

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
STROJNÍ**



**BAKALÁŘSKÁ
PRÁCE**

2019

**ŘÍZENÍ KOTLE POMOCÍ
JEDNODESKOVÉHO
MIKROPOČÍTAČE ARDUINO**



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jahoda** Jméno: **Ondřej** Osobní číslo: **456758**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Strojírenství**
Studijní obor: **Informační a automatizační technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Řízení kotle pomocí jednodeskového mikropočítače Arduino

Název bakalářské práce anglicky:

Control of a boiler with single-board microcontroller Arduino

Pokyny pro vypracování:

Popište řízený kotel
Popište jednodeskový mikropočítač Arduino
Navrhněte hardware nutný k řízení kotle
Navrhněte algoritmus řízení kotle

Seznam doporučené literatury:

Pavel Beneš, Branislav Lacko, Ladislav Maixner a kol. Automatizace a automatizační technika 2. Praha : Computer Press, 2013, ISBN 9788025141069
Zbyška Voda: Průvodce světem Arduina

Jméno a pracoviště vedoucí(ho) bakalářské práce:


Ing. Stanislav Vrána, Ph.D., U12110.3


Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

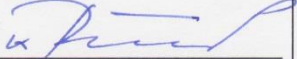
Datum zadání bakalářské práce: **26.04.2019**

Termín odevzdání bakalářské práce: **12.06.2019**

Platnost zadání bakalářské práce:


Ing. Stanislav Vrána, Ph.D.
podpis vedoucí(ho) práce


podpis vedoucí(ho) ústavu/katedry

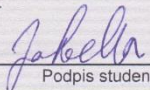

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

26-04-2019

Datum převzetí zadání


Podpis studenta

Poděkování

Rád bych poděkoval svému otci, Ing. Petru Jahodovi, za jeho pomoc a rady při tvorbě této bakalářské práce. Dále mé díky patří i mému vedoucímu Ing. Stanislavovi Vránovi, Ph.D. a v neposlední řadě Bc. Magdaléně Mudrochové za korekci gramatiky a formátování.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně. Dále prohlašuji, že jsem všechny použité zdroje správně a úplně citoval a uvádím je v příloženém seznamu použité literatury.

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

V Praze dne: 6. června 2019

Podpis:

Abstrakt

Cílem této bakalářské práce je sestavit funkční řešení, které řídí vytápění rodinného domu za pomoci dvoustavového kotle a mikropočítače Arduino.

V teoretické části jsou obsaženy možné prostředky automatizace a souhrn použitého hardwaru i softwaru včetně kotle samotného.

V praktické části je popsán program nahraný v mikrokontroleru, konkrétní řešení zapojení a určení k čemu dané části v tomto projektu slouží.

Abstract

Goal of this bachelor's thesis is to get fully working solution of family house heating system with two state boiler and microcontroller Arduino.

In theoretic part are described automation instruments and complete hardware and software used in this project, including boiler itself.

In practical part are described parts of program in Arduino, meanings of each hardware part and specific connections and cable management.

Obsah

1	Úvod	10
1.1	Slovo úvodem	10
1.2	Volba tématu	10
1.3	Cíl práce	10
2	Automatizace	13
2.1	Arduino	13
2.2	Raspberry Pi	14
2.3	PLC	15
3	Kotel VIADRUS G 27 ECO	16
4	Použitý hardware	17
4.1	DCcEle DCcduino MEGA 2560	17
4.2	RTC Hodiny reálného času DS3231	19
4.3	Arduino Ethernet LAN modul	20
4.4	Arduino čtečka microSD karet	21
4.5	Arduino relé 4 kanály 5 V s optočlenem	21
4.6	Display I2C 20x4 a řadič	22
4.7	Nepájivé kontaktní pole ZY-170 R	23
4.8	Teploměr a vlhkoměr DHT22	23
4.9	Teplotní čidlo DS18B20	24
5	Použitý Software	26
5.1	Fritzing	26
5.2	Arduino IDE	26
5.3	Programovací jazyk Wiring	27
6	Funkcionalita komponent	29
6.1	Relé	29
6.2	Teplotní čidla	29
6.3	Ethernetový modul	30
6.4	Hodiny reálného času	30
7	Popis částí programu	32
7.1	Implementace potřebných knihoven	32
7.2	Časy pro časovač	32
7.3	Logika vypínání a zapínání kotle a čerpadla	33

7.4	Ukládání hodnot na SD kartu.....	37
8	Zapojení	38
8.1	Zdroj	38
8.2	Schéma	39
9	Výstup programu	41
10	Závěr	43

Seznam obrázků

Obrázek 1 - Kotel Viadrus G27 ECO [5]	16
Obrázek 2 - Arduino MEGA2560 [11]	19
Obrázek 3 - Hodiny reálného času [12]	20
Obrázek 4 - Ethernet LAN modul [14]	20
Obrázek 5 - čtečka microSD karet [16]	21
Obrázek 6 - Relé modul [17]	22
Obrázek 7 - Display [19]	22
Obrázek 8 - Nepájivé kontaktní pole [20]	23
Obrázek 9 - Čidlo DHT22 [22]	24
Obrázek 10 - teplotní čidlo DS18B20 [24]	25
Obrázek 11 - Schéma zapojení	40
Obrázek 12 - Graf teplot 6.2.2019	41
Obrázek 13 - Graf teplot 6.2.2019 - 20:00 - 21:00	42

Seznam zkratek

PMW	Pulse Width Modulation
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
RISC	Reduced Instruction Set Computer
AVR	Advanced Virtual RICS
EEPROM	Electrically Erasable Programmable Read-Only Memory
SRAM	Serial Random-Access Memory
RTC	Real Time Clock
PCB	Printed Circuit Board
IDE	Integrated Development Enviroment
PLC	Programable Logic Controller
STL	Statement List
LD	Ladder Diagram
FBD	Functional Block Diagram
GND	Ground

1 ÚVOD

1.1 Slovo úvodem

V dnešní době plné automatizace a snahy o co nejefektivnější řešení problémů je vytápění rodinného domu beze sporu jedna z věcí, na které většina domácností může ušetřit. Složitá a drahá řešení od výrobců, která nejsou vyrobena přesně na míru domu, nejsou vždy optimální volbou. Přístroje v hodnotě několika desítek tisíc korun může každý z nás nahradit za levnější, starší model kotle a Arduino s několika rozšiřujícími moduly a tím celé vytápění optimalizovat podle své potřeby na míru domu pouze za zlomek ceny řešení od výrobce.

1.2 Volba tématu

Volbu tématu ovlivnila zejména vysoká poptávka po prvcích chytré domácnosti a touha po úspoře výdajů za vytápění, která v současnosti nelze uspokojit produkty dostupnými na trhu. Produkty, které jsou v současné době volně k zakoupení, nenabízejí téměř žádnou možnost individuálního rozšíření. Například možnost rozšířit projekt za hranice řízení kotle a postupovat v žebříčku inteligentní domácnosti dále směrem k minimalizaci nutných nákladů na domácnost.

1.3 Cíl práce

Hlavním cílem mé bakalářské práce je sestavit plně funkční automatický systém, který dokáže, bez jakéhokoliv zásahu člověka, řídit vytápění celého

domu podle teploty v jedné místnosti, a to včetně korekce teplot způsobených dopravním zpožděním kotle – doba od začátku, respektive konce ohřívání vody v oběhu do doby, než se tato změna projeví na teplotě ve vytápěném pokoji. Dále v tomto projektu implementuji tabulku časů, podle kterých bude kotel topit v daný den a čas v týdnu. A v neposlední řadě se snažím v projektu o maximalizaci účinnosti vytápění.

TEORETICKÁ ČÁST

2 AUTOMATIZACE

Pojem automatizace označuje použití řídicích systémů k ovládání zařízení a procesů s co možná nejmenší potřebou přítomnosti člověka. Avšak samotná zařízení vyžadují něco, kam pošlou vyhodnocená data a od čeho jim přijde příkaz, co mají dělat. Tuto úlohu plní řídicí modul s náležitým programem. K řídicímu modulu se připojí všechny periférie projektu (senzory, čidla atp.), který posléze funguje jako mozek celého procesu.

Řídicí modul může být malý a levný, jako třeba Arduino, ideální pro domácí automatizační projekty. Nebo třeba Raspberry Pi, který je vhodný pro pokročilejší projekty. Řídicí modul však může být i velice drahý a sofistikovaný, který je určený především pro průmyslové automatizace - PLC.

