ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra měření

Studijní program: Otevřená informatika

Studijní obor: Internet věcí

Flexibilní modul pro domácí automatizaci

Flexible module for home automation

BAKALÁŘSKÁ PRÁCE

Autor: Patrik Vele

Vedoucí práce: doc. Ing. Jiří Novák, Ph.D.

Praha 2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vele** Jméno: **Patrik** Osobní číslo: **465899**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra měření**

Studijní program: **Otevřená informatika**

Studijní obor: **Internet věcí**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Flexibilní modul pro domácí automatizaci

Název bakalářské práce anglicky:

Flexible Module for Home Automation

Pokyny pro vypracování:

Navrhněte a implementujte technické a programové vybavení modulu pro domácí automatizaci s následujícími vlastnostmi:
- hlavní funkcí modulu bude spínání spotřebičů napájených z rozvodné sítě 230 V,
-.modul musí poskytovat rozhraní pro vzdálenou konfiguraci a ovládání, ideálně WiFi,
- zvolená platforma by měla umožnit další rozšiřitelnost hardware,
- programové vybavení musí umožnit jak manuální ovládání jednotlivých výstupů, tak provádění scénářů na základě časových rozvrhů a případně i dalších informací získaných buď místně připojenými senzory, nebo prostřednictvím sítě,
- musí být implementována vhodná forma zabezpečení.

Seznam doporučené literatury:

[1] Kocourek, P., Novák, J.: Přenos informace. Skripta ČVUT FEL, Praha 2003
[2] Tanenbaum, A. S., Wethetral, D.J.: Computer Networks, Prentice Hall 2010, ISBN-13: 978-0132126953
[3] Parziale, L. at al.: TCP/IP Tutorial and Technical Overview, IBM Redbooks, ISBN 9780738494685

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Jiří Novák, Ph.D., K 13138 - katedra měření

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

Datum zadání bakalářské práce: **12.02.2019** Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

doc. Ing. Jiří Novák, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání	Podpis studenta

Prohlášení

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

V Praze dne 24. 5. 2019 .....

Patrik Vele

Poděkování

Tímto bych rád poděkoval doc. Ing. Jiřímu Novákovi, Ph.D., vedoucímu mé bakalářské práce, za odborné vedení, za pomoc a cenné rady.

Abstrakt

Tato bakalářská práce se zabývá problematikou tvorby flexibilního modulu pro domácí automatizaci. Jsou zde popsány jednotlivě zvolené komponenty, jejich zapojení, následný návrh desky plošných spojů včetně závěrečného osazení a konstrukce. Druhá část je věnována tvorbě softwaru, který z modulu vytváří samostatně pracující inteligentní jednotku disponující připojením do internetu a možností vzdáleného ovládání.

Klíčová slova: Arduino, domácí automatizace, internet věcí

Abstract

This thesis deals with implementation of the flexible module for home automation. Chosen concept, components and arrangement are described as well as the design of the printed circuit board, including the final attachment and construction. The second part is dedicated to software development, which results in an individually working, intelligent unit featuring internet connection and the possibility of remote control of the module.

Keywords: Arduino, home automation, internet of things

Obsah

Seznam obrázků

1 Úvod

V poslední době se čím dál víc dostává do povědomí společnosti takzvaný internet věcí. Jedná se o koncept, který není přesně definován. Lze ho však chápat jako velké množství chytrých zařízení, které na základě jejich vzájemné komunikace poskytují nějakou službu. Příkladem by mohla být domácnost, která sama řídí vytápění. Tato domácnost se sama učí a zároveň dokáže inteligentně reagovat na danou situaci bez zásahu člověka. Díky těmto možnostem si internet věcí nalezl uplatnění i v jiných odvětvích. K integraci dochází například v oblasti zemědělství, automobilového průmyslu, měst, energetiky, ale i mnoha dalších. Jedná se tedy o rychle se rozvíjející technologii, která by se během několika následujících let měla výrazně rozšířit a stát se neodmyslitelnou součástí našich životů.

Osobně se o internet věcí, jeho možnosti a budoucnost velice zajímám. V současné době se také jedná o oborové zaměření v rámci mého studia. Mám tedy již teoretické i praktické zkušenosti, které bych však rád v budoucnu dále rozšiřoval. Dané téma jsem si zvolil především proto, že zde vidím dobrou příležitost pro moji seberealizaci v této oblasti.

Cílem práce je vytvoření modulu pro spínání spotřebičů napájených z rozvodné sítě 230 V. Modul bude disponovat bezdrátovým připojením do internetu, které přinese především dvě hlavní výhody. První z nich bude možnost vzdáleného ovládání pomocí libovolného zařízení připojeného do stejné sítě. Z bezpečnostních důvodů budou však tyto akce umožněny na základě autentizace pomocí hesla. Druhou výhodou bude schopnost rozhodování modulu v závislosti na informacích získaných na internetu. Protože se bude jednat o flexibilní modul, bude kladen velký důraz na široké spektrum použití a na případnou možnost budoucího rozšíření.

Samotná práce bude věnována části hardwarové i části softwarové. Postupně popíši jednotlivé mnou zvolené komponenty, následné schéma zapojení, vývoj základní desky, až po samotné osazení a dokončení modulu po hardwarové stránce. Poté se budu věnovat vývoji softwaru. Konkrétně tedy části programového spínání výstupů, vytvoření vlastní sítě, konektivitou do internetu, stahováním dat a implementací webového serveru včetně jednoduchého grafického rozhraní pro vzdálenou správu. Na závěr zmíním možnou budoucí rozšiřitelnost modulu.

2 Hlavní komponenty

2.1 Arduino UNO

Jako první bylo nezbytné vybrat součást, která bude mozkem celého modulu. Výsledná aplikace by neměla být nijak náročná na výpočetní výkon ani na složité grafické operace. Díky tomu jsem získal možnost výběru správného výpočetního modulu z velice širokého spektra. Konkrétně se jednalo například o mikrokontroléry od firmy STM, Arduino, případně známé Raspberry PI 3 či Raspberry PI Zero.

Osobně se již nějakou dobu zajímám o populární vývojové desky z rodiny Arduino, primárně o model Arduino UNO, který také vlastním. Z časových důvodů jsem se však s touto deskou k žádnému většímu projektu nedostal. Rozhodl jsem se tedy využít tuto příležitost a jako výpočetní prvek použít tento zmíněný model.

