



SPŠT

Střední průmyslová škola Třebíč

Maturitní práce

DÁLKOVĚ OVLÁDANÁ ZÁSUVKA

Profilová část maturitní zkoušky

Studijní obor: Informační technologie

Třída: ITB4

Školní rok: 2023/2024 David Suchý

Zadání práce



SPŠT

Střední průmyslová škola Třebíč
Manželů Curieových 734, 674 01 Třebíč

Zadání ročníkové práce

Obor studia: **18-20-M/01 Informační technologie**

Celé jméno studenta:	David Suchý	Školní rok:	2023/2024
Třída:	ITB4		
Číslo tématu:	30		
Název tématu:	Dálkově ovládaná zásuvka		
Rozsah práce:	15 - 25 stránek textu		

Specifické úkoly, které tato práce řeší:

Navrhněte a zrealizujte dálkově ovládanou síťovou zásuvku pro připojení spotřebiče 230V max. příkon 1000W. Ovládání bude přes komunikační standard Bluetooth pomocí mobilního telefonu. Napište mobilní aplikaci (pro platformu Android), která umožní okamžité zapnutí a vypnutí zásuvky a dále zapnutí na definovanou dobu. Pro realizaci HW části využijte platformu Arduino, pro SW část vývojové prostředí Android Studio.

Termín odevzdání:	27. března 2024, 23.00
Vedoucí projektu:	Ing. Ladislav Havlát
Oponent:	Ing. Marek Veselý
Schválil:	Ing. Petra Hrbáčková, ředitelka školy

ABSTRAKT

Cílem této maturitní práce je vývoj a realizace projektu bezdrátově ovladatelné zásuvky s využitím desky Arduino a jeho programovacího jazyka. Tento projekt umožňuje uživateli komunikaci skrz rozhraní Bluetooth a mobilní android aplikace. Teoretická část se zaměřuje na popis a informace o využitých technologiích, jejich funkcích a fyzickém hardwaru. Praktická část obsahuje popis samotného vývoje projektu a souhrn využití jednotlivých technologií.

KLÍČOVÁ SLOVA

maturitní práce, arduino, displej, relé, Bluetooth, zásuvka, bezdrátově

ABSTRACT

The aim of this graduation thesis is to develop and implement a wirelessly controllable socket project using the Arduino board and its programming language. This project shows the user the possibilities of communication through Bluetooth and mobile android applications. In the theoretical part there is a description and information about the technologies used, their functions and physical hardware. The practical part contains a description of the actual development of the project and an introduction to the use of each technology.

KEYWORDS

graduation thesis, arduino, display, relay, Bluetooth, socket, wireless

PODĚKOVÁNÍ

Děkuji Ing. Ladislavu Havlátovi za cenné připomínky a rady, které mi poskytl při vypracování maturitní práce.

V Třebíči dne 27. března 2024

podpis autora

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil.

V Třebíči dne 27. března 2024

podpis autora

Obsah

Úvod.....	7
1 Teoretická část.....	8
1.1 Arduino	8
1.1.1 Historie.....	8
1.1.2 Mise.....	8
1.1.3 Deska.....	8
1.1.4 Specifikace	9
1.1.5 Sériová komunikace	9
1.2 Jazyk.....	10
1.2.1 Wiring	10
1.2.2 Analogové a digitální vstupy a výstupy	10
1.2.3 Struktura.....	10
1.2.4 Proměnné	11
1.2.5 Datové typy	11
1.2.6 Knihovny.....	12
1.2.7 Cloud.....	12
1.3 LCD displej.....	12
1.3.1 Funkčnost	12
1.4 Relé	13
1.4.1 Princip	13
1.4.2 Dělení	13
1.5 Bluetooth modul	14
1.5.1 Piny	14
1.6 LED	14
2 Praktická část	16
2.1 Zapojení	16
2.2 Displej.....	16
2.3 Zapojení LCD.....	16
2.3.1 Obsluha	17
2.3.2 Zobrazení času	18
2.3.3 LED	19
2.4 Relé	19

2.4.1	Zapojení relé.....	20
2.4.2	Řízení stavu.....	21
2.4.3	Reset.....	22
2.5	Bluetooth modul HC-05.....	22
2.5.1	Zapojení modulu	22
2.5.2	Komunikace s mobilním zařízením	23
2.6	Verzování a zálohy	24
2.7	Problémy	26
2.7.1	Zapojení.....	26
2.7.2	Inspirace	26
2.7.3	Program Arduino.....	26
2.7.4	Android studio.....	27
2.7.5	Github.....	27
2.7.6	Testování	27
2.8	Ochrana zapojení	27
	Závěr.....	29
	Seznam použitých zdrojů	30
	Seznam použitých symbolů a zkratek	32
	Seznam obrázků	33

Úvod

Při rozhodování o tématu mé ročníkové práce jsem vycházel ze své zkušenosti s vánočním osvětlením na našem domě. Jako dobrý nápad mi přišlo zrealizovat zásuvku na dálkové ovládání pomocí mobilního telefonu skrz rozhraní Bluetooth. Dalším důvodem bylo také to, abych rozšířil své znalosti a schopnosti v rámci mikroprocesorové techniky a práci s Arduinem.

Cílem práce je realizace bezdrátové zásuvky se základními funkcemi. Jedna z hlavních funkcí je zapínání a vypínání relátka. Jako další důležitá funkce je výpis stavu a času, jak dlouho je relé v zapnutém stavu. Zásuvka je spojena skrz Bluetooth s mobilní aplikací android. Uživatel v reálném čase pošle informaci přes aplikaci, zásuvka ji zpracuje a vyhodnotí adekvátní reakci.

Dále jsou pro uživatelské pohodlí výhodné již přednastavené hodnoty pomocí tlačítek v android aplikaci. Samozřejmě je stále možné zadat hodnoty ručně.

Jako poslední je využito prostředí Arduino, které zajišťuje, že program bude běžet bez chyb v reálném čase. Tento systém je založen na jazyku C, který je především určen pro mikroprocesorové programy a využívá uživatelské knihovny a knihovny zabudované v aplikaci přímo od vývojářů Arduina.

V teoretické části je pojednáváno o technologiích, které byly využity při vývoji programu a jejich podrobném popsání. Naopak v praktické části je popsáno, jak byly tyto technologie využity při programování a zapojení komponentů.

1 Teoretická část

Tato část práce se zaměřuje na použité technologie a programovací jazyky potřebných k vývoji projektu. Jsou zde podrobně popsány at' už jejich funkce, výhody nebo nevýhody. Také zde nalezneme obecné schéma zapojení jednotlivých komponent do desky Arduino.