2.1 Arduino

Arduino je klasický jednodeskový mikrokontroler. Hodí se především pro domácí řízení a regulace. Větší a složitější aplikace zde mohou narazit na problém, že Arduino není klasický počítač. Celý kód je vyvíjen na jiném zařízení a běží v nekonečné smyčce, a to může být pro větší aplikace nepřekonatelná překážka.

Existuje mnoho typů Arduina, které se liší velikostí, pamětí, konektory, cenou, počtem vstupů, respektive výstupů a v mnoha dalších aspektech. Samotné Arduino pak lze rozšířit o funkce, které samotný mikrokontroler nemá, pomocí tzv. Arduino Shieldů. Tyto Shildy mohou přidat Arduinu funkce jako je možnost ukládání dat na SD kartu (pokud již tuto funkci nemá

samotný mikrokontroler), možnost připojení se k internetu přes ethernetový kabel či Wi-Fi a spoustu dalších.

Arduino je kompletně open-source a má obrovskou podporu komunity. Existuje mnoho internetových fór, ve kterých lze nalézt vše o Arduino, od detailních návodů až po vyřešené problémy a časté zádrhele, které se uživatelům stávají [1, 2].

2.2 Raspberry Pi

Stejně jako Arduino je i Raspberry Pi jednodeskový počítač navržený tak, aby měl přibližně velikost platební karty. Obdobně jako Arduino má velkou podporu komunity a lze o Raspberry Pi na fórech dohledat veškeré potřebné informace. Používá se k domácímu řízení a také často ke školní výuce. Na rozdíl od Arduina Raspberry Pi lze považovat za slabší klasický počítač, a proto je mnoha uživateli využíván například jako domácí server, nebo přehrávač videí atp. Dále také lze, po připojení nezbytných periférií jako jsou klávesnice, myš a monitor, na Raspberry Pi vyvíjet přímo kód.

Raspberry Pi má již mnoho generací od doby prvního modelu. Nejnovější Raspberry Pi 3, vydané v roce 2016, je již výkonem porovnatelné se slabším stolním počítačem nebo výkonným smartphonem.

Podobně jako Arduino má i tento mikropočítač mnoho možností, jak si pomocí přídatných modulů rozšířit funkce, avšak s tím rozdílem, že už často samotná deska obsahuje např. slot na SD kartu nebo ethernetový port.

Raspberry Pi běží na operačním systému Raspbian a je pod ochrannou známkou, takže klony se objevují v jiné barvě a je o poznání složitější si

nechat vyrobit vlastní mikropočítač dělaný na míru vytvářeného projektu [1, 3].

2.3 PLC

PLC neboli programovatelný logický automat je průmyslový počítač bez operačního systému navržený k řízení automatického procesu (ovládání výrobních linek a řízení strojů) v reálném čase. PLC je relativně malý počítač, který stejně jako Arduino běží v cyklech. Obecně se o programovatelných logických automatech dá říct, že jsou velmi spolehlivé a vydrží v provozu i desítky let nepřetržitého provozu. Cenově se PLC pohybuje od desítek tisíc až po stovky tisíc Kč.

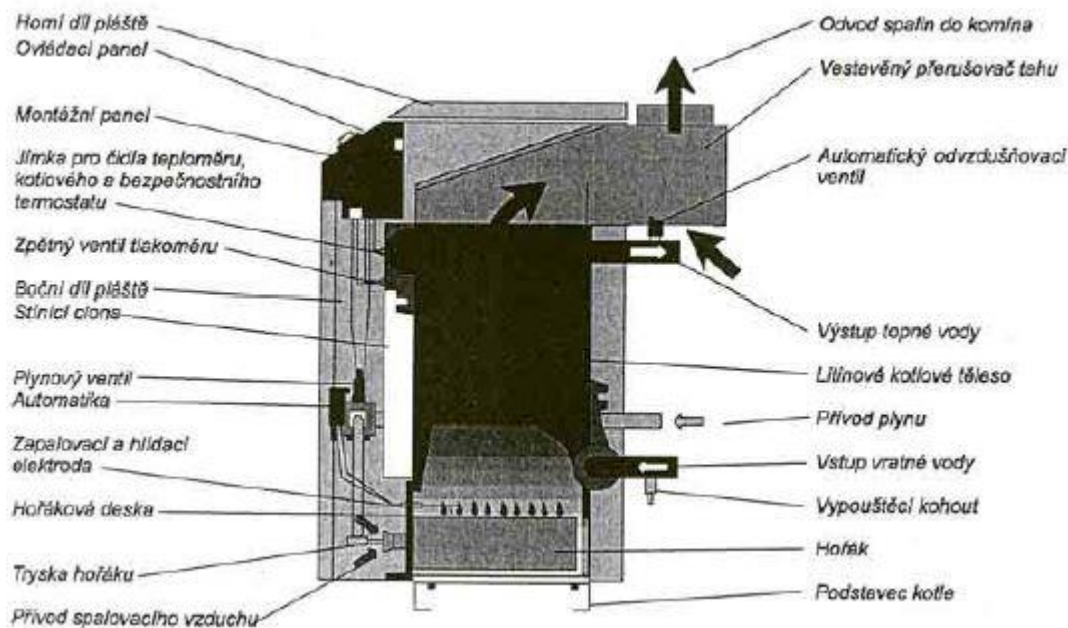
K programování PLC se používá software jako ProLeit, Tia Portal nebo předchůdce Simatic Step 7, oproti kterému má ale Tia Portal výhodu, že v něm lze nastavit i všechny použitý hardware a nadesignovat vizualizaci ovládacího panelu. Což pokud je PLC programováno ve Step 7, musí programátor udělat v jiném programu, například Simatic WinCC. Ohledně programovacího jazyku se nejčastěji používá řádkový jazyk Statement list (STL), nebo grafické jazyky Ladder diagram (LD) a Functional block diagram (FBD) [4].

Hlavním výrobcem PLC je firma Siemens, ale jsou i jiní výrobci, jako například Teco, Honeywell a další.

3 KOTEL VIADRUS G 27 ECO

Tento litinový čtyř-člankový plynový kotel je vybaven atmosférickým hořákem (spaluje plyn smíšený se vzduchem při tlaku 101 HPa) a je určen pro spalování nízkotlakého zemního plynu a propanu. Viadrus G27 ECO je teplovodní s nuceným oběhem topné vody a pracovním přetlakem 0,4 MPa. Výrobce udává účinnost kotle 92 %.

Tento kotel je relativně starý a jednoduchý oproti kotlům, které se prodávají v dnešní době. Na rozdíl od moderního kotle, který zvládá zahřívání oběhovou vodu jen na určité procento svého výkonu, Viadrus G 27 ECO má pouze dva stavy, a to zapnuté nebo vypnuté vyhřívání oběhové vody [5]. Dále má kotel k dispozici čerpadlo, které pohání oběh vody celým systémem. Čerpadlo je naprosto nezbytné pro rozvod ohřáté vody do všech radiátorů celého domu.



Obrázek 1 - Kotel Viadrus G27 ECO [5]

4 POUŽITÝ HARDWARE

Tato kapitola obsahuje popis veškerého hardware užitého při tomto projektu a jejich specifikaci.

4.1 DCcEle DCcduino MEGA 2560

Tento jednodeskový mikropočítač běžící na procesoru ATmega od firmy Atmel je 100 % klon jednodeskového mikropočítače Arduino MEGA 2560 se stejným procesorem.

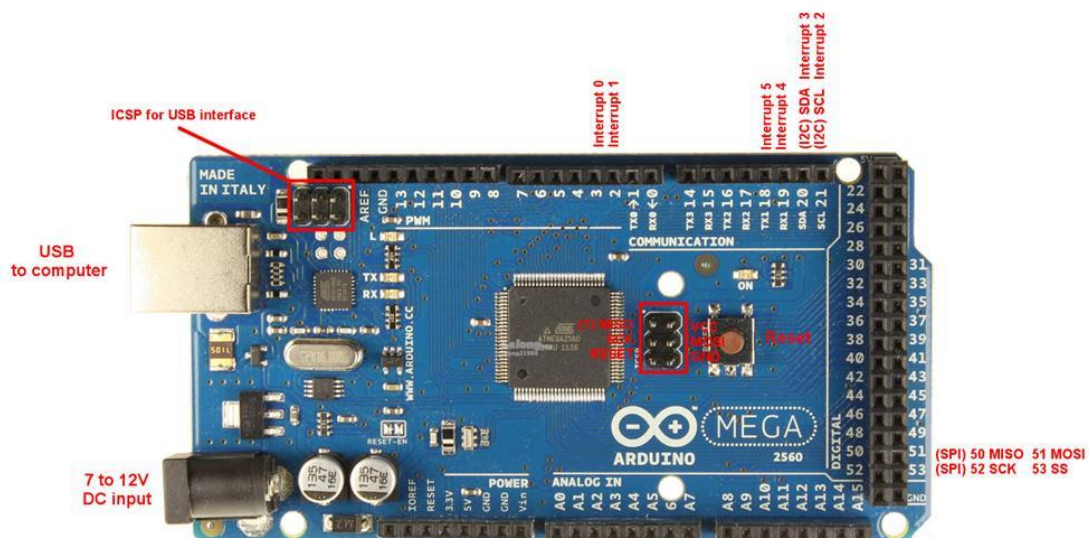
DCcduino MEGA 2560 obsahuje celkem 54 digitálních inputů, respektive outputů, z nichž 14 lze použít jako pulzně šířkově modulovaný (PMW) output. To je typ kódování analogového signálu pomocí dvoustavového signálu jako poměr mezi stavy zapnuto/vypnuto. Tomuto poměru se říká tzv. střída. Tím se docílí přenosu relativní hodnoty v rozsahu 0-100% [6].

Dále má mikropočítač k dispozici 16 analogových vstupů, 4 vstupy UART, neboli universální asynchronní přijímač/vysílač. To je část DCcduina zajišťující asynchronní komunikaci pomocí pinů RX a TX, které jsou zodpovědné za přijímání a odesílání dat [7]. DCcduino také obsahuje 16 MHz krystalový oscilátor, který plní funkci hodin, 6 pinů sériového periferního rozhraní (SPI), které komunikují mezi řídicím mikroprocesorem a ostatními integrovanými obvody [8]. Všechny piny operují s 5 V a jsou schopny poskytnout, respektive přijmout až 40 mA. A v neposlední řadě má DCcduino USB konektor, napájecí konektor a resetovací tlačítko [9].

DCcduino MEGA 2560 běží na procesoru ATmega 2560. Tento 8bitový procesor zprostředkovává 256 kB programovatelné paměti s AVR jádrem. AVR

je označení 8bitových mikročipů typu RISC (počítač s redukováným souborem instrukcí) s Harvardskou architekturou od firmy Atmel. AVR procesory se skládají ze 32 stejných 8bitových registrů obsahujících jak adresy, tak data [10].

Procesor ATmega 2560 má 256 kB programovatelné paměti Flash, 4 kB trvalé paměti dat EEPROM (elektronicky vymazatelná paměť pouze pro čtení) a 8 kB paměti SRAM. Všechny tyto paměti jsou uvnitř čipu AVR. Flash paměť je programovatelná a rozdělená po blocích, z čehož každý blok lze plnit informacemi nezávisle na ostatních. Výhodou této paměti je, že ji lze libovolně znovu přeprogramovat a obsah zůstane zachován i po odpojení napájení (nevolatilní). Stejně jako paměť Flash i EEPROM zůstane zachována po vypnutí napájení, ale EEPROM uchovává různá nastavení za běhu mikrokontroleru, nikoliv program samotný. SRAM neukládá data po vypnutí napájení (volatilní) a slouží k zápisu i čtení právě běžícího programu [11].



Obrázek 2 - Arduino MEGA2560 [11]

4.2 RTC Hodiny reálného času DS3231

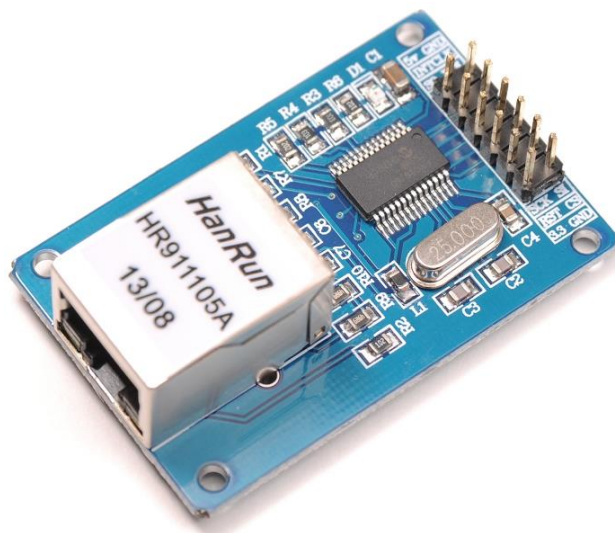
Modul reálného času (RTC) DS3231 je rozšíření pro Arduino, které uchovává informace o aktuálním čase a datu. Toto zařízení podporuje sekundy, minuty, hodiny, dny, měsíce i roky. Je založeno na teplotně kompenzovaném krystalovém oscilátoru a krystalu a dokáže držet přesný čas s téměř nulovou odchylkou. V modulu je nutná součást baterie typu LIR2032 nebo CR2032. Celý modul potom pracuje při provozním napětí 3,3 až 5,5 V [12].



Obrázek 3 - Hodiny reálného času [12]

4.3 Arduino Ethernet LAN modul

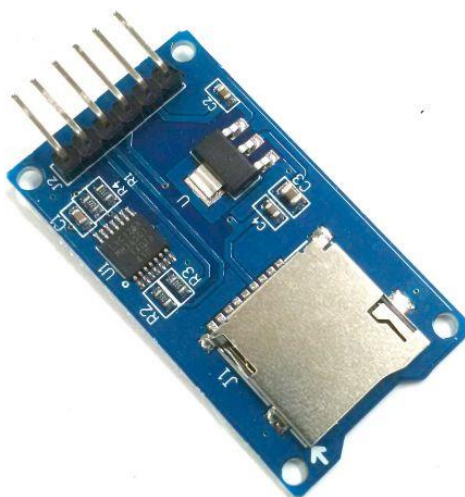
Tento modul typu HR911105A umožní platformám jako Arduino nebo Raspberry Pi připojení k LAN síti. Komunikuje s Arduinem pomocí protokolu SPI. Provozní napětí je 3,3 V, nicméně zvládne i 5 V díky zabudovanému napěťovému převodníku [13, 14].



Obrázek 4 - Ethernet LAN modul [14]

4.4 Arduino čtečka microSD karet

Modul čtečky karet umožňuje číst, respektive zapisovat data a podporuje jak SD, tak SDHC karty (vysoko-rychlostní karty). Komunikuje s Arduinem pomocí protokolu SPI. Napájen může být buď 5 V nebo 3,3V [15, 16].



Obrázek 5 - čtečka microSD karet [16]

4.5 Arduino relé 4 kanály 5 V s optočlenem

Tento modul je nejzákladnější prvek automatizace – spínací prvek. Je ovládaný přímo 5 V z Arduina a obsahuje diodu, která signalizuje, zda je prvek sepnutý nebo rozepnutý. Modul se velice jednoduše ovládá a to tak že, pokud Arduino pošle na výstupní pin 5 V (HIGH), relé se sepne. Pokud pošle 0 V (LOW), relé je rozepnuté. Princip funkce relé spočívá v tom, že v klidovém stavu díky pružině kotva nespojuje obvod. Zatímco když je přivedeno 5 V do ovládacího obvodu (cívky), vytvoří se elektromagnet, který kotvu přitáhne a ta propojí řízený obvod [17].



Obrázek 6 - Relé modul [17]

4.6 Display I2C 20x4 a řadič

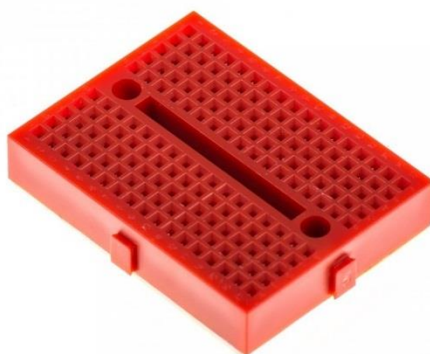
Display má k dispozici 20 znaků x 4 řádky a má žluté podsvícení na zeleném poli. Stejně jako většina modulů pro Arduino je napájen 5 V. Pro ovládání displaye jsou možné dvě varianty. Buď připojit minimálně 10 vodičů z 16ti pinového hřebínku, respektive 6 datových vodičů do Arduina, nebo využít řadič PCF8574, který funguje jako 8 bitový převodník na I2C sběrnici. Díky tomu stačí připojit 4 vodiče do Arduina, respektive 2 datové vodiče [18, 19].



