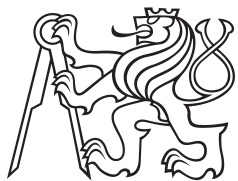


Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra mikroelektroniky

RFID časomíra

Tomáš Hrdina

Vedoucí: Ing. Tomáš Teplý
Studijní program: Elektronika a komunikace
Srpen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hrdina** Jméno: **Tomáš** Osobní číslo: **492162**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra mikroelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

RFID časomíra

Název bakalářské práce anglicky:

RFID Timing System

Pokyny pro vypracování:

Rozeberte možnosti využití RFID při odměřování časových úseků pro sportovní účely, zvolte nejvhodnější řešení vzhledem k požadavkům na přesnost měření časových úseků a spolehlivost načítání/identifikace závodníků, realizujte prototyp časomíry pro tréninkové použití.

Postup řešení:

- 1) Proveďte rešerši trhu v oblasti sportovních časomír se zaměřením na časomíry využívající bezkontaktní identifikaci.
- 2) Proveďte návrh vlastního řešení.
- 3) Zařízení realizujte.
- 4) Otestujte funkčnost Vašeho řešení.
- 5) Shrňte dosažené výsledky a uveďte výhody a nevýhody ve srovnání s běžně užívanými časomírami.

Seznam doporučené literatury:

- [1] Robert B. Reese: Microprocessors From Assembly Language to C Using the PIC18Fxx2, Da Vinci Engineering Press, Hingham, Massachusetts, 2005
- [2] Lehpamer, H.: RFID Design Principles, Artech House, 2012
- [3] Dobkin, D. – Dobkin, M., D.: The RF in RFID: Passive UHF RFID in Practice, Elsevier Science & Technology, 2007

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Tomáš Teplý katedra mikroelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **31.01.2022**

Termín odevzdání bakalářské práce: **15.08.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Tomáš Teplý
podpis vedoucí(ho) práce

prof. Ing. Pavel Hazdra, CSc.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval svému vedoucímu práce za cenné rady a připomínky, které mi poskytl při psaní této práce. Dále bych rád poděkoval své rodině a svým přátelům za podporu při studiu a při sportu, který mi byl oporou.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a s použitím uvedené literatury.

V Praze, 15. srpna 2022

Abstrakt

Práce se zabývá bezkontaktním měřením času s využitím v atletických trénincích. V textu jsou popsány již dostupné způsoby měření času. Dále se věnuji mému zpracování bezdrátové časomíry s využitím RFID technologie umožňující měření ve sportu. Součástí je i otestování zařízení v reálných tréninkových podmínkách, porovnání s ostatními časomírami a uvedení výhod a nevýhod jednotlivých typů časomír. V poslední části je představen i vlastní návrh DPS a její realizace.

Klíčová slova: časomíra, RFID technologie, mikrokontroler, UHF, tag

Vedoucí: Ing. Tomáš Teplý

Abstract

The work is focused timing during sports training. In the text below there are described available means of timing with the description how they work. Part of this work is also realisation of athletics timing system with usage of RFID reader including testing in real conditions during sports training. Next, there is comparison with other timing systems and description of their advantages and disadvantages. There is also my realization of athletics timing with my design of PCB.

Keywords: timing, RFID technology, microcontroller, UHF, tag

Title translation: RFID timing system

Obsah

1 Úvod	1
1.1 Cíle práce	1
2 RFID technologie	3
2.1 Princip	4
2.2 Typy RFID	4
2.2.1 Induktivní spojení	5
2.2.2 Radiační spojení	5
2.3 Tag	5
2.3.1 Aktivní čipy	5
2.3.2 Pasivní čipy	6
2.3.3 Použití RFID	7
2.3.4 Anténa	7
3 Dostupné časomíry	9
3.1 Trvalé řešení	9
3.1.1 Výhody trvalého řešení	10
3.1.2 Nevýhody trvalého řešení	10
3.2 Rámové řešení	10
3.2.1 Výhody rámového řešení	10
3.2.2 Nevýhody rámového řešení	11
3.3 BLE Technologie	11
3.3.1 Výhody BLE technologie	11
3.3.2 Nevýhody BLE technologie	11
3.4 Závodní časomíra	11
3.4.1 Výhody a nevýhody	12
4 Mé řešení atletické časomíry	13
4.1 RFID čtečka	14
4.1.1 Tag	14
4.1.2 Mikrokontroler	15
4.1.3 Displej	19
4.1.4 Měření, otestování časomíry a čtečky	21
4.1.5 Deska	22
4.2 Použité typy komunikace	23
4.2.1 USART	24
4.2.2 SPI	25
4.2.3 USB	25
5 Závěr	27
Seznam použitých zkratk	29
Literatura	31
Přílohy	34
A Zdrojový kód	35
B Osazená deska	47

Obrázky

2.1 Elektromagnetické spektrum [3] .	3
2.2 RFID technologie [1]	3
2.3 Detekce tagu [1]	4
2.4 Pasivní tag [1]	6
2.5 Různé podoby UHF tagu	6
2.6 Směrnost antény[1]	7
3.1 Konfigurace trvalého řešení [4] ...	9
3.2 Rámové řešení [5]	10
3.3 BLE technologie [6]	11
3.4 Závodní časomíra (obrázek startovací pistole převzat z [7], displeje z [8], fotobuněk z [9] a cílové kamery z [10])	12
4.1 Schéma sestavené časomíry (obrázek notebooku použit pod licencí Creative Commons [11], obrázek desky z [12])	13
4.2 B-Code [18]	20
4.3 7segmentový displej [18]	20
4.4 Graf počtu úspěšně zaregistrovaných pohybů čipu kolem čtečky v závislosti na vzdálenosti mezi čipem a čtečkou	22
4.5 Návrh desky v prostředí KiCad .	23
4.6 Propojení zařízení přes UART [21]	24
B.1 Osazená deska	47
B.2 Osazená deska s displeji	48

Tabulky

2.1 Druhy RFID	5
4.1 Propojení displeje č.1 a č.2	14
4.2 Propojení pinů desky s displeji a čtečkou	14
4.3 Fáze tréninku	18
4.4 Vliv umístění čipu na počet úspěšně registrovaných průběhů cílem	21
4.5 Počet úspěšně registrovaných průběhů	21
4.6 Test časomíry při tréninku	22

Kapitola 1

Úvod

Čas je jedna ze základních fyzikálních veličin používaných v každodenním životě. Setkáváme se s ním každý den a ani si to neuvědomujeme. Pro sportovce je čas zásadní veličinou, především v běžeckých disciplínách. Umožňuje nám porovnávat se mezi sebou, stanovovat světové rekordy, zlepšovat si svá osobní maxima a nastavit si pravidelnost během atletických tréninků, která přispívá ke zlepšení. Pro závodníky na středních tratích (1500 m, 3 000 m) jsou intervalové tréninky součástí mnoha tréninků a čas v nich hraje důležitou roli.

Bezdrátové technologie měření jsou založeny na principu měření času závodníka, který probíhá danou oblastí, ve které je registrovaná jeho přítomnost. Měřit čas lze mnoha způsoby, mezi ně patří RFID technologie, BLE nebo závodní časomíra, která se používá např. při atletických závodech. RFID technologie má dosah až několik metrů, BLE dosahuje až na několik stovek metrů a závodní časomíra registruje pohyb pouze na cílové čáře.

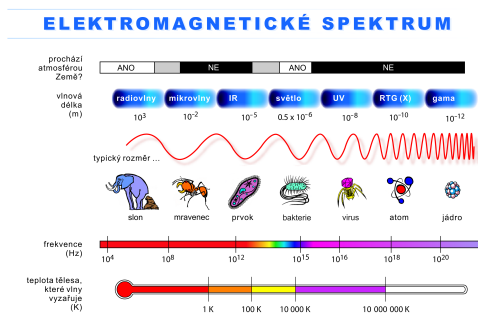
1.1 Cíle práce

Cílem mé práce je zanalyzovat trh sportovních časomír pro bezkontaktní identifikaci a uvést jejich vlastnosti. Dalším úkolem je časomíru zkonstruovat a otestovat. Posledním úkolem je vytvořenou časomíru porovnat s nalezenými časomírami a shrnout dosažené výsledky.

Kapitola 2

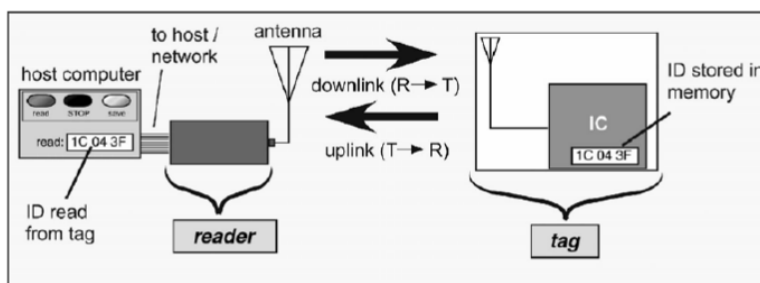
RFID technologie

RFID technologie je založena na principu identifikování objektů nebo osob pomocí radiových vln bez nutné přímé viditelnosti [1]. Rádiové vlny řadíme mezi neionizující záření 2.1 s frekvencemi v rozmezí 125 kHz až 2,45 GHz [2]. Velké množství identifikačních čísel umožňuje jednoznačnou identifikaci ob-



Obrázek 2.1: Elektromagnetické spektrum [3]

jektů. Systém obsahuje tyto součásti: čtečka, tag a systém pro zpracování dat (nejčastěji připojený počítač, ale data mohou být odesílána na server, případně do databáze) 2.2. Dosah technologie se liší v závislosti na typu

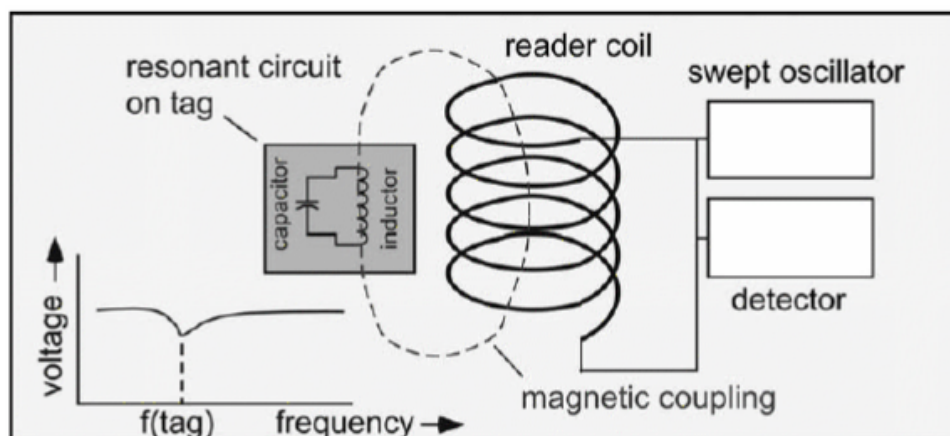


Obrázek 2.2: RFID technologie [1]

použití, dosahuje od několika centimetrů až po stovky metrů.