1.1 Arduino

Arduino je italská firma fungující už od roku 2005, jakožto nástroj pro studenty v Itálii. Těmto studentům chtěla firma umožnit levnou a spolehlivou variantu, kde můžou využít právě mikroprocesorové prostředí s již předprogramovanými funkcemi. **Chyba! Nenalezen zdroj odkazů.**

1.1.1 Historie

Nyní společnost Arduino navrhuje, vyrábí, podporuje elektronická zařízení a software a umožňuje lidem na celém světě snadný přístup k pokročilým technologiím, které komunikují s reálným světem. Jejich produkty jsou přímočaré, jednoduché a výkonné a jsou připraveny pokrýt potřeby všech uživatelů od studentů přes tvůrce až po profesionální vývojáře. [1]

1.1.2 Mise

Posláním společnosti Arduino je umožnit každému, aby si zlepšil svůj život prostřednictvím dostupné elektroniky a digitálních technologií. Kdysi existovala bariéra mezi světem elektroniky, designu a programování a zbytkem světa. Arduino tuto bariéru prolomilo. V průběhu let se jejich produkty staly mozkiem tisíců projektů, od předmětů denní potřeby až po složité vědecké přístroje. Kolem této open-source platformy se shromáždila celosvětová komunita složená ze studentů, amatérů, umělců, programátorů a profesionálů, jejichž příspěvky tvoří neuvěřitelné množství dostupných znalostí. [1]

1.1.3 Deska

V tomto projektu je využita Arduino deska s názvem – „Laskakit UNO R3 – ATmega328P“ což je precizní klon originální desky. Tato deska s vlastním programovacím jazykem a procesorem Arduino zajišťuje jednoduché rozhraní

s velkou škálou knihoven a možností úpravy. Obsahuje například obsluhu LCD displeje nebo Bluetooth modulu. Arduino je open-sourceový hardware a software, který umožňuje studentům nebo začínajícím programátorům mikroprocesorů vyzkoušet, jak vlastně mikroprocesory fungují. Taktéž je to jedna z nejrozšířenějších



Obrázek 1 Arduino Uno [2]

desek na trhu, která je ideální pro začátečníky i profesionály. [2]

Na obrázku je možné vidět reálnou fotku desky Arduino Uno 3.

1.1.4 Specifikace

Na desce se nachází mikroprocesor/mikrokontrolér ATmega328P s frekvencí až 20MHz. Také obsahuje paměti FLASH s 32KB, EEPROM s 1KB a SRAM 2KB. Na desce je možné nalézt 14x digitální I/O, 1x 10-bit ADC 6 kanálů, 6 PWM kanálů, 2x 8 bitový časovač a 1x 16 bitový časovač oba s programovatelným watchdog časovačem s odděleným integrovaným oscilátorem. Vstupní napětí desky UNO R3 je 6 až 12 voltů, poté však zdroje na desce vyvíjí napětí 5 nebo 3,3 volty. Lze ji napájet pomocí USB konektoru, napájecích konektorů nebo napájecích pinů. [2]

1.1.5 Sériová komunikace

Aby Arduino správně spolupracovalo s počítačem, musí být splněno několik základních požadavků: Potřebujete počítač s portem USB a kabel USB pro připojení desky Arduino k počítači. Potřebujete také převodník, který je obvykle součástí čipu (ATmega32u4...). Lze jej však k desce Arduino připájet i samostatně. Při

programování můžete potřebovat také externí převodník, který připojíte. A nakonec potřebujete mozek, který bude přijímat instrukce z převodníku. Díky této komunikaci může komunikovat s Arduinem i za běhu programu. Může také číst hodnoty ze senzorů a pracovat s nimi. Arduino je také možné ovládat pomocí jednoduchých textových příkazů. Sériovou komunikaci lze využít i pro jiné aplikace než programování. [4]

1.2 Jazyk

Jazyk Arduina se především odvíjí od knihoven, které uživatel používá. Přesto však má nějaké základní příkazy, které jsou neměnné. Arduino tudíž dává začátečníkům jednoduché prostředí v podobě jejich aplikace a snadné programování pomocí knihoven a několika stálých příkazů. Na konec se program přeloží do programovacího jazyka C a pošle pomocí USB do již zmíněné desky. [1]

1.2.1 Wiring

Programovací jazyk Wiring je určený k obsluze mikroprocesoru bez znalostí hardwaru. V této době je neznámější v platformách Arduino, kde se nachází jako framework C++. Byl vytvořen pro podobnou sadu jako Arduino a vychází z open-source projektu - Processing. [3]

1.2.2 Analogové a digitální vstupy a výstupy

Analogové signály nejsou omezeny na HIGH a LOW jako digitální signály. Proto se pro zpracování analogových hodnot využívá výbava Arduina - analogový digitální převodník. Tento převodník má za úkol převést analogové napětí na digitální hodnoty. Používá se funkce `analogRead(pin)`. Převádí tedy hodnotu analogového vstupu na pinu a vrací digitální hodnotu v rozsahu 0 – 1023. Výchozí napětí je 5V nebo 3,3V. **Chyba! Nenašle zdroj odkazů.**

1.2.3 Struktura

Projekt v Arduino IDE se nazývá sketch. Počáteční kód obsahuje dvě hlavní části. První část se nazývá „`setup()`“, což je část kódu, která se provede pouze jednou při každém spuštění programu. Často obsahuje počáteční nastavení pinů a různé inicializace proměnných a inicializace funkcí knihoven. Druhou částí je „`loop()`“, v této části programu se kód provádí neustále do ukončení. Nachází se v ní obsluha společně s logikou. Pod touto částí je možné definovat vlastní funkce neboli přerušení.

Jediná možnost pro dočasné opuštění smyčky `loop()` je přerušení právě vyvolané funkce. Po opuštění se provedou příkazy ve funkci a vrátí se zpět do hlavní smyčky.
Chyba! Nenalezen zdroj odkazů.

1.2.4 Proměnné

Proměnná je definována jako vyhrazené místo v paměti, díky kterému se uchovávají data. Každá proměnná má vlastní název, datový typ i hodnotu. Pokud jsou všechny veličiny stejné, může to vyústit k problémovým situacím. Proměnné je potřeba deklarovat a inicializovat. Je možné zkombinovat deklaraci s inicializací a provést je zároveň. Pakliže je proměnná již deklarovaná, ale nemá inicializovanou hodnotu, je možné s ní nadále pracovat. Pokud však hodnota přesáhne maximální velikost datového typu, dojde k přetečení a program se může ukončit. **Chyba! Nenalezen zdroj odkazů.**[6]

1.2.5 Datové typy

Datový typ udává informaci, jaké hodnoty můžeme do proměnných ukládat. Například se může jednat o hodnoty `true` nebo `false`, čísla, znaky či znakové řetězce. Mezi základní datové typy řadíme typy číselné. Do nich se řadí `byte`. `Byte` je tedy číselná proměnná o velikosti 8 bitů a rozsahu 2^8 či 0 až 255. Uchovávají se v ní celá čísla. Další typ je `Integer`, ten je v programech psán zkráceně, jakožto `int`. Slouží k ukládání větších celých čísel. Jeho rozsah se počítá od záporných čísel až po kladné. Číslo 0 je přesně v polovině celého rozsahu. Celý rozsah u desek Arduino činí 16 bitů, což znamená, že používá čísla od -32 768 až po číslo 32 767. Dalším datovým typem, který obsahuje pouze celá čísla je typ `long`. Ten má již rozsah 32 bitů, tudíž přibližně od -2 miliard až po 2 miliardy. Dalším číselným datovým typem, který již dokáže uchovat desetinná čísla je `float`. Je také 32 bitový. Do skupiny logických datových typů patří typ `bool`, který má pouze hodnoty `true` – pravda, nebo `false` – nepravda.

