

**Gymnázium a střední odborná škola Mikulov,
příspěvková organizace**



Arduino – DIY chytré hodinky

Seminární práce

Mikulov, 2022

ZADÁNÍ SEMINÁRNÍ PRÁCE

Název práce:	Arduino-DIY chytré hodinky		
Jméno autora:	Zbyněk Dohnálek		
Obor vzdělání:	79-41-K/81 Gymnázium	Třída:	7.0
Seminář:	IVT SMF	Školní rok:	2021/2022
Termín odevzdání:	15.3. 2022	Rozsah	10 stran

Zásady pro vypracování:

1.	Z dostupné literatury nastudujte a stručně shrňte historii Arduina. Přidejte příklady jeho využití v dnešní době.
2.	Naučte se základy jazyka C a naprogramujte hodinky.
3.	Z dostupné literatury čerpejte informace o fungování IOT vývojové platformy Arduino a různých periférií s ním spojené, které použijete u hodinek.
4.	Vyjádřete svůj vlastní názor na využití Arduina ve vašem projektu a celkově ve světě.


Literatura:

- Zbyšek Voda & Tým HW Kitchen, Průvodce světem arduina, 2.vydání,ebook, 2018,
Dostupné z: <https://bastlirna.hwkitchen.cz/>
- Herbert Schildt, Nauč se sám C, SoftPress, 2001 ISBN: 80-86497-16-X
Případně lze rovněž čerpat z materiálů dostupných online

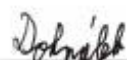
V Mikulově 13.9.2021



Roman Pavlačka
ředitel



Miroslav Siebert
vedoucí práce



autor

Souhrn

Seminární práce se zabývá návrhem chytrých hodinek. První část se zabývá historií a principy komunikace Arduina. Druhá část samostatným popisem využitých periférií, realizací a naprogramování zařízení. Na konci je uveden i způsob případného vylepšení hodinek.

Klíčová slova

Arduino, Bluetooth, Nano, sériová komunikace, zařízení

Resumé

This seminar work attempts with the design of a smart watch. The first part deals with the history and principles of Arduino communication. The second part deals with a separate description of the peripherals used, implementation and programming of the device. At the end, there is also a way to improve the watch.

Keywords

Arduino, Bluetooth, Nano, serial communication, device

Prohlášení

Prohlašuji, že jsem seminární práci na téma „Arduino-DIY chytré hodinky” vypracoval samostatně, s využitím pouze citovaných literárních pramenů, dalších informací a zdrojů 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 Mikulově 22.března 2022

Dohnálek

.....

Zbyněk Dohnálek

Poděkování

Děkuji vedoucímu práce Ing. Miroslavu Siebertovi, Ph.D., za vstřícnost, odborné vedení a cenné rady při psaní této seminární práce. Škole za umožnění využití 3D tiskárny a Adamovi Krškovi za pomoc při tisknutí.

Obsah

1	Úvod	7
2	Historie Arduina	8
2.1	Využití Arduina v dnešní době.....	9
3	Principy komunikace	10
3.1	Sériová komunikace	10
4	Periferie použité v projektu	11
4.1	Bluetooth HC-05	11
4.2	Real Time Clock (Hodiny reálného času)	11
4.3	I2C OLED display.....	11
4.3.1	Sběrnice I2C.....	11
4.4	Step-up měnič DC-DC	11
4.5	USB nabíječka TP4056	12
4.6	Arduino Nano V3.0	12
5	Návrh a realizace zařízení.....	13
5.1	Zapojení komponent.....	13
5.2	Realizace zařízení.....	14
5.3	Programování Arduina	15
5.4	Konečný produkt.....	16
6	Návrhy pro zlepšení zařízení	17
7	Závěr.....	18
8	Seznam použité literatury	19
9	Přílohy	20
9.1	Seznam obrázků	20
9.2	Seznam tabulek	20
9.3	Zdrojové kódy (důležité části).....	20
10	Dokumentace	23

1 Úvod

V dnešním světě se můžeme setkat s velkým počtem moderních technologií v podobě drobných zařízení, které nám umožňují vytvářet nové projekty, jako jsou roboti, mini-počítače, případně si jimi můžeme vybavit i celou domácnost. Tyto zařízení využívají nespočet periférií, díky kterým jsou rozšiřovány o nové možnosti jejich využití. Jedním z takových zařízení je například Arduino, které se díky jeho jednoduchosti a velké využitelnosti v každodenním životě stalo populárním po celém světě.

V dřívějších letech většina běžných uživatelů využívající elektronická zařízení neměla v podstatě ani tušení, jak určité zařízení jimi zadané funkce vykonává a z čeho se skládá. To se postupem času změnilo. Čím dál více lidí se začalo zajímat, jak určitá elektronika funguje, ze kterých komponentů se skládá a co v ní která součástka vykonává. Proto nastala i pro mnohé otázka, zda by si místo koupené meteostanice z obchodu nemohli „zbastlit“ svoji vlastní. Díky tomu, že Arduino stále navrhuje, vyrábí a podporuje elektronická zařízení a programy, lidé tuto možnost mají. Umožňuje také přístup k pokročilejším technologiím jako je AI¹ a pracovat s nimi.

