

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA ELEKTRONIKY A INFORMAČNÍCH TECHNOLOGIÍ

BAKALÁŘSKÁ PRÁCE

Balistický chronometr

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Pavel TRÁVNÍČEK**
Osobní číslo: **E18B0187P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a telekomunikace**
Téma práce: **Balistický chronometr**
Zadávající katedra: **Katedra elektroniky a informačních technologií**

Zásady pro vypracování

Navrhňte balistický chronometr, schopný měřit rychlost vystřeleného projektilu různých zbraní.

1. Prostudujte způsob a možnosti měření rychlosti vystřelených projektilů.
2. Navrhňte zapojení umožňující měřit rychlost projektilů zvolených typů zbraní.
3. Navržené řešení realizujte.
4. Zhodnoťte dosažené parametry a možnosti realizovaného řešení.

Rozsah bakalářské práce: **30 – 40 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Jiří Pinker: Mikroprocesory a mikropočítače. BEN – technická literatura ISBN 80-7300-110-1

Vedoucí bakalářské práce: **Ing. Kamil Kosturik, Ph.D.**
Katedra elektroniky a informačních technologií

Datum zadání bakalářské práce: **9. října 2020**
Termín odevzdání bakalářské práce: **27. května 2021**



Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Doc. Ing. Jiří Hammerbauer, Ph.D.
vedoucí katedry

Abstrakt

Cílem této bakalářské práce byla realizace balistického chronometru. Jedná se o elektronické zařízení, které dokáže změřit rychlost vystřeleného projektilu ze zbraně. Zařízení je tvořeno dvěma optickými závory, které tvoří dvě klíčové součástky, a to vysílač (IR LED dioda) a přijímač (OPL550A). Balistický chronometr jsou v podstatě stopky, které začnou čítat, když projektil přeruší první optickou závoru a zastaví čítání při přerušení druhé optické závory. Díky tomu je známá doba (t) průletu projektilu mezi závory a vzdálenost (s) závor je pevně daná konstrukcí. Nakonec mikrokontrolér ATmega16A provede výpočet rychlosti ze vzorce $v = \frac{s}{t}$ a vypočtená rychlost se poté zobrazí na LCD displeji.

Klíčová slova

Balistický chronometr, balistický chronograf, LCD displej, mikrokontrolér ATmega 16A-PU, chrono, sportovní střelba, měření rychlosti projektilu

Abstract

The goal of this bachelor thesis was the implementation of a ballistic chronometer. It is an electronic device that can measure the speed of a projectile fired from a weapon. The device consists of two optical barriers, which form two essential components, namely transmitter (IR LED diode) and a receiver (OPL550A). A ballistic chronometer is basically a stopwatch that starts counting when a projectile interrupts the first optical barrier and stops counting when the projectile interrupts the second optical barrier. As a result, the time (t) of flight of the projectile between the barriers and the distance (s) of the barriers is given by construction. Finally, the microcontroller ATmega16A calculates the speed from the formula $v = \frac{s}{t}$ and calculated velocity is then displayed on LCD display.

Key words

Ballistic chronometer, ballistic chronograph, LCD display, microcontroller ATmega 16A-PU, chrono, sport shooting, projectile velocity measurement

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....
podpis

V Plzni dne 26.5.2021

Pavel Trávníček

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Kamilovi Kosturikovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce.

Obsah

OBSAH	8
SEZNAM OBRÁZKŮ A TABULEK	9
SEZNAM SYMBOLŮ A ZKRATEK	10
ÚVOD	11
1 POROVNÁNÍ RŮZNÝCH POUŽITELNÝCH DRUHŮ SENZORŮ	12
1.1 OPTICKÉ SENZORY	12
1.2 ULTRAZVUKOVÉ SENZORY	12
1.3 DOPPLERŮV RADAR.....	13
2 POUŽITÉ KOMPONENTY Z POHLEDU TEORIE	13
2.1 ŘÍDÍCÍ JEDNOTKA – ARCHITEKTURA AVR.....	13
2.2 MIKROKONTROLÉR ATMEGA 16A A JEHO HLAVNÍ PARAMETRY	15
2.3 POPIS VYUŽITÝCH FUNKCÍ ATMEGA 16A.....	17
2.3.1 Rozhraní SPI.....	17
2.3.2 Přerušení	19
2.3.3 Čítač/časovač	21
2.4 OPTICKÉ ZÁVORY	23
2.4.1 Přijímače OPL 550A	23
2.4.2 Vysílače.....	24
2.5 ZOBRAZOVACÍ JEDNOTKA - LCD DISPLEJ.....	24
2.6 KRYSTALOVÝ OSCILÁTOR 16 MHZ	26
3 NÁVRH A VÝVOJ HARDWARU A SOFTWARE	26
3.1 HARDWAROVÁ ČÁST	26
3.1.1 Napájení	26
3.1.2 Optoelektronické závory.....	27
3.1.3 Zapojení mikrokontroléru ATmega 16A.....	27
3.1.4 Zapojení LCD displeje.....	27
3.1.5 ISP programování.....	28
3.2 SOFTWARE	29
3.2.1 Externí přerušení	30
3.2.2 Využití čítače / časovače.....	31
3.2.3 Výpočet rychlosti	32
3.3 NÁVRHY JEDNOTLIVÝCH DPS	32
3.4 OVĚROVACÍ MĚŘENÍ.....	34
ZÁVĚR	36
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	37
PŘÍLOHY	1

Seznam obrázků a tabulek

OBR. 1: BLOKOVÉ SCHÉMA AVR MCU [9]	14
OBR. 2: PINOUT ATMEGA 16A [9]	15
OBR. 3: BLOKOVÉ SCHÉMA ATMEGA16A [9]	16
OBR. 4: ZNÁZORNĚNÍ SYSTÉMU SE SBĚRNÍCI SPI [10]	18
OBR. 5: ISP PINOUT [12]	19
OBR. 6: ČASOVÝ PRŮBĚH U CTC MÓDU [9]	22
OBR. 7: SCHÉMA ZAPOJENÍ OPL550A [16]	24
OBR. 8: ZNAKOVÁ SADA HD44780(TYP A00) [17]	25
OBR. 9: UKÁZKA 4-BITOVÉ KOMUNIKACE [17]	25
OBR. 10: ZAPOJENÍ KRYSTALU	26
OBR. 11: ZAPOJENÍ LCD DISPLEJE	28
OBR. 12: PROSTŘEDÍ SOFTWARE AVR LOADER	29
OBR. 13: PROSTŘEDÍ ATMEL STUDIA 7.0	30
OBR. 14: POHLED SHORA NA CENTRÁLNÍ DPS	33
OBR. 15: POHLED SHORA NA DPS SE SENZORY	33
OBR. 16: POHLED SHORA NA DPS S LED DIODOU	34
TAB. 1: PŘEHLED VEKTORŮ PŘERUŠENÍ[9]	21
TAB. 2 : TABULKA NAMĚŘENÝCH HODNOT	34

Seznam symbolů a zkratek

C	Programovací jazyk
CTC	Pracovní režim čítače/časovače
DIP	Druh pouzdra, které má dvě řady pinů
DPS	Deska plošných spojů
EEPROM	Elektricky přemazatelná programovatelná paměť
FLASH	Elektricky programovatelná paměť
HASL	Povrchová úprava DPS, kdy je DPS ponořena do cínové pájecí slitiny
IR LED	Infračervená LED dioda
ISP	Programování v systému
LCD	Displej skládající se z tekutých krystalů
MCU	Mikrokontrolér
MISO	Datová linka SPI kanálu
MOSI	Datová linka SPI kanálu
SCK	Synchronizační hodinový signál SPI
SMT	Povrchová montáž
SPI	Sériové periferní rozhraní
SRAM	Statická paměť
SS	Pin pro výběr obvodu
THT	Technologie osazování DPS součástkami s drátovými vývody
TTL	Tranzistorová logika
USBasp	Programátor

Úvod

Jako téma své bakalářské práce jsem si zvolil balistický chronometr, z toho důvodu, že jsem několik let aktivním členem sportovního střeleckého klubu. Balistický chronometr najde své využití hlavně u střelců, kteří si přebíjejí náboje a potřebují si ověřit skutečnou rychlost svých nábojů. Důvod proč si chtít ověřovat rychlost svých projektilů je hlavně ten, že v určitých sportovních disciplínách existují tzv. výkonnostní faktory, kde hraje hlavní roli právě rychlost střely a její hmotnost. Pokud střelec poddimenzuje náboje mohl by získat určité výhody, hlavně z hlediska chování zbraně při rychlé střelbě. Naopak u disciplín kde rychlost střelby nehraje roli by naopak mohl získat výhodu ten střelec, který by měl předimenzované náboje, jelikož by projektily měly lepší balistickou křivku. Výhodou oproti komerčnímu chronografu je daleko příznivější pořizovací cena, samozřejmě záleží na kvalitě a propracovanosti komerčního chronometru. Přesnost měření je srovnatelná, což je hlavní požadavek na toto zařízení. Balistický chronometr řešený v této práci by měl být schopen změřit i ty nejmenší projektily, jako jsou plastové kuličky o průměru 6 mm, olovené diabolky do vzduchových zbraní, ale také i střely velkorážných zbraní, např. ráže 9 mm luger. Vzhledem k této univerzálnosti zařízení je vhodná konstrukce umístěná mimo zbraň, kdy je chronometr položen před hlavní zbraň. Není zde vhodná konstrukce umístěná přímo na hlaveň, jak tomu bývá u chronografů určených zejména pro vzduchové zbraně. Toto u velkorážných zbraní nelze realizovat, jelikož mají značný zpětný ráz a při výstřelu vzniká v okolí hlavně značná tlaková vlna. Další problém s umístěním přímo na zbraň by nastal u krátkých zbraní z důvodu pohyblivého závěru.

Nejdůležitějším bodem při návrhu chronometru je způsob řešení snímání projektilů. Navržený chronometr v této práci využívá optoelektronických senzorů. Ty se skládají z přijímače a vysílače, tedy IR LED dioda. Z těchto komponentů jsou sestaveny dvě optické závory. Poté co projektil proletí první závorou odstartuje se čítání času a přeruší se v momentě, když projektil proletí druhou závorou. Jako řídicí jednotka slouží mikrokontrolér ATMEGA 16A-PU. O zobrazení výsledku se stará dvouřádkový LCD displej.

1 Porovnání různých použitelných druhů senzorů

V této kapitole jsou uvedeny tři možnosti jakými by bylo možné provést snímání projektilů. Vzhledem k tomu, že zařízení má být schopno detekovat a naměřit rychlosti i nekovových projektilů, tak indukční senzory jsou vyloučeny.

1.1 Optické senzory

Optické senzory se skládají ze dvou základních částí, a to ze zdroje (vysílače) optického záření a přijímače, který toto záření zpracovává. Zdroji tohoto záření jsou nejčastěji LED diody (IR LED diody) anebo laserové diody. K detekování se používají jako přijímače nejčastěji fotodiody, fototranzistory, detektory citlivé na polohu (PSD – position sensitive device) nebo také detektory se strukturou vázanou na náboje (CCD – charge coupled device). Mají tu vlastnost, že nejsou citlivé na rušení elektromagnetickým polem. Dále mají kladnou vlastnost, že disponují poměrně velkým rozsahem vzdáleností, ve kterém zvládnou spolehlivě fungovat. Naopak nevýhodou může být větší citlivost na denní světlo. [1], [2]

Téměř všechny vyráběné komerční chronometry využívají optické senzory. Jako zdroje optického záření se používají infračervené LED diody a pro detekování projektilu v drtivé většině fotodiody, fototranzistory anebo integrované obvody optických detektorů obsahující fotodiodu. Řešení pomocí optoelektronických senzorů je asi nejvýhodnější ze všech hledisek, ať už hlediska funkčnosti, z ekonomického hlediska, tak i jednoduchosti celého systému. Jedinou nevýhodou by mohla být citlivost na denní světlo, ale tento potenciální problém lze vyřešit konstrukcí chronometru.

