

# **Embedded komunikační zařízení pro nastavení parametrů a vyčítání hodnot ze simulátoru měřiče rychlosti**

Bc. Pavel Janás

---

Diplomová práce  
2019



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Pavel Janás**  
Osobní číslo: **A17660**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Počítačové a komunikační systémy**  
Forma studia: **prezenční**

Téma práce: **Embedded komunikační zařízení pro nastavení parametrů a včítání hodnot ze simulátoru měřiče rychlosti**

Téma anglicky: **Embedded Communication Device for Setting Parameters and Reading Values from the Speed Meter Simulator**

### Zásady pro vypracování:

1. **Nastudujte komunikační rozhraní RS-232 a definujte soubor parametrů pro nastavení měřiče rychlosti.**
2. **Navrhněte schéma hardwarové desky s výstupem na displej, vstupem definovaného počtu tlačítek a komunikačním rozhraním RS-232.**
3. **Implementujte jednoduchou grafickou PC aplikaci, která bude simulovat činnost měřiče rychlosti a komunikovat pomocí RS-232 rozhraní.**
4. **Implementujte na zvolený mikropočítač definovanou obousměrnou komunikaci pro odeslání hodnot pro nastavení parametrů a jednoznačnou interpretaci výsledků ze simulačního programu měřicího zařízení.**
5. **Vytvořte funkční prototyp ovládacího a zobrazovacího zařízení komunikujícího se simulační aplikací na PC.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **KAINKA, Burkhard. Měření, řízení a regulace pomocí PC: [vývoj hw a sw pro praxi].** Praha: BEN – technická literatura, 2003. PC & elektronika. ISBN 80-7300-089-x.
2. **Opatření obecné povahy: kterým se stanovují metrologické a technické požadavky na stanovená měřidla, včetně metod jejich zkoušení při schvalování typu a ověřování stanovených měřidel: "silniční rychloměry používané při kontrole dodržování pravidel silničního provozu".** In: . Brno: Český metrologický institut, 2010, ročník 2010, 0111-OOP-C005-09. Dostupné z: [https://www.cmi.cz/sites/all/files/public/download/Uredni\\_deska/3405-ID-C\\_3405-ID-C.pdf](https://www.cmi.cz/sites/all/files/public/download/Uredni_deska/3405-ID-C_3405-ID-C.pdf)
3. **PINKER, Jiří. Mikroprocesory a mikropočítače.** Praha: BEN – technická literatura, 2004. ISBN 80-7300-110-1.
4. **JURÁNEK, Antonín a Miroslav HRABOVSKÝ. EAGLE pro začátečníky I: uživatelská a referenční příručka .: 2. vydání.** Praha: BEN – technická literatura, 2007, 191 s. ISBN 80-730-0213-2.
5. **GOOK, Michael. Hardwarová rozhraní: průvodce programátora.** Brno: Computer Press, 2006. ISBN 80-251-1019-2.

Vedoucí diplomové práce:

**Ing. Petr Dostálek, Ph.D.**

Ústav automatizace a řídicí techniky

Konzultant:

**Ing. Josef Filip**

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

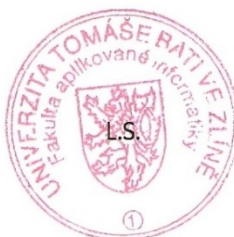
**30. listopadu 2018**

Termín odevzdání diplomové práce:

**17. května 2019**

Ve Zlíně dne 10. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



Ing. Miroslav Matýsek, Ph.D.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 20.5.2019

Pavel Janás, v.r.

## **ABSTRAKT**

Tato diplomová práce se zabývá návrhem a realizací embedded komunikačního zařízení sloužícího k odesílání hodnot pro nastavení parametrů měření simulátoru měřiče rychlosti motorových vozidel a k interpretaci získaných výsledků ze simulátoru. Dalším cílem práce je vytvořit jednoduchou simulační počítačovou aplikaci měřiče v libovolně zvoleném programovacím jazyce. Výstupem této práce je funkční prototyp ovládacího a zobrazovacího zařízení včetně programového vybavení provozovaný na vestavěný akumulátor komunikující s implementovanou simulační aplikací na PC přes Bluetooth rozhraní podporující standard RS-232.

Klíčová slova: RS232, Bluetooth, MC9S08AC128, UART, SPI, sériová komunikace, simulátor.

## **ABSTRACT**

This diploma thesis deals with design and realization of the embedded communication device used for sending values for setting the measurement parameters of the traffic enforcement camera and to interpret the results obtained from the simulator. Another aim of this work is to create a simple simulation computer application of traffic enforcement camera in arbitrarily programming language. The output of this work is a functional prototype of control and display device operating on battery including software solution communicating with implemented simulation application on PC via Bluetooth interface supporting RS-232 standard.

Keywords: RS232, Bluetooth, MC9S08AC128, Traffic enforcement camera, SPI, serial communication, simulator.

Rád bych poděkoval vedoucímu mé diplomové práce panu Ing. Petru Dostálkovi, Ph.D. za odbornou pomoc, cenné rady a vstřícné jednání v průběhu tvorby této práce. Mé poděkování patří také konzultantovi diplomové práce Ing. Josefu Filipovi a Jiřímu Tomeškovi.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 PŘENOS INFORMACE</b> .....	<b>11</b>
1.1 TYPY PŘENÁŠENÝCH INFORMACÍ.....	11
1.1.1 Analogové informace .....	11
1.1.2 Diskrétní informace.....	11
1.2 DRUHY SPOJENÍ.....	11
1.3 SIMPLEXNÍ SPOJENÍ .....	12
1.4 POLODUPLEXNÍ SPOJENÍ .....	12
1.5 PLNĚ DUPLEXNÍ SPOJENÍ.....	13
1.6 SÉRIOVÁ A PARALELNÍ KOMUNIKACE.....	13
1.6.1 Sériová komunikace .....	13
1.6.2 Paralelní komunikace .....	14
<b>2 SÉRIOVÝ PŘENOS</b> .....	<b>15</b>
2.1 SYNCHRONNÍ PŘENOS DAT .....	15
2.2 ASYNCHRONNÍ PŘENOS DAT .....	16
2.2.1 Baud Rate .....	16
2.2.2 Datové bity .....	17
2.2.3 Synchronizační bity.....	17
2.2.4 Paritní bit.....	17
2.2.5 Sériový paket.....	17
2.3 RS-232.....	18
2.3.1 Asynchronní přenos RS-232 .....	19
2.3.2 UART (Universal Asynchronous Receiver-Transmitter) .....	20
2.3.3 UART vs USART rozhraní.....	22
<b>3 BLUETOOTH KOMUNIKACE</b> .....	<b>23</b>
3.1 KOMUNIKACE A PŘIPOJENÍ .....	24
3.2 PROTOKOL RFCOMM .....	25
<b>4 JEDNOČIPOVÉ MIKROPOČÍTAČE</b> .....	<b>26</b>
4.1 MC9S08AC128.....	27
4.1.1 Vlastnosti.....	27
4.1.2 Pracovní režimy: .....	28
4.1.3 Přerušovací subsystém .....	28
4.1.4 Periférie mikrokontroléru.....	29
4.1.4.1 Čítač/časovač .....	29
4.1.4.2 A/D převodník .....	31
4.1.4.3 UART(SCI).....	31
4.1.4.4 Externí přerušení KBI.....	32
4.1.5 Rozmístění pinů .....	32
<b>II PRAKTICKÁ ČÁST</b> .....	<b>35</b>
<b>5 NÁVRH A REALIZACE ZAŘÍZENÍ</b> .....	<b>36</b>

5.1	OBVOD NABÍJENÍ .....	37
5.2	NAPÁJENÍ OVLÁDACÍ A ZOBRAZOVACÍ JEDNOTKY .....	39
5.2.1	Zvyšující DC-DC měnič .....	40
5.3	MIKROKONTROLÉR .....	40
5.3.1	Paralelní I/O .....	41
5.3.2	SCI.....	41
5.3.3	A/D převodník.....	43
5.3.4	KBI přerušení .....	45
5.3.5	Využití časovače .....	46
5.4	LCD DISPLEJE.....	47
5.5	OBVOD REÁLNÉHO ČASU .....	48
5.6	PAMĚŤ EEPROM.....	50
5.7	BLUETOOTH MODULY .....	50
5.7.1	Bluetooth modul HC-05 .....	51
5.7.2	RS232 převodník Bolutek BLK–MD-BC04-B.....	54
5.7.3	Nastavení parametrů pro komunikaci .....	55
5.7.3.1	Příklad nastavení modulu Bolutek BLK–MD-BC04-B.....	55
5.8	REALIZACE ZAŘÍZENÍ A DODATEČNÉ ÚPRAVY .....	56
<b>6</b>	<b>SOFTWAREVÉ ŘEŠENÍ OVLÁDACÍHO A ZOBRAZOVACÍHO ZAŘÍZENÍ .....</b>	<b>58</b>
6.1	SOUBOR POŽADOVANÝCH PARAMETRŮ PRO NASTAVENÍ SIMULÁTORU MĚŘIČE RYCHLOSTI .....	58
6.2	HLAVNÍ PROGRAM OVLÁDACÍ A ZOBRAZOVACÍ JEDNOTKY .....	59
6.3	OVLÁDÁNÍ ZAŘÍZENÍ .....	59
6.3.1	Výpis informací na LCD.....	65
6.3.2	Maticová klávesnice.....	65
6.3.3	Úsporný režim .....	66
<b>7</b>	<b>SOFTWAREVÉ ŘEŠENÍ SIMULÁTORU RYCHLOMĚŘU.....</b>	<b>67</b>
7.1	POPIS VYBRANÝCH METOD.....	67
7.2	POPIS APLIKACE .....	69
	<b>ZÁVĚR .....</b>	<b>71</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>73</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>75</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>78</b>
	<b>SEZNAM TABULEK.....</b>	<b>80</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>81</b>



## ÚVOD

V dnešní době, kdy je na silnicích k vidění čím dál více motorových vozidel a téměř každá domácnost vlastní minimálně jedno auto se zvyšuje pravděpodobnost vzniku dopravních nehod zapříčiněných překročením dovolené rychlosti. Je tedy nutné nějakým způsobem zajistit dodržování této bezpečné dovolené rychlosti a přinutit řidiče jezdit pomaleji a obezřetněji ať už na dálnicích nebo obecně v úsecích, ve kterých může docházet k vysokým počtům nehod. K tomuto slouží radarový měřič rychlosti, který musí být pro každou určitou dopravní situaci přizpůsoben, například ve městech, na dálnicích nebo okresních cestách nastavením jeho měřících parametrů.

Hlavním cílem této práce je vyvinout ovládací a zobrazovací jednotku, která slouží k nastavení těchto parametrů policejního měřiče rychlosti motorových vozidel a interpretuje získané naměřené hodnoty na displeji jednotky.

Z důvodu požadavku na větší komfort při práci se zařízením, zobrazování měření s přesností na desetiny, možnosti přizpůsobit si jednotlivá ovládací menu, odstranění nedostatků předchozích modelů a realizování dalších úprav uživatelského rozhraní na základě předchozích zkušeností s používáním předešlého modelu má být vyvinut modernizovaný model ovládacího a zobrazovacího zařízení. Zařízení bude mít možnost využívat sériové komunikace jak pomocí bezdrátové technologie Bluetooth RS-232, tak i klasickou kabelovou sériovou linkou RS-232, dále bude ukládat jednotlivá nadlimitní měření s přiřazením datumu a času z obvodu reálného času do externí nevolatilní paměti odkud bude možnost data exportovat sériovou linkou do počítače pro další zpracování.

Při zpracování této práce nebude k dispozici reálný měřič rychlosti, proto dalším cílem diplomové práce je vytvořit jednoduchou grafickou simulační aplikaci měřiče rychlosti pro operační systém Windows využitím programovacího jazyka C#.

Práce je rozdělena na teoretickou a praktickou část. Teoretická část se věnuje přenosu informace, sériovému přenosu, technologii Bluetooth, jednočipovým mikropočítačům a jeho periferiím. Praktická část se zabývá popisem návrhu a realizace ovládací a zobrazovací jednotky a simulační aplikace měřiče. V neposlední řadě práce obsahuje také rozbor programového kódu obou cílů práce.

## **I. TEORETICKÁ ČÁST**

## 1 PŘENOS INFORMACE

Přenášené informace představují nějaký podnět, který má určitý význam pro stranu příjemce. Může jít o kódovaná data nesoucí údaj o stavu systému, vhodně uspořádaná data, kterými lze řídit dané zařízení. Tyto informace můžeme vysílat, přijímat mezi určitými zařízeními a následně dále zpracovávat. Přenos informace je realizován na základě modelu:

Informační zdroj - vysílač - komunikační cesta - přijímač - cílová destinace.

### 1.1 Typy přenášených informací

#### 1.1.1 Analogové informace

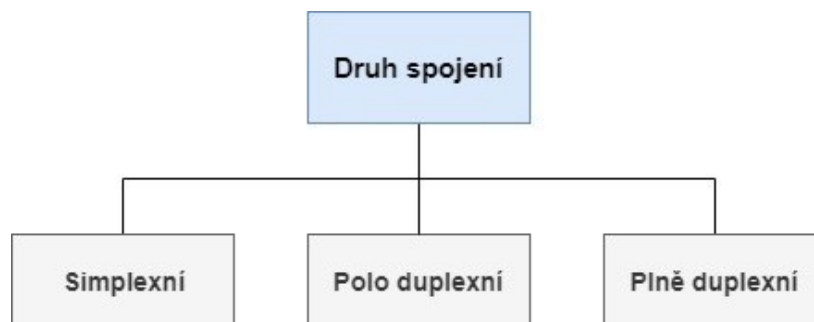
Analogové informace odrážejí proces, jehož okamžitá hodnota se mění spojitě s časem. Tento typ informace může tedy na omezeném intervalu nabývat libovolné hodnoty z nekonečného počtu hodnot. Jde například o zvuk. Abychom byli schopni přenos takovéto informace zajistit je nejprve zapotřebí využít elektronického zařízení, které převede tlak vzduchu na změny elektrického napětí. Takovým zařízením je mikrofón, snímač. [5]

#### 1.1.2 Diskrétní informace

Diskrétní informace popisují proces konečným počtem hodnot a na rozdíl od analogové informace/signálu se okamžitá hodnota nemění spojitě s časem. Základní a nejmenší jednotkou diskrétní informace je 1 bit, tento bit může nabývat buď logické 1 (Pravda), nebo logické 0 (Nepravda). Skupina 8 bitů se nazývá 1 byte, který již představuje konkrétní znak. Digitalizací analogové informace tedy získáme značné výhody, a to neomezenou reprodukovatelnost informace, možnosti dalšího zpracování, manipulovatelnosti, odolnosti proti rušení.

### 1.2 Druhy spojení

Pro přenos informace je nutné definovat způsob spojení a výměny dat mezi dvěma zařízeními. Druhy spojení jsou klasifikovány jako simplexní, polo duplexní, plně duplexní. Každý druh spojení má vždy zdroj a cílovou destinaci.



Obr. 1. Druhy přenosu informací.

### 1.3 Simplexní spojení

Režim simplexního spojení je model přenosu dat, ve kterém jsou data odesílána vždy jen v jednom směru. Odesílatel(A) pouze odesílá data a nemá schopnost je přijímat. Jako běžný příklad simplexního vysílání lze uvést televizní a rádiové vysílání.



Obr. 2. Simplexní model přenosu.

### 1.4 Poloduplexní spojení

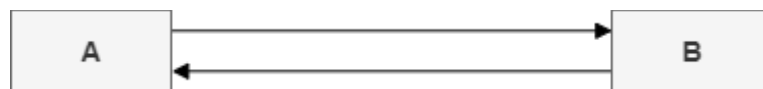
Poloduplexní přenosový režim už zahrnuje obousměrnou komunikaci, ale využívá se jen jednoho kanálu, vysílání a příjem probíhá na stejné frekvenci. U tohoto modelu tedy není možné vysílat a přijímat data současně, v daný okamžik mohou data putovat jen jedním směrem a je zde tudíž zpoždění v komunikaci. Jde o střídavé odesílání informací. V případě vysílání obou stran současně dochází ke ztrátě informace. Příkladem je obousměrný rádiový vysílač a přijímač – vojenská vysílačka.



Obr. 3. Poloduplexní(Half-Duplex) model přenosu.

## 1.5 Plně duplexní spojení

Poslední přenosový model je konstruován jako dvojice simplexních spojení, která umožňuje obousměrný simultánní přenos. Přenos probíhá pomocí jednoho kanálu. Model využívá toho, že signály, které jsou přenášeny v jednom směru, sdílejí kapacitu spojení se signály, které jdou směrem opačným. Kapacita přenosového kanálu je tedy rozdělena mezi oba směry. Plně duplexní režim již nabízí vysokou efektivitu komunikace. Příkladem může být telefon, smartphone.



Obr. 4. Plně duplexní (Full Duplex) model přenosu.

## 1.6 Sériová a paralelní komunikace

Přenos informace může probíhat dvěma způsoby. Jedná se o sériovou komunikaci a paralelní komunikaci.

### 1.6.1 Sériová komunikace

Sériová komunikace může probíhat v simplexním, poloduplexním nebo plně duplexním modelu přenosu informace. Je to typ komunikace, který slouží pro přenos větších objemů diskrétních dat. Pomocí tohoto typu komunikace je možné přenášet spolehlivě informace mezi dvěma v některých případech i více různými zařízeními na větší vzdálenosti. Tato přenášená data jsou reprezentována pomocí binárních signálů v podobě pulzů, tzn. logických hodnot 1 a 0, kde binární 1 je reprezentována jako log. 1(TRUE) nebo jako hodnota napětí 5 V a více a binární 0 jako log. 0(FALSE) nebo napěťová hodnota 0 V. [5]

Jde tedy o proces, při kterém dochází k odesílání dat v binární podobě sekvenčním způsobem přenosovým kanálem nebo po sběrnici na základě předem definovaného komunikačního protokolu. Protokol zajišťuje bezpečnou a spolehlivou formu komunikace pomocí definovaného souboru pravidel pro zdroj dat(odesílatel) a cílový bod(přijímač). Těchto protokolů existuje velké množství a každé zařízení schopné vzdálené komunikace nějaký využívá.

### 1.6.2 Paralelní komunikace

Využitím paralelního rozhraní je získána výhoda přenesení celého jednoho nebo více byte v jednom časovém úseku, tzn. vyšší přenosovou rychlostí, ovšem za cenu vyšších pořizovacích nákladů a vznikem možných problémů s nutností prostoru pro použití více vodičů a hrozba rozfázování signálů, která roste se zvyšováním vzdálenosti přenosu.

Maximální přenosová rychlost dat se rovná násobku frekvence a šířky rozhraní. Sériové rozhraní má šířku jednoho bitu, u paralelního rozhraní záleží na počtu signálových linek pro jednotlivé datové bity. S využitím paralelního rozhraní při propojení zařízení pomocí kabelů v délce desítek metrů není možné překročit frekvence desítek MHz.[5]

Tento typ přenosu je tedy vhodný pouze pro kratší vzdálenosti.

## 2 SÉRIOVÝ PŘENOS

Sériový přenos je způsob přenosu, který je založen pouze na dvou samostatných kabelech. U sériového přenosu můžeme rozlišovat, jestli jde o synchronní nebo asynchronní transakci.

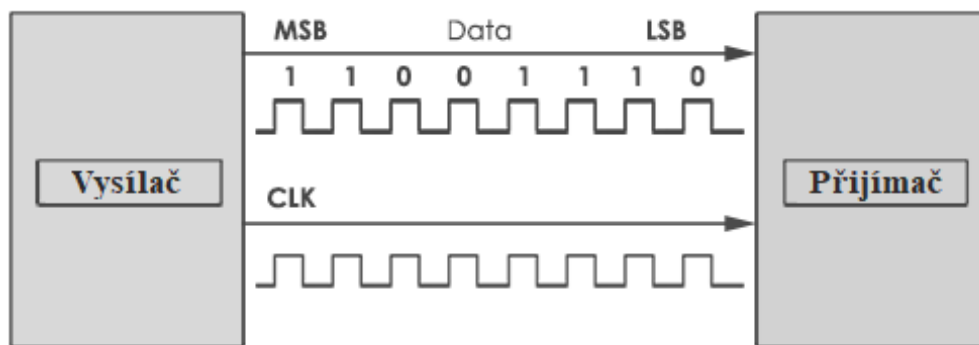
### 2.1 Synchronní přenos dat

Je to přenos s konstantní okamžitou rychlostí.[5]

U tohoto typu protokolu je vždy datová linka párována s hodinovým (CLK) signálem. To znamená, že všechna zařízení na této sériové sběrnici sdílejí společný taktovací signál. Tok bitů je přenášen přes jedinou datovou linku konstantní rychlostí. Komunikační kanál CLK (clock) zajišťuje synchronizaci mezi vysílačem a přijímačem.

Existuje možnost, kdy je časovací signál zahrnut do společného signálu s přenášenými daty při využití kódů s vlastní synchronizací. [5]

Tohoto způsobu komunikace využívají známé sériové sběrnice SPI a I2C a nabízí za jistých okolností rychlejší přenos dat.



Obr. 5. Schéma synchronní sériové komunikace.[6]

Z výše uvedeného diagramu lze vidět, že při každém taktu hodinového synchronizačního signálu je z vysílače odeslán jeden bit dat přijímači. Pokud dojde ke ztrátě synchronizace mezi vysílačem a přijímačem například následkem rušení nebo kapacitními vazbami mezi vodiči, může dojít ke zkreslení a částečné ztrátě informace až po úplnou ztrátu dat. Při sériové komunikaci je také určováno, jestli bude nejprve vyslán LSB nebo MSB datový bit. Nejčastěji jde o přenos datových bitů od LSB (nejméně významný bit) po MSB (nejvíce významný bit), tento způsob je vyznačen na schématu. Dále je zde zapotřebí brát v úvahu

Endianitu, která určuje, jestli se na nejnižší adresu v paměti uloží nejprve nejméně významný byte (Little Endian), nebo nejvíce významný byte (Big Endian).[6]

## 2.2 Asynchronní přenos dat

Jde o přenos, kde data jsou přenášena bez podpory externího synchronizačního hodinového signálu. Při asynchronním přenosu může jedna strana zahájit přenos v jakýkoli čas a v případě, že cílové zařízení není připraveno nadále obsluhovat tento přenos, má schopnost ho pozastavit. [5]

Touto metodou je minimalizován počet potřebných signálových vodičů a vstupně/výstupních pinů, ale je také zapotřebí vynaložit další úsilí pro zajištění spolehlivého bezchybného přenosu a příjmu dat. Proto u asynchronního přenosu musejí mít přijímač a vysílač nastaveny stejné komunikační parametry:

- Synchronizační bity - start bit, počet stop bitů
- Datové bity
- Paritní bit (sudá, lichá, bez parity)
- Přenosovou rychlost (Baud Rate)

U tohoto typu přenosu má každá strana vlastní zdroj hodinového signálu, díky kterému mohou odečítat data v přesně definovaných intervalech.

