

**STREDNÁ PRIEMYSELNÁ ŠKOLA JOZEFA MURGAŠA**

**Hurbanova 6, 975 18 Banská Bystrica**

**AUTOMATIZOVANÉ OVLÁDANIE ÚSTREDNÉHO KÚRENIA**

Stredoškolská odborná činnosť

Č. odboru : 11

Banská Bystrica

2023

Riešiteľ: Andrej Bočkaj

Ročník štúdia : štvrtý

**STREDNÁ PRIEMYSELNÁ ŠKOLA JOZEFA MURGAŠA**

**Hurbanova 6, 975 18 Banská Bystrica**

**AUTOMATIZOVANÉ OVLÁDANIE ÚSTREDNÉHO KÚRENIA**

Stredoškolská odborná činnosť

Č. odboru : 11

Banská Bystrica

2023

Riešiteľ: Andrej Bočkaj

Ročník štúdia : štvrtý

Školiteľ : Ing. Vladimír Perešíni

## **Čestné vyhlásenie o originalite súťažnej práce SOČ**

Čestne prehlasujem, že svoju súťažnú prácu SOČ „Automatizované ovládanie ústredného kúrenia“ som neprihlásil a neprezentoval v žiadnej inej súťaži , gestorom ktorej je Ministerstvo školstva, vedy, výskumu a športu Slovenskej republiky.

V Banskej Bystrici    dňa 10.3.2023

.....

podpis autora práce

# OBSAH

<b>ZOZNAM SKRATIEK, ZNAČIEK A SYMBOLOV .....</b>	<b>7</b>
<b>ZOZNAM OBRÁZKOV, TABULIEK A GRAFOV .....</b>	<b>8</b>
<b>ÚVOD .....</b>	<b>9</b>
<b>1 TEORETICKÉ VÝCHODISKÁ .....</b>	<b>10</b>
1.1 Regulačný obvod .....	10
1.2 Dvojpohodová regulácia .....	10
1.3 PD regulátor .....	10
1.3.1 <i>Proporcionálny regulátor</i> .....	10
1.3.2 <i>Derivačný regulátor</i> .....	10
1.4 SPI.....	11
1.5 JSON.....	11
1.6 PWM.....	11
1.7 I <sup>2</sup> C .....	12
1.8 Komunikačný modul ESP32 .....	12
1.9 Firebase .....	12
1.10 ESP-NOW protokol .....	13
1.10.1 <i>Zabezpečenie</i> .....	13
1.11 UART .....	14
1.12 Snímače .....	14
<b>2 CIELE PRÁCE.....</b>	<b>15</b>
<b>3 MATERIÁL A METODIKA PRÁCE .....</b>	<b>16</b>
3.1 Prvé verzie DPS .....	16
3.1.1 <i>Ovládacia jednotka ústredného kúrenia</i> .....	16
3.1.1.1 <i>Návrh Schémy</i> .....	16
3.1.1.2 <i>Testovanie</i> .....	17
3.1.1.3 <i>Realizácia prototypu</i> .....	17
3.1.1.4 <i>Oživovanie prototypu</i> .....	17
3.1.1.5 <i>Návrh DPS</i> .....	17
3.1.1.6 <i>Oživovanie DPS</i> .....	18

3.1.2	<i>Termostaty</i> .....	18
3.1.2.1	<i>Návrh Schémy</i> .....	18
3.1.2.2	<i>Realizácia a oživovanie prototypu</i> .....	19
3.1.2.3	<i>DPS</i> .....	19
3.2	<i>Súčasná verzia DPS</i> .....	19
3.2.1	<i>Ovládacia jednotka ústredného kúrenia</i> .....	19
3.2.1.1	<i>Návrh Schémy</i> .....	19
3.2.1.2	<i>Návrh DPS</i> .....	20
3.2.1.3	<i>Oživovanie</i> .....	20
3.2.2	<i>Termostaty</i> .....	21
3.2.2.1	<i>Návrh Schémy</i> .....	21
3.2.2.2	<i>Návrh DPS</i> .....	21
3.2.2.3	<i>Oživovanie</i> .....	21
3.3	<i>Objednanie DPS</i> .....	21
3.4	<i>Návrh zdrojového kódu</i> .....	22
3.4.1	<i>PLC</i> .....	22
3.4.2	<i>Štruktúra</i> .....	22
3.4.2.1	<i>Vstupy a výstupy</i> .....	22
3.4.2.2	<i>Logika</i> .....	23
3.4.2.3	<i>Parametre výhrevných okruhov</i> .....	23
3.4.2.4	<i>Vizuálne nástroje</i> .....	23
3.4.2.5	<i>HMI</i> .....	23
3.4.3	<i>Algoritmy</i> .....	23
3.4.3.1	<i>Hysteréza</i> .....	23
3.4.3.2	<i>PD</i> .....	24
3.4.3.3	<i>Intervaly vykurovania</i> .....	24
3.4.3.4	<i>Časovač (Timer)</i> .....	25
3.4.3.5	<i>Dotyk</i> .....	25
3.4.3.6	<i>Posielanie informácií ESP-NOW</i> .....	26
3.4.3.6.1	<i>Termostaty</i> .....	26
3.4.3.6.2	<i>Ovládacia jednotka</i> .....	26
3.4.3.7	<i>Ukladanie do pamäte flash</i> .....	27

3.4.3.8	<i>Jazyk</i>	27
3.4.4	<i>Princíp</i>	27
3.4.4.1	<i>Ovládacia jednotka ústredného kúrenia</i>	27
3.4.4.2	<i>Termostaty</i>	28
3.5	<i>Mechanická konštrukcia</i>	28
3.5.1	<i>SolidWorks a 3D tlač</i>	28
3.5.2	<i>Prvé verzie</i>	29
3.5.2.1	<i>Ovládacia jednotka ústredného kúrenia</i>	29
3.5.2.2	<i>Termostaty</i>	29
3.5.3	<i>Súčasná verzia</i>	29
3.5.3.1	<i>Ovládacia jednotka ústredného kúrenia</i>	29
3.5.3.2	<i>Termostaty</i>	29
3.6	<i>Vytvorenie webovej aplikácie</i>	30
3.6.1	<i>Databáza</i>	30
3.6.2	<i>Webové rozhranie</i>	30
3.6.3	<i>Arduino IDE</i>	30
4	<b>VÝSLEDKY PRÁCE A DISKUSIA</b>	31
4.1	<i>Výsledky</i>	31
4.2	<i>Zistenia</i>	31
4.3	<i>Pozorovania</i>	31
4.4	<i>Diskusia</i>	31
5	<b>ZÁVER PRÁCE</b>	32
5.1	<i>Splnenie cieľov</i>	32
5.2	<i>Využitie v praxi</i>	32
5.3	<i>Zlepšenie</i>	32
	<b>ZHRNUTIE</b>	33
	<b>ZOZNAM POUŽITEJ LITERATÚRY</b>	34
	<b>ZOZNAM PRÍLOH</b>	36

## **ZOZNAM SKRATIEK, ZNAČIEK A SYMBOLOV**

**CCMP** - Counter Mode with Cipher Block Chaining Message Authentication Code Protocol.

**DPS** – Doska plošných spojov

**PWM** - Pulse width modulation

**ADC** – Analog to Digital Convertor

**DAC** – Digital to Analog Convertor

**SPI** – Serial peripheral interface

**I2C** - Inter-Integrated Circuit

**JSON** - JavaScript Object Notation

**TFT** - Thin Film Transistor

**EAGLE** - Easily Applicable Graphical Layout Editor

**UART** - universal asynchronous receiver-transmitter

**DC** - direct current

**AC** - alternating current

**SMD** - surface mount devices

**RTC** – Real time clock

**ULR** - Uniform Resource Locator

## **ZOZNAM OBRÁZKOV, TABULIEK A GRAFOV**

<i>OBRÁZOK 1 HYSTERÉZA (13)</i> .....	10
<i>OBRÁZOK 2 SPI PRINCÍP (4)</i> .....	11
<i>OBRÁZOK 3 SPI ZAPOJENIE (4)</i> .....	11
<i>OBRÁZOK 4 JSON (5)</i> .....	11
<i>OBRÁZOK 5 PWM (14)</i> .....	11
<i>OBRÁZOK 6 I2C ZAPOJENIE (15)</i> .....	12
<i>OBRÁZOK 7 I2C (16)</i> .....	12
<i>OBRÁZOK 8 ESP32 (8)</i> .....	12
<i>OBRÁZOK 9 FIREBASE (9)</i> .....	12
<i>OBRÁZOK 10 ESP-NOW ŠPECIFICKÝ AKČNÝ RÁMEC (10)</i> .....	13
<i>OBRÁZOK 11 ESP-NOW KONTENT (10)</i> .....	13
<i>OBRÁZOK 12 ESP-NOW (17)</i> .....	13
<i>OBRÁZOK 13 UART (11)</i> .....	14
<i>OBRÁZOK 14 UART RÁMEC (11)</i> .....	14
<i>OBRÁZOK 15 SENZOR (18)</i> .....	14
<i>OBRÁZOK 16 PRAKTICKÁ REALIZÁCIA PLC CYKLU V ARDUINO IDE</i> .....	22
<i>OBRÁZOK 17 PRAKTICKÁ REALIZÁCIA HYSTERÉZY V PROGRAME</i> .....	24
<i>OBRÁZOK 18 GRAFICKÉ ZNÁZORNENIE FUNKCIE PD ALGORITMU</i> .....	24
<i>OBRÁZOK 19 PRAKTICKÁ REALIZÁCIA PD REGULÁTORA V PROGRAME</i> .....	25
<i>OBRÁZOK 20 GRAFICKÉ ZNÁZORNENIE PRINCÍPU KOMUNIKÁCIE</i> .....	26



## ÚVOD

Na začiatku bola zima. Teda až do doby, kým človek neobjavil oheň a teplo z nebo vyžarujúce. Vznikali prvé zdroje tepla určené k vykurovaniu, ako napríklad otvorené ohniská. Postupom času dochádzalo k čoraz väčšej túžbe cieľavedome kontrolovať oheň, čo viedlo k zrodeniu krbu. Rovnako ako všetky súčasné zariadenia podľahli modernizácií, tak aj vykurovanie sa postupne vyvíjalo až do dnešnej doby.

Súčasný systémy na reguláciu a ovládanie výhrevných zariadení v domácnosti dokážu priniesť používateľovi dostatočný komfort. Existuje množstvo variant izbových termostátov, ako napríklad elektrické, analógové alebo digitálne.

Analógové termostaty patria k tým starším a lacnejším. Môžu fungovať na 2 princípoch. Prvým princípom, ktorým je dnes využívaný častejšie, je bimetalický prúžok. Prúžok sa pod vplyvom tepla ohýba, čím pri zmenách teploty zapína a prepína obvody v termostate a vykurovacom telese. Analógové termostaty dokážu teplotu iba regulovať.

Digitálne termostaty snímajú a následne regulujú teplotu na základe informácií v digitálnej forme v závislosti na teplote okolia. Rovnako bývajú vybavené farebným dotykovým displejom, ktorý umožňuje intuitívne a jednoduché ovládanie bez nutnosti stáleho študovania príručky. Zväčša majú možnosť pripojenia na lokálnu sieť, čo umožní užívateľovi prístup k aplikácii, ktorá dokáže detailne zobrazovať charakteristiku vykurovania podľa zadaných parametrov.

Aktuálna situácia je z pohľadu energetickej náročnosti osudová, ohľadom na financie a životné prostredie. Inteligentné vykurovanie je čoraz potrebné v rodinných domoch alebo bytoch s necentralizovaným vykurovacím systémom. So správnymi nastaveniami vykurovania je možné ušetriť množstvo financií.

Cieľom práce bolo vytvoriť zariadenie, ktoré bude intuitívne ovládať dva výhrevné okruhy ústredného kúrenia využitím dvoch bezdrôtových termostátov na meranie teploty v daných okruhoch. Zariadenie musí byť jednoduché na manipuláciu a rovnako nenáročné na ovládanie pri nastavovaní parametrov vyhrievania. Hlavná činnosť, ktorú musí zariadenie zabezpečiť je bezdrôtová komunikácia medzi termostatmi a poskytovať webové rozhranie umožňujúce ovládať parametre vykurovania z ľubovoľného miesta na zemi.

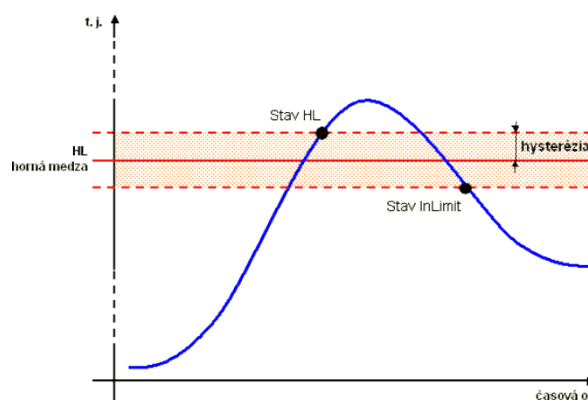
# 1 TEORETICKÉ VÝCHODISKÁ

## 1.1 Regulačný obvod

Je model obvodu ktorý popisuje správanie sa regulovanej sústavy na základe regulátora. Regulátor porovnáva skutočnú a žiadanú hodnotu regulovanej veličiny a v závislosti od regulačnej odchýlky ovplyvňuje akčnú veličinu ktorá riadi regulovanú sústavu. (1)

## 1.2 Dvojpolohová regulácia

Dvojpolohová regulácia využíva len krajné hodnoty meranej veličiny. Výstupný signál z regulátora je digitálny signál ktorý sa preklápa pri dosiahnutí krajnej hodnoty. Využíva sa hlavne pri jednoduchších objektoch riadenia napríklad výhrevné zariadenia a špirály. Nevýhodou dvojpolohovej regulácie je časté spínanie kontaktov a mechanických častí regulátora. Na obmedzenie častého spínania sa do obvodu zavádza hystereza. Pri regulácii pomocou hysterézy sa nereguluje objekt riadenia na konštantnú hodnotu ale na určený rozsah hodnôt. Pásmi pri ktorom sa výstupný signál nemení sa nazýva pásmo necitlivosti (deadband). (2)



OBRÁZOK 1 HYSTERÉZIA (13)

## 1.3 PD regulátor

Vzniká paralelným zapojením proporcionálneho a derivačného regulátora. Zlepšuje stabilitu sústavy a vyhovuje pre astatické sústavy. (3)

### 1.3.1 Proporcionálny regulátor

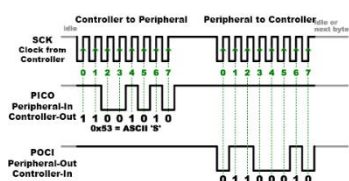
Pracuje so stálou regulačnou odchýlkou. Regulátor zhoršuje stabilitu systému no rýchlo približuje aktuálny stav regulovanej sústavy na požadovaný stav. Pri využití nastáva zvýšenie prekmitu sústavy.

