

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

Bakalářská práce

2024

Pavel Tlustoš

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Časomíra
Bakalářská práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Pavel Tlustoš**
Osobní číslo: **I21071**
Studijní program: **B0714A150008 Automatizace**
Zadaná práce: **Časomíra**
Zadávací katedra: **Katedra řízení procesů**

Zásady pro vypracování

Cílem práce je návrh časomíry pro měření časů u sportů se vzdáleným startem od cíle, s vizualizací a ukládání výsledků.

Teoretická část práce provede rozbor problematiky detekce závodníka pro spuštění/zastavení měření času, problematiku bezdrátové komunikace cílového a startovacího modulu měření, přehled dostupných technologií, možnost automatické identifikace závodníka a výběr vhodné platformy pro řešení.

Praktická část pak bude řešit návrh, konstrukci a oživení řešení časomíry pro vybraný typ sportu s bezdrátovou komunikací startovací a cílové brány a automatickou detekcí projetí závodníka. Součástí řešení bude vizualizace, případně uložení výsledku.

Rozsah pracovní zprávy: **30-50**
Rozsah grafických prací: **-**
Forma zpracování bakalářské práce: **Tištěná/elektronická**

Seznam doporučené literatury:

- [1] VANA, V. Mikrokontrolery ATMEL AVR: popis procesoru a instrukční soubor. Praha: BEN technická literatura, 2003. 336 s. ISBN 978-80-7300-083-0.
[2] VANA, V. Mikrokontrolery ATMEL AVR: programování v jazyce C. Praha: BEN technická literatura, 2003. 216 s. ISBN 978-80-7300-102-0.
[3] VLACH, J. Řízení a vizualizace technologických procesů. Praha: BEN technická literatura, 2002. 160 s. ISBN 978-80-86056-66-X.
[4] BRTNÍK, B. Základní elektronické obvody. Praha: BEN technická literatura, 2011. 156s. ISBN 978-80-7300-408-8
[5] RIPKA, P.; TIPEK, A. Master Book of Sensors. Praha: BEN, 2003. ISBN 0-12-752184

Vedoucí bakalářské práce: **Ing. Pavel Rozsival**
Katedra elektrotechniky

Datum zadání bakalářské práce: **15. prosince 2023**
Termín odevzdání bakalářské práce: **10. května 2024**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Daniel Honc, Ph.D. v.r.
vedoucí katedry

Prohlašuji:

Práci s názvem časomíra jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne

Pavel Tlustoš

PODĚKOVÁNÍ

Chtěl bych poděkovat mé rodině a všem přátelům, kteří mi byli oporou při těžkých časech psaní a tvoření mé bakalářské práce. Velké díky patří mému vedoucímu Ing. Pavlu Rozsívalovi, který byl vždy připraven a ochoten mi poradit, či pomoci s problémem, který během mé práce nastal. Závěrem bych chtěl ještě poděkovat všem profesorům, doktorům, inženýrům a bakalářům, kteří mě učili a na jejich předaných vědomostech a poznacích jsem mohl začít tvořit mojí bakalářskou práci.

ANOTACE

Požadavkem práce je navrhnout časomíru pro sportovní účely se vzdáleným startem od cíle, s vizualizací výsledného času a uložení výsledku. Teoretická část bude obsahovat přehled možných řešení použitelných pro zadanou práci. V praktické části bude obsažen způsob řešení vizualizace výsledků, bezdrátová komunikace, ukládání výsledků a automatická detekce projíždějícího závodníka.

KLÍČOVÁ SLOVA

mikroprocesor, časomíra, modulace, senzor, FSK, AVR128DB28, ATMEGA328PB, displej

TITLE

Timer

ANNOTATION

The job requirement is to design a timer for sports purposes with a remote start from finish, with visualization of the resulting time and saving of the results. The theoretical part will contain an overview of possible solutions applicable to the assigned work. The practical part will include method of solving results and automatic detection of a passing competitor.

KEYWORDS

microprocessor, timer, modulation, sensor, FSK, AVR128DB28, ATMEGA328PB, display

OBSAH

Seznam ilustrací.....	9
Seznam tabulek.....	10
Seznam zkratk (terminologie) a značek.....	12
Úvod.....	13
1 Problematika.....	14
1.1 Časomíra.....	14
1.2 Bezdrátová komunikace.....	15
1.2.1 CSS.....	16
1.2.2 FSK.....	16
1.2.3 ASK.....	17
1.2.4 PSK.....	17
1.3 Detekce závodníka.....	18
1.3.1 Infračervený senzor.....	18
1.3.2 Laserový senzor.....	18
1.3.3 Optický senzor.....	18
1.3.4 Koncový senzor.....	18
1.4 Identifikace závodníka.....	19
1.4.1 AIDC.....	19
1.4.2 Manuální identifikace.....	21
1.5 Problematika měření času.....	21
1.6 Zobrazení času.....	21
1.7 Výběr hlavních komponentů.....	22
2 Tvorba časomíry.....	23
2.1 Představa.....	23
2.2 Komponenty.....	25
2.2.1 Modul s LLCC68.....	25
2.2.2 ATMEGA328PB-XMINY.....	25
2.2.3 AVR128DB48-CNANO.....	26
2.2.4 Sensory SS-01GL13-E a BOS016P.....	26
2.2.5 Klávesnice a displej.....	27
2.3 Program.....	28
2.4 Příprava pro tvorbu programů.....	28

2.5	Tvorba programu.....	29
2.5.1	Modul s LLCC68.....	29
2.5.2	SD karta.....	33
2.5.3	Měření času.....	33
2.5.4	Sedmisegmentový led displej.....	36
2.5.5	Klávesnice.....	40
2.5.6	OLED displej.....	43
2.5.7	Konečné programy.....	45
2.6	Schéma časomíry.....	46
2.6.1	Start.....	47
2.6.2	Cíl.....	49
2.6.3	Zobrazovací stanoviště.....	50
3	Konstrukce časomíry.....	52
3.1	Tvorba PCB.....	52
3.1.1	PCB start.....	52
3.1.2	PCB cíl.....	54
3.1.3	PCB zobrazovací stanoviště.....	55
3.2	Časomíra.....	56
	Závěr.....	58
	Použitá literatura.....	59
	Seznam příloh.....	61

SEZNAM ILUSTRACÍ

Obrázek 1 Využití rádiového spektra.....	14
Obrázek 2 Modulace FSK.....	16
Obrázek 3 Modulace ASK.....	17
Obrázek 4 Modulace PSK.....	18
Obrázek 5 Blokové schéma časomíry	19
Obrázek 6 ATMEGA328PB-XMINY.....	21
Obrázek 7 AVR128DB48	22
Obrázek 8 Schéma zapojení prvního řádku klávesnice.....	23
Obrázek 9 Schéma zapojení modulu rádia.....	24
Obrázek 10 Schéma zapojení klávesnice.....	25
Obrázek 11 Schéma zapojení stanoviště start.....	26
Obrázek 12 Zapojení senzoru pro detekci závodníka.....	26
Obrázek 13 Schéma zapojení stanoviště cíl.....	28
Obrázek 14 Zapojení spínání napájení pro segmentové displeje.....	29
Obrázek 15 Napájení stanoviště LED.....	30
Obrázek 16 Schéma zapojení stanoviště LED.....	30
Obrázek 17 PCB start.....	31
Obrázek 18 Osazovací PCB start.....	32
Obrázek 19 PCB cíl.....	34
Obrázek 20 Osazovací PCB cíl.....	37
Obrázek 21 PCB LED.....	37
Obrázek 22 Osazovací PCB LED.....	38
Obrázek 23 Stanoviště start.....	39
Obrázek 24 Stanoviště cíl.....	40
Obrázek 25 Stanoviště LED.....	45

SEZNAM TABULEK

Tabulka 1 Změřené vzdálenosti časomíry.....	28
---	----

SEZNAM ZKRATEK (TERMINOLOGIE) A ZNAČEK

PCB	Printed Circuit Board
FSK	Frequency shift keying
ISM	Industrial, Scientific, Medical
SF	Spreading Factor
SNR	Signal to noise ration
LoRa	Long Range
ASK	Amplitude shift keying
PSK	Phase shift keying
RFID	Radio Frequency Identification
AIDC	Automatic Identification and Data Capture
CSS	Chirp Spread Spectrum
RFID	Radio Frequency Identification
OCR	Optical Character Recognition
SS	Slave Select
CS	Chip select
SCLK	Serial Clock
MISO	Master In Slave Out
MOSI	Master Out Slave In
SRAM	Static Random-Access Memory
SPI	Serial Peripheral Interface
DNF	Did not finish
DIO	Digital input/output
CRC	Cyclic redundancy check
PCB	Printed Circuit Board
USB	Universal Serial Bus
SD	Secure digital
SMD	Surface Mounted Device
EEPROM	Electrically Eraseble Programmable Read Only Memory
UART	Universal asynchronous receiver-transmitter

ACK

acknowledgement

ÚVOD

Cílem mé práce je vytvořit bezdrátovou časomíru. Bezdrátové komunikace vhodné pro tuto práci jsou LoRa, FSK, ZigBee atd. Časomíra bude automaticky detekovat projíždějícího závodníka jak na startu, tak v cíli. Výsledný čas bude následně zobrazen a uložen.

V první kapitole se vysvětluje problematika detekce závodníka pro spuštění/zastavení měření času, problematika bezdrátové komunikace cílového a startovacího modulu měření, přehled dostupných technologií, možnosti automatické identifikace závodníka a výběr vhodné platformy pro řešení.

V druhé kapitole se řeší návrh, konstrukce a oživení časomíry pro sjezdové lyžování. Vybrané řešení bezdrátové komunikace mezi startem, cílem a zobrazovacím modulem. Způsob řešení zobrazování výsledku a uložení výsledku na SD kartu. Řešení detekce startujícího a cílového závodníka. Softwarové řešení komunikace mezi moduly.

Ve třetí kapitole se zabývám řešením konstrukce jednotlivých stanovišť. Nakonec řeším tvorbu PCB jednotlivých stanovišť.

1 PROBLEMATIKA

Sjezdařská časomíra se dnes používá jako nezbytná součást jak závodů, tak i každého tréninku pro hledání té ideální sjezdařské techniky pomocí výsledných časů po projetí tréninkové dráhy. Časomíry musí mít velký dosah a spolehlivost, aby bylo docíleno, že každý závodník bude mít řádně a bezchybně změřený čas a nemusel startovat opakovaně při závodě, což by pravděpodobně znamenalo prohru z důvodů sjetí vosku. Pro tréninky je ideální, aby byla časomíra snadno ovladatelná a především lehce přenosná na požadované pozice bez potřeby napájení ze sítě, což ovšem klade požadavek na bateriové napájení a na kvalitu baterií pro napájení časomíry. Pro tréninky je velice vhodné, aby byla časomíra schopna zaznamenávat aspoň dva závodníky na trati v jednom okamžiku. Sjezdařské časomíry pracují často na velkých vzdálenostech cíle od startu a často bez přímého vidění cíle a startu, to klade požadavky na bezdrátovou komunikaci s velkým dosahem a vysokou odolností proti šumu a rušení.

1.1 Časomíra

Časomíra je zařízení, které počítá (měří) dobu od spuštění až po zastavení. Časomíry mohou být mechanické, elektromechanické nebo softwarové, které spoléhají na přesnost generátoru hodin. U mikroprocesorů je zdrojem hodin krystal nebo RC oscilátor.

1.2 Bezdrátová komunikace

Každá služba má vymezené frekvenční pásmo, ve kterém může komunikovat. Skoro všechny frekvenční pásma jsou licencovaná. Pro volné užití jsou frekvence 433MHz, 868MHz (v Evropě), 915MHz (v Americe), 2,4GHz a 5,8GHz. Tato bezlicenční pásma jsou označována jako ISM pásmo (volná pásma rádiového vysílání). Nevýhodou vyšších frekvencí je, že se stoupající frekvencí klesá dosah vzdáleností, ale roste přenosová rychlost. Důvodem, proč vyšší frekvence mají nižší dosah, jsou rozptýlení signálu, absorpce rádiových vln, vyšší ovlivnění atmosférickými jevy atd. Tudíž nižší frekvence jsou vhodné pro nižší požadavky na rychlost na větší vzdálenosti s vyšším výskytem rušivých vlivů. Zato vyšší frekvence jsou vhodné pro rychlé komunikace na krátké vzdálenosti s menším výskytem rušivých vlivů. Dalším vlivem, co ovlivňuje rádiovou komunikaci, je použitý typ modulace například LoRa, FSK, GFSK atd. Níže je uveden obrázek použití frekvencí pro různé účely.



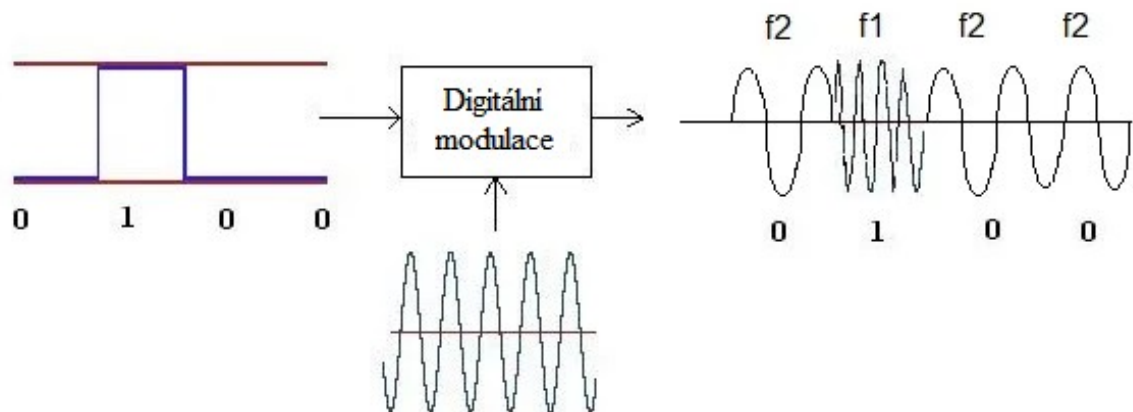
Obrázek 1 Využití rádiového spektra

1.2.1 CSS

CSS využívá frekvence 433MHz, 868MHz a 915MHz, tedy pásma ISM. Tato modulace umožňuje obousměrně komunikovat a díky tomu má nízkou hladinu šumu. Rychlost a dosah jde u CSS ovlivnit pomocí SF (faktor šíření). CSS má velký dosah až desítky kilometrů v přímé viditelnosti a má malou spotřebu energie. Nevýhodou CSS je malá šířka pásma, malá přenosová rychlost, problém pro použití v aplikacích pracujících v reálném čase a kvůli většímu využívání CSS může být rušena okolními zařízeními.

