



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Zařízení pro sledování domácnosti pomocí vestavěných systémů využívající GSM síť k přenosu dat
Student:	Bc. Vojtěch Procházka
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Studijní obor:	Návrh a programování vestavných systémů
Katedra:	Katedra číslicového návrhu
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

- 1) Prozkoumejte existující řešení.
- 2) Navrhněte a zrealizujte vlastní zařízení umožňující sledování domácnosti.
- 3) Zaměřte se zejména na možnost přenášet data mezi zařízením a vzdáleným serverem s co možná nejnižší spotřebou energie.
- 4) Zařízení navrhněte tak, aby bylo možné přenášet větší objem dat.
- 5) Pro demonstraci správné funkce připojte k zařízení kameru a v určitých intervalech přenášejte statické fotografie.
- 6) Vytvořte jednoduchý server pro příjem dat ze vzdáleného zařízení.
- 7) Výsledné řešení otestujte a proveďte analýzu spotřeby v závislosti na množství přenášených dat.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Hana Kubátová, CSc.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 11. února 2020



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Diplomová práce

**Zařízení pro sledování domácnosti pomocí
vestavěných systémů využívající GSM síť
k přenosu dat**

Bc. Vojtěch Procházka

Katedra číslicového návrhu

Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

27. května 2020

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Pavlu Kubalíkovi, Ph.D. za cenné rady a veškerou pomoc, kterou při tvorbě diplomové práce poskytl. Velice si cením přístupu vedení.

Zároveň bych rád poděkoval všem, kteří mě podpořili při tvorbě a korektuře.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisu. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisu, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programu, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 27. května 2020

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2020 Vojtěch Procházka. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Procházka, Vojtěch. *Zařízení pro sledování domácnosti pomocí vestavěných systémů využívající GSM síť k přenosu dat.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Diplomová práce se zabývá návrhem a realizací sledovacího zařízení pro internet věcí (IoT), které pomocí vestavěných systémů využívá GSM síť. Vytvořené sledovací zařízení odesílá pořizené fotografie na vzdálený server přes existující mobilní síť s běžnou SIM kartou. Samotné sledovací zařízení se skládá ze čtyř hlavních komponent: řídicí Arduino, adaptér na SD kartu, kamera a GSM modul.

Nedílnou součástí práce je měření vytvořeného IoT zařízení při výkonu různých úkolů. K měření bylo použito druhé Arduino a byla měřena spotřeba energie, propustnost a latence. Metodologie měření ohledně získání jednotlivých naměřených hodnot je detailně popsána a jsou vytvořeny závěry ohledně využitelnosti tohoto zařízení v IoT. V práci jsou taktéž uvedena zjištění ohledně výdrže baterie pro napájení sledovacího zařízení. Pro daný GSM modul je prozkoumána využitelnost rychlosti LTE sítě a je vypočítáno, jaké celkové zrychlení odesílání souborů by bylo dosaženo při použití této rychlosti.

Klíčová slova GSM, vestavěné systémy, Arduino, mobilní síť, bezdrátový přenos dat, spotřeba energie, baterie, propustnost, LTE, IoT, low power, latence, kamera

Abstract

This diploma thesis is centered around the design and realization of a surveillance device for the internet of things (IoT), which utilizes the GSM network using embedded systems. The created surveillance device sends taken photos to a remote server via the existing cellular network with a common SIM card. The surveillance device consists of four main components: driving Arduino, SD card adapter, camera and GSM module.

Measuring this IoT device when it is performing various tasks is an integral part of the work. A second Arduino was used for measuring. Power consumption, throughput and latency were measured. Measurement methodology of acquiring individual measured values is described in detail and findings regarding the usability of this device in IoT are shown. Battery lifespan for this device is also discussed. The usability of LTE speeds for the given device is explored and a speedup ratio for sending files while using this speed is calculated.

Keywords GSM, embedded systems, Arduino, cellular network, wireless data transfer, power consumption, battery, throughput, LTE, IoT, low power, latency, camera

Obsah

Úvod	1
1 Cíle práce	3
2 Existující řešení	5
3 Teoretický úvod	9
3.1 GSM síť	9
3.2 Sada protokolů TCP/IP	9
3.3 Vývojová prostředí	11
3.3.1 ArduinoIDE	11
3.3.2 Microsoft Visual Studio	11
3.3.3 Wolfram Mathematica	12
4 Analýza a návrh	13
4.1 Výběr platformy	13
4.1.1 Vybrané komponenty	14
4.2 Programové řešení	17
4.2.1 Komunikace mezi programy	18
4.2.2 Sledovací zařízení	19
4.2.3 Server	20
4.2.4 Měřící program	20
4.3 Metodologie měření	22
4.3.1 Spotřeba energie	22
4.3.2 Propustnost	23
4.3.3 Latence	24
5 Realizace	27
5.1 Sledovací zařízení	27
5.1.1 Hlavní překážky realizace	29

5.2	Server	33
5.3	Měřicí program	34
5.4	Zpracování naměřených hodnot	36
6	Testování	37
6.1	Testování funkcionality	37
6.2	Naměřené hodnoty	39
6.2.1	Propustnost	40
6.2.2	Latence	41
6.2.3	Spotřeba energie	41
6.3	Hlavní omezující faktory zařízení	46
	Závěr	51
	Literatura	53
	A Seznam použitých zkratk	57
	B Obsah příloženého CD	59

Seznam obrázků

2.1	Zapojení sledovacího zařízení z reference [5].	6
2.2	Zapojení monitorovacího zařízení z reference [7].	7
3.1	ISO/OSI vrstvy TCP/IP protokolu	10
3.2	Připojování hlaviček při odesílání dat z aplikace.	10
4.1	Arduino MEGA 2560	15
4.2	SIM7000E Arduino NB-IoT/LTE/GPRS/GPS Expansion Shield DFR057213	15
4.3	Catalex MicroSD Card Adapter	16
4.4	Arducam 5MP Plus OV5642 Mini Module	16
4.5	SIM7000E	17
4.6	Diagram zapojení jednotlivých komponent	18
4.7	Zapojení při měření proudu určité komponenty.	21
5.1	Architektura sledovacího zařízení	28
5.2	Fotka realizace pro měření komponent.	29
5.3	Rozpojený pin 13 u čipu LVC125A.	32
5.4	Modifikovaný adaptér, pohled z boku	32
5.5	Modifikovaný adaptér, pohled přímo	33
6.1	Procházející proud při odesílání 1 B souboru z GSM modulu.	43
6.2	Zachycený snímek s rozlišením 320 x 240	45
6.3	Proud procházející kamerou při zachycení snímku a zápisu na SD.	47
6.4	Proud procházející adaptérem na SD kartu při čtení a zápisu 10 kB souboru.	48
6.5	Proud procházející adaptérem na SD kartu při čtení a zápisu 100 kB souboru.	49

Seznam tabulek

3.1	Výběr GSM standardů	10
5.1	Propojení řídicího Arduina a GSM modulu	28
5.2	Propojení řídicího Arduina a adaptéru na SD kartu	28
5.3	Propojení řídicího Arduina a kamery	29
5.4	Propojení řídicího Arduina a měřícího Arduina	35
6.1	Naměřené doby přenosu jednotlivých částí	40
6.2	Naměřené spotřeby energie pro odesílání různých souborů	42
6.3	Naměřené spotřeby energie adaptéru a kamery	44

Úvod

Vestavěné systémy jsou v dnešní době populární a široce používané. Vestavěný systém je systém s mikroprocesorem navrženým pro vykonávání specifické funkce nebo více funkcí. Na rozdíl od počítače není vestavěný systém navržený tak, aby jeho software byl měněn koncovým uživatelem [1]. Obecně řečeno, vestavěné systémy něco řídí nebo sbírají data.

Shromážděná data jsou poté buď přímo použita tímto vestavěným systémem, nebo je potřeba data někam odeslat. V tomto bodě začíná být relevantní termín Internet věcí (Internet of Things, IoT). IoT je vestavěný systém se schopností přenášet data přes určitou síť [2]. Existuje mnoho různých sítí, například WiFi a mobilní síť. Hlavní výhodou mobilní sítě je pokrytí rozsáhlé oblasti. S vestavěnými systémy posílající data přes mobilní síť souvisí pojem "cellular IoT" - mobilní internet věcí.

Cellular IoT nabízí alternativu k sítím s nízkou spotřebou energie a rozsáhlým pokrytím [3]. Je důležité zmínit dva nejběžnější standardy pro cellular IoT, což jsou LTE-M (Long-Term Evolution) a NB-IoT (Narrowband IoT) [4]. Oba standardy fungují s běžnou GSM sítí, poskytovanou telefonními operátory, avšak jejich specifikace jsou rozdílné. Požadavek na nízkou spotřebu energie je důležitý, pokud je zařízení umístěno v oblasti bez elektřiny (např. vzdálený dohled, vzdálený monitorovací systém pro pacienty). NB-IoT má nízkou spotřebu energie, ale hlavní nevýhodou je rychlost přenosu. Na druhou stranu LTE-M má vyšší rychlost přenosu, ovšem je to za cenu zvýšené spotřeby energie.

Jedním z cílů této práce je navrhnout, zrealizovat a popsat zařízení, které zachycuje fotografie a používá mobilní síť pro posílání těchto fotografií na vzdálený server. Zároveň je snaha, aby zařízení mělo co nejnižší spotřebu energie.

Dalším z cílů je prozkoumání existujících řešení vestavěných systémů používajících mobilní síť ke komunikaci. Obzvláště zajímavá jsou existující řešení, která uvádějí spotřebu energie, pro možné porovnání se zařízením, jež je prezentované v této práci.

ÚVOD

Součástí práce jsou uvedené naměřené hodnoty spotřeby energie, propustnosti a latence realizovaného sledovacího zařízení. Zároveň je detailně popsána metodologie získání těchto naměřených hodnot.

Cíle práce

Diplomová práce si pokládá jako hlavní cíle:

1. Navrhnout a realizovat vlastní zařízení umožňující sledování domácnosti.
2. Prozkoumat existující řešení pro možné porovnání naměřených hodnot.
3. Zaměřeni se zejména na možnost přenášet data mezi zařízením a vzdáleným serverem s dbáním na nízkou spotřebu energie.
4. Zařízení by mělo být schopné přenášet větší objem dat.
5. Připojit k zařízení kameru a v určitých intervalech přenášet statické fotografie.
6. Vytvořit jednoduchý server pro příjem dat ze vzdáleného zařízení.
7. Výsledné řešení otestovat.
8. Provést analýzu spotřeby energie v závislosti na množství přenášených dat.

Existující řešení

Problém bezdrátového posílání dat z vestavěných systémů přes GSM síť za účelem sledování domácnosti je již řešený v různých článcích. Vybrané články jsou popsány v této sekci.

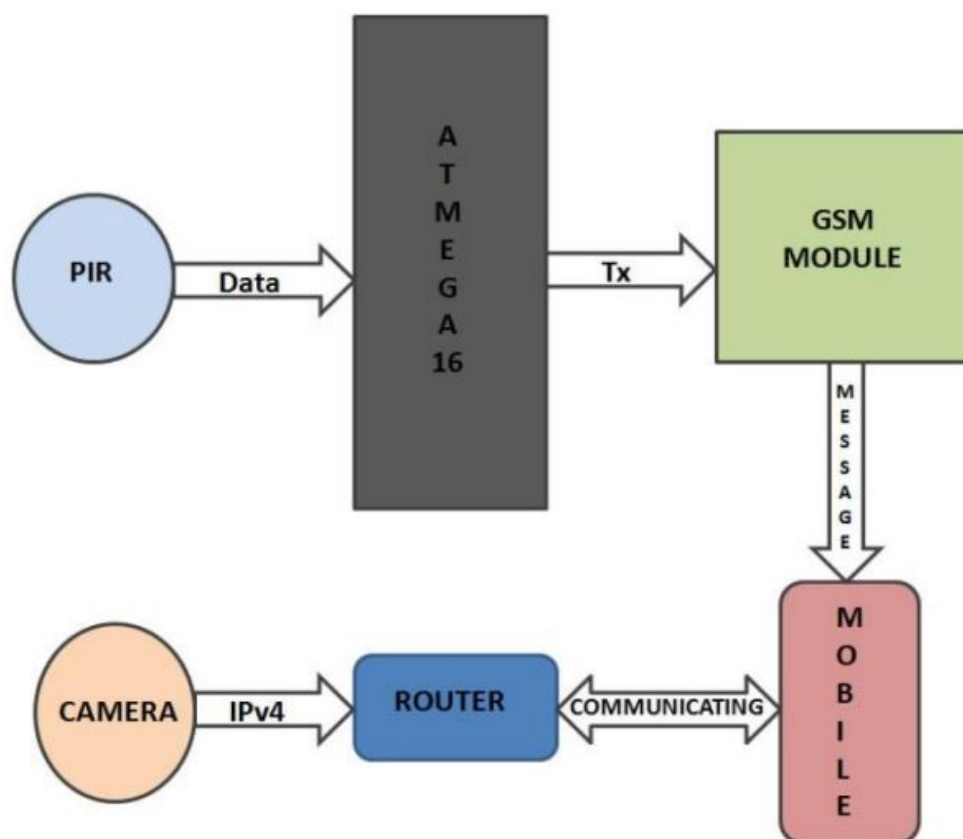
Práce [5] navrhla a realizovala zařízení, které sleduje domácnost pomocí infračerveného senzoru a kamery. Pokud infračervený port zachytí pohyb, pošle se přes GSM síť zpráva na mobilní telefon s adresou, na níž se může uživatel kouknout na živý přenos z kamery. Tato práce bohužel neudává žádné hodnoty spotřeby energie.

Zapojení komponent citované práce je možné vidět na obrázku 2.1. PIR je infračervený senzor. Ten je využíván čipem ATmega16, který komunikuje s GSM modulem. GSM modul odesílá zprávu na mobil. Ve zprávě je internetový odkaz na stránku, na které se může uživatel připojit k přenosu z kamery.

Dříve vytvořená práce [6] zkoumala spotřebu energie u mobilních telefonů při přenosu dat pomocí 3G, GSM a WiFi. Autoři shledali, že WiFi má mnohem úspornější spotřebu energie oproti 3G a GSM, ale je to za cenu nižšího pokrytí. Spotřebu energie v práci měří pomocí aplikace Nokia Energy Profiler, která běží během přenosu dat na pozadí telefonu Nokia N95. Udávají spotřebu energie 4.0 Ws pro stáhnutí 10 kB souboru a 5.0 Ws pro stáhnutí 50 kB souboru pomocí GSM sítě.

Studie [7] prezentuje monitorovací systém reálného času pro kontrolu teploty pacientů přijatých do nemocnice. Používá GSM modem SmartG100 a mikrokontrolér PIC16F877A. Algoritmus spánku/probuzení je implementovaný v mikrokontroléru pro snížení celkové spotřeby energie. Tento algoritmus spánku/probuzení redukuje procházející proud z 300 mA na 2.695 mA. Zařízení je aktivní pouze 221 ms a ve stavu spánku mezi jednotlivými aktivními fázemi je 60 s. Tento GSM modem má prezentovaný procházející proud 1.6 mA ve stavu spánku.

