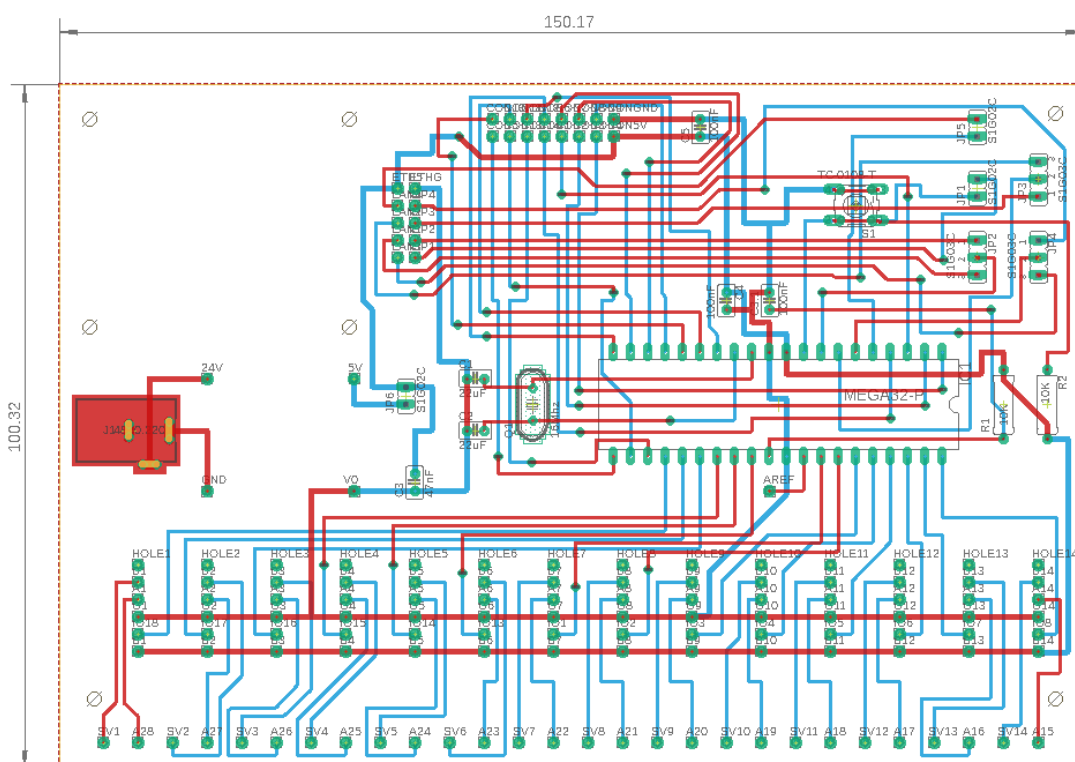
## 4 Návrh DPS

V následující části je rozebrán návrh každé s DPS, tj. jejich návrh a popis.

### 4.1 Základní deska

Jak bylo již bylo zmíněno v kapitole 3.1, účelem základové desky je, aby obsahovala pouze nezbytné součástky, ale aby umožňovala propojení všech modulů. V návrhu lze pozorovat, že na svorkovnice umístěné ve spodní části návrhu, je připojena pouze polovina možných programovatelných pinů. Důvodem pro toto rozhodnutí byla velikost. Myšlenkou byl návrh finálního produktu, který by mohl být využit ve vícero použitích. To znamená, že zde vznikl požadavek na velikost, což bylo přímým rozporu s modulárností. Proto v použitích, kde není potřeba využití velkého množství vstupů a výstupů, bylo potřeba zařízení zmenšit. Toho bylo docíleno tak, že byla polovina programovatelných pinů vyvedena na konektor. Finální návrh tak dosáhl velikosti 10x15 cm.

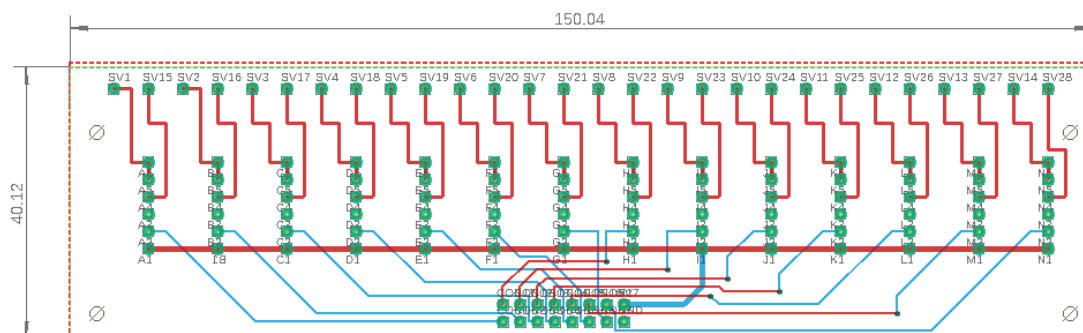
Nakonec zde lze vidět umístění ethernetového modulu, napravo od něj poté SPI konektor a nakonec také místo pro step-down modul označený jako 24V -> 5V.



Obr. 4.1: Návrh DPS základní desky

## 4.2 Rozšíření pro základní desku

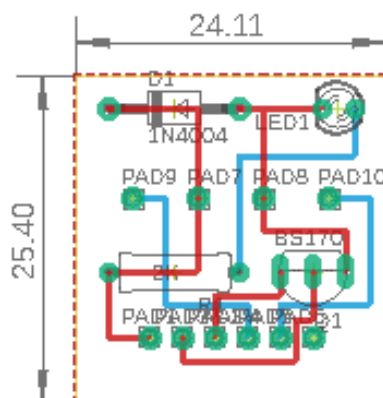
Zde se jedná o jednotku umožňující rozšíření počtu vstupů a výstupů pro základní desku. Jednotka je pak v případě nutnosti připojena do již zmíněného konektoru na základové desce. Finální návrh je o velikosti 4x15 cm, tedy došlo k 5 cm redukci na šíři základové desky.



Obr. 4.2: Návrh DPS rozšíření pro základní desku

## 4.3 Výstup periferie

Návrh periferií byl založen na základu jímž je šesti pinová patice, která musí dodržovat zapojení základní desky. První z periferií byla navržena periferie výstupní, protože obsahovala největší součástky. Následně byla priorita přiřazena LED1. Tato LED musí být na stejné pozici pro všechny periferie tak jako patice. Poslední snahou byla pak snaha o posunutí relé co nejnižze pro větší mechanickou stabilitu.



Obr. 4.3: Návrh DPS výstupní periferie

### 4.3.1 Parametry

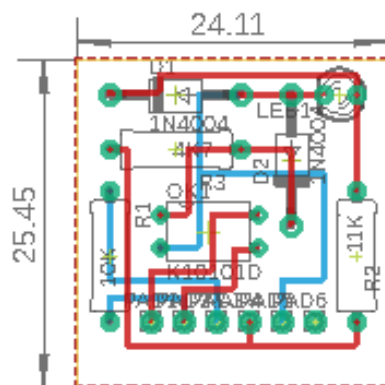
Tab. 4.1: Tabulka vlastností výstupní periferie 4.3

Napájecí napětí	5 V
Napájecí proud	12 mA
Čas sepnutí	0.5 ms
Setnutelné napětí	200 VDC
Maximální zatížitelnost	100 W
Životnost [sepnutí]	$5 \cdot 10^7$

Výsledný návrh byl zmenšen na velikost 25x24 mm, což se stalo předlohou pro periferii vstupní. Tento požadavek, stejně jako požadavek na LED je hlavně estetického základu, avšak přináší pozitivna pro případnou snahu zařízení zavřít do krabice. V tomto případě pak bude možné vytvořit přehledný pás stavových LED v daném krytí.

## 4.4 Vstupní periferie

Jak tedy již bylo zmíněno v sekci 4.3, požadavkem s ohledem na funkčnost je identické zapojení patice. Dále lze z obrázku 4.4 pozorovat, že byly dodrženy i požadavky estetické, tj. identické uložení LED a dodržena velikosti 25x24 mm.



