



Středoškolská technika 2022

Setkání a prezentace prací středoškolských studentů na ČVUT

**Naprogramování a sestavení maticového LED panelu na
zobrazování textu**

Ondřej Weigert

VOŠ,SŠ, Centrum odborné přípravy, Sezimovo Ústí
Budějovická 421, Sezimovo Ústí

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Sezimově Ústí dne 05. 04. 2022

Ondřej Weigert

Poděkování

Maturitní práce byla zpracována jako závěrečný projekt v rámci řádného ukončení 4. ročníku maturitního studia Elektrotechnika – počítače a robotika. Vedoucím práce byl pan učitel Ing. Miloň Jedlička, kterému tímto děkuji za odborné konzultace a cenné rady týkající se struktury i obsahu práce.

Zároveň děkuji vedení Vyšší odborné školy, Střední školy, Centra odborné přípravy a pedagogům této školy za umožnění využít ke zpracování projektu moderní zařízení odborných laboratoří. Děkuji zejména panu Miloši Blechovi za přínosné maturitní semináře. Cenné rady mi poskytl také odborný konzultant pan Jiřímu Bumbovi

Anotace

Projekt byl vypracován jako nezbytná součást povinné SOČ zkoušky...

Tato práce se zabývá zapojením Arduina a led matice a jeho naprogramováním.

Po nalezení informací k naprogramování Arduina proběhlo následné sestavení programu pro zobrazování textu na led matici. Cílem práce je zapojit Arduino a Matici to naprogramovat, a následně umístit do pouzdra vytvořeného na 3D tiskárně.

Anotation

The project was made as a necessary part of the compulsory SOČ examination...

This work deals with the connection of Arduino and led matrix and its programming. After finding out the information for programming the Arduino, a program for displaying the text to led matrix was subsequently compiled. The aim of the work is to involve Arduino and matrix to program it and then place it to the 3D printed housing.

Klíčová slova a vybrané pojmy

Arduino

Led matice

3D tisk

Programovací jazyk C

PN – Polovodič

LCD – Liquid Crystal Display

Keywords

Arduino

Led matrix

3D print

C Programming Language

PN – Semiconductor

LCD – Liquid Crystal Display

Obsah

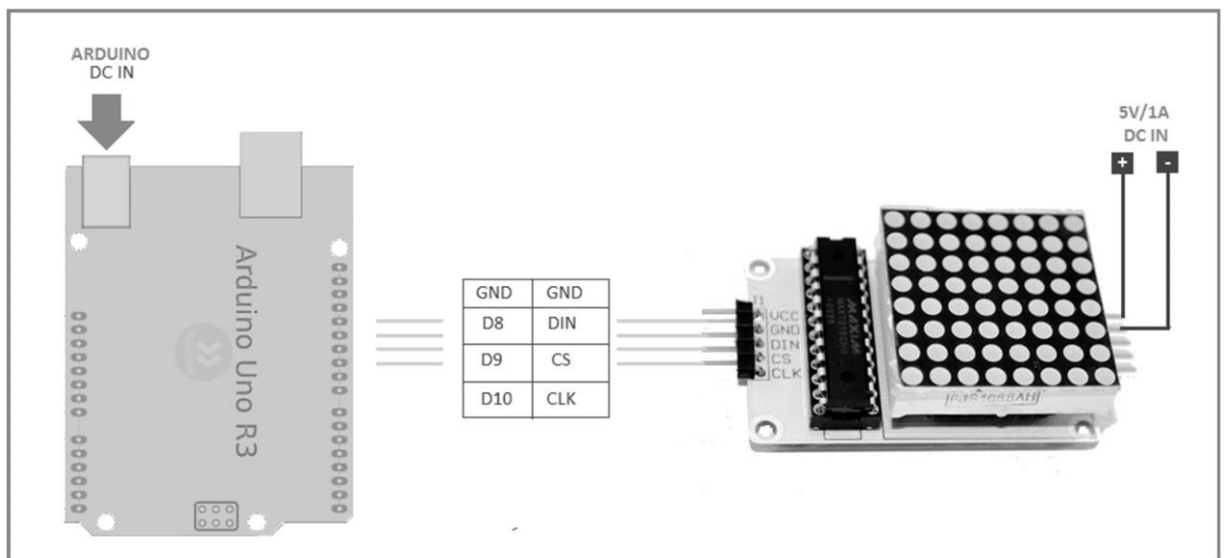
1. Úvod	3
2. Teoretická část	3
Arduino a jeho charakteristika	4
Typy desek Arduino	5
Arduino Uno	5
Arduino Esplora	6
Arduino Mega	6
Arduino Mini	7
Arduino Nano	8
Led diodový maticový display	9
Historie Arduina	11
Hardware Arduina	12
Led Dioda	13
PROGRAMOVACÍ JAZYK C	15
SPI PROPOJENÍ	16
Stručná historie řešeného problému	17
3. Praktická část	18
Postup řešení	22
4. Závěr – zhodnocení dosažených výsledků	22
5. Seznam použitých zdrojů a literatury	23
6. Seznam obrázků	24
7. Přílohy	25

1. Úvod

Arduino se v poslední době stalo docela dost velkým trendem a dnes na internetu najdete spousty a spousty programů které už někdo vytvořil a stačí vám pouze nakoupit si součástky a sestavit je a následně už jen nahrát program.

Právě proto jsem se rozhodl že budu mít v SOČ práci Arduino. Poté jsem se přemýšlel nad tím, co si udělám napadlo mě, že bych si udělal digitální stolní hodiny, ale pak jsem dostal nápad udělat si display na text. Jakožto hráč počítačový her, co hraje s kamarády jsem si říkal, že by se to občas dalo použít k něčemu vtipnému ale dokonce i k něčemu užitečnému. Jelikož jsem člověk, co docela dost zapomíná tak jsem si říkal, že by se to dalo použít třeba i jako display na poznámky a vím že ho nepřehlédnu, jelikož mi to bude přímo svítit do očí.

Posledním důvod, proč jsem si vybral zrovna tuto SOČ práci byl důvod, že mám aspoň trošku nějaký přehled od programovacím jazyce C což je právě programovací jazyk pro Arduino.



Obr: 1 Zapojení Arduina a matice

2. Teoretická část

Tato SOČ práce bude fungovat tak že větu kterou bude chtít uživatel zadat zadá do programu přesně tak jak by ji napsal a program si to přebere a zobrazí tuto větu na matici. Základem bude Program v jazyku C. Program bude začínat listem který přiřadí každému písmenu v abecedě kód na zobrazení na matici. Poté bude následovat cyklus který poté co uživatel zadá větu tak si jí přebere a bude jí zobrazovat, dokud se zařízení neodpojí od napětí anebo nezadá další větu. Velikost matice bude 8x64 aby tam byl ten text jasně vidět.

Cílem mé SOČ práce bude zobrazit text zadaný uživatelem a zobrazit ho na matici.