2.1 Princip

Technologie se liší v závislosti na použitém způsobu propojení čtečky a tagu. Přítomnost tagu v dosahu čtečky registrujeme náhlým poklesem napětí na čtecí cívce 2.3. Aby mohl tag fungovat, získává energii usměrněním příja-



Obrázek 2.3: Detekce tagu [1]

tého signálu. Rozlišujeme dva způsoby propojení tagu a čtečky, induktivní a radiační.

V případě induktivního propojení se využívá toho, že čtečka obsahující čtecí cívku vytváří ve svém okolí elektromagnetické pole jehož energii tag využívá. Data uložená v tagu regulují odpor zátěže a tím ovlivňují amplitudovou modulaci napětí na cívce čtečky. Mohou nastat tři případy. Impedance zátěže je shodná s tzv. intrinzickou impedancí, pak je koeficient odrazu roven nule a nedochází k odrazu energie zpět ke čtečce. V jiném případě se stane, že impedance zátěže je rovna nekonečnu, koeficient odrazu je roven 1 a nastává odraz energie ve fázi s energií dopadající. Poslední případ, který může nastat, je ten, že impedance zátěže je rovna nule a nastává odraz energie v protifázi. Způsob této modulace se nazývá zátěžová modulace. Tento způsob propojení čtečky a tagu je typický pro nízké a vysoké frekvence [2].

V druhém případě se využívá elektromagnetického pole a zpětně rozptýleného signálu. Čtečka vyzařuje nemodulovaný signál, který je anténou tagu zachycen. Část signálu je zachycena, část pohlcena a zbývající část znovu vyzářena. Signál je modulovaný modulátorem, což je FET tranzistor pracující v několika režimech 2.3.2.

2.2 Typy RFID

RFID technologie dělíme na několik pásem podle toho v jakém daná technologie vysílá. Rozlišujeme nízké frekvence (LF), vysoké frekvence (HF) a velmi vysoké frekvence (UHF) (tabulka 2.1). Se změnou frekvence se také mění vlnová délka, definovaná vztahem 2.1, kde λ je vlnová délka v metrech,

Tabulka 2.1: Druhy RFID

typ RFID	frekvence [MHz]	typ spojení
LF	0,125/0,134	induktivní
HF	13,56	induktivní
UHF	890-960, 2400	radiační

c rychlost světla v m/s a f frekvence v Hz.

$$\lambda = \frac{c}{f} \quad (2.1)$$

Vlnovou délkou rozumíme vzdálenost dvou nejbližších míst kmitajících se stejnou fází. Pro RFID technologii platí, že vlnová délka se pohybuje v rozmezí od 12,5 cm (UHF) do 3 km (LF), pokud počítáme s mezními frekvencemi 125 kHz a 2,4 GHz.

■ 2.2.1 Induktivní spojení

Induktivní spojení se používá u systémů, kde je vlnová délka mnohem delší než délka antény, patří sem systémy pracující na nízkých a vysokých frekvencích. Spojení strmě klesá s rostoucí vzdáleností tagu od čtečky. Cívka v tagu obsahuje desítky až stovky závitů [2].

■ 2.2.2 Radiační spojení

Naopak u systémů, kde je vlnová délka srovnatelná s rozměry antény se používá radiační spojení. K těmto systémům řadíme ty, které pracují na velmi vysokých frekvencích. Spojení tagu a čtečky klesá postupně s rostoucí vzdáleností. Pro velmi vysoké frekvence se nepoužívají antény tagů s cívkami, ale s dipóly. Délka dipólu je obvykle polovinou vlnové délky [2].

■ 2.3 Tag

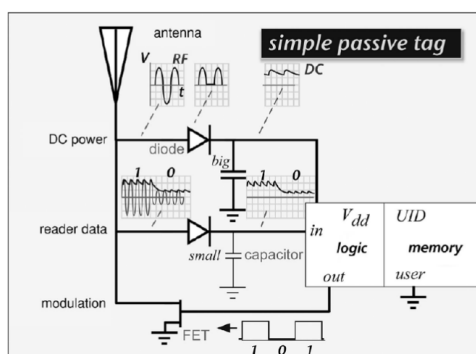
Tag neboli čip je označení pro prvek jehož pohyb je snímán anténou čtečky. Jedná se o rezonanční obvod složený z cívky a kondenzátoru, které společně generují unikátní frekvenci. Podle toho, zda jsou napájeny nebo ne, rozlišujeme dvě skupiny čipů, pasivní a aktivní [1].

■ 2.3.1 Aktivní čipy

Aktivní čipy pro své fungování vyžadují napájení a samy vysílají další informace mezi, které patří např. úroveň nabití baterie. Jejich výhodou je také schopnost být ve spojení s anténou na vzdálenost několik stovek metrů až kilometrů. Při jejich použití lze využít frekvenční pásmo více tagy zároveň.

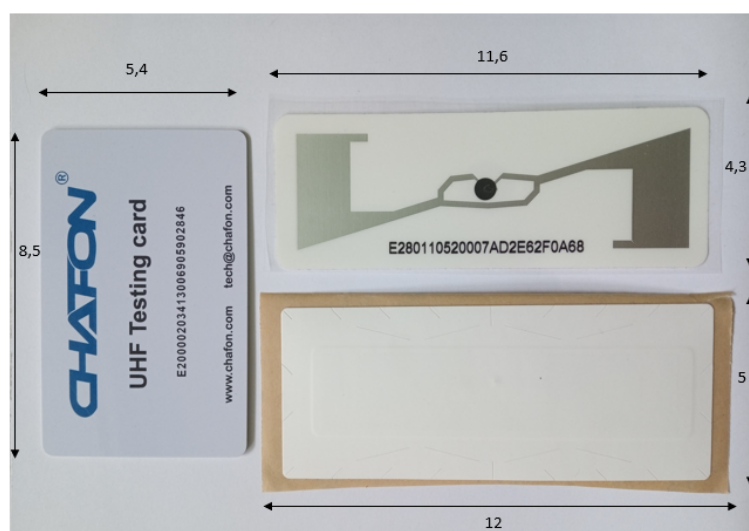
2.3.2 Pasivní čipy

Jedná se o čipy, které pro své fungování nepotřebují vlastní napájení. Některé mohou disponovat zapisovatelnou pamětí. Pasivním tagem se rozumí usměrňovací obvod složený z antény, několika diod, kondenzátorů, logického obvodu a tranzistoru FET (obr. 2.4). Při použití těchto čipů se využívá toho,



Obrázek 2.4: Pasivní tag [1]

že čtečka vysílá signál a usměrňovací obvod tagu tento signál anténou zachytí a zmoduluje ho. FET tranzistor zde plní funkci spínače. V první poloze je sepnutý, tzn. anténa je uzemněna. V druhé poloze je rozepnutý, anténou tagu prochází malý proud a modulovaný signál je směřován do okolí tagu, kde je registrován čtečkou. Ve čtečce dochází k demodulaci signálu obvodem, který je velmi podobný tomu usměrňovacímu. Různé podoby UHF tagů jsou vyobrazeny na obr. 2.5 (uvedené rozměry jsou v cm) - vlevo testovací UHF karta (velikost odpovídá platební kartě), vpravo dvě samolepky, vše součástí balení se čtečkou.



Obrázek 2.5: Různé podoby UHF tagu

■ Použití

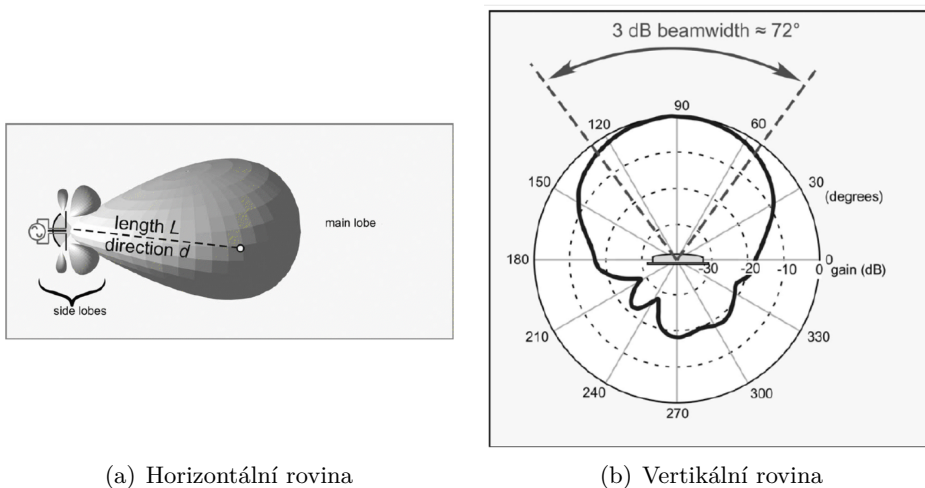
Kvůli jejich nízké ceně je lze používat na závodech, kde startuje velké množství lidí a nemusí se vracet zpět pořadateli. Lze je také využít pro identifikaci zboží ve skladech a mít přehled třeba o skladových zásobách. Naopak registrovat jejich pohyb lze pouze na jednotky metrů. Umístění čipu je závislé na typu použití nebo na sportu při, kterém je použit. Umístění na řídkách nebo na zádech je typické pro cyklozávodce. Naopak nejvhodnější umístění pro běžecké disciplíny je na hrudi (v případě čipu připevněném na startovním čísle) případně na noze.

■ 2.3.3 Použití RFID

Tato technologie nachází široké využití v praxi. Ve sportovním odvětví, jako identifikátor dané osoby, jejíž pohyb je díky čipu registrován, ale také v dalších oblastech, např. sport, obchod, doprava (identifikace vlakových vagonů), a přístupové systémy [1].

■ 2.3.4 Anténa

Jednou ze základních charakteristik antény je směrovost (viz obr. 2.6). Směrovost vyjadřuje do kterých směrů anténa nejvíce a nejméně vyzářuje. Znalost této vlastnosti využijeme při nastavení konfigurace pro daný účel. Čím je výseč vyzářování v horizontální rovině užší, tím je směrovost lepší. Ve vertikální rovině požadujeme, aby hlavní lalok byl největší a postranní laloky nejmenší, nejlépe žádné.



Obrázek 2.6: Směrovost antény[1]

Anténa pracuje buď v režimu half-duplex nebo full-duplex. Half-duplex představuje režim, ve kterém jsou data vysílána nebo přijímána, naopak v režimu full-duplex je možný tok dat oběma směry zároveň [1].

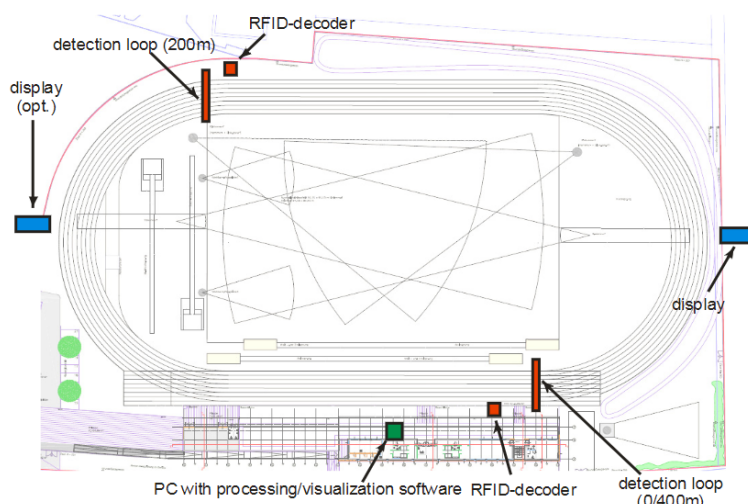
Kapitola 3

Dostupné časomíry

V této části jsou jednotlivá řešení již dostupných časomír představena. Převážně se jedná o řešení vhodná pro závodní účely.

