

# Hlídání úniku vody

Bakalářská práce



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra elektrotechnologie

Výpracoval: Vácha Ondřej  
Vedoucí práce: Ing. Karel Künzel, CSc.



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vácha** Jméno: **Ondřej** Osobní číslo: **483812**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra elektrotechnologie**  
Studijní program: **Elektrotechnika, energetika a management**  
Specializace: **Aplikovaná elektrotechnika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Hlídní úniku vody**

Název bakalářské práce anglicky:

**Water leakage monitoring**

Pokyny pro vypracování:

1. Aktualizujte rešerši systémů pro hlídání úniku vody z domácího rozvodu
2. Navrhněte zapojení a provedení instalace takového zařízení
3. Navrhněte zkušební zapojení vybraných komponent pro ověření funkčnosti
4. Ověřte a zdokumentujte funkčnost
5. Zpracujte dokumentaci zařízení
6. Porovnejte výsledné řešení s komerčně nabízenými detektory

Seznam doporučené literatury:

Kadlec K.: Snímače průtoku - principy, vlastnosti a použití, Automa 2006  
Vácha: Bakalářský projekt, ČVUT FEL 2020  
Firemní literatura Honeywell, Grohe a další

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Karel Künzel, CSc., katedra elektrotechnologie FEL**

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

Datum zadání bakalářské práce: **28.01.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Karel Künzel, CSc.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
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.

\_\_\_\_\_ Datum převzetí zadání

\_\_\_\_\_ Podpis studenta



## Čestné prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne: .....

Podpis: .....

## **Poděkování**

Rád bych poděkoval vedoucímu mé práce Ing. Karlu Künzelovi ,CSc., za připomínky a konstruktivní kritiku, které mě vedly vpřed.

## **Abstrakt**

Tato bakalářská práce se věnuje návrhu a výrobě zařízení pro hlídání úniku vody v domácím rozvodu. Obsahuje návrh plošného spoje, napsání programu pro centrální jednotku a pro WiFi modul. Dále obsahuje uvedení zařízení do provozu a jeho odzkoušení. Cílem této práce bylo vytvořit zařízení, které řeší nedostatky aktuálně na trhu dostupných řešení. Všechny stanovené cíle byly dosaženy.

**Klíčová slova:** Arduino, impulsní vodoměr, kulový ventil, PCB, LCD, WiFi modul, 3D modelování

## **Abstrakt**

This bachelor thesis focuses on creating complete system for water leakage detection in households. This thesis contains design of PCB (printed circuit board), creation of program for central unit and WiFi module, installing system inside switchboard and testing. Main goal of this thesis was creation of a system, which solves some shortcomings of commercially available solutions. All of declared goals were successfully achieved.

**Keywords:** Arduino, impuls water meter, ball valve, PCB, LCD, WiFi module, 3D modeling

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
1.1	Požadavky a cíle . . . . .	9
<b>2</b>	<b>Koncepce zařízení</b>	<b>9</b>
2.1	Návrh blokového schématu . . . . .	10
<b>3</b>	<b>Výběr komponent pro realizaci</b>	<b>11</b>
3.1	Centrální jednotka . . . . .	11
3.2	Uzávěrový ventil . . . . .	12
3.3	Impulsní vodoměr . . . . .	15
3.4	Periférie . . . . .	15
3.4.1	WiFi Modul ESP01S . . . . .	15
3.4.2	LCD displej . . . . .	17
3.4.3	GSM modul, EZS a další . . . . .	18
<b>4</b>	<b>Návrh zařízení</b>	<b>18</b>
4.1	Návrh a výroba desek plošného spoje . . . . .	18
4.1.1	Návrh desky plošného spoje pro uživatelské rozhraní . . . . .	19
4.1.2	Návrh hlavní desky plošných spojů . . . . .	22
4.1.3	Výroba desek plošných spojů . . . . .	24
4.2	Návrh programového vybavení . . . . .	26
4.2.1	Programové vybavení pro Arduino Nano . . . . .	26
4.2.2	Programové vybavení pro WiFi modul ESP01S . . . . .	28
<b>5</b>	<b>Realizace a instalace</b>	<b>28</b>
5.1	Instalace impulsního vodoměru a kulového ventilu . . . . .	28
5.2	Čelní kryt krabičky . . . . .	29
5.3	Instalace zařízení do rozvaděče . . . . .	31
<b>6</b>	<b>Testování programového vybavení</b>	<b>31</b>
6.1	První fáze . . . . .	31
6.1.1	První fáze část A . . . . .	32
6.1.2	První fáze část B . . . . .	34
6.1.3	První fáze část C . . . . .	34
6.2	Druhá fáze . . . . .	35
6.2.1	Otestování v provozu . . . . .	35
6.2.2	Dokumentace kódového označení poruch . . . . .	37
<b>7</b>	<b>Závěr</b>	<b>37</b>
7.1	Návrh pro vylepšení . . . . .	38
<b>8</b>	<b>Použitá zdroje</b>	<b>39</b>



# 1 Úvod

Tato bakalářská práce navazuje na bakalářský projekt [1], který zpracovává řešerši k této práci. V této práci se zabývám návrhem programového vybavení, výrobou desek plošných spojů, sestavením a uvedením do provozu zařízení pro hlídání úniku vody pro domovní rozvod vody.

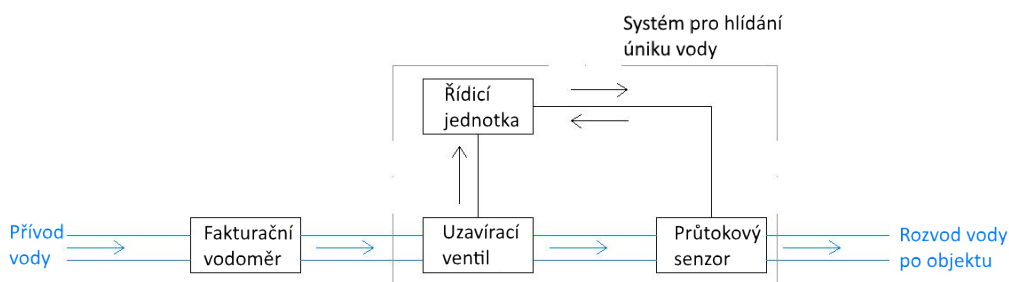
Zařízení pro hlídání úniku vody slouží k detekci a zabránění škod vzniklých únikem vody v chráněném objektu a informování majitele o vzniklé havárii. Havárií může být například: protékající toaleta, nedovřený kohoutek vody, havárie na vodovodním potrubí nebo neočekávaný velký odběr vody [1].

## 1.1 Požadavky a cíle

Požadavky a cíle této práce byly vytvořeny ze závěru bakalářského projektu [1]. Hlavními požadavky jsou návrh a provedení zařízení, které bude řešit nedostatky aktuálně na trhu dostupných řešení. Celý systém bude napájen z 12 V DC, s co nejmenší spotřebou. S možností na dálku ovládat uzavěrový ventil a celý systém mít zálohovaný Pb baterií s dlouhou zálohou v případě výpadku napájení. Při návrhu budu také klást důraz na spolehlivost, funkčnost a cenu.

## 2 Koncepce zařízení

Zařízení pro hlídání úniku vody se skládá z těchto hlavních částí: senzoru průtoku vody s impulsním výstupem, řídicí jednotky a uzavíracího ventilu. Řídicí jednotku lze také propojit s dalšími periferiemi jako zabezpečovacím systémem (dále jen EZS), GSM a Wi-Fi modulem nebo s další signalizací stavů.



Obrázek č. 1: Schématické zapojení jednotlivých modulů zařízení pro hlídání úniku vody na přívodu vody, zdroj: [1]

## 2.1 Návrh blokového schématu

Pro návrh blokového schématu potřebuji znát jaké vstupy a výstupy v zařízení pro hlídání úniku vody použiji.

**Vstupy:** Vstupů bude celkem 13. 4 pro trimry, kde trimry budou v zařízení plnit funkci pro nastavování hlídaných parametrů.

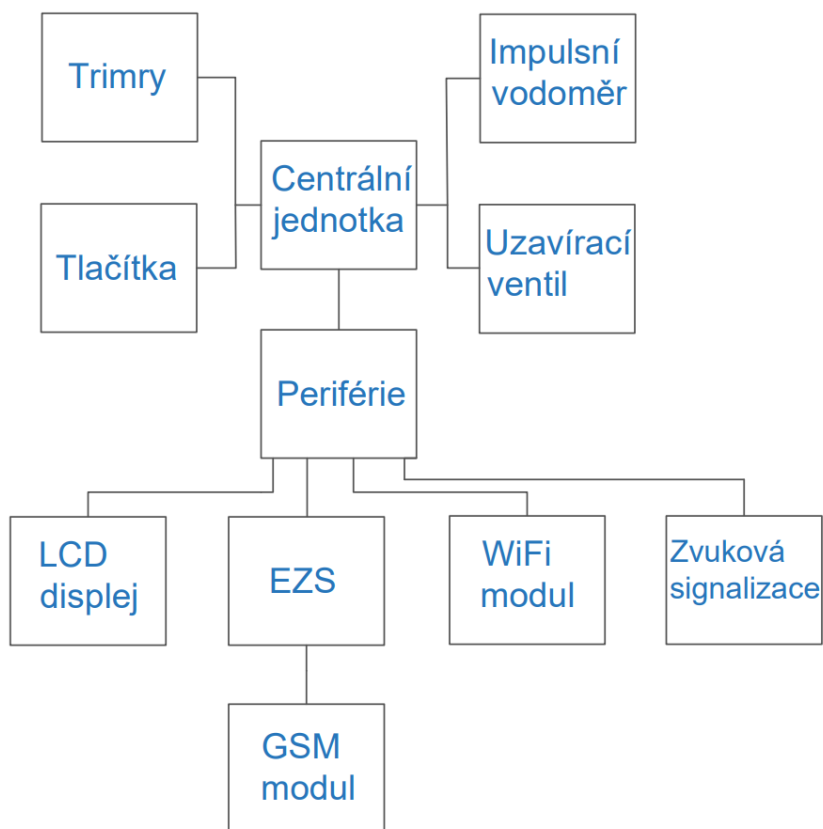
Další 4 vstupy jsou pro tlačítka - reset centrální jednotky, externí zavření ventilu, tlačítko pro rozsvícení displeje a režim bez kontroly odběru vody.

Externí zavření ventilu přes tlačítko bude navíc zapojeno v logickém zapojení OR s externím uzavřením vody z EZS.

Dalšími vstupy jsou: stav ventilu nastavený v online rozhraní, koncové stavy ventilu, vstup od impulsního vodoměru a vstup od EZS.

**Výstupy:** Výstupů je dohromady 6. Dva jsou pro LCD displej a jsou to kanály SDA a SCL. Další dva jsou pro ovládání pohonu kulového ventilu. Ventil bude ovládán přes tranzistor, který sepne GND cívice relátka. Tranzistor bude ovládán z výstupního pinu centrální jednotky.

Poslední dva výstupy budou spínány přes tranzistory NPN a jsou signalizačního charakteru. Je to zvuková signalizace a výstup pro GSM modul.



Obrázek č. 2: Blokové schéma zařízení pro hlídání úniku vody

## 3 Výběr komponent pro realizaci

### 3.1 Centrální jednotka

Při výběru centrální jednotky jsem se rozhodoval mezi PLC, Arduino, ESP32, Nucleo a PyCom. Protože pro mě hlavním kritériem bylo napájení celého systému 12 V DC a nízká pořizovací cena. Rozhodl jsem se pro Arduino Nano. Také mám s Arduinem nejvíce zkušeností. Bližší informace o důvodu výběru centrální jednotky je možné nalézt v dokumentu [1].

Arduino Nano je vybaveno dvěma asynchronními přerušeními, konkrétně na digitálních pinech 2 a 3. Protože tato přerušení jsou na bázi hardwaru, tak je možné na nich sledovat změnu pinu na GND nebo +5 V DC.