(obr. 1) Arduino UNO [1]

Základním kamenem celého modulu bude tedy vývojová deska Arduino UNO, konkrétně se bude jednat o její čínskou kopii. Tato deska disponuje 16 MHz procesorem Atmega328, 32 kB flash pamětí, šesti analogovými vstupními piny a čtrnácti digitálními vstupně výstupními piny. Všechny tyto piny pracují s pěti-voltovou logikou. Mezi specifické funkce, které jsou zde hardwarově implementovány patří sériová komunikace UART, dále je zde podpora sběrnic SPI a I2C [2].

Díky těmto specifikacím jsou splněny požadavky z hlediska výkonu, počtu pinů i dostatku paměti pro samotný program. Výhodou je kompatibilita s velkým množstvím rozšiřujících modulů. Jistou nevýhodou by mohla být omezená výbava z hlediska funkcí či výkonu v porovnání s jinými mikrokontroléry. Tato nevýhoda je však spojena s nízkou pořizovací cenou a malou spotřebou okolo 50 mA [2]. Protože se bude jednat o modul s nepřetržitým provozem, spotřeba zde hraje zásadní roli. Deska Arduino UNO bude tedy dobrou volbou.

2.2 I/O expander

Hlavní funkcí modulu bude spínání spotřebičů. Pro dostatečnou konektivitu jsem se k tomuto účelu rozhodl implementovat celkem 16 nezávislých spínacích výstupů. Arduino UNO však disponuje pouze 14 digitálními vývody [3]. K vyřešení tohoto problému jsem zvolil I/O expander PCAL6416APW, který bude řízen pomocí desky Arduino. Na tuto komunikaci využiji I2C sběrnici, což mi umožní ovládat 16 nezávislých výstupů pomocí dvojice pinů SDA a SCL [3].

2.3 Optotriak

Velice důležitá byla volba správného spínacího prvku. V této oblasti bude docházet k práci s nízkým napětím ze sítě, bude se tedy jednat o nejkritičtější část z hlediska bezpečnosti. Já jsem pro tento účel zvolil optotriak TLP360J, který galvanicky oddělí obvod modulu od síťových 230 V a díky jeho dobrým fyzikální vlastnostem, zejména prací i s 600 V a izolačním napětím 5000 V, si bezpečně poradí i s případnými napěťovými výkyvy [4].

2.4 ESP07

Modul bude mimo jiné disponovat jednoduchým webovým serverem. Hlavním úkolem tohoto serveru bude informovat uživatele o stavu jednotlivých výstupů modulu a zároveň umožnění ručního ovládání nebo změny nastavení. Z bezpečnostních důvodů budou všechny tyto úkony umožněny na základě autentizace pomocí hesla.

Arduino UNO bohužel neumožňuje konektivitu do internetu pomocí WiFi. Bylo tedy nezbytné využít samostatný prvek, který tuto práci zastane. Jako vhodný kandidát se jevil některý z modulů označených jako ESP, který by měl Arduinu umožnit bezdrátové připojení do internetu pomocí rozhraní UART. Každý z těchto modulů je založen na čipu ESP8266 a liší se navzájem pouze svou výbavou tvořenou například počtem pinů, typem antén nebo velikostí. Pro můj projekt jsem zvolil typ ESP07, který oproti své konkurenci vyniká třemi vlastnostmi. První z nich je dobře zpracovaná anténní část. Model disponuje keramickou anténou, která na rozdíl od antény

vytvořené na plošném spoji, slibuje lepší vlastnosti. Pokud by však tato malá keramická anténa nestačila, je zde konektor pro připojení antény externí. Druhou pozitivní vlastností je dostatek pinů, které dávají mnohem více možností pro případné rozšíření. Poslední vlastností je absence nadbytečných funkcí, což implikuje velice malé rozměry modulu a dobré rozložení pinů.

(obr. 2) ESP07 [5]

Hlavním výpočetním prvkem je jednojádrový 32 bitový procesor Tensilica L106. Ke klíčovým vlastnostem patří podpora šifrování typu WEP/TKIP/AES a zabezpečení typu WPA/WPA2. Co se bezdrátové komunikace týče, čip podporuje protokoly 802.11 b/g/n. Pro přenos informací s Arduinem použiji podporovanou sériovou linku UART, která i přes svou nízkou komunikační rychlost bude pro jednoduchý server dostačující. Na závěr je třeba zmínit, že napájecí napětí čipu ESP8266 je 3,3 V [6].

Díky ESP07 bude možné připojit modul k místní lokální síti a tím z něho vytvořit síťový prvek. Poté se na tento prvek bude možné připojit z libovolného zařízení spadajícího do stejné sítě. Výhodou této funkcionality bude pohodlná správa z libovolné části našeho domu, bytu či zahrady. Další velkou výhodou bude možnost umístění modulu i do hůře přístupných oblastí.

2.5 AC/DC napájecí zdroj

Modul bude obsahovat z hlediska napětí 2 různé části. První bude vysílací, do které bude spadat oblast okolo modulu ESP07 pracující s 3,3V napěťovou úrovní. Druhá bude pracovat s napětím 5 V. Do této oblasti budou patřit ostatní komponenty mého modulu.

Druhým kritériem pro volbu správného zdroje byl proudový odběr. Oblast, která bude pracovat s 3,3 V, bude obsahovat pouze jeden výkonný prvek, čímž bude čip ESP8266. Tento čip by měl mít proudový odběr okolo 70 mA, avšak při svém startu či době vysílání bude schopen krátkodobě odebírat proud přes 300 mA [7]. Arduino UNO sice disponuje pinem s výstupním napětím 3,3 V, ale maximální výstupní proud tohoto pinu je pouhých 40 mA [2]. Z tohoto důvodu není možné vysílací část napájet pomocí Arduina. Oblast, která bude pracovat s 5 V logikou, bude mít výkonných prvků více. Prvním z nich bude Arduino, které by při svém chodu mělo odebírat konstantní proud asi 50 mA [2]. Dále to bude všech 16 výstupů tvořených optotriaky, kde každým z nich bude moci protékat proud až 10 mA [4]. Pro 5V oblast tedy bylo nezbytné počítat se spotřebou 210 mA.

Poslední podmínkou bylo zvýšení přehlednosti pomocí eliminace vodičů. Té bude možné docílit přivedením napájení ze sítě společně s vodiči spotřebičů, které budou spínány. Napájecí zdroj bude tedy nezbytné zakomponovat na základní desku modulu.

Za těchto výše uvedených podmínek jsem zvolil 5W zdroj VTX-214-005-0503, který na svém výstupu obsahuje úroveň 5 V s garantovaným proudem 500 mA a úroveň 3,3 V s garantovaným proudem 757 mA [7]. Zdroj je dostatečně výkonný, aby byl schopný pokrýt proudový odběr obou částí a zároveň má patřičnou rezervu pro možné budoucí rozšíření modulu o další komponenty.