Poslední skupina jsou znakové datové typy. Jeden z nich se nazývá `char`. Tento datový typ slouží k uchování pouze jednoho znaku. Znak je uložen jako číselná hodnota v ASCII tabulce znaků. Řetězec těchto znaků se nazývá `String`. Může obsahovat více než 1 znak. [6]

1.2.6 Knihovny

Knihovny v Arduino IDE softwaru lze nainstalovat přímo z aplikace, ale je možné si vlastní knihovny vytvořit a poté importovat. Knihovny slouží pro získání více funkcí a možností práce s hardwarem pomocí kódu. Je to jednoduše kolekce funkcí, které lze přidat do aplikace a volat je podle potřeby stejně jako jiné funkce v aplikaci. Přesněji řečeno, knihovna může obsahovat jakýkoli objekt, nejen funkce, ale většina knihoven obsahuje pouze funkce. Mezi nejznámější knihovny patří především knihovny Wire společně s knihovnou LiquidCrystal, které jsou použity i v tomto projektu. [7]

1.2.7 Cloud

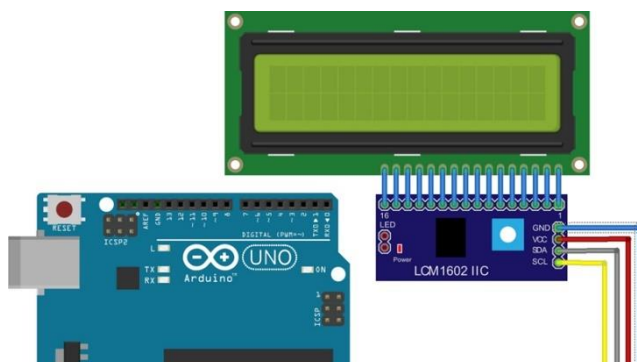
Arduino nabízí možnost cloud, což je platforma, která umožňuje uživateli spravovat, vytvářet a testovat IoT projekty online a jednoduše. Emuluje již zmíněnou aplikaci Arduino IDE ve webovém rozhraní a dodává hardwarové prostředky bez potřeby uživatele vlastnit jakýkoliv hardware.[1]

1.3 LCD displej

Základní LCD displej, který je použit má 2 řádky a na každém řádku je 16 znaků. Displej je zeleně podsvícený s tmavými písmeny.

1.3.1 Funkčnost

LCD technologie využívá vlastnosti tekutých krystalů (liquid crystal) a závislosti na změnách elektrostatického pole. LCD displej je tvořen z několika vrstev. Molekulová vrstva je vložena mezi 2 zarovnávací, které mají drobné drážky vyrovnávající



Obrázek 2 Obecné schéma zapojení LCD displeje

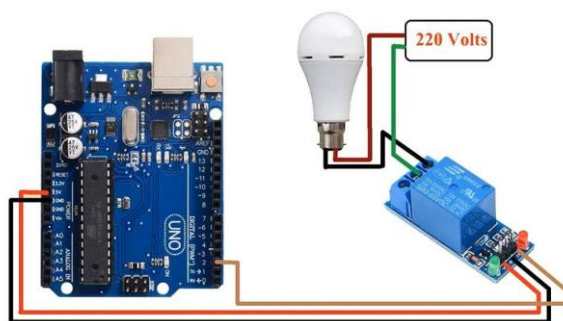
Chyba! Nenalezen zdroj odkazů.

molekuly. Pro vytvoření barevného obrazu je možné aplikovat barevné filtry na vrchol LCD buněk. [9]

Zde je možné vidět obecné zapojení LCD displeje do desky Arduino UNO. Lze si povšimnout barevného rozdělení propojovacích drátků. Například VCC jakožto červený a GND jako modrý.

1.4 Relé

Pětivoltové relé je deska, pomocí které je uživatel schopen ovládat různé spotřebiče, ale také i další zařízení s velkými proudy. Je možné jej ovládat přímo



Obrázek 3 Obecné zapojení relé [11]

mikrokontrolérem nebo i čidlem na 5 voltů. Relé je opticky odděleno a jeho pracovní proud je 185 miliampérů. [12]

Na tomto obrázku můžeme pozorovat obecné zapojení relé v desce Arduino UNO. Poté i zapojení žárovky na již zmíněné relé.

1.4.1 Princip

Při přivedení elektrického napětí na cívku relé vznikne magnetické pole a nastane přetížení kotvy s pohyblivými kontakty. Pružné kontakty se spojí s pevnými a přeruší tak stávající spojení. [14]

1.4.2 Dělení

Relé se dělí podle buzení a jmenovitého napájení cívky na: [14]

1. AC/DC – Střídavé i stejnosměrné napětí
2. DC – Pouze jednosměrné napětí
3. AC – Pouze střídavé napětí

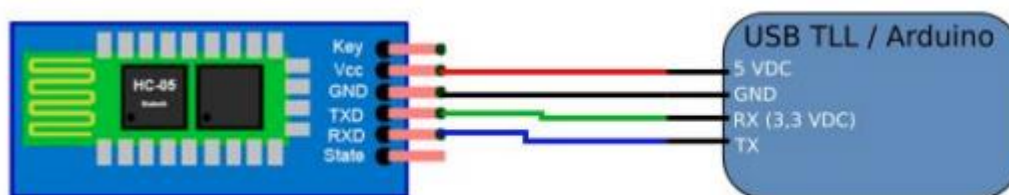
V tomto přehledu je možné vidět jmenovité rozdělení napájení cívky pomocí zkratk a popisů u nich.

1.5 Bluetooth modul

Bluetooth modul použitý v tomto projektu se nazývá HC-05. Je to univerzální jednoduše použitelný modul pro přenos dat, který podporuje protokol Bluetooth V2.0. Má sériové rozhraní a dosah až 10 metrů. Ukazatel statusu ukazuje LED, která svítí modře nebo červeně. Může fungovat jako Master i Slave. Rychlost komunikace s mobilním telefonem je přibližně 9600bps v režimu 8 N 1. Typicky se využívá pro dálkové ovládání pomocí mobilního telefonu nebo Arduino jednotky. Ovládat je možné například LED pásek, odemykat auto, rozsvěcovat světla nebo zásuvky. [15]

1.5.1 Piny

Na desce modulu HC-05 lze nalézt 6 pinů. Pin STATE je výstupní vstup, který slouží pro kontrolu správného spárování. Piny RXD a TXD jsou datové vstupy.