### 1.3.2 Derivačný regulátor

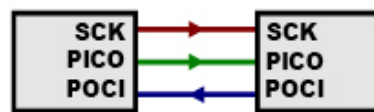
Deriváciou funkcie sústavy získame smernicu dotýčnice ku grafu funkcie sústavy. Čím rýchlejšie funkcia sústavy rastie, tým väčšia bude hodnota derivácie. Derivácia bude mať zápornú hodnotu v mieste, kde bude graf funkcie sústavy klesať. Regulátor má pri konštantnom vstupe nulový prenos z dôvodu žiadnej zmeny stavu.

## 1.4 SPI

Sériové periférne rozhranie je synchronné sériové periférne rozhranie, primárne využívané v rámci vstavaných systémov pre komunikáciu na krátke vzdialenosti medzi riadiacimi mikrokontrolérmi a ostatnými integrovanými obvodmi resp. modulmi. obojsmerná full-duplex komunikácia prebieha na spoločnej zbernici, adresácia obvodov je riešená prostredníctvom samostatných vodičov, ktoré zmenou stavu na logickú nulu aktivujú príjem a vysielanie zvoleného zariadenia. (4)



OBRÁZOK 2 SPI PRINCÍP (4)



OBRÁZOK 3 SPI ZAPOJENIE (4)

## 1.5 JSON

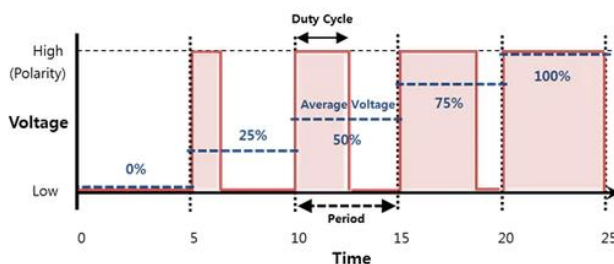
JSON je jednoduchý textový formát na ukladanie a výmenu údajov, ktorý používa ľudsky čitateľný text. Je úplne nezávislý na jazyku, ale používa konvencie, ktoré poznajú programátori z rodiny jazykov C, C++, C#, Java, JavaScript. (5)

```
'{"name": "John", "age": 30, "car": null}'
```

OBRÁZOK 4 JSON (5)

## 1.6 PWM

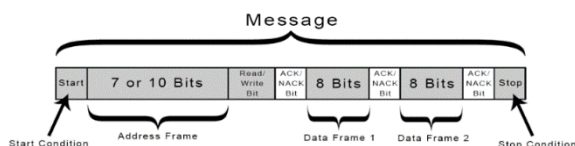
Modulácia je nelineárny elektrický proces, pri ktorom sa parametre jedného signálu (nosného) menia iným signálom (modulačným, informačným). Pri šírkovopulznej modulácii zostávajú amplitúda, frekvencia a fáza pôvodného signálu nezmenené. Trvanie (šírka) štvorcového impulzu je ovplyvnené informačným signálom. V anglickej technickej literatúre sa označuje skratkou PWM - pulse-width modulation. (6)



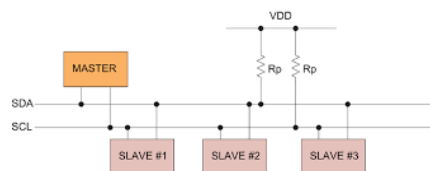
OBRÁZOK 5 PWM (14)

## 1.7 I<sup>2</sup>C

I<sup>2</sup>C je dvojvodičové obojsmerné synchronné sériové periférne rozhranie, primárne využívané v rámci vstavaných systémov pre komunikáciu na krátke vzdialenosti medzi radiacimi mikrokontrolérmi a ostatnými integrovanými obvodmi resp. modulmi. Zbernica umožňuje, v závislosti od pripojených zariadení, komunikáciu prenosovými rýchlosťami od 100 kbps až po 5 Mbps a zariadenia sú adresované 7-bitovými ( $2^7 = 128$  adres) prípadne 10-bitovými ( $2^{10} = 1\,024$  adres) adresami. (7)



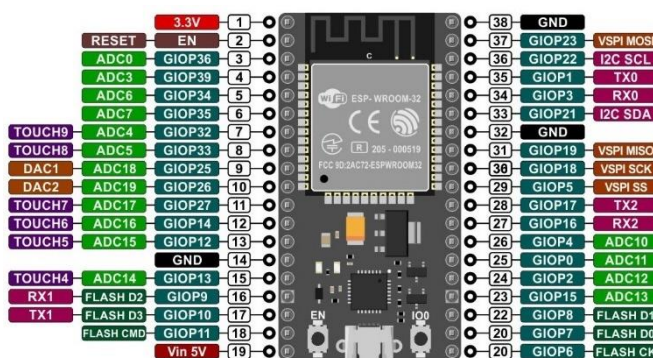
OBRÁZOK 7 I2C (16)



OBRÁZOK 6 I2C ZAPOJENIE (15)

## 1.8 Komunikačný modul ESP32

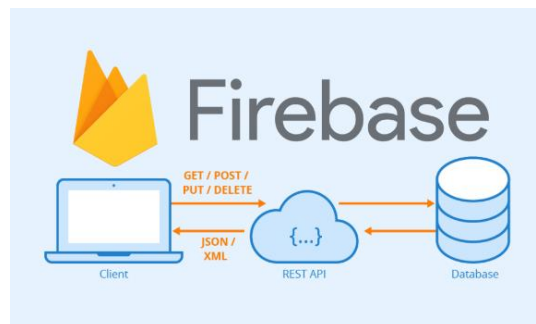
Komunikačný modul ESP32 je séria cenovo dostupných a energeticky nenáročných systémov na jednom čipe vyvinutých spoločnosťou Espressif ktoré obsahujú Wi-Fi a Bluetooth. Podporujú veľké množstvo periférnych vstupných a výstupných rozhraní ako napríklad ADCs a DACs, SPI, I<sup>2</sup>C a PWM. (8)



OBRÁZOK 8 ESP32 (8)

## 1.9 Firebase

Google Firebase je softvér na vývoj aplikácií podporovaný spoločnosťou Google, ktorý umožňuje vývojárom vyvíjať aplikácie pre iOS, Android a webové aplikácie. Firebase poskytuje nástroje na sledovanie analytiky, nahlasovanie a opravu zlyhaní aplikácií, vytváranie marketingových a produktových experimentov. (9) (10)



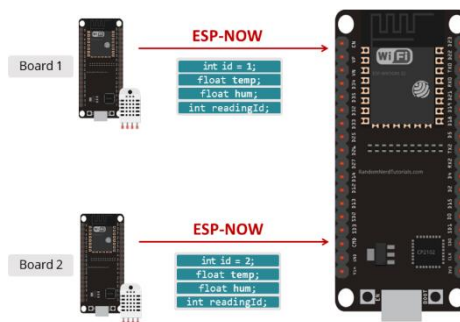
OBRÁZOK 9 FIREBASE (9)

## 1.10 ESP-NOW protokol

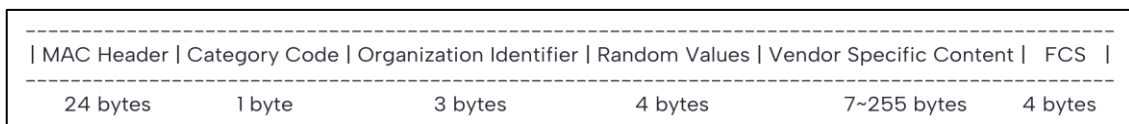
ESP-NOW je rýchla, bezdrôtová komunikačná technológia vytvorená na prenos dát. Informácie určené na prenos sa odosielajú a prijímajú prostredníctvom štruktúry vytvorenej v zdrojovom programe. Dátová štruktúra je zapuzdrená do špecifického akčného rámca a odoslaná bez použitia WI-FI. Odosielanie štruktúr sa vykonáva prostredníctvom protokolu podobný energeticky-úspornému bezdrôtovému pripojeniu 2.4GHz ktorý sa často využíva pri bezdrôtových myšiach. Predvolená bitová rýchlosť je 1Mbps. (10) (12)

### 1.10.1 Zabezpečenie

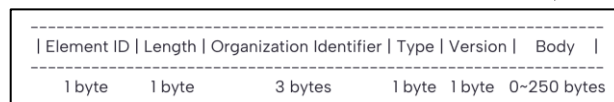
ESP-NOW umožňuje využívať CCMP protokol ktorý dokáže chrániť špecifický akčný rámec. CCMP protokol bol navrhnutý pre ochranu bezdrôtovej LAN. Zariadenie má primárny kľúč (PMK) a niekoľko lokálnych kľúčov (LMK) ktorých dĺžka je 16bajtov.



OBRÁZOK 12 ESP-NOW (17)



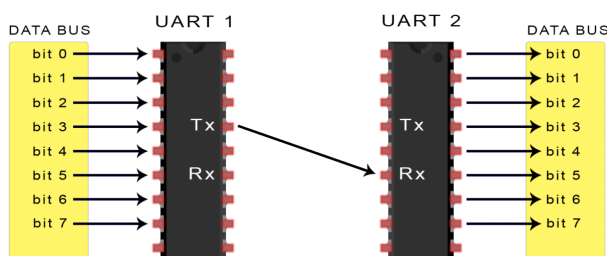
OBRÁZOK 10 ESP-NOW ŠPECIFICKÝ AKČNÝ RÁMEC (10)



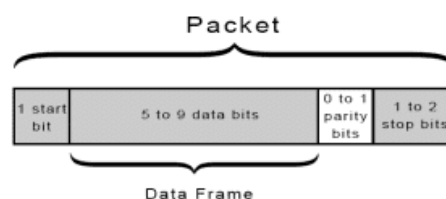
OBRÁZOK 11 ESP-NOW KONTENT (10)

## 1.11 UART

Je integrovaný obvod, ktorý využíva na prenos dát dva vodiče. Nepodobá sa komunikačným protokolom SPI a I2C pretože na jeho funkciu treba fyzický integrovaný obvod. Dáta sú prenášané z paralelnej dátovej zbernice bez potrebného hodinového signálu rýchlosťami od 9600 baud po maximálnu prenosovú rýchlosť 115200 baud. Komunikácia odosiela dáta v špecifickom dátovom rámci. Štartovací a končiaci byt plus parity bit, ktorý nesie informáciu o chybe pri komunikácii. (11)



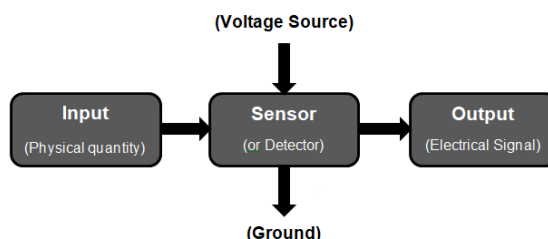
OBRÁZOK 13 UART (11)



OBRÁZOK 14 UART RÁMEC (11)

## 1.12 Snímače

Jedným z členov celej regulačnej sústavy sú snímače. Ich úlohou je premena fyzikálnej veličiny (neelektrického vstupného signálu) na výstupný elektrický signál, prípadne zmena niektorej elektrickej vlastnosti snímača. Najčastejšie ide o zmenu impedancie, odporu, kapacity prípadne indukčnosti. Podľa typu prevodovej funkcie rozdeľujeme snímače na lineárne a nelineárne. Podľa druhu interakcie, teda vzájomného pôsobenia s meraným objektom na kontaktné a bezkontaktné. Podľa spôsobu prevodu nameranej veličiny snímače delíme na priame a nepriame. Snímače v priemyselných prevádzkach sú často vystavené okrem vplyvu meranej veličiny aj vedľajším nežiaducim vplyvom okolitého prostredia. Je preto nevyhnutné pred inštaláciou snímača poznať prostredie v ktorom bude inštalovaný a následne vybrať snímač vhodného prevedenia. (12)



OBRÁZOK 15 SENZOR (18)

## 2 CIELE PRÁCE

Našou úlohou bolo vytvoriť zariadenie, ktoré bude automaticky ovládať dva výhrevné okruhy ústredného kúrenia využitím dvoch bezdrôtových termostátov na meranie teploty v daných okruhoch. Chceli sme, aby zariadenie bolo jednoduché na manipuláciu a rovnako nenáročné na ovládanie pri nastavovaní parametrov vyhrievania.

Na dosiahnutie cieľa sme zvolili komunikačný modul ESP32, ktorý je energeticky nenáročný a dostatočne výkonný pre všetky požadované úkony. Hlavná činnosť, ktorú musel komunikačný modul zabezpečiť bola bezdrôtová komunikácia medzi komunikačnými modulmi. Po výbere správneho komunikačného modulu nasledovalo vhodné navrhnutie dosiek plošných spojov, ktoré by boli minimálne energeticky náročné za účelom dlhej výdrže batérie a vhodnej veľkosti, aby bol termostat kompaktný. Rovnako bolo nutné navrhnuť dosku plošných spojov pre riadiacu jednotku, ktorá bude schopná riadiť dva výhrevné okruhy ústredného kúrenia prostredníctvom vhodných relé. Pri návrhoch bolo nutné zachovať rozmiestnenie pinov pre sériové periférne rozhranie na displej rovnaké, aby bolo možné naprogramovať komunikačné moduly bez úpravy používateľskej knižnice určenej pre TFT displeje.

Po výrobe dosiek plošných spojov bolo potrebné vytvoriť hlavnú logiku spínania výhrevných okruhov. V jednom cykle komunikačný modul nahrá všetky vstupy. Po nahraní vstupov sa vykoná logický vyhodnocovací algoritmus, ktorý po jeho skončení dá na výstupy dané hodnoty. Z dôvodu umožnenia upravovania parametrov logického vyhodnocovacieho algoritmu sa stalo nevyhnutným vytvorenie rozhrania, cez ktoré bude možné jednoducho upravovať parametre výhrevných okruhov. Na vykreslenie údajov na displej sme zvolili knižnicu, ktorá podporovala komunikačný prevodník na displeji. Na vytvorenie jednoduchého používateľského rozhrania sme museli vytvoriť s elementárnych objektov sústavu, z ktorej sa skladal jeden nástroj. Pomocou nástrojov sme boli schopný vytvoriť prehľadné strany s informáciami a aj strany na úpravu parametrov výhrevných okruhov.

Aby bolo možné upravovať nastavenia automatizovaného ovládania ústredného kúrenia bez prítomných ovládacích jednotiek, bolo potrebné využiť databázu firebase. Tá nám umožnila prístup k ovládaniu a sledovaniu automatizovaného ovládania ústredného kúrenia z ľubovoľného miesta na zemi len pomocou mobilných dát alebo WI-FI.

### **3 MATERIÁL A METODIKA PRÁCE**

Prvé kroky na vytvorenie automatizovaného riadenia ústredného kúrenia pozostávali zo správneho výberu prvkov, elektronických zariadení a súčiastok.