3 Návrh desky plošných spojů

Pro návrh desky plošných spojů jsem použil program Eagle od společnosti Autodesk. S tímto programem jsem se již v minulosti setkal, což mi umožnilo čerpat z dříve nabytých zkušeností.

3.1 Schéma zapojení

Vývoj schématu zapojení celého modulu probíhal v několika fázích. Nejdříve jsem samostatně navrhnul zapojení pro jednotlivé hlavní komponenty za použití pasivních součástek. Poté jsem tyto části postupně spojoval do jednoho celku. V této fázi jsem objevil jisté nedostatky, jejichž popis a následná řešení uvádím níže. Na závěr jsem do schématu zakomponoval některé pasivní součástky a podpůrné prvky.

3.1.1 Zapojení jednotlivých komponent

Tato kapitola bude věnována pouze dvojici z výše uvedených komponent, které na rozdíl od ostatních vyžadují složitější zapojení a použití pasivních součástek.

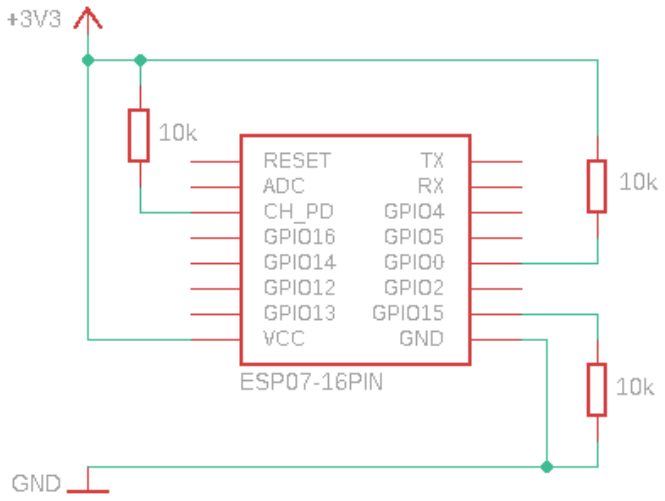
3.1.1.1 ESP07

Prvním krokem pro úspěšné zapojení byla znalost samotného prvku. Tato část nebyla tak snadná, jak jsem předpokládal. Na oficiálních stránkách firmy Espressif system, což je výrobce čipu ESP8266, jsem narazil na velké množství dokumentů. Mezi nimi se mi podařilo nalézt soubor s názvem ESP8266EX datasheet, který mě sice s čipem seznámil, ale z hlediska zapojení mne nijak neposunul. Nejdůležitější informace se nacházela v poznámce za čarou, kde byla drobná zmínka o možnostech bootování při startu čipu. Po zdlouhavějším hledání jsem nalezl podstatné informace.

Existuje několik způsobů bootování, kde každý je zvolen na základě kombinace logických úrovní na pinech GPIO15, GPIO0 a GPIO2. Standardní je bootování z flash paměti, kde se nachází již zabudovaný software od výrobce. Pro tuto možnost by měla být daná kombinace GPIO15 = 0, GPIO0 = 1 a GPIO2 = 1 [8]. Častokrát jsem se ale setkal se zapojením, kde byl pin GPIO2 nepřipojen, pravděpodobně díky internímu pull-up rezistoru. Protože jsem nenašel důvod pro jeho trvalé připojení, rozhodl jsem se tento pin nechat plovoucí. V případě budoucích komplikací s bootováním nebude problém provést dodatečné připojení k napájení.

Posledním důležitým pinem je CH_PD, který je také označován jako CH_EN, tedy chip enable. Jak již název napovídá, tak pro správnou funkci celého modulu je nezbytné, aby tento pin byl trvale nastaven na hodnotu logické 1 [8].

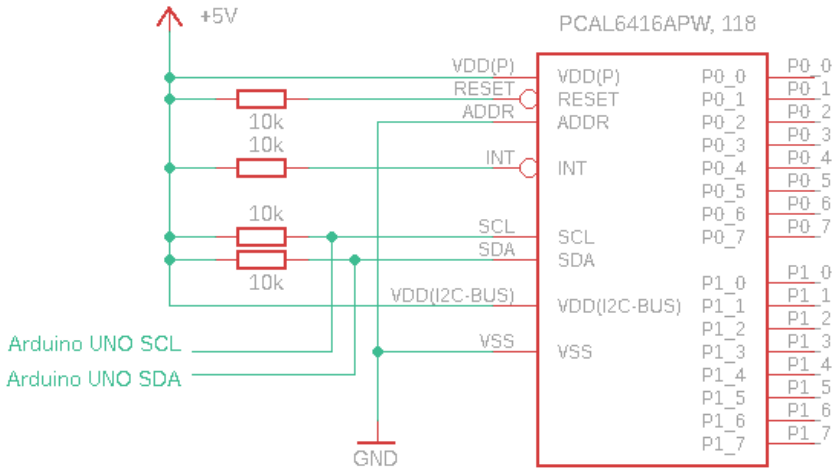
Všechny zmíněné piny připojím přes 10kΩ rezistory. Tato volba přispěje k jednoduchosti, protože se jedná o hodnotu, která bude využita i v jiné oblasti výsledného modulu.



(obr. 3) Schéma zapojení ESP07

3.1.1.2 I/O expander

Tato komponenta disponuje velice podrobnou dokumentací, která obsahuje mimo jiné i doporučená zapojení. Na základě těchto informací nebyl žádný problém při sestavení schématu zapojení a navrhnutí vhodných hodnot použitých součástek. Jedinou překážkou bylo, že tato komponenta se nenachází ve standardních knihovnách programu Eagle. Z tohoto důvodu jsem si s pomocí dokumentace součástku sám v programu vytvořil.



(obr. 4) Schéma zapojení I/O expanderu

Ze zapojení je patrné, že nebudu využívat piny RST a INT. Důvodem nevyužití hardwarového resetu je fakt, že všechny výstupy I/O expanderu budou pracovat jako spínače, které bude v případě potřeby možné resetovat softwarově pomocí I2C sběrnic. Samotný informační pin INT neposkytuje v této implementaci spínačů žádnou podstatnou výhodu. Dále je zde vyobrazen pin ADDR, díky kterému je umožněno připojit k I2C sběrnici až dvě různá zařízení typu slave se stejnou adresou. Tento pin obě zařízení odliší pomocí logické 1 a 0 [3].