1.2 Ultrazvukové senzory

Dalším možným druhem senzoru je senzor ultrazvukový. Princip činnosti ultrazvukových senzorů je založen na tom, že se měří doba, která uplyne od vyslání impulsu ke snímanému objektu, jehož vzdálenost se měří do doby než se od tohoto objektu

vrátí zpět. Jako vysílač a přijímač se používá nejčastěji piezoelektrický krystal. Hlavní jeho výhodou je, že dokáže detekovat v podstatě jakýkoli materiál. Hlavní využití najde v průmyslu, například jako detektor předmětů na dopravníku, dále jako detektor lesklých a průhledných předmětů (fólie, PET lahve) nebo je hojně využíván pro monitorování výšky hladiny v nádrži. [3], [4]

Pro použití v této bakalářské práci to není nejvhodnější způsob detekce projektilů. Oproti fotodiodám a LED diodám je pořizovací cena daleko vyšší. Konstrukčně se také jeví optoelektronické senzory jako daleko lepší volba, jelikož jsou rozměrově menší.

1.3 Dopplerův radar

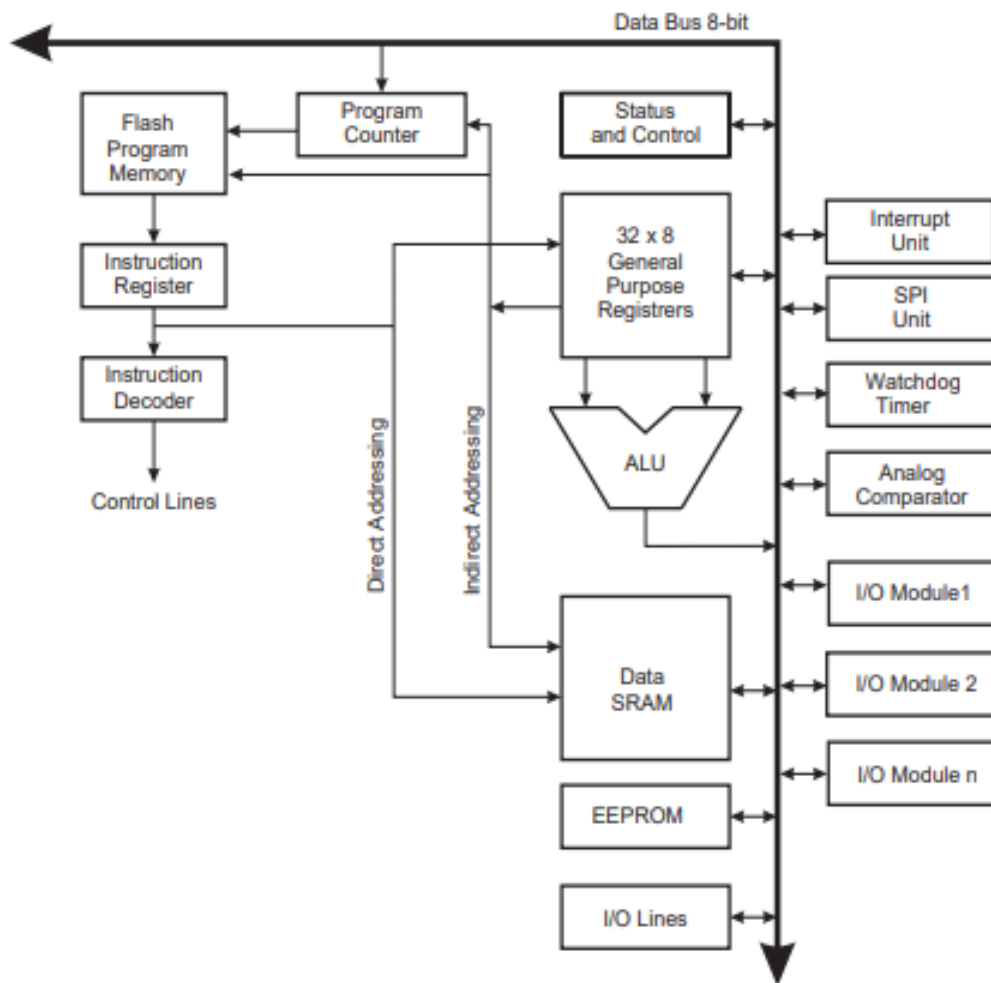
Využití radaru jako balistického chronografu je zde uvedeno spíše jen pro zajímavost. Bylo by poměrně nákladné zrealizovat takové zařízení. Takové typické uplatnění Dopplerova radaru je u policie, kdy radar měří rychlost jedoucích automobilů. Dále najde uplatnění v letectví, ve vojenském sektoru anebo také v meteorologii, kde takový radar dokáže měřit jak rychle se pohybují srážkové částice v oblacích. Jak název napovídá, princip činnosti je založen na Dopplerově jevu. Ten spočívá v tom, že při vzájemném pohybu zdroje, který vysílá vlnění a pozorovatele se mění frekvence vyslaného vlnění. [5]

2 Použité komponenty z pohledu teorie

2.1 Řídící jednotka – architektura AVR

Použitý mikrokontrolér v této práci, tedy ATmega 16A, patří do rodiny AVR mikrokontrolérů. Tyto mikrokontroléry se vyznačují harvardskou architekturou, tedy oddělenou pamětí pro data a pro program. Dále se vyznačují architekturou RISC (Advanced), tedy pokročilou. Zkratka RISC (Reduced Instruction Set Computer) znamená počítač s redukovanou instrukční sadou. Instrukční sada u ATmega 16 obsahuje 131 instrukcí. Mezi hlavní rysy konstrukce RISC patří to, že v každém strojovém cyklu by vždy měla být dokončena jedna instrukce. U MCU ATmega odpovídá jeden strojový cyklus přesně hodinovému cyklu. Dále je pro ATmega typické zřetěžené zpracovávání

jednotlivých instrukcí. Nejprve se načte první instrukce, ta se v dalším hodinovém cyklu provádí a současně se už vybírá předem druhá instrukce, která bude prováděna až v dalším cyklu.[6], [15]

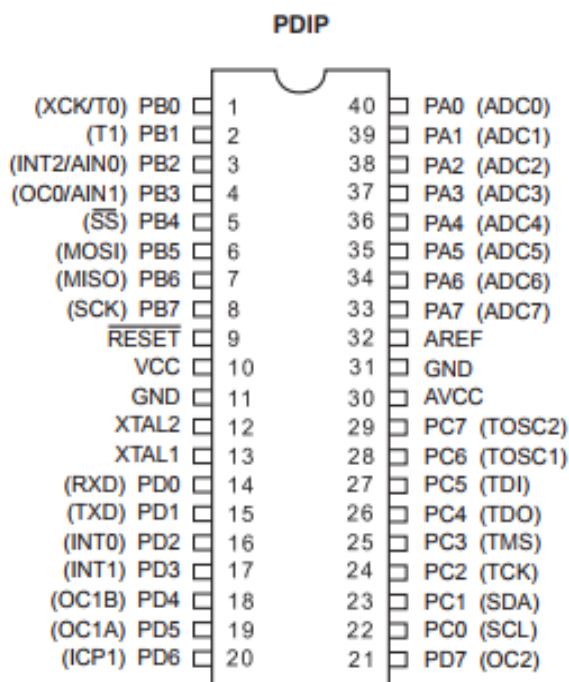


Obr. 1: Blokové schéma AVR MCU [9]

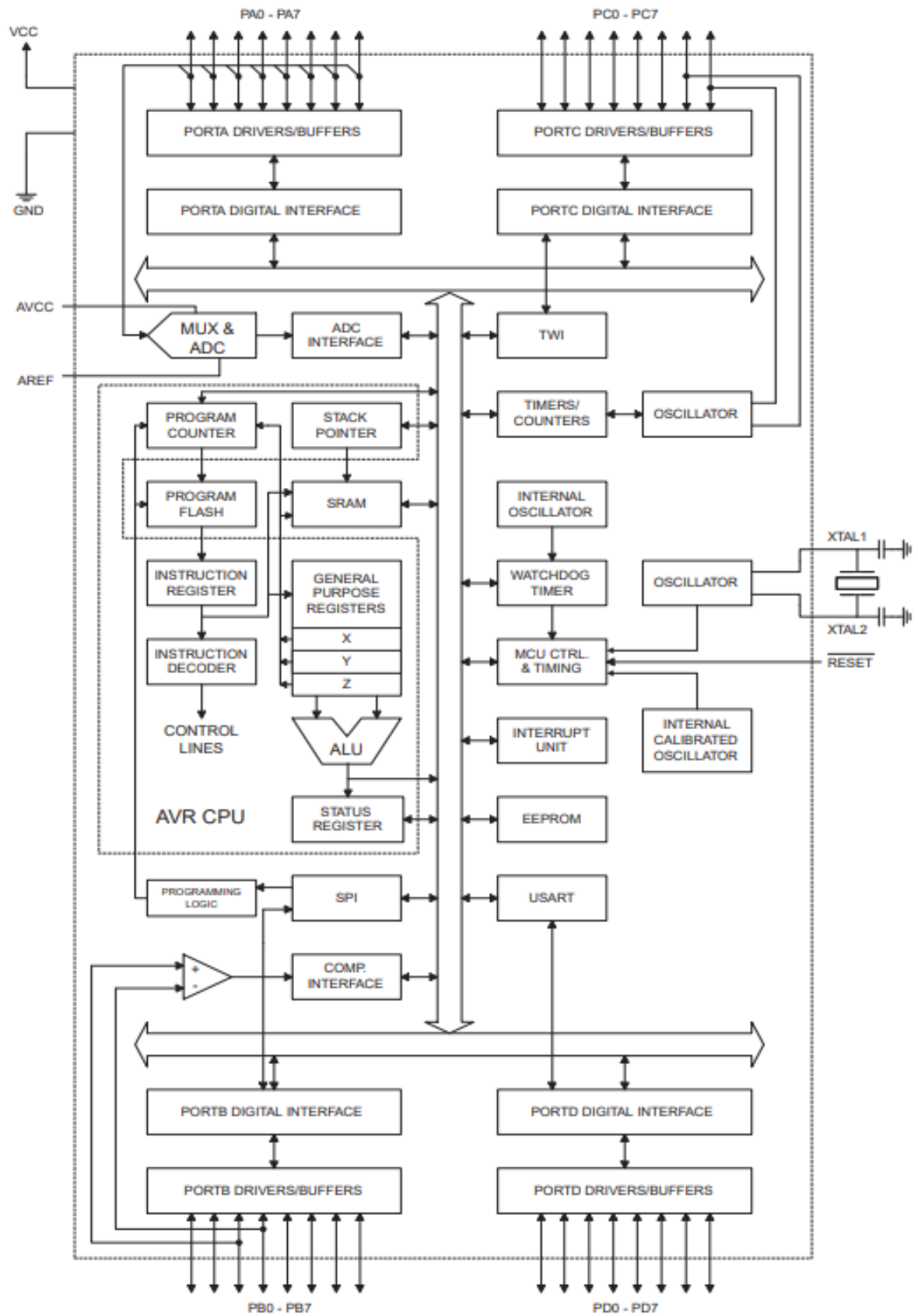
2.2 Mikrokontrolér ATmega 16A a jeho hlavní parametry

Řídící jednotkou byl v této práci zvolen mikrokontrolér ATmega16A. Pro tuto aplikaci je se svými parametry naprosto vyhovující. A to jak z pohledu velikosti paměti, dostatečného počtu I/O pinů. Další požadavek byl, aby mikrokontrolér měl alespoň dva piny pro externí přerušení, což také splňuje, jelikož má tři takové piny (INT0, INT1 a INT2). Další nárok byl, aby byla možnost ISP programování, což je možné.