### 2.2.1 Baud Rate

Baud Rate specifikuje datovou rychlost, což je míra počtu změn stavu digitálního datového signálu za sekundu. [7] Udává tedy počet symbolů (signálových prvků) přenesených za sekundu.

Při metodě kódování NRZ (Non Return To Zero) je přenášený symbol jednobitový (změna přenášeného signálu představuje dvouhodnotovou informaci), takže přenosová rychlost (bity za sekundu nebo bps) a modulační rychlost (Baud Rate) je stejná. Toto platí pro sériové rozhraní UART. U složitějších metod, například použitím fázové modulace, kdy symbol představuje více bitů, je možno přenést čtyři bity na baud. [7]

Standardní rychlosti jsou 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bitů za sekundu [b/s] nebo baud [Bd].



### 2.2.2 Datové bity

Mají logickou úroveň danou vahou bitu (1 nebo 0). Data, která mají být přenášena, jsou rozdělena do bloků a vložena do paketů, které obsahují další konkrétní bitové sekvence.

Standardní velikost přenášené informace může být 5 až 9 bitů, toto však závisí, jestli je použita standardní znaková sada ASCII (7 bitů) nebo rozšířená znaková sada využívající 255 znaků (8 bitů).[7] V případě uplatnění paritního bitu je to 5 až 8 bitů.

### 2.2.3 Synchronizační bity

**Startovací bit** má pevně definovanou hodnotu logické nuly, umožňuje synchronizaci příjemce pomocí signálu z vysílače.[5] Signalizuje začátek přenosu.

**Stop bit** se používá k označení konce jednoho paketu a zajišťuje prodlevu mezi přenosy.[5] Jeho délka může být 1, 1.5, 2 bity. Čím více bitů je využito pro Stop bit, tím lépe dochází k synchronizaci obou zařízení, kde může docházet k výchytkám rychlosti, z důvodu, že každé zařízení má svůj vlastní zdroj hodinového signálu, ovšem za cenu zpomalení přenosu.

### 2.2.4 Paritní bit

Jde o volitelný detekční bit pro kontrolu chyby přenosu. Parita může být lichá nebo sudá. Při liché paritě je hodnota paritního bitu určena a odeslána tak, aby část přenášeného rámce (datové slovo + paritní bit) vždy obsahovala lichý počet výskytu bitů s logickou hodnotou 1. Při sudé paritě je počet bitů s logickou hodnotou 1 části rámce sudý.

### 2.2.5 Sériový paket



Obr. 6. Sériový paket/rámec.

Každý byte přenášených dat v asynchronním sériovém přenosu je odeslán v paketu nebo rámci bitů. Paket tedy tvoří synchronizační, paritní a datové bity. Nejčastěji je posílán nejdříve nejméně významný bit (LSB). Minimální jednotka informace, která je přenášena je jeden byte. [5]

### 2.3 RS-232

Tento standard byl vytvořen jako komunikační rozhraní zařízení DTE (počítač, tiskárna) s jedním koncovým zařízením datových kanálů DCE (modem) do vzdálenosti 20 metrů. [5]

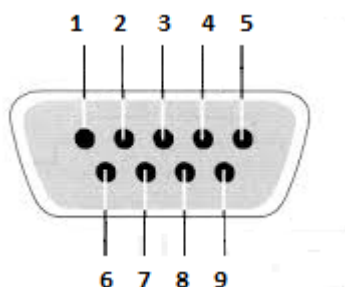
Jde o plně duplexní komunikaci, kde sériový přenos informací může probíhá asynchronně, pomocí pevně nastavené přenosové rychlosti a synchronizace sestupnou hranou startovacího impulsu. Standard určuje řídicí signály, přenos dat, elektrické rozhraní, typy konektorů. [5]

Pro větší odolnost proti rušení je informace po propojovacích vodičích přenášena větším napětím, než je standardních 5 V TTL logiky. Rozhraní je definováno nesymetrickými úrovněmi napětí podle tabulky 1.

Tab. 1. Napěťové úrovně RS-232.

Logická úroveň	TxD	RxD
0	+5 až +15 V	+3 až +25 V
1	-5 až -15 V	-3 až -25 V
nedefinováno		-3 až +3 V

Pro standard RS-232 se používá 9 pinový konektor nebo méně využívaná varianta 25 pinového konektoru. Ve většině případů se využívá pouze signálů TxD (vysílání), RxD (příjem) a GND (signálová zem) a ostatní zapojeny nemusí být. Pro propojení je tedy nutno signálového kabelu, který je 3 a více pramenný. Význam jednotlivých pinů je popsán v tabulce č. 2.



Obr. 7. Konektor sériového portu DB-9P.

Připojení dvou zařízení může být provedeno pomocí principu 1:1, kdy jsou piny male konektoru na jedné straně propojeny napřímo s female zásuvkou na druhém konci. Tohoto je

využito, pokud je připojován DCE modem se zařízením DTE, protože DCE modem na výstupu TXD přijímá a na vstupu RXD vysílá. Druhý způsob připojení se jmenuje Null modem, kde jeho typ zapojení může být minimální nebo úplné. Tohoto řešení se využívá, pokud je nutné propojit dvě zařízení stejného typu. U minimálního připojení jsou překříženy vodiče tak, že TxD je připojeno na RxD. V případě úplného připojení jsou navíc překříženy i piny DTR a DSR, RTS a CTS.[5]

Tab. 2. Funkce pinů sériového portu.[5]

Pin	Signál	Směr	Popis signálu
1	DCD	Input	<b>Data Carrier Detected</b> - Vstup oznamuje, že byl detekován nosný signál modemu
2	RXD	Input	<b>Receive Data</b> příjem dat, vstup přijímače
3	TXD	Output	<b>Transmit Data</b> vysílání dat, výstup vysílače
4	DTR	Output	<b>Data Terminal Ready</b> - posílá informaci, že je terminál připraven k výměně dat
5	PG, SG	-	<b>Protected ground, Signal Ground</b> ochranné uzemnění, signálová zem obvodu
6	DSR	Input	<b>Data Set Ready</b> - vstup indikuje, že je zařízení připraveno přenášet data
7	RTS	Output	<b>Request to Send</b> – aktivní stav informuje modem, že terminál má data k přenosu
8	CTS	Input	<b>Clear To Send</b> - povolení terminálu, aby odeslal data. Vypnutý stav přenos zakáže
9	RI	Input	<b>Ring Indicator</b> - Signál indikuje, že modem přijímá signál vyzvánění

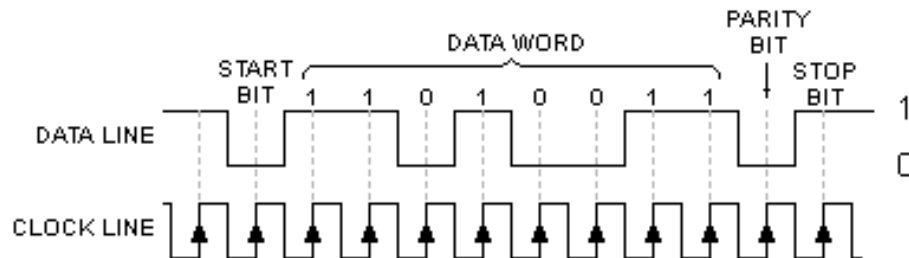
### Řízení toku dat

Jedná se o hardwarový nebo softwarový handshaking. Piny RTS a CTS slouží k řízení toku dat pomocí hardware. Softwarové řízení toku využívá komunikačních protokolů ZMODEM, KERMIT, kde pomocí datového kanálu přijímač informuje vysílače, jestli je schopen data přijmout a zpracovat. Používají se znaky ASCII XON/XOFF.[8]

#### 2.3.1 Asynchronní přenos RS-232

Před přenosem se datový kanál nachází v klidovém stavu, jde o vysokou úroveň napětí (log 1). Každý přenos slova (byte) v asynchronním sériovém přenosu začíná startovacím bitem

(nízká úroveň napětí – log. 0) z důvodu nutnosti synchronizace, protože data jsou přenášena konstantní rychlostí. Synchronizace je zajištěna pomocí sestupné hrany start bitu. Dále následují datové bity a případně i paritní bit. Přenos je ukončen stop bitem, který má vysokou úroveň napětí. Všechny tyto přenášené bity představují jeden přenosový rámeček.



Obr. 8. Přenosový rámeček asynchronní sériové komunikace RS-232.[8]

Z obrázku 8 lze vidět, že tento přenosový rámeček se skládá ze start bitu, 8 datových bitů, paritní bit: sudá parita, jeden stop bit. Signál představuje binární data 0b11001011 vyslaná od nejméně významného bitu po nejvíce významný bit. Takovéto datové slovo odpovídá hexadecimální hodnotě 0xCB, decimální hodnotě 203 a ASCII znaku *Ě*.

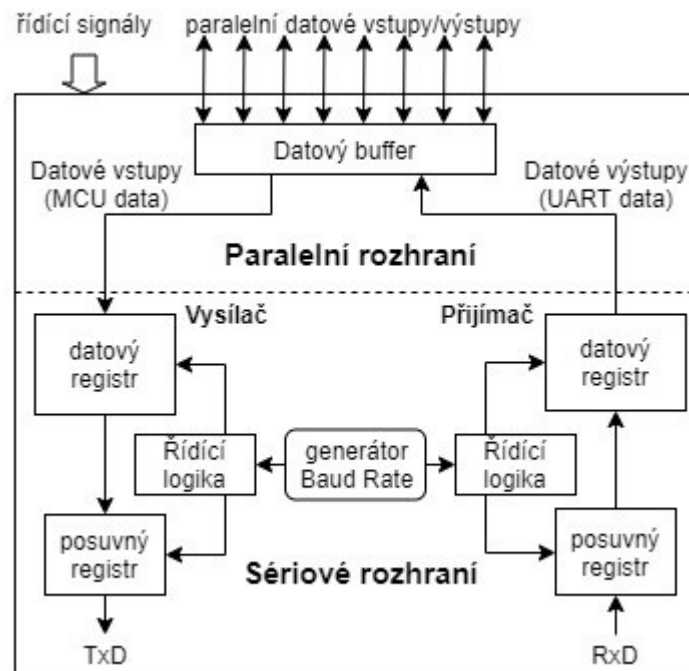
V dnešní době se už od využití RS-232 v oblasti počítačové techniky ustoupilo a bylo nahrazeno výkonnějším rozhraním USB. V průmyslu je však stále používáno zejména díky své jednoduchosti.

### 2.3.2 UART (Universal Asynchronous Receiver-Transmitter)

Sériový port mikropočítače obsahuje interní integrovaný obvod UART (Universal Asynchronous Receiver Transmitter) zajišťující převod paralelních dat (byte) na sériovou bitovou formu. [6] UART může být ve formě samostatného integrovaného obvodu, ale nejčastěji je integrovanou součástí mikropočítače. Mikropočítač může obsahovat i více těchto obvodů. Jde o hardwarovou implementaci sériové komunikace, UART funguje tedy jako prostředník, mezi paralelním a sériovým rozhraním, kde na jedné straně je připojena datová sběrnice sestávající se ze čtyř nebo osmi signálových linek, řídicích signálů a vyrovnávací paměti (datový buffer). Zde jsou paralelně přiváděny jednotlivé byte z mikropočítače, které jsou dále zpracovávány pro sériový přenos.

Na straně sériového rozhraní je sériový vysílač TxD skládající se z vysílacího datového registru, posuvného registru a řídicí logiky. Sériový přijímač RxD má taktéž svůj přijímací

datový registr, posuvný registr a řídicí logiku. Řídicí signály určují, kdy má docházet k zápisu nebo čtení dat. [12]



Obr. 9. Schéma UART rozhraní.

### Vysílač

Datový registr vysílače obsahuje paralelní data (byte) zasláná z mikrokontroléru, která jsou určena pro odesílání. Řídicí logika vysílače má za úkol sestavit přenosový rámec na základě nakonfigurovaných parametrů přenosu (start bit, paritní bit, počet stop bitů) a tento rámec je umístěn do posuvného registru vysílače. Obsah posuvného registru je postupně odesílán výstupem TxD s přesným časováním na základě zvolené přenosové rychlosti Baud Generátorem. [12]

### Přijímač

Přijímač vstupem RxD odebírá tyto bity přenosového rámce na základě očekávané přenosové rychlosti, odděluje synchronizační bity a datové bity ukládá do posuvného registru, dokud není přijat celý byte. Z posuvného registru jsou všechny bity poslány najednou do datového registru, poté do datového bufferu a odtud jsou odeslány přes paralelní sběrnici do mikrokontroléru. [12]

Převodu paralelních dat na sériová nebo naopak může být dosaženo využitím D-klopných obvodů, kde pro každý jednotlivý bit je zapotřebí jeden D-klopný obvod. D-klopný obvod je řízen hranou, posouvá tedy bit ze vstupu na výstup na základě detekce náběžné nebo sestupné hrany hodinového signálu. [12]

UART pracuje na napěťové úrovni TTL logiky podle rozmezí napájecího napětí mikropočítače, což může být standardně 3 V, 3,3 V, 5 V, proto je nutné pro přenos na větší vzdálenosti konvertovat toto napětí například na napěťovou úroveň linky RS 232, nebo je možno využít Bluetooth nebo Wifi modulů podporujících sériové rozhraní, které vyžadují stejné napájecí napětí, jako mikropočítač.

### 2.3.3 UART vs USART rozhraní

Rozdíl mezi rozhraními je ten, že USART podporuje synchronní i asynchronní komunikaci, zatímco UART podporuje pouze asynchronní způsob komunikace. Dalšími vylepšeními USART jsou:

- Podpora různých komunikačních protokolů jako LIN, RS-485, IrDA, Smart Card.
- Hodinový signál může být generován zařízením, které figuruje jako vysílač [12]
- Využitím externího hodinového signálu může být přenosová rychlost USART mnohem vyšší než u UART [12]

### 3 BLUETOOTH KOMUNIKACE

Bluetooth je bezdrátová technologie určena pro levné zařízení, komponenty malých rozměrů, umožňující rádiovou výměnu dat mezi počítači, mobilními telefony, smartphony a dalšími přenosnými zařízeními. Tato technologie je určena pro přenos dat na krátké vzdálenosti a je definována standardem IEEE 802.15.1.

Zařízení Bluetooth má bezdrátový vysílač a přijímač využívající rádiových vln UHF s krátkou vlnovou délkou pracující ve frekvenčním rozsahu od 2,4 do 2,480 GHz. Toto frekvenční pásmo je vymezeno pro průmyslové, zdravotnické a vědecké účely, tudíž všechna zařízení podporující toto rozhraní má velké možnosti využitelnosti. Bluetooth rozhraní bylo původně navrženo jako bezdrátová alternativa kabelového propojení pomocí rozhraní RS-232.[9] Bluetooth využívá rádiových kanálů s diskretní frekvenční modulací, kde nosná frekvence kanálu je:  $F = 2402 + k(\text{MHz})$ ,  $k = 0, \dots, 78$ . Kódování probíhá tak, že kladná frekvenční odchylka představuje logickou 1, záporná frekvenční odchylka logickou 0.[5] Vysílače jsou zařazeny do následujících tří výkonnostních tříd a nabízejí režim nízké spotřeby:

Tab. 3. Výkonnostní kategorie Bluetooth zařízení.

Třída	Maximální povolený výkon		Dosah
	mW	dBm	
1	100	20	až 100 m
2	2,5	4	až 10 m
3	1	0	až 1 m

Střídáním nosné frekvence je zajištěn přenos z jednoho kanálu na jiný, tímto způsobem je minimalizován útlum signálu a rušení.[5]

Bluetooth rozděluje přenášená data na pakety a každý paket je přenášen jedním ze 79 přidělených Bluetooth kanálů. Každý kanál má šířku pásma 1 MHz. Obvykle je prováděno 1600 přeladění za sekundu mezi 79 kanály při aktivované metodě přenosu v rozprostřeném spektru (Adaptive Frequency Hopping). Personální síť Bluetooth Low Energy využívá odstup 2 MHz, který pojme 40 kanálů. Komunikace je možná do vzdálenosti 1-100 m, ale pouze za podmínek provozu ve volném prostoru. Pokud jsou v cestě překážky, dosah rychle klesá, nedochází však ke skokové ztrátě spojení, ale postupně se zvyšuje počet chybně přenesených paketů.[9]

Bluetooth využívá těchto modulačních schémat:

- Gaussovo klíčování frekvenčním posuvem (GFSK).

Zařízení pracující na tomto principu modulace pracují v režimu Basic Rate (BR), kde je možná okamžitá přenosová rychlost 1 Mbit / s.

Příchodem verze Bluetooth 2.0 + EDR jde o:

- Diferenciálně kvadrurní klíčování fázovým posuvem ( $\pi/4$ -DQPSK).
- Diferenciální klíčování fázovým posuvem (8-DPSK).

Zde zařízení pracují v režimu Enhanced Data Rate (EDR) o přenosové rychlosti 2 a 3 Mbit/s. [9]

Bluetooth je paketový protokol postaven na architektuře Master/Slave. Zařízení v režimu Master může komunikovat až se sedmi zařízeními Slave v pikosíti (Piconet). Zařízení Slave zase může mít více než jedno nadřazené zařízení. Pomocí hodinového signálu zařízení, které je v Master režimu jsou synchronizována všechna ostatní zařízení (režim Slave). Na vývoji tohoto standardu se podílely významné počítačové, síťové a telekomunikační společnosti 3Com, Agere Systems, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia, Toshiba. [5]

### 3.1 Komunikace a připojení

Zařízení typu EDR nebo BR v režimu Master může komunikovat s maximálně sedmi zařízeními v ad-hoc počítačové síti využívající technologii Bluetooth (Piconet). Zařízení si mohou po dohodě vyměňovat jejich role. Zařízení v režimu Slave se tedy stane zařízením Master. Toto se odehrává například při komunikaci, kdy bezdrátová náhlavní souprava zahajuje spojení s počítačem nebo smartphonem. Pro navázání spojení musí být v roli Master, posléze jsou role prohozeny a funguje tedy jako Slave. Data mohou být přenášena v jakýkoli čas mezi zařízením v roli Master a jiným Slave zařízením, Master si vybírá, jaké zařízení osloví, a komunikace probíhá po určitý čas. Po uběhnutí tohoto času je vybráno jiné Slave zařízení, tímto způsobem je rychle přepínáno z jednoho zařízení na druhé.[9]

Jakékoli zařízení Bluetooth ve zjištěném režimu vysílá na požádání následující informace:

- Jméno zařízení
- Třidu
- Seznam služeb
- Technické informace - použitá Bluetooth specifikace, výrobce.



Každé zařízení má unikátní 48bitovou adresu, která však není viditelná, namísto toho je využíváno pojmenování zařízení nastavené uživatelem nebo defaultně výrobcem a toto jméno zařízení se zobrazuje při zjišťování dostupných zařízení nebo v seznamu již spárovaných zařízení. V případě, kdy se jedno zařízení snaží navázat komunikaci se zařízením, jehož adresa je mu známa, vždy reaguje přímými požadavky na připojení a předá výše uvedené informace, pokud jsou vyžadovány.[9]

Využití služeb zařízení může vyžadovat párování nebo potvrzení/přijetí ze strany jeho vlastníka.

### 3.2 PROTOKOL RFCOMM

Bluetooth standard obsahuje množství různých protokolů, jedním z nich je protokol RFCOMM. Jde o emulaci sériového portu RS232, variantu s 9 vodiči přes L2CAP (Logical Link Control and Adaptation Protocol). Protokol také podporuje vícenásobné spojení (až 60 současných připojení) a je tedy možno připojit jedno zařízení s více zařízeními. Radiové spojení nahrazuje standardní propojení pomocí vodičů, a tudíž není nutné multiplexorů a kabelů.[5]

RFCOMM obsahuje vestavěné schéma pro emulaci nulového modemu. Protokol vychází ze standardu ETSI TS 07.10 a je využíván v mnoha Bluetooth aplikacích zejména pro svou rozšířenost a také veřejně dostupnému API. Pokud se jedná o přenos stavů řídicích obvodů a ne obvodů datových, standard TS 07.10 nerozlišuje mezi DTE a DCE zařízeními. Řídicí signály RS-232 jsou vysílány jako počet nezávislých DTE / DCE signálů viz tabulka 4:

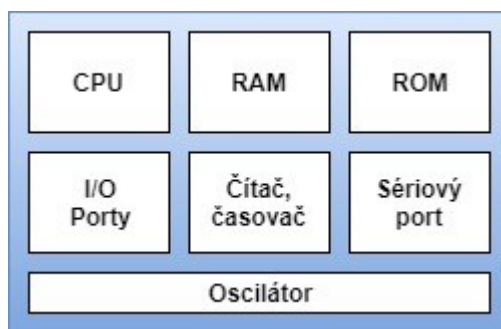
*Tab. 4. TS 07.10 řídicí signály sériového portu.*

Signály TS 07.10	Odpovídající RS-232 řídicí signály
RTC	DSR, DTR
RTR	RTS, CTS
IC	RI
DV	DCD

Způsobem, jakým standard TS 07.10 přenáší řídicí signály RS-232 vytváří propojení nulového modemu, pokud jsou propojována dvě zařízení stejného druhu.[10]

## 4 JEDNOČIPOVÉ MIKROPOČÍTAČE

Jednočipový mikropočítač je integrovaný obvod, ve kterém jsou soustředěny všechny podstatné části mikropočítače. Nejčastěji je využíváno Harvardské koncepce, která má oddělený adresní prostor pro paměť programu a paměť dat.



Obr. 10. Zjednodušené schéma jednočipového mikropočítače.

### Základní obvody jednočipového mikropočítače:

- Centrální procesorová jednotka.
- Paměť programu (EPROM nebo Flash, ROM, pokud jde o jednočipový mikropočítač určený pro konkrétní aplikaci s pevně daným programem).
- Paměť dat (RAM).
- Periferní obvody pro vstup a výstup dat. (I/O porty)
- Generátor hodinového signálu.
- Sériový port

Dále mohou mít A/D a D/A převodníky, řadič přerušování, kontrolní obvod pro správnou činnost.