Arduino by se dalo přirovnat k Legu. Máme neomezené zdroje, jak něco vytvořit. Můžeme stavět podle návodu nebo vytvořit něco dle vlastní fantazie. Není určené jen pro profesionální vývojáře, ale i pro studenty, kteří teprve začínají s programováním. Pokud si vytvoří vlastní projekt, u kterého uvidí, že ve skutečnosti něco vykonává, např. po zhasnutí světla se automaticky rozsvítí jejich lampička, motivuje je v tom pokračovat.

Z toho důvodu jsem si i já vybral projekt, který mi pomůže naučit se základy v programování a pochopit, jak periferie, které jsem v projektu použil, společně fungují.

Cílem této seminární práce je pomocí Arduina navrhnout a naprogramovat hodinky, které bude uživatel schopen bezdrátově propojit s mobilem a za pomoci aplikace ovládat a měnit jejich nastavení.

¹ Umělá inteligence – Artificial intelligence (AI) je schopnost počítačového systému napodobovat lidské kognitivní funkce, jako je učení nebo řešení problémů [18]. Ve skutečnosti AI nemá přesnou definici a stále se upravuje.

2 Historie Arduina

Arduino se začalo vyvíjet od roku 2005 v malém městě Ivra na Institutu interakčního designu (v originále Interaction Design Institute Ivrea) v Itálii. Většina uživatelů v té době používalo desky BASIC Stamp od společnosti Parallax, které byli poměrně drahé (vycházely průměrně na 76 €). Bylo proto zapotřebí vytvořit levný, jednoduchý a výkonný set s deskou, který si budou moci dovolit i studenti. Vznikly tedy první prototypy Arduina, které se prodávali pouze mezi známými. Pro Arduino nadešla chvíle, kdy vznikly plně sestavené a otestované verze, které se prodaly za výrobní cenu do místních škol jako test. Mezi studenty se Arduino díky své jednoduchosti a přijatelné ceně uchytilo a postupem času se začalo prodávat v obchodech jako je SparkFun Electronics nebo Adafruit [2]. Aby se Arduino rozšířilo do celého světa, rozhodli se tvůrci z Arduina udělat Open Source projekt [1].

Spolu s Arduinem vznikl programovací jazyk Processing, který je složen z grafické knihovny pro jazyk Java a integrovaného vývojové prostředí (GUI). Byl navržen pro zjednodušení výuky díky propojení a sloučení grafického designu s programováním. Jeho tvůrci věřili, že propojení umění s moderní technologií je nezbytnou součástí pro vzdělávání od útlého věku. Inspirací pro vytvoření jazyku Processing byli programovací jazyky BASIC a Logo [10]. S jeho zaměřením na vizuální, interaktivní media se často využívá designéry, umělci a architekty k vytváření jejich projektů [3]. Pozměněnou verzí programovacího jazyka Processing je jazyk Wiring, který se používá při práci s Arduinem k naprogramování jejich projektů. Jeho myšlenkou je napsání kódu o pár řádcích, připojit pár komponentů k hardwaru (Arduinu) a sledovat, jak se dioda zapne, když se člověk přiblíží. Připsat dalších pár řádků kódu, přidat další sensor a sledovat, jak se na základě světla v místnosti mění jas diody. Tento proces vytváření je nazývaný „Sketching with hardware“ a je jedním ze způsobů, jak objevit nové myšlenky rychlou cestou [4].

Vzhledem k tomu, že Arduino je Open Source projekt a všechna schémata desek i data ohledně softwaru jsou volně dostupná online, se na trhu objevují i klony Arduina, které vystupují pod jinými, někdy téměř podobnými názvy. K nejznámějším klonům patří Infidduino a Freeduino. Díky tomu si i my můžeme vytvořit nebo koupit klon, který je většinou levnější a téměř identický k oficiálním modelům Arduina.

2.1 Využití Arduina v dnešní době

V dnešní době se s Arduinem můžeme setkat téměř všude. Můžeme si vyrobit vlastního drona, který byl dříve využíván pouze jako vojenská technologie. Nastavit si čas, kdy chceme zatáhnout rolety na oknech a Arduino to za nás udělá. Někteří lidé si vytvoří i chytrý květináč, aby mohli komunikovat se svojí rostlinou doma v pokoji. Ke květináči dají Arduino, které má k sobě připojené sensory teploty, vlhkosti, WiFi modul a atd. Poté, pokud uvnitř místnosti bude příliš vysoká teplota nám rostlina může poslat zprávu na mobil: “Je mi horko.” nebo “Potřebuji vodu!” Někdo zase nechce poslouchat určité osobnosti v televizi. Z toho důvodu si může v určitou chvíli, kdy určitá osoba začne mluvit, pomocí Arduina ztlumit zvuk.

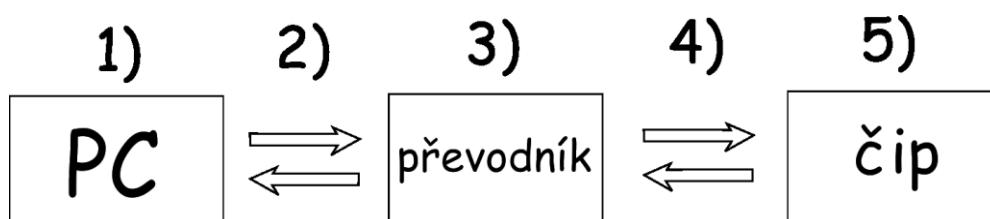


Obr. 1 Oficiální logo platformy Arduino. [1]

3 Principy komunikace

Pro komunikaci Arduina s PC musíme mít několik součástí. Následující popis postihuje většinu desek při programování, se kterým se setkáváme u Arduina [1].