Obrázek 4 Připojení k USB TTL převodníku [17]

GND a VCC jsou piny napájení, GND připojený k zemi (0 voltů) a VCC do napájecího napětí 5 voltů. Poté pin EN, jehož funkce je závislá na úrovni napětí. Pokud je úroveň „low“, tak je modul nastaven v provozním režimu, naopak při úrovni „high“ je modul v režimu programovacím. [16]

1.6 LED

LED diody jsou specializované polovodičové součástky známé jako "světelné diody"; jako zdroj světla se běžně používají bílé LED diody, které byly do praxe zavedeny v roce 1993 vývojem modrých LED diod s vysokým jasnem na bázi nitridu galia. Tyto LED fungují na základě kombinace polovodiče typu P (vysoká koncentrace děr) s

polovodičem typu N (vysoká koncentrace elektronů). Po přivedení dostatečného napětí dochází k rekombinaci elektronů a děr při přechodu P-N a uvolnění energie ve formě světla. Ve srovnání s běžnými světelnými zdroji, které přeměňují elektrickou energii na teplo a poté na světlo, LED diody přeměňují elektrickou energii přímo na světlo, a poskytují tak účinné vyzařování světla s minimálními ztrátami energie. Existují dva hlavní typy LED diod: lampový typ (s vývody) a čipový typ (pro povrchovou montáž). Barva světla vyzařovaného LED diodami závisí na použitém materiálu a barvu lze přizpůsobit podle specifických vlastností vlnové délky požadovaných pro konkrétní aplikaci, jako jsou dopravní světla nebo automobilové lampy. [18]

2 Praktická část

V této části je podrobně popsán vývoj projektu, postup, zpracování použitých technologií, problémy, které vznikly při vývoji a popis programu samotného. Část obsahuje kapitoly zaměřené na dané komponenty, jejich důmyslné zapojení v programu a interpretaci.

2.1 Zapojení

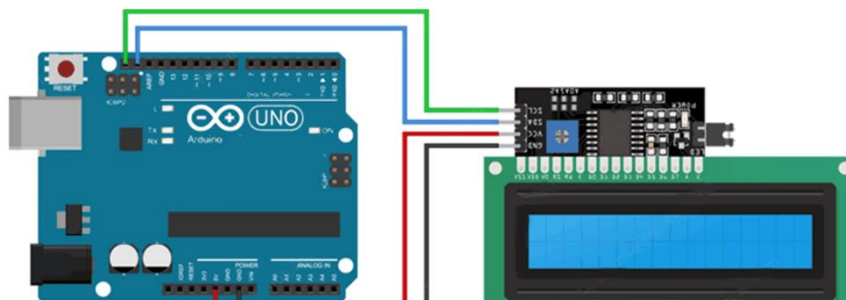
Každá komponenta je zapojena do desky Arduino Uno 3. Moduly jsou zapojeny samostatně a specificky pomocí vodičů a příslušných konektorů. Každý vodič je zapojen na pin modulu Arduino. Piny modulu se dělí do specifických částí, například do analogových či digitálních. Je také možné piny dělit na 5 voltové, 3,3 voltové a zem. Všechny zapojené komponenty jsou v připojeny do příslušných pinů 5V nebo 3,3V, podle jejich maximálního napětí. Některé z modulů potřebují specifický pin pro jejich ovládání, a proto jim byly potřebné piny přiděleny.

2.2 Displej

Displej je komponenta, které si uživatel všimne jako první. Displeje jsou obecně používané pro zobrazování informací. V tomto projektu to není jinak. Po spuštění programu displej ukáže stav relé. Informuje, zda je zapnuto nebo vypnuto. Poté po připojení a spuštění časově omezeného zapnutí, vypisuje informaci o tom, jak dlouho program ještě poběží než dojde k vypnutí relátka.

2.3 Zapojení LCD

V tomto projektu je LCD displej zapojený pomocí sběrnice I2C. Je to proto, aby došlo k zmenšení počtu vodičů potřebných k zapojení displeje.



Obrázek 5 Schéma zapojení LCD displeje v projektu

Na tomto obrázku je možné vidět přímé zapojení v tomto projektu jako schéma, pomocí Arduino Circuit Designeru.

2.3.1 Obsluha

Obsluha displeje je zajištěna pomocí knihovny LiquidCrystal_I2C.h. Díky této knihovně a přiřazení příslušných hodnot podle zapojení, je možné displej jednoduchými funkcemi obsluhovat.

```
#include <LiquidCrystal_I2C.h>
#include <SimpleRelay.h>
#include <SoftwareSerial.h>

#define RX 11
#define TX 10
SoftwareSerial bluetooth(TX, RX);
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

Obrázek 6 Knihovny a přiřazení informací v kódu

Základními funkcemi, které je potřeba provést vždy při spuštění je „lcd.init()“ společně s „lcd.backlight()“. Pomocí těchto funkcí, se předá knihovně informace v rámci údajů nastavených v kódu a zapne podsvícení, které je potřebné k zobrazení

```
lcd.init();
lcd.backlight();

lcd.setCursor(4, 0);

lcd.printstr("Vypnuto");
```

Obrázek 7 Základní příkazy LCD

informací na displeji. Poté se nastaví pozice, kde se bude zapisovat a může vypsát první informaci. V tomto případě – Vypnuto.

Samozřejmě je možné pozici zápisu přenastavit. Dokonce i celý displej vymazat a to pomocí „`lcd.clear()`“.

2.3.2 Zobrazení času

Zobrazení času pomocí displeje je v projektu řešeno funkcí „`milis()`“ a definovaného intervalu. Nastaví se hodnota sekund a poté pomocí funkce „`Casovac`“, která probíhá zároveň jako hlavní kód, počítá 1 sekundu. V této funkci nalezneme poměrně jednoduchý kód, který se skládá z časování a výpisu zbývajcího času na displej.

```
void Casovac()
{
    if (millis() > time_for_action) {
        time_for_action = millis() + (unsigned long)INTERVAL;
        led = !led;
        digitalWrite(13, led);

        casovac--;

        if(casovac==9);
        {
            lcd.clear();
            lcd.setCursor(4, 0);
            lcd.printstr("Zapnuto");
            lcd.setCursor(7, 1);
        }
        lcd.print(casovac);
    }
}
```

Obrázek 8 Kód funkce Časovač

Zde je možné pozorovat podmínku, která kontroluje, zda čas pro akci je menší než počet milisekund od startu programu. Pokud tedy je menší, provede se kód, který nastaví čas pro akci právě milisekundy od startu programu + náš definovaný interval, což je v tomto případě 1000 milisekund. Poté zapne nebo vypne diodu a zapíše její stav na příslušný výstup. Odečte se hodnota 1 od proměnné časovač. Zkontroluje se, zda hodnota na proměnné časovač není 9, protože pokud ano, tak se vykoná kód, který vymaže displej a znovu zapíše údaje. Tato kontrola je zde z důvodu přehlednosti při přeskokování z čísla 10 na číslo 9, aby nebyla viditelná zbývajcí 0, kterou již jednociferná čísla nepotřebují. Hned poté se vypíše zbývajcí čas.

2.3.3 LED

Součástí znázornění doby výkonu relé je blikající dioda, která běží ve stejném intervalu jako ukazatel času na displeji. Jak již bylo popsáno v kapitole Zobrazení času. Dále je potřeba definovat výstupový pin a interval času. Dioda je ovládaná pomocí nastavení pinu a následném definování proměnné led, díky které program zná hodnotu vypnuto či zapnuto. Díky těmto informacím je možné diodu zapínat nebo vypínat a to pomocí příkazu `digitalWrite(13,led)`, který nastaví hodnotu proměnné led na pin 13. Je možno tento příkaz provést pouze v případě, že je pin 13 nastavený na mód OUTPUT.

```
#define INTERVAL 1000
byte led;
unsigned long time_for_action;
pinMode(13, OUTPUT);
```

Obrázek 9 Definování intervalu a nastavení pinu

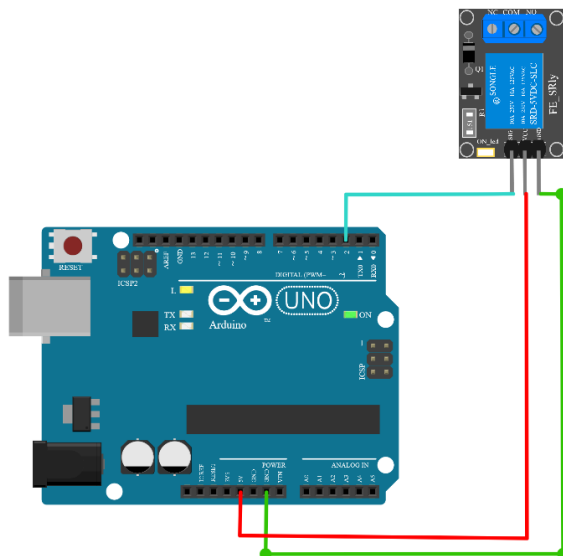
Zde je viditelné definování intervalu na 1000ms a módu pinu 13 na output, což je výstupový režim pro již zmiňovanou diodu.

2.4 Relé

Relé je komponenta často využívaná při spínání určitých elektrických spotřebičů. Zde zastupuje pozici zásuvky a je možné jej nastavit na pozici 1 = zapnuto, nebo na pozici 2 = vypnuto. V tomto projektu je stav relátka důležitý především pro výpis informací na displej a pro správný chod projektu. Pokud by nastala chyba ve stavu relé, tak by to znamenalo, že by celý program katastroficky selhal.

2.4.1 Zapojení relé

Relé je zapojeno standartním a obecným způsobem a to tak, že je zapojená zem do země na desce, proud 5V do proudu 5V a řízení stavu na pin 2.



Obrázek 10 Schéma zapojení relé do desky Arduino

Lze zde vidět výše popsané zapojení v názorném schématu vytvořeného pomocí Arduino Circuit Designeru.

2.4.2 Řízení stavu

Řízení stavu relé je umožněno knihovnou SimpleRelay.h, která obsahuje funkce k obsluze relátka. Hlavními funkcemi této knihovny jsou „relay.off()“ a „relay.on()“, tyto funkce zajišťují zapínání a vypínání relé. V programu je také využita funkce „relay.isRelayOn()“, díky ní lze jednoduše zjistit, zda je relé v režimu zapnuto nebo

```

{
void RelayOn()
{
    if (relay.isRelayOn())
    {
        lcd.clear();
        lcd.setCursor(4,0);
        lcd.printstr("Zapnuto");

        if (cas)
        {
            casovac++;

            while (n < casovac)
            {
                lcd.setCursor(7, 1);
                Casovac();
            }

            relay.off();
            lcd.clear();
            lcd.setCursor(4,0);
            lcd.printstr("Vypnuto");
        }
    }
    else
    {
        lcd.clear();
        lcd.setCursor(4,0);
        lcd.printstr("Vypnuto");
    }
}
}

```

Obrázek 11 Kód funkce RelayOn

v režimu vypnuto. Tato funkce je využita ve vlastní funkci, kde se kontroluje stav relé a poté obsluhuje displej a funkci Casovac. Vlastní funkce „RelayOn()“ se nachází vždy

na konci hlavního kódu, a to pro kontrolu, zda je relé zapnuto nebo není. V této funkci, pokud je kontrola vyhodnocena jako „true“, tudíž zapnuto, se provede vymázení displeje, nastavení kurzoru a vypsání informace na displej. Poté se kontroluje, zda byl zadán čas, jak dlouho má být relé zapnuto. Pokud byl čas nastaven správně, tak se pokračuje v časovači, který potom obsluhuje funkce Casovac. Po skončení odpočtu času je relé vypnuto a displej je znovu vymazán, aby byl nadále přehledný. Poté se znovu nastaví kurzor a vypíše se nový stav relé. Pokud ale čas správně zadáný nebyl, dojde k vymázení displeje a vypsání nové informace a to – Vypnuto.

2.4.3 Reset

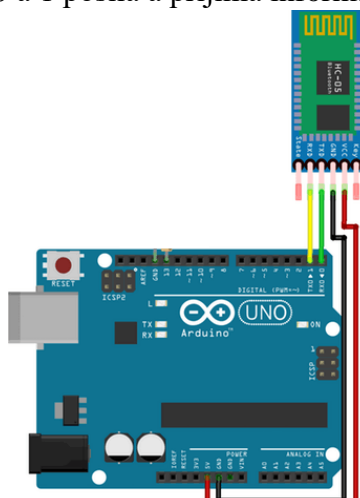
Jediný způsob, jak uvést relé do základního stavu, stavu před jakoukoliv manipulací, je přes tlačítko reset. Toto tlačítko se nachází na desce Arduino Uno 3 a je přístupné pouze fyzicky. Žádné tlačítko nebo příkaz v mobilní aplikaci není možný. Toto platí pro všechny komponenty použité v tomto projektu.

2.5 Bluetooth modul HC-05

Bluetooth modul HC-05 je využit jakožto sériové komunikační zařízení s mobilním telefonem pomocí rozhraní Bluetooth.

2.5.1 Zapojení modulu

Bluetooth modul je zapojen jako master zařízení, které se zapíná společně s Arduino deskou. Celé zařízení je zapojeno jednoduchým způsobem na proud 5V a zem, poté pomocí pinů na desce 0 a 1 posílá a přijímá informace od Arduina.



Obrázek 12 Schéma zapojení bluetooth modulu

Na tomto obrázku je možné pozorovat funkční zapojení modulu HC-05 s deskou Arduino.

2.5.2 Komunikace s mobilním zařízením

Komunikace je zajištěna pomocí knihovny SoftwareSerial.h, kde se nastaví rozhraní Bluetooth a piny přes které má operovat. Poté se již v kódu provedou příkazy „bluetooth.begin(9600)“, který započne komunikaci s telefonem a příkaz „bluetooth.println(„Arduino zapnuto!“)“, který oznámí uživateli stav arduina v jeho mobilní aplikaci.

```
if (bluetooth.available() > 0)
{
    BluetoothData = bluetooth.read();

    switch (BluetoothData) {
        case 'a':
            relay.on();

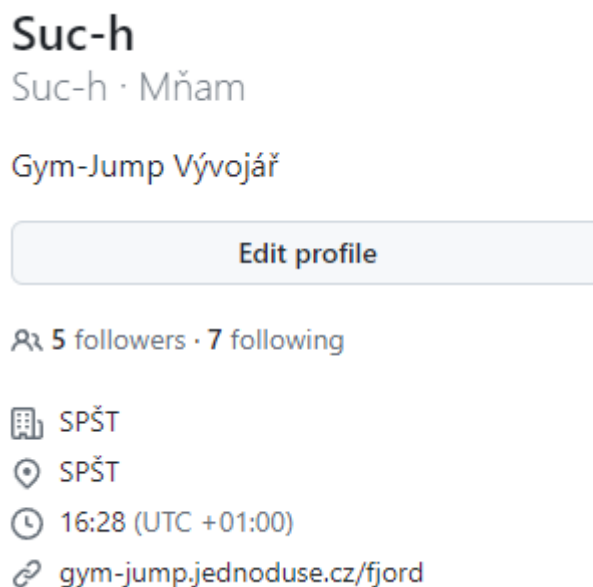
            cas = true;
            casovac = 5;
            break;
    }
}
```

Obrázek 13 Kód komunikace bluetooth

V tomto kódu lze vidět, že zařízení funguje na jednosměrné komunikaci. Modul čte bit po bitu a podle nich určuje další akce. V případě bitu „a“ zařízení provede zapnutí relé, nastavení proměnné čas na hodnotu „true“, díky které informuje program, že byl nastaven čas, a poté přiřadí časovači 5 sekund stavu zapnuto.

2.6 Verzování a zálohy

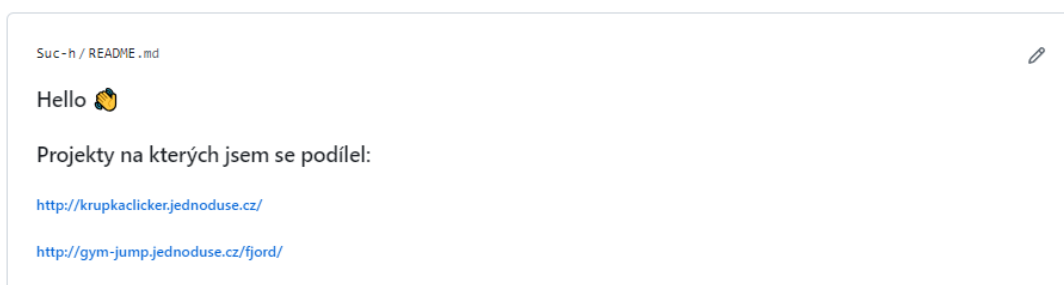
Zálohování projektu a jeho verzování je důležitá součást vývoje jakékoliv aplikace či projektu. Je totiž nutné mít při práci z domova kód přístupný online. K tomuto účel jsem si vybral Github, což je populární systém pro zálohování. Používá se převážně



Obrázek 14 Profil

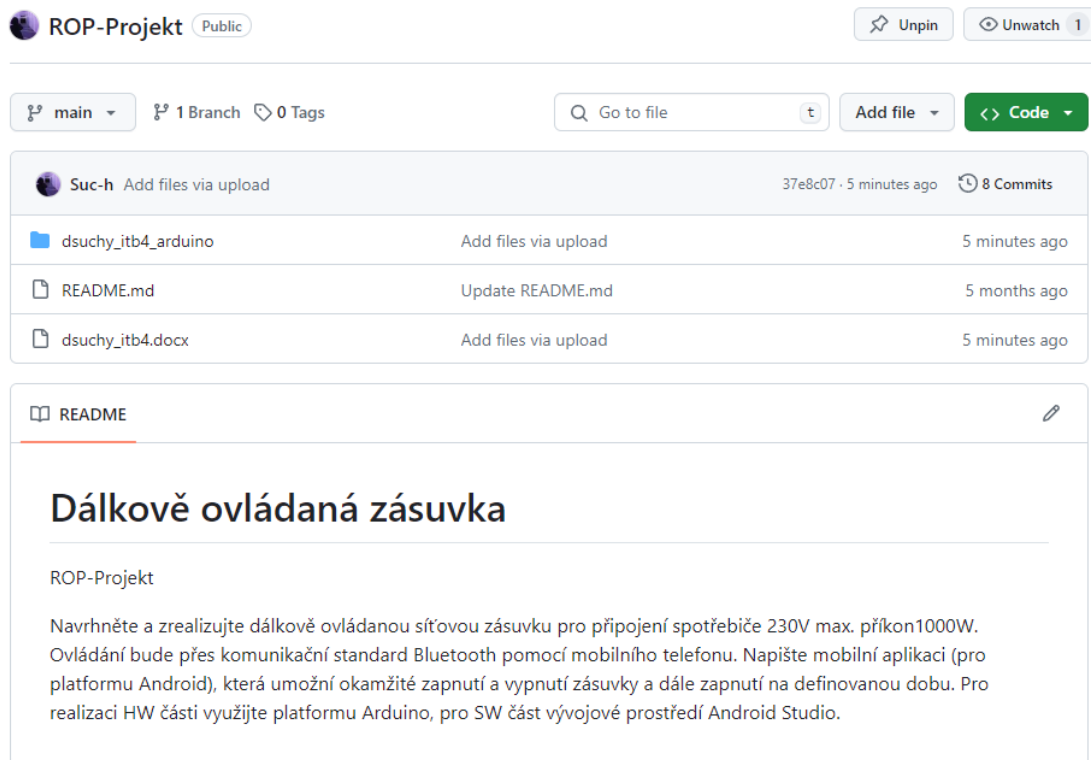
pro ukládání kódu a verzování aplikací, ale je možné do jeho úložiště zálohovat i různé soubory a dokumenty typu .docx nebo .txt.

Vlastní profil Github lze také modifikovat. Můj profil je například obohacený o místo, kde programuji většinu svých aplikací, o moji časovou zónu a odkaz na stránku.



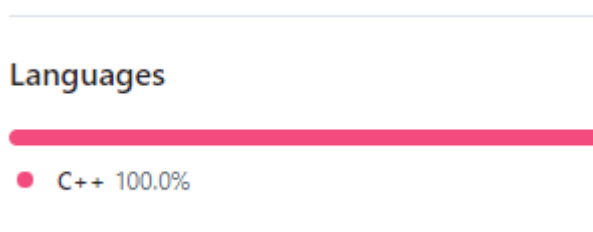
Obrázek 15 Informace o mém profilu

Při využívání Github prostředí je často důležité novým uživatelům, kteří narazí na váš profil říct, kdo jste, co děláte a jaké jsou vaše zájmy. Můj profil toto téma řeší poněkud lehkomyšlně a pouze zdraví a informuje o tom, na jakých projektech jsem se podílel.



Obrázek 16 Repositář

Na tomto obrázku můžeme pozorovat můj repositář na platformě Github. Je zde vidět, jak se soubory ukládají, a to například program do složky. Také si lze povšimnou uložení mé dokumentace jako soubor s příponou .docx. Soubor README.md obsahuje informace o zadání projektu a jeho názvu, který je viditelný na obrázku v rámečku nazvaném README.



Obrázek 17 Použitý programovací jazyk.

Github umožňuje uživateli nahlédnout do rozložení použitých programovacích jazyků. V mém případě, kdy jsem používal pouze platformu Arduino IDE a jeho programovací

jazyk je vidět, že využívá pouze jazyk C. Důvod toho je jednoduchý kód, který je v programu napsaný se překládá do jazyku C++.

2.7 Problémy

V této části budou popsány problémy, které vznikly při realizaci projektu. Jsou zde popsány obtíže z části hardwaru, které se vyskytly problémy při zapojení, tak komplikace z části softwaru. V aplikaci Arduino IDE i android studio.

2.7.1 Zapojení

V případě zapojení byl nalezen velký problém. Tato záležitost se týkala LCD displeje, který nezobrazoval data a pouze svítil. Naštěstí byl vyřešen tím, že vzadu na displeji byl špatně nastavený kontrast. Tudiž oprava byly rychlá a nečekaná. Jeden z dalších problémů, který nastal bylo samovolné vypnutí bluetooth modulu. Toto bylo vyřešeno jednoduchým vyměněním vodiče. Důvod, proč problém nastal, byl ten, že se na jednom konci vodiče nacházel poškozený konektor.

2.7.2 Inspirace

Při vymýšlení programu bylo obtížné nalezení nějakého podrobného návodu nebo dokonce i jednoduchého vyřešení spojení bluetooth a modulu Arduina. Kvůli tomuto problému jsem byl nucen většinu logiky kódu vymyslet sám. Z této komplikace vzešlo spoustu problémů, které zde nejsou popsány, jelikož byly vzápětí vyřešeny. Naštěstí byl po dlouhém hledání nalezen datasheet, díky kterému bylo umožněno ulehčení programovací části spojené s propojením rozhraní bluetooth mezi modulem HC-05 a mobilním telefonem.

2.7.3 Program Arduino

V oblasti programování nastaly 2 menší komplikace. Jedna z nich byla komunikace bluetooth modulu s mobilním zařízením. V tomhle ohledu byla chyba v nastavení pinů a v připojení skrz rozhraní bluetooth. Po úpravě nastavení a změně funkce mobilní aplikace se stala komunikace funkční. Druhá potíž, která nastala, bylo chybné počítání času. Ze začátku se čas přičítal namísto odčítal, a proto byl celý kód předělán od základu. Po přestavbě kódu se čas správně odčítal od nastavené hodnoty a vypisoval se správně na 2. řádku LCD displeje.