Vzájemné propojení jednotlivých obvodů je zajištěno soustavou sběrnic, ty umožňují rozšiřování počítače o další jednotky, aniž by bylo změněno vnitřní zapojení těchto jednotek. V každý okamžik může na sběrnici být připojen pouze jeden zdroj dat. [5]

Jednočipové počítače lze vhodně naprogramovat pro vybranou oblast použití, programování může být realizováno například pomocí USB programátoru odpovídajícího výrobce. Pro programování jednočipových mikropočítačů jsou využívány jazyky vyšší úrovně: C, C++,

ale stále v některých případech je nutno použít jazyka symbolických adres pro zajištění úsporného, optimálního kódu pro vestavěnou (embedded) aplikaci.

Jednočipové mikropočítače, které jsou užívány v řízení procesů v reálném čase se nazývají mikrokontroléry.[3]

Mikrokontroléry jsou nejčastěji využívány ve vestavěných systémech řízení a regulace. Jejich využitím dochází ke zjednodušení procesu vývoje a výroby zařízení a tím i ke snížení celkových nákladů.

## 4.1 MC9S08AC128

Tento mikrokontrolér od firmy NXP pochází z rodiny HCS08, tudíž jde o 8bitový mikrokontrolér nabízející maximální frekvenci sběrnice 20 MHz s centrální procesorovou jednotkou o maximální taktovací frekvenci 40 MHz. Mikrokontrolér je dostupný v různých verzích pouzdra (package), a to: 80pin LQFP, 64pin QFP, 48pin QFN, 44pin LQFP, kde jednotlivé provedení se liší v obsahu periferií, počtu vstupně výstupních pinů, časovačů a kanálů časovačů, AD převodníků, sériových periferních rozhraní a vzdáleností mezi jednotlivými vývody.

### 4.1.1 Vlastnosti

Mikrokontrolér je určen pro všeobecné průmyslové aplikace a do prostředí se zvýšenou hladinou rušení, tudíž se vyznačuje velmi dobrými vlastnostmi EMC, EMI a nabízí:

- až 40 MHz taktovací frekvenci CPU
- až 20 MHz taktovací frekvenci sběrnice
- 128 kB paměti FLASH.
- až 8 kB paměti RAM.
- Až 70 GPIO pinů (v závislosti na použitém pouzdru)
- KBI (Keyboard interrupt module), SCI (Serial Communication Interface), SPI, IIC (I2C), RTI (Real Time Interrupt), Background Debug Mode (BDM)
- 1 dvoukanálový, 2 šestikanálové 16bitové časovače s pulsně šířkovou modulací (PWM)
- Možnosti volby zdroje hodinového signálu

#### 4.1.2 Pracovní režimy:

- **Run mode:** normální pracovní režim mikrokontroléru, CPU vykonává kód uložený v paměti pro program.
- **Wait mode:** vykonáním instrukce WAIT CPU vstoupí do režimu čekání. Nachází se ve stavu nízké spotřeby a není taktován, v tomto režimu jsou povolena přerušení.
- **Stop mode:** MC9S08AC12 se může nacházet v jednom ze dvou různých Stop režimů (stop2, stop3), a to nastavením bitu STOPE v system option registru a odpovídajících bitů v registru SPMSC2. Ve stop2 režimu jsou všechny zdroje hodinového signálu zastaveny, periférie, Flash paměť a centrální procesní jednotka jsou také vypnuty. Jde o stav velmi nízké spotřeby energie v pohotovostním režimu, přičemž obsah paměti RAM a stav I/O pinů je stále udržován. [11]

#### 4.1.3 Přerušovací subsystém

Přerušení umožňuje mikrokontroléru reagovat na nahodilé, asynchronní události. Taková událost vyvolá přerušení vykonávání dosud běžícího programu v hlavním programu (cyklu) - dokončí se právě rozpracovaná strojová instrukce a činnost CPU je přerušena. Následně je vyvolána obsluha přerušení daného zdroje a začne se vykonávat podprogram, který se nenachází v hlavní smyčce. Po dokončení obsluhy přerušení činnost procesoru pokračuje provedením následující instrukce z místa v hlavním programu, kde bylo přerušení vyvoláno.

Požadavků na přerušení může být více a jsou zpracovávány pomocí řadiče přerušení.

Řadič přerušení má za úlohu:

- Zaregistrovat požadavky na přerušení, které jsou aktivní.
- V případě více aktivních požadavků najednou dojde k jejich zařazení podle nastavené priority.
- Předat informaci procesoru o tom, který požadavek byl vybrán.
- Oznámit procesoru existenci alespoň jednoho aktivního požadavku.[3]

#### Přerušení od periférií

Velká část požadavků na přerušení jsou vnitřní, například od periférií, diagnostických obvodů. Může jít o přerušení od časovače, KBI, UART, ADC modulu.

#### IRQ

Mikrokontrolér má ale také IRQ pin, který je určen jako vstup pro požadavek na přerušení od vnější události. Tato externí přerušení jsou řízena pomocí IRQSC stavového a kontrolního registru. Vnitřní obvody IRQ reagují buď na detekci zvolené polaritě hrany přerušovacího signálu (vzestupná nebo sestupná) nebo na detekci hrany a logické úrovně na pinu. Pokud je funkce tohoto pinu povolena, je možné také pomocí něj probudit mikrokontrolér ze stop režimu.[11]

#### 4.1.4 Periferie mikrokontroléru

V této kapitole jsou uvedeny vybrané periferie mikrokontroléru vztahující se k této práci. Každá periferie má své řídicí a stavové registry, díky kterým je možné nastavovat jejich funkci.

##### 4.1.4.1 Čítač/časovač

Slouží k počítání událostí, generování událostí s přesným načasováním, modulace šířky pulsu, k vytvoření časového intervalu (funkce zpoždění). Dále je nutný pro správnou synchronizaci s dalšími obvody a událostmi.

MC9S08AC128 má tři nezávislé moduly časovačů (TPM), každý kanál umožňuje pulzně šířkovou modulaci (PWM). Celý časovací systém MC9S08AC128 nabízí 6kanálový TPM1, druhý 6kanálový TPM2 a 2kanálový TPM3. Časovače jsou na sobě nezávislé. Každý kanál může pracovat v režimu input capture, output compare, edge-aligned PWM, center-aligned PWM.

Časovač se sestává z čítače, který čítá s předem nastavenou frekvencí až do určité hodnoty, poté dojde k přetečení jeho obsahu a nastavení příznaku přetečení, následuje vynulování a čítač začne čítat od začátku. V době, kdy je nastaven příznak přetečení (hodnota bitu příznaku přetečení je 1) může být využito jako zdroj přerušení za podmínky, že je povoleno přerušení od časovače. K tomuto nastavení slouží stavový a řídicí registr.

Hodnota přetečení čítače je určena pomocí 16bitového modulu registru podle vzorce:

$$\text{Modulo} = \frac{t_{TOF} * f_{source}}{\text{prescaler}} \quad (1)$$

Kde:

- $t_{TOF}$ : požadovaný čas do přetečení v sekundách

- $f_{\text{source}}$ : frekvence hodinového kmitočtu časovače
- prescaler: hodnota předděličky
- Modulo: hodnota modulo registru

Čítač má 16 bitů, hodnota čítače je rozdělena do dvou 8bitových registrů, kde první obsahuje vyšší byte, druhý registr nižší byte. Při čtení z těchto registrů jsou jednotlivé byte ukládány do vyrovnávací paměti a zůstávají zde uloženy, dokud nejsou přečteny oba byte, tím je zajištěno koherentní čtení hodnoty čítače.

#### Režimy čítače:

- **Vzestupný (inkrementační):** čítání směrem nahoru od hodnoty 0x0000 do hodnoty nastavené modulo registrem (Output compare - režim reagující na shodu, Edge aligned PWM režimy) nebo do přetečení čítače - hodnota 0xFFFF (normální režim).
- **inkrementační a dekrementační:** čítání nahoru od hodnoty 0x0000 po hodnotu nastavenou v modulo registru a poté čítá zpět dolů do hodnoty 0, příznak přetečení čítače je nastaven při dosažení nuly. (Center aligned PWM režim). Když během čítání ve směru dolů dojde ke shodě s hodnotou v registru hodnot kanálu, výstup PWM se nastaví do log. 1, v případě shody s hodnotou v registru hodnot kanálu při vzestupném čítání se výstup PWM nastaví do log.0. [11]

#### Zdroj hodinového signálu:

- frekvence sběrnice vydělena hodnotou předděličky (prescaler): 1, 2, 4, 8, 16, 32, 64, 128.
- fixní systémový: zdroj hodin krystalu, v případě aktivní smyčky PLL, FLL je nutno synchronizačního obvodu, jinak je shodný jako hodinový signál sběrnice
- externí

Externí zdroj hodinového signálu lze zvolit pro jednotlivé obvody časovačů, jsou k tomu určeny piny TPMCLK, TPM1CLK, and TPM2CLK, kde maximální frekvence může být: frekvence sběrnice / 4.[11]

Každý kanál má svou vlastní obsluhu přerušení.

#### 4.1.4.2 A/D převodník

MC9S08AC128 má 10bitový ADC modul s přístupnými 18 analogovými vstupy. Napětíové reference A/D převodníku jsou brány z hlavního obvodu napájení ovládacího zařízení. Referenční napětí pro vysokou úroveň je tedy shodné s napájením mikrokontroléru. ADC modul obsahuje:

- Lineární postupný aproximační algoritmus s rozlišením 10 bitů.
- Teplotní čidlo, jehož výstup je připojen k jednomu z analogových vstupů kanálu.
- Převodník nabízí nastavení jednorázového nebo kontinuálního převodu, nastavení optimalizace konverze.[11]

#### 4.1.4.3 UART(SCI)

MC9S08AC128 disponuje dvěma nezávislými SCI rozhraními podporujícími plně duplexní, asynchronní sériovou komunikaci s kódováním NRZ. SCI pracuje v těchto režimech:

- normální režim: Je využíváno výstupu TxD vysílače a vstupu RxD přijímače, jde o režim asynchronního přenosu.
- režim smyčky (Loop mode): Výstup vysílače je interně připojen ke vstupu přijímače. Režim slouží k odlaďování uživatelského programu sériové komunikace.
- režim jednoho vodiče (Single-Wire): Vstup přijímače je připojen ke vstupu vysílače TxD. Tento režim se používá k implementaci poloduplexního sériového přenosu. [11]

Dále má 3 oddělené kontrolní registry přerušení, možnost volby polarity vysílaných dat a možnost generování nebo přijímání handshake signálů. Přerušení je generováno:

- při vyslání posledního znaku z bufferu vysílače nebo umístění jednoho znaku do bufferu vysílače
- při úspěšně přijatém znaku
- při zaplnění bufferu přijímače
- při příjmu signalizačních bitů overrun, parity error, framing error, noise error
- při detekci LIN break znaku [11]

SCI obsahuje generátor přenosové rychlosti (baud rate generátor), blok vysílače a přijímače. Přijímač a vysílač pracují nezávisle na sobě i přesto že sdílejí stejný generátor přenosové rychlosti. Zdrojem hodinového signálu pro baud rate generátor je hodinový signál sběrnice mikrokontroléru. Přenosová rychlost SCI je udávána vztahem:

$$\text{Baud rate} = \frac{f_{bus}}{\text{prescale divisor} * 16} \quad (2)$$

Kde:

- Baud rate – výsledná přenosová rychlost
- $f_{bus}$  – frekvence sběrnice
- prescale divisor – hodnota modulo dělitele pro nastavení různých přenosových rychlostí

Při výběru normálního režimu je možno zvolit, jestli bude přenášeno 8 nebo 9 datových bitů. SCI má možnost nastavení parity (sudá, lichá, bez parity). Paritní bit může být generován a kontrolován hardwarově pomocí interního logického obvodu, nebo softwarově. V případě nastavení parity je nejvýznamnější bit (MSB) vysílaného byte považován za paritní bit (osmý nebo devátý datový bit). [11]

#### 4.1.4.4 Externí přerušení KBI

Modul zajišťuje možnost generování přerušení až pro 8 pinů, každý tento pin má v řídicím registru vlastní bit pro nastavení. Přerušení může být naprogramováno tak, aby bylo generováno v závislosti na:

- sestupné hraně signálu a citlivosti na nízkou úroveň (LOW) signálu
- sestupné hraně a citlivosti na nízkou úroveň (LOW) signálu nebo vzestupné hraně a citlivosti na vysokou úroveň (HIGH) signálu.
- pouze sestupné nebo náběžné hraně.
- na úrovních signálu.

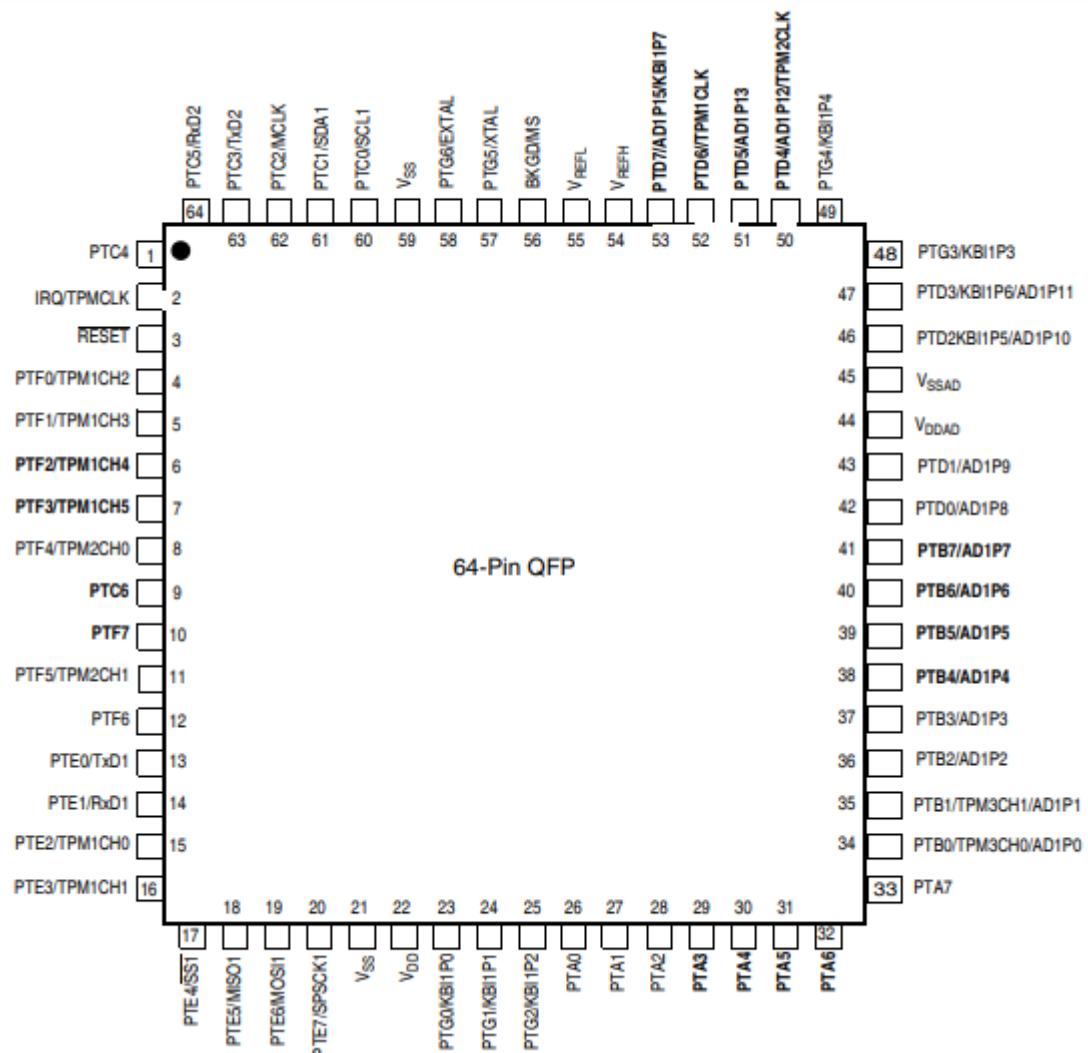
Z toho pouze 4 piny a to 53 (KBI1P7), 47 (KBI1P6), 46 (KBI1P5), 49 (KBI1P4) lze nastavit pro detekci vzestupné hrany a citlivosti na vysokou úroveň (HIGH) signálu. Modul má společný bit pro povolení přerušení. Pomocí takto nakonfigurovaných pinů lze mikrokontrolér probudit z režimu „wait“ nebo „stop3“ režimu externími signály zajištěnými například tlačítky, prepínači, maticovou klávesnicí.[11]

#### 4.1.5 Rozmístění pinů

Následující schéma znázorňuje přiřazení signálů vnitřních obvodů k jeho jednotlivým pinům (vývodům). Téměř u všech vývodů mikrokontroléru může být zvolena jedna z jeho více



funkcí, nastavení je zajištěno pomocí odpovídajících registrů dané periferie. Například vývod 49 (PTG4/KBI1P4) je přiřazen portu G, defaultně je nastaven jako vstupně/výstupní (GPIO) pin, ale jeho funkce může být přenastavena pro generování přerušování z externího tlačítka pomocí registru KBI Pin Enable Register (KBIPE). Pokud je pin nastaven jako analogový pomocí registrů periferie, jeho digitální funkce je zakázána.



Obr. 11. Rozmístění pinů u MC9S08AC128, 64 pinové pouzdro, QFP[11].

Vstupně-výstupní digitální piny (I/O) jsou určeny pro komunikaci pomocí logických úrovní. U pinů s funkcí I/O musí být určen směr toku dat, nastavujeme tedy, jestli jde o pin vstupní nebo výstupní zápisem logické hodnoty do registru řízení směru daného pinu portu (PTxDDn). Zápisem do datového registru (PTxDn) nastavujeme na výstupní pin logickou úroveň 0 nebo 1, v případě vstupního pinu tyto logické hodnoty čteme. Písmeno x určuje označení portu, n určuje číslo bitu v registru, který ovládá daný pin (PTD0 – port D, nultý

bit, vývod 42 z mikrokontroléru). *MC9S08AC128* má k tomuto účelu k dispozici 54 pinů. U pinů, kde jejich směr je nastaven jako vstupní mohou být aktivovány pull-up rezistory odpovídajícími řídicími registry. Pull-up rezistory zajišťují, že pin má pevně definovanou počáteční logickou úroveň 1 v případě, kdy na tento vstup není nic připojeno.

Dále je možné nastavit Low nebo High Drive Strength, v druhém případě výstupní pin může dodávat zátěži větší proud, avšak nesmí být překročen maximální dovolený proudový limit pinu, což je 25 mA. [11]

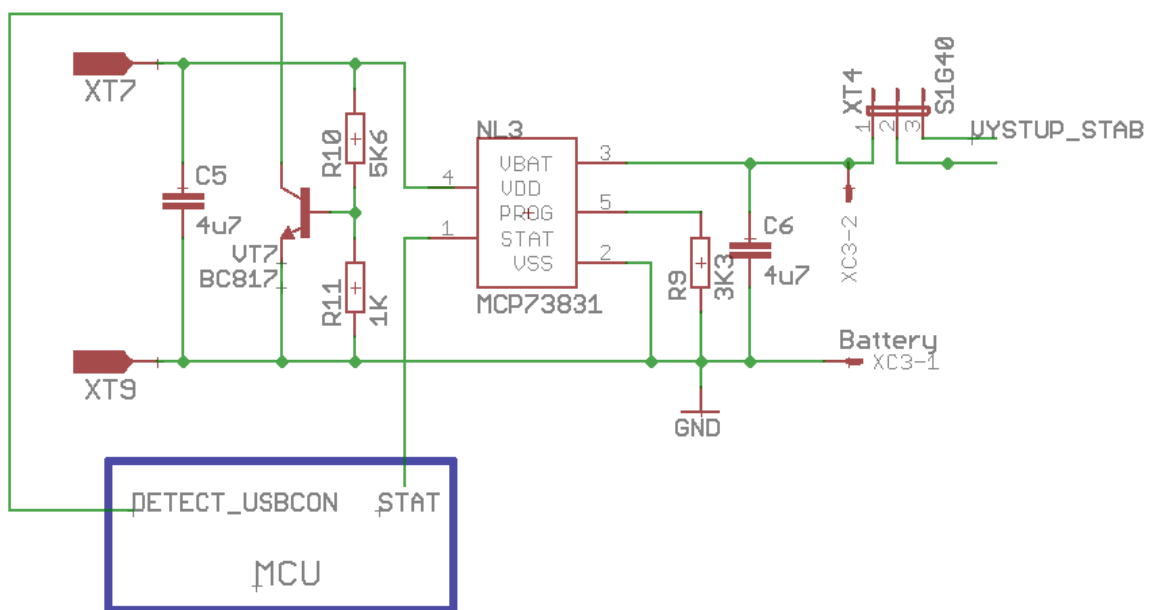
## **II. PRAKTICKÁ ČÁST**



Kromě hardwarové a softwarové realizace ovládací/zobrazovací jednotky bylo zapotřebí navrhnout a vytvořit počítačovou aplikaci s použitím libovolného programovacího jazyka, která bude simulovat měřič rychlosti. Obousměrná komunikace mezi realizovanou ovládací, zobrazovací jednotkou a simulační aplikací měřiče rychlosti probíhá přes RS-232 rozhraní.

## 5.1 Obvod nabíjení

Použitý SMD obvod MCP73831-2 je určen pro nabíjení Lithium-iontových/polymerových akumulátorů a je díky svým velmi malým rozměrům (SOT-23 pouzdro) uzpůsoben pro použití v prostorově omezených a přenositelných aplikacích, což je aktuální případ. Velkou výhodou je nutnost malého počtu externích součástek pro jeho funkci a nízká cena. Obvod MCP73831 byl také zvolen vzhledem k tomu, že zařízení bude nabíjeno pomocí USB. Zdroj nabíjení je připojen pomocí konektoru USB B mini (XT7).



Obr. 13. Schéma obvodu pro nabíjení akumulátoru a detekce USB nabíjení.

Napájecí napětí  $V_{DD}$  obvodu je 5 V. K pinu  $V_{BAT}$  je připojena lithium-polymerový akumulátor o napětí 3,7 V s kapacitou 1600 mAh. Akumulátor byl vybrán na základě regulačního napětí  $V_{REG}$  obvodu MCP73831-2 a její kapacita podle požadované výdrže provozu zařízení minimálně 3 hodiny.