Arduino a jeho charakteristika

Arduino je mikrokontroler s otevřeným zdrojem, který lze použít pro mnoho různých aplikací. Je to pravděpodobně nejjednodušší a nejlevnější možnost mikrokontroleru pro fanoušky, studenty a profesionály, kteří vyvíjejí projekty založené na mikrokontrolerech. Arduino používá buď mikrokontroler Atmel AVR nebo Atmel ARM a některé verze mají rozhraní USB. Mají také šest nebo více analogových vstupních kolíků a čtrnáct nebo více kolíků pro vstup / výstup (I / O), které se používají pro připojení snímačů, akčních členů a dalších periferních obvodů k mikrokontroleru. Arduino stojí kdekoli od pěti do šesti dolarů až do výše 40 dolarů, v závislosti na sadě funkcí.

Arduino je otevřená platforma s grafickým vývojovým prostředím, které vychází z prostředí Wiring (podobný projekt jako Arduino, tedy deska s mikrokontrolerem a IDE)^[1] a Processing (prostředí pro výuku programování).^[2] Arduino bylo poprvé představeno v roce 2005. Může být použito k vytváření samostatných interaktivních zapojení nebo může být připojeno k software na počítači (např. Adobe Flash, Processing, Max/MSP, Pure Data, SuperCollider). Momentálně lze koupit verze, které jsou už zkompletované; schéma a návrh plošného spoje je dostupný pro ty, kteří si chtějí postavit Arduino sami.

Na rozdíl od Raspberry Pi není Arduino zamýšleno jako plnohodnotný stolní počítač. Řídící program je vyvíjen zvlášť (na stolním počítači) a do Arduina je posléze nahrán a spuštěn. Uvnitř Arduina je pak spuštěn jen tento program, který typicky obsahuje smyčku, která se neustále opakuje (Arduino neustále zjišťuje stav svého okolí a na změny reaguje). Díky tomu má nízkou spotřebu (je možné napájení malou baterií) a hodí se například pro řízení dronů, robotů a podobně. [7]

Typy desek Arduino

Existuje mnoho různých typů Arduin, jak je uvedeno v následující tabulce, z nichž každá má vlastní sadu funkcí. Jsou lišící se rychlostí zpracování, pamětí, I / O porty a konektivitou, ale jejich základní funkce je stejná.

Arduino Uno

Arduino Uno je v současné době asi nejčastěji používaný typ desky. Je přímým pokračovatelem hlavní vývojové linie, která započala prvním Arduinem se sériovým portem místo USB, pokračující přes Arduino Extreme, NG, Diecimila a Duemilanove až k dnešnímu Uno. Na desce najdeme procesor ATmega328 a již klasické USB. Z této hlavní linie se vyvinuly i další dvě speciální desky. První z nich je Arduino Ethernet. Které má stejnou výbavu jako Uno. Místo USB portu zde ale najdeme Ethernet port pro připojení k síti. Příjemná je přítomnost slotu pro microSD karty. Druhou deskou je Arduino Bluetooth. Jak už název napovídá, místo USB zde najdeme Bluetooth modul pro bezdrátovou komunikaci. Velmi odlehčenou verzí Arduina Uno je Arduino Pro. To postrádá USB port a je to tedy nutné ho programovat externím převodníkem. Je určeno spíše k pevnému zabudování do nějakého projektu.

Arduino Uno jsem si vybral z důvodu toho, že jsem si nebyl jistý, zda nebudu ještě chtít připojit další externí zařízení. Dalším důvodem byla možnost pouze připojení, kde



Obr: 2 Arduino Uno

nemusím nic pájet, proto to je jednodušší pro používání než jako pájení u Arduino Mini. Posledním důvodem bylo možnost připojit Arduino přes USB port.

Arduino Esplora

Arduino Esplora je v informatice název malého jednodeskového počítače založeného na mikrokontrolerech ATmega od firmy Atmel. Tato deska je zvláštní už od pohledu, vypadá totiž jako ovladač, a také pro tyto účely byl vytvořen. Vlastní teploměr, bzučák, akcelerometr, a i piny pro LCD displej. Jedná se totiž o druh Arduina, ze kterého můžeme vytvořit herní set nebo vlastní konzoli. [1]



Obr: 3 Arduino Esplora

Arduino Mega

Arduino Mega2560 je v informatice název malého jednodeskového počítače založeného na mikrokontrolerech ATmega od firmy Atmel. Tato deska je zvětšením desky Arduina

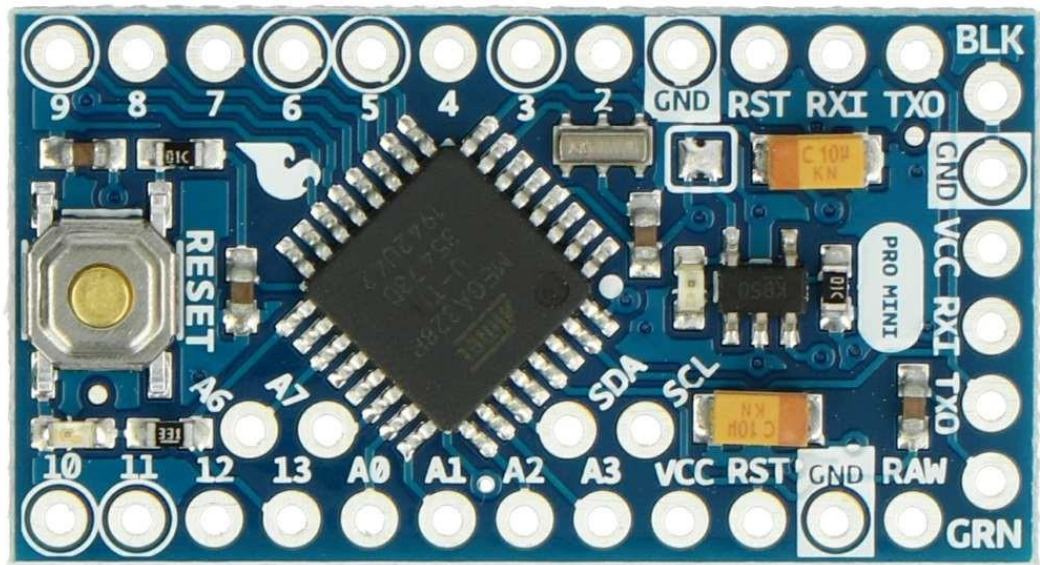


Obr: 4 Arduino Mega

Uno, používá výkonnější čip, více pinů. Další rozšíření má ještě deska Arduino Mega ADK, které má USB navíc pro připojení zařízení s Androidem. [2]