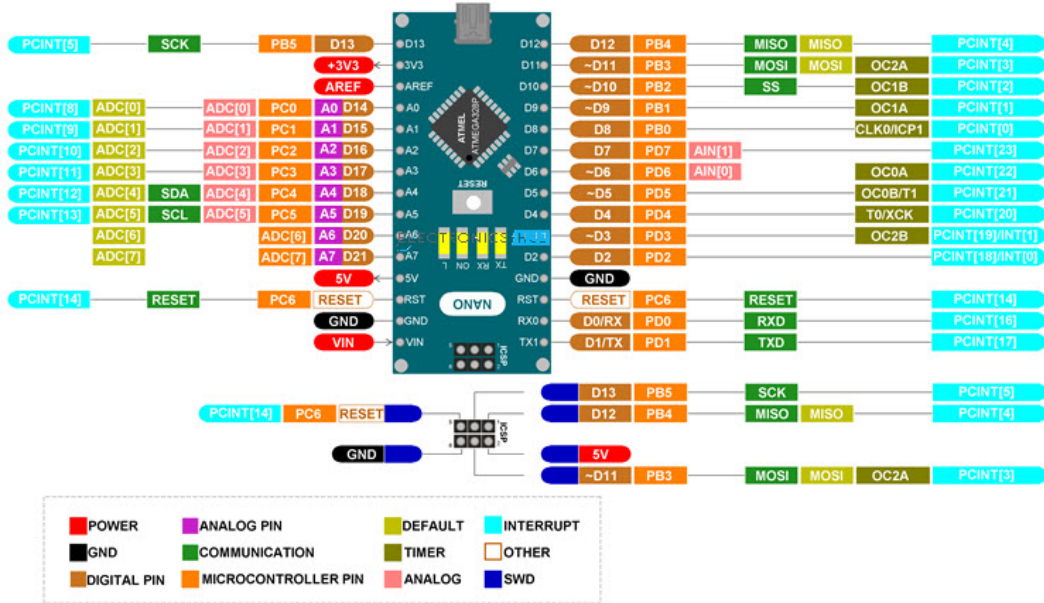
Pro potřeby většího počtu přerušovacích vstupů je možné využít registr PCICR (pin change interrupt control register). První 3 bity tohoto registru se týkají třech portů Arduina. Tyto porty jsou označeny B, C a D. Odpovídají pinům D8 až D12 (označení B), respektive D0 až D7 (označení C), respektive A0 až A5 (označení D). Při využívání registru PCICR je možné sledovat pouze náběžnou nebo sestupnou hranu. Tento registr je využit v kapitole č. 4.2 při návrhu programového vybavení.

Na analogových pinech A0-A5 má Arduino zapojen desetibitový analogově-digitální převodník. Na analogových pinech A4 (SDA) a A5 (SCL) umožňuje komunikaci na bázi protokolu I2C, například s LCD displejem, který je vybaven I2C převodníkem. Dále je mikrokontrolér vybaven vnitřními hodinami, které lze použít například pro měření doby, jak dlouho je mikrokontrolér zapnutý.

Program na Arduino se píše v jazyce Wiring, který vznikl na bázi jazyka C/C++. Jeho hlavními dvěma funkcemi jsou *setup* a *loop*. Funkce *setup* je zavolána jako první, pouze jednou, a to po startu Arduina. V této funkci se nastavují piny, konstanty a další proměnné. Funkce *loop* je nekonečná smyčka do které se píše hlavní tělo programu.

Mikroprocesor	ATmega328P
Provozní napětí (logická úroveň)	5 V
Počet digitálních I/O pinů	22 pinů
Počet analogových vstupů	8 pinů (označení A0 až A7)
Maximální proudové zatížení na 1 pin	40 mA
Celkové maximální zatížení všech pinů	200 mA
Rychlost hodin	16 MHz

Tabulka č. 1: Technické informace o Arduino Nano, zdroj: [2]



Obrázek č. 3: Popis pinů Arduina Nano, zdroj: [3]

### 3.2 Uzávěrový ventil

Při výběru uzávěrového ventilu jsem měl tato kritéria: pořizovací cena, nízká spotřeba, napájení 12 V DC a rychlé otevírání/zavírání ventilu. Z výsledků rešerše z dokumentu [1] jsem zjistil, že na výběr mám tedy pouze dvě možnosti. Kulový a membránový ventil. V tabulce níže je vidět porovnání základních vlastností obou ventilů. Kulový ventil má oproti membránovému velkou nevýhodou, dlouho se zavírá/otevírá. Ale jinak je jeho provoz i pořizovací cena mnohem nižší, a proto jsem se rozhodl zvolit ten. Konkrétní zvolený typ je dostupný z [4]. U konkurenčních řešení se používají oba typy. Například firma Vodostop používá kulový a firma Sonytron využívá membránový ventil.

Typ ventilu	Pořizovací cena	Spotřeba	Rychlost otevření/zavření	Spínací cykly
Kulový	cca 2 000 Kč	-	jednotky s	20k
Membránový	cca 3 500 Kč	7 W stálý odběr	stovky ms	100k

Tabulka č. 2: Porovnání kulového a membránového ventilu s průměrnými hodnotami několika použitelných konkrétních typů

Konkrétní zvolený uzávěrový ventil je v zapojení označovaném CR2. Zapojení CR2 znamená bez vyvedených koncových spínačů. Při hledání kulových ventilů s elektropohonem mimo ČR jsem narazil na mimo evropský produkt, u kterého bylo schéma zapojení s vyvedenými koncovými spínači s označením CR5. Nyní popíši rozdíl v ovládání a výhody těchto dvou typů zapojení.

Princip ovládání ventilu se zapojením CR2 využívá tři vodiče OPEN, CLOSE a COMMON. Vodiče OPEN a CLOSE jsou jeden pro otevírání a druhý pro zavírání ventilu. Na vodič COMMON je přivedena GND nebo VCC, toto se liší podle výrobce.

Pokud na vodič COMMON je přivedeno VCC, tak pro zavírání/otevírání je třeba přivést na příslušný vodič GND. Zem je přivedena vždy jen na jeden vodič, ten druhý zůstává odpojen. Dojde k uzavření obvodu a ventil se zavírá/otevívá tak dlouho dokud není sepnut koncový spínač, který obvod rozpojí. Tento koncový spínač pak blokuje i další otáčení ventilu ve stejném směru. Pokud je na vodič COMMON přivedena GND, tak se na vodiče pro otevírání/zavírání musí přivést VCC.

Princip ovládání ventilu s vyvedenými koncovými spínači je založen na tom, že mám 6 vodičů. Dva jsou společné a je na nich GND nebo VCC. V mém případě na ně připojím VCC, protože výstup z koncového spínače půjde do anody optočlenu. Tyto společné vodiče jsou připojeny na svorky C koncového spínače pohonu. Další dva vodiče jsou připojené na svorky NO koncových spínačů a přivádí informaci o jejich sepnutí do centrální jednotky. Zbylé dva vodiče jsou pro napájení motoru a jsou označovány OPEN a CLOSE. V tabulce níže jsou ukázané jejich kombinace a jaký to má vliv na pohyb ventilu.

open	close	pohyb ventilu
HIGH	HIGH	zastavení pohybu
HIGH	LOW	otevírání
LOW	HIGH	zavírání
LOW	LOW	zastavení pohybu

Tabulka č. 3: Ovládání kulového ventilu s elektropohonem v zapojení CR5

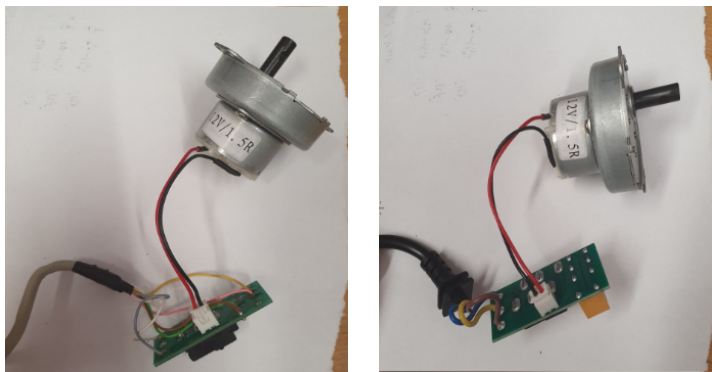
Zapojení ventilu CR2 má výhodu menšího počtu vodičů potřebných k ovládání ventilu, ale nevýhodou je, že nemám informaci o koncovém stavu ventilu. Mám na mysli specifickou skutečnost, kdy byl vydán pokyn k uzavření ventilu, ale ventil se neuzavřel. V programu to mohu snadno ošetřit tak, že pokud došlo k vydání pokynu a voda stále protéká, ventil není uzavřen. Tuto informaci si, ale centrální jednotka pamatuje pouze dokud nedojde k jejímu restartu. Pokud budu, ale chtít vědět v jaké pozici se ventil nachází po restartu centrální jednotky, bez koncového spínače nemám snadnou možnost, jak to zjistit.

Proto jsem se rozhodl modifikovat ventil zakoupený v ČR, který má zapojení bez koncových spínačů a koncové spínače vyvést ven. V případě, když bych kupoval uzávěrový ventil znovu, jistě bych vybral s vyvedenými koncovými spínači z důvodu přidané informace o poloze ventilu po restartu centrální jednotky.

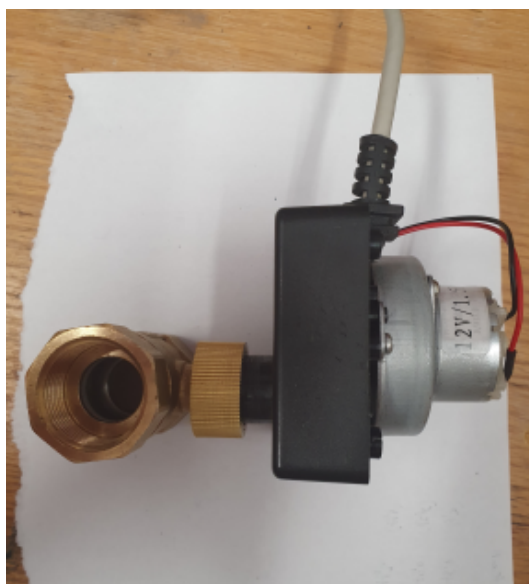
Postup modifikace ventilu s označením CR2 na CR5 je znázorněn na obrázcích níže. Celý proces byl navíc zjednodušen díky tomu, že koncové spínače jsou i v zapojení CR2 použity, jen nejsou vyvedeny ven. Pro modifikaci mi stačí tyto koncové spínače vyvést ven a motor napájet přímo, ne z přepínacího relé jako je tomu u zapojení CR2. Potom stačí přidat kód do programu, který ošetří, kdy má dojít k zastavení pohybu ventilu, to nastane vždy, když bude sepnut koncový spínač.

Nejprve jsem odletoval všechny přivedené vodiče na desku plošného spoje motoru a přepínací relé. Potom jsem naletoval vodiče nové na VCC a GND konektoru motoru. Dále jsem jen naletoval vodiče pro koncové spínače.

Nyní když chci, aby se ventil začal zavírat/otevírat, přivedu pokyn dle tabulky výše a pouze v programu čekám na to, než dojde k sepnutí koncového spínače. V momentu, kdy je koncový spínač sepnutý, ventil dosáhl nové koncové pozice a v programu to vyhodnotím jako zastavení pohybu ventilu.



Obrázek č. 4: Vlevo - upravený kulový ventil, vpravo - kulový ventil před upravením



Obrázek č. 5: Složený upravený uzavírací ventil s kulovým uzávěrem bez horního krytu, společně s přípojkou na vodovodní přívod

### 3.3 Impulsní vodoměr

Aktuálně nabízené impulsní vodoměry na trhu, které jsou použitelné pro mou aplikaci z cenového hlediska a přesnosti měření jsou pouze 2 typy. Suchoběžné a mokroběžné turbínové průtokoměry. Obecně platí, že mokroběžné z jejich principu funkce dosahují přesnějšího měření průtoku vody, mají menší rozběhový průtok, ale o to mají vyšší cenu.

Zvolil jsem tedy suchoběžný od firmy Enbra, který se svými parametry přibližuje k mokroběžnému, dostupný je např. zde: [5]. U zvoleného impulsního vodoměru je konstanta převodu proteklého množství vody na 1 impuls 0.3 l, to bude důležité v kapitole č. 4.2 při návrhu programu pro centrální jednotku.

Zapojení impulsního výstupu z vodoměru je provedeno dvěma vodiči. Jeden přivádí 5 V DC na reed kontakt. Druhý vodič z reed kontaktu přivádí informaci o průtoku zpět na vstup centrální jednotky.