1.2.2 FSK

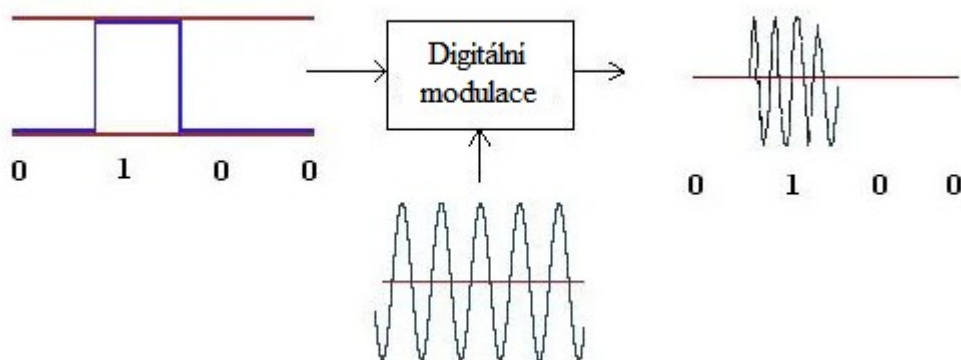
FSK patří mezi digitální modulace a pracuje na principu posouvání frekvence nosné vlny. Tato modulace využívá frekvence k reprezentování různých hodnot. Binární FSK (tedy BFSK) využívá dvou frekvencí k reprezentování binární 1 a 0. Dosah FSK je několik kilometrů v přímé viditelnosti. Má nižší pravděpodobnost chyb a dobrý poměr signálu k šumu(SNR). Aplikace FSK pro nízké rychlosti je velice snadná. Má nižší dosah a využívá velkou šířku pásma.



Obrázek 2 Modulace FSK

1.2.3 ASK

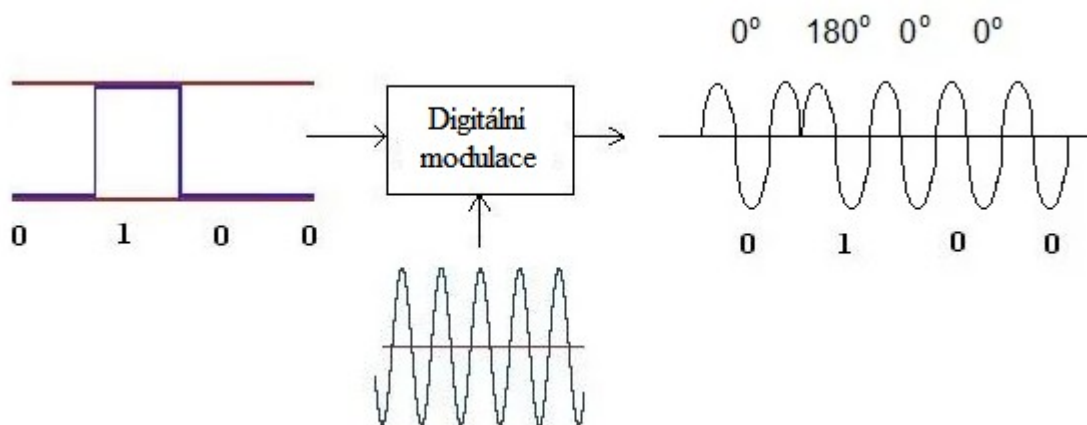
ASK modulace patří mezi digitální modulace a pracuje na principu změny amplitudy. Různé hodnoty výšky amplitudy reprezentují různou logickou hodnotu. Fáze signálu a frekvence signálu se v této modulaci nemění. Modulace ASK je velice jednoduchá a levná. Je vhodná pro aplikace, kde máme omezenou šířku pásma. Nevýhodou ASK je velká náchylnost k rušení a šumu. Má malou datovou rychlost.



Obrázek 3 Modulace ASK

1.2.4 PSK

PSK je digitální modulace využívající fázový posun. Logická hodnota je vyjadřována fázovým posunem, například jedna je reprezentována posunem o 180° a nula o 0° . U této modulace se nemění frekvence ani amplituda. Má vysokou přenosovou rychlost. Je málo odolná k chybám. Má dosah od stovek metrů až po kilometry. Je to forma FSK modulace, proto má také horší využití šířky pásma.



Obrázek 4 Modulace PSK

1.3 Detekce závodníka

Aby byl závodník vždy detekován na startu a v cíli, musí být vybrány spolehlivé a zároveň odolné senzory. Pro start jsou vhodné mechanické senzory jako koncový spínač, aby mohl závodník vědět, kdy přesně odstartuje. Při použití bezdotykového senzoru by mohlo dojít k neúmyslnému spuštění časomíry. Pro cíl či brány pro měření mezičasu jsou vhodné dvoupolohové (binární) optické senzory s vysokou reakční frekvencí, aby byl schopen zachytit spolehlivě projíždějícího závodníka.

1.3.1 Infračervený senzor

Infračervený senzor využívá infračervené vlny. Využívá se k detekci pohybu, v průmyslu jako komunikační rozhraní IrDA nebo pro snímání teploty. Snímače polohy mají dosah až 30 metrů. Mají malou spotřebu energie, tudíž jsou vhodné pro aplikace, kde je bateriové napájení. Infračervený senzor je spolehlivý jak ve dne, tak v noci. Při vysokých výkonech může poškodit zrak. Proto v případě oblastí s výskytem lidí je vhodný pouze při malých výkonech. Je vhodné senzor používat pro menší vzdálenosti, protože pro velké vzdálenosti potřebuje velký výkon a je méně efektivní.

1.3.2 Laserový senzor

Laser je zesílené světlo stimulované emisí záření. Všechny částice světla se pohybují v jednom směru a fázi, díky tomu se paprsek nerozprostře jako třeba světlo z lampy a nezviditelňuje nerovnosti na povrchu například na stěně. To znamená, že jde o velmi intenzivní paprsek světla, který lze použít od snímačů nebo laserových ukazovátek, až po lasery řezající různé materiály od dřeva po kovy. Laser má velmi vysoký dosah, může dosahovat až kilometry. Je velmi přesný a odolný proti rušení. Ovšem laser je velice škodlivý pro lidské oko, i při malých výkonech dochází ke zhoršení zraku. Proto je nevhodný pro oblasti, kde se vyskytují lidé.

1.3.3 Optický senzor

Optický senzor převádí světlo na elektrický signál. Při použití optických snímačů se měří změna signálu místo intenzity, protože při nastavení a po nastavení se může intenzita změnit, třeba změnou žárovky s nižší či vyšší luminiscencí, a pak by senzor už nefungoval správně. Optické senzory mají vysokou citlivost, jsou jednoduché a spolehlivé. Jsou však citlivé na rušení vlivem okolního prostředí.

1.3.4 Koncový senzor

Koncový senzor je mechanický snímač, který má dva stavy logickou jedna a logickou nula. Koncový spínač má dva vstupy NO (v klidu otevřený) a NC (v klidu sepnutý). Často se používají k detekci dorazu či polohy objektu. Je jednoduchý na aplikaci a má velkou opakovatelnost. Je spolehlivý a má nízkou spotřebu elektrické energie. Nevýhodou je opotřebení mechanické části a také musí mít přímý kontakt s objektem.

1.4 Identifikace závodníka

Identifikace závodníka je velice důležitá pro následné přiřazení a uložení výsledného času k identifikačnímu číslu závodníka. Proto je důležité zvolit správnou optimální volbu identifikace závodníka. Může být identifikace automatická (AIDC) nebo manuální. Mezi AIDC patří čárový kód, RFID, OCR, magnetické proužky, chytré karty a hlasové rozpoznání. Mechanická identifikace využívá mechanické rozhraní, pomocí kterého se lze identifikovat, například pomocí klávesnice.

1.4.1 AIDC

AIDC je proces, při kterém jsou identifikována a přečtena data, následně jsou uložena a poté zpracována počítačem nebo mikroprocesorem. Výhoda AIDC je velmi rychlé zpracování dat a pro uživatele velmi jednoduché řešení.

Čárový kód

Čárový kód patří mezi nejznámější AIDC. Setkáváme se s ním prakticky ve všech odvětvích, například v obchodech a výrobních procesech. Čárový kód se skládá z černých čar a bílých mezer různých tloušťek reprezentující logické jedničky a nuly. Tyto jedničky a nuly reprezentují informaci v binární formě o produktu či objektu. Kód se čte pomocí čtečky čárových kódů, který předává získané informace dál, například počítači.

RFID – radiofrekvenční identifikace

RFID se využívá, pokud chceme znát informace o objektu v reálném čase třeba o poloze, kde se aktuálně nachází produkt ve skladu. RFID pracuje na principu, že objekt má RFID tagy obsahující vysílač/přijímač a čip s unikátními identifikačními daty. RFID tagy komunikují s RFID čtečkami a předávají jim data.

OCR – optické rozpoznávání znaků

OCR funguje na rozpoznávání textů nebo obrázků a z nich získává data. Tato AIDC se používá v procesech digitalizace. Nejprve v OCR dojde ke skenování textu nebo obrázku a zároveň k potlačení šumu a rušení. Poté jsou rozpoznávány segmentové znaky a přiřazovány odpovídajícím znakům v abecedě.

Magnetické proužky

Magnetické proužky jsou proužky, v kterých je uložena informace pomocí proužků s různou polaritou. Tyto polarizace reprezentují logické jedničky a nuly. Tato data z magnetického proužku přečte magnetická čtečka a dále zpracuje. Tento způsob má nevýhodu v slabém zabezpečení a hrozbě snadného duplikování magnetického proužku. Tato nevýhoda je zásadním problémem kreditních karet, které jdou snadno duplikovat.

Čipové karty

Čipové karty jsou formou pokročilé technologie magnetických pásků. Zde je umístěn integrovaný čip, z kterého je možné číst. Tato metoda má vyšší zabezpečení než magnetické proužky a používá se v moderních platebních kartách.

Hlasové rozpoznání

Funguje na principu, že je zaznamenán hlas pomocí mikrofону, který je převeden na elektrický signál, který je filtrován, aby se zbavil šumu a nadbytečných informací. Zvuk je poté analyzován a jsou identifikovány klíčové rysy řeči. Poté se vytvoří model řeči, který slouží k dekodování a rozpoznávání řeči. Výsledný rozpoznávaný zvuk je dále zpracován podle dané aplikace. Hlasové rozpoznání se používá třeba u hlasových asistentů v chytrých telefonech.

1.4.2 Manuální identifikace

Manuální identifikace je naproti AIDC méně efektivní kvůli faktorům, jako je lidská chyba či pomalejší identifikace. Na druhou stranu je jednoduchá a k ovládání nepotřebujeme kartu, ani jiné identifikační zařízení jako u většiny AIDC. K manuální identifikaci můžeme použít třeba klávesnici nebo dotykový displej. Přes vstupní periférii jsou vkládána data, která jsou poté zpracovávána počítačem či mikroprocesorem.

1.5 Problematika měření času

Měření času je velice obtížný úkon kvůli mnoha vlivům, které ovlivňují měření. Při použití mikroprocesoru, který využívá oscilátor pro zdroj hodin oscilátor RC či krystal, je nutné počítat s tím, že frekvence oscilátoru se bude měnit s měnící se teplotou. To je závažný problém při měření reálného času. Pokud bychom nastavili úspěšně mikroprocesor na generátor reálných hodin při teplotě 25°C, tak při změně teploty by se čas začal odchylovat od reálného času. Tento problém můžeme vyřešit měřením teploty a pomocí změřené okolní teploty lze kompenzovat vzniklou chybu okolí. Výrobci mikroprocesorů, například Microchip Technology, mají na stránkách, jak tyto vlivy kompenzovat.

U časomíry s rádiovou komunikací také hraje vliv na výsledný čas zpoždění signálu. Také opakované odeslání zprávy o tom, že závodník už vyrazil ze startu z důvodu znehodnocené první zprávy z různých důvodů jako šumu a rušení. A volba spuštění a zastavení měření času, jestli se časomíra bude spouštět a zastavovat na náběžnou či sestupnou hranu signálu.

1.6 Zobrazení času

Pro zobrazení výsledného času je potřeba zobrazovací zařízení, které zobrazí čas čitelně a jasně i na větší vzdálenosti. Pro tyto účely se používají sedmsegmentové displeje, maticové displeje nebo LCD displeje. Při použití displeje s LED diodami nebo pixely lze budít jednotlivé LED diody a pixely staticky či dynamicky.

Statické buzení spočívá v tom, že LED dioda nebo pixel je po celou dobu, kdy má být rozsvícen, buzen napětím. Tento způsob buzení má mnoho nevýhod proti dynamickému buzení. Statické buzení je náročnější na spotřebu energie, a tudíž není vhodné pro aplikace, kde potřebujeme nízkou spotřebu energie. Hardware je složitý kvůli tomu, že na každou LED diodu nebo pixel musí být vyhrazen zvlášť port mikroprocesoru. Na druhou stranu je statické buzení jednoduché na ovládání a má stabilní obraz na rozdíl od dynamického buzení.

Dynamické buzení spočívá v tom, že jednotlivé LED diody nebo pixely jsou střídavě buzeny a tedy pro lidské oko svítí stejně jako při statickém buzení, ale ve skutečnosti jsou krátký čas zapnuty a po zbytek vypnuty. Dynamické buzení využívá nedokonalost lidského oka, které je schopné postřehnout 25 snímků za sekundu. Dynamické buzení je jednodušší na hardware a má nižší spotřebu, tudíž je vhodné pro nízkoenergetické aplikace. Nevýhodou dynamického buzení je, že vyžaduje vyšší nároky na řídicí algoritmus a hrozí problíkávání displeje při zvolení špatné frekvence rozsvěcování a zhasínání LED diod nebo pixelů. Také může v některých případech snížit životnost displeje.

1.7 Výběr hlavních komponentů

Jako řídicí jednotky pro časomíru byly zvoleny dva vývojové kity AVR128DB48 Curiosity Nano a jeden vývojový kit ATMEGA328PB – XMINI. Vývojový kit ATMEGA328PB – XMINI byl vybrán, protože mikroprocesor je téměř kompatibilní s mikroprocesorem ATMEGA328P, který je na deskách Arduino. Tato možnost byla zvolena, protože pro ukládání výsledků byla zvolena SD karta a díky knihovnám Arduino lze snadno inicializovat a zapisovat hodnoty na SD kartu pomocí rozhraní SPI.

Jako rádiový modul byla zvolena LAMBDA A68-8D využívající digitální modulaci FSK. Tento modul má dobrý poměr výkonu k ceně.

Pro detekci závodníka byl zvolen pro start mechanický koncový mikrospínač SS-01GL13-E a pro cíl reflexní optická závora BOS016P.

Pro zobrazení času bylo zvoleno pět sedmsegmentových displejů MP005916.

Pro inicializaci závodníka byla zvolena klávesnice, tedy manuální způsob inicializace s OLED displejem.

2 TVORBA ČASOMÍRY

K vytvoření funkční časomíry je zapotřebí dvou či více stanovišť, které spolu budou automaticky komunikovat.

2.1 Představa

Pro komunikaci mezi stanovišti byla zvolena frekvence 868MHz, protože tato frekvence je bezlicenční. Modulace byla vybrána FSK, protože má nízký výskyt chyb a i přes její menší dosah pro účely časomíry, kde se maximální vzdálenosti budou pohybovat okolo jednoho kilometru, je dostatečná. Také aplikace FSK je pro účely řídké komunikace jednoduchá.