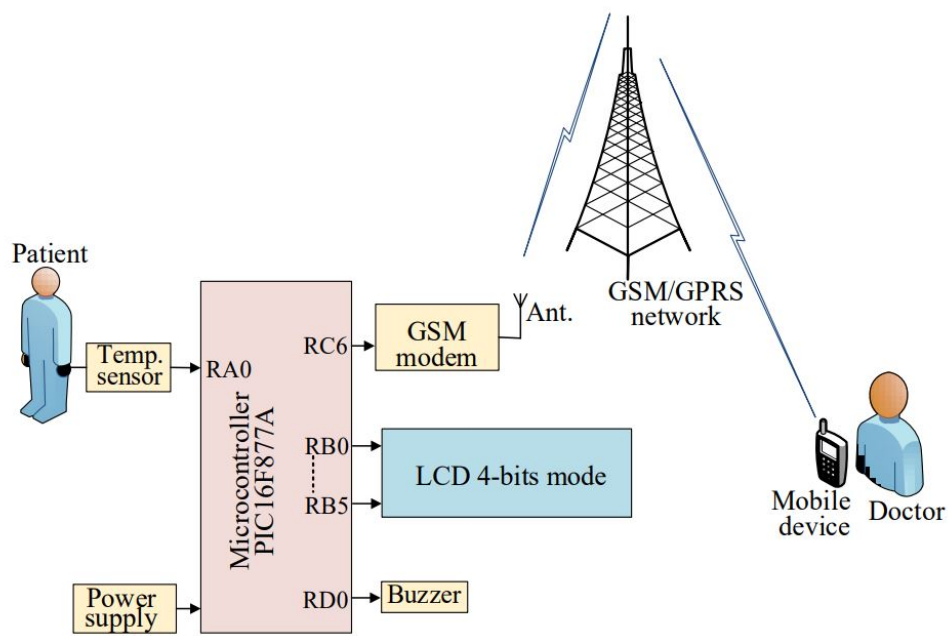
Realizace této studie je zobrazena na obrázku 2.2. Senzor teploty monitoruje pacientovu teplotu a ta je následně zpracována mikrokontrolérem PIC16F877A. Mikrokontrolér následně odešle pomocí GSM modemu přes GSM



Obrázek 2.1: Zapojení sledovacího zařízení z reference [5].

sít tuto informaci příslušnému lékaři. V diagramu jsou zobrazeny i další periferie, jako například LCD (Liquid-crystal display) a bzučák.

Existují další návrhy používající GSM modem s mikrokontrolérem, například pro posílání dat přes SMS (Short Message Service) [8], nebo pro posílání naměřených parametrů (např. napětí, proud, frekvence) z "inteligentní sítě" [9]. Jiné studie zkoumají modely spotřeby energie, které pomáhají identifikovat hlavní spotřebitele energie při používání chytrého telefonu s LTE [10]. Další studie se zaměřuje na optimalizace pro šetření energie pro mobilní síť [11], jež navrhuje další nastavitelné řízení spotřeby pro vylepšený diskontinuální příjem LTE. Studie [12] prezentuje schéma rovnováhy mezi spotřebou energie a latencí.



Obrázek 2.2: Zapojení monitorovacího zařízení z referencie [7].

Teoretický úvod

Kapitola obsahuje obecný popis oblastí, které bylo potřeba nastudovat pro úspěšné vypracování práce. Popsaná témata zahrnují pouze klíčové oblasti, neboť k úspěšnému splnění cílů práce bylo potřeba nastudovat větší množství materiálů, jejichž popis již není předmětem diplomové práce.

3.1 GSM síť

Global System for Mobile Communications (GSM) je nejpobulárnějším standardem pro mobilní komunikaci [13]. GSM síť je systém používaný většinou mobilních sítí na světě [14]. Díky velké popularitě a globálnímu užití nabízí GSM síť vysoké pokrytí. Pojem GSM síť je zpravidla používán i pro označení mobilní sítě, a nejen jako označení protokolu pro druhou generaci (2G).

Pro připojení k GSM síti jsou využívány Subscriber Identity Module (SIM) karty. SIM karty se vkládají do zařízení, a ta se připojují do GSM sítě. Na SIM kartě jsou uložena data k identifikaci uživatele, který ji používá [15]. SIM karty jsou poskytovány operátory, a to jak při koupi telefonu, tak samostatně. SIM karty se dají vložit také do vestavěných systémů pro připojení ke GSM síti. SIM karty mohou být zabezpečeny PINem (Personal Identification Number), tedy číslem, které je potřeba zadat k odemčení SIM karty.

Na GSM síti běží současně více standardů, každý na jiné frekvenci. Pro koncového uživatele je toto nejvíc poznat v rychlosti datových přenosů jednotlivých standardů. Pro tuto práci jsou významné standardy popsány v tabulce 3.1 [16].

3.2 Sada protokolů TCP/IP

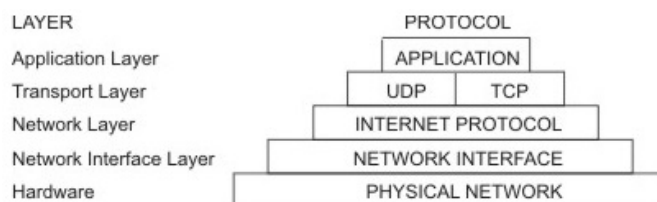
Protokoly obecně obsahují množiny pravidel pro formáty zpráv a postupy pro komunikaci [17]. TCP/IP sada protokolů obsahuje několik ISO/OSI vrstev, od fyzické vrstvy až po aplikační vrstvu viz obrázek 3.1 [17]. Ke každé zprávě je

3. TEORETICKÝ ÚVOD

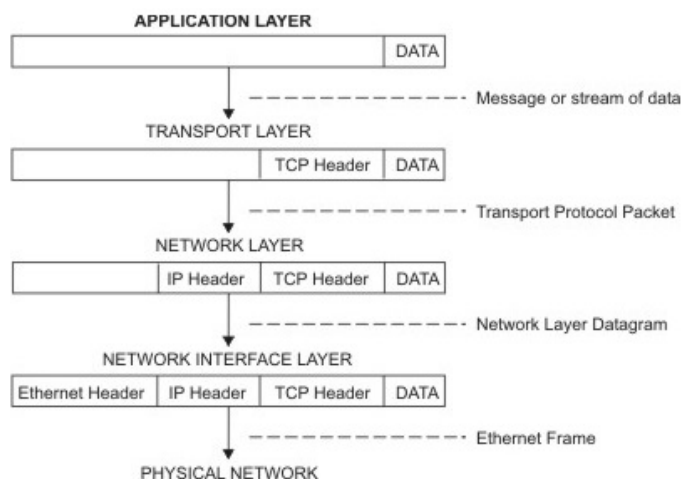
Tabulka 3.1: Výběr GSM standardů

Standard	Rychlost stahování	Rychlost nahrávání
GSM	1.8 kB s^{-1}	1.8 kB s^{-1}
GPRS	6.7 kB s^{-1}	3.35 kB s^{-1}
EDGE	27.2 kB s^{-1}	13.6 kB s^{-1}
LTE	12.5 MB s^{-1}	6.25 MB s^{-1}

při použití TCP/IP protokolu připojena hlavička. Ta může mít různé velikosti a přímo ovlivňuje velikost užitečných dat při přenosu. V hlavičce se udržují informace jako například cílová IP adresa, zdrojová IP adresa, velikost dat atp. Připojování různých hlaviček při odesílání dat z aplikace lze vidět na obrázku 3.2 [17].



Obrázek 3.1: ISO/OSI vrstvy TCP/IP protokolu



Obrázek 3.2: Připojování hlaviček při odesílání dat z aplikace.

Data jsou posílána v paketech. Každý paket má maximální velikost. Ta je dána buď hardware specifikací sítě, nebo je limitovaná softwarově. Softwarová limitace by nikdy neměla přesáhnout velikost hardwarové limitace, neboť pak by se paket musel fragmentovat, což by znamenalo celkové zpomalení

přenosu. Hardwarová specifikace maximální velikosti paketu pro síť Ethernet je 1500 B [18].

3.3 Vývojová prostředí

Při práci na praktické části byla využita vývojová prostředí, o kterých je v této sekci více diskutováno:

1. ArduinoIDE
2. Microsoft Visual Studio
3. Wolfram Mathematica

Nejsou to sice jediná použitelná prostředí pro dosažení cílů práce, avšak jsou velmi vhodnými nástroji a zároveň díky předchozí zkušenosti s těmito nástroji byla práce s nimi jednodušší.

3.3.1 ArduinoIDE

Pro vypracování praktické části bylo zvoleno ArduinoIDE (Integrated development environment) [19], které je populární platformou pro programování vestavěných systémů.

ArduinoIDE ulehčuje programování všech podporovaných vestavěných systémů. Programátor píše kód a pro překlad a nahrání do vestavěného systému stačí zmáčknout příslušné tlačítko a ArduinoIDE to provede automaticky. IDE je potřeba vždy správně nastavit, tzn. minimálně zvolit programovaný vestavěný systém a port, přes který je vestavěný systém připojen k PC.

Kód je psaný ve zjednodušeném C/C++ a je překládán pomocí `avr-gcc/avr-g++` [20]. Díky své popularitě existuje mnoho knihoven, které se dají do ArduinoIDE přidat, aby se dalo IDE použít i pro další vestavěné systémy a moduly.

3.3.2 Microsoft Visual Studio

Pro tvorbu serveru bylo zvoleno Microsoft Visual Studio 2017 (dále VS) [21]. Kromě překladače disponuje také velmi užitečným debuggerem, který pomáhá při ladění programu. Pokud je program přeložený v debug módu, je možné program kdykoliv zastavit uprostřed běhu a rovnou se kouknout na aktuální hodnoty proměnných, bez nutnosti pomocných výpisů do konzole. Program lze v debug módu také krokovat.

Do VS se dají nahrát přídatné pluginy, jimiž se dá upravit uživatelské rozhraní tak, aby to bylo pro programátora co nejpříjemnější. Díky těmto pluginům je možné například simulovat programování ve Vimu [22] se všemi klávesovými zkratkami, zatímco programátor může využívat všech výhod VS, jako například výše zmíněný debugger.

3.3.3 Wolfram Mathematica

V praktické části jsou mimo jiné měřeny hodnoty procházejícího proudu jednotlivých komponent, a tyto hodnoty je poté potřeba zpracovat a vypočítat z nich spotřebu energie. Ke zpracování spotřeb energie byl napsán skript ve Wolfram Mathematice [23], jenž ulehčuje různé matematické výpočty a vykreslování grafů. Zároveň Mathematica plně podporuje importování ze souborů a je vhodná i pro zpracování větších objemů dat.

V Mathematice je programováno pomocí jazyku Wolfram Language [24]. Wolfram Language je plnohodnotný programovací jazyk, dají se v něm naprogramovat i programy nesouvisející s matematikou. Na druhou stranu je Mathematica orientovaná spíše na výpočty. Mathematica podporuje různé složitější algoritmy bez nutnosti importovat jakoukoliv knihovnu. Zároveň při psaní v Mathematice se dá využívat speciálních symbolů, které blíže připomínají standardní symboly používané při popisu jisté funkce nebo rovnice. Toto má ovšem i své úskalí v podobě možných obtíží při vkládání těchto speciálních symbolů.

Analýza a návrh

Obsahem kapitoly je popis klíčových prvků v analýze a návrhu.

4.1 Výběr platformy

Jedním z cílů diplomové práce je navrhnout a zrealizovat vlastní zařízení, jež umožní vzdálené sledování domácností. Zařízení by mělo komunikovat pomocí GSM sítě, s možností přenášet větší objem dat a dbát na nízkou spotřebu energie. Zároveň pro diplomovou práci je potřeba vytvořit jednoduchý server, na který budou posílána data ze sledovacího zařízení.

Pro další cíle diplomové práce je potřeba změřit propustnost, latenci a spotřebu energie, aby mohly být vyvozeny závěry ohledně výdrže baterie, díky nimž se dá zhodnotit využitelnost zařízení v oblasti bez přístupu k síti.

Hlavními faktory při výběru komponent byly popularita a cena. Důvod zvolení těchto faktorů je očekávané využití zařízení v IoT. Vysoká popularita pomáhá s hledáním zdrojů a materiálů při programování jednotlivých komponent, zatímco nízká cena umožňuje lehčí škálovatelnost.

K vypracování práce je nutno rozhodnout a vybrat tyto části:

1. mikrokontrolér pro ovládání sledovacího zařízení,
2. GSM modul pro odesílání dat ze sledovacího zařízení,
3. Adaptér na SD (Secure Digital) karty pro rozšíření úložného prostoru,
4. kameru,
5. měřící zařízení,
6. server.

Jako mikrokontrolér je možné použít například Arduino, PIC32, STM32, NodeMCU a další. Termín Arduino zastřešuje více mikrokontrolérů společně

s programovacím prostředím ArduinoIDE, knihovny a online komunitou, která vytváří návody a další knihovny. PIC32 i STM32 mikrokontroléry také nabízí vlastní vývojové prostředí a jsou vhodné pro programy, ve kterých jsou složité výpočty. STM32 mikrokontroléry používají ARM (Advanced RISC Machine) procesory. NodeMCU je nejznámější čipy ESP8266 a ESP32 a má jako jednu z výhod k dispozici WiFi přímo na desce a nízkou cenu. Z těchto možností se prozatím Arduino jeví jako ta nejvhodnější, neboť hlavní výhodou NodeMCU v podobě podpory WiFi není žádaná a zároveň nejsou potřeba pro sledovací zařízení žádné složitější výpočty.

Výběr GSM modulu je omezený požadavkem možnosti LTE rychlosti a možností použití běžných SIM karet. Adaptér na SD kartu by měl být pro microSD karty, jelikož kartu lze snadno a levně zvětšit v případě, že se vybraná karta projeví jako nedostatečná. Další výhodou je možnost použít microSD karty v jiných zařízeních, například v chytrém telefonu nebo fotoaparátu. Obě komponenty, GSM modul i adaptér na SD karty, musí být kompatibilní s vybraným mikrokontrolérem.

Jelikož se jedná o jednoduchý server, na zvoleném jazyku příliš nezáleží a je důležitější, aby programátor byl se zvoleným jazykem dostatečně obeznámen.

Při výběru kamery je potřeba brát zřetel na to, jestli je kompatibilní s již vybraným mikrokontrolérem a jestli pro ni existují napsané knihovny. Další podmínkou je, aby kamera měla alespoň 5MP, za účelem možnosti posílat lepší fotky, pokud to rychlost přenosu dovolí.

Jako měřicí zařízení je možné použít samostatné produkty, které je možné vložit do přívodu energie pro mikrokontrolér. Další možností je využít jiný mikrokontrolér pro měření sledovacího zařízení. Zároveň oproti jednoúčelovým měřičům spotřeby energie lze mikrokontrolér mnohem lépe přizpůsobit specifickým potřebám měření.

4.1.1 Vybrané komponenty

Arduino MEGA 2560 [25], obrázek 4.1, bylo vybráno jako mikrokontrolér pro ovládání sledovacího zařízení (dále řídící Arduino), SIM7000E Arduino NB-IoT/LTE/GPRS/GPS Expansion Shield DFR057213 [26], obrázek 4.2, na pozici GSM modulu a Catalox MicroSD Card Adapter, obrázek 4.3, byl vybrán pro rozšíření úložného prostoru. Kamera byla použita Arducam 5MP Plus OV5642 Mini Module [27], obrázek 4.4.