Obr. 4.4: Návrh DPS vstupní periferie

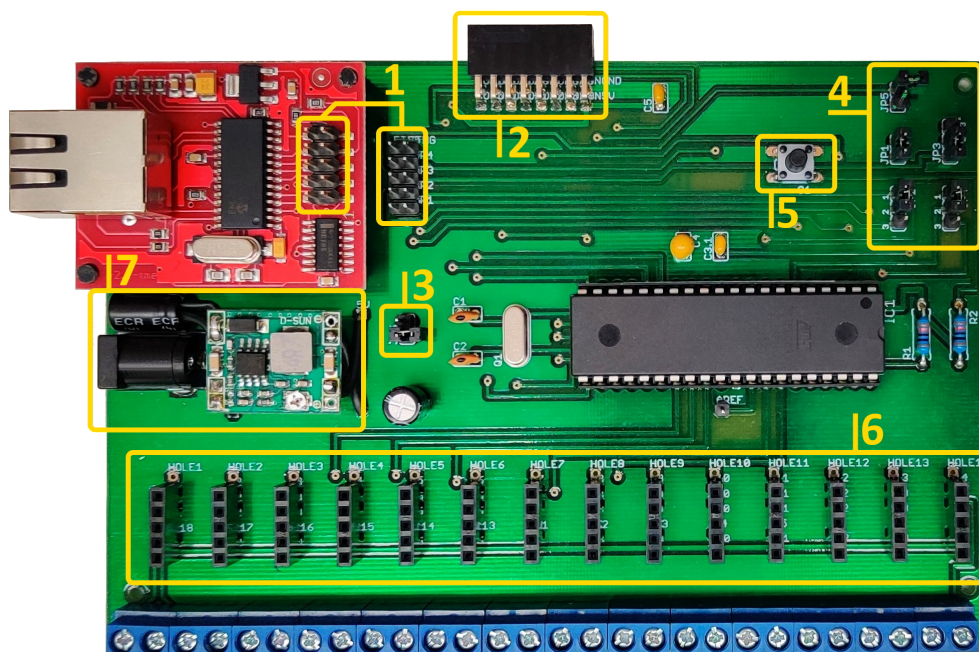


## 5 Konstrukční uspořádání

Tato kapitola je věnována pohledu na finální stav hardwaru vyvíjeného zařízení. Nachází se zde vizuální shrnutí zadaných požadavků a také základní informace o obsluze. V podkapitole 5.1.1 se nachází grafické znázornění připojení ethernetového modulu k základové desce. Následující podkapitole 5.1.2 se dále věnuje nutnostem, které musí být splněny před programováním.

### 5.1 Základní deska

Na obrázku 5.1 lze pozorovat zhotovený kus základní desky. Modul ethernetu a modul napájení step-down jsou na desku přidělány ve výšce 1 cm. Pro modul ethernetu byli vytisknuty speciální distační sloupky, které umožňují vysunutí koncovky ethernetového modulu, což nabízí lepší přístup a možnost ukotvení v případě zavření do krabice. Dále se tak zlepšují vlastnosti chlazení dané desky. Z tohoto důvodu je step-down modul pro napájení umístěn do stejné výše. Jedinou vadou je DC konektor, jež byl na desku navržen dříve a nectí tak možnost zakrytování.



Obr. 5.1: Realizace základní desky

Pro přesnější popis bude využito žlutého číselného značení v obrázku 5.1:

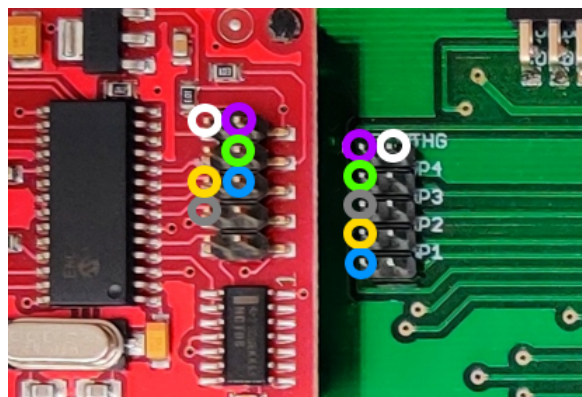
- 1 -> konektory pro ISP rozhraní
- 2 -> konektor pro připojení Rozšiřujícího modulu základové desky
- 3 -> zkratovací propoj sloužící k odpojení step-down modulu od základní desky
- 4 -> zkratovací propoje sloužící k přepnutí ISP kontaktů (více v 5.1.2)
- 5 -> reset tlačítko
- 6 -> patice pro připojení periférií
- 7 -> napájecí obvod (step-down modul)

### 5.1.1 Propojení ethernetového modulu

Pro správnou funkci musí být modul zapojen následujícím způsobem:

Tab. 5.1: Tabulka propojení pro obrázek 5.2

Barva	ATMEGA32	AND-ETH v2
Modrá	MOSI	SI
Žlutá	SCK	SCK
Šedá	MISO	SO
Zelená	SS	/CS
Fialová	+5V	VCC
Bílá	GND	GND



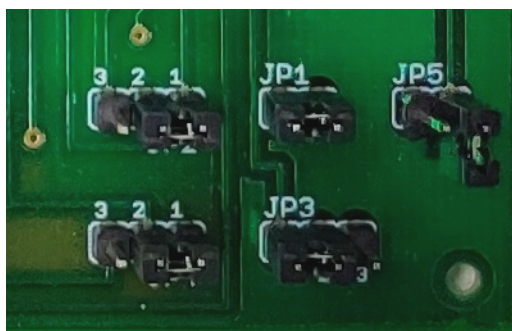
Obr. 5.2: Propojení ISP komunikace

## 5.1.2 Příprava na programování zařízení

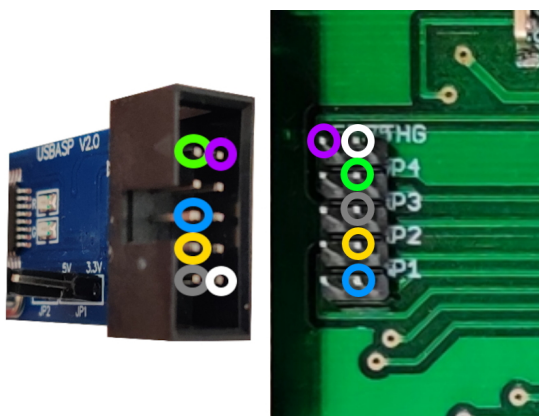
Následující podkapitola je věnována potřebným opatřením, které je nutno splnit předtím, než budu zařízení převedeno do programovacího stádia.

Tab. 5.2: Tabulka konfigurace pro programátor 5.3

JP2	JP1	JP5
2+1	✓	–
JP4	JP3	JP6
2+1	2+1	–



Obr. 5.3: Konfigurace pro programátor



Obr. 5.4: Zapojení programátoru

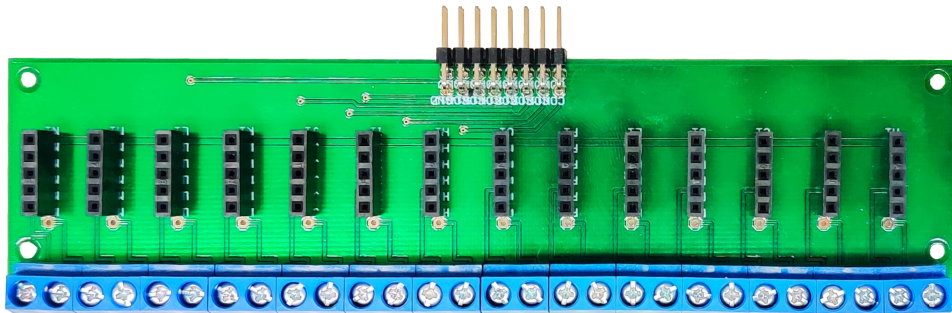
Tab. 5.3: Tabulka konfigurace pro Ethernet

JP2	JP1	JP5
3+2	–	✓
JP4	JP3	JP6
3+2	3+2	✓

Zkratovací propoj s pojmenováním JP6 se nachází na obrázku 5.1 a je žlutě označen číslem 3. Pro připojení programátoru je nutné zkratovat pin 2 a 1 na jumperu JP2, JP3 a JP4. Dále je nutné zkratovat JP1, avšak JP5 a JP6 musí zůstat rozpojeny. Pro připojení ethernetového modulu musí dojít ke zkratování pinu 2 a 3 na JP2, JP3 a JP4. Následně se rozpojí JP1 a zkratuje JP5 a JP6.

## 5.2 Rozšiřující modul

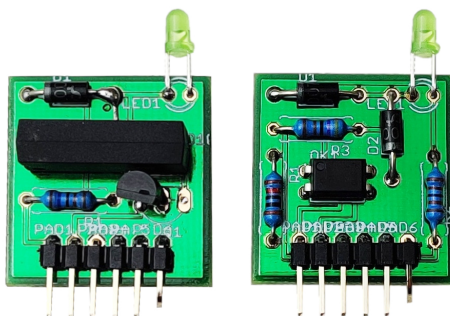
Jak lze pozorovat, tak rozšiřující modul se skládá ze stejné patice pro připojení periférií (číslo 6) a protikusů pro konektor (číslo 2) viditelných na obrázku 5.1.