3.1 Trvalé řešení

Jedním ze systémů využívající RFID technologii je měřicí systém jehož prvky je nutno zabudovat již při stavbě stadionu [4]. Tato technologie využívá detekční smyčky umístěné přímo pod cílovou čarou. Další smyčka je umístěna na opačné straně stadionu, na startu 200m trati. Na stadionu jsou také umístěny displeje pro zobrazování mezičasů, každý v polovině zatáčky (viz obr. 3.1). Kvůli vyšším vzdálenostem se využívá propojení jednotlivých prvků



Obrázek 3.1: Konfigurace trvalého řešení [4]

optickými kabely případně pomocí bezdrátového WIFI připojení. Závodník má umístěný čip na kotníku, na hrudi nebo na bocích. Nejvhodnější je umístění na kotníku, aby byla dodržena doporučená vzdálenost 0,9 m mezi čipem a detekční smyčkou.

3.1.1 Výhody trvalého řešení

Řešení je implementováno přímo pod tartanovou dráhu. Techniku lze využívat i na trénincích a na displejích zobrazovat mezičasy nebo časy ostatních účastníků tréninku.

3.1.2 Nevýhody trvalého řešení

Již při samotném pokládání tartanu je nutné vědět, zda se bude instalovat tato technologie. Použité zařízení lze používat pouze na jednom stadionu, nelze jej přesunout na jiný.

3.2 Rámové řešení

Další možností je umístění antén a čtečky na rám, kterým závodníci probíhají. Dvě antény jsou umístěny na rámu a nakloněny tak, aby svíraly 45° úhel se zemí (obr. 3.2). Rám, o výšce 2,3 m, plní funkci držáku komponent systému (dvě antény a čtečka) [5].



Obrázek 3.2: Rámové řešení [5]

3.2.1 Výhody rámového řešení

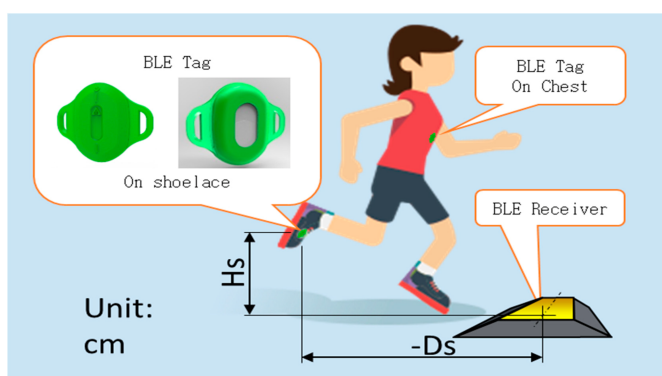
Mezi výhody patří fakt, že čtečka a antény jsou umístěny přímo na rámu a není potřeba řešit jejich umístění. Umístění antén a čtečky blízko sebe umožňuje jednoduché zapojení bez speciálních nároků na délku kabelů. Konfigurace je vhodná do městského prostředí a zároveň vymezuje prostor, kudy mají běžci běžet. Toto řešení je použitelné pro širokou škálu sportů, např. běh, cyklistika,...

3.2.2 Nevýhody rámového řešení

Jedinou nevýhodou tohoto řešení je možnost instalovat tento rám pouze do prostředí s dostatkem místa.

3.3 BLE Technologie

Tato technologie je velmi podobná technologii RFID. Systém se skládá z BLE tagů, které mají běžci na sobě a čteček rozmístěných podél trasy běhu v 500m vzdálenostech, na startu a v cíli (obr. 3.3). Přijímač, který registruje přítomnost tagu, je umístěn v panelu, přes který je nutné, aby běžec přeběhl. Tagy, použité v této technologii, se řadí mezi aktivní čipy, neboť neustále vysílají informace obsahující startovní číslo běžce a úroveň baterie. Propojení komponent zajišťuje síť 4G LTE [6].



Obrázek 3.3: BLE technologie [6]

3.3.1 Výhody BLE technologie

Tuto technologii lze využít při bězích na dlouhé vzdálenosti, kde startuje velké množství lidí. Přijímače registrující průběh startem lze zapojovat do série a díky tomu lze zaznamenávat průběh až 20 osob za sekundu [6]. Další výhodou je umístění přijímače v panelu, který slouží jako ochrana před vnějšími vlivy a zároveň je barevně označen, aby byl dobře vidět.

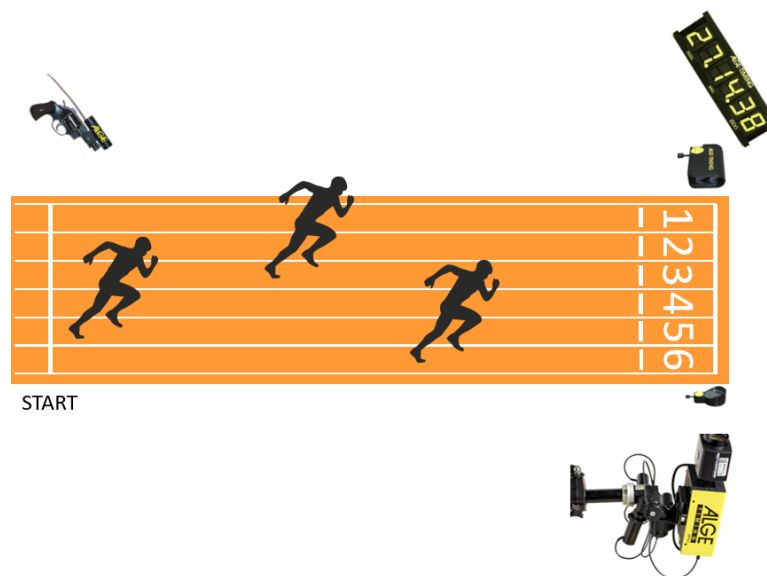
3.3.2 Nevýhody BLE technologie

Nevýhodu této technologie shledávám v nutnosti napájení tagu a případné výměně napájecího zdroje. K nevýhodám řadím také vyšší cenu za jeden tag.

3.4 Závodní časomíra

Tento odstavec bych rád věnoval časomíře používané při atletických závodech, neboť atletikou se zabývá již několik let a na některých závodech obsluhuji

cílovou kameru. Závodní systém se skládá z cílových fotobuněk, kamery a startovací pistole 3.4. Rázová vlna vyvolaná startovním výstřelem z pistole je signál pro spuštění časomíry. Zastavení časomíry probíhá na základě pohybu přes cílovou čáru. Orientační čas je zobrazován na připojeném šesticiferném displeji, přesnější čas je určen ze záznamu z kamery, která registruje všechny pohyb přes cílovou čáru. Ač se používá označení kamera, výsledkem není video, ale obrázek, ze kterého lze pomocí softwaru určit pořadí a čas doběhu závodníka do cíle.



Obrázek 3.4: Závodní časomíra (obrázek startovací pistole převzat z [7], displeje z [8], fotobuněk z [9] a cílové kamery z [10])

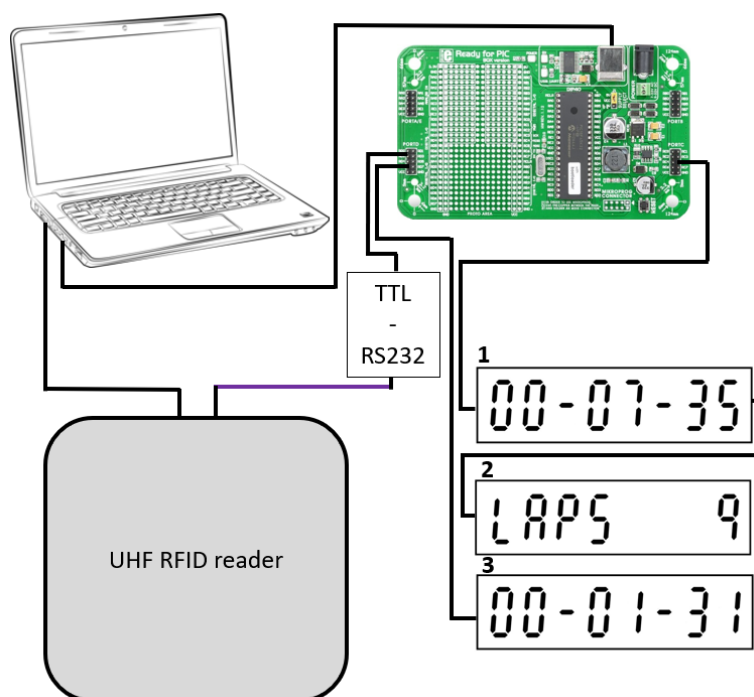
3.4.1 Výhody a nevýhody

Mezi největší výhody tohoto systému patří jeho přesnost, která umožňuje toto zařízení používat při oficiálních soutěžích a měřit časy s přesností na jednu setinu sekundy. Výhodou je určitě dostatečně velký displej, jehož čitelnost není snížena ani přímým slunečním zářením, zato nemá podsvícení, které by umožňovalo využití zařízení i za snížených světelných podmínek. K nevýhodám patří složitá instalace systému vyžadující zhruba 30 minut. Obsluhu musí být svěřena proškolené obsluze.

Kapitola 4

Mé řešení atletické časomíry

Pro mé řešení jsem použil RFID čtečku, RFID tag a mikrokontroler na desce od firmy MikroElektronika (obr. 4.1). K desce jsem připojil tři displeje pro



Obrázek 4.1: Schéma sestavené časomíry (obrázek notebooku použit pod licencí Creative Commons [11], obrázek desky z [12])

zobrazování času posledního kola, celkového času a počtu zbývajících kol do konce daného úseku. Počet úseků v tréninku a počet kol úseku je nutné nastavit před zahájením tréninku. Každý displej se skládá z osmi cifer. Každou cifru tvoří 8 segmentů. Součástí každého displeje je integrovaný mikrokontroler s řadičem MAX7219, který umožňuje kaskádové zapojení displejů 4.1 a řízení pomocí sběrnice SPI. Čtečka s deskou jsou napájeny z notebooku pomocí USB portu. Čtečka je připojena k převodníku TTL-RS232 propojovacím sériovým kabelem se zakončením DB9 - male na obou stranách. Převodník je propojen s deskou čtyřmi vodiči (VCC, GND, RXD a TXD). Propojení

Tabulka 4.1: Propojení displeje č.1 a č.2

pin displeje č.1	pin displeje č.2
VCC	VCC
GND	GND
DOUT	DIN
LODA	CS
CLK	CLK

displejů a převodníku jsem uvedl do tabulky 4.2.