Pro účely měření spotřeby energie bylo zvolené druhé Arduino MEGA 2560 (dále měřicí Arduino), z důvodů programovatelnosti a podobnosti programovacích zdrojů s řídícím Arduinem.

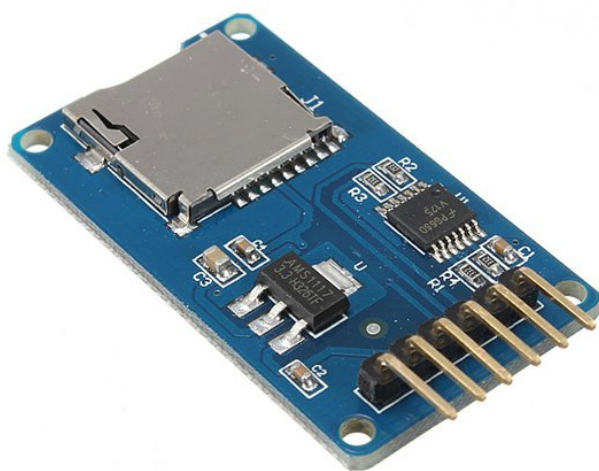
Součástí zmíněného GSM modulu je modul SIM7000E, používající AT příkazy. Seznam těchto AT příkazů lze nalézt v dokumentaci pro sérii modulů SIM7000 [28]. GSM modul poskytuje knihovnu s funkcemi, usnadňující komunikaci s SIM7000E modulem, obrázek 4.5.



Obrázek 4.1: Arduino MEGA 2560



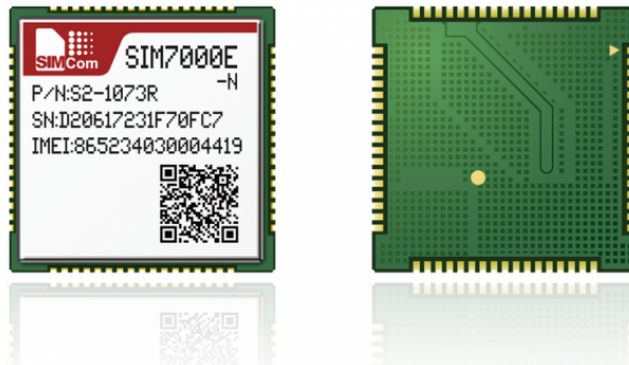
Obrázek 4.2: SIM7000E Arduino NB-IoT/LTE/GPRS/GPS Expansion Shield DFR057213



Obrázek 4.3: Catalex MicroSD Card Adapter



Obrázek 4.4: Arducam 5MP Plus OV5642 Mini Module



Obrázek 4.5: SIM7000E

Server je napsaný v C# a běží na PC. Tento server funguje primárně pro testovací účely celého zapojení, ale lze ho beze změn použít i pro reálné využití. Sice se nabízí psát server v C++, jelikož jsou v C++ psané programy pro Arduina, ovšem v C# je jednodušší napsat jednoduchý server.

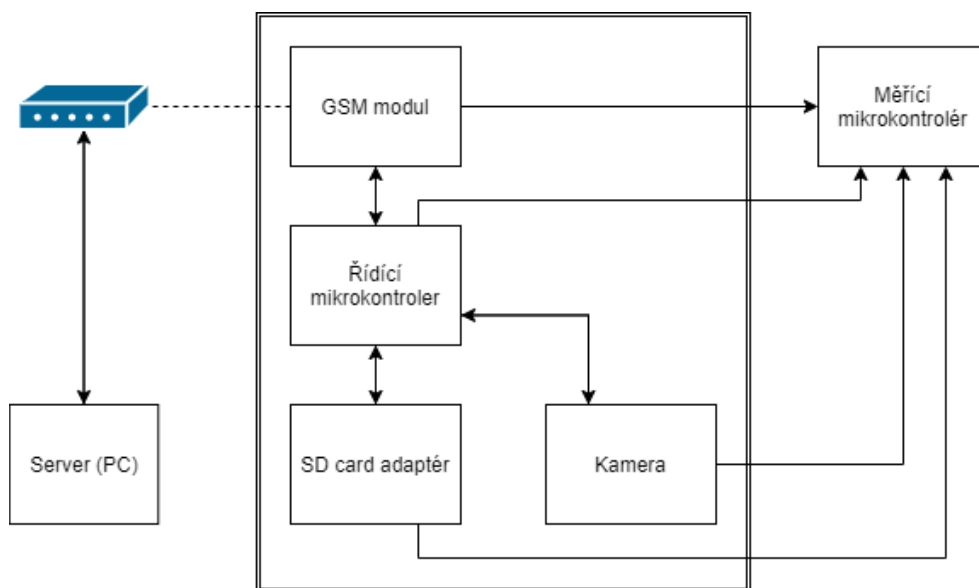
Výsledné zapojení komponent a komunikace mezi nimi lze vidět na obrázku 4.6.

Zvolení Arduina a modulů (štítů) kompatibilních s Arduinem dává možnost používat pro programovací potřeby ArduinoIDE, což je populární IDE pro programování vestavěných systémů a díky tomu lze online nalézt mnoho zdrojů pomáhajících s programováním.

Jedna z výhod celé konfigurace je možnost posílat velké soubory a fotky přes již existující GSM síť, s velkým pokrytím a velkým dosahem. Navíc v této konfiguraci lze pozměnit jakákoliv část za jinou bez velkých zásahů, např. místo/vedle kamery může být připojen teploměr pro odesílání teploty v daném místě, jako server může být použit jiný vestavěný systém atp.

4.2 Programové řešení

Pro splnění cílů práce je zapotřebí vytvořit tři různé programy; program pro řídicí Arduino, server a program pro měřící Arduino. Programy musí mezi sebou umět předávat určité informace.



Obrázek 4.6: Diagram zapojení jednotlivých komponent

4.2.1 Komunikace mezi programy

Příkaz `"AT+CIPSEND?"`, který je jeden z možných AT příkazů daný GSM modulem, udává limit maximálně 1460 B pro velikost užitečných dat v jednom paketu při odesílání. To znamená, že soubor, větší než 1460 B, je nutno odeslat po částech. Komunikace probíhá na základě TCP/IP a dalšího komunikačního protokolu.

Komunikační protokol mezi řídicím Arduinem a serverem je navrhnout následovně:

1. Řídící Arduino odesílá data (fotku) na server a začne odesláním hlavičky.
2. Server čeká na přijetí hlavičky ve správném formátu.
 - Návrh formátu hlavičky je popsán v ukázce kódu 4.1. Na konci obou řádků musí být odřádkování.
 - Příklad návrhu formátu hlavičky je v ukázce kódu 4.2.
3. Po poslání hlavičky může řídicí Arduino odeslat data na server.
 - Odesílání dat může probíhat po částech, nemusí být vše v jednom paketu.
4. Po přijetí dat, jejichž velikost je udaná v hlavičce, by se server měl vrátit do stavu očekávání hlavičky.

```
Content-length: SIZE_IN_BYTES
Filename: NAME_OF_THE_FILE
```

Ukázka kódu 4.1: Návrh formátu hlavičky při posílání na server

```
Content-length: 2200
Filename: picture.jpg
```

Ukázka kódu 4.2: Příklad návrhu formátu hlavičky při posílání na server

Komunikace mezi řídicím a měřicím Arduinem by měla probíhat na základě stavového automatu uvnitř měřícího Arduina a signálů pocházejících z řídicího Arduina.

Z teoretického hlediska není žádná komunikace mezi Arduiny potřeba, neboť měřící Arduino může číst proud protékající jednotlivými komponentami bez signálu od řídicího Arduina. Dodatečné signály ovšem mají z praktického hlediska tyto výhody:

1. Spouštěcí signál, jenž udává dobu, po kterou má smysl provádět měření.
2. Alternující signály pro udání momentálního stavu řídicího Arduina.
 - Díky tomu je možné naměřené hodnoty analyzovat lépe a přesněji, například možnost udání spotřeby energie během určitého stavu.

4.2.2 Sledovací zařízení

Jak lze z diagramu 4.6 vyčíst, sledovací zařízení se stává ze čtyř komponent; řídicí Arduino, GSM modul, adaptér na SD kartu a kamera.

Sledovací zařízení má za účel v této práci v určitých intervalech odesílat statické fotografie na server. Pro tento účel lze navrhnout strukturu programu takto:

1. Inicializace všech komponent.
2. Vyfocení fotky, uložení na SD kartu.
3. Odeslání fotky na server.
4. Počkání určitého časového intervalu a návrat k bodu 2.

Celkově se jedná o jednoduše popsatelný program, pokud komunikace s jednotlivými komponenty probíhá hladce. Situace je složitější, protože se jedná o programování vestavěných systémů, nikoli softwaru pro počítač. Možnými komplikacemi může být například špatné zapojení komponent.

Pro komunikaci s jednotlivými komponenty je potřeba využít příslušné knihovny, která usnadňuje komunikaci a zrychluje tím celkovou práci. Program je psaný v ArduinoIDE, tedy ve zjednodušeném C++.

4.2.3 Server

Server je psaný v Microsoft Visual Studio v jazyce C#. Úkol serveru je přijímat data odeslaná z řídicího Arduina a zároveň vypisovat časy, kdy se určité události dějí, pro výpočet latence a propustnosti. Funkcionalita tohoto serveru by mohla vypadat následovně:

1. Zapnutí serveru na přidělené IP adrese a zvoleném portu.
2. Naslouchání na daném portu.
3. Po připojení klienta server očekává přijmutí hlavičky v daném formátu viz ukázka kódu 4.1.
4. Otevření souboru.
5. Průběžné ukládání přijatých paketů do souboru, dokud není přijato tolik B, kolik bylo udáno v přijaté hlavičce.
6. Zavření souboru, návrat k bodu 2.

Takovýto server, ačkoliv nijak složitý, naplní všechny potřeby diplomové práce pro ověření přenosu dat přes GSM síť.

4.2.4 Měřící program

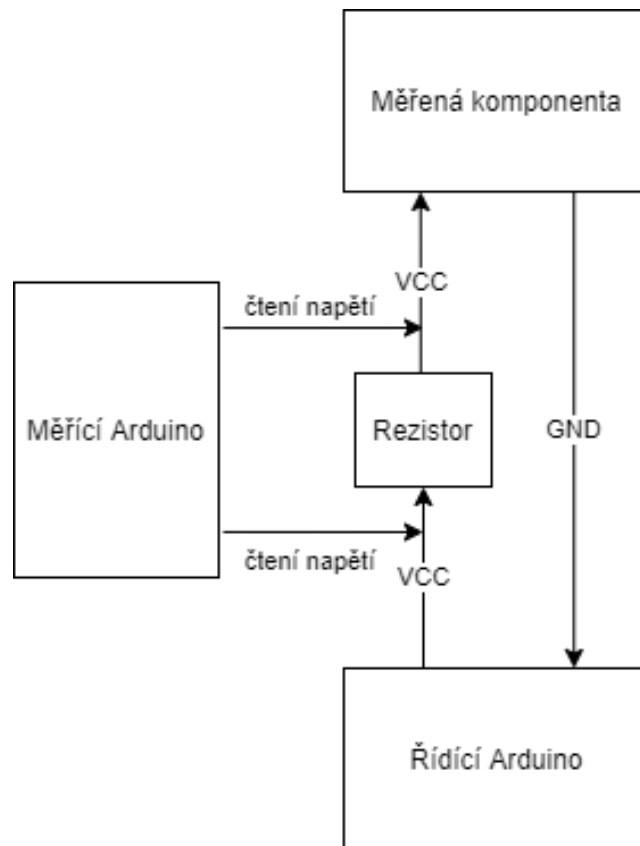
Program v měřícím Arduinu má za úkol měřit procházející proud. Navržené zapojení komponent při měření spotřeby lze vidět na obrázku 4.7.

Analogové vstupy Arduina dokáží rozpoznat napětí v rozmezí 0–5 V. Toto napětí lze pomocí A/D převodníku použít v programu v rozmezí 0–1023 a zpětně vypočítat naměřené napětí.

Limitace A/D převodníku způsobují, že nejmenší možné měřitelné napětí je $5/1023$ V. Citlivost měření závisí na zvoleném rezistoru. Čím má rezistor větší odpor, tím je měření přesnější, na druhou stranu je ale více omezen shora.

Návrh programu pro měření spotřeby energie je následovný:

1. Změření napětí v bodě před rezistorem vůči referenci uvnitř měřícího Arduina.
2. Změření napětí v bodě za rezistorem vůči referenci uvnitř měřícího Arduina.
3. Získání napětí na rezistoru z předchozích dvou měření odečtením jednoho od druhého.
4. Vypočítání proudu.
5. Vypsání proudu do konzole společně s časovou značkou.



Obrázek 4.7: Zapojení při měření proudu určité komponenty.

6. Kontrola signálů pro konečný automat.
7. Počkání určitého časového intervalu a návrat na začátek.

Tedy nejdříve je navrženo změření napětí U_{bfr} a U_{aft} před a za rezistorem, který je vložený do napájení měřené komponenty. Odečtením jednoho napětí od druhého je možné získat napětí na rezistoru U_R v závislosti na čase t dle rovnice:

$$U_R(t) = U_{bfr} - U_{aft} \quad (4.1)$$

Vypočítání proudu procházejícího rezistorem $I(t)$ v daný čas t lze pomocí Ohmova zákona za použití známé hodnoty odporu rezistoru R viz rovnice:

$$I(t) = \frac{U_R(t)}{R} \quad (4.2)$$

Vzhledem k tomu, že měřená komponenta je připojena k rezistoru sériově, viz obrázek 4.7, tak daný proud I je roven i proudu protékajícímu měřenou

komponentou. Hodnota tohoto proudu v daný časový okamžik je společně s časovou značkou přes UART (Universal Asynchronous Receiver/Transmitter) vypsaná do konzole po uplynutí nastavitelného časového intervalu.

Součástí programu by měl být konečný automat, který vypisuje s časovou značkou změnu stavu, pokud mu přijde signál od řídicího Arduina.

Celé měření je možno řídit dalším signálem z řídicího Arduina. K měření a zápisu naměřených hodnot dochází pouze v případě, kdy je řídicí signál aktivní. Tím se zbytečný výstup měřicího programu minimalizuje a zvětší se část výstupu přímo použitelná pro analýzu.

Měřicí Arduino by mělo být připojeno pomocí USB k PC, aby mohly být zaznamenány všechny výstupy s časovou značkou, za níž je vypsan buď proud, a nebo označení změny vnitřního stavu komponenty, například přechod z inicializační fáze do fáze odesílání souboru. Znalost momentálního stavu umožňuje vytvoření detailního grafu, na němž lze vidět jednotlivé fáze odběru energie, a tudíž je možné na první pohled identifikovat energeticky náročnější fáze.

4.3 Metodologie měření

V této sekci je detailněji rozebrána metodologie měření spotřeby energie, propustnosti a latence. Při testech se dají řídicí i měřicí Arduino zapojit na stejné PC, na němž běží server. Takovéto zapojení umožňuje lepší synchronizaci časů a umožňuje porovnávání časů z výpisů do konzole.