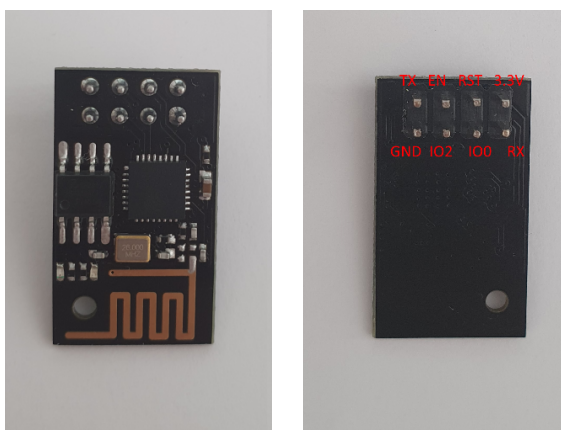
Princip funkce a porovnání ostatních impulsních vodoměrů je možné nalézt v dokumentu [1].

### 3.4 Periférie

Periférie, které používám: WiFi modul, zvuková signalizace, LCD displej 20x4 zelenožlutý, elektronická zabezpečovací signalizace od firmy Jablotron (EZS), která bude zároveň sloužit i jako GSM modul.

#### 3.4.1 WiFi Modul ESP01S

WiFi modul ESP01S je 32bitový mikrokontrolér na bázi ESP8266. Modul ESP01S se programuje stejně jako Arduino jazykem Wiring. Využiji ho pro připojení k internetu a budu ho ovládat přes iot-cloud. Iot-cloudem se rozumí internetová služba, která propojuje zařízení s přístupem na internet a umožňuje jim vyměňovat si mezi sebou data.



Obrázek č. 6: Přední strana modulu ESP01S a zadní strana modulu ESP01S s popsányými piny s dostupnou dokumentací z [6].

Pro modul ESP01S jsou napsány knihovny ESP8266WiFi a MQTT, které umožňují připojení k síti WiFi a vytvoření MQTT objektu pro komunikaci s iot-cloudem. Protože tento mikrokontrolér není vybaven žádným typem USB, je třeba pro nahrání programu do paměti propojit počítač s modulem ESP01S přes převodník USB na TTL komunikaci a pin GPIO0 připojit na GND, aby modul ESP01S nabootoval v programovacím režimu.

V zařízení pro hlídání úniku vody je funkce modulu ESP01S předávat informaci o aktuálně žádaném stavu ventilu z internetového rozhraní centrální jednotce. Uvedu tedy dvě možnosti, jak zrealizovat propojení s centrální jednotkou a jejich výhody a nevýhody.

První možností je propojit pin GPIO0 s pinem na centrální jednotce přímo. Protože WiFi modul je v tomto zapojení jako výstup, není třeba používat převodník logické úrovně z 3.3 V na 5 V, protože budu spínat výstup na GND nebo 3.3 V a oboje se nachází v toleranci 5 V vstupu centrální jednotky. Logická úroveň pro napětí 3.3 V je brána jako logická 1 v Arduinu Nano.

Druhou možností je použít propojení mezi modulem ESP01S a centrální jednotkou pomocí sériové komunikace. Zde je třeba použít převodník napěťové úrovně. Výhodou je pak možnost z online rozhraní ovládat takový počet výstupů, kolik dovolí volné piny na centrální jednotce.

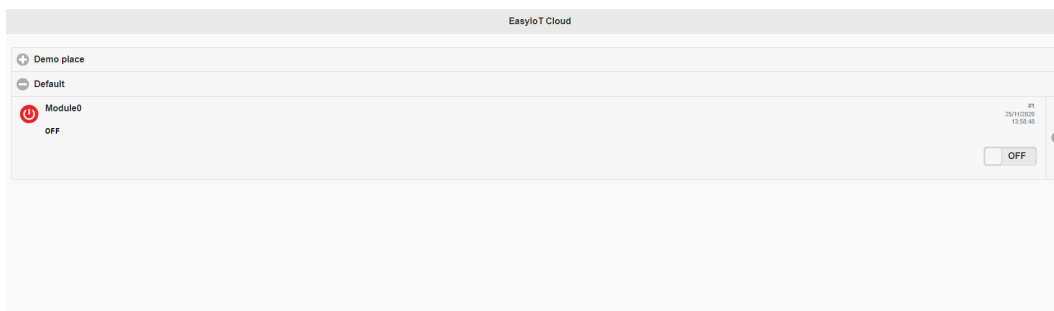
První způsob oproti druhému je výhodnější pro ovládání maximálně dvou výstupů, protože zabere méně nebo stejný počet pinů na centrální jednotce a využívá o jeden převodník napěťové úrovně méně. Protože z online rozhraní potřebuji ovládat pouze uzávěrový ventil, zvolím první možnost, tedy přímé propojení WiFi modulu s centrální jednotkou. K tomuto propojení využiji pin GPIO0 jako výstup z modulu ESP01S. Logickou 0 budu v programu interpretovat jako požadavek na zavření ventilu a logickou 1 pro otevření.

Pro online rozhraní použiji iot-cloud, který má volně dostupné vytvořené rozhraní pro komunikaci mezi klientem a serverem, zkráceně se tomuto rozhraní říká api. Iot-cloudů je na výběr spousta. Já zvolil iot-cloud s názvem iot-playground, protože měl volně dostupné api, bezplatné vytvoření a provoz účtu s prakticky neomezeným množstvím volání procedur mezi serverem a klientem.

Dostupný je zde: <https://easyiot-cloud.com/> [online 20.03.2021]. Do rozhraní iot-playground je možné přistupovat přes prohlížeč nebo přes mobilní aplikaci. Iot-playground má mobilní aplikaci v době psaní práce dostupnou pouze pro uživatele telefonů Android. Uživatelé iOS v době psaní práce mohou využívat pro ovládání pouze webové prostředí.

Návrh prostředí je jednoduchý. Nejprve přejdu na odkaz uvedený výše a vytvořím si účet. Po přihlášení do prostředí stačí přidat jeden modul, který bude simulovat tlačítko. Vytvořím ho přes Configure-> Modules-> Add module a v module type zvolím Digital output, zbytek nastavení nechám a uložím. Prostředí by ve finále mělo vypadat takto:





Obrázek č. 7: Prostředí v iot-playground s jedním modulem jako tlačítkem

WiFi modul ESP01S umožňuje práci v módech AP/STA/STA+AP. V tabulce níže jsou tyto pojmy vysvětleny. Pro mou aplikaci využiji mód STA, protože chci WiFi modul připojit do domácí sítě.

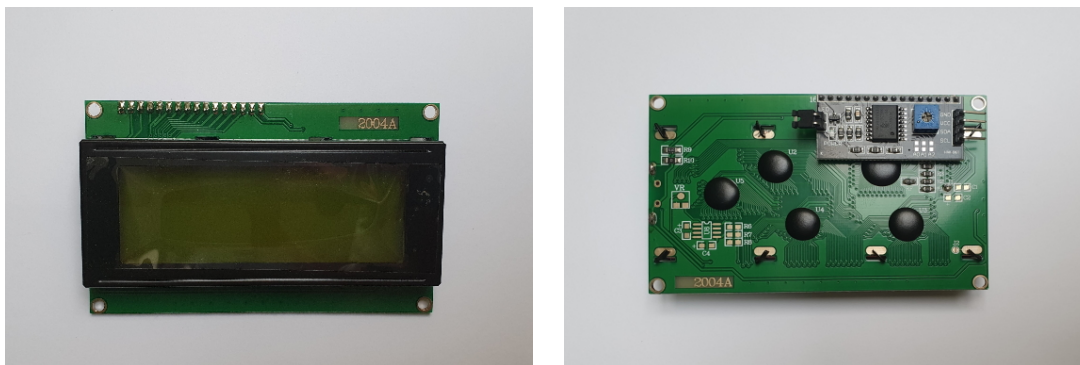
AP	Access point, neboli přístupový bod. V tomto režimu ESP01S vytváří WiFi, na kterou se mohou připojit.
STA	Station, v tomto režimu se modul ESP01S připojuje do sítě WiFi.
STA+AP	Kombinace módu AP a STA. Modul ESP01S je připojen do sítě, a navíc vytváří WiFi síť vlastní, na kterou je možné se připojit.

Tabulka č. 4: Pracovní módy WiFi modulu ESP01S

Nyní mám nastavené webové rozhraní a zbývá napsat program pro WiFi modul ESP01S, ten je možné nalézt v kapitole č. 4.2.

### 3.4.2 LCD displej

Pro zobrazování informací o stavu ventilu, odběru vody, typu poruchy a dalších používám 20x4 zelenožlutý LCD displej. Ovládám ho přes sériový synchronní přenos dat pomocí I2C protokolu, který funguje na bázi master-slave. Jeho výhodou oproti standardnímu ovládání LCD displeje je, že potřebuje pouze dva piny na centrální jednotce. Jeden pin, kde přenáším synchronní hodiny (SCL) a druhý na přenos dat a adresy periférie (SDA). Pro standardní ovládání LCD displeje je potřeba 16 pinů. Rozhodl jsem se mít trvale podsvícený displej na minimální jas místo nepodsvíceného displeje. Rozdíl v odběru proudu byl pouze 1 mA a čitelnost displeje se mnohonásobně zlepšila. Stiskem tlačítka je pak možné zapnout maximální podsvícení jen na omezenou dobu.



Obrázek č. 8: Přední a zadní strana LCD displeje. Na zadní straně je naletovaný převodník I2C komunikace (černá destička) pro LCD displeje

Ukázka složení komunikace na kanálu SDA. Uvažuji, že v defaultním stavu je kanál v logické 1. Pro zahájení komunikace přijde start bit. Start bit je vždy přechod z logické 1 do logické 0. Následuje adresa zařízení, se kterým si přeji komunikovat, tato adresa může být například v hexa formátu. Dále přichází READ/WRITE bit, kterým určuji, kdo bude číst a kdo bude zapisovat. Pokud budu zapisovat, tak pak přichází data, po nich acknowledge bit, kterým mi slave říká, že data přijal, a nakonec stop bit. Stop bit je přechod z logické 0 do logické 1.

### 3.4.3 GSM modul, EZS a další

V mé práci využiji EZS, která bude zároveň sloužit i jako GSM modul. Pro tento účel připravím jeden externí vstup a výstup. Vstup bude sloužit k uzavření vody, když dojde k zabezpečení domu. Výstup bude sloužit pro informování GSM modulu uvnitř EZS, že nastala porucha a má se uvědomit majitel.

## 4 Návrh zařízení

### 4.1 Návrh a výroba desek plošného spoje

Desku plošných spojů jsem navrhl v programu Fritzing. Je to open source program, který umožňuje exportovat desku plošného spoje jako pdf pro domácí výrobu. Návrh plošného spoje jsem rozdělil na dvě fáze.

V první fázi jsem navrhl plošný spoj pro uživatelské rozhraní s tlačítky, trimry a LCD displejem. V druhé fázi jsem navrhl plošný spoj pro hlavní desku se všemi výstupy/vstupy a ostatními komponenty.

Než začnu s návrhem desky, je třeba znát jaké konkrétní součástky použiji a jaké mají rozměry, to je obzvlášť důležité, protože celé hotové zařízení se umístí do krabičky do rozvaděče.

Je třeba připomenout, že místo součástek, které v programu Fritzing neměly správné parametry, nebo pokud jsem nenašel přesný model součástky, který použiji, byly tyto součástky nahrazeny svorkami typu male s rozestupem 2.54 mm. Navíc pro svorky, které mají vývod mimo krabičku v rozvaděči, je třeba udělat svorky dvojité s rozestupem 5 mm, protože ty budou vyvedeny přes Wago svorky ven.

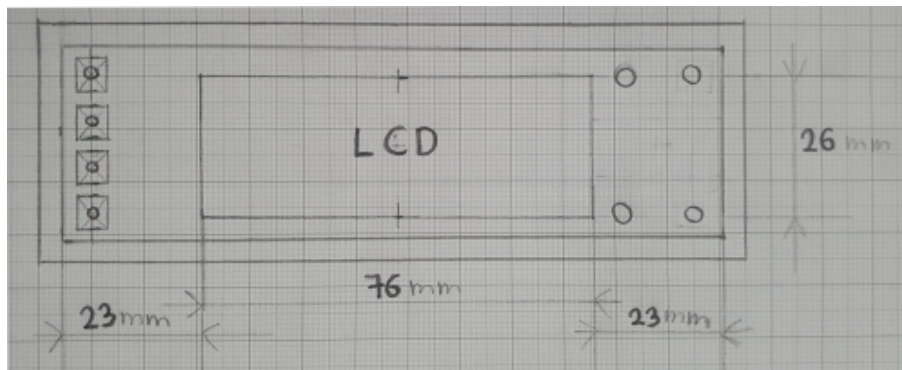
#### 4.1.1 Návrh desky plošného spoje pro uživatelské rozhraní

Návrh desky plošného spoje pro uživatelské rozhraní je ovlivněn volným místem na čelním krytu krabičky. Z důvodu velikosti displeje a potřeby místa na tlačítka s trimry a popisky zvolím 8 modulovou krabičku na DIN lištu.