Tabulka 4.2: Propojení pinů desky s displeji a čtečkou

	pin na desce	pin na připojeném zařízení
displej č.1	VCC	VCC
	GND	GND
	RC5	DIN
	RC0	CS
	RC3	CLK
displej č.3	VCC	VCC
	GND	GND
	RD7	DIN
	RD4	CS
převodník RS232-TTL	RD0	CLK
	VCC	VCC
	GND	GND
	RD7	RXD
	RD6	TXD

4.1 RFID čtečka

Použil jsem čtečku s udávaným rozsahem 5-6 m v závislosti na použitém tagu [13]. Zisk antény dle výrobce činí 6 dBi. Čtečka disponuje 5 konektory, z toho dva jsou určeny pro napájení. Jeden je možné přes adaptér připojit k síťovému napájení a druhý lze propojit s USB portem počítače. Zbývající konektory slouží pro přenos dat. Jedná se o konektor RS232, relay a WG26.

4.1.1 Tag

V mém projektu jsem využil UHF RFID kartu dodávanou s RFID čtečkou a čip na startovním čísle z absolvovaných závodů. Všechny tyto čipy se řadí do skupiny pasivních čipů.

4.1.2 Mikrokontroler

Zařízení, které celý systém řídí se nazývá mikrokontroler. Jedná se o mikro-počítač na čipu s integrovanými komponentami jako jsou operační paměť, paměť ROM, časovače, CPU, vstupní a výstupní porty [14]. Mezi dostupné periferie patří rozhraní USART a SPI, které se zde vyskytují dvakrát [15]. Obě SPI rozhraní jsem v projektu využil pro displeje, první rozhraní USART je použito pro nahrávání programu do mikrokontroleru a druhé je využito pro připojení RFID čtečky. Na desce od firmy MikroElektronika byl použit mikrokontroler PIC18F45K22. V mém návrhu jsem použil mikrokontroler PIC18F24K22, z důvodu nízké dostupnosti mikrokontrolerů PIC na trhu. Tento mikrokontroler disponuje téměř totožnými periferiemi jako ten umístěný na desce, rozdíl lze najít ve velikosti paměti, která je poloviční (16 kB). Také se liší v zapojení jednotlivých periferií.

Projekt byl napsán a kompilován v programu MikroC pro PIC v jazyce C. Tento program nám zkompilováním vytvoří soubor s příponou .hex, který se nahrává pomocí softwaru mikroBootloader. Data jsem odesílána z počítače USB-B kabelem přes převodník UART - USB.

Kód z mikrokontroleru

Zde jsem uvedl důležité části z kódu práce, které představují základní nastavení pinů mikrokontroleru, displejů a komunikace mezi čtečkou a mikrokontrolerem. Pro některé činnosti, které se často opakují jsem si vytvořil zvlášť funkce. Některé z nich zde uvedu. Celý kód se nachází v příloze A.

V první části jsem využil již definovaných názvů registrů a funkcí a k nim přiřazených adres, kde se nachází [16]. V prvním řádku došlo k přiřazení adresy k názvu registru pro nastavení počtu zobrazených číslic. V druhém řádku byla k názvu instrukce přiřazena adresa instrukce. Touto instrukcí dojde k zobrazení všech číslic na displeji. Např. pro nastavení počtu zobrazených číslic:

```
#define DECODE_MODE_REG    0x09
#define DECODE_ALL         0xFF
```

První dva řádky slouží pro nastavení portů C a D jako digitálních. Třetím a čtvrtým řádkem jsem nastavil, zda port nebo jednotlivé piny na něm budou vstupní - 1 nebo výstupní - 0.

```
ANSELB = 0;    // PORTC digital
ANSELD = 0;    // PORTD digital
TRISC = 0b00000000;    // PORTC - output
TRISD = 0b11111111;
```

$$\text{Desired baud rate} = \frac{F_{\text{OSC}}}{64 \cdot (\text{SPBRGH} : \text{SPBRG} + 1)} \quad (4.1)$$

$$\text{SPBRGH} : \text{SPBRG} = \frac{F_{\text{OSC}}}{64 \cdot \text{Desired baud rate}} - 1 \quad (4.2)$$

Ze vztahu 4.1 pro nízkou ($SPBRGH2 = 0$) přenosovou rychlost [17] jsem si vyjádřil na jednu stranu $SPBRGH:SPBRG$ a dosadil frekvenci oscilátoru 32 MHz (4x8 MHz) [15]. Požadovanou rychlost přenosu jsem nastavil na stejnou, na které pracuje RFID čtečka, tedy 9600 baud. Vypočítanou hodnotu $SPBRGH:SPBRG$ jsem převedl z decimální soustavy do hexadecimální a nastavil tak hodnotu $SPBRG2$ bitu. Nastavení transmit status a control registru jsem provedl na čtvrtém řádku. Pátým bitem ($TXEN$) jsem povolil vysílání po UART rozhraní, ostatní bity jsem nechal na hodnotě 0. Receive status a control register jsem nastavil v dalším řádku. Sedmým a čtvrtým bitem jsem povolil použití sériového portu a příjmu dat po UART rozhraní.

```
SPBRG2 = 0x33;
SPBRGH2 = 0;
BAUDCON2 = 0;
TXSTA2 = 0b00100000;
RCSTA2 = 0b10110000;
```

Abych mohl nastavit požadované parametry displeje, využil jsem funkci `MAX7219_write`, která použije adresu registru `reg1` a adresu funkce `data1`, ještě před tím je nutné odeslat dvakrát No-op hodnotu (`0x00`), která nám zajistí ovládní prvního připojeného displeje (více o nastavení ovládaného displeje v kapitole 4.1.3).

```
void MAX7219_write(char reg1, char data1)
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(0x00);
    SPI1_write(0x00);
    SPI1_write(reg1);
    SPI1_write(data1);
    MAX7219_CS_LOAD = 1;
}
```

Pro zobrazení číslice na displeji jsem si vytvořil funkci `write_number` (funkce `write_laps`, `write_letter` a `write_dash` fungují podobně), jejíž vstupní hodnoty jsou `position` - pozice na displeji, na kterou chci zobrazovat a `number` - zobrazované číslo.

```
void write_number(int position, int number)
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(0x00);
    SPI1_write(0x00);
    SPI1_write(position);
    SPI1_write(number);
    MAX7219_CS_LOAD = 1;
}
```

Zobrazení mezičasu kola na třetím displeji zajišťuje následující funkce

(nejdříve je displej zapnut a poté jsou postupně zobrazeny všechny cifry času, na třetí a šesté pozici je zobrazena pomlčka):

```
void show_lap_time(int seconds, int seconds_tens,
int minutes, int minutes_tens, int hours, int hours_tens)
{
    Display_On_3();
    write_number_3(1,seconds);
    write_number_3(2,seconds_tens);
    write_dash_3(3);
    write_number_3(4,minutes);
    write_number_3(5,minutes_tens);
    write_dash_3(6);
    write_number_3(7,hours);
    write_number_3(8,hours_tens);
}
```

V následující části dochází k ověření, zda se na předchozí pozici času vyskytuje číslice 0. Pokud ano, hodnota 1 je zapsána do pole s názvem `check`. Pokud toto 6místné pole obsahuje alespoň 5 jedniček (počet jedniček v poli `check` je uložen v proměnné `num`) pak je jedna z podmínek pro ukončení pauzy je splněna.

```
if(h_tens_3 == 0)
{
    check[0] = 1;
    if (h_3 == 0)
    {
        check[1] = 1;
        if (m_tens_3 == 0)
        {
            check[2] = 1;
            if (m_3 == 0)
            {
                check[3] = 1;
                if(s_tens_3 == 0)
                {
                    check[4] = 1;
                    if (seconds_3 == 0)
                    {
                        check[5] = 1;
                    }
                }
            }
        }
    }
}
```

Obsluhu přerušení nebo-li odklon od činnosti programu zajišťuje funkce interrupt. Po provedení této obsluhy se pokračuje ve vykonávání zbývající části programu v místě, kde došlo k přerušení.

V první části obsluhy přerušení je registrován příjem dat na rozhraní UART. Pokud jsou nějaká data přijata na druhém rozhraní UART, pak jsou přečtena funkcí `UART2_Read_Text()` s argumenty `ID_Read`, `"\n"` a `32`. Prvním argumentem je pole, do kterého je text uložen. Druhým argumentem je oddělovač dat. Funkce ukládá znaky dokud nenarazí na tento symbol, v mém případě konec řádku. Posledním argumentem je počet znaků, ve kterých lze očekávat oddělovač. Použil jsem číslo `32`, protože sekvence potvrzující přítomnost čipu je právě `32` znaků dlouhá. Dále následuje porovnání se sekvencí, která je unikátní pro můj čip. Pokud se všechny znaky shodují, pak je hodnota proměnné `b` rovna `30` (poslední dva znaky ze sekvence se liší, ověřovat lze pouze pomocí prvních `30` znaků) a nastává obsluha přerušení. Dále se ověřuje fáze tréninku dle proměnné a nutné podmínky (tabulka 4.3).

Před první fází se na displeji zobrazí čas `00-00-00` a čeká se na průběh cílem, pak nastává první fáze - první úsek tréninku.

Tabulka 4.3: Fáze tréninku

číslo	fáze tréninku	podmínka
1.	start (první úsek)	start = 1, pause = 0
2.	běh (úsek)	pause = 0
3.	mezičas	laps > 1, stop = 1
4.	pauza	pause = 1
5.	konec tréninku	tot = 0

```
void interrupt()
{
    if (UART2_Data_Ready() == 1)
        UART2_Read_Text(ID_Read, "\n", 32);

    for(i = 0; i < 30; i++)
    {
        if(ID_Read[i] == ID[i])
            b++;
    }
    if(b == 30)
    {
        if(start == 0)
        { // start training
            start = 1;
            l--;
        }
        else if (pt == 1 && pause == 0)
        { // minimal time for one lap is set one minute,
            l--; // decrease number of laps
        }
    }
}
```

```

h_tens_lap = h_tens_2, h_lap = h_2,
m_tens_lap = m_tens_2, m_lap = m_2,
s_tens_lap = s_tens_2, seconds_lap = seconds_2;
h_tens_2 = 0, h_2 = 0, m_tens_2 = 0, m_2 = 0,
s_tens_2 = 0, seconds_2 = 0;
stop = 1;
pt = 0;
if(l == -1) // if the lap of distance is last
{
    pause = 1;
    tot--;
}
}
else if(pause == 1 && num > 4 && seconds_3 < 5)
{ // end of pause?
    num = 0;
    stop = 0;
    h_tens_2 = 0, h_2 = 0, m_tens_2 = 0, m_2 = 0,
    s_tens_2 = 0, seconds_2 = 0;
    pause = 2;
    tot--;
    l = laps;
    l--;
}
}
b = 0;
RC2IF_bit = 0;
}

```

4.1.3 Displej

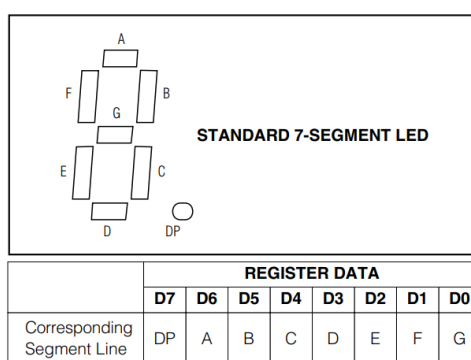
Pro zobrazení údajů jsem použil tři 7segmentové displeje, každý s osmi ciframi s řadičem MAX7219. Při volbě vhodného displeje jsem se rozhodoval, zda všechny požadované údaje zobrazovat na jediném (velkém) displeji nebo každý údaj na samostatném menším displeji. Zvolil jsem více displejů, a to z důvodu jednoduchosti zapojení a ceny pořízení. Displej pracuje ve dvou režimech zobrazování znaků, tzv. code nebo decode.