3.1.2 Vzniklé komplikace

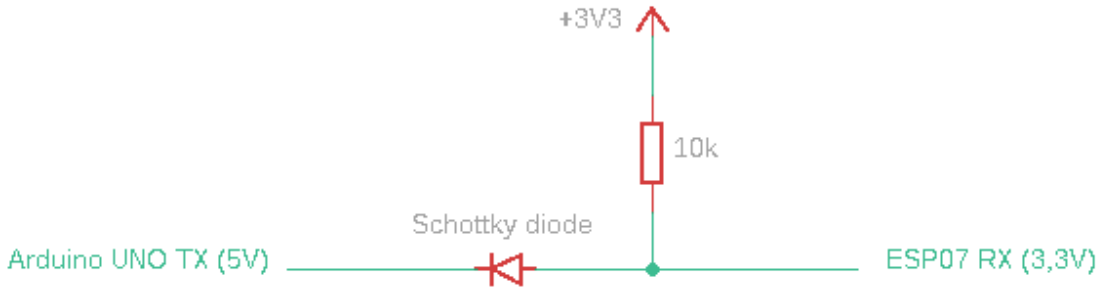
Při spojování jednotlivých komponent v jedno ucelené schéma jsem narazil na několik problémů. V této části se budu těmto vzniklým nedostatkům a jejich následným řešením věnovat.

3.1.2.1 Kompatibilita napěťových úrovní

Při představení jednotlivých komponent jsem mimo jiné zmiňoval, že Arduino UNO pracuje s 5 V a ESP07 s 3,3 V napěťovou úrovní [2] [6]. Pro umožnění jejich vzájemné komunikace přes UART bylo tedy nezbytné zaručit napěťovou kompatibilitu těchto zařízení.

Z pohledu Arduina jako přijímače nebylo třeba žádných úprav, protože logická 1 bude díky napěťovému rozsahu 3 až 5 V detekována správně [9].

Problém nastal na přijímacím pinu RX modulu ESP07, který nemá 5V toleranci [10]. Bylo tedy nezbytné změnit vysílací úroveň Arduina na 3,3 V. Možností na vyřešení tohoto problému jistě existuje velké množství. Osobně jsem k tomuto účelu zvolil zapojení s jedním pull-up rezistorem a Schottkyho diodou zapojenou v závěrném směru.



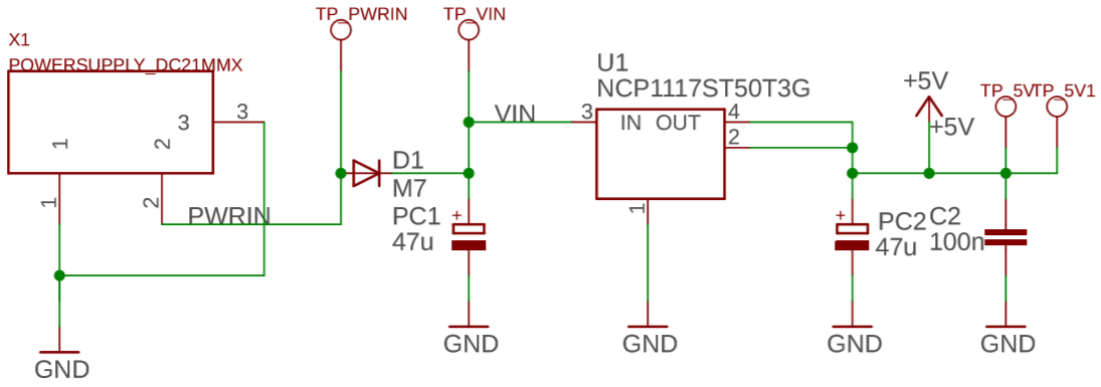
(obr. 5) Schéma zapojení obvodu pro kompatibilitu napěťových úrovní

Velkou výhodou tohoto zapojení je nízká energetická náročnost daná velice malým proudem protékajícím pinem Arduina do země v případě vstupní logické 0. Tento proud je ovlivněn velikostí pull up rezistoru, který jsem zvolil 10 kΩ. Pro představu hodnoty proudu lze využít známý Ohmův zákon, kdy po dosazení hodnot 3,3 V a 10 kΩ do vztahu I = U/R dostáváme proud 330 uA. Zdůraznil bych, že se jedná pouze o přibližnou hodnotu z důvodu zanedbání malého úbytku napětí na diodě a vedení. Druhou hlavní komponentou je Schottkyho dioda, kterou jsem zvolil na základě její schopnosti rychlého přechodu z propustného do nepropustného stavu. Díky této vlastnosti je možné diodu použít i v obvodech s frekvencí v jednotkách GHz, tedy pro UART komunikaci s maximální rychlostí 115200 bitů za sekundu bude tato komponenta naprosto dostačující.

3.1.2.2 Napájení desky Arduino UNO

Jedná se o model, který disponuje třemi možnostmi napájení. Prvním a nejznámějším způsobem je napájení přes USB, dále pak pomocí konektoru JACK nebo přes dvojici pinů VIN a GND [2]. V mém projektu se bude Arduino osazovat na navrhnutou základní desku podobně jako procesor u stolních počítačů. Z tohoto důvodu jsem zvolil napájení pomocí pinů VIN a GND, které na rozdíl od dalších dvou možností nevyžaduje žádnou kabeláž.

Problémem je, že tento způsob je určen primárně pro napájení z baterie. Výrobce zde proto přidal regulátor napětí, na kterém vzniká úbytek 2 V. Tedy na rozdíl od napájení přes 5 V USB je zde pro správnou funkci zapotřebí 7-12 V [2].



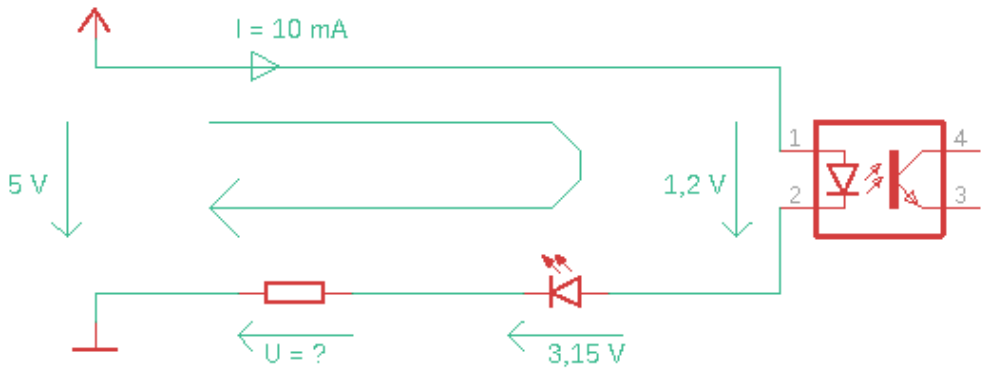
(obr. 6) Schéma zapojení napájecí části desky Arduino UNO [11]

Můj obvod obsahuje stabilní 5V zdroj, tedy regulátor napětí v tomto případě nemá žádné využití. Problém tedy vyřeším propojením vstupu a výstupu regulátoru, čímž odstraním jeho funkci. Poté by mělo být možné napájet Arduino přes VIN pouze 5 V a zároveň zaručit jeho správný chod.

3.1.2.3 Indikace sepnutého výstupu

Velmi důležitou vlastností tohoto systému bude indikace stavu jednotlivých výstupů. Ta bude implementována softwarově a poskytována za pomoci webového serveru. Pro případné selhání této softwarové části jsem se rozhodl zakomponovat ještě indikaci hardwarovou. Protože však I/O expander nedisponuje žádnými signalizátory stavu výstupů, použil jsem pro tento účel malé SMD LED diody v kombinaci s ochrannými rezistory, jejichž hodnotu jsem stanovil na základě informací o jednotlivých prvcích dané smyčky.

Proud protékající obvodem by měl být minimálně 10 mA, aby i v nejhorším případě došlo ke správné funkci triggerovací LED diody uvnitř optočlenu [4]. Druhá podmínka byla určena maximálním proudem 200 mA, který může zemnícím pinem I/O expanderu protékat [3]. Tedy pro případ sepnutí všech 16 výstupů zároveň jde o maximálně 12,5 mA pro každý z nich. Na základě těchto dvou omezení jsem odečetl z voltampérových charakteristik udávané úbytky napětí na prvcích a vytvořil následující schéma zapojení [12] [4].



(obr. 7) Schéma zapojení pro výpočet hodnoty rezistoru

Použitím druhého Kirchhoffova zákona, tedy součtem všech úbytků napětí ve smyčce, jsem získal napětí na rezistoru 0,65 V. Následně jsem použil Ohmův zákon, kdy jsem tuto hodnotu vydělil protékajícím proudem. Výsledkem byl odpor 65 Ω. Pro reálné řešení jsem zvolil rezistor s hodnotou 56 Ω, což je nejbližší nižší hodnota rezistoru, která se nachází v řadě E12. Tato volba povede ke zvýšení protékajícího proudu, čímž vznikne dostatečná rezerva pro případné změny fyzikálních vlastností obvodu, které mohou být způsobeny například dlouhodobým používáním.

3.1.3 Závěrečné úpravy

Ucelené schéma zapojení jsem na závěr obohatil o některé komponenty a úpravy, které celému modulu přidaly dané pozitivní vlastnosti.

3.1.3.1 Kondenzátory

Na obou napěťových částech obvodu se budou vyskytovat výkonové špičky. Tyto špičky budou způsobeny prudkými změnami proudu při spínání výstupů a také případným vysíláním modulu ESP07. Z tohoto důvodu jsem obě napěťové části doplnil o dvojici blokovacích kondenzátorů, které zajistí požadovanou stabilitu napájecího napětí při všech změnách odběru proudu. Dvojice se skládá z rychlého keramického kondenzátoru o kapacitě 68nF a tantalového kondenzátoru s kapacitou 47 uF.

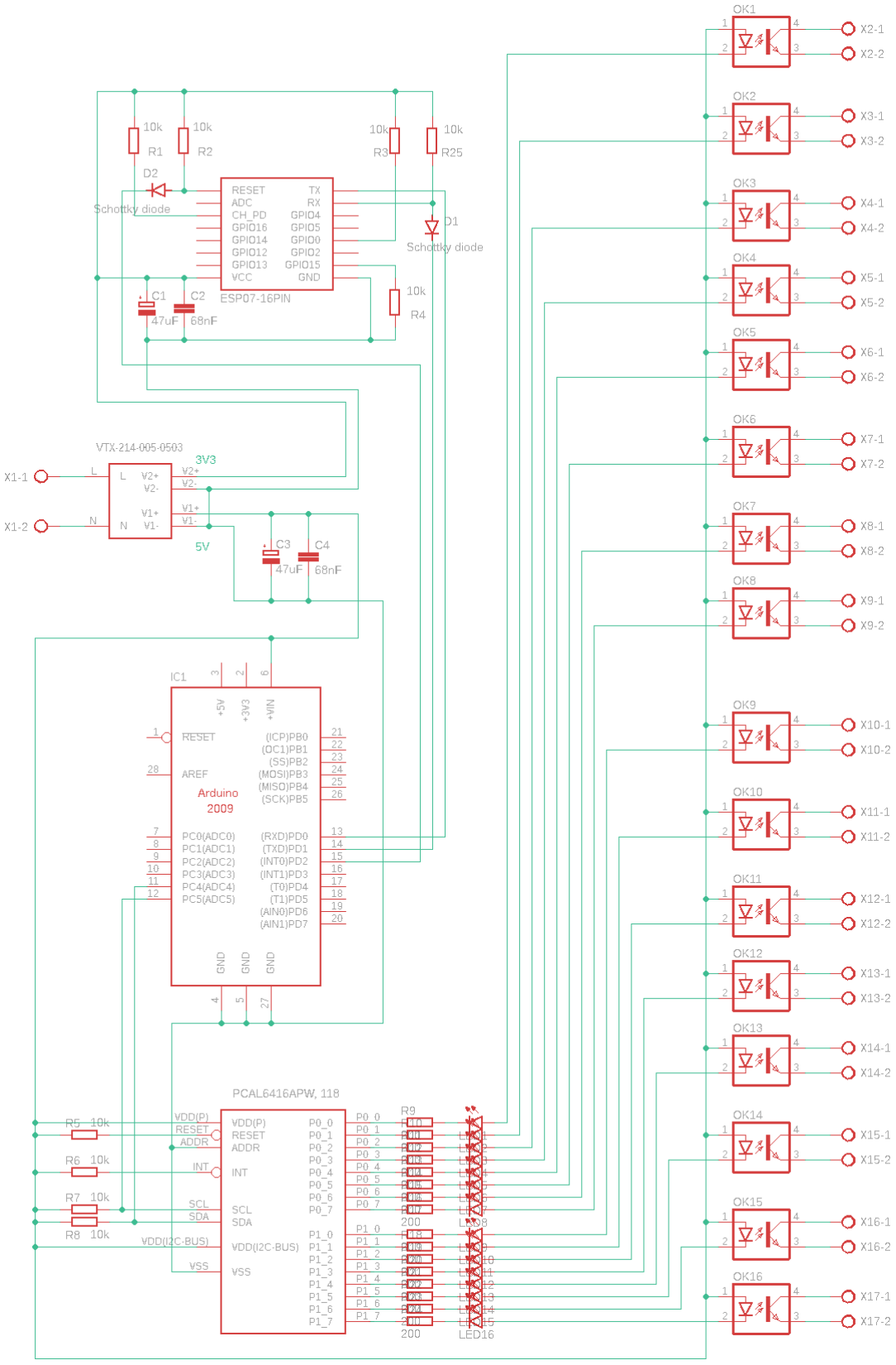
3.1.3.2 Pájecí svorkovnice

Modul by měl umožňovat rychlé a spolehlivé připojení spotřebičů napájených ze sítě. Pro zaručení těchto vlastností jsem pro každý výstup přidal dvoupólovou pájecí svorkovnici. Zároveň jsem jednu svorkovnici použil pro připojení napájecích vodičů.

3.1.3.3 Softwarový restart bezdrátové komunikace

Rozhodl jsem se k Arduinu připojit pin modulu ESP07, který je označen jako RST, čímž dostanu možnost případného softwarového restartu této WiFi části. Vzhledem k tomu, že potřebuji 3,3V logiku řídit pomocí 5 V, použiji zapojení pro zajištění napěťové kompatibility vstupu, které bylo popsáno výše.

3.1.4 Výsledné schéma zapojení



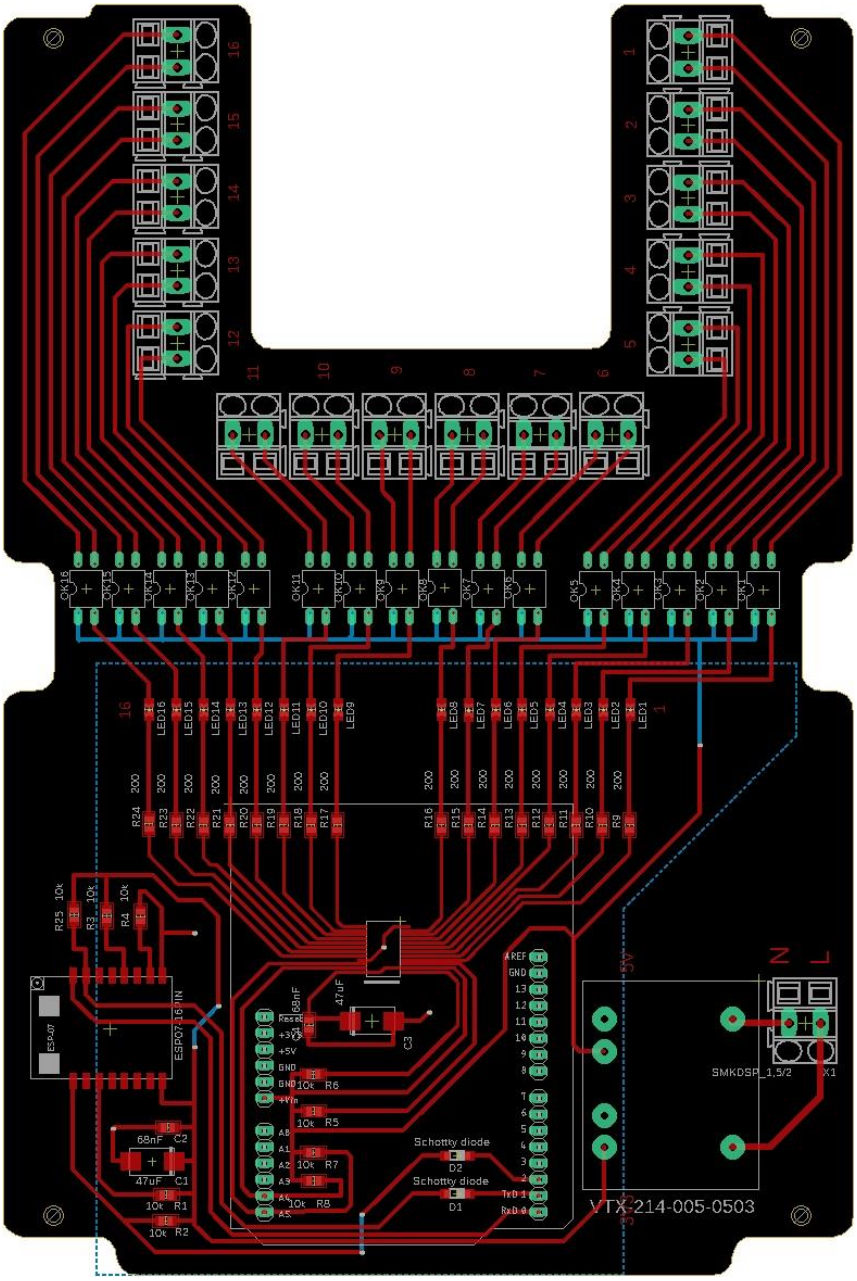
(obr. 8) Výsledné schéma zapojení

3.2 Návrh desky plošných spojů

Pro tento návrh desky plošných spojů byla použita 2 vrstvá architektura. Horní vrstva obsahuje většinu spojů a spodní vrstva je primárně zemnící, výjimečně se zde nachází krátké pomocné spoje pro vrstvu horní. Tyto pomocné spoje napomáhají k minimalizaci délky vodičů, čímž eliminují nežádoucí fyzikální vlastnosti, šetří materiál a zvyšují celkovou přehlednost.

Úvodní rozmístění jednotlivých komponent mi dalo představu o rozměrech výsledné základní desky. V této fázi bylo nezbytné vybrat vhodný box, kam bude deska následně umístěna. Už dříve mi bylo jasné, že výsledný produkt bude pravděpodobně připevněn na zdi či stropě. S tímto předpokladem jsem se rozhodl vybrat box větší, a z důvodu bezpečnosti přivést veškerou kabeláž vytvořeným otvorem na spodní straně. Pokud by tedy byly všechny vodiče spotřebičů přivedeny do jednoho místa ve zdi, do mého modulu by z viditelné části nic nevedlo. Posledním požadavkem byla volba nevodivého materiálu, který nebude bezdrátovou komunikaci nijak omezovat. Za výše uvedených podmínek jsem tedy vybral plastovou průmyslovou krabičku s označením 627-002, která má rozměry 222 krát 146 krát 56 mm a disponuje integrovanými distančními sloupky pro upevnění mé základní desky [13].

V této chvíli začala tvorba výsledného grafického návrhu. Upravil jsem tedy rozměry a tvar základní desky a začal na ni logicky umisťovat jednotlivé komponenty, které jsem propojoval vodivými cestami o šířce 0,4 nebo 0,6 mm. Po dokončení této fáze bylo nezbytné zkontrolovat a případně upravit vzdálenosti vodivých částí v oblastech s 230 V. Úpravy vedly také k eliminaci zemnící plochy, aby nedošlo k průrazu dielektrika a následnému zkratu. Poté jsem odstranil zemnící plochu i z oblasti anténní částí modulu ESP07, která by omezovala bezdrátovou komunikaci. Na závěr jsem ještě přidal čtyři otvory pro uchycení desky ve vybrané krabičce.

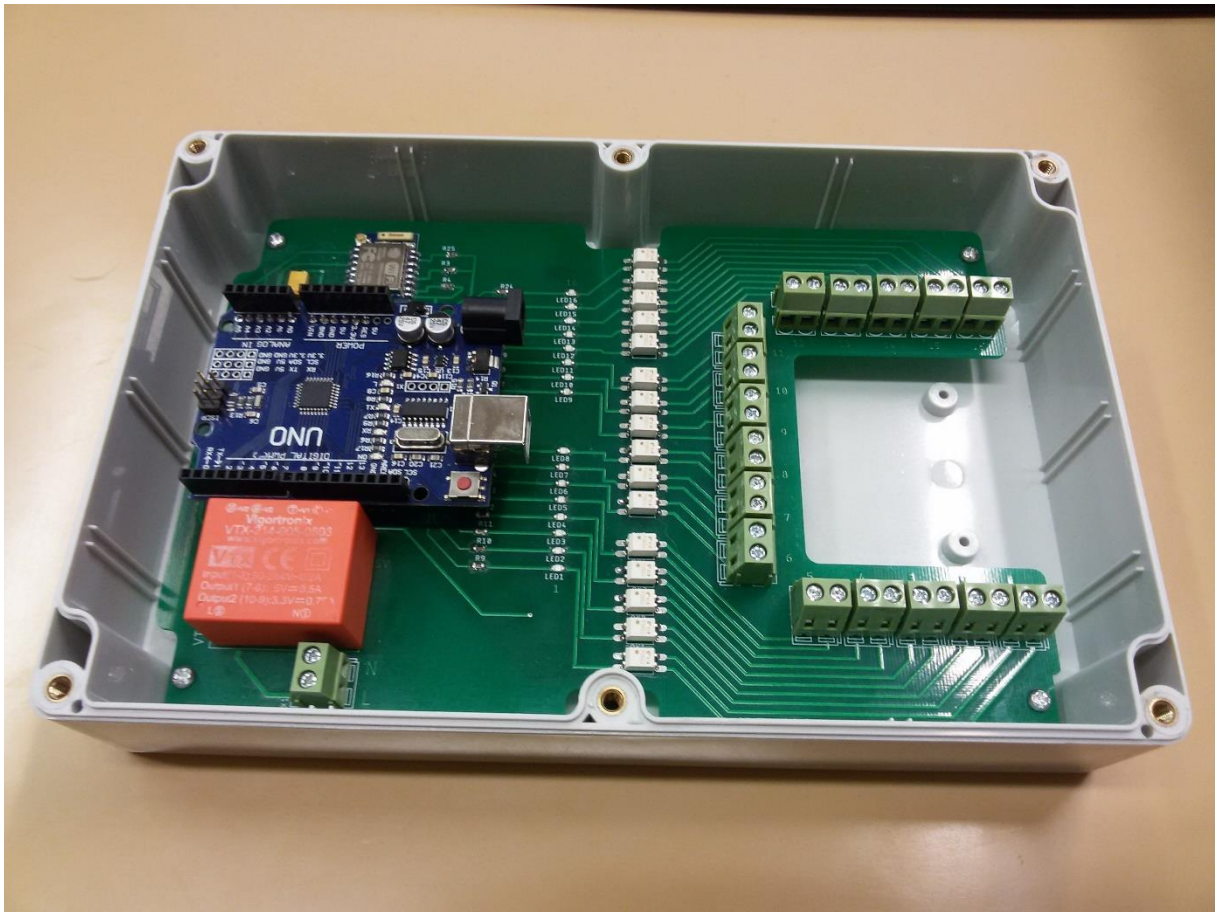


(obr. 9) Grafický návrh desky plošných spojů

Jak je z obrázku patrné, tak většina komponent bude připájena přímo k navrhnuté základní desce. Výjimkou bude pouze Arduino UNO, které bude osazeno pomocí oboustranných kolíků připájených k jeho pinům a dutinkových lišt připevněných k navrhnuté desce. Tato konstrukce pro jeho snadné odpojení přináší řadu výhod. První z nich je možnost rychlé výměny v případě poruchy. Dále jde o dobré využití prostoru díky umístění nad ostatními součástkami. Největší výhodou je ale možnost pohodlného a snadného nahrání programu. Pro tento případ není nutné nijak manipulovat s celým modulem, ale pouze s deskou Arduino UNO.

4 Praktická konstrukce modulu

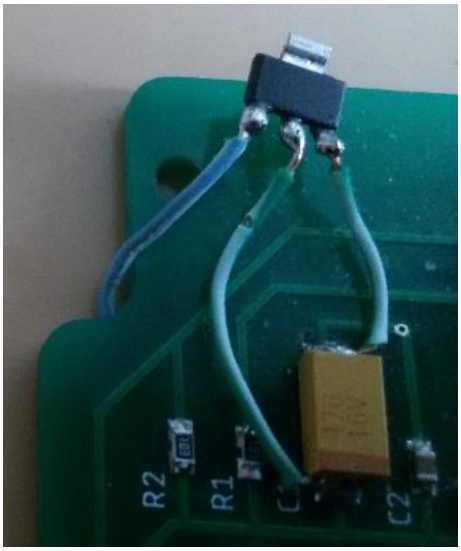
Na základě grafického návrhu jsem si nechal desku vyrobit zahraniční firmou. Následně jsem ji osadil všemi vybranými součástkami a umístil do předem připravené krabičky.



(obr. 10) Výsledný modul

5 První spuštění

Tato fáze měla především otestovat funkčnost jednotlivých komponent a správnost celkového zapojení. Ihned po startu byl odhalen problém se startem modulu ESP07. Po krátkém elektrotechnickém měření byla chyba detekována ve zdroji. Konkrétně se jednalo o vadnou 3,3V část, která je schopna dodávat zhruba desetkrát menší proud, než je udáváno výrobcem. Rozhodl jsem se tedy použít jako dočasné řešení lineární regulátor napětí, který provede konverzi ze vstupních 5 V na požadované napětí 3,3 V. Touto úpravou jsem celý modul uvedl do provozu a připravil ho tak na následný vývoj softwaru bez zbytečného čekání na reklamaci součástky.

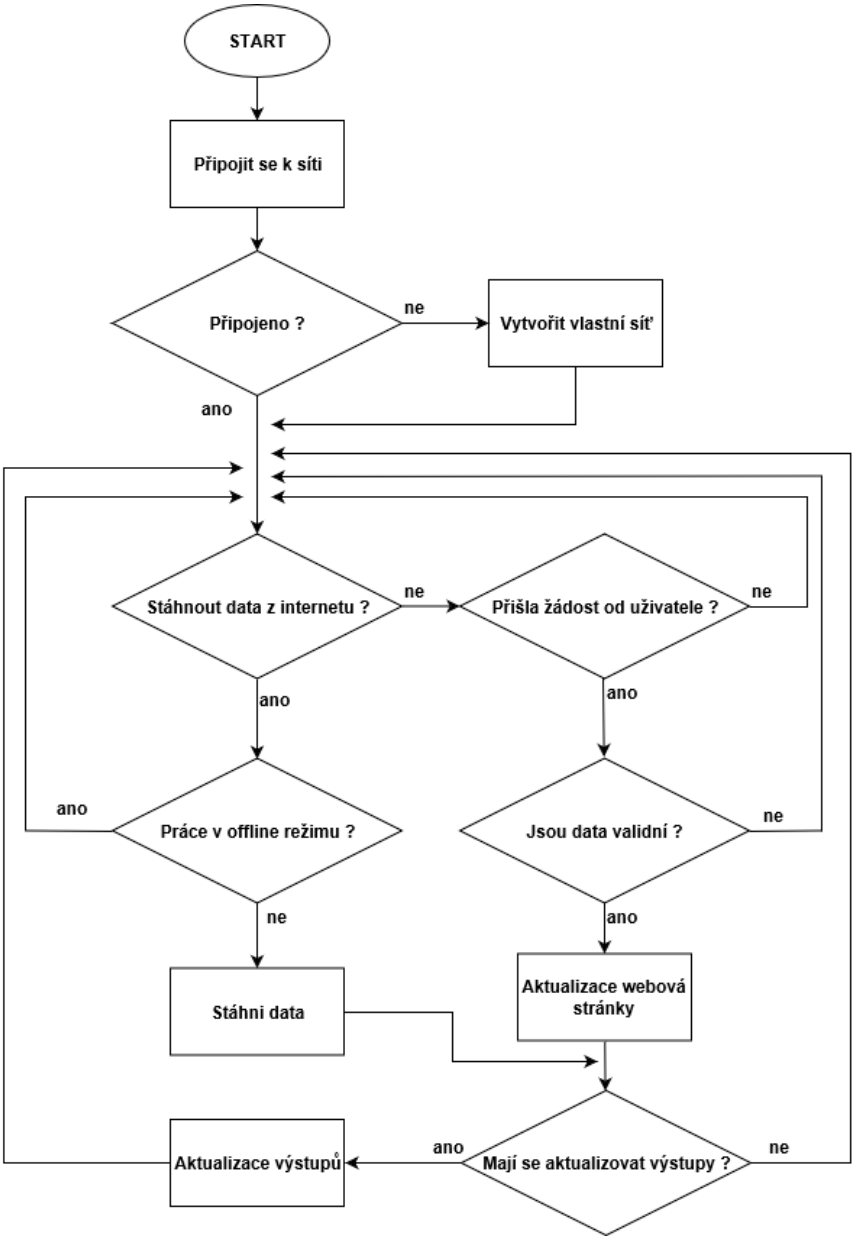


(obr. 11) Dočasné řešení s použitím lineárního regulátoru napětí

Bohužel po komunikaci s prodejcem mi bylo sděleno, že se jedná o vadu celé výrobní série tohoto produktu. Nezbývalo mi tedy než toto dočasné řešení ponechat prozatím i jako konečné.

6 Software

Jedná se o modul, který by měl umožňovat široké spektrum použití, proto jsem se snažil software vytvořit z několika základních částí, kde u každé z nich byl kladen důraz především na univerzálnost.



(obr. 12) Blokové schéma programu

Po vytvoření tohoto obecného konceptu jsem aplikací drobných úprav provedl specifikaci pro konkrétní použití, kterým bude ovládání okenních žaluzií.

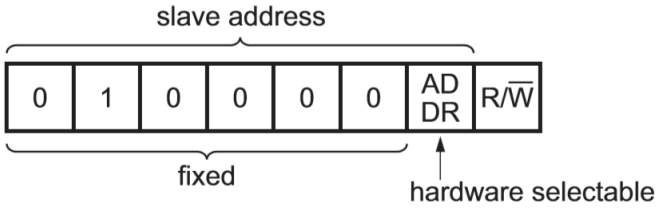
6.1 Zajištění komunikace s periferiemi

Prvním krokem při tvorbě softwaru bylo zajištění komunikace mezi mikrokontrolérem Arduino UNO a jeho periferiemi, tedy modulem ESP07, I/O expanderem a sériovým monitorem pro kontrolní výpisy.

6.1.1 I/O expander

Jak je již známo, tak komunikace s touto komponentou bude probíhat po I2C sběrnici. Pro tento účel jsem použil knihovnu *Wire.h*, která je jednou ze standartních knihoven podporovaných oficiální stránkou Arduino. Ta obsahuje veškeré potřebné funkce pro implementaci komunikace po I2C sběrnici [14].

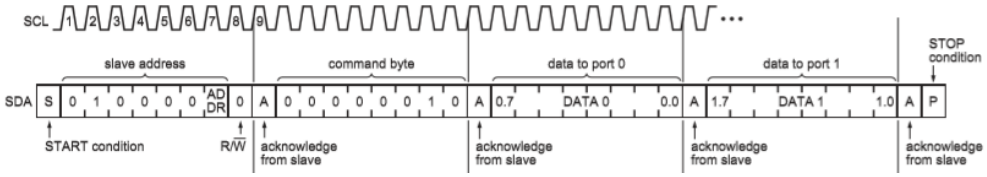
Hlavním parametrem pro navázání komunikace je fyzická adresa zařízení. Ta se v případě I/O expanderu skládá z fixní části danou výrobcem a části hardwarově nastavitelnou pinem ADDR. Díky tomu je možné připojit k jedné I2C sběrnici až dvě zařízení se stejnou fixní adresní částí, protože obě zařízení budou odlišena právě tímto jedním bitem [3].



(obr. 13) