



Vstupně/výstupní moduly pro řídicí systém chytré budovy

Bakalářská práce

Studijní program:

B2612 Elektrotechnika a informatika

Studijní obor:

Elektronické informační a řídicí systémy

Autor práce:

Miloslav Černý

Vedoucí práce:

Ing. Jan Kraus, Ph.D.

Ústav mechatroniky a technické informatiky





Zadání bakalářské práce

Vstupně/výstupní moduly pro řídicí systém chytré budovy

Jméno a příjmení: **Miloslav Černý**
Osobní číslo: M18000021
Studijní program: B2612 Elektrotechnika a informatika
Studijní obor: Elektronické informační a řídicí systémy
Zadávací katedra: Ústav mechatroniky a technické informatiky
Akademický rok: 2020/2021

Zásady pro vypracování:

1. Seznamte se s obecnými principy návrhu spolehlivých digitálních vstupů a výstupů pro řídicí systémy a zejména pro specifickou oblast metrologie.
2. Pro existující modulární zařízení s procesorem STM32 navrhnete a vyrobte vlastní vstupně/výstupní moduly.
3. Implementujte kompletní program pro jejich správné odečítání (zejména registraci a generování pulzů), a pro záznam a přenos naměřených hodnot po vybrané komunikační lince do nadřazeného systému.
4. Své řešení důkladně otestujte, výsledky přehlednou formou shrňte a v závěru diskutujte možnosti dalšího rozvoje tohoto zařízení.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
30–40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] STM32F413/423 [online]. [cit. 2020-10-05]. Dostupné z:
<https://www.st.com/en/microcontrollers-microprocessors/stm32f413-423.html>
- [2] STM32CubeIDE: Integrated Development Environment for STM32 [online]. [cit. 2020-10-05].
Dostupné z: <https://www.st.com/en/development-tools/stm32cubeide.html>
- [3] KING, Craig L. IEC Compliant Active-Energy Meter Design Using The MCP3905/6. 2005.
- [4] AVRAM, Sebastian; PLOTENCO, Vladislav; PAVEN, Loredana-Nicoleta. Design and development of an electricity Meter Test Equipment. In: 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP). IEEE, 2017. p. 96-101.

Vedoucí práce:

Ing. Jan Kraus, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce:

9. října 2020

Předpokládaný termín odevzdání:

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

11. května 2022

Miloslav Černý

Poděkování

Rád bych poděkoval své rodině, která mě při tvorbě práce podporovala. Panu Ing. Janu Krausovi, Ph.D. za vedení při práci, trpělivost a ochotu.

Abstrakt

Tato práce se zabývá odečtem spotřeby vody v domácnostech z pulzního výstupu vodoměrů. V práci je popsán výběr vhodného mikrokontroleru pro realizaci odečtů impulsů z vodoměru. Uvádí srovnání obdobných mikrokontrolerů dostupných na našem trhu. Dále jsou v práci popsány všechny použité periferie a programové řešení, jak správně pulzy odečítat. Následně je vytvořen rozšiřující modul pro existující desku STM32, který umožňuje zobrazovat data na displej, zápis na SD kartu a ovládání relé.

Klíčová slova

ARM, STM32, spotřeba vody, pulzní výstup S0, Nucleo

Abstract

This paper deals with the reading of water consumption in households from the pulse output of water meters. The paper describes the selection of a suitable microcontroller to implement the pulse readings from the water meter. It provides a comparison of similar microcontrollers available in our market. Furthermore, the paper describes all of the used peripherals and software solution to read the pulses correctly. Subsequently, an extension module for an existing STM32 board is developed which allows to show data on the display, write to the SD card and to control the relays.

Key words

ARM, STM32, water consumption, pulse output S0, Nucleo

Obsah

Úvod	1
1 Architektura procesorů ARM	10
1.1.1 Nucleo H743ZI2	11
1.2. Vývojové desky Arduino	13
1.2.1 Arduino vs nucleo	13
1.3. Programování procesorů STM32	14
1.4. IDE	14
1.4.1 IAR Systems	14
1.4.2 CubeIDE	14
1.4.3 SW4STM32	16
1.4.4 TrueStudio	16
1.5. Vodoměry v domácnosti	16
1.6. Technologie cyble	18
2 Hardwarové řešení	20
2.1. Displej LCD1602	20
2.1.1 8bitové zapojení	21
2.1.2 I2C sběrnice	23
2.2. Displej SSD1306	25
2.2.1 Konfigurace I2C v CubeIDE	25
2.2.2 Programování oled displeje	26
2.3. MicroSD modul	27
2.3.1 Konfigurace v CubeIDE	27
2.3.2 Programování modulu	28
2.4. Relé	29
2.5. Ochrana vstupů	29
2.5.1 Pull-down rezistor	30
2.5.2 Pull-up rezistor	30
2.6. Ochrana proti přepětí	31
2.6.1 Zenerova dioda	31
2.6.2 Optočlen	32
2.7. Ochrana proti rušení	32
2.7.1 Stíněný vodič	32
2.7.2 Ochrana vstupů proti rušení a přepětí	32
2.7.3 Ochrana proti rušení a přepětí na externím modulu	33
2.8. Návrh externího modulu	34

3	Softwarové řešení	36
3.1.	Princip odečtu otáček.....	36
3.1.1	Programové řešení odečtu pulzů	37
3.2.	Sériová komunikace	37
3.3.	Hodiny reálného času.....	38
3.4.	Časovač.....	38
3.5.	Webserver.....	39
3.5.1	Konfigurace webserveru	39
3.5.2	Spuštění webserveru.....	40
3.6.	Webový skriptovací jazyk.....	41
3.7.	CGI standard.....	41
3.8.	Webová stránka	42
4	Dosažené výsledky a testování.....	43
	Závěr.....	45
	Literatura.....	47
	Přílohy	50
	A Rozdělení desek STM32	50
	B 3D model externího modulu	51
	C Princip odečtu pulzů.....	52
	D Popis pinů procesoru	53
	E Kód webové stránky	54

Seznam obrázků

Obr. 1- Popis desky nucleo H743ZI2 [3]	11
Obr. 2- Nucleo H743ZI pinout, modrá: GPIO Nuclea, růžová: piny pro arduino, zelená: použité piny v práci [3]	12
Obr. 3-Principy používaných vodoměrů [12]	17
Obr. 4-Možnosti zapojení senzoru Cybleltron [15]	19
Obr. 5- Zapojení periferií a nuclea	20
Obr. 6-Smazání displeje	22
Obr. 7-Zápis na displej.....	22
Obr. 8-Princip I2C sběrnice [17]	23
Obr. 9-Přenos dat po I2C [18]	24
Obr. 10-Nastavení I2C v CubeIDE	25
Obr. 11-Nastavení FatFS v CubeIDE	28
Obr. 12-Výkonová relé	29
Obr. 13-Princip pull-down rezistoru [23]	30
Obr. 14-Princip pull-up rezistoru [23]	30
Obr. 15-Přepětová ochrana se Zenerovou diodou [23]	31
Obr. 16-Galvanické oddělení pomocí optočlenu [23]	32
Obr. 17-Dvoustupňová ochrana proti přepětí a rušení.....	33
Obr. 18-Spodní a vrchní strana DPS externího modulu	35
Obr. 19-Impulz při protečení jednoho litru	36
Obr. 20-Hotový modul připojený k desce Nucleo	44

Seznam zdrojových kódů

Zdrojový kód 1 - Zobrazení dat na OLED displej.....	26
Zdrojový kód 2 - Inicializace OLED displeje	26
Zdrojový kód 3 - Inicializace FatFS + SPI sběrnice	28
Zdrojový kód 4 - Zápis a ukládání na SD kartu	28
Zdrojový kód 5 - Programová realizace odečtu pulzů	37

Seznam tabulek

Tabulka 1-Vlastnosti H743ZI2 a F7F7ZI	12
Tabulka 2-Popis pinů displeje 1602.....	21
Tabulka 3-Provozní parametry zařízení.....	43

Seznam grafů

Graf 1-Spotřeba vody v domácnosti	44
---	----

Seznam zkratek

SD	Secure digital
LCD	Liquid crystal display
ARM	Advanced RISC Machine
RAM	Random access memory
IDE	Integrated Development Environment (Vývojové prostředí)
GPIO	General Purpose Input Output (Uživatelsky konfigurovatelné piny)
LED	Light-Emitting Diode (Svítivá dioda)
UART	Universal asynchronous receiver-transmitter (univerzální asynchronní přijímač-vysílač)
GSM	Groupe Special Mobile (standart pro mobilní sítě)
EMC	Elektromagnetická kompatibilita
I2C	Inter-Integrated Circuit (počítačová sběrnice)
EEPROM	Electrically Erasable Programmable Read-Only Memory (elektricky mazatelná paměť)
OLED	Organic Light-Emitting Diode
MCU	Microcontroller unit (centrální procesorová jednotka)
DPS	Deska plošných spojů
HTML	Hypertext Markup Language (značkovací jazyk pro webové stránky)
IP	Internet Protocol
SSI	Server side includes
CGI	Common gateway interface
SDA	Synchronous data
SCL	Synchronous clock
SWD	Serial wire debbuger
DDRAM	Display Data Random Access Memory

Úvod

K měření spotřeby vody se využívají vodoměry. Kvalitní a přesný vodoměr umožňuje spravedlivé rozdělení spotřeby vody a spolehlivé měření, které je základním předpokladem pro přesné údaje, a tudíž spravedlivé účtování nákladů za odpovídající množství spotřebované vody koncovým uživatelům. Tato práce se zabývá tím, jak pomocí mikrokontroleru z rodiny STM32 programově realizovat správný záznam spotřeby vody v domácnostech.

Cílem této práce je vytvořit zařízení pro detekci pulzů z výstupů S0 bytových vodoměrů a následně je převést na množství spotřebované vody. Data o množství spotřebované vody se ukládají na SD kartu a posílají se přes sériovou komunikaci do PC. Data se také po ethernetu odesílají na jednoduchý webový server. Součástí práce je výběr vhodné vývojové desky z rodiny STM32 STmicroelectronics (dále jen „ST“). Dále je zde také srovnání s ostatními vývojovými deskami nabízenými na trhu a porovnání jejich klíčových vlastností, výhod a nevýhod.

Práce je rozdělená na 3 části, a to na hardwarové řešení, ve kterém jsou popsány vybrané periferie použité ve finálním externím modulu pro desku Nucleo a poté samotný návrh modulu. Jedná se o rozšiřující modul, který rozšiřuje Nucleo o možnost zobrazování dat na displeji, zápis na SD kartu a využití relé. V další části je popis softwarového řešení detekce pulzů, jejich převod pomocí koeficientu K na množství protečené vody a následné odesílání na webserver či do PC přes sériovou linku. V následujících kapitolách bude popsán výběr desky a periférií. Dále je zde podrobný popis všech použitých periférií a je vysvětleno, proč jsem je vybral a jak s nimi lze pracovat. Je zde také vysvětlen princip standardního výstupu vodoměrů S0 a popis, jak pomocí pulzů zjistit aktuální množství spotřebované vody.

V závěru práce je zhodnocení vyrobeného modulu a jeho otestování v praxi. Modul byl zapojen v domácnosti a byl naměřen graf spotřeby vody. Z údajů zaznamenaných na SD kartě může uživatel zjistit celkovou spotřebu vody za určité období a také v jakém čase dochází během dne k větším odběrům vody. Ze záznamu lze např. snadno poznat i případný únik vody v noci nebo v době, kdy nikdo v domácnosti nebyl přítomen.

Změřením spotřeby vody v průběhu běžných pracovních činností v domácnosti jako jsou mytí nádobí, praní prádla, sprchování, splachování lze zjistit konkrétní spotřebu vody pro jednotlivou činnost. Z úsporných důvodů to pak může být podnětem pro rozhodnutí o výměně spotřebiče (úspornější pračka, myčka na nádobí) nebo případné výměně zařízení (úspornější sprchová hlavice nebo kuchyňská baterie). V úvahu připadá i využití dešťové vody místo pitné např. pro splachování WC.

1 Architektura procesorů ARM

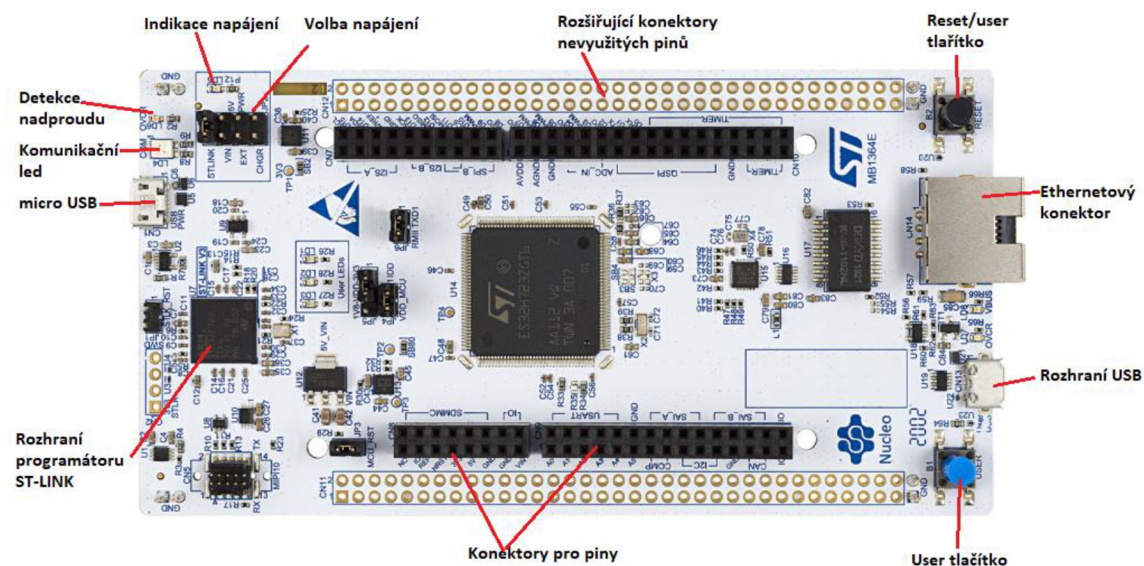
ARM je v informatice označení architektury procesorů. Tato architektura se používá hlavně díky své nízké spotřebě elektrické energie a velkému výpočetnímu výkonu. Z těchto důvodů se používá nejvíce v oblasti mobilních zařízení jako jsou mobilní telefony a tablety. Dále pak ve vestavěných systémech (pevné disky, USB flash disky, Wi-fi čipy a routery). V dnešní době ARM proniká čím dál více do počítačových systémů. V roce 2013 je ARM nejvíce zastoupenou architekturou mikroprocesorů. V roce 2013 bylo vyrobeno 10 miliard ARM procesorů, v roce 2014 se jich vyrobilo 50 miliard [1].