4.3.1 Spotřeba energie

Návrh měření procházejícího proudu komponentou je popsán v předchozí sekci 4.2.4 a v této sekci je blíže vysvětlen princip měření spotřeby energie.

V zmíněném obvodu, viz obrázek 4.7, zastává řídicí Arduino zdroj napětí, jenž lze označit $U_S(t)$. Prochází-li rezistorem a měřenou komponentou stejný proud $I(t)$ a je-li $U_R(t) + U_K(t) = U_S(t)$ a zároveň $U_R(t) \ll U_K(t)$ po nějaký časový interval, pak po tento časový interval je spotřeba energie rezistorem zanedbatelná vzhledem ke spotřebě komponenty a $U_S(t) \approx U_K(t)$.

Příkon $P(t)$ je dán vztahem v rovnici:

$$P(t) = U_K(t) \cdot I(t) \tag{4.3}$$

Integrace příkonu přes daný časový úsek udává spotřebu energie měřené komponenty E za tento čas viz rovnice:

$$E = \int_t P(t) dt = \int_t U_K(t) \cdot I(t) dt \tag{4.4}$$

Je-li zdroj napětí konstantní $U_s(t) = U_s = konst$ a $U_S = U_K$ pak je možné rovnici dále zjednodušit takto:

$$E = U_S \cdot \int_t I(t) dt \quad (4.5)$$

Pro možnost integrování je potřeba z naměřených hodnot proudu udělat spojitou funkci. Toho je dosaženo využitím lineární interpolace. Integrací interpolované funkce je získaná plocha pod křivkou, jenž v má jednotku As. Pro získání spotřeby energie E , již jednotka se dá vyjádřit jako Ws, je plocha pod křivkou vynásobena napětím.

S naměřenými hodnotami se dá vypočítat výdrž baterie $t_{lifespan}$ v následující rovnici, kde $B_{capacity}$ udává kapacitu baterie, která bývá uvedena v mAh a znamená množství elektrického náboje, jenž je baterie schopna poskytnout [29]. $D_{average}$ udává průměrný proud procházející zařízením.

$$\frac{B_{capacity}}{D_{average}} = t_{lifespan} \quad (4.6)$$

4.3.2 Propustnost

Propustnost udává množství dat přenesených za vteřinu. Existují minimálně dva různé druhy propustnosti, jenž jsou relevantní pro tuto práci; propustnost GSM sítě a propustnost mezi GSM modulem a serverem.

Propustnost mezi GSM modulem a serverem lze počítat zahrnutím všech inicializačních fází řídicího Arduina. Toto je relevantní v případě, kdy zařízení pracuje se spacími cykly mezi odesíláním jednotlivých souborů, tedy v momentě, kdy se vyplatí vypínat GSM modul z důvodu šetření energie. K výpočtu této propustnosti lze použít jednoduché rovnice:

$$\frac{S}{t} = T_{p_{to\ server}} \quad (4.7)$$

kde S je velikost posílaného souboru, t je celkový čas přenosu a $T_{p_{to\ server}}$ je propustnost mezi GSM modulem a serverem.

Pro výpočet propustnosti GSM sítě jde využít faktu, že soubory jsou posílány v paketech o velikosti užitečných dat nejvíce 1460 B. U přenášení části souboru lze očekávat, že existuje statická doba přenosu, která není závislá na velikosti přenášených dat, což bylo při realizaci potvrzeno. Například statická doba připravení modulu na přenos, a poté nestatická doba nahrávání a odesílání souboru. U té již závisí na velikosti souboru.

Pro výpočet propustnosti $T_{p_{no_UART}}$ GSM sítě bez dalších parametrů při přenosu souboru většího než 1460 B lze použít následující rovnice:

$$\frac{(P_{t_{msize}} - P_{t_{psize}})}{(t_{msize} - t_{psize})} = T_{p_{no_UART}} \quad (4.8)$$

kde Pt_{msize} reprezentuje maximální velikost paketu, 1460 B, a Pt_{psize} je velikost části, která je menší než maximální velikost paketu. t_{msize} a t_{psize} jsou časy přenosu těchto částí.

Mělo by být poznamenáno, že tato propustnost Tp_{no_UART} nezahrnuje čas přenosu dat z řídicího Arduina do GSM modulu. Arduino komunikuje s GSM modulem pomocí UART rozhraní s baud rate 19 200. To znamená, že přenosová rychlost mezi řídicím Arduinem a GSM modulem je limitována rychlostí 1920 B s^{-1} (8 datových bitů, start bit, stop bit). Protože tato rychlost UARTu přímo ovlivňuje dobu přenosu a je závislá na velikosti přenášeného souboru, lze čas strávený přenosem dat pomocí UARTu přímo využít k získání propustnosti Tp s rychlostí přenosu UARTu Tp_{UART} následovně:

$$\frac{(Pt_{msize} - Pt_{psize})}{((t_{msize} - \frac{Pt_{msize}}{Tp_{UART}}) - (t_{psize} - \frac{Pt_{psize}}{Tp_{UART}}))} = Tp \quad (4.9)$$

Dle článků [16] and [31] dosahuje GSM síť (2G) rychlosti 1200 B s^{-1} nebo dokonce až 1800 B s^{-1} pro nahrávání, v závislosti na kanále. Toto je důležitá rychlost pro porovnání naměřených výsledků.

Při znalosti propustnosti Tp z rovnice (4.9) a rychlosti přenosu UART rozhraní Tp_{UART} , se může vypočítat statická doba přenosu t_{static} v rovnici:

$$t_{part} - \frac{Pt}{Tp} - \frac{Pt}{Tp_{UART}} = t_{static} \quad (4.10)$$

kde Pt je velikost přenášené části a t_{part} doba přenosu této části.

Vypočítaná statická doba přenosu může být využita pro výpočet doby přenosu při jiné rychlosti sítě $Tp_{network}$. Následující rovnice je využitelná při počítání času $t_{diff_network}$ pro poslání jedné části souboru s jinou rychlostí sítě:

$$t_{static} + \frac{Pt}{Tp_{network}} + \frac{Pt}{Tp_{UART}} = t_{diff_network} \quad (4.11)$$

Pro výpočet zrychlení R_{diff} při použití jiné sítě je možno použít rovnici:

$$1 - \frac{t_{diff_network}}{t_{part}} = R_{diff_network} \quad (4.12)$$

4.3.3 Latence

Latencí je bráno zpoždění mezi akcí a reakcí [30], například doba od vyslání signálu v jednom bodě sítě do přijetí signálu v jiném bodě sítě. Skutečnost, že server běží na stejném PC, ke kterému je zapojeno řídicí Arduino, byla využita při počítání latence. Díky tomu jsou výpočty přesnější, neboť server i řídicí Arduino zapisují se stejným zdrojem času.

Při počítání latence je důležité nezapomenout na TCP/IP hlavičku o velikosti 40 B [18]. Následující rovnici lze použít pro odhadnutí latence mezi GSM modulem a serverem za předpokladu nutnosti nahrát nějaká data Pt z řídicího

Arduina do GSM modulu s propustností UARTU Tp_{UART} a propustnost GSM sítě Tp :

$$t_{measured} - \frac{Pt + Pt_{overhead}}{Tp} - \frac{Pt}{Tp_{UART}} = t_{approximated} \quad (4.13)$$

kde $t_{measured}$ je změřená latence z výpisů do konzole a $t_{approximated}$ je odhad latence mezi serverem a GSM modulem.

Realizace

Kapitola pojednává o výsledné realizaci a zapojení všech komponent. Dále jsou popsány hlavní překážky, které bylo potřeba překonat při tvorbě praktické části.

5.1 Sledovací zařízení

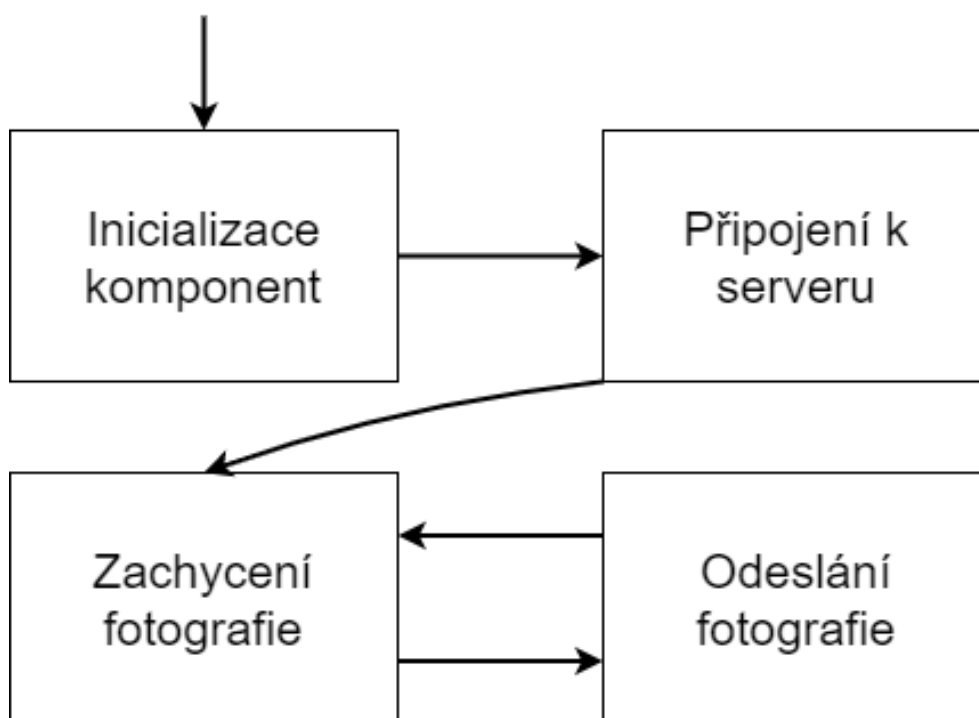
Architektura programu pro sledovací zařízení je popsána na obrázku 5.1. Jednotlivé části programu jsou realizovány dle návrhu a jsou přesněji popsány v sekci 4.2.

Program nejdříve inicializuje všechny komponenty a poté se pokusí připojit na vzdálený server. IP adresu a port tohoto serveru lze nastavit změnou příslušných konstantních proměnných. Po připojení na server je kamerou vyfocen snímek, jenž je následně uložen na SD kartu. Nakonec je tento snímek odeslán na připojený server. Jelikož jedním z cílů práce je odesílat statické fotografie v určitém intervalu, po odeslání fotografie se program vrací po uplynutí určité doby do bodu před zachycením snímku.

Modularita této architektury umožňuje lehké využití jednotlivých komponent v jiných projektech, případně zaměnění některé komponenty za jinou, například novější a výkonnější.

Hardwarové zapojení s řídicím Arduinem je popsáno:

1. v tabulce 5.1 pro GSM modul,
2. v tabulce 5.2 pro adaptér na SD kartu,
3. v tabulce 5.3 pro kameru.



Obrázek 5.1: Architektura sledovacího zařízení

Tabulka 5.1: Propojení řídicího Arduina a GSM modulu

Pin řídicí Arduino	Pin GSM modul
VIN	VIN
GND	GND
RESET	RESET
IOREF	IOREF
AREF	AREF
7	7
10	8
12	12

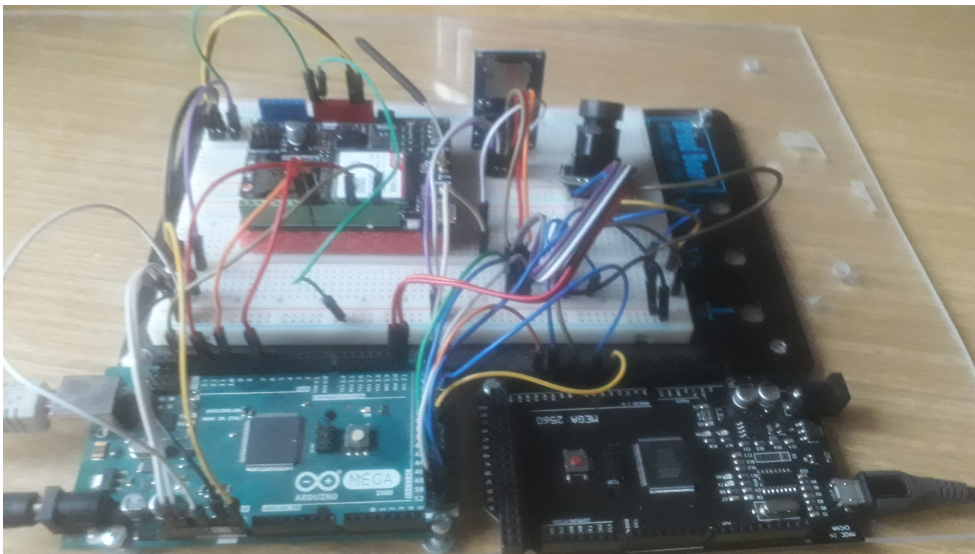
Tabulka 5.2: Propojení řídicího Arduina a adaptéru na SD kartu

Pin řídicí Arduino	Pin adaptér na SD kartu
5V	VCC
GND	GND
50	MISO
51	MOSI
52	SCK
22	CS

Tabulka 5.3: Propojení řídicího Arduino a kamery

Pin řídicí Arduino	Pin kamera
5V	VCC
GND	GND
SDA (20)	SDA
SCL (21)	SCL
53	CS
50	MOSI
51	MISO
52	SCK

Na obrázku 5.2 lze vidět fotku celého zařízení, včetně měřicího Arduino. Některé komponenty nejsou upevněny, neboť v stabilním prostředí nemá jejich upevněnost na přesnost naměřených dat vliv. Další možností je toto rozestavení dát do kartonové nebo 3D vytištěné krabice, udělat otvor pro kameru a připevnit jednotlivé komponenty přímo ke krabici.



Obrázek 5.2: Fotka realizace pro měření komponent.

5.1.1 Hlavní překážky realizace

Během realizace došlo k určitým komplikacím, jež jsou v této podsekcí blíže rozebrány. Nejvíce komplikací bylo s GSM modulem. Chyby zpravidla bylo mnohem náročnější odhalit než opravit. V podsekcí jsou popsány tyto překážky:

1. Chyba v knihovně pro GSM modul, spočívající v absenci funkce pro ověření SIM karet s PINem.
2. Chybná funkce pro odesílání dat z GSM modulu, pokud data nejsou pouze textové řetězce.
3. Hardwarový nedostatek adaptéru na SD kartu, znemožňující použití SPI (Serial Peripheral Interface) rozhraní pro více modulů zároveň.

DFRobot [26] poskytuje na svých stránkách knihovny a ukázky pro jednodušší rozchození GSM modulu pro programátory. Ačkoliv tyto knihovny fungují pro poskytnuté ukázky, během realizace vyšlo najevo, že jsou v knihovně chyby, a absence komentářů opravu neulehčila.

První chybou, bránící správné funkci, byl předpoklad tvůrce knihovny, že všechny SIM používané GSM modulem nebudou mít PIN. Pro opravu bylo potřeba dopsat do knihovny funkci, která přijímá PIN při ověření SIM karty. Stačilo v dokumentaci pro modem SIM7000E [28] najít příslušnou funkci **AT+CPIN=<pin>**, jež při ověření SIM karty použije daný pin v parametru. Bodově:

1. Knihovna nepodporovala SIM karty s PINem.
2. K opravě bylo potřeba vytvořit novou funkci v knihovně.
3. Nová funkce přijímá PIN SIM karty v parametru.
4. PIN je předán modemu pomocí **AT+CPIN=<pin>**.