Obrázek 7 - Display [19]

4.7 Nepájivé kontaktní pole ZY-170 R

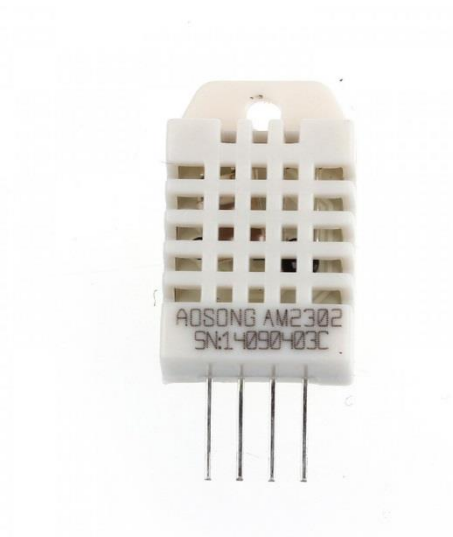
Toto nepájivé pole má 170 pinů rozdělených do 34 řad po pěti vzájemně propojených pinech. Výhoda nepájivého pole je ve snadné konstrukci i destrukci obvodů bez nutného pájení pouze za pomoci propojovacích drátků [20].



Obrázek 8 - Nepájivé kontaktní pole [20]

4.8 Teploměr a vlhkoměr DHT22

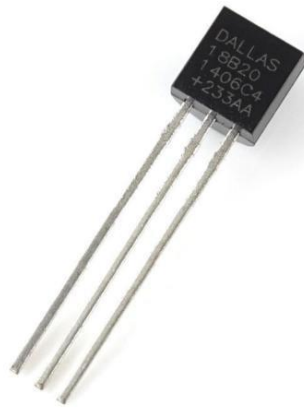
Teplotní čidlo DHT22 měří jak teplotu, tak vlhkost. Má celkem 4 piny, ale stačí připojit pouze tři, neboť dva z nich jsou GND, čili zem. Nicméně je nutné umístit jeden 10k odpor mezi datový a zdrojový kabel. Čidlo zvládá měřit teploty v rozmezí -40 až +80°C s přesností $\pm 0,5$ °C a vlhkost v rozsahu 0- 100% s přesností $\pm 2\%$ [21, 22].



Obrázek 9 - Čidlo DHT22 [22]

4.9 Teplotní čidlo DS18B20

Toto teplotní čidlo dokáže měřit teplotu mezi -55°C a 125°C , avšak při rozsahu -10°C až 85°C má přesnost $\pm 0,5^{\circ}\text{C}$. Dále toto čidlo zvládá takzvanou parazitní komunikaci. To znamená, že stačí připojit pouze dva dráty (ze tří) k Arduinu, a to propojením napájecího pinu s pinem země a připojením obou na zem Arduina. Datový pin senzoru připojíme přes $4700\ \Omega$ odpor ke zdroji a zároveň na komunikační pin Arduina. Dále toto čidlo podporuje OneWire komunikaci. Díky unikátnímu 64-bitovému sériovému číslu každého čidla, můžeme komunikovat s více čidly po jednom drátu [23, 24].



Obrázek 10 - teplotní čidlo DS18B20 [24]

5 POUŽITÝ SOFTWARE

5.1 Fritzing

Fritzing je OpenSource program umožňující vytváření snadných a přehledných výkresů projektů založených na mikrokontroleru Arduino a nepájivém poli. Program byl vytvořen v Německu na univerzitě aplikovaných věd v Postupimi. Fritzing nabízí hned tři pohledy na projekt. Pohled "montážní deska", "schéma" a "deska plošných spojů" (PCB). Pohled "montážní deska" je pohled reálně vypadajících součástek zadrátovaný stejně jako ve fyzickém projektu, kterého děláme výkres. Pohled "schéma" je zobrazení každé součásti jako je soubor vstupů a výstupů. A poslední pohled "deska plošných spojů" je vygenerovaný pohled, jak by vypadal tištěný spoj celého zapojení [25].

Výkres se vždy začíná kreslit v pohledu "montážní deska". Program automaticky vygeneruje ostatní dva pohledy, avšak je třeba ještě zkontrolovat pohled "schéma" a ujistit program o tom, že každý drát je tak, jak má být. Na pohledu "deska plošných spojů" není třeba nic měnit, takže je výkres hotový a máme vše, co je třeba k návrhu tištěného spoje [25].

5.2 Arduino IDE

Arduino IDE je vývojové prostředí k programování mikrokontrolerů typu Arduino. Program se píše v jazyce Wiring. Prostředí je velice intuitivní a obsahuje řadu příkladů jednoduchých programů a základní knihovny. Knihovny lze ovšem i nahrát vlastní, pokud není výběr implicitních knihoven dostatečný [26].

5.3 Programovací jazyk Wiring

Wiring je programovací jazyk, sloužící k programování jednodeskových mikropočítačů typu Arduino bez konkrétních znalostí hardwaru. Byl vymyšlen v roce 2003 Hernandem Barragánem na Institutu Interaktivního Designu ve městě Ivrea v severní Itálii. Wiring nevznikl přímo pro vývojový kit Arduino, ale pro velmi podobný vývojový kit a vychází z dalšího open-source projektu Processing.

Tento programovací jazyk je navržen a nazván tak, aby se daný projekt pouze propojil drátky, například přes nepájivé pole, a hned naprogramoval a spustil. Základní struktura jazyku je tvořena dvěma částmi. A to částí `setup()`, která se vždy vykoná jednou za běh programu a částí `loop()`, která běží stále dokola po celou dobu běhu programu. Využíváme ovšem i místo mimo tyto dvě části, a to například na připojení knihoven nebo definici globálních proměnných. Jazyk Wiring má velmi blízko k jazyku C++. Arduino IDE dokonce obsahuje převodník C++ do Wiring, tudíž je možné programovat Arduino přímo v C++, ale to se dělá většinou jen v pokročilejších a delších programech.

Nejběžnější software k programování v jazyce Wiring je výše zmíněné Arduino IDE, ale existují i jiné programy k naprogramování mikrokontrolerů, jako například Arduino Eclipse IDE, který je dělaný na větší a složitější programy [27].

PRAKTICKÁ ČÁST

6 FUNKCIONALITA KOMPONENT

V této kapitole popisuji, k čemu používám dané komponenty a proč jsou v projektu zahrnuty.

6.1 Relé

V celém projektu jsou použita pouze dvě relé i přesto, že v samotném rozšiřujícím modulu Arduina jsou hned čtyři. Další volná relé jsou připravena na budoucí rozšíření projektu.

Spínací obvod relé je připojen k datovému pinu Arduina, 5 V zdroji (viz kapitola 8.1) a k uzemnění. Zatímco spínaný obvod je připojen ke klasické domácí síti 230 V a k uzemnění. Díky tomu může Arduino pomocí 5 V sepnout obvod a ovládat spínaný okruh s 230 V.

Spínaný okruh prvního relé vede přímo na hořák kotle, což znamená, že v případě sepnutí okruhu se spustí hořák a začne se ohřívat oběhová voda. Zatímco druhé relé propojuje okruh vedoucí do čerpadla kotle a tím zapíná oběh vody celým topným systémem skrz všechny radiátory.

6.2 Teplotní čidla

K celému řízení jsou za potřebí pouze tři teplotní čidla, nicméně jsem pro porovnání přidal i čtvrté čidlo ven.

Čidlo DHT22 je umístěné v pokoji a podle teploty v tomto pokoji program reguluje vypínání a zapínání hořáku.

Další dvě čidla DS18B20 jsou připevněna na vstupu a výstupu ohřívané vody v kotli a třetí je venku. Venkovní čidlo má před sebou dostatečný zákryt,

aby v žádnou denní hodinu nebylo přímo osvětleno sluncem, což by značně ovlivňovalo měření.

Jelikož právě ohřátá voda má jinou teplotu než voda, která už proběhla celým okruhem a všemi radiátory, je nezbytné mít čidla na vstupu i výstupu. Díky tomu je možné optimalizovat dobu zahřívání ohřevné vody a vypnout hořák dřív než čerpadlo v případě, že se teplota v pokoji blíží žádané teplotě. Díky tomu dojde k lepšímu využití energie uložené v ohřevné vodě, než kdyby se hořák i čerpadlo vypínaly ve stejnou dobu (více v kapitole 7.3).