#### **3.1 Prvé verzie DPS**

Na začiatku sme začali experimentovať so samostatnými modulmi. Hlavne s komunikačným modulom ESP32, ktorý sme si zvolili ako jadro zariadení. Podstatou bolo pochopenie princípu činnosti spolu s praktickou realizáciou požadovaných protokolov, ako je napríklad komunikačná technológia ESP-NOW.

##### **3.1.1 Ovládacia jednotka ústredného kúrenia**

###### **3.1.1.1 Návrh Schémy**

Pre postupný vývoj zariadenia sme museli ako prvé navrhnuť elektrotechnickú schému, z ktorej sme mohli ďalej vychádzať. Pri návrhu sme dbali na výber vhodných elektronických zariadení a súčiastok s ohľadom na cenu a kvalitu.

Hlavný obvod ktorý sme museli vytvoriť bolo zapojenie s relé. Zvažovali sme použitie polovodičových relé z dôvodu nízkej hlučnosti a malých rozmerov. Napokon sme sa rozhodli použiť elektromagnetické relé, ktoré najlepšie vyhovovalo menovitému napätiu a prúdu riadiacich signálov ústredného kúrenia. Aby sme boli schopný bezpečne ovládať zvolené elektromagnetické relé vyžadovalo sa zapojenie obvodu s NPN tranzistorom. Priviedli sme výstup cez rezistor z komunikačného modulu na bázu tranzistora. Kolektor sme spojili s výstupným kontaktom cievky a emitor pripojili na vetvu GND. Rovnako bolo potrebné paralelne zapojiť schotkyho diódu k cievke relé tak aby bola v závernom smere k privedenému napätiu 5 V. Tým sme zamedzili možnému poškodeniu elektronických súčiastok pred často vznikajúcim spätným indukovaným napätím vytvoreným rozopnutou cievkou.

Aby sme boli schopný využívať intervaly na vykurovanie, museli sme zariadenie obstarat' hodinovým RTC modulom. Zvolili sme hodinový modul s komunikáciou I2C v kolmom zapojení kontaktov na DPS modulu. Pre vloženie modulu do elektrotechnickej schémy sme si navrhli vlastnú knižnicu pre hodinový modul. Vytvorili sme schematickú značku a podľa rozmerov z datasheetu vytvorili aj footprint. Pripojili sme konkrétne kontakty zo schematickej značky s návrhom z footprintu a následne sme vytvorili súbor obsahujúci navrhnutú knižnicu. Vytvorenú knižnicu sme už len načítali v programe EAGLE a mohli sme ju začať používať.



Praktické užívateľské prostredie sme zabezpečili zapojením 2,8 palcového displeja cez SPI zbernicu ku komunikačnému modulu. Rozhodli sme sa použiť nedotýkový TFT displej a ako ovládaciú jednotku využiť inkrementálny rotačný enkodér.

Na napájanie sme využili 5 V DC zdroj z 230 V AC.

### **3.1.1.2 Testovanie**

Aby sme boli schopný navrhnuť prvú dosku plošných spojov museli sme vytvorené obvody otestovať pomocou kontaktného poľa. Potrebné bolo aby sme si obstarali všetky potrebné súčiastky spolu s vodičmi na prepojovanie. Obvody sme testovali oddelene aby nedošlo k neúmyselnému skratu a teda aj zničeniu súčiastky, ktoré by bolo zavinené našou chybou.

### **3.1.1.3 Realizácia prototypu**

Zapojenie všetkých zariadení a súčiastok súčasne sme zrealizovali pomocou prototypovej spájkovacej dosky. Bolo potrebné zaobstarat' si dostatočne veľkú dosku pre všetky potrebné súčiastky.

Osadili sme elektronické prvky, skontrolovali polaritu a začali sme spájkovačkou vytvárať cínové prepojenia zo spodnej strany prototypovej dosky. Riadili sme sa dopredu navrhnutou elektrotechnickou schémou.

Súčiastky neboli prispôsobené pre použitý typ dosky čo nám z veľkej miery skomplikovalo praktickú realizáciu. Nevhodné súčiastky sme museli upraviť ohnutím kontaktov aby pasovali do prototypovej dosky.

### **3.1.1.4 Oživovanie prototypu**

Po osadení súčiastok na prototypovú dosku sme začali testovať celkové zapojenie obvodu. Prototypová spájkovacia doska nám umožnila nefunkčne navrhnutú časť obvodu jednoducho opraviť alebo odstrániť.

Zistené nedostatky, ako napríklad chýbajúci pull-up rezistor pri rotačnom enkodéry, sme dopĺňovali do elektrotechnickej schémy.

### **3.1.1.5 Návrh DPS**

Pri návrhu dosky plošného spoja sme vychádzali z navrhnutých obvodov realizovaných na prototypovej doske. Zmeny oproti prototypu:

- Pridali sme do obvodu konektor pre ultrazvukový snímač
- Displej sme vyviedli mimo DPS.

- inkrementálny rotačný enkodér sme vyviedli mimo DPS.

### **3.1.1.6 Oživovanie DPS**

Funkčné súčiastky vložené v prototypovej doske sme museli odspájkovať, aby sme ich mohli znova využiť pri DPS.

Jedinou chybou návrhu boli zle zvolené tranzistory. Zistili sme, že rozloženie reálnych tranzistorových kontaktov nie je totožné s tranzistorovými kontaktami v schéme. Problém sme vyriešili obrátením oboch tranzistorov.

## **3.1.2 Termostaty**

### **3.1.2.1 Návrh Schémy**

Príjemné užívateľské prostredie závisí od použitého displeja. Z toho dôvodu sme sa rozhodli využiť dotykový Nextion displeje, ktorý podporuje vývojové prostredie na návrh vizualizácií. Displeje sme zapojili cez UART zbernicu ku komunikačnému modulu ESP32.

Aby bolo možné s termostatmi jednoducho manipulovať bolo potrebné vybaviť zariadenia batériou. Rozhodli sme sa využiť akumulátorový typ batérií s ohľadom na životné prostredie. Chceli sme cieľiť na kompaktnosť zariadenia, teda bolo vyžadované, aby batéria zaberala čo najmenej miesta, no zároveň ponúkala nemalú kapacitu pre dostatočnú výdrž na jedno nabitie. Typ Li-Pol batérií bol pre nás najlepšou voľbou. Batérie sú vyhotovené v plochom prevedení čo nám umožní vloženie batérie za displej. Nevýhodou bolo nízke výstupné napätie nepostačujúce použitému displeju. Aby bolo možné batériu vložiť pod displej nesmela obsahovať väčšie rozmery než diery na uchytenie displeja. Kapacita batérie závisela od rozmerov, ktoré sme boli schopný implementovať do zariadenia.

Kvôli nízkeho výstupnému napätiu batérie bolo potrebné napätie znásobiť. Sérioovo sme zapojili step-up menič k batérií a paralelnému obvodu displeja a stabilizátora na 3,3 V.

Ako posledné bolo náležité do schémy pridať teplotný snímač s malými rozmermi. Ten sme vyberali podľa presnosti snímania a spotreby elektrickej energie. Prepojenie medzi komunikačným modulom a teplotným snímačom sme realizovali rozhraním I2C.

### **3.1.2.2 Realizácia a oživovanie prototypu**

Pomocou schémy sme prepojili na kontaktnom poli modul komunikačného modulu ESP32 s dotykovým displejom cez UART zbernicu. Pokúšali sme sa o vzájomnú komunikáciu zariadení. Prvé pokusy skončili zlyhaním, z dôvodu zlého baud ratu. Následne po úprave na totožný baud rate komunikácie sme boli schopní využívať prepojenie.

Pokračovali sme v zapojení batérie so step-up meničom a snímačom teploty. Bolo potrebné aby sme odskúšali činnosť všetkých zariadení súčasne. Rovnako z hľadiska zdrojového kódu tak aj zapojenia.

### **3.1.2.3 DPS**

Prvý návrh sme vytvorili osadením modulov na DPS. Výsledkom bolo rozmerovo väčšie zariadenie, ktoré neponúkalo dostatočne veľkú funkcionálnosť.

## **3.2 Súčasná verzia DPS**

Po prvých pokusoch o vytvorenie funkčného zariadenia na ovládanie výhrevných okruhov ústredného kúrenia, sme boli schopní navrhnuť omnoho komplexnejšie zariadenie. Mali sme príležitosť vychádzať zo vzniknutých chýb prvých verzií. Časť zariadenia, ktoré boli dostatočne funkčné sme prevzali a zamerali sme sa viac na vylepšenie nedostatočne funkčných celkov. Rovnako sme sa rozhodli odstrániť kompletný vývojový modul ESP32 a nahradiť ho vlastným zapojením napájania a obvodov určených na programovanie samotného komunikačného modulu ESP32.

### **3.2.1 Ovládacia jednotka ústredného kúrenia**

#### **3.2.1.1 Návrh Schémy**

Oproti prvej verzii sme sa zbavili rotačného enkodéra a neprimerane malého displeja. Začali sme sa sústreďovať na jednoduchosť ovládania. Preto sme zvolili 4 palcový TFT dotykový displej. Prevodník na displeji podporoval výkonnú knižnicu pre vykresľovanie objektov na displej spolu so SPI rozhraním. V tomto prípade sme mali SPI zbernicu so 14 vodičmi. 3 vodiče boli zdieľané medzi dotykom a vykresľovaním.

Vymenili sme aj obvod s relé. Na prvej verzii zariadenia sa nachádzal klopný obvod pre signalizáciu napätového stavu. Obvod sme odstránili a rovnako aj použité relé, kvôli veľkej hlučnosti a rozmerom. Do schémy sme vložili menšie a menej hlučné relé. Zbavili sme sa aj napätového zdroja z 230 V AC na 5V DC.

Na napájanie sme sa rozhodli využiť externý zdroj pre väčšiu bezpečnosť a zmenšenie rozmerov. Externý zdroj taktiež ponúkal vyšší výstupný výkon.

Kvôli zväčšujúcej sa nepresnosti a možným častejším výpadkom, zapríčinených zvyšujúcou sa teplotou, sme museli pridať do zapojenia obvod na napájanie a riadenie aktívneho chladenia. Dvojpólový 5V ventilátor sme zapojili s vhodným NPN tranzistorom pre PWM riadenie.

### **3.2.1.2 Návrh DPS**

Najväčší dôraz sme kládli na kompaktnosť. Zariadenie muselo nadobúdať menšie rozmery ako diery pre uchytenie displeja. Návrh sme začali tvoriť ako obojstranné zapojenie DPS.

Pre jednoduchosť zapojenia sme museli komunikačný modul ESP32 osadiť zo spodnej strany DPS. Umožnilo nám to možnosť lepšie využiť rozloženie kontaktov, vďaka čomu sme mohli väčšinu SMD súčiastok vložiť pod hodinový RTC modul. Po usporiadaní súčiastok sme vytvorili dostatočne otvorený priestor pre pasívne chladenie zariadenia hliníkovým chladičom.

Požadovali sme aby bolo zariadenie z určitej miery symetrické. Rovnaké súčiastky sme sa snažili rozmiestniť totožne podľa stredu DPS.

Zabezpečenie jednoduchšej výmeny poškodenej externej periférie alebo jej výmeny za užívateľsky vhodnejšiu sme dosiahli modulárnym zapojením externých periférií.

### **3.2.1.3 Oživovanie**

Aby sme boli schopný spájať súčiastky SMD a komunikačný modul ESP32, bolo potrebné si zaobstaráť spájkovaciu cínovú pastu. Malý kus pasty sme naniesli na miesto spájkovania a stačilo len nahriať. Súčiastka sa jednoducho spojila s DPS.

Po spájkovaní sme testovali vodivosť zapojení elektronickým multimetrom. Zistili sme, že konektor na pripojenie displeja disponuje jedným nefunkčným kontaktom. Konektor sme museli odspájať a vymeniť za plnohodnotný.

Rovnako sa zistila chyba pri prerušení napájania. Zariadenie nebolo schopné spustiť nahraný program bez stlačenia tlačidla reset. Po odpojení a opätovnom pripojení napájania sa doska dostala do boot load módu. Chyba zapojenia spočívala vo vynechanom pull-up rezistore na kontakte komunikačného modulu ESP32.

## **3.2.2 Termostaty**

### **3.2.2.1 Návrh Schémy**

Prvé verzie neboli dostatočne vhodné pre dlhodobé využitie, z dôvodu malej výdrže na jedno nabitie. Zariadenia rovnako nepodporovali očakávanú vzdialenosť bezdrôtového prenosu dát.

Návrh vylepšenej elektronickej schémy obsahoval výmenu dotykového Nextion displeja za dotykový TFT displej totožný s vybratým displejom z ovládacej jednotky ústredného kúrenia. Aby sme docielili väčší dosah bezdrôtovej komunikácie, vybrali sme komunikačný modul s konektorom na pripojenie externej WI-FI antény. Potrebným bolo taktiež pridať do schémy load-sharing obvod pre nezávislé nabíjanie batérie a napájania zariadenia.

Do schémy sme rovnako zakomponovali obvod na napájanie a programovanie komunikačného modulu ESP32 z ovládacej jednotky ústredného kúrenia.

### **3.2.2.2 Návrh DPS**

Návrh kompaktného zapojenia spočíval v správnom rozložení súčiastok. Komunikačný modul sme osadili zo spodnej strany rovnako ako pri návrhu DPS pre ovládaciu jednotku ústredného kúrenia. Vyviedli sme kontakty pre externé periférie ako napríklad tlačidlo a spínač pre možnosť vypnutia zariadenia.

Vhodné rozmery DPS záviseli od veľkosti teplotného snímača a displeja. DPS sa spolu s teplotným snímačom museli rozmermi zmestiť pod vybraný displej. Myslieť sa muselo aj na medzi priestor na teplotnú izoláciu obvodov.

### **3.2.2.3 Oživovanie**

Na doske sa nachádzali len SMD súčiastky, ktoré sme spájkovali spájkovacou cínovou pastou. Daný nedostatok sa zistil len pri chybné navrhnutom obvode operačného zosilňovača s priamou spätnou väzbou. Obvod mal zabezpečovať sledovanie vstupného napätia bez odberu elektrického prúdu.

## **3.3 Objednanie DPS**

Na výrobu DPS sme zvolili spoločnosť JLCPSB zo sídlom v Číne. Aby nám mohla spoločnosť vyrobiť navrhnuté DPS, bolo potrebné vygenerovať gerber súbor. Pomocou vývojového programu EAGLE na vytváranie elektronických schém a DPS sme vygenerovali potrebný gerber súbor jednoduchým klikom.

### 3.4 Návrh zdrojového kódu

Vytvorenie používateľsky jednoduchého prostredia spolu s bezporuchovým návrhom systému, vyžadovalo premyslenú štruktúru zdrojového kódu. Pri neustálom upravovaní, menení a zväčšovaní programu sme museli dodržiavať určité konvencie. Nastolili sme si preto vlastné udržiavané pravidlá, ktoré nám zaručili jednoduchú manipuláciu s programom.