### Nastavení regulace nabíjecího proudu PROG

Nabíjecí obvod disponuje sadou režimů pro regulaci proudu: Kondicionování, rychlé nabíjení, konec nabíjení. Hodnoty proudu ve všech režimech jsou určeny na základě zvolené hodnoty rezistoru  $R_{PROG}$ . [13]

V režimu kondicionování je nabíjecí proud určen podle vzorce:  $I_{PREG} / I_{REG}$ . Přejít mezi režimy je uskutečněno při přesáhnutí prahového napětí definovaného pro každý režim. V případě dosažení regulačního napětí  $V_{REG} = 4,2$  V (určeno typem obvodu) začíná regulace pro udržení konstantní hodnoty napětí. Při režimu rychlého nabíjení může být maximální nabíjecí proud 500 mA. Kvůli zahřívání integrovaného obvodu byla hodnota nabíjecího proudu zmenšena přidáním rezistoru s hodnotou 3K3 mezi pin  $R_{PROG}$  a napájecí zem  $V_{SS}$ . [13]

Hodnota byla vypočtena na základě výpočetního vztahu:

$$I_{REG} = \frac{1000}{R_{PROG}} \quad (3)$$

Kde:

$R_{PROG}$  [K $\Omega$ ] – rezistor pro nastavení hodnoty nabíjecího proudu

$I_{REG}$  [mA] – nabíjecí proud

$$R_{PROG} = \frac{1000}{300} = 3.33 \text{ K}\Omega$$

Před akumulátor je paralelně připojen výstupní keramický kondenzátor C6 pro zajištění stability výstupního proudu.

### Výstup stavu nabíjení STAT

Pin STAT obvodu MCP73831 je třístavový výstup s úrovněmi:

- Low – Tento stav indikuje, že právě probíhá nabíjení. Na výstupu je logická 0 představující napětíovou hodnotu 0 V.
- High – V případě plného nabití dojde k nastavení na výstup log. 1, představovaná napětíovou úrovní 5 V. Tato hodnota je čtena vstupem STAT mikrokontroléru.
- High-Z – stav vysoké impedance, indikuje, že baterie není připojena.

Tento výstup je připojen k vstupnímu pinu STAT mikrokontroléru a je zde čten aktuální stav na základě výše uvedených stavů výstupu. Vstup má aktivován pull up rezistor.

Detekce připojeného USB nabíjení je řešena pomocí bipolárního tranzistoru s vyvedenou bází na výstup děliče napětí. Na vstupním pinu DETECT\_USBCON mikrokontroléru je nastaven pull up rezistor zajišťující log. 1. Tranzistor je zapojen v režimu s otevřeným kolektorem. Při připojení USB napájení přejde tranzistor VT7 do režimu saturace přes dělič napětí (zajišťující napětí  $U_{BE} > 0,7 \text{ V}$ ), tím dojde k sepnutí tranzistoru a nastavení pinu DETECT\_USBCON na log. 0.

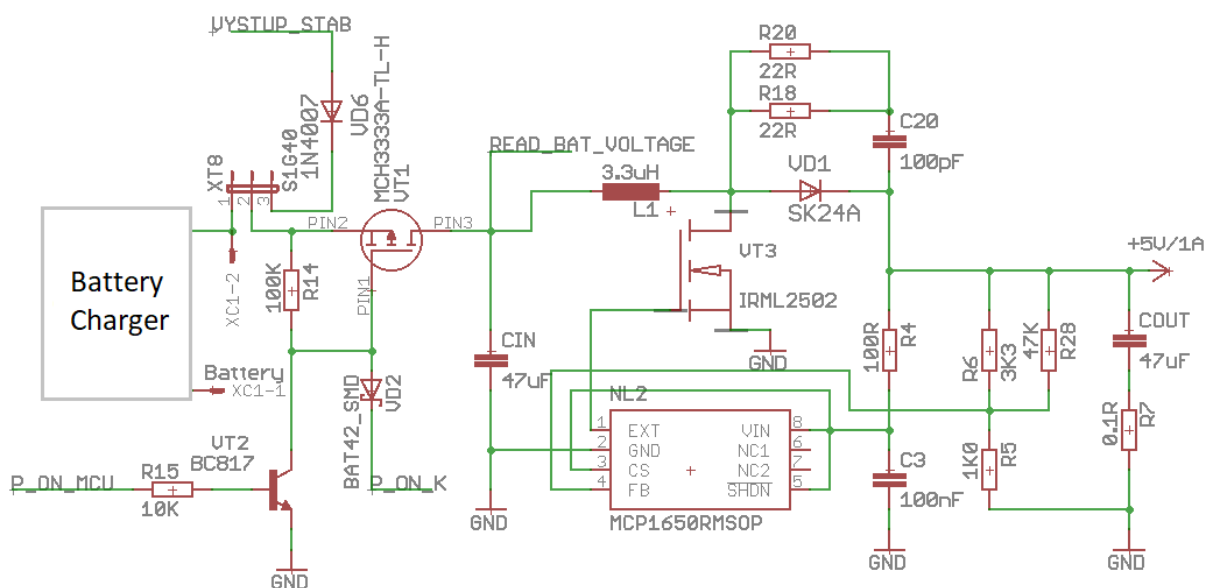
## 5.2 Napájení ovládací a zobrazovací jednotky

Všechny integrované obvody ovládací jednotky včetně displejů jsou napájeny 5 V ze stejnosměrného měniče napětí pracujícího ve zvyšujícím režimu. Pomocí jumperu XT8 je možno volit ze dvou zdrojů napětí pro zvyšující DC měnič:

- Akumulátor
- Externího napájení

Napájení z externího zdroje je stabilizováno obvodem 78M05 na pevnou hodnotu 5 V – 0,7 V (úbytek napětí na diodě), tedy kolem 4,3 V. Pomocí Schottkyho diody je vstup stabilizátoru chráněn proti přepólování tohoto zdroje.

Zapínání a vypínání jednotky je řešeno spínáním P-MOSFETu VT1. Za tranzistorem je snímáno aktuální napětí baterie pomocí A/D převodníku (pin READ\_BAT\_VOLTAGE).



Obr. 14. Schéma obvodu napájení ovládací jednotky.

### 5.2.1 Zvyšující DC-DC měnič

Výstupní napájecí napětí 5 V je zajištěno obvodem MCP1650R. Jeho vstupní napětí  $V_{IN}$  se může pohybovat od 2,7 do 5,5 V, proto je vhodný pro využití v přenositelných aplikacích napájených z lithiových akumulátorů v daném napěťovém rozsahu.

Pro dosažení požadovaných parametrů měniče byly použity následující externí součástky:

- N-MOSFET IRML2502
- Cívka s indukčností 3,3  $\mu$ H, Schottkyho dioda SK24A
- Odporový dělič napětí pro nastavení výstupního napětí, vstupní a výstupní keramické kondenzátory.

Hodnota indukčnosti cívky byla zvolena na základě doporučení v datovém listu k MCP1650R pro tuto aplikaci, dále nebylo nutné přidání odporu pro omezení špičkové hodnoty proudu cívkou z důvodu malého poměru převodu těchto napětí. Pinem EXT je řízeno rychlé spínání/rozpínání výkonového N-MOSFET tranzistoru. K zamezení vybíjení kondenzátoru na zem přes sepnutý tranzistor IRML2502 slouží dioda SK24A. Dostatečnou frekvencí spínání je dosaženo vyššího výstupního napětí, než je vstupní.

Regulace výstupního napětí je prováděna porovnáváním hodnoty výstupního napětí měniče přes vhodně nastavený napěťový dělič [(hodnota odporů R6, R28 připojených paralelně) a R5] proti vnitřnímu referenčnímu napětí 1,22 V na pinu FB (Feedback). Protože je jednotka vypínána odpojením napájecího zdroje pro měnič, pin SHDN (Shutdown) je tedy stále připojen k pinu  $V_{IN}$ .

Pro aktuální firmwarové řešení ovládací jednotky byl zvolen způsob bezdrátového provozu, zdrojem je tedy akumulátor.

## 5.3 Mikrokontrolér

Jádrum ovládací a zobrazovací jednotky je mikrokontrolér MC9S08AC128. Byl vybrán především na základě požadavků:

- Napájecí napětí 5 V
- Dostatečný počet digitálních I/O pinů v kombinaci s piny využívající interních periférií (ADC, UART, a další)



- 2x SCI (UART), jeden pro bezdrátový režim, druhý pro využití kabelového propojení
- SPI pro komunikaci s externí EEPROM, I2C pro komunikaci s obvodem reálného času DS3231.
- Dostatečné rychlosti a velikosti FLASH paměti pro firmware jednotky.
- Možnosti zapůjčení programátoru BDM USB Multilink HCS08/HCS12.
- SMD provedení vhodné pro ruční pájení – pouzdro 64pin QFP.

Ovládací a zobrazovací jednotka využívá téměř všechny hlavní periferie, které MC9S08AC128 nabízí.

### 5.3.1 Paralelní I/O

V ovládacím zařízení měřiče rychlosti jsou I/O piny použity následovně:

- datové a řídicí signály LCD displejů.
- spínání tranzistoru sirény (Buzzer).
- řídicí signály Bluetooth modulu (Indikace spojení, device enable).
- řídicí signály externí EEPROM (Chip Select, Write Protect).
- detekce nabíjení přes USB konektor.
- řízení tranzistoru pro vypnutí/zapnutí zařízení.

### 5.3.2 SCI

Sériové UART rozhraní je využito pro přenos nastavovacích parametrů do simulační aplikace měřícího přístroje.

Pro zajištění větší přesnosti nastavení přenosové rychlosti byl jako zdroj hodinového signálu pro mikrokontrolér použit externí krystal. Při zvoleném externím zdroji hodin a aktivní smyčce FLL (FEE režim) byly hodnoty parametrů N, P, R pro výpočet interního hodinového signálu mikrokontroléru o pevně dané frekvenci zvoleny podle ukázkového příkladu výpočtu z datového listu mikrokontroléru. Hodnota externího krystalu pro výpočet byla vybrána z běžně dostupných krystalů na českém trhu. Frekvence vybraného krystalu je 3,6864 MHz. Pro zajištění požadované přenosové rychlosti SCI je nutné znát frekvenci sběrnice. Ta byla vypočítána podle výpočetního vztahu:

$$f_{BUS} = \frac{f_{ext\_crystal} * P(N/R)}{2} \quad (4)$$

Kde:

- $f_{BUS}$  – frekvence sběrnice
- $f_{EXT\_CRYSTAL}$  – frekvence externího krystalu
- $P$  – hodnota 1 nebo 64
- $N$  – násobící faktor
- $R$  – dělicí faktor

$$f_{BUS} = \frac{3,6864 * 1(10/1)}{2} = 18,432 \text{ MHz}$$

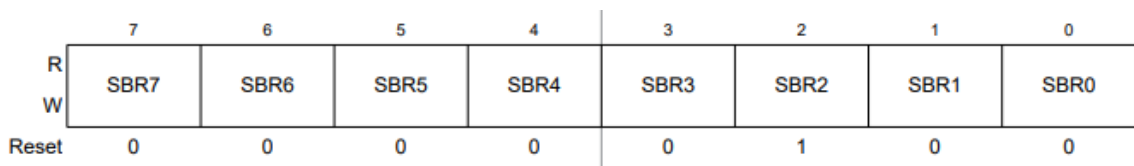
Externí krystal byl tedy zvolen tak, aby nedošlo k nesouladu přenosové rychlosti mezi komunikujícími zařízeními. Povolená hodnota nesouladu udávaná výrobcem mikrokontroléru při použití interního zdroje hodinového signálu je 4,5 % pro 8bitový datový režim. Pomocí rovnice (2) byla určena hodnota modulo děličky [SBR12:SBR0] pro generování přenosové rychlosti 9600 Bd:

$$[SBR12: SBR0] = \frac{f_{BUS}}{\text{Baud rate} * 16}$$

$$[SBR12: SBR0] = \frac{18,432 * 10^6}{9600 * 16}$$

$$[SBR12: SBR0] = 120$$

Pro zapsání hodnoty modulo dělitele 120 (binárně 01111000) je nutno nastavit hodnoty v Baud Rate registrech SCI1BDH a SCI1BDL tak, aby kombinace 13 bitů SBR12 až SBR0 odpovídala této hodnotě. Hodnota dělitele je reprezentována 8 bity, tudíž pro nastavení požadované přenosové rychlosti stačí zapsat těchto 8 bitů do registru SCI1BDL (SCI1BDL = 0x78) a zbytek bitů v SCI1BDH nastavit na 0 (SCI1BDH = 0x00). Při frekvenci externího krystalu 3,6864 MHz s využitím smyčky FLL vychází přenosová rychlost přesně 9600 Bd.



Obr. 15. Baud Rate registr SCI1BDL.[11]

V této práci SCI pracuje v asynchronním režimu (využití přijímače RxD i vysílače TxD) s následující konfigurací:

- režim přenosu 8 datových bitů
- zvolená parita je lichá
- paritní bit je generován hardwarově.

Dále je povoleno přerušení při úspěšném příjmu znaku. Tato nastavení jsou provedena zápisem bitů do odpovídajících stavových, kontrolních a řídicích registrů.

### 5.3.3 A/D převodník

Ovládací zařízení využívá analogově digitální převodník pro snímání aktuální hodnoty napětí akumulátoru.

#### Napájení

Jako napájení převodníku  $V_{DDAD}$  je použito napájení  $V_{DD}$  mikrokontroléru. Analogová zem převodníku  $V_{SSAD}$  je připojeno na společnou zem v celém zařízení.

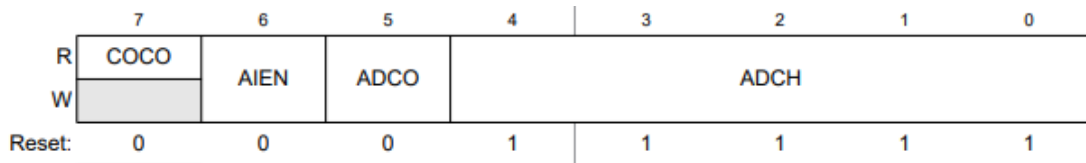
#### Napět'ové reference

Mikrokontrolér umožňuje připojení externího zdroje napět'ové reference. V případě tohoto ovládacího zařízení dostačuje, aby pin pro referenční hodnotu HIGH (VREFH) byl připojen na vstupní napájení A/D převodníku ( $V_{DDAD}$ ). Pin pro referenční hodnotu LOW (VREFL) je připojen na společnou zem.

#### Řídicí registry

Pro využití A/D převodníku je nutné nejprve nastavit odpovídající registry, důležitý je stavový a řídicí registr ADCSC1. V registru je vyhrazeno 5 bitů pro výběr vstupního kanálu, jde o pole ADCH, daný kanál je vybrán zápisem kombinace bitů dle tabulek uvedených v manuálu (datasheet) mikrokontroléru. Zápisem kombinace 11111 dojde k přímé deaktivaci

A/D převodníku a izolaci vstupního kanálu od všech zdrojů, tím je zabráněno dalším převodům.



Obr. 16. Stavový a řídicí registr ADCSC1.[11]

Pole AIEN je určeno k zapnutí nebo vypnutí generování přerušení po dokončení převodu, bit COCO (Conversion Complete Flag) je bit pouze pro čtení, který je nastaven při dokončení převodu. Bit ADCO je nastaven na 0 pro jednotlivý převod. Stavový a kontrolní Registr ADCSC2 nastavuje softwarový trigger pro zahájení převodu, zjištění, jestli je konverze aktivní, funkce pro porovnávání hodnot. Zápisem do konfiguračního registru ADCCFG je nastaven režim převodu: 10 bit, interní ADC hodinový signál je určen podle vztahu: zdroj hodinového signálu/dělička hodinového signálu ( $BUS\_CLK/4$ ), nízkopříkonový režim a dlouhá doba vzorkování.

K ovládání jednotlivých pinů kanálů 0 až 7 slouží Pin Control 1 registr (APCTL1). Používají se k zablokování řízení vstupně výstupních portů pinů mikrokontroléru používaných jako analogové vstupy. Signál pro snímání hodnoty napětí baterie je připojen k vývodu AD1P0, tudíž byl nastaven odpovídající bit ADPC0 na hodnotu 0.

Po dokončení převodu lze k výsledku převodu přistupovat pomocí datových registrů ADCRH a ADCRL. Pro převod bylo využito 10bitového A/D, v tomto případě je výsledek zaokrouhlen na 10 bitů.

Hodnota napětí akumulátoru  $U_{BAT}$  je vypočítána podle výpočetního vztahu:

$$U_{BAT} = \frac{ADC_{VREFH}}{2^{(ADC_{RES}-1)}} * ADC_{VAL} \quad (5)$$

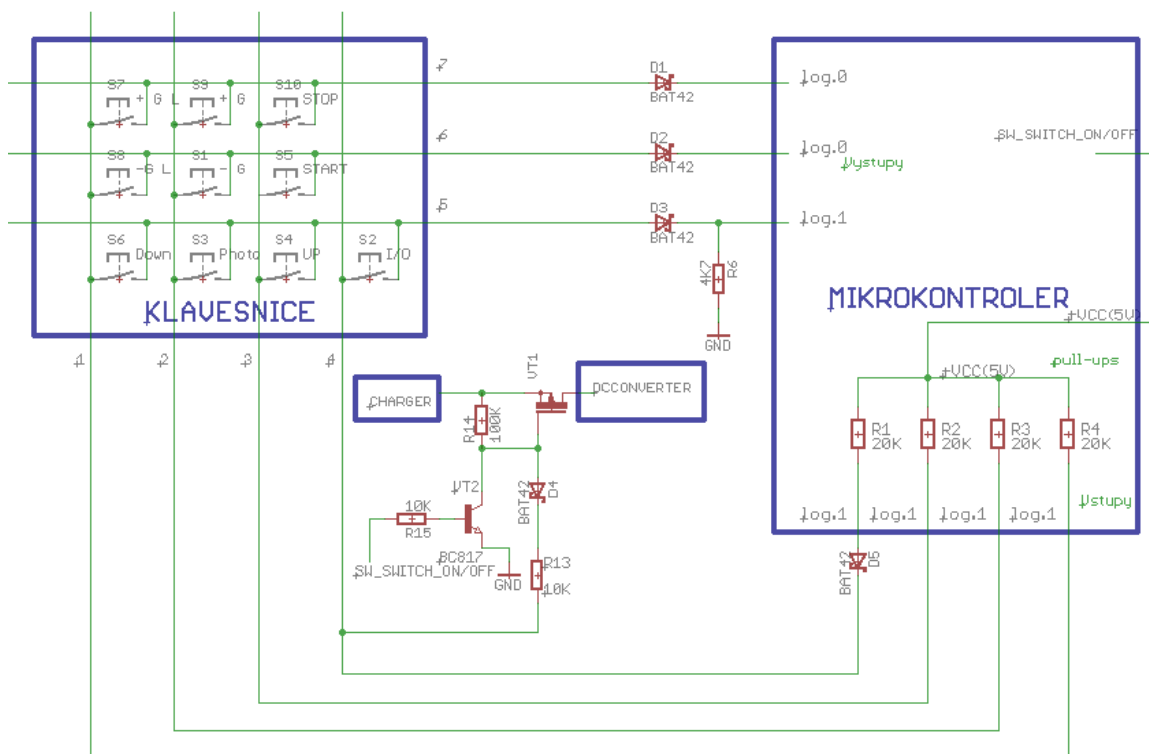
Kde:

- $ADC_{VREFH}$  – referenční napětí pro ADC modul
- $ADC_{RES}$  – rozlišení ADC
- $ADC_{VAL}$  – výsledná hodnota převodu

Pro zajištění větší přesnosti ADC hodnoty je provedeno 32 převodů a jejich následné zprůměrování.

### 5.3.4 KBI přerušení

Vstupní piny mikrokontroléru 53 (KBI1P7), 47 (KBI1P6), 46 (KBI1P5), 49 (KBI1P4) jsou nastaveny pomocí KBI Pin Enable registru (KBIPE) jako vstupy generující přerušení při stisku klávesy. K takto nakonfigurovaným pinům MCU s aktivovanými pull up rezistory jsou připojeny sloupce maticové klávesnice. Řádky jsou připojeny k výstupním pinům přes ochranné diody proti zkratu při případnému stisku více kláves najednou.



Obr. 17. Schéma zapojení maticové klávesnice.

KBI přerušení je generováno při detekci sestupné hrany – změně stavu z aktivní úrovně (log.1) na log.0 na vstupních pinech. V obsluze přerušení je následně zjišťována aktuálně stisknutá klávesa na klávesnici.

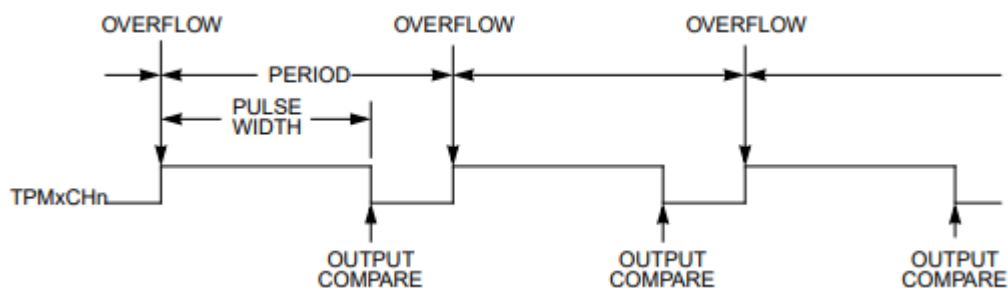
Schéma výše znázorňuje i zapínání ovládací a zobrazovací jednotky, to je realizováno následovně:

Po připojení akumulátoru je P-MOSFET VT1 rozepnut (mezi vývody source a drain neprotéká proud), protože rezistor R14 zajišťuje, že napětí na gate je stejné jako na source. Při

stisku I/O tlačítka dojde k uzavření obvodu přes rezistor R14, Schottkyho diody D4 a D3 a rezistor R6. Dojde tedy k sepnutí VT1 a na vstup DC/DC měniče MCP1650R je přivedeno napětí z akumulátoru. Měnič vytvoří na výstupu 5 V a následně se ziniculuje mikrokontrolér. Ten ihned po startu nastaví na jeho výstupním pinu SW\_SWITCH\_ON/OFF log. 1, čímž se otevře bipolární tranzistor VT2 do saturace a gate P-MOSFETu VT2 je připojen na zem a je tím udržován v sepnutém stavu. Vypnutí ovládací a zobrazovací jednotky je zajištěno zápisem log. 0 na výstupní pin SW\_SWITCH\_ON/OFF pomocí maticové klávesnice.

### 5.3.5 Využití časovače

V zařízení je využito kanálu časovače v režimu edge-aligned PWM pro změnu intenzity podsvícení LCD displejů v různých provozních stavech. Tento režim se vyznačuje dvounásobnou pracovní frekvencí proti center-aligned PWM. Režim může být aktivní i když ostatní kanály v tomto samém TPM modulu jsou nastaveny pro režimy input capture nebo output compare. Nastavuje se zápisem hodnoty bitu CPWMS = 0 ve stavovém a kontrolním registru TPM0SC. Časovač TPM1 kanál 0 je určen pro řízení podsvícení LCD 16x2, TPM11 pro LCD 12x2.