ATmega16A je 8-bitový mikrokontrolér s vysokým výkonem a malým příkonem. Je vyroben CMOS technologií. Rozmezí napájecího napětí uvádí výrobce 2,7 V až 5,5 V. Paměť pro program a dat je oddělena, jelikož se jedná o harvardskou architekturu. Uživatel má k dispozici 32 8-bitových pracovních registrů. Celkový počet I/O pinů je 32 (4 porty po 8 pinech), přičemž každý z těchto pinů může být použit jako vstupní nebo výstupní. Mikrokontrolér má dohromady celkem 40 pinů, v této práci je využito pouzdro DIP. Čip dokáže poskytnout 16 KB Flash paměti, operační paměť SRAM má velikost 1024 B a velikost EEPROM paměti je 512 B. ATmega16A obsahuje dva 8-bitové a jeden 16-bitový časovače/čítače. Co se týče komunikačních rozhraní, tak je k dispozici sériové rozhraní USART a sériové rozhraní SPI. [8], [9]



Obr. 2: Pinout ATmega 16A [9]



Obr. 3: Blokové schéma ATmega16A [9]

2.3 Popis využitých funkcí ATmega 16A

2.3.1 Rozhraní SPI

Sériové periferní rozhraní (SPI) umožňuje vysokorychlostní synchronní přenos dat mezi několika mikrokontroléry navzájem nebo mezi mikrokontrolérem a periferními zařízeními. Přenos po SPI je duplexní, to znamená že tok dat probíhá v obou směrech. Princip SPI komunikace je celkem jednoduchý, systém má vždy jen jeden obvod nazývaný Master (Hlavní, řídicí) a obvykle několik obvodů Slave (podřízený). Tyto obvody se propojují čtyřmi následujícími vodiči: MISO (Master In, Slave Out), MOSI (Master Out, Slave In), SCK (Clock) a SS (Slave Select). Po lince MOSI se posílají data z obvodu Master do obvodu Slave. Naopak po lince MISO tečou data z obvodu Slave do obvodu Master. Hodinový signál SCK je generován Master obvodem a je připojen ke všem SCK vstupům obvodů Slave. Poslední čtvrtá linka, tedy SS slouží pro výběr obvodu. [10]

Sériový kanál SPI je ovládaný třemi registry. Prvním, řídicím registrem je SPCR, stavovým registrem je SPSR a datovým registrem je registr SPDR. V SPCR je 8 následujících bitů.

- ***SPIE***

Tento nejvýznamnější bit povoluje přerušení.

- ***SPE***

Stará se o aktivaci kanálu. Když je SPE rovné 0, tak je kanál aktivovaný. Pro připojení slouží u mikrokontroléru ATmega16A piny PB7 až PB4.

- ***DORD***

Udává v jakém pořadí se přenáší data, zda v LSB nebo MSB. Když je DORD rovné nule, tak se data přenáší od MSB (nejvýznamnějšího bitu) a přenos ukončuje LSB (nejméně významný bit).

- ***MSTR***

V případě MSTR = 1 konfiguruje MCU jako Master, v případě 0 jako Slave.

- ***CPOL a CPHA***

CPOL nastavuje polaritu synchronizačního hodinového signálu SCK. CPHA fázi tohoto signálu.

- ***SPR0 a SPR1***

Tyto 2 bity určují kmitočet hodinového signálu SCK.

V registru SPSR nalezneme 2 příznaky – SPIF a WCOL a 1 bit SPI2X.

- ***SPIF a WCOL***

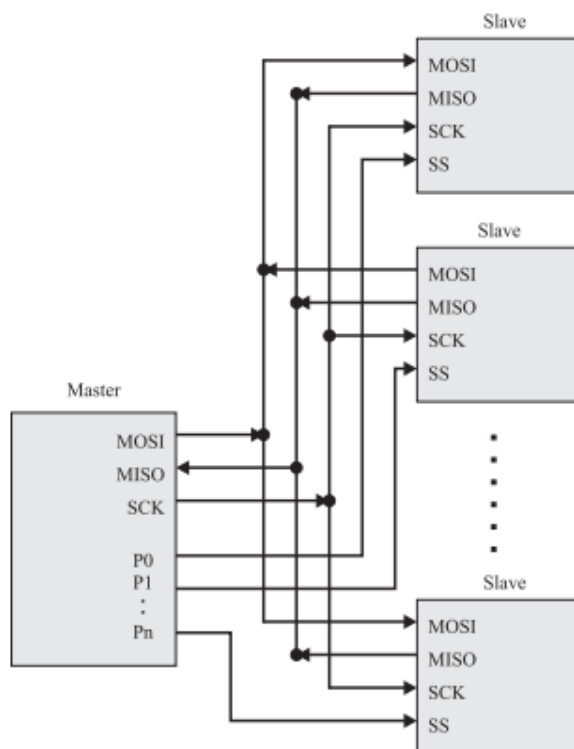
Po skončení přenosu dat se SPIF nastaví na hodnotu 1. WCOL signalizuje kolizi při zápisu dat. WCOL je nastaven v případě, že se zapsala data do SPDR během přenosu dat.

- ***SPI2X***

Tímto bitem se nastavuje dvojnásobná přenosová rychlost.

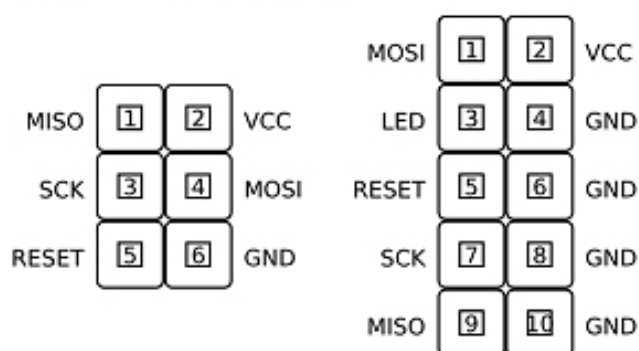
Do datového registru SPDR se po skončení přenosu zapisují data. Po zápisu se tyto data vysílají po SPI. [15]

Největší výhodou SPI je její jednoduchost. Přispívá k tomu také to, že komunikace probíhá v obou směrech po samostatných linkách a není třeba přepínat mezi módy, kdy se mají data vysílat a kdy přijímat. [11]



Obr. 4: Znárodnění systému se sběrnici SPI [10]

V této práci je využita komunikace SPI k naprogramování mikrokontroléru ATmega16A. Jedná se o ISP (In System Programming) programování. A jak již název napovídá, jde o programování mikrokontroléru přímo v systému. To přináší velkou výhodu a usnadnění práce. Zejména při vývoji zařízení to ušetří mnoho času, jelikož se nemusí zbytečně manipulovat s mikrokontrolérem.



Obr. 5: ISP Pinout [12]

2.3.2 Přerušeni

Mikrokontrolér Atmega 16A má vyhrazené 3 piny pro externí přerušeni (INT0-pin PD2, INT1-pin PD3 a INT2-pin PB2). Externí přerušeni se stará o to, aby se aktuální zpracováváný běh programu přerušil a přesměroval ho na „rutinu pro obsluhu přerušeni“. Po vykonání této rutiny se vrátí zpátky na místo, kde byl běh programu přerušeni a odtud pokračuje standardně dále. Zde je dobré si uvědomit, že v přerušeni by se neměly měnit obsahy registrů. Je to z toho důvodu, že kdyby se v přerušeni změnil obsah registru, který je využíván v hlavním programu, tak by se po skončení rutiny obsluhy přerušeni mohlo stát něco nepředvídatelného. ATmega 16A má celkem 21 možných zdrojů přerušeni, každé toto přerušeni má vyhrazenou speciální adresu. Každé toto přerušeni má svoji prioritu a adresu. Priorita je daná vektorem přerušeni, čím je vektor nižší, tím je priorita vyšší. To znamená, že vektor 1 má nejvyšší prioritu a vektor 21 nejnižší. Nejvyšší prioritu má RESET. ATmega16A má pět následujících zdrojů resetu – reset po připojení napájení, externí reset, Watchdog reset, reset když poklesne napětí a reset od JTAG AVR rozhraní. Pro tuto práci jsou důležitá externí přerušeni INT0 a INT1, která mají vektory 2 a 3. Vstupy externího přerušeni jsou ovládány pomocí následujících registrů.

- **GICR**

Jedná se o masku přerušení, díky které lze povolit nebo zakázat externí přerušení.

- **GIFR**

V případě, že přijde požadavek na aktivaci přerušení, tak se nastaví příznak v tomto registru. Pokud přerušení bylo zakázáno a bylo vyvolané, tak si příslušný příznakový bit tento požadavek zapamatuje a přerušení se provede, jakmile bude přerušení povoleno. Toto ovšem platí jen pro případ, že je vstup hranově citlivý.