Dále je třeba vybrat konkrétní typy tlačítek a trimrů, protože pokud nebudu znát jejich rozměry, nemohu správně rozměrově navrhnout desku plošného spoje. Konkrétní zvolený typ trimru je P3362 10 k $\Omega$  s rozměry 6x6 mm, pro tlačítka jsem zvolil typ ISC 1516-4 s rozměry 7x7 mm. Využívám tlačítka jak s aretací, tak bez aretace. Se znalostí rozměrů tlačítek a trimrů se mohu pustit do návrhu desky plošného spoje pro uživatelské rozhraní.

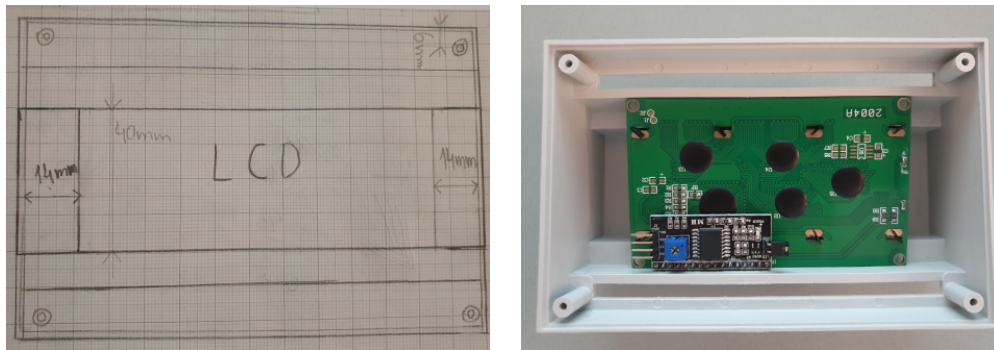
Rozměry vrchní strany 8 modulové DIN krabičky jsou 135x45 mm (bez okrajů) s okraji je efektivní velikost 125x36 mm.

Displej pro co největší úsporu místa umístím doprostřed a navíc ho "utopím" v otvoru krabičky. To znamená, že efektivní velikost displeje při pohledu shora bude 76x27 mm. Z toho vyplývá, že mi zbude na každé straně 23 mm široký úsek, kam se musí vejít trimr/tlačítko a popis. To při velikosti tlačítek 7x7 mm a trimru 6x6 mm nebude problém.



Obrázek č. 9: Skica - čelní pohled na krabičku, vlevo rozmístění trimrů, uprostřed LCD displej a vpravo rozmístění tlačítek

Ve skutečnosti, ale musím ještě uvažovat, že deska plošného spoje bude umístěna z vnitřní strany krabičky. Deska plošného spoje pro trimry a tlačítka tím bude mít rozměry do šířky zmenšené o přesahující rozměr displeje. Rozměry desky plošného spoje budou 14x40 mm. Proto nemohu mít tlačítka ve dvou sloupcích, ale budu je muset umístit na desku plošného spoje stejně jako mám naznačeny trimry na obrázku č. 9.

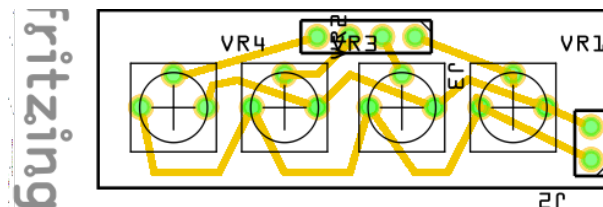


Obrázek č. 10: Vlevo - skica půdorysu krabičky s vyznačeným LCD displejem a místem pro desku plošného spoje pro trimry a tlačítka, vpravo skutečné umístění displeje v krabičce

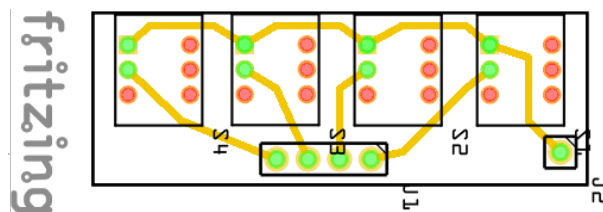
Při návrhu desky plošných spojů jsem si nejprve vytvořil schématické zapojení. Na propojení s hlavní deskou si připravím konektory typu male s roztečí 2.54 mm. Tyto dvě desky propojím vodiči s konektory typu female.

Po navrženém zapojení Fritzing automaticky v záložce "PCB návrh" zobrazuje fyzickou velikost součástek, a i jaký pin má být s jakým propojen, dle návrhu ze schématického zapojení. Fritzing umí také autorouting pro automatický návrh desky plošného spoje, který ale provádí vždy ve dvou vrstvách. Mám v plánu si desku plošných spojů vytvořit doma. Funkci autoroutingu tedy nemohu použít a desku si musím navrhnout sám.

Na obrázcích níže jsou vidět schématická zapojení desky plošných spojů, na které budou umístěny tlačítka, displej a trimry.



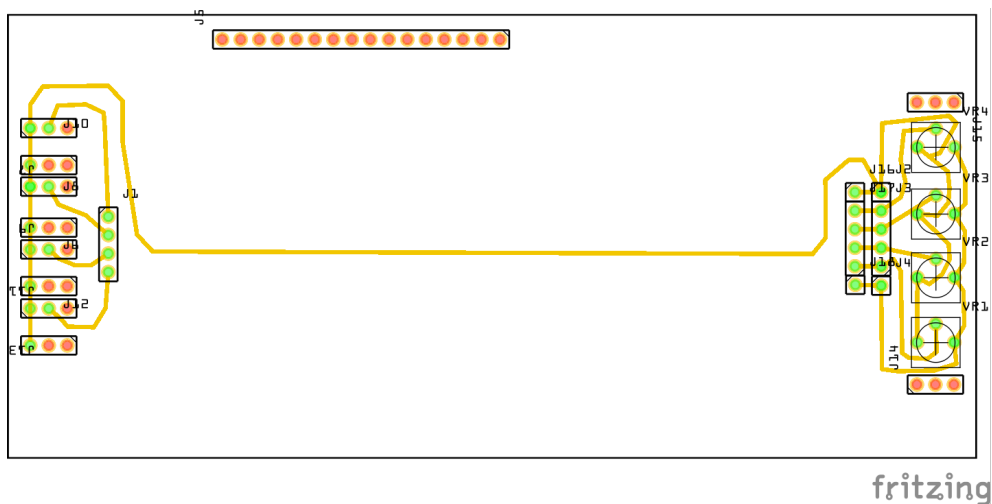
Obrázek č. 11: Deska plošného spoje s trimry



Obrázek č. 12: Deska plošného spoje s tlačítky

Tyto dvě desky plošných spojů jsem navrhl do jedné desky, která je vidět na obrázku níže. 16 pinový konektor, je zde umístěn pouze pro účely, výroby desky. Pomůže mi správně určit místo, kde mám vyříznout otvor pro I2C převodník a určit místa, kde budou udělány otvory pro uchycení desky a LCD displeje. Mám totiž změřeno, jak daleko od kraje musí být umístěn první otvor relativně vůči prvnímu pinu, tak aby LCD displej byl přesně uprostřed desky.

Na levé straně desky jsou zapojena tlačítka a na pravé jsou připraveny konektory pro propojení desky pro uživatelské rozhraní s hlavní deskou. Také jsem si na pravé straně připravil zapojení trimrů. Při výrobě toto zapojení trimrů vynechám a nakreslím ho na další desku zvlášť. Důvodem je, že trimry a tlačítka mají jinou výšku. Takže trimry budou podepřeny ve výšce, tak aby správně dosedaly na horní kryt krabičky.



Obrázek č. 13: Deska plošných spojů vytvořená z obrázků výše

#### 4.1.2 Návrh hlavní desky plošných spojů

Po návrhu desky plošných spojů pro uživatelské rozhraní jsem začal zpracovávat návrh hlavní desky. Zde je zapojení složitější než u předchozích, proto popíši, jak jsem postupoval.

Jako první jsem na breadboard umístil Arduino a rovnou mu určil pozici na desce plošných spojů. Na desku plošných spojů jsem ho umístil tak, aby jeho konektor byl přístupný z vnějšku, až bude zařízení zapojeno v rozvaděči. Přístupný konektor je důležitý, aby se dal snadno aktualizovat program v paměti Arduina.

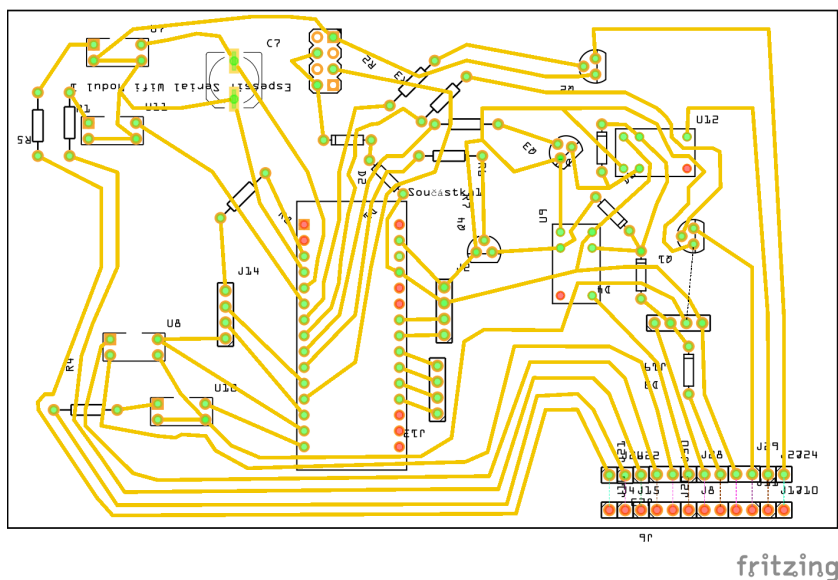
Arduino je napájeno z 5 V DC, k tomu jsem použil DC-DC měnič z 12 V na 5 V (B1205s 1 W). Napájení pro modul ESP01S jsem získal snížením 5 V na 3.3 V pomocí dvou diod. Další možností, jak napájet modul ESP01S by bylo použít stabilizátor napětí LD1117, odporový dělič nebo další DC-DC měnič.

Pro ovládání kulového ventilu použiji dvě relátka, která jsou spínaná tranzistory. Tranzistory jsem zvolil typu NPN a přes odpor v bázi je spínám z výstupního pinu centrální jednotky. Detailnější ovládání ventilu je popsáno v kapitole č. 3.2.