Obr. 18. Perioda PWM a šířka pulsu.[11]

Perioda signálu je určena dobou do přetečení (overflow) čítače, tedy hodnotou + 1 v modulo registrech (TPM1MODH:TPM1MODL). Poměr aktivní úrovně pulzu (pulse width) a periody signálu se nazývá střída (duty cycle). Tento poměr je nastavován pomocí hodnotových registrů vybraného TPM1 kanálu - TPM1C0VH:TPM1C0VL (Output compare). Polarita PWM signálu je dána nastavením kontrolního bitu ELS0A ve stavovém a kontrolním registru TPM1C0SC na hodnotu 0, to znamená, že po dosažení hodnoty v modulo registru dojde k nastavení výstupu z aktivní úrovně na neaktivní (HIGH to LOW).

U každého z displejů je nastavena jiná hodnota bitu ELS0A z důvodu, že první je spínán pomocí NPN a druhý PNP tranzistorem. Napětí podsvícení je tedy řízeno spínáním bipolárního tranzistoru, kde na jeho bázi je přiváděn PWM signál o určité střídě.

V režimu nečinnosti ovládacího zařízení je nastaven poměr na 0 % u obou řízení PWM, tudíž podsvícení displejů je vypnuto, v případě vypnutého měřicího režimu zařízení je pro LCD1 nastaven na 10 % (představuje hodnotu 0x1996 při využití zápisu pomocí 16bitového bezznaménkového celého čísla).

TPM3 časovač je využit pouze jako zdroj přerušení, toto přerušení je generováno každé 4,551 sekundy. V obsluze přerušení je umístěn kód pro hlášení nízkého stavu nabití baterie, indikace USB nabíjení, vypnutí podsvícení displejů a Bluetooth modulu. K tomuto účelu je využit další časovač TMP2 kanál 0, který počítá čas, po který je ovládací zařízení neaktivní, po předem nastavené době (cca 15 minut) dojde k vypnutí podsvícení displejů a Bluetooth modulu.

## 5.4 LCD Displeje

Pro zobrazení nastavovaných hodnot a přijímaných informací byly vybrány tyto Alfanumerické LCD displeje:

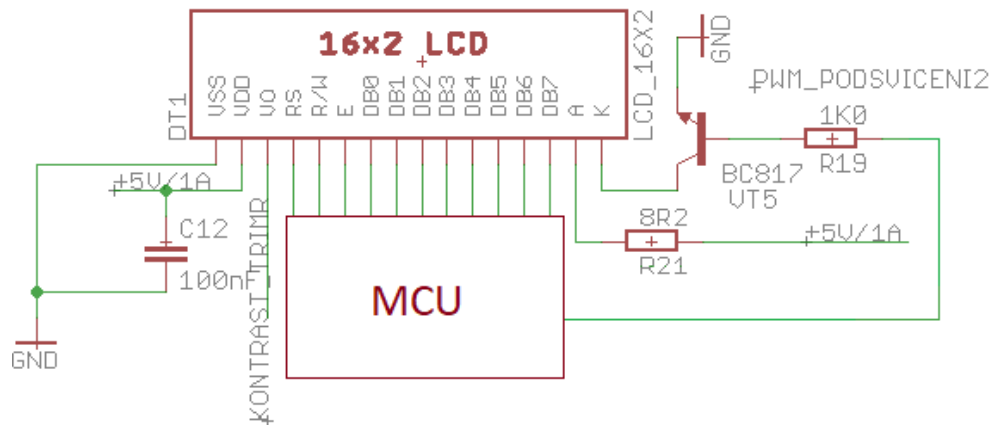
### RC1202A-YHY-ESX

- má 2 řádky, 12 sloupců, 15 pinů
- typ zobrazení: pozitivní, transflektivní
- barva podsvícení: žlutá
- rozměry modulu (délka, šířka, výška): 55,7 x 32,0 x 9,7 mm
- rozteč kontaktů: 1,27 mm
- slouží k zobrazení přijatého záznamu měření ze simulátoru měřiče

### POWERTIP NPC1602LRU

- má 2 řádky, 16 sloupců, 16 pinů
- typ zobrazení: pozitivní, transflektivní
- barva podsvícení: žlutá
- rozměry modulu (délka, šířka, výška): 80 x 36 x 12,7 mm
- rozteč kontaktů: 2,54 mm
- slouží k zobrazení nastavených hodnot parametrů

Každý z displejů používá určitou znakovou sadu uvedenou v datovém listu.



Obr. 19. Schéma zapojení LCD displeje POWER TIP NPC1602LRU.

Oba displeje pracují v 8bitovém datovém režimu (piny DB0 až DB7) a mají řadič ST7066U. Řídicími piny jsou:

- RS (Register Select): Určuje, zda jsou zapisována data nebo příkaz.
- R/W (Read/Write): Určuje, zda se jedná o čtení nebo zápis.
- E (Enable): Při sestupné hraně řídicího signálu (změna z logické úrovně 1 na log. 0) dojde k přijetí aktuálně zapsaných logických hodnot na datových pinech řadičem displeje.

Displeje mají napájecí napětí 5 V, napětí je přivedeno na pin  $V_{DD}$  a napájecí zem  $V_{SS}$ . Pro podsvícení obou displejů byly vypočteny hodnoty předřadného odporu, jas je řízen pulzně-šířkovou modulací generovanou časovačem mikropočítače. Příslušný výstup je proudově zesílen bipolárním tranzistorem v zapojení se společným emitorem. Na pin VO u obou displejů je připojen trimr s hodnotou 10 k $\Omega$ , kterým je nastavován kontrast zobrazení.

## 5.5 Obvod reálného času

Pro zprostředkování hodin reálného času je využit RTC obvod Maxim DS3231, který se vyznačuje vysokou přesností a nízkou spotřebou. Čtení hodnot z obvodu probíhá po datové sběrnici IIC (I2C). DS3231 je zařízení Slave a je adresováno mikrokontrolérem (Master).

Přenos probíhá po dvou vodičích:

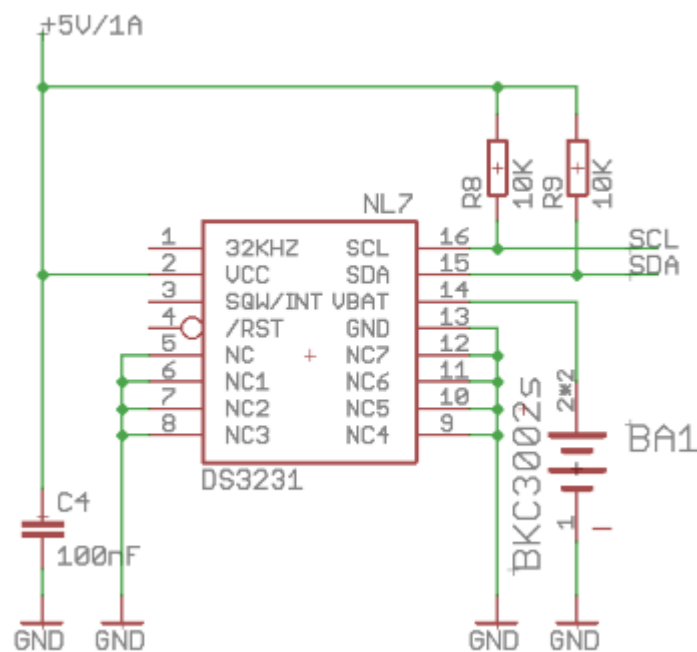
- SCL (Serial Clock): vstupní pin pro hodinový signál, který je generován mikrokontrolérem. Frekvence hodinového signálu je nastavena na 100 kHz.



- SDA (Serial Data): pin pro obousměrný sériový datový přenos.

U obou vodičů je zajištěno jejich nastavení na log. 1 pomocí pull up rezistorů, aby DS3231 neobsazoval sběrnici v době, kdy by přenos zrovna neměl probíhat.

Pro komunikaci mikrokontroléru s tímto obvodem je nejdříve vyslán signál start (sestupná hrana detekovaná na pinu SDA) za přítomnosti hodinového signálu v log. 1. Signál start tedy udává informaci všem obvodům na sběrnici, že se jedná o nový přenos dat). Dále je vysílána adresa DS3231 spolu s bitem (Read/Write) určujícím, jestli jde o operaci čtení nebo zápisu. Po každém správně přijatém byte je vyslán potvrzující bit Ack zařízením, které tyto data přijímalo. Data jsou přenášena po jednom byte. Pro ukončení přenosu je vyslán signál stop, který je reprezentován jako změna z log. 0 na log. 1 při aktivní úrovni hodinového signálu. Poté se zařízení nachází v klidovém stavu zajištěném pull-up rezistory R8 a R9.



Obr. 20. Schéma zapojení RTC obvodu DS3231.

Obvod je využíván k přiřazení aktuálního datumu a času k datům (jednotlivým měření rychlosti) zaslaných ze simulační aplikace. Udržování aktuálního datumu a času ve vypnutém stavu ovládací a zobrazovací jednotky zajišťuje knoflíková baterie CR2032. Baterie je uložena do kovového pouzdra BKC3002S na DPS.

## 5.6 Paměť EEPROM

V návrhu zařízení je použita externí paměť AT25512 od výrobce ATMEL o velikosti 512 Kbit (524288 bitů). Do paměti je tedy možné uchovat 65536 byte. Zápis probíhá vždy po blocích (stránkách) o velikosti 128 byte. Prvním bitem, který je vysílán nebo přijímán, je MSB. Napájecí napětí je 5 V a maximální frekvence hodinového signálu pro komunikaci s pamětí je 20 MHz. Paměť má 8bitový stavový registr, do kterého jsou zapisovány všechny řídicí instrukce (povolení/zákaz zápisu do registru, přístup k registru, nastavení ochrany proti zápisu pro blok dat).

Komunikace probíhá přes SPI, EEPROM je podřízeným zařízením (slave) a má připojeny následující piny:

- SCK (Serial Clock): Vstupní pin pro hodinový signál generovaný mikrokontrolérem. Polarita hodinového signálu musí být nastavena na log. 0 v době, kdy komunikace není aktivní.
- SI (Slave Input): datový vstup
- SO (Slave Output): datový výstup
- CS (Chip Select/Slave Select) – slouží pro povolení paměti EEPROM komunikovat na SPI sběrnici. Pokud tento pin není nastaven na log. 0, zasláná data mikrokontrolérem nejsou pinem SI přijata. Samotný zápis nebo čtení dat proběhne až na základě detekce vzestupné hrany signálu.
- WP (Write protect) – Při nízké napěťové úrovni (log.0) zajišťuje hardwarovou ochranu proti neúmyslnému zapisování do stavového registru paměti.
- HOLD: je připojen k napájecímu napětí (log. 1), protože návrh nepočítá s pozastavením sériové komunikace.

Do této nevolatilní paměti jsou ukládány aktuálně zvolené hodnoty parametrů pro nastavení simulační aplikace měřiče rychlosti, hodnoty jsou po startu ovládací jednotky znovu načteny. Dále jsou zde ukládány nadlimitní měření (jednotlivé měření rychlosti vyhodnocené jako přestupky) zasláné simulační aplikací včetně přiřazení data a času z obvodu reálného času DS3231.

## 5.7 Bluetooth moduly

Bluetooth moduly slouží jako náhrada kabelového rozhraní RS232 a umožňují bezdrátovou komunikaci se simulační aplikací.

Obě zařízení mají integrovanou anténu, podporují Enhanced Data-Rate modulační schéma s maximální přenosovou rychlostí 2,1 Mbit/s a využívají pásma 2,4 GHz. Zařízení jsou třídy 2 - maximální povolený výkon je 2,5 mW a jejich maximální dosah 10 metrů. Běžná spotřeba modulů je 30 mA, 10 mA v případě spárování se zařízením.

U obou modulů je možnost volby, jestli budou pracovat v režimu Master nebo Slave a podporují kromě komunikace přes rozhraní UART i rozhraní SPI, ale tyto piny nejsou zapojeny v tomto návrhu. Moduly jsou napájeny 5 V, ale vstupní a výstupní signály pracují s napětíovou úrovní 3,3 V. Pro dva signály TxD a BTMODUL\_ENABLE ve směru komunikace od mikrokontroléru k Bluetooth modulu musely být přizpůsobeny napětíové úrovně pomocí děliče napětí podle vztahu:

$$U_{výstupní} = \frac{R_2}{R_1 + R_2} * U_{vstupní} \quad (6)$$

$$U_{výstupní} = \frac{2000}{1000 + 2000} * 5$$

$$U_{výstupní} = 3,33 \text{ V.}$$

V opačném směru komunikace mikrokontrolér spolehlivě rozpozná logickou 1, protože její napětíová úroveň je definována od 2,7 V pro napájecí napětí  $V_{DD}$ .

### 5.7.1 Bluetooth modul HC-05

Modul je umístěn v ovládací jednotce na straně displejů a je nastaven na roli Master, je tedy iniciátorem spojení.



Obr. 21. HC-05 bluetooth modul.

**Funkce pinů:**

- **Enable:** Tento vstup slouží k zapnutí/vypnutí modulu. Modul je zapnut při aktivní logické úrovni (HIGH) nebo v nezapojeném stavu (vnitřní Pull-up rezistor). Pin je řízen vývodem 27 (PTA1) na mikrokontroléru.
- **VCC:** napájecí napětí 3,3 až 6 V. Na desce je integrován vlastní stabilizátor napětí z maximálně 6 na 3,3 V, které je nutné pro vnitřní modul s hlavními integrovanými obvody.
- **GND:** signálová a napájecí zem.
- **TXD:** Výstup modulu pro zasílání dat vstupu UARTu MCU. Tento signál je zapojen k RXD pinu na mikrokontroléru.
- **RXD:** Vstup modulu pro příjem/čtení dat z výstupu UARTu MCU. Tento signál je zapojen k TXD pinu na mikrokontroléru. Správná napěťová úroveň je zajištěna pomocí děliče napětí.
- **State:** Výstupní pin, který slouží jako indikace úspěšného připojení k druhému Bluetooth zařízení, tento vývod je připojen ke vstupnímu pinu (PTA4) mikrokontroléru. Vysoká úroveň signálu signalizuje úspěšné spárování/připojení se zařízením. Tento pin je také propojen s indikační interní LED na modulu, pomocí které jsou signalizovány 3 stavy, ve kterých se modul může nacházet:
  - Komunikační režim (Communication mode): Probíhá vyhledávání zařízení ke spárování. Indikováno přibližně 3 bliknutími za sekundu.
  - Komunikační režim: Zařízení bylo úspěšně spárováno a připojeno. Indikováno dvojitým probliknutím každých 5 sekund.
  - Příkazový režim (Command mode): V tomto režimu je zařízení nastavováno AT příkazy. Indikováno jedním bliknutím každé 2 sekundy.

### **Komunikační režim:**

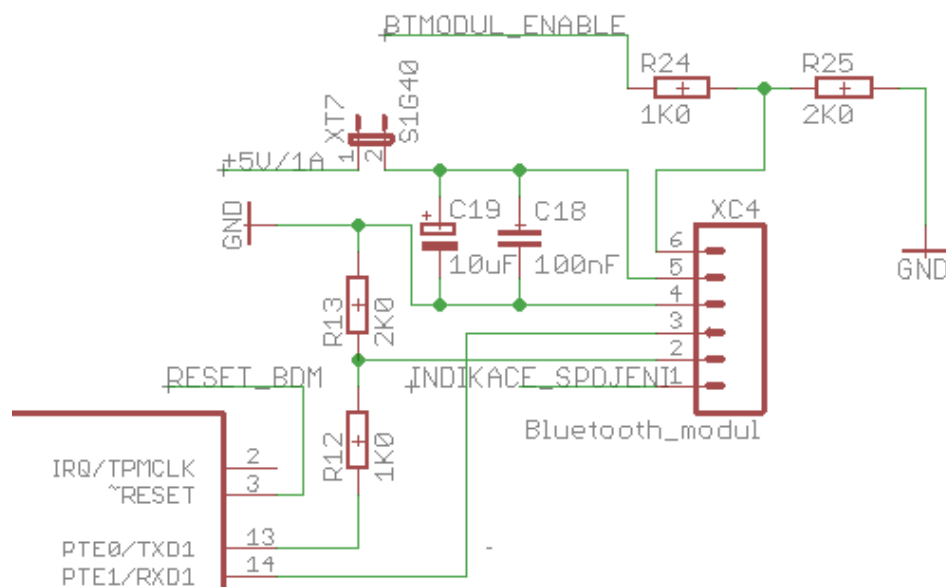
V tomto režimu jsou všechny přijaté informace přes sériové UART spojení považovány pouze jako datové. Pokud dojde ke spárování a připojení s jiným zařízením, data mohou být vysílána mezi těmito zařízeními. V případě neaktivního připojení dochází stále k vyhledávání zařízení ke spárování.

### **Příkazový režim:**

V tomto režimu jsou všechny přijaté informace přes sériové UART rozhraní považovány za příkazy. Nastavují se zde například parametry pro komunikaci, nastavení master / slave režimu a názvu modulu, párovacího klíče, přístupové dotazy na adresy zařízení, volba režimu úspory energie, obnovení továrního nastavení a další. Režim se dělí ještě na dva pod režimy:

- S omezenými příkazy: Přenosová rychlost je shodná jako v komunikačním režimu. Jsou dostupné jen některé AT příkazy.
- Úplné řízení: V tomto režimu je definována přenosová rychlost 38400bps a je možno používat všechny dostupné příkazy modulu.

Restartováním modulu je automaticky nastaven komunikační režim.



Obr. 22. Schéma zapojení HC-05 modulu k mikrokontroléru.

Kromě keramického a elektrolytického blokovacího kondenzátoru je v obvodu napájení jumper, který slouží k aktivaci režimu úplného řízení pro zadávání příkazů. Tohoto režimu je dosaženo rozpojením obvodu napájení, stisknutím a přidržením tlačítka na modulu a současném připojením napájení. Režimu s omezenými příkazy lze využít pouze při stisknutém tlačítku na modulu bez nutnosti manipulace s napájením.

### 5.7.2 RS232 převodník Bolutek BLK–MD-BC04-B

Tento modul na rozdíl od HC-05 disponuje obvodem Maxim 3221e pro převod napěťových úrovní RS232 na TTL a opačně. Na desce je integrován konektor DB9P, modul lze tedy přímo připojovat k zařízením komunikujících signály na úrovni standardu RS-232. Modul potřebuje ke správné funkci externí napájení. Napájení může být buď z akumulátoru nebo přemostěním externího USB napěťového signálu na vstup stabilizátoru napětí obsaženého na DPS.

Modul v realizaci figuruje v roli Slave. Pomocí tohoto modulu komunikuje simulační aplikace běžící na PC, modul je propojen s PC využitím převodníku USB rozhraní na RS-232 značky PremiumCord obsahujícím čip FTDI. Protože převodník i modul mají zakončení typu male (zástrčka), bylo nutné mezi ně vřadit redukci Female to Female zapojení typu 1:1. Modul je zařízení typu DCE, převodník úrovní DTE.



Obr. 23. Zařízení Bolutek BLK–MD-BC04-B připojené k převodníku USB to RS232.

Na rozdíl od HC-05 má BC04 jen komunikační režim, který má rozdílnou funkci v různých rolích modulu. AT příkazy lze zadávat, pokud je modul v roli Master nebo Slave. Pokud je modul v roli Slave a je právě spojen s jiným zařízením, nelze zadávat příkazy.

Tab. 5. Provozní stavy Bolutek BLK–MD-BC04-B.

Role	LED indikace (modrá)	Popis
------	----------------------	-------

<b>Master</b>	rychlé blikání (3 bliknutí za sekundu)	Hledání zařízení pro spárování (Inquiring)
	nepřetržitý svit	Spárováno/ připojeno k zařízení
<b>Slave</b>	Jedno bliknutí za sekundu	Čekání na příchozí spojení (Pairable)
	nepřetržitý svit	Spárováno/ připojeno k zařízení

Modul disponuje tlačítkem na desce, dlouhý stisk (> 3 s) slouží k resetu parametrů na výchozí hodnoty, krátkým stiskem dojde ke smazání uložené Bluetooth adresy zařízení, ke kterému byl nebo má být modul spárován.

Tyto moduly byly vybrány pro návrh ovládací jednotky z důvodu jejich nízké pořizovací ceny, velmi malým rozměrům a vyhovujícím provozním parametrům, které nabízejí.

### 5.7.3 Nastavení parametrů pro komunikaci

Moduly jsou nastavovány AT příkazy pomocí sériové linky, každý modul má tyto příkazy mírně odlišné. HC-05 byl nastaven za pomoci vývojové desky Arduino Uno. Bolutek BLK–MD-BC04-B byl nakonfigurován za pomoci terminálového emulátoru Tera Term. Parametry obou modulů v defaultní konfiguraci jsou:

Baud Rate: 9600 Bd, Datové bity: 8 bitů, Stop Bit: 1, Parita: bez paritního bitu, Handshake: bez handshake, Heslo pro spárování: 1234.

#### 5.7.3.1 Příklad nastavení modulu Bolutek BLK–MD-BC04-B

Aby bylo možné modulu posílat AT příkazy aplikací Tera Term, je zapotřebí v programu nastavit potřebné parametry přenosu – komunikační port (COM), Baud Rate, datové bity, paritní bit, stop bit tak, aby byly shodné s parametry modulu. Dále je nutné za každým AT příkazem posílat speciální řídicí znaky, aby modul rozlišil jednotlivé příkazy:

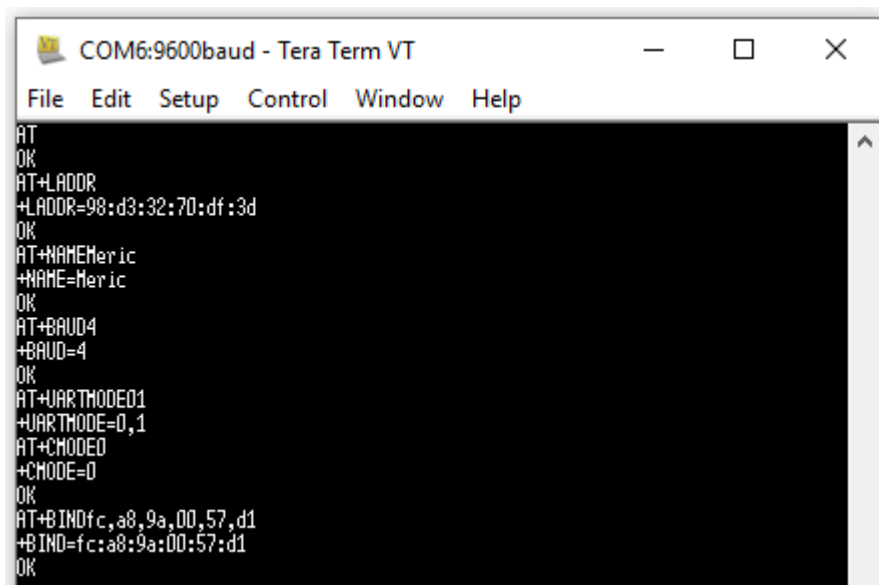
- Carriage Return (posun kurzoru na začátek řádku): \r
- Line Feed (Nový řádek): \n

Tab. 6. Syntaxe AT příkazů.