Obr. 5.5: Realizace rozšiřující desky

## 5.3 Periferie

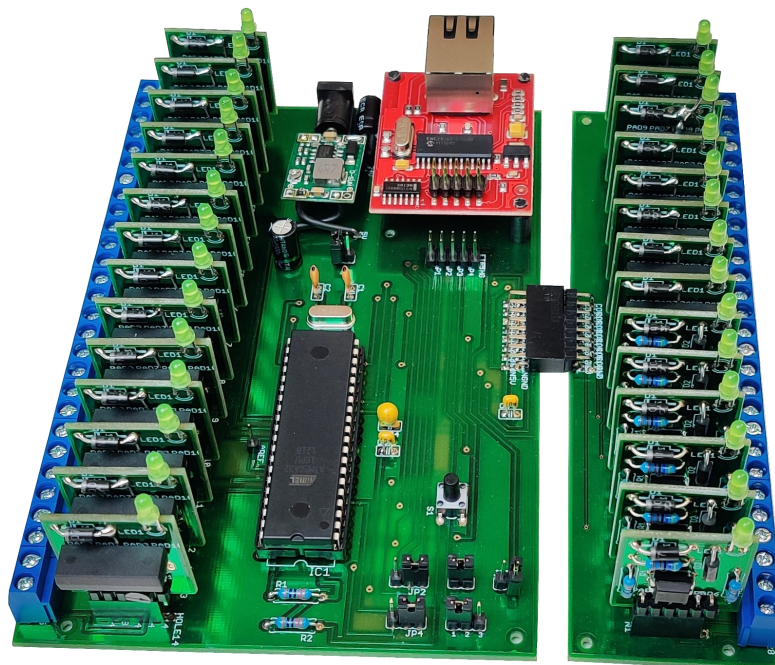
Periferie používají 5 pinů, proto byla šestá nožička využita jako pomůcka pro správné zapojení. Tato nožička musí být vždy nad prázdnou pájecí ploškou (viditelná na obrázku 5.6). Na daném obrázku se vlevo nachází výstupní periferie a vpravo vstupní.



Obr. 5.6: Realizace periférií

## 5.4 Finální produkt

V této sekci je ukázána finální podoba plně osazeného zařízení. Při plném osazení toto zařízení disponuje dvaceti osmi perifერიemi a ethernetovou komunikací. Pro přehlednost fotografie není modul ethernetu zapojen, avšak jeho zapojení lze najít v kapitole 5.1.1 na obrázku 5.2.



Obr. 5.7: Finální produkt

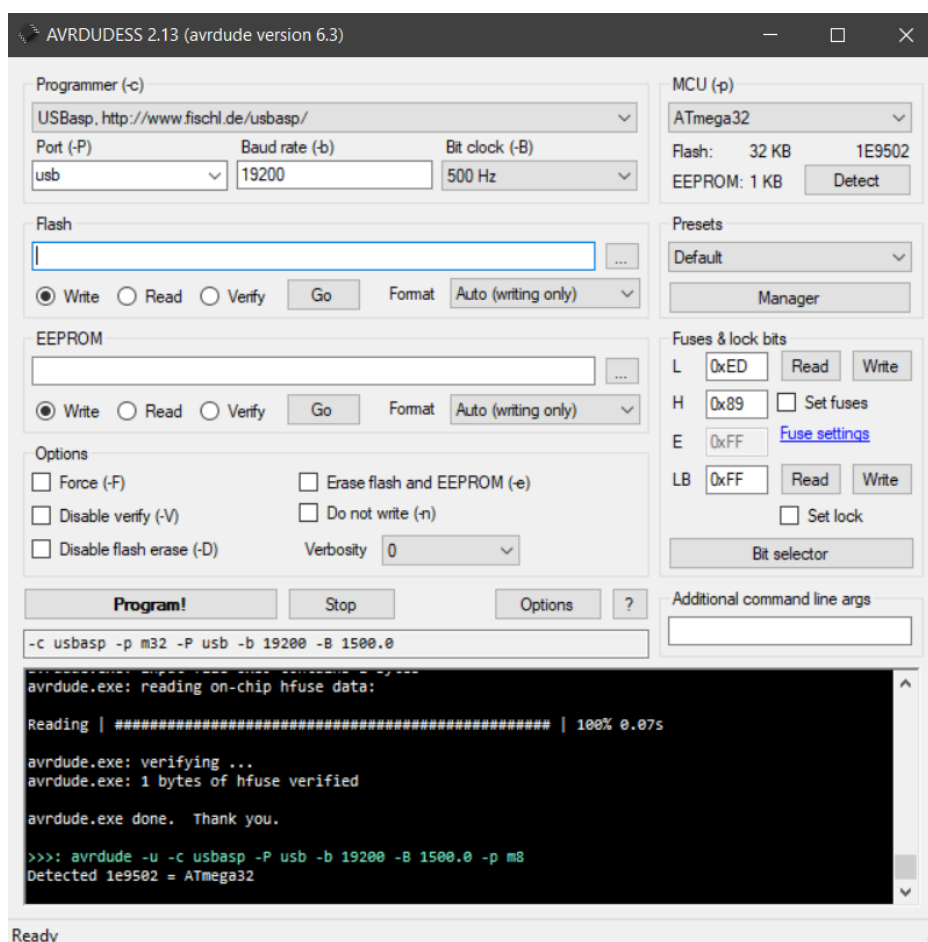


## 6 Software pro rozhraní

Následující část je věnována softwarové části této práce a to na úrovni programu pro mikrokontroler, ale i na úrovni programů pro práci se zařízením. Mezi důležité programy patří microchip studio a avrdude. Microchip studio je programovací prostředí určeno pro AVR mikrokontrolery. V tomto programu je možno programovat v C/C++ a obsahuje důležité knihovny jako avr/io.h, která je nezbytnou součástí pro práci s vstupy a výstupy mikrokontroleru. Další důležitou funkcí je, že program po úspěšném přeložení vygeneruje soubor formátu .hex, který je nezbytně důležitý pro nahrání programu přes AVRDUDE.

### 6.1 AVRDUDE

AVRDUDE je program, starající se o nahrávání .hex souboru do daného mikrokontroleru. Prvním úkonem je správné nastavení požadovaného programátoru, v tomto případě USBasp. Následně se nastaví Port, Baud rate a Bit clock viz. obrázek 6.1.

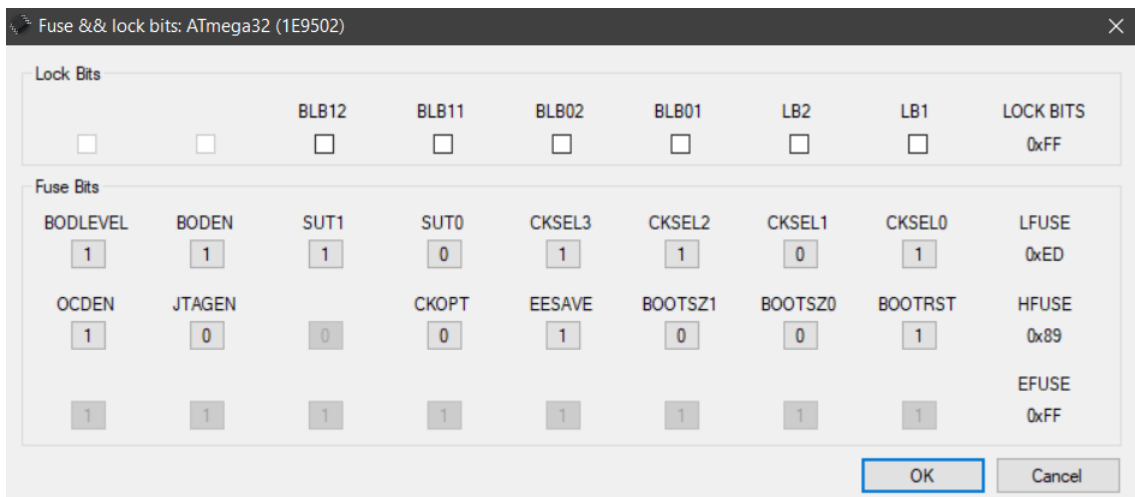


Obr. 6.1: Nastavení programu AVRDUDE

Bit clock je zle nastaven na nejnižší hodnotu. Tato hodnota je nejbezpečnější za cenu dlouhého nahrávání. Pro zrychlení lze použít hodnotu 16 kHz. Pokud je korektně zapojen programátor, pak je možno kliknout na tlačítko Detect v sekci MCU. Pokud je vše v pořádku, bude k konzoly vypsána hláška Detected a název použitého mikrokontroleru, viz. konzole na obrázku 6.1. Následně je potřeba vybrat vhodnou paměť a nastavit cestu k .hex souboru. Posledním důležitým krokem je nastavení Fuses.

### 6.1.1 Fuses

Fuses nastavení je velmi důležitou částí programování mikrokontroleru. Je to však také jednou z nejvíce nebezpečných částí a je tedy velmi důležité mít jistotu správnosti. Při špatném nastavení dojde k zamknutí daného mikrokontroleru, tedy nebude možno ho znovu programovat. V nejhorších případech může dojít k až úplnému zničení. Na obrázku 6.2 je nastavení pro 8 MHz externí crystal clock s 4.1 ms zpožděním při zapnutí mikrokontroleru.



