|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| **Závěrečná studijní práce**  **dokumentace** | | |  |
| **Distance Warning Device - DWD** | | |  |
| Andreas Piskoř | | |  |
|  | | |  |
|  | |  |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |  |
| **Třída:**  **Školní rok:** | IT4  2020/2021 | |  |

#### Poděkování

*Děkuji panu učiteli Ing. Petru Grussmannovi za rady a vylepšení projektu, panu učiteli Mgr. Markovi Lučnému za cenné rady, panu učiteli Mgr. Marcelu Godovskému za rady ohledně napajení a predevším panu Ing. Jiřímu Miekischovi za pomoc se zapojením hardwaru a ukázku z pájení.*

Prohlašuji, že jsem závkiěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2020

*podpis autora práce*

**ANOTACE**

Projekt se zabývá drzěním bezpečných odstupů od ostatních lidí kvůli pandemii. Skládá se z hardwarové a softwarové části. Softwárová část je napsaná v Platformiu IDE, což znaména, že kód je napsán v kombinaci jazyků C a C++. V této části si může uživatel měnit maximální vzálenost od osob jak jen ultrazvukový senzor dovolí. V hardwarové části je základem čip Arduino nano, který celý projekt řídí. Ultrazvukový senzor měrí vzdálenost od objektů (lidí), hodnoty ze senzoru se vypisují na LDC Display a je-li maximální dovolená hodnota překročena, začne tonizovat bzučák a svítit červená LED, naopak když je hodnota vzdálenosti v pořádku svítí zelená LED. Vše je propojeno na nepájivém poli.

**Klíčová slova**

Platformio IDE; C++; C; Arduino nano; ultrazvukový senzor; LCD Display; LED

OBSAH

[Úvod 6](#_Toc370246085)

[1 Výroba distančního systému 7](#_Toc370246086)

[2 Využité technologie 8](#_Toc370246087)

[2.1 Hardware 8](#_Toc370246087)

[2.1.1 Seznam součástek 8](#_Toc370246087)

[2.1.2 Arduino nano 9](#_Toc370246087)

[2.1.3 Ultrazvokový senzor HC-SR04](#_Toc370246087) 11

[2.1.4 LCD Display 16x02 I2C 1](#_Toc370246087)2

[2.1.5 Bzučák 14](#_Toc370246087)

[2.1.6 Tlačítko 14](#_Toc370246087)

[2.1.7 LED 15](#_Toc370246087)

[2.1.8 Resistor 15](#_Toc370246087)

[2.2 Napájení 9V baterie 16](#_Toc370246087)

[2.3 Software 17](#_Toc370246087)

[2.3.1 VS Code 17](#_Toc370246087)

[2.3.2 Platformio IDE 17](#_Toc370246087)

[3 Způsoby řešení a použité postupy 18](#_Toc370246088)

[3.1 Funkčnost součástek 18](#_Toc370246087)

[3.1.1 SEnzorová část 18](#_Toc370246087)

[3.1.2 displayová část 19](#_Toc370246087)

[3.1.3 bzučáková část 20](#_Toc370246087)

[3.2 vše dohromady 21](#_Toc370246087)

[3.2.1 problém s napájením 21](#_Toc370246087)

[3.2.2 Problémy s knihovnami 21](#_Toc370246087)

[4 Výsledky řešení 22](#_Toc370246089)

[4.1 Finální Podoba Hardwaru 22](#_Toc370246087)

[4.2 finální podoba kódu 23](#_Toc370246087)

[4.3 Schéma 24](#_Toc370246087)

[Závěr 25](#_Toc370246090)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 26](#_Toc370246091)

[Seznam příloh 27](#_Toc370246092)

Úvod

Rozhodl jsem se udělat tento projekt z toho důvodu, že na jaře tohoto roku vypukla celosvětová pandemie COVIDU-19. Chci aby lidé dbali na rozestupy, které si mají držet od ostatních lidí tak, aby nemoc nedostali či neroznesli. U zařízení se může nastavit hodnota v kódu, kde se může libovolně měnit. Zařízení nemusí nutně sloužit jen na tento případ, když si vezmete novější auto a začnete couvat zjistíte, že zařízení v tom autě funguje na obdobném principu, akorát trochu inteligentněji.

Na tento projekt jsem vzhlížel tak, aby skutečně fungoval a měl i svoje vychytávky, protože to by nemohl být projekt jak se patří. Hardwarovou část tvoří tudíž čip Arduino nano, který má hodně výhod. Je zde propojeno mnoho součástí, o kterých se rozepíšu později u Hardwarové části. U softwarové části jsem místo původního Arduina IDE vybral Platformio IDE. Cíle tutíž byly v tento moment jasné, udělat nejlépe funkční měřění se senzorem a aby se hodnota vypisovala na display a když se hodnota překročí začne bzučet bzučák.

V této dokumentaci se rozepíšu hlavně o problematice u součástek a napájení. Nebude chybět ani samotné schéma projektu a taky doufám, že u čtení neusnete. Na závěr práci zhodnotím a kouknu co se mi povedlo a co naopak ne.

# Výroba distančního systému

První část projektu se zabývala postavením a konstrukcí zařízení. Bylo to moje poprvé co jsem musel něco pájet, aby se to nepodělalo, protože součástky byli koupeny s mojí peněženky. Součástky jsem vybíral bystře, protože jak jeden moudrý člověk řekl “Jak si usteleš, tak si také lehneš.”, a proto jsem si radši připlatil za lepší. Tudíž jsem nekupoval nic z Číny, ale u překupníku v České Republice a to se vyplatí.

Nejdříve jsem si říkal jestli nepoužít jako jádro projektu ESP8266, jenže jsem si řekl, že je zbytečné tento modul kupovat. Když projekt nevyžaduje Wi-Fi a také když už doma mám Arduino nano. Arduino nano je vytvořené na takové malé projekty jaký je třeba iten můj. Je taky založeno na mikrokontroleru ATmega328 od firmy Atmel.

Ostatní součastky jsem nějak moc neřešil, takže jsem se podíval co mám doma a co ne a objednal ty co nemám z Arduino shopu. Jednalo se především o senzor, display a resistory.

Po zapojení všech součástek, jsem se rozhodoval zda použít programovací prostrědí Arduino IDE nebo Platformino IDE. A na základě tvaru obličeje pana učitele Grussmanna jsem se rozhodl pro Platformio. I když si do téď říkam, že to byla chyba. Ale na druhou stranu jsem si procvičil jazyk C a C++.

# Využité technologie

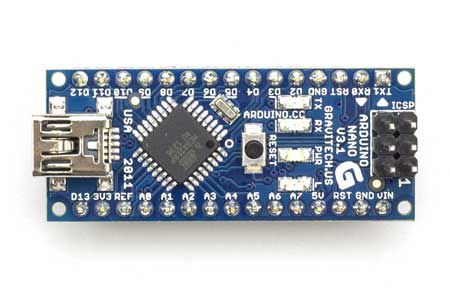
**2.1 Hardware**

2.1.1 Seznam součástek

* Arduino nano,
* ultrazvukový senzor HC-SR04,
* LCD display 16x02 I2C,
* bzučák,
* tlačítko,
* červená LED,
* zelená LED,
* resistor 100 Ω
* 2x resistor 220 Ω

**2.1.2 Arduino nano**

Je v [informatice](https://cs.wikipedia.org/wiki/Informatika" \o "Informatika) název malého [jednodeskového počítače](https://cs.wikipedia.org/wiki/Jednodeskov%C3%BD_po%C4%8D%C3%ADta%C4%8D" \o "Jednodeskový počítač) z [otevřené](https://cs.wikipedia.org/wiki/Otev%C5%99en%C3%BD_software" \o "Otevřený software) [platformy](https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A1_platforma" \o "Počítačová platforma) [Arduino](https://cs.wikipedia.org/wiki/Arduino" \o "Arduino), který je založen na [mikrokontrolerech](https://cs.wikipedia.org/wiki/Jedno%C4%8Dipov%C3%BD_po%C4%8D%C3%ADta%C4%8D" \o "Jednočipový počítač) [ATmega](https://cs.wikipedia.org/wiki/AVR" \o "AVR) od firmy [Atmel](https://cs.wikipedia.org/wiki/Atmel" \o "Atmel). Arduino Nano je podobné Arduinu Mini, s tím rozdílem, že má vlastní USB port a převodník. Je tedy trošku větší, ovšem není třeba mít extra převodník při vkládání programu.



*Obrázek č. 1 Arduino nano*

Vybral jsem si Arduino nano, protože jsem s ním pracoval i v minulosti a pracovalo se mi s ním velice slušně. Sice jedno během projektu to nepřežilo, ale to nebyl podmět k tomu vzdávat se. Taky jsem si ho vybral proto, že jsem už ho doma měl a nechtělo se mi pracovat s něčím co moc neznám např: ESP8266.

Dělal jsem na něm hodně pokusů ještě než jsem se dostal k zavěrečnému projektu a byli pozitivní a už jen z tohoto hlediska jsem usoudil, že Arduino nano je to správné řešení na projekt.

## **Technické informace o Arduinu**

|  |  |
| --- | --- |
| Mikroprocesor | ATmega328 |
| Architektura | AVR |
| Provozní napětí (logická úroveň) | 5V |
| Vstupní napětí (doporučeno) | 7-12V |
| Počet digitálních I/O pinů | 22 pinů, z toho 6 s PWM |
| Počet analogových vstupů | 8 pinů |
| Proudové zatížení na 1 pin | 40 mA |
| Flash paměť | 32 KB |
| SRAM | 2 KB |
| EEPROM | 1KB |
| Rychlost hodin | 16 MHz |
| Výška | 18 mm |
| Šířka | 45 mm |
| Váha | 7g |

Arduino vlastní 22 digitálních pinů, což umožňuje přidávat hodně vychytávek na projekty a má i 8 digitálních. Do arduina se můžou i ukládat hodnoty pomocí EEPROM. Je malé a dobře se s ním manipuluje. Prostě na můj projekt jak dělané.

**2.1.3 Ultrazvokový senzor HC-SR04**

Automatické měření vzdálenosti nemusí být zdaleka takový problém, jak by se mohlo na první pohled zdát. **Pomocí ultrazvukového principu můžeme vzdálenost měřit velmi pohodlně**, bezkontaktně a dokonce přesně. Ultrazvukový měřič vzdálenosti najde využití v mém projekt úplně přesně, kde se kontroluje vzdálenost od objektu. K dispozici je Arduino knihovna a tak je získávání dat ze senzoru velmi jednoduché.



*Obrázek č. 2 HC-SR04*

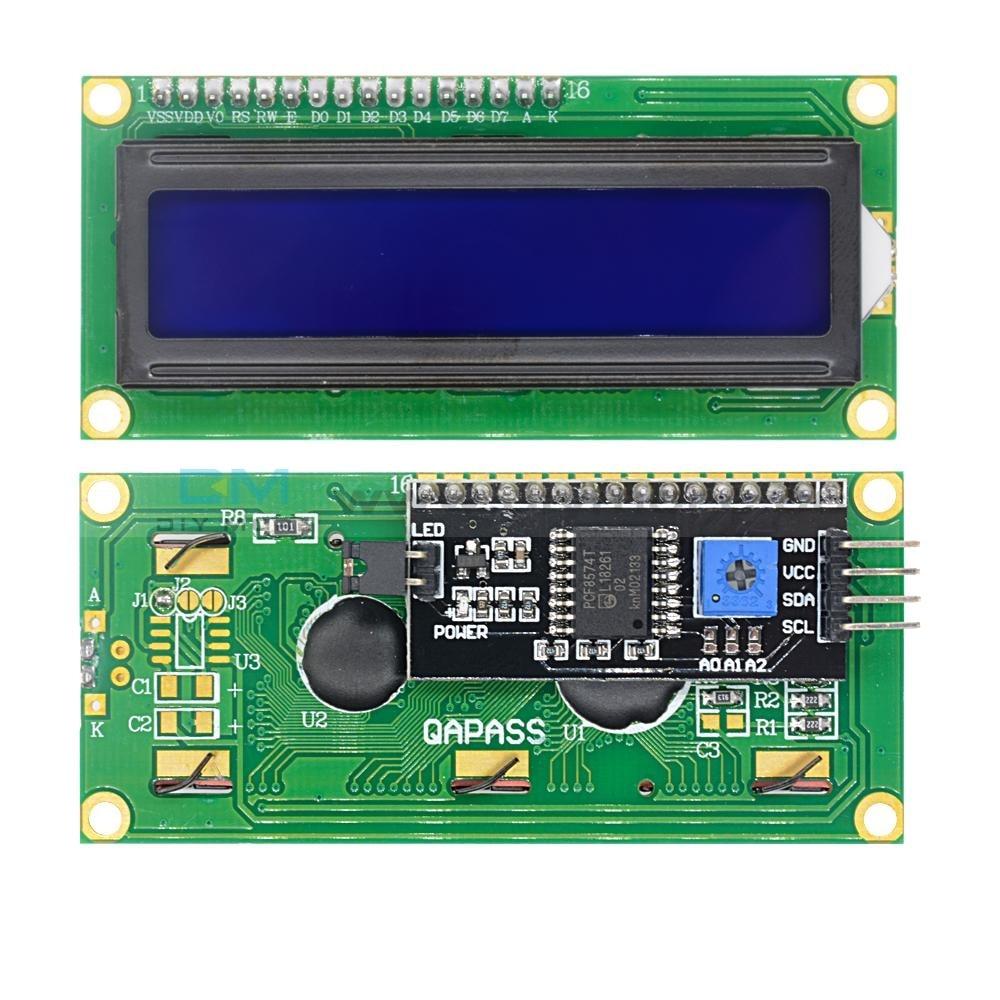
#### ****Specifikace:****

* Pracovní rozsah: 2-400cm
* Vysoká přesnost: až 3mm
* Pracovní úhel: <15 stupňů
* Klidový proud:< 2mA
* Napájecí napětí: 5VDC
* Velikost: 45x20x16mm

Tento senzor jsem si vybral, protože mi ho pan učitel Grussmann prodal.

**2.1.4 LCD Display 16x02 I2C**

I2C\_LCD je snadno použitelný zobrazovací modul, který může usnadnit zobrazení. Jeho použití může snížit obtížnost zapojení, aby jsem se mohl soustředit na jádro práce. Knihovna Arduino pro I2C\_LCD usnadňuje uživatelům práci, potřebuje jen pár řádků kódu, aby dosáhl složitých funkcí grafiky a zobrazení textu. Může na nějakém místě nahradit sériový monitor Arduina, můžete získat informace o běhu bez počítače a to já zrovna potřeboval.



*Obrázek č. 3 LCD Display 16x02 I2C*

Tento display má strašně moc výhod oproti normálnímu LCD 16x02. Nejenže se lépe pracovalo s knihovnami, ale bylo i lehčí zapojení. Velkou výhodou je i podsvícení, takže ve i ve tmě se dá dobře přečíst co zrovna vypisuje.

**Funkce displaye**

* Obsazeny jsou pouze 2 piny Arduino (použijte rozhraní I2C).
* Podporuje standardní režim I2C (100 Kbit / s) a rychlý režim I2C (400 Kbit / s).
* Kompatibilní s několika úrovněmi komunikační logiky: 2,8 ~ 5VDC.
* Knihovna Arduino podporována, k dokončení zobrazení použijte řádek kódu.
* Integrujte 7 velikostí písem ASCll, 5 grafických funkcí.
* Poskytněte vyhrazený software pro převod obrazových dat (Bitmap Converter).
* Většinu složitých operací zpracovává nezávislý řadič I2C\_LCD, což šetří prostředky uživatelského řadiče.
* Podporuje funkci kurzoru, může nastavit frekvenci 16 blikání kurzoru.
* Podporuje 128 úrovní nastavení podsvícení.
* Podpora 64úrovňového nastavení kontrastu obrazovky.
* Podpora úpravy adresy zařízení.
* Podporuje paralelní práci 127 I2C\_LCD.
* Při ladění kódu může místo sériového monitoru sledovat stav běhu programu.
* K dispozici jsou dvě neobvyklé metody obnovení: reset a obnovení továrního nastavení.
* 4 symetrické pevné otvory pro snadnou instalaci.

**Specifikace:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parametry | Hodnota |  |  |
| Typ obrazovky | Dvoubarevný LCD |  |  |
| Rozlišení obrazovky | 128 \* 64 pixelů |  |  |
| Velikost jednotlivých pixelů | 0,33 \* 0,33 mm |  |  |
| Režim komunikace | I2C (100 Kbit / s a 400 Kbit / s) |  |  |
| Ovladač | STM8S005KBT6 |  |  |
| Provozní frekvence | 16 MHz |  |  |
| Hmotnost | 20 g |  |  |

**2.1.5 Bzučák**

Je elektro-akustický měnič, který přeměňuje [elektrickou energii](https://cs.wikipedia.org/wiki/Elektrick%C3%A1_energie" \o "Elektrická energie) na [mechanickou energii](https://cs.wikipedia.org/wiki/Mechanick%C3%A1_energie" \o "Mechanická energie) ve formě [zvuku](https://cs.wikipedia.org/wiki/Zvuk" \o "Zvuk). Obvykle se skládá z [membrány](https://cs.wikipedia.org/w/index.php?title=Technick%C3%A9_membr%C3%A1ny&action=edit&redlink=1" \o "Technické membrány (stránka neexistuje)), z pohonné části, do které je přiváděn vstupní [signál](https://cs.wikipedia.org/wiki/Sign%C3%A1l" \o "Signál) a dalších dílů. V mém projektu bude sloužit k tomu, aby bzučel když je porušena podmínka v kódu. To je tehdy když je překročena maximální dovolená vzdálenost od objektu.



*Obrázek č. 4 Bzučák*

**2.1.6 Tlačítko**

Je jednoduchý [spínač](https://cs.wikipedia.org/wiki/Sp%C3%ADna%C4%8D" \o "Spínač), který slouží k ručnímu ovládání elektrického zařízení. Hmatník tlačítka je obvykle vyroben z tvrdého materiálu (plastu nebo kovu) a tvar je přizpůsoben tak, aby bylo možné tlačítko obsluhovat tlakem prstu nebo ruky. Já talčítko v tomto případě potřebuji, abych mohl vpustit napětí z baterky do oběhu kdy chci.

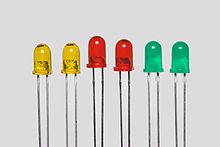


*Obrázek č. 5 Tlačitko*

**2.1.7 LED**

Je označení v elektrotechnice pro diodu, která emituje světlo případně [infr-červené](https://cs.wikipedia.org/wiki/Infra%C4%8Derven%C3%A9_z%C3%A1%C5%99en%C3%AD" \o "Infračervené záření) nebo [ultra-fialové](https://cs.wikipedia.org/wiki/Ultrafialov%C3%A9_z%C3%A1%C5%99en%C3%AD" \o "Ultrafialové záření) záření, čímž se liší od standardní diody. LED vyzařuje z obnaženého [PN přechodu](https://cs.wikipedia.org/wiki/PN_p%C5%99echod" \o "PN přechod), a vede [stejnosměrný proud](https://cs.wikipedia.org/wiki/Stejnosm%C4%9Brn%C3%BD_proud" \o "Stejnosměrný proud) pouze jedním směrem. Na rozdíl od [žárovky](https://cs.wikipedia.org/wiki/%C5%BD%C3%A1rovka" \o "Žárovka) dosahuje vysoké účinnosti, je mechanicky odolná, levná na výrobu, a proto je čím dál více využívána.

Já v projektu využívám 2 LED a to jednu červenou a jednu zelenou. Červená svítí když je překročena maximální dovolená vzdálenost od objektu. Zelená naopak když je vše v pořádku a je vzálenost dodržena.



*Obrázek č. 6 LED*

**2.1.8 Resistor**

Je pasivní [elektrotechnická součástka](https://cs.wikipedia.org/wiki/Elektrotechnick%C3%A1_sou%C4%8D%C3%A1stka" \o "Elektrotechnická součástka) projevující se v [elektrickém obvodu](https://cs.wikipedia.org/wiki/Elektrick%C3%BD_obvod" \o "Elektrický obvod) v ideálním případě jedinou vlastností – [elektrickým odporem](https://cs.wikipedia.org/wiki/Elektrick%C3%BD_odpor" \o "Elektrický odpor) (jednotka [Ohm](https://cs.wikipedia.org/wiki/Ohm" \o "Ohm), značka Ω). Důvodem pro zařazení rezistoru do obvodu je obvykle snížení velikosti [elektrického proudu](https://cs.wikipedia.org/wiki/Elektrick%C3%BD_proud" \o "Elektrický proud) nebo získání určitého úbytku [napětí](https://cs.wikipedia.org/wiki/Elektrick%C3%A9_nap%C4%9Bt%C3%AD" \o "Elektrické napětí). Rezistory se také mohou používat jako topné články, testovací zátěže pro [generátory](https://cs.wikipedia.org/wiki/Elektrick%C3%BD_gener%C3%A1tor" \o "Elektrický generátor) apod. Rezistory rozdělujeme na pevné a proměnné. Pevné rezistory mají pevně danou hodnotu odporu, která se mírně mění pouze v závislosti na teplotě, procházejícím proudem a životnosti rezistoru a proměnné nepoužívám, ale z nazvů chytíte, že se dá měnit jejich hodnota.

Resistory mám v projektu přesně 3. Jeden s hodnotou 100 Ω a dva s hodnotou 220 Ω. V mém případě složí k tomu, aby do součástek nešlo moc napětí a nezničili se tak.

**2.2 Napajení 9V baterie**

Devíti-voltovka nebo 9V baterie, je obyčejná velikost baterie, která byla zavedena počátkem [tranzistorov](https://en.wikipedia.org/wiki/Transistor_radio" \o "Tranzistorové rádio)é doby. Má obdélníkový hranolový tvar se zaoblenými hranami a nahoře má polarizovaný snap konektor. Formát devíti voltových baterií je běžně dostupný v primární uhlík-zinkové podobě.

Většina 9V alkalických baterií je vyrobena ze šesti samostatných článků 1,5 V LR61 uzavřených v obalu. Tyto články jsou o něco menší než články [AAAA](https://en.wikipedia.org/wiki/AAAA_battery" \o "Baterie AAAA) LR8D425 a lze je místo nich použít pro některá zařízení, i když jsou o 3,5 mm kratší. Uhlík-zinkové typy jsou vyrobeny se šesti plochými buňkami v hromadě, uzavřené v obalu odolném proti vlhkosti, aby se zabránilo vysychání.

A přesně tohle je moje baterie. Jediný problém co jsem s ní měl byl ten, že jsem musel ještě na ní koupit konektor, abych s ní mohl napájet součástky v nepájivém poli.



*Obrázek č. 7 9V baterie*

**2.3 Software**

**2.3.1 VS Code**

je [editor zdrojového kódu](https://cs.wikipedia.org/w/index.php?title=Editor_zdrojov%C3%A9ho_k%C3%B3du&action=edit&redlink=1" \o "Editor zdrojového kódu (stránka neexistuje)) vyvíjený společností [Microsoft](https://cs.wikipedia.org/wiki/Microsoft" \o "Microsoft) pro [operační systémy](https://cs.wikipedia.org/wiki/Opera%C4%8Dn%C3%AD_syst%C3%A9m" \o "Operační systém) [Windows](https://cs.wikipedia.org/wiki/Windows" \o "Windows), [Linux](https://cs.wikipedia.org/wiki/Linux" \o "Linux) a [macOS](https://cs.wikipedia.org/wiki/MacOS" \o "MacOS). Obsahuje podporu pro [Git](https://cs.wikipedia.org/wiki/Git" \o "Git) (a pro [GitHub](https://cs.wikipedia.org/wiki/GitHub" \o "GitHub)), [zvýraznění syntaxe](https://cs.wikipedia.org/wiki/Zv%C3%BDrazn%C4%9Bn%C3%AD_syntaxe" \o "Zvýraznění syntaxe), kontextový [našeptávač](https://cs.wikipedia.org/wiki/Na%C5%A1ept%C3%A1va%C4%8D" \o "Našeptávač) a podporu pro [ladění](https://cs.wikipedia.org/wiki/Lad%C4%9Bn%C3%AD_(programov%C3%A1n%C3%AD)" \o "Ladění (programování)) a [refaktorizaci](https://cs.wikipedia.org/wiki/Refaktorizace" \o "Refaktorizace). Zdrojový kód je [svobodný software](https://cs.wikipedia.org/wiki/Svobodn%C3%BD_software" \o "Svobodný software) pod [licencí MIT](https://cs.wikipedia.org/wiki/Licence_MIT" \o "Licence MIT). Sestavené binárky nabízené přímo Microsoftem jsou [freewarem](https://cs.wikipedia.org/wiki/Freeware" \o "Freeware) obsahujícím [telemetrii](https://cs.wikipedia.org/wiki/Telemetrie" \o "Telemetrie), ale existuje i komunitně sestavovaná varianta VSCodium.

VS Code jsem vybral proto, že už vím jak se s ním pracuje z minulých ročníků a taky jsem potřeboval extension Platformio IDE. Ve VS Code se dělají především webové stránky, ale může se zde i programovat v C++, čehož jsem využil.

Jak už jsem se zmiňoval tak jsem neměl moc na výběr kde můžu programovat, protože pan učitel Grussmann má rád, když se všechno stěžuje. Takže o Arduinu IDE jsem si mohl jen nechat zdát.

´ **2.3.2 Platformio IDE**

Nejlepší vývojové IDE pro Microsoft VS Code. Uživatelsky přívětivé a rozšířitelné integrované vývojové prostředí se sadou nástrojů pro profesionální i amaterský vývoj. Poskytuje moderní a výkonné funkce pro zrychlení a zjednodušení vytváření kódu.

Platformio jsem si vybral, protože už mám s ním zkušenost z minulých let.

**Proč zrovna Platformio IDE?**

Lehký, ale výkonný editor zdrojového kódu pro různé platformy.

Inteligentní dokončení kódu na základě typů proměnných, definic funkcí a závislostí knihoven.

Pracovní postup pro více projektů se snadnou navigací po kódové základně projektu, více podoknech a podpoře motivů.

# Způsoby řešení a použité postupy

**3.1 Funkčnost součástek**

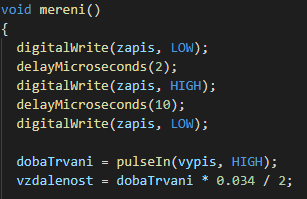
Funkčnost součástek jsem řešil tak, jak by to řešil každý člověk. Vetšinu jsem změřil volt metrem jestli jimi vůbec prochází nějaký proud. Hned tedy ze začatku jsem se potýkal s problémy, že mi nějaké součástky nešli. Tak jsem narychlo musel znánět některé z nich a naštěstí mi přišli včas, takže jsem mohl rychle zase pracovat.

Projekt jsem si pak rozdělil na 3 části.

* Senzorová
* Displayová
* Bzučáková

**3.1.1 Senzorová část**

V této části jsem se zabýval hlavně jestli senzor bude fungovat samostatně a jestli Serial monitor bude vypisovat hodnoty měřené senzorem. Hned na začátek se musím přiznat jsem neměl nejmeší ponětí jak jím měřit. Musel jsem zkouknout vice než pár videí a gihubů, než jsem našel to co jsem hledal. Nakonec se mi podařilo najít efektivní měřící metodu, která byla rychlá a spolehlivá.



*Obrázek č. 8 Ukázka měření*

Měřění funguje na tom principu, že když je status HIGH, provede se operace pulseIn a vypíše měřenou hodnotu.

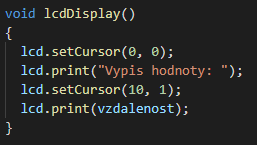
**3.1.2 Displayová část**

Tato část se zabývá výpisem na LCD display. K tomuto jsem využil knihovnu LiquidCrystal\_I2C, která vlastně umožňuje i jednoduší zápis v kódu. Místo asi 8 řádku jsem tak napsal jen jeden, ve kterém je napsané všechno potřebné k fungování displaye.



*Obrázek č. 8 Display identifikace*

Na výpis jsem přišel dokonce sám, protože tato část kódu byla jednoduchá. Stačilo přidat funkci lcd a jen vypsat pomocí printu a ještě nastavit kde se má hodtota vypsat na displayi. V tom mi pomohla funkce setCursor.

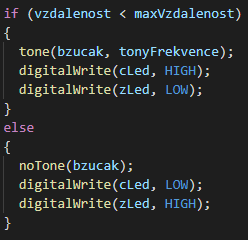


*Obrázek č. 9 Ukázka výpisu na display*

Na displayi bylo jen nutné to, aby jste si nespletli dráty a správně zapojily. Pak napsali kus kódu a měli jste jistotu, že bezchybně funguje.

**3.1.3 Bzučáková část**

V poslední části jsem zkoušel jestli funguje bzučák společně s ledkami. Tudíž jsem zapojil a zkusil jsem napsat kus kódu. Tady jsem dlouho vymýšlel podmínku, kdy bude bzučák bzučet a jak budou svítit či blikat ledky. Nakonec jsem vymyslel, že ledky budou jen svítit a to zase statusem HIGH nebo LOW. Bzučák je vyřešený statusem tone a notone. A to všechno v jedné podmínce, která je zárověň jednoduchá a zarověň řídí skoro všechno.



*Obrázek č. 10 Hlavní podmínká*

Zkoušel jsem o dost více promakanější podmínky, ale buď nefungovaly tak jak jsem chtěl nebo zkrátka nefungovali vůbec. Tak sem dal na zlatou střední cestu a tato funguje úplně bezchybně.

**3.2 Vše dohromady**

Když jsem viděl, že všechno funguje jak má tak jsem části spojil. Projekt se mi zdál hotov, ale ne nadarmo se říká “Neříkej hop dokud nepřeskočíš”. Všechno začalo naráz blbnout a já musel vymýšlet co s tím. Jednatk senzor měřil jen do té doby než bzučák začal bzučet a display vypisoval tím pádem jen 0 nebo strašně nízké hodnoty. Pár dní jsem přemýšlel, čím by to mohlo byt a prozřel jsem.

**3.2.1 Problém s napájením**

Do projektu jsem šel totiž s tím, že budu napájet všechno pomocí Arduino konektoru, který má chabých 5V. Proč chabých? No... protože 5V nestačí napájet naráz LCD display, ultrazvukový senzor, bzučák a ještě ledky. Proto jsem musel jít do opatření a poradit se s panem učitelem Miekischem. Který mi řekl ať vemu baterku a to nejlépe 9V. To jsem také udělal a z nefunkčního projektu se stal najdenou kouzlem funkční. Nevyřešilo to všechny problémy co jsem měl, ale vyřešilo to ten základní za což panu Miekischovi jěště jednou děkuji.

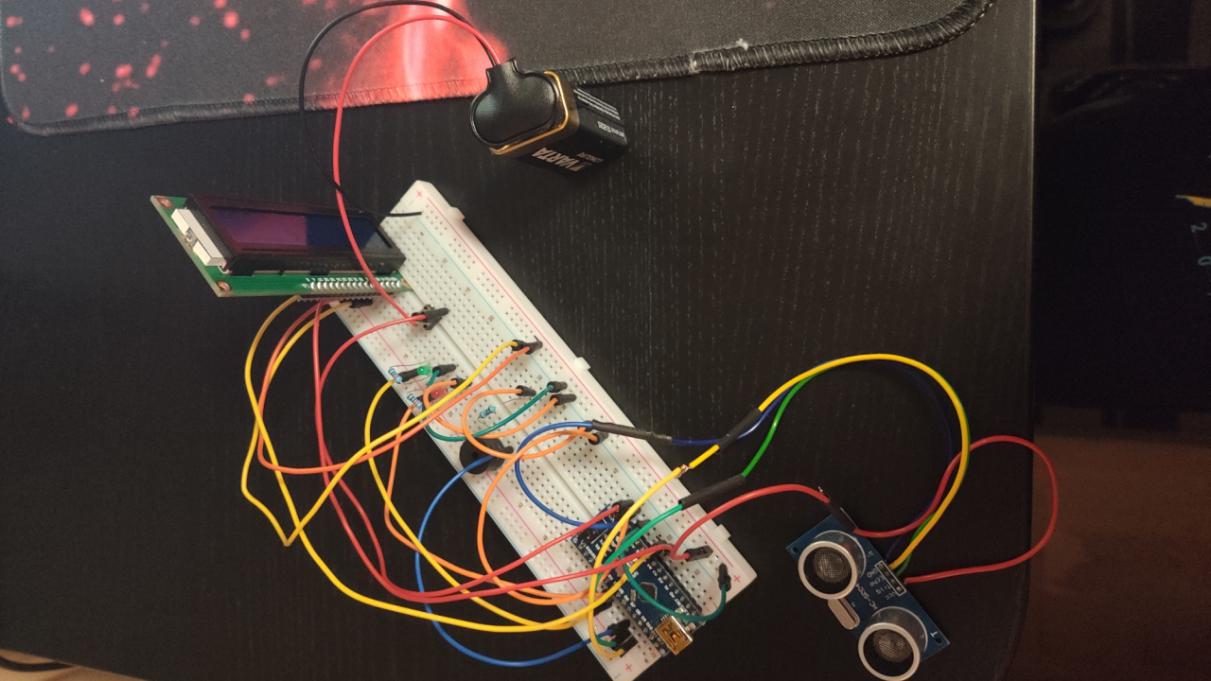
**3.2.2 Problémy s knihovnami**

Problémy se neobjevovali jen v hardwarové části, ale i v softwarové. Několikrát jsem totiž musel měnit knihovny. Některé z toho důvodu že neobsahovali to co jsem potřeboval a některé zase nebyly na tento projekt stavěné. Takže jsem velkou část smazal a nechal jen jednu jedinou a to LiquidCrystal\_I2C na zprovoznění displaye.

# Výsledky řešení

**4.1 Finální Podoba Hardwaru**

Finální podoba hardwaru je duležitá na udělání dobrého dojmu u učitele a samozřejmě zákazníka, který by si chtěl výsledný produkt třeba zakoupit. Já bohužel dobrý dojem obravdu udělat nemohu, protože můj celý projekt je na nepájivém poli a složí fakt jen na ukázku. Ale i přes to jsem s hardwarovou částí spokojen, protože zapojení celého projektu jsem řešil samostatně.



*Obrázek č. 11 Závěrečná podoba hardwaru*

**4.2 Finální podoba kódu**

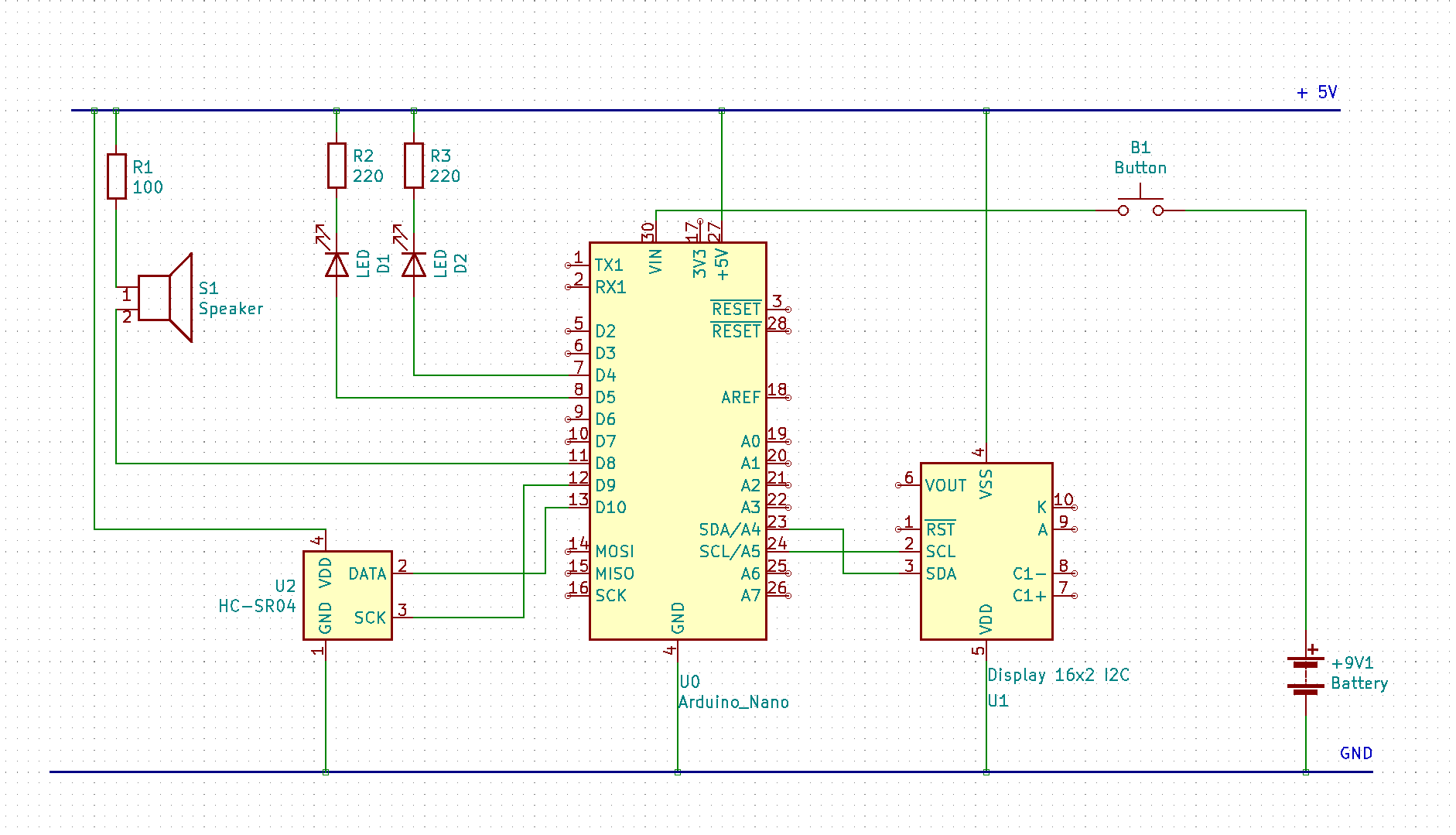
S finální podobou kódu jsem taktéž spokojený. Až na pár nedostatků a mojeho hodně jednoduchého uvažování tomu není co moc vytknout. Kód jsem se snažil správně oddělovat a dávat důležité části do funkcí.