- **MCUCR**

Pomocí tohoto registru lze volit, za jakých podmínek se bude aktivovat externí přerušení. Pro vstup INT0 to jsou bity ISC01 a ISC00, pro přerušení INT1 - ISC11 a ISC10. Čtyřmi různými variantami nastavení dvojic bitů ISCXX, lze nastavit, aby se vstup aktivoval logikou 0 (pro INT0 : ISC01 = 0, ISC00 = 0), jakoukoliv změnou stavu INT (pro INT0 : ISC01 = 0, ISC00 = 1), sestupnou hranou INT (pro INT0 : ISC01 = 1, ISC00 = 0) anebo vzestupnou hranou INT (pro INT0 : ISC01 = 1, ISC00 = 1). [13], [15]

Tab. 1: Přehled vektorů přerušení[9]

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	\$000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$002	INT0	External Interrupt Request 0
3	\$004	INT1	External Interrupt Request 1
4	\$006	TIMER2 COMP	Timer/Counter2 Compare Match
5	\$008	TIMER2 OVF	Timer/Counter2 Overflow
6	\$00A	TIMER1 CAPT	Timer/Counter1 Capture Event
7	\$00C	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	\$00E	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	\$010	TIMER1 OVF	Timer/Counter1 Overflow
10	\$012	TIMER0 OVF	Timer/Counter0 Overflow
11	\$014	SPI, STC	Serial Transfer Complete
12	\$016	USART, RXC	USART, Rx Complete
13	\$018	USART, UDRE	USART Data Register Empty
14	\$01A	USART, TXC	USART, Tx Complete
15	\$01C	ADC	ADC Conversion Complete
16	\$01E	EE_RDY	EEPROM Ready
17	\$020	ANA_COMP	Analog Comparator
18	\$022	TWI	Two-wire Serial Interface
19	\$024	INT2	External Interrupt Request 2
20	\$026	TIMER0 COMP	Timer/Counter0 Compare Match
21	\$028	SPM_RDY	Store Program Memory Ready

2.3.3 Čítač/časovač

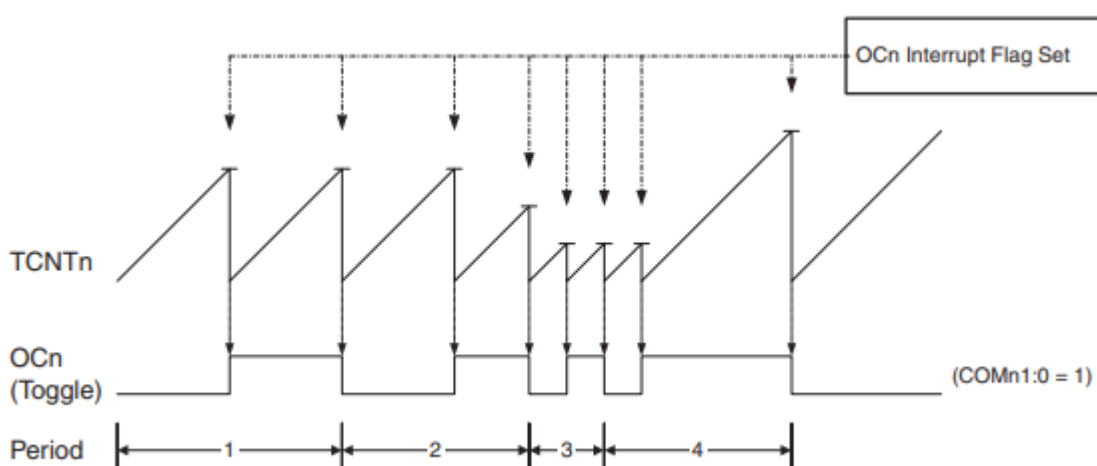
Čítače/časovače umožňují časování běhu akcí, počítání vnějších impulsů, dělení a měření frekvence a generování PWM signálů. Rozdíl mezi časovačem a čítačem je takový, že časovač využívá hodinového kmitočtu mikrokontroléru, zatímco čítač čítá impulsy externího signálu. Zachycení stavu časovače poskytuje záchytný registr. Toto zachycení je aktivováno vstupem ICP. Srovnání obsahu časovače se speciálním registrem zajišťuje výstupní komparátor (Output Compare). V případě shody se změní výstup OC.

Mikroprocesor ATmega 16A obsahuje 3 časovače/čítače, 2 jsou 8bitové (Časovač/čítač 0 a 2) a 1 je 16bitový (časovač/čítač 1). Pro tuto práci byl využit 8bitový časovač/čítač 2. Co se aktuálně nachází v časovači/čítači říká registr TCNT2. Jako

porovnávací registr slouží registr OCR2. Registr TCCR2 je řídicí registr časovače/čítače 2. Když časovač/čítač přeteče nebo dosáhne při čítání hodnoty rovné OCR2, tak se nastaví příznaky přerušení TOV2 a OCF2. Tyto signály souhlasí s maskami přerušení z registru TIMSK2 a příznaky z registru TIFR2. Když je přerušení povoleno, zavolá se rutina obsluhy přerušení. Podle toho jaký se zvolí pracovní režim časovače/čítače je buďto nulován nebo je obsah zvyšován/snižován s každým hodinovým signálem. Předdělička se nastavuje pomocí bitů CS22-CS20. Způsob jakým bude časovač/čítač čítat je určeno nastavením WGM20, WGM21, tedy bity z registru TCCR2. Výstupní signál OC2 je tedy dán tím jak se nastaví bity WGM a COM v registru TCCR2. Čítač/časovač může pracovat ve čtyřech různých režimech a těmi jsou normální režim, CTC režim, vysokorychlostní PWM a fázově korigovaný PWM režim. V této práci byl využit režim CTC, proto je zde detailněji popsán jen tento režim.

- **CTC režim**

Pro tento režim se musí nastavit bity WGM21 = 1 a WGM20 = 0. V CTC (Clear Timer on Compare Match) režimu se porovnaná hodnota registru TCNT2 s hodnotou OCR2. V případě, že dojde ke shodě těchto dvou hodnot, tak se čítač vynuluje a nastaví se příznak OCF2 a dojde k přerušení, za předpokladu, že je povoleno (je nastavena maska OCIE2 a I bit v registru SREG). Nastavením bitů COM21 = 0, COM20 = 1 lze nastavit toggle režim. Což znamená, že signál vývodu OC2 bude negovaný pokaždé, když TCNT2 bude mít stejnou hodnotu jako OCR2. [13], [14] a [15]



Obr. 6: Časový průběh u CTC módu [9]

2.4 Optické závory

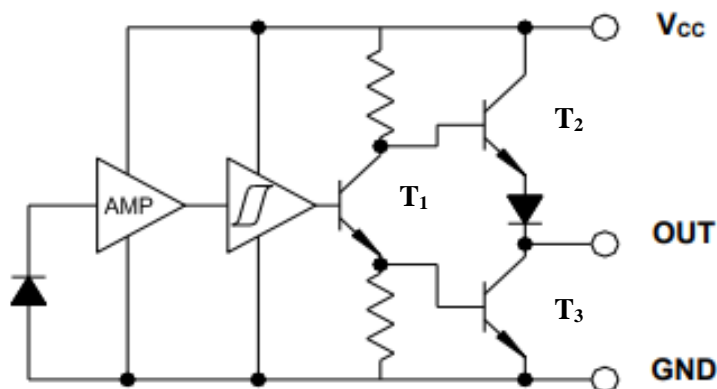
Optické závory se skládají ze dvou částí a to z vysílače a přijímače. Tyto dva prvky jsou popsány v následující dvou podkapitolách. Balistický chronograf disponuje dvěma takovými závory.

2.4.1 Přijímače OPL 550A

Senzor OPL550A je součástka v podobě integrovaného obvodu. Křemíkový čip obsahuje fotodiodu, zesilovač, Schmittův klopný obvod a tranzistory NPN. Součástka je v plastovém pouzdře, které zajišťuje dobré optické vlastnosti a také ochranu proti mechanickému poškození. Vyznačuje se výstupem logické úrovně kompatibilní s TTL / LSTTL logikou. Logické úrovně u TTL obvodů jsou následující : logická 1 je rovna napájecímu napětí, většinou +5 V a logická 0 je rovna 0 V. Pro to, aby nedošlo k tomu, že by se obvod dostal do mezistavu, kdy by se nedalo předvídat, zda je na výstupu log. 0 nebo log. 1, je potřeba aby napětí na výstupu nebylo menší než 2,4 V a zároveň vyšší než 0,4 V. Hodnota 0,4 V se označuje jako V_{OL} a hodnota 2,4 V jako V_{OH} . [22]

Výstup součástky OPL550A je typu Totem-pole. Údajně se mu říká Totem-pole z důvodu, že uspořádání výstupní brány připomíná indiánský totem. Myšleno symbolicky v tom smyslu, že jednotlivé komponenty - NPN tranzistory, dioda a rezistor jsou ve schéma „naskládáné“ jedna na druhé, podobně jak je tomu u zmiňovaného indiánského totému.

Výstupní obvod obsahující tranzistory T_2 a T_3 je zde buzen tranzistorem T_1 . V případě, že je T_1 nevybuzen, zůstane i T_1 a T_2 zavřený. Na T_2 je objeví kladné předpětí a otevře se. Na výstupu se objeví kladné napětí, blízké napájecímu, tedy logická 1. V případě, že je T_1 vybuzen a otevře se, tak na emitoru T_1 vzroste napětí a tím pádem se otevře tranzistor T_3 . Ale na kolektoru T_1 napětí klesne a předpětí pro T_2 ho nestačí vybudit a T_2 se zavře. Na výstupu se přes T_3 objeví logická 0. [23]



Obr. 7: Schéma zapojení OPL550A [16]

2.4.2 Vysílače

Vysílače, v našem případě LED (Light emitting diode) diody jsou polovodičové součástky, které obsahují jeden PN přechod. Po zapojení diody v propustném směru se stane dioda vodivou. Záření LED diody je důsledkem rekombinace elektronů s dírami. Přímá rekombinace je jev, při kterém se energie elektronů, které přecházející z vodivostního pásu do valenčního, uvolní formou světelných kvant (fotonů), jedná se o tzv. zářivou rekombinaci. Při nepřímé rekombinaci dochází k uvolnění energie formou tepla (fonony). Kvantová účinnost LED diod pracujících ve viditelném spektru je oproti IR LED diodám daleko nižší. Je to způsobeno tím, že je u nich využíváno nepřímé rekombinace, zatímco u IR LED je využíváno přímé rekombinace. LED diody jsou zdrojem nekoherentního záření, což znamená, že vyzářené fotony mají různou frekvenci. Na rozdíl od laserových diod se vyznačují menší směrovostí a poměrně velkou spektrální šířkou. Mezi typické materiály pro luminiscenční diody patří GaAs, GaP, GaAlAs, GaAsP. Pro infračervené LED diody to jsou právě materiály GaAs a GaAlAs. [21]

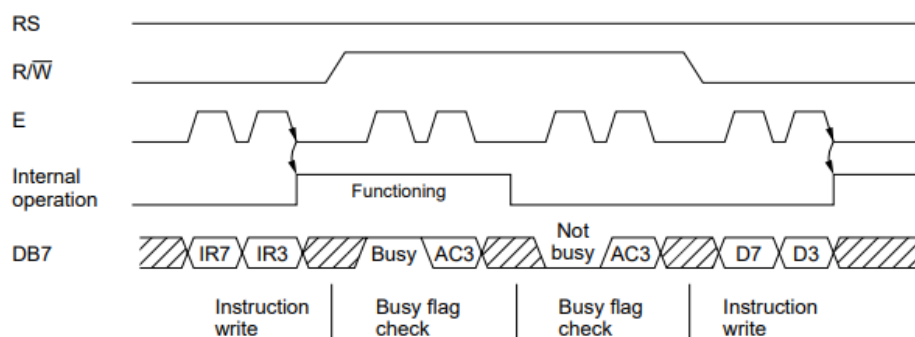
2.5 Zobrazovací jednotka - LCD displej

V této práci je využit dvouřádkový LCD displej (2x16 znaků) s řadičem HD44780. Princip činnosti LCD displeje je založen na natáčení tekutých krystalů, odtud ten název LCD - Liquid Crystal Display. K dispozici je kompletní znaková sada, která obsahuje základní alfanumerické znaky, řecké, německé, japonské znaky a další speciální znaky. Jednotlivé znaky se poté zobrazují v matici 5 x 8. Displej má celkem dohromady šestnáct vývodů. Osm vývodů, značených DB0 až DB7, slouží pro přenos dat. Pin RS slouží

k určení, zda jde o data nebo o instrukce. Pin R/W říká, zda se jedná o zápis nebo o čtení dat z displeje. Pin E řídí vzorkování doběžnou hranou signálu. Pin V_{ee} slouží k nastavení kontrastu. Nakonec už zbývají jen dva piny, to jsou piny napájecí a dva piny pro podsvětlení displeje. Pro komunikaci LCD displeje s mikrokontrolérem jsou dvě možnosti provedení. Prvním způsobem je využití všech osmi datových pinů. Druhou možností je využití jen čtyř datových pinů, konkrétně pinů DB4 až DB7, zbytek pinů je uzemněn. Princip 4bitové komunikace spočívá v tom, že data jsou vždy posílána nadvakrát. V této práci je využita právě tato 4bitová komunikace.[19], [20]

