

Světelná tabule pro zobrazování skóre řízená mikropočítačem

Filip Matějček

Bakalářská práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Filip Matějček**
Osobní číslo: **A11140**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Světelná tabule pro zobrazení skóre řízená mikropočítačem**
Téma anglicky: **An Electronic Scoreboard Display with a Microcontroller**

Zásady pro vypracování:

1. **Prostudujte a popište existující konstrukce tabulí pro zobrazení skóre při sportovních zápasech.**
2. **Navrhňte zapojení vlastního zobrazovače řízeného zvoleným mikropočítačem.**
3. **Realizujte prototyp zobrazovače podle svého návrhu.**
4. **Vytvořte programové vybavení pro použitý mikropočítač.**

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **BARR, Michael a Anthony J MASSA. Programming embedded systems. 2nd ed. Sebastopol: O'Reilly, 2006, xxi, 301 s. ISBN 978-0-596-00983-0.**
2. **CATSULIS, John. Designing Embedded Hardware. Sebastopol: O'Reilly Media, 2005. ISBN 978-0-596-00755-3.**
3. **HEROUT, Pavel. Učebnice jazyka C. 6. vyd. České Budějovice: Kopp, 2010, 271, viii s. ISBN 978-80-7232-383-8.**
4. **MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6.**
5. **NOVÁK, Petr. Mobilní roboty - pohony, senzory, řízení, Praha: BEN - technická literatura, 2005. ISBN/EAN 80-7300-141-1 / 9788073001414.**
6. **PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.**

Vedoucí bakalářské práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

6. března 2015

Termín odevzdání bakalářské práce:

22. května 2015

Ve Zlíně dne 6. března 2015



L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně 22.5.2015


.....
podpis

ABSTRAKT

Bakalářská práce se zabývá návrhem světelné tabule pro zobrazování skóre při fotbalových zápasech. Práce pojednává o technickém a elektronickém návrhu světelné tabule, která se skládá ze dvou sedmi-segmentových displejů a informace o poločase. Dále obsahuje návrh a kód programu pro řídicí jednotku tabule, která je řízena pomocí elektronické desky Arduino, ve které je osazen mikropočítač Atmel. Oba displeje jsou ovládány pomocí dálkového ovládání.

Klíčová slova: Světelná tabule, sedmi-segmentový displej, řídicí jednotka, Arduino, Atmel, dálkové ovládání

ABSTRACT

The bachelor thesis deals with the project of a scoreboard which displays a score during the football match. Thesis talks about the technical and electronic proposal of the scoreboard which is composed of two seven-segments displays and information about the half. It also includes the design and program code for the control unit of board which is control by electronic Arduino board which is equip with Atmel microcontroller. Both displays are controlled by remote control.

Keywords: Scoreboard, seven-segments display, control unit, Arduino, Atmel, remote control

V rámci této práce bych chtěl poděkovat panu Ing. Janu Dolinayovi, Ph.D. za poskytnutí pomoci při vypracovávání bakalářské práce.

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

OBSAH

| | |
|---|-----------|
| ÚVOD | 8 |
| I TEORETICKÁ ČÁST | 9 |
| 1 KONSTRUKCE TABULÍ | 10 |
| 1.1 KONSTRUKCE TABULÍ U RŮZNÝCH SPORTŮ | 10 |
| 1.1.1 Basketbal | 10 |
| 1.1.2 Fotbal a Ragby | 10 |
| 1.1.3 Florbal a Házená | 11 |
| 1.1.4 Lední hokej..... | 11 |
| 1.1.5 Volejbal | 12 |
| 1.1.6 Tenis | 12 |
| 1.1.7 Bojové sporty | 13 |
| 2 SEGMENTOVÝ DISPLEJ | 14 |
| 2.1 SEDMISEGMENTOVÝ DISPLEJ | 14 |
| 2.2 DEVÍTISEGMENTOVÝ DISPLEJ | 14 |
| 2.3 ČTRNÁCTISEGMENTOVÝ DISPLEJ | 15 |
| 2.4 PATNÁCTISEGMENTOVÝ DISPLEJ | 15 |
| 2.5 ŠESTNÁCTISEGMENTOVÝ DISPLEJ | 15 |
| 2.6 TECHNOLOGIE INDIKACE SEGMENTŮ | 16 |
| 2.6.1 LED (Ligth Emitting Diode) | 16 |
| 2.6.2 OLED (Organic Ligth Emitting Diode) | 16 |
| 2.6.3 Tekutý krystal (Liquid crystal)..... | 17 |
| 3 MIKROPOČÍTAČ | 18 |
| 3.1 SBĚRNICE | 18 |
| 3.1.1 Datová sběrnice | 18 |
| 3.1.2 Adresová sběrnice | 18 |
| 3.1.3 Řídící sběrnice..... | 18 |
| 3.2 MIKROPROCESOR | 19 |
| 3.3 PAMĚŤ PROGRAMU | 19 |
| 3.4 PAMĚŤ DAT | 19 |
| 3.5 PERIFERNÍ OBVODY | 19 |
| 4 PROGRAMOVÁNÍ MIKROPOČÍTAČŮ | 20 |
| 4.1 PROGRAMOVACÍ JAZYKY..... | 20 |
| 4.1.1 Assembler..... | 20 |
| 4.1.2 C | 20 |
| 4.1.3 C++..... | 20 |
| 4.1.4 Ada | 21 |
| 4.2 VÝVOJOVÉ PROSTŘEDÍ | 21 |
| 5 PLATFORMA ARDUINO | 22 |

| | | |
|-----------|--|-----------|
| 5.1 | ARDUINO UNO R3 | 22 |
| 5.2 | NAPÁJENÍ | 22 |
| 5.3 | PAMĚŤ..... | 22 |
| 5.4 | VSTUPY A VÝSTUPY | 23 |
| 5.5 | KOMUNIKACE..... | 23 |
| II | PRAKTICKÁ ČÁST | 24 |
| 6 | NÁVRH SVĚTELNÉ TABULE..... | 25 |
| 6.1 | NÁVRH ZAPOJENÍ SEMISEGMENTOVÉHO DISPLEJE | 25 |
| 6.2 | DÁLKOVÉ OVLÁDÁNÍ | 26 |
| 6.2.1 | Přijímač | 26 |
| 6.2.2 | Vysílač..... | 27 |
| 7 | REALIZACE ZAPOJENÍ..... | 28 |
| 7.1 | POPIS FUNKČNOSTI TABULE..... | 30 |
| 7.2 | POPIS PROGRAMU | 31 |
| 7.2.1 | Proměnné..... | 31 |
| 7.2.2 | Funkce setup()..... | 32 |
| 7.2.3 | Funkce chNum()..... | 32 |
| 7.2.4 | Funkce loop() | 33 |
| | ZÁVĚR | 36 |
| | SEZNAM POUŽITÉ LITERATURY..... | 38 |
| | SEZNAM OBRÁZKŮ | 40 |
| | SEZNAM PŘÍLOH..... | 41 |

ÚVOD

Už od dávných dob se lidé snažili měřit čas. Na začátku to bylo pouze střídání dne a noci, poté i fáze Měsíce a také změnu ročních období. Dalším poznatkem bylo vrhání stínu, díky kterému zaznamenávali už pouze délku dne. Jak čas šel a člověk vymyslel nové sporty, tak začal i shromažďovat statistiky a snažil se je nějak zaznamenávat, aby informoval ostatních lidí o průběhu utkání a jeho délce. Nejdříve textovým zápisem na tabuli pomocí křídly, ale v průběhu času se záznam vyvíjel a přišli nové způsoby.

V nynější době se začali používat modernější informační tabule, které se skládají z několika displejů, které používají LED technologii a stávají se tak spolehlivým informačním prostředkem, který se může elektronicky obsluhovat na dálku a spoří tak spotřební energii, ale také energii člověka.

Z těchto důvodů jsem se rozhodl navrhnout vlastní tabuli pro předávání informací sportovního utkání, která by se měla používat při fotbalových zápasech TJ Sokol Bystřice pod Lopeníkem.