Co je ke komunikaci potřeba:



Obr. 2 Schéma sériové komunikace. [1]

- 1) Počítač s USB portem.
- 2) USB kabel
- 3) Převodník, který se nachází na základní desce Arduina, přímo v jeho čipu nebo připojíme externí převodník. Všechny převodníky se liší pouze způsobem propojení, jejich fungování zůstává stejné [1].
- 4) Pomocí převodníku se připojíme na Arduino, na kterém se nachází čip.
- 5) Čip přijímá instrukce z převodníku.

3.1 Sériová komunikace

Základem sériové komunikace jsou dva stavy (vede/nevede, svítí/nesvítí). V historii se s ní setkáváme již před naším letopočtem, kdy se využívalo kouřových signálů jako známka nebezpečí. Později se začalo využívat morseovy abecedy v přístroji známý jako telegraf. Protože měl každý znak jinou délku, využívání se stalo nevhodné a hledala se alternativa, která sjednotí délku znaků. Začal se používat přístroj telex, který fungoval na principu relátek, které spínají a rozepínají elektrické spojení. Přes sériovou linku se přenášela 5-ti bitová slova rychlostí až 50 bitů/s. Postupem času se stala 5-ti bitová slova nedostatečná a tak začala nová zařízení využívat 8 bitů. Začala vznikat první polovodičová zařízení, která stále zvyšovala svou rychlost, až se dostala na 300 bitů/s. Rychlost se dále násobila a komunikace se rozšiřovala přidáváním vodičů. S nárůstem mobilních zařízení se začali nahrazovat vodiče a vznikala zařízení využívající infračerveného světla [5]. Stále s přibývajícím nárokem na vysokou přenosovou rychlost vznikl i nový komunikační systém založený na sériové komunikaci a to USB (Universal Serial Bus).

4 Periferie použité v projektu

4.1 Bluetooth HC-05

S Arduinem komunikujeme po sériové lince piny označenými RX (receive) a TX (transmit). Díky těmto pinům můžeme komunikovat s Arduinem, když už na něm máme nahraný program. Pomocí komunikace můžeme ovládat Arduino nebo přijímat naměřené hodnoty ze senzorů. Za pomoci Bluetooth máme možnost komunikovat s Arduinem bezdrátově. Obr. 6

4.2 Real Time Clock (Hodiny reálného času)

Hodiny reálného času (dále jen RTC) se dříve využívali jen u počítačů. Při vypnutí se čas přenesl do RTC a uchová se až do následného zapnutí. Po zapnutí a připojení k internetu se synchronizuje pomocí síťového časového protokolu (NTP). Moduly jsou vybaveny bateriemi, díky kterým ukládají čas při malém odběru elektřiny po vypnutí zařízení [6]. Obsahují křemenový krystal, který způsobuje vibrace o určité frekvenci a počtem těchto vibrací udržují stálý čas [7]. Modul DS1307 použitý v projektu dokáže měřit čas a má přesný kalendář až do roku 2100 [8]. Na základě vlastního měření jsem zjistil, že tento typ modulu se předbíhá až o 90 sekund za měsíc. Z toho důvodu musíme RTC pravidelně synchronizovat nahráním času z počítače. Modul je propojen pomocí pinů SDA a SCL. Viz. 4.3.1. Obr. 4

4.3 I2C OLED display

V projektu jsem využil OLED displej, který má díky možnosti ovládání jednotlivých pixelů nezávisle na sobě nízkou spotřebu. S Arduinem je propojen pomocí pinů SDA a SCL, kterými komunikuje po rozhraní I2C sběrnice. Obr. 5

4.3.1 Sběrnice I2C

Pomocí I2C sběrnice můžeme propojit jedno řídicí zařízení (master), v našem případě Arduino, až se 128 dalšími zařízeními, která jsou řízena (slave). V mém projektu je řízený displej a RTC. Sběrnice využívá dva vodiče nazývané SCL (clock line), který udává takt komunikace a SDA (data line) přenášející data [1][15].

4.4 Step-up měnič DC-DC

Modul slouží k převodu napěťových úrovní a používá se jako zdroj DC napětí. Tento typ označujeme jako step-up boost, výstupní napětí má tedy vždy vyšší hodnotu než vstupní. U tohoto modulu je výstupní hodnota pevně daná na 5 VDC [11]. Obr. 7

4.5 USB nabíječka TP4056

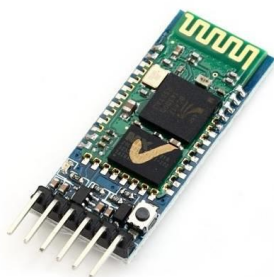
Nabíječka je určená pouze pro lithium-polymerové (Li-Pol) baterie. Hlavním prvkem modulu je čip TP4056. Má na sobě další dva čipy, které chrání před přepětím, podpětím a nadproudem [12]. Obr. 8

4.6 Arduino Nano V3.0

V projektu je využito precizního klonu, který má stejné osazení jako oficiální verze Arduina. Jedná se o jednu z nejmenších verzí Arduina vhodnou pro využití v malých projektech. Na rozdíl od Arduina Mini s podobným osazením obsahuje USB port a převodník, díky kterým nám usnadňuje jeho programování. Běží na procesoru ATmega328 s provozní frekvencí 16 MHz. Jeho flash paměť dokáže pojmout až 32 kB [9]. (V mém projektu jsem využil přibližně 20,3 kB) Obr. 3