6.3 Ethernetový modul

Tento modul Arduina je zde zahrnut pouze pro budoucí rozšíření.

6.4 Hodiny reálného času

I přes to, že samotné Arduino MEGA obsahuje krystalický oscilátor k určování reálného času, jsem v kapitole 4.2 uvedl, že používám přídatný modul hodin reálného času.

Důvodem je, že jsem před přidáním RTC modulu zaznamenával obrovské časové odchylky od reálného času, a to v rámci několika minut za týden, což je bohužel pro konzistentní fungování projektu naprosto nepřijatelné. Tudiž bylo nezbytné přidat do projektu přesnější určování reálného času. Tuto funkci zde plní modul RTC DS3231, u kterého je odchylka od reálného času téměř nulová.

K určování reálného času by také bylo možné použít ethernetový modul a internet. Tuto variantu jsem ovšem nezvolil, protože ethernetový modul je

pouze zapojený, nikoliv zprovozněný a komunikace přes ethernet není v tomto projektu zahrnuta.

7 POPIS ČÁSTÍ PROGRAMU

V této kapitole podrobně vysvětlím a popíšu celý program nahraný v Arduino, zmíním neúspěšné pokusy a vysvětlím, proč byly neúspěšné.

7.1 Implementace potřebných knihoven

Implementace knihoven je naprosto nezbytná část programu, která přidává Arduino rozšiřující funkce, díky kterým je Arduino schopno například komunikovat s přídatnými moduly, což samo o sobě není možné.

V jazyce Wiring se to dělá pomocí příkazu `#include <>`. Přičemž se mezi znaky větší a menší vepíše název přidávané knihovny. Tento příkaz je v programu Arduino IDE generován automaticky po nahrání knihovny.

Já v programu implementuji knihovny ke komunikaci s teplotním čidlem a vlhkoměrem DHS22 (`dht.h`), teplotním čidlem DS18B20 (`OneWire.h`), SD kartou (`Fat16.h`), modulem reálného času (`DS1307.h`), I2C komunikací pro display (`Wire.h` a `LiquidCrystal_I2C.h`) a k operování s EEPROM pamětí (`EEPROM.h`).

7.2 Časy pro časovač

Časovač jsem v projektu zahrnul pro funkci jednorázového naprogramování časů, kdy má kotel topit na danou teplotu. Díky tomu bude mít kotel přesně dané časy a teploty, jak má topit a tudíž nepotřebuje žádný zásah člověka a udržuje naprogramovaný týdenní rytmus.

Pro realizaci časovače používám jednu proměnnou typu float se čtyřmi řádky a 16 ti sloupci. Řádky popisují den v týdnu, čas, kdy začít topit, čas, kdy přestat topit a žádanou teplotu v pokoji.

Řádek dnů v týdnu je definován jako 0 = vše, 1 = pondělí, 2 = úterý, 3 = středa, 4 = čtvrtek, 5 = pátek, 6 = sobota, 7 = neděle, 8 = hodnota z webu. Osmý člen prvního řádku je zde připraven na rozšíření projektu o přenastavení hodnot této proměnné přes internet. To ale v této práci není zahrnuto. Každý den (všechny indexy řádku kromě indexů 0 a 8) je v tomto řádku zapsán dvakrát. To slouží k nastavení teploty při vstávání a příchodu domů ze školy, nebo třeba práce. V mezičase, kdy nikdo není v domě, topí kotel na jinou teplotu. Ostatní řádky proměnné už jen kopírují formát daný prvním řádkem tak, aby si všechny dny, časy a teploty odpovídaly. Což znamená, že každý den v týdnu, čas kdy začít topit, čas kdy přestat topit a teplota jsou sepsané v jednotlivých sloupcích.

Po experimentálním pozorování jsem zjistil, že je třeba začít topit 40 minut před dobou, než má být v pokoji určená teplota a stačí přestat topit 10 minut před daným časem, aby teplota v pokoji přesně odpovídala času a teplotě. To je dané dopravním zpožděním zahřívání oběhové vody a musí se to do proměnné zadat po odečtení těchto časů.

7.3 Logika vypínání a zapínání kotle a čerpadla

Celý průběh programu zajišťuje hlavní cyklus `loop()`, který proběhne každých 20 sekund. Ovšem tento cyklus pouze zkontroluje, zda čidla načetla své hodnoty správně a poté volá ostatní podprogramy.

Napřed zavolá podprogram, který načte hodnoty aktuálního času z RTC modulu, poté přečte všechny hodnoty ze všech čidel a zkontroluje, zda byly přečteny správně a jsou v reálném rozmezí a pokud je vše v pořádku, zavolá podprogram *regulace()*. A nakonec zapíše hodnoty na SD kartu a zobrazí na display.

Podprogram *regulace()* určuje, kdy se má a smí kotel či čerpadlo zapnout a kdy se má a smí kotel či čerpadlo vypnout. To z něj dělá nejdelší a nejsložitější část celého programu. Podprogram napřed zjistí z tabulky časovače, na jakou teplotu by v tuto chvíli měl topit (vždy nějakou hodnotu musí nalézt) a uloží si ji. Následuje série podmínek určujících, kdy kotel smí topit a kdy nesmí. To jsem nastavil tak, aby kotel nikdy netopil víc než pět minut, pokud teplota na výstupu kotle překročí o 0,1 °C teplotu v pokoji. Tato podmínka má omezit vliv dopravního zpoždění topného systému. Pokud tento stav nastane, pak kotel nesmí osm minut topit. Poté následuje část, která po splnění všech předešlých podmínek a podmínky, že teplota vody na výstupu nepřesahuje maximální teplotu (47 °C, kvůli rosnému bodu spalin v kotli), sepne relé a tím spustí čerpadlo topné vody a hořák v kotli a zapíše čas zapnutí.

Další části podprogramu slouží k vypínání kotle. Zde je vypsáno mnoho případů, při kterých je třeba kotel vypnout. A pokud některý z případů nastane, relé se rozepne a tím se vypne kotel (nikoliv čerpadlo) a zapíše se čas vypnutí kotle. Čerpadlo pak běží dál dokud není rozdíl teploty vody vystupující z kotle a vstupující do kotle menší než 5 °C a až potom se vypne i čerpadlo. Na samém konci podprogramu je pravděpodobně nejdůležitější část, a to část, která kontroluje, zda není zapnutý hořák kotle, zatímco je

čerpadlo vypnuté. Takový stav se nikdy nesmí stát, nicméně jsem přidal tuto softwarovou ochranu, že pokud tento stav nastane, pak se okamžitě zapne čerpadlo a na SD kartu se zapíše chybové hlášení.

Softwarová ochrana ovšem nemusí vždy stačit kvůli možným chybám programátora, a proto by bylo vhodné jako rozšíření projektu přidat hardwarovou ochranu, pro zabránění, aby takovýto stav nemohl nastat.

```
void loop(void) {
    if (millis() - predchoziMillis > cetnost) { // kazde 20sec
        se vyctou hodnoty cidel a zapisi na SD
        predchoziMillis = millis();
        RTC.get(rtc, true);
        cticidla();
        if (cidlot3 < 85.00) regulace();
        sdzapisdata();
        lcdzobraz();
    }
}

void regulace(void) {
    // casove rele nastavi pozadovanou teplotu podle hodnot v
    poli CASOVAC
    tempfloat = rtc[2] + (float(rtc[1]) / 100.00); // aktualni
    cas [hod.min]
    // rtc[3] je DOW
    for (byte i = 0; i < pocetcasu; i++) {
        if ((tempfloat >= casovac[1][i]) && (tempfloat <
        casovac[2][i]) && ((casovac[0][i] == 0.00) || (casovac[0][i]
        == rtc[3]))) {
            teppokojmin = casovac[3][i];
        }
    }

    // stav reregulace
    tempfloat = 0; //promenna pro opravu casu kdy kotel topi
    pres pulnoc
    if (kdyzavyp[0][0] > rtc[2]) tempfloat = 1440;
    if (60 * kdyzavyp[1][0] + kdyzavyp[1][1] + kotelvyucas < 60
    * rtc[2] + rtc[1] + tempfloat)
        stavregulace = 'A'; // ukonceni stavu prekroceni
    maximalniho casu souvisleho topeni
    if (stavkotle && (60 * kdyzavyp[0][0] + kdyzavyp[0][1] + 5 <
    60 * rtc[2] + rtc[1] + tempfloat) && (teppokojmin - 0.1 <
    cidloh1t))
```

```

    stavregulace = 'P'; // prekroceni maximalniho casu
souvisleho topeni

    // regulace
    // zatop max do tepkotelmax
    if ((cidlot1 < tepkotelmax) || !stavcerpadla) &&
!stavkotle)
        if ((cidlohlt < teppokojmin))
            if (stavregulace < 'P') {
                stavkotle = 1; // topi
                stavcerpadla = 1; // cerpadlo jede
                digitalWrite(rele2, LOW); // cerpadlo zapnout
                delay(1000);
                digitalWrite(rele1, LOW); // kotel zapnout
                delay(2000);
                kdyzavyp[0][0] = rtc[2]; //hod
                kdyzavyp[0][1] = rtc[1]; //min
                kdyzavyp[0][2] = rtc[0]; //sec
            }

    // vypnout kotel v tepkotelmax
    if (stavkotle && ((cidlot1 > (tepkotelmax +
float(hystkotel))) || (((cidlohlt > (teppokojmin +
float(hystpokoj))) || (stavregulace >= 'P'))))) {
        if (cidlot1 > (tepkotelmax + float(hystkotel)))
stavregulace = 'B'; // prekroceni max teploty kotle
        stavkotle = 0; // netopi
        digitalWrite(rele1, HIGH); // kotel vypnout
        delay(2000);
        kdyzavyp[1][0] = rtc[2]; //hod
        kdyzavyp[1][1] = rtc[1]; //min
        kdyzavyp[1][2] = rtc[0]; //sec
    }

    // vypne cerpadlo když rozdíl na kotli < 5°C
    if (!stavkotle && stavcerpadla && ((cidlot1 - cidlot2) <
5.0))
        if (((cidlohlt > teppokojmin))) {
            digitalWrite(rele2, HIGH); // cerpadlo vypnout
            delay(2000);
            stavcerpadla = 0;
        }

    if (stavkotle && !stavcerpadla) { // bezpecnostni kontrola
zapnuti cerpadla
        digitalWrite(rele2, LOW); // cerpadlo zapnout
        stavcerpadla = 1; // cerpadlo jede
        sdzapiserr("err nezapnute cerpadlo");
    }
}

```

7.4 Zobrazení hodnot na display

V první fázi, při inicializaci, display zobrazí pouze text „Regulace kotle“ a „Made by OJ“. V druhé fázi po inicializaci program smaže vše, co je doposud zobrazeno a na display zobrazí všechny teploty teplotních čidel DS18B20 (teplota na vstupu a výstupu oběhové vody a venkovní teplota) a proměnnou *stavchyb*, která signalizuje, jestli se vyskytla nějaká chyba v průběhu programu. Hodnota „A“ značí, že se žádná chyba nevyskytla, hodnota „B“ značí, že se vyskytla chyba v programu a hodnota „C“ říká, že se vyskytla chyba při zápisu na SD kartu. Dále zobrazí teplotu a relativní vlhkost z čidla DHT22 (teplota a vlhkost pokoje, podle kterého se reguluje) společně s časy zapnutí a vypnutí kotle. A jako poslední zobrazí reálný čas společně s proměnnou *teppokojmin*, která udává teplotu, na kterou kotel topí, proměnnou *stavcerpadla* a *stavkotle*, které udávají, zda kotel a čerpadlo běží.

7.5 Ukládání hodnot na SD kartu

SD karta není nezbytnou součástí regulace. Slouží pro mou kontrolu a pozorování teplot. Na SD kartu se zapisuje datum, reálný čas, teplota a vlhkost regulovaného pokoje (čidlo DHT22), teploty vstupu, výstupu ohřívání vody z kotle a venkovní teplota (čidla DS18B20) a proměnné *teppokojmin*, *stavcerpadla*, *stavkotle*, popsané v předešlé kapitole. V neposlední řadě se na SD kartu zapisuje konkrétní chybová hláška při případné chybě programu.

SD karta tedy neslouží k ukládání konstant nebo proměnných programu, se kterými se bude později pracovat. K tomuto účelu slouží vnitřní paměť mikrokontroleru.

8 ZAPOJENÍ

V této kapitole popisuji a ukazuji přesné zapojení celého Arduina a všech modulů včetně několika netradičních vylepšení, které jsem do projektu přidal.

8.1 Zdroj

Celkové zapojení tohoto projektu má poměrně netradičně řešený elektrický zdroj. A to tak, že jsou zde zdroje hned dva.

Prvním zdrojem je 9 V nabíječka, která vede do napájecího konektoru na desce Arduina. Avšak toto napájení není využito pro relé, ovládající spouštění a vypínání vytápění ohřevné vody a čerpadla zajišťující oběh ohřevné vody. Druhý zdroj je 5 V nabíječka, která je připojena přímo na ovládací relé. Tato nabíječka poskytuje stejných 5 V jako zdroj Arduina, jelikož je hned za napájecím konektorem v Arduinu stabilizátor napětí 7805, který napětí sníží z 9 V na 5 V.

Toto dvojitě řešení zdroje jsem uskutečnil z důvodu velkého zahřívání stabilizátoru napětí v desce Arduina a proto jsem část zátěže přesunul na další zdroj. Důvodem proč toto řešení vůbec může fungovat je to, že jsou všechny prvky připojeny na stejnou hodnotu země – zem Arduina. Pokud bych použil i zem z 5 V nabíječky, pak by datový kabel neměl správnou referenční hodnotu napětí a pravděpodobně by nefungoval tak, jak má.

Toto vylepšení značně pomohlo se zahříváním zabudovaného zdroje, ovšem já jsem ho ještě vylepšil o přídatný pasivní chladič v podobě

měděného plíšku připájeného přímo na chladič stabilizátoru napětí 7805 a mnohonásobně tím zvětšil jeho plochu.

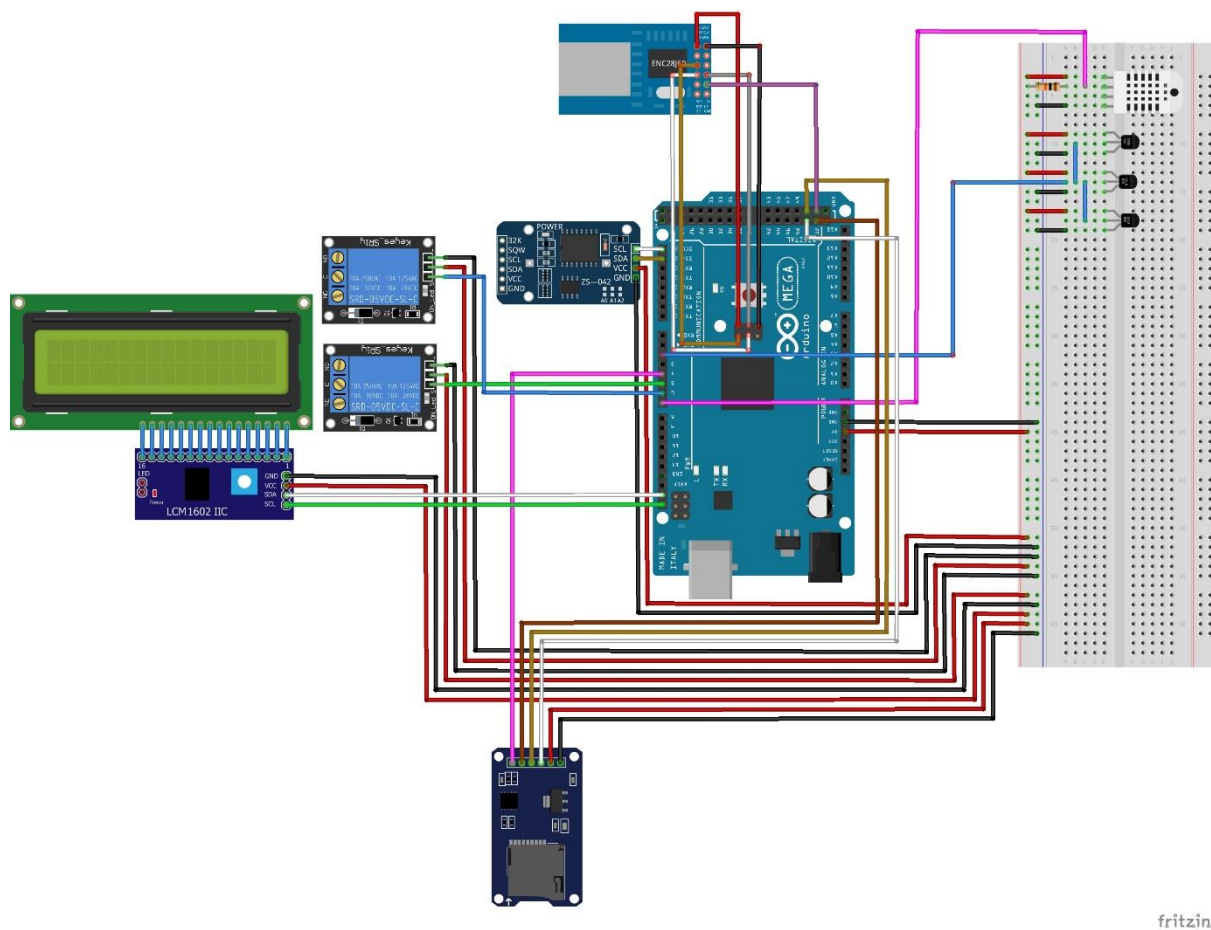
Pomocí těchto dvou vylepšení jsem byl schopen na omak znatelně snížit teplotu stabilizátoru. Bez těchto vylepšení nebylo možné na zdroji udržet prst více jak vteřinu. Po přidání je zdroj na omak pouze teplý.

Další řešený problém se zdrojem byl nedostatek výstupů 5 V a GND na desce Arduina. To jsem řešil připájením kabelu ze zdroje Arduina k jednostrannému plošnému spoji – cuprextitu. A ke cuprextitu jsem připájel kabely, které měly být připojeny na zdroj. Totéž jsem udělal pro GND, čili zem. Stejného výsledku bych mohl dosáhnout připojením zdroje do nepájivého pole, avšak nepájivé pole, které jsem měl k dispozici bylo příliš malé a již zabrané ostatními prvky projektu, a navíc je mnohem snazší omylem vytáhnout kabel z nepájivého pole než odtrhnout připájený kabel. Proto jsem zvolil více permanentní řešení.

8.2 Schéma

Schéma celého propojení Arduina s moduly a čidly jsem tvořil v programu Fritzing. Červené kabely jsou kabely zdroje a černé kabely jsou země. Ostatní barvy jsem vybíral náhodně.

V tomto schématu jsou teplotní čidla znázorněna na nepájivém poli, což ale není reálné zapojení, neboť v reálném zapojení jsou vedeny dráty o několik metrů dále do jiných místností.



fritzing

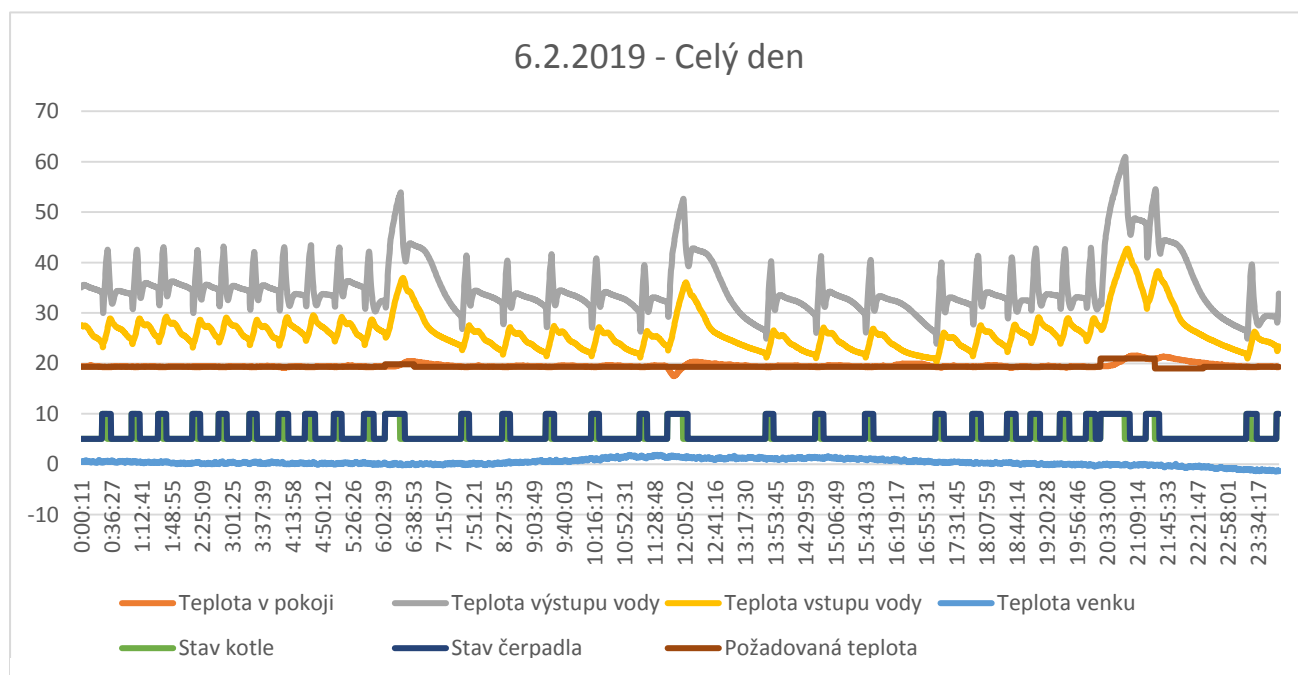
Obrázek 11 - Schéma zapojení

9 VÝSTUP PROGRAMU

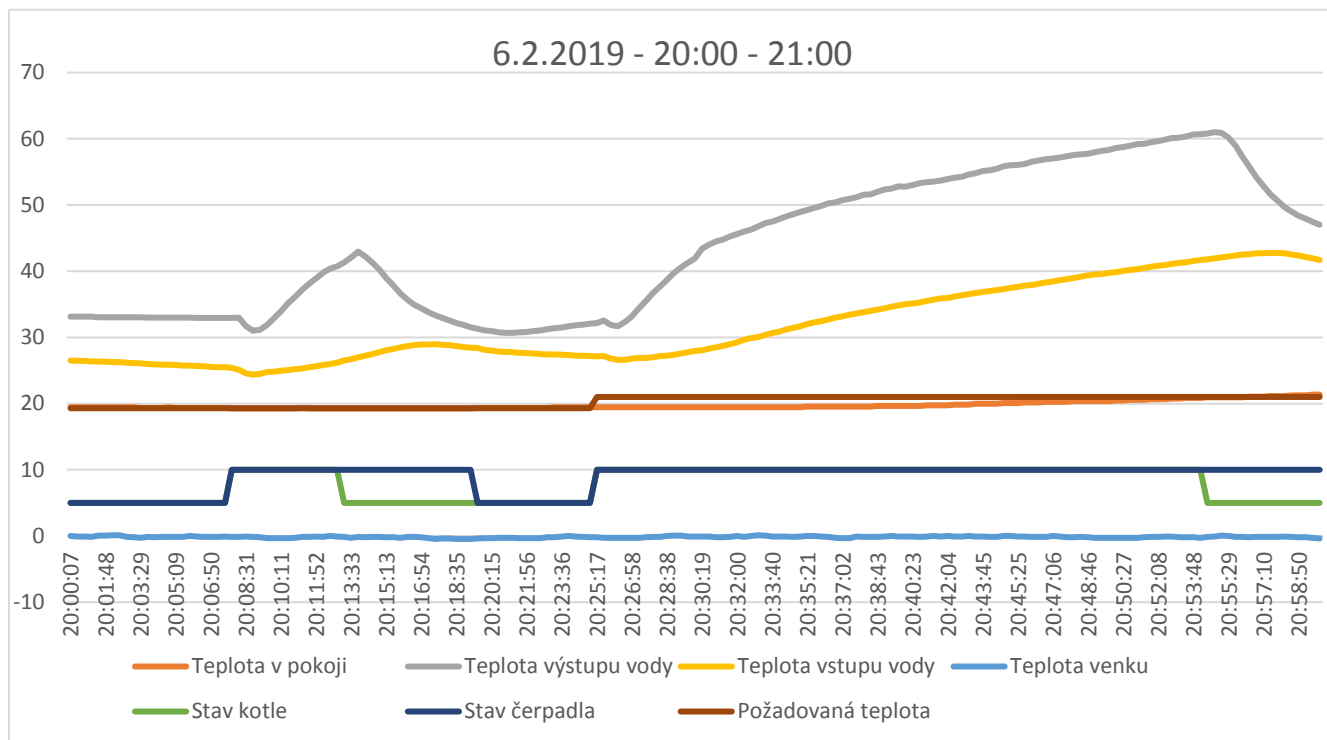
Jak jsem psal v kapitole 7.5, všechny hodnoty čidel program zapisuje každých 20 sekund na SD kartu do textového dokumentu. A díky tomu jsem schopen vytvořit následující grafy.