I. TEORETICKÁ ČÁST

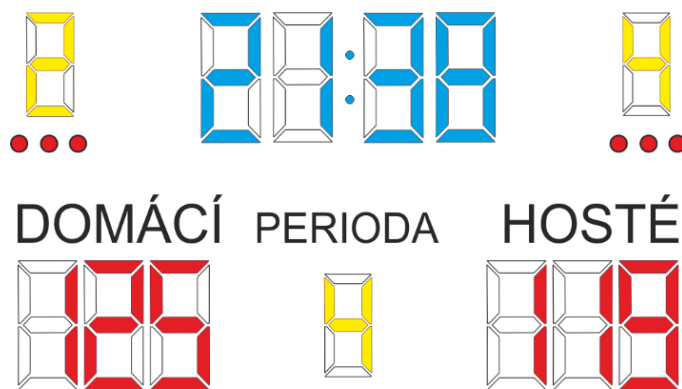
1 KONSTRUKCE TABULÍ

Světelné tabule se nejčastěji skládají s několika LED displejů, které informují nejen o stavu zápasu, ale používají se i v jiných oborech než u sportovních zápasů. Ve větších firmách se používají pro informování pracovníků o jejich produktivnosti za směnu či měsíc. Dalším příkladem, jsou benzinové pumpy, které díky nim informují o ceně paliva. Nejčastěji je však používají jako digitální hodiny na různých místech, autobusových nádraží, vlakových nádraží, nebo i ve školách. Dají se ovládat buď na dálku, nebo na přímo. U sportovních zápasů existují různé typy světelných tabulí, například univerzální tabule, které se dají požívat na většinu známých sportů a jsou tak více flexibilní. A však při každém sportu jsou odlišné formy zobrazení informací, které univerzální tabule nemusí obsahovat. Proto se nejčastěji používají ty tabule, které jsou vyrobeny přímo pro daný sport a zobrazují tak kompletní informace o zápase.

1.1 Konstrukce tabulí u různých sportů

1.1.1 Basketbal

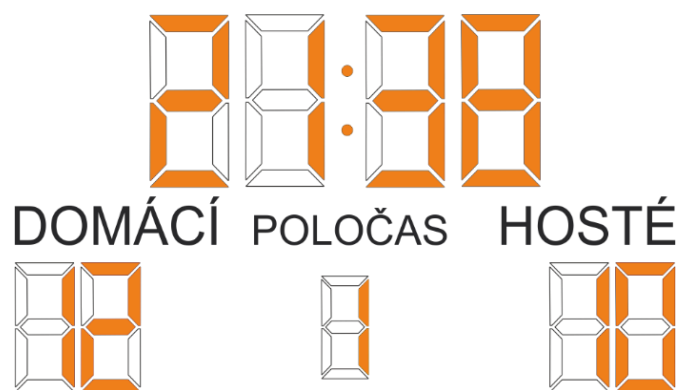
Pro Basketbal jsou různé typy světelných tabulí, základní výbavou většiny z nich je čas utkání, perioda (čtvrtina), skóre obou mužstev, počet a doba Time Out-ů.



Obrázek 1 - Znáznornění světelné tabule pro Basketbal

1.1.2 Fotbal a Ragby

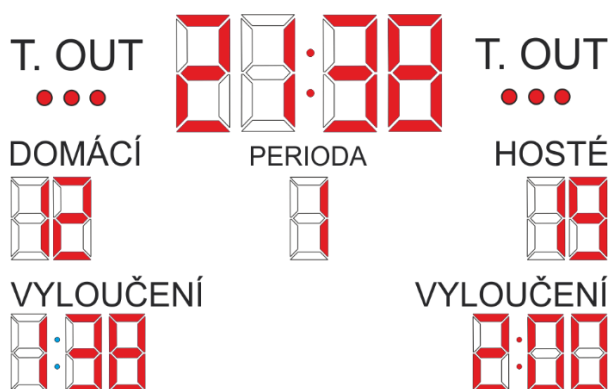
Tyto dva sporty mají nejjednodušší strukturu tabulí. Je na nich znázorněn pouze čas utkání, perioda (poločas) a skóre obou týmů.



Obrázek 2 - Znáznornění světelné tabule pro Fotbal a Ragby

1.1.3 Florbal a Házená

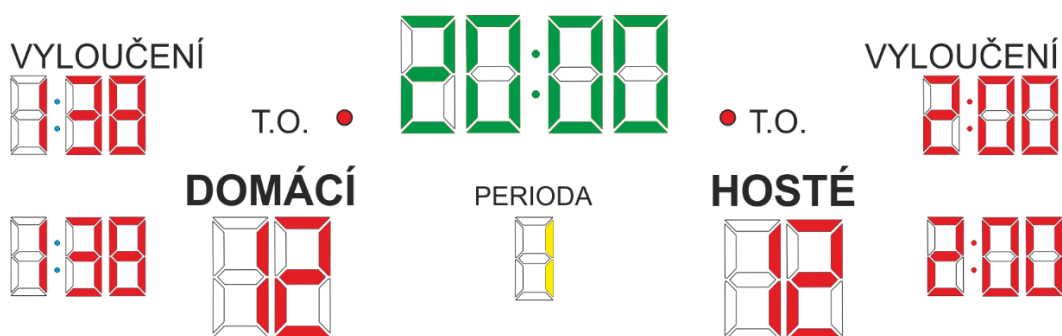
Každý se hraje jinak a má odlišné pravidla hry ovšem u obou těchto sportů se dají použít totožné světelné tabule. Většina se skládá opět z ukazatele času utkání, periody, počtu Time Out-ů a době trestného času s číslem hráče, který si jej odpykává.



Obrázek 3 - Znáznornění světelné tabule pro Florbal a Házenou

1.1.4 Lední hokej

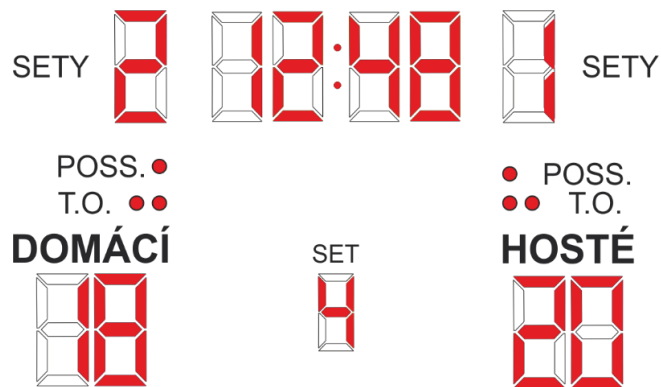
U ledního hokeje jsou totožné tabule s předchozím příkladem, ovšem je možné vyloučit více hráčů naráz, tedy u těchto světelných tabulí je více řádků pro vyloučené hráče.



Obrázek 4 - Znáznornění světelné tabule pro Lední hokej

1.1.5 Volejbal

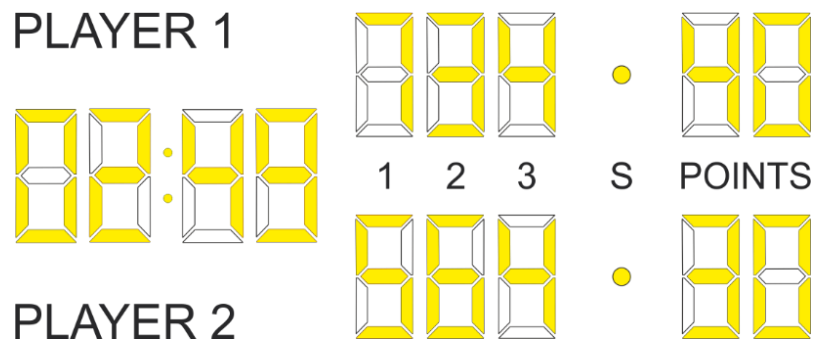
Světelné tabule pro volejbal jsou skoro totožné s ostatními, informují opět o času utkání, skóre obou týmů, počtu Time Out-ů mají však navíc číslo setu a počet odehraných setů, dále mohou obsahovat skóre jednotlivých setů.