Další chyba v GSM modulu byla plně odhalena při pokusu o odeslání fotky. Pro odeslání dat z GSM modulu jsou poskytnuty dvě funkce; **send(char *data)** a **send(char *buffer,size_t len)**.

První funkce zjistí délku dat pomocí funkce `strlen` a poté odešle celá data. Po odeslání čeká, dokud SIM7000E modem nepošle zpátky OK, případně ERROR. Tato funkce funguje pro odesílání textových řetězců v pořádku. Pro tuto diplomovou práci je tedy vhodná pro odeslání hlavičky na server.

Druhá funkce má navíc parametr pro označení délky dat, které se z bufferu mají poslat, ta ovšem nefungovala korektně. První chyba funkce je v zapisování celého bufferu do modemu, nehledě na daném parametru délky. Toto je hlavně problém kvůli tomu, že kdykoliv je cokoli zapsáno do modemu, tak je to zároveň vystaveno na výstup z modemu. Tedy při kontrolování, jestli byla data úspěšně odeslána, je informace o stavu odesílání až za odeslanými daty. Tato funkce používá pro kontrolu jinou knihovní funkci, jež čte z výstupu modemu pouze 100 znaků, a tudíž odpověď modemu nemusela být vůbec zkontrolována. Další chyba je, že je použita funkce `strstr` při hledání odpovědi modemu. Funkce `strstr` se zastaví, kdykoliv při čtení bufferu přečte null. Avšak null mohou obsahovat i odeslaná data.

Oprava funkce spočívá v použití jiné funkce pro zápis z GSM modulu do SIM7000E modemu. Funkce pošle pouze daný počet znaků z bufferu. Další opravou je nepoužívání funkcí pro textové řetězce, při kontrolování správného odeslání, ale kontrolovat ručně přímo nad jednotlivými přečtenými znaky. Poslední změnou je použití posuvného bufferu při kontrolování odezvy, aby nedošlo k situaci, že odpověď je rozdělena do více čtení, tím by byla vyhodnocena jako chybějící.

Tato překážka stručněji v bodech:

1. Knihovna implementovala dvě různé funkce pro odesílání dat.
2. Funkce pro odesílání textových řetězců fungovala správně.
3. Funkce pro odesílání i nespécifikovaných dat o dané délce obsahovala dvě chyby.
4. První chybou bylo ignorování parametru délky, a tudíž zapsání celého bufferu do modemu.
5. To způsobilo odeslání špatných dat.
6. Oprava spočívala v použití jiné funkce pro zápis dat do modemu. Nová funkce má jeden z parametrů délku dat. Parametr udává, kolik dat z daného bufferu má být zapsáno.
7. Druhou chybou bylo nesprávné kontrolování korektnosti odeslání dat.
8. Příčinou bylo kontrolování pouze 100 znaků odpovědi modemu a kontrolování pomocí funkce `strstr`.
9. Opravou bylo kontrolování celé odpovědi bez pomoci funkcí pracujících s textovými řetězci.

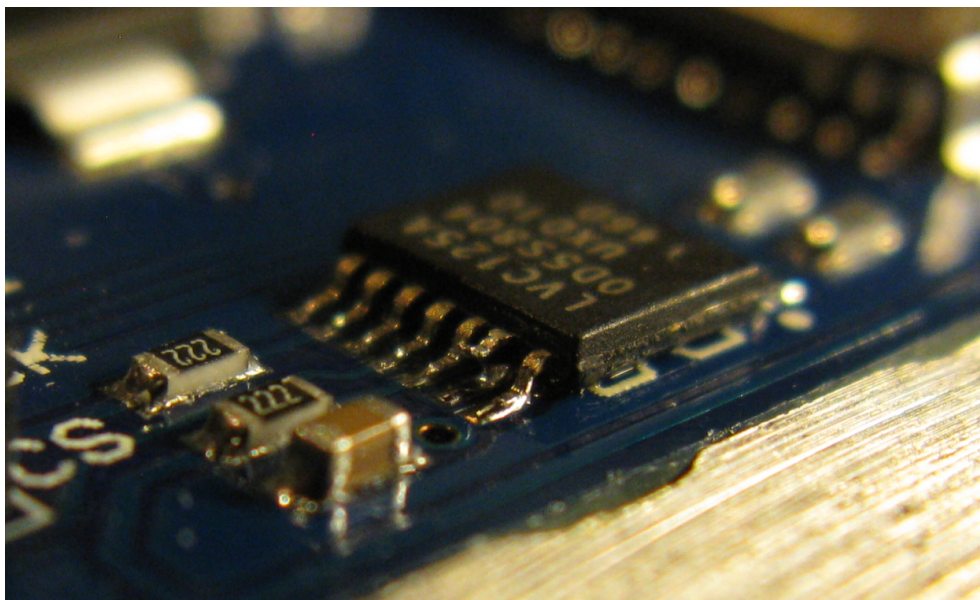
Poslední velká překážka byla způsobena zvoleným adaptérem na SD kartu. Adaptér, chybou návrhu, neumí umožnit dalším komponentám komunikovat přes SPI rozhraní. Naštěstí lze na internetu najít návod, jak modifikovat adaptér, aby fungoval správně [32].

Adaptér na SD kartu používá čip LVC125A. Pro modifikaci je potřeba odpojit pin 13 (zobrazeno na obrázku 5.3), který je momentálně zapojený na GND, a připojit ho k pinu 8 (signál pro CS). Díky tomu adaptér korektně neposílá na MISO sběrnici nic, pokud není CS aktivní pro adaptér. Výslednou modifikaci lze vidět na obrázcích 5.4 a 5.5. Bodové shrnutí této chyby:

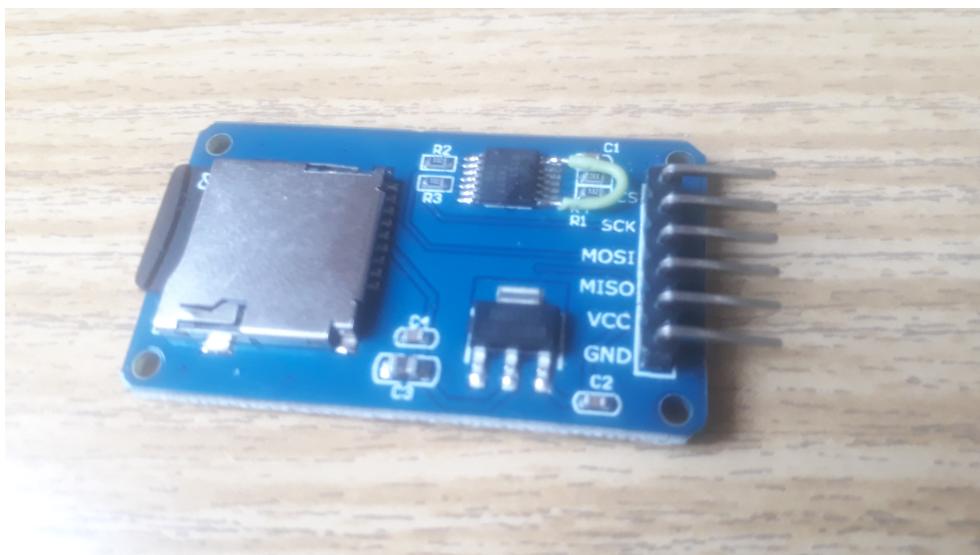
1. Adaptér na SD kartu neumožňoval jiným zařízením používat stejné SPI rozhraní.
2. Oprava spočívala v hardwarové úpravě adaptéru.

5. REALIZACE

3. Bylo potřeba odpojit pin 13 čipu LVC125A od GND.
4. Na závěr stačilo připojit pin 13 k pinu 8.



Obrázek 5.3: Rozpojený pin 13 u čipu LVC125A.



Obrázek 5.4: Modifikovaný adaptér, pohled z boku



Obrázek 5.5: Modifikovaný adaptér, pohled přímo

S kamerou žádné komplikace nenastaly, akorát z důvodu množství produktů, které ArduCAM nabízí, bylo obtížné se ze začátku zorientovat v jejich ukázkách. Jakmile se člověk zorientoval, práce s kamerou šla hladce.

5.2 Server

Server byl realizován podle analýzy a návrhu; čeká na připojení klienta, přijetí hlavičky, a následně zapisuje do souboru data, která přijme od klienta. Po odpojení klienta se může připojit nový klient. Při spuštění serveru je do konzole vypsáno, kam server ukládá soubory. Poté je vypsána lokální IP adresa a port.

Popořadě tedy server provádí tyto činnosti:

1. Vypsání místa na disku, kde jsou soubory ukládány.
2. Vypsání lokální IP adresy a port, na kterém bude server spuštěn.
3. Spuštění serveru.
4. Čekání na připojení klienta.
5. Čekání na přijetí hlavičky ve správném formátu.
 - Pokud hlavička nemá správný formát, je spojení s klientem ukončeno a návrat k bodu 4.
6. Otevření souboru pro zápis dat.

7. Přijímání a postupné zapisování přijatých dat do otevřeného souboru.
8. Zavření souboru.
9. Ukončení spojení a návrat k bodu 4.

Port používaný serverem bylo potřeba přeměřovat na správné PC, aby byla umožněna komunikace se sledovacím zařízením. Přesměrování se dělá v nastavení modemu.

Pro zjištění IP adresy, na níž se má sledovací zařízení připojit, je důležité zjistit veřejnou IP. Ta se dá zjistit například napsáním do Googlu "What's my IP".

5.3 Měřící program

Měřící program má za úkol změřit procházející proud tří komponent; kamery, adaptéru na SD kartu a GSM modulu. Komponenty jsou při měření zapojeny dle návrhu viz obrázek 4.7.

Program plní funkci tří individuálních programů pro měření proudu. Každé rozdělení je určeno pro měření jiné komponenty - GSM modulu, adaptéru na SD kartu a kamery. Rozčlenění hlavní funkce programu je provedeno pomocí "#define" direktivy. Měření by mohlo být provedeno pro všechny komponenty zároveň, nevýhodou je však nutnost rozdělit výstup měření jednotlivých komponent. Proto je zvoleno měřit komponenty samostatně a rozčlenit funkcionality měřícího programu na tři.

Funkce programu je popsána v návrhu a průběh měření lze rozdělit do dvou částí:

1. Měření proudu na rezistoru.
2. Stavový automat vypisující popisky, podle příchozích signálů z řídicího Arduina.

Měření proudu probíhá každých 5 ms změřením napětí na rezistoru o odporu $0.22\ \Omega$, jež je vložený do napájení dané komponenty. Proud procházející rezistorem, vypočítaný pomocí Ohmova zákona, je poté vypsán do konzole společně s časovou značkou. Pro časovou značku je použita funkce `millis()`, která vrací počet milisekund od spuštění programu.

Kvůli limitacím A/D převodníku lze tímto způsobem měření se zvoleným rezistorem naměřit nejmenší nenulový proud $0.022\ 21\ \text{A}$. Se stejným rezistorem je možno měřit proud až do $22.7272\ \text{A}$.

Stavový automat funguje na principu Mealy - výstup je vypisován při změně stavu. Automat je řízen dvěma signály, které pocházejí z řídicího Arduina. Jednotlivé stavy čekají na aktivaci příslušného signálu. Po aktivaci

přejde stavový automat do nového stavu a při přechodu vypíše do konzole časovou značku společně s předem definovanou zprávou pro každý stav.

Cílem stavového automatu je možnost identifikace jednotlivých fází na řídicím Arduino pro lepší a přehlednější grafy zpracovaných výpisů s procházejícím proudem komponent.

Zároveň je z řídicího Arduina řízen další signál, udávající, kdy má měřicí Arduino pracovat. Při aktivaci signálu si měřicí Arduino resetuje stavový automat, což umožňuje provádět více měření postupně, bez nutnosti ručně resetovat Arduino po každém měření.

Shrnutí funkce měřicího programu:

1. Počkání na měřicí signál z řídicího Arduina, jenž povoluje měření.
2. Reset stavového automatu.
3. Změření napětí před rezistorem a za rezistorem.
4. Výpočet napětí na rezistoru.
5. Použití Ohmova zákona k výpočtu proudu.
6. Vypsání časové značky a naměřeného proudu do konzole.
7. Přechzení signálů z řídicího Arduina, jež hýbají stavovým automatem.
8. Vypsání časové značky a připravené zprávy do konzole při změně stavu.
9. Čekání určitého časového intervalu.
10. Po vypnutí měřicího signálu konec měření, jinak návrat k bodu 3.

Hardwarové zapojení měřicího Arduina je zobrazeno v tabulce 5.4. Dále jsou piny A8 a A10 na měřicím Arduino zapojeny před a za rezistor, použitý pro měření spotřeby.

Tabulka 5.4: Propojení řídicího Arduina a měřicího Arduina

Pin řídicí Arduino	Pin měřicí Arduino
39	A4
35	A6
31	A12

5.4 Zpracování naměřených hodnot

Naměřené hodnoty proudu bylo potřeba nadále zpracovat tak, aby se z toho dala vytěžit použitelná data. Pro tuto potřebu byl vytvořen skript v Mathematice.

Skript má za úkol vzít soubor s naměřenými hodnotami a vytvořit z něj graf měření, spočítat celkovou spotřebu energie a dobu měření. Architektura skriptu je popsána v následujícím seznamu:

1. Načtení souboru.
2. Rozdělení dat ze souboru do dvou až čtyřech menších částí, v závislosti na měřené komponentě.
3. Výpočet doby měření.
4. Výpočet spotřeby energie.
5. Vykreslení grafu procházejícího proudu.
6. Přidání označení začátku jednotlivých fází do grafu procházejícího proudu, aby bylo lépe v grafu vidět, co je měřeno.

Skript nejdříve otevře daný soubor a rozdělí ho do dvou proměnných. V jedné z nich jsou pouze naměřené hodnoty a v druhé jsou pouze všechna označení změn fází. Během zpracování souboru s naměřenými hodnotami při odesílání z GSM modulu proběhne ještě další rozdělení tak, aby bylo možné určit spotřebu během samotného odesílání souboru. Doba měření je spočítána jako čas mezi prvním a posledním měřením.

Naměřená data udávají naměřený proud procházející rezistorem vždy v určitém čase - není to spojitá funkce. Jak již bylo popsáno v sekci 4.3.1, nejdříve je provedena lineární interpolace naměřených dat, aby se z jednotlivých bodů stala funkce. Pro lineární interpolaci je použit jednoduchý mód, ten pouze přímo spojí jednotlivé body. Funkce je poté zintegrována od prvního časového bodu do posledního časového bodu. Výsledkem je plocha pod křivkou, neboli celková spotřeba energie v daném čase po vynásobení napětím.

Dále jsou data vykreslena do grafu s jednotlivými body spojenými křivkou. Na závěr jsou do tohoto grafu přidána označení jednotlivých fází. Při tvoření těchto označení jsou jednotlivé popisy fází rozděleny s jinou hodnotou osy y, aby se texty nepřekrývaly.