Obr. 6.2: Nastavení fuses

Pro potřeby této práce jsou důležité Low bity SUT1-0 a CKSEL3-0, následně H bit CKOPT. Zbytek bitů je ponecháno v základní hodnotě. Další důležitou informací je, že při nastavení bitu do 0, je bit aktivní. Při nastavení do 1, je bit neaktivní. JTAGEN bit je také změněn, ale programově. Tento bit zakazuje JTAG komunikaci a dovoluje tak použít D registr jako vstup nebo výstup. Následující tabulky jsou z datasheetu mikrokontroleru, který lze najít na [17].

Tab. 6.1: Výběr zařízení pro hodinový takt

Možná zařízení	CKSEL3..0
Externí krystal / keramický rezonátor	1111 - 1010
Externí nízkofrekvenční krystal	1001
Externí RC oscilátor	1000 - 0101
Interní RC oscilátor	0100 - 0001
Externí hodiny	0000

Základním nastavením je použití interního RC oscilátoru s taktem 1 Mhz. Tento takt je při nastavení CKSEL3..0 = 0001 a je možno ho zvednout až na 8 Mhz nastavením na CKSEL3..0 = 0100. Avšak v ideálním případě má tento takt stabilitu  $\pm 1\%$ . Proto zde byl použit externí přesnější krystal také o hodnotě 8 MHz (CKSEL3..0 = 1101). Tento krystal je možno následně změnit až na hodnotu 16 MHz pro CKSEL3..0 = 1111. Důvodem pro nepoužití 16 MHz byla menší frekvenční stabilita. První 3 čísla jsou nastaveny dle tabulky 6.2 a poslední dle 6.3

Tab. 6.2: Nastavení CKOPT vůči frekvenci

CKOPT	CKSEL3..1	Frekvenční rozsah (MHz)
1	110	0.9 - 3.0
1	111	3.0 - 8.0
0	101,110,111	1.0 $\leq$

Dle tabulky 6.2 byl nastaven CKOPT do 0. Toto nastavení by pak odpovídalo i pro použití 16 MHz krystalu. CKSEL poté odpovídá frekvencím 1 MHz pro nastavení bitů do 101, 8 MHz pro 110 a 16 MHz pro 111.

Tab. 6.3: Nastavení CKSEL0 a SUT

CKSEL0	SUT1..0	Čas spuštění po zapnutí	Zpoždění po resetu
1	00	1K CK	65 ms
1	01	16K CK	–
1	10	16K CK	4.1 ms
1	11	16K CK	65 ms

Pozn. CKSEL0 je pouze pro keramické rezonátory. CK je perioda hodinového taktu.



## 6.2 Oživení základové desky a periférií programem

K prvnímu oživení základní desky a periférií byl napsán program kontrolující vstup a pinu A0. Pokud došlo k přivedení 24V na vstupní periférii, mikrokontroler to vyhodnotil a sepnul výstupy A1 až A7. Tímto způsobem bylo možné rychle ověřit funkčnost všech vstupních a výstupních periférií. Obdobným způsobem byli otestovány všechny registry, aby bylo možné konstatovat základní funkčnost hardwaru. Výhodou tohoto oživení poté bylo bližší seznámení s prací s vstupy a výstupy.

Výpis 6.1: Oživovací program blink

```
#define F_CPU 8000000UL      //definice taktu mikrokontroleru 1
#include <avr/io.h>         //knihovna pro vstupy/výstupy      2
#include <util/delay.h>     //knihovna zpoždění              3
                                                                    4
int main(void)             5
{                                                                    6
  DDRA = 0b11111110;      //A0 = vstup, A1-7 = výstup        7
                                                                    8
  while(1)                 9
  {                                                                    10
    if((PINA & 0b00000001) == 1) //kontrola vstupu          11
    {                                                                    12
      PORTA = 0b11111110;    //zapnutí výstupů              13
    }                                                                    14
    else                     15
    {                                                                    16
      PORTA = 0b00000000;    //vypnutí výstupů            17
    }                                                                    18
  }                                                                    19
}                                                                    20
```

## 6.3 Firmware pro ATMEGU32

Jako finální program, byl využit již vytvořený kód z diplomové práce [14]. Práce se zabývala stejnou problematikou a tak zde byla již vytvořena knihovna MODBUS a také vyřešený protokol TCP a ICMP. Pro komunikaci jsou zde použity čtyři knihovny `enc28j60`, `ip_config`, `ip_arp_udp_tcp` a `net`. Knihovna `enc28j60` zajišťuje ovládání ethernetového modulu. Knihovna `ip_config` obsahuje definice pro knihovnu `ip_arp_udp_tcp`. Ta pak slouží k řízení protokolů ARP, TCP a ICMP. Poslední knihovnou je `net`, který je využit pro inicializaci komunikace. Původní neupravená verze těchto knihoven lze najít na [26].

Další z knihoven byly použity knihovny `Modbus`, `FC_IO`, `Casovac` a `Pamet`. Tyto knihovny bylo potřeba upravit pro tuto práci, protože využívali knihovnu `Stav`, jejíž použití bylo nežádoucí. Tato knihovna byla určena pro přepínání pracovních módů.

**Casovac** nese informace o hodinovém taktu, který je použit, tak aby bylo možné přesně čas.

**Knihovna Pamet** spravuje dvě paměťová místa. První místo je určeno pro ukládání aktuálních hodnot výstupů. Druhé pro ukládání dat mezi mikrokontrolerem a ethernetovým modulem. Přesnější informace a popis jednotlivých funkcí lze najít v diplomové práci [14] na straně 48 až 52.

**FC\_IO** slouží pro ovládání vstupů a výstupů. Obsahuje funkce pro zápis na daný port a funkce pro čtení vstupní hodnoty. Přesnější informace a popis jednotlivých funkcí lze najít v diplomové práci [14] na straně 43 až 48.

**Modbus** zpracovává přijaté Modbus TCP pakety. Z daného paketu zjistí jakou funkci má vykonat, provede ji a otestuje zdali vše proběhlo korektně. Následně na tento požadavek odpoví, že vše proběhlo korektně a nebo vrátí chybové hlášení proč nebylo možné tento požadavek splnit. Přesnější informace a popis jednotlivých funkcí lze najít v diplomové práci [14] na straně 53 až 64.

Soubor `main` obsahuje funkce pro volání inicializace ethernetového modulu, dále volá funkce pro zpracování příchozích paketů a funkce pro odpovídání. Tento soubor byl upraven tak, že z něm byla použita pouze vyjmenovaná část. Druhou část tvořil podprogram, který využíval knihovnu `Stav` pro možnost automatického vypnutí přenosu a zapnutí předdefinovaného chování výstupů.

### 6.3.1 Testování komunikace

Pomocí příkazu ping v příkazovém řádku byla ověřena funkčnost ARP a ICMP protokolu (echo reply). Podrobnější výpis komunikace byl získán programem Wireshark, viz 6.4.

```
C:\Users\pajah>ping 192.168.0.50

Pinging 192.168.0.50 with 32 bytes of data:
Reply from 192.168.0.50: bytes=32 time=5ms TTL=64
Reply from 192.168.0.50: bytes=32 time=2ms TTL=64
Reply from 192.168.0.50: bytes=32 time=2ms TTL=64
Reply from 192.168.0.50: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.0.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 5ms, Average = 2ms

C:\Users\pajah>
```

Obr. 6.3: Výpis z příkazového řádku na příkaz ping

Dále byl využit program Wireshark pro sledování paketů. Wireshark nabízí možnost kontroly, že je ve funkcích správně nastavený popis protokolu a náhled co jednotlivý paket obsahuje. Tato možnost se zvláště hodí při kontrole MODBUS komunikace.

Source	Destination	Protocol	Length	Info
192.168.0.101	192.168.0.50	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (reply in 6057)
192.168.0.50	192.168.0.101	ICMP	74	Echo (ping) reply id=0x0001, seq=18/4608, ttl=64 (request in 6056)
192.168.0.101	192.168.0.50	ICMP	74	Echo (ping) request id=0x0001, seq=19/4864, ttl=128 (reply in 6143)
192.168.0.50	192.168.0.101	ICMP	74	Echo (ping) reply id=0x0001, seq=19/4864, ttl=64 (request in 6142)
192.168.0.101	192.168.0.50	ICMP	74	Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 6261)
192.168.0.50	192.168.0.101	ICMP	74	Echo (ping) reply id=0x0001, seq=20/5120, ttl=64 (request in 6260)
192.168.0.101	192.168.0.50	ICMP	74	Echo (ping) request id=0x0001, seq=21/5376, ttl=128 (reply in 6326)
192.168.0.50	192.168.0.101	ICMP	74	Echo (ping) reply id=0x0001, seq=21/5376, ttl=64 (request in 6325)