Obrázek 5 - Znáznornění světelné tabule pro Volejbal

1.1.6 Tenis

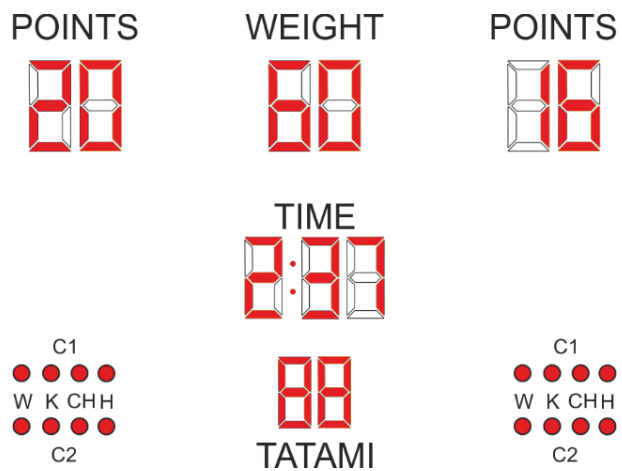
Tenis má odlišné zobrazení informací světelných tabulí. Některé tabule zobrazují jméno obou hráčů nebo pouze signalizaci, který hráč podává. Dalším ukazatelem je skóre v setech, buď ve 3, nebo v 5 a zobrazuje také stav hry.



Obrázek 6 - Znáznornění světelné tabule pro Tenis

1.1.7 Bojové sporty

Pro různé bojové sporty jsou světelné tabule ve směs podobné, zobrazují čas utkání, body obou soupeřů, váhovou kategorii, číslo kola a počet vyhraných kol.



Obrázek 7 - Znárodnění světelné tabule pro Judo

2 SEGMENTOVÝ DISPLEJ

Je elektronické, nebo může být i mechanické zařízení, které slouží pro zobrazování různých znaků nebo číslic. Skládá se z několika segmentů, které mají dva stavy buď je segment sepnut, nebo vypnut. Různými kombinacemi sepnutých segmentů dostaneme znak, který chceme na výstupu zobrazit. Elektronické segmentové displeje používají pro indikaci sepnutého segmentu buď LED technologii, OLED, nebo se také používají tekuté krystaly. Mají velkou škálu použití, nejčastěji se nachází v elektronických zařízeních, které potřebují zobrazovat nějaké informace, například v digitálních hodinkách, kalkulačky, budík a také v různých světelných tabulích. [7]

2.1 Sedmisegmentový displej

Nejpopulárnější a nejjednodušší displej z jednoduchým zapojením a ovládním. Skládá se ze 7 segmentů, které umí zobrazovat dekadické číslice od 0 do 9, někdy je se na displeji nachází i tečka. [7]



Obrázek 8 - Sedmisegmentový displej

2.2 Devítisegmentový displej

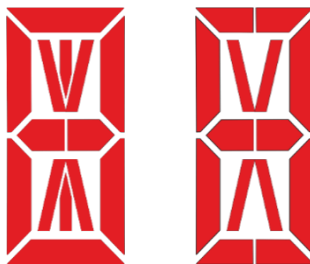
Obsahuje o dva šikmé segmenty navíc oproti předchozímu displeji, se kterými se pohodlněji zobrazují číslice, ale pro zobrazení textu jsou tyto dva displeje stále nevhodné. [7]



Obrázek 9 - Devítisegmentový displej

2.3 Čtrnáctisegmentový displej

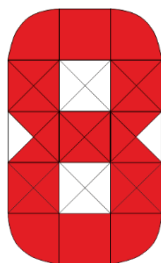
Složen ze čtrnácti segmentů, ze kterých jsou 4 segmenty šikmé a směřují do středu znaku. Existují dvě verze tohoto displeje, se středními vertikálními segmenty a rozpojeným středním horizontálním segmentem, druhá verze neobsahuje střední vertikální segmenty, ale má rozdvojené všechny horizontální segmenty. Zobrazí číslice, ale také písmena latiny. Používá se ve výtazích a autorádiích. [7]



Obrázek 10 - Čtrnáctisegmentový displej

2.4 Patnáctisegmentový displej

Segmenty má uspořádané tak aby rozlišoval ostré hrany u číslic nebo oblouky znaků. Používá se u návěstidel, nebo u informačních tabulí o příjezdu či odjezdu vlaků. [7]



Obrázek 11 - Patnáctisegmentový displej

2.5 Šestnáctisegmentový displej

Má stejné rozložení segmentů jakou čtrnáctisegmentového displeje, pouze má navíc dva vertikální střední segmenty. Je přezdíváný jako „Union Jack“ díky tomu, že se podobá vlajce Spojeného království. Zobrazuje číslice i písmena latiny a používá se v autorádiích, výtazích či multimetrech. [7]



Obrázek 12 - Šesnásctisegmentový displej

2.6 Technologie indikace segmentů

2.6.1 LED (Light Emitting Diode)

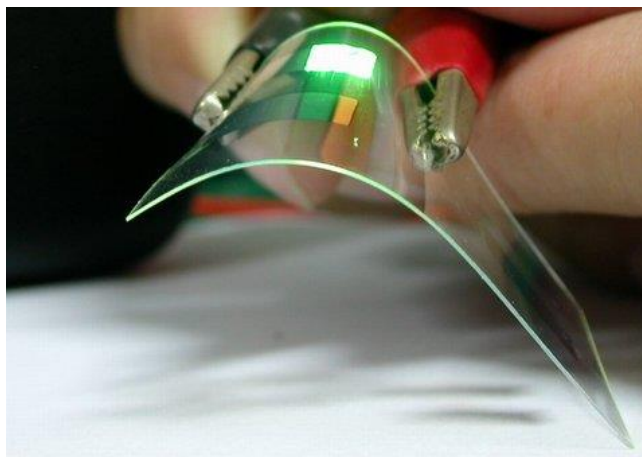
Jedná se o polovodičovou součástku obsahující přechod P-N, která slouží k vyzařování světla a také infračerveného, nebo ultrafialového záření, čím se odlišuje od ostatních diod. Používá se pro světelnou signalizaci v různých zařízeních. Český název pro LED je elektroluminiscenční dioda, světelná dioda, ale nejčastěji se používá ledka. [8]



Obrázek 13 - Sedmissegmentový displej s LED podsvícením [12]

2.6.2 OLED (Organic Light Emitting Diode)

Je typ displeje, který používá technologii organických luminiscenčních diod. Skládá se z průhledné anody a kovové katody, mezi kterými je několik vrstev organické látky. Nejčastěji se používá v mobilních telefonech. [9]



Obrázek 14 - Displej s OLED technologií [9]

2.6.3 Tekutý krystal (Liquid crystal)

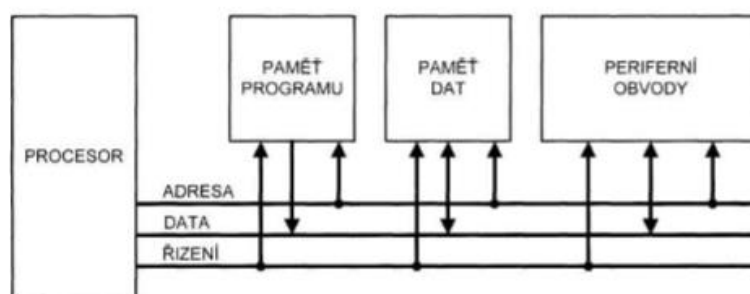
Je hmota, která má vlastnost měnit svůj stav z tekutého na pevný a zpět. Existují různé druhy tekutých krystalů, u elektronických segmentových displejích se používají termotropní nematické krystaly, které mění své optické vlastnosti při působení elektrického pole. Nejčastěji se používají u LCD displejů. [10]



Obrázek 15 - Displej z tekutých krystalů [13]

3 MIKROPOČÍTAČ

Jak název napovídá, jedná se o klasický počítač, ovšem, který je svými rozměry daleko menší, o čemž vypovídá předpona mikro. Toto ale neznamená, že by mikropočítač ztrácel na výkonu, díky vysoké integraci dosahují mikropočítače velké výpočetní výkonosti a funkční variabilitě. Skládá se z několika základních částí, mikroprocesoru, paměti programu, paměti dat a z periferních obvodů. [6]



Obrázek 16 - Schéma mikropočítače [6]

3.1 Sběrnice

Sběrnice slouží k propojení jednotlivých částí s procesorem a z pravidla jsou 3 druhy sběrnic. [6]

3.1.1 Datová sběrnice

Slouží k přenosu dat a její šířka u 8 bitových mikropočítačů je většinou 1 byte, dá se využívat pro příjem dat, nebo pro odesílání dat, případně střídáním obou variant. [2]

3.1.2 Adresová sběrnice

Další sběrnici je adresová, která slouží pro adresování paměti a rozlišováním mezi jednotkami, které jsou připojeny na datovou sběrnici. Šířka adresové sběrnice je nejčastěji 16 bitů. [2]

3.1.3 Řídící sběrnice

Poslední sběrnici je řídicí, která řídí čtení, zápis a další aktivity jednotek. Většina těchto signálů jsou řízena mikroprocesorem, ale některé jsou generovány i ostatními jednotkami. [2]

3.2 Mikroprocesor

Je nejdůležitější částí mikropočítače, který řídí veškerou jeho činnost. Zajišťuje správnou funkčnost provádění instrukcí, které jsou uloženy v paměti programu, řídí zpracování dat v paměti, řídí také tok dat z mikropočítače ven přes výstupní obvody a také tok dat přes vstupní obvody do mikropočítače. [6]

3.3 Paměť programu

V této paměti se nacházejí instrukce, které při postupném provádění zajišťují požadovanou činnost mikropočítače, dále se zde nachází různé konstanty nebo tabulky, které jsou používány v programu. Pokud se program nemění, používá se pro uložení paměť ROM (EPROM, EEPROM, FLASH). Při testování programu je potřeba program neustále měnit a proto se používají paměti typu RAM, ovšem u těchto variant jsou mikropočítače vybaveny menší programovou pamětí typu ROM, u kterých se při připojení napájení začne provádět program právě zde a vyvolá čtení z velkokapacitní diskové paměti, ze které se nahraje program, který přebere řízení mikropočítače. [2]

3.4 Paměť dat

Slouží pro dočasné uložení dat, získaných z výstupních obvodů nebo také pro uložení mezi-výsledků výpočtů. Typ těchto pamětí je vždy typu RAM. [2]

3.5 Periferní obvody

Mezi ně patří vstupní a výstupní obvody, které slouží ke komunikaci s vnějšími prostředky. Mikropočítač může obsahovat různé množství vstupně/výstupních bran, tedy připojovacích míst, rozlišených adresově. [2]

4 PROGRAMOVÁNÍ MIKROPOČÍTAČŮ

Je nespočet firem, které vyrábějí mikropočítače, z těch nejznámějších například Atmel, Freescale, nebo Intel. Mikropočítače od těchto firem používají několik jazyků pro programování, které pracují s různými knihovny, ze kterých jsou používány funkce pro psaní algoritmu programu. Pro nahrání programu do mikropočítače se používá několik komunikačních rozhraní, nejčastěji přes sériový port, nebo USB. Tyto rozhraní pracují buď s programátorem, přímo se zařízením, nebo s vývojovým kitem, která nahrají příslušný program do paměti mikropočítače. [4]

4.1 Programovací jazyky

4.1.1 Assembler

Assembler (Assembly language) je nástavba strojového kódu, proto se taky nazývá jako jazyk symbolických adres. Program se skládá z posloupnosti strojových instrukcí, které se nachází v dokumentaci každého mikropočítače. Assembler se v nynější době používá pouze na doplňky kódů pro jazyky na vyšší úrovni, nebo v kódech, které musí být efektivní, ultra kompaktní, nebo v situacích kdy to nemůže být napsáno jiným způsobem. [1]

4.1.2 C

Jazyk C byl následovníkem assembleru a v nynější době se jedná o nejefektivnější jazyk, který má nespočet výhod. Je velmi jednoduché se tento jazyk naučit, proto jej využívá skoro každý programátor a také díky tomu, že je jazyk na vyšší úrovni oproti assembleru, což znamená, že je nezávislý na architektuře procesoru a dá se s ním tak vyvíjet různé algoritmy, nebo aplikace. Podporují ho všechny kompilátory mikropočítače a je vhodný pro 8bitové i 32bitové procesory. [1][3]

4.1.3 C++

Jedná se o objektovou formu jazyka C, který je rovněž populární mezi programátory. Svými rysy je stejný jako jazyk C, ale podporuje funkce pro lepší datovou abstrakci a objektové programování. Nicméně tyto funkce snižují účinnost programu. Nejčastěji je využívají softwaroví vývojáři, u kterých převládají přínosy nad účinností. [1]

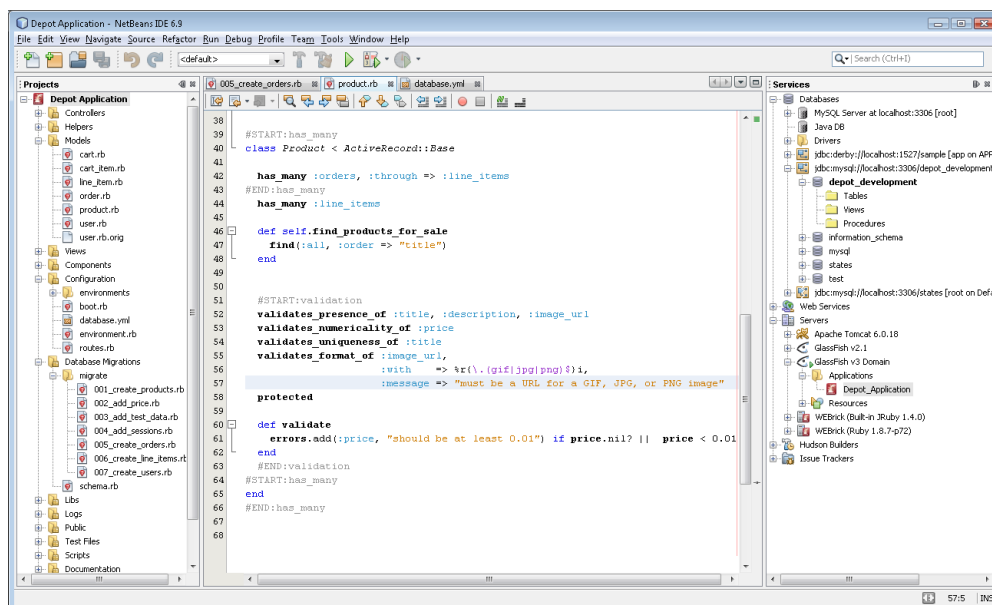
4.1.4 Ada

Objektově orientovaný jazyk jako C++, i když jsou od sebe odlišné. Byl vytvořen americkým ministerstvem obrany pro rozvoj vojenského softwaru. Je dvakrát přijat jako mezinárodní standart, Ada83 a Ada95, které byly využity spíše v obraném a leteckém průmyslu. Není moc populární i přes to, že jeho funkce by zjednodušili vývoj softwaru na úkor C nebo C++. [1]

4.2 Vývojové prostředí

Vývojové prostředí (IDE) je software, který programátorům usnadňuje jejich práci, pro psaní kódu a jeho následného nahrání do mikropočítače. Všeobecně podporuje ten jazyk, který se při vytváření projektu nastaví. Skládá se s editoru zdrojového kódu, do kterého se píše algoritmus, deklarace proměnných a knihoven. Další funkcí je debugger, který programátorovi pomáhá k postupnému projíždění kódu a tak lépe najde chyby v kódu. Poslední funkce je kompilátor, který převádí program v určitém jazyku na strojový kód. [11]

Většina firem mikropočítačů mají zdarma ke stažení své vývojové prostředí pro jejich výrobky, ale existují i open source prostředí, které jsou vhodné pro mikropočítače, které nemají své vlastní, nebo pokud je jejich prostředí zpoplatněno.



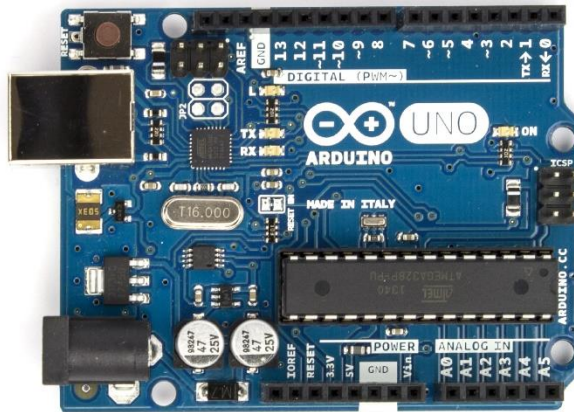
Obrázek 17 - Vývojové prostředí NetBeans [16]

5 PLATFORMA ARDUINO

Arduino je počítačová open-source platforma, která využívá jednoduché mikropočítačové desky a vývojového prostředí pro zápis softwaru na desku. Používá se na různé ovládání fyzických objektů, světel, motorů a jiných. [15]

5.1 Arduino Uno R3

Uno R3 je osazen 8bitovým mikropočítačem ATmega328. Deska podporuje 14 digitálních I/O pinů a 6 z nich může být použito na PWM modulaci. Osazeno je také 6 analogových vstupů, 16 MHz keramický rezonátor, resetovací tlačítko, USB pro komunikaci s deskou a vstup pro externí napájení. [15]



Obrázek 18 - Arduino UNO R3 [15]

5.2 Napájení

Arduino má dva vstupy pro napájení, buď prostřednictvím USB, nebo externím napájením. Deska si sama přepíná mezi zdroji napájení. Externí zdroj může být buď AC/DC adaptér nebo baterie v rozmezí napětí od 6 do 20V, doporučené napětí je od 7 do 12V. Jsou zde osazeny i další piny, které slouží k napájení externího zařízení, jedná se o pin 5V a 3V3, které dávají napětí 5V, nebo 3,3V. [15]

5.3 Paměť

ATmega328 má 32kB flash paměti pro program, dále má 2kB paměti SRAM a 1kB paměti EEPROM. [15]

5.4 Vstupy a výstupy

Osazeno je 14 digitálních vstup/výstup pinů, které pracují s napětím 5V a s maximálním proudem 40mA. Mají také 20-50 kOhm pull-up rezistory. Piny 3, 5, 6, 9, 10 a 11 se dají použít k 8bitové PWM modulaci. Dalšími vstupy jsou analogové, kterých je 6 a používají 10bitový převodník. [15]

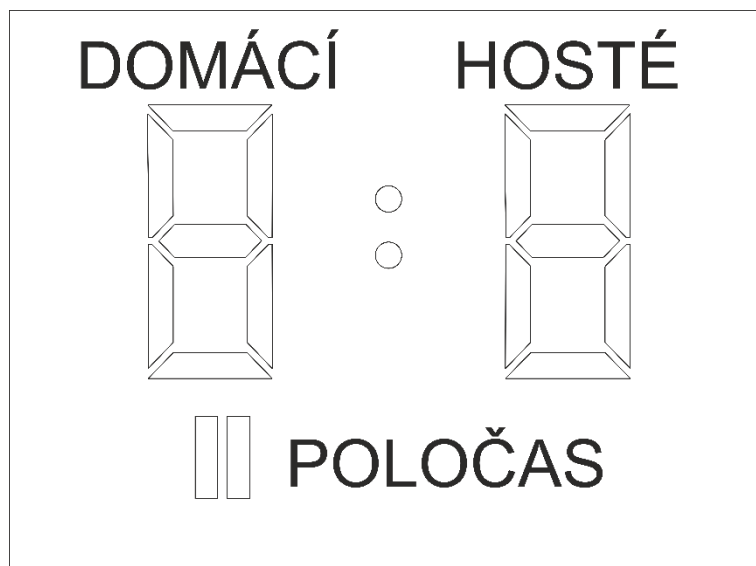
5.5 Komunikace

Ke komunikaci mezi PC a deskou slouží rozhraní USB. Je potřeba nainstalovat i ovladač, ten se nainstaluje buď samotným stažením, nebo při instalaci vývojového prostředí od Arduina. Prostřednictvím USB se dá používat i sériová komunikace. [15]

II. PRAKTICKÁ ČÁST

6 NÁVRH SVĚTELNÉ TABULE

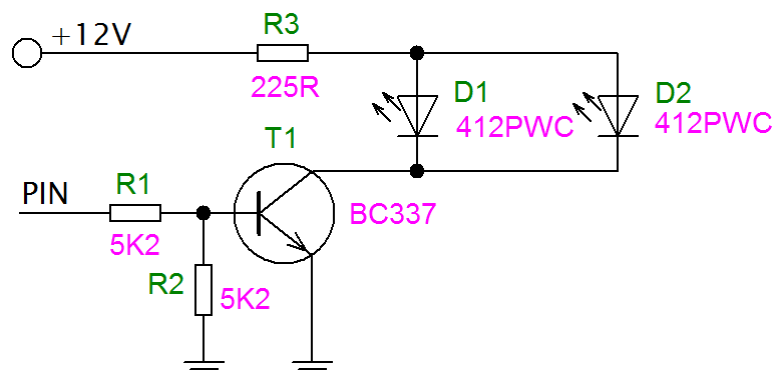
Tabule je vyrobena z centimetr tlustého plastu, ve kterém jsou osazeny dva sedmissegmentové displeje, dvojtečka a 2 čárky pro poločas. Celá bude řízena mikropočítačem Atmel a o její ovládání se bude starat dálkové ovládání s plovoucím kódem. Její rozměry jsou 80,4 x 60 cm.



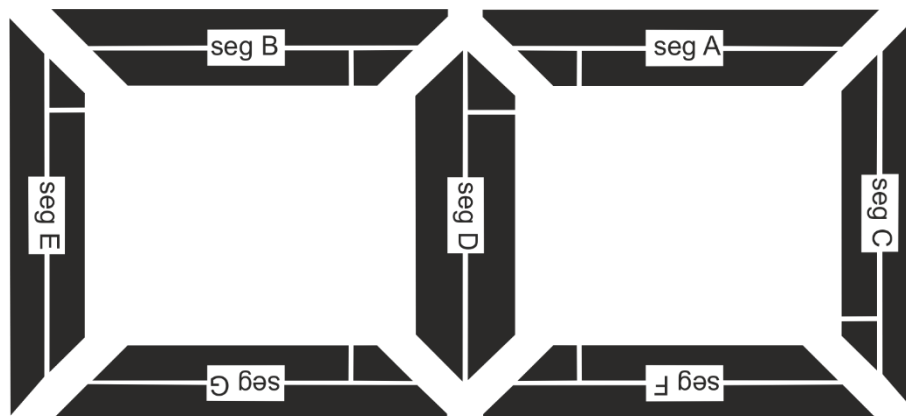
Obrázek 19 - Návrh tabule

6.1 Návrh zapojení sedmissegmentového displeje

Pro displej je použito 14 vysoce svítivých diod, s větším úhlem svítivosti. Zapojení má společnou zem a 7 vstupů pro každý segment. Každý segment je osazen 2 diodami a odporem pro snížení napětí na 3V. Pro rozsvícení diod slouží tranzistor, který se po sepnutí výstupního pinu otevře a uzemní tak obvod.



Obrázek 20 - Schéma segmentu



Obrázek 21 - Návrh plošného spoje displeje

6.2 Dálkové ovládání

Skládá se z 5 kanálového přijímače a dálkového ovládání rovněž s 5 kanály. Díky plovoucímu kódu je ovládání zabezpečeno a nelze jej ovládat jinými vysílači. [14]

6.2.1 Přijímač

Přijímač je ve formě stavebnice s označením K5PRX, která je řízena mikroprocesorem Microchip, který se stará o přijatá data a ovládání výstupů. Data se přijímají pomocí vysoko frekvenčního Rx modulu, díky čemuž je sestavení snadnější, neboť se nemusí nastavovat vysoko frekvenční část. Deska se napájí v rozsahu 10-14V, doporučené napětí je 12V. Na výstupu je osazeno 5 prepínacích relé na 250V/10A. [14]



Obrázek 22 - Přijímač Flajzar K5PRX [14]

6.2.2 Vysílač

Vysílač má označení KV5TX a skládá se ze 4 tlačítek ze přední strany a jednoho tlačítka z boční strany. Jeho dosah je až 150 m a je napájen 12V baterií typu 23A. [14]



Obrázek 23 - Dalkový ovladač KV5TX [17]

7 REALIZACE ZAPOJENÍ

Na tabuli jsem načrtl návrh dle kótování a pomocí přímočaré pily jsem vyřezal všechny otvory, pouze dvojtečku jsem vyřezal pomocí vrtáku. Po napájení plošných spojů pro displej jsem je upevnil do krabice tak, aby byly alespoň 25 mm od konce zadní strany tabule, je to z toho důvodu, aby diody měly širší záběr osvětlení.



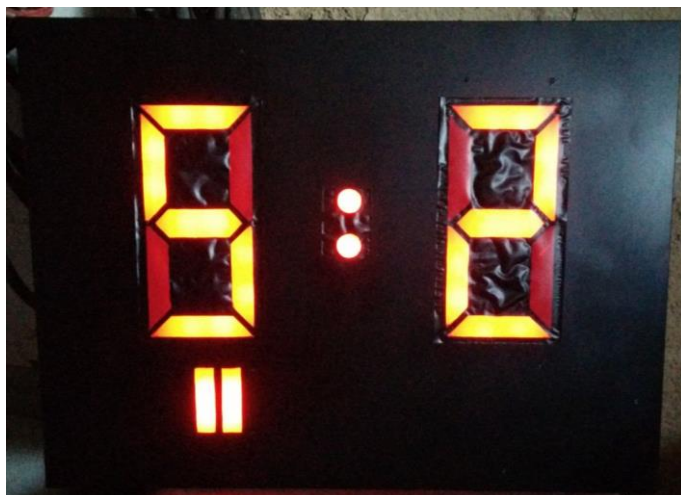
Obrázek 24 - Vyřezané otvory do desky

Po upevnění do krabice jsem od sebe oddělil všechny segmenty, aby nedocházelo k prosvi-
tování mezi nimi. Krabice jsem nastavil tak, aby se kryty mezi segmenty kryly s nevyřezanými místy displeje na tabuli. Krabice jsem upevnil pomocí čtyř závitových tyčí, které prochází skrz tabuli a ze přední strany jsou zašroubovány do zapuštěné matky. Ze zadní strany je do nich osazena krabice a pomocí matek upevněna.



Obrázek 25 - Uchycení plošného spoje displeje do krabice

Ze přední strany jsou displeje zalepeny pomocí červené folie, která je z vnitřní strany olepena oranžovou páskou, díky které neprosvitají diody, a lépe vede světlo po segmentu. Stejně jsou upevněny i ostatní zobrazovací prvky na tabuli.

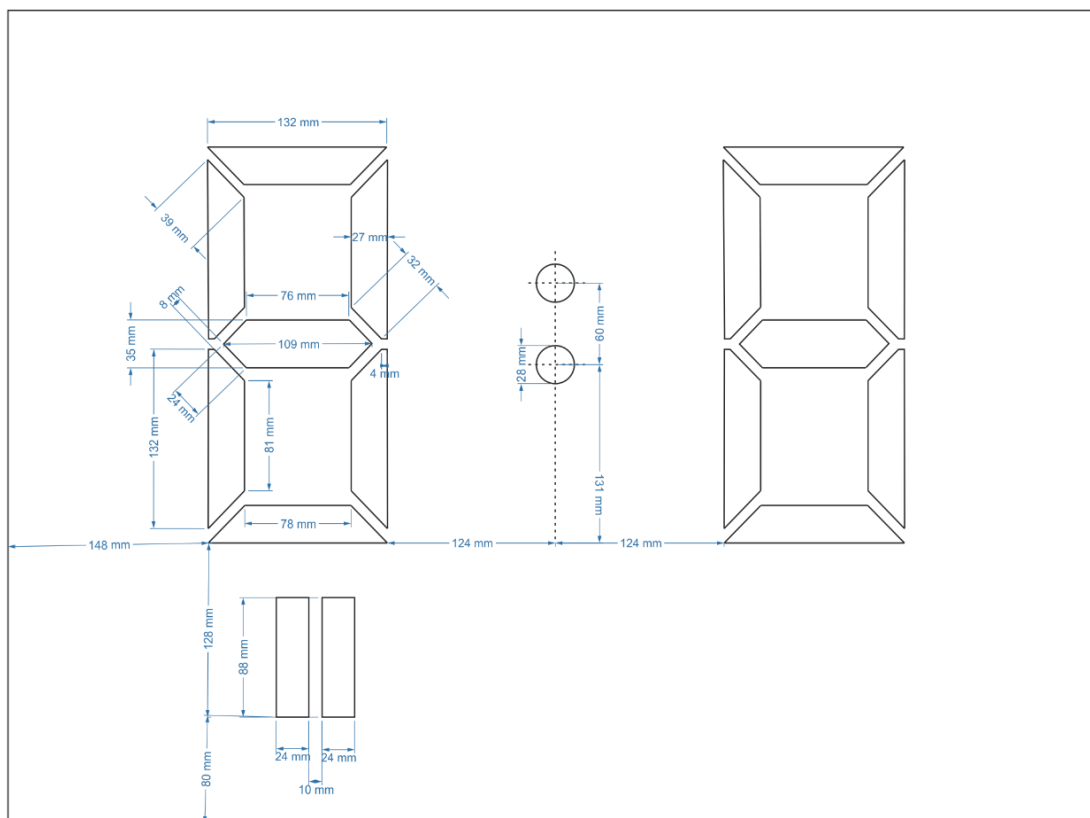


Obrázek 26 - Tabule ze přední strany

Všechny tyto výstupy se sbíhají do jedné krabice, ve které je upevněn přijímač pro dálkové ovládání, který generuje signály na analogové vstupy mikropočítače. Rovněž je zde umístěn mikropočítač a deska se spínacími tranzistory, které řídí digitální výstupy na oba displeje.



Obrázek 27 - Tabule ze zadní strany

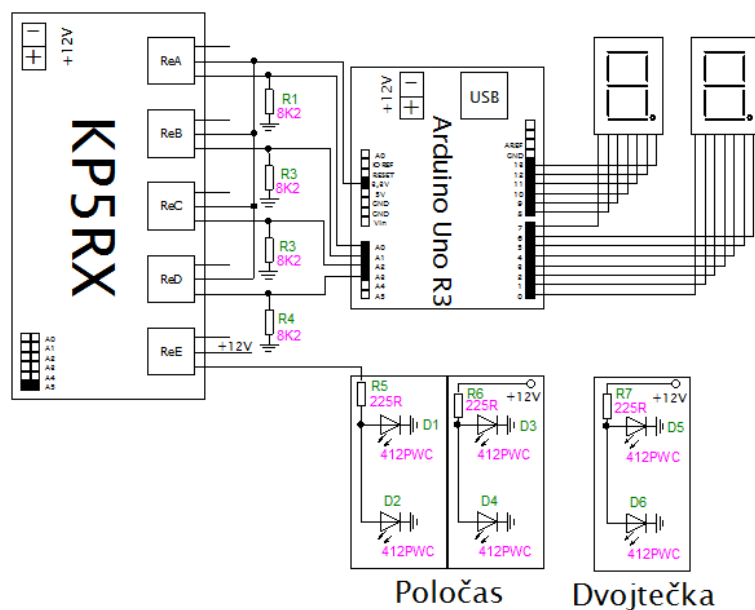


Obrázek 28 - Kótování tabule

7.1 Popis funkčnosti tabule

Při stlačení kteréhokoli tlačítka se vyšle signál do přijímače, který přepne příslušné relé pro dané tlačítko. Pomocí jumperu se dá nastavit, zda má relé při stlačení tlačítka zůstat sepnuté nebo sepnout jen na pár milisekund. Do prostředních vstupů relé jsem přivedl napětí 3,3V, které se při sepnutí relé přivádí na analogové vstupy mikropočítače. Po rozepnutí relé není na vstupu definováno napětí a díky rezistoru se zajistí, že je na vstupu 0V.

Každý displej je ovládán dvěma tlačítky, ze přední strany se horními tlačítky posouvá na displejích číslo nahoru a dolními směrem dolů, bočním tlačítkem se zapíná druhý poločas. O funkčnost displejů se stará mikropočítač, ovšem dvojtečka a signalizace prvního poločasu jsou připojeny na 12V, pouze signalizace druhého poločasu je připojen na 5 kanál přijímače, u kterého je osazen jumper, což způsobuje, že při stlačení tlačítka zůstane relé sepnuté a při dalším stlačení se rozepne.



Obrázek 29 - Schéma zapojení

7.2 Popis programu

Algoritmus je napsaný v jazyce C prostřednictvím vývojového prostředí Arduino. Obsahuje dvě systémové funkce `loop()` a `setup()`, dále jednu vlastní funkci `chNum()`. Na začátku jsou deklarované digitální výstupy, několik základních proměnných a dvou rozměrné pole, ve kterém jsou binární hodnoty pro výstupy. [16]

7.2.1 Proměnné

Konstantní proměnné `LEDA1` – `LEDG1` deklarují čísla pinů pro první displej a `LEDA2` – `LEDG2` deklarují piny pro druhý displej. Proměnné `num1` a `num2` se rovnají počátečnímu číslu, se kterým se bude od začátku pracovat, čili na displejích se zobrazí nula a poté se pomocí ovládání bude číslo zvyšovat, nebo snižovat. Proměnná `start` se použije pro nastavení výstupů při prvním spuštění programu. Dvourozměrné pole `arrayNum[11][7]` obsahuje v jednom řádku binární hodnoty pro výstup, které nastavují, jestli bude segment sepnut či nikoli. Pole je uspořádáno tak, že na prvním řádku jsou výstupy pro nulu, na druhém pro jedničku a takhle to pokračuje až do devítky.

```
const int LEDA1 = 0, LEDB1 = 1, LEDC1 = 2,
        LEDD1 = 3, LEDE1 = 4, LEDF1 = 5, LEDG1 = 6;
const int LEDA2 = 7, LEDB2 = 8, LEDC2 = 9,
        LEDD2 = 10, LEDE2 = 11, LEDF2 = 12, LEDG2 = 13;
int num1 = 0, num2 = 0;
int start = 0;
int arrayNum[11][7] = {{1,1,1,0,1,1,1},{0,0,0,0,0,1,1},{0,1,1,1,1,1,0},
                      {0,0,1,1,1,1,1},{1,0,0,1,0,1,1},{1,0,1,1,1,0,1},{1,1,1,1,1,0,1},
```

```
{0,0,1,0,0,1,1},{1,1,1,1,1,1,1},{1,0,1,1,1,1,1},{1,1,1,1,1,0,0}};
```

7.2.2 Funkce setup()

Tato funkce se provede pouze jednou a většinou slouží pro deklarování digitálních pinů do stavu vstup/výstup. Pro nastavení se používá funkce PinMode(), do které se zapisují dvě hodnoty, první je číslo pinu a druhá zda se jedná o vstup či výstup. U mého mikropočítače jsem nastavil deklarované piny jako výstupy, které budou podle potřeby buď sepnuty, nebo vypnuty. [16]

```
void setup(){
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
  pinMode(LED6, OUTPUT);
  pinMode(LED7, OUTPUT);
  pinMode(LED8, OUTPUT);

  pinMode(LED9, OUTPUT);
  pinMode(LED10, OUTPUT);
  pinMode(LED11, OUTPUT);
  pinMode(LED12, OUTPUT);
  pinMode(LED13, OUTPUT);
  pinMode(LED14, OUTPUT);
  pinMode(LED15, OUTPUT);
}
```

7.2.3 Funkce chNum()

Jedná se o mou vlastní funkci, která přebírá pouze dvě proměnné. První proměnnou je dis, která rozhoduje, na který displej se má vykreslit číslo, v případě jedničky jej vykreslí na první, v případě dvojky na druhý. Druhou proměnnou je num, která přebírá číslo pro vykreslení. Ve funkci jsou dvě podmínky, které díky proměnné dis testují, o který displej se jedná, po výběru displeje se pomocí funkce switch() testuje z proměnné num číslo, které se má na daný displej vykreslit, při nalezení schody se provede několik funkcí digitalWrite(). Do téhle funkce se zapisuje číslo digitálního pinu a poté binární hodnota, pokud bude rovna nule, pin bude vypnut, při rovnosti jedničky se pin sepne. Podle mého algoritmu se předávají proměnné, které odkazují na dané piny a binární hodnota pro nastavení pinu se předává z pole array-Num[11][7], ve kterém jsou pro dané číslo nastaveny výstupy. Pokud uživatel navýší číslo na deset, na displej se vypíše písmeno E, jako error. [16]

```
void chNum(int dis, int num){
  if(dis == 1){
    switch(num) {
```

```

        case 0:
            digitalWrite(LED1A, arrayNum[0][0]);
            digitalWrite(LED1B, arrayNum[0][1]);
            digitalWrite(LED1C, arrayNum[0][2]);
            digitalWrite(LED1D, arrayNum[0][3]);
            digitalWrite(LED1E, arrayNum[0][4]);
            digitalWrite(LED1F, arrayNum[0][5]);
            digitalWrite(LED1G, arrayNum[0][6]);
            break;
        case 1:
            digitalWrite(LED1A, arrayNum[1][0]);
            digitalWrite(LED1B, arrayNum[1][1]);
            digitalWrite(LED1C, arrayNum[1][2]);
            digitalWrite(LED1D, arrayNum[1][3]);
            digitalWrite(LED1E, arrayNum[1][4]);
            digitalWrite(LED1F, arrayNum[1][5]);
            digitalWrite(LED1G, arrayNum[1][6]);
            break;
        ...
    }
}
if(dis == 2){
    switch(num) {
        case 0:
            digitalWrite(LED2A, arrayNum[0][0]);
            digitalWrite(LED2B, arrayNum[0][1]);
            digitalWrite(LED2C, arrayNum[0][2]);
            digitalWrite(LED2D, arrayNum[0][3]);
            digitalWrite(LED2E, arrayNum[0][4]);
            digitalWrite(LED2F, arrayNum[0][5]);
            digitalWrite(LED2G, arrayNum[0][6]);
            break;
        case 1:
            digitalWrite(LED2A, arrayNum[1][0]);
            digitalWrite(LED2B, arrayNum[1][1]);
            digitalWrite(LED2C, arrayNum[1][2]);
            digitalWrite(LED2D, arrayNum[1][3]);
            digitalWrite(LED2E, arrayNum[1][4]);
            digitalWrite(LED2F, arrayNum[1][5]);
            digitalWrite(LED2G, arrayNum[1][6]);
            break;
        ...
    }
}
}

```

7.2.4 Funkce loop()

Po deklaraci pinů se nachází za funkcí setup() a chNum() funkce loop(). Tato funkce se opakuje stále dokola a vykonává tak hlavní program, který pracuje s algoritmem. V případě našeho programu se při prvním startu nатаví na displeje nuly, o to se postará podmínka if, která bude splněna v případě, že se proměnná start rovná nule. Na konci podmínky se start

posune na jedničku, aby se víckrát neprováděl. Dále se nacházejí čtyři podmínky, které díky funkci `analogRead()` čtou vstupní hodnoty na analogových vstupech a detekují tak sepnutí tlačítka ovládání, při kterém se přivede napětí na vstup a díky 10bitovému převodníku bude na vstupu číslo větší než nula. Do funkce `analogRead()` se zadává pouze číslo pinu, ze kterého se má číst. V každé z těchto podmínek se nachází ještě jedna podmínka, která porovnává, zda se může následující číslo vypsat, při zvyšování aby nepřekročilo číslo devět a při snižování číslo nula. Pokud jsou i tyto podmínky splněny tak se proměnná s aktuálním číslem posune o jedno nahoru nebo dolů a zavolá se funkce `chNum`, která číslo vypíše na displeji, poté se program funkcí `delay()` pozastaví na jednu sekundu, pro lepší snímání vstupů. Funkce `delay()` pracuje s milisekundami, takže se do ní místo jedničky musí zapsat tisícovka. [16]

```
void loop() {
  if(start == 0){
    chNum(1, num1);
    chNum(2, num2);
    start++;
  }
  if(analogRead(A0) > 50)
  {
    if(num1 < 10){
      num1++;
      chNum(1, num1);
      delay(1000);
    }
  }
  if(analogRead(A2) > 50)
  {
    if(num1 > 0){
      num1--;
      chNum(1, num1);
      delay(1000);
    }
  }
  if(analogRead(A1) > 50)
  {
    if(num2 < 10){
      num2++;
      chNum(2, num2);
      delay(1000);
    }
  }
  if(analogRead(A3) > 50)
  {
    if(num2 > 0){
      num2--;
      chNum(2, num2);
      delay(1000);
    }
  }
}
```

}
}

ZÁVĚR

Bakalářská práce se zabývá návrhem světelné tabule pro zobrazování skóre při fotbalových zápasech. Práce pojednává o technickém a elektronickém návrhu světelné tabule. Technický návrh zahrnuje veškerý postup od výroby tabule až po její elektronickou část.