Arduino Mini

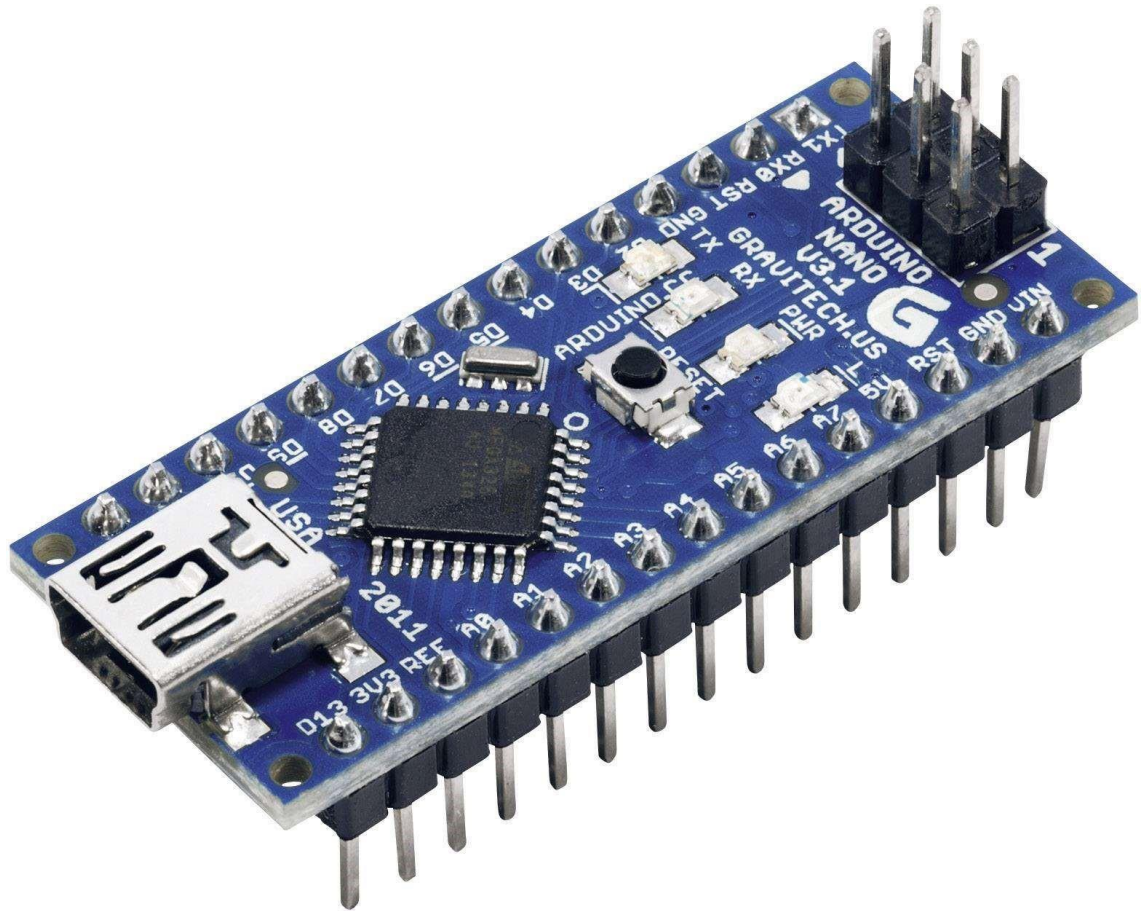
Arduino Mini je v informatice název malého jednodeskového počítače z otevřené platformy Arduino, který je založen na mikrokontrolerech ATmega od firmy Atmel. Je asi nejmenší oficiální verzí Arduina a je navržen tak, aby ušetřil co nejvíce místa. Kvůli tomu nemá USB port. K programování je proto potřeba použít externí USB převodník. Funguje na procesoru ATmega328 s taktem 16 MHz. Díky tomu, jak je malý, ho můžeme použít například v dálkových ovladačích, pro ovládání klávesnice ke vstupním dveřím nebo například k chytrému osvětlení podle denního světla. [3]



Obr: 5 Arduino Mini

Arduino Nano

Arduino Nano je v informatice název malého jednodeskového počítače z otevřené platformy Arduino, který je založen na mikrokontrolerech ATmega od firmy Atmel. Arduino Nano je podobné Arduino Mini, s tím rozdílem, že má vlastní USB port a převodník. Je tedy trochu větší, ovšem není třeba mít extra převodník při vkládání programu. [4]



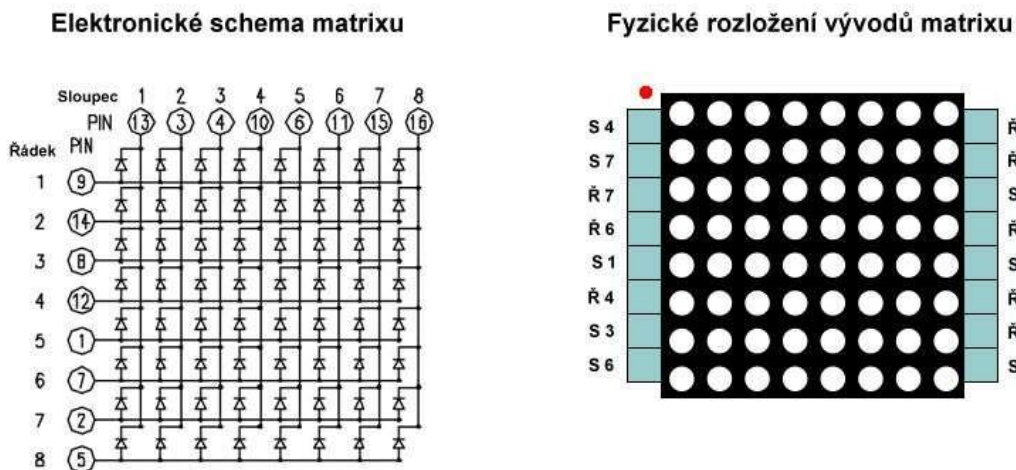
Obr: 6 Arduino Nano

Led diodový maticový display

Digitální obvody nacházejí v současné době širokou škálu aplikací. Hlavní výhodou digitálních elektronických obvodů je lepší, resp. přesnější kontrola nad jejich logickými stavy, které jsou vybírány konečné posloupnosti hodnot. Využití je možné pro výpočty matematických operací nebo provádění jiných přesně požadovaných funkcí. Splňují tedy požadavky pro řízení a kontrolu všemožných aplikací v oblastech elektrotechniky a výrobních procesů. Práce v první části shrnuje základní teorii integrovaných řadičů (budičů) pro displeje. Obvykle nacházejí aplikace v řízení sedmi-segmentových displejů nebo maticových LED displejů. Budiče jako samostatný blok nevykonávají žádné procesy bez vstupního podnětu. Z tohoto důvodu je v teoretické části popsáno jako budiče použít a také naznačeno, jak tyto obvody řídit pomocí hardwarových modulů s mikroprocesory. Výhoda použití budičů také spočívá v principu řízení pomocí sériového toku dat. Tento fakt vede také k redukci počtu vodičů mezi řídicím obvodem a budičem a mimo jiné také k požadavku použití oscilátoru pro hodinový signál s vyšší frekvencí. Druhá část práce popisuje výsledky praktické realizace displeje pomocí budičích obvodů a programovatelného mikroprocesoru, který je řízen na úrovni lokální sítě v reálném čase.