Obr. 3 Precizní klon Arduino Nano V3.0. [9]



Obr. 6 Bluetooth HC-05 [14]



Obr. 4 RTC hodiny reálného času [14]



Obr. 5 I2C OLED displej [14]



Obr. 7 Step Up měnič napětí 5V [14]



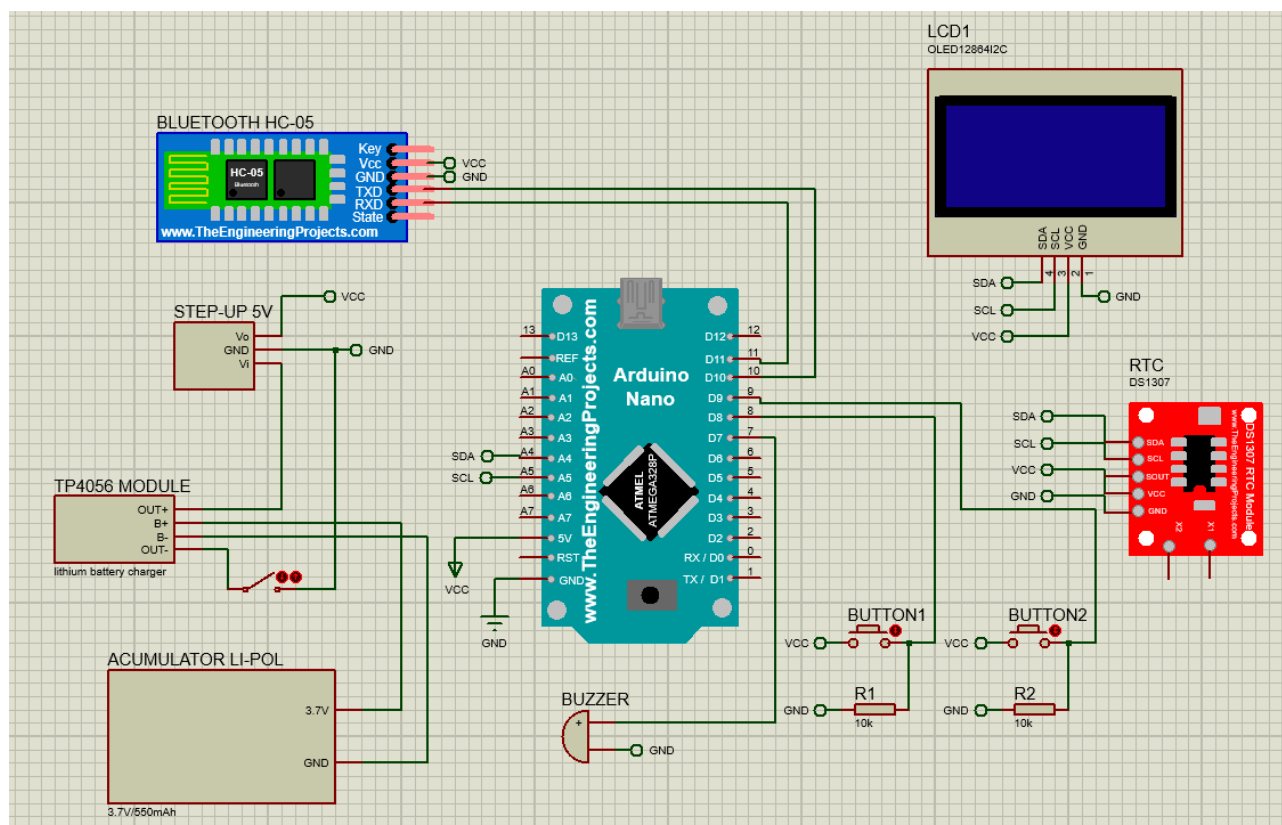
Obr. 8 TP4056 Micro USB nabíječka [14]

5 Návrh a realizace zařízení

Cílem této seminární práce je navrhnout DIY hodinky, které budou zvládat základní operace běžných hodinek. Dále budou schopné propojit se s mobilem pomocí Bluetooth technologie. Připojené zařízení bude možné ovládat a měnit nastavení za pomoci mobilu.

5.1 Zapojení komponent

Navržené zařízení využívá vývojové desky Arduino Nano V3.0 (dále jen Arduino) na které jsou připojeny ostatní periferie. Arduino a jejich periferie jsou popsány v kapitole 4. Arduino je napájeno pomocí akumulátoru Cellevia Li-pol (3,7 V/550 mAh). Díky micro USB nabíječce můžeme akumulátor dobít. Vzhledem k tomu, že Arduino funguje na napětí 5 V, je připojen k zařízení STEP-UP převodník, který nám zvýší dodávané napětí 3,7 V z baterie na 5 V. Mezi nabíječkou a převodníkem se nachází přepínač, kterým můžeme celé zařízení vypnout nebo zapnout. Celkové schéma zapojení můžeme vidět na Obr. 9 (schéma bylo vytvořeno pomocí programu Acronis).