Lower 4 Bits	Upper 4 Bits	CG RAM (1)	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	(1)				0	1	F	\	P				-	9	3	α	ρ	
xxxx0001	(2)		!	1	Q	a	q						◻	7	4	ä	q	
xxxx0010	(3)		"	2	B	R	b	r					"	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s					」	ウ	テ	モ	ε	ω
xxxx0100	(5)		\$	4	D	T	d	t					、	エ	ト	μ	Ω	
xxxx0101	(6)		%	5	E	U	e	u					・	オ	ナ	1	σ	Ü
xxxx0110	(7)		&	6	F	V	f	v					ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w					ア	キ	ヌ	ラ	q	π
xxxx1000	(1)		<	8	H	X	h	x					イ	ク	ネ	リ	∫	×
xxxx1001	(2)		>	9	I	Y	i	y					ウ	ケ	ル	ル	'	∫
xxxx1010	(3)		*	:	J	Z	j	z					エ	コ	ハ	レ	j	∫
xxxx1011	(4)		+	;	K	C	k	c					オ	サ	ヒ	ロ	*	∫
xxxx1100	(5)		,	<	L	¥	l	l					ヤ	シ	フ	ワ	φ	∫
xxxx1101	(6)		-	=	M	J	m	>					ユ	ズ	ハ	ン	モ	÷
xxxx1110	(7)		.	>	N	^	n	→					ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?	O	_	o	←					ッ	リ	マ	°	ö	■

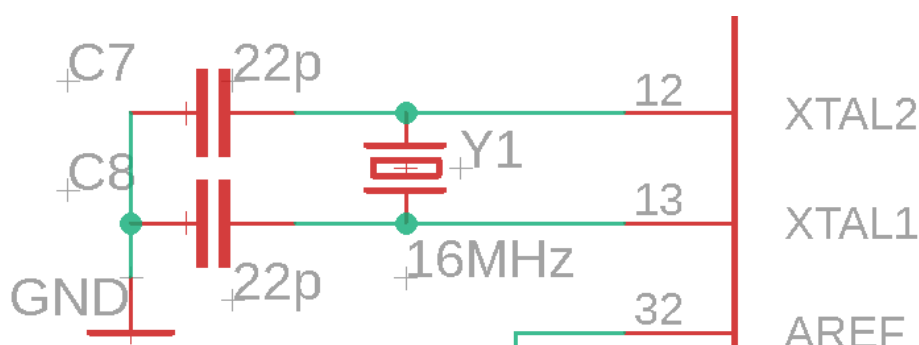
Obr. 8: Znaková sada HD44780(typ A00) [17]



Obr. 9: Ukázka 4-bitové komunikace [17]

2.6 Krystalový oscilátor 16 MHz

Krystalový oscilátor se u ATmega 16A připojuje na piny XTAL1 a XTAL2. Tyto piny představují vstup a výstupního invertujícího zesilovače vnitřního krystalového oscilátoru. Společně s krystalem je potřeba mezi tyto piny také zapojit dva kondenzátory (12 - 22pF). Oproti vnitřnímu RC oscilátoru, který mikrokontrolér obsahuje, zajistí připojený krystalový oscilátor daleko větší přesnost a stabilitu kmitočtu. Vzhledem k této aplikaci je přesnost oscilátoru velice důležitá.[9]



Obr. 10: Zapojení krystalu

3 Návrh a vývoj hardwaru a softwaru

3.1 Hardwarová část

V této části je popsána hardwarová stránka projektu, to znamená veškerá elektronika a konstrukce celého zařízení.

3.1.1 Napájení

Napájení chronometru obstarává 9V baterie. Toto napětí z baterie je připojeno na vstup lineárního napěťového stabilizátoru 7805. Na výstupu tohoto stabilizátoru je k dispozici stejnosměrné napětí o hodnotě 5V. Toto napětí napájí řídicí jednotku ATmega16A, LCD displej, dále LED diody optických závor a samozřejmě i senzory OPL550A. Teoreticky je možné přivést na vstup stabilizátoru napětí z rozmezí 7,5V až 25V. Stabilizátor je nutné, dle datasheetu, doplnit kondenzátory. K ochraně před prepólováním je zapojení doplněno dvojicí usměrňovacích diod.

3.1.2 Optoelektronické závory

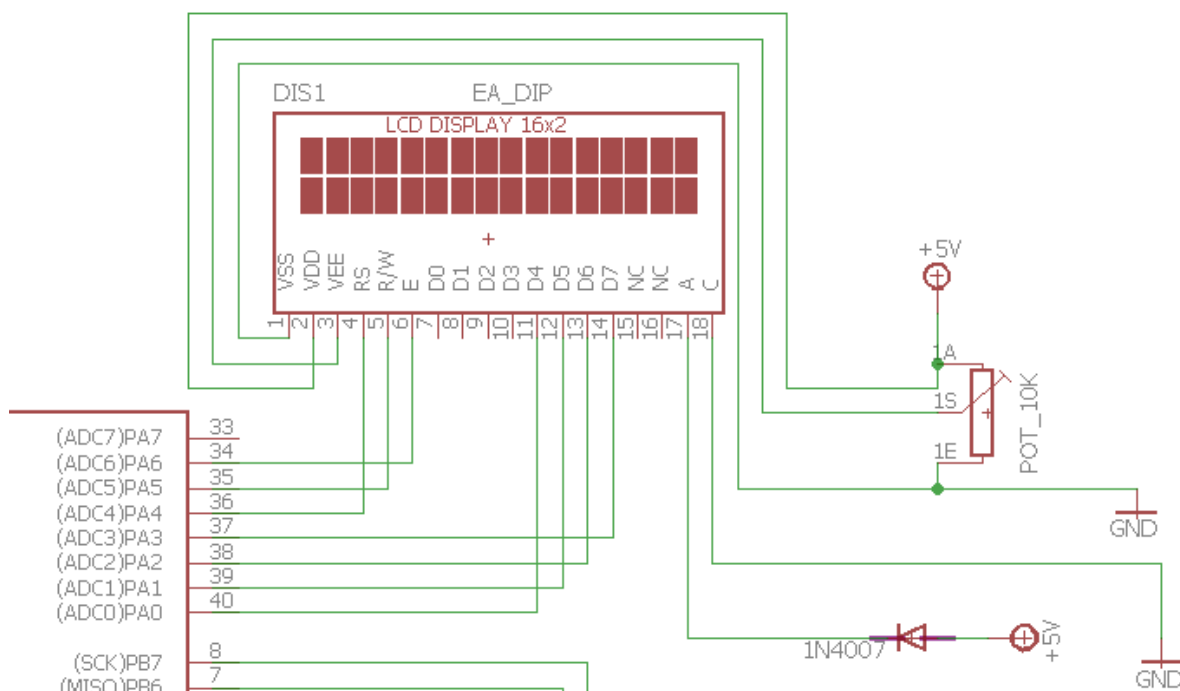
Závory se skládají z vysílačů - IR LED diody a přijímačů - OPL 550A. Parametry a vlastnosti těchto komponentů jsou již uvedeny v teoretické části práce. V této kapitole bude popsána výsledná konstrukce závor. Výsledná konstrukce závor poskytuje dostatečně velký snímací prostor, ve kterém je jisté zaregistrování letícího projektilu. Tato snímací plochy rozměry 40 x 75 mm. Vzhledem ke specifčnosti zařízení bylo nutné si vyrobit závory s pomocí 3D tisku.

3.1.3 Zapojení mikrokontroléru ATmega 16A

K řízení je použit mikrokontrolér ATmega16A, který je mozkiem celého zařízení. U portu A jsou využity piny PA0 až PA7, pomocí kterých je připojen LCD displej. Piny na portu B, PB5 až PB7, jsou použity pro ISP programování. Piny PB5 a PB6 jsou určeny pro přenos dat, tedy MISO a MOSI linky. Pin PB7 je určen pro synchronizační hodinový signál SCK. Piny PD2 a PD3 na portu D jsou piny externího přerušení INT0 a INT1. Na INT0 je přiváděn signál z první optické závory, který v případě zaclonění senzorů projektilem, startuje čítání časovače. Časovač přestane čítat v okamžiku, kdy projektil zacloní senzory druhé závory.

3.1.4 Zapojení LCD displeje

LCD displej s řadičem HD44780 byl již popsán v teoretické části. V této práci je využita 4bitová komunikace s mikrokontrolérem ATmega16A. Čtyřbitová, protože jsou využity jen čtyři datové piny z celkových osmi, konkrétně to jsou piny D4 až D7. Pro co nejlepší čitelnost, lze potenciometrem regulovat podsvícení displeje. LCD displej je zapojen dle datasheetu výrobce a částečně dle [18]. Ze zdroje [18] je využita knihovna pro ovládání LCD displeje. Vzhledem ke konstrukci zařízení se displej nenachází přímo na DPS, ale je propojen s mikrokontrolérem pomocí propojovacích vodičů a konektoru, který je umístěn na DPS. Detailní zapojení displeje lze vidět na obrázku níže.

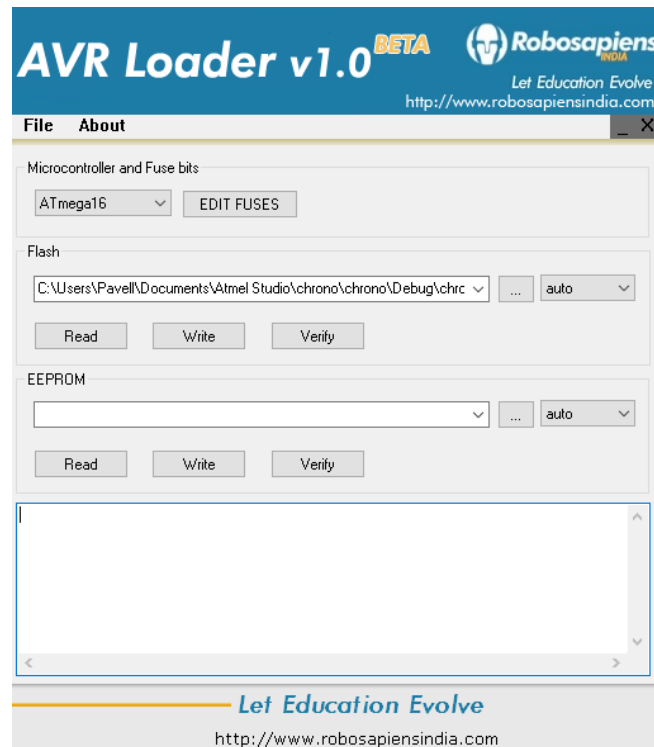


Obr. 11: Zapojení LCD displeje

3.1.5 ISP programování

Jak již bylo v teorii popsáno, jedná se o metodu, která umožňuje programování obvodů přímo v zařízení. To má obrovskou výhodu, když je třeba nahrát jiná verze programu do MCU.