Taková realizace vyžaduje zajištění kompatibility mezi tokem dat ze sítě a mikroprocesorem. Na základě libovolného internetového nebo smyšleného protokolu a vhodného převodu dat sítě a sběrnici kompatibilní pro daný mikroprocesor

Pole diod - LED matrix



Obr: 7 Vnitřní zapojení led matice

je možné kompatibilitu zajistit. Předností použití digitálních zobrazovacích zařízení (displejů) ve srovnání s obrázkem či textem tištěným na daném podkladu je vzdálená kontrola nad zařízením (komunikace pomocí logických stavů) nebo např. velká univerzálnost (rychlá možnost přepisu). Za pomoci řídicích obvodů je lze využívat pro potřebné účely, umožňuje-li to zařízení. S ohledem na cenu, tedy i spotřebu je nutno

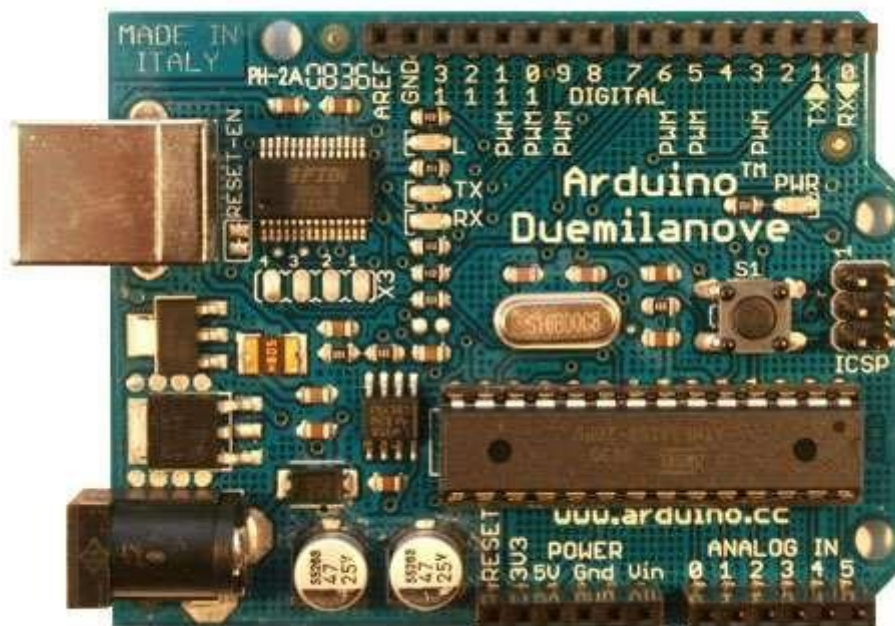
pohlížet na fakt, že LED diody se vyznačují vysokou účinností přeměny elektrické energie na světlo. [5]



Obr: 8 Led matice

Historie Arduina

Projekt vznikl v roce 2005 v Itálii ve městě Ivrea. Jeho cílem bylo vytvořit jednoduchou prototypovací platformu pro studenty, která umožní rychlý vývoj a jednoduché používání.



Obr: 9 Arduino Duemilanove

Projekt zaznamenal velký úspěch a později začaly vznikat jeho další, novější verze. Do února 2010 se údajně prodalo více než 120 tisíc kusů.

Zakladatelé Massimo Banzi a David Cuartielles pojmenovali projekt po Arduinovi Ivrejském, významné historické postavě města.

Projekt Arduino získal ocenění v kategorii digitálních komunit na Prix Ars Electronica 2006.

V roce 2010 vyšla, zatím poslední, verze Arduino desky s označením Uno, která nahradila svého předchůdce Duemilanove. Od roku 2012 jsou v prodeji desky Due (s procesorem ARM) a Leonardo a také nová revize Una R3. [7]

Hardware Arduina

Desky Arduino obsahují 8bitové mikrokontrolery z rodiny AVR od firmy Atmel a množství dalších podpůrných obvodů. Oficiální vydání Arduina, které vyrábí a prodává Italská firma Smart Projects, používají čipy ATmega8, ATmega168, ATmega328, ATmega1280 a ATmega2560. Arduino Due obsahuje 32bitový ARM procesor Atmel SAM3X8E. Každá deska má většinu I/O pinů přístupných přes standardizované patice, do kterých se jednoduše připojují další obvody, kterým ve světě Arduina říká Shieldy. Na deskách bývá několik diod, resetovací tlačítko, konektory pro ICSP programování,



Obr: 10 ATmega1280

napájecí konektor, oscilátor a obvod zprostředkovávající komunikaci po USB. Základní verze Arduina, Arduino Uno, poskytuje celkem 14 I/O digitálních pinů a 6 pinů analogových. Šest digitálních pinů je také možné použít na softwarově řízený PWM výstup.

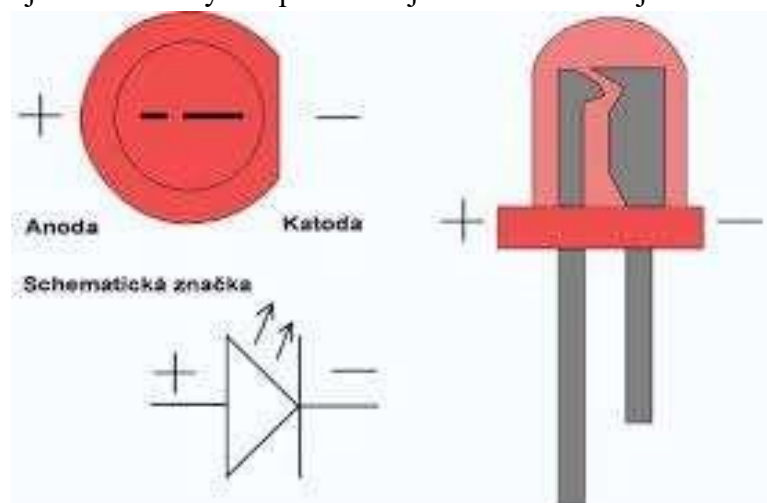
Hlavní mikrokontroler, který je uživatelsky programovatelný, již má bootloader (kód, který se po spuštění postará o základní nastavení mikrokontroleru, jako jsou interní časovače, nastavení rozhraní USART a další) a nastavené potřebné fuses bajty (těmi se nízkou úrovní nastavují některé vlastnosti čipu). Díky tomu se uživatel nemusí starat o detaily a své programy píše v jazyce podobném C/C++.

Ačkoliv je Arduino připojeno k počítači pomocí rozhraní USB, je softwarově simulována sériová komunikace přes linku RS-232. Ve starších deskách, jako je například Duemilanove nebo Diecimila, se pro tyto účely používaly FTDI čipy, v desce Uno toto obstarává ATmega8U2 s přeprogramovaným firmware (ten je, jako vše ostatní, volně dostupný v podobě zdrojových kódů). [7]

Led Dioda

Závislost proudu na napětí (voltampérová charakteristika) má podobný průběh jako běžná polovodičová dioda. Je jiný pouze hodnotami hraničních napětí v propustném a závěrném směru.