Obr. 9 Schéma zapojení Arduino

5.2 Realizace zařízení

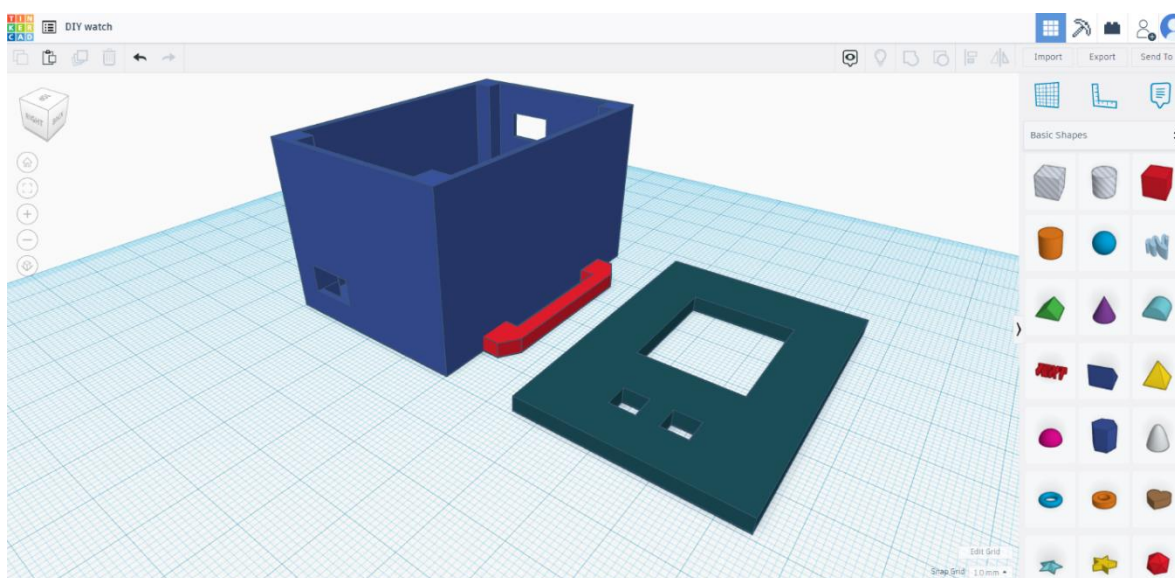
Prvním krokem bylo uspořádat periferie připojené na Arduino do co nejmenšího celku. Vzhledem k tomu, že většina periférií je upravena pro nepájivé pole, bylo nutné z některých periférií vypájet předpájené piny. Periferie jsem tímto krokem ještě více zmenšil a docílil toho, aby hodinky bylo možné instalovat do co nejmenšího pouzdra pro běžné nošení na zápěstí ruky.

K propojení zařízení s mobilem jsem potřeboval funkční aplikaci. V mém případě se mi nabízela aplikace MIT App Inventor, ve které si mohu vytvořit vlastní aplikaci na míru. Aplikace funguje na principu skládání bloků kódu, se kterým se setkáváme převážně u známé aplikace Scratch.



Obr. 10 Vytvořená aplikace za pomoci MIT App Inventor

Dalším krokem bylo vytvořit pouzdro, do kterého se zařízení umístí. V takovém případě jsem využil aplikace Tinkercad, ve které jsem si pouzdro vymodeloval, následně nahrál do Creality Slicer a pomocí 3D tiskárny vytiskl.