Obr. 6.4: Výpis z Wiresharku na příkaz ping

Source	Destination	Protocol	Length	Info
192.168.0.50	192.168.0.101	Modbus...	66	Response: Trans: 2; Unit: 1, Func: 15: Write Multiple Coils
192.168.0.101	192.168.0.50	TCP	54	64019 → 502 [ACK] Seq=43 Ack=37 Win=64204 Len=0
192.168.0.101	192.168.0.50	Modbus...	68	Query: Trans: 3; Unit: 1, Func: 15: Write Multiple Coils
192.168.0.50	192.168.0.101	TCP	60	502 → 64019 [ACK] Seq=37 Ack=57 Win=1024 Len=0
192.168.0.50	192.168.0.101	Modbus...	68	Response: Trans: 3; Unit: 1, Func: 15: Write Multiple Coils
192.168.0.101	192.168.0.50	TCP	54	64019 → 502 [ACK] Seq=57 Ack=49 Win=64192 Len=0
192.168.0.101	192.168.0.50	Modbus...	68	Query: Trans: 4; Unit: 1, Func: 15: Write Multiple Coils

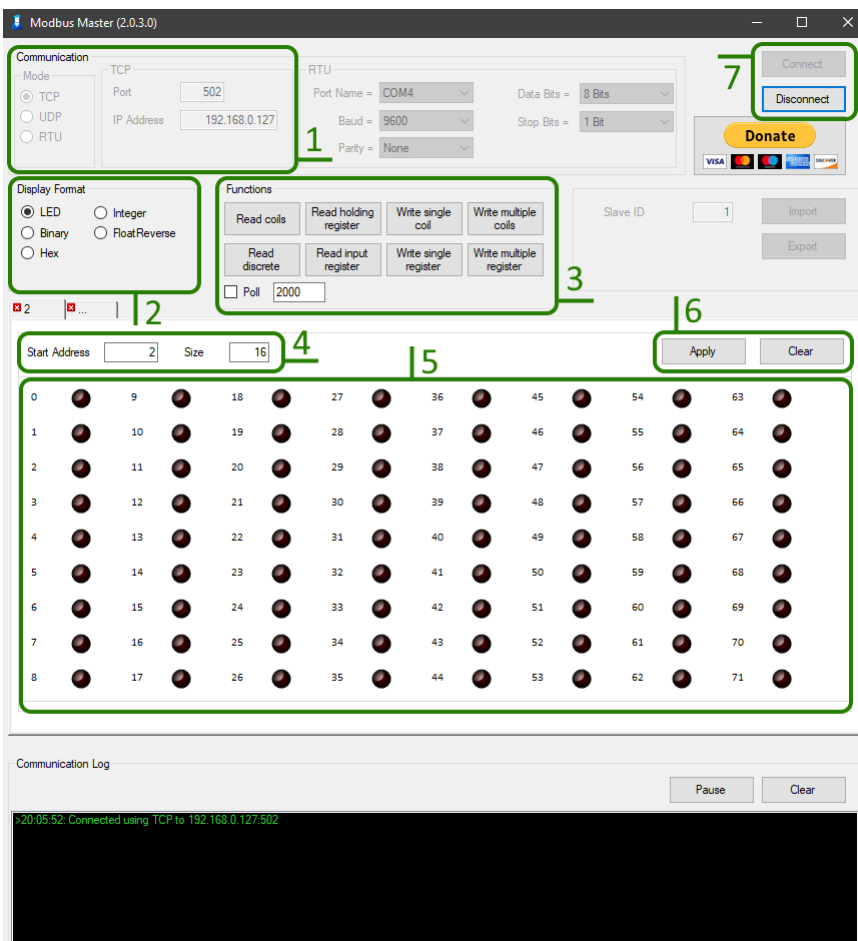
0000	62 5f 70 72 61 79 ac ed 5c cf c0 74 08 00 45 00	b_pray... \..t..E..
0010	00 36 54 06 40 00 80 06 24 d4 c0 a8 00 65 c0 a8	.6T @... \$....e..
0020	00 32 fa 13 01 f6 52 4c ad eb 00 00 0a 25 50 18	.2....RL ....%P.
0030	fa cc 2a 80 00 00 00 03 00 00 00 08 01 0f 00 01	..*... .....
0040	00 08 01 00	....

Obr. 6.5: Výpis z Wiresharku na Write Multiple Coils

Ve spodní části obrázku 6.5 lze pozorovat označenou část paketu. Tato část lze pak přirovnat k dané funkci v příloze A.1 až A.8. Tedy určit validnost zprávy.

## 6.4 Modbus tool

Program Modbus tool od firmy codeplex je simulátorem master i slave zařízení komunikujícím po MODBUS/TPC a je dostupný z [23]. V této práci byl využit pro kontrolu funkčnosti finálního firmwaru zařízení.



Obr. 6.6: Modbus master tool

skládá se ze 7 oken:

- 1 → Nastavení komunikace, portu a IP adresy požadovaného serveru
- 2 → Nastavení formátu v jakém je možno zapisovat hodnotu do 5
- 3 → 7 funkcí pro čtení a zapisování vstupů a výstupů
- 4 → Nastavení velikosti a počáteční adresy pro funkce
- 5 → LED pole reprezentující hodnoty pinů
- 6 → Apply tlačítko pro uložení nastavení a přípravu na odeslání požadavku
- 7 → Tlačítko připojení a odpojení od serveru

Postup práce s programem je pak následující. Nejprve se nastaví komunikace, port a IP adresa v kolonce **1**. Dále se připojí k serveru v kolonce **7**. Poté se vybere způsob zápisu hodnot v kolonce **2**. Následuje nastavení kolonky **4** a **6** a následného potvrzení přes apply v **6**. Posledním krokem je stisknutí potřebné funkce v kolonce **3**.

### 6.4.1 Nastavení pro jednotlivé funkce

**Funkce 1 - Read coils** slouží k vyčtení hodnoty z výstupního registru C a D.

- Size musí být nastaven na hodnotu 1
- Start adress je pak možno nastavit od 0 do 15, kdy 0 představuje pin  $C_0$  a 15  $D_7$

**Funkce 2 - Read discrete** složí ke čtení hodnoty na vstupu.

- Size musí být nastaven na hodnotu 1
- Start adress je pak možno nastavit od 0 do 7, kdy 0 představuje pin  $A_0$  a 7  $A_7$

**Funkce 3 - Read holding register** složí ke čtení hodnot na výstupů.

- Size musí být nastaven na hodnotu 1
- Start adress je pak možno nastavit od 0 do 2, kdy 0 představuje registr C, 1 poté registr D a 2 registr C i D.

**Funkce 4 - Read input register** složí k vyčtení hodnot portu A

- Size musí být nastaven na hodnotu 1
- Start adress musí být nastavena na hodnotu 0

**Funkce 5 - Write single coil** složí k zápisu hodnoty jednoho výstupu

- Size musí být nastaven na hodnotu 1
- Start adress je možno nastavit od 0 do 15, kde 0 představuje port  $C_0$  a 15  $D_7$
- Následně je nutné zaznačit v kolonce **5** na obrázku 6.6 led s označením 0, jestli má být port v logické 1 nebo 0.

**Funkce 6 - Write single register** složí k zápisu hodnoty všech výstupů jednoho registru

- Size musí být nastaven na hodnotu 1
- Start adress musí být nastavena na hodnotu 0
- Následně je nutné zaznačit v kolonce **5** na obrázku 6.6 led s označením 0 až 15 jestli má být daný port v logické 1 nebo 0. Zde led 0 označuje  $C_0$  a led 15  $D_7$ .

**Funkce 7 - Write multiple coils** složí k zápisu hodnoty všech výstupů jednoho registru

- Size musí být nastaven na hodnotu 8
- Start adress je možno nastavit od 0 do 1, kde 0 představuje registr C a 1 registr D
- Následně je nutné zaznačit v kolonce **5** na obrázku 6.6 led s označením 0 až 7 jestli má být daný port v logické 1 nebo 0. Zde led 0 označuje nultý port registru.

## 7 Návrhy na zlepšení

Tato kapitola je věnována možným vylepšením, které by mohli být v budoucích pracích udělány. Jedná se o věci vyzpozorované během vývoje, které nebyli stěžejní pro požadované výsledky práce a nebylo je z časového hlediska udělat.

První z těchto věcí je náhrada jumper pinů viditelných na obrázku 5.3 za pás posuvných spínačů. Důvodem použití jumper pinů byla jejich cena, která byla doprovázena myšlenkou, že do zařízení se v budoucnu program nahraje a poté již nebude potřeba přepínání. Během ladění programu se tento problém projevoval dlouhým časem nutným pro přípravu na programování.

Dalším problémem je použití ISP programátoru. Ten byl vybrán z důvodu šetření pinů, protože během programování není potřeba mít připojen ethernetový modul. Velkým problémem se ukázala nemožnost debugování. Při ladění tak nemohlo být zjištěno, co se v programu odehrává, což velmi ztížilo danou práci. Možným vylepšením by tedy bylo nahrazení ISP programátoru za programátor JTAG, který možnost debugu nabízí.

Poslední změnou týkající se hardwaru, je úprava SPI patice viditelné na obrázku 5.2 tak, aby bylo možné ethernet modul otočit a přímo zasunout. Tímto způsobem se zlepší přenos tím, že zde nebude narušována integrita přenosu rušením ve vodičích.

V rámci programové části upravit knihovnu modbus tak, aby bylo možné dynamicky určit, kde se bude nacházet výstup a kde vstup. V současné podobě je pro vstupy vyhrazen port A a pro výstupy C a D.

Nakonec upravit knihovnu FC\_IO tak, aby načítala informace o pozici vstupů a výstupů z konfiguračního souboru uloženého na EEPROM. Tímto způsobem by pak mohlo být dosaženo zrychlení při změně hardwarové konfigurace. Znamenalo by, že program může být zapsán v paměti FLASH bez nutnosti změny a nahrával by se pouze konfigurační soubor do EEPROM.

## Závěr

V této práci proběhlo seznámení s problematikou řízení křižovatek. Následně proběhlo seznámení s používanými průmyslovými komunikacemi, přesněji pak s komunikací Modbus/TCP. Po seznámení bylo nutné sepsat všechny požadavky na finální produkt a jeho funkčnost. Tyto požadavky pak dali náhled, kterým směrem zařízení směřovat. Základem návrhu byl výběr vhodného mikrokontroleru, jímž se stal procesor z rodiny AVR, přesněji tedy ATMEGA32. Pro tento procesor pak bylo převzato zapojení z vývojového kitu.

Toto zapojení se pak stalo hlavní částí základní desky. Následoval návrh výstupních periférií. Pro jednoduchost a robustnost bylo nakonec vybráno zapojení vybavené mini relé. Relé bylo nutné řídit tak, aby nezatěžovalo mikrokontroler a bylo tedy vybaveno N-kanálovým tranzistorem. Zde došlo k chybě v návrhu, která byla pro tuto práci již opravena a setrvává tak pouze na fyzickém kusu již vyrobených výstupních perifériích. Velikost a zapojení výstupní periferie se pak stalo předlohou pro vstupní periferii. Zde působí jako spínací prvek a současně jako galvanické oddělení optočlen.

Základní deska navrženého zařízení bylo následně rozdělena pro využití, kde je potřeba méně periférií. Vznikla tak tedy deska pro rozšíření základní desky. Zde došlo také k chybě v návrhu, která je již opravena a setrvává tak pouze na již vyrobené desce rozšíření. Následně bylo celé zařízení oživeno a byla otestována jeho funkčnost. Po otestování byl čas připojit ethernetový modul. Byl vybrán modul s čipem enc28j60 od stejné firmy od které bylo použito zapojení pro mikrokontroler.

Následující úkoly obsahovali pouze práci s kódem. Kód byl převzat s diplomové práce, která se zabývala stejnou problematikou. Tento kód byl následně upraven tak, aby odpovídal požadavkům této práce. Po odladění kódu byla otestována funkčnost pomocí programu Wireshark a Modbus tool.

# Literatura

- [1] ZÁVIŠ, Jan. Model soustavy křižovatek [online]. Brno, 2018 [cit. 2021-10-30]. Dostupné z URL: <[https://www.vut.cz/studenti/zav-prace?zp\\_id=111135#\\_ga=2.103198340.508419854.1635603246-662265911.1630413204](https://www.vut.cz/studenti/zav-prace?zp_id=111135#_ga=2.103198340.508419854.1635603246-662265911.1630413204)>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Radek Štohl.
- [2] PÁSEK, Jan. Programovatelné automaty v řízení technologických procesů [online]. Brno, 30.11. 2007 [cit. 2021-12-25]. Dostupné z URL: <[https://www.vut.cz/www\\_base/priloha\\_fs.php?dpid=185135&skupina=dokument\\_priloha](https://www.vut.cz/www_base/priloha_fs.php?dpid=185135&skupina=dokument_priloha)>. Scripta. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky.
- [3] MICHALEC, Libor. Ethernet vs. sériové protokoly v průmyslu [online]. Praha, 4. 10. 2018 [cit. 2021-12-25]. Dostupné z URL: <<https://automatizace.hw.cz/ethernet-vs-seriove-protokoly.html>>.
- [4] FILIP, Jiří. Beckhoff real-time Ethernet na MSV [online]. Děčín, 2016 [cit. 2021-12-25]. Dostupné z URL: <<https://tinyurl.com/3ywb2ptp>>.
- [5] ZEŽULKA, František. HYNČICA Ondřej. Průmyslový Ethernet [online]. Děčín, 2005 [cit. 2021-12-25]. Dostupné z URL: <[https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-2005\\_04\\_30417\\_493/](https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-2005_04_30417_493/)>.
- [6] BÍLEK, Karel. Ethernet Powerlink [online]. Děčín, 2001 [cit. 2021-12-26]. Dostupné z URL: <[https://automa.cz/cz/casopis-clanky/ethernet-powerlink-komunikace-v-realnem-case-2001\\_03\\_33500\\_2769/](https://automa.cz/cz/casopis-clanky/ethernet-powerlink-komunikace-v-realnem-case-2001_03_33500_2769/)>.
- [7] ZEŽULKA, František. HYNČICA Ondřej. Průmyslový Ethernet CC-Link IE controller network [online]. Děčín, 2009 [cit. 2021-12-26]. Dostupné z URL: <[https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-cc-link-ie-controller-network-2009\\_05\\_38986\\_5912/](https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-cc-link-ie-controller-network-2009_05_38986_5912/)>.
- [8] elektroprumysl.cz . Průmyslový Ethernet [online]. Brno, 12. 1. 2014 [cit. 2021-12-26]. Dostupné z URL: <<https://www.elektroprumysl.cz/automatizace/prumyslovy-ethernet>>.



- [9] iPC2U s.r.o. . Podrobný popis protokolu Modbus TCP s příklady příkazů [online]. Praha, 14. 12. 2020 [cit. 2021-12-26]. Dostupné z URL: <<https://ipc2u.cz/blogs/news/podrobny-popis-protokolu-modbus-tcp-s-priklady-prikazu>>.
- [10] Acromag, Inc. . INTRODUCTION TO MODBUS TCP/IP [online]. Wixom, MI 48393-7037 U.S.A., 2005 [cit. 2021-12-26]. Dostupné z URL: <[https://www.prosoft-technology.com/kb/assets/intro\\_modbustcp.pdf](https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf)>.
- [11] GM electronic s r. o. . Modul DC/DC měnič step-down 1-17V/2A, miniaturní [online]. Praha, [cit. 2021-4-25]. Dostupné z URL: <<https://www.gme.cz/step-down-napajeci-modul-1-17v-2a-miniaturni#product-detail>>.
- [12] LaskaKit . USBASP - ISP programátor pro ATMEL [online]. Rychnov nad Kněžnou, [cit. 2021-12-26]. Dostupné z URL: <[https://www.laskakit.cz/usbasp-isp-programator-pro-atmel/?gclid=Cj0KCQiAmeKQBhDvARIsAHJ7mF5yLJd4xz-b7QPMYTXAS\\_K-YvX3bNwJ3oD9S\\_5BCVcsCsU7rqf0VgQaAoEjEALw\\_wcB](https://www.laskakit.cz/usbasp-isp-programator-pro-atmel/?gclid=Cj0KCQiAmeKQBhDvARIsAHJ7mF5yLJd4xz-b7QPMYTXAS_K-YvX3bNwJ3oD9S_5BCVcsCsU7rqf0VgQaAoEjEALw_wcB)>.
- [13] GM electronic s r. o. . Vývojový kit EvB 5.1 - ATmega32 [online]. Praha, [cit. 2021-12-26]. Dostupné z URL: <<https://www.gme.cz/vyvojovy-kit-evb-5-1-atmega32>>.
- [14] ZÁVIŠ, Jan. Model křižovatky s průmyslovou komunikací [online]. Brno, 2021 [cit. 2021-12-26]. Dostupné z URL: <[https://www.vut.cz/studenti/zav-prace?zp\\_id=134528#\\_ga=2.157456186.1417805463.1640645764-662265911.1630413204](https://www.vut.cz/studenti/zav-prace?zp_id=134528#_ga=2.157456186.1417805463.1640645764-662265911.1630413204)>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Radek Štohl.
- [15] GM electronic s r. o. . LED 3mm, zelená, 3,5mcd, 35°, difúzní BL-B2141-L [online]. Praha, [cit. 2021-12-26]. Dostupné z URL: <<https://www.gme.cz/data/attachments/dsh.511-200.2.pdf>>.
- [16] TME [online]. Łódź, Polsko: Transfer Multisort Elektronik Sp. z o.o., [cit. 2021-12-27]. Dostupné z URL: <[https://www.tme.eu/cz/katalog/mikrokontrolery-a-mikroprocesory\\_100590/](https://www.tme.eu/cz/katalog/mikrokontrolery-a-mikroprocesory_100590/)>.
- [17] GM electronic s r. o. . Mikrokontrolér AVR, EEPROM: 1kB, SRAM: 2kB, flash: 32kB, THT ATMEGA32-16PU [online]. Praha, [cit. 2021-12-27]. Dostupné z URL: <<https://www.gme.cz/data/attachments/dsh.432-180.1.pdf>>.

- [18] GM electronic s r. o. . Modul Ethernet AND-ETH v2 [online]. Praha, [cit. 2021-12-27]. Dostupné z URL: <<https://www.gme.cz/modul-ethernet-and-eth-v2>>.
- [19] VŠB-Technická univerzita Ostrava. Zvýšení vědeckovýzkumného potenciálu pracovníků a studentů technických vysokých škol v oblasti dopravy [online]. Ostrava, 2009 [cit. 2021-12-27]. Dostupné z URL: <[http://projekt150.ha-vel.cz/node/6?fbclid=IwAR2t5Idl\\_5IPCXMe6Qo\\_95aZVxa65nySJq6tYFz8wILtsett1IaAn4x04fM](http://projekt150.ha-vel.cz/node/6?fbclid=IwAR2t5Idl_5IPCXMe6Qo_95aZVxa65nySJq6tYFz8wILtsett1IaAn4x04fM)>.
- [20] SMĚLÝ, Martin. DOPRAVNÍ INŽENÝRSTVÍ, ŘÍZENÉ ÚROVŇOVÉ KŘÍŽOVATKY ČÁST 1 [online]. Brno, 2007 [cit. 2021-12-27]. Dostupné z URL: <[http://lences.cz/domains/lences.cz/skola/subory/Skripta/CM04-Dopravn%C3%AD%20in%C5%BEen%C3%BDrstv%C3%AD%20\(DST\)/M04-%C5%98%C3%ADzen%C3%A9%20%C3%BArov%C5%88ov%C3%A9%20k%C5%99i%C5%BEovatky%20I.pdf](http://lences.cz/domains/lences.cz/skola/subory/Skripta/CM04-Dopravn%C3%AD%20in%C5%BEen%C3%BDrstv%C3%AD%20(DST)/M04-%C5%98%C3%ADzen%C3%A9%20%C3%BArov%C5%88ov%C3%A9%20k%C5%99i%C5%BEovatky%20I.pdf)>.
- [21] POLIŠENSKÝ, Karel. Řešení křižovatek [online]. Tábor, [cit. 2021-12-27]. Dostupné z URL: <<https://www.naucseridit.cz/reseni-krizovatek/>>.
- [22] AND-TECH, EvB 5.1 v5 Uživatelská příručka [online]. [cit. 2022-4-18]. Dostupné z URL: <<https://www.gme.cz/data/attachments/czn.772-044.1.pdf>>.
- [23] Codeplex, Codeplex modbus tool [online]. [cit. 2022-5-18] Dostupné z URL: <<https://www.modbustools.com/download.html>>.
- [24] AVR TCP/IP Examples. Github [online]. San Francisco(Kalifornie, USA): GitHub, © 2021 [cit. 2022-5-17]. Dostupné z URL: <<https://github.com/ptrks/AVR-ENC28J60-Examples>>.
- [25] GM electronic s r. o. . Optočlen, výstup: tranzistorový, izolační pevnost 8,2kV, THT K10101D [online]. Praha, [cit. 2022-5-18]. Dostupné z URL: <<https://www.gme.cz/data/attachments/dsh.523-226.1.pdf>>.
- [26] AVR TCP/IP Examples. Github [online]. San Francisco(Kalifornie, USA): GitHub, © 2021 [cit. 2022-5-18]. Dostupné z URL: <<https://github.com/ptrks/AVR-ENC28J60-Examples>>.

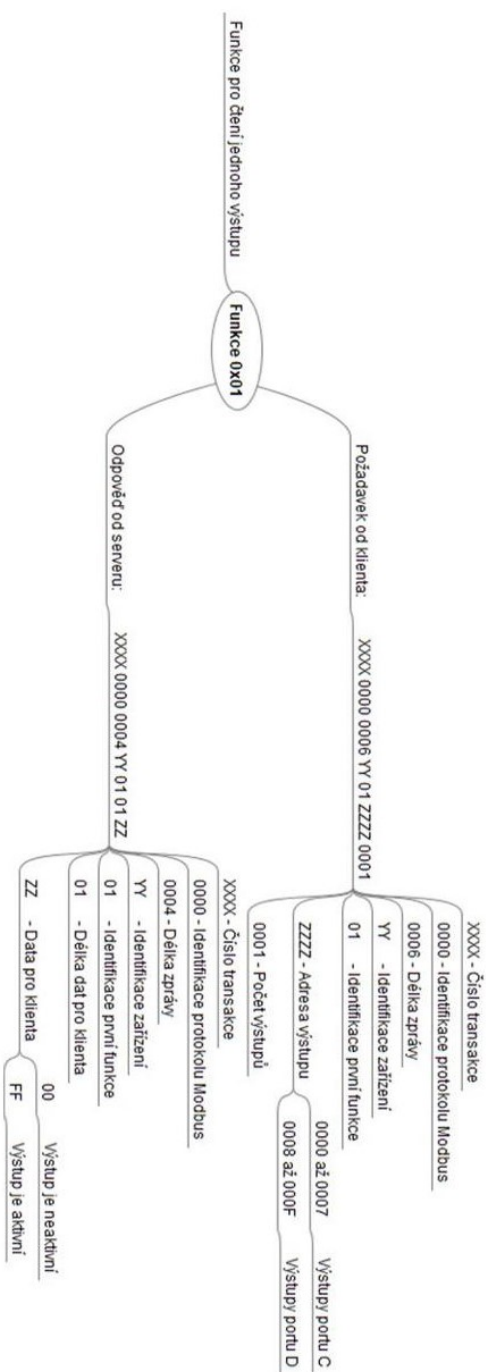
## Seznam symbolů a zkratek

<b>Mb/s</b>	megabyte za sekundu
<b>Kb/s</b>	kilobyte za sekundu
<b>DPS</b>	deska plošného spoje
<b>W</b>	watt
<b>V</b>	volt
<b>U</b>	značka pro napětí
<b>I</b>	značka pro proud
<b>R</b>	značka pro odpor
<b>P</b>	značka pro výkon
<b>mA</b>	miliampér
<b>A</b>	ampér
<b>SPI = ISP</b>	sériové periferní rozhraní
<b>GND</b>	označení větve s 0V
<b>MHz</b>	megahertz
<b>HW</b>	hardware (fyzické součástky)
<b>SW</b>	software (programová část)
<b>ms</b>	milisekunda

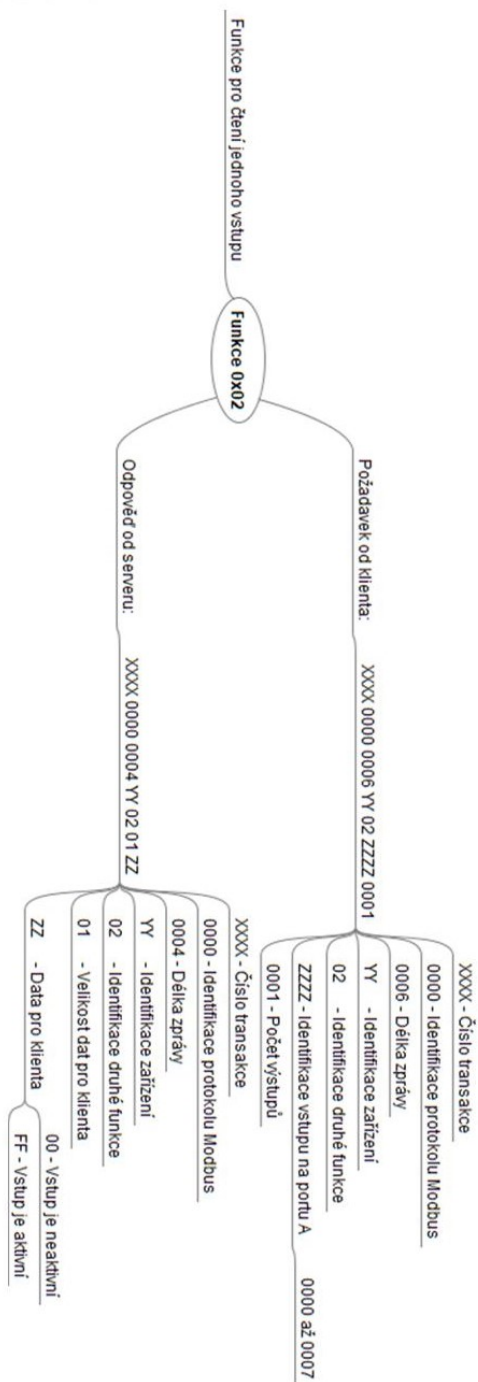
# A Packet Modbus funkcí

Všechny diagramy pro následující podkapitoly byli použity z příloh diplomové práce [24].