Zvyšujeme-li od nuly napětí na diodě v propustném směru (tj. kladný pól zdroje je připojen na anodu a záporný pól zdroje na katodu diody), zpočátku LED neprochází téměř žádný proud - chová se, jako by byla nevodivá. Od určité výše přiloženého napětí dojde ke zlomu – dioda se začne otevírat a na další drobné zvyšování napětí reaguje prudkým nárůstem procházejícího proudu. Proud vzhledem k napětí roste přibližně exponenciálně. Tehdy se dioda začíná také rozsvěcet. Zvyšujeme-li dále napětí, dioda zvyšuje svůj jas, až dosáhne svého maxima. Závislost svítivosti LED na proudu je v této oblasti přibližně lineární, tj. např. při dvojnásobném zvýšení proudu dojde ke zhruba dvojnásobnému



Obr: 11 Led dioda červená

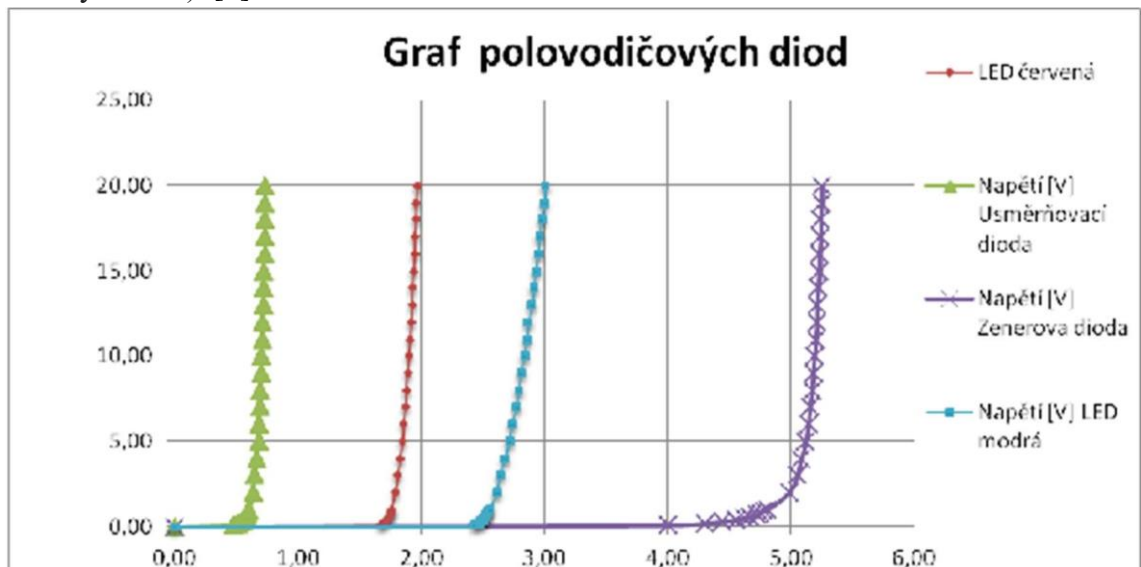
zvýšení intenzity světla vydávaného diodou. Při dalším zvyšování proudu nad povolenou mez může dojít k tepelnému přetížení polovodičového přechodu a poruše diody.

Vzhledem ke tvaru voltampérové charakteristiky se tyto diody musejí napájet ze zdroje proudu. Pokud nevyžadujeme vysokou účinnost zapojení (například u různých indikátorů), můžeme se přiblížit proudovému zdroji tím, že připojíme LED v sérii s rezistorem omezujícím protékající proud ke zdroji stálého napětí (změny napětí vyvolají menší změny proudu). Tento způsob je běžně používán.

Většina LED má také nízké průrazné napětí, takže mohou být zničeny přiložením závěrného napětí i o výši jen několika voltů. Protože se nelze vždy spolehnout na obecná pravidla určení polaritý vývodů, je vhodné se o jejich polaritě přesvědčit nahlédnutím do katalogového listu. Případně lze polaritu zjistit připojením diody ke zdroji nízkého napětí v sérii s ochranným rezistorem. [5]

Princip:

Prochází-li PN přechodem světelné diody elektrický proud v propustném směru, přecházejí volné elektrony z části typu N (s přebytkem volných elektronů) do části typu P (s nedostatkem volných elektronů, s tzv. dírami). Když excitovaný volný elektron z části N zapadne do orbity atomu v polovodiči typu P, dostane se na nižší energetickou hladinu. Přebytková energie je v podobě vlnění vyzářena mimo materiál (různými druhy záření, např. viditelné světlo, infračervené nebo ultrafialové záření). Při vhodné volbě příměsí v polovodičovém materiálu je v LED na odhaleném PN přechodu přebytková energie vyzářena ve viditelném spektru jako nekoherentní světlo s úzkým spektrem (úzký vlnový rozsah). [5]



Obr: 12 Diody graf

PROGRAMOVACÍ JAZYK C

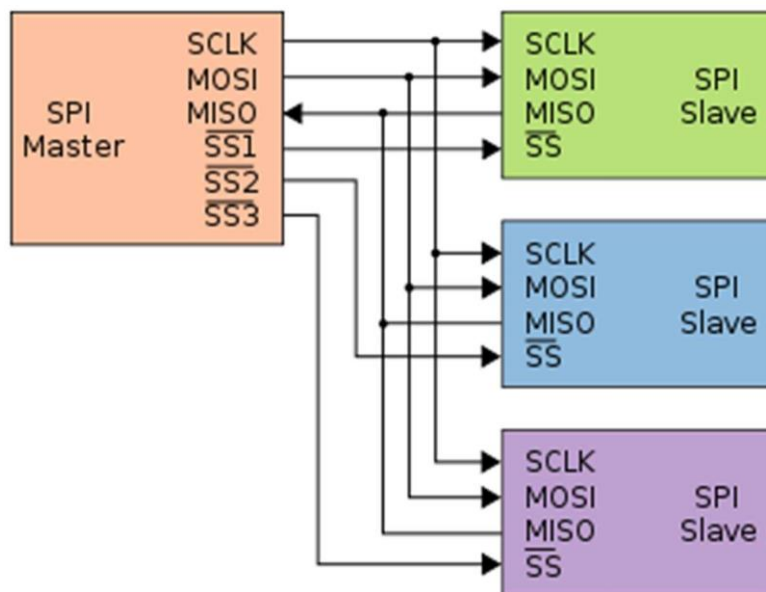
V současné době jeden z nepoužívanějších standardů (jazyků) pro psaní systémového softwaru a aplikací (kompilátory, operační systémy, knihovny). Jeho použití je tedy možné ve většině systémového programování, je velmi snadno čitelný (ve srovnání s jinými programy) a navíc i velmi univerzální, tedy přenositelný do jiných architektur beze změn nebo s minimálními úpravami. V roce 1983 se American National Standards Institute (ANSI) dohodla na sestavení komise X3J11, aby byla vytvořena standardní specifikace C. Po dlouhém a pracném procesu byl standard dokončen v roce 1989 a schválen jako ANSI X3.159-1989 „Programming Language C“. Tato verze jazyka je často stále označována jako ANSI C. V roce 1990 byl standard ANSI C (s drobnými změnami) adoptován institucí International Organization for Standardization (ISO) jako „ISO 9899|ISO/IEC 9899:1990“. Jednalo se v podstatě o vytvoření jakéhosi standartu, resp. vytvořit nadmnožinu (soubor) zahrnující mnoho vlastností jazyka doposud používané pouze lokálně. Samotný programovací standart ANSI C je v současnosti kompatibilní s většinou aplikací. Téměř každý překladač je s tímto standardem schopný pracovat. Z důvodu častého používání různých knihoven, a tedy i příkazů ve zdrojovém kódu programu se někdy jazyk stává přeložitelným pouze pro určené platformy z obecného pohledu. Z pohledu konkrétní aplikace se jedná opět o standart (například knihovny pro řízení LCD displeje se ve většině variant není o moc jiný). V současnosti dobře existuje také několik rozšíření jazyka C např. jazyk C++, které úzce spojen s jazykem C. C++ je od jazyka C odvozeno. Pro dané aplikace se C++ chová striktněji a se standardním C není úplně kompatibilní. [13] Použitím jazyka C lze provádět matematické operace, řízení procesorů nebo programování softwarových aplikací. Základem pro použití jazyka je vybrat vhodné vývojové softwarové prostředí, které podporuje překlad tohoto jazyka. Základem je nadefinování použitých knihoven pro jazyk C a příslušné operandy, se kterými se v daném programu pracuje. Následuje zápis těla programu, kde se definuje vykonávání daných funkcí pro danou aplikaci. Při správnosti se poté zdrojový kód překládá do spustitelné podoby pro procesor (kontrolér). Jedná se o převod zdrojového kódu do bitové podoby. V této podobě se program již velmi těžce upravuje. Proces kompilace zajišťuje nejčastěji vývojový software. Vývojový software obvykle umožňuje také použití funkce debug, která ve fázi, kdy je program laděn kontroluje chyby ve zdrojovém kódu, které potenciálně zamezují chodu programu. [8]