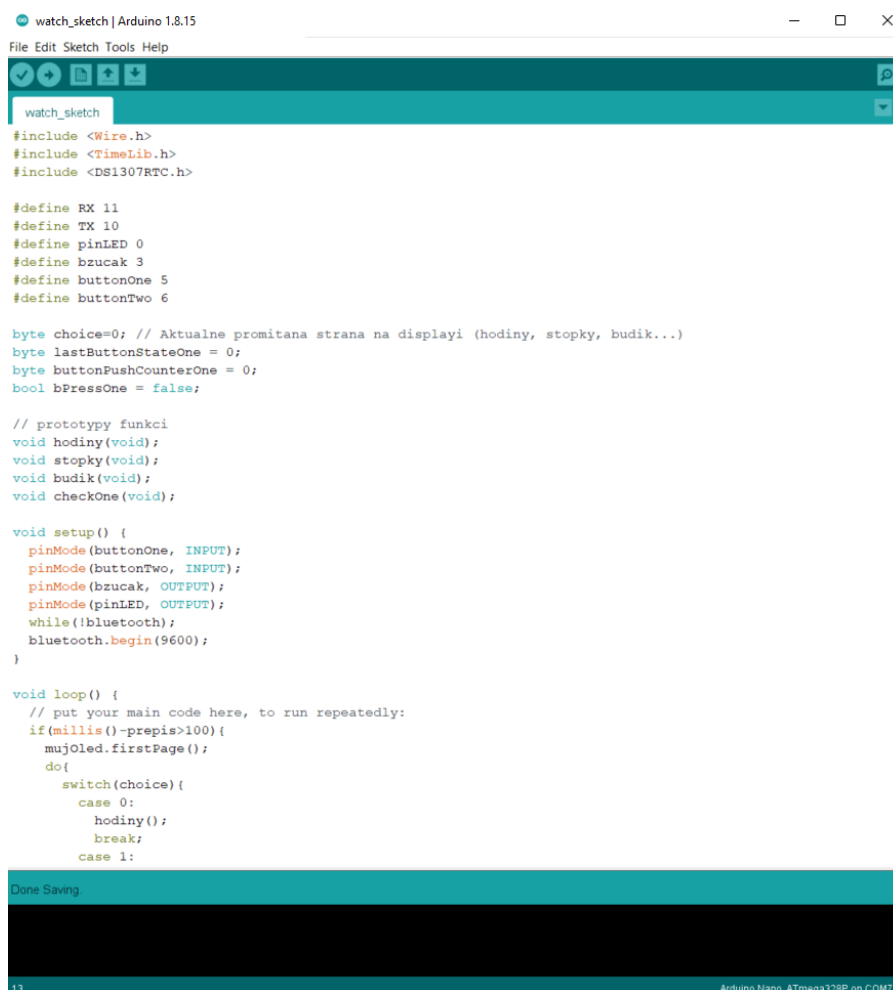


Obr. 11 Vymodelování krabičky v Tinkercad

5.3 Programování Arduina

Arduino můžeme programovat ve více programovacích jazycích jako jsou C/C++, MicroPython nebo Java. Samotné Arduino nám nabízí k využití vývojového prostředí Arduino IDE, které je jednou z nejpoužívanějších možností. Arduino IDE bylo napsané v jazyce Java a vzniklo z již zmíněného výukového prostředí Processing. V Arduino IDE budu využívat jazyku Wiring, který je zjednodušenou verzí C/C++ [1].

Kód psaný v Arduino IDE se skládá z několika částí. Nejprve jsou do kódu přidány knihovny a deklarovány globální proměnné. Dále jsou inicializovány vstupně-výstupní porty, na které jsou periférie připojeny. Všechno, co se nachází uvnitř funkce „setup()” se provede pouze na začátku po spuštění programu. Jako poslední částí je samotná funkce „loop()”, která běží, dokud se zařízení nevypne nebo program sám nezahavaruje (nenastane chyba). Ukázku programovacího prostředí je možné vidět na Obr. 12.



```
watch_sketch | Arduino 1.8.15
File Edit Sketch Tools Help

watch_sketch
#include <Wire.h>
#include <TimeLib.h>
#include <DS1307RTC.h>

#define RX 11
#define TX 10
#define pinLED 0
#define buzcak 3
#define buttonOne 5
#define buttonTwo 6

byte choice=0; // Aktualne promitana strana na displayi (hodiny, stopky, budik...)
byte lastButtonStateOne = 0;
byte buttonPushCounterOne = 0;
bool bPressOne = false;

// prototypy funkci
void hodiny(void);
void stopky(void);
void budik(void);
void checkOne(void);

void setup() {
  pinMode(buttonOne, INPUT);
  pinMode(buttonTwo, INPUT);
  pinMode(bzucak, OUTPUT);
  pinMode(pinLED, OUTPUT);
  while(!bluetooth);
  bluetooth.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(millis()-prepis>100){
    mujOled.firstPage();
    do{
      switch(choice){
        case 0:
          hodiny();
          break;
        case 1:

```

Obr. 12 Ukázka programovacího prostředí Arduino IDE 1.8.15

5.4 Konečný produkt

Zařízení se ovládá pomocí dvou tlačítek. Horním tlačítkem přepínáme údaje (čas, budík, stopky...), dolním tlačítkem nastavujeme samotné funkce.



Obr. 13 Arduino-DIY hodinky

Technické parametry

Název	Arduino - DIY hodinky
Záruka	Neomezená
Určení	Pánské
Styl	Inženýrský
Tvar	Hranatý
Hmotnost	59 g
Voděodolnost	Ne
Pouzdro	Plast
Řemínek	Pletený
Rozměry	60 x 40 x 25 mm
Baterie	Li-Pol (550 mAh)

Tabulka 1 Technické parametry hodinek

Funkce hodinek

Místní čas
Stopky
Alarm
Kalendář
Osvětlení
Bluetooth
Slovník (ENG)

Tabulka 2 Funkce hodinek

6 Návrhy pro zlepšení zařízení

Každé zařízení se dá vždy něčím vylepšit. Někomu může chybět, že zařízení nemá určitou funkci a některým by to naopak mohlo připadat zbytečné. V této kapitole shrnu návrhy možného vylepšení, a to jak softwarového, tak hardwarového.

Pokud bych zanedbal nedostatečné znalosti v oboru, mohl bych si místo využití Arduina Nano vyrobit vlastní plošný spoj, který by již všechny periferie obsahoval. Ušetřilo by se tím místo a celé hodinky by byly mnohem menší. Dále bych mohl přidat další senzory dle vlastního výběru a měřit více funkcí. Pro pohodlnější zacházení přidat třetí tlačítko a zjednodušit tím přecházení v menu.

Zda bych si nechtěl vyrobit vlastní plošný spoj a použít Arduino, dalo by se využít vývojové desky TinyPICO. Jedná se o nejmenší desku na světě s čipem ESP32, který disponuje WiFi a Bluetooth. Díky své váze pouhé 2 g by se zmenšila i celková váha [15].

Co se týče softwarové části, mohl bych si vytvořit pokročilejší a přehlednější menu, které by zpříjemnilo používání jednotlivých funkcí. Přidat viditelný stav baterie a světové časy. Pomocí Bluetooth umožnit, aby mi chodila upozornění z mobilu. Jako vychytávky bych mohl přidat jednoduché hry (např. snake) nebo umožnit změnu rozvržení hlavního panelu s časem dle výběru uživatele.

7 Závěr

Jedním důležitým faktorem je neustále se rozvíjet a učit se novým věcem. Abychom se mohli rozvíjet v oblasti moderních technologií, potřebujeme nasbírat nespočet zkušeností, které získáme převážně tvorbou nových projektů a jejich následným vylepšováním.