Pro detekci závodníka byl zvolen na startu mechanický koncový mikropínač, protože je velmi jednoduchý a spolehlivý. K detekci závodníka v cíli byla zvolena optická závora s infračerveným světlem s malým bezpečným výkonem, aby při provozu nedošlo k újmě na zraku. Ze stejného důvodu byla vyřazena laserová závora, která by také měla neblahý účinek na zrak okolních osob. Optická závora s infračerveným světlem má výhodu, že funguje bez problému jak ve dne, tak v noci. Infračervené snímače mají sice malý dosah, ale pro účely použití časomíry jsou dostatečné.

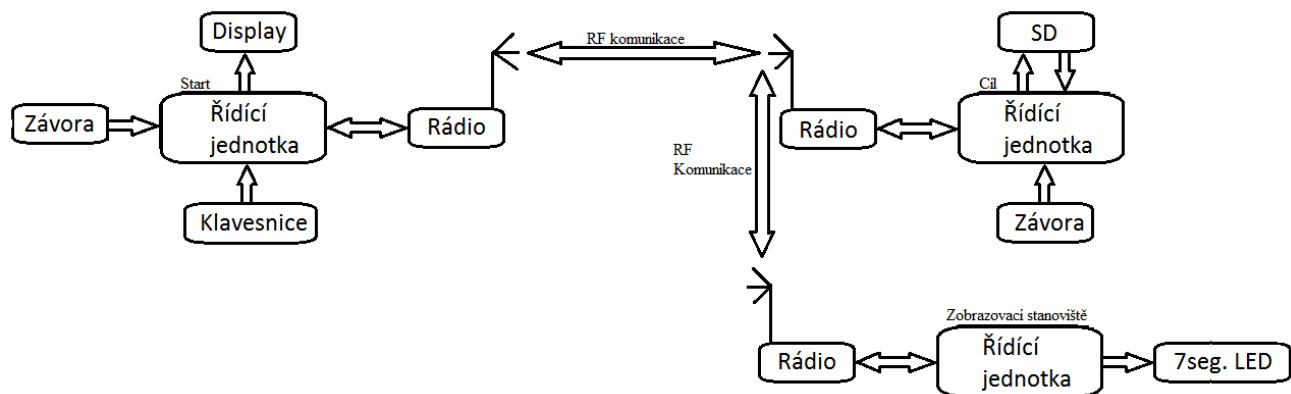
Jako způsob ukládání dat bylo zvoleno uložení dat na SD kartu. Tento způsob byl zvolen jak kvůli jeho jednoduchosti, tak kvůli podpoře knihoven Arduino pro SD kartu. Díky tomu je práce s SD kartou velice jednoduchá a rychlá.

Pro zobrazení výsledného času závodníka bylo zvoleno zobrazování pomocí sedmi-segmentového led displeje. Buzení jednotlivých led displejů bylo provedeno pomocí dynamického buzení led diod. Tento způsob byl zvolen kvůli úspoře energie a úspoře pinů mikroprocesoru.

Pro inicializaci závodníka byla nakonec zvolena manuální inicializace pomocí klávesnice místo automatické inicializace závodníka. Automatická inicializace závodníka nebyla vybrána, protože by k časomíře byly zapotřebí další komponenty k provozu časomíry a časomíra by byla pouze pro osoby s identifikačním prostředkem. Proto byla zvolena manuální inicializace závodníka a díky tomu může užívat časomíru každý závodník či amatér, který si to chce jen příležitostně zkusit a mít při tom uložený výsledný čas.

Pro zobrazování informací, ať už o inicializaci, počtu závodníků na trati nebo informacích o stavu časomíry, byl zvolen OLED displej. Byl zvolen kvůli tomu, že má lepší viditelnost než třeba LCD displej. OLED displej má také jednoduché ovládání, které je stejné v podstatě u všech LCD a OLED displejů. Díky tomu je mnoho programů a řešení, jak pracovat s těmito displeji.

Pro začátek bylo vytvořeno blokové schéma, aby byla vytvořena představa o tom, jak má projekt vypadat.



Obrázek 5 Blokové schéma časomíry

2.2 Komponenty

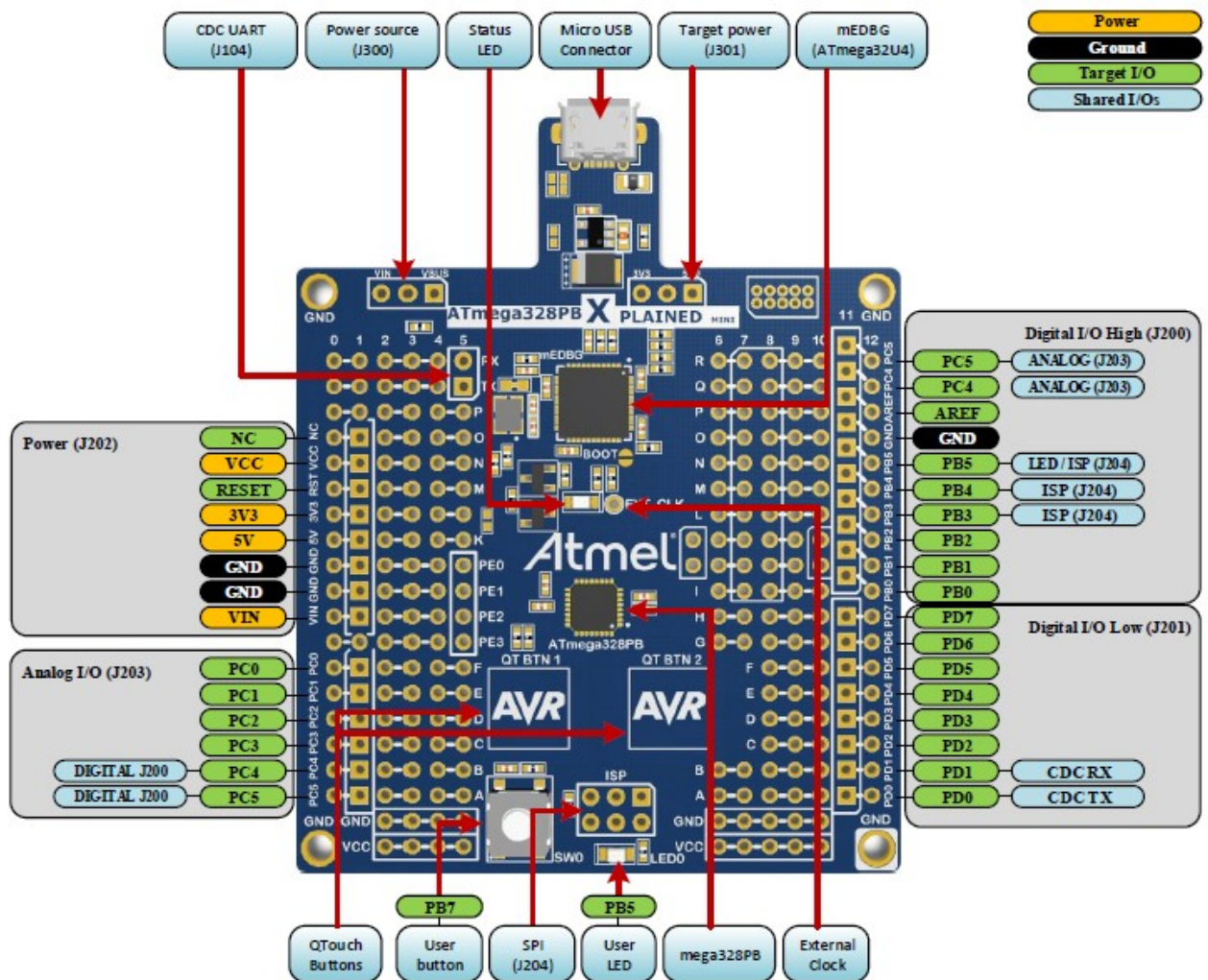
Aby byla časomíra funkční, musely být vybrány komponenty schopné fungovat i v teplotách pod bodem mrazu a zároveň byly úsporné jak na energii, tak kvůli výběru bateriového napájení a úspoře financí.

2.2.1 Modul s LLCC68

Pro komunikaci byl zvolen modul s čipem LLCC68 od firmy SEMTECH. Čip LLCC68 je nástupcem starších čipů jako SX1262. LLCC68 má maximální dosah dvacet kilometrů při použití se správnou anténou. Má nízký odběrový proud při příjmu 4,6mA. Při vysílání se ziskem 22dB má čip odběr 118mA. Klidový odběr je 70 μ A.

2.2.2 ATMEGA328PB-XMINY

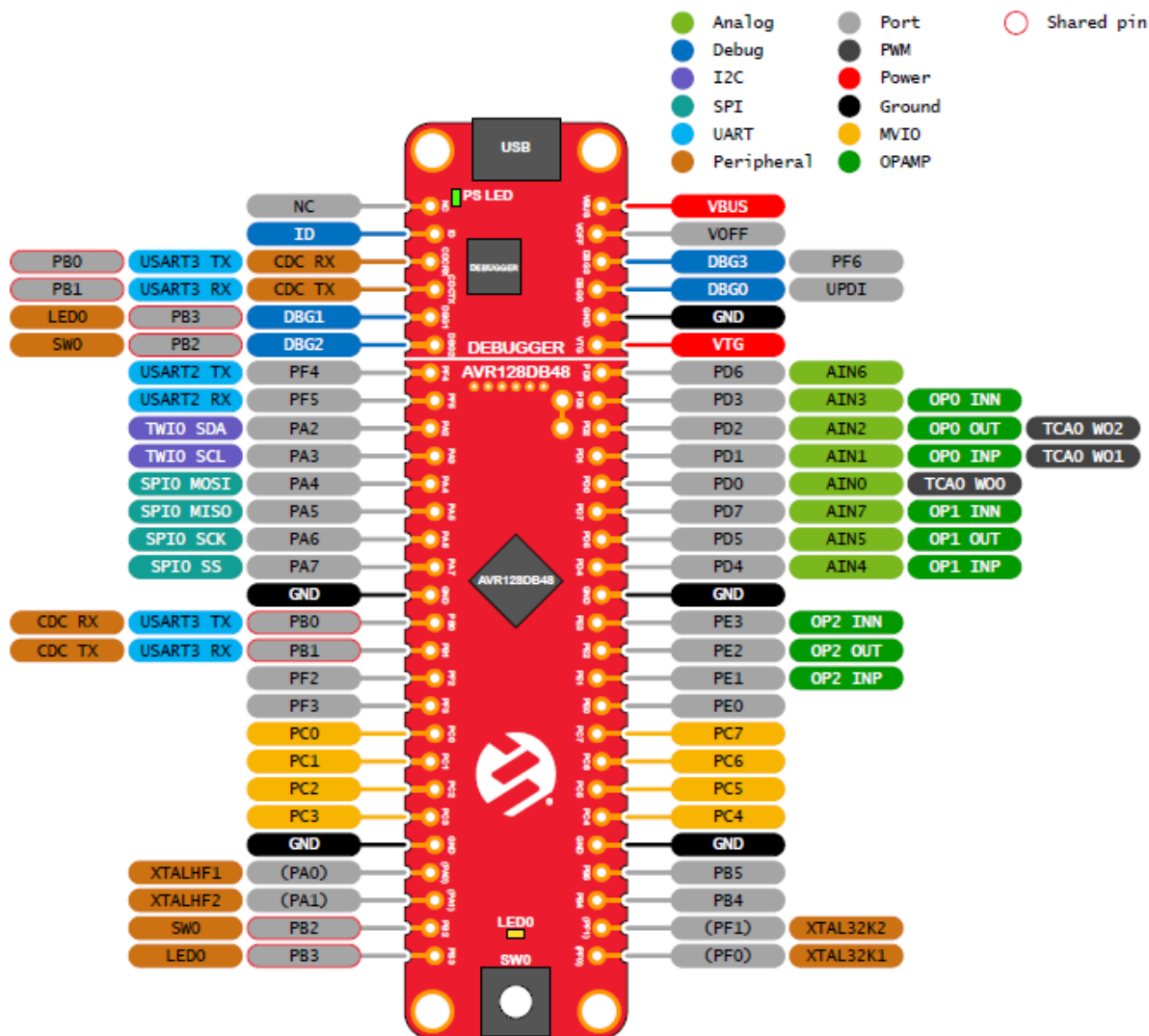
Jako řídicí jednotku pro stanoviště cíl byl zvolen vývojový kit s čipem ATMEGA328PB, kvůli jeho nízké ceně a podobě k čipu ATMEGA328P, který využívají desky Arduino. Po drobných úpravách v knihovnách lze použít knihovny i pro ATMEGA328PB. Tento mikroprocesor využívá jádro AVR a má programovou paměť 32KB, paměť EEPROM 1KB a paměť SRAM 2KB. ATMEGA328PB je čip od výrobce mikroprocesorů Microchip Technology. ATMEGA328PB má maximální frekvenci až 20MHz. Tento vývojový kit se programuje pomocí micro USB.



Obrázek 6 ATMEGA328PB-XMINY

2.2.3 AVR128DB48-CNANO

Pro stanoviště cíl a zobrazovací stanoviště byla vybrána jako řídicí jednotka vývojový kit s čipem AVR128DB48. Tento mikroprocesor využívá jádro AVR a má programovou paměť 128KB, paměť EEPROM 512B a paměť SRAM 16KB. AVR128DB48 je čip od firmy Microchip Technology. Programová paměť čipu je 128KB. AVR128DB48 má maximální frekvenci až 24MHz. Tento vývojový kit se také programuje pomocí micro USB.



Obrázek 7 AVR128DB48

2.2.4 Senzory SS-01GL13-E a BOS016P

Pro stanoviště cíl byla vybrána infračervená optická závora BOS016P od firmy BALLUFF, kvůli jejímu bezpečnému provozu pro osoby v prostředí s časomírou. Tento senzor má dosah až 14 metrů a provozní napětí 10-30V.

Pro stanoviště start byl zvolen mechanický mikrospínač s kontakty NC a NO. Tento mikrospínač je od firmy Omron.

2.2.5 Klávesnice a displej

Inicializace závodníka se bude provádět na stanovišti start pomocí klávesnice od firmy Storm. Pro zobrazování informací byl vybrán OLED displej od firmy MIDAS.

2.3 Program

Pro zprovoznění a spolehlivou funkci časoměry musí mít mikroprocesory vytvořené softwarové algoritmy. Programy byly vytvořeny v prostředí Atmel Studio 7.0, které podporuje programování mikroprocesorů jak od firmy ATMEL, tak i od firmy Microchip Technology, která vlastní ATMEL.

Mikroprocesory byly naprogramovány v jazyce C+. Než byly vytvořeny konečné programy pro jednotlivá stanoviště, tak bylo napsáno mnoho malých programů.

Tyto malé programy byly vytvořeny pro jednotlivé úkony jako ovládání rádiového modulu pomocí rozhraní SPI a pomocných I/O pinů, dynamické ovládání sedmissegmentových led displejů, ovládání SD karty pomocí SPI atd.

2.4 Příprava pro tvorbu programu

Než se mohly začít vytvářet menší programy pro hlavní programy, tak bylo nutné nastudovat z data sheet funkčnost jednotlivých komponentů a rozhraní, přes které se ovládají. V práci je využíváno SPI k programování SD karty a modulu s čipem LLCC68. Pro komunikaci s OLED displejem se využívá paralelní osmibitová komunikace s pomocnými I/O piny jako RS, E, V0 a R/W. K ovládání tlačítek bylo zvoleno postupné čtení řádků. Pro zobrazení výsledného času bylo vybráno dynamické zobrazování a díky tomu bylo zabráno pro pět sedmissegmentových led displejů pouze třináct portů mikroprocesoru místo čtyřiceti pěti při použití statického zobrazení.