Obr: 13 Programovací jazyk C

SPI PROPOJENÍ

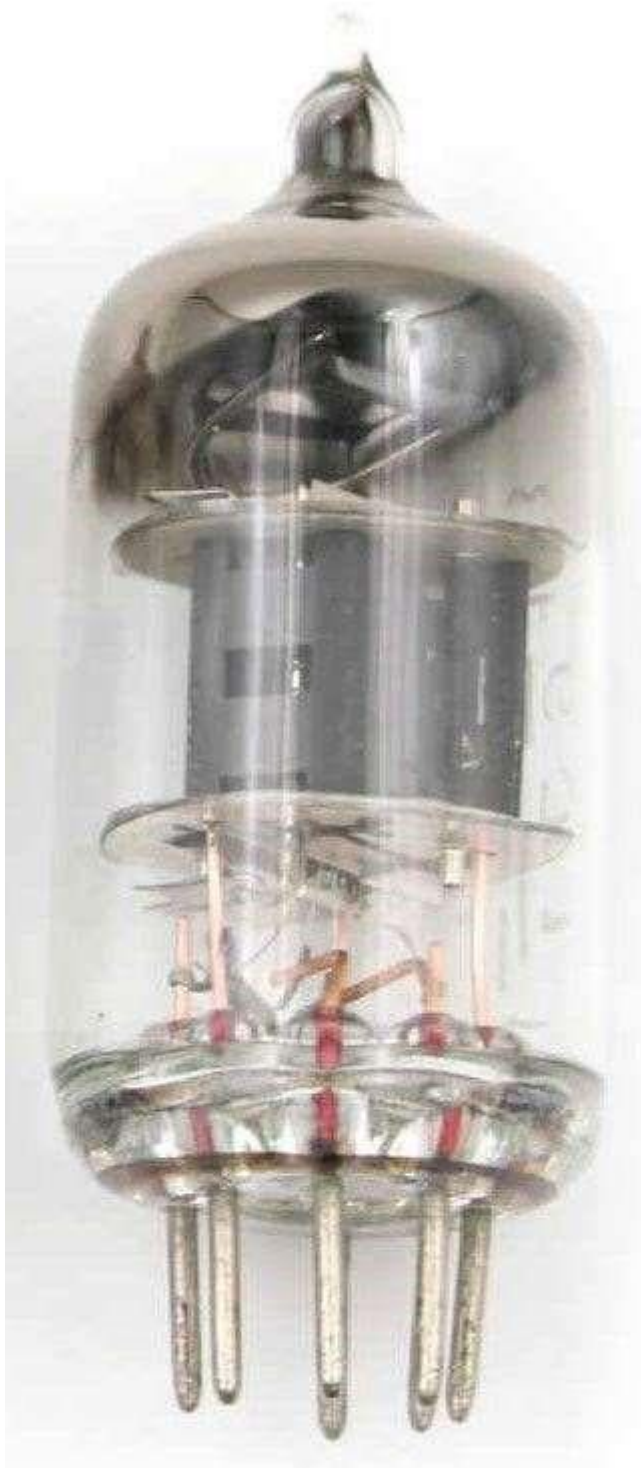
Použití knihovny není pro funkci nezbytné, avšak pouze výhodné při tvorbě programu. Na ethernet dokáže samostatně obsloužit ethernetovou komunikaci, včetně komunikace mezi převodníkem na SPI a MCU. Zaslání dat na sériovou linku nám umožňuje kontrolu při inicializaci ethernetu. Například je možné kontrolovat nadefinovanou IP adresu (případně i MAC adresu) zařízení. Při změně stavu na ethernetu nebo například v případě, že na ethernet přichází různé bloky dat lze pomocí sledování sériové linky pomocí sériového monitoru v prostředí IDE správnost příchozích dat. Tato data mohou představovat jak text (typ CHAR), tak číslo (datový typ INT aj.). Při dalším ladění programu je praktické použití této funkce pro kontrolu výstupních veličin funkcí postavených na základě cyklů (cyklicky je tedy zapisována daná hodnota do daného objektu programu). [8]



Obr: 14 SPI propojení

Stručná historie řešeného problému

Pokud bychom chtěli řešit historii daného tématu tak bych se podíval do historie na elektronkové žárovky a jejich ovládání. Takové elektronky byli velice jednoduché zařízení a postupem času se dostali na diody a z diod led display. A co se ovládacího prvku týče tak bych se podíval na staré mikroprocesory, které se programovali v programovacím jazyce basic anebo assembler.



Obr: 15 Elektronka

3. Praktická část

Mou SOČ práci jsem začal hledání informací a součástek která bych mohl použít.