Open Source platformy, jako je Arduino, umožňují lidem stavět na něčem, co již bylo vytvořeno. Jejich komunity se neustále rozrůstají a díky tomu vzniká i mnoho nových projektů, na kterých se můžeme podílet, nebo se jimi alespoň inspirovat. Díky tomu máme také možnost zadarmo číst kódy zkušených vývojářů z celého světa a učit se od nich.

V této seminární práci se podařilo navrhnout a pomocí schématu vytvořit jednoduché hodinky se základními funkcemi, osvojit si základy v programování a propojit zařízení s mobilem. Než jsem začal s programováním a návrhem samotných hodinek, musel jsem se seznámit, jak Arduino samo o sobě funguje a rozhodnout se, které periferie v projektu využiji. Bylo to klíčovou fází, od kterého se odvíjel celkový vývoj projektu.

V sekci 5.4 můžeme vidět plně sestavené a naprogramované hodinky, které obsahují základní funkce hodinek, jsou schopné se bezdrátově propojit a ovládat pomocí mobilu. Všechny zásady pro vypracování byly úspěšně splněny a můžeme tedy považovat projekt za dokončený.

8 Seznam použité literatury

- [1] VODA, Zbyšek. *Průvodce světem Arduina*. Vydání druhé. Bučovice: Martin Stříž, 2017. ISBN 978-80-87106-93-8
- [2] Arduino The Documentary (2010) Dostupné z:
https://www.youtube.com/watch?v=D4D1WhA_mi8
- [3] Introduction to the Processing software. Dostupné z: <https://processing.org/overview/>
- [4] Wiring. An environment and electronics. Dostupné z: <http://wiring.org.co/about.html>
- [5] Jednočipové mikroprocesory a způsoby jejich komunikace. Dostupné z:
<https://www.fi.muni.cz/usr/jkucera/pv109/2003/xharaga.htm>
- [6] How do computers keep track of time? Dostupné z:
<https://cs.stackexchange.com/questions/54933/how-do-computers-keep-track-of-time>
- [7] A real-time clock (RTC). Dostupné z: <https://www.electronics-tutorials.ws/connectivity/real-time-clocks.html>
- [8] Drátek-návody. RTC DS1307. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/hodiny-realneho-casu-ds1307.html>
- [9] Arduino nano V3.0 precizní klon. Dostupné z: <https://dratek.cz/arduino/823-arduino-nano-v3.0-atmega328-precizni-klon.html>
- [10] The history of Processing by Casey Reas and Ben Fry. Dostupné z:
<https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85>
- [11] DC-DC Step Up měnič napětí 5V. Dostupné z:
<https://dratek.cz/docs/produkty/0/992/1557993668.pdf>
- [12] TP4056 module specs. Dostupné z: <https://bikepixels.com/2019/11/10/how-to-add-a-lipo-battery-in-our-arduino-projects/>
- [13] OLED displej. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/oled-displej-ssd1306.html>
- [14] Dratek.cz obrázky periférií. Dostupné z: <https://dratek.cz/>
- [15] NĚMEC, Radek. Systém ovládání počítačové prezentace s využitím gest. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/134430>
- [16] Definice umělé inteligence. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-artificial-intelligence/>

9 Přílohy

9.1 Seznam obrázků

Obr. 1 Oficiální logo platformy Arduino. [1]	9
Obr. 2 Schéma sériové komunikace. [1]	10
Obr. 3 Precizní klon Arduino Nano V3.0. [9]	12
Obr. 4 RTC hodiny reálného času [14]	12
Obr. 5 Bluetooth HC-05 [14]	12
Obr. 6 I2C OLED displej [14]	12
Obr. 7 TP4056 Micro USB nabíječka [14]	12
Obr. 8 Step Up měnič napětí 5V [14]	12
Obr. 9 Schéma zapojení Arduino	13
Obr. 10 Vytvořená aplikace za pomoci MIT App Inventor	14
Obr. 11 Vymodelování krabičky v Tinkercad	14
Obr. 12 Ukázka programovacího prostředí Arduino IDE 1.8.15	15
Obr. 13 Arduino-DIY hodinky	16
Obr. 14 Propojení displeje s Arduino NANO	23
Obr. 15 Propojení displeje s Arduino UNO	23
Obr. 16 Vložení Arduino do krabičky z Lega	23
Obr. 17 Hotový produkt	23
Obr. 18 První tisknutí krabičky + Aplikace na hodinky	23

9.2 Seznam tabulek

Tabulka 1 Technické parametry hodinek	16
Tabulka 2 Funkce hodinek	16

9.3 Zdrojové kódy (důležité části)

```
void setup() {  
    // inicializace pinu  
    pinMode(buttonOne, INPUT);  
    pinMode(buttonTwo, INPUT);  
    pinMode(bzucak, OUTPUT);  
    while (!bluetooth);  
    // zahájení komunikace s bluetooth pomocí seriové komunikace  
    bluetooth.begin(9600);  
}
```

Funkce void setup() se provede pouze jednou na začátku programu. Používá se k inicializaci konkrétních pinů a jejich režimů, k zahájení používání knihoven atd. Funkce setup() společně s funkcí loop() musí být v programu obsaženy vždy. Při jejich absenci by program skončil s chybou [1].

```

void checkOne(void) {
    byte buttonStateOne = 0;
    buttonStateOne = digitalRead(buttonOne);

    if (buttonStateOne != lastButtonStateOne) {
        if (buttonStateOne == LOW) {
            bPressOne = true;
            buttonPushCounterOne++;
        }
    }
    lastButtonStateOne = buttonStateOne;
}