#### 3.4.1 PLC

Inšpiráciou navrhnutia plynulého chodu zariadenia bolo pochopenie princípu činnosti PLC. Toto zariadenie funguje na zákonitosti:

- I. Čítanie hodnôt všetkých vstupov
- II. Logické vyhodnotenie vstupov podľa programu
- III. Priradenie hodnôt na výstup

```
void loop() {  
    hardware.Inputs();  
    main_logic.fcMain();  
    hardware.Outputs();  
}
```

OBRÁZOK 16 PRAKTICKÁ REALIZÁCIA  
PLC CYKLU V ARDUINO IDE

#### 3.4.2 Štruktúra

Prvé pokusy o vytvorenie funkčnej štruktúry programu sme museli mnohonásobne meniť z dôvodu zložitosti či nespoľahlivosti. Až po určitom čase sa nám podarilo vytvoriť systém, ktorý nám pri programovaní umožnil obrovskú prehľadnosť zdrojového kódu. Pozri prílohu A.

##### 3.4.2.1 Vstupy a výstupy

Pre jednoduché čítanie zo vstupov a rovnako aj priradzovanie hodnôt na výstup komunikačného modulu sme vytvorili triedu Hardware. Pomocou tejto triedy sme schopný priradzovať určité funkcie na požadované kontakty komunikačného modulu. Trieda Hardware pozostáva z troch hlavných metód:

- I. Setup() – metóda, v ktorej sa všetky periférie nastavujú, aby splňovali požadované vlastnosti s použitým hardwarom.
- II. Inputs() – metóda na čítanie alebo meranie požadovanej periférie
- III. Outputs() – metóda na priradzovanie určitého stavu ku konkrétnemu kontaktu

Taktiež sme triedu Hardware obstarali funkciami na čítanie zo vstupov pre konkrétne objekty podľa nastaveného časového intervalu. Pri výstupoch sme pridali do programu funkcie, ktoré priradia logicky vyhodnotený stav na nastavený kontakt.

### **3.4.2.2 Logika**

V logickej časti volanej funkciou `fcMain`, bolo vyžadované len logické vyhodnocovanie, či zdieľanie informácií. Pre prehľadnosť sme zaviedli konvenciu prístupovania k požadovaným funkciám len prostredníctvom objektov na to určených. Funkcia `fcMain` sa stará o správnu činnosť programu. Pri vzniku chyby je vo funkcii nastavený chybový stav, ktorý zaručí odpojenie ovládacích signálov pre ochranu riadeného zariadenia.

### **3.4.2.3 Parametre výhrevných okruhov**

Na zabezpečenie správneho ukladania parametrov výhrevných okruhov ústredného kúrenia sme vytvorili triedu `Oven`. Triedu sme zapuzdрили, čo zabráni nechcenému chybnému zápisu do konkrétneho parametra. Rovnako sme v triede vytvorili metódy na pre vyhodnocovanie spínania určitého okruhu. K triede sme priradili vlastný hardware, ktorým dokáže podľa parametrov ovládať riadiace signály výhrevných okruhov ústredného kúrenia.

### **3.4.2.4 Vizuálne nástroje**

Pre možnosť jednoduchého nastavovania a ovládania výhrevných okruhov sme vytvorili skupinu vizuálnych nástrojov, pomocou ktorých je používateľ schopný intuitívne vyvolávať naprogramované akcie. Pod pojmom vizuálne nástroje myslíme softwarové tlačidlá, posúvače, textové objekty a objekty s ikonami. Vizuálne nástroje sme vytvárali prostredníctvom „TFT\_eSPI.h“ knižnice, ktorá podporuje funkciu `sprite`. Vytvorili sme si triedu v hlavičkovom súbore „Faceplate\_Class.h“ a rovnako aj objekt v hlavičkovom súbore „HMI.h“, cez ktorý prístupujeme k požadovaným vizuálnym pomôckam.

### **3.4.2.5 HMI**

Využitím vizuálnych nástrojov sme dokázali vytvoriť užívateľské prostredie, ktoré ponúkne užívateľovi komfortné nastavovanie a prehľadné zobrazovanie parametrov pre výhrevné okruhy ústredného kúrenia.

## **3.4.3 Algoritmy**

### **3.4.3.1 Hysteréza**

Na riadenie signálu využívame dvojpolohové prvky, preto nám stačí využívať princíp hysterézy. Zariadenie sme naprogramovali, aby reagovalo zmenou stavu len pri prekročení vymedzenej hranice. To znamená že ak sa teplota nachádza v pásme

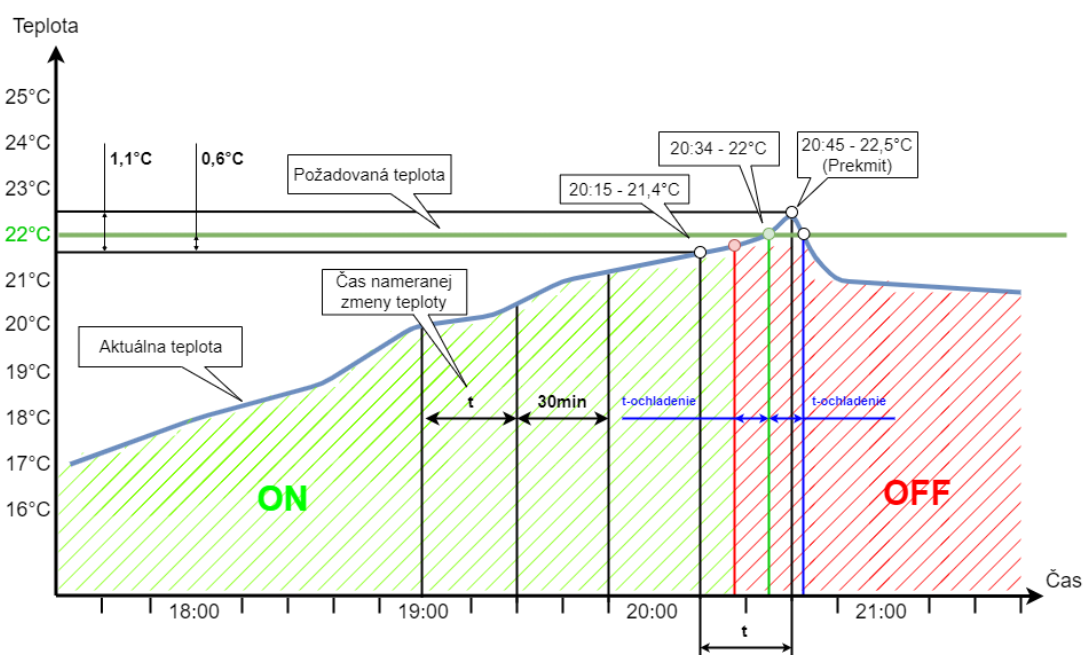
necitlivosti (deadband), predchádzajúci stav sa nebude meniť. Pri prekročení teploty nad alebo pod požadovanú teplotu s toleranciou sa stav preklopí.

```
else if (State.Temperature >= Control.Parameters.Required_T + Control.Parameters.Positive_tolerance) State.Heating = false;
else if (State.Temperature <= Control.Parameters.Required_T - Control.Parameters.Negative_tolerance) State.Heating = true;
```

OBRÁZOK 17 PRAKTICKÁ REALIZÁCIA HYSTERÉZY V PROGRAME

### 3.4.3.2 PD

Aby sme sa vyhli teplotným prekmitom, museli sme zariadenie vybaviť naprogramovaným PD regulátorom. Vytvorili sme algoritmus, ktorý predvída budúci stav teploty podľa predchádzajúcich nameraných hodnôt. Dovoľuje odchytku aktuálnej teploty oproti predchádzajúcej teplote a ukladá do premennej čas, za ktorý sa teplota zmenila. Výsledkom uložených hodnôt je určitá rýchlosť stúpania alebo klesania nameranej teploty. Rovnako sme z predchádzajúcich stavov namerali čas ochladenia alebo zohriatia miestnosti. Pomocou vybraných parametrov sme boli schopní vytvoriť algoritmus, ktorý dokáže zistiť vhodný čas za ktorý sa vykurovací stav musí zmeniť.



OBRÁZOK 18 GRAFICKÉ ZNÁZORNENIE FUNKCIE PD ALGORITMU

### 3.4.3.3 Intervaly vykurovania

Zabezpečenie vykurovania v nastavených časových intervaloch bolo neoddeliteľnou súčasťou funkcie zariadenia. Mali sme k dispozícii aktuálny čas získaný cez NTP protokol alebo cez hodinový RTC modul. Pre jednoduché ukladanie alebo nastavovanie doby vyhrievania sme zvolili ukladanie nastaveného času v minútach.



Taktiež bolo nevyhnutné zabezpečiť nezávislosť medzi výhrevnými okruhmi a vybranými časovými intervalmi vykurovania. Pre možnosť priradovania časových intervalov k výhrevným okruhom sme zvolili číslo v dvojkovej sústave. Určitá hodnota čísla od 1 po 31 dokáže priradiť ľubovoľnú kombináciu piatich časových intervalov k požadovaným výhrevným okruhom.

```

if (value - 16 >= 0) {
    value = value - 16;
    for (int i = 0; i < 5; i++) {
        if (((Real_time / 60) >= Interval[4].zap[i]) && ((Real_time / 60) < Interval[4].vyp[i]))
            && (Interval[4].Days[State.WeekDay - 1]) && Interval[4].enable[i]) return true;
    }
}
if (value - 8 >= 0) {
    value = value - 8;
    for (int i = 0; i < 5; i++) {
        if (((Real_time / 60) >= Interval[3].zap[i]) && ((Real_time / 60) < Interval[3].vyp[i]))
            && (Interval[3].Days[State.WeekDay - 1]) && Interval[3].enable[i]) return true;
    }
}
}

```

OBRÁZOK 19 PRAKTICKÁ REALIZÁCIA PD REGULÁTORA V PROGRAME

#### 3.4.3.4 Časovač (Timer)

Nie je možné, aby boli všetky funkcie vykonávané pri každom cykle programu. Niektoré činnosti sa vykonávajú dlhšiu dobu, čo by zvyšovalo čas jedného cyklu.

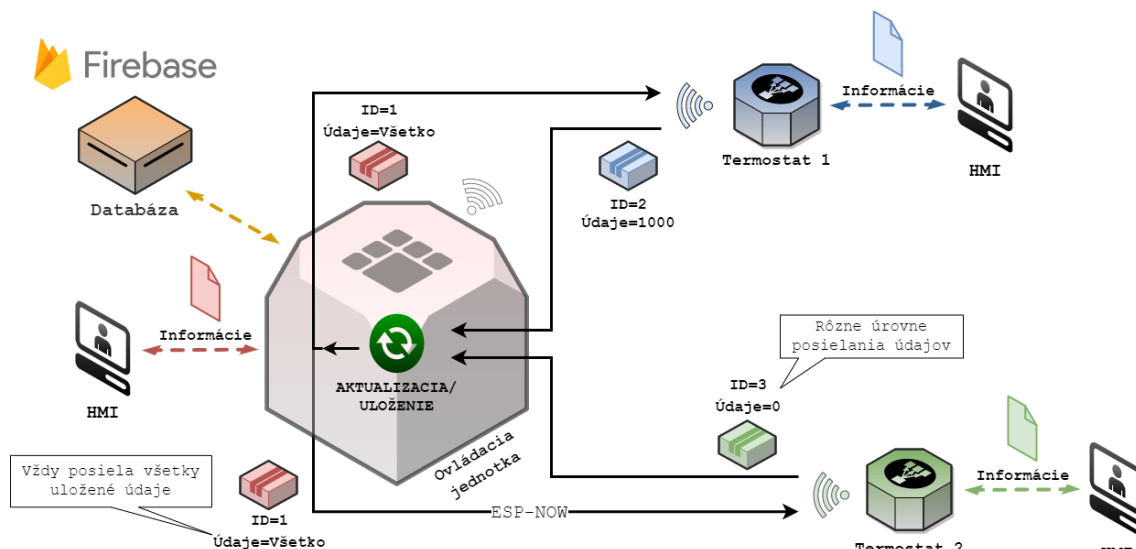
Časovač je v programe jednou z hlavných funkcií na posielanie, prijímanie, meranie a ukladanie dát v programe. Vytvorili sme časovač, ktorý využíva funkciu millis() na určovanie uplynutého času. Časovač začne ubiehať až pri jeho prvom príkaze.

#### 3.4.3.5 Dotyk

Pri programovaní užívateľského prostredia na displej sa vyskytol problém v snímaní stáleho dotyku. Snímanie dotyku na displeji sa so zvyšujúcou teplotou zhoršuje. Pridali sme preto k funkcií na snímanie dotyku časovač, ktorý drží aktívnu hodnotu dotyku aj po malých prerušeníach. Časovač v termostatoch sme navrhli adaptívne v závislosti na dobe vykresľovania, aby nedochádzalo k zvyšovaniu latencie. Na rozdiel od ovládacej jednotky, v ktorej je časovač nastavený v závislosti od teploty aby sa znížilo riziko neschopnosti snímať dotyk.

### 3.4.3.6 Posielanie informácií ESP-NOW

Pre dostatočne dlhú výdrž na jedno nabitie a taktiež odpojenie určitého výhrevného okruhu pri strate termostatu, bolo nutné vymyslieť správny princíp komunikácie.



#### 3.4.3.6.1 Termostaty

Zariadenie sme naprogramovali, aby bolo väčšinu času v režime spánku. V tom čase ovládacia jednotka nedokáže komunikovať s termostatom čo by zapríčinilo okamžité odpojenie výhrevného okruhu.

Vytvorili sme princíp komunikácie, v ktorom len termostat odosiela údaje podľa užívateľom nastaveného času. Termostat sa po dosiahnutí užívateľom nastavenej doby spánku zobudí. Nastaví do základného stavu všetky parametre a odošle údaj o aktuálnej nameranej teplote do ovládacej jednotky. Po odoslaní dát, termostat čaká na spätné prijatie informácií od ovládacej jednotky. Všetky prijaté dáta sa uložia a dokáže nám zobrazovať, nami nastavené parametre vyhrievania.

#### 3.4.3.6.2 Ovládacia jednotka

Zabezpečili sme, aby komunikácia prebiehala len po obdržaní dát. Zariadenie prijíma určitú úroveň dát z termostatu a podľa aktuálnych informácií upraví alebo nastaví parametre vykurovania. Následne aktualizuje informácie na displeji a odošle všetky uložené dáta do termostatu, od ktorého bola prijatá správa. Rovnako sme vytvorili snímacie komunikácie prostredníctvom časovača. Keďže ovládacia jednotka nedokáže komunikovať s termostatom v režime spánku, boli sme nútení vytvoriť algoritmus, ktorý zabezpečí kontrolu komunikácie. Algoritmus prijíma a uchováva informácie o dobe

spánku v časovači. Komunikácia prebieha ak bol časovač počas doby svojej činnosti prerušený prijatím dát. V opačnom prípade, ak časovač dovŕši nastavený čas, vypíše na displej ovládacej jednotky chybu komunikácie a odpojí daný výhrevný okruh.