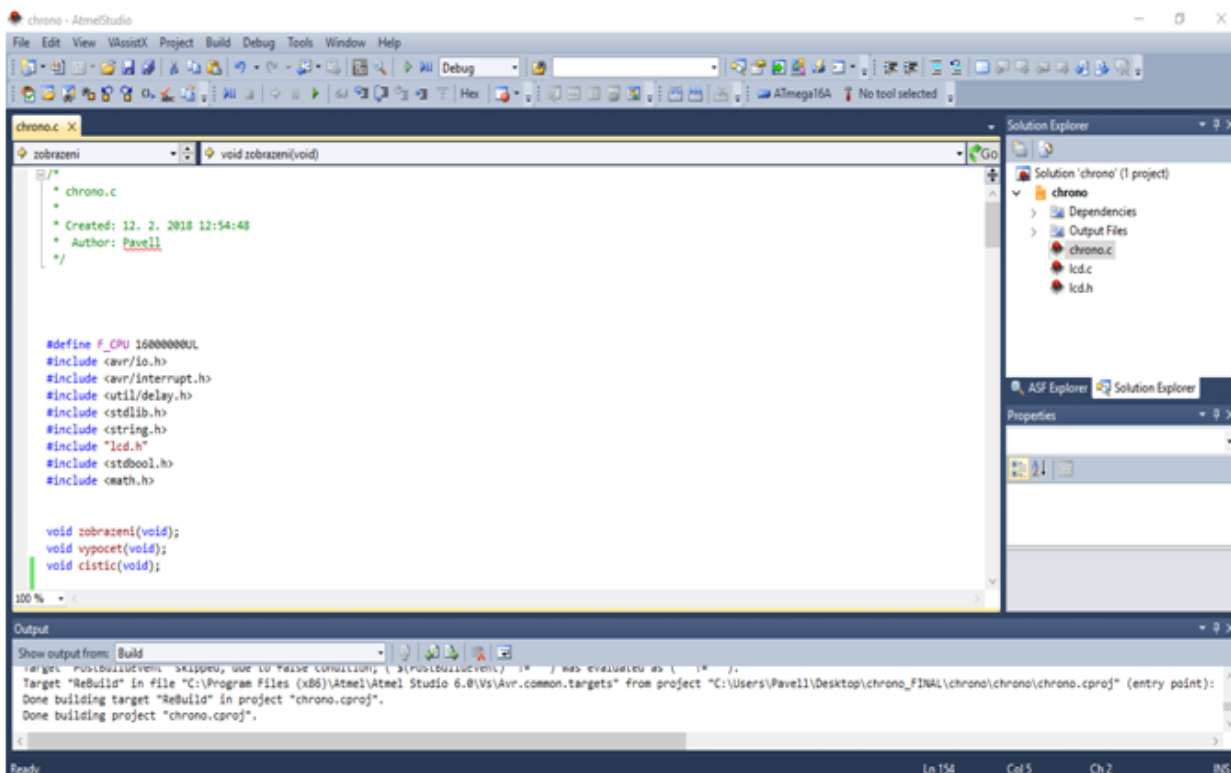
Pro nahrávání programu do mikrokontroléru byl použit programátor USBasp, který se jedním koncem zapojí do USB slotu počítače a druhý konec se zapojí do pinheaderu na DPS. Dále pro nahrání programu do mikroprocesoru posloužil příslušný software, v této práci jsem použil konkrétně AVR Loader v1.0. Do příslušné kolonky se vloží zvolený hexadecimální kód (soubor HEX) a program se nahraje do mikrokontroléru.



Obr. 12: Prostředí softwaru AVR Loader

3.2 Software

Program pro chronograf je napsaný v programovacím jazyku C. Zdrojový kód byl vytvořen v programu Atmel Studio 7.0, který je volně dostupný a stažitelný. V následujících podkapitolách bude popsáno konkrétní použití různých funkcí a specifických příkazů v této práci.



Obr. 13: Prostředí Atmel Studia 7.0

3.2.1 Externí přerušení

Externí přerušení INT0 a INT1 mají druhou a třetí nejvyšší prioritu, hned po přerušení vyvolané signálem reset. Jeho funkci jsem využil v projektu tak, že mi zajišťuje, to že když projektil proletí skrz optickou závoru, tak na pinu přerušení INT0 nastane změna. Logika senzorů je taková, že pokud je některý ze senzorů zacloněn, tak se na jeho výstupu objeví logická nula. V případě první optické závory, která startuje čítání čítače A přerušení zareaguje na tuto změnu a okamžitě zastaví běh programu a skočí do námi požadovaného řádku.

- *Ukázky ze zdrojového kódu*

//nastavení externího přerušení

```
MCUCR |= (1 << ISC01);           // Dobežná hrana INT0
MCUCR |= (1 << ISC11);           // Dobežná hrana INT1
GICR  |= (1 << INT1) | (1 << INT0); // Povolit přer. od INT1 a INT0
```

```
//Externí přerušení INT0 (SENZOR1_START)

ISR (INT0_vect) // Start
{
    TCCR2 |= ( 1 << CS20);

    GICR &= ~((1 << INT0)); // Zakázat přerušení
}

```

3.2.2 Využití čítače / časovače

Čítač je použit k odměření časových intervalů, a je nastaven do CTC módu. To znamená, že se porovnává hodnota z registru TCNT2 s nastavenou hodnotou v OCR2. Když se hodnota načítaných pulzů shoduje s nastavenou hodnotou, nastane přerušení. Přerušení jsem zvolil jednou za 10 mikrosekund, jak lze vidět ve výpočtech níže.

$$T = \frac{1}{f} = \frac{1}{16 \cdot 10^6} = 6,25 \cdot 10^{-8} s$$

$$\text{Počet pulzů} = \frac{T_{IK}}{T} = \frac{10 \cdot 10^{-6}}{6,25 \cdot 10^{-8}} = 160 - 1 = 159$$

- *Ukázky ze zdrojového kódu*

```
//nastavení časovače

TCCR2 |= ( 1 << WGM21 ) | ( 0 << WGM20 ); // CTC mod
OCR2 = 160-1; // TOP porovnání s TCNT2
TCCR2 &= ~ ( ( 1 << CS20 ) ); // clk 0
TIMSK |= ( 1 << OCIE2 ); // povolení přerušení
sei();

//Přerušení čítače

ISR(TIMER2_COMP_vect) //přerušení čítače
{
    desetMikroSek ++;

}

```

3.2.3 Výpočet rychlosti

Výpočet rychlosti vychází z podílu dráhy a času. Dráha je pro nás v tomto případě konstantou, s pomocí časovače / čítače zjišťujeme čas. Níže je funkce *void vypocet(void)*, která obstarává tento výpočet.

- *Funkce pro výpočet rychlosti*

```
//Výpočet rychlosti
```

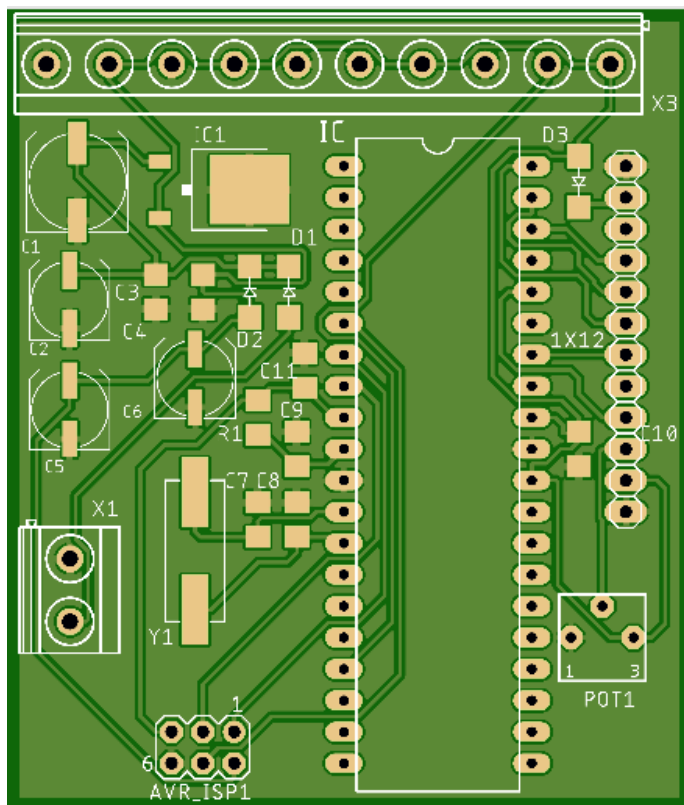
```
void vypocet(void){  
  
    rychlost = draha / (desetMikroSek * pow (10,-5)) ;  
    dtostrf(rychlost,6,3,vypis);  
    stop=false;  
  
    return;  
  
}
```

3.3 Návrhy jednotlivých DPS

Celý systém obsahuje dohromady pět DPS, přičemž jednu hlavní centrální DPS a dvě dvojice menších DPS pro senzory a LED diody. Všechny DPS jsou navrženy v softwaru Eagle 9.6.2 (free verze). DPS jsou oboustranné, díky čemuž lze dosáhnout menších rozměrů DPS. Materiál DPS je FR4 a tloušťka desek je 1,55 mm. Tloušťka mědi je 35 μm. Jsou využity jak součástky pro SMT montáž, tak i klasické vývodové THT součástky. Jako povrchová úprava DPS byla zvolena technologie HASL s olovem. Všechny DPS jsou zhotoveny asijskou společností JLCPCB. Zvolil jsem asijského výrobce z finančních důvodů, jelikož ceny za DPS jsou nesrovnatelně nižší než u českých firem.

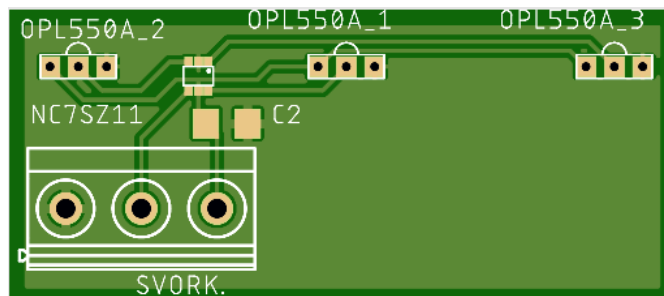
Centrální DPS, kde je hlavním komponentem mikrokontrolér, má rozměry 65 x 55 mm a obsahuje 23 součástek. Na DPS nejsou konstrukční otvory pro umístění, a to z důvodu, že je krabice poměrně rozměrná oproti velikosti DPS a nejrozumnější variantou bylo DPS přilepit. Na této DPS se nachází velká svorkovnice s 10-ti kontakty. Tato svorkovnice slouží pro napájení DPS s LED diodami a pro propojení DPS se senzory s centrální DPS. Dále je součástí 12-ti pinový konektor pro připojení LCD displeje.

Vzhledem k celkovému provedení zařízení nebylo příliš vhodné umístit displej přímo na DPS. Dále obsahuje obvod, který zajišťuje napájecí napětí 5V, kterým jsou všechny DPS napájeny.