Tabule se skládá ze dvou sedmi-segmentových displejů. Každý segment je tvořen dvěma vysoce svítivými diodami.

Celá tabule je řízena vývojovou deskou Arduino, která je osazena 8bitovým mikropočítačem Atmel328. Deska je osazena 14 digitálními vstupy/výstupy a také 6 analogovými vstupy. Některé z digitálních vstupů podporují různé funkce, například PWM modulaci.

Displeje se ovládají na dálku pomocí 5 kanálového přijímače a vysílače s plovoucím kódem. Vysílač má 5 tlačítek, z toho 4 ovládají změnu výsledku na displejích a jedno přepíná mezi prvním a druhým poločasem. Přijímač bere signály od ovladače a při stisku tlačítka sepne příslušné relé, které je pro dané tlačítko přiřazeno a tím vyšle signál na analogový vstup mikropočítače.

Z nalezených informací by světelná tabule pro fotbalové zápasy měla obsahovat dva displeje pro skóre zápasu, značení poločasů a časový údaj o délce hry. Ovšem kvůli náročnosti displejů vyrobená tabule časoměru neobsahuje.

Do tabule jsou vyřezány dva sedmi-segmentové otvory pro displeje, mezi nimi jsou vyvrtány dvě díry pro dvojtečku a dva obdélníky pro značení poločasu. Jak bylo již zmíněno, displeje se skládají ze sedmi segmentů, z nichž každý je napojen na pin výstupu mikropočítače. Každý segment má externí napájení 12V, které je díky rezistoru potlačeno na 3,3V a napájí tak obě diody. K sepnutí diod je zapotřebí změny logické hodnoty na pinu pro daný segment a dojde tak k otevření tranzistoru, který diody uzemní. Plošné spoje displejů jsou uchyceny 2,5 cm od zadní části tabule do 3 cm krabice, aby měly diody větší rozptyl svítivosti, a také jsou odděleny segmenty, aby nescházelo k prosvěcování mezi nimi. Celá krabice je uchycena pomocí závitové tyče, která prochází skrz celou tabuli, ze přední strany je uchycena do zapuštěné matky, ze zadní strany prochází tyč skrz celou krabici, která je upevněna matkou. Ostatní části tabule jsou rovněž zakrytovány stejně. Ze přední strany je nalepena červená folie, která má funkci jednak izolace a jednak znázornění sepnutých segmentů.

Veškerou činnost tabule řídí program, který je nahrán do vývojové desky Arduino. Na začátku programu se provádí funkce `setup()`, poté se neustále opakuje funkce `loop()`, která pracuje s funkcí `chNum()`.

Vytvořená tabule je funkční a bude používána při zápasech místního spolku TJ Sokol Bystřice pod Lopeníkem, který mě o vytvoření tabule požádal.

SEZNAM POUŽITÉ LITERATURY

- [1] BARR, Michael a Anthony J MASSA. Programming embedded systems. 2nd ed. Sebastopol: O'Reilly, 2006, xxi, 301 s. ISBN 978-0-596-00983-0.
- [2] CATSOULIS, John. Designing Embedded Hardware. Sebastopol: O'Reilly Media, 2005. ISBN 978-0-596-00755-3.
- [3] HEROUT, Pavel. Učebnice jazyka C. 6. vyd. České Budějovice: Kopp, 2010, 271, viii s. ISBN 978-80-7232-383-8.
- [4] MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6.
- [5] NOVÁK, Petr. Mobilní roboty - pohony, senzory, řízení, Praha: BEN - technická literatura, 2005. ISBN/EAN 80-7300-141-1 / 9788073001414.
- [6] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.
- [7] Segmentový displej. 2010. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-05-11]. Dostupné z: http://cs.wikipedia.org/wiki/Segmentov%C3%BD_displej
- [8] LED. 2005. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-05-11]. Dostupné z: <http://cs.wikipedia.org/wiki/LED>
- [9] OLED. 2006. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-05-11]. Dostupné z: <http://cs.wikipedia.org/wiki/OLED>
- [10] Displej z tekutých krystalů. 2006. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-05-11]. Dostupné z: http://cs.wikipedia.org/wiki/Displej_z_tekut%C3%BDch_krystal%C5%AF
- [11] Vývojové prostředí. 2006. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-05-12]. Dostupné z: http://cs.wikipedia.org/wiki/V%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD
- [12] Zobrazovací jednotky. *SPŠ Mohlenice* [online]. [cit. 2015-05-13]. Dostupné z: <http://www.spsemoh.cz/vyuka/zel/zobrazovaci-jednotky.htm>
- [13] 20x4 White on Blue Character LCD with Backlight. 2006. *Gravitech Electronic Experimental Solutions* [online]. [cit. 2015-05-13]. Dostupné z: <http://www.gravitech.us/20chblcd.html>
- [14] *FLAJZAR: Vývoj, výroba a prodej elektroniky* [online]. [cit. 2015-05-18]. Dostupné z: <http://www.flajzar.cz/>
- [15] *Arduino* [online]. 2015. [cit. 2015-05-14]. Dostupné z: <http://www.arduino.cc/>

[16] Ruby and Ruby on Rails. *NetBeans* [online]. [cit. 2015-05-14]. Dostupné z: <https://netbeans.org/features/ruby/index.html>

[17] *GSMALARM* [online]. [cit. 2015-05-18]. Dostupné z: <http://www.gsmalarm.sk>

SEZNAM OBRÁZKŮ

| | |
|---|----|
| Obrázek 1 - Znázornění světelné tabule pro Basketbal | 10 |
| Obrázek 2 - Znázornění světelné tabule pro Fotbal a Ragby | 11 |
| Obrázek 3 - Znázornění světelné tabule pro Florbal a Házenou..... | 11 |
| Obrázek 4 - Znázornění světelné tabule pro Lední hokej | 11 |
| Obrázek 5 - Znázornění světelné tabule pro Volejbal | 12 |
| Obrázek 6 - Znázornění světelné tabule pro Tenis | 12 |
| Obrázek 7 - Znázornění světelné tabule pro Judo..... | 13 |
| Obrázek 8 - Sedmissegmentový displej | 14 |
| Obrázek 9 - Devítisegmentový displej..... | 14 |
| Obrázek 10 - Čtrnáctisegmentový displej..... | 15 |
| Obrázek 11 - Patnáctisegmentový displej..... | 15 |
| Obrázek 12 - Šesnáctisegmentový displej | 16 |
| Obrázek 13 - Sedmissegmentový displej s LED podsvícením [12]..... | 16 |
| Obrázek 14 - Displej s OLED technologií [9] | 17 |
| Obrázek 15 - Displej z tekutých krystalů [13]..... | 17 |
| Obrázek 16 - Schéma mikropočítače [6] | 18 |
| Obrázek 17 - Vývojové prostředí NetBeans [16] | 21 |
| Obrázek 18 - Arduino UNO R3 [15] | 22 |
| Obrázek 19 - Návrh tabule..... | 25 |
| Obrázek 20 - Schéma segmentu..... | 25 |
| Obrázek 21 - Návrh plošného spoje displeje | 26 |
| Obrázek 22 - Přijímač Flajzar K5PRX [14]..... | 26 |
| Obrázek 23 - Dalkový ovladač KV5TX [17]..... | 27 |
| Obrázek 24 - Vyřezané otvory do desky | 28 |
| Obrázek 25 - Uchycení plošného spoje displeje do krabice | 28 |
| Obrázek 26 - Tabule ze přední strany | 29 |
| Obrázek 27 - Tabule ze zadní strany | 29 |
| Obrázek 28 - Kótování tabule | 30 |
| Obrázek 29 - Schéma zapojení | 31 |

SEZNAM PŘÍLOH

PI CD