SPI sběrnice má čtyři vodiče (SS/SC, SCLK, MOSI, MISO) a jedná se o sériovou sběrnici. Vodič pro výběr zařízení je SS nebo SC. Vodič pro generování hodin je SCLK. MISO a MOSI jsou datové vodiče. Díky dvěma datovým vodičům se jedná o duplexní komunikaci, a proto lze v jednom okamžiku vysílat i přijímat data. Tato sběrnice funguje na principu detekce náběžných a sestupných hran, na rozdíl třeba od sběrnice TWI (I2C), která pracuje na principu velikosti napětí a je také poloduplexní.

MISO – Pokud je zařízení v režimu master, tak tento vodič slouží pro příjem dat.
Pokud je zařízení v režimu slave, tak tento vodič slouží pro odesílání dat.

MOSI – Pokud je zařízení v režimu master, tak tento vodič slouží pro odesílání dat.
Pokud je zařízení v režimu slave, tak tento vodič slouží pro příjem dat.

Při použití osmibitové paralelní datové sběrnice se vysílají jednotlivé bity najednou. Proto bylo rozhodnuto vyhradit celý osmibitový port mikroprocesoru pro komunikaci s displejem. Díky tomu můžu rychle a jednoduše zapisovat žádanou hodnotu rovnou na výstup portu. Přídavné tři piny slouží k různým účelům.

R/W – Slouží k možnosti zapsání či čtení dat.

RS – Slouží k možnosti komunikovat s daty či instrukcemi.

E – Slouží k začátku zápisu nebo čtení dat/instrukcí.

Modul s čipem LLCC68 má rozhraní SPI s dalšími pomocnými I/O piny.

2.5 Tvorba programu

Před vytvořením hlavního programu bylo vytvořeno několik menších programů, které byly poté použity v konečných programech.

2.5.1 Modul s LLCC68

Před zahájením tvorby programu bylo nutné nastudovat data sheet od firmy SEMTECH pro čip LLCC68 a s využitím kódu od firmy RF solutions se začalo s ožívováním rádiového čipu.

Čip LLCC68 využívá komunikační rozhraní SPI spolu s pomocnými I/O piny.

RESET – Tento pin slouží k resetování čipu při nízké hladině napětí. Při vysoké hladině napětí čip není resetován.

BUSSY – Tento pin signalizuje zaneprázdnění čipu. Dokud je BUSSY na vysoké hladině napětí, tak je čip v zaneprázdněném stavu a není možné posílat další data ke zpracování. Jakmile je BUSSY nízko, čip čeká na nová data.

RXTEN – Tento pin slouží k tomu, aby se přepnul do přijímacího režimu a hodnota na tomto pinu musí být negovaná hodnota z výstupu pinu TXTEN.

TXTEN – Tento pin udává, že je při vysoké hladině napětí čip ve vysílajícím módu.

DIOx – Piny DIO1, DIO2 a DIO3 jsou konfigurační piny. Všechny mohou být použity ke generování přerušení. Piny DIO2 a DIO3 mají ovšem volitelné další funkce pro provoz čipu na rozdíl od pinu DIO1. DIO2 může fungovat jako RF spínač a řídit tak, jestli bude čip vysílat, či bude v jiném režimu. Pokud je DIO2 na vysoké úrovni napětí, tak je čip ve vysílacím režimu a pokud je na nízké hladině napětí, tak je v jiném režimu. Pomocí pinu DIO3 můžeme nakonfigurovat velikost napájení pro TCXO.

Kvůli využití SPI rozhraní a rozhodnutí nepoužít knihovny od Arduina pro čipy AVR128DB48 a použít je pouze pro čip ATMEGA328PB kvůli komunikaci s SD kartou, tak ze znalostí SPI rozhraní a kódu od RF solutions, byla upravena funkce pro AVR128DB48 a druhá byla vytvořena.

Funkce transmit byla upravena pro AVR128DB48. Jedná se o funkci typu void. Tedy o funkci, která nevrací hodnotu. Funkce má vstupní data Array a Size. Array jsou data, která se mají odeslat a Size je kolik osmibitových dat se má poslat.

- 1) První je ve funkci busy. Tato funkce čeká, dokud je čip zaneprázdněný. (Tato funkce bude uvedena dále.)
- 2) Dále se nastaví pin SS na nízkou úroveň a tím se vybere čip LLCC68 pro komunikaci s čipem AVR128DB48.
- 3) Poté je cyklus for, který předává data, která se mají odeslat funkci spi_komunikace, která odesílá data čipu LLCC68.
- 4) Po skončení odeslání program chvíli čeká a poté nastaví pin SS vysoko, čímž ukončí komunikaci s čipem LLCC68.

```
void transmit(uint8_t Array[], int Size) {
    busy();
    PORTA.OUTCLR=ss;
    for (int i = 0; i < Size; i++) {
        spi_komunikace(Array[i]);
    }
    _delay_ms(10);
    PORTA.OUTSET=ss;
}
```

Funkce spi_komunikace slouží k odesílání a přijímání dat. Funkce má návratovou osmibitovou proměnnou. Jako vstup do funkce jsou osmibitová data.

- 1) V prvním kroku se odešlou data.
- 2) Poté je vykonána instrukce NOP, která vytváří zpoždění jeden takt.
- 3) Dále se čeká, dokud není přenos ukončen.
- 4) Po ukončení přenosu funkce vrátí osmibitová přijatá data.

```
uint8_t spi_komunikace(uint8_t data){
    SPI0.DATA = data;
    asm volatile("nop");
    while (!(SPI0.INTFLAGS & (0x80))) ;
    return SPI0.DATA;
}
```

Funkce busy čeká, dokud je čip LLCC68 zaneprázdněný.

```
void busy() {
    while (PORTA.IN & (bussy)){
    }
}
```

Funkce Status je podobná funkci transmit a slouží ke zjištění, v jakém stavu je čip LLCC68. Nejprve se pošle adresa z jaké chceme číst, tedy šestnáctkově 0xC0, a poté je odeslána nula a jsou vrácena přijatá data z funkce spi_komunikace a následně uloženy do globální proměnné byte1.

```
void Status(){
    Size = sizeof(get_status);
    transmit(get_status, Size);
    PORTA.OUTCLR=ss;
    byte1 = spi_komunikace(0x00);
    PORTA.OUTSET=ss;
}
```

Po zhotovení a upravení funkcí byly upraveny ostatní funkce a přidány funkce Tx a Rx. Tyto funkce provádějí přepínání režimu vysílání či jiného režimu pomocí pinů DIO2, TXTEN a RXTEN. Tx nakonec odešle data. Rx na konci funkce čeká na příjem dat. Upravené funkce od RF solutions jsou setup, lora_setup, TX_Setup, TX_Send, RX_Setup, RX_read a RX_wait.

Setup - Ve funkci setup nastavuji piny, jestli jsou vstupní či výstupní.

- 1) Ve funkci lora_setup nejprve provedu resetování čipu LLCC68 za dodržení potřebných časových intervalů pro dodržení úspěšného resetování čipu.
- 2) V prvním kroku odešlu nastavení zdroje frekvence jako krystal.
- 3) Poté čtu kontrolní stav čipu LLCC68.
- 4) Dále nastavím modulaci na FSK modulaci na typ paketu GFSK.
- 5) V dalším kroku nastavuji parametry GFSK jako přenosovou rychlost, filtr, šířku pásma a frekvenční odchylku.
- 6) Poté se nastaví parametry paketu GFSK jako preamble, délka detekce preamble, délka synchronizačního slova, filtr adres, jestli zná délku paketu přijímač i vysílač, nebo jestli jsou délky paketu proměnné, velikost maximální přijatelné informace, zabezpečení polynomem CRC a kódování.
- 7) Dále nastavím zvýšený zisk přijímače.
- 8) Potom nastavím frekvenci rádia.
- 9) Poté se provede kalibrace pro frekvence ISM v rozsahu 863-870MHz.
- 10) Nastavím vysílací zisk na maximum (tedy na 22dBm).
- 11) Nakonec nastavím DIO2 na nastavení režimů RF.

TX_Setup - Ve funkci TX_Setup nastavuji čip LLCC68 na vysílač a jeho parametry při vysílání.

- 1) V prvním kroku zde nastavuji vysílací výkon.
- 2) Poté nastavím provozní režim pa s optimálním nastavením.
- 3) Pak nastavím přerušení na DIO1.
- 4) Dále vymažu všechny chyby detekované zařízením.
- 5) Vymažu také všechny přerušení pomocí DIO pinů.
- 6) Nastavím adresu přijímače a vysílače.
- 7) Zapišu data do registru a nastavím bajt synchronizačního slova.
- 8) Na konec nastavím přerušení na DIO1.

TX_Send – Ve funkci TX_Send posílám data čipu LLCC68, která se mají odeslat.

- 1) Pošlu data, která se mají odeslat čipu LLCC68.
- 2) Nastavím, že čip zůstane v odesílacím režimu po dobu, dokud se neodešlou data.
- 3) A na konec zkontroluji stav čipu LLCC68.

RX_Setup – Ve funkci RX_Setup nastavuji čip LLCC68 do přijímacího režimu.

- 1) Vymažu všechny chyby detekované zařízením.
- 2) Vymažu také všechny přerušení pomocí DIO pinů.
- 3) Nastavím adresu přijímače a vysílače.
- 4) Zapišu data do registru a nastavím bajt synchronizačního slova.
- 5) Nastavím zvýšený přijímací zisk.
- 6) Nastavím přerušení na DIO1.
- 7) Nastavím, že je čip v přijímacím režimu, dokud nepřijme data od vysílače.

RX_Read – Funkce RX_Read slouží k přečtení přijatých dat. Tato funkce využívá funkci spi_komunikace k uložení dat do globální proměnné byte1.

RX_Wait – Funkce RX_Wait slouží k tomu, že se čeká, dokud pin DIO1 nevygeneruje přerušení, že byla přijata data od vysílače.

2.5.2 SD karta

Pro práci s SD kartou byla zvolena knihovna Arduino, která ulehčila práci s SD kartou. Tato knihovna také využívá knihovnu od Arduina pro komunikační rozhraní SPI, které SD karta využívá. Pro ovládání SD karty byly vytvořeny dvě funkce využívající knihovnu SD a SPI od Arduina. Knihovna pro SD kartu je SD. Knihovna pro SPI se jmenuje SPI.

SD_setup – Tato první funkce provádí inicializaci SD karty.

- 1) Nejprve provedu inicializaci karty a vyberu pin mikroprocesoru pro SS.
- 2) Pokud se inicializace zdařila, uložím do globální proměnné stav_sd jedničku.
- 3) Poté provedu nízkoúrovňové operace s SD kartou pro nastavení rychlosti SPI.
- 4) Pokud se nastavení zdařilo, je uložena do stav_sd jednička.
- 5) Poté se provede detekce typu SD_karty.

SD_zapis – Funkce SD_zapis slouží k zápisu dat na SD kartu.

- 1) Jako první otevřu textový dokument pro zápis dat.
- 2) Pokud se soubor úspěšně otevřel, pak se do sd_stav uloží jedna, pokud se soubor neotevřel, pak se do sd_stav uloží nula.
- 3) Jestliže se soubor otevřel, pak se do něj zapíše požadovaná data.
- 4) Po zápisu se soubor zavře.

2.5.3 Měření času

Pro měření času byl zvolen šestnáctibitový časovač v ATMEGA328PB. Čas byl měřen pomocí přerušení, ve kterém se přičítají hodnoty k proměnným `set0`, `sek0` a `min0`. Do inicializační části byla vložena i inicializace externího přerušení z optické závory, která detekuje závodníka a tím zastaví měření času. Pro toto byla napsána inicializační funkce a dva podprogramy pro přerušení.

setup_casovac_preruseni – Funkce `setup_casovac_preruseni` provádí inicializaci externího přerušení a inicializaci šestnáctibitového časovače.

- 1) Nejprve zapnu pull-up rezistor na portu D.
- 2) Poté nastavím, že se přerušení vykoná vždy, kdy je změna logické hodnoty.
- 3) Poté zapnu přerušení pro INT0.
- 4) Povolím změny v předděličce hodin.
- 5) Nastavím předděličku hodin na osm, čímž se bude zdroj hodin dělit osmi.
- 6) Poté nastavím hodnotu do registru OCR1A. Tato hodnota se porovnává s obsahem časovače. Pokud je shoda hodnot, tak se vygeneruje v mém případě přerušení.
- 7) Následně toto přerušení při shodě povolím.
- 8) Nastavím, že se provede přerušení při maximální hodnotě časovače.
- 9) Nastavení hodin bez předděličky.
- 10) Nakonec povolím globální přerušení.

```
void setup_casovac_preruseni(){
    pinMode(2, INPUT);
    pinMode(2, INPUT_PULLUP);
    EICRA=0b00000110;
    EIMSK|=(1<<INT0);
    CLKPR=(1<<CLKPCE);
    CLKPR=(1<<CLKPS0)|(1<<CLKPS1);
    OCR1A=2499;
    TIMSK1|=(1<<OCIE1A);
    TCCR1B|=(1<<WGM12);
    TCCR1B|=(1<<CS11);
    sei();
}
```

ISR(INT0_vector) – `ISR(INT0_vector)` funkce se zavolá, pokud je vyvoláno externí přerušení na pinu PD2.

```
ISR(INT0_vect){
    prujezd=1;
}
```

ISR(TIMER1_COMPA_vect) – Funkce ISR(TIMER1_COMPA_vect) měří čas závodníkovi či závodníkům (až třem), pokud jsou na trati. Celý vnitřek funguje na tom, že se měří setiny. Ty se po dosažení sta posunou na sekundy. Po dosažení šedesáti sekund se sekundy vynulují a přičte se jedna minuta.

```
ISR(TIMER1_COMPA_vect)
{
    set0++;
    if(set0>=100)
    {
        set0=0;
        sek0++;
    }
    if(sek0>=60){
        sek0=0;
        min0++;
    }
}
if (razeni[1]>0)
{
    set1++;
    if(set1>=100)
    {
        set1=0;
        sek1++;
        if(sek1>=60){
            sek1=0;
            min1++;
        }
    }
}
if (razeni[2]>0)
{
    set2++;
    if(set2>=100)
    {
        set2=0;
        sek2++;
        if(sek2>=60){
            sek2=0;
            min2++;
        }
    }
}
}
```

2.5.4 Sedmisegmentový led displej

Jak už bylo zmíněno výše, tak pro buzení jednotlivých led diod v sedmisegmentovém displeji bylo vybráno dynamické buzení kvůli úspoře energie a úspoře portů mikroprocesoru. Pro jednodušší ovládání segmentů byl vyhrazen celý PORTD na vybrání jednotlivých led pro rozsvícení a dalších pět pinů mikroprocesoru bylo vyhrazeno pro ovládání napájení jednotlivých led displejů.