2.7.4 Android studio

Při řešení mobilní android aplikace se vyskytl problém při spojení hardwaru s mobilním telefonem. Hlavní chybou bylo to, že při spuštění programu, aplikace nenalezla žádné bluetooth zařízení, přestože byl bluetooth na zařízení zapnutý a modul se nacházel ve stavu online. Vlastní aplikace byla nevyhovující a proto byla nalezena aplikace nová, která byla již funkční. Aplikace byla uzpůsobena k mým potřebám. Zde se poté přidaly funkce k jednoduššímu využívání aplikace. Tyto funkce jsou tlačítka, která zastávají písmena nebo čísla. Tyto znaky se poté posílají přes sériovou komunikaci pomocí rozhraní bluetooth. Modul poté čte znaky a vyhodnocuje jakou akci provést.

2.7.5 Github

Při verzování na platformu Github nastal problém při nahrávání projektu. Ze začátku Github nebyl schopný verzovat Arduino IDE soubor, tak jak bylo potřeba. Z tohoto důvodu byl Github ze začátku nevyužívaný, ale bylo zjištěno, že při nahrávání projektu pomocí Github desktop je možné program nahrát. Bohužel však nebylo již možné program pomocí aplikace Github desktop stáhnout, a proto platforma sloužila spíše jako úložný prostor pro posílání nových verzí programu a dokumentace mezi školním počítačem a počítačem nacházejícím se doma.

2.7.6 Testování

Při testování a následném hodnocení ve škole byl nalezen problém při posílání dat na desku Arduino. Přestože doma data se přes USB kabel posílali bez jakéhokoli problému, ve škole tato možnost nebylo funkční. Tato komplikace byla vyřešena způsobem, výměny USB kabelu a přepojení do USB portu jiného počítače.

2.8 Ochrana zapojení

Jakožto ochrana projektu byla zvolena dřevěná krabice, která byla vytvořena z existující fošny. Tato fošna se uřízla na míru pro tento projekt. Poté do ní byla vyfrézovaná vnitřní část ve které bude celé zapojení vloženo. Deska je zašroubovaná přímo do spodní části krabice. Relé je také přišroubováno. Bluetooth modul je zastrčen v pravé části krabice, aby bylo viditelné blikání ledek. Celý projekt je zakryt plexisklem, na které je přichycen LCD displej pro lepší viditelnost zobrazovaných

informací. Krabička je natřena barvou, aby nevypadala nevzhledně. V krabičce jsou také vyvrtané díry pro napájení desky Arduino a pro přístup k tlačítku Reset. Tyto kroky byly udělány z důvodu jednoduššího přístupu k napájení, vodičům a tlačítkům na desce.

Závěr

V průběhu mé ročníkové práce jsem zjistil spoustu nových informací o systému Arduino a o jeho programovacím jazyku. Naučil jsem se lépe orientovat v jeho programovacím prostředí a aplikaci Arduino IDE, kde jsem program vytvářel. Během práce bylo potřeba vyřešit několik problémů. Mezi ty nejtěžší patřila komunikace mobilního zařízení s bluetooth modulem, který jsem vyřešil pomocí předělání celého kódu a android aplikace. Při srovnání se zadáním práce si myslím, že zadání jsem splnil tak, jak bylo napsáno. Jediná věc, kterou jsem splnil z poloviny, bylo vytvoření vlastní aplikace v android studiu. Tento bod jsem vyřešil využitím již hotové aplikace, kterou jsem uzpůsobil svým potřebám. Zbytek hardwaru a softwaru je zapojen a vytvořen mnou. Můj projekt je možné využít například v domácnosti, kde se například čerpá voda ze studny a je potřeba v daných intervalech zapínat a vypínat sání vody. Toto není jediné využití a je také možné pomocí tohoto projektu časovaně napájet vánoční osvětlení domu, nebo auta. Hlavní přínos práce je obohacení mých zkušeností v prostředí Arduino a celkové zlepšení mých schopností v oblasti mikroprocesorové techniky.

Seznam použitých zdrojů

- [1] *Arduino*. Online. ARDUINO.CC. Arduino. 2021, September 15 2021. Dostupné z: <https://www.arduino.cc/>. [cit. 2024-03-20].