Syntax			Odpověď modulu
AT+	Příkaz	Parametr	OK nebo ERROR

Pro nastavení přenosové rychlosti 9600 Bd je v manuálu modulu vyhledán odpovídající parametr a to je 4. Příklad správného zápisu je tedy: **AT+BAUD4\r\n**. Při nesprávném zadání

příkazu modul odpoví výpisem chyby (Error) s odpovídajícím kódem chyby, kterou lze také vyhledat v manuálu. Na následujícím obrázku jsou zobrazeny všechny příkazy potřebné k nakonfigurování modulu pro účely komunikace v této práci.



```
COM6:9600baud - Tera Term VT
File Edit Setup Control Window Help
AT
OK
AT+LADDR
+LADDR=98:d3:32:70:df:3d
OK
AT+NAMEMer ic
+NAME=Mer ic
OK
AT+BAUD4
+BAUD=4
OK
AT+UARTMODE01
+UARTMODE=0,1
AT+CMODE0
+CMODE=0
OK
AT+BINDfc,a8,9a,00,57,d1
+BIND=fc:a8:9a:00:57:d1
OK
```

Obr. 24. Konfigurace BC04 modulu v programu Tera Term pomocí sériové linky.

### Konfigurace:

Pro lepší orientaci mezi Bluetooth zařízeními je vhodné zařízení pojmenovat, k tomu slouží příkaz AT+NAME. Příkazem AT+UARTMODE01 jsou nastaveny parametry pro UART: 0 – jeden stop bit, 1 – lichá parita. Aby bylo zajištěno, že se moduly navzájem připojí po startu a nedojde tedy ke spárování s jiným náhodným zařízením, bylo zapotřebí pevně přiřadit (nabindovat) navzájem jejich Bluetooth adresy příkazem AT+BIND. Ke zjištění adresy BC04 slouží příkaz AT+LADDR. AT+CMODE0 aktivuje režim pro připojení pouze k nabindované Bluetooth adrese modulu HC-05 ovládací jednotky. Moduly mají pevně definován maximální počet 8 přenášených datových bitů.

## 5.8 Realizace zařízení a dodatečné úpravy

Návrh desky plošných spojů byl zaslán na výrobu firmě Gatema a.s. Vyrobená deska byla osazena ručně. Po zprovoznění zařízení bylo nutné vyřešit některé nedostatky v návrhu a modifikovat desku plošných spojů tak, aby realizované zařízení mohlo pracovat spolehlivě. Šlo o přidání Shottkyho diody v závěrném směru vůči signálu směřujícím na vstup mikrokontroléru KBI1P4 v obvodu zapínání/vypínání ovládací jednotky, aby nedocházelo ke



proudovým svodům přes pin mikrokontroléru ve stavu bez napájení. Další komplikace nastala, když integrovaný obvod nabíjení po nabití baterie nastavil na výstupu STAT log.1, přes tento výstupní pin a dělič napětí bylo na vstupní pin MCU přivedeno napětí 5 V, díky kterému docházelo k zapnutí mikrokontroléru ve stavu bez napájení. Pro detekci USB nabíjení byl přidán bipolární tranzistor VT7 a dělič napětí, kolektor tranzistoru byl připojen vodičem k pinu PTA0. Tyto úpravy byly zapracovány do schématu a návrhu desky plošného spoje obsažených v přílohách této práce. Reset mikrokontroléru může být proveden hardwarovým tlačítkem na desce plošného spoje. Baterie byla přilepena k desce plošného spoje pomocí oboustranné lepící pásky a Bluetooth modul HC-05 byl napevno připájen k DPS. LCD displeje byly podloženy distančními sloupky, tak aby se nacházely oba ve stejné výšce a přitom byla dodržena dostatečná výška pro viditelnost na displeje při zapouzdření zařízení. DPS je umístěna do vodě odolné plastové krabičky.

## 6 SOFTWAREVÉ ŘEŠENÍ OVLÁDACÍHO A ZOBRAZOVACÍHO ZAŘÍZENÍ

Program pro ovládací jednotku byl napsán v jazyce C ve vývojovém prostředí CodeWarrior od firmy NXP. Dále byl při programování v tomto prostředí použit integrovaný vývojový nástroj Processor Expert, který generuje inicializační zdrojový kód a základní funkce pro jednotlivé vybrané periferie mikrokontroléru na základě navolených parametrů v uživatelském rozhraní pro vybranou komponentu. Processor Expert však nelze použít pro každý stavební prvek ovládací jednotky, tudíž je nutné například pro LCD displeje nebo maticovou klávesnici ovladače/knihovny vytvořit. Knihovny dalších periférií jsou postaveny na základních funkcích generovaných Processor Expertem.

### 6.1 Soubor požadovaných parametrů pro nastavení simulátoru měřiče rychlosti

Pro komunikaci se simulační aplikací je využito sériového asynchronního přenosu dat dle standardu RS232. Parametry pro přenos jsou nastaveny takto:

- přenosová rychlost 9600 Baud
- start bit ("0")
- 8 informačních bitů
- paritní bit – lichá parita (ODD)
- jeden stop bit ("1")
- bez handshake

#### Soubor řídicích parametrů simulátoru měřiče rychlosti

- **limit rychlosti:**
  - osobní auto: 0 až 250 km/h
  - nákladní auto: 0 až 250 km/h
- **směr:** příjezd/ odjezd/ oba směry
- **dosah:** 20 m/ 30 m/ 60 m
- **režim:** automaticky/ručně
- **příkazy:** start / stop měření

Jde o parametry zadané na ovládací a zobrazovací jednotce pomocí klávesnice, tyto parametry jsou vyobrazeny na velkém LCD displeji (16x2) ovládací a zobrazovací jednotky.

### Soubor parametrů pro zobrazení na ovládací a zobrazovací jednotce

- rychlost změřeného vozidla (000,0 – 999,9):  
stovky [km/h], desítky [km/h], jednotky [km/h], desetiny [km/h].
- směr: příjezd/ odjezd
- typ vozidla: osobní/ nákladní auto

Informace zaslané simulační aplikací zobrazované na malém LCD displeji ovládací a zobrazovací jednotky.

## 6.2 Hlavní program ovládací a zobrazovací jednotky

Logická struktura ovládání ovládací a zobrazovací jednotky může být popsána jako konečný automat. Ovládací a zobrazovací jednotka má 8 stavů (režimů), kdy právě v jednom z nich je možno se nacházet. Pomocí tlačítek maticové klávesnice „klávesa nahoru“, „klávesa dolů“ jsou realizovány přechody mezi těmito režimy. Výstup v jednotlivých režimech tedy závisí na vstupu z klávesnice (proměnná **stav**) a vnitřním stavu (proměnná **rezim**) ve kterém se právě automat nachází. Výstupy těchto režimů mohou být například výpisy přijatých a zpracovaných hodnot, odstartování převodu A/D převodníku, inkrementace a dekrementace řídicích proměnných a jejich výpis na LCD displej, odeslání nastavených hodnot sériovou linkou a další.

## 6.3 Ovládání zařízení

Konečný automat je realizován nekonečným cyklem **while** (1) a přepínačem **switch**. Je definován počáteční stav (**int rezim** = 0). Proměnná **extern int stav** je aktualizována asynchronně v podprogramu přerušení od tlačítka (KBI). V případě, kdy je hodnota proměnné **stav** = 0 nebo **stav** = 6, došlo ke stisku tlačítek pro přechod mezi režimy. Konečný stav (vypnutí ovládací a zobrazovací jednotky) může nastat asynchronně v jakýkoli čas, protože se nalézá v podprogramu obsluhy přerušení od tlačítka.

```

while(1) {
    switch(rezim)
    {
        case 0 //režim pro nastavení limitu rychlosti
            while(rezim == 0){
                switch(stav){
                    case 0
                        //výpis na LCD, odeslání hodnoty limitu
                        stav = -1;
                        break;
                    case 1
                        //zvuková signalizace
                        stav = -1;
                        break;
                }
                controlsRezim();
            };
            break;
        ..case 1 //režim pro zobrazení stavu baterie a nabíjení
        ..case 2 //režim měření
        ..case 3 až case 7
    }
    controlsRezim();
}

```

Obr. 25. Obecná logická struktura řešení jednotlivých menu ovládací jednotky.

Funkce `controlsRezim()` zajišťuje inkrementaci nebo dekrementaci proměnné režim a také ošetřuje překročení počtu definovaných stavů.

```

void controlsRezim() {
    if(stav == 0){rezim++; stav=-1;} // sipka nahoru
    else if(stav == 6){rezim--; stav=-1;} // sipka dolu
    if(rezim < 0){rezim = 7;}
    else if(rezim > 7){rezim = 0;}
}

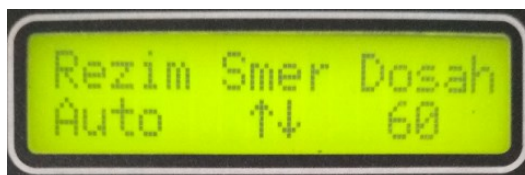
```

### Menu 1 (case 0):

Slouží k volbě režimu měření, směru, dosahu radaru. Hodnoty pro každý z těchto parametrů jsou pevně definovány v polích znaků jako řetězce. Jejich nastavování je realizováno inkrementací a dekrementací indexovacích proměnných pro daná pole. Po nastavení požadované hodnoty dojde k okamžitému vyslání odpovídajícího řetězce z pole pomocí funkce `uart-Dosah(byte *string)` obsažené v knihovně `uartDriver.c`. Při hodnotě `param_cont.parDosah = 2` je odeslán parametr D60 z následujícího pole:

```
unsigned char paramDosah[3][6] = {"D20\r\n", "D30\r\n", "D60\r\n"};
```

D představuje kategorizační znak pro dosah. Všechny funkce rozhraní UART posílají tyto řetězcová data po jednotlivých znacích pomocí cyklu for.



Obr. 26. Menu pro nastavení režimu, směru, dosahu měřiče.

### Menu 2 (case 1):

Slouží k nastavení limitu pro osobní a nákladní auta. Pro odeslání je nastavené celé číslo rozloženo na stovky (hundreds), desítky (tens), jednotky (units). Následně jsou pomocí funkce `uartLimitN(units,tens,hundreds)` uloženy do pole bytů ve tvaru:

```
unsigned char digits[] = {'N','hundreds','tens','units','\r','\n'};
```

Pole obsahuje také kategorizační znak (N – limit nákladní) a speciálními řídicími znaky `\r\n` pro zajištění jednoznačného příjmu a zatřídění daného parametru simulátorem.



Obr. 27. Menu pro nastavení limitů rychlosti.

Nastavování může probíhat buď krátkými stisky tlačítka nebo při delším přidržení cca na 3 sekundy dojde k automatické inkrementaci nebo dekrementaci hodnoty. Rychlost nastavování je určena použitím komponenty časovače funkcí `FC321_GetTimeMS(&time)`. Funkce vrací aktuální hodnotu času časovače. Pokud dojde k napočítání do určité hodnoty času, je provedena inkrementace/dekrementace proměnné a časovač je resetován. Takto je realizováno automatické změny hodnoty v určité rychlosti, že na ni uživatel může reagovat.

### Ukládání nastavených hodnot

Nastavené hodnoty pomocí menu 1 a 2 jsou ukládány do externího obvodu EEPROM funkcí `save_parameters()` při vypínání ovládací jednotky. Funkce zapisuje strukturu `pcontrol` obsahující všechny nastavované parametry na adresu EEPROM 0x00. Při každém zapnutí jed-

notky tedy dojde k načtení této uložené struktury z EEPROM funkcí struct pcontrol readparameters\_eeprom(0x00), takže není zapotřebí nastavovat tyto hodnoty znovu, pokud je není zapotřebí měnit.

### Menu 3 (case 2):

Zde jsou zobrazovány informace o počtu změřených vozidel a počtu přestupků pomocí funkce print\_treti\_rezim(); Funkce se nachází v knihovně controls.c, informace o nadlimitním měření (přestupku) je přijímána ze simulační aplikace, počet změřených vozů je počítán na základě všech přijímaných měření ze simulátoru. Tyto informace je možno vynulovat odpovídajícími tlačítky.



*Obr. 28. Menu pro zobrazení počtu měření a přestupků.*

### Menu 4 (case 3):

Režim měření, aktuálně zvolené parametry pro nastavení měřiče rychlosti pomocí předchozích menu jsou zobrazeny na spodním LCD 16x2 displeji.

V tomto režimu lze měnit režim měření (Automatický/Manuální) a směr, ve kterém budou jednotlivá měření zaznamenávána. O – osobní limit, N – nákladní limit, 60 je hodnota v metrech určující dosahu radaru, šipky určují směr měření. Stiskem tlačítka START dojde ke spuštění měřicího cyklu aplikace měřiče rychlosti. Na horním LCD displeji 12x2 jsou zobrazovány informace zaslané simulační aplikací, a to **aktuální rychlost s přesností na desetiny, směr a druh vozidla, číslo snímku měření**. Informace o nadlimitním měření je indikováno zvukovým signálem pomocí buzzeru. Po stisknutí tlačítka STOP dojde přerušeni cyklu měření v počítačové aplikaci a do EEPROM je zapsána hodnota počtu přijatých nadlimitních měření, pro případné smazání jen právě těchto hodnot. Při manuálním režimu měření je přijat pouze jeden záznam měření a poté je automaticky odeslán příkaz stop do simulátoru.

Tato data měření jsou přijímána v obsluze přerušeni od UART rozhraní, přerušeni je generováno při korektním přijetí znaku ze sériové linky. Jde o událost void AS1\_OnRxChar(void). Při každém vyvolání této události je jeden přijatý znak uložen do

odpovídající části bytového pole, přijaté znaky jsou ukládány do bytového pole v neměnném pořadí. V okamžiku, kdy jsou v poli uloženy všechny očekávané parametry dojde k jejich vypsání na horní displej spolu s aktuálním časem z obvodu RTC, poté je vynulováno bytové pole spolu s indexační proměnnou a ovládací jednotka je připravena přijmout data dalšího záznamu měření.

Přijatému záznamu měření zobrazenému na obrázku č. 28 odpovídá bytové pole ve tvaru:

```
array[message_count] = {stovky, desítky, jednotky, desetiny, směr, druh vozidla, pře-  
stupek}, message_count = 0 až 6
```

```
array[7] = { ' ', '6', '0', '5', '0xC6', 'N', '0x01' };
```

Datum a čas je přiřazován z obvodu reálného času DS3231 pomocí funkcí RTC1\_GetTime(&currTime) a RTC1\_GetDate(&date). Proměnné currTime a date jsou strukturovaného typu.



Obr. 29. Režim měření.

Všechna tato data měření jsou ukládána do externí nevolatilní EEPROM paměti po každém výpisu na displej. Zapisována je struktura **struct measurements** na počáteční adresu v EEPROM = 0x14. Pro další zápis je adresa inkrementována o velikost této struktury.

**Menu 5 (case 4):**

Režim zobrazuje aktuální stav nabití akumulátoru v procentech a jeho hodnotu napětí ve Voltech. Dále indikuje stav nabíjení baterie přes USB rozhraní, a to pomocí funkce chargeState(), která čte na vstupních pinech PTDD1(STAT), PTAD3(USBdetect) mikrokontroléru logické hodnoty a na základě jejich kombinací je vyhodnocován aktuální stav.

Pomocí funkce get\_adcValue() je realizován A/D převodníkem převod analogové hodnoty napětí akumulátoru na digitální. Pomocí vztahu (5) je vypočítána hodnota napětí ve voltech, toto napětí je poté přepočítáváno na procentuální hodnotu vztahem:

$$Battery[\%] = \frac{U_{BAT\_ACTUAL} - U_{BAT\_DEPLETED}}{U_{BAT\_FULL} - U_{BAT\_DEPLETED}} * 100 \quad (7)$$

Využitím přerušení od časovače TII\_OnInterrupt(void) je stav nabíjení aktualizován v určeném časovém intervalu, tak aby došlo ke včasnému rozpoznání změny tohoto stavu. K aktualizaci hodnoty napětí baterie dochází každé 4 minuty. Při nízkém stavu baterie (Batt [%] < 8%) dojde ke zvukové signalizaci. Před téměř plným vybitím baterie dojde k uložení dat do EEPROM paměti a vypnutí jednotky, aby nedošlo ke ztrátám dat a chybám při zápisu.



*Obr. 30. Režim aktuálního stavu baterie a detekce nabíjení.*

#### **Menu 6 (case 5) a Menu 7 (case 6):**

Tyto režimy jsou určeny pro nastavování času a data. Nastavované hodnoty jsou odesílány po sběrnici I2C pomocí funkcí RTC1\_SetTime(uint8\_t hourS, uint8\_t minS, uint8\_t secS, uint8\_t secS\_hund), RTC1\_SetDate(word year, uint8\_t month, uint8\_t day) do obvodu reálného času. Poslání hodnot je provedeno stisknutím tlačítka PHOTO.



*Obr. 31. Režim 6 pro nastavení času a 7 pro nastavení datumu.*



**Menu 8 (case 7):**

Toto menu nabízí zobrazení aktuálního času a datumu v reálném čase s pomocí komponenty časovače FC321, indikaci navázání spojení Bluetooth modulů. BT ! indikuje, že připojení nebylo navázáno, BT C že k připojení došlo. Toto menu také slouží pro výpis nebo mazání obsahu paměti EEPROM.



Obr. 32. Režim 8.

Použité funkce:

`erase_eeprom(int address, int numof_chunks)` – funkce maže uložená měření, první argument udává adresu, druhý udává počet měření ke smazání z paměti. Smazání dat je potvrzeno výpisem EE DEL na displej.

`readmeasure_eeprom(int address)` – pomocí této funkce jsou vypsaný na terminál (například Tera Term) všechna zapsaná měření. Přečtení dat je potvrzeno výpisem EE RD na displej.

`vypisEEPROM()` – vypíše obsah části paměti na terminál ve formátu matice.

### 6.3.1 Výpis informací na LCD

Pro výpis řetězce na LCD displeje jednotky slouží funkce `LCD16_printChar()`, `LCD16_printStr()`, `LCD_printChar()`, `LCD_printStr()` obsažené v knihovně `LCDdriver.c`. Pro převedení číselných hodnot na řetězec je využíváno funkce `sprintf()`.

### 6.3.2 Maticová klávesnice

Knihovna `keyboard.c` obsahuje funkce pro vyhodnocení stisknutého tlačítka na maticové klávesnici o velikosti 3x3 plus tlačítka I/O pro vypnutí/zapnutí jednotky připojené k jednomu z řádků. Vyhodnocování je řešeno následovně:

1. Všechny výstupní piny řádků jsou nastaveny na logickou 0. Na vstupech jsou aktivovány pull-up rezistory. Stisknutím jakékoli stisknuté klávesy dojde k vygenerování přerušení.

2. V obsluze přerušení je volána funkce `detekceKlaves()`, která nejprve nastaví výstupní piny všech řádků na log. 1, poté je postupně v cyklu na každý řádek zapisována logická 0 a jsou čteny hodnoty na jednotlivých vstupních pinech. Pokud na jednom ze vstupních pinů je přečtena logická 0, dojde k přiřazení předem určené číselné hodnoty do proměnné **state** definující stisk odpovídajícího tlačítka v maticové klávesnici.

U některých tlačítek jsou rozlišovány stavy pro dlouhý stisk a krátký stisk tlačítka. To je řešeno pomocí inkrementace hodnoty proměnné **Button\_Count** a neustálém čtení hodnoty na vstupním pinu v cyklu `while` spolu s přidáním zpoždění a následném vyhodnocení hodnoty **Button\_Count** podmínkou. Tím je zjištěno, že tlačítko bylo stisknuto déle.

### 6.3.3 Úsporný režim

V přerušení od časovače `T11_OnInterrupt()` je pravidelně zjišťováno, jestli byla stisknuta nějaká klávesa. Pokud je po dobu 15 minut čtena stále stejná hodnota proměnné stav, ke stisku klávesy nedošlo a ovládací a zobrazovací jednotka není zrovna používána. Dojde tedy k vypnutí Bluetooth modulu, snížení intenzity podsvitu displejů pomocí PWM modulace. Z tohoto režimu je možno se dostat stiskem jakékoli klávesy.

## 7 SOFTWAREVÉ ŘEŠENÍ SIMULÁTORU RYCHLOMĚŘU

Tato aplikace je naprogramována v jazyce C# s využitím knihovny tříd pro tvorbu grafického rozhraní Windows Forms (WinForms), která je součástí .NET frameworku.

Aplikace měřiče rychlosti využívá 3 vlákna (thready), každé plní určitou funkci:

- Vlákno vytvořené událostí DataReceived při přijetí znaku ze sériové linky.
- Background worker, ve kterém běží metoda simulace měřiče rychlosti.
- UI vlákno

### 7.1 Popis vybraných metod

`void` `getAvailablePorts()`

Metoda získá názvy aktuálně dostupných sériových COM portů, hodnoty uloží do řetězcového pole a hodnoty z pole jsou vypsané do ovládacího prvku `comboBox`.

`private void` `initSerial(string chosenPort)`

Zde jsou nastavovány jednotlivé parametry sériové komunikace. Argumentem metody je aktuálně zvolený název portu pomocí GUI. Dále je nastavován baud rate 9600, lichá parita, jeden stopbit, 8 datových bitů, nadefinování rozpoznání ukončení řádku speciálními řídicími znaky `\r\n`. Vnitřní metodou `serialPort1.Open()` dojde k otevření sériové komunikace na zvoleném portu.

`private void` `DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)`

Metoda/událost je volána pokaždé, když dojde k přijetí dat přes sériový port. Tato metoda je volána v novém vlákně. Uvnitř události je použito metody `serialPort.ReadLine()`, která čte data z bufferu přijímače až po výskyt znaků `\r\n`. Tím je zaručeno jednoznačné přečtení každého jednotlivého nastavovacího parametru zasláného z ovládací jednotky. Všechny příchozí nastavovací parametry simulátoru měřiče jsou ukončeny znaky `\r\n`. Každý přijatý parametr je dále kategorizován ve funkci `categorizeIncomingData(string inData)` podle předem definovaného znaku, který je v něm obsažen.