V prvním kroku je definován výběr pinů pro napájení.

```
#define S1 PIN2_bm
#define S2 PIN2_bm
#define S3 PIN3_bm
#define S4 PIN4_bm
#define S5 PIN5_bm
```

V druhém kroku jsou definována čísla pro zobrazení čísel bez desetinné čárky, pro zobrazení desetinné čárky se od hodnoty musí odečíst hodnota 128, tedy 2^7 .

```
#define nula 96
#define jedna 250
#define dva 164
#define tri 176
#define ctyri 58
#define pet 49
#define sest 35
#define sedm 248
#define osm 32
#define devet 56
```

Dále je už napsaná funkce pro inicializaci pinů mikropočítače pro ovládání sedmisegmentového displeje.

setup_7segment – Ve funkci setup_7segment se nastaví požadované piny na výstup a nastaví se tak, aby byly po inicializaci všechny segmenty zhasnuté.

```
void setup_7segment(){
    PORTF.DIRSET |=S2|S3|S4|S5;
    PORTF.OUTSET |=S2|S3|S4|S5;
    PORTD.DIRSET=0xFF;
    PORTD.OUTSET=244;
    PORTC.DIRSET |=S1;
    PORTC.OUTSET |=S1;
}
```

uloz_segmenty – Funkce uloz_segmenty slouží k tomu, aby po příjmu dat byl čas uložen a zpracován pro zobrazení na displeji.

V této funkci je využívána lokální proměnná pocet, tato proměnná slouží k rozpoznání čísel, jaká se mají zobrazit. Tím, že v cyklech while se odečítá od času hodnota jedna nebo deset a počítá se počet odečtení pomocí proměnné pocet, tak se zjistí, jaká čísla se zobrazí na jednotlivých displejích. Tyto hodnoty se poté předají funkci prirazeni.

```
void uloz_segmenty(uint8_t cas1,uint8_t cas2,uint8_t cas3){
    uint8_t pocet=0;
    if (cas1>9)
        segmenty[0]=63;
    while (cas1>0)
    {
        cas1=cas1-1;
        pocet++;
    }
    prirazeni(pocet,0);
    pocet=0;
    while (cas2>9)
    {
        cas2=cas2-10;
        pocet++;
    }
    prirazeni(pocet,1);
    pocet=0;
    while (cas2>0)
    {
        cas2=cas2-1;
        pocet++;
    }
    prirazeni(pocet,2);
    pocet=0;
    while (cas3>9)
    {
        cas3=cas3-10;
        pocet++;
    }
    prirazeni(pocet,3);
    pocet=0;
    while (cas3>0)
    {
        cas3=cas3-1;
        pocet++;
    }
    prirazeni(pocet,4);
    pocet=0;
}
```

Prirazení – Ve funkci prirazení pomocí funkce switch se přiřadí jednotlivé hodnoty pro zobrazení pomocí vstupních dat jako pocet a pozice. Pocet je číslo, které se má zobrazit a pozice je, na jakém místě se má zobrazit. Na konci funkce se poté k hodnotám na pozicích minuty (pozice 0) a sekundy (pozice 2) odečte 128, čímž se zobrazí tečka. Výsledné hodnoty se poté ukládají do globální proměnné segmenty.

```
void prirazení(uint8_t pocet, uint8_t pozice){
    switch (pocet){
        case 0:{
            segmenty[pozice]=nula;
            break;
        }
        case 1:{
            segmenty[pozice]=jedna;
            break;
        }
        case 2:{
            segmenty[pozice]=dva;
            break;}
        case 3:{
            segmenty[pozice]=tri;
            break;}
        case 4:{
            segmenty[pozice]=ctyri;
            break;}
        case 5:{
            segmenty[pozice]=pet;
            break;}
        case 6:{
            segmenty[pozice]=sest;
            break;}
        case 7:{
            segmenty[pozice]=sedm;
            break;}
        case 8:{
            segmenty[pozice]=osm;
            break;}
        case 9:{
            segmenty[pozice]=devet;
            break;}
    }
    if((pozice==0)|| (pozice==2))
        segmenty[pozice]=segmenty[pozice]-32;
}
```

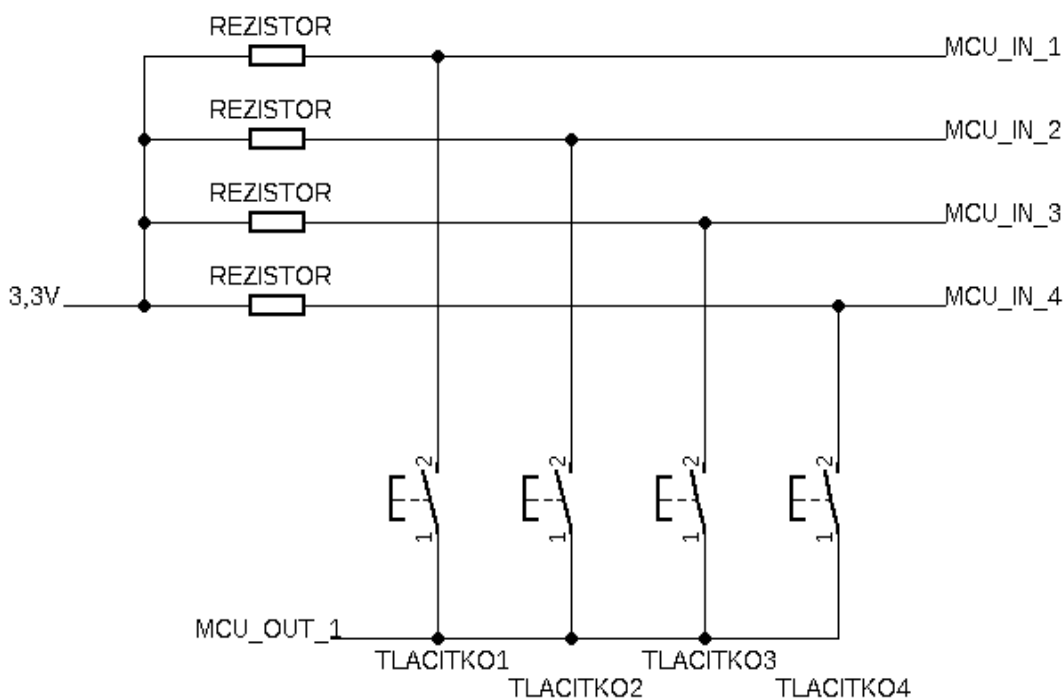
zobraz – Funkce zobraz slouží k zobrazení času na displeji. Čísla se zobrazují dynamicky. V této funkci se opakuje pětkrát podobný cyklus pro všech pět displejů.

- 1) Nejprve v něm zapnu napájení k vybranému displeji pomocí snížení napětí na daném pinu nízko, čímž otevřu NPN tranzistor a PNP tranzistor. Tím propustí napětí pro displej.
- 2) Po tomto kroku na výstup dám hodnotu čísla, která se má zobrazit pomocí proměnné segmenty, ve které jsou z předchozích hodnot uložená čísla.
- 3) Poté zobrazím krátkou dobu číslo, které díky rychlému opakování této funkce se zdá pro lidské oko pořád svítící.
- 4) Po této časové prodlevě příslušný pin nastavím opět na vyšší hodnotu napětí a tím uzavřu NPN tranzistor. V tomto důsledku se napětí na PNP tranzistoru vyrovnají a tím se uzavře a následně přeruší napájení displeje.

```
void zobraz(){
    uint8_t doba=2;
    PORTC.OUTCLR=S1;
    PORTD.OUT=segmenty[0];
    _delay_ms(doba);
    PORTC.OUTSET=S1;
    PORTD.OUT=segmenty[1];
    PORTF.OUTCLR=S2;
    _delay_ms(doba);
    PORTF.OUTSET=S2;
    PORTD.OUT=segmenty[2];
    PORTF.OUTCLR=S3;
    _delay_ms(doba);
    PORTF.OUTSET=S3;
    PORTD.OUT=segmenty[3];
    PORTF.OUTCLR=S4;
    _delay_ms(doba);
    PORTF.OUTSET=S4;
    PORTD.OUT=segmenty[4];
    PORTF.OUTCLR=S5;
    _delay_ms(doba);
    PORTF.OUTSET=S5;
}
```

2.5.5 Klávesnice

Pro inicializaci a ovládání časomíry byla vybrána dvacetitlačítková klávesnice. Klávesnice je tvořena pěti řádky a čtyřmi sloupci. Hodnoty se čtou tak, že mám pět výstupů mikroprocesoru a čtyři vstupy mikroprocesoru. Nejprve je přepnut řádek, který chci číst do logické nuly a poté se čtou všechny řádky, pokud je jeden z řádků v logické nule, pak je daná klávesa stisknuta a provede se odpovídající akce. Takto se čtou pro každý řádek opakovaně. Zapojení níže je zapojení s pull-down rezistorem. Toto zapojení je pro první řádek. Problém s překmity byl řešen pomocí softwarového řešení místo hardwarového s využitím Schmittova klopného obvodu a kondenzátoru.



Obrázek 8 Schéma zapojení prvního řádku klávesnice

klavesnice_setup – klavesnice_setup je funkce pro nastavení požadovaných vstupů a výstupů pro funkční provoz klávesnice.

```
#define SLOUPEC0 0x04
#define SLOUPEC1 0x08
#define SLOUPEC2 0x10
#define SLOUPEC3 0x20
#define RADEK0C 0x04
#define RADEK1E 0x01
#define RADEK2E 0x02
#define RADEK3B 0x10
#define RADEK4B 0x20
void klavesnice_setup(){
    PORTF.DIRCLR |=SLOUPEC0 |SLOUPEC1 |SLOUPEC2 |SLOUPEC3;
    PORTC.DIRSET |=RADEK0C;
    PORTE.DIRSET |=RADEK1E |RADEK2E;
    PORTB.DIRSET |=RADEK3B |RADEK4B;
    PORTC.OUTSET |=RADEK0C;
    PORTE.OUTSET |=RADEK1E |RADEK2E;
    PORTB.OUTSET |=RADEK3B |RADEK4B;
    PORTC.DIRSET |=PIN1_bm;
    PORTC.OUTSET |=PIN1_bm;
}
```

klavesnice_cti_radu – klavesnice_cti_radu je funkce pro čtení stisku klávesy. Má vstupní proměnnou rada pro určení, jaká řada se čte, tedy první řada má vstup nula, druhá řada má vstup čtyři, třetí řada má vstup osm atd. Další vstupní proměnná je klavesa, která je zároveň i výstupní proměnnou pro určení, jaká klávesa byla zmáčknuta.

- 1) Nejprve se čte, jestli je daná klávesa zmáčknuta.
- 2) Pokud je, pak se vykoná zpoždění pro potlačení překmitů.
- 3) Dále se čeká, dokud nebude tlačítko uvolněno.
- 4) Nakonec je přiřazena hodnota do proměnné klavesa, která určuje, jaká klávesa byla zmáčknuta.

```
uint8_t klavesnice_cti_radu(uint8_t rada,uint8_t klavesa){
    if(!(PORTF.IN&(SLOUPEC0))){
        _delay_ms(30);
        while(!(PORTF.IN&(SLOUPEC0)));
        klavesa=1+rada;
    }
    else if(!(PORTF.IN&(SLOUPEC1))){
        _delay_ms(30);
        while(!(PORTF.IN&(SLOUPEC1)));
        klavesa=2+rada;
    }
    else if(!(PORTF.IN&(SLOUPEC2))){
        _delay_ms(30);
        while(!(PORTF.IN&(SLOUPEC2)));
        klavesa=3+rada;
    }
    else if(!(PORTF.IN&(SLOUPEC3))){
        _delay_ms(30);
        while(!(PORTF.IN&(SLOUPEC3)));
        klavesa=4+rada;
    }
    return klavesa;
}
```

klavesnice_cti – klavesnice_cti je funkce pro čtení stisku klávesy pomocí funkce klavesnice_cti_radu a vykonání daného úkolu pro danou klávesu. Protože tato funkce je velice rozsáhlá, tak je uveden pouze začátek programu.

- 1) Nejprve se inicializují proměnné rada a klavesa.
- 2) Poté se nastaví první řádek na logickou nulu.
- 3) Po nastavení se vykoná funkce klavesnice_cti_radu a uloží výsledek do proměnné klavesa.
- 4) Nakonec se nastaví řádek opět na logickou jedna.
- 5) Toto se opakuje pro všech pět řádků.
- 6) Po získání hodnoty klávesy se hodnota srovná a určí se, co se má provést. Mohou se provést tři akce. První a druhá jsou návrat do původního zobrazení OLED displeje a třetí je provedení akce podle hodnoty proměnné klavesa.

```
void klavesnice_cti(){
    uint8_t rada=0,klavesa=0;
    PORTC.OUTCLR=RADEK0C;
    klavesa=klavesnice_cti_radu(rada,klavesa);
    PORTC.OUTSET=RADEK0C;
    rada=rada+4;
    PORTE.OUTCLR=RADEK1E;
    klavesa=klavesnice_cti_radu(rada,klavesa);
    PORTE.OUTSET=RADEK1E;
    rada=rada+4;
    PORTE.OUTCLR=RADEK2E;
    klavesa=klavesnice_cti_radu(rada,klavesa);
    PORTE.OUTSET=RADEK2E;
    rada=rada+4;
    PORTB.OUTCLR=RADEK3B;
    klavesa=klavesnice_cti_radu(rada,klavesa);
    rada=rada+4;
    PORTB.OUTCLR=RADEK4B;
    klavesa=klavesnice_cti_radu(rada,klavesa);
    PORTB.OUTSET=RADEK4B;
    if(klavesa>0){
        if((informace>0)&&(!(klavesa==1)||klavesa==2))){
            informace=0;
            prirad_cislo(21);
        }
        else{
            switch (klavesa){
                case 1:
                {
                    if(informace>0){
                        if(informace<15)
                            informace++;.....}
                        napoveda(informace);
                    }
                    break;
                }
                case 2:
                {
                    if(informace>1){
                        informace--;
                        napoveda(informace); ...}
                }
            }
        }
    }
}
```

2.5.6 OLED displej

Pro zobrazení byl vybrán OLED displej. Tyto displeje mají snadné ovládání. Pro komunikaci s displejem byla vybrána osmibitová paralelní komunikace se třemi pomocnými výstupy.

OLED_setup – OLED_setup je funkce pro inicializování OLED displeje.

- 1) Nejprve se definují tři pomocné piny.
- 2) Poté se provede inicializace portů mikroprocesoru.
- 3) Pak pomocí zpoždění a funkce OLED_zapisdat se inicializuje OLED_display.
- 4) Nakonec programu se vypíše na displej základní zobrazení pomocí funkce OLED_zapisznak.

```
#define RS 0x20
#define RW 0x40
#define ENEABLE 0x80

void OLED_setup(){
    PORTD.DIRSET|=0xFF;
    PORTC.DIRSET|=RS;
    PORTC.DIRSET|=RW;
    PORTC.DIRSET|=ENEABLE;

    PORTD.OUTCLR=0xFF;
    PORTC.OUTCLR|=RS|RW|ENEABLE;

    _delay_ms(100);
    OLED_zapisdat(0,0x33);
    _delay_ms(100);
    OLED_zapisdat(0,0x33);
    _delay_ms(100);
    OLED_zapisdat(0,0x33);
    _delay_ms(20);
    OLED_zapisdat(0,0x33);
    _delay_ms(10);

    OLED_zapisdat(0,0x38);
    _delay_ms(8);
    OLED_zapisdat(0,0x0C);
    _delay_ms(1);
    OLED_zapisdat(0,0x01);
    _delay_ms(20);

    OLED_zapisdat(0,0x06);
    _delay_ms(10);
    OLED_zapisdat(0,0x02);
    strcpy(text,"Cislo: ");
    OLED_zapisznak(text,1);
    strcpy(text,"Pocet jezdcu: 0");
    OLED_zapisznak(text,2);
    strcpy(text,"Cil: ");
    OLED_zapisznak(text,3);
    strcpy(text,"LED: ");
    OLED_zapisznak(text,4);
}
```

OLED_zapisdat – OLED_zapisdat je funkce pro zápis dat do OLED displeje.

Využívá jednobitovou proměnnou rs pro určení, jestli se mají zapisovat data či instrukce, a proměnnou data, jaká data(instrukce) se mají zapsat.

- 1) Nejprve se nastaví pin pro zápis/čtení na nízkou úroveň napětí, čímž dojde k nastavení OLED displeje do režimu zápisu.
- 2) Pak se nastaví pin podle toho, jestli jde o instrukce nebo data.
- 3) Pak zobrazím data na osmibitový PORTD.
- 4) Pak pošlu puls pomocí pinu ENEABLE, který zahájí a ukončí zápis do OLED displeje.
- 5) Na konci funkce vynuluji PORTD a RS pin.

```
void OLED_zapisdat(bool rs,uint8_t data){
    PORTC.OUTCLR=RW;
    if(rs==1)PORTC.OUTSET=RS;
    else PORTC.OUTCLR=RS;
    PORTD.OUT=data;
    PORTC.OUTCLR=ENEABLE;
    _delay_us(1);
    PORTC.OUTSET=ENEABLE;
    _delay_us(1);
    PORTC.OUTCLR=ENEABLE;
    _delay_us(100);
    PORTC.OUTCLR=RS;
    PORTD.OUT=0x00;
}
```

OLED_zapisznak – OLED_zapisznak je funkce pro zápis dat do OLED displeje,

která se mají zobrazit. Má vstupní proměnnou data, která je typu char a určuje, jaká data se mají zobrazit. Proměnná radek určuje, na jaký řádek se mají zapisovat data.

- 1) V prvním kroku se určí řádek, do kterého se má zapisovat.
- 2) Poté se určí délka řetězce znaků, které se mají zobrazit.
- 3) Nakonec se pomocí cyklu for pošlou data do OLED displeje pomocí funkce OLED_zapisdat.

```
void OLED_zapisznak(char *data,uint8_t radek){
    if (radek==1){
        OLED_zapisdat(0,0x01);
        _delay_ms(1);
        OLED_zapisdat(0,0x80);
    }
    else if (radek==2)OLED_zapisdat(0,0xC0);
    else if (radek==3)OLED_zapisdat(0,0x94);
    else if (radek==4)OLED_zapisdat(0,0xD4);
    uint8_t delka=0;
    delka = strlen(data)-1;
    if (delka>20)delka=20;
    for (uint8_t i=0;i<=delka;i++){
        OLED_zapisdat(1,data[i]);}
}
```

2.5.7 Konečné programy

Z menších programů byly sestaveny tři výsledné programy pro každé stanoviště. Stanoviště start je hlavní řídicí jednotkou, která ovládá celou časomíru. Přes stanoviště start závodník ovládá časomíru a ta mu zobrazuje pomocí OLED displeje informace o trati a samotné funkčnosti stanovišť. Ověřování funkčnosti stanovišť probíhá v periodickém vysílání a čekání na odezvu od stanovišť. Pokud start nedostane odpověď od stanovišť do tří period, tak se stanoviště počítají jako nefunkční. Periodické vysílání je přerušeno pouze ve dvou případech. V prvním je, když je zadáno DNF daného závodníka či všech závodníků. V druhém je, pokud je potvrzeno zadané číslo ke startu. Toto přerušení je zavedeno kvůli utlumení komunikace mezi zařízeními a dosažení lepší jistoty, že budou data dodána do cíle, kde jsou dále zpracována a tím omezit množství opakovaných startů závodníka.

Program stanoviště cíl má za úkol odpovídat stanovišti start a předávat informace o funkčnosti SD karty a stanovišti LED. Pokud cíl přijme číslo závodníka začne měřit jeho čas až do chvíle, kdy je zaznamenán senzorem. Tento program je uzpůsoben tak, aby byl schopen měřit čas až třem závodníkům najednou. Toto sice není praktické pro závod, ale pro využití na trénincích je to velice přínosné. Pokud závodníka zaznamená cílový senzor průjezdu, pak je jeho číslo spolu s časem uloženo na SD kartu. Nedostatkem tohoto programu je způsob zjištění funkčnosti stanoviště LED kvůli tomu, že stanoviště cíl posílá požadavek na odpověď od stanoviště LED pouze pokud senzor průjezdu zaznamená pohyb. Tento způsob byl vybrán kvůli tomu, aby se snížilo množství vadných zpráv z důvodů vyrušení ze dvou vzájemných zpráv. Tento problém by šel řešit pomocí vzájemné synchronizace stanovišť.

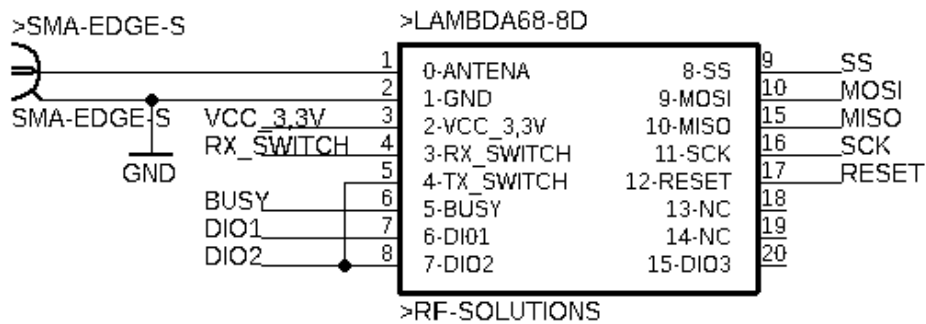
Program stanoviště LED má pouze úkol přijímat čas od cíle a následně ho zobrazit a předat potvrzení o přijetí zprávy. Čas zobrazuje pomocí dynamického buzení jednotlivých segmentů displeje, aby se ušetřila energie baterie.

2.6 Schéma časomíry

Než se mohla dát vytvořit PCB, tak bylo nutné vytvořit schéma zapojení. Pro vytvoření schémat i návrhu PCB byl vybrán návrhový program Eagle.

Také bylo nutno nakreslit nové knihovny pro vývojové kity AVR128DB28-CNANO a ATMEGA328PB-XMINY. Také pro rádiový modul LAMBDA68-8D s čipem LLCC68.

Všechny stanoviště mají rádiový modul, pomocí kterého mezi sebou komunikují, zapojení pro všechny tři stanoviště jsou stejná.



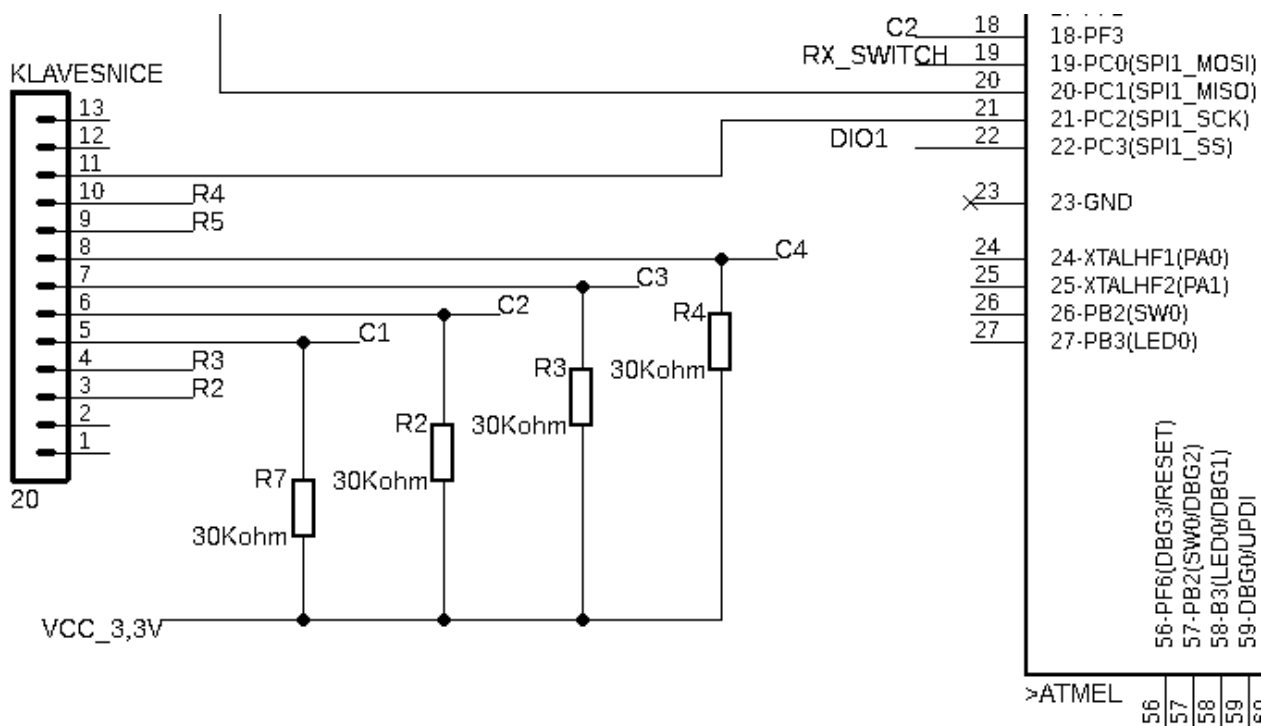
Obrázek 9 Schéma zapojení modulu rádia

2.6.1 Start

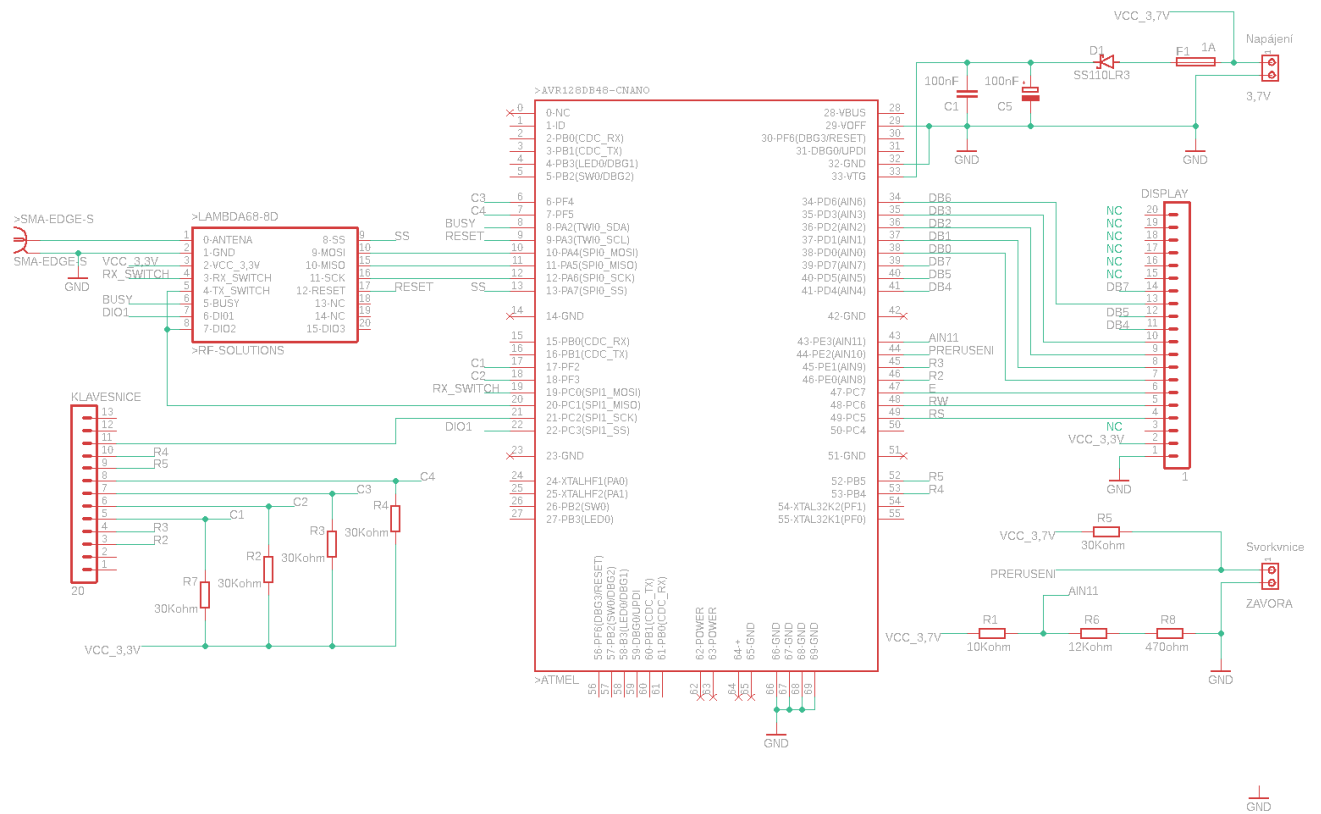
Stanoviště start slouží jak k odstartování závodníka, tak i ke komunikaci mezi obsluhou a časomírou. Pomocí OLED displeje je časomíra schopná zobrazovat stav funkčnosti jednotlivých stanovišť, zobrazit potvrzení o doručení zprávy, možnost zobrazit návod pro použití časomíry. Obsluha komunikuje s časomírou pomocí dvacetitlačítkové klávesnice. Základních deset tlačítek slouží k zadání čísla závodníka. Ostatní tlačítka slouží k vykonání činnosti, třeba k vymazání veškerých zadaných dat do klávesnice, potvrzení zadání čísla, zobrazení nápovědy, ovládání nápovědy, zadání DNF k zadanému číslu atd. Po potvrzení čísla může závodník odstartovat a vydat se do cíle.

Napájení pro toto stanoviště řeším pomocí 3,7V baterie. Před napájením čipu je zapojena ochranná dioda kvůli přepólování, což by jinak znamenalo zničení stanoviště. Dioda je typu schottky, aby úbytek napětí byl co nejmenší.

V zapojení klávesnice se využívají pull-down rezistory, aby při klidovém stavu byla logická úroveň na pinu jedna. A při stisku klávesy se snížila na nulu.



Obrázek 10 Schéma zapojení klávesnice

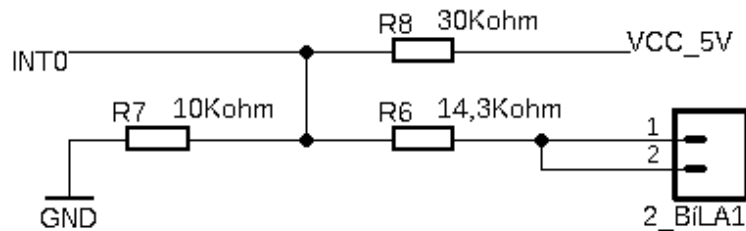


Obrázek 11 Schéma zapojení stanoviště start

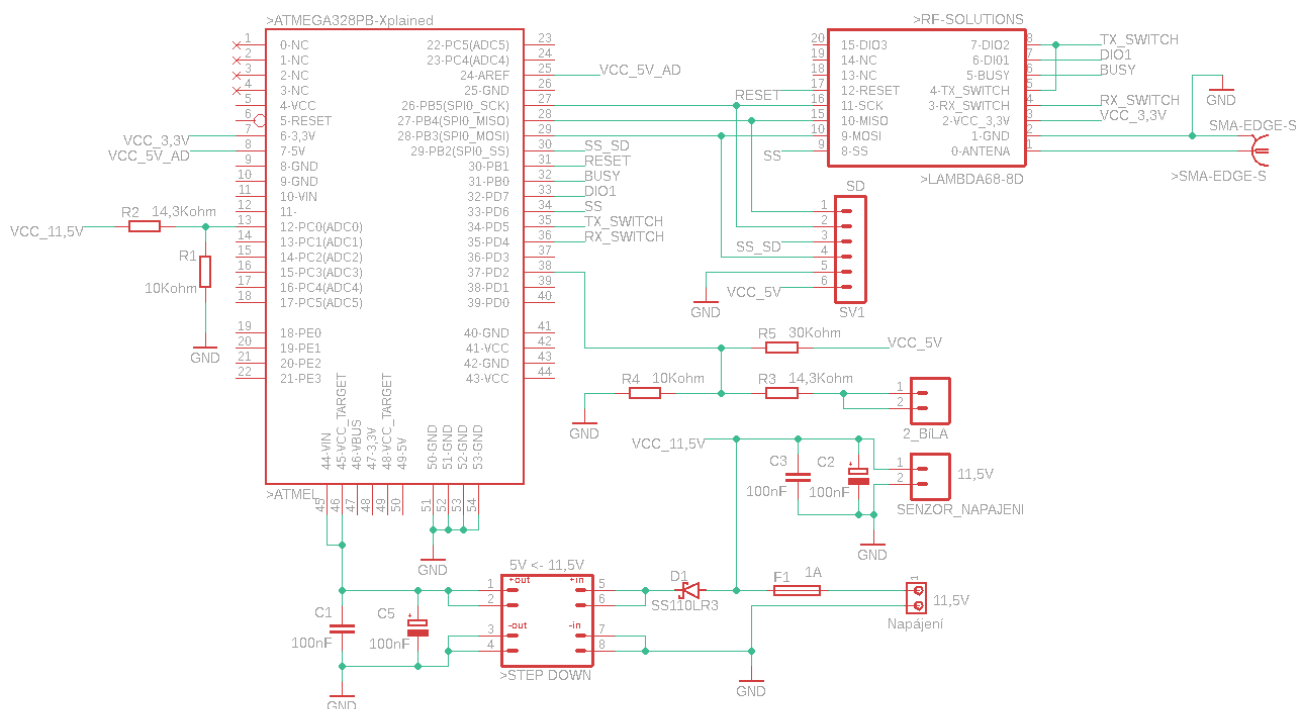
2.6.2 Cíl

Napájení je zde děláno pomocí snížení napětí na pět voltů pomocí obvodu step-down. Modul step-down je realizován integrovaným obvodem MP1584EN.

Stanoviště cíl plní úkony jako měření času závodníka, následné uložení času závodníka, detekce závodníka a komunikování s ostatními stanovišti pomocí rádiové komunikace. Při návrhu musel být vyřešen problém s příliš vysokým výstupním napětím ze senzoru detekce závodníka, který pracuje na úrovních napětí mezi deseti až třiceti volty. Výstupní napětí bylo zmenšeno pomocí děliče napětí s rezistory v řádech desítek kiloohmů kvůli snížení odběru proudu.



Obrázek 12 Zapojení senzoru pro detekci závodníka

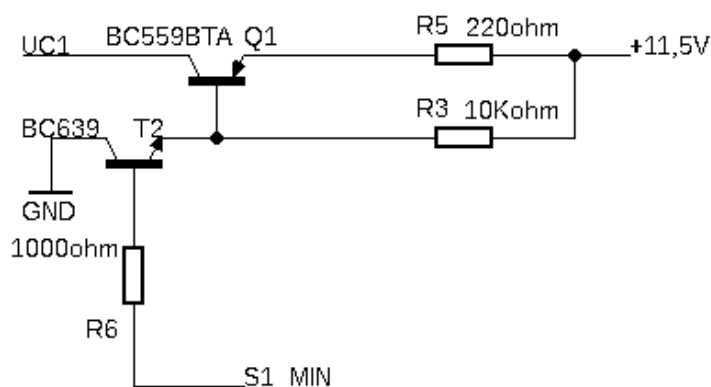


Obrázek 13 Schéma zapojení stanoviště cíl

2.6.3 Zobrazovací stanoviště

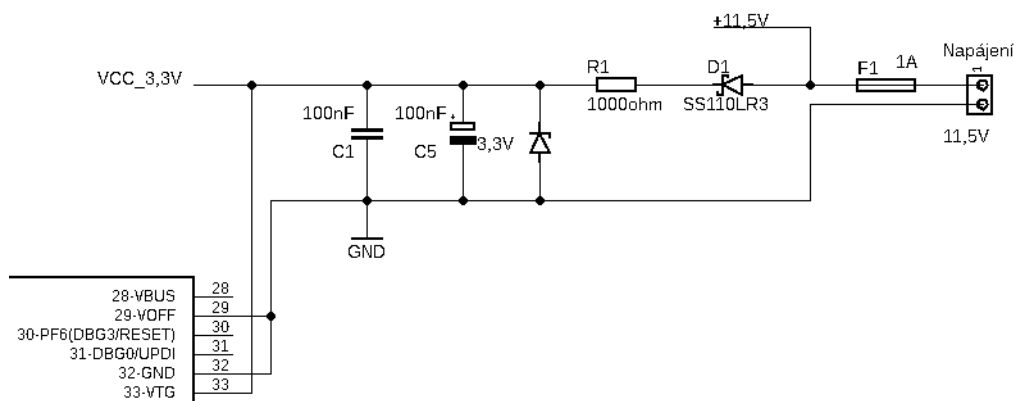
Zobrazovací stanoviště má za úkol zobrazovat výsledný čas závodníka. Při návrhu stanoviště musel být vyřešen problém se spínáním jednotlivých displejů. Tyto displeje potřebují napájení aspoň osm voltů pro červené diody, aby se rozsvítily. Tento požadavek vznikl kvůli sériovému zapojení čtyř diod v displeji pro každý segment mimo desetinné čárky, která je tvořena jednou diodou.

Tento problém byl vyřešen pomocí zapojení níže pro napájení displeje zobrazující minuty, snížením napětí na pinu S1_MIN nízko, čímž se otevře NPN tranzistor a PNP tranzistor a tím se propustí napětí pro displej. V případě, kdy chceme ukončit napájení displejů, tak nastavíme pin S1_MIN na vyšší hodnotu napětí a tím uzavřeme NPN tranzistor. V tomto důsledku se napětí na PNP tranzistoru vyrovnají a tím se uzavře a následně přeruší napájení displejů.

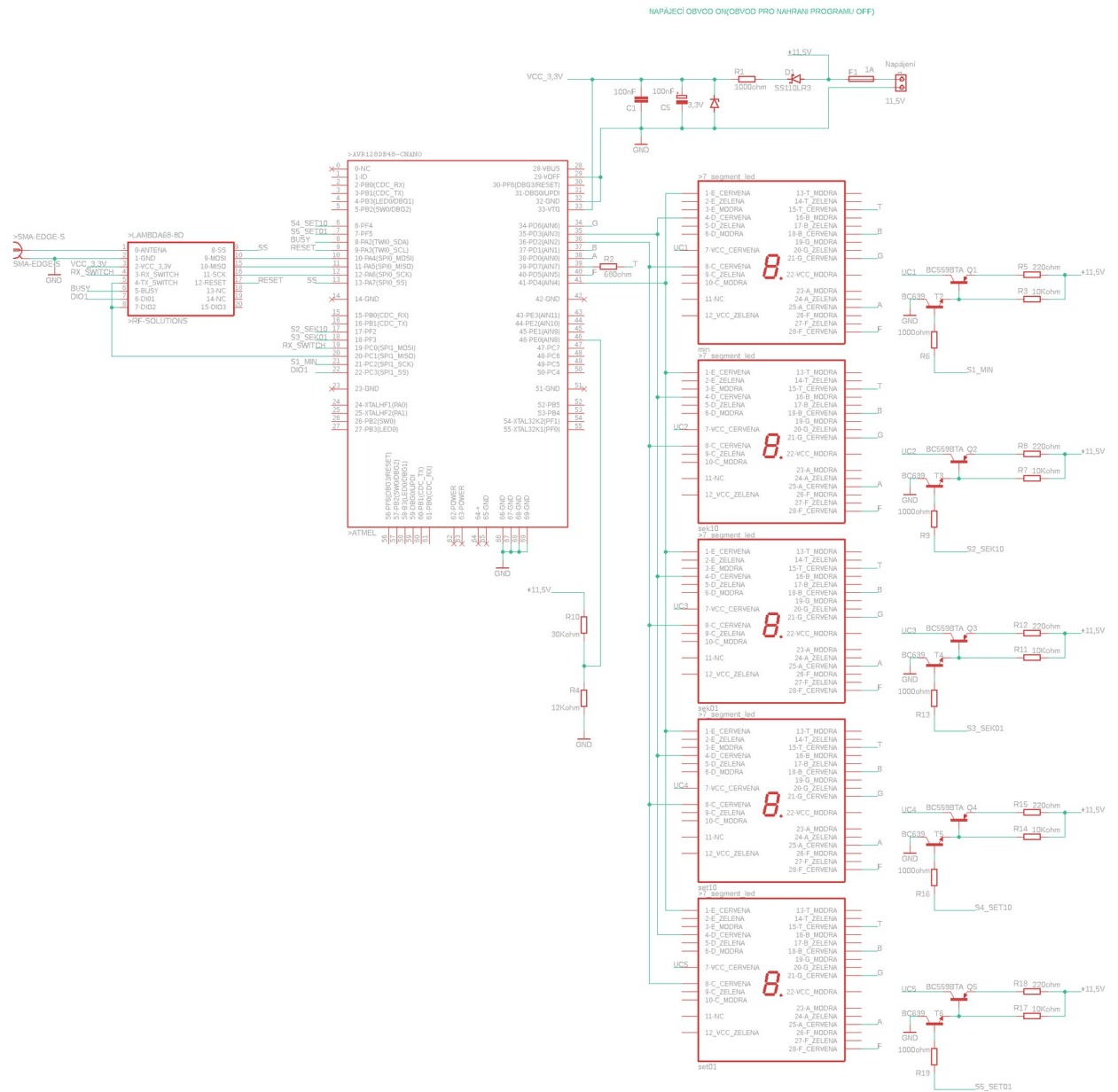


Obrázek 14 Zapojení spínání napájení pro segmentové displeje

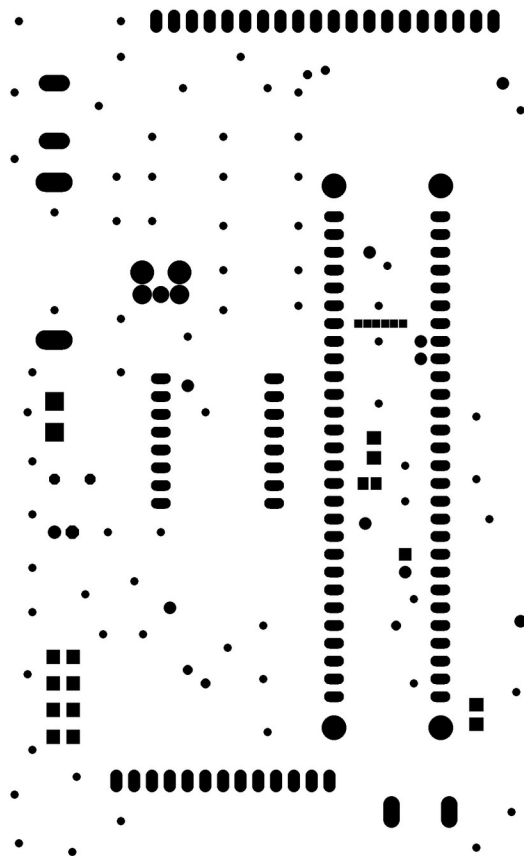
Napájení bylo v tomto případě řešeno pomocí Zenerovy diody. Kvůli hrozbě prohození napájení byla vložena do obvodu Schottkyho dioda, aby se obvod vlivem prohození polarit nezničil. Tato dioda byla vložena, až za odbočku pro napájení displeje, protože by vznikl úbytek napětí a samotný obvod pro napájení displeje je odolný proti přepólování sám od sebe.



Obrázek 15 napájení stanoviště LED



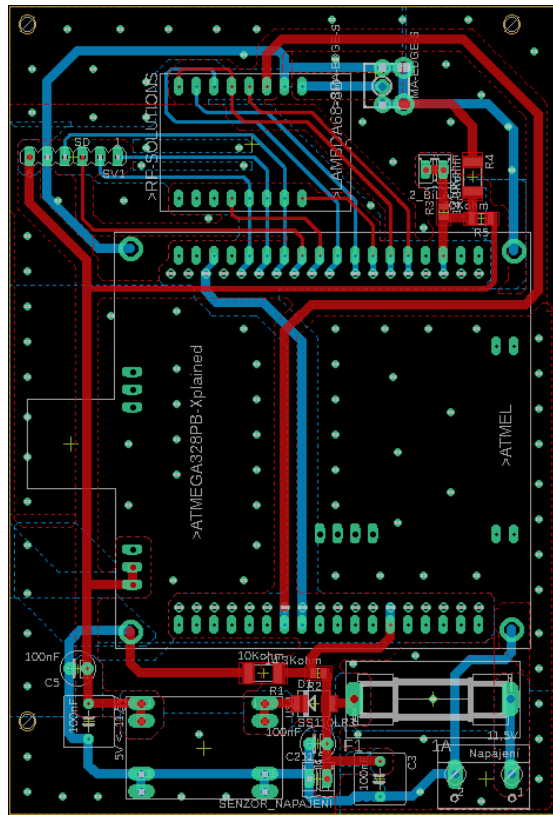
Obrázek 16 Schéma zapojení stanoviště LED



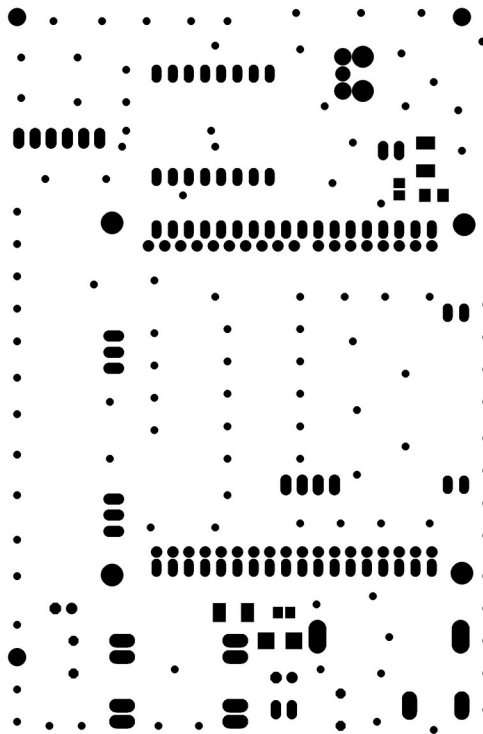
Obrázek 18 Osazovací PCB start

3.1.2 PCB cíl

Při návrhu stanoviště cíl se dbalo na to, aby stanoviště bylo co možná nejmenší, aby se v případném praktickém použití snadno nainstalovalo do ochranného krytu.



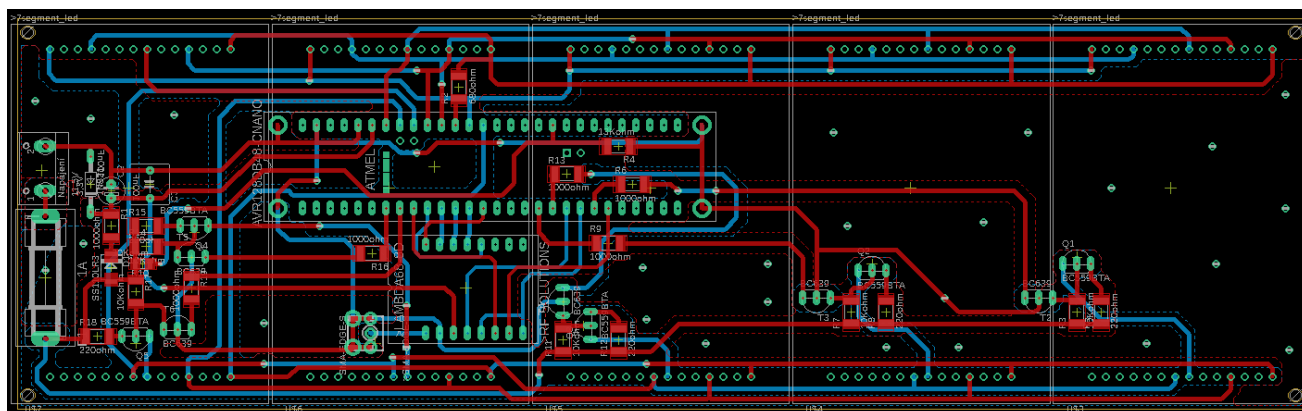
Obrázek 19 PCB cíl



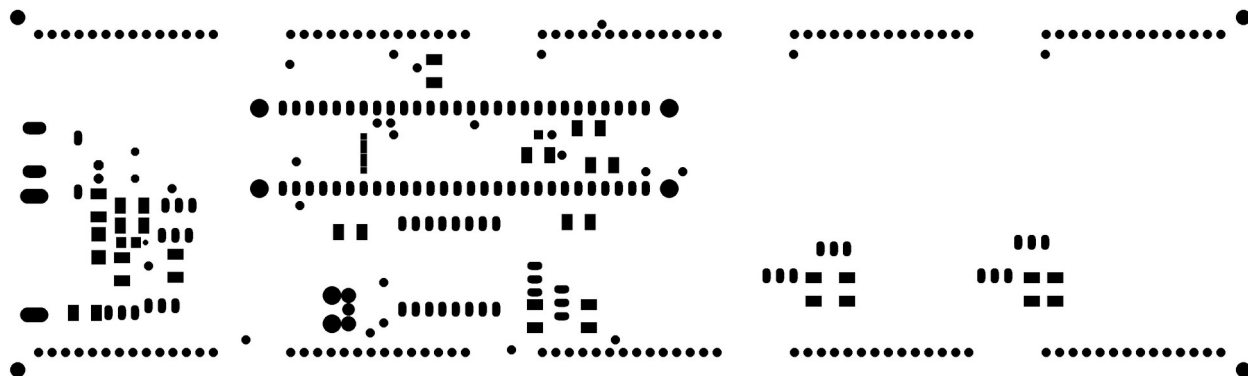
Obrázek 20 Osazovací PCB cíl

3.1.3 PCB zobrazovací stanoviště

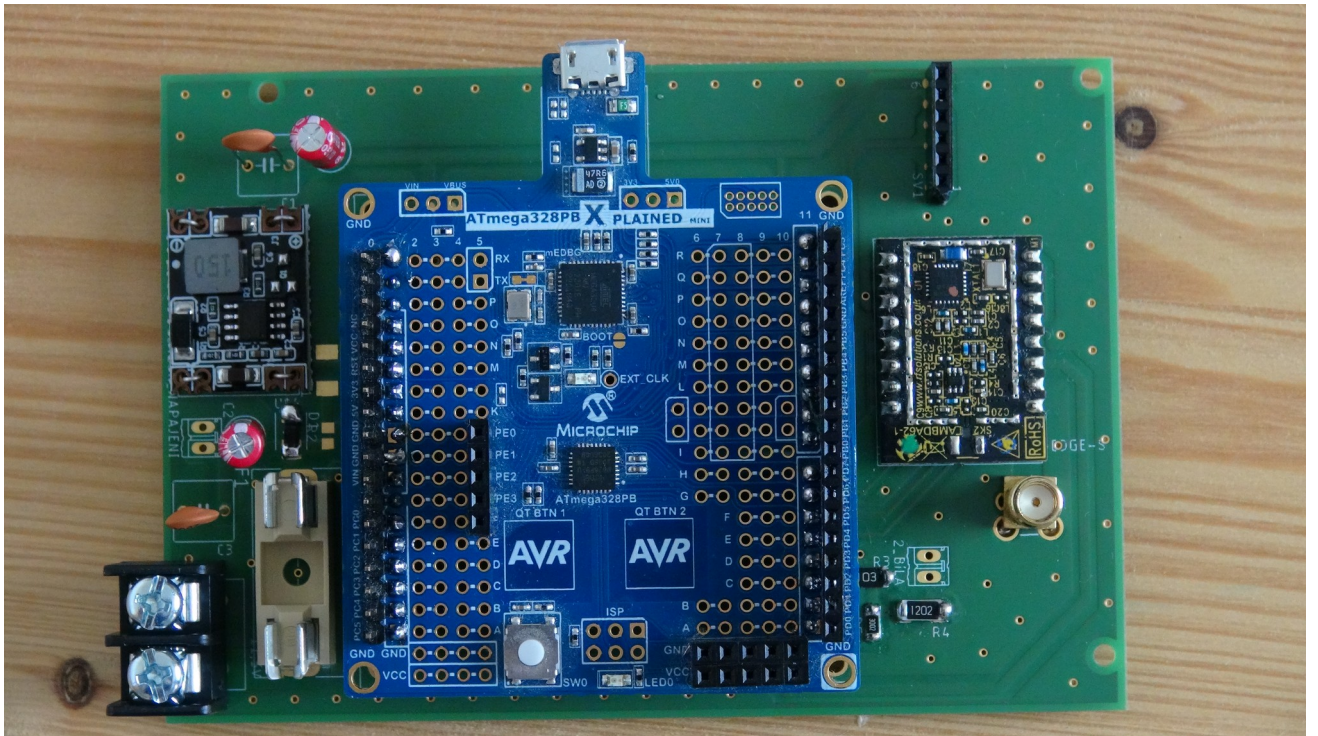
Podle schématu byla navrhnutá PCB a poté se dala deska vyrobit. Po výrobě desky se osadila součástkami a oživila.



Obrázek 21 PCB LED



Obrázek 22 Osazovací PCB LED



Obrázek 24 Stanoviště cíl



Obrázek 25 Stanoviště LED

ZÁVĚR

Hlavními body práce bylo navrhnout a sestavit časomíru se vzdáleným startem od cíle a následným zobrazením výsledného času.

Pro dálkovou komunikaci mezi startem a cílem byla zvolena bezdrátová komunikace v bezlicenčním pásmu o frekvenci 868MHz. Pro tuto komunikaci byl zvolen rádiový modul s modulací FSK. Nejprve byla zprovozněna pouze simplexní komunikace mezi moduly. Po úspěšném zprovoznění modulů v simplexním režimu byly moduly zprovozněny v poloduplexní komunikaci. V poloduplexní komunikaci může vysílat vždy pouze jedno zařízení a ostatní musí naslouchat.

Po vytvoření rádiové komunikace byly přidány senzory pro start a přijetí do cíle. Jakmile byla funkčnost senzorů odzkoušena, tak bylo přidáno měření času. Měření času bylo provedeno pomocí 16-bitového časovače. Po úspěšném odzkoušení měření času při pokojové teplotě bylo přidáno třetí stanoviště LED, které má za úkol zobrazovat výsledný čas na pět sedmsegmentových displejů.

Po splnění cílů bylo doděláno ke stanovišti start komunikační rozhraní, aby bylo možné časomíru ovládat a dostávat od ní zpětnou vazbu o stavu časomíry. Jako vstupní periférie byla zvolena klávesnice a jako výstupní periférie byl zvolen OLED displej se čtyřmi řádky. Pomocí dodělání těchto periférií je možné ovládat časomíru a použít ji i pro účely, kde je potřeba identifikovat závodníka a poté mu následně uložit čas. Pro uložení času byla vybrána SD karta.

Na závěr byla pro každé stanoviště vyrobena PCB. A poté osazena součástkami a patičkami pro vývojové kity, aby bylo možné jejich snadné vyměnění při jejich zničení.

V této práci byla zprovozněna a zkonstruována plně funkční časomíra, avšak při jejím zkušebním provozu byly objeveny nedostatky jako zjišťování stavu stanoviště LED či nekomfortní ukládání výsledků na SD kartu.

V této práci byly splněny veškeré zadané požadavky na tuto bakalářskou práci s názvem Časomíra.

POUŽITÁ LITERATURA

- [1] MATOUŠEK, David a BRTNÍK, Bohumil. *Programování mikrokontrolérů s jádrem 8051 v jazyce C: názorné příklady a funkční programy pro AT89S52*. MC & praxe. Praha: BEN - technická literatura, 2010. ISBN 978-80-7300-264-0.
- [2] LLCC68 Datasheet. In: Semtech [online]. [cit. 2024-01-29]. Dostupné z: <https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000HTJR/Tem0gUxGfOZ2Qn3bUzmV2zKNQRYJ3bpobPfoQ7B.erE>
- [3] LAMBDA80 - Example Code [software]. In: RF Solutions [online]. [cit. 2024-01-29]. Dostupné z: <https://www.rfsolutions.co.uk/arduino-i49>
- [4] SD [software]. In: Arduino [online]. [cit. 2024-01-29]. Dostupné z: <https://www.arduino.cc/reference/en/libraries/sd/>
- [5] ATmega328PB Xplained Mini Datasheet. In: Microchip [online]. [cit. 2024-01-29]. Dostupné z: <https://onlinedocs.microchip.com/pr/GUID-FC2A0384-AC9D-45B4-951E-5C0DEF8B2E9-en-US-5/index.html>
- [6] ATmega328PB Datasheet. In: Microchip [online]. [cit. 2024-01-29]. Dostupné z: <https://onlinedocs.microchip.com/pr/GUID-0EC909F9-8FB7-46B2-BF4B-05290662B5C3-en-US-12.1.1/index.html?GUID-1C7FC5CD-E28F-4D27-95A1-FD8BAD20788A>
- [7] AVR128DB48 Curiosity Nano Datasheet. In: Microchip [online]. [cit. 2024-01-29]. Dostupné z: <https://ww1.microchip.com/downloads/en/DeviceDoc/AVR128DB48-Curiosity-Nano-HW-UserG-DS50003037A.pdf>
- [8] AVR128DB28/32/48/64 Datasheet. In: Microchip [online]. [cit. 2024-01-29]. Dostupné z: <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/AVR128DB28-32-48-64-DataSheet-DS40002247.pdf>
- [9] 8 Advantages and Disadvantages of LoRa Module. In: MINERWSEMI [online]. [cit. 2024-01-29]. Dostupné z: <https://en.minersemi.com/blog/8-advantages-and-disadvantages-of-lora-module>
- [10] FSK Advantages | FSK Disadvantages | Frequency Shift Keying. In: RF Wireless World [online]. [cit. 2024-01-29]. Dostupné z: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-FSK.html>
- [11] VYUŽITÍ RÁDIOVÉHO SPEKTRA. In: Český telekomunikační úřad [online]. [cit. 2024-01-29]. Dostupné z: <https://spektrum.ctu.cz/>
- [12] Advantages and Disadvantages of Infrared sensor. In: RF Wireless World [online]. [cit. 2024-01-29]. Dostupné z: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Infrared-Sensor.html>
- [13] Advantages of Laser | Disadvantages of Laser. In: RF Wireless World [online]. [cit. 2024-01-29]. Dostupné z: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Laser.html>
- [14] Advantages of optical sensor | Disadvantages of optical sensor. In: RF Wireless

- World [online]. [cit. 2024-01-29]. Dostupné z: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Optical-Sensor.html>
- [15] Automatic Identification and Data Capturing (AIDC) Technology. In: Elprocus [online]. [cit. 2024-01-29]. Dostupné z: <https://www.elprocus.com/aidc-what-is-automatic-identification-and-data-capture-technology/>
- [16] LAUZIER, Jákob. In: machinemetrics.com [online]. 2022 [cit. 2024-01-29]. Dostupné z: <https://www.machinemetrics.com/blog/manual-data-collection>
- [17] Different Types of Crystal Oscillators: A Comprehensive Guide. In: SIWARD [online]. 2023 [cit. 2024-01-29]. Dostupné z: https://www.siward.com/en/about/industry/Different_Types_of_Crystal_Oscillators__A_Comprehensive_Guide
- [18] AVR053: Internal RC Oscillator Calibration for tinyAVR and megaAVR Devices In: Microchip [online]. [cit. 2024-01-29]. Dostupné z: www.microchip.com/content/dam/mchp/documents/FOTH/ApplicationNotes/ApplicationNotes/Atmel-2555-Internal-RC-Oscillator-Calibration-for-tinyAVR-and-megaAVR-Devices_ApplicationNote_AVR053.pdf&usg=AOvVaw00_B2fzMYsikomtYDW4HdA&opi=89978449
- [19] LCD Fundamentals and the LCD Driver Module of 8-Bit PIC® Microcontrollers In: Microchip [online]. [cit. 2024-01-29]. Dostupné z: www.microchip.com/downloads/en/Appnotes/FAN658-LCD-Fundamentals-and-the-LCD-Driver-Module-of-8-Bit-PIC-MCUs-00000658C.pdf&usg=AOvVaw07jVruuiOoiBBemOwD7hGq&opi=89978449

SEZNAM PŘÍLOH

PŘÍLOHA A

- Zdrojový kód stanoviště start
- Zdrojový kód stanoviště cíl
- Zdrojový kód stanoviště LED