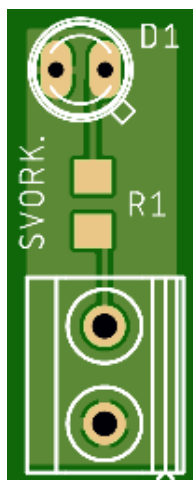
Obr. 14: Pohled shora na centrální DPS

DPS obsahující senzory OPL 550A má rozměry 45 x 20 mm. Deska je umístěna v postranních drážkách optických závor, které mají tvar písmene L.



Obr. 15: Pohled shora na DPS se senzory

Poslední DPS obsahuje optický vysílač, tedy LED diodu TSAL4400. Tato deska má rozměry 10 x 25 mm.



Obr. 16: Pohled shora na DPS s LED diodou

3.4 Ověřovací měření

Pro kontrolní otestování zařízení byla použita airsoftová krátká zbraň, u které výrobce udává orientační rychlost vystřelených projektilů až 90 m/s. Ovšem není upřesněno pro jakou hmotnost střeliva je tato rychlost uváděna.

Tab. 2 : Tabulka naměřených hodnot

Č. měření	Rychlost [m/s]
1.	85,714
2.	84,581
3.	84,956
4.	85,351
5.	84,958
6.	85,322
7.	84,631
8.	85,407
9.	85,128
10.	84,791

Aritmetický průměr naměřených hodnot rychlostí:

$$\bar{v} = \frac{v_1 + v_2 + \dots + v_n}{n} = \frac{850,8390}{10} = 85,0839 \text{ m/s}$$

Průměrná odchylka:

$$\Delta v = \frac{|\bar{v} - v_1| + |\bar{v} - v_2| + \dots + |\bar{v} - v_n|}{n} = \frac{3,0050}{10} = 0,3005 \text{ m/s}$$

Určení relativní průměrné odchylky měření (v procentech):

$$\delta_{v\%} = \frac{\Delta v}{\bar{v}} \cdot 100 = \frac{0,3005}{85,0839} \cdot 100 = 0,3532 \%$$

Rychlost pro těchto provedených 10 měření by se dala zapsat takto:

$$v = (85,0839 \pm 0,3005) \text{ m/s}$$

Závěr

Cílem této bakalářské práce bylo zkonstruovat balistický chronometr, tedy zařízení určené k měření rychlosti vystřelených projektilů.

V první kapitole jsou popsány možnosti realizace snímání vystřelených projektilů ze zbraně. Tento rozbor je poměrně krátký, a to z důvodu, že ostatní alternativní řešení byla spíše jen teoretická. Využití optických senzorů zde bylo téměř jednoznačné.

Zhotovené zařízení se skládá ze dvou optických závor, přičemž každá je sestavena z vysílače a přijímače. Naměřená rychlost projektilu je zobrazována na LCD displeji. K řízení je využit mikrokontrolér Atmega16A, který je naprogramovaný v jazyku C.

Finálním výrobkem je balistický chronometr schopný měření rychlosti vystřelených projektilů různých ráží. Kontrolní měření bylo provedeno s airsoftovou zbraní, u které dosahuje rychlost projektilů až 90 m/s. Z výsledků měření vyplívá, že průměrná odchylka byla přibližně 0,3 m/s a relativní průměrná odchylka v procentech byla přibližně 0,35 %.

Odhadované náklady na zhotovení chronometru jsou 1200 až 1500 Kč, což je zhruba třetinová až čtvrtinová cena oproti typické ceně komerčního chronometru.

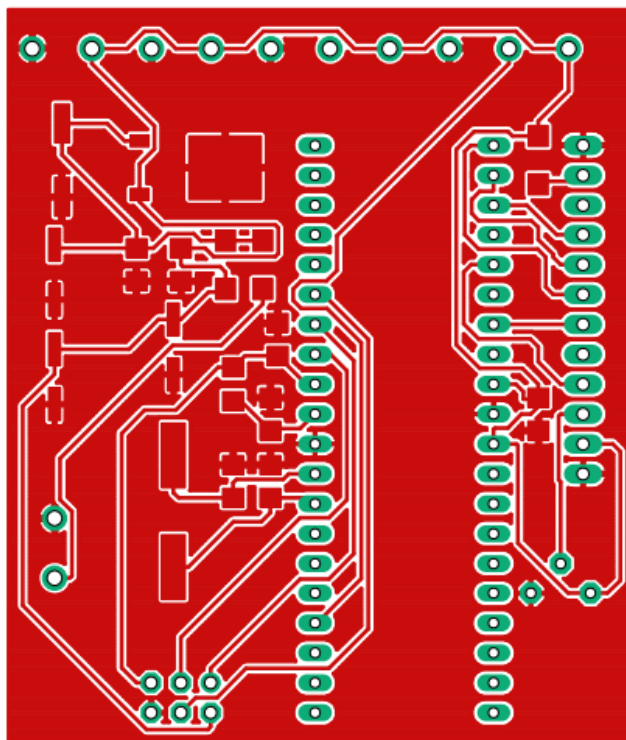
Seznam literatury a informačních zdrojů

- [1] JENČÍK, Josef, VOLF, Jaromír a kol. *Technická měření*. Vydavatelství ČVUT, Praha 2003. 212 s. ISBN 80-01-02138-6
- [2] ELUC. *Snímače polohy* [online]. [Cit. 26.4.2021]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/960>
- [3] FORMÁNEK, Josef. *Základní charakteristika a demonstrování základních principů měření veličin* [online]. [Cit. 26.4.2021]. Dostupné z: <http://home.zcu.cz/~formanek/mmvyuka/Data/ivk-mt-soubory/01-F.pdf>
- [4] Elektro. *Nové ultrazvukové senzory pro průmyslovou automatizaci* [online]. [Cit. 26.4.2021]. Dostupné z: <http://www.odbornecasopisy.cz/elektro/casopis/tema/nove-ultrazvukove-senzory-pro-prumyslovou-automatizaci--14485>
- [5] Matfyz. *Zajímavosti z metrologie VII: Dopplerovské radary* [online]. Poslední změna 1.10.2018. [Cit. 26.4.2021]. Dostupné z: <https://www.matfyz.cz/clanky/zajimavosti-z-meteorologie-vii-dopplerovske-radary>
- [6] OLIVKA, Petr. *Procesory CISC a RISC* [online]. [Cit. 1.5.2021]. Dostupné z: <https://poli.cs.vsb.cz/edu/arp/down/procrisc.pdf>
- [7] POKORNÝ, Martin. *Úvod do mikrokontrolérů ATMEL AVR* [online]. [Cit. 1.5.2021]. Dostupné z: <http://noel.feld.cvut.cz/vyu/scs/prezentace2005/ATmega/ATmega.pdf>
- [8] Microchip. *ATmega16A* [online]. [Cit. 1.5.2021]. Dostupné z: <https://www.microchip.com/wwwproducts/en/ATmega16A>
- [9] Microchip. *Datasheet ATmega16A* [online]. 2014. [Cit. 1.5.2021]. Dostupné z: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet.pdf
- [10] DUDÁČEK, Karel. *Sériová rozhraní SPI, Microwire, I²C a CAN* [online]. 2002. [Cit. 3.5.2021]. Dostupné z: http://home.zcu.cz/~dudacek/NMS/Seriova_rozhrani.pdf
- [11] Root. *Externí sériové sběrnice SPI a I²C* [online]. Poslední změna 30.12.2008. [Cit. 5.5.2021]. Dostupné z: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>
- [12] Hobbytronics. *AVR ISP Connectors* [online]. [Cit. 5.5.2021]. Dostupné z: <https://www.hobbytronics.co.uk/avr-isp-connectors>
- [13] SOUEPL. *Mikrokontrolér ATMEL ATmega 32* [online]. [Cit. 5.5.2021]. Dostupné z: https://www.souepi.cz/wp-content/ucitele/valecka/at_mega_32_popis%20%28Opraveno%29.htm

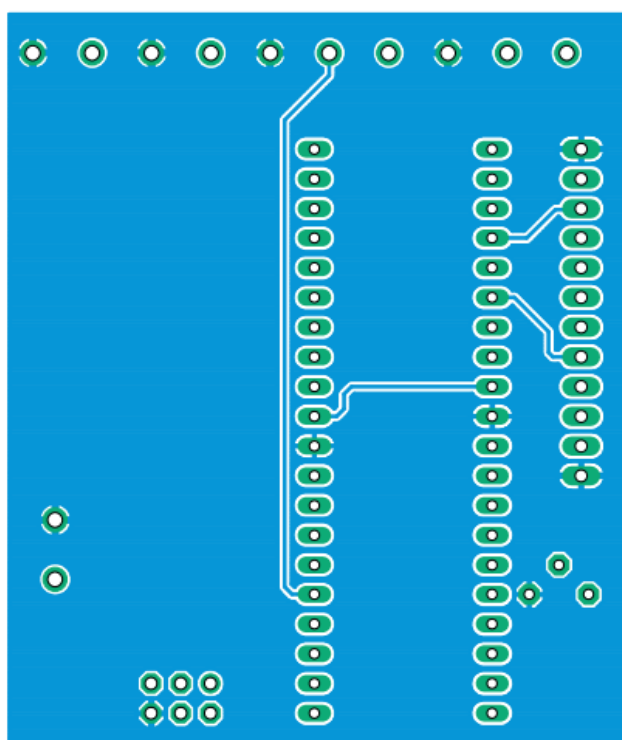
- [14] Tajned. *Základy ovládání mikrokontroléru ATmega(16/32) 11. Díl – Čítač / Časovač* [online]. [Cit. 10.5.2021]. Dostupné z: <http://www.tajned.cz/2014/10/zaklady-ovladani-mikrokontroleru-atmega-11-dil-citac-casovac-1-cast-zakladni-popis/>
- [15] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR ATmega 16 – 4. díl*. 1. vyd. Praha: BEN – technická literatura, 2006. 320 s. ISBN 80-7300-174-8
- [16] Ttelectronics. *Photologic Sensor OPL550 Datasheet* [online]. [Cit. 12.5.2021]. Dostupné z: <https://www.ttelectronics.com/TTElectronics/media/ProductFiles/Optoelectronics/Datasheets/OPL530-550-560.pdf>
- [17] Alldatasheet. *HD44780 Datasheet* [online]. [Cit. 12.5.2021]. Dostupné z: <https://www.alldatasheet.com/datasheet-pdf/pdf/63673/HITACHI/HD44780.html?>
- [18] FLEURY, Peter. *Interfacing a HD44780 Based LCD to an AVR* [online]. [12.5.2021]. Dostupné z: <http://www.peterfleury.epizy.com/avr-lcd44780.html?i=1>
- [19] Elektronika kvalitně. *Ovládání znakových LCD s řadičem HD44780* [online]. [Cit. 12.5.2021]. Dostupné z: <http://www.elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatice.html>
- [20] Pandatron. *Ovládání znakových LCD s řadičem HD44780* [online]. [Cit. 12.5.2021]. Dostupné z: http://pandatron.cz/?685&ovladani_znakovych_lcd_s_radice_m_hd44780_%EF%BF%BD_1._dil
- [21] MARTINEK, Radislav. *Senzory v průmyslové praxi*. 1. vyd. Praha: BEN – technická literatura, 2004. 200s. ISBN 978-80-7300-114-4
- [22] MALÝ, Martin. *Hradla, volty, jednočipy* [online]. [Cit. 15.5.2021]. 1. vyd. Praha: CZ.NIC, 2017. 508 s. ISBN 978-80-88168-26-3. Dostupné z: https://knihy.nic.cz/files/edice/hradla_volty_jednocipy.pdf
- [23] OLIVKA, Petr. *Technologie výroby číslicových obvodů* [online]. 2010. [Cit. 17.5.2021]. Dostupné z: <https://poli.cs.vsb.cz/edu/apps/down/technologie.pdf>
- [24] PINKER, Jiří. *Mikroprocesory a mikropočítače*. 1. vyd. Praha: BEN – technická literatura, 2004. 160 s. ISBN 80-7300-110-1