V režimu Code se používá tzv. B-Code (obr. 4.2), kdy je přesně definovaná sada znaků, které lze zobrazit (čísla 0-9, písmena E, F, L, H, P a -). Naopak v režimu Decode lze rozsvěcet libovolné segmenty a to s využitím zápisu 8bitového čísla (viz obr.4.3). Např. pro rozsvícení čísla 0, je nutné rozsvítit segmenty označené písmeny A-F, bitový zápis by vypadal takto: 0b01111110. Osmý segment tvoří desetinnou tečku a je rozsvěcen osmým bitem.

Data do displeje jsou posílána s každou náběžnou hranou signálu CS ve formě 16bitového slova. V případě kaskádního zapojení dvou displejů, se navíc odesílá tzv. No-Op hodnota. Jedná se o 16bitově zapsanou nulu. Pro

7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6-D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

Obrázek 4.2: B-Code [18]



Obrázek 4.3: 7segmentový displej [18]

odeslání dat do displeje zapojeného blíže k desce je nutné odeslat nejdříve No-Op hodnotu a pak teprve požadovaný příkaz, tento displej obdrží námi požadovaný příkaz a druhý obdrží No-Op hodnotu. V případě, že chci odesílat data do vzdálenějšího displeje, odesílám nejprve požadovaný příkaz a poté No-Op hodnotu. Při tvorbě kódu, který jsem nahrával do mikrokontroleru, jsem využil kód uvedený v [16].

Zobrazované údaje

Požadované údaje jsou zobrazovány na třech displejích v závislosti na fázi tréninku. Z toho dva jsou zapojeny kaskádně k prvnímu rozhraní SPI a třetí displej je připojen ke druhému rozhraní SPI. Celý trénink jsem rozdělil na několik fází. První a poslední fáze proběhne vždy pouze jednou ostatní se mohou opakovat vícekrát. Třetí fáze nemusí nastat vůbec, pokud je počet kol roven 1.

V první a druhé fázi se na prvním displeji zobrazuje celkový čas úseku. Je odstartován prvním pohybem čipu kolem RFID čtečky a zastaven posledním průchodem kolem RFID čtečky. Na druhém displeji též připojeném k prvnímu SPI rozhraní se zobrazuje počet zbývajících kol do konce daného úseku.

Mezičas je zobrazen pouze pokud byl počet kol nastaven na hodnotu větší

než 1. Po 5 s tento čas přestane svítit a počet kol je snížen o 1 tak, aby závodník probíhající cílem znal počet kol zbývajících do konce daného úseku.

Fáze pauzy nastává, když běžec doběhne do cíle posledního kola úseku a zároveň je hodnota proměnné $tot > 0$. Fáze pauzy může být ukončena během posledních 5 s pauzy, případně i později.

Konec tréninku nastává ve chvíli, kdy je hodnota proměnné tot rovna 0, poté se zobrazí čas posledního úseku, případně i posledního kola (pokud je hodnota $laps > 1$). Druhý displej zobrazí vodorovnou linku značící konec tréninku.

4.1.4 Měření, otestování časomíry a čtečky

Nejprve jsem testoval čtečku s čipy v různých pozicích (testovací karta umístěná v kapse sportovních kalhot, čip se startovním číslem umístěný na hrudi a na boku). S každým čipem jsem proběhl 20x kolem čtečky ve vzdálenosti 50 cm. Do tabulky 4.4 jsem uvedl v kolika případech z 20 byl čip registrován. V dalším kroku jsem vyzkoušel do jaké vzdálenosti je čtečka schopna čip

Tabulka 4.4: Vliv umístění čipu na počet úspěšně registrovaných průběhů cílem

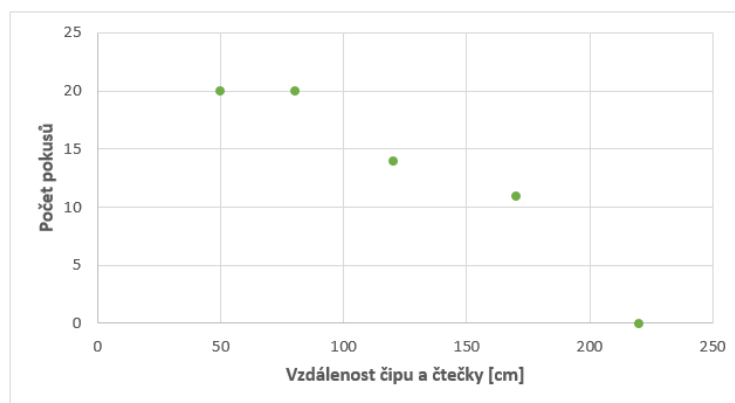
umístění čipu	počet úspěšných pokusů
testovací karta v kapse	18
startovní číslo s čipem na hrudi	10
startovní číslo s čipem na boku	20

ještě zaregistrovat. Pro každou vzdálenost (50, 80, 120, 170 a 220 cm) jsem 20x proběhl kolem čtečky. Během tohoto testování jsem měl čip umístěný na boku, jelikož se toto umístění na základě předchozího měření jeví jako nejspolehlivější. Naměřená data jsem uvedl do tab. 4.5 a do grafu 4.4.

Tabulka 4.5: Počet úspěšně registrovaných průběhů

vzdálenost čtečky a tagu [cm]	počet úspěšných průběhů
50	20
80	20
120	14
170	11
220	0

Časomíru jsem testoval během atletického tréninku 10x400 m. Umístil jsem ji k cílové čáře ve výšce cca 80 cm a namířil směrem do cílové rovinky tak, aby čtečka a cílová čára spolu svíraly úhel o velikosti 45°. Vzdálenost mezi čipem a čtečkou tak byla zhruba 50 cm. Test proběhl s číslem umístěným na boku. Naměřené časy ve tvaru M:SS pomocí časomíry a sportovních hodinek jsem uvedl do tabulky 4.6. Dále jsem uvedl rozdíl těchto dvou časů v sekundách. Čas změřený pomocí hodinek jsem zaokrouhloval na celé sekundy.



Obrázek 4.4: Graf počtu úspěšně zaregistrovaných pohybů čipu kolem čtečky v závislosti na vzdálenosti mezi čipem a čtečkou

Tabulka 4.6: Test časomíry při tréninku

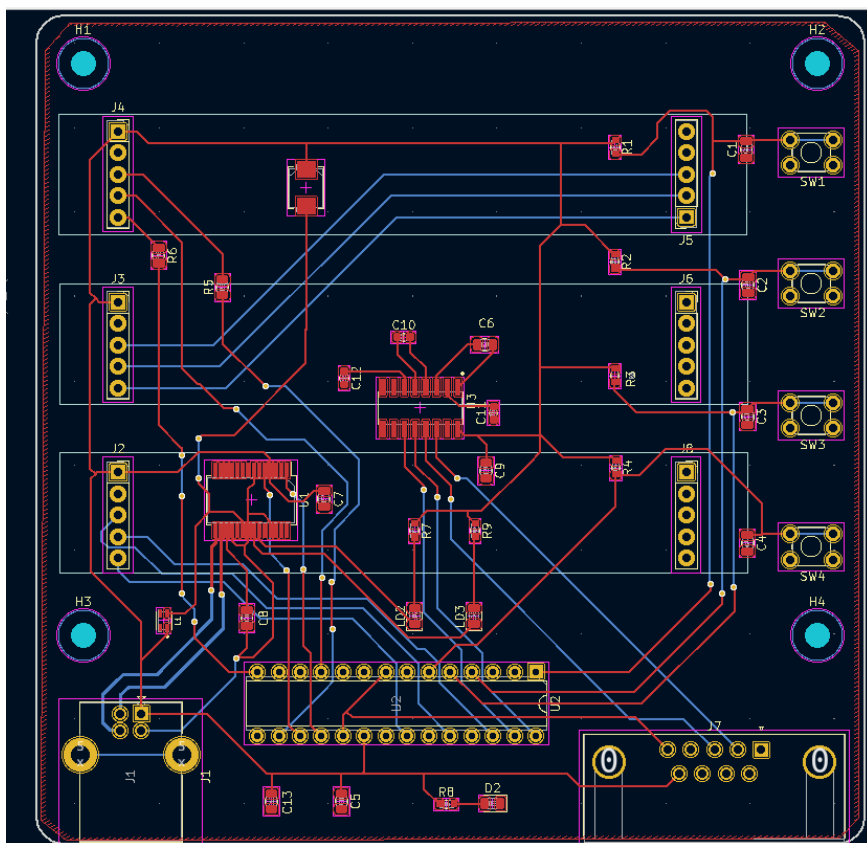
úsek č.	čas změřený hodinkami	čas určený časomírou	rozdíl
1	1:23	1:22	-1
2	1:33	1:32	-1
3	1:29	1:28	-1
4	1:28	1:28	0
5	1:28	1:27	-1
6	1:32	1:31	-1
7	1:29	1:28	-1
8	1:28	1:28	0
9	1:29	1:29	0
10	1:30	1:30	0

4.1.5 Deska

Pro návrh vlastní desky jsem se rozhodl využít software KiCad. Při návrhu jsem se inspiroval zapojením mikrokontroleru a dalších komponent na desce od firmy Mikroelektronika (schéma [19]). Všechny komponenty jsem umístil na desku 4.5, tak aby se k desce připojilo napájení pomocí portu USB (konektor USB-B) a s pomocí sériového kabelu UHF RFID čtečka.

V levé horní části se nachází 6 konektorů pro tři 7segmentové displeje, které se připojí pomocí konektorů typu dutinka. Vpravo od displejů jsou umístěny čtyři tlačítka pro nastavení a resetování programu. Pod displeji se nachází dva převodníky, jeden pro převod USB-UART (FT232RL) a druhý pro převod mezi sériovou linkou a technologií TTL (MAX3232EIDR). Níže pod displej jsem umístil mikrokontroler PIC18F24K22 (v návrhu desky označení U2). Ve spodní části desky se nachází vlevo konektor USB-B a vpravo konektor RS232. V desce jsem nechal vytvořit 4 otvory pro případné připevnění k podkladu (v návrhu jsou označeny modrou barvou a písmeny H1-H4).

Při návrhu jsem zvolil použití dvou vrstev mědi. V první vrstvě jsem nechal rozlít GND signál a propojit napájení. V druhé vrstvě jsem propojil zbývající



Obrázek 4.5: Návrh desky v prostředí KiCad

spoje. Některé komponenty jsou typu SMD. To znamená, že jsou připevněny pomocí povrchové montáže pájením. Ostatní komponenty jsou typu THT. Jsou také pájené, ale upevněné skrz desku. Rezistory, kondenzátory a diody mají pouzdra s označením 0805, v této velikosti jsou vhodné k SMD pájení.

Desku jsem si nechal vyrobit firmou JLCPCB. Komponenty jsem si osazoval sám za pomoci vedoucího. Fotky osazené desky jsou k dispozici v příloze B.

Seznam použitých součástek: mikrokontroler PIC18F24K22, konektor USB-B, konektor RS232 (DB9) - female, MAX3232EIDR, FT232RL, tlumivka DLW21SN121SQ2L, DIP 28, schottkyho dioda MBRS140T3, tlačítko tactile x4, 100nF kondenzátor x8, 10 μ F kondenzátor, 0,47 μ F kondenzátor x3, konektor dutinka s roztečí 2,54 mm x6, LED x3, 10k Ω rezistor x4, 27 Ω rezistor x2, 2,2k Ω rezistor x3.

4.2 Použité typy komunikace

V práci jsem využil tyto typy komunikace: USART, SPI a USB. USART komunikace je použita pro propojení RFID čtečky a desky s mikrokontrolerem a v kombinaci s USB technologií také jako propojení desky a počítače, které slouží pro nahrávání dat do mikrokontroleru. SPI komunikace slouží ke komunikaci mezi deskou a displeji. Níže jsou tyto technologie detailně

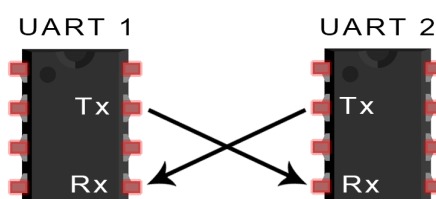
popsány.

■ 4.2.1 USART

Základní vodiče sběrnice:

- **TX** - vodič pro odesílání dat.
- **RX** - vodič pro přijímání dat.
- **GND** - vodič pro uzemnění, sloužící také jako referenční.

USART představuje sériovou komunikaci pro propojení tzv. embedded zařízení a počítače nebo více embedded zařízení mezi sebou. Nejjednodušší variace vyžaduje tři vodiče TX, RX a GND. Rozlišujeme dvě varianty sériové komunikace. Synchronní označovanou jako USART, kde se kromě základních vodičů používá vodič pro synchronizaci přenosu dat [20]. Pro propojení dvou zařízení je nezbytné propojení vodičů křížem tak, aby vodiče jednoho zařízení propojovaly opačné piny druhého zařízení (dle obr.4.6). Základní nastavení



Obrázek 4.6: Propojení zařízení přes UART [21]

rychlosti přenosu je 9600 baud, žádný parity bit, 8 datových bitů a 1 stop bit. První parametr představuje rychlost přenosu dat, obvykle se jedná o hodnotu 9600, ale je možné se setkat i s dalšími, např. 300, 600, 1200, 1800, 2400,... Přenos informací mezi čtečkou a deskou se odehrává s rychlostí 9600 baud, naopak komunikace mezi deskou a PC probíhá s rychlostí 115 200 baud.

■ Výhody USART komunikace

- Slouží k přenosu různých typů informace. [22]
- Hardware je cenově dostupný.
- Kabely pro propojení zařízení mohou dosahovat dlouhých délek (několik desítek metrů).
- Použité kabely nemusí být stíněné.
- Komunikace může být jak synchronní, tak asynchronní.

■ Nevýhody USART komunikace

- Lze propojit pouze dvě zařízení.
- Rychlost přenosu je limitovaná hodnotou 20 kbs. [22]

■ 4.2.2 SPI

Propojení displeje a mikrokontroleru je realizováno pomocí sběrnice SPI za použití pinů označených VCC, GND, DIN, CS a CLK. Vodiče VCC a GND slouží k připojení napájecího napětí respektive zemnicího vodiče, pin CS slouží k výběru slave (řízeného) zařízení, tento výběr provádí master (řídící zařízení), pin CLK je připojen na zdroj hodinového signálu, kterým je krystalový oscilátor s frekvencí 8 MHz. Sběrnice umožňuje kaskádní zapojení displejů, tzn. k desce s mikrokontrolerem je připojen pouze první displej, k prvnímu druhý displej, atd. Zařízení master jako jediné může řídit hodinový signál, který slouží k tomu, abychom určili, jak s daty pracovat (upravovat je nebo číst).

■ 4.2.3 USB

Universální sériová sběrnice je rozhraní, které slouží k propojení počítače s periferními zařízeními, jako jsou klávesnice, myši, tiskárny, skenery a další. Rozhraní umožňuje napájení připojených zařízení. Délka kabelu propojující zařízení je limitovaná hodnotou 5 m. Počet připojených zařízení je omezen hodnotou 127. Rozhraní se dále dělí na kategorie USB 1.1, USB 2.0 a USB 3.0. Jednotlivé kategorie se liší především v rychlostech přenosu, který umožňují (např. standard USB 2.0 je schopen přenosu o rychlosti až 480 Mb/s, což je hodnota čtyřicetkrát větší než je maximální rychlost přenosu pomocí standardu 1.1 [23]).



Kapitola 5

Závěr

Ve své práci jsem zkonstruoval časomíru řízenou mikrokontrolerem, která je schopná zobrazovat údaje důležité pro běhání intervalových tréninků. Z výsledků je patrné, že změřený čas se nepatrně liší ve srovnání s časem změřeným pomocí sportovních hodinek, ale stále se jedná o 1 sekundový rozdíl, čip umístěný na boku se dle výsledků jeví jako nejspolehlivější místo pro umístění čipu. Pro atletické tréninky není důležitá až tak přesnost časů, ale právě vytrvalost běžce. Ve srovnání s ostatními časomírami se domnívám, že její výhodou je právě jednoduchost zapojení, kterou zvládne kdokoliv. Ostatní časomíry jsou vhodné především pro závodní účely. Nevýhodou mé časomíry je možnost měřit čas pouze jednomu závodníkovi a nutnost nastavení programu před kompilací a nahráním výsledného souboru do mikrokontroleru.

V dalším vývoji bych se chtěl zaměřit na ukládání naměřených dat, jejich zpracování a porovnání s atletickým plánem. Rád bych také doplnil více mezičasů a možnost startovat z kteréhokoli místa na stadionu (např. pro běhání úseků o délce 1 km).



Seznam použitých zkratk

RFID	radio frequency identification
DPS	deska plošných spojů
UHF	ultra high frequency
PCB	printed circuit board
BLE	bluetooth low energy
FET	field effect transistor
LF	low frequency
HF	high frequency
WIFI	wireless fidelity
LTE	long term evolution
SPI	serial peripheral interface
TTL	transistor-transistor logic
USB	universal serial bus
ROM	read only memory
CPU	central processing unit
UART	universal asynchronous receiver transmitter
USART	universal synchronous asynchronous receiver transmitter
PIC	programmable interface controllers
DIN	data input
CS	chip select
CLK	clock
RXD	receive data
TXD	transceive data
TX	transiever
RX	receiver
THT	through hole technology
SMD	surface mount technology

Literatura

- [1] DOBKIN, Daniel, et al. *The RF in RFID : Passive UHF RFID in Practice*. Burlington: Newnes, 2008. [15. 4. 2022]. ISBN 978-0-7506-8209-1. [online] Dostupné také z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=319166>
- [2] LEHPAMER, Harvey. *RFID Design Principles*. Second Edition. Boston: Artech House, 2012. ISBN 978-1-60807-470-9. [cit. 13. 5. 2022]. Dostupné také z: <https://books.google.rw/books?id=Cm7O1ppKte8C&printsec=frontcover#v=onepage&q&f=false>
- [3] Elektromagnetické Spektrum [online]. Aldebaran Group For Astrophysics [cit. 26. 7. 2022]. In: Aldebaran. Dostupné z: https://www.aldebaran.cz/tabulky/tb_spektrum.php
- [4] WOELLIK H., A. Mueller. J. Herriger. Permanent RFID Timing System in a Track and Field Athletic Stadium for Training and Analysing Purposes. In: *Procedia Engineering* [online]. 2014, Pages 202-207. ISSN 1877-7058. [cit. 12. 1. 2022]. Dostupné z: doi: 10.1016/j.proeng.2014.06.034
- [5] KOLAJA J., J. K. Ehlerova, Effectivity of Sports Timing RFID System, Field Study. In: *IEEE International Conference on RFID Technology and Applications (RFID-TA)*. 2019, p. 220-223 [cit. 5. 5. 2022]. ISBN 978-1-7281-0589-5. Dostupné z: doi:10.1109/RFID-TA.2019.8892108
- [6] SUN, C.-I., J.-T. Huang, S.-C. Weng, M.-F. Chien. City Marathon Active Timing System Using Bluetooth Low Energy Technology. *Electronics* [online] 2019, Volume 8, p. 252 [cit. 5. 5. 2022]. Dostupné z: <https://doi.org/10.3390/electronics8020252>
- [7] Obrázek startovací pistole. In: alge-timing.com [online]. [cit. 3. 7. 2022]. <https://alge-timing.com/alge/download/brochure/SM8+SV4+HS2-2/SM8-PE.pdf>
- [8] Obrázek displeje. In: phoenix-sports.com [online] Phoenix Sports Technology 2022 [cit. 3. 7. 2022]. http://www.phoenix-sports.com/alge_images/gaz4_master.gif

- [9] Obrázek cílových fotobuňek. In: hssports.co.uk [online]. HS Sports Ltd 2020 [cit. 3. 7. 2022]. <https://hssports.co.uk/wp-content/uploads/2020/10/PR1A-Photocell-2.jpg>
- [10] Obrázek cílové kamery [online]. In: alge-timing.com [cit. 3. 7. 2022] https://alge-timing.com/AlgeTiming_V2/assets/images/4/optic3_26-f5f9d154.jpg
- [11] Obrázek notebooku. In: Wisc-online.com [online]. [cit. 3. 8. 2022]. Dostupné z: <https://www.wisc-online.com/assetrepository/viewasset?id=4072>
- [12] Obrázek desky s mikrokontrolerem. In: Microe.com [online]. MikroElektronika 2022. [cit. 3. 8. 2022]. Dostupné z: https://cdn1-shop.mikroe.com/img/product/ready-pic/ready-pic-thickbox_default-1.jpg?v=100
- [13] Informace o UHF RFID čtečce In: Chafon.com [online]. [cit. 17. 5. 2022]. Dostupné z: <http://www.chafon.com/productdetails.aspx?pid=765>
- [14] DAWOUD, Shenouda, R. Peplow. *Digital System Design: Use of Microcontroller*. Volume 2, River Publishers, 2010. ISBN 978-87-92329-40-0. [cit. 10. 5. 2022]. Dostupné také z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=3400111>
- [15] Datasheet mikrokontroleru PIC18F45K22. In: Microchip.com [online]. Microchip Technology 2020. [cit. 10. 4. 2022] Dostupné z: [https://ww1.microchip.com/downloads/en/DeviceDoc/PIC18\(L\)F2X-4XK22-Data-Sheet-40001412H.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/PIC18(L)F2X-4XK22-Data-Sheet-40001412H.pdf)
- [16] MAX7219 Interfacing with 8-digit 7 Segment Display and PIC16F877A. In: microcontrollerslab.com [online]. 2003-2022. [cit. 18. 4. 2022]. Dostupné z: <https://microcontrollerslab.com/max7219-interfacing-8-digit-7-segment-display-pic16f877a/>
- [17] UART Communication with Pic Microcontroller (Programming in MPLAB XC8). In: Microcontrollerslab.com 2013-2022 [online]. [cit. 25. 7. 2022]. Dostupné z: <https://microcontrollerslab.com/uart-communication-pic-microcontroller-mplab-xc8/>
- [18] 7segmentový displej [datasheet]. In: Laskakit [online]. ©2003 Maxim Integrated. [cit. 21. 3. 2022]. Dostupné z: https://www.laskakit.cz/user/related_files/max7219-max7221.pdf
- [19] Schema použité desky s mikrokontrolerem. In: Microe.com [online]. MikroElektronika 2012. [cit. 2. 7. 2022]. Dostupné z: <https://download.mikroe.com/documents/starter-boards/ready-pic/dip28/ready-pic-dip28-schematic-v112.pdf>
- [20] DAWOUD, Shenouda, Peter Dawoud. *Serial Communication Protocols and Standards: RS232/485, UART/USART, SPI, USB, INSTEON, Wi-Fi and WiMAX*. River Publishers, 2020. ISBN 978-87-7022-154-2. [cit.