Hlavní výhodou tohoto skriptu je jeho přenositelnost na měření jiných komponent. Dokud měření dodržuje formát s časovými značkami na první pozici a na druhé pozici je buď naměřená hodnota proudu, a nebo text udávající změnu fáze, tak lze aplikovat tento skript na měření jakékoliv komponenty. Jediné, co je potřeba změnit pro jiné měření, je první řádek, v němž se odkazuje na soubor s naměřenými hodnotami.

Testování

Zde je popsán způsob testování realizace. Součástí kapitoly jsou všechny naměřené hodnoty a jejich analýza.

6.1 Testování funkcionality

Při testování byly nejdříve jednotlivé komponenty a server testovány samostatně, až poté byly testovány všechny naráz. Postupné testování pomohlo odhalit chyby při práci na nižší úrovni. V případě, kdy by byl tento krok přeskočen, mohlo by hledání skryté chyby zabrat mnohem déle. Zároveň díky tomuto rozdělení dojde k bližšímu seznámení s jednotlivými komponentami. Seznam popsaných testů:

1. Test serveru pomocí klienta v C#.
2. Test adaptéru na SD kartu.
3. Otestování kamery pomocí softwaru od ArduCAM.
4. Zachycení snímku z kamery na SD kartu.
5. Odeslání textové hlavičky z GSM modulu na server.
6. Odeslání malých a velkých souborů z GSM modulu.
7. Test všech komponent zároveň.
8. Vyzkoušení UDP protokolu.

Pro otestování správné funkce serveru byl napsán klient v C#. Tento klient se připojí k serveru, odešle hlavičku, poté odešle soubor po částech. Během testování byl klient připojen k serveru jak přes lokální IP adresu, tak přes

veřejnou IP adresu. Také byla testována různá velikost částí, po kterých byl soubor odeslán.

Adaptér byl otestován pomocí zápisů do souboru a čtení ze souborů z SD karty, na níž byly předem nahrané v PC soubory. Dále byl vytvořen na SD kartě, zapojené do adaptéru, soubor. Do souboru byla zapsána série znaků. Poté byl soubor uzavřen, znovu otevřen v módu pro čtení a celý soubor byl vypsan. Na závěr testu byla daná SD karta zapojená zpět do PC pro kontrolu, jestli (textový) soubor jde přečíst i v PC.

Kamera byla nejdříve samostatně otestována pomocí programu od ArduCAM [33], umožňující přenášet fotky z kamery přes UART rozhraní. Během tohoto testu se již objevily první problémy s kompatibilitou s adaptérem na SD kartu, neboť i přes to, že nebyl zrovna adaptér používán, byl zapojený ke stejnému Arduinu. Řešení tohoto problému viz podsekcce 5.1.1. Po odpojení adaptéru a ujištění se, že všechny piny kamery jsou správně zapojené, nebyl s kamerou žádný další problém.

V druhé řadě bylo otestováno zachycení snímku na SD kartu. S hardwarově opraveným adaptérem na SD kartu proběhl test bez problému. SD karta byla po testu vyjmuta z adaptéru a zachycené snímky byly otevřeny v PC pro ověření správné funkce.

Všechny testy obsahující GSM modul proběhly za použití T-Mobile SIM karty s koupeným 4G internetem. Byla vyzkoušena taktéž TESCO SIM karta, s tou byl ovšem problém připojit se do sítě. TESCO operátor je virtuální a samotný přenos probíhá po O2 síti. S T-Mobile SIM kartou se šlo připojit do sítě, když byl v GSM modulu nastaven mód "GSM only" a nebo "GSM and LTE only". Nešlo se připojit, pokud byl nastavený mód "LTE only". U obou SIM karet 4G internet funguje v chytrých telefonech.

První testy GSM modulu proběhly nejdříve pro odeslání textové zprávy (hlavičky) na již dříve otestovaný a funkční server. Další na řadě bylo odesílání souborů. V první řadě testů, kde byly odeslány pouze malé soubory nebyly odhaleny žádné chyby.

Bylo vyzorováno, že při odesílání souborů se samotné odeslání paketu děje ve dvou fázích. V první fázi se SIM7000E modem připravuje odeslat data a v druhé fázi jsou teprve nahrávána data z řídicího Arduina a odeslána.

Při poslání větších souborů s dobou přenosu 5 minut došlo k nesprávnému odeslání souboru. GSM modul považoval přenos za hotový, zatímco server ještě nepřijal celý soubor. Oprava (již popsána v podsekcce 5.1.1) nejdříve spočívala v čtení celého výstupního bufferu z SIM7000E modulu. Tato oprava se jevila jako dostatečná, dokud byly posílané pouze textové soubory.

Až u testu celého sledovacího zařízení bylo objeveno, že předchozí oprava nebyla dostatečná. Ačkoliv předchozí oprava kontrolovala, jestli se zpráva odeslala, kontrola byla prováděna pomocí funkcí operujících s textovými řetězci. To způsobilo, že při pokusu o odeslání fotky, obsahující i byte 0x00, nefungovalo potvrzení odeslání korektně, protože kontrolující funkce se zastavila

na daném bytu. Oprava spočívala v manuálním procházení bytů pro kontrolu odeslání.

Během testování došlo také k vytvoření serveru a klienta, jenž používají protokol UDP. Při pokusu odeslání z GSM modulu na tento server bylo ovšem okamžitě jasné, že UDP protokol není vhodný pro přenos souborů.

Shrnutí této sekce:

1. Testy serveru a adaptéru proběhly bez komplikací.
2. Testování kamery odhalilo hardwarovou chybu v adaptéru na SD karty.
 - Po opravení této hardwarové chyby fungovala kamera i adaptér na SD kartu zároveň korektně.
 - Oprava je popsána v podsekcí 5.1.1.
3. Testy GSM modulu odhalily problémy s odesíláním větších souborů.
 - Oprava spočívala v předělání jedné ze dvou knihovních funkcí pro odesílání dat.
4. Test celku odhalil nedostatečnou opravu knihovní funkce pro odesílání.
 - Byl problém s kontrolou správnosti odeslání dat, pokud data obsahovala byte 0x00.
 - Oprava spočívala v odstranění používání funkcí pro textové řetězce při kontrole korektnosti odeslání dat.
5. Protokol UDP se prokázal jako nevhodný pro přenos souborů.

6.2 Naměřené hodnoty

Výsledky měření jsou prezentovány v této sekci. Jedná se o měření:

1. propustnosti,
2. latence,
3. spotřeby energie.

Při prezentování výsledků je často použit medián místo průměru, protože hodnoty mediánu nejsou tak moc ovlivněny extrémními naměřenými hodnotami v porovnání s průměrnými hodnotami. Extrémy obsahují mimo jiné různé zpoždění na GSM síti (např. připojení k operátorovi), a hlavně nejsou zmíněné extrémy ovlivnitelné zařízením.

6.2.1 Propustnost

Během posílání souboru o velikosti 3000 B, a zvolení maximální velikosti užitečných dat v jednom paketu 1460 B, je soubor poslán ve třech částech; 2x 1460 B a 1x 80 B. Naměřené mediánové doby pro odeslání těchto částí je zobrazeno v tabulce 6.1.

Tabulka 6.1: Naměřené doby přenosu jednotlivých částí

Velikost části [B]	Čas přenosu části [s]
80	9.533
1460	11.682

První rovnice pro výpočet propustnosti GSM sítě je rovnice (4.8). Po dosazení naměřených hodnot vyjde 642.159 B s^{-1} . Nicméně tato propustnost vypočítaná v rovnici (4.8) obsahuje nejenom propustnost GSM sítě, ale zároveň je ve výpočtu zahrnut i čas strávený přenosem dat z řídicího Arduina do modulu přes UART rozhraní.

Když se použije rychlost UART rozhraní $Tp_{UART} 1920 \text{ B s}^{-1}$, vyjde propustnost Tp z rovnice (4.9) 964.866 B s^{-1} . Články [16] a [31] udávají rychlost GSM sítě (2G) 1200 B s^{-1} , případně až 1800 B s^{-1} při nahrávání. Rychlost vypočítaná v rovnici (4.9) může být označena za náležící GSM síti. To potvrzuje, že GSM modul přenášel data rychlostí 2G a ne LTE.

Když se použije výsledek rovnice (4.9) v rovnici (4.10) pro výpočet statické doby přenosu, s velikostí části 1460 B, vyjde 9.408 s.

Statická doba přenosu může být využita v rovnici (4.11) pro výpočet délky přenosu jedné části 1460 B pomocí LTE rychlosti $t_{diff_network}$, jenž může dosáhnout až 6250 kB s^{-1} [16]. Doba přenosu 1460 B paketu pomocí LTE rychlosti je 10.168 s.

Dle očekávání je tato doba přenosu kratší, než při přenášení pomocí GSM sítě. Výsledek výpočtu celkového zrychlení pro přenos jedné 1460 B části z rovnice (4.12) je 12.96 %.

Toto ukazuje, že propustnost sítě má nízký vliv na celkovou dobu přenosu souboru. Statická doba přenosu má mnohem větší vliv na celkovou dobu přenosu souboru, a proto při snaze o celkové zrychlení by bylo důležitější se zaměřit na redukování statické doby přenosu namísto zrychlení sítě.

Ohledně propustnosti jsou vytvořena tato shledání:

1. Propustnost GSM sítě Tp vychází 964.866 B s^{-1} .
2. Statická doba přenosu při odesílání z GSM modulu trvá 9.408 s neohledě na propustnosti sítě.
3. Pokud by byly pakety odesílány pomocí LTE rychlosti, vyšlo by celkové zrychlení 12.96 %.

4. Rychlejší posílání souborů je vysoce závislé na zkrácení statické doby přenosu, neboť zrychlení propustnosti sítě má v porovnání malý vliv.

6.2.2 Latence

K měření latence bylo využito skutečnosti, že řídicí Arduino bylo připojeno k PC, na kterém běžel server, tedy se stejným zdrojem hodin. Vzhledem k tomu, že zvolený GSM modul neposkytuje přímo ping funkcionalitu, bylo potřeba odhadnout latenci jinak. Jednosměrná latence lze odhadnout jako čas mezi odesláním hlavičky z GSM modulu a její přijetí na serveru.

Hlavička v tomto testu má velikost 40 B. Zvolená metoda měření latence není stoprocentně přesná, ale poskytuje dobrý odhad. Medián naměřené latence je 321.5 ms. Mělo by ovšem být poznamenáno, že tento odhad obsahuje následující operace: nahrání 40 B dat z řídicího Arduina do GSM modulu, odeslání přes GSM síť na server a jakékoliv další zpoždění spojené s TCP/IP protokolem.

Jelikož propustnost T_p byla již vypočítaná v předchozí sekci 6.2.1, propustnost $T_{p_{UART}}$ a velikost TCP/IP hlavičky $P_{t_{overhead}}$ jsou známé, tak mohou být použity pro lepší odhad latence. Po dosazení do rovnice (4.13) s P_t rovno 40 B vyjde po konverzi jednotek 217.75 ms.

Chytrý telefon se stejnou SIM kartou a s nastaveným 2G internetem má průměrnou latenci při pingování (dvousměrná latence) 189.015 ms, neboli 94.507 ms jednosměrné latence. Dříve odhadnutá latence stále nepočítá s latencí USB (přenos z řídicího Arduina do konzole v PC) ani s jakýmkoliv dalším zpožděním probíhajícím mezi GSM modulem a serverem, jako například latence na síti, latence mezi modemem a připojeným PC atd.

Shrnutí měření latence:

1. Byla použita doba mezi odesláním paketu a přijetím paketu na serveru.
2. K lepšímu odhadu bylo potřeba počítat s velikostí TCP/IP hlavičky.
3. Propustnost z předchozí sekce 6.2.1 byla využita k zlepšení odhadu.
4. Jednosměrná latence byla odhadnuta na 217.75 ms.
5. Chytrý telefon se stejnou SIM kartou má jednosměrnou latenci 94.507 ms při použití 2G sítě.
6. Odhadnutá latence nepočítá s různými dílčími zpožděními, například latence mezi modemem a serverem, jejichž znalost by zlepšila odhad latence.

6.2.3 Spotřeba energie

V této podsekci jsou prezentovány a analyzovány spotřeby energie pro:

1. GSM modul,
2. kameru,
3. adaptér na SD kartu.

Zároveň je provedena analýza výdrže baterie, jež by napájela různé komponenty s uvedenými spotřebami energie.

Naměřený procházející proud GSM modulem při posílání 1 B souboru je zobrazen na obrázku 6.1, na kterém lze vidět jednotlivé fáze inicializace a odesílání souboru z GSM modulu: zapnutí GSM modulu, nastavení baud rate, kontrola SIM kart, nastavení net módu, kontrola signálu, připojení k operátorovi, připojení k serveru, odesílání hlavičky a souboru, odpojení od serveru a vypnutí GSM modulu. Poslání pouze 1 B souboru je identifikováno nejvíce statických spotřeb energie.

V tabulce 6.2 jsou naměřené mediánové hodnoty spotřeby energie z alespoň pěti měření pro každou velikost souboru. Z této tabulky jde shledat, že část odesílání roste úměrně s velikostí souboru; což jasně značí, že poslání jednoho velkého souboru je mnohem úspornější vzhledem ke spotřebě energie, oproti poslání více malých souborů s dohromady stejnou velikostí.

Tabulka 6.2: Naměřené spotřeby energie pro odesílání různých souborů

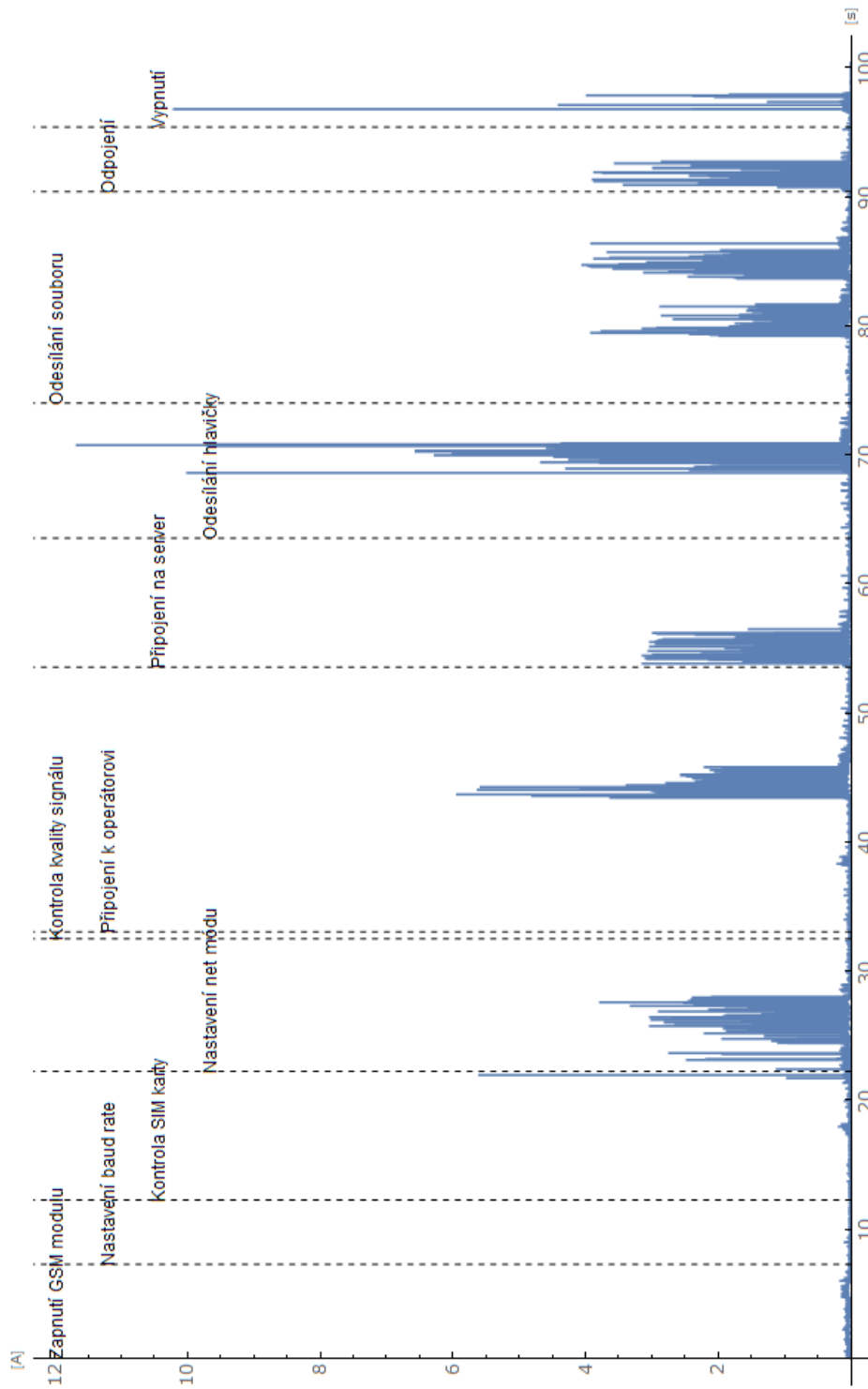
Velikost [B]	Spotřeba [Ws]	Doba [s]	Část odesílání ^a [%]
1	44.587	100.365	33.050
3 000	91.173	119.809	47.917
10 000	98.332	169.943	63.546
100 000	795.75	704.742	91.046

^aZobrazuje spotřebu energie při posílání souboru (včetně hlavičky) jako procento celkové spotřeby.

Jak je patrné z 6.1, malé soubory spotřebují hodně energie na inicializaci a dalších fázích, které nejsou závislé na velikosti souboru. Velké soubory mají podobné grafy procházejícího proudu s jediným rozdílem, že fáze odesílání souboru je delší. Poměr nezávislých fází na velikosti souboru vůči odesílací fázi se zmenšuje zvětšováním velikosti souboru.

Spotřeba energie z [26] je 4.0 Ws pro stáhnutí 10 kB souboru přes GSM síť. Pro porovnání s tímto číslem je použita část odesílání spotřeby energie z tabulky 6.2 pro odeslání 10 kB, což vychází 62.486 Ws. Ovšem nutno zdůraznit, že [26] používá chytrý telefon a ne vestavěný systém.

Studie [28] udává, že použitý GSM modem má průměrný procházející proud 300 mA. Udané číslo lze srovnat s průměrným procházejícím proudem GSM modulem použitým v sledovacím zařízení, jenž je včetně všech inicializačních fází 115.723 mA pro odeslání 10 kB souboru.



Obrázek 6.1: Procházející proud při odesílání 1 B souboru z GSM modulu.

Pro GSM modul:

1. Byly prezentovány naměřené hodnoty v tabulce 6.2.
2. Tyto hodnoty byly konvertovány pro možné porovnání s jinými studiemi [26] a [28].
3. Také je ukázán graf s proudem procházejícím GSM modulem v jednotlivých fázích.

V tabulce 5.2 jsou naměřené spotřeby energie adaptéru na SD kartu a kamery. Všechny tyto hodnoty jsou průměrnými hodnotami z alespoň 10 měření. Medián zde není zvolen z důvodu absence extrémů, které byly způsobeny v GSM modulu hlavně připojením k operátorovi a dalším zpožděním na GSM síti.

Pro měření kamery byl napsán program, který inicializuje modul, poté kameru, pak připraví kameru a zachytí snímek, a na závěr zapíše tento snímek na SD kartu. Spotřeba energie je v tomto případě měřena pouze u kamery, to znamená, že není zachycená spotřeba energie, kterou potřebuje adaptér při zapisování dané fotografie. Bylo měřeno s rozlišením 1024 x 768 a 320 x 240.

Na obrázku 6.3 je graf naměřeného procházejícího proudu při zachycení a zápisu snímku s rozlišením 1024 x 768, který má velikost 21 837 B. Fotka 6.2 je s rozlišením 320 x 240 a má velikost 12 271 B. Lepší rozlišení snímku než 1024 x 768 nebylo měřeno, z důvodu doby odeslání 100 kB souboru z GSM modulu, která přesahuje 11 min viz tabulka 6.2.

Adaptér byl měřen pomocí programu, jenž nejdříve přečte soubor o dané velikosti, a poté zapíše do jiného souboru, se stejnou velikostí. Byly provedeny dvě sady měření, jedna pro velikost čtení a zápisu 10 kB, druhá 100 kB. Na obrázku 6.4 je vykreslený procházející proud při měření s velikostí souboru 10 kB a na obrázku 6.5 je graf procházejícího proudu s velikostí souboru 100 kB.

Jeden z důvodů, proč zápis je rychlejší, je způsob, jakým byl program realizován - čtení bylo prováděno jednotlivě po bytech (tak, jak je čteno ze souboru při odeslání z GSM modulu), zatímco zápis byl prováděn po 10 B najednou (kamera zapisuje po více).

Tabulka 6.3: Naměřené spotřeby energie adaptéru a kamery

Komponenta	Spotřeba [W s]	Doba měření [s]
Adaptér, 10 kB	0.463	0.716
Adaptér, 100 kB	11.744	5.341
Kamera, 320 x 240	17.742	3.148
Kamera, 1024 x 768	140.928	13.071

Shrnutí měření adaptéru na SD kartu a kamery:



Obrázek 6.2: Zachycený snímek s rozlišením 320 x 240

1. Naměřené hodnoty jsou prezentovány v tabulce 6.3.
2. Grafy naměřeného procházejícího proudu pro obě komponenty jsou ukázány.
3. Je vysvětlen nepoměr zapisovací vůči čtecí fázi u grafu spotřeb adaptéru na SD kartu.

S naměřenými hodnotami GSM modulu lze vypočítat výdrž baterie, například 9.6 V baterie s parametrem $B_{capacity}$ 2500 mAh. GSM modul se spotřebami energie uvedenými v tabulce 6.2 má průměrnou spotřebu proudu $D_{average}$ 115.723 mA, za celou dobu odesílání 10 kB souboru.

Při použití popsané baterie v rovnici (4.6) vyjde výdrž baterie $t_{lifespan}$ 21.603 h. To znamená 457.628 odeslání 10 kB souboru, včetně všech inicializačních fází. Nutno znovu zdůraznit, že tahle hodnota počítá pouze se spotřebou GSM modulu a ne se všemi komponentami.

Pro přesnější výpočet výdrže baterie lze počítat se spotřebou energie pro odeslání 10 kB souboru z GSM modulu z tabulky 6.2, spotřebou energie při zápisu a čtení 10 kB souboru pomocí adaptéru, a navíc zachycení a zápis snímku kamerou o rozlišení 320 x 240 z tabulky 6.3. Tento snímek má sice větší velikost souboru, než 10 kB, ale pro výpočet výdrže baterie je to přijatelné, neboť velikost snímku záleží také na tom, co je foceno - snímky zachycené s nasazenou krytkou, tedy plně černé, mají velikost souboru okolo pouhých 3 kB, díky JFIF formátu.

Průměrný spotřebovaný proud je vypočítán jako součet uvedených spotřeb energie v příslušných tabulkách 6.2 a 6.3, vydělený součtem doby měření, bez započítání doby pro adaptér. Doba měření adaptéru není započítána, neboť kromě inicializace, která tvoří malou část celého měření viz obrázek 6.4, je tato doba obsažena při zápisu snímku u kamery, případně odeslání souboru v GSM modulu. Průměrný spotřebovaný proud s třemi komponentami vyjde 134.654 mA s celkovou dobou aktivní fáze 173.091 s.

Po dosažení do rovnice (4.6) s výše popsanou baterií s parametrem $B_{capacity}$ 2500 mAh a výše vypočítaným průměrným proudem $D_{average}$ 134.654 mA vyjde očekávaná výdrž baterie $t_{lifespan}$ 18.566 h. Toto umožní až 386.141 zachycení, uložení a odeslání snímku.

V případě, že by došlo k odeslání jednoho snímku za hodinu by tato baterie zvládla odeslat až 386 snímků, neboli baterie by vydržela 386 h, či 16.08 dní.

Dohromady tyto výpočty ohledně výdrže baterie obsahují:

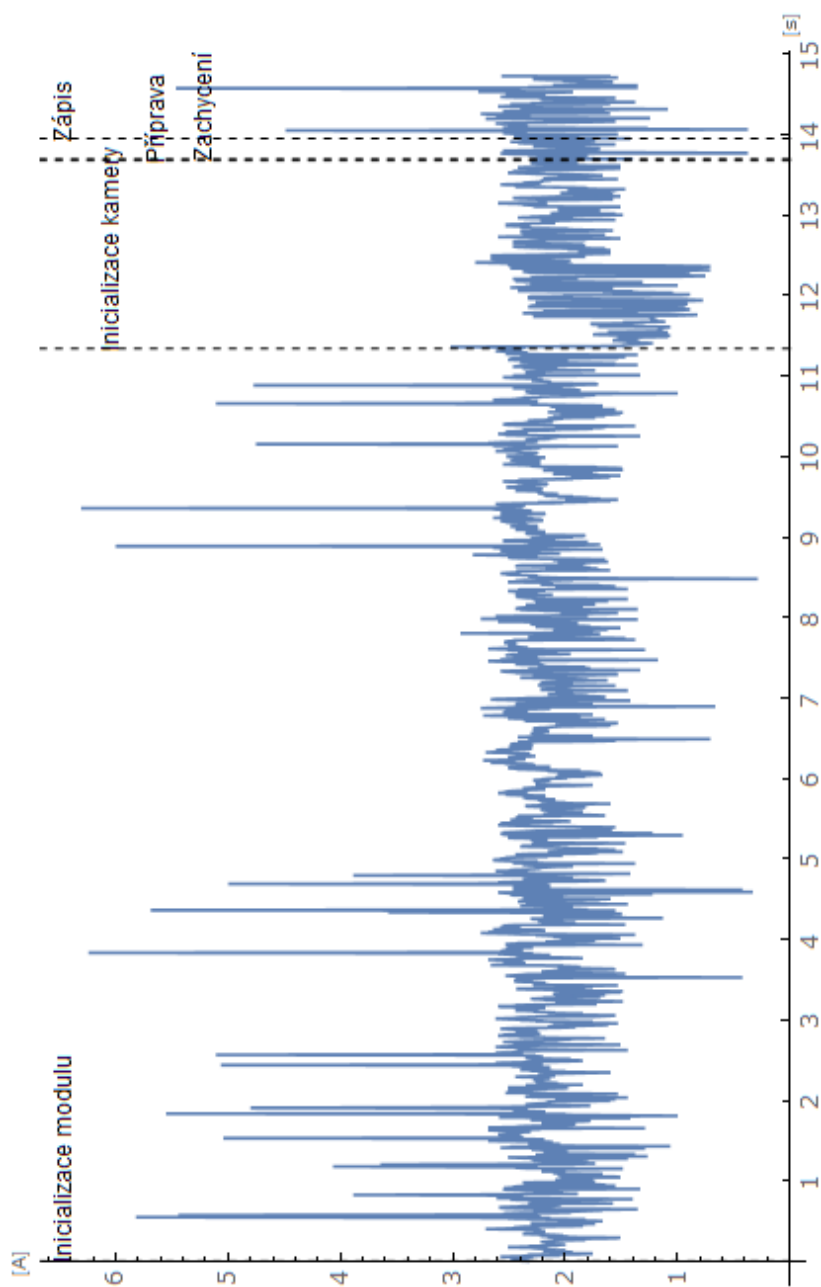
1. Je počítáno s 9.6 V baterií, která má kapacitu 2500 mAh.
2. Při výpočtech jsou použity naměřené spotřeby energie z tabulek 6.2 a 6.3.
3. Baterie by při počítání pouze s GSM modulem vydržela 457.628 odeslání 10 kB souboru
4. Po započítání spotřeby energie kamery a adaptéru se výdrž baterie sníží na 386.141 zachycení, uložení a odeslání snímku.

6.3 Hlavní omezující faktory zařízení

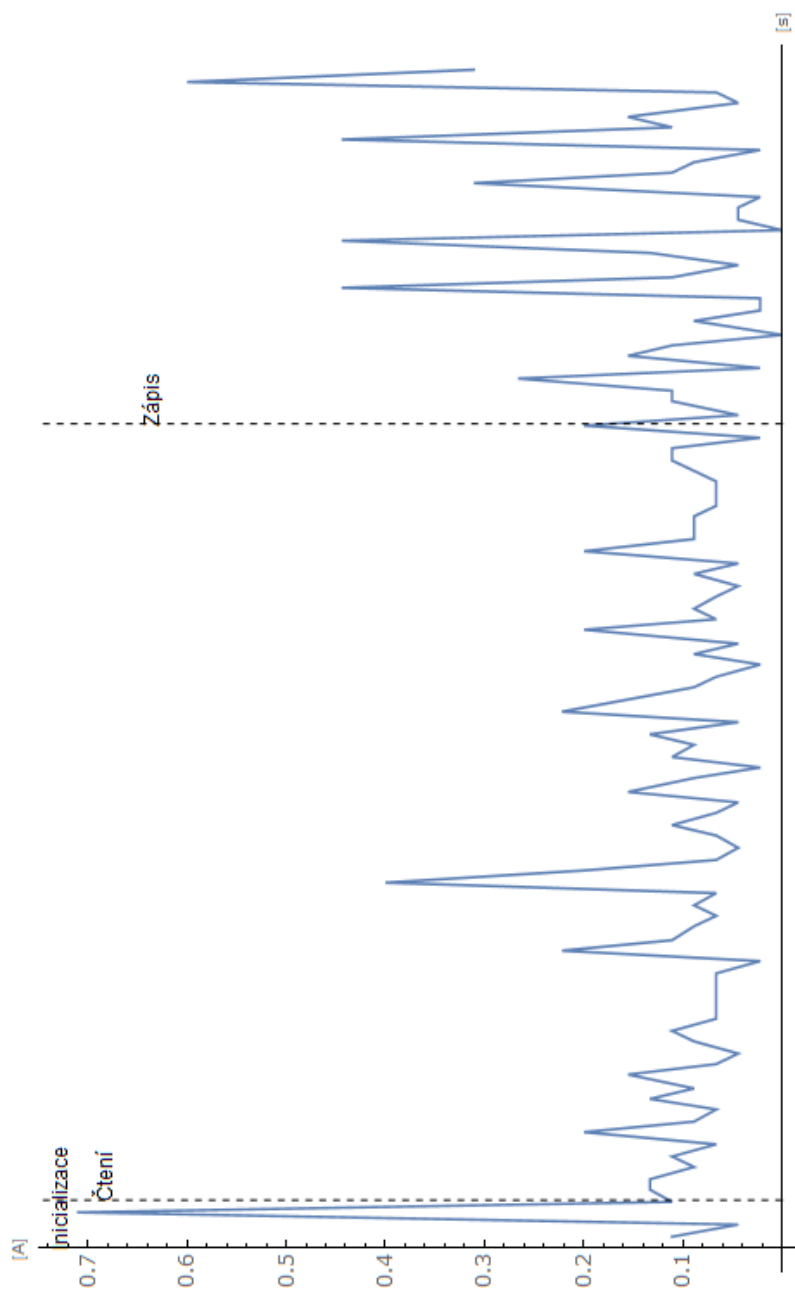
Podrobným testováním celého zařízení se podařilo identifikovat několik hlavních faktorů významně omezujících technické možnosti celého zařízení:

1. Doba přenosu je příliš závislá na statické době přenosu.
2. Se zvoleným GSM modulem nešlo dosáhnout LTE rychlosti sítě ani s jednou z použitých SIM karet.
3. Při případné duplikaci celého zařízení by bylo vhodné vybrat jiný adaptér na SD kartu, neboť tento je potřeba hardwarově opravit.

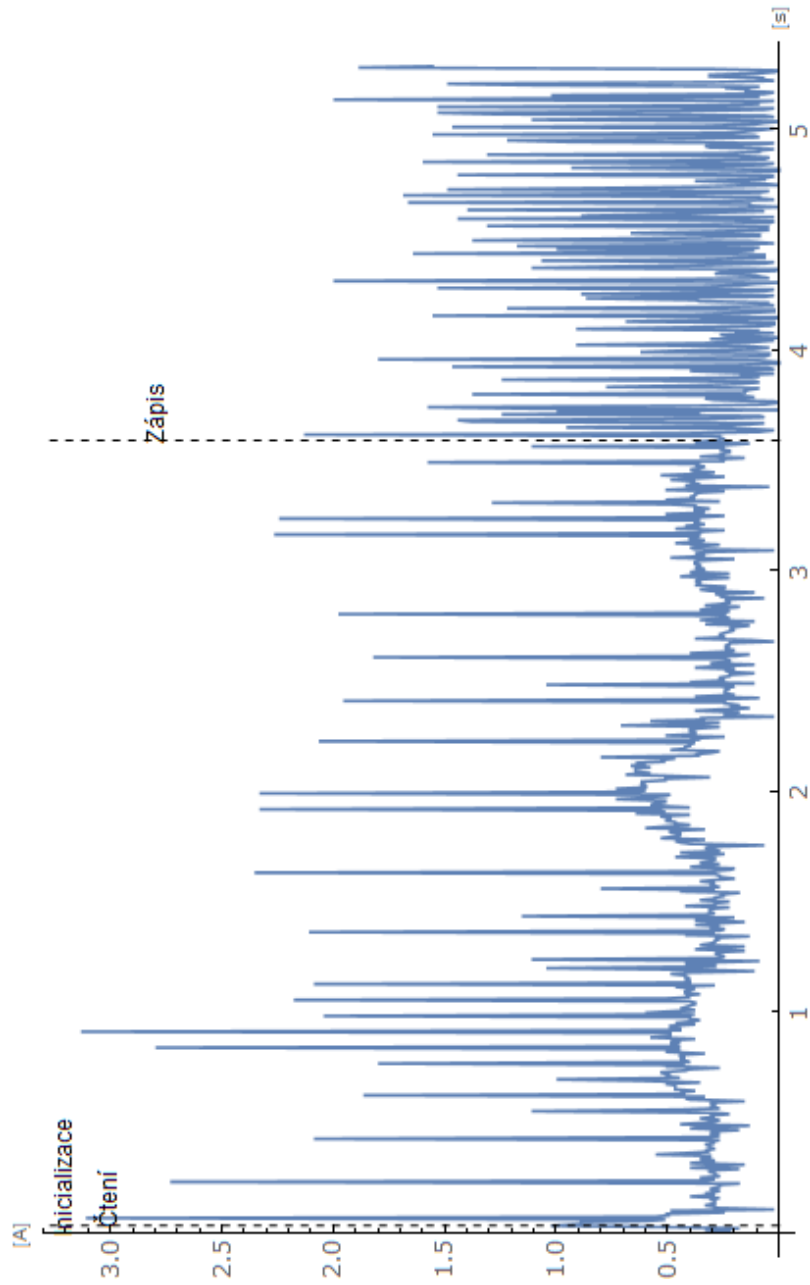
Díky identifikování úzkých hrdel je jasnější, na co se v případné budoucí práci zaměřit.



Obrázek 6.3: Proud procházející kamerou při zachycení snímku a zápisu na SD.



Obrázek 6.4: Proud procházející adaptérem na SD kartu při čtení a zápisu 10 kB souboru.



Obrázek 6.5: Proud procházející adaptérem na SD kartu při čtení a zápisu 100 kB souboru.

Závěr

Cílem práce bylo navrhnout a realizovat IoT zařízení umožňující sledování domácnosti a tento cíl byl naplněn. Součástí práce je popis metodologie měření a prezentované naměřené hodnoty spotřeby energie, propustnosti a latence. Pro demonstraci správné funkce sledovacího zařízení byl vytvořen jednoduchý server, na nějž jsou v pravidelných intervalech přenášeny fotografie. Sledovací zařízení se skládá ze čtyř hlavních komponent: řídicí mikrokontrolér Arduino, kamera, GSM modul a adaptér na SD kartu.

Práce začíná teoretickým úvodem popisující nezbytné znalosti a pokračuje analýzou a návrhem programového řešení. Pro tuto práci bylo napsáno více programů, ale hlavní jsou tři: program pro sledovací zařízení, server a program pro měření spotřeby. Mezi podpůrné programy patří klient pro testování serveru, skript v Mathematice pro analýzu naměřených hodnot procházejícího proudu a programy jednotlivých komponent pro měření proudu.

Součástí práce je kapitola realizace, v níž jsou mimo jiné popsány hlavní překážky při realizaci. Překážky bylo potřeba překonat pro úspěšné splnění cílů. Také je ve stejné kapitole blíže popsán skript, použitý pro analýzu naměřených hodnot a pro tvorbu grafů, které jsou prezentovány v kapitole testování.

Kapitola testování popisuje kroky pro testování funkcionality sledovacího zařízení. Zároveň je v dané kapitole analýza spotřeby energie, latence a propustnosti.

Z analýzy spotřeby energie při odesílání GSM modulu v tabulce 6.2 vyplývá, že je úspornější poslat jeden větší soubor, než odeslat více menších souborů o stejné velikosti. Toto je způsobeno především spotřebou během inicializačních fází, ale také například menším množstvím odeslaných hlaviček. Ze stejné tabulky lze odvodit, že je potřeba poslat alespoň 100 kB soubor pro dosažení alespoň 90 % pro spotřebu energie během odesílacích fází vůči inicializaci.

Pro účel efektivního využití energie je pomocí naměřených hodnot spotřeby energie také vypočítána výdrž baterie. V případě, kdy zařízení pošle fotografii, a poté hodinu čeká, by 9.6 V 2500 mAh baterie vydržela až 16 dní,

při započítání spotřeby GSM modulu, adaptéru na SD kartu a kamery.

Vytvořené sledovací zařízení lze využít například na chatě, pro kontrolu, jestli na ní nespádl strom a podobně. Metodologie měření a analýza naměřených hodnot mohou být použity i při tvorbě jiných IoT zařízení pro stejný, a nebo podobný účel.

Literatura

- [1] S. Heath, *Embedded systems design*, 2nd ed., pp. 1-5, 2002.
- [2] M. Rouse, “internet of things (IoT)”, <https://internetofthingsagenda.techtarget.com>, 2019.
- [3] M. Wedd, “What Is Cellular IoT? “
,<https://www.iotforall.com/what-is-cellular-iot/>, 2018.
- [4] Y. Hwang, “Cellular IoT Explained: NB-IoT vs. LTE-M vs. 5G and More”, <https://medium.com/iotforall/cellular-iot-explained-nb-iot-vs-lte-m-vs-5g-and-more-8f26496df5d4>, 2018.
- [5] S. Kumar, S. Akashe, “Implementation of GSM Based Security System with IOT Applications”, *International Journal of Computer Network and Information Security(IJCNIS)*, Vol.9, No.6, pp.13-20, 2017.DOI: 10.5815/ijcnis.2017.06.02.
- [6] N. Balasubramanian, A. Balasubramanian and A. Venkataramani, “Energy Consumption in Mobile Phones: a Measurement Study and Implications for Network Applications“. In *IMC*, 2009.
- [7] S. K. Gharghan, “Energy-Efficient Remote Temperature Monitoring System for Patients Based on GSM Modem and Microcontroller”, *Journal of Communications* Vol. 12, No. 8, August 2017.
- [8] K. Aziz, S. Tarapiah, S. H. Ismail, and S. Atalla, “Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies,” in *Proc. 3rd IEEE MEC International Conference on Big Data and Smart City*, Muscat, Oman, 15-16 March 2016.
- [9] A. Sachan, “GSM Based Automated Embedded System for Monitoring and Controlling of Smart Grid”. *International Journal of Electrical*,

Computer, Energetic, Electronic and Communication Engineering Vol:7, No:12, 2013.

- [10] M. Lauridsen, L. Noel, T. B. Sorensen, and P. Mogensen, "An Empirical LTE Smartphone Power Model with a View to Energy Efficiency Evolution". Intel Technology Journal, Volume 18, Issue 1, 2014.
- [11] K. Gye, K. Suki, H. Yeong. "An Adjustable Power Management for Optimal Power Saving in LTE Terminal Baseband Modem". Consumer Electronics, IEEE Transactions on. 55. 1847 - 1853. 10.1109/TCE.2009.5373741. 2009.
- [12] A. Koc, S. Jha, R. Vannithamby, M. Torlak. "Device Power Saving and Latency Optimization in LTE-A Networks Through DRX Configuration". Wireless Communications, IEEE Transactions on. 13. 2614-2625. 10.1109/TWC.2014.031914.131298. 2014.
- [13] Adam Fendelman, "What Is GSM in Cellular Networking?" [Online]. <https://www.lifewire.com/definition-of-gsm-578670>. Accessed [16. 4. 2020].
- [14] J Chavis, "What is a GSM Network?" [Online]. Available: <https://www.wisegeek.com/what-is-a-gsm-network.htm>. Accessed [16. 4. 2020].
- [15] D. BADER, G. and B. WOLFE, "What is a SIM card and what does it do?" [Online]. <https://www.imore.com/sim-card>. Accessed [16. 4. 2020].
- [16] E-Tinker, "Mobile Communication: What are GSM, GPRS, EDGE, UMTS, HSPA and LTE?". [Online]. Available: <https://www.askingbox.com/info/mobile-communication-what-are-gsm-gprs-edge-umts-hspa-and-lte>. [Accessed: 16. 2. 2020].
- [17] IBM Knowledge Center , "TCP/IP protocols" [Online]. https://www.ibm.com/support/knowledgecenter/en/ssw_aix_71/network/tcpip_protocols.html. Accessed [17. 4. 2020].
- [18] M. Bradley, "Packet Size: Network MTU Vs. Maximum TCP". [Online]. Available: <https://www.lifewire.com/definition-of-mtu-817948>. [Accessed 24. 2. 2020].
- [19] Arduino, "ArduinoIDE" [Online]. <https://www.arduino.cc/en/Main/Software>. Accessed [17. 4. 2020].
- [20] Arduino, "Sketch build process" [Online]. <https://arduino.github.io/arduino-cli/sketch-build-process/>. Accessed [17. 4. 2020].

-
- [21] Microsoft, “Microsoft Visual Studio” [Online].
<https://visualstudio.microsoft.com>. Accessed [17. 4. 2020].
- [22] Vim, “Vim the editor” [Online]. <https://www.vim.org>. Accessed [17. 4. 2020].
- [23] Wolfram, “Wolfram Mathematica” [Online]. Available:
<https://www.wolfram.com/mathematica/>. [Accessed: 18. 2. 2020].
- [24] Wolfram Language, “Programming with Built-in Computational Intelligence” [Online]. <https://www.wolfram.com/language/>. Accessed [11. 5. 2020].
- [25] Arduino, “ARDUINO MEGA 2560 REV3”. [Online]. Available:
<https://store.arduino.cc/arduino-mega-2560-rev3>. [Accessed: 29. 1. 2020].
- [26] DFrobot, “Module specification - SIM7000E Arduino NB-IoT/LTE/GPRS/GPS Expansion Shield DFR0572”. [Online].
Available: wiki.dfrobot.com/SIM7000_Arduino_NB-IoT_LTE_GPRS_Expansion_Shield_SKU_DFR0505_DFR0572. [Accessed: 30. 1. 2020].
- [27] ArduCAM, “Arducam 5MP Plus OV5642 Mini Module” [Online].
<https://www.arducam.com/product/arducam-5mp-plus-spi-cam-arduino-ov5642/>. Accessed [20. 4. 2020].
- [28] SIMCom, “SIM7000 Series_AT Command Manual_V1.03”. [Online].
Available: https://simcom.ee/documents/SIM7000E/SIM7000%20Series_AT%20Command%20Manual_V1.03.pdf. [Accessed: 11. 2. 2020].
- [29] skyfly, “Kapacita baterie”. [Online]. Available:
<https://skyfly.cz/pristroj/slovník/kapacitabat.htm>. [Accessed: 14. 5. 2020].
- [30] Patrick Zandl, “Jaká je kvalita GPRS připojení?”. [Online]. Available:
https://www.idnes.cz/mobil/tech-trendy/jaka-je-kvalita-gprs-pripojeni.A040513_5258077_mob_tech. [Accessed: 16. 5. 2020].
- [31] R. Stuhlfauth, “GSM data transfer capabilities”. [Online]. Available:
http://read.pudn.com/downloads161/ebook/733562/GSM/GSM_chap8.pdf.
[Accessed 19. 2. 2020].
- [32] someuser, “Re: SOLVED. Nrf24 (Mirf lib) + Micro SD-card works OK together” [Online]. <https://forum.arduino.cc/index.php?topic=360718.msg2942160#msg2942160>. Accessed [30. 4. 2020].

- [33] ArduCAM, “Arducam host app” [Online]. https://github.com/ArduCAM/Arduino/tree/master/ArduCAM/examples/host_app. Accessed [14. 5. 2020].

Seznam použitých zkratek

IoT Internet of Things

GSM Global System for Mobile Communications

2G Second-generation cellular network

3G Third-generation cellular network

SIM Subscriber Identity Module

IDE Integrated development environment

VS Microsoft Visual Studio

SMS Short Message Service

PIN Personal Identification Number

SD Secure Digital

SPI Serial Peripheral Interface

LTE Long-Term Evolution

NB-IoT Narrowband IoT

UART Universal Asynchronous Receiver/Transmitter

LCD Liquid-crystal display

ARM Advanced RISC Machine

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	src		
		implzdrojové kódy implementace
		thesiszdrojová forma práce ve formátu \LaTeX
	text	text práce
		thesis.pdftext práce ve formátu PDF