#### **3.4.3.7 Ukladanie do pamäte flash**

Aby sme boli schopný vyvarovať sa strate užívateľom nastavených údajov, využili sme integrovanú EEPROM pamäť v komunikačnom module ESP32. Pre jednoduchšie zaobchádzanie s čítaním a vpisovaním do pamäte sme vytvorili triedu nazvanú „Flash.h“. Obsahovala funkcie pre čítanie a vpisovanie dát do pamäte len prostredníctvom identifikačného čísla.

Ukladanie informácií do EEPROM pamäte v module ESP32 je obmedzené na určitý počet. Aby sme sa vyhli poškodeniu pamäte pri vpisovaní, nastavili sme časovač, ktorý po dovŕšení nastaveného času porovná všetky uložené dáta s hodnotami, ktoré by sa mali uložiť do rovnakého miesta pamäte podľa identifikačného čísla.

#### **3.4.3.8 Jazyk**

Pre možnosť ovládania zariadení v dvoch jazykoch sme vytvorili triedu, ktorá uchováva nastavený jazyk. Pred každým výpisom textu na displej sme priradili funkciu, do ktorej bolo potrebné zadať dva parametre pre text v slovenskom a anglickom jazyku. Následne pri výpise textu trieda „Language.h“ vložila text v užívateľom nastavenom jazyku.

### **3.4.4 Princíp**

Funkcia činnosti hlavnej ovládacej jednotky ústredného kúrenia s funkciou termostatov bola diametrálne odlišná. V programoch sa využívali rôzne knižnice, čo nám zvyšovalo čas nahrávania programu do zariadení. Na vyriešenie problému bolo potrebné zabezpečiť ku každému zariadeniu konkrétny program.

#### **3.4.4.1 Ovládacia jednotka ústredného kúrenia**

Zariadenie sme nastavili tak, aby sa prístroj po pripojení k zdroju napájania napojil na lokálnu sieť WI-FI. Po napojení nastaví všetky použité objekty cez funkciu „Setup“ na správnu činnosť. Rovnako priradí objektovým atribútom hodnoty uložené vo flash pamäti z predchádzajúceho zapnutého stavu. Tým sme zabezpečili opätovný funkčný priebeh aj po výpadku napájania. Ovládacia jednotka ústredného kúrenia pracuje na PLC cykle.

### **3.4.4.2 Termostaty**

Aby sme predišli krátkej výdrže na jedno nabitie, vytvorili sme v zariadení cyklus zabezpečujúci uspanie komunikačného modulu (Sleep mode). Naprogramovali sme, aby sa zariadenie začalo režim spánku len po uplynutí užívateľom nastaveného času od posledného dotyku na displeji. Termostat bude v dobe spánku v neaktívnom režime, čo znemožní užívateľovi použitie zariadenia. Prístroj sa prebudí z uspatého stavu len po uplynutí užívateľom nastaveného času, alebo tlačidlom na hornej strane termostatu. Priradili sme do zdrojového kódu program, ktorý po prebudení zariadenia zaistuje skenovanie dostupných prístupových bodov o kmitočte 2,4GHz. Nájdením správnej siete modul ESP32 získa komunikačný kanál na komunikáciu s ovládacou jednotou ústredného kúrenia.

Zobrazenie aktuálnych parametrov a nameraných hodnôt termostatu po zobudení alebo resetovaní, sme zabezpečili odoslaním požiadavky do ovládacej jednotky ústredného kúrenia. Naprogramovali sme, aby bola požiadavka odoslaná hneď po prvom nameraní aktuálnej teploty. Po odoslaní požiadavky vo forme určitej úrovne odoslaných dát, termostat čaká na spätné prijatie údajov. Prijaté údaje obsahujú aktuálne nastavené parametre vykurovania, uložené v ovládacej jednotke. Termostat po načítaní užívateľského prostredia zobrazí aktuálne parametre vyhrievania. Termostat pracuje na PLC cykle.

## **3.5 Mechanická konštrukcia**

Po vytvorení dosiek plošných spojov a ich oživení sme mohli vytvoriť kryt pre ochranu elektronických súčiastok a jednoduchú manipuláciu.

### **3.5.1 SolidWorks a 3D tlač**

Na vytvorenie požadovaného krytu sme využili jednoduchý a intuitívny software, navrhnutý tak, aby maximálne uľahčoval prácu konštruktérom. V programe SolidWorks sme vytvorili 3D modeli krytov, ktoré sme následne vytlačili na 3D tlačiarňi. 3D tlačiareň sme nemali k dispozícii preto som oslovil firmu MTS s.r.o., ktorá nám umožnila tlač navrhnutých 3D modelov.

### **3.5.2 Prvé verzie**

#### **3.5.2.1 Ovládacia jednotka ústredného kúrenia**

Konštrukciu sme vytvárali aby kopírovala línie vytvorenej DPS pre prijateľné rozmery. Nepožadovali sme krytie voči vode ani prachu, čo sa časom ukázalo ako chybné rozhodnutie. Nami navrhnutý 3D model mal otvorený priestor pre vyvedenie napájania a vodiče pre riadenie výhrevných okruhov ústredného kúrenia. Zariadenie neobsahovalo žiadny druh chladenia. Dochádzalo k prehrievaniu spolu s väčšími nepresnosťami hodinového RTC modulu.

#### **3.5.2.2 Termostaty**

Aby sme zaistili dostatočnú výdrž krytu pri bežnom používaní, museli sme vytvoriť konštrukciu, ktorá by pevne držala displej, batériu a základnú dosku. Kryt pozostával z troch odnímateľných častí. Nevýhodou prvej verzie návrhu boli veľké rozmery a problém s meraním teploty. Teplotný snímač nebol oddelený od základnej dosky dostatočne veľkým priestorom a ani žiadnou tepelnou izoláciou. Teplomer vždy nameral teplotu s parazitnou teplotou vytvorenou komunikačným modulom.

### **3.5.3 Súčasné verzie**

#### **3.5.3.1 Ovládacia jednotka ústredného kúrenia**

Po vytvorení novej DPS sme museli do krytu zakomponovať viacero vylepšení. DPS obsahovala navyše aktívne aj pasívne chladenie spolu s indikačnou LED diódou a konektorom pre napájanie. Rozhodli sme sa zvýšiť krytie voči vniknutiu tuhých pevných telies väčších ako 12,5 mm, čo odpovedá krytiu IP2X. Kvôli stále veľkým rozmerom novej DPS a viacerým externým zariadeniam sme neboli schopný vytvoriť rovnako malú konštrukciu ako konštrukcia pre termostaty. Vytvorili sme kryt, ktorý ponúka rovnako malý pôdorys ako kryt termostatov, ale museli sme zvýšiť jeho výšku o 2cm.

#### **3.5.3.2 Termostaty**

Pri návrhu sme vychádzali z prvej verzie 3D modelu. Boli sme si vedomí vzniknutých nedostatkov, ktoré sa prejavili pri bežnom používaní.

Podmienkou pri vytvorení novej konštrukcie bolo zabezpečenie správnej hodnoty nameranej teploty. Snímač teploty sme externe vyviedli pomocou štyroch vodičov mimo základnú DPS termostatu. Medzi teplotným snímačom a DPS s komunikačným modulom

ESP32 sme vytvorili vzduchovú bariéru. Vzduchová bariéra tepelne izoluje časť DPS s komunikačným modulom a snímač teploty.

Oproti prvej verzii sme sa zbavili jednej časti konštrukcie, ktorá plnila funkciu uchytania batérie. Museli sme z prvej verzie odstrániť použitý displej, aby sme mohli ušetriť miesto vložení batérie na DPS displeja. Displej z prvej verzie obsahoval viacero obvodov na vlastnej DPS, ktoré bránili vloženiu batérie. Batériu s displejom už stačilo len dotlačiť aby termostat nebol labilný či menej odolný.

### **3.6 Vytvorenie webovej aplikácie**

Pre možnosť zobrazovania a ovládania parametrov výhrevných okruhov z ľubovoľného miesta na zemi, sme zvolili databázu firebase.

#### **3.6.1 Databáza**

Aby sme mohli databázu využívať bolo potrebné, aby sme ju najprv nastavili podľa určitých kritérií. Museli sme si vytvoriť nový projekt a nastaviť overovacie metódy pre posielanie nameraných hodnôt a prístupu do webovej aplikácie. Zvolili sme si overovaciu metódu pomocou g-mail účtu. Na záver bolo potrebné si zvoliť server, kde budú naše dáta ukladane a aj prístupové pravidlá pre čítanie databázy a písanie do databázy.

#### **3.6.2 Webové rozhranie**

Webové rozhranie sme programovali cez vývojové prostredie Visual Studio Code pomocou rozšírenia pre posielanie zdrojového kódu do firebase serverov. Zdrojový kód pozostával zo skupiny prepojených jazykov html, css a javascript.

#### **3.6.3 Arduino IDE**

Pre prístup k databáze sme využili knižnicu „Firebase\_ESP\_Client.h“, ktorá podporovala čítanie a posielanie údajov z firebase databázy. Nevyhnutným bolo, aby sme zadefinovali potrebné parametre k pripojeniu do databázy, ako napríklad zvolený g-mail alebo URL adresu. Po povolení databázy na pripojenie komunikačnému modulu ESP32 sme boli schopný posielat' požadované dáta na databázu prostredníctvom JSON objektu.

## **4 VÝSLEDKY PRÁCE A DISKUSIA**

### **4.1 Výsledky**

Výsledkom našej práce bolo vytvorenie užívateľsky prehľadného funkčného zariadenia na ovládanie výhrevných okruhov ústredného kúrenia. Zariadenie dokáže spoľahlivo fungovať aj pri nepredvídateľných situáciách, ako je napríklad výpadok elektrickej energie. Vďaka tejto funkcionalite sme schopní chrániť regulované zariadenie ako aj užívateľský komfort.

Funkcionalita je zameraná na spektrum možných parametrov vykurovania. Ponúka možnosť nastavovania hysteréznej slučky, od výberu teplotného snímača až po nastavenie vhodných vykurovacích intervalov.

Zariadenie ponúka informácie o aktuálnej hodnote spaľovacieho paliva pomocou ultrazvukového senzora. Pre správne zobrazovanie kapacity v percentách je užívateľ schopný nastaviť senzor v stave, keď je zásobník naplnený a rovnako aj prázdny.

Prístroj sme navrhovali tak, aby spĺňalo odolnosť voči vniknutiu pevných cudzích telies väčších ako 12mm odpovedajúce stupňu ochrany krytom IP 2X.

### **4.2 Zistenia**

Vyber typu modulu ESP32 sme zvolili správne. Podporuje dostatočne výkonný procesor pre vykresľovanie užívateľky príjemného prostredia. Rovnako obsahuje dostatok kontaktov pre všetky nami použité periférie. Nevýhodou bolo zistenie že komunikačný modul nedisponuje dostatočne veľkou SRAM pamäťou. Zabránilo nám to možnosť vykreslenia pomocou spritu o rozmeroch displeja.

### **4.3 Pozorovania**

Pozorovaním chodu zariadenia sme boli schopní zistiť možné poruchy pri vzniknutí nepriaznivých podmienok, ako napríklad výpadku elektrického prúdu.

Taktiež sme pozorovali výdrž termostatu na jedno nabitie z čoho nám vychádzala neuveriteľná hodnota jeden mesiac pri správne nastavenom posielaní údajov.

### **4.4 Diskusia**

Pri práci na výrobku sme nadobudli množstvo praktických skúseností. Oboznámili sme sa s tvorbou samostatne funkčného zariadenia a prehĺbili sme svoje znalosti v oblasti elektroniky a programovania. Poznatky získané pri tvorbe zariadenia nám ponúkajú nové možnosti pri tvorbe a organizovaní zložitejších projektov.

## **5 ZÁVER PRÁCE**

Výsledkom našej práce bolo prepojenie elektroniky, programovania a konštrukcie v jedno funkčné zariadenie. Celkový vývoj postupoval systematicky od prvých verzií prototypov až k výslednému prístroju. Práve vďaka správne načasovaniu riešenia komplexných problémov sme boli schopní dospieť k funkčnému výrobku.

### **5.1 Splnenie cieľov**

- Kompaktnosť zariadení sme s veľkou mierou splnili. Zariadenia vyzerajú dostatočne prakticky a funkčne.
- DPS pre termostaty spotrebúvajú minimálne množstvo elektrického prúdu, z čoho termostat dokáže vydržať na jedno nabitie viac ako mesiac.
- Zabezpečenie nepretržitej prevádzky aj po strate a opätovnom nahodení napájania.
- Termostaty dokážu komunikovať s ovládacou jednotkou ústredného kúrenia bez prekážok vo vzdialenosti do 25m.
- Vytvorenie komunikácie s databázou.
- Vytvorenie vlastných obvodov na programovanie a napájanie modulu ESP32.

### **5.2 Využitie v praxi**

Zariadenie malo od prvých analýz jasné určenie v praxi. Ovládanie výhrevných okruhov pomocou dvoch relé, ktoré zopínajú riadiaci signál podľa nastavených parametrov. Prístroj je vhodný pre všetky druhy vykurovacích zariadení, ktoré na svoje riadenie potrebujú dva riadiace dvojpolohové prvky.

### **5.3 Zlepšenie**

Budúci vývoj práce by smeroval k modernejšiemu dizajnu užívateľského prostredia. Zariadenie sme programovo navrhli aby bola možnosť rozšíriť ovládacie prvky pre vykurovacie okruhy.

Pridali by sme do krytu ovládacej jednotky ústredného kúrenia možnosť uchýtenia na DIN lištu.



## **ZHRNUTIE**

Cieľom našej práce bolo vytvoriť zariadenie pre automatické ovládanie výhrevných okruhov ústredného vykurovania využívajúce bezdrôtové termostaty. Požadovali sme aby zariadenie fungovalo sebestačne podľa nastavených parametrov.

Vytvorili sme kompaktné termostaty s energeticky nenáročnou DPS pre dlhú výdrž na jedno nabitie. Ovládaciu jednotku sme navrhli s riadiacimi prvkami na ovládanie vykurovania pomocou relé.

Po vytvorení funkčného hardwaru sme navrhli program, ktorého jadro tvorí PLC cyklus. V programe sme vytvorili vlastnú štruktúru aby nedošlo k výpadkom alebo chybám pri používaní.

Kryty zariadení sme navrhovali tak, aby spĺňali minimálne krytie certifikácie IP 2X podľa rozmerov DPS. Navrhnuté modeli sme vytlačili na 3D tlačiarňu v spoločnosti MTS s.r.o..

Nami určený výsledok sa nám podarilo splniť systematickým plnením čiastočných cieľov. Nadobudli sme skúsenosti pre vytvorenie komplexného programu, kompaktného DPS ako aj určovania vlastných priorít s ohľadom na čas.

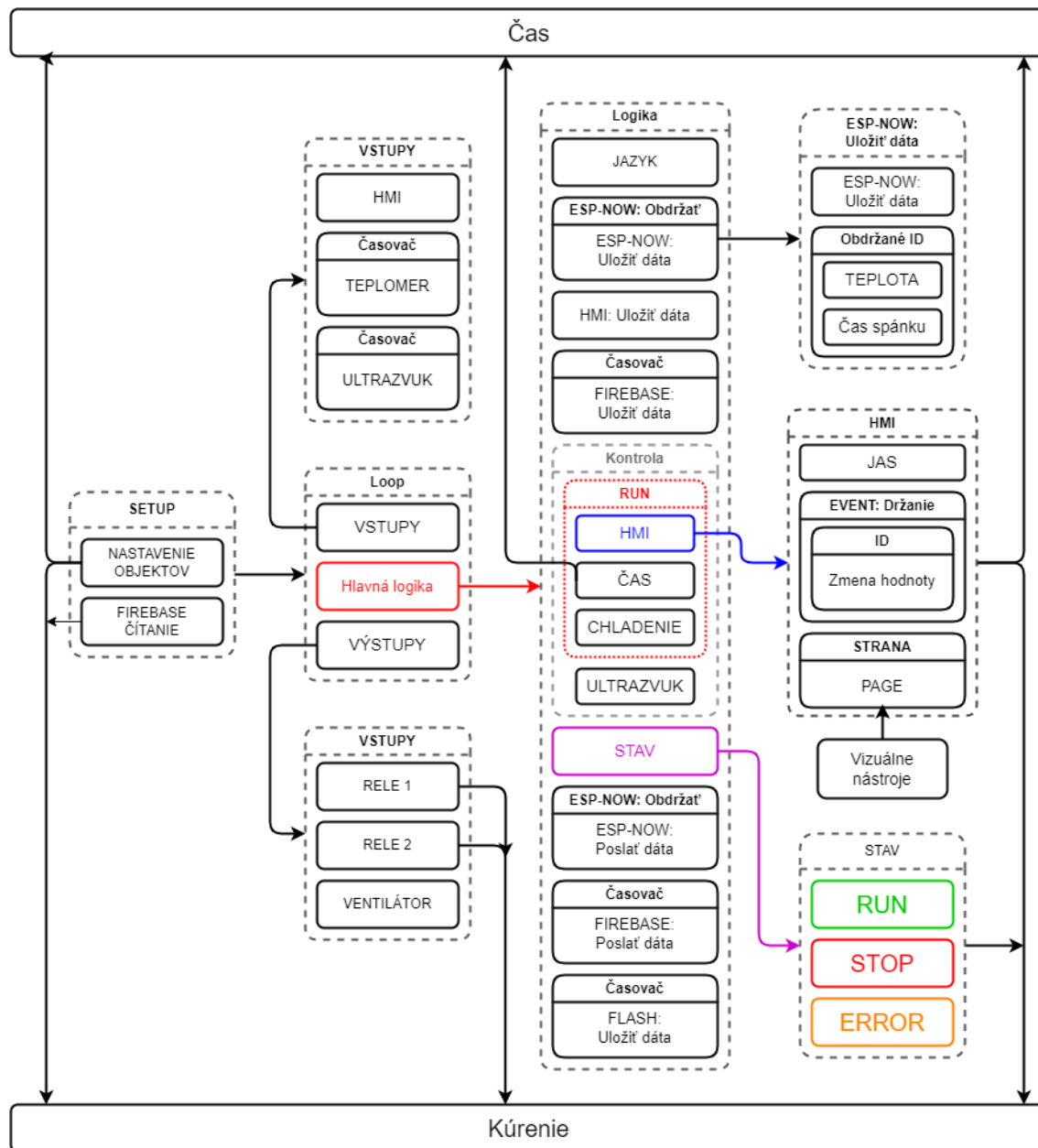
# ZOZNAM POUŽITEJ LITERATÚRY

1. *Regulacny obvod*. **Mydlo, Peter**. Banská Bystrica SPŠJM : Peter Mydlo, 2018. 02.01 .
2. *Dvojpolohová regulácia*. **Mydlo, Peter**. Banská Bystrica SPŠJM : Peter Mydlo, 2018. 02.05.
3. **ok2jnj**. Program PID regulátoru mikropájky. *ok2jnj.ok2kld*. [Online] 14. 2 2015. [Dátum: 2. 2 2023.] <https://ok2jnj.ok2kld.cz/ok2jnj/cs/2015/02/program-pi-regulatoru-mikropajky/>.
4. **MIKEGRUSIN**. Serial Peripheral Interface (SPI). *sparkfun*. [Online] 2018. [Dátum: 20. 1 2023.] <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/res>.
5. Introducing JSON. *json.org*. [Online] [Dátum: 20. 1 2023.] <https://www.json.org/json-cz.html>.
6. Čo je PWM. *electricianexp*. [Online] 30. 11 2020. [Dátum: 20. 1 2023.] <https://electricianexp.com/41/sk/knowledgebase/chtotakoe-shim>.
7. I<sup>2</sup>C. *wikipedia*. [Online] 29. 1 2023. [Dátum: 7. 2 2023.] <https://en.wikipedia.org/wiki/I%C2%B2C>.
8. Getting Started with the ESP32 Development Board. *randomnerdtutorials*. [Online] [Dátum: 2. 5 2023.] <https://randomnerdtutorials.com/getting-started-with-esp32/>.
9. Getting started with Firebase. *cloud.google*. [Online] 7. 2 2023. [Dátum: 7. 2 2023.] <https://cloud.google.com/firestore/docs/client/get-firebase>.
10. ESP32: Getting Started with Firebase (Realtime Database). *randomnerdtutorials*. [Online] [Dátum: 20. 2 2023.] <https://randomnerdtutorials.com/esp32-firebase-realtime-database/>.
11. ESP-NOW User Guide. *espressif*. [Online] 20. 7 2016. [Dátum: 7. 2 2023.] <https://www.espressif.com/en/products/software/esp-now/resources>.
12. ESP-NOW with ESP32: Receive Data from Multiple Boards (many-to-one). *randomnerdtutorials*. [Online] [Dátum: 18. 2 2023.] <https://randomnerdtutorials.com/esp-now-many-to-one-esp32/>.
13. Campbell, Scott. BASICS OF UART COMMUNICATION. *circuitbasics*. [Online] 2016. [Dátum: 2. 7 2023.] <https://www.circuitbasics.com/basics-uart-communication/>.
14. What is a Sensor? *analog*. [Online] [Dátum: 7. 2 2023.] <https://www.analog.com/en/design-center/glossary/sensor.html>.
15. Team, D2000 Dev. Limity. *doc.ipesoft*. [Online] 12. 10 2017. [Dátum: 15. 2 2023.] <https://doc.ipesoft.com/display/D2DOCV12SK/Limity>.
16. Create analog voltage on ESP32 with PWM. *upesy*. [Online] 23. 12 2022. [Dátum: 23. 1 2023.] <https://www.upesy.com/blogs/tutorials/how-to-use-pwm-on-esp32-with-examples>.
17. Afzal, Sal. I2C Primer: What is I2C? (Part 1). *analog*. [Online] [Dátum: 10. 2 2023.] <https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html>.

18. FastBitLab. STM32 I2C Lecture 3 : I2C protocol explanation. *fastbitlab*. [Online] 5. 8 2019. [Dátum: 8. 1 2023.] <https://fastbitlab.com/stm32-i2c-lecture-3-i2c-protocol-explanation/>.
19. ESP32: ESP-NOW Web Server Sensor Dashboard (ESP-NOW + Wi-Fi). *randomnerdtutorials*. [Online] [Dátum: 15. 1 2023.] <https://randomnerdtutorials.com/esp32-esp-now-wi-fi-web-server/>.
20. Chaudhari, Dipali. 20 Different Types of Sensors Used in Industry. *dipslab*. [Online] [Dátum: 8. 2 2023.] <https://dipslab.com/sensor-types/>.

## **ZOZNAM PRÍLOH**

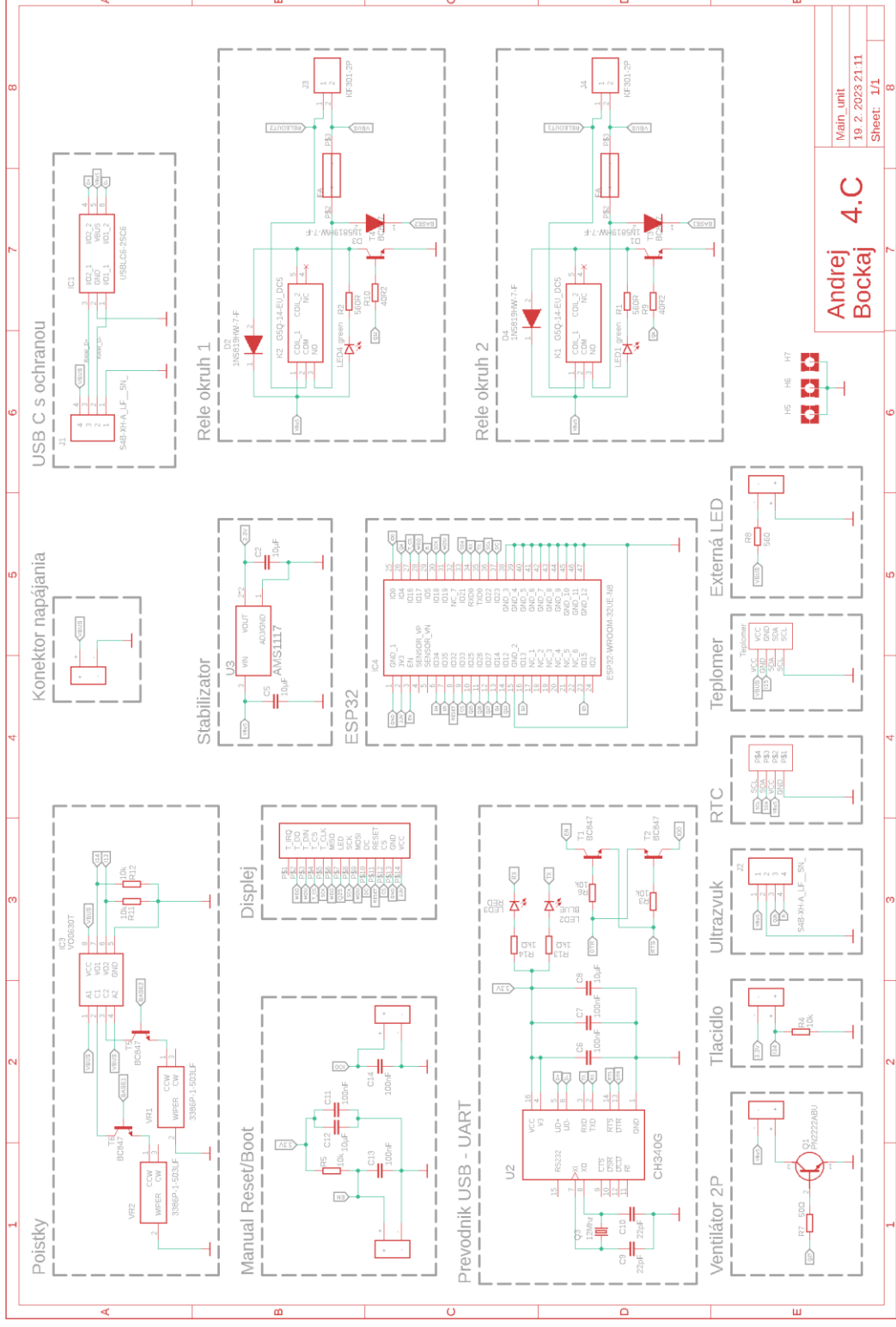
<i>PRÍLOHA A GRAFICKÉ ZNÁZORNENIE ŠTRUKTÚRY PROGRAMU .....</i>	<i>37</i>
<i>PRÍLOHA B TABUĽKA ELEKTRONICKÝCH SÚČIASTOK .....</i>	<i>38</i>
<i>PRÍLOHA C SCHÉMA OVLÁDACEJ JEDNOTKY .....</i>	<i>39</i>
<i>PRÍLOHA D DPS OVLÁDACEJ JEDNOTKY .....</i>	<i>40</i>
<i>PRÍLOHA E SCHÉMA TERMOSTATU .....</i>	<i>41</i>
<i>PRÍLOHA F DPS TERMOSTATU .....</i>	<i>42</i>
<i>PRÍLOHA G 3D MODEL OVLÁDACEJ JEDNOTKY .....</i>	<i>43</i>
<i>PRÍLOHA H 3D MODEL TERMOSTATU .....</i>	<i>44</i>
<i>PRÍLOHA I PRVÉ VERZIE OVLÁDACEJ JEDNOTKY .....</i>	<i>45</i>
<i>PRÍLOHA J PRAKTICKÁ REALIZÁCIA OVLÁDACEJ JEDNOTKY .....</i>	<i>46</i>
<i>PRÍLOHA K PRAKTICKÁ REALIZÁCIA TERMOSTATU .....</i>	<i>47</i>
<i>PRÍLOHA L PRAKTICKÁ REALIZÁCIA ULTRAZVUKOVÉHO SNÍMAČA .....</i>	<i>48</i>
<i>PRÍLOHA M VÝVOJ DPS OVLÁDACEJ JEDNOTKY .....</i>	<i>49</i>
<i>PRÍLOHA N VÝVOJ DPS TERMOSTATU .....</i>	<i>50</i>
<i>PRÍLOHA O VÝVOJ KONŠTRUKCIE TERMOSTATU .....</i>	<i>51</i>
<i>PRÍLOHA P VÝVOJ KONŠTRUKCIE OVLÁDACEJ JEDNOTKY .....</i>	<i>52</i>
<i>PRÍLOHA Q ZDROJOVÝ KÓD OVLÁDACEJ JEDNOTKY - VIĎ ELEKTRONICKÁ FORMA .....</i>	<i>53</i>
<i>PRÍLOHA R ZDROJOVÝ KÓD TERMOSTATU - VIĎ ELEKTRONICKÁ FORMA .....</i>	<i>54</i>

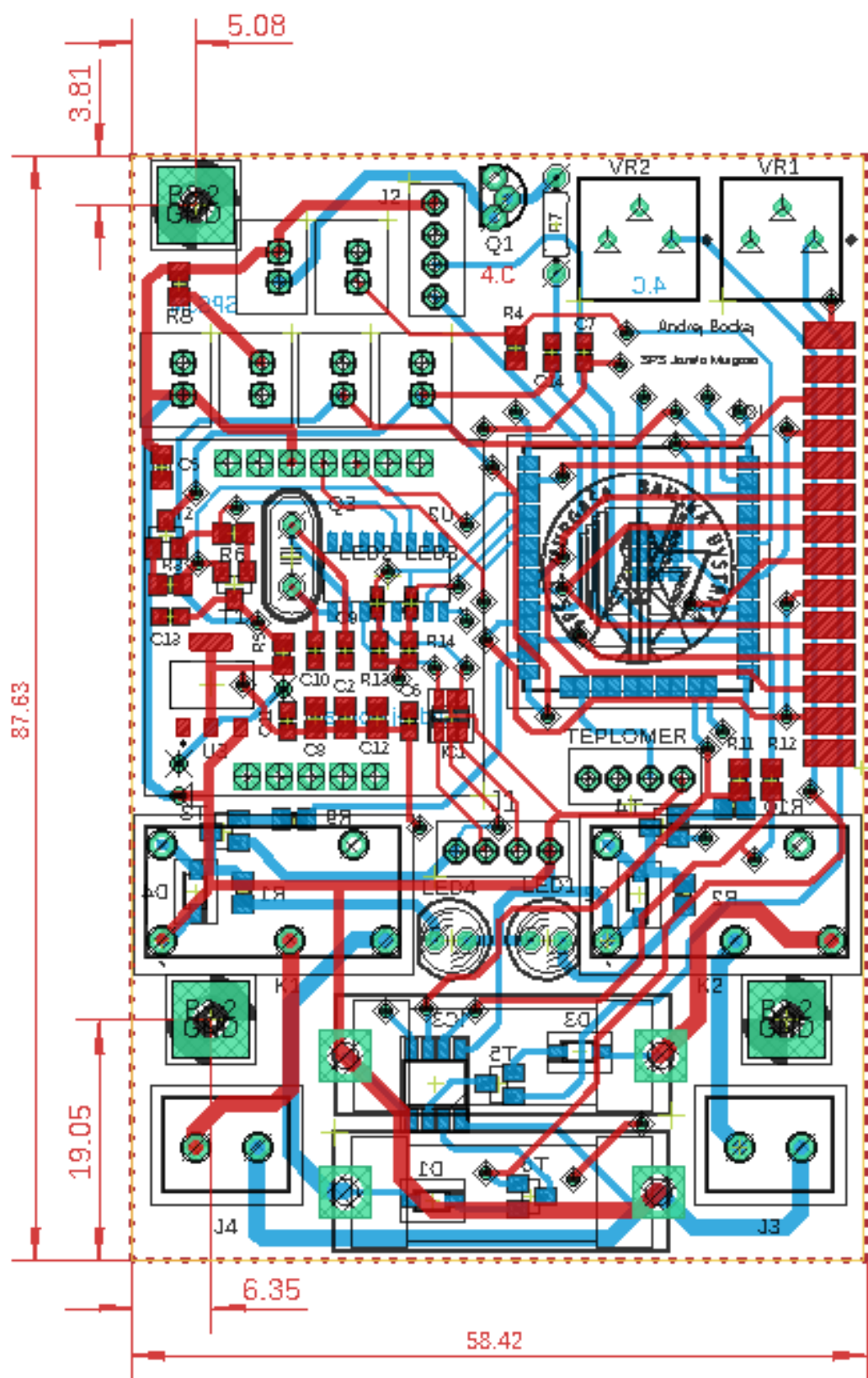


PRÍLOHA B TABUĽKA ELEKTRONICKÝCH SÚČIASTOK

Názov	Značka / Stranka	Typ / Hodnota	Počet kusov		Cena za kus	Výsledná cena		
			Ovládacia jednotka	Termostaty		Ovládacia jednotka	Termostaty	Celkovo
Antena	LaskaKit.cz	3dB 2.4G U.FL (IPEX)	1	0	0,76 €	0,76 €	0,00 €	0,76 €
	MOLEX	47950-1011	0	1	3,78 €	0,00 €	3,78 €	3,78 €
Dioda	ODES INCORPORAT	1N5819HW-7-F	4	2	0,15 €	0,60 €	0,30 €	0,90 €
Displej	aliexpress.com	4.0 inch LCD 480*320 MCU SPI Serial TFT LCD ST7796S	1	1	13,20 €	13,20 €	13,20 €	26,40 €
ESP32	ESPRESSIF	ESP32-WROOM-32U-N4	1	1	5,78 €	5,78 €	5,78 €	11,56 €
Hodinový modul	LaskaKit.cz	DS1306	1	0	1,60 €	1,60 €	0,00 €	1,60 €
Chladič	LaskaKit.cz	Hliníkový 25x25x10	1	0	0,59 €	0,59 €	0,00 €	0,59 €
Kondenzátor	KYOCERA AVX	22pF	2	2	0,13 €	0,25 €	0,25 €	0,51 €
	WALSIN	100nF	5	6	0,02 €	0,08 €	0,10 €	0,18 €
	WALSIN	10uF	4	6	0,03 €	0,12 €	0,18 €	0,30 €
Konektor	LaskaKit.cz	JST-PH 2mm 2pin SMD	0	1	0,08 €	0,00 €	0,08 €	0,08 €
	LaskaKit.cz	2P JST-XH 2.5mm	1	0	0,02 €	0,02 €	0,00 €	0,02 €
	LaskaKit.cz	4P JST-XH 2.5mm	1	0	0,10 €	0,10 €	0,00 €	0,10 €
	LaskaKit.cz	12mm GX12 - 4p (samec)	1	0	0,93 €	0,93 €	0,00 €	0,93 €
	LaskaKit.cz	12mm GX12 - 4p (samica)	1	0	0,93 €	0,93 €	0,00 €	0,93 €
Kryštál	LaskaKit.cz	YIC 12.00M-HC49-S 12MHz	1	1	0,46 €	0,46 €	0,46 €	0,92 €
LED	LaskaKit.cz	5mm	3	0	0,04 €	0,12 €	0,00 €	0,12 €
	Orient	ORH-x36x 0603	2	4	0,03 €	0,06 €	0,12 €	0,18 €
Nabíjací IC	TOPPOWER	TP4056	0	1	0,42 €	0,00 €	0,42 €	0,42 €
Napájecí konektor	LaskaKit.cz	DC-022 5,5/2,1mm	1	0	0,25 €	0,25 €	0,00 €	0,25 €
Ochranná ESD	STMicroelectronics	USBLC6-2SC6	1	1	0,23 €	0,23 €	0,23 €	0,46 €
Optočlen	VISHAY	VO0630T	1	0	1,32 €	1,32 €	0,00 €	1,32 €
Poistka	LaskaKit.cz	5x20mm	2	0	0,06 €	0,12 €	0,00 €	0,12 €
Potenciometer	SR PASSIVES	20kΩ	2	0	0,46 €	0,92 €	0,00 €	0,92 €
Prepínač	LaskaKit.cz	0,5A 50VDC	0	2	0,25 €	0,00 €	0,50 €	0,50 €
Prevodník	WCH	CH340G USB-UART SOP-16	1	1	0,78 €	0,78 €	0,78 €	1,56 €
Priechodka	Copper.sk	19432	2	0	0,40 €	0,80 €	0,00 €	0,80 €
Relé	OMRON OCB	G5Q-14-EU 5VDC	2	0	1,67 €	3,34 €	0,00 €	3,34 €
Rezistor	YAGEO	1kΩ	2	4	0,02 €	0,04 €	0,09 €	0,13 €
	YAGEO	10kΩ	4	5	0,02 €	0,08 €	0,11 €	0,19 €
	ROYAL OHM	1k32	0	1	0,01 €	0,00 €	0,01 €	0,01 €
	WALSIN	560Ω	3	1	0,02 €	0,06 €	0,02 €	0,08 €
	ROYAL OHM	100kΩ	0	2	0,02 €	0,00 €	0,04 €	0,04 €
	ROYAL OHM	40,2Ω	3	0	0,03 €	0,09 €	0,00 €	0,09 €
Senzor teploty	ASAIR	DHT22	1	0	5,20 €	5,20 €	0,00 €	5,20 €
	LaskaKit.cz	SHT31	0	1	7,08 €	0,00 €	7,08 €	7,08 €
	TechFun.sk	HTU21D	0	1	4,90 €	0,00 €	4,90 €	4,90 €
Stabilizátor	LaskaKit.cz	AMS1117	1	1	0,08 €	0,08 €	0,08 €	0,16 €
Step-up menič	LaskaKit.cz	MT3608	0	1	1,01 €	0,00 €	1,01 €	1,01 €
Svorkovnica	LaskaKit.cz	KF301-2P	2	0	0,10 €	0,20 €	0,00 €	0,20 €
Tlačítka	Copper.sk	B3F-4000	1	1	0,50 €	0,50 €	0,50 €	1,00 €
Tranzistor	ONSEMI	PN2222ABU	1	0	0,13 €	0,13 €	0,00 €	0,13 €
	ONSEMI	FDC606P	0	1	0,78 €	0,00 €	0,78 €	0,78 €
	HM SEMICONDUCT	2SA1036KT146Q	0	1	0,09 €	0,00 €	0,09 €	0,09 €
	CDIL	TBC847	4	3	0,05 €	0,21 €	0,16 €	0,37 €
Ultrazvukový senzor	TechFun.sk	HY-SRF05	1	0	2,50 €	2,50 €	0,00 €	2,50 €
USB C	LUMBERG	2435 02	0	1	3,00 €	0,00 €	3,00 €	3,00 €
USB C adapter	LaskaKit.cz	40x25mm	1	0	1,60 €	1,60 €	0,00 €	1,60 €
Ventilátor	SUNON	MF30100V1-1000U-G99	1	0	8,16 €	8,16 €	0,00 €	8,16 €
Vodič	LaskaKit.cz	JST-XH-2 2.5mm 2pin	3	2	0,27 €	0,81 €	0,54 €	1,35 €
	LaskaKit.cz	JST-XH-3 2.5mm 3pin	1	1	0,34 €	0,34 €	0,34 €	0,68 €
	LaskaKit.cz	JST-XH-4 2.5mm 4pin	1	1	0,51 €	0,51 €	0,51 €	1,02 €
	LaskaKit.cz	JST-XH-4 2.5mm 4pin	1	1	0,59 €	0,59 €	0,59 €	1,18 €
Celkové sčítanie	1 Termostat		73	59		54,48 €	46,03 €	100,51 €
	2 Termostaty		73	118		54,48 €	92,06 €	146,54 €

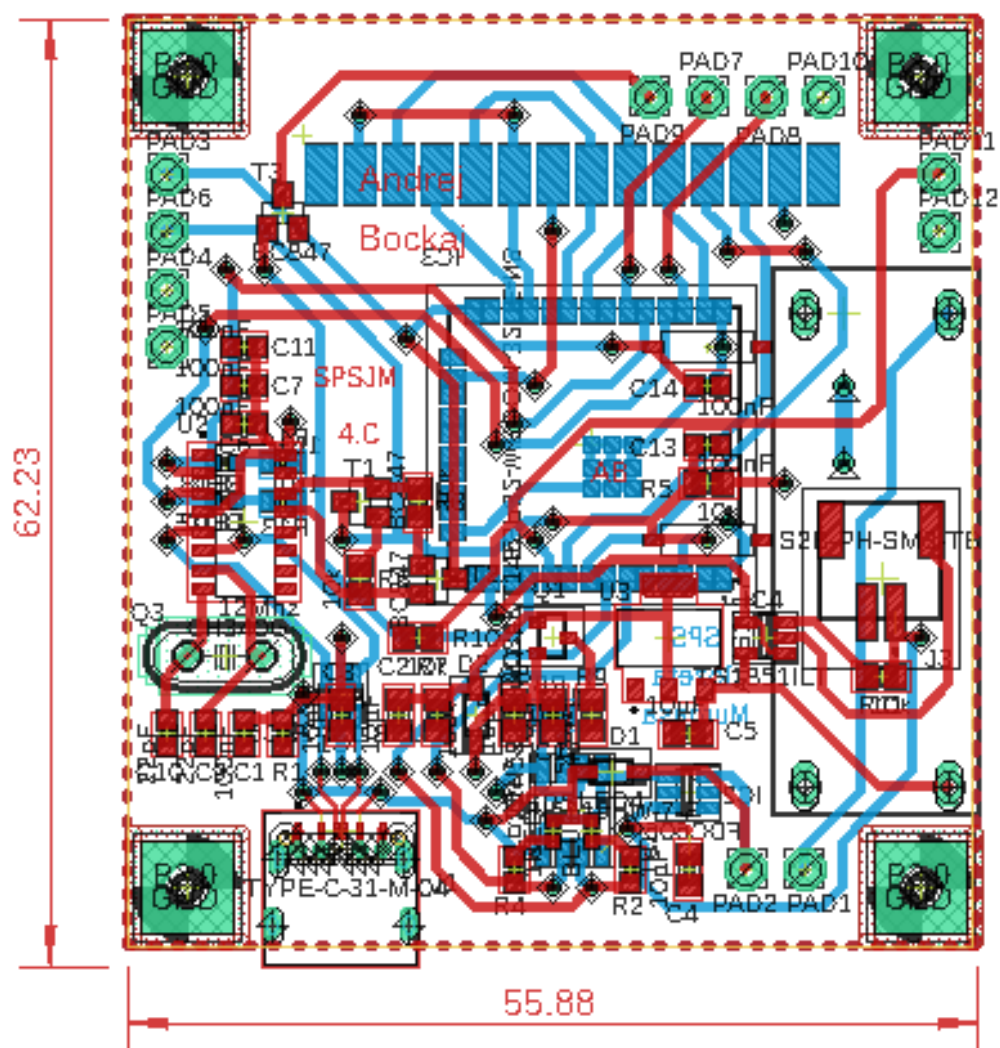
# PRÍLOHA C SCHÉMA OVLÁDAČEJ JEDNOTKY



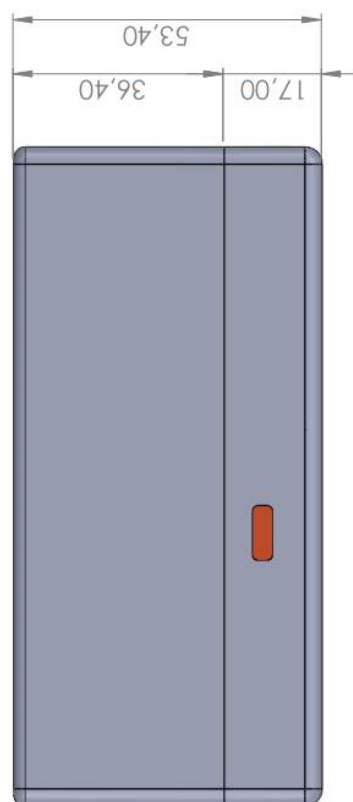
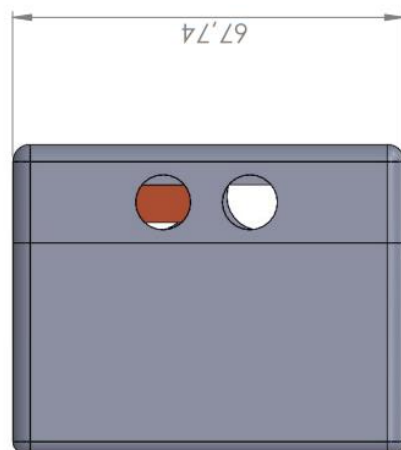