3. 6. 2022]. Dostupné také z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=6300567>
- [21] Propojení dvou zařízení s využitím UART rozhraní. In: circuitbasics.com[online]. [cit. 12. 5. 2022]. Dostupné z: <https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-UART-Basic-Connection-Diagram.png>
- [22] AXELSON, Jan. *Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems*. Madison: Lakeview Research, 2007. ISBN 978-1931448-06-2. Dostupné také z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=316101>
- [23] Barry, Peter, Patrick Crowley. *Modern Embedded Computing - Designing Connected, Pervasive, Media-Rich Systems - Universal Asynchronous Receiver/Transmitter*. Waltham: Elsevier, 2012. [cit. 12. 5. 2022]. ISBN: 978-0-12-391490-3. Dostupné také z: <https://app.knovel.com/hotlink/pdf/id:kt00BPZB2P/modern-embedded-computing/universal-asynchronous>

Příloha A

Zdrojový kód

```
#define DECODE_MODE_REG 0x09
#define BRIGHTNESS_REG 0x0A
#define SCAN_LIMIT_REG 0x0B
#define DISPLAY_REG 0x0C
#define TEST_REG 0x0F
#define No_Op_Reg 0x00
#define DISABLE_DECODE 0x00
#define DECODE_ALL 0xFF
#define DECODE_LAPS 0x01
#define BRIGHTNESS 0x0F
#define SCAN_ALL_DIGITS 0x07
#define TEST_ON 0x01
#define TEST_OFF 0x00
#define DISPLAY_ON 0x01
#define DISPLAY_OFF 0x00

sbit MAX7219_CS_LOAD at RC0_bit;
sbit MAX7219_CS_LOAD_2 at RD5_bit;
sbit MAX7219_CS_LOAD_DIRECTION at TRISC0_bit;
sbit MAX7219_CS_LOAD_2_DIRECTION at TRISD5_bit;

int i, j=0, start = 0, stop = 0, pause = 0, num = 0, l = 0, x = 0,
pt = 0, tot = 0;
int h_tens = 0, h = 0, m_tens = 0, m = 0, s_tens = 0, seconds = 0;
// time format (h_tens h ) - (m_tens m) - (s_tens seconds)
int h_tens_2 = 0, h_2 = 0, m_tens_2 = 0, m_2 = 0, s_tens_2 = 0,
seconds_2 = 0; // variables of lap time int h_tens_lap = 0, h_lap = 0,
m_tens_lap = 0, m_lap = 0, s_tens_lap = 0, seconds_lap = 0;
int h_tens_3 = 0, h_3 = 0, m_tens_3 = 0, m_3 = 0, s_tens_3 = 0,
seconds_3 = 0; // pause time
int check[6]={0,0,0,0,0,0};
unsigned char ID[] = {0x00, 0x54, 0x00, 0x1C, 0x00, 0x45, 0x01, 0xC3, 0x87,
0x21, 0x07, 0x28, 0x5F, 0xF6, 0x01, 0x0F, 0x01, 0x01, 0xE2, 0x00, 0x00,
0x20, 0x34, 0x13, 0x00, 0x69, 0x05, 0x90, 0x28, 0x46};
```

```
unsigned char ID_read[32];
int b = 0;

// settings of training
int laps = 2; // number of laps in one distance
int count = 2; // number of distances
int h_tens_pause = 0, h_pause = 0, m_tens_pause = 0, m_pause = 0,
s_tens_pause = 2, seconds_pause = 0; // duration of pause

void MAX7219_write(char reg1, char data1)
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(0x00);
    SPI1_write(0x00);
    SPI1_write(reg1);
    SPI1_write(data1);
    MAX7219_CS_LOAD = 1;
}
void MAX7219_write_2(char reg1, char data1)
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(reg1);
    SPI1_write(data1);
    SPI1_write(0x00);
    SPI1_write(0x00);
    MAX7219_CS_LOAD = 1;
}
void MAX7219_write_3(char reg1, char data1)
{
    MAX7219_CS_LOAD_2 = 0;
    SPI2_write(reg1);
    SPI2_write(data1);
    MAX7219_CS_LOAD_2 = 1;
}
void MAX7219_init()
{
    MAX7219_write(DECODE_MODE_REG,DECODE_ALL);
    MAX7219_write(BRIGHTNESS_REG,BRIGHTNESS);
    MAX7219_write(SCAN_LIMIT_REG,SCAN_ALL_DIGITS);
    MAX7219_write(DISPLAY_REG,DISPLAY_ON);
    MAX7219_write(TEST_REG,TEST_OFF);
}

void MAX7219_init_2()
{
    MAX7219_write_2(DECODE_MODE_REG,DECODE_LAPS);
```

```
MAX7219_write_2(BRIGHTNESS_REG,BRIGHTNESS);
MAX7219_write_2(SCAN_LIMIT_REG,SCAN_ALL_DIGITS);
MAX7219_write_2(DISPLAY_REG,DISPLAY_ON);
MAX7219_write_2(TEST_REG,TEST_OFF);
}
void MAX7219_init_3()
{
    MAX7219_write_3(DECODE_MODE_REG,DECODE_ALL);
    MAX7219_write_3(BRIGHTNESS_REG,BRIGHTNESS);
    MAX7219_write_3(SCAN_LIMIT_REG,SCAN_ALL_DIGITS);
    MAX7219_write_3(DISPLAY_REG,DISPLAY_OFF);
    MAX7219_write_3(TEST_REG,TEST_OFF);
}
void Display_On()
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(DISPLAY_REG);
    SPI1_write(DISPLAY_ON);
    MAX7219_CS_LOAD = 1;
}
void Display_On_2()
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(DISPLAY_REG);
    SPI1_write(DISPLAY_ON);
    SPI1_write(0x00);
    SPI1_write(0x00);
    MAX7219_CS_LOAD = 1;
}
void Display_On_3()
{
    MAX7219_CS_LOAD_2 = 0;
    SPI2_write(DISPLAY_REG);
    SPI2_write(DISPLAY_ON);
    MAX7219_CS_LOAD_2 = 1;
}
void Display_Off_3()
{
    MAX7219_CS_LOAD_2 = 0;
    SPI2_write(DISPLAY_REG);
    SPI2_write(DISPLAY_OFF);
    MAX7219_CS_LOAD_2 = 1;
}
void write_number(int position, int number)
{
    MAX7219_CS_LOAD = 0;
```

```
SPI1_write(0x00);
SPI1_write(0x00);
SPI1_write(position);
SPI1_write(number);
MAX7219_CS_LOAD = 1;
}
void write_laps(int position, int number)
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(position);
    SPI1_write(number);
    SPI1_write(0x00);
    SPI1_write(0x00);
    MAX7219_CS_LOAD = 1;
}
void write_number_3(int position, int number)
{
    MAX7219_CS_LOAD_2 = 0;
    SPI2_write(position);
    SPI2_write(number);
    MAX7219_CS_LOAD_2 = 1;
}
void write_dash(int position)
{
    MAX7219_CS_LOAD = 0;
    SPI1_write(0x00);
    SPI1_write(0x00);
    SPI1_write(position);
    SPI1_write(0b00001010);
    MAX7219_CS_LOAD = 1;
}
void write_letter(int position, int letter)
{
    // display letter on the second display
    MAX7219_CS_LOAD = 0;
    SPI1_write(position);
    SPI1_write(letter);
    SPI1_write(0x00);
    SPI1_write(0x00);
    MAX7219_CS_LOAD = 1;
}

void write_dash_3(int position)
{
    MAX7219_CS_LOAD_2 = 0;
```

```

    SPI2_write(position);
    SPI2_write(0b00001010);
    MAX7219_CS_LOAD_2 = 1;
}
void show_lap_time(int seconds, int seconds_tens, int minutes, int minutes_tens, int hours, int hours_tens)
{
    Display_On_3();
    write_number_3(1,seconds);
    write_number_3(2,seconds_tens);
    write_dash_3(3);
    write_number_3(4,minutes);
    write_number_3(5,minutes_tens);
    write_dash_3(6);
    write_number_3(7,hours);
    write_number_3(8,hours_tens);
}
void write_time(int seconds, int seconds_tens, int minutes, int minutes_tens, int hours, int hours_tens)
{
    write_number(1,seconds);
    write_number(2,seconds_tens);
    write_number(4,minutes);
    write_number(5,minutes_tens);
    write_number(7,hours);
    write_number(8,hours_tens);
}
void write_word_laps()
{
    write_letter(8,0b00001110); // letter L
    write_letter(7,0b01110111); // letter A
    write_letter(6,0b01100111); // letter P
    write_letter(5,0b01011011); // letter S
    write_letter(4,0); // digit 4 turn off
    write_letter(3,0);
    write_letter(2,0);
}
void interrupt()
{
    if (UART2_Data_Ready() == 1)
        UART2_Read_Text(ID_Read, "\n",32);

    for(i = 0; i < 30; i++)
        if(ID_Read[i] == ID[i])
            b++;
    if(b == 30)

```

```

{
    if(start == 0)
    {
        start = 1;
        l--;
    }
    else if (pt == 1 && pause == 0)
    {
        l--;
        h_tens_lap = h_tens_2, h_lap = h_2, m_tens_lap = m_tens_2,
m_lap = m_2, s_tens_lap = s_tens_2, seconds_lap = seconds_2,
h_tens_2 = 0, h_2 = 0, m_tens_2 = 0, m_2 = 0, s_tens_2 = 0,
seconds_2 = 0;
        stop = 1;
        pt = 0;
        if(l == -1)
        {
            pause = 1;
            tot--;
        }
    }
    else if(pause == 1 && num > 4 && seconds_3 < 5)
    {
        num = 0;
        stop = 0;
        h_tens_2 = 0, h_2 = 0, m_tens_2 = 0, m_2 = 0, s_tens_2 = 0,
seconds_2 = 0;
        pause = 2;
        tot--;
        l = laps;
        l--;
    }
}
b = 0;
RC2IF_bit = 0;
}

int main()
{
    ANSELC = 0; // PORTC digital
    ANSELD = 0; // PORTD digital
    TRISC = 0b00000000; // PORTC - output
    TRISD = 0b11111111;
    SPBRG2 = 0x33; // baudrate 9600
    SPBRGH2 = 0;
    BAUDCON2 = 0;
}