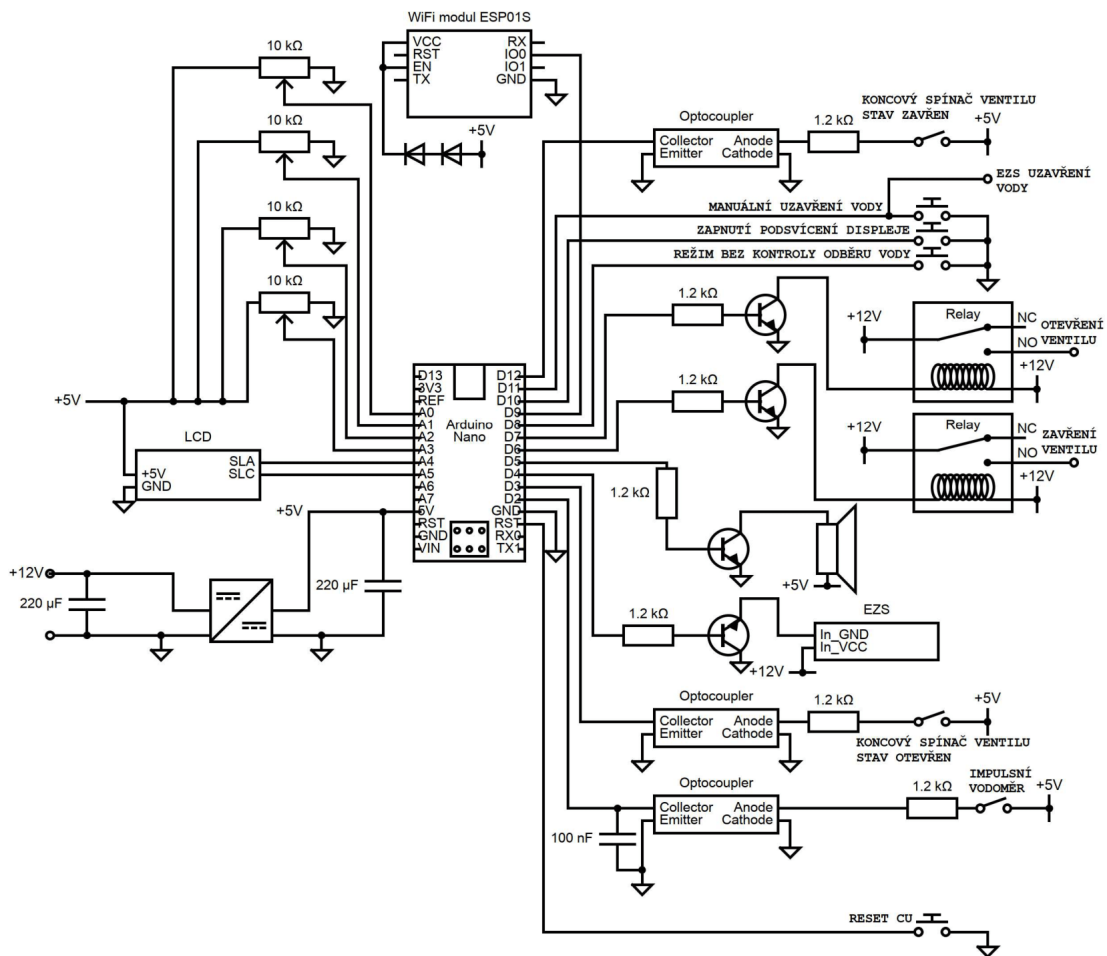
Vývody pro trimry, tlačítka a LCD displej jsou všechny v návrhu schématu desky plošných spojů zakresleny jako svorky typu male s roztečí 2.54 mm.

Koncové spínače kulového ventilu, vstup od impulsního vodoměru a vstup pro externí uzavření z EZS musí být kvůli možnému rušení galvanicky oddělené. Vzhledem k tomu, že impulsní vodoměr může být daleko od zařízení, přidal jsem pro jistotu k jeho optočlenu kondenzátor pro filtrování rušení.

Při návrhu desky plošných spojů jsem musel brát v potaz jakým způsobem upravit samotnou desku do krabičky. Hlavní deska plošných spojů se bude nacházet ve spodní části krytu, přichycená 4 šrouby. Deska plošných spojů pro uživatelské rozhraní bude uchycena také 4 šrouby. Těmto 8 místům se musím při návrhu desky plošných spojů vyhnout.



Obrázek č. 14: Návrh hlavní desky plošných spojů

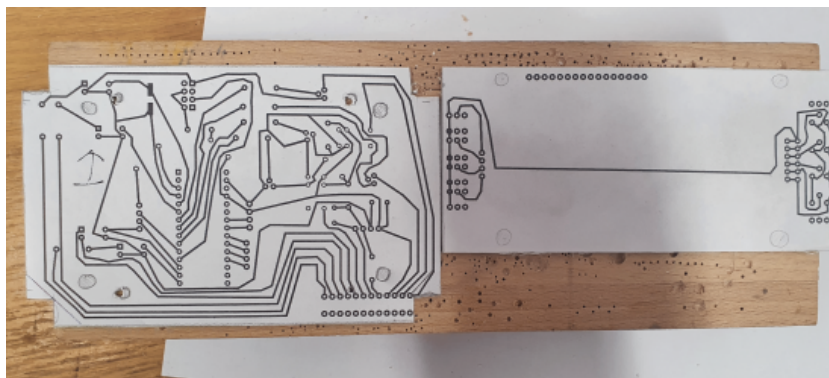


Obrázek č. 15: Schéma zapojení



### 4.1.3 Výroba desek plošných spojů

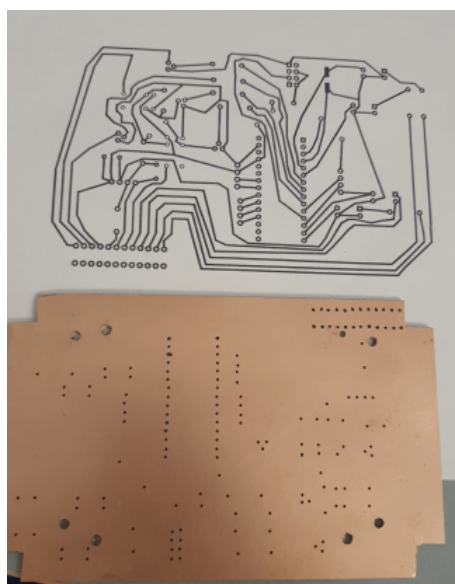
Všechny desky plošných spojů jsem vyráběl najednou a postup u každé z nich byl identický. Nejprve jsem si vyexportoval z programu Fritzing pdf všech desek, které bylo z pohledu součástek a vytiskl v měřítku 1:1 vůči finálním rozměrům. Vystříhl a nalepil na cuprexit ze strany součástek.



Obrázek č. 16: Nalepení vytisknutých pdf z Fritzing na cuprexit

Po nalepení jsem si pro všechna místa, která plánuji vyvrtat připravil důlčíkem dŕolíky, abych snadněji chytil hrot vrtáku a přesně vyvrtal všechny otvory pro uchytení šroubů a součástek. Pro součástky jsem používal vrták o průměru 0.9 mm a pro šrouby 3 mm.

Po vyvrtání je důležité zkontrolovat, zdali všechny součástky připravenými otvory projdou. Dále jsem desku ze strany plošného spoje mechanicky očistil a odmastil technickým benzínem.

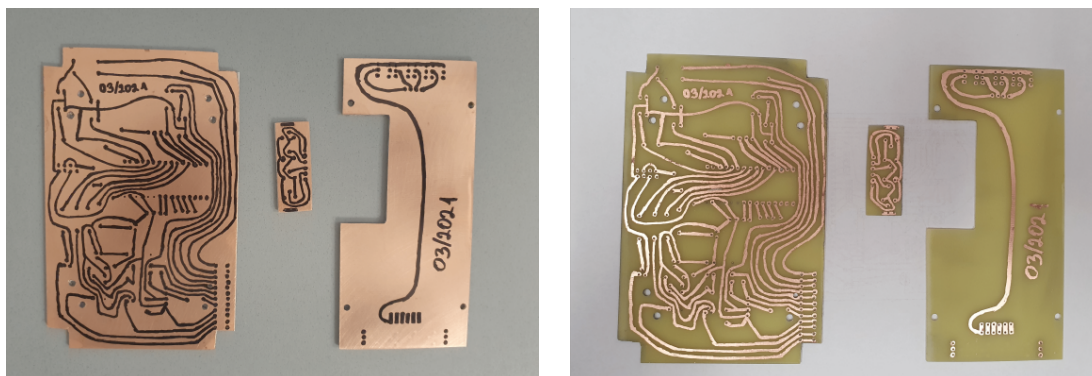


Obrázek č. 17: Zajištěná a umytá vodivá strana cuprexitu



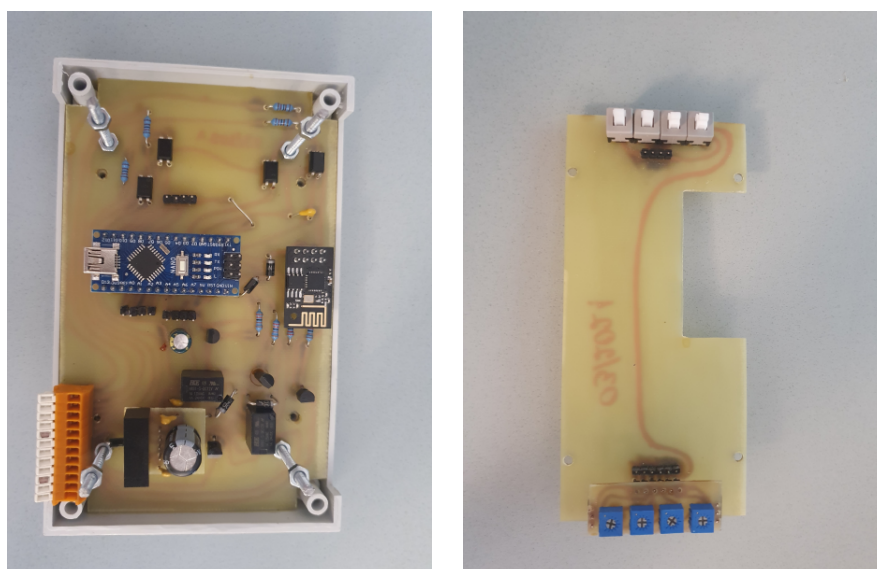
Dále je třeba nakreslit vodivé spoje. Z Fritzing jsem si vyexportoval zrcadlově obrácené desky plošných spojů, vytiskl si je v měřítku 1:1 ku skutečnému a lihovým fixem jsem od ruky začal překreslovat vodivé spoje. Po nakreslení jsem zkontroloval, zdali jsou spoje dostatečně silné, a navzájem se mezi sebou nikde nedotýkají.

Dále jsem si připravil nádobu s chloridem železitým pro leptání. Je důležité kontrolovat proces odleptávání. Pokud by někde byl spoj udělán tenčí, mohlo by dojít v tomto místě k odleptání mědi a přerušení vodivého spoje. V mém případě leptání trvalo 33 minut. Po vyleptání jsem nakreslený vzor umyl lihem.



Obrázek č. 18: Nakreslené vodivé cesty, vyleptání v chloridu železitým a umytí nakresleného vzoru lihem

Dále následuje osazení desky plošných spojů součástkami a jejich přiletování. Plošný spoj jsem na závěr ošetřil pájitelným ochranným lakem, který zabraňuje oxidaci plošného spoje.



Obrázek č. 19: Osazené všechny součástky

U hlavní desky jsem musel ještě v této fázi udělat úpravu a nahradit DC-DC měnič B1205s 1 W za 2 W verzi. Odběr WiFi modulu, Arduina a LCD displeje po zapnutí překračoval povolený maximální proud DC-DC měniče B1205s 1 W. WiFi modul potřebuje krátký impuls kolem 350 mA při zapnutí, jinak nedojde k připojení do WiFi sítě. Proto jsem musel nahradit DC-DC měnič B1205s o výkonu 1 W za B1205s 2 W, který má jiný rozměr a jinak vyvedené piny. Oproti 1 W verzi má prohozené GND a VCC napájení. Abych udělal co nejmenší zásah do již hotové desky, vyrobil jsem ještě jednu desku plošných spojů a tu jsem přes 4 male konektory napojil na připravené piny pro předchozí měnič. S tímto měničem mohu dosáhnout proudu až 400 mA, který je už dostatečný.

Při pájení součástek jsem zjistil, že mi u vstupu optočlenu pro manuální uzavření vody z externího zdroje chybí rezistor. Vyvrtil jsem tedy dva nové otvory, přerušil vodivý spoj a překlenul ho rezistorem.

## 4.2 Návrh programového vybavení

### 4.2.1 Programové vybavení pro Arduino Nano

Celý program bude napsán v jazyce Wiring. Nejprve si musím stanovit prioritu ovládání ventilu. Ventil bude možno ovládat ve 4 situacích. V normálním provozu, manuálně přes tlačítko, z online rozhraní a v poruchovém stavu. Vznikne-li požadavek z důvodu poruchy na uzavření ventilu, tak se bude ignorovat nastavení ventilu v online rozhraní a dojde k uzavření ventilu. Pokud bude zařízení přepnuto do režimu bez kontroly odběru vody, ventil zůstane vždy otevřen. Ovládání ventilu z online rozhraní bude fungovat pouze pokud není centrální jednotka v manuálním režimu nebo v poruše.

Celý program je možné rozdělit do několika bloků. Toto rozdělení jsem zvolil, aby programování bylo, co nejsnazší a nejpřehlednější. Také mi to umožní snadněji hledat případné chyby.

**První blok:** V tomto bloku kódu se věnuji vysvětlení funkce pro zaznamenání impulsu od vodoměru, výpočtu proteklého množství vody pro pozdější zobrazení na displej a vyhodnocením jednotlivých typů poruch.

Zvolený vodoměr má konstantu převodu množství proteklé vody na 1 impuls 0.3 l, to je poměrně velké množství vody a nechci, aby jakýkoliv impuls byl ztracen/zpožděn kvůli tomu, že program vykonával jinou funkci (např. reakci na tlačítko). Proto vstup od impulsního vodoměru bude připojen na digitální pin D2, protože ten má na sobě hardwarové přerušování. Základní pravidlo při práci s přerušováními je, aby jejich obsluha trvala co nejkratší dobu, proto pouze nastavím proměnnou typu bool do stavu true a na tuto změnu budu reagovat v hlavní smyčce programu, a ne v obsluze přerušování.

V hlavní smyčce programu bude každý příchozí impuls od vodoměru vyhodnocen, že proteklo 0.3 l vody. K proměnné typu float budu tedy přičítat 0.3, a dále si budu ukládat hodnotu vnitřních hodin Arduina, kdykoliv přijde impuls od vodoměru.

Hodnotou vnitřních hodin se rozumí hodnota v milisekundách od začátku běhu programu. K přečtení této hodnoty slouží funkce *millis*.

Uložení okamžiku, kdy přišel impuls mi dovolí vyhodnotit, jestli další následovné impulsy by mohly být známkou poruchy *prasklé potrubí*. Pokud praskne potrubí očekávám impulsy s velmi krátkou časovou mezerou mezi sebou. To mi umožní programem zareagovat rychleji, než když bych čekal například na vyhodnocení poruchy *maximálního dovoleného odebraného množství vody*.

Vyhodnocení poruchy *překročení maximálního odebraného množství vody* udělám následovně. Proměnnou typu float, která se bude neustále zvětšovat při protékající vodě budu porovnávat s maximální nastavenou hodnotou dovoleného množství odebrané vody. Jakmile dojde k překročení nastaveného maximálního množství vody je to vyhodnoceno jako porucha.

Vyhodnocení poruchy *překročení doby odběru bez přerušení* udělám následovně. Při příchodu pouze prvního impulsu uložím čas, který budu porovnávat s časem dalšího následujícího impulsu. Rozdílem těchto dvou časů získám dobu, jak dlouho odebírám vodu bez přerušení. Pokud doba odběru překročí hlídání parametr nastavený trimrem, vím že se jedná o poruchu.

**Druhý blok:** V tomto bloku se věnuji začlenění tlačítek a trimrů do programu. Pro okamžitou reakci tlačítek za jakýchkoliv podmínek je realizuji pomocí přerušení. Využiji k tomu registr PCICR a budu pouze kontrolovat, jestli došlo ke změně hodnoty pinu na portu C. Port C odpovídá digitálním pinům, na kterých jsou zapojena tlačítka.

Pokud je detekována sestupná hrana, nastavím bool proměnnou na true a obsluhu přerušení udělám v hlavní smyčce programu, kde pomocí funkce *digitalRead* si přečtu v jaké hodnotě se tlačítko nachází. Všechna tlačítka v sepnutém stavu připojují GND na příslušné vstupní piny centrální jednotky. Pro zamezení záskmitům od tlačítek, budu realizovat obsluhu přerušení jednou za 100 ms.

Hlídání parametry jsou: maximální množství odebrané vody, maximální čas odběru vody bez přerušení, maximální doba mezi jednotlivými impulsy pro vyhodnocení poruchy prasklé potrubí a čas nulování měření. Vzhledem k rozdílnostem odběru vody v domácnostech je třeba, aby všechny tyto parametry byly snadno měnitelné. K tomu využiji funkci *analogRead* a přes trimr budu měnit velikost napětí na příslušném vstupním pinu centrální jednotky a následně toto napětí převádět na požadovanou veličinu.

Zápis např. po jak dlouhé době nepřicházejících impulsů se má vynulovat měření: *time = map(analogRead("příslušný pin trimru"), 0, 1023, 45000, 300000)*. V jednom řádku jsou zapsány dvě funkce. Vnitřní funkcí je *analogRead*, která slouží k převodu analogové hodnoty napětí na pinu Arduina na digitální. Vnější funkce přepočítává přečtenou hodnotu, která může být v rozsahu od 0 do 1023 (odpovídá přesnosti převodníku) na hodnoty 45000 až 300000 ms. U ostatních veličin, je to provedeno stejně, pouze se mění přepočet hodnoty, použitý pin trimru a proměnná, do které se výsledná hodnota uloží.

**Třetí blok:** V tomto bloku se věnuji integraci LCD displeje do programu. Komunikaci s displejem realizuji pomocí sériové komunikace s protokolem I2C. Pro určení adresy, na kterou budu posílat z centrální jednotky data jsem použil návod zpracovaný v [7].

Adresa pro můj LCD displej je: 0x3F. Pro vytvoření komunikace použiji knihovnu `LiquidCrystal_I2C`, která je volně dostupná v Arduino IDE, ver. 1.8.13 pod záložkou projekt-> přidat knihovnu. Dokumentace k této knihovně je dostupná z [8] Příkazem `LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x3F, 20, 4)` vytvořím `lcd` objekt, který pouze funkcí `lcd.init` inicializuji. Zobrazení znaku na displej se dělá pomocí dvou příkazů. První stanoví na jaké řádce a sloupci chci mít první znak a druhý předá data přes sériovou komunikaci, která se budou zobrazovat na displej: `lcd.setCursor(0, 0)` a `lcd.print("text")`.

## 4.2.2 Programové vybavení pro WiFi modul ESP01S

V kapitole č. 3.4.1 jsem popsal, jak si připravit a nastavit webové rozhraní `iot-playground`. V této kapitole rozeberu strukturu programu pro modul ESP01S a vytvoření komunikace mezi `iot-cloudem` a modulem ESP01S.

Využiji k tomu dvě knihovny `ESP8266WiFi` a `MQTT`. Obě jsou volně dostupné v IDE ver. 1.8.13 pod záložkou nástroje a knihovny. Knihovna `MQTT` mi umožní vytvořit klienta, který se připojí na `iot-playground`. Při připojování je třeba vědět na jakou adresu a port. Oba tyto údaje získám z `api iot-playground`. Adresa je `cloud.iot-playground.com` a port je 1883. Pro úspěšné připojení k serverům `iot-playground` je nutno také vyplnit uživatelské jméno a heslo k vytvořenému účtu z kapitoly č. 3.4.1.

Dále je třeba, než se přejde do hlavního těla programu, přidat nekonečnou smyčku, která bude testovat podmínku, jestli se modul ESP01S úspěšně připojil k WiFi. Až po úspěšném připojení se bude pokračovat dále. K WiFi síti se připojuje příkazem `WiFi.begin(SSID, PASSWORD)`, kde `SSID` je název WiFi a `PASSWORD` je heslo pro připojení. Oba tyto údaje je třeba zadat jako text ohraničený uvozovkami.

Hlavní tělo programu tvoří funkce `subscribe`. `Subscribe` metoda je volána, kdykoliv dojde k požadavku ovládat ventil z online rozhraní. Při požadavku na otevření ventilu je sepnut výstup `GPIO0` na `HIGH` a při zavření na `LOW`.

# 5 Realizace a instalace

## 5.1 Instalace impulsního vodoměru a kulového ventilu

Pozice impulsního vodoměru a kulového ventilu je definována pozicí domovního vodoměru. Dle platného zákona je zakázáno cokoli zapojovat před domovní vodoměr. V našem domě se hlavní vodoměr nachází ve sklepě a při stavbě domu k němu byl připraven vývod kabelu SYKFY 5x2x0,5 mm<sup>2</sup>, který je zaveden do hlavního rozvaděče, který se nachází v 1.NP a do kterého umístím hotové zařízení. Pokud by nebyl připravený vývod kabelu u vodoměru, je třeba ho přivést, například v liště. Modifikovaný ventil a impulsní vodoměr jsem s kabelem SYKFY propojil přes svorkovnici. Místa pro instalaci kulového ventilu i impulsního vodoměru bylo za hlavním vodoměrem málo, byl jsem tedy nucen vybočit s přívodní trubkou do prostoru, jak je vidět na obrázku č. 20.

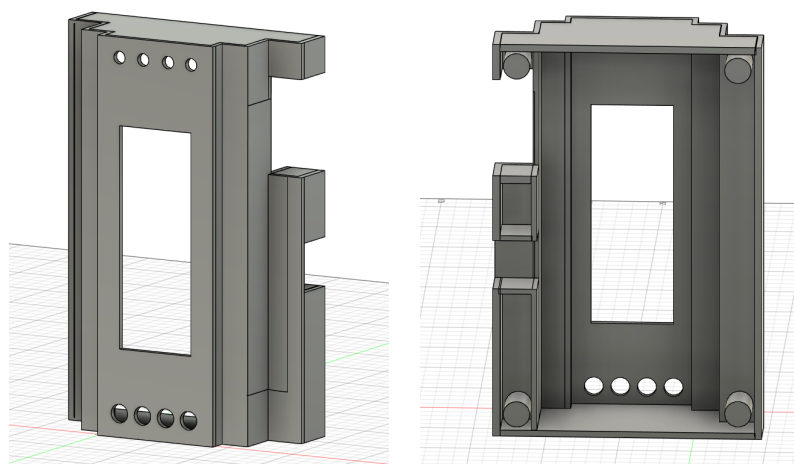


Obrázek č. 20: Nainstalovaný impulsní vodoměr spolu s kulovým ventilem na hlavním přívodu vody

## 5.2 Čelní kryt krabičky

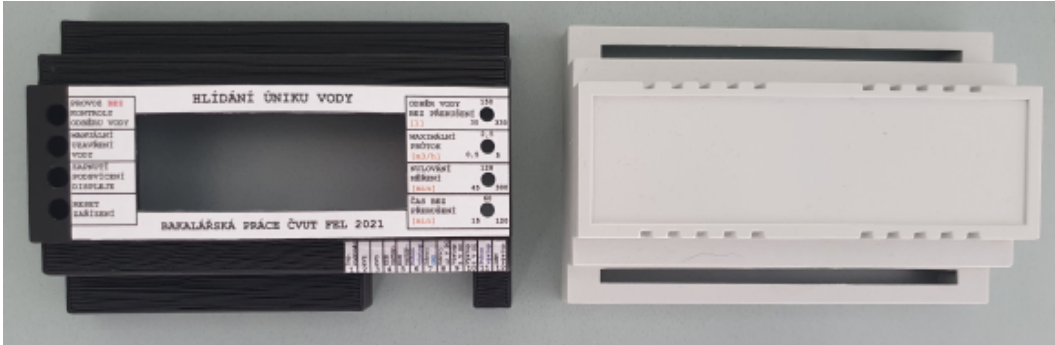
S hotovými deskami je čas sestavit a nastavit výšku desky pro uživatelské rozhraní tak, aby displej a trimry dosedaly na vnitřní stranu krabičky.

Výška displeje, tlačítek a trimrů je rozdílná a výška krabičky, která se prodává jako komplet na DIN lištu o velikosti 8 modulů je moc vysoká. Musel jsem si tedy horní kryt krabičky navrhnout sám. Ten jsem vymodeloval v programu Fusion360 a vytiskl na 3D tiskárně. V modelu jsem změnil výšku a zároveň jsem si připravil otvory pro LCD displej, tlačítka, trimry, Wago svorky a USB vývod z Arduina.



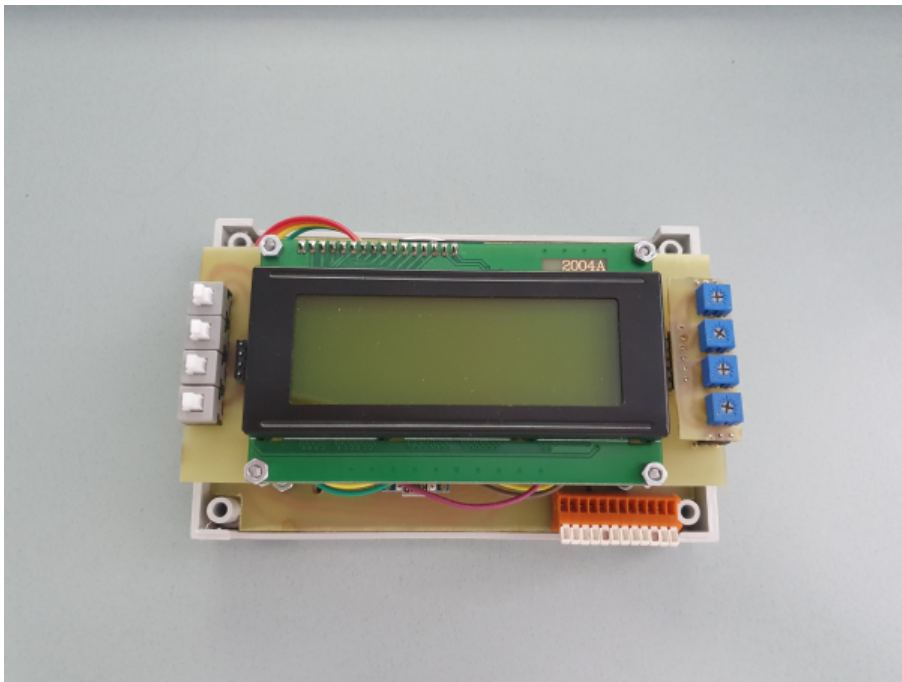
Obrázek č. 21: Model 8 modulové DIN krabičky, navržen byl 1:1 ku skutečné krabičce v programu Fusion360

Na vytisklou krabičku jsem si v programu PDF-Xchange editor vytvořil popisek, který jsem vytiskl. Opatřil ochrannou fólií, vystříhl otvory a nalepil na přední stranu krabičky.



Obrázek č. 22: Porovnání vytisklého horního krytu krabičky na 3D tiskárně a krytu 8 modulové DIN krabičky

S dokončeným horním krytem jsem mohl sestavit celé zařízení dohromady. Deska plošných spojů pro uživatelské rozhraní je uchycena přes 4ks závitové tyče o průměru 3 mm. Délka závitových tyčí je 2ks 42 mm a 2ks 36 mm, protože 2 otvory vyšly na vyvýšené místo v dolním krytu krabičky, musely být tedy zkráceny o 6 mm.



Obrázek č. 23: Sestavení celého výrobku bez horního krytu



## 5.3 Instalace zařízení do rozvaděče

Sestavené zařízení jsem nainstaloval do rozvaděče na DIN lištu. Potom jsem připojil všechny vstupy a výstupy přes svorkovnici. Navíc jsem přidal tavnou pojistku 1 A pro jistění napájení celého systému.



Obrázek č. 24: Zapojení zařízení v rozvaděči

## 6 Testování programového vybavení

Proces testování bych rozdělil na dvě fáze. V první fázi se zaměřím na to otestovat vytvořený program s umělým zdrojem impulsů se zapojením na breadboardu. Ve druhé fázi otestuji zařízení v provozu umístěné v rozvaděči, se zapojenými všemi vstupy a výstupy.

### 6.1 První fáze

Umělý zdroj impulsů vytvořím Arduinoem, kde budu tlačítka přepínat jaký typ odběru budu chtít na výstupu. Pro účely testování zvolím 3 různé typy odběru vody:

- 1) velký odběr vody, předpokládám 4 impulsy za 1 s
- 2) malý odběr vody, předpokládám 1 impuls za 30 s
- 3) normální odběr vody, předpokládám 1 impuls za 1 s

Struktura kódu bude pro všechny 3 případy stejná, jen změním dobu, kdy výstup bude v logické 0.

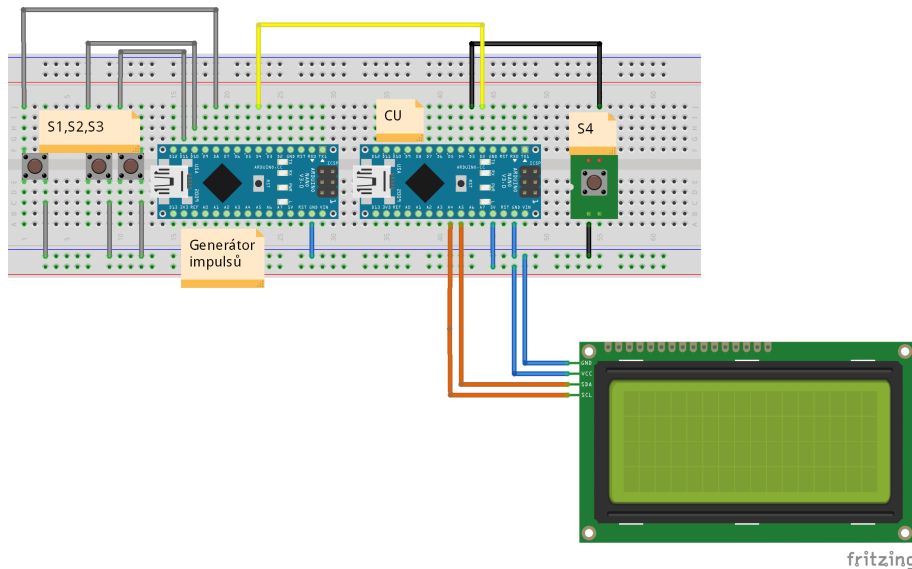
Pro co nejsnazší detekci chyb v programu bude nejlepší rozdělit postup testování do několika jednotlivých kroků. Začnu testovat detekci na jednotlivé typy průtoků a zobrazovat důležité informace na displeji - první fáze část A.

Potom přidám ovládání ventilu přes WiFi a manuální režim - první fáze část B.

Nakonec zapojím trimry a tlačítka, abych mohl měnit požadované měřené veličiny vždy při restartu centrální jednotky - první fáze část C.

### 6.1.1 První fáze část A

Schématické zapojení jsem vytvořil v programu Fritzing. Tlačítka S1 až S4 jsou nahrazena ve skutečném zapojení pro testovací účely vodiči. Funkce tlačítek S1 až S3 je čistě pro výběr typu impulsů, které dává generátor impulsů hlavnímu Arduino (na schématu označeno CU). Tlačítko S4 slouží pro rozsvícení displeje.



Obrázek č. 25: Schématické zapojení pro testování první fáze část A

Protože jsem se rozhodl pro displej místo signalizačních LED, musel jsem vymyslet, jak ho co nejlépe využít. K dispozici jsou 4 řádky, každý o 20 znacích.

- na první řádek zobrazuji aktuální odebrané množství vody [l] od posledního nulování,
- na druhý řádek zobrazuji zbývající čas, než dojde k překročení odběru vody bez přerušení [min:s],
- na třetí řádek zobrazuji stav ventilu, tedy zavřen, otevřen nebo manuální režim,
- na poslední zobrazuji stav, jestli nastala porucha nebo ne.



Obrázek č. 26: Zobrazení ve stavu bez poruchy



Jako první jsem otestoval výpočet odběru vody na displeji, samotné zobrazování na displej a funkčnost tlačítka pro rozsvěcování displeje. Dalším krokem bylo testovat jednotlivé typy poruch. Začal jsem poruchou *prasklé potrubí*. U té je nastaveno, že pokud čas mezi impulsy byl menší než 300 ms je vyhodnoceno, že impulsy přicházejí velmi rychle za sebou a s největší pravděpodobností se jedná o poruchu *prasklé potrubí*.



Obrázek č. 27: Zobrazení ve stavu porucha 1 - prasklé potrubí

Ještě jsem při testování narazil na pár nedostatků, které nebyly zohledněny při návrhu programu. Pokud byl vyslán pokyn k zavření/otevření ventilu a nepříjde informace od koncového spínače, vyhodnocuji to jako poruchu. Dále jsem ještě přidal kontrolu, pokud dojde k posláni informace od koncového spínače, že ventil byl zavřen, ale nadále budou přicházet impulsy od vodoměru, vyhodnocuji to také jako poruchu.

U kulového ventilu je doba zavírání/otevírání kolem 9 sekund. Při vyslání požadavku na zavření, může tedy stále protéct nějaké malé množství vody, a to by bylo vyhodnoceno jako porucha. Proto jsem zde přidal limit 5 impulsů, které smí přijít potom co je vyslán pokyn pro uzavření vody.

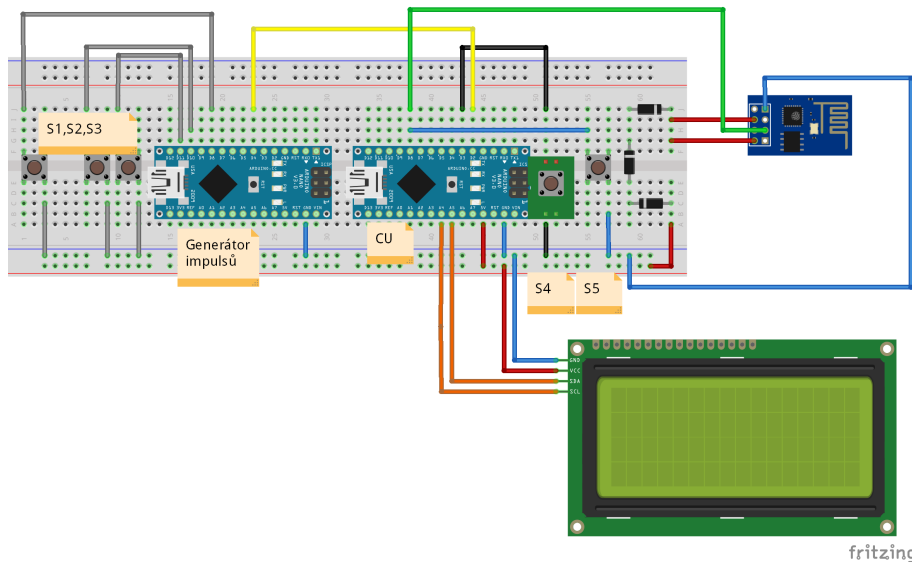


Obrázek č. 28: Zobrazení ve stavu prasklého potrubí a nedošlo k uzavření ventilu

Zbývající 2 typy poruch, jako *překročený maximální čas odběru vody bez přerušení* a *překročení maximálního množství odebrané vody* fungují správně.

### 6.1.2 První fáze část B

Oproti předchozí fázi testování jsem doplnil tlačítko S5, které reprezentuje zapnutí/vypnutí režimu bez kontroly odběru vody a přidal jsem WiFi modul.



Obrázek č. 29: Schématické zapojení pro testování první fáze část B

Při testování této fáze jsem se zaměřil na to, abych měl správně určené priority ovládání. Režim bez kontroly odběru vody musí mít největší prioritu. Pouze pokud je vypnutý, má další prioritu porucha a po ní nakonec ovládání z online rozhraní. Při testování jsem narazil na problém s WiFi modulem ESP01S. Při výpadku připojení k internetu a jeho obnovení modul ESP01S nereagoval na požadavky ovládání ventilu z online rozhraní. Vyřešil jsem to tak, že pokud nastane výpadek připojení k internetu, po půl vteřinových intervalech dochází ke kontrole, jestli je připojení obnoveno. Pokud připojení obnoveno není, čeká se dál. Ve chvíli, kdy je připojení obnoveno, restartuji modul ESP01S a načítám aktuální nastavenou hodnotu ventilu v online rozhraní.

Dále jsem v této fázi ještě doplnil možnost pro externí zavření vody, např. z EZS. Po opravě uvedené chyby s modulem ESP01S vše fungovalo v pořádku.

### 6.1.3 První fáze část C

V této fázi testování jsem přidal 4 trimry, které slouží ke změně parametrů, které hlídám v programu. Tyto parametry jsou: jak rychle za sebou musí přicházet impulsy, aby byla vyhodnocena porucha prasklé potrubí, maximální dovolené množství odebrané vody, jak velká časová mezera musí být mezi impulsy, aby došlo k vynulování měření a jak dlouho mohou odebírat vodu bez přerušení.

Protože nastavená hodnota trimru se mění v závislosti na velikosti napětí a chybě analogově-digitálního převodníku Arduina, rozhodl jsem se mít tuto možnost čtení nastavených hodnot jen pokud dojde k restartu centrální jednotky. Také to zamezí vzniku chyby, že dojde k obsluze trimru zrovna ve chvíli, kdy by měla být vyhodnocena případná porucha, či nulování měření.

Nastavené hodnoty, může uživatel sledovat na LCD displeji po restartu centrální jednotky po dobu 10 sekund. Dále je u trimrů připraven rozsah se stupnicí, kterou se může uživatel také řídit při nastavování hlídaných parametrů.

Možnost změnit veličiny, které se hlídají, je důležitá, protože žádná domácnost nemá stejný odběr vody (netráví stejnou dobu ve sprše apod.). Proto i rozsahy hodnot, je možné měnit ve velmi širokém spektru.

Hlídaný parametr	rozsah	
	od	do
Jak velký průtok je vyhodnocen jako prasklé potrubí [ms]	100	800
Maximální množství odebrané vody [l]	30	330
Po jak dlouhé nečinnosti dochází k vynulování počítání [min:s]	0:45	3:00
Jak dlouho je možné odebírat vodu bez přerušení [min]	15	120



Obrázek č. 30: Zobrazení hlídaných parametrů nastavených trimry

## 6.2 Druhá fáze

### 6.2.1 Otestování v provozu

Rozsah hlídaných parametrů jsem nastavil po změření časů strávených ve sprše členy domácnosti a hrubým odhadem. Konkrétní nastavené parametry byly:

Jak velký průtok je vyhodnocen jako porucha prasklé potrubí = 300 ms - odpovídá 3.6 m<sup>3</sup>/h odběru vody.

Maximální množství odebrané vody bez přerušení = 120 litrů.

Po jak dlouhé nečinnosti dochází k vynulování počítání = 1 minuta.

Jak dlouho je možné odebírat vodu bez přerušení = 15 minut.

Tyto nastavené parametry se postupně během provozu měnily. Například, někdo strávil ve sprše více než 15 minut. Výsledné nastavené parametry, které jsou aktuálně použitelné jsou:

Jak velký průtok je vyhodnocen jako prasklé potrubí = 550 ms - odpovídá 2 m<sup>3</sup>/h odběru vody.

Maximální množství odebrané vody bez přerušení = 150 litrů.

Po jak dlouhé nečinnosti dochází k vynulování počítání = 45 sekund.

Jak dlouho je možné odebírat vodu bez přerušení = 20 minut.



Obrázek č. 31: Zapojení zařízení v rozvaděči v provozu se zapnutým podsvícením displeje

Při testování v provozu jsem narazil na 4 problémy. První byl, že při ovládání motoru docházelo k rušení Arduina a to vyvolávalo jeho restart. Tento problém jsem vyřešil nalepováním 100 nF keramického kondenzátoru na přívodní svorky motoru. Druhý byl ten, že jsem zapomněl připravit jednu Wago svorku pro vstup do GSM modulu a vstup do impulsního vodoměru. Oba tyto problémy jsem vyřešil tím, že jsem zrušil výstup pro externí zvukovou signalizaci, a tím se mi uvolnily dvě svorky.

Třetí problém byl, že při zavření ventilu na delší dobu, například zajistí se EZS a celý den je ventil zavřený, tak tlak v trubkách klesne, protože žádné z odběrných míst není dostatečně těsné. Po opětovném otevření vody, dojde k utržení nánosů na stěnách trubky. Po nějakou dobu teče zakalená voda a proud vody není konstantní, ale nárazový. Toto trvá, dokud není tlak v trubkách vyrovnán.

Posledním problémem bylo, že pokud například myčka napouští vodu, tak hned při zajištění EZS dojde k vypnutí vody. To způsobí, že myčka mycí cyklus nedokončí. Tento problém, by šel vyřešit v programu a to tak, že pokud dojde k zabezpečení domu přes EZS, ventil se ihned neuzavře. Centrální jednotka by v tomto stavu čekala 45 minut a po uplynutí doby 45 minut by vyhodnocovala každý odběr jako havarijní.

Vzhledem k zapojení, v jakém je provedeno manuální uzavření vody a uzavření vody z EZS, tak to v mém případě není možné provést, protože nelze rozlišit, jestli byl požadavek pro zavření z EZS nebo z tlačítka. Oba dva jsou totiž zapojeni v logickém OR. Proto jsem se rozhodl prozatím nezapojovat EZS do zařízení, protože je to spíše nepraktické než praktické.

## 6.2.2 Dokumentace kódového označení poruch

Při poruše se zařízení může nacházet v jedné z 5 poruch. Každá z nich má kvůli úspoře místa na displeji vlastní označení jako porucha1 až porucha5, místo celého slovního popisu. Vysvětlení jednotlivých označení:

- Porucha1 = došlo k moc velkému odběru za krátký čas (prasklé potrubí)
- Porucha2 = došlo k dlouhému stálému odběru vody
- Porucha3 = došlo k překročení nastaveného limitu na maximální dovolené množství odebrané vody
- Porucha4 = došlo k průtoku vody, i když měl být zavřený ventil
- Porucha5 = byl zavolán pokyn pro otevření/zavření ventilu, ale centrální jednotka neobdržela do stanoveného času informaci od koncového spínače pohonu, že motor pohyb vykonal

## 7 Závěr

V této práci jsem se zabýval návrhem a vytvořením zařízení pro hlídání úniku vody. Struktura této práce byla zvolena tak, aby čtenáře nejprve seznámila se základním konceptem zařízení pro hlídání úniku vody a zvolenými komponenty pro realizaci. V dalších kapitolách byl popsán návrh a výroba desek plošných spojů pro zařízení pro hlídání úniku vody a návrh programového vybavení pro centrální jednotku a pro WiFi modul. Následující kapitola byla o nainstalování impulsního vodoměru s kulovým ventilem na hlavní přívod vody a zapojení postaveného zařízení do rozvaděče. V závěrečné kapitole došlo k otestování programového vybavení zařízení. Cílem této práce bylo navrhnout a vyrobit zařízení pro hlídání úniku vody v domácnostech, které bude řešit některé nedostatky komerčně dostupných řešení např.: dlouhodobá záloha zařízení při výpadku napájení, napájení celého systému z 12 V DC, nízká pořizovací cena a nízká spotřeba. Všech těchto cílů bylo úspěšně dosaženo.

	Vodostop [9]	eVodník [10]	Grohe [11]	Sonytron [12]	Mé řešení
<b>cena</b>	20 000 Kč (V3 verze)	20 000 Kč (Home)	11 520 Kč	od 9800 Kč	5000 Kč
<b>napájení</b>	230 V AC	230 V AC	230 V AC	12 V DC	12 V DC
<b>online komunikace</b>	ANO	ANO za příplatek	ANO	ANO za příplatek	ANO
<b>externí vstup</b>	ANO	ANO	NE	ANO	ANO
<b>manuální režim</b>	ANO	Neuvádí	NE	ANO	ANO
<b>typ ventilu</b>	NO / NC	NC	NO	Neuvádí	Dostupný v [1]
<b>GSM modul</b>	Za příplatek	Za příplatek	-	NE	-
<b>Čidla vlhkosti</b>	Nelze	Nelze	ANO	Nelze	Nelze
<b>Typ vody</b>	Studená i teplá	Studená	Studená	Studená i teplá	Studená/teplá

Tabulka č. 5: Tabulka shrnující základní vlastnosti aktuálně na trhu dostupných systémů a mého řešení

V tabulce č. 5 je možné vidět porovnání aktuálně na trhu dostupných systémů s mým řešením. Cenu mého zařízení jsem bral jako komplet, tedy veškerá elektronika, tisk krabičky, impulsní vodoměr a kulový ventil.

Spotřeba není uvedena u aktuálně na trhu dostupných řešení, protože žádný výrobce ji na svých stránkách neuvádí. Mé řešení má spotřebu při 12 V DC do 120 mA s rozsvíceným displejem a pohybujícím se ventilem. Typická spotřeba v 99 % času je do 70 mA. Při této spotřebě a při zálohování celého systému z Pb baterie o kapacitě 7 Ah, zařízení bude provozu schopné minimálně 4 dny.

## 7.1 Návrh pro vylepšení

Jedná se o zařízení, které je prototypem. Proto zde uvedu i to, jak by se celé zařízení dalo vylepšit. Desku plošných spojů bych navrhl celou znovu, tentokrát s Wago svorkami navíc, ať si může majitel zapojit například dvakrát externí uzavření vody nebo více spínaných výstupů, například zvlášť pro GSM modul a externí zvukovou signalizaci. Dále bych Arduino Nano nahradil ESP32 s vestavěným WiFi modulem dostupné z [13]. Důvod nahrazení by byl ten, protože na Arduino Nano mám vyčerpány skoro všechny piny, ESP32 disponuje dostatečným počtem pinů pro případné rozšíření počtu vstupů/výstupu na hlavní desce.

Také jsem zjistil, že ne všichni mají doma/v bytě volných 8 modulů v rozvaděči. Proto by se také dala udělat menší varianta zařízení bez LCD displeje, ale pouze se signalizačními diodami.

## 8 Použité zdroje

- [1] Ondřej Vácha. *Zařízení pro hlídání úniku vody - řešerše*. Jan. 2021.
- [2] *Důležité informace o HW Arduino Nano*. Mar. 2021. URL: [https://cs.wikipedia.org/wiki/Arduino\\_Nano](https://cs.wikipedia.org/wiki/Arduino_Nano).
- [3] Ravi Teja. *Arduino Nano specifikace a popis pinů*. Feb. 2021. URL: <https://www.electronicshub.org/arduino-nano-pinout/>.
- [4] *Zakoupený kulový ventil*. Apr. 2021. URL: <https://www.elventil.cz/variant/f.s.a.-kulovy-mosazny-kohout-0-20bar-s-pohonem-12v/28186/168>.
- [5] *Zakoupený impulsní vodoměr*. Apr. 2021. URL: <https://www.mikromarz.com/www-mikromarz-cz/eshop/50-1-Vodomery/-4-/5/712-ENBRA-ER-AM-3-4-pulsni-vystup-studena-voda>.
- [6] *Firemní literatura mikrokontroleru ESP01S*. Apr. 2021. URL: [https://docs.ai-thinker.com/\\_media/esp8266/esp8266\\_series\\_modules\\_user\\_manual\\_en.pdf](https://docs.ai-thinker.com/_media/esp8266/esp8266_series_modules_user_manual_en.pdf).
- [7] Benne de Bakker. *Character I2C LCD with Arduino Tutorial*. Jan. 2020. URL: <https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/>.
- [8] John Johnrickman. *johnrickman/LiquidCrystal\_I2C*. Sept. 2015. URL: [https://github.com/johnrickman/LiquidCrystal%5C\\_I2C](https://github.com/johnrickman/LiquidCrystal%5C_I2C).
- [9] *Firemní literatura Vodostop*. Apr. 2021. URL: <http://vodostop.cz/navody-ke-stazeni/>.
- [10] *Firemní literatura eVodník*. Apr. 2021. URL: <https://evodnik.cz/>.
- [11] *Firemní literatura Grohe*. Apr. 2021. URL: [https://www.sanitino.cz/grohe-sense-guard-intelligentni-centralni-jednotka-s-uzaverem-privodu-vody-22500ln0?utm\\_source=GooglePlatformy&utm\\_medium=Search&utm\\_campaign=GoogleMerchant&gclid=CjwKCAjwpKCDBhBPEiwAFgBzj30nzpjq7m1tYPJqmSrIq3Vz6k3PKEODQhy9EFwOYmONJmZsRZd6FB0CSnMQAvD\\_BwE#informace-o-produktu](https://www.sanitino.cz/grohe-sense-guard-intelligentni-centralni-jednotka-s-uzaverem-privodu-vody-22500ln0?utm_source=GooglePlatformy&utm_medium=Search&utm_campaign=GoogleMerchant&gclid=CjwKCAjwpKCDBhBPEiwAFgBzj30nzpjq7m1tYPJqmSrIq3Vz6k3PKEODQhy9EFwOYmONJmZsRZd6FB0CSnMQAvD_BwE#informace-o-produktu).
- [12] *Firemní literatura Sonytron*. Apr. 2021. URL: <https://www.sonytron.cz/ke-stazeni>.
- [13] *ESP32 s wifi modulem*. Feb. 2021. URL: [https://www.laskarduino.cz/espressif-esp32-wroom-32-2-4ghz-wifi-bluetooth-modul/?gclid=Cj0KCQjw0oCDBhCPARIsAII3C\\_F8ECMaH5AQhKFQNOV0l9aIpYFD\\_SyU\\_hWFhFwyGGwIej9SFpP4QaQaAgEiEALw\\_wcB](https://www.laskarduino.cz/espressif-esp32-wroom-32-2-4ghz-wifi-bluetooth-modul/?gclid=Cj0KCQjw0oCDBhCPARIsAII3C_F8ECMaH5AQhKFQNOV0l9aIpYFD_SyU_hWFhFwyGGwIej9SFpP4QaQaAgEiEALw_wcB).