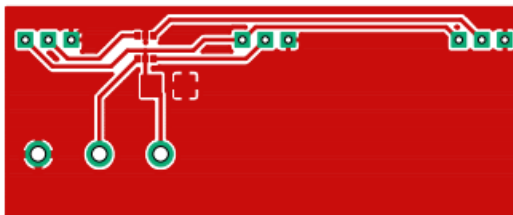
Příloha B – Vodivé vrstvy DPS

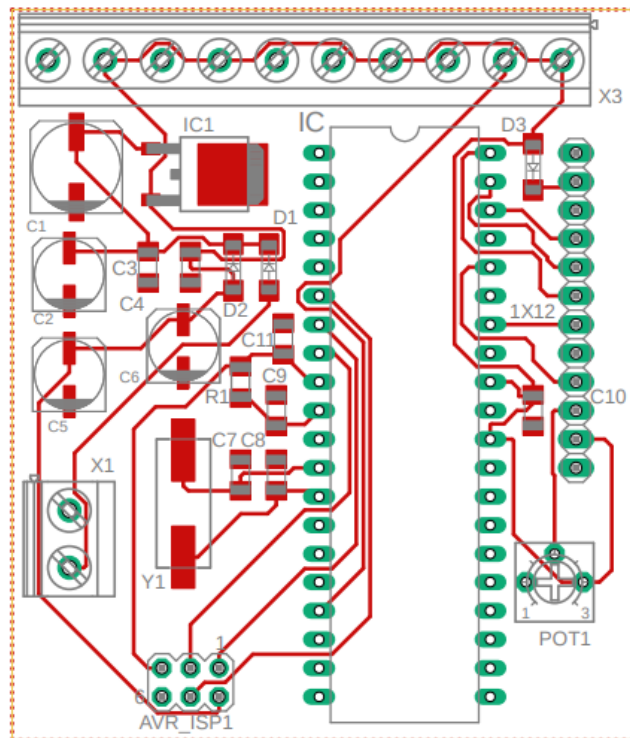
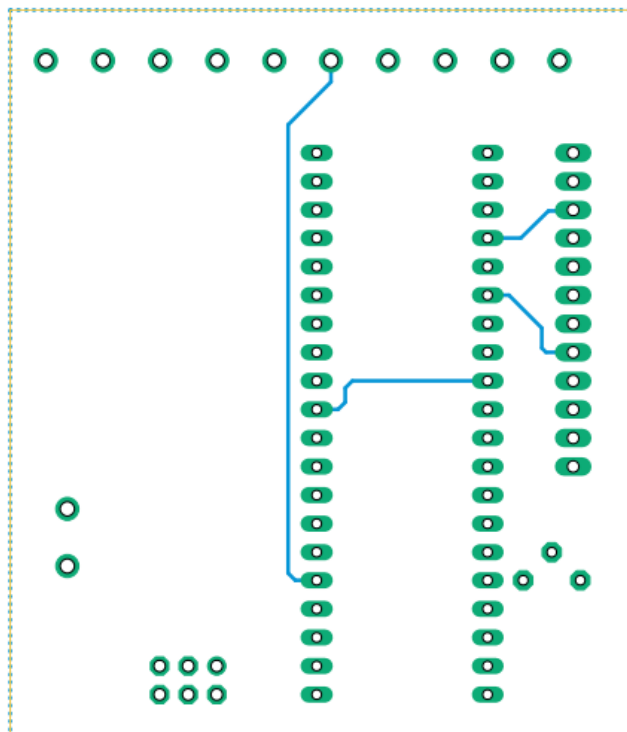
B.1 – Vodivá vrstva - TOP – centrální DPS

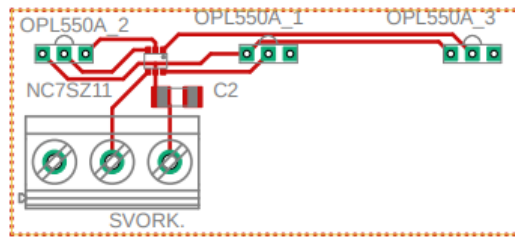
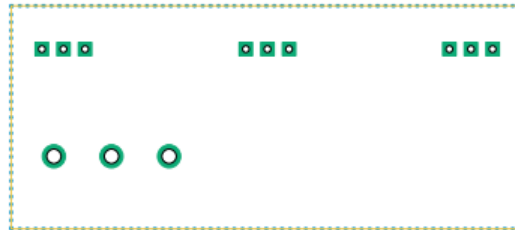
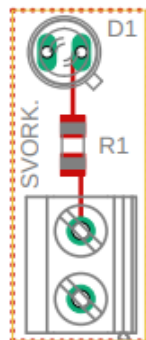
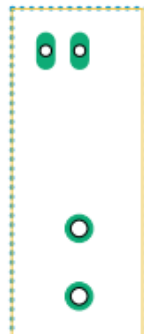


B.2 – Vodivá vrstva - BOTTOM – centrální DPS



B.3 – Vodivá vrstva - TOP – DPS se senzory**B.4 – Vodivá vrstva - BOTTOM – DPS se senzory****B.5 – Vodivá vrstva - TOP – DPS s LED diodou****B.6 – Vodivá vrstva - BOTTOM – DPS s LED diodou**

Příloha C – Osazovací výkresy**C.1 – Osazovací výkres - TOP – centrální DPS****C.2 – Osazovací výkres - BOTTOM - centrální DPS**

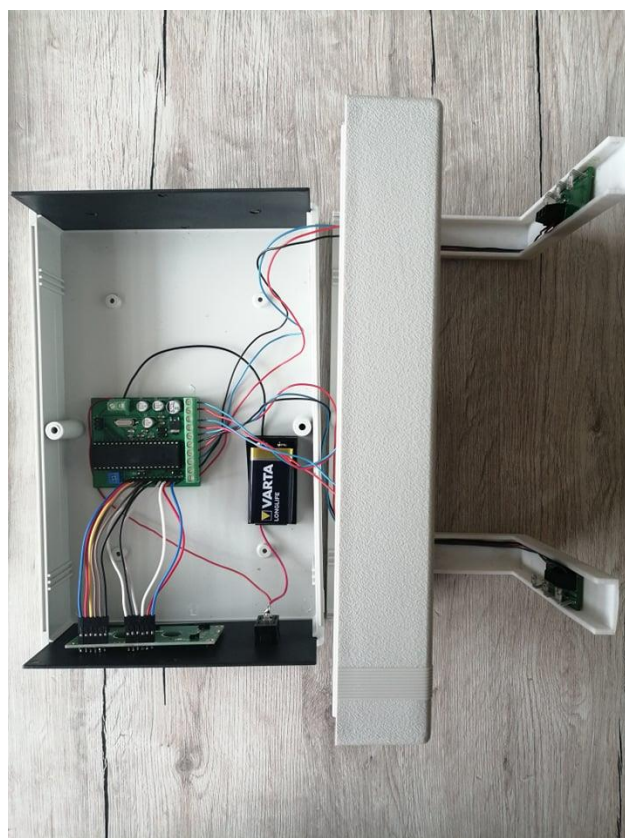
C.3 – Osazovací výkres - TOP - DPS se senzory**C.4 – Osazovací výkres - BOTTOM - DPS se senzory****C.5 – Osazovací výkres - TOP - DPS s LED diodou****C.6 – Osazovací výkres - BOTTOM - DPS s LED diodou**

Příloha D - Fotodokumentace

D.1 – Fotografie chronometru – pohled na zařízení zepředu



D.2 – Fotografie chronometru – vnitřek zařízení



Příloha E – Zdrojový kód

```

#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
#include <string.h>
#include "lcd.h"
#include <stdbool.h>
#include <math.h>

void zobrazeni(void);
void vypocet(void);
void cistic(void);

//Proměnné

volatile bool stop = false;
volatile bool stop1 = true;
volatile unsigned long desetMikroSek = 0 ;

volatile float rychlost = 0;

volatile uint8_t i = 0 ;

char vypis[100]={'\0'};

int main(void)
{
    for(i=0;i<100;i++)
    {
        vypis[i]='\0';
    }
    desetMikroSek=0;
    rychlost=0;

    // nastavení vstupu

    DDRD &= ~(1<<PD2)|(1<<PD3);
    DDRD |= (1<<PD7);

    // nastavení vstupu do log1

    PORTD |= (1<<PD2)|(1<<PD3)|(1<<PD7);

    //nastavení časovače

    TCCR2 |= ( 1 << WGM21 ) | ( 0 << WGM20 ); // CTC mod
    OCR2 = 160-1; // TOP porovnání s TCNT2 = 0;
    TCCR2 &= ~ (( 1 << CS20 )); // clk 0
    TIMSK |= ( 1 << OCIE2 ); // povolení přerušení
    sei();

    //nastavení externího přerušení

    MCUCR |= (1 << ISC01); // Dobežná hrana INT0
    MCUCR |= (1 << ISC11); // Dobežná hrana INT1
    GICR |= (1 << INT1) | (1 << INT0); // Povolení přerušení od INT1 a INT0

```

```
//spuštění displeje

    lcd_init(LCD_DISP_ON);
    lcd_clrscr();
    _delay_ms(50);

    lcd_gotoxy(0,0);
    lcd_puts("RYCHLOST [m/s]");

//nekonečný cyklus
while(1)
{
// volání funkcí
    if(stop)                //upravit ve finále
    {
        vypocet();
        zobrazeni();

        main();                //čistič();
    }
}

//Přerušení čítače

ISR(TIMER2_COMP_vect)      //přerušení čítače
{
    desetMikroSek ++;
}

//Externí přerušení INT0 (SENZOR1_START)

ISR (INT0_vect)            // start
{
    TCCR2|= ( 1 << CS20);

    GICR &= ~(1 << INT0);    // zakáže přerušeni
}

//Externí přerušeni INT1 (SENZOR2_STOP)

ISR (INT1_vect)           // stop
{
    TCCR2 &= ~(1 << CS20 ); // clk 0

    GICR &= ~(1 << INT1);    // zakáže přerušeni
    TCNT2 = 0;
    stop = true;
}
```

```
//Výpočet rychlosti
void vypocet(void){

    rychlost = 0.227 / (desetMikroSek * pow (10,-5)) ;
    dtostrf(rychlost,6,3,vypis);
    stop=false;

    return;

}

//Zobrazení na LCD
void zobrazeni(void){

    lcd_gotoxy(0,1);
    lcd_puts(vypis);

    _delay_ms(5000);

    lcd_clrscr();
    lcd_gotoxy(0,0);
    //lcd_puts("RYCHLOST [m/s]");
    lcd_puts(vypis);
return;

}

void cistic(void){

//další střelba

    GICR |= (1 << INT1) | (1 << INT0); // Povol přerušení od INT1 a INT0
    desetMikroSek=0;
    rychlost=0;

    for(i=0;i<100;i++)
    {
        vypis[i]='\0';
    }

    return;
}
```