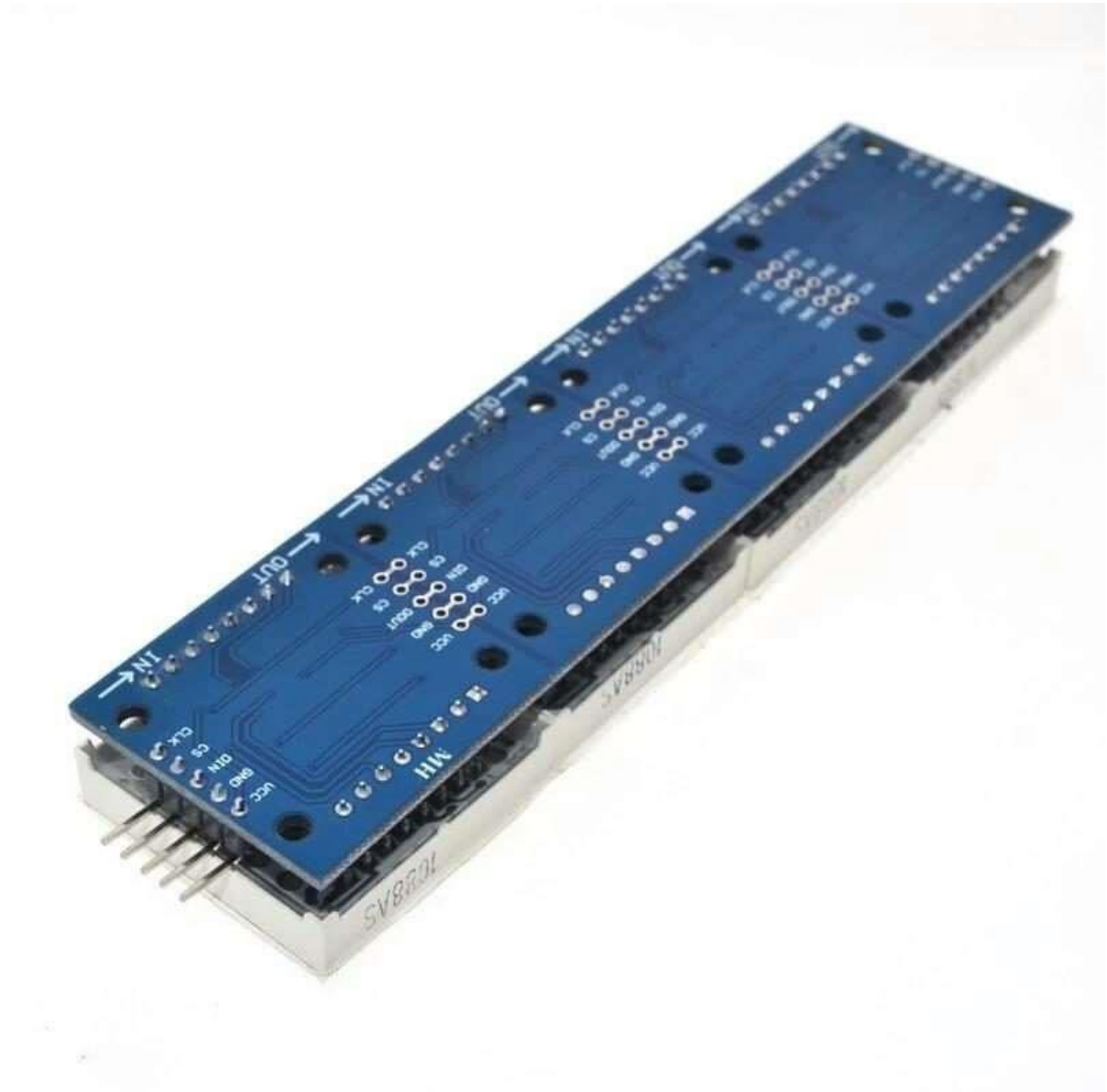
Na začátku jsem si říkal že by bylo třeba dobré si spájet ledky a udělat si plochu a tu pak programovat ale nakonec jsem se rozhodl pro to, že koupím hotovou led matici. Poté jsem se rozhodoval přes co budu matici programovat pomocí jakého jazyka a čím to budu pohánět. Jelikož jsme nějakou dobu dělali ve škole s Arduinem i když jsem okrajově tak jsem se rozhodl pro Arduino a programovací jazyk c++. Poté co jsem vybrané dané součástky a objednané tak jsem koukal po internetu po nějakých projektech které fungují na stejném nebo podobném principu. První zmínku, kterou jsem našel tak byl přímo daný program na Arduino a led matici. Když jsem se na program koukal tak jsem zjistil, že je sestaven ze 2 knihoven a minima čistě psaného textu. Řekl jsem si, že bych to mohl upravit přeložit knihovny a použít to ale nakonec jsem se rozhodl že budu ještě hledat dál. Po nějaké době jsem našel použití pouze jedné knihovny I které jsem nakonec zůstal s touto knihovnou jsem sepsal program na staticky text na matici. Když jsem začínal tento program psát začal jsem si hledat co vše tato knihovna dovede abych byl schopný program sepsat a rozuměl tomu. Začal jsem nejdříve s tím, jak připojím Arduino s maticí a přes jaké piny to připojím.

Zapojení bylo jednoduché a programově také díky knihovně MD_MAX72xx jsem nastavení digitálních vstupů měl jednoduché.

```
#define      MAX_DEVICES    8

#define      CLK_PIN        6
#define      DATA_PIN      7
#define      CS_PIN          5
```

```
MD_MAX72XX mx = MD_MAX72XX (CS_PIN, MAX_DEVICES);
```



Obr: 16 Led matice piny

Díky jednoduchým příkazům z knihovny jsem si nastavil maximální počet matic a k tomu dané vstupy. Pak přišel na řadu samotný program, kde jsem začal tím, že jsem si nastavil proměnou, do které si nahraji danou větu na zobrazení.

```
#define CHAR_SPACING 1
#define BUF_SIZE 75
char message [BUF_SIZE] = {"Dobry den"};
```

Nejdříve jsem si definoval mezeru, která bude mezi znaky. Poté jsem si definoval buffer, co je proměnná na dané znaky a určil jsem jí velikost. Poté jsem si vytvořil proměnou message, do které jsem si nahrál danou větu a dal jí stejnou velikost jako bufferu. Jako další krok přichází na řadu samotné načtení textu na matici. Prvně jsem si ručil proměnou,

kteřá mi bude hlídat počet znaků v textu a po dosažení posledního znaku zastaví zobrazování dalšího znaku.

```
void readSerial(void)
{
    static uint8_t putIndex =
    0;

    while (Serial.available())
    {
        message[putIndex] = (char)Serial.read();          if
        ((message[putIndex] == '\n') || (putIndex >= BUF_SIZE-3))

        {
```

Touto část programu jsem nevymyslel já sám, ale využil jsem ji z programu, který vytvořil někdo jiný. Ale nejsem schopný na tuto část udělat citaci, jelikož je to jen v programu a ani nevím název muže ani firmy. Proto bychom tímto chtěl sdělit, že toto není má část programu nýbrž vypůjčená.

Poslední příkazový řádek jediné, co dělá je vlastně to, že čeká buď na to až počítač dojde na poslední znak anebo až přeteče buffer. Jako další na řádu přijde samotné zobrazování textu na matici. Na této části jsem se na nějakou dobu zasekl, jelikož jsem si nevěděl rady, jak to přesně udělat, ale pak jsem našel video, na kterém jsem našel docela zajímavý způsob, jak to udělat, a to tím že se použije funkce Switch. Jako první na řádu přišlo založení potřebných proměnných, která budeme potřebovat, a to byla proměnná na jmenovaný switch poté proměnou která bude ukazovat na znaky které jsme zadali a proměnou která bude brát daná písmena z listu s fontem a proměnou která uchovává hodnotu zrovna ukazovaného znaku a proměnou která udává velikost sloupce na matici. Hlavní program začíná načtením znaku z listu s fontem a pokud se dostaneme s ukazatelem který ukazuje na zadanou větu na konec tak se nastaví zpět na začátek a přesune nás ve switchi do druhé části, pokud ne tak nahrajeme další znak z listu s fontem a záměrně se přesuneme do druhé části vrchní přesunutí do druhé části je jen pro případ, kdyby počítač načetlo špatnou hodnotu tak aby se nezobrazovali špatné znaky. V druhé části zobrazíme následující znaky a poté zobrazíme prázdné sloupce které jsou na matici a pošle nás do třetí části. Ve třetí části ošetříme to, aby se koncem znaků nezačali znovu zobrazovat ty samé znaky, a proto po přečtení posledního znaku v zadané větě vyplní program zbytek matice prázdnými sloupci a ukončí celý cyklus. Poté co jsem dopsal tento program jsem se snažil najít, jak bych mohl tento text rozpořehovat našel jsem docela hodně možností, jak to dělat, ale některé z nich mi přišli až moc jednoduché a že by tam