`private void` `categorizeIncomingData(string inData)`

Přijatý nastavovací parametr je ukládán do řetězcové proměnné `inData` a jeho význam je rozeznáván na základě kategorizačního znaku. Například u přijatého textu „Sprijezd“ je metodou `inData.Substring(0, 1)` extrahován kategorizační znak S, a pomocí větviček podmínek

(if, else) je tento text(parametr) zařazen do odpovídající části pole parametrů a zároveň dojde k jeho výpisu v GUI pomocí ovládacího prvku textBox.

```
private byte limitControl(string limit, string actualMeasurement)
```

Metoda vrací informaci o přestupku. Pokud je hodnota aktuálního měření rychlosti větší, než je nastavený limit, metoda vrací hodnotu 1, v opačném případě 0.

```
private void writeToUart(string limit, string actualMeasurement, string ind, string category, string direction, byte tOffence)
```

Pomocí této metody je záznam měření vyhodnocený na základě nastavených kritérií zobrazen v GUI a odeslán přes sériovou linku. Hodnota aktuální rychlosti měření je rozčleněna na hodnoty stovek, desítek, jednotek a desetin. Metodami SerialPort.Write (zapisuje řetězec) a WriteByte (zapisuje pouze jeden byte) dojde k zapsání těchto dat do výstupního bufferu sériového portu. Aby nedošlo k chybám při příjmu dat u ovládací jednotky, mezi vysláním jednotlivých fragmentů záznamu měření je definováno zpoždění 20 milisekund.

```
public void meric(string[] parametersArray)
```

Metoda simuluje měřič rychlosti. Zajišťuje cyklické generování jednotlivých záznamů měření rychlosti. Cyklus je aktivován při přijetí příkazu „start“ z ovládací jednotky za podmínky, že všechny potřebné nastavovací parametry jsou uloženy v poli. Poté jsou náhodně vygenerovaná jednotlivá měření z tabulky záznamů zobrazována v GUI. Na základě uložených nastavovacích parametrů je metodami (limitControl, writeToUart) a podmínkami vyhodnocováno, jestli má být daný záznam měření zaslán do ovládacího zařízení. Příkazem „stop“ dojde k zastavení cyklu měřiče a uvedení aplikace do nečinného režimu.

Tato metoda běží asynchronně ve vláknu backgroundWorker. Dále jsou pomocí BackgroundWorkeru zajištěny výpisy systémových informací.

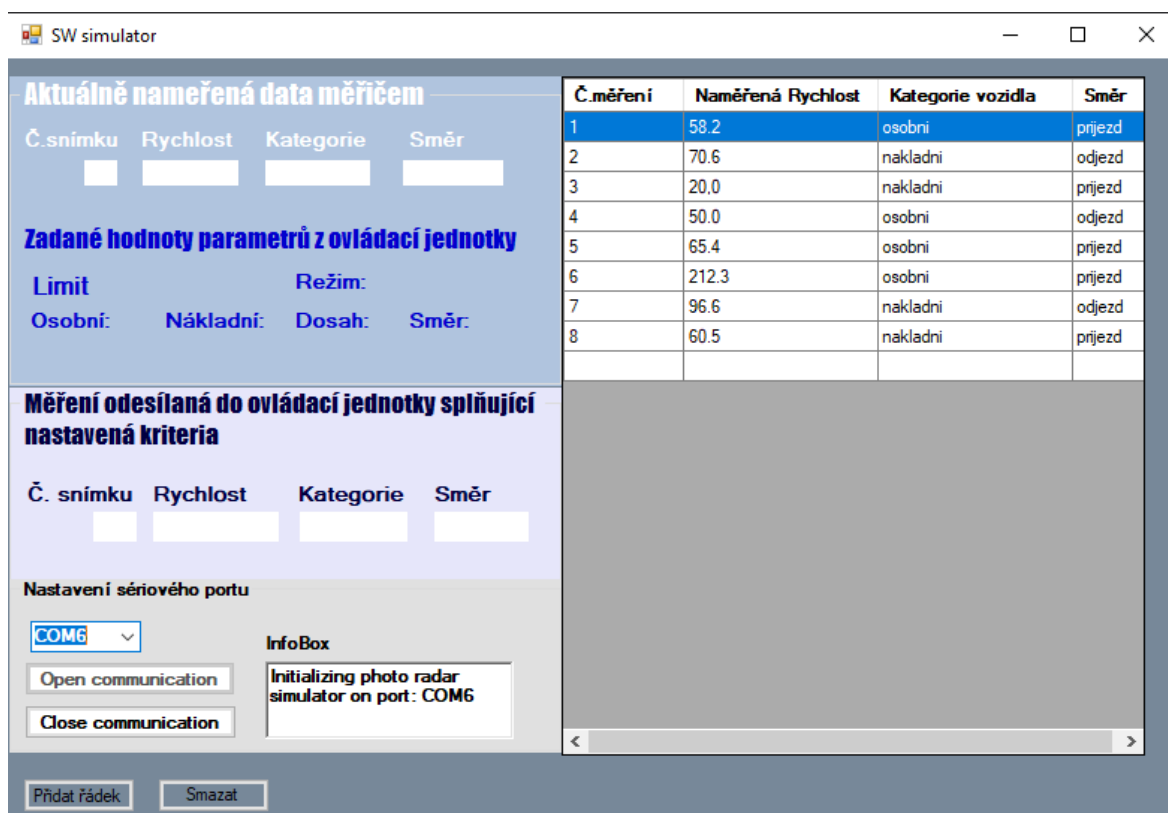
V případě potřeby přistoupit nebo změnit hodnoty ovládacích prvků ve vláknech UI (například změnit text v poli, zobrazit upozornění) z jiného vlákna, než ze kterého byly tyto prvky vytvořeny, je nutné použít metody Invoke/BeginInvoke/EndInvoke/InvokeRequired. K tomu slouží interface:

```
public static class ISynchronizeInvokeExtensions s rozšiřující metodou public static void InvokeEx<T>(this T @this, Action<T> action) where T : ISynchronizeInvoke obsahující všechny výše uvedené metody.
```

## 7.2 Popis aplikace

Simulační aplikace slouží pro příjem hodnot nastavovaných parametrů jejich zpracování a zobrazení pomocí GUI, na základě těchto vstupních dat jsou vyhodnocována jednotlivá měření generovaná z tabulky a tyto vyhodnocená data jsou odesílána zpět do ovládacího zařízení pomocí Bluetooth sériového přenosu RS-232.

Po spuštění programu je nejprve nutné vybrat sériový port z rozbalovacího seznamu všech dostupných portů, přes který bude probíhat komunikace. Stisknutím tlačítka „**Open communication**“ dojde k inicializaci vybraného sériového portu a vypsání potvrzující informace do informačního pole **InfoBox**. V tomto stavu aplikace čeká na řídicí povel z ovládací jednotky ke spuštění měření.



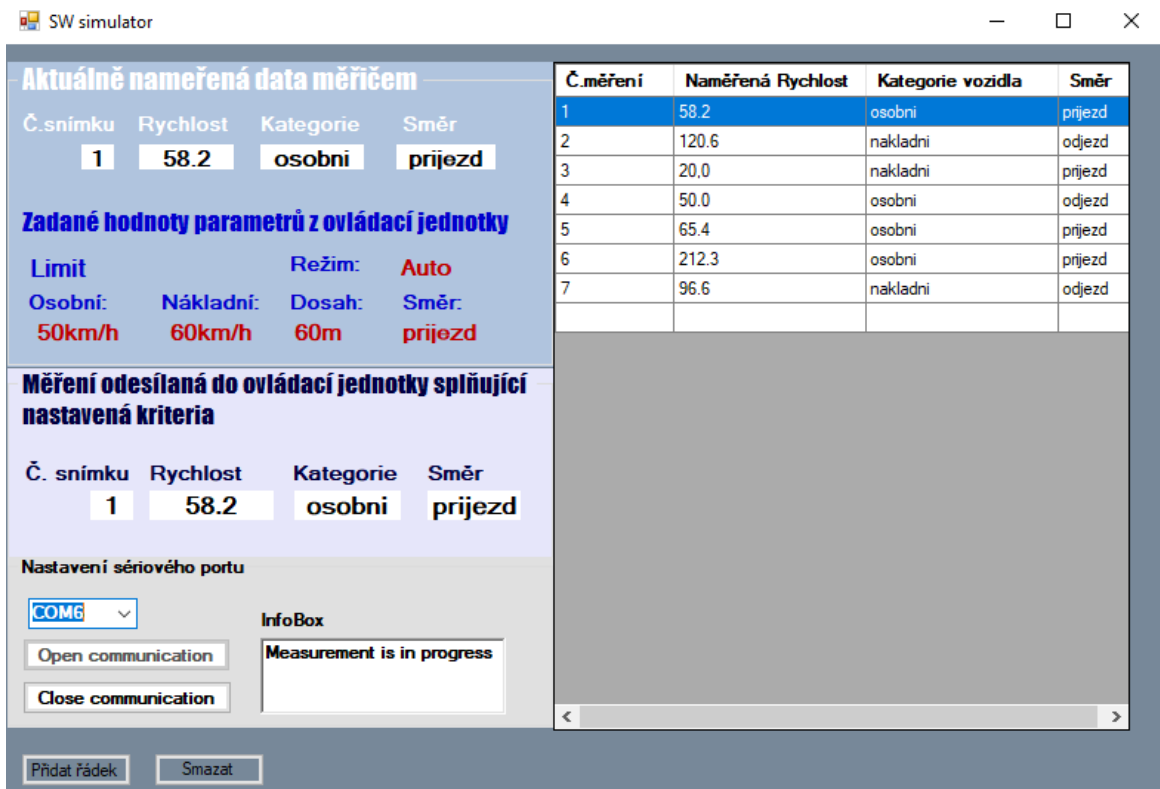
Obr. 33. GUI simulátoru měřiče rychlosti, nečinný režim.

Dále je nutné zaslat pomocí ovládací jednotky hodnoty parametrů **režim, limity pro osobní a nákladní vozy, směr a dosah radaru**. Dokud všechny parametry nejsou zvoleny, není možné měření spustit, parametr směru lze měnit i při spuštěném režimu měření. Po příjmu řídicího příkazu **start** dojde každých 5 sekund k vygenerování náhodného záznamu měření z tabulky do odpovídajících polí spadající pod blok „**Aktuálně naměřená data měřičem**“.

Do tabulky lze zadávat nové záznamy (tlačítko přidat řádek) nebo vybrané záznamy smazat (tlačítko Smazat), úpravy lze provádět za běhu i v režimu měření. Na základě zvolených parametrů dochází k vyhodnocování jednotlivých záznamů měření.

Pokud je zvoleno měření jen při příjezdu vozidla, do ovládací jednotky jsou zasílány pouze všechny záznamy splňující toto kritérium. V případě, kdy hodnota rychlosti v záznamu měření přesahuje nastavený limit pro odpovídající druh motorového vozidla, je ovládacímu zařízení zaslána informace o přestupku.

Na následujícím obrázku je již spuštěn režim měření rychlosti a je zde uvedeno změřené příjíždějícího osobního vozidla, které překročilo rychlostní limit 50 km/h, číslo snímku je 1. Informace zobrazené v bloku „Měření odesílané do ovládací jednotky splňující nastavená kritéria“ jsou zaslány do ovládací a zobrazovací jednotky.



Obr. 34. GUI simulátoru měřiče rychlosti, režim měření.



Obr. 35. Odpovídající záznam měření zobrazený na ovládací a zobrazovací jednotce.

## ZÁVĚR

Cílem diplomové práce bylo navrhnout a vytvořit prototyp ovládacího a zobrazujícího zařízení komunikujícího se simulační aplikací na PC. Vyvinutý prototyp ovládací a zobrazovací jednotky splňuje všechna kritéria a požadavky, které byly na něj kladeny. Kromě nastavování hodnot parametrů zařízení obsahuje také obvod reálného času se záložní baterií, Bluetooth modul pro možnost bezdrátového přenosu dat. Dále zařízení disponuje interním ukládáním naměřených dat s přiřazením datumu a času do externího obvodu paměti EEPROM, nechybí zvuková signalizace a samozřejmostí jsou použité podsvícené LCD displeje. Zařízení je možno provozovat zcela bezdrátově s napájením z vestavěného akumulátoru nebo může být pevně propojeno a napájeno kabelovým spojením. Návrh jednotky byl proveden tak, aby rozměry desky plošných spojů byly kompatibilní s mechanikou a zapouzdřením předchozího typu ovládací jednotky.

V rámci práce byla vytvořena také grafická počítačová aplikace simulátoru měřiče rychlosti z důvodu absence možnosti pracovat s reálným radarovým měřičem. Funkčnost prototypu byla tedy ověřena za pomoci simulační aplikace. Pro obousměrnou komunikaci byl sestaven vlastní protokol pro předávání hodnot parametrů a naměřených dat.

Práce je rozdělena na teoretickou a praktickou část. První kapitola uvádí obecnou problematiku přenosu informace. Druhá kapitola pojednává o sériovém přenosu informace. Třetí kapitola se zabývá problematikou Bluetooth komunikace a v poslední kapitole teoretické části jsou popisovány jednočipové mikropočítače spolu s konkrétním použitým mikrokontrolérem. V praktické části je popisován návrh zapojení zařízení a problémy, které nastaly po návrhu a realizaci zařízení a bylo je nutné pro korektní elektronickou i programovou funkčnost vyřešit. V poslední kapitole bylo popsáno programové vybavení zařízení, popis ovládání a funkce jednotky a simulační aplikace měřiče rychlosti. Přílohy práce zahrnují elektronické schéma zapojení jednotky, předlohu DPS, osazovací plán a seznam použitých součástek.

Všechny body zadání práce se podařilo splnit. Vylepšeními nad rámec práce by se mohlo jednat o přidání podpory SD karty pro ukládání záznamů měření, při stávajícím řešení je nutné uložená data odesílat z jednotky pomocí sériového rozhraní na terminál, nebo do případné aplikace, která bude zaznamenávat hodnoty zaslané po sériové lince. Dalším vylepšením může být využití výkonnějších Bluetooth modulů nebo Wi-Fi modulů. Vybrané Bluetooth moduly mají dosah pouze do 10 m, což je dostačující, ale na trhu v podobné cenové

relaci lze nalézt i výkonnější zařízení. Od toho se odvíjí vhodnost implementace dalšího režimu v ovládacím zařízení, určeného pro nastavování parametrů sériové komunikace u zvoleného bezdrátového modulu.



**SEZNAM POUŽITÉ LITERATURY**

- [1] KAINKA, Burkhard. *Měření, řízení a regulace pomocí PC: vývoj hw a sw pro praxi*. Praha: BEN, 2003. PC & elektronika. ISBN 80-7300-089-x.
- [2] *Opatření obecné povahy, kterým se stanovují metrologické a technické požadavky na stanovená měřidla, včetně metod jejich zkoušení při schvalování typu a ověřování stanovených měřidel: Silniční rychloměry používané při kontrole dodržování pravidel silničního provozu*. [online]. Brno: Český metrologický institut. ©2010 [cit. 2019-04-02]. 0111-OOP-C005-09. Dostupné z: [https://www.cmi.cz/sites/all/files/public/download/Uredni\\_deska/3405-ID-C\\_3405-ID-C.pdf](https://www.cmi.cz/sites/all/files/public/download/Uredni_deska/3405-ID-C_3405-ID-C.pdf)
- [3] PINKER, Jiří. *Mikroprocesory a mikropočítače*. Praha: BEN, 2004. ISBN 80-7300-110-1.
- [4] JURÁNEK, Antonín a Miroslav HRABOVSKÝ. *EAGLE pro začátečníky: uživatelská a referenční příručka: 2. vydání*. Praha: BEN, 2007, 191 s. ISBN 80-730-0213-2.
- [5] GOOK, Michael. *Hardwarová rozhraní: průvodce programátora*. Brno: Computer Press, 2006. ISBN 80-251-1019-2.
- [6] Codrey Electronics - Electronics and Embedded Systems. *What is Serial Communication and How it works?* [online]. ©2019 [cit. 2019-04-02]. Dostupné z: <https://www.codrey.com/embedded-systems/serial-communication-basics/>
- [7] Embedded Systems Course. *Module7: SERIAL COMMUNICATION* [online]. ©2017 [cit. 2019-04-03]. Dostupné z: <http://www.eeherald.com/section/design-guide/esmod7.html>
- [8] Vývoj.HW.cz. *HW server představuje: Sériová linka RS-232*. [online]. ©1997 [cit. 2019-04-04]. Dostupné z: <https://vyvoj.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>
- [9] Wikipedia, The Free Encyclopedia. *Wikipedia contributors. Bluetooth*. [online]. 2019 [cit. 2019-04-09]. Dostupné z: <https://en.wikipedia.org/wiki/Bluetooth>.
- [10] Serial Port Emulation. *RFCOMM with TS 07.10*. [online]. ©1999 [cit. 2019-04-10]. Dostupné z: [http://www.dcs.ed.ac.uk/home/slipb/Bluetooth%20VCC%20Models/Documents/bluetooth\\_fl.pdf](http://www.dcs.ed.ac.uk/home/slipb/Bluetooth%20VCC%20Models/Documents/bluetooth_fl.pdf)

- [11] NXP Semiconductors. *MC9S08AC128 Reference Manual*. [online]. NXP, 2008 [cit. 2019-04-12]. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/MC9S08AC128RM.pdf>
- [12] Codrey Electronics - Electronics and Embedded Systems. *UART Communication Protocol - How it works?* [online]. ©2018 [cit. 2019-04-21]. Dostupné z: <https://www.codrey.com/embedded-systems/uart-serial-communication-rs232/>
- [13] Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers. *MCP73831/2 - Miniature Single-Cell*. [online]. ©2014 [cit. 2019-04-23]. Dostupné z: [https://cdn.sparkfun.com/assets/learn\\_tutorials/6/9/5/MCP738312.pdf](https://cdn.sparkfun.com/assets/learn_tutorials/6/9/5/MCP738312.pdf)

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

A/D	Analog to Digital
ADC	Analog to Digital Converter
ASCII	American Standard Code for Information Interchange
BDM	Background Debug Mode
CLK	Clock
COM	Communication port
CPU	Central Processing Unit
CS	Chip Select
DC	Direct Current
DCE	Data Circuit-Terminating Equipment
DPS	Deska plošných spojů
DTE	Data Terminal Equipment
EDR	Enhanced Data Rate
EEPROM	Electrically Erasable Programmable Read-Only Memory
EPROM	Erasable Programmable Read-Only Memory
ETSI	European Telecommunications Standards Institute
FEE	FLL Engaged External
FLL	Frequency Locked Loop
FTDI	Future Technology Devices International
GND	Signal Ground
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
I/O	Input/Output
IIC/I2C	Inter-Integrated Circuit
IrDA	Infrared Data Association

---

IRQ	Interrupt ReQuest
KBI	Keyboard Interrupt
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LIN	Local Interconnect Network
LSB	Least Significant Bit
MCU	Microcontroller Unit
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MSB	Most Significant Bit
NRZ	Non Return To Zero
PC	Personal Computer
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
RTC	Real Time Clock
RTI	Real Time Interrupt
RxD	Receive Data
SCI	Serial Communication Interface
SCL/SCK	Serial Clock
SDA	Serial Data
SI	Slave Input
SMD	Surface Mount Device
SO	Slave Output
SPI	Serial Peripheral Interface Bus
TPM	Timer/PWM Module

TS	Technical Specification
TTL	Transistor-Transistor Logic
TxD	Transmit Data
UART	Universal Asynchronous Receiver Transmitter
UHF	Ultra High Frequency
UI	User Interface
USART	Universal Synchronous/Asynchronous Receiver and Transmitter
USB	Universal Serial Bus
WP	Write Protect

## SEZNAM OBRÁZKŮ

Obr. 1. Druhy přenosu informací.....	12
Obr. 2. Simplexní model přenosu.....	12
Obr. 3. Poloduplexní(Half-Duplex) model přenosu.....	12
Obr. 4. Plně duplexní (Full Duplex) model přenosu.....	13
Obr. 5. Schéma synchronní sériové komunikace.[6].....	15
Obr. 6. Sériový paket/rámec.....	17
Obr. 7. Konektor sériového portu DB-9P.....	18
Obr. 8. Přenosový rámec asynchronní sériové komunikace RS-232.[8].....	20
Obr. 9. Schéma UART rozhraní.....	21
Obr. 10. Zjednodušené schéma jednočipového mikropočítače.....	26
Obr. 11. Rozmístění pinů u MC9S08AC128, 64 pinové pouzdro, QFP[11].....	33
Obr. 12. Blokové schéma ovládací a zobrazovací jednotky.....	36
Obr. 13. Schéma obvodu pro nabíjení akumulátoru a detekce USB nabíjení.....	37
Obr. 14. Schéma obvodu napájení ovládací jednotky.....	39
Obr. 15. Baud Rate registr SCI1BDL.[11].....	43
Obr. 16. Stavový a řídicí registr ADCSC1.[11].....	44
Obr. 17. Schéma zapojení maticové klávesnice.....	45
Obr. 18. Perioda PWM a šířka pulsu.[11].....	46
Obr. 19. Schéma zapojení LCD displeje POWER TIP NPC1602LRU.....	48
Obr. 20. Schéma zapojení RTC obvodu DS3231.....	49
Obr. 21. HC-05 bluetooth modul.....	51
Obr. 22. Schéma zapojení HC-05 modulu k mikrokontroléru.....	53
Obr. 23. Zařízení Bolutek BLK–MD-BC04-B připojené k převodníku USB to RS232. .....	54
Obr. 24. Konfigurace BC04 modulu v programu Tera Term pomocí sériové linky.....	56
Obr. 25. Obecná logická struktura řešení jednotlivých menu ovládací jednotky.....	60
Obr. 26. Menu pro nastavení režimu, směru, dosahu měřiče.....	61
Obr. 27. Menu pro nastavení limitů rychlosti.....	61
Obr. 28. Menu pro zobrazení počtu měření a přestupků.....	62
Obr. 29. Režim měření.....	63
Obr. 30. Režim aktuálního stavu baterie a detekce nabíjení.....	64
Obr. 31. Režim 6 pro nastavení času a 7 pro nastavení datumu.....	64

---

Obr. 32. Režim 8.....	65
Obr. 33. GUI simulátoru měřiče rychlosti, nečinný režim. ....	69
Obr. 34. GUI simulátoru měřiče rychlosti, režim měření. ....	70
Obr. 35. Odpovídající záznam měření zobrazený na ovládací a zobrazovací jednotce. .....	70