*Obrázek č. 12 Závěrečná podoba kódu*

**4.3 Schéma**

Schéma je na přání pana učitele Grussmanna udělané v KiCadu. Jsem rád, že jsem si mohl vyzkoušet nějaké schéma vytvořit sám, protože to byla pro mě nová zkušenost jak s KiCadem tak se schématy obecně. Trvalo mi než jsem se do toho dostal, ale myslím si, že kdybych takových projektů udělal v budoucnu více, šlo by mi to časem od ruky.



*Obrázek č. 13 Schéma v KiCadu*

# **Závěr**

Projekt bych označil za dokončený a hlavně fukční. Bylo zde hodně věcí nad, kterými jsem se pozastavil a nad, kterými budu přemýšlet jěště dlouho. Nasbíral jsem nové zkušenosti, co se týče psaní kódu v Platformiu IDE, vytváření schémat v KiCadu, spojování hardwaru a softwaru, pracování s Githubem a v neposlední řadě psaní dokumentace.

Myslím, že tímto si musí projít každý maturant, ať už slabší nebo jedničkař. Já se naučil jedno. Když chcete něco vytvořit, musíte do toho vrazit spoustu svého volného i nevolného času.

Do projektu je spoustu co přidávat. Napadlo mě i ukládat hodnotu do EEPROM od Arduina, ale už jsem neměl tolik času na realizaci, tak jsem radši ustoupil než byjsem se brodil v dlaších problémech.

Další vylepšení by byl větší display, kde by se dalo udělat LiquidMenu a ovládal by se pomocí tlačítek nebo malé klávesnice.

A na konec by možná mohlo být praktické dát celý projekt do pěkné krabičky, aby mohl využít svůj plný potenciál.

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] BOHMAN, Ludvík. Zákon o pojistné smlouvě. Praha: Linde Praha a. s., 2004. 381 s. ISBN80-7201-504-4

[2] DUCHÁČKOVÁ, Eva. Principy pojištění a pojišťovnictví. 3. aktualizované vydání. Praha: Ekopress 2009. 224 s. ISBN 978-80-86929-51-4

[3] KUBALA, Petr. Planetární dvojcata - Věda a technika (Český rozhlas) [online].   
Č. 2000-2008, poslední revize 19. 3. 2008 [cit. 2008-03-20].  
<http://www.rozhlas.cz/veda/vesmir/\_zprava/435849>.

[4] KULDOVÁ, O., FLEISCHMANNOVÁ, E. Metodická příručka k technice administrativy a obchodní korespondence. 1.vyd. Praha: Fortuna 1998. 111 s.   
ISBN 80-7168-574-7. Kapitola 6, Metody nácviku psaní hmatovou metodou,   
s. 28-29.

[5] VLACH, J. JE Temelín a zásobování teplem. Energetika, 2001, roč. 51, č. 3, s. 84 -85. ISSN 0375-8842.

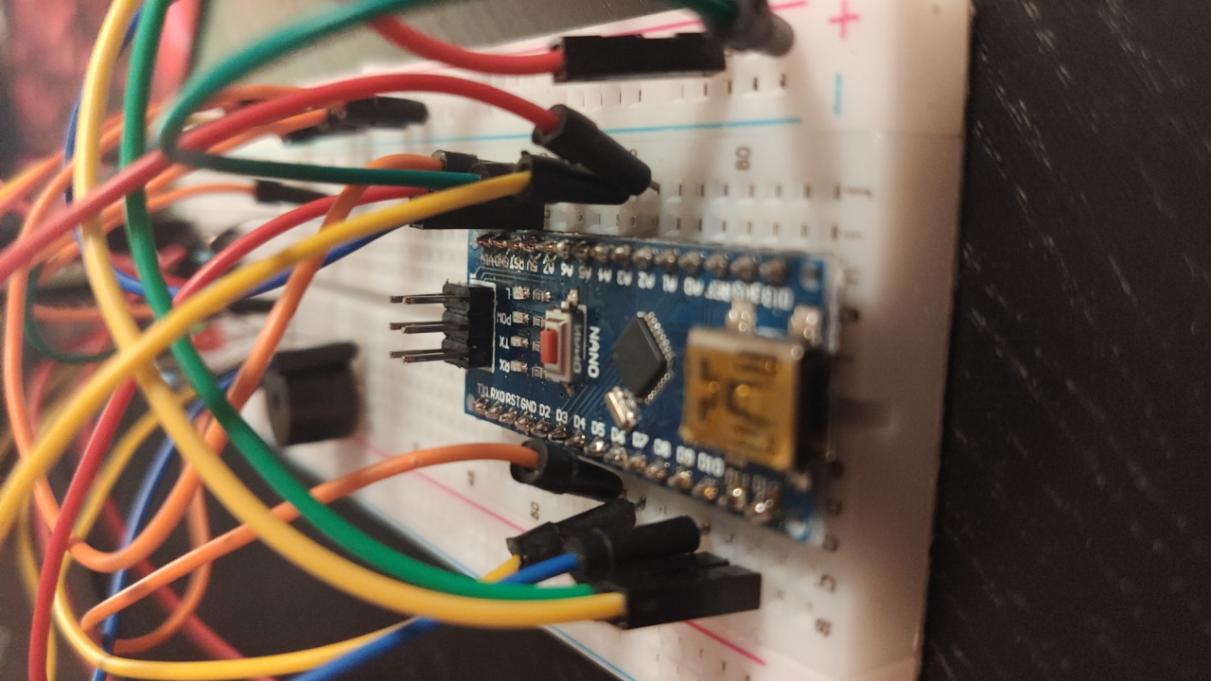
**Chtěl bych se zeptat jak mám tuhle část udělat?**