```



```

TXSTA2 = 0b00100000; // enable transmnision
RCSTA2 = 0b10110000; // enable reception
MAX7219_CS_LOAD = 0; // Deselect DAC
MAX7219_CS_LOAD_DIRECTION = 0; // Set CS pin as Output
MAX7219_CS_LOAD_2 = 0; // Deselect DAC
MAX7219_CS_LOAD_2_DIRECTION = 0;
SPI1_Init(); // Initialize SPI module
SPI2_Init(); MAX7219_init();// first display initialization
MAX7219_init_2(); // second display initialization
MAX7219_init_3(); // third display initialization
UART2_Init(9600);
INTCON.GIE = 1;
INTCON.PEIE = 1;
PIE3.RC2IE = 1;
PIR3.RC2IF = 0;

l = laps; // show remaining laps to go
write_word_laps();
write_laps(1,laps); // number of laps

for(i = 1; i < 9; i++)
{
    if(i%3 == 0)
        write_dash(i);
    else
        write_number(i,0);
}
tot = 2*count-1;
while(start != 1){}
while(tot != 0)
{
    if(pause == 0)
    {
        h_tens = 0, h = 0, m_tens = 0, m = 0, s_tens = 0, seconds = 0;
        write_word_laps();
        write_laps(1,laps);
        pt = 0;
        num = 0;
        while(l != -1)
        {
            if(stop == 1 && laps > 1)
                show_lap_time(seconds_lap,s_tens_lap,m_lap,m_tens_lap,
h_lap,h_tens_lap);
            write_time(seconds, s_tens, m, m_tens, h, h_tens);
            if(m_2 > 0)
                pt = 1;
        }
    }
}

```

```
delay_ms(1000);
seconds++;
seconds_2++;
if(seconds_2 == 5)
{
    Display_Off_3();
    write_laps(1,1);
    stop = 0;
}
if(seconds_2 == 10)
{
    seconds_2=0;
    s_tens_2++;
    if (s_tens_2 == 6)
    {
        s_tens_2=0;
        m_2++;
        if (m_2 == 10)
        {
            m_2=0;
            m_tens_2++;
            if (m_tens_2 == 6)
            {
                m_tens_2 = 0;
                h_2++;
                if(h_2 == 10)
                {
                    h_2 = 0;
                    h_tens_2++;
                }
            }
        }
    }
}
if(seconds == 10)
{
    seconds=0;
    s_tens++;
    if (s_tens == 6)
    {
        s_tens=0;
        m++;
        if (m == 10)
        {
            m=0;
            m_tens++;
        }
    }
}
```

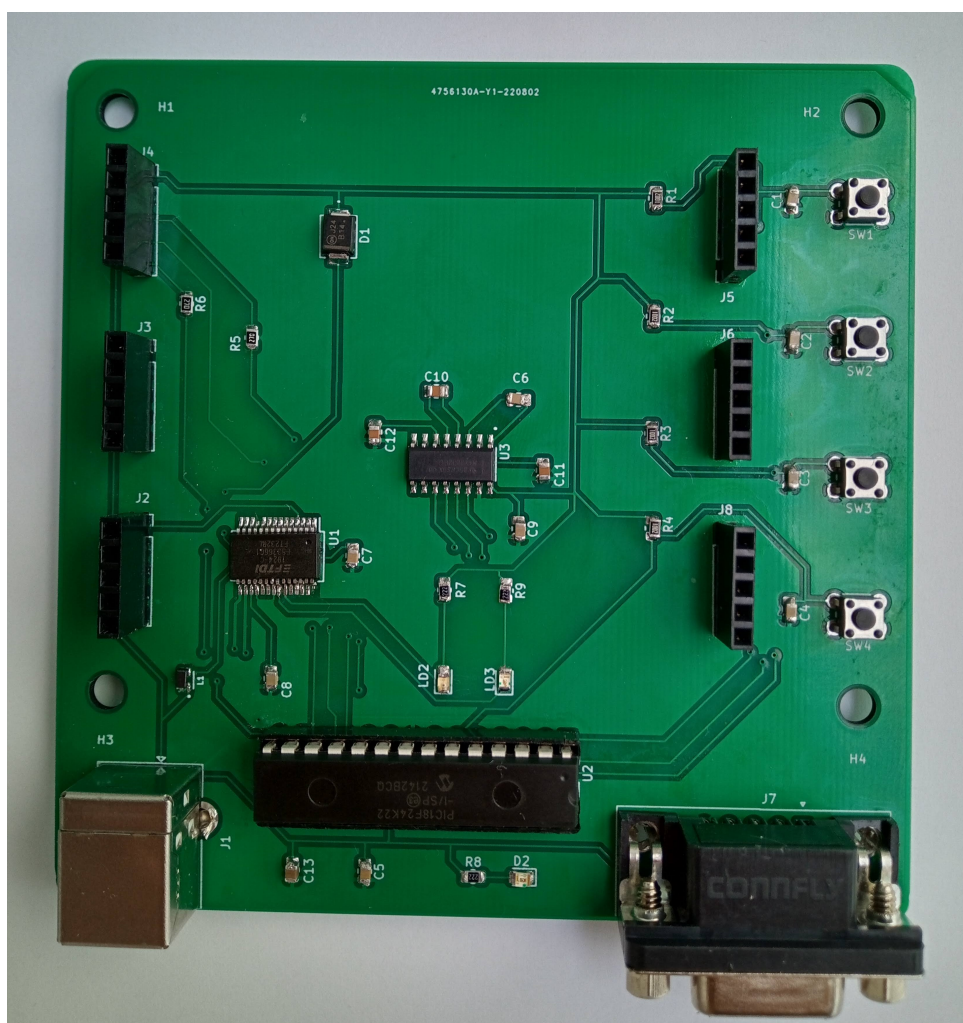
```
        if (m_tens == 6)
        {
            m_tens = 0;
            h++;
            if(h == 10)
            {
                h=0;
                h_tens++;
            }
        }
    }
}
}
}
}
}
if(laps > 1)
    show_lap_time(seconds_lap,s_tens_lap,m_lap,m_tens_lap,
h_lap,h_tens_lap);
}
else if(pause == 1)
{
    x = 0;
    h_tens_3 = h_tens_pause, h_3 = h_pause,
m_tens_3 = m_tens_pause, m_3 = m_pause,
s_tens_3 = s_tens_pause, seconds_3 = seconds_pause;
    for(i = 0; i < 6; i++)
        check[i] = 0;
    write_letter(8,0b01100111); // P
    write_letter(6,0b00111110); // U
    write_letter(4,0b01001111); // E
    write_laps(1,15);
    while(num != 6 && pause != 2)
    {
        if(x > 9)
        {
            write_time(seconds_3,s_tens_3,m_3,m_tens_3,h_3,h_tens_3);
            Display_Off_3();
        }
        delay_ms(1000);
        x++;
        seconds_3--;
        if(h_tens_3 == 0)
        {
            check[0] = 1;
            if (h_3 == 0)
            {
                check[1] = 1;
            }
        }
    }
}
```

```
        if (m_tens_3 == 0)
        {
            check[2] = 1;
            if (m_3 == 0)
            {
                check[3] = 1;
                if(s_tens_3 == 0)
                {
                    check[4] = 1;
                    if (seconds_3 == 0)
                    {
                        check[5] = 1;
                    }
                }
            }
        }
    }
}
if(seconds_3 < 0 && check[4] != 1)
{
    seconds_3 = 9;
    s_tens_3--;
    if(s_tens_3 < 0 && check[3] != 1)
    {
        s_tens_3 = 5;
        m_3--;
        if(m_3 < 0 && check[2] != 1)
        {
            m_3 = 9;
            m_tens_3--;
            if(m_tens_3 < 0 && check[1] != 1)
            {
                m_tens_3 = 5;
                h_3--;
                if(h_3 < 0 && check[0] != 1)
                {
                    h_3 = 9;
                    h_tens_3--;
                    if (h_tens_3 < 0 )
                    {
                        h_tens_3 = 0;
                    }
                }
            }
        }
    }
}
```

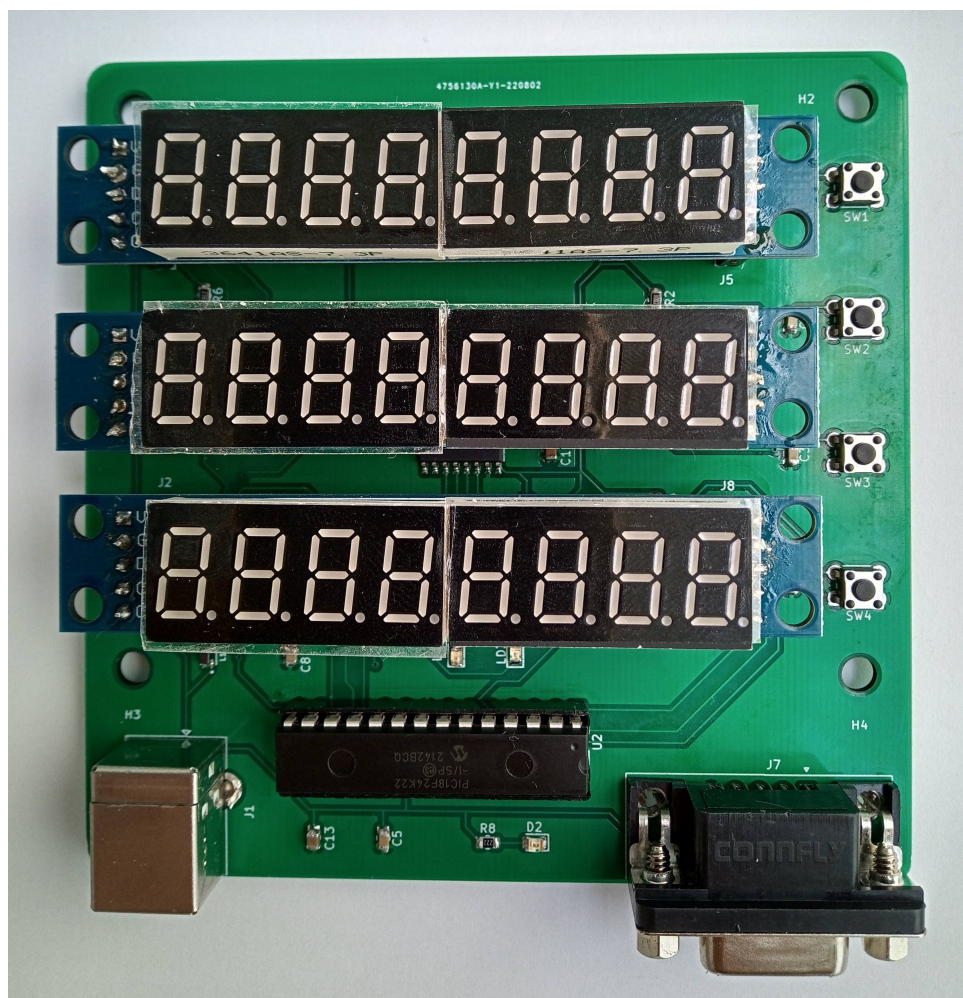
```
    }
    num = 0;
    for(i = 0; i < 6; i++)
        if (check[i] == 1)
            num++;
    }
    write_number(1,seconds_3);
    while(pause != 2){}
    pause = 0;
    }
}
for(i = 2; i < 9; i++)
    write_letter(i,0b00000001);
write_laps(1,10);
if(laps > 1)
    show_lap_time(seconds_lap,s_tens_lap,m_lap,m_tens_lap,h_lap,
h_tens_lap);
}
```


Příloha B

Osazená deska



Obrázek B.1: Osazená deska



Obrázek B.2: Osazená deska s displeji