**SEZNAM TABULEK**

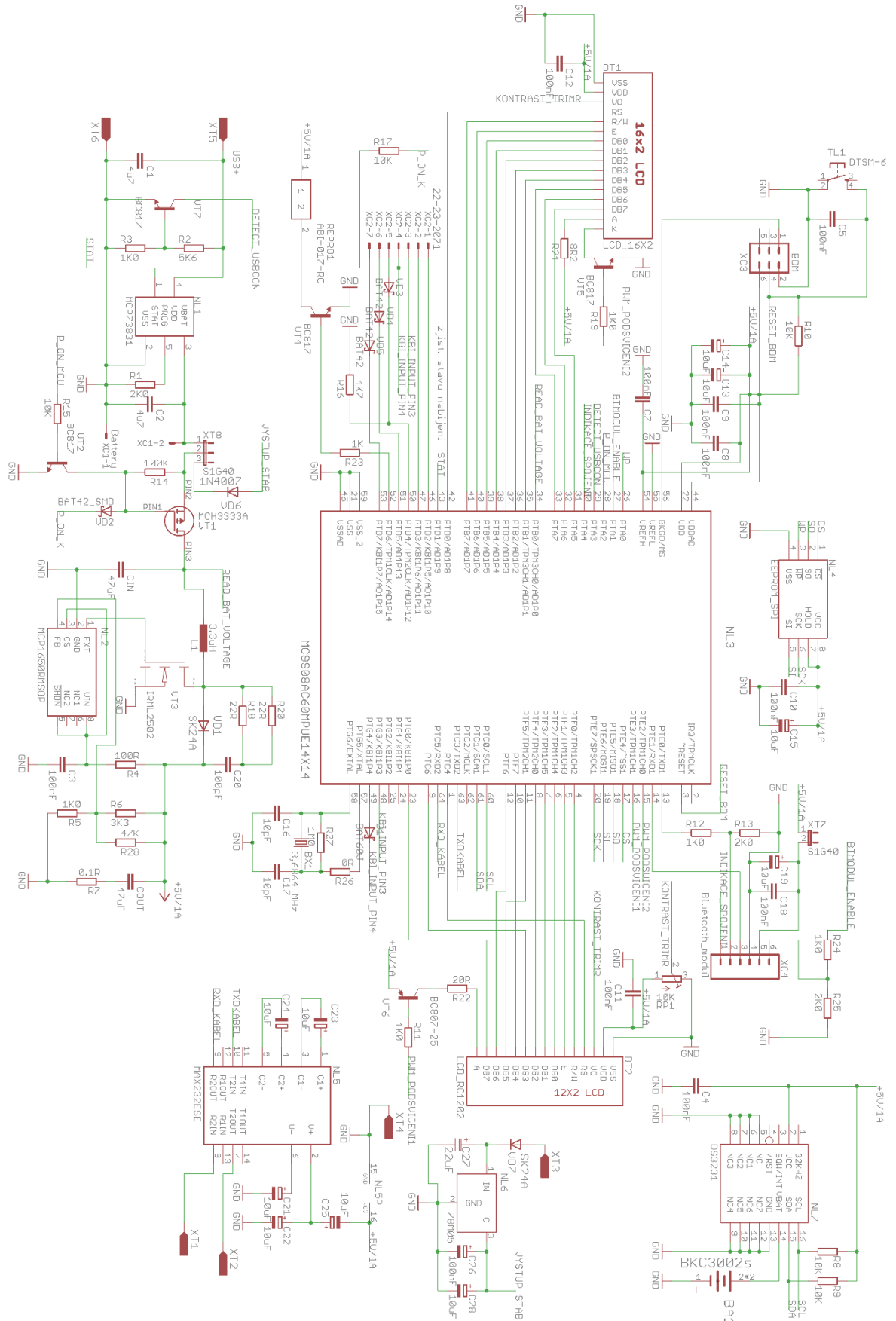
Tab. 1. Napěťové úrovně RS-232.....	18
Tab. 2. Funkce pinů sériového portu.[5].....	19
Tab. 3. Výkonnostní kategorie Bluetooth zařízení. ....	23
Tab. 4. TS 07.10 řídicí signály sériového portu.....	25
Tab. 5. Provozní stavy Bolutek BLK–MD-BC04-B.....	54
Tab. 6. Syntaxe AT příkazů. ....	55



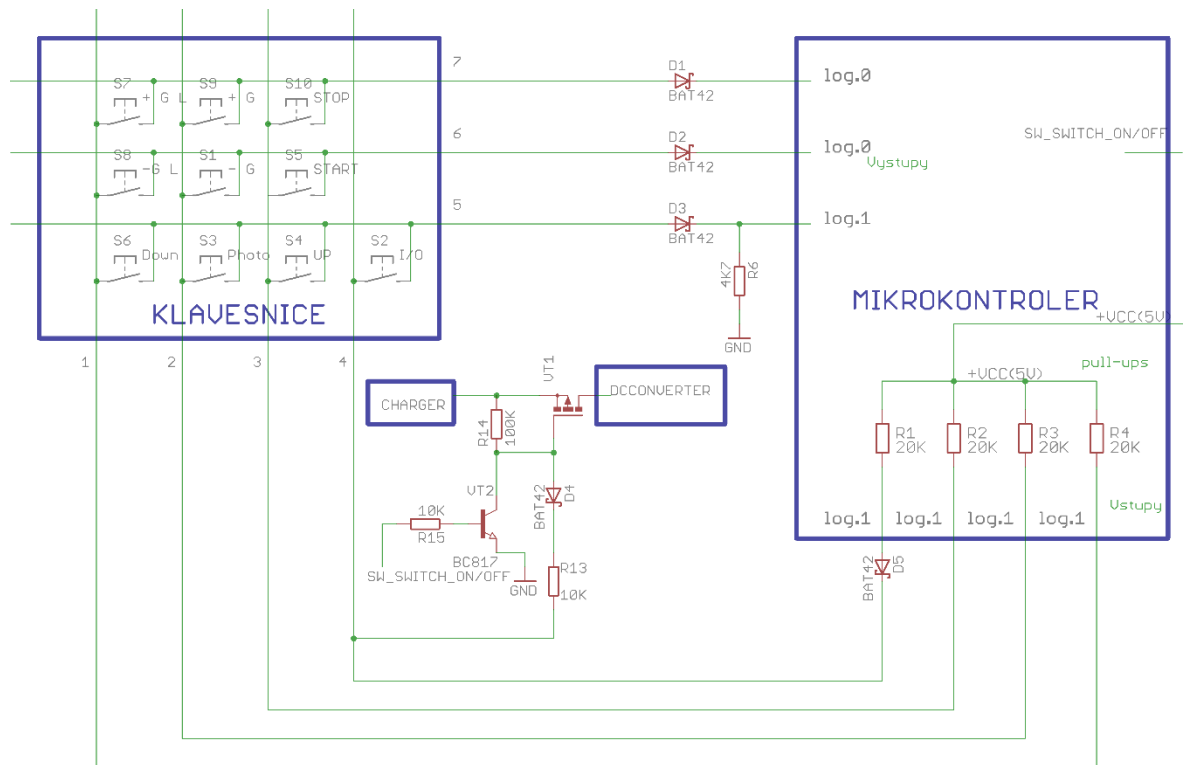
**SEZNAM PŘÍLOH**

P I	Schéma zapojení ovládací jednotky
P II	Schéma zapojení maticové klávesnice
P II	Schéma zapojení maticové klávesnice
P III	Deska plošného spoje horní část
P IV	Deska plošného spoje spodní část
P V	Osazovací plán horní strana
P VI	Osazovací plán spodní strana
P VII	Seznam použitých součástek
P VIII	Realizovaná ovládací a zobrazovací jednotka

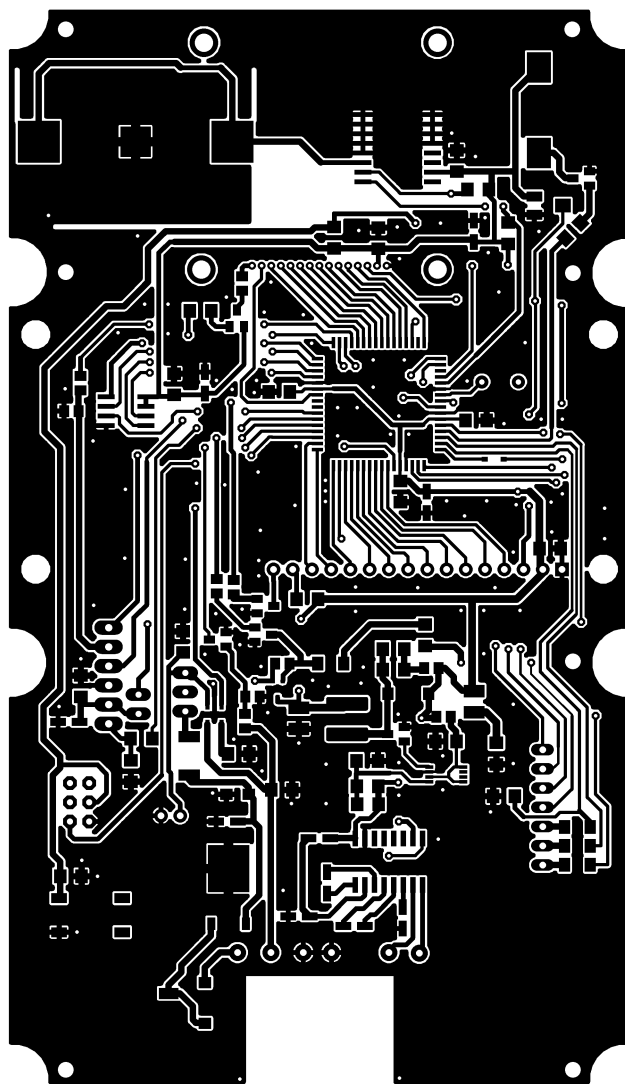
# PŘÍLOHA P I: SCHÉMA ZAPOJENÍ OVLÁDACÍ JEDNOTKY



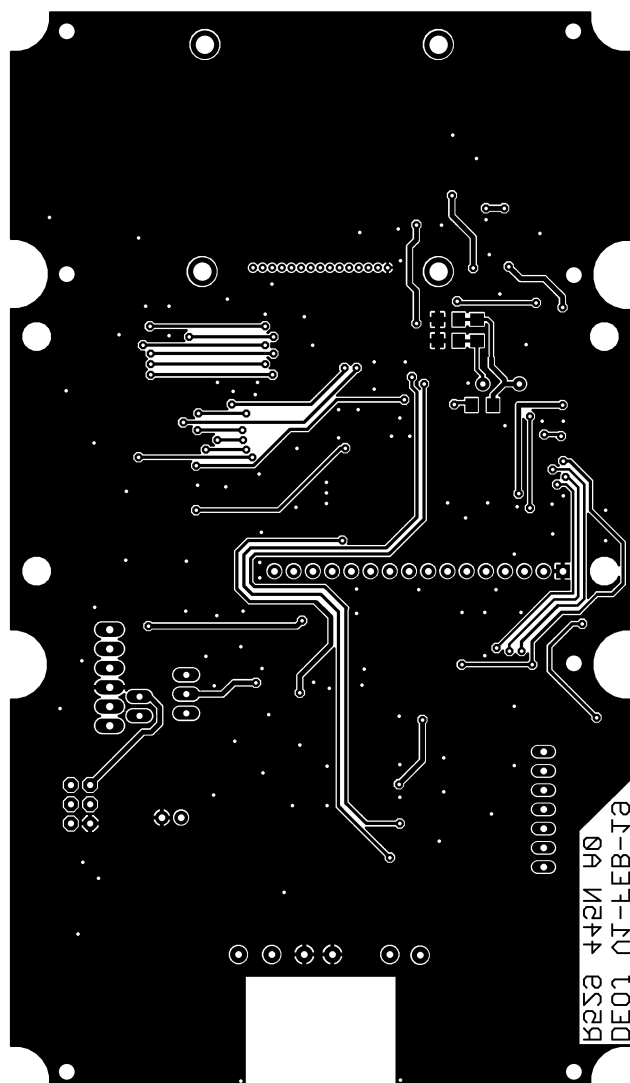
# PŘÍLOHA P II: SCHÉMA ZAPOJENÍ MATICOVÉ KLÁVESNICE



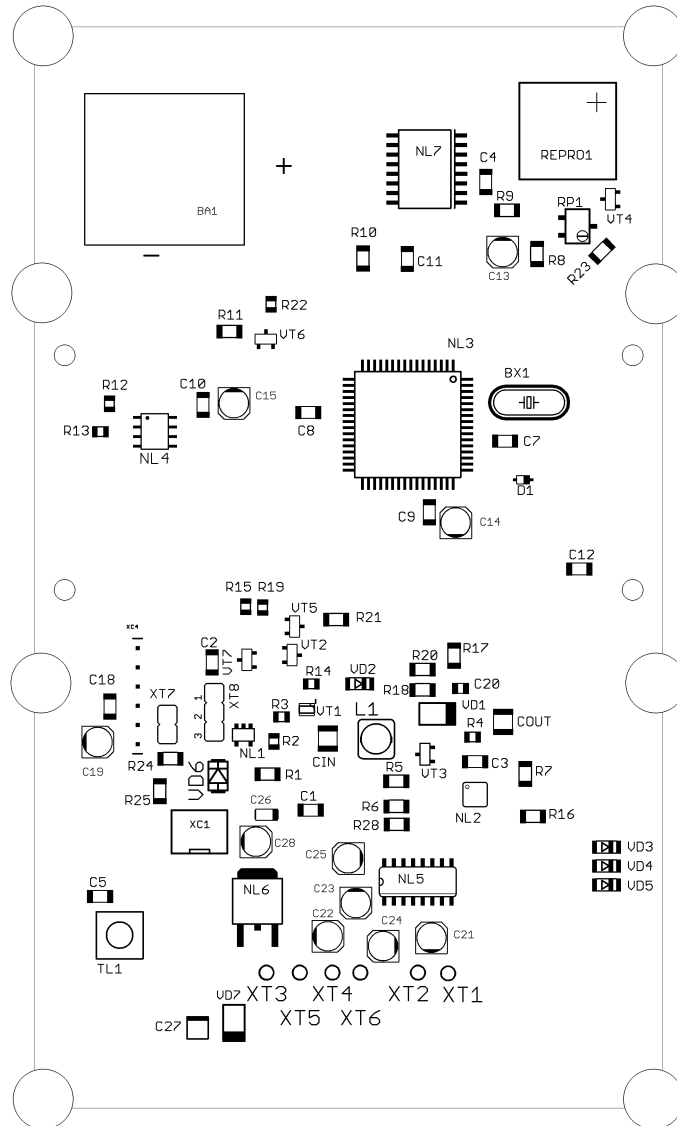
## PŘÍLOHA P III: DESKA PLOŠNÉHO SPOJE HORNÍ ČÁST



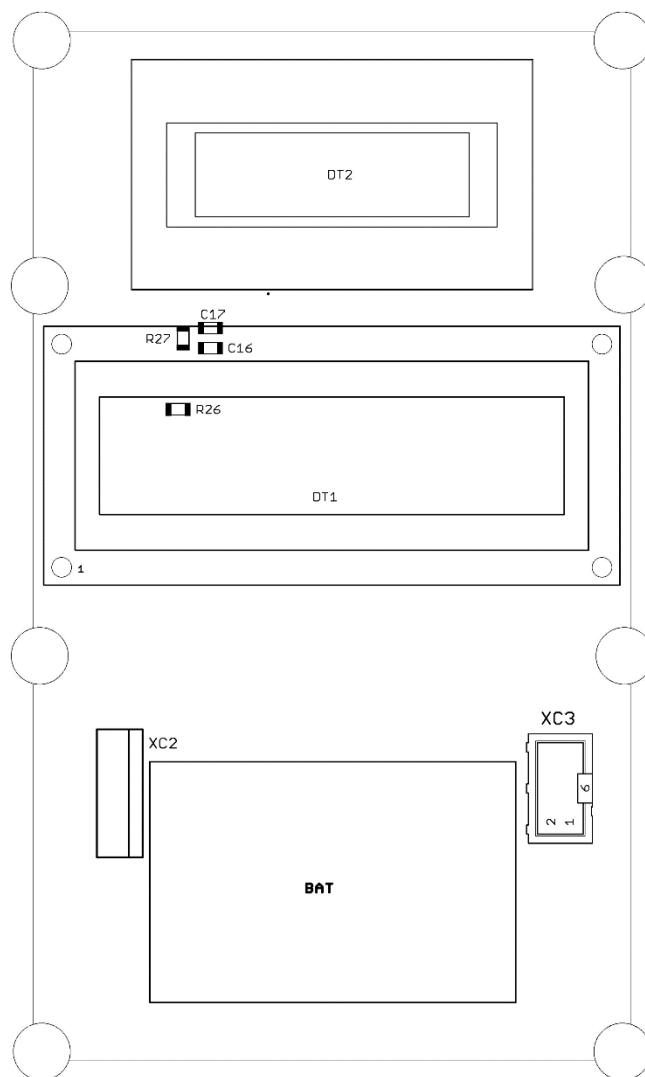
# PŘÍLOHA P IV: DESKA PLOŠNÉHO SPOJE SPODNÍ ČÁST



# PŘÍLOHA P V: OSAZOVACÍ PLÁN HORNÍ STRANA



## PŘÍLOHA P VI: OSAZOVACÍ PLÁN SPODNÍ STRANA



## PŘÍLOHA P VII: SEZNAM POUŽITÝCH SOUČÁSTEK

Označení na schématu	Hodnota	Zatížení	Druh	Číselný znak	
BA1	3,7V	1600mAh	BATERIE	3,7V/1600mAh	LP903450
BX1	3,6864MHz		KRYSTAL	3,6864 MHz	HC49US
C1, C2	4U7	25	KONDENZÁTOR	4U7	SMD1206
C3 ÷ C12, C18	100N	25V	KONDENZÁTOR	100N	SMD1206
C13 ÷ C15, C19, C21 ÷ C25, C28	10U	16V	KONDENZÁTOR	10U RM5,5	SEMT-CT B 4x5,4
C16, C17	10P	50V	KONDENZÁTOR	10P	SMD1206
C20	100P	50V	KONDENZÁTOR	100NP	SMD0805
C26	100N	35V	KONDENZÁTOR	100N	SMC_A
C27	22U	16V	KONDENZÁTOR	22U	SMD3528
CIN, COUT	47U	16V	KONDENZÁTOR	47U	SMD1210
DT1			LCD	POWERTIP NPC1602LRU-GWB-H	
DT2			LCD	RC1202A-YHY-ESX	
L1	3,3 uH		INDUKTOR	3,3 uH	SRR4528A
NL1			OBVOD INTEGROVANÝ	MCP73831	SOT23-5
NL2			OBVOD INTEGROVANÝ	MCP1650R	MSOP08
NL3			OBVOD INTEGROVANÝ	MC9S08AC60MPUE	QFP64_14X14
NL4			OBVOD INTEGROVANÝ	AT25512	SOIC8_JEDEC
NL5			OBVOD INTEGROVANÝ	MAX232ESE	SO16
NL6		1,5A	OBVOD INTEGROVANÝ	78M05	TO252
NL7			OBVOD INTEGROVANÝ	DS3231	SO16W
R1, R25	2K0	0,25W	ODPOR	2K0	SMD1206
R2	5K6	0,125W	ODPOR	5K6	SMD0805
R3, R12, R19	1K0	0,125W	ODPOR	1K0	SMD0805
R4	100R	0,125W	ODPOR	100R	SMD0805
R5,R11, R23, R24	1K0	0,25W	ODPOR	1K0	SMD1206
R6	3K3	0,25W	ODPOR	3K3	SMD1206
R7	0R1	0,25W	ODPOR	0R1	SMD1206
R8, R9, R10, R17	10K	0,25W	ODPOR	10K	SMD1206
R13	2K0	0,125W	ODPOR	2K0	SMD0805
R14	100K	0,125W	ODPOR	220R	SMD0805



R15	10K	0,125W	ODPOR	10K	SMD0805
R16	4K7	0,25W	ODPOR	4K7	SMD1206
R18, R20	22R	0,25W	ODPOR	22R	SMD1206
R20	10K	0,125W	ODPOR	10K	SMD0805
R15	10K	0,125W	ODPOR	10K	SMD0805
R3, R12, R19	1K0	0,125W	ODPOR	1K0	SMD0805
R4	100R	0,125W	ODPOR	100R	SMD0805
R5, R11, R23, R24	1K0	0,25W	ODPOR	1K0	SMD1206
R21	8R2	0,25W	ODPOR	8R2	SMD1206
R22	20R	0,125W	ODPOR	22R	SMD0805
R23	10K	0,25W	ODPOR	10K	SMD1206
R26	0R0	0,25W	ODPOR	0R0	SMD1206
R27	1M0	0,25W	ODPOR	1M0	SMD1206
R28	47K	0,25W	ODPOR	47K	SMD1206
REPRO			BUZZER	ABI-017-RC	
RP1	10K		TRIMR ODPOROVÝ	10K/N	TS53YLK010
TL1			TLAČÍTKO	TD-03XA-T	DTSM-6
VD1, VD7	U <sub>fm</sub> = 0,5V	40V/2A	DIODA	SK24A	DO214AC
VD2, VD3, VD4, VD5	U <sub>fm</sub> = 0,4V	30V/0,2A	DIODA	BAT42	MINIMELF
VD6	U <sub>fm</sub> = 1,1V	1000V/1A	DIODA	1N4007	SMD4007
VD8	U <sub>fm</sub> = 0,35V	10V/3A	DIODA	BAT60J	SOD323_ST
VT1		0,9 W	TRANZISTOR	MCH3333A P-MOSFET SC-70FL-3	
VT2, VT4, VT5, VT7		0,25 W	TRANZISTOR	BC817 NPN	SOT23-BEC
VT3		0,8 W	TRANZISTOR	IRML2502 N-MOSFET SOT23	
VT6		0,25 W	TRANZISTOR	BC807-25 PNP	SOT23-BEC
XC1			KONEKTOR	JST XH 2PIN	
XC2	rozteč 2,54mm		KONEKTOR	1x7pinů dutinková lišta	
XC3			KONEKTOR	BDM	ML6
XC4	rozteč 2,54mm		KONEKTOR	1x6pinů dutinková lišta	
XT1 ÷ XT6	průměr 39 mil		PÁJECÍ BOD		
XT7			JUMPER	2-pin	S1G40
XT8			JUMPER	3-pin	S1G40

## PŘÍLOHA P VII: REALIZOVANÁ OVLÁDACÍ A ZOBRAZOVACÍ JEDNOTKA