nebyla vůbec žádná přidaná moje hodnota, proto jsem hledal dále a našel jsem knihovnu Max Matrix je to docela dost jednoduše sepsaná knihovna, která se jednoduše používá. Při začátku jsem se seznamoval s příkazy knihovny a po chvíli už jsem chápal, jak funguje sepsal jsem si vlastní font list a začal jsem psát hlavní program. Na začátku mi to přišlo vážně jednoduché všechno šlo, jak jsem chtěl, ale pak když jsem vyzkoušel program zjistil jsem, že text mi neběhá tím směrem, kam jsem chtěl. Snažil jsem se přijít na problém proč se mi to otáčí jiným směrem pokusil jsem se změnit program dokonce jsem hledal i problém v zapojení ale problém jsem i tak nenašel. Dokonce i když jsem změnil směr otáčení přímo v programu tak místo toho, aby se to otočilo tak mi program dokonce přestal i fungovat. Ale jelikož už mi nezbývalo moc času na celkové dokončení SOČ práce tak jsem se rozhodl, že pouze připišu můj pokus o rozpořívání textu i když se nepovedl.

Postup řešení

Začátek postupu řešení začne výměrem správných součástek. Vytvoření výkresu a následné sestavení. A Poté jde už jen o část programu. Jako první vytvořím vývojový diagram poté následně program a dozkoušení daného programu a na konec vytvoření krabičky pro celek.

Panel na zobrazování textu – nenašel jsem přímo naprogramovaný panel pouze dané součástky, ale našel jsem stejný projekt až na to, že tento panel byl z matice 8x16 což si myslím, že není až tak dobrý na čitelnost, a navíc program byl napsaný pro angličtinu. Pro zprovoznění jsem si vybral Arduino a přímo typ jsem zvolil Uno, protože mega by byla moc velký a zbytečná a nano je zase moc malé a má málo vstupů. Matici jsem vybral 8x64 aby byl text jasně čitelný. Pro napsání programu jsem vybral přímo program od Arduina, jelikož z něj to budu rovnou posílat do Arduina a zkoušet.

4.Závěr – zhodnocení dosažených výsledků

Mou SOČ práci se mi povedlo dovést ke zdárnému konci. Dle zadání SOČ práce se mi povedli splnit veškeré body které jsem si uvedl ve své SOČ práci. V průběhu dělání své SOČ práce jsem se chtěl rozhodnout, že bych místo statického textu použil text s

animací, ale když jsem začal dělat textovou animaci tak se naskytlo mnoho problémů. Prvním problémem byla knihovna, kterou používám na statický text tak byla příliš složitá na to abych pochopil animaci. Druhý problém bylo najít jednoduchou pomocnou knihovnu na animaci tu jsem hledal poměrně dlouho ale postupem času jsem našel. Když už jsem našel použitelnou knihovnu začal jsem dělat na animaci. Když už jsem se blížil ke konci kdy to vypadalo, že už by animace mohla fungovat přišel největší problém. Pomocná knihovna měla příkaz na otočení textu u animace, ale místo toho, aby se mi otočil text na led matici a začal se pohybovat přes všech osm led matic tak se začal hýbat v každé matici zvlášť. Tento problém jsem si myslel že vyřeším vcelku jednoduše ale postupem času jsem začal zjišťovat, že to nebude až tak jednoduché. A z důvodu toho že jsem se blížil k datumu odevzdání mé práce tak jsem se rozhodl, že bude jednodušší nechat SOČ práci se statickým textem a mít splněné celé zadání než se do konce termínu snažit zprovoznit animaci, a nakonec to nemít celé hotové. Ale z důvodu toho že budu chtít pokračovat na vysokou školu tak jsem si z toho nedělal až tak velkou hlavu, jelikož aspoň vím, že pokud se na vysokou školu dostanu budu moci pokračovat ve své SOČ práci a dostat jí na vyšší stupeň a dodělat zmiňovanou animaci. Při děláním mé SOČ práce jsem si zvýšil znalosti v programování v programovacím jazyce C a C++ a dále ještě práci v programech na výkresy a vývojové diagramy. Svou práci bych zhodnotil asi tak že si myslím, že se mi povedla, ale je tam ještě docela hodně věcí, které bych mohl dotáhnout do dokonalosti.

5. Seznam použitých zdrojů a literatury

[5] Průvodce [online]. 2014 [cit. 2022-04-11]. Dostupné z:

<https://www.robotikabrno.cz/docs/arduino/Průvodce-světlem-Arduina-CZ.pdf>

[7] Arduino [online]. [cit. 2022-04-11]. Dostupné z:

<https://cs.wikipedia.org/wiki/Arduino>

[1] Arduino Esplora [online]. [cit. 2022-04-11]. Dostupné z:

https://cs.wikipedia.org/wiki/Arduino_Esplora

[2] Arduino Mega2560 [online]. [cit. 2022-04-11]. Dostupné z:

https://cs.wikipedia.org/wiki/Arduino_Mega2560

[3] Arduino Mini [online]. [cit. 2022-04-11]. Dostupné z:

https://cs.wikipedia.org/wiki/Arduino_mini

[4] Arduino Nano [online]. [cit. 2022-04-11]. Dostupné z:

https://cs.wikipedia.org/wiki/Arduino_Nano

[6] Led [online]. [cit. 2022-04-11]. Dostupné z: <https://cs.wikipedia.org/wiki/LED>

[8] MAREK, Michal. MATICOVÝ LED DISPLAY 8X64 [online]. Brno, 2015 [cit. 202204-11]. Vysoké učení technické v Brně.

6. Seznam obrázků

Obr: 1 Zapojení Arduina a matice	3
Obr: 2 Arduino Uno	5
Obr: 3 Arduino Esplora	6
Obr: 4 Arduino Mega	6
Obr: 5 Arduino Mini	7
Obr: 6 Arduino Nano	8
Obr: 7 Vnitřní zapojení led matice	9
Obr: 8 Led matice.....	10
Obr: 9 Arduino Duemilanove	10
Obr: 10 ATMega1280	11
Obr: 11 Led dioda červená	12
Obr: 12 Diody graf	14
Obr: 13 Programovací jazyk C	15
Obr: 14 SPI propojení	16
Obr: 15 Elektronka	17
Obr: 16 Led matice piny	19

7. Přílohy

1. Výkres vnější
2. Výkres vnitřní
3. Vývojový diagram
4. Video