**Vím, že jsme ve škole něco podobného dělali, ale ani za boha si nevzpomínam jak.**

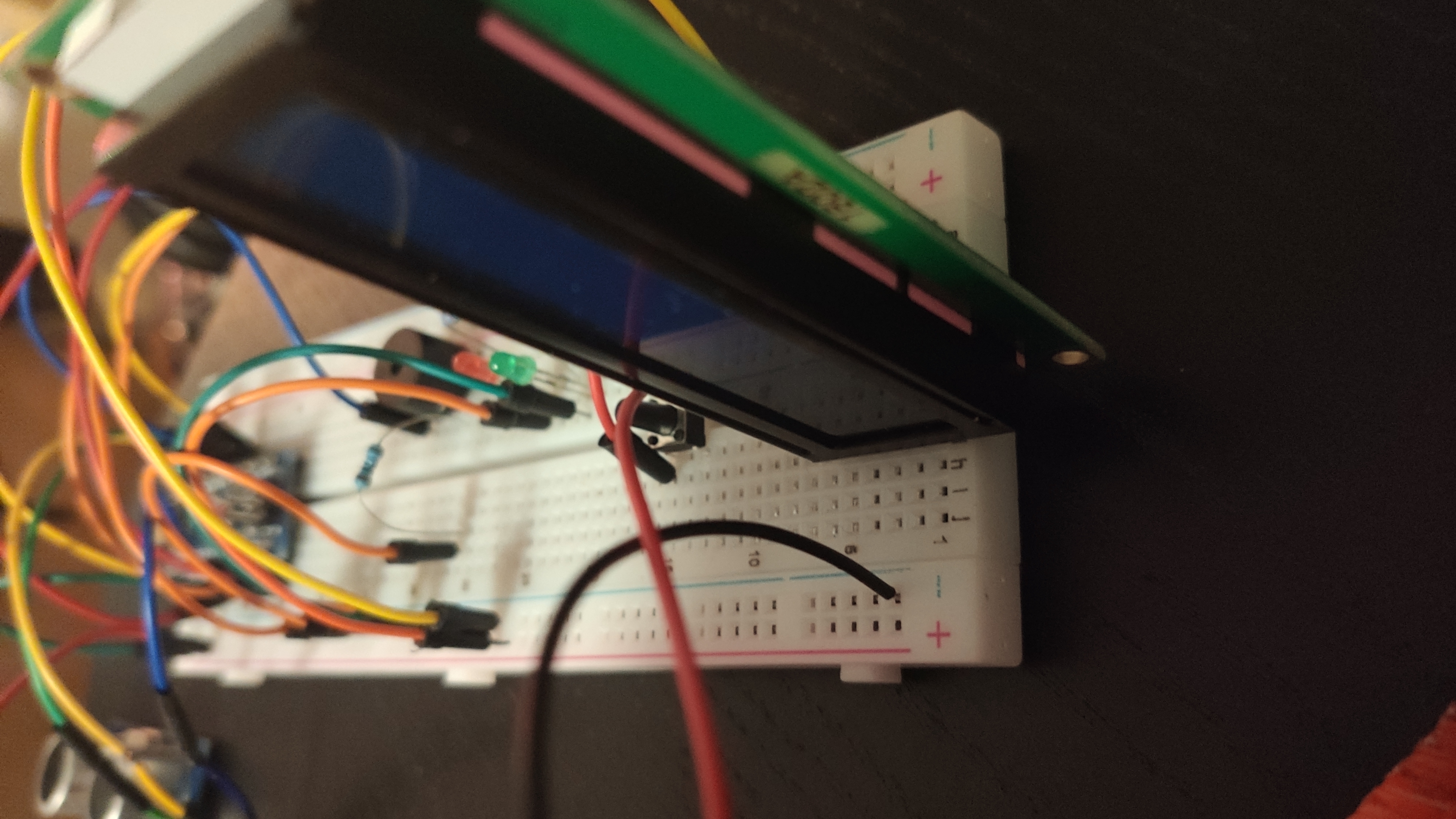
Seznam příloh

č. 1 Fotodokumentace projektu

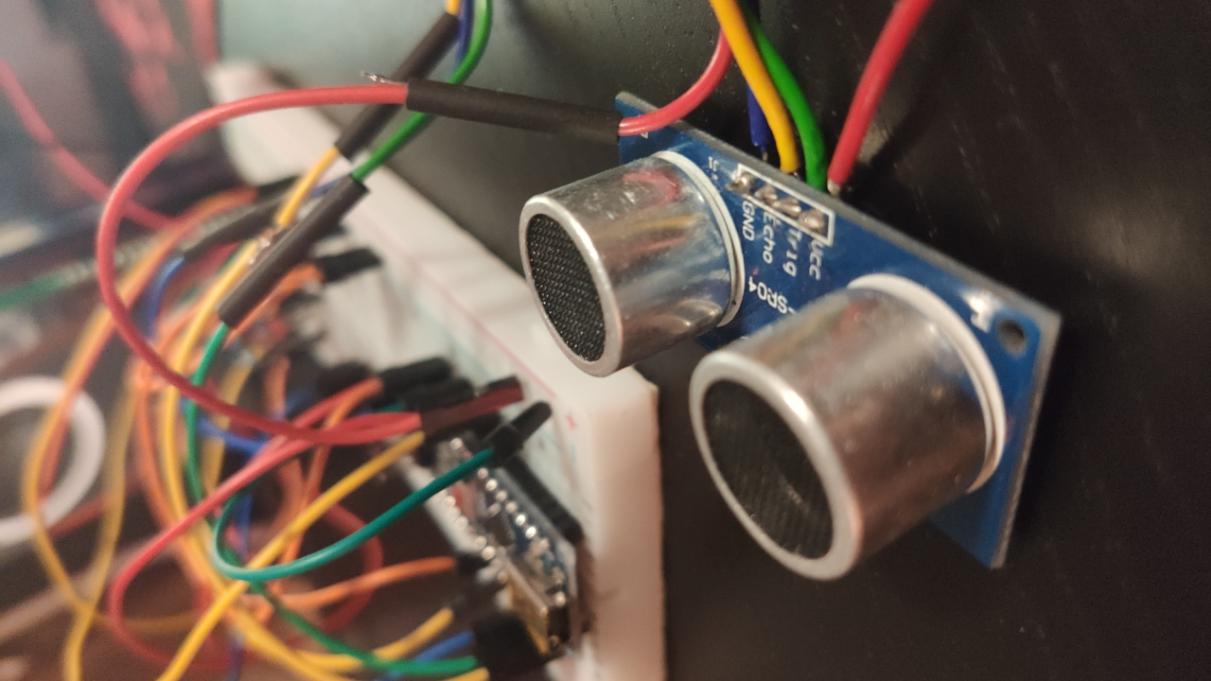
**Příloha č. 1: Fotodokumentace projektu**



*Obrázek arduina\_nano*



*Obrázek LCD displaye*



*Obrázek bzučáku*



*Obrázek napájení 9V baterkou*