- [2] *Arduino UNO R3*. Online. ECLIPSERA, Laskakit, c2024 Dostupné z: <https://www.laskakit.cz/arduino-uno-r3--atmega328p--klon/>. [cit. 2024-02-29].
- [3] *Wiring*. Online. Wiring. Processing, 1999. Dostupné z: <https://wiring.org.co/about.html>. [cit. 2024-03-20].
- [4] VODA, Zbyšek. Základní struktury jazyka Wiring. *Bastlirna* [online]. hwkitchen, 2014. Dostupné z: <https://bastlirna.hwkitchen.cz/zakladni-struktury-jazyka-wiring/>. [cit. 2024-03-26]
- [5] YOUNGBLOOD, Tim. *How To Use Arduino's Analog and Digital Input/Output (I/O)*. Online. All About Circuits. EETech, c2024. Dostupné z: <https://www.allaboutcircuits.com/projects/using-the-arduinios-analog-io/>. [cit. 2024-03-20].
- [6] TŮMA, Ondřej. Arduino - programování v čistém C(++). *Bastlirna* [online]. McBig, 2011. Dostupné z: <https://zeropage.cz/a/arduino-programovani-v-cistem-c> [cit. 2024-03-26].
- [7] NĚMEČEK, Dominik. *Lekce 4 - Proměnné - Datové typy a deklarace*. Online. ITnetwork. Praha: ITnetwork, 2011. Dostupné z: <https://www.itnetwork.cz/hardware-pc/arduino/programovaci-jazyk/promenne-datove-typy-a-deklarace>. [cit. 2024-03-26].
- [8] *What is a code library?* Online. Quora. Quora, c2024. Dostupné z: <https://www.quora.com/What-is-a-code-library>. [cit. 2024-03-26].
- [9] *Wiring: programovací jazyk*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: [https://cs.wikipedia.org/wiki/Wiring_\(programovac%C3%AD_jazyk\)](https://cs.wikipedia.org/wiki/Wiring_(programovac%C3%AD_jazyk)). [cit. 2024-03-26].
- [10] *LCD monitor*. Online. Lcd monitor. Dostupné z: https://old.pf.jcu.cz/stru/katedry/fyzika/prof/Tesar/diplomky/pruvodce_hw/komponenty/zobraz-zarizeni/monitor/lcd.htm. [cit. 2024-02-29].
- [11] *Electrovygian*. Online. Electrovygian. 2023. Dostupné z: <https://www.electrovygian.com/arduino/1-channel-relay-with-arduino/>. [cit. 2024-03-25]
- [12] ČERNÝ, Michal. *I2C adaptér pro LCD displej – Arduino*. Online. RoboDoupě. RoboDoupě, 2015. Dostupné z: <https://robodoupe.cz/2015/i2c-adapter-pro-lcd-displej/>. [cit. 2024-03-26].
- [13] *Relé modul*. Online. Relé modul. LASKAKIT 2024. Dostupné z: <https://www.laskakit.cz/1-kanal-5v-rele-modul--high-low-250vac-30a/>. [cit. 2024-03-26].

2024-02-29].

- [14] *El. mag. relé*. Online. El. mag. relé. Elektrokomponenty AZ 2014 - 2024 ©. Dostupné z: <http://www.eshop.elektrokomponenty.cz/novinky/elektromagneticke-rele-co-je-a-jak-funguje/>. [cit. 2024-02-29].
- [15] *Bluetooth modul*. Online. Bluetooth modul. ECLIPSERA 2024. Dostupné z: <https://dratek.cz/arduino/1005-bluetooth-modul-hc-05.html>. [cit. 2024-02-29].
- [16] *Bluetooth modul*. Online. Bluetooth modul. HWKITCHEN 2024. Dostupné z: <https://www.hwkitchen.cz/bluetooth-modul-hc-05/>. [cit. 2024-02-29].
- [17] *Bluetooth modul*. Online. Bluetooth modul. ECLIPSERA 2021. Dostupné z: <https://dratek.cz/docs/produkty/1/1401/1427822941.pdf>. [cit. 2024-02-29].
- [18] *What are LEDs and How Do They Work*. Online. ROHM CO. ROHM CO., 2016. Dostupné z: <https://www.rohm.com/electronics-basics/leds/what-are-leds>. [cit. 2024-03-27].

Seznam použitých symbolů a zkratek

Zkratka	Význam
LCD	Liquid Crystal Display (displej z tekutých krystalů)
USB	Univerzální Sériová Sběrnice
V	Volty
VCC	Napětí společného kolektoru
GND	Uzemnění

Seznam obrázků

Obrázek 1 Arduino Uno [2]	9
Obrázek 2 Obecné schéma zapojení LCD displeje Chyba! Nenalezen zdroj odkazů.	12
Obrázek 3 Obecné zapojení relé [11].....	13
Obrázek 4 Připojení k USB TTL převodníku [17]	14
Obrázek 5 Schéma zapojení LCD displeje v projektu	16
Obrázek 6 Knihovny a přiřazení informací v kódu.....	17
Obrázek 7 Základní příkazy LCD	17
Obrázek 8 Kód funkce Časovač	18
Obrázek 9 Definování intervalu a nastavení pinu	19
Obrázek 10 Schéma zapojení relé do desky Arduino	20
Obrázek 11 Kód funkce RelayOn	21
Obrázek 12 Schéma zapojení bluetooth modulu.....	22
Obrázek 13 Kód komunikace bluetooth.....	23
Obrázek 14 Profil	24
Obrázek 15 Informace o mém profilu.....	24
Obrázek 16 Repositář.....	25
Obrázek 17 Použitý programovací jazyk.	25