```

Funkce, která nám hlídá, zda bylo tlačítko zmáčknuto.

```

if (millis() - prepis > 100) {
    mujOled.firstPage();
    do {
        switch (choice) {
            case 0:
                hodiny();
                break;
            case 1:
                stopky();
                break;
            case 2:
                budik();
                break;
            case 3:
                slovník();
                break;
            default:
                hodiny();
        }
    } while (mujOled.nextPage());
    prepis = millis();
}

```

Příkaz, který kontroluje rozdíl mezi aktuálním a posledním vykreslením displeje. Tělo tohoto příkazu obsahuje skupinu příkazů, které zaručují překreslení displeje. Jeho součástí jsou podprogramy, které jsou volány.

```

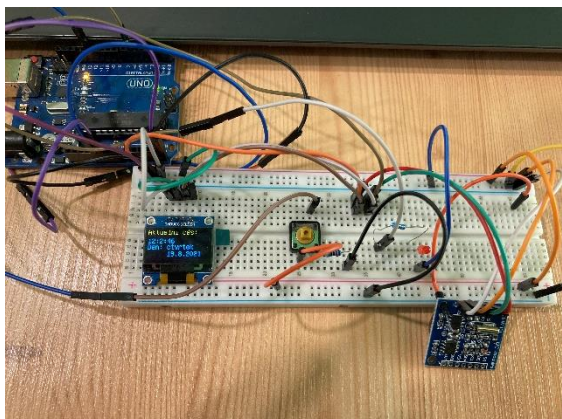
void hodiny(void) {
    tmElements_t tm;
    mujOled.setFont(u8g_font_unifont);
    checkThree();

    if (RTC.read(tm)) {
        mujOled.setPrintPos(0, 10);
        mujOled.print("Aktualni cas: ");
        mujOled.setPrintPos(0, 27);
        print2digits(tm.Hour);
        mujOled.print(":");
        print2digits(tm.Minute);
        mujOled.print(":");
        print2digits(tm.Second);
        mujOled.setPrintPos(0, 40);
        mujOled.print("Den:");
        print2digits(tm.Day);
        mujOled.setPrintPos(8, 55);
        if (choiceTwo == 0) {
            cas_nyni = millis();
            blinkStopky = (int)((cas_nyni - last_time) / 1000);
            if (blinkStopky % 2 == 0) mujOled.print("t");
            else mujOled.print("");
        }
        mujOled.setPrintPos(0, 55);
        if (bPressFour) {
            mujOled.print("B");
            if ((buttonPushCounterFour > 1) && (choice == 0)) {
                bPressFour = false;
                digitalWrite(bzucak, LOW);
                buttonPushCounterFour = 0;
            }
        }
        if ((buttonPushCounterH == tm.Hour) && (buttonPushCounterM == tm.Minute) &&
bPressFour) {
            current_rozdil = millis();
            b_melodie(melodie);
        }
        mujOled.setPrintPos(40, 55);
        print2digits(tm.Day);
        mujOled.write('.');
        print2digits(tm.Month);
        mujOled.write('.');
        print2digits(tmYearToCalendar(tm.Year));
    }
}

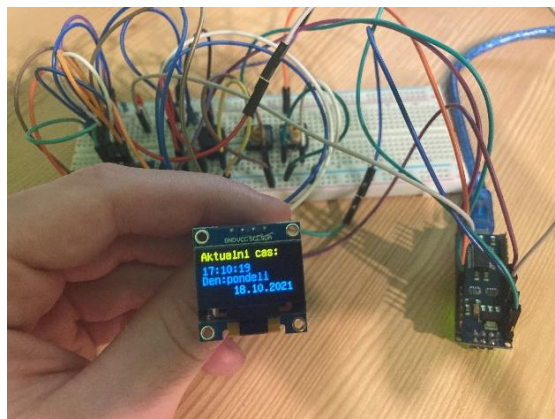
```

Funkce hodiny() nám vykresluje čas, datum, zapnuté a vypnuté stopky/budík.

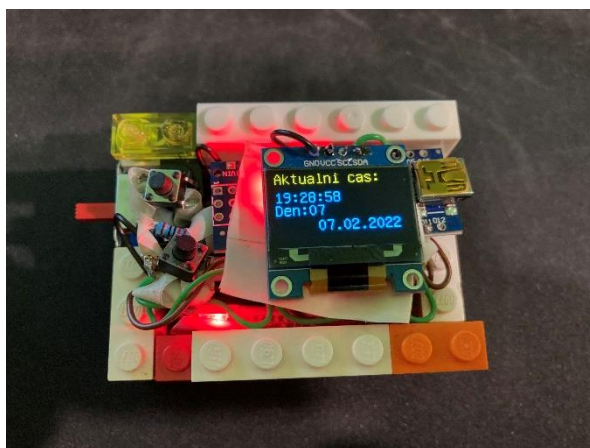
10 Dokumentace



Obr. 15 Propojení displeje s Arduino UNO



Obr. 14 Propojení displeje s Arduino NANO



Obr. 16 Vložení Arduina do krabičky z Lega



Obr. 18 První tisknutí krabičky + Aplikace na hodinky



Obr. 17 Hotový produkt