V rámci zpřehlednění grafu, jsem odebral hodnotu vlhkosti v regulovaném pokoji a nastavil hodnotu, když je čerpadlo, respektive hořák kotle zapnutý, na 10 a vypnutý na 5.

Jak lze vypočítat z prvního grafu, od začátku dne do 6:05 je požadovaná teplota v pokoji 19,3 °C. Následuje 35 minut topení na 19,8 °C, poté až do 20:25 na 19,3 °C. Nicméně okolo 11:45 je patrný výkyv teploty v pokoji a následný překmit, nad požadovanou teplotu. To je způsobené větráním a následuje okamžitá reakce systému na změnu teploty. Večer je ještě patrné zvýšení požadované teploty společně s lehce opožděnou reálnou teplotou.



Obrázek 12 - Graf teplot 6.2.2019



Obrázek 13 - Graf teplot 6.2.2019 - 20:00 - 21:00

10 ZÁVĚR

Plýtvání přírodními zdroji je v dnešní době velmi řešené téma. Proto je důležité, aby se každý snažil omezit plýtvání těmito neobnovitelnými zdroji. Jedním z nich je i zemní plyn, který pohání většinu vytápěcích kotlů.

A právě proto jsem vyvinul specifické řešení pro konkrétní rodinný dům, které topí podle přednastavených časů na předem danou teplotu v řízeném pokoji. Dané řešení poskytuje i možnost vytvářet grafy teplot a kompenzuje dopravní zpoždění topného systému.

V teoretické části jsem se zabýval prostředky automatizace projektů o různých velikostech, od domácích projektů až po prostředky využívané v průmyslu. Součástí teoretické části je i popis všech hardwarových součástí projektu včetně kotle a popis programů, ve kterých celý projekt vznikal.

V praktické části jsem rozebral k čemu je každá komponenta použita a proč je důležitá pro projekt, případně zda by mohlo být použito i jiné řešení. Dále jsem se v praktické části zabýval specifickým zapojením hardwaru a v neposlední řadě popsáním samotného programu.

Zpracováním tohoto projektu tedy považuji cíl mé bakalářské práce za splněný.

Vypracováváním mé bakalářské práce jsem získal mnoho nových znalostí v této problematice a chtěl bych se tomu oboru i nadále věnovat.

Seznam použité literatury

- [1] MACH, Marek. *BP - Automatizace pomocí Raspberry Pi* [online]. B.m., 2017 [vid. 2019-04-24]. ČVUT. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/70723/F2-BP-2017-Mach-Marek-Bakalarska-prace-rev-1306.pdf?sequence=1&isAllowed=y>
- [2] ZBYŠEK VODA. *Průvodce světem Arduina*. nedatováno.
- [3] Raspberry Pi. *Root.cz* [online]. 24. duben 2019 [vid. 2019-04-24]. Dostupné z: <https://www.root.cz/n/raspberry-pi/>
- [4] PAVEL BENEŠ. *Automatizace a automatizační technika 2*. Praha: Computer Press, nedatováno.
- [5] VIADRUS G 27 ECO. B.m.: ŽDB a.s.
- [6] *PWM signál* [online]. [vid. 2019-04-02]. Dostupné z: <https://domat-int.com/produkty/online-navody/pwm-signal>
- [7] *UART* [online]. 23. září 2011 [vid. 2019-04-04]. Dostupné z: <https://uart.cz/139/arduino-a-seriova-komunikace/>
- [8] *SPI* [online]. [vid. 2019-04-04]. Dostupné z: <https://www.arduino.cc/en/Reference/SPI>
- [9] *Arduino MEGA Datasheet* [online]. [vid. 2019-04-02]. Dostupné z: <https://www.robotshop.com/media/files/PDF/ArduinoMega2560Datasheet.pdf>
- [10] FÜHRER, Ondřej. *AVR* [online]. B.m., nedatováno [vid. 2019-04-04]. Vysoká škola Báňská. Dostupné z: [http://wh.cs.vsb.cz/mil051/images/c/ca/PAP_Architektura_procesoru_AVR_ATMega_\(Ond%C5%99ej_F%C3%BChrer\).pdf](http://wh.cs.vsb.cz/mil051/images/c/ca/PAP_Architektura_procesoru_AVR_ATMega_(Ond%C5%99ej_F%C3%BChrer).pdf)

- [11] *Paměti* [online]. 4. prosinec 2018 [vid. 2019-04-05]. Dostupné z: <https://cs.wikipedia.org/wiki/AVR>
- [12] *RTC hodiny reálného času* [online]. 2017 [vid. 2019-04-05]. Dostupné z: <https://arduino-shop.cz/docs/produkty/0/74/1459971336.pdf>
- [13] *Ethernet LAN modul* [online]. 2016 [vid. 2019-04-06]. Dostupné z: <https://arduino-shop.cz/docs/produkty/0/139/1500636003.pdf>
- [14] *Ethernet LAN modul - obrázek* [online]. [vid. 2019-04-06]. Dostupné z: <https://www.santy.cz/moduly-c22/ethernet-arduino-mega-uno-2560-1280-328-hr911105a-online-sd-enc28j60-mini-i146/>
- [15] *Čtečka SD karet* [online]. 2016 [vid. 2019-04-06]. Dostupné z: <http://navody.arduino-shop.cz/docs/texty/0/64/1427787012.pdf>
- [16] *Čtečka SD karet - obrázek* [online]. [vid. 2019-04-06]. Dostupné z: <https://www.gme.cz/modul-ctecky-micro-sd-karet>
- [17] *Relé modul* [online]. [vid. 2019-04-08]. Dostupné z: <https://arduino-shop.cz/arduino/2190-arduino-rele-4-kanaly-5v-s-optoclenem.html>
- [18] *LCD Displej* [online]. [vid. 2019-04-09]. Dostupné z: <https://navody.arduino-shop.cz/zaciname-s-arduinem/lcd-displej.html>
- [19] *Display* [online]. [vid. 2019-04-08]. Dostupné z: https://arduino-shop.cz/arduino/1421-arduino-display-zlutý-20x4-znaku-hd44780.html?gclid=EA1aIQobChMIpOX89bq74QIVUOJ3Ch0p8A-fEAQYBSABEGIQyPD_BwE
- [20] R.O, CzechProject spol s. Nepájivé kontaktní pole. *arduino-shop.cz* [online]. [vid. 2019-04-09]. Dostupné z: <https://arduino-shop.cz/arduino/1425-nepajive-kontaktni-pole-zy-170-r.html>

- [21] *DHT22* [online]. [vid. 2019-04-10]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-dht11.html>
- [22] R.O. DHT22 - obrázek. *arduino-shop.cz* [online]. [vid. 2019-04-10]. Dostupné z: <https://arduino-shop.cz/arduino/1188-arduino-dht22-teplomer-a-vlhkomer-digitalni.html>
- [23] *DS18B20* [online]. 28. červen 2016 [vid. 2019-04-17]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-ds18b20.html>
- [24] DS18B20 - obrázek. *eBay* [online]. [vid. 2019-04-17]. Dostupné z: https://www.ebay.com/itm/5x-DS18B20-DALLAS-18B20-TO-92-1-Wire-Digital-Temperature-Sensor-IC-NEW-z3-/252023599865?_ul=CZ
- [25] *Fritzing* [online]. [vid. 2019-04-17]. Dostupné z: <http://fritzing.org/>
- [26] *Arduino IDE* [online]. [vid. 2019-04-17]. Dostupné z: <https://www.arduino.cc/en/main/software>
- [27] *Arduino Eclipse IDE* [online]. [vid. 2019-04-23]. Dostupné z: <http://www.baeyens.it/eclipse/>