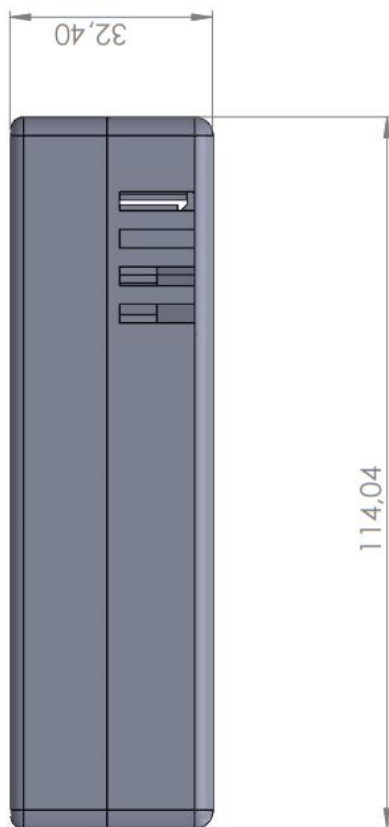
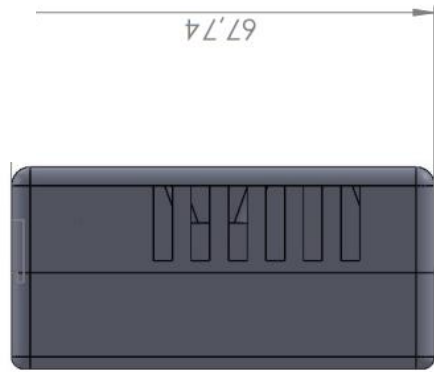




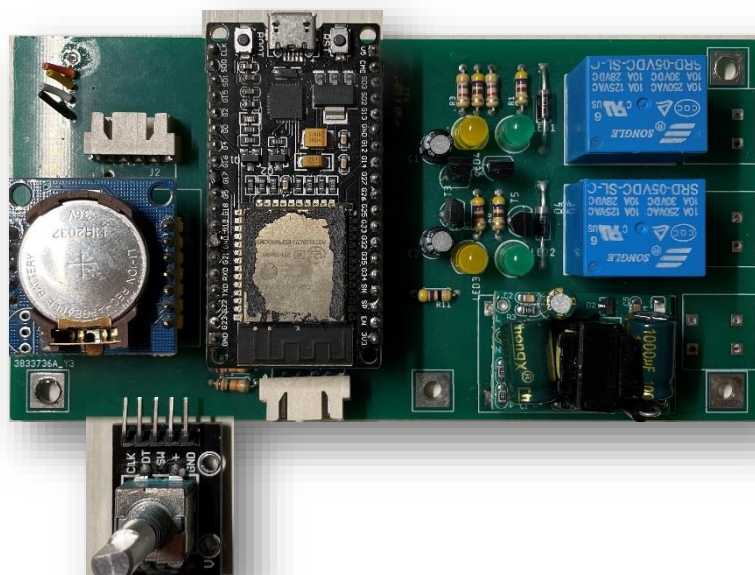
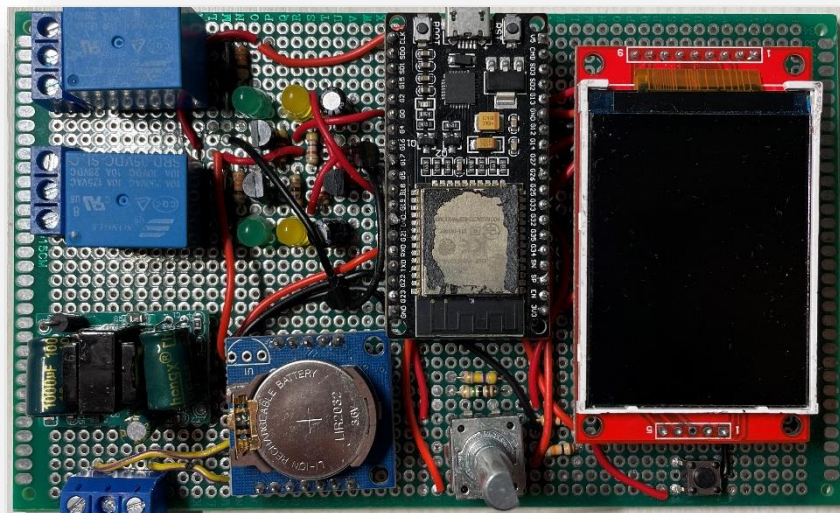
PRÍLOHA G 3D MODEL OVLÁDACEJ JEDNOTKY



PRÍLOHA H 3D MODEL TERMOSTATU

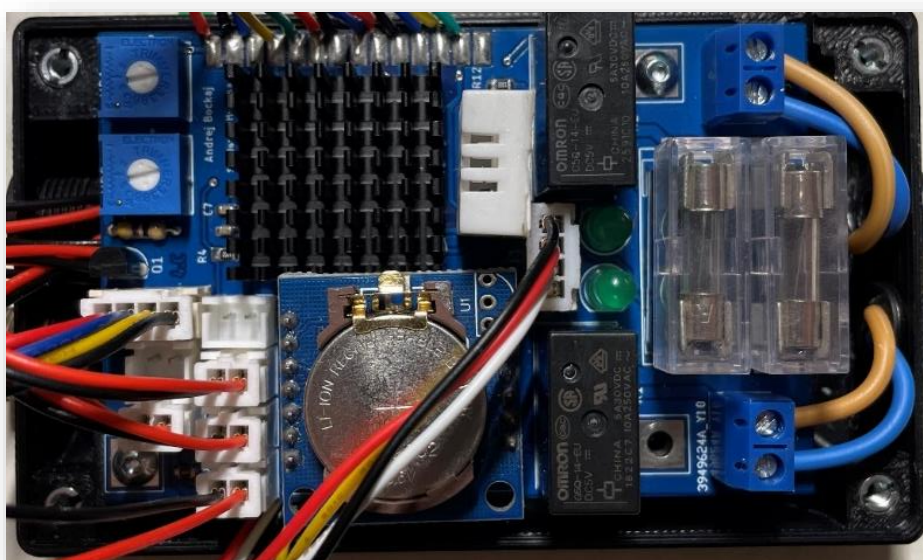


PRÍLOHA I PRVÉ VERZIE OVLÁDACEJ JEDNOTKY

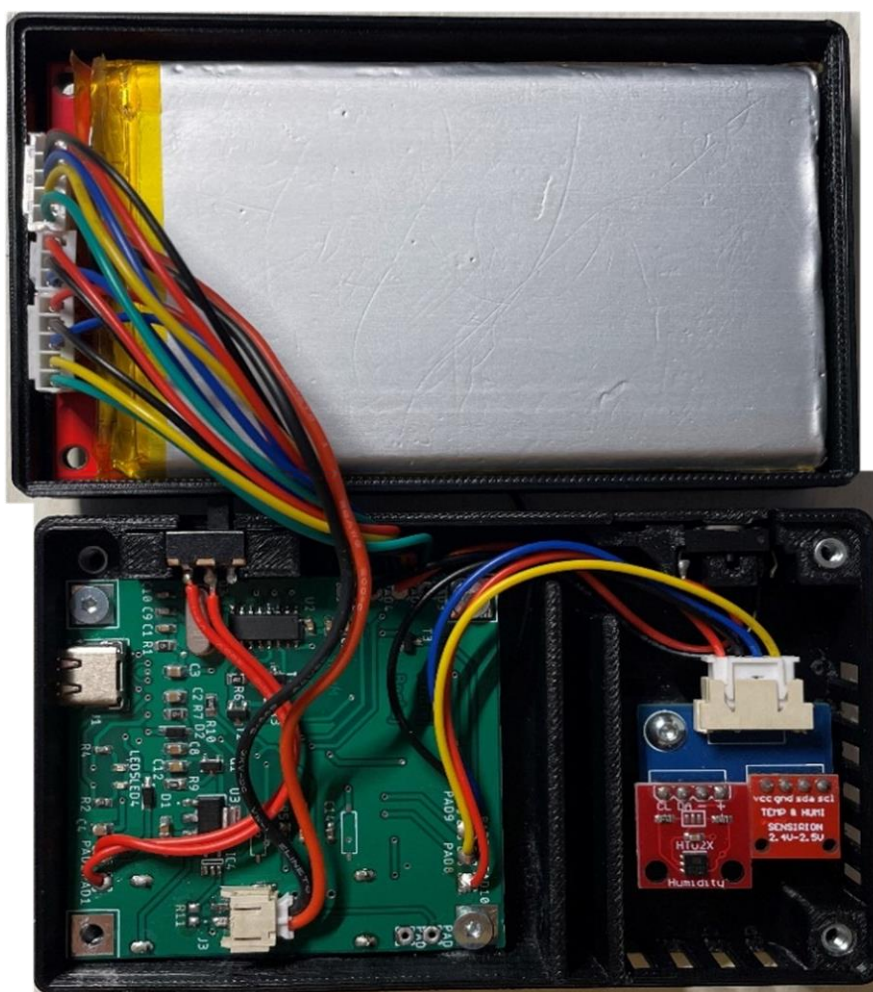




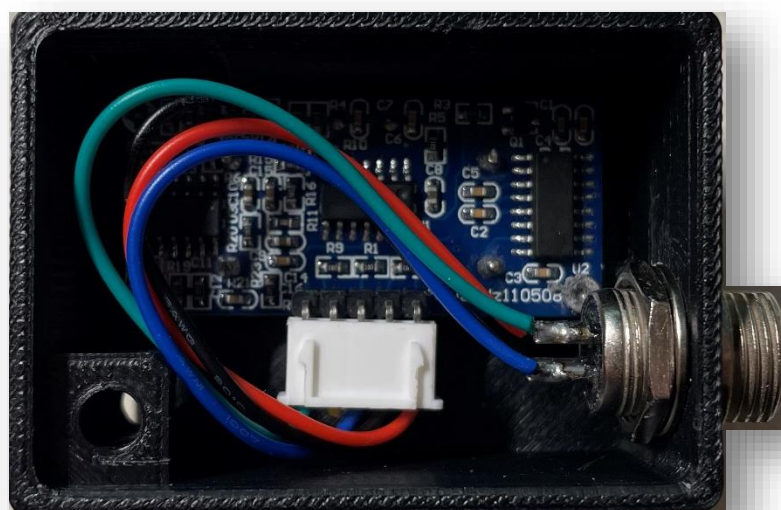
*PRÍLOHA J PRAKTICKÁ REALIZÁCIA OVLÁDAČEJ  
JEDNOTKY*



*PRÍLOHA K PRAKTICKÁ REALIZÁCIA  
TERMOSTATU*

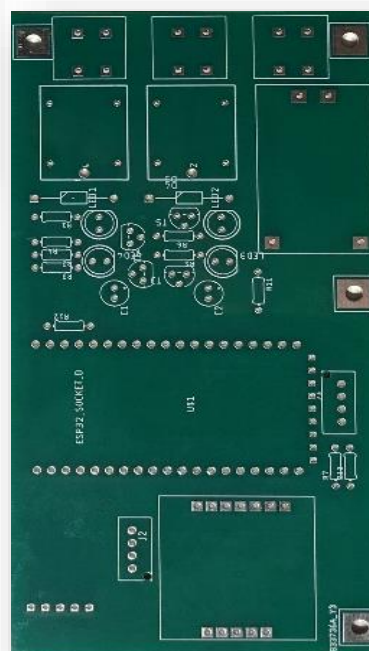
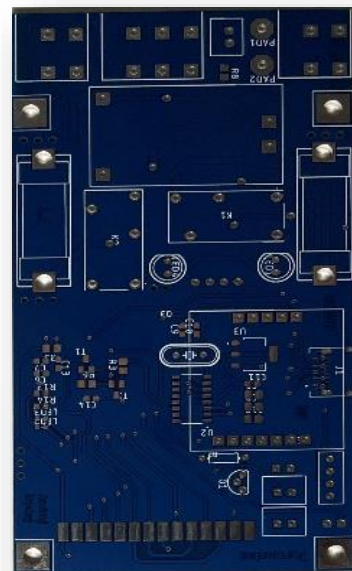
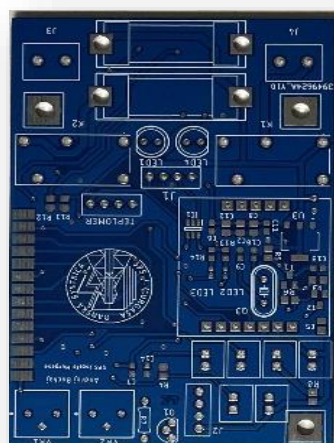


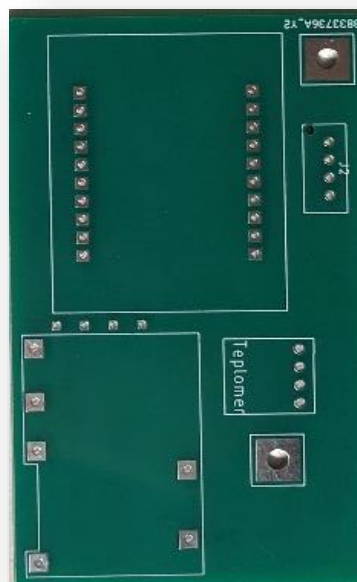
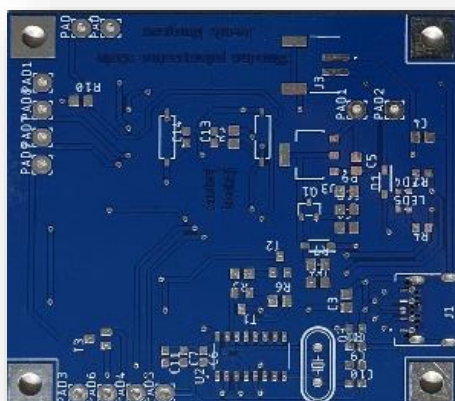
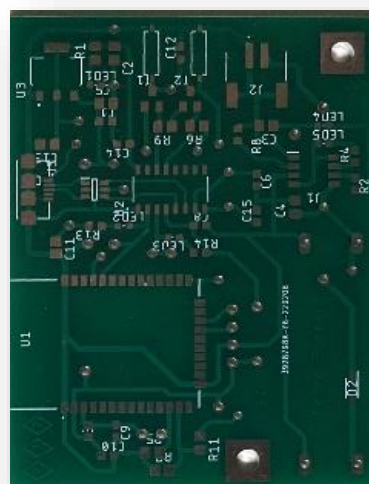
*PRÍLOHA L PRAKTICKÁ REALIZÁCIA ULTRAZVUKOVÉHO SNÍMAČA*





PRÍLOHA M VÝVOJ DPS OVLÁDACEJ JEDNOTKY





*PRÍLOHA O VÝVOJ KONŠTRUKCIE TERMOSTATU*



*PRÍLOHA P VÝVOJ KONŠTRUKCIE OVLÁDACEJ JEDNOTKY*



*PRÍLOHA Q ZDROJOVÝ KÓD OVLÁDACEJ JEDNOTKY - VIĎ ELEKTRONICKÁ FORMA*

*PRÍLOHA R ZDROJOVÝ KÓD TERMOSTATU - VIĎ ELEKTRONICKÁ FORMA*