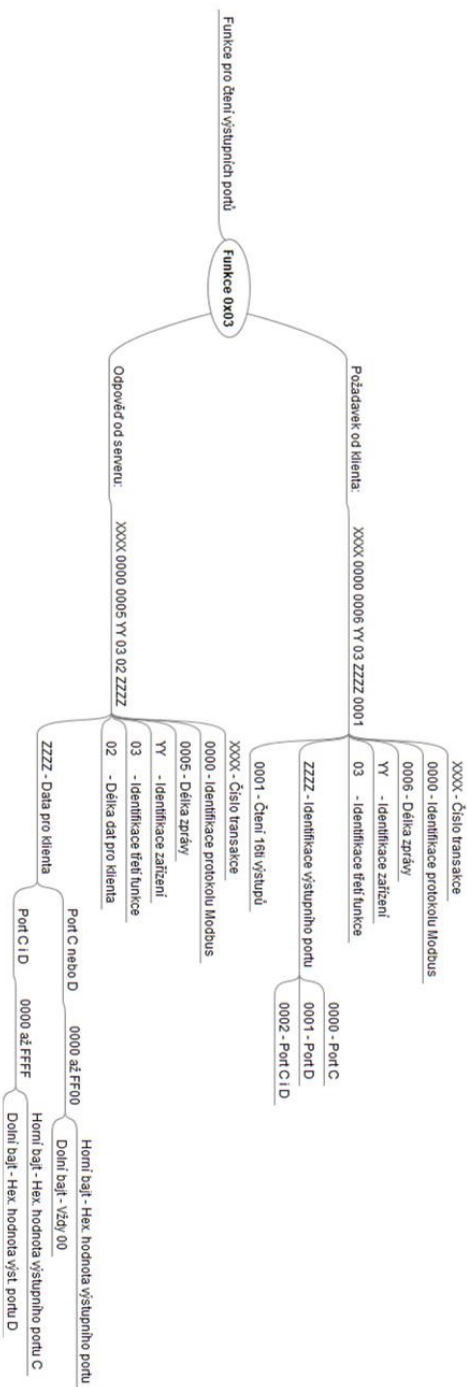
## A.1 Funkce 1



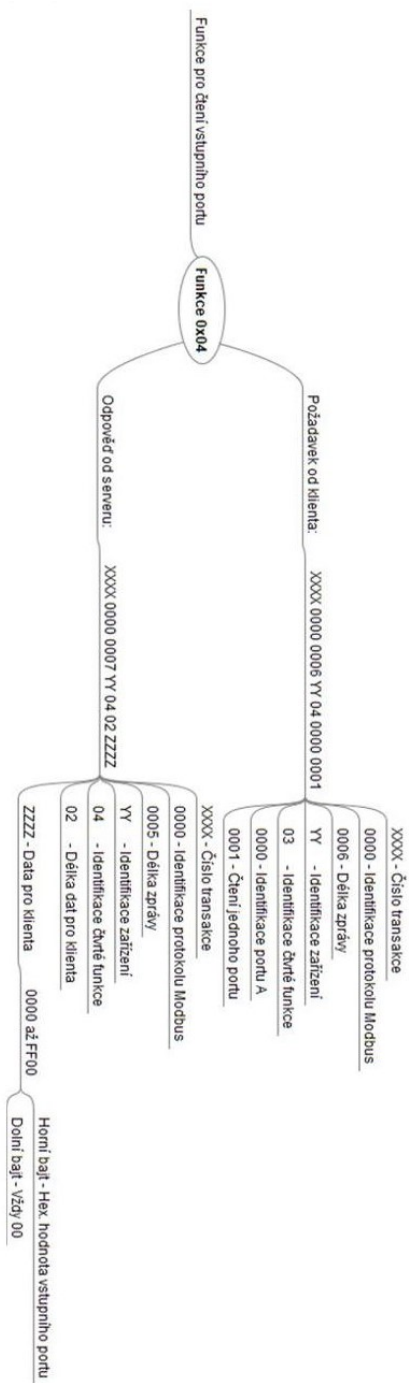
## A.2 Funkce 2



## A.3 Funkce 3



## A.4 Funkce 4

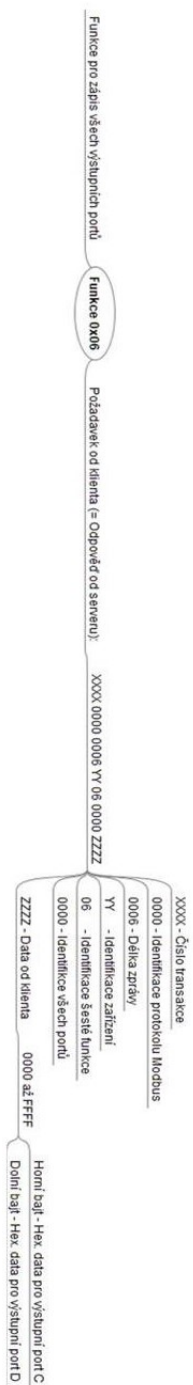


## A.5 Funkce 5

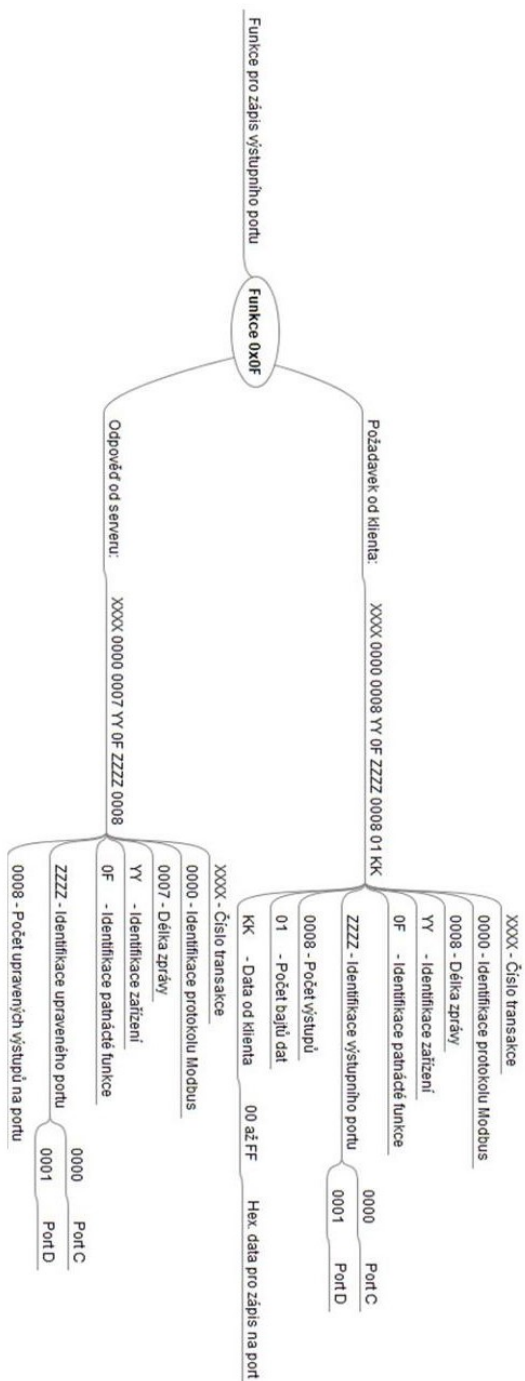




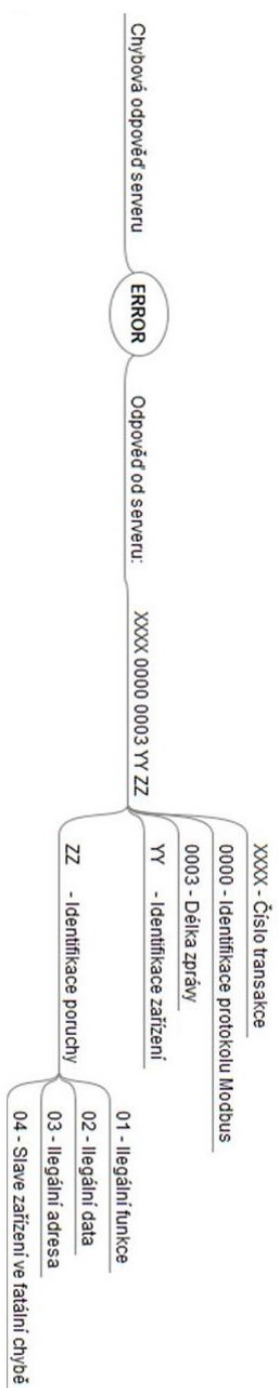
## A.6 Funkce 6



# A.7 Funkce 7



## A.8 Chybové hlášení



## **B Elektronická příloha**

### **B.1 Software pro ATMEGA32**

- Finální verze zdrojového kódu
- Všechny použité knihovny
- Na CD nebo jako elektronická příloha pod názvem - Příloha 1

### **B.2 Návrhy DPS**

- Opravená verze DPS
- Všechny použité návrhy
- Na CD nebo jako elektronická příloha pod názvem - Příloha 2