STM32 je rodina 32bitových integrovaných obvodů mikrokontrolerů od společnosti ST. Čipy jsou řazeny do řad, které jsou založeny na stejném 32bitovém jádru procesoru ARM. Každý mikrokontroler se skládá z jádra procesoru, statické volatilní paměti RAM, paměti FLASH, ladícího rozhraní a různých dalších periférií. Jednotlivé desky se liší ve spotřebě, v počtu pinů, ve velikosti programovatelné paměti a perifériích, která daná deska nabízí. V příloze A Rozdělení desek STM32 je vidět detailnější rozdělení desek Nucleo z hlediska spotřeby, velikosti paměti apod. [2].

V současné době existují tři základní typy desek ST Nucleo: Nucleo-32, Nucleo-64 a Nucleo-144. Každá deska nabízí jiné parametry a využití. Celkem si můžeme vybrat z 27 různých typů Nucleo desek. Snadno dostupné kity Nucleo umožňují vytvářet vlastní nápady a projekty a rychle tvořit prototypy s jakýmkoliv MCU z řad STM32. Desky v sobě mají zabudovaný programátor ST-link, takže není potřeba k desce připojovat externí programátor. Softwarové balíčky pro vývojové prostředí STM32CubeIDE (dále jen „CubeIDE“) obsahují demo příklady od jednoduchých projektů až po rozsáhlé aplikace. Tyto příklady fungují bezproblémově i v jiných vývojových prostředích jako je IAR, TrueStudio, Keil MDK – ARP apod. [3].

1.1.1 Nucleo H743ZI2

Deska Nucleo H743ZI2 viz obr. 1 je vývojový kit optimalizovaný pro vysoký výkon poháněný procesorem Cortex-M7 s pamětí L1 cache o velikosti 32Kb (16Kb pro data a 16Kb pro program). Maximální frekvence procesoru je až 400 MHz. Disponuje 2 MB programovatelnou pamětí a až 168 libovolně připojitelných pinů, které lze využít jako vstup nebo výstup. Deska obsahuje jedno resetovací tlačítko a jedno uživatelské tlačítko, které můžeme libovolně naprogramovat a používat v aplikacích. Na desce jsou také (mimo různých signalizačních LED pro indikaci napájení desky, komunikace při programování přes ST-LINK apod.) 3 uživatelské LED, se kterými můžeme pracovat. Hlavní potřebnou periferií, kterou deska disponuje, je konektor pro ethernet. Ten dále používám pro svojí aplikaci v bakalářské práci. K programování desky není potřeba žádný externí programátor, protože v desce je zabudovaný programátor ST-LINK. Konkrétně na této desce je to ST-LINK V3, který také funguje jako virtuální COM pro sériovou komunikaci UART. Deska pro své správné fungování potřebuje napájecí napětí 5 V, které je dodáváno přes kabel microUSB. Přes tento kabel se deska i programuje [3].



Obr. 1- Popis desky nucleo H743ZI2 [3]

1.2. Vývojové desky Arduino

Kvůli své jednoduchosti jsou vývojové desky Arduina jedny z nejpoužívanějších pro jednodušší aplikace. Nejsou tak složité jako třeba desky ST nucleo, které na rozdíl od Arduina disponují větší pamětí a rychlostí zpracování. Mají už také zabudované některé hodně používané periferie jako je třeba konektor pro ethernetovou komunikaci nebo slot na SD kartu. Desky Arduino jsou levné, velice jednoduché na naprogramování a nabízí velikou variaci počtu pinů.

1.2.1 Arduino vs nucleo

Výhody:

- Jednoduché programování
- Cena
- Velký výběr desek od malých modulů (Arduino Nano) až po větší a výkonnější desky (Arduino Mega)
- Platformní nezávislost
- Jednoduché připojení přes USB (není potřeba driverů)
- Rozsáhlá uživatelská komunita
- Zdarma tutoriály

Nevýhody:

- Menší paměť
- Pomalejší zpracování
- Větší spotřeba

1.3. Programování procesorů STM32

Programování procesorů se provádí v programovacích jazycích uživatelsky čitelných pro programátora jako je třeba jazyk C nebo C++. Pomocí kompilátoru se kód z jazyka C přeloží do binárního kódu souboru instrukcí, které dále zpracovává procesor. Tento soubor se uloží pomocí programátoru ST-LINK do paměti FLASH. ST-LINK je obvodový debugger a programátor pro mikrokontrolery STM8 a STM32. Moduly jednovodičového rozhraní a rozhraní JTAG/SWD se používají pro komunikaci s jakýmkoliv kontrolérem řady STM na desce. Po připojení desky přes microUSB kabel se ST-LINK chová jako FLASH disk. Po nahrání binárního kódu na tento disk se přeprogramuje paměť FLASH mikrokontroleru. ST-LINK umožňuje také velmi snadný upgrade firmwaru [4].

1.4. IDE

Rodina procesorů STM32 je podporovaná velkou škálou softwarových vývojových prostředí s podporou programovacích jazyků C, C++, Pascal a JAVA s debuggery od ST, které jsou doplněny nástroji od ST. Ty umožňují další konfigurace a inicializace MCU nebo sledování chování za běhu. Vývojová prostředí jsou jak placená, tak i zdarma [5].

1.4.1 IAR Systems

Je velmi propracované vývojové prostředí s nástroji pro správu a editování projektů. Zahrnuje velké množství příkladů s konfiguračními soubory, příklady kódů a šablony projektů. To je velkou výhodou, která umožní každému projektu rychlý start. Podporuje všechny 32bitové procesory a vybrané 64bitové procesory. Komplexní debugger C-SPY Debugger poskytuje simulátor instrukcí ARM a rozsáhlou podporu pro ladění cílových systémů. Zahrnuje pluginy RTOS a širokou podporu komunikačních zásobníků a middlewarů. Toto prostředí je sice placené, ale je zde verze zdarma na dobu 30 dní nebo do omezené paměti pro program na 32Kb [6].

1.4.2 CubeIDE

Je bezplatné a bohaté vývojové prostředí poskytované firmou ST. Je k dispozici pro Windows, macOS i Linux. Je to pokročilá vývojová platforma v C/C++ s konfigurací periférií, generováním kódu, kompilací kódu a funkcemi pro následné ladění

mikrokontrolerů a mikroprocesorů. CubeIDE zahrnuje CubeMx platformu, což je grafické prostředí pro konfiguraci různých parametrů procesoru, a hlavně k definování pinů. Vývojové prostředí obsahuje seznam všech dostupných desek a při vytváření nového projektu si musíme vybrat jakou desku používáme. Při výběru desky jsou v prostředí nabízeny i příklady právě pro konkrétní typy desek. Po zvolení správné desky a vytvoření projektu jsou zde dvě možnosti, a to načíst procesor v CubeMx viz příloha D Popis pinů procesoru se všemi piny v defaultním nastavení, nebo můžeme nechat všechny piny nenastavené. Defaultní nastavení má tu výhodu, že zde jsou popsány hlavní piny, například jsou zde uživatelské LED1 – LED3 a tlačítka, takže už nemusíme hledat v dokumentaci, na kterém pinu je co zapojené. Volné piny můžeme dále pak nastavit jako GPIO vstupy nebo výstupy. Po nastavení všech potřebných parametrů a pinů můžeme projekt uložit a tím se nám automaticky vygeneruje kód a prostředí nás přepne do programovacího okna. Vygenerovaný kód je jakousi hlavičkou, kde jsou programově definované a inicializované periferie, vstupy a výstupy podle toho, jak jsme je zvolili v CubeIDE. Jsou zde pomocí komentářů oddělené různé sekce pro psaní vlastních proměnných nebo přidávání potřebných knihoven. Ve `while(1)` smyčce se píše hlavní program, pokud není v projektu použit FreeRTOS či RTOS. Při použití FreeRTOS bychom se do `while(1)` smyčky neměli nikdy dostat a jednotlivé tasky se vykonávají v příslušných funkcích vygenerovanými pro jednotlivé tasky při vygenerování hlavičkového kódu [7].

Ze všech zde zmiňovaných vývojových prostředí je CubeIDE nejvíce uživatelsky přívětivé a umožňuje příjemné a efektivní pracování se STM32 deskami, a proto jsem se rozhodl ho používat při svém řešení. Výhodou je i zakomponovaný grafický designér CubeMX, ve kterém můžeme konfigurovat jednotlivé piny procesoru a připojené periferie. Při potřebě změny jakýchkoliv parametrů při programování stačí jen otevřít designér a potřebné změnit. Po změnách se kód vygeneruje znovu a doplní se o inicializační kódy podle toho, co jsme v nastavení změnili či přidali.

1.4.3 SW4STM32

Je vývojové prostředí pro více operačních systémů, které je založené na platformě Eclipse. Podporuje celou řadu mikrokontrolerů STM32 a souvisejících desek. Nejprve je nutné vybrat a nakonfigurovat vstupy a výstupy a také nastavení procesoru v CubeMX. Poté se může vygenerovat inicializační kód. Je zde nutno zvolit, že chceme používat SW4STM32. Vytvoří se tak projekt v jazyce C, který dále otevřeme přes programovací prostředí Eclipse. Prostředí v Eclipse je téměř totožné s CubeIDE, ale neobsahuje již CubeMx, kde můžeme konfigurovat procesor a vstupy/výstupy, což je hlavní nevýhodou tohoto prostředí [8].

1.4.4 TrueStudio

Podobná platforma jako SW4STM32. Nejdříve je potřeba inicializace v STM32CubeMX a poté můžeme programovat v TrueStudios. Aktualizace TrueStudios jsou od roku 2017 pozastaveny, ale všechny hlavní funkce jsou v bezplatném prostředí CubeIDE, které je uživatelsky přívětivější [9].

1.5. Vodoměry v domácnosti

K měření spotřeby vody v domácnostech jsou určeny bytové vodoměry. Kvalitní a přesný bytový vodoměr umožňuje spravedlivé rozdělení spotřeby vody a spolehlivé měření, které je základním předpokladem pro přesné údaje, a tedy spravedlivé účtování finančních nákladů mezi spotřebitele vody. Pro uživatele to je také jediná možnost, jak si hlídat či ověřit správnost naúčtovaných nákladů za spotřebovanou vodu [10].

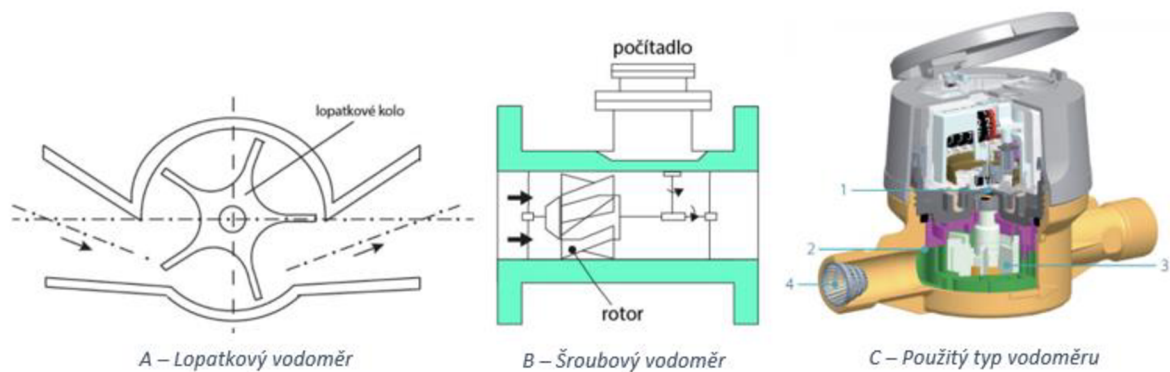
Bytových vodoměrů lze na trhu najít nezměrné množství. Liší se nejen v ceně, ale také v konstrukci. Nejdůležitějšími prvky každého vodoměru je jeho přesnost a odolnost proti okolnímu ovlivňování (manipulace s měřidly).

Z hlediska legislativy je vodoměr považován za měřidlo. Požadavky na měřidla udává Zákon č. 505/1990 Sb. Podle tohoto zákona je Český metrologický institut věcně a místně příslušný ve věci stanovování metrologických a technických požadavků na stanovené měřidlo [11].

Vodoměry se dělí podle principu funkce na lopatkové a šroubové. Hlavní součástí lopatkového vodoměru je lopatkové kolo, které se roztáčí působením protékající

vody. Otáčivý pohyb lopatkového kola se převádí přes převodové ústrojí na rotační pohyb na číselník [12].

Šroubové vodoměry pracují na principu roztočení speciálního šroubového kola, které je roztáčeno díky proudící vodě. Stejně jako u lopatkových vodoměrů se přenáší rotační pohyb na číselník. Šroubové kolo je umístěno buď horizontálně nebo vertikálně. Tyto vodoměry se používají v průmyslových objektech s vysokým odběrem vody. Zjednodušený princip viz obr. 3 B.



Obr. 3-Principy používaných vodoměrů [12]

V domácnostech se používají vodoměry s maximálním průtokem $Q_3 = 4 \text{ m}^3/\text{h}$ viz obr. 3 C. Tyto vodoměry jsou většinou jednovtokové s nátokem vody na oběžné kolo. Otáčivý pohyb oběžného kola je přímo přenášen magnetickou spojkou (1) na počítadlo bez použití dalších převodových kol. Proud vody je usměrněn v hrdle vodoměru (2) a tím se zredukuje nerovnoměrnosti proudu. Není zde potřeba používat žádné uklidňující délky potrubí. Oběžné kolo (3) je konstruováno tak, aby bylo dosaženo vysoké citlivosti při nízkých průtocích a odolnosti v případě přetížení. Filtr (4) má za úkol zachytit případné nečistoty v hrdle z potrubí [13]. Ze štítku vodoměru lze vyčíst parametry vodoměru. Na štítku každého vodoměru je uveden maximální trvalý průtok Q_3 , který je u bytových vodoměrů nejčastěji $3 - 4 \text{ m}^3/\text{h}$.

Každý vodoměr, elektroměr nebo plynoměr má možnost pulzního výstupu. Ten může být buď optický tzn. využití optočlenu nebo elektrický tzv. S0 pulzní výstup.

S0 je průmyslový standard pulzního výstupu vodoměrů a elektroměrů definovaný normou DIN 43864 [13].

Impulsy odpovídají sepnutí kontaktu na 35 až 65ms viz obr. 20. Pulzní výstup je většinou tvořen tranzistorem, který pracuje v režimu otevřeného kolektoru. V závislosti na tom, jaký typ měřiče používáme, může jeden pulz znamenat různé množství spotřebované veličiny.

- Vodoměry: 1 impuls odpovídá většinou 1 l spotřebované vody. Dále se používá i 10 l nebo 100 l, ale 1 ku 1 je nejčastější využití.
- Plynoměry: 1 impuls odpovídá 0,01 krychlového metru spotřebovaného zemního plynu apod.
- Elektroměry: u elektroměrů jeden impuls znamená většinou 1, 10, 100 nebo 1000 kWh

U většiny vodoměrů většinou jeden impuls odpovídá přímo množství dané veličiny. V dokumentacích k měřičům musí být vždy uveden koeficient K, který značí převod z počtu impulsů na množství dané veličiny.

1.6. Technologie cyble

Vodoměry od firmy Itron jsou předpřipraveny pro montáž komunikátoru pro dálkové odečty. Tyto komunikační moduly umožňují integraci vodoměrů Itron do všech typů odečtových a monitorovacích systémů. Itron poskytuje uživatelům komplexní hardwarové a softwarové řešení na bázi mobilních odečtových systémů, systémů s pevnou sítí nebo GSM přenosů dat. Mají standardní komunikační výstup. Díky nemagnetickému principu je nelze ovlivnit vnějším magnetickým polem. Výrobce zaručuje odolnost proti rušení podle norem EN 50081-1, EN 50081, EN 50082-1, EN 50082-2 [15].

Komunikační moduly jsou konstruovány tak, aby naplnily požadavky vodárenských společností zamýšlejících dálkově odečítat svá měřidla. Unikátní patentovaná technologie detekce zpětného průtoku zajišťuje kompenzaci pulzů tak, že stav počítadla a hodnota dálkového odečtu jsou vždy identické [15].

Komunikátory se vyrábějí v několika variantách, které se liší ve způsobu komunikace, zapojení, převodní konstanty K a maximálního průtoku Q_3 . Vyrábí se ve dvou variantách, a to dvou vodičová a pětivodičová viz obr. 4. V pětivodičovém zapojení jsou dva pulzní výstupy VF a HF s detekcí přerušení zemního vodiče a detekcí zpětného toku vody. Pulzní výstupy VF a LF se liší v délce pulzu. Komunikátor, se kterým byla práce otestována, využívá pětivodičové zapojení s konstantou $K=1$ pro NF pulzní výstup [15].



Obr. 4-Možnosti zapojení senzoru Cybleltron [15]

DIR – směrový vstup indikuje směr průtoku (normální nebo zpětný).

VF – vysokofrekvenční signál detekující otáčky terče senzoru Cyble.

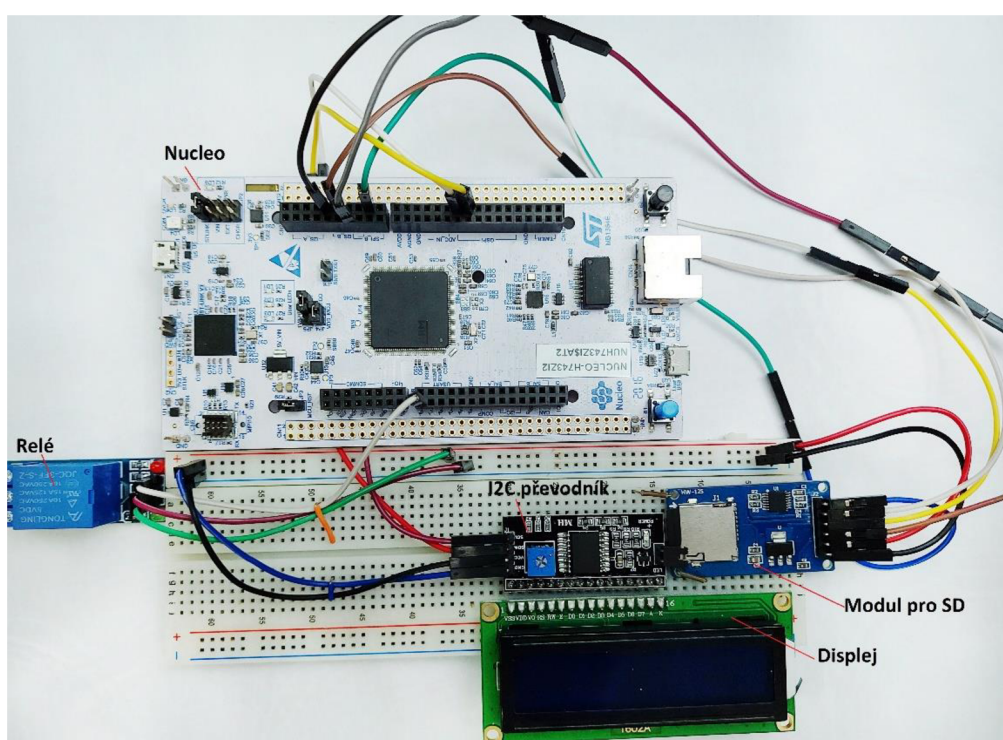
NF – nízkofrekvenční signál detekující otáčky terče senzoru Cyble.

Poškození kabelu – detekce poškození kabelu, kabel je monitorován změnou proudu v zemní smyčce.

2 Hardwarové řešení

V této kapitole jsou popsány všechny periferie, které jsem použil pro návrh a následnou realizaci externího modulu pro odečítání pulzů z pulzních výstupů S0 bytových vodoměrů. Na obr. 5 je vidět prvotní a také testovací zapojení Nuclea a všech periférií pomocí nepájivého pole.

Na dlouhodobější používání je tento způsob zapojení nevhodný. Pro sjednocení periférií a Nuclea do jednoho celku byl navrhnout externí modul, který v sobě implementuje všechny používané periferie jako je SD karta, relé, displej a ochranu vstupů proti rušení či přepětí, jak již bylo zmíněno v kapitole 2.7.



Obr. 5- Zapojení periférií a nuclea

2.1. Displej LCD1602

LCD displeje se u nás objevily koncem 80. let. Dříve byly považovány za zázrak techniky. Dnes už jsou LCD displeje všude kolem nás. Setkáváme se s nimi v autobuse při nákupu jízdenek, v telefonních automatech nebo na parkovištích.

Každý má v sobě speciální integrovaný obvod neboli řadič, který celý displej ovládá a komunikuje s okolím. Standardem se stal řadič HD44780 od firmy HITACHI. Všichni výrobci ho používají, anebo používají nějaký upravený ekvivalent tohoto řadiče. Při výběru LCD displeje tedy tolik nezáleží na výrobci, protože všichni používají stejnou nebo téměř identickou technologii. Displeje se vyrábí v několika variantách a liší se počtem znaků a řádků [16].

Je to asi jedna z nejjednodušších možností z hlediska ceny a složitosti ovládání, kterou můžeme použít pro zobrazení textu či nějakých dat. Umožňují 8bitové nebo 4bitové zapojení. Použitý způsob zapojení se odráží na počtu potřebných vodičů k zapojení. Při použití rozšiřujícího modulu převodníku na I2C komunikaci lze řízení displeje výrazně zjednodušit a také se nám zredukuje počet vodičů na minimální možné množství.

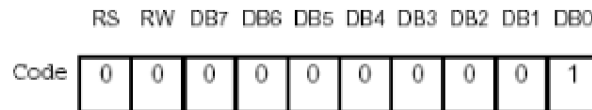
2.1.1 8bitové zapojení

Pro komunikaci s displejem je potřeba minimálně 6-11 vodičů a 2 vodiče pro přivedení napájecího napětí viz tabulka 2. V případě 8bitové komunikace (DB0-DB7) se pošle všech 8 bitů najednou a potvrdí se signálem enable. Po připojení procesoru a displeje je nutné inicializovat displej v programovacím prostředí. Je nutné nastavit počet bitů datové komunikace, směr psaní znaků, posun řádků, pozice kurzoru, blikání kurzoru. Tento způsob komunikace je jednoduchý a výkonný z hlediska zaneprázdnění procesoru.

Pin	Název	Popis	Pin	Název	Popis
1	VSS	GND	9	DB2	Data bus 2
2	VDD	5 V	10	DB3	Data bus 3
3	VO	kontrast displeje	11	DB4	Data bus 4
4	RS	Register select	12	DB5	Data bus 5
5	R/W	Read/write	13	DB6	Data bus 6
6	E	Enable	14	DB7	Data bus 7
7	DB0	Data bus 0	15	A	Podsvícení LCD anoda
8	DB1	Data bus 1	16	K	Podsvícení LCD katoda

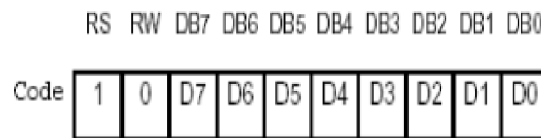
Tabulka 2-Popis pinů displeje 1602

Na obr. 6 je znázorněná instrukce pro smazání displeje. Bit RS je nastaven na nulu, což znamená, že je zapisována instrukce. Zapiše kód 20H (prázdná instrukce) do všech adres DDRAM. Potom je nastavena paměť DDRAM na adresu 0 adresového čítače a návrat displeje k jeho originálnímu stavu [16].



Obr. 6-Smazání displeje

Na obr. 7 je znázorněn princip způsobu zapisování znaků na displej. Každý znak, který je zobrazen na displeji, má svou adresu v DDRAM paměti. Na displeji je možné zobrazit znak na konkrétním místě [16].



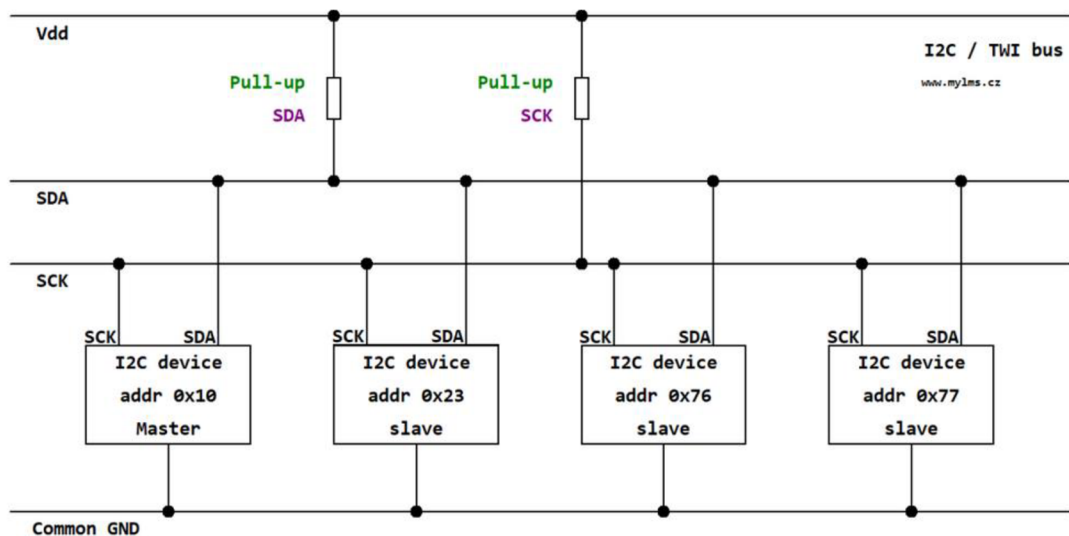
Obr. 7-Zápis na displej

Bit RS nastaven na 1 = data. Bit RW nastaven na 0 = zápis dat DB7-DB0.

Pro ještě větší úsporu datových vodičů je možné použít převodník pro LCD displeje na datovou komunikaci I2C viz obr. 5. Na komunikaci je nutno použít pouze dva datové vodiče a dva vodiče pro napájení. Na modulu je zabudovaný i napěťový dělič, kterým můžeme volit kontrast displeje. Nevýhodou je však nutnost použít k samotnému displeji další modul navíc.

2.1.2 I2C sběrnice

Sběrnice I2C je dvou vodičové datové spojení mezi jedním nebo několika procesory (Masters) a speciálními periferními součástkami (Slaves). Všechny součástky jsou připojeny na stejnou sběrnici a jsou cíleně vybírány svými adresami. Adresa a data se přenášejí stejnými vodiči. Sběrnice umožňuje velmi jednoduché propojení mezi několika integrovanými obvody a bezproblémové dodatečné rozšiřování. Mohou být připojeny všechny integrované obvody, které zvládají speciální protokol sběrnice. Mimo integrovaných obvodů RAM, EEPROM, obvodů pro rozšíření portů, A/D a D/A převodníků existuje celá řada speciálních integrovaných obvodů, jako jsou budiče displejů nebo integrovaných obvodů pro televizní a audio techniku. Sběrnice I2C používá sériovou datovou linku SDA a linku hodinového signálu SCL viz obr. 8. Data a adresy se přenášejí podobně jako v posuvných registrech společně s hodinovými impulzy. Obě linky je možno využít jako obousměrné [17].



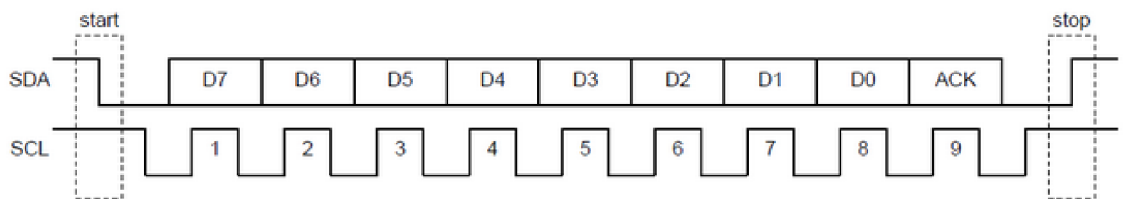
Obr. 8-Princip I2C sběrnice [17]

Neaktivní účastníci sběrnice mají většinou vysokou impedanci, neustále však vyhodnocují signály na sběrnici. Je-li použit jen jeden master, vydává hodinové signály jen on. Data však může vysílat jak master, tak i slave.

Protokol I2C má řadu definovaných stavů, které každému účastníkovi umožňují rozeznat začátek i konec přenosu a také své možné adresování.

1. Klidový stav: SDA i SCL jsou na úrovni HIGH a tím jsou tedy neaktivní.
2. Podmínka startu: SDA je masterem stažena na nízkou úroveň, zatímco SCL zůstává HIGH
3. Podmínka stopu: SDA přejde z úrovně LOW na HIGH, SCL zůstává v úrovni HIGH.
4. Potvrzení: Příslušný přijímač potvrzuje příjem bytu nízkou úrovní na SDA, dokud master nevyšle devátý hodinový impuls na SCL. Potvrzení současně znamená, že se má přijímat další byte. Požadované ukončení se musí ohlásit neexistencí potvrzení. Vlastní ukončení viz bod 3.

Pokud je na linku připojen jeden master a několik slave obvodů, tak pro poslání datového rámce některému ze slave zařízení master pošle v prvních sedmi bitech adresu slave zařízení. Osmý bit obsahuje log. 1 a čeká se na příjem potvrzovacího bitu. Poté master vyšle data po lince SDA. V případě, že je potřeba získat data ze slave zařízení, pak master opět pošle adresu, ale v osmém bitu je místo log. 1 log. 0. Slave tak pozná, že má vysílat data. Synchronizace přenosu dat probíhá hodinovými impulsy na lince SCL o frekvenci 10 až 400 kHz. Průběh zapisování dat je vidět na obr. 9 [18].



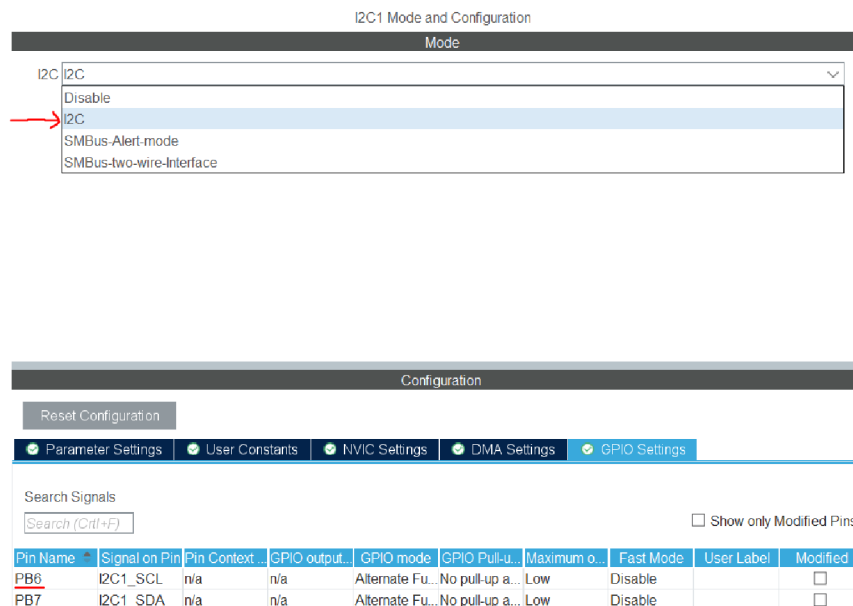
Obr. 9-Přenos dat po I2C [18]

2.2. Displej SSD1306

Displej na obr. 5, který jsem používal dříve při softwarovém návrhu, se dříve velmi běžně používal. V nynější době však působí až archaicky. Hlavní nevýhodou, kvůli které jsem se rozhodl použít jiný displej, jsou jeho velké rozměry. Disponuje vcelku omezenými možnostmi v množství informací, které se na něm dají zobrazit. V kombinaci s potřebným externím modulem pro I2C se cena vyšplhá až dvojnásobek ceny, za kterou se dá pořídit modernější OLED displej, který už je sám schopen komunikovat po I2C sběrnici. Také je rozměrově mnohem menší a dá se na něm zobrazit daleko více informací. Z těchto důvodů jsem se nakonec rozhodl využít v práci OLED displej.

2.2.1 Konfigurace I2C v CubeIDE

Po správném zapojení displeje a I2C převodníku s deskou nucleo je nutné nastavit v programovacím prostředí CubeIDE typ I2C sběrnice, se kterou budeme pracovat a jaké piny použijeme jako SCL a SDA. Podrobnější nastavení v IDE je vidět na obr. 10.



Obr. 10-Nastavení I2C v CubeIDE

2.2.2 Programování oled displeje

Po nastavení v CubeIDE už můžeme začít psát kód. Je vhodné si stáhnout už nějakou hotovou knihovnu pro komunikaci přes sběrnici I2C pro tento displej. Usnadní nám to práci i programování a nemusíme psát celou knihovnu pro jednotlivé funkce. Základem pro správnost funkčnosti displeje je jeho inicializace v inicializační části programu viz zdrojový kód 2. V Main funkci programu už můžeme začít psát nějaké funkce podle knihovny `ssd1306.c`. Instrukce v Main funkci se vykonají po spuštění programu pouze jednou, proto se zde nacházejí různá nastavení a inicializace. Pokud bychom chtěli vytvořit např. čítač, tak je nutné kód psát už do `while` smyčky, ve které se kód opakuje neustále dokola pokud program neukončíme [19].

```
SSD1306_Init();//Inicializace displeje
```

Zdrojový kód 2 - Inicializace OLED displeje

```
while (1)
{
    /* USER CODE END WHILE */
    SSD1306_Clear();
    SSD1306_GotoXY(10,10);
    SSD1306_Puts("Webserver started",&Font_11x18,1);
    SSD1306_GotoXY(10,28);
    SSD1306_Puts("WaterMeasure",&Font_11x18,1);
    SSD1306_GotoXY(0,53);
    SSD1306_Puts("V3.0",&Font_7x10,1);
    SSD1306_UpdateScreen();
    HAL_Delay(2000);
    SSD1306_Clear();
    /* USER CODE BEGIN 3 */
}
```

Zdrojový kód 1 - Zobrazení dat na OLED displej

2.3. MicroSD modul

Jelikož náš vývojový kit Nucleo v sobě nemá přímo zabudovaný slot pro SD kartu, musíme pro její používání připojit k desce modul pro SD karty. Jedná se o vcelku levnou a velmi dostupnou periférii. Například vývojový kit STM32H747I-DISCO v sobě má přímo slot pro SD kartu a příslušné zapojení, takže se karta může rovnou připojit do desky a lze s ní dále pracovat.

Na obr. 6 je použitý modul pro SD kartu, který připojíme k naší desce. Modul používá SPI komunikační protokol. K napájení je zde potřeba 5 V, takže lze použít napájení rovnou z vývojové desky. Je nutné dávat pozor na typ použitého modulu pro SD kartu, protože většina z nich nedisponuje stabilizátorem, který z 5 V vyrobí 3,3 V. Ty jsou potřeba pro řídicí obvod modulu [20].

Připojení k desce je velmi jednoduché, stačí se podívat do dokumentace, kterou SPI sběrnici budeme používat a najít odpovídající MOSI, MISO, CSK a CS piny. Připojit napájení a zem a připojení máme hotové. Složitější je nastavení pro práci s SD kartou v CubeIDE viz obr. 11.

2.3.1 Konfigurace v CubeIDE

Nejdůležitějším krokem je povolit v CubeIDE práci s FatFS, což je modul souborového systému FAT/exFAT pro malé vestavěné systémy. Modul FatFs je napsán v souladu s ANSI C a je zcela oddělen od diskové I/O vrstvy. Proto je nezávislý na platformě. Může být začleněn do malých mikrokontrolerů s omezenými zdroji, jako je třeba 8051, AVR, ARM atd.

Následuje nastavení sběrnice SPI ve Full-Duplex-Master režimu. Pro správnou funkčnost zápisu a čtení na SD kartu je doporučeno volit rychlost sběrnice okolo 10Mbit/s. Při vyšších nebo nižších rychlostech by SD karta nemusela fungovat správně nebo vůbec.

2.3.2 Programování modulu

Do příslušných adresářů Inc a Scr, kde se nacházejí hlavičkové soubory, je potřeba nakopírovat stažené knihovny pro práci s SD. V inicializační části programu musíme nastavit všechny používané periferie viz zdrojový kód 3.

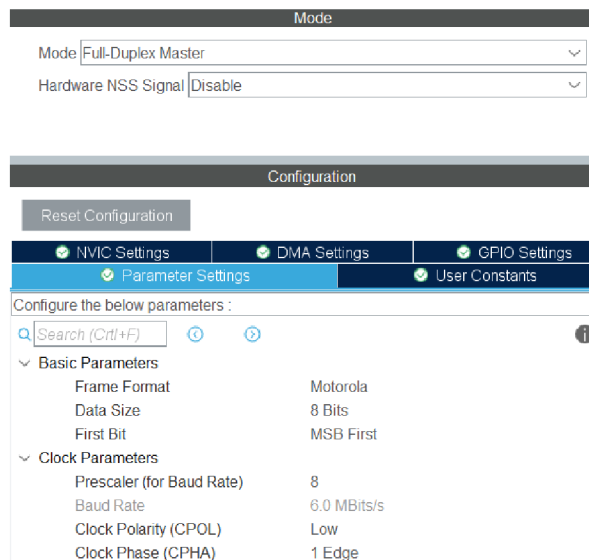
```
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USB_OTG_FS_PCD_Init();
MX_SPI1_Init();
MX_FATFS_Init();
/* USER CODE BEGIN 2 */
/* Waiting for the Micro SD module to initialize */
```

Zdrojový kód 3 - Inicializace FatFS + SPI sběrnice

Po inicializaci už můžeme psát příkazy pro SD kartu z knihoven. Nejprve je nutné říci kartě, že se má připojit. Poté můžeme založit nový soubor, nebo otevřít již vytvořený textový dokument uložený na SD. Po zapsání textu na kartu je nutné soubor zavřít a kartu zase odpojit.

```
fres = f_mount(&fs, "", 1);
if (fres == FR_OK) {
    sprintf(sdinfo, "Micro SD card is mounted successfully!\n");
    HAL_UART_Transmit (&huart3, sdinfo, sizeof (sdinfo), 10);
} else if (fres != FR_OK) {
    sprintf(sdinfo, "Micro SD card's mount error!\n");
    HAL_UART_Transmit (&huart3, sdinfo, sizeof (sdinfo), 10);
}
```

Zdrojový kód 4 - Zápis a ukládání na SD kartu



Obr. 11-Nastavení FatFS v CubeIDE

2.4. Relé

Pro ovládání elektromagnetického ventilu je zde použito výkonové relé. Důležitým parametrem při volbě typu relé je spínací napětí, tedy napětí, při kterém vznikne v cívce taková síla, která přitáhne kontakt. Spínací napětí bývají většinou 5 V, 12 V nebo 24 V. Dalším důležitým parametrem je velikost spínaného napětí. Většina elektromagnetických ventilů potřebuje spínací napětí 230 V. Potřebujeme relé, které budeme moci spínat pomocí 5 V a bude schopno sepnout 230 V ze sítě. Pro tyto účely by stačil standardní typ relé SRD-05VDC-SL-C viz obr. 12 A [21].



Obr. 12-Výkonová relé

S tímto relé jsem ze začátku práce pracoval, ale při návrhu DPS modulu pro Nucleo viz příloha B 3D model externího modulu jsem zjistil, že mi nevyhovují rozměry relé. Použil jsem nakonec relé FTR K1CK005W [22], které má téměř identické vlastnosti viz obr. 12 B.

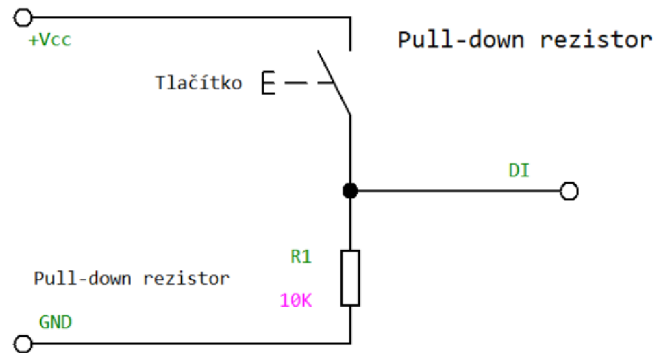
Relé se spíná pomocí 5 V, tudíž lze použít napětí z Nucleo desky a je schopno spínat až 440 VAC nebo 300 VDC. Má dielektrickou pevnost 5 kV a kategorii přepětí III a [22].

2.5. Ochrana vstupů

Pokud je potřeba připojit ke vstupním pinům napájecí napětí 5 V nebo 3,3 V, je možné to udělat přímo. Problém může nastat, pokud je napětí menší nebo větší. V tomto případě je potřeba jej oddělit. Digitální vstupy detekují buď logickou 0 nebo logickou 1. Všechny GPIO jsou v režimu vysoké impedance nebo v nedefinovaném stavu. Vstupní impedance bude velmi vysoká (řádově 100 M Ω). Z toho vyplývá, že odebíraný proud bude velmi nízký. Pokud na pinu nebude nic připojeno, může se stát, že se na pinu indukuje napětí, které náhodně mění logickou hodnotu. Pin pak bude v tzv. nedefinovaném stavu. Vstup je nutné vždy připojovat přes tzv. pull-up nebo pull-down rezistor. Tím se zajistí stálá nízká nebo vysoká logická úroveň [23].

2.5.1 Pull-down rezistor

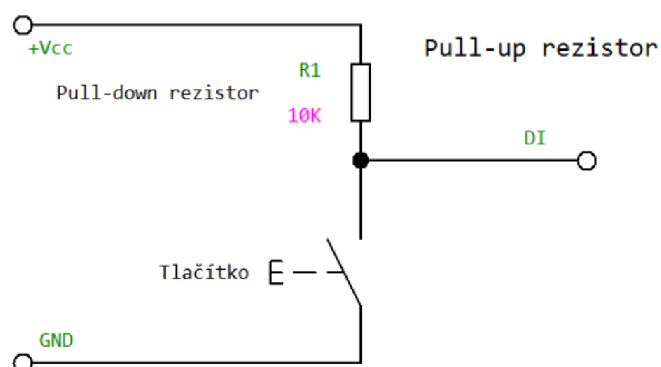
Je připojen mezi svorku GND a vstupní pin. Rezistor stáhne případné rušivé signály na zem, a tak bude na nepřipojeném vstupu stále log úroveň 0. V případě připojení signálu na vstup díky velkému odporu stoupne napětí na takovou úroveň, že se vstup bezpečně sepne.



Obr. 13-Princip pull-down rezistoru [23]

2.5.2 Pull-up rezistor

Jeho funkce je podobná jako u pull-down rezistoru. Jediný rozdíl je, že se připojuje proti kladnému napájecímu napětí a vstupní signál do procesoru je tak negovaný. V případě připojení tlačítka je třeba jednu stranu tlačítka připojit na GND a druhou na vstup. Použít lze také 10 K Ω odpor. Procesory mají již pull-up rezistory zabudované a je jen nutné nastavit v používaném IDE, že je chceme používat. V případě Nuclea je to příkaz GPIO_PULLUP. Tím dojde k připojení vnitřního pull-up rezistoru ke vstupnímu pinu. Hodnoty se liší v závislosti na typu procesoru. Většinou se hodnoty pohybují okolo 20 K Ω do 150 K Ω [23].



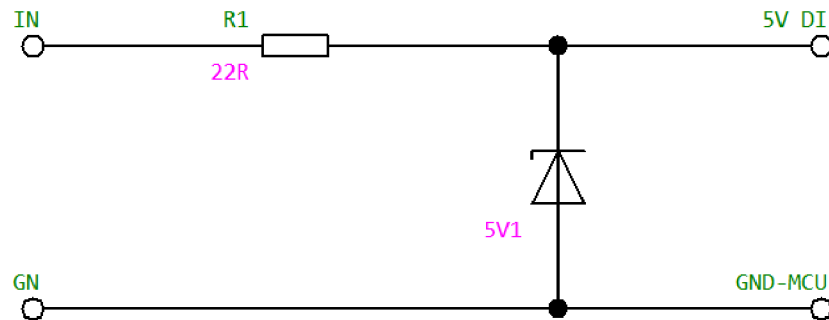
Obr. 14-Princip pull-up rezistoru [23]

2.6. Ochrana proti přepětí

Přepětová ochrana chrání elektrická zařízení před poškozením přepětím vyšším, než které je zařízení schopno vydržet. Za přepětí lze považovat napětí, které je 2x větší než jmenovité dovolené napětí. Nejčastější hodnota se používá odporu 10 k Ω .

2.6.1 Zenerova dioda

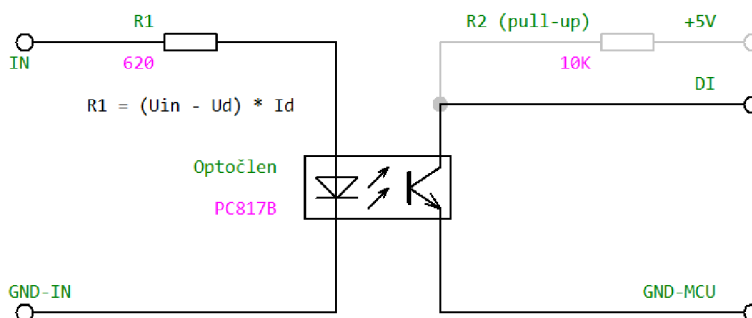
K omezení vstupního napětí je možné použít jednoduchý parametrický stabilizátor se Zenerovou diodou, jejíž napětí bude nastavené na vstupní napětí mikrokontroleru. Při přivedení přepětí na vstupní svorky Zenerova dioda omezí napětí na cca 5 V, při mnohonásobně větším napětí na vstupu je zde odpor R1 jako destruktivní součástka, který přehoří a rozpojí tak obvod. Výhodou tohoto zapojení je možnost použití širšího rozsahu vstupního napětí.



Obr. 15-Přepětová ochrana se Zenerovou diodou [23]

2.6.2 Optočlen

Galvanicky odděluje vstup mikrokontroleru od napětí na vstupu. Vhodnou konstrukcí DPS je pak zajištěna ochrana před napětovými špičkami až v řádech kV. Při přepětí dojde k poškození optočlenu, ale samotné MCU je nepoškozeno. K diodě je nutno vypočítat předřadný odpor R1.



Obr. 16-Galvanické oddělení pomocí optočlenu [23]

2.7. Ochrana proti rušení

2.7.1 Stíněný vodič

Stíněný vodič je elektrický kabel, který má kolem svých vodičů další vodivou vrstvu pro elektromagnetické stínění. Tato vrstva plní funkci ochrany proti rušení. Stínění je většinou realizováno jako opletení stávajících vodičů měděnými drátky. Tento oplet bývá zpravidla ještě zakryt izolační vrstvou [24].

Funguje na principu Faradayovy klece. Snižuje elektrický šum a zlepšuje tak kvalitu přenášeného signálu. Omezuje elektromagnetické záření, které je vyzařováno vodičem samotným a může nepříznivě působit na zařízení v jeho okolí. Stínění také chrání vodič před rušivým šumem ostatních zařízení [24].

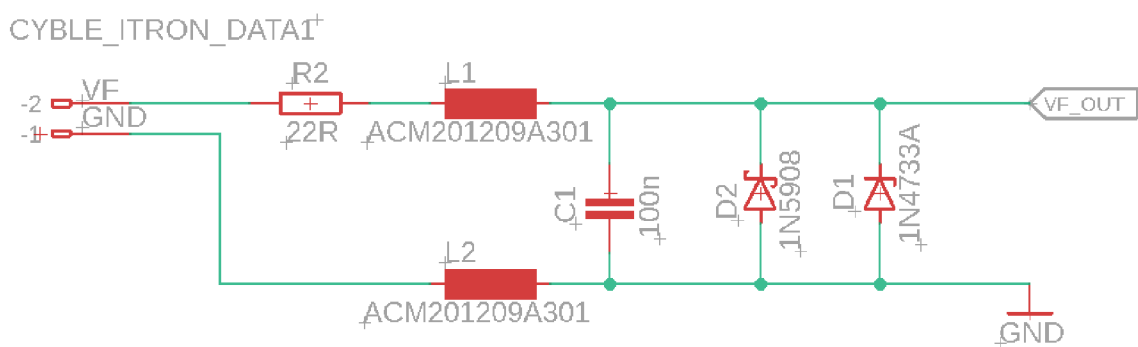
2.7.2 Ochrana vstupů proti rušení a přepětí

Vstupy MCU je třeba chránit proti přepětí, pokud je obvod pouze na jedné DPS bez vnějších obvodů. V prostředí bez rušení není většinou třeba řešit nějaká opatření. Často se lze setkat s tím, že jsou jednotlivé vstupy procesoru použity jako I2C sběrnice atd. pro teplotní čidla či jiná zařízení. Bez stíněného kabelu je velice pravděpodobný vznik rušení. Vodiče slouží jako anténa a může se na nich naindukovat napětí, které je schopné zničit nějaký vstup procesoru v horším případě i celý procesor.

Je vhodné vstupy zabezpečit před rušením i přepětí. Lze použít diodu paralelně ke vstupu, která v ideálním případě chrání vstup proti připojení napětí opačné polaroty (také možno použít vhodně dimenzovaný transil nebo Zenerovu diodu). V obvodu na obr. 17 jsou na vstupu zapojeny cívky, jejichž funkcí je ochrana proti rušení a také i přepětí. Při přepětí slouží víceméně jako pojistka a shoří. Vhodné je i vstupy provléknout feritovým kroužkem, který je schopen odstínit vyšší frekvence. Na vstup se také zapojují odpory o malých hodnotách např. 22 Ω , které fungují na podobném principu.

2.7.3 Ochrana proti rušení a přepětí na externím modulu

Na obr. 17 je schéma přepětové ochrany použité na externím modulu Nuclea, jehož návrh bude popsán detailněji v dalších kapitolách.



Obr. 17-Dvoustupňová ochrana proti přepětí a rušení

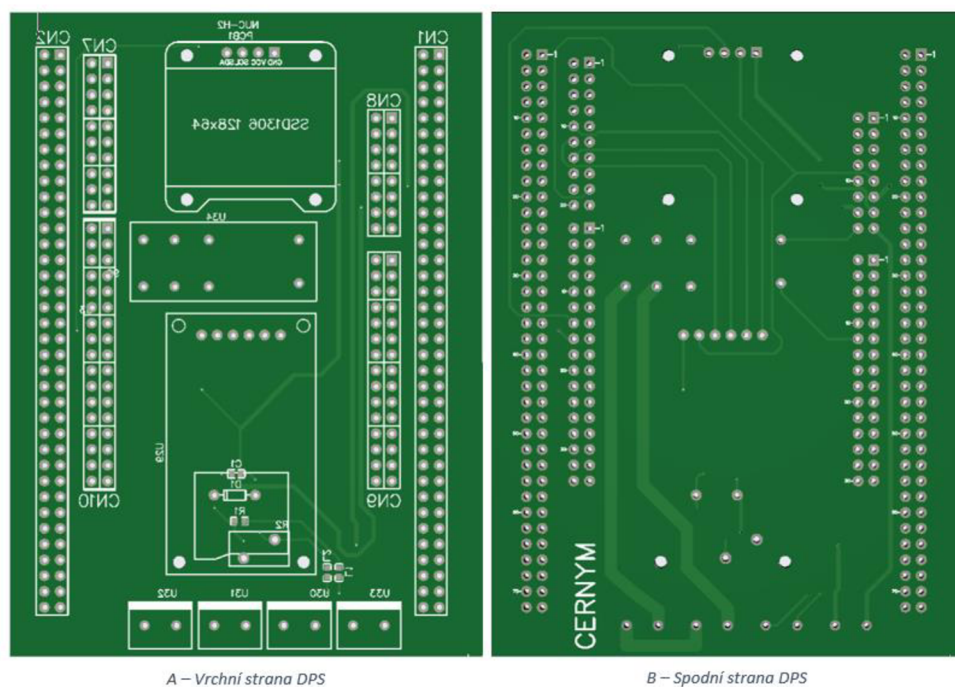
Cívky L1 a L2 jsou zde kvůli potlačení rušivých signálů, ale fungují také jako destruktivní součástka tzn. při velkém přepětí se přepálí a je nutno je vyměnit. Do série s cívkou L1 je zapojen odpor R1, na kterém v případě přepětí vznikne příslušný úbytek napětí. Při velkém přepětí dojde k jeho poškození. Zenerova dioda D1 a transil D2 chrání vstup před napětím opačné polaroty a také omezují případné přepětí.

2.8. Návrh externího modulu

Hlavní myšlenkou, jak by měl externí modul vypadat, byl princip zacvaknutí do volných dutinek pro piny, které jsou na Nucleu. To znamená, že nemusíme nic připojovat kabely apod. Jednoduše se modul připojí i odpojí. Do volných dutinek je možné připojovat další periferie či signály podle potřeby. Hlavní část DPS modulu tvoří OLED displej, na kterém se zobrazuje aktuální čas a spotřeba. Hned pod ním je SD karta, která měla být původně ze spodní strany DPS. To by bylo esteticky lepší, ale při nasazovacím způsobu by nebyl dostatek místa mezi deskou Nucleo a modulem. Na spodním okraji jsou čtyři svorkovnice sloužící:

- 1.) Pro přívod vodičů z vodoměru.
- 2.) Vstup pro 230 V, který má následně všechny měděné cesty dimenzované pro maximální napětí a proud, které relé zvládne bezproblémově sepnout.
- 3.) Vstup pro vodiče elektromagnetického ventilu.
- 4.) Externí napájení Nuclea.

Dosud bylo Nucleo napájeno přímo z microUSB z PC. V případě nasazení snímače v praxi je nutné řešit napájení pomocí nějakého externího, spínaného zdroje. Celý modul byl navržen nejprve v programu Eagle, ale následně jsem ho navrhl ještě v prostředí EasyEDA, které je zdarma a umožňuje vytvořit uspokojivý 3D model výrobku. DPS je možné vyrobit hned několika způsoby. Nejlepším způsobem je nechat si desku vyrobit nějakou společností zabývající se výrobou DPS. DPS by bylo možné vyrobit i na Technické univerzitě, ale zdálo se mi vhodnější využít společnost JBLPCB. U té je možné nechat desku vyrobit s všelijakými popisky na vrchní i dolní straně desky, maskou osazovacího plánu součástek na DPS a podobnými vymoženostmi. Na základě doložených gerber dat vám DPS vyrobí a dodají. Je potřeba jen osadit součástky a odzkoušet.



Obr. 18-Spodní a vrchní strana DPS externího modulu

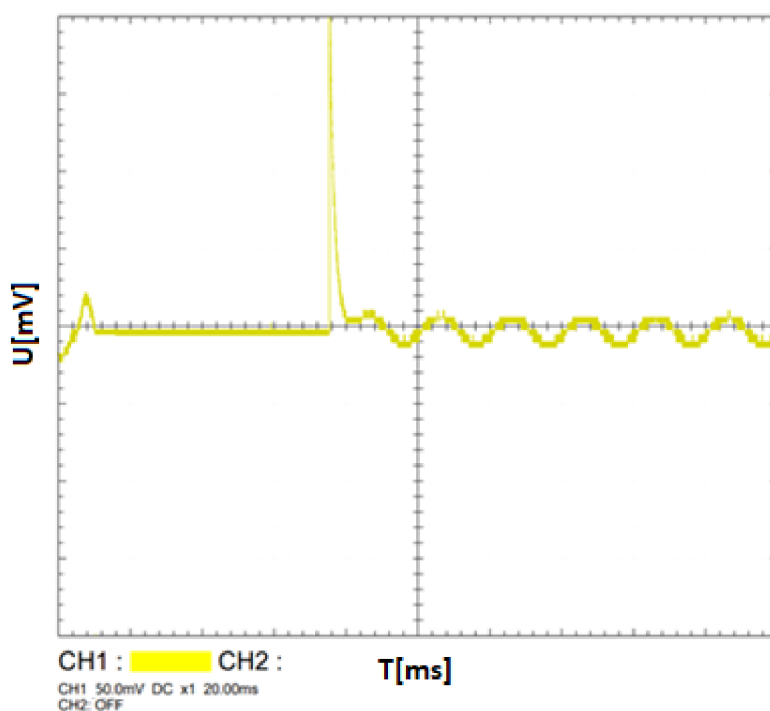
V příloze B 3D model externího modulu je detailnější model toho, jak deska reálně vypadá. Hotový modul viz obr. 20 po osazení součástek je plně funkční a implementuje v sobě všechny funkce, které byly popsány v kapitolách výše. Vznikl zde problém s malým místem pro manipulaci s SD kartou. Ten jsem nakonec vyřešil umístěním modulu SD karty na distanční sloupky. Modul je ve větší výšce, než bylo zamýšleno při návrhu, ale nijak to neovlivňuje jeho funkci.

3 Softwarové řešení

V této části a následujících kapitolách bude popsáno, jak programově zachytávat pulzy na pulzním výstupu S0 vodoměru a dále s touto informací pracovat.

3.1. Princip odečtu otáček

Podle uživatelského manuálu by měl senzor při nulové spotřebě vykazovat klidový signál obr. 19 pravá strana grafu a v momentě protečení jednoho litru vody by se měl objevit krátký pulz ve formě náběžné hrany viz obr. 19 levá strana grafu, která pomalu klesá až se signál zase ustálí na klidové hodnotě. Signál by měl podle dokumentace s naším nastavením senzoru trvat 35ms. Průběh byl zobrazen na přenosném osciloskopu Hantek 60002BE. Z obr. 19 je vidět, že signál je v ustáleném stavu a pak je zde náběžná hrana, která signalizuje protečení právě jednoho litru vody. Zachycený pulz na osciloskopu trvá okolo zmiňovaných 35ms, a to je standartní délka pro pulzní výstup S0. Tímto jsme si ověřili správnost měření podle dokumentace. Tyto náběžné hrany lze programově zaznamenávat a přičítat si zachycené pulzy do nějaké proměnné. Počet pulsů vynásobených konstantou K odpovídá množství spotřebované vody v litrech.



Obr. 19-Impulz při protečení jednoho litru

3.1.1 Programové řešení odečtu pulzů

Následující zdrojový kód 5 vysvětluje princip odečtu pulzů. Princip je detailněji vysvětlen na následujícím vývojovém diagramu viz příloha C Princip odečtu pulzů.

```
CybleState = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_3);
if(CybleState != LastCybleState)
{
    if(CybleState == GPIO_PIN_RESET)
    {
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
        CybleStateCounter++;
        printf("Number of pulses: %d\r\n", CybleStateCounter);
    }
    HAL_Delay(500);
}
LastCybleState = CybleState;
```

Zdrojový kód 5 - Programová realizace odečtu pulzů

3.2. Sériová komunikace

Zprovoznění sériové komunikace byla jedna z prvních věcí, kterou byla potřeba v programu udělat, abychom si mohli vypisovat informace o proměnných či nějaké chybové hlášky na PC. Informaci o aktuální spotřebě samozřejmě vypisujeme i na displej, ale je dobré mít k dispozici sériovou komunikaci s PC. Nucleo, se kterým pracujeme má k dispozici hned několik sériových portů. Je však potřeba dát si pozor, který pro co můžeme použít. Většinu z nich bychom použili pro komunikaci Nuclea s dalším kitem apod. V našem případě se nabízí využít UART3, protože piny jsou vyvedeny přímo na microUSB, kterým je Nucleo spojeno s PC. Přes microUSB se deska programuje a je z něj také napájena. Může však sloužit také ke komunikaci s PC. Je zde potřeba nastavit určité parametry, aby komunikace fungovala. Důležité parametry jsou baudrate, což je vlastně údaj o tom, kolik bitů může být posláno za časovou jednotku. Dále je nutno vybrat správný port, který byl počítačem přiřazen např. COM4. Pro práci se sériovou komunikací jsem zkusil využít různé programy k tomu určené. Nejvíce se mi osvědčil program Hercules, který je dostupný zdarma [27]. Z placených programů, které jsem měl možnost využít jen po dobu free trial verze jsem byl spokojený se Serial Port monitorem [28]. Nevýhodou u některých programů byla nemožnost otevřít nebo uzavřít komunikační port.

3.3. Hodiny reálného času

Na displeji jsem chtěl zobrazovat kromě údaje o spotřebě vody také aktuální čas. První myšlenka, jak toho dosáhnout, bylo využít nějaký modul reálného času, který má svůj vlastní krystal. Využít jsem chtěl modul DS3231, který komunikuje po I2C sběrnici a má i záložní baterii pro případ výpadku napájecího napětí. Tento modul jsem však nakonec nepoužil, protože Nucleo, se kterým pracuji má v sobě zabudovaný modul RTC s vlastním krystalem s frekvencí 32 KHz. Nevýhodou je však absence záložní baterie. Pokud nastane výpadek napájení, po dalším zapnutí napájení se tak čas nastaví opět do hodnot nastavených v programu. Řešením je využít záložní registr, který do sebe uloží poslední hodnotu času, než nastane výpadek.

3.4. Časovač

Čítače/Časovače se nejvíce používají k odpočítání nějakého časového úseku následovaného přerušením či nějakou jinou akcí. Zápis na SD kartu je třeba provádět v pravidelných intervalech. Zvolil jsem časový úsek jedné minuty, což myslím, plně postačuje. Každou minutu by tedy měl nastat zápis na SD kartu. K tomu jsem jako první řešení využil jeden z časovačů, kterými Nucleo disponuje. Po nastavení časovače v CubeMx jsem tímto způsobem dosáhl spolehlivého zapisování dat na SD kartu ve stále se opakujícím časovém úseku jedné minuty.

Dalším řešením, které jsem ověřil byl softwarový časovač, který je dostupný při použití FreeRTOSU. Výhodou je, že není potřeba nastavovat žádné parametry. Je potřeba jen zavolat v kódu nebo nějakém tasku funkci, která časovač spustí s daným časovým úsekem. Lze zde nastavit přímo jedna minuta. Tímto způsobem jsem dosáhl stejného výsledku jako u normálního časovače popsaného výše.

Posledním možným způsobem je využití RTC. Tento způsob ale není vhodný pro měření menších časových úseků. Využitelný je při měření hodinových, nebo delších intervalů. Pro finální řešení použité pro zapisování na SD kartu jsem využil nakonec softwarový časovač.

3.5. Webserver

Hlavní myšlenkou celé práce je vytvořit webovou stránku, na které se bude zobrazovat aktuální spotřeba vody a bude možno odsud ovládat přívod vody. Kit má v sobě zabudovaný čip LAN8720, což je nízkoenergetický transiever, který podporuje komunikaci s ethernetem přes standardní RMII rozhraní. To nám dává možnost komunikace přes ethernetové rozhraní. Nabízí se dvě možnosti, jak vytvořit na kitu webserver a obě se zásadně liší. První možností je vytvořit dynamickou stránku. Ta se bude automaticky obnovovat. IDE dokonce obsahuje příklad s webserverem vytvořeným tímto způsobem. Výhodou je, že se stránka sama dynamicky obnovuje. Zásadní nevýhodou je nutná hlubší znalost HTML a zkušenosti s tvorbou webových stránek. Z tohoto důvodu jsem přistoupil na druhý způsob a tím je SSI. Pomocí SSI lze celkem jednoduše odeslat nějaká data na web. Nevýhodou tohoto způsobu je, že se stránka sama neobnovuje a je nutno to zajistit jinak. Největší komplikací bylo nastavení všech parametrů tak, aby webserver správně fungoval. Každý kit, byť jsou skoro stejné, vyžaduje trochu jiné nastavení [25].

3.5.1 Konfigurace webserveru

Pro práci s ethernetem je používán TCP/IP stack, který implementuje všechny datové přenosy potřebné k realizaci samotného webserveru. Pro tyto účely byl zvolen velmi populární LwIP stack. Je to volně dostupný open source TCP/IP stack pro embedded systémy [26].

Problém při konfiguraci nastal hned při nastavování LwIP stacku, který nebylo možno vybrat pro používání v IDE. U většiny ostatních kitů je možné stack vybrat ihned bez nějakých předchozích nastavení. U kitu, který je použit zde, to tak není. Je nutno povolit LCache a DCache, jinak nelze povolit LwIP. Jak už bylo zmíněno v kapitole 3.5, tak transiever LAN8720 pracuje v režimu RMII. Toto je nutné dodržet i při konfiguraci parametrů. Některé kity pracují v režimu MII. Rozdílem je pouze způsob připojení periferie ethernetu. MII je připojeno pomocí 16 vodičů, zatímco RMII je trochu šetrnější a potřebuje jen 7 vodičů pro připojení. Obě varianty ale v principu fungují stejně.

Při konfiguraci se ještě objevuje parametr PHY, který slouží jako adresa určující, zda je použit externí modul pro ethernet např. W5500 nebo je použit transiever z Nuclea. Při nastavení špatné adresy pak webserver nebude fungovat.

Dalším problémem bylo nastavení IP adresy. IP adresu lze přiřadit zařízení buď staticky nebo dynamicky. Záleží na tom, jestli víme, v jakém rozsahu IP adres se můžeme pohybovat a které jsou volné. Pokud jsme v rodinném domě, je nejjednodušší možností využití DHCP a následně se v nastavení routeru podívat, jaké zařízení přibýlo v síti. Přes zjištěnou IP adresu se v prohlížeči dostaneme na webserver. DHCP je funkce schopná přidělit IP adresu, masku podsítě, výchozí bránu. V domácích podmínkách jsem využil DHCP a vyčetl si IP adresu. Bohužel v případě univerzitních kolejí jsem měl trochu problém. DHCP jsem zde nevyužil, protože jsem pak nemohl vyčíst IP adresu. Údajně by měla být v programu nějaká proměnná, do které se při používání DHCP načte přiřazená IP adresa. To se mi však nedařilo. Přešel jsem tedy na statické nastavení a nastavil vše ručně. Byl zde problém s tím, že jsem nevěděl, jak je síť na kolejích řešená a jaký je rozsah IP adres, které jsou volné k přiřazení. Zkoušel jsem postupně, které jsou volné, a nakonec jsem desce nějakou volnou přiřadil.

3.5.2 Spuštění webserveru

Desky Nucleo, Discovery apod. nemají vlastní file systém. To znamená, že nemáme kam uložit html soubor webové stránky. HTML kód stránky je nutno konvertovat na data v jazyce C. ST na to má přímo napsaný skript v Pearlu, který je dostupný na jejich stránkách. Pomocí skriptu se vytvoří zkonvertovaný soubor, který už lze přiřadit na příslušné místo v IDE. Při vypsání IP adresy Nuclea do prohlížeče se už dostaneme na webovou stránku.

3.6. Webový skriptovací jazyk

SSI je v podstatě jednoduchý skriptovací jazyk, který se provádí na straně serveru těsně před odesláním stránky uživateli. Výhodou je, že se do HTML kódu zapisuje pomocí komentáře viz příloha E Kód webové stránky. Pokud není SSI podporované, příkaz se jednoduše neprovede, protože se bere jako komentář. Pokud chceme SSI používat, je nutno ho v nastavení stacku LwIP povolit a místo stránky ve formátu html je nutno ji uložit jako shtml a až poté vyvolat C soubory pomocí Pearl scriptu.

SSI funguje tak, že při startu webserveru se vyvolá funkce `httpd_set_ssi_handler` a ta načte data do tag komentářů. Tato funkce se znovu již nevyvolá. Funkce se vyvolá pouze po spuštění programu nebo reloadu stránky. Proto je v HTML kódu stránky použit javascript, který pomocí čítače napočítá do daného času a pak vyvolá obnovení stránky. Tím se znovu stránka načte a spustí se tak funkce SSI a načtou se nová data.

3.7. CGI standard

Je protokol pro propojení externích aplikací s webovým serverem. To serveru umožňuje poslat požadavek od klienta externí aplikaci, která na tento požadavek vrátí příslušný výstup. CGI je zde využito kvůli práci s tlačítky pro ovládání relé. Při kliknutí na tlačítko se vytvoří v url adrese webové stránky parametr `relay=1`. Tento index si pak přečte funkce v programu Nuclea a přiřadí jí příslušný výstup.

3.8. *Webová stránka*

Pro psaní HTML stránek se používá značkovací jazyk, který je propojený s hypertextovými odkazy. Je to hlavní jazyk pro vytváření html stránek. Pro testování byla vytvořena jednoduchá html stránka s prostorem pro data a nějakými ovládacími tlačítky. Tlačítka se dá ovládat relé a tím tedy elektromagnetický ventil. Jak už bylo zmíněno v kapitolách výše, při použití SSI se stránka sama dynamicky neobnovuje.

V příloze E Kód webové stránky je html kód jednoduché webové stránky. Pro jednoduchost zde nejsou ani využity CSS styly, které slouží pro formátování a celkový návrh, jak bude stránka vypadat. Jelikož se stránka při použití SSI neobnovuje periodicky, je nutné to zajistit programově. Pro tento účel je zde jednoduchý javascript, který má za úkol odpočítat nějaký zvolený čas, po kterém se vyvolá reload stránky a data se načtou znovu. Dále jsou v kódu tagy, které slouží pro zapsání dat na určité místo na stránce viz kapitola 3.6.

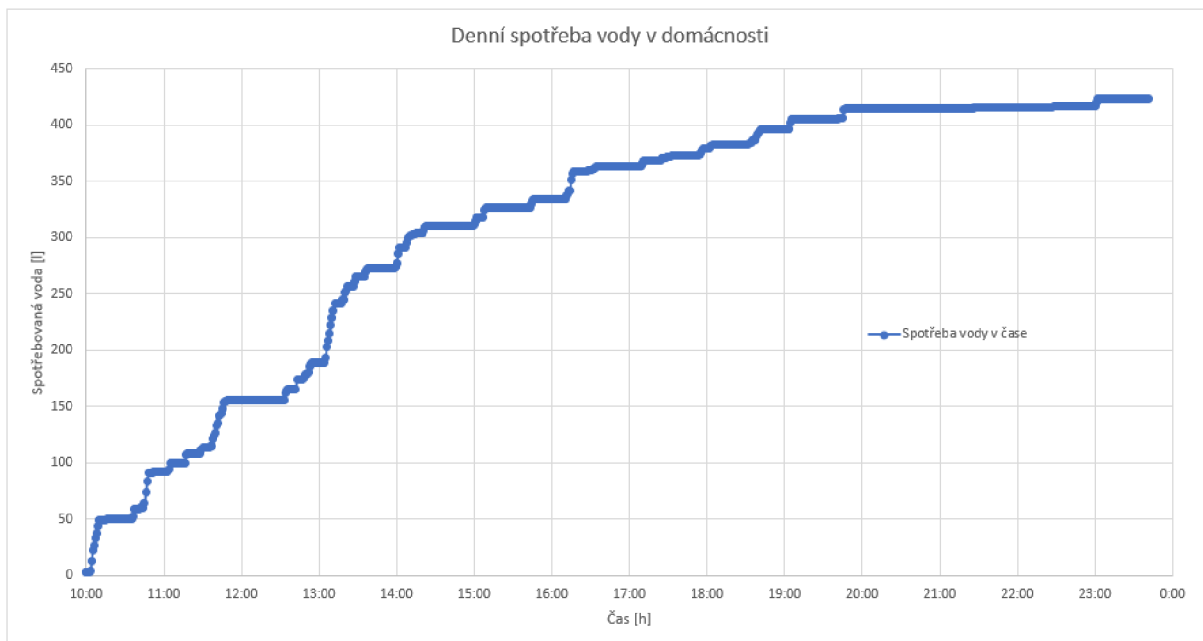
4 Dosažené výsledky a testování

Funkce modulu viz obr. 20 byla otestována nejprve na vodoměru Flodis od firmy Itron a poté na generátoru funkcí Siglent SDG 830. Pomocí generátoru propojeného s osciloskopem byla stanovena šířka frekvenčního pásma, ve kterém je zařízení schopné pulzy správně detekovat. Vodoměr Flodis má maximální průtok $Q_3 = 4 \text{ m}^3/\text{h}$, to odpovídá 1,1 l/s. Měřením bylo zjištěno frekvenční pásmo od 0 Hz do 20 Hz. Tento rozsah je naprosto dostačující pro vodoměry v bytových jednotkách. Větší průtok vody v těchto podmínkách by nemělo smysl řešit, protože taková situace nenastane. V průmyslu či podobném prostředí se používají jiné vodoměry, které mají průtok Q_3 definovaný nad $4 \text{ m}^3/\text{h}$. Pokud bychom chtěli použít naše zařízení v průmyslu s vodoměrem s vyšším průtokem $Q_3 > 4 \text{ m}^3/\text{h}$, tak by zařízení detekovalo pulzy správně do průtoku $K \cdot 20 \text{ l/s}$. Tabulka 3 obsahuje provozní parametry celého zařízení.

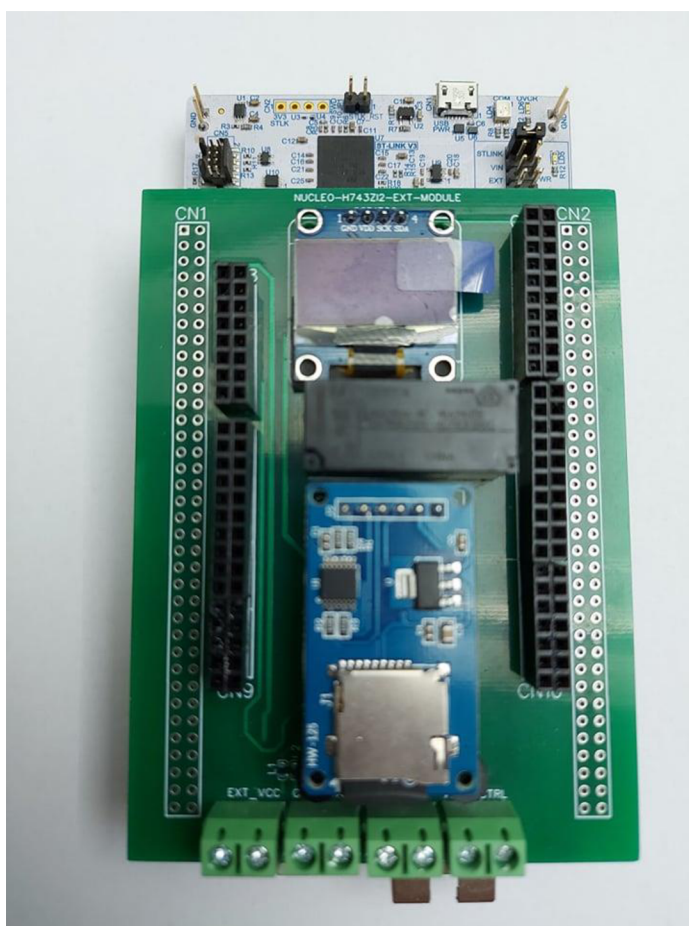
Veličina	Hodnota	Jednotka
Napájecí napětí	7 až 12	V
Maximální proud	800	mA
Polarizace signálu S0	ANO	-
Provozní teplota	od -10 do 55	°C
Délka kabelu	5	m
Koeficient K	1	-
Max frekvence	20	Hz
Min frekvence	>0	Hz
Proudová spotřeba	105	mA

Tabulka 3-Provozní parametry zařízení

V grafu 1 lze vidět příklad naměřené denní spotřeby v rodinném domě o 4 dospělých osobách. Lze vidět, že největší spotřeba vody nastává kolem odpoledních hodin. Naopak v pozdních večerních hodinách je spotřeba vody téměř nulová.



Graf 1-Spotřeba vody v domácnosti



Obr. 20-Hotový modul připojený k desce Nucleo

Závěr

Cílem této bakalářské práce bylo navrhnout způsob sběru dat v IoT a realizovat modul k desce STM32 pro pulzní výstup. V našem případě jde o sběr dat spotřeby vody v domácnosti. Jednou z důležitých částí je způsob, jak získávat data ze senzoru pro dálkové odečty Cyble Itron a následně je dále zpracovávat.

Práce se dělí na čtyři hlavní části. První část je věnována popisu dostupných vývojových kitů na trhu a jejich vzájemné porovnání. Shrnuje, jaké mají výhody a nevýhody a pro co jsou vhodné viz tabulka 1. Následně je vybrán nejvhodnější kit pro naši aplikaci.

V druhé části je popsáno hardwarové řešení externího rozšiřujícího modulu pro desku Nucleo viz příloha B 3D model externího modulu. Tento modul Nucleo rozšiřuje o možnost rychlého a jednoduchého zobrazování dat na OLED displej, zápis na SD kartu po SPI sběrnici a výkonová relé pro spínání elektromagnetického ventilu. Modul také obsahuje ochranu vstupů proti přepětí a rušení viz kapitola 2.7.3.

Třetí část je věnována softwarovému návrhu a způsobu, jak detekovat pulzy z pulzního výstupu S0 bytových vodoměrů a dále je převést pomocí koeficientu K na hodnotu spotřeby vody v litrech. Data jsou ukládány po jedné minutě na SD kartu viz kapitola 3.4. Dále se data odesílají na jednoduchý webserver a zobrazují se na OLED displeji.

Výsledkem práce je funkční rozšiřující modul k vývojovému kitu STM32 H743ZI2. Na modulu je výkonové relé použité ke spínání elektromagnetického ventilu, který ovládá přívod vody. S deskou Nucleo pak externí modul tvoří celek, který je schopen detekovat signál z pulzního výstupu S0 vodoměru a převést ho na hodnotu v litrech. Tuto hodnotu následně zobrazí na OLED displej a odešle na webserver, na kterém je vidět aktuální spotřeba vody. Z webserveru je také možné pomocí tlačítek ovládat relé, které je schopno spínat ventil. Na SD kartu jsou v pravidelných intervalech po 1 minutě zapisovány data o spotřebě vody s přiřazeným časem zápisu. Jelikož všechny pulzní výstupy S0 u vodoměrů fungují na stejném principu jen se liší koeficientem K viz kapitola 1.5, tak je zařízení univerzální a bude fungovat s jakýmkoliv bytovým vodoměrem.

V případě průmyslových vodoměrů, které mají stanovený vyšší průtok, jak je zmíněno v kapitole 4, bude modul fungovat správně do průtoku vody $K \cdot 20$ l/s.

Do budoucna je možno toto zařízení rozšířit o další funkce a vylepšení jako je třeba nastavení určitých režimů např. všední den, víkend a dovolená. Každý režim by měl definovanou jinou denní hodnotu spotřeby vody. V režimu dovolená by měla být spotřeba téměř nulová. Také by bylo možno přidat funkci na automatické zavírání ventilu místo aktuálního manuálního zavírání z webserveru podle toho, jak by se lišil poslední průtok od aktuálního. Tzn. v případě zvýšení spotřeby vody např. o více než 50 l by se mohl ventil automaticky zavřít. Při použití dynamické webové stránky by se dala vylepšit její grafická úprava a také by se dalo zlepšit její rozhraní, což využití SSI příliš neumožňuje.

Literatura

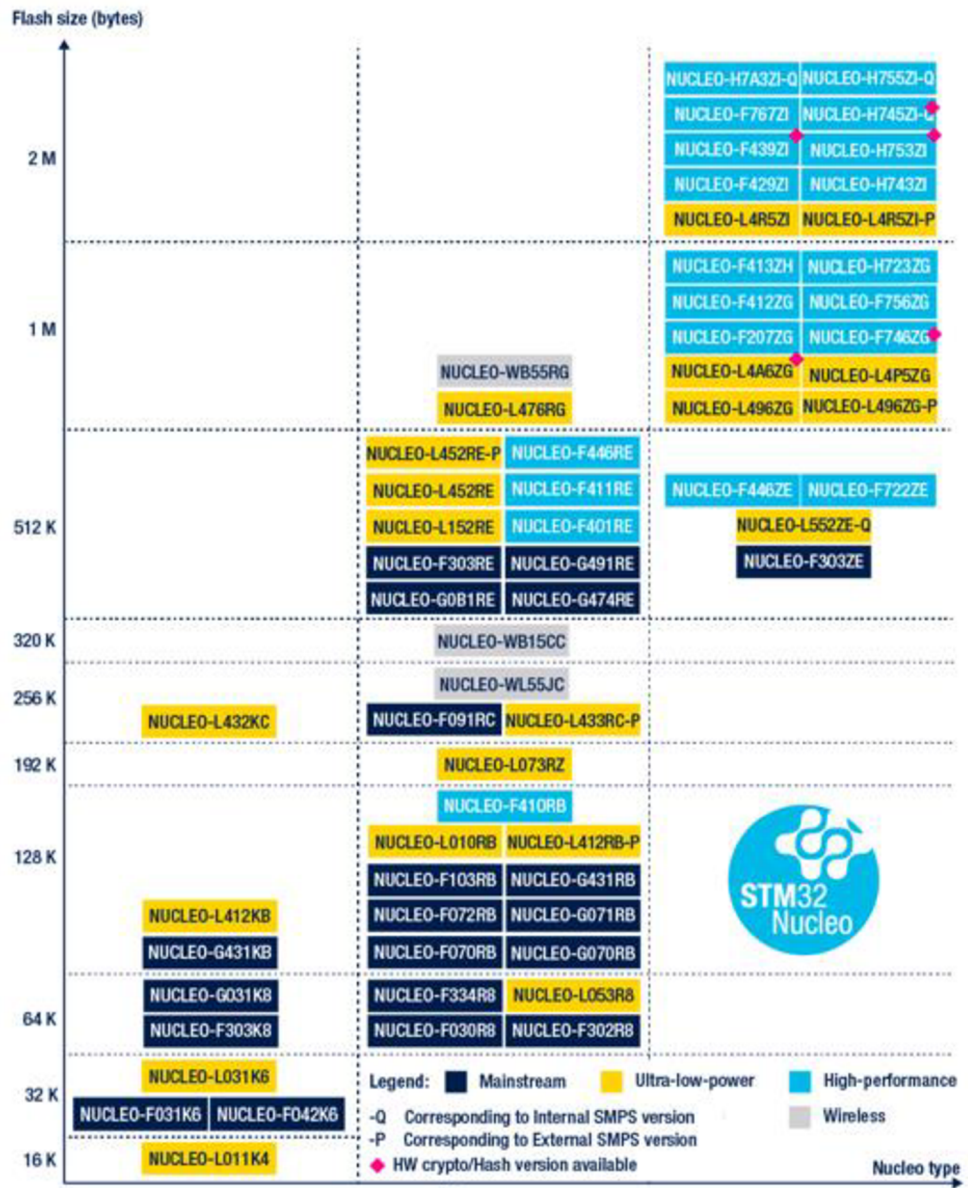
- [1] ARM. *Cs.wikipedia.org* [online]. Wikipedie: Otevřená encyklopedie., 2022 [cit. 2022-05-01]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=ARM&oldid=21178900>
- [2] STM32 Nucleo Boards. *St.com* [online]. Switzerland: STMicroelectronics, 2022 [cit. 2022-03-04]. Dostupné z: <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>
- [3] NUCLEO-H743ZI. *St.com* [online]. Switzerland: STMicroelectronics, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.st.com/en/evaluation-tools/nucleo-h743zi.html>
- [4] ST-LINK/V2. *St.com* [online]. Switzerland: STMicroelectronics, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.st.com/en/development-tools/st-link-v2.html>
- [5] STM32 IDEs. *St.com* [online]. Switzerland: STMicroelectronics, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.st.com/en/development-tools/stm32-ides.html>
- [6] IAR Embedded Workbench for Arm. *Iar.com* [online]. Sweden: IAR Systems, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.iar.com/ewarm>
- [7] STM32CubeIDE, STM32CubeMX, STM32 Finder: 3 Tools, 1 Unique Ecosystem to Empower Creators. *Blog.st.com* [online]. Switzerland: STMicroelectronics, 2022 [cit. 2022-05-01]. Dostupné z: <https://blog.st.com/stm32cubeide-free-ide/>
- [8] System Workbench for STM32: free IDE on Windows, Linux and OS X. *St.com* [online]. Switzerland: STMicroelectronics, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.st.com/en/development-tools/sw4stm32.html>
- [9] A powerful eclipse-based C/C++ integrated development tool for your STM32 projects. *St.com* [online]. Switzerland: STMicroelectronics, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.st.com/en/development-tools/truestudio.html>
- [10] Vodoměry pro měření a rozúčtování spotřeby vody: BYTOVÉ VODOMĚRY. *Mereni-tepla.cz* [online]. Praha: net.pointers.cz, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.mereni-tepla.cz/presne-neovlivnitelne-vodomery>
- [11] Vodoměr. *Cs.wikipedia.org* [online]. Wikipedie: Otevřená encyklopedie., 2022 [cit. 2022-05-02]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Vodom%C4%9Br&oldid=18561218>
- [12] Vodoměr. *Cs.wikipedia.org* [online]. Wikipedie: Otevřená encyklopedie., 2022 [cit. 2022-05-02]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Vodom%C4%9Br&oldid=18561218>

- [13] Domovní vodoměr Itron, Flodis TVM, DN 20, 30° C, PN 16, 190 mm. *Tzbtechnika.cz* [online]. Praha: TZB technika .cz, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.tzbtechnika.cz/domovni-vodomery/itron-domovni-vodomer-flodis-tvm--dn-20--30-c--pn-16--190-mm/>
- [14] Měření a regulace: Vodoměry - 1. část. *Topin.cz* [online]. Praha: Topin Media, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.topin.cz/clanky/vodomery-1-cast-detail-8336>
- [15] ITRON CYBLE NF. *Kapka-vodomery.cz* [online]. Čelákovice: Pub-Me webdesign, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.kapka-vodomery.cz/e-shop/vodomery/domovni-vodomery/itron-prislusenstvi/itron-cyble-nf>
- [16] LCD displej. *Pic16fxx.sweb.cz* [online]. DOVEDA, 2022 [cit. 2022-05-01]. Dostupné z: <http://pic16fxx.sweb.cz/programovani/lcd/index.html>
- [17] Interface LCD 16×2 with STM32 without I2C. *Controllerstech.com* [online]. India: Controllerstech, 2022 [cit. 2022-05-01]. Dostupné z: <https://controllerstech.com/interface-lcd-16x2-with-stm32-without-i2c/>
- [18] I2C (TWI) – sériová komunikace po dvou vodičích s adresací. *Zavavov.cz* [online]. ZAVAVOV, 2022 [cit. 2022-05-01]. Dostupné z: <http://www.zavavov.cz/cz/elektrotechnika/komunikacni-sbornice/68-i2c-twi-seriova-komunikace-po-dvou-vodicich-s-adresaci/>
- [19] OLED display using I2C with STM32. *Controllerstech.com* [online]. India: Controllerstech, 2022 [cit. 2022-05-01]. Dostupné z: <https://controllerstech.com/oled-display-using-i2c-stm32/>
- [20] Čtečka microSD karet. *Dratek.cz* [online]. Havlíčkův Brod: ECLIPSE s.r.o, 2022 [cit. 2022-05-01]. Dostupné z: https://dratek.cz/arduino/993-ctecka-microsd-karet.html?gclid=CjwKCAiAJoeRBhAJEiwAYY3nDLnsu-RG_tVh-yT_1WMC04ZgUPrXGmiAhXIKeEqVXdKRom-EGQCtiRoCPnIQAvD_BwE
- [21] Relé 3V Songle SRD-03VDC-SL-C. *Pajenicko.cz* [online]. Zbožíčko: Pájeničko s.r.o, 2022 [cit. 2022-05-01]. Dostupné z: <https://pajenicko.cz/rele-3v-songle-srd-03vdc-sl-c-250vac30vdc-10a>
- [22] FTR-K1AK005T FUJITSU. *Tme.eu* [online]. Ostrava: TME, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.tme.eu/cz/details/ftr-k1ak005t/elektromagneticka-rele-miniaturni/fujitsu/>
- [23] Ošetření vstupů Arduina a jiných IO. *Mylms.cz* [online]. 2022 [cit. 2022-05-01]. Dostupné z: <https://www.mylms.cz/osetreni-vstupu-arduina-a-jinych-io/>

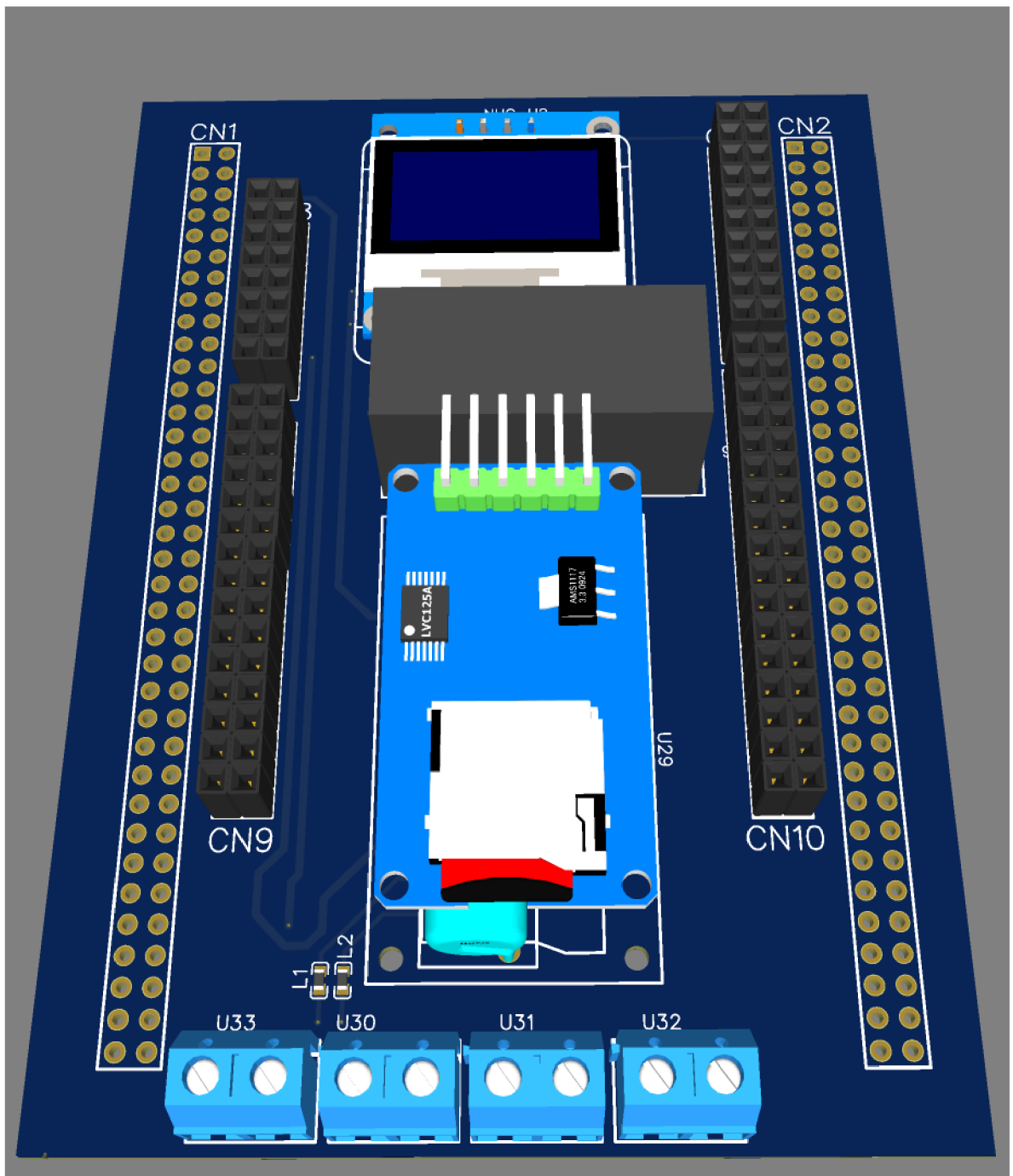
- [24] Stíněný kabel. *Cs.wikipedia.org* [online]. Wikipedie: Otevřená encyklopedie., 2022 [cit. 2022-05-01]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=St%C3%ADn%C4%9Bn%C3%BD_kabel&oldid=20352279
- [25] MII and RMI interfaces of STM32 network. *Developpaper.com* [online]. Develop Paper, 2022 [cit. 2022-05-01]. Dostupné z: <https://developpaper.com/mii-and-rmii-interfaces-of-stm32-network/>
- [26] LwIP. *En.wikipedia.org* [online]. Wikipedie: Otevřená encyklopedie., 2022 [cit. 2022-05-01]. Dostupné z: <https://en.wikipedia.org/wiki/LwIP>
- [27] Hercules. *Hw-group.com* [online]. [cit. 2022-03-04]. Dostupné z: <https://www.hw-group.com/software/hercules-setup-utility>
- [28] Serial Port Monitor Track and analyze the activity of your COM ports. *Com-port-monitoring.com* [online]. Alexandria: Electronic Team, 2022 [cit. 2022-05-01]. Dostupné z: <https://www.com-port-monitoring.com>

Přílohy

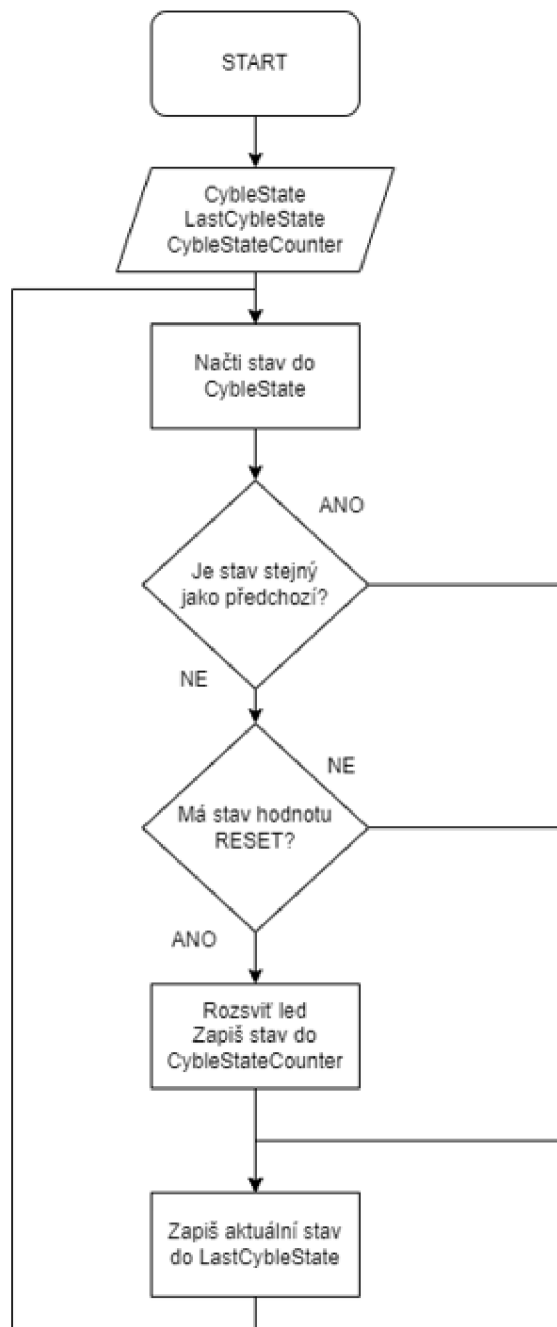
A Rozdělení desek STM32



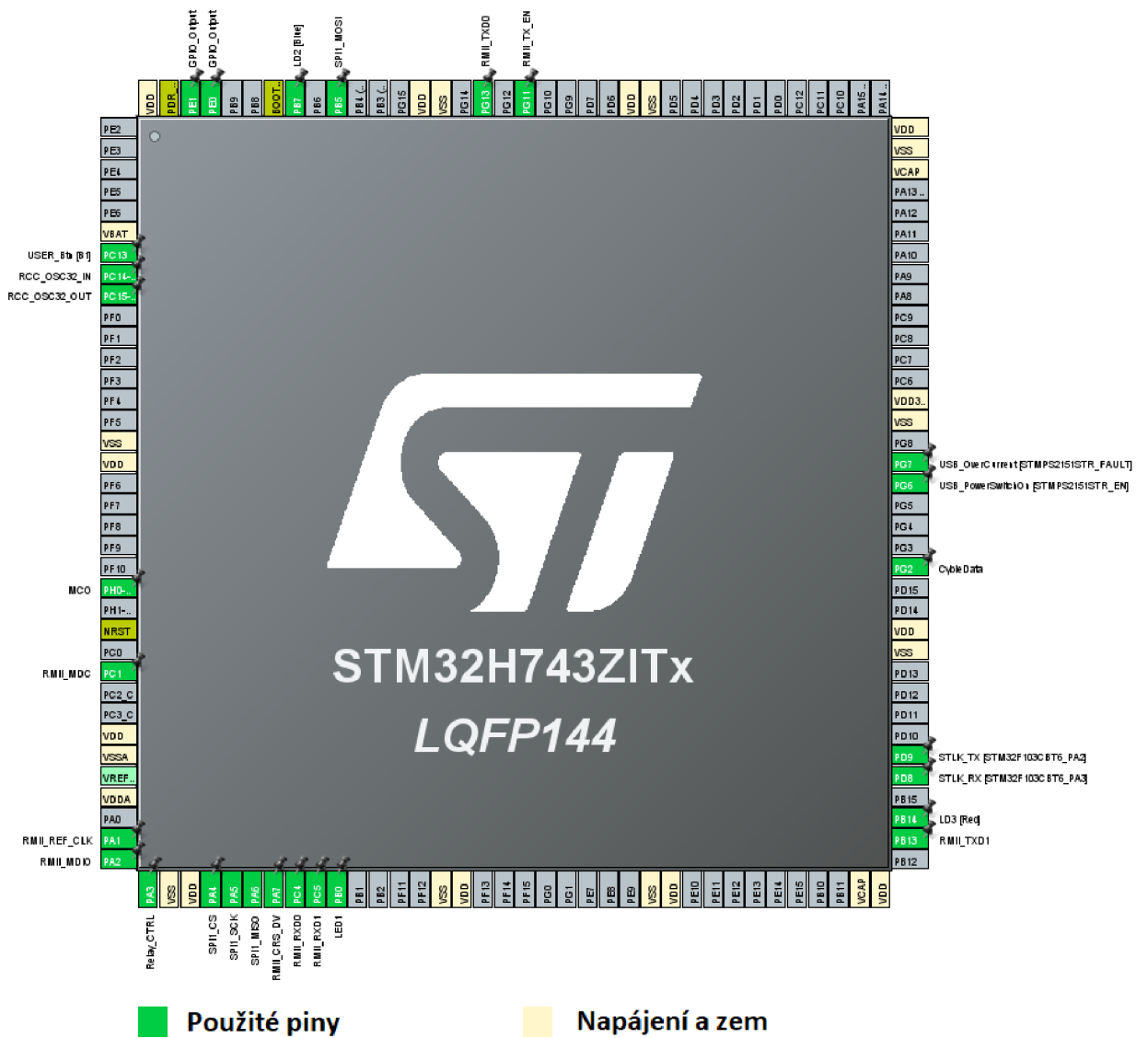
B 3D model externího modulu



C Princip odečtu pulzů



D Popis pinů procesoru



E Kód webové stránky

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>STM32 httpd server</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <script type="text/javascript">
9     let counter = 1;
10    setInterval(() => {
11      document.querySelector('h1').innerText = counter;
12      counter++;
13      if(counter > 5) location.reload();
14    }, 1000);
15  </script>
16
17  <script type="text/javascript">
18    document.getElementsByName("ipaddr").value = /*#tag1*/;
19  </script>
20
21 <h1 style="text-align: center">Reload</h1>
22
23 <p>Ovladani LED</p>
24 <form method="get", action="/leds.cgi">
25   <!-- <input type="checkbox" name="led" value="1">LED1<br> -->
26   <!-- <input type="checkbox" name="led" value="2">LED2<br> -->
27   <input type="submit" name="led" value="1">OPEN<br>
28   <input type="submit" name="led" value="2">CLOSE<br>
29   <br>
30   <input onload="myFunction" value="" name="ipaddr">
31   <p>text for tag2: <!--#tag2--></p>
32   <!-- <input type="submit" value="send"> -->
33 </form>
34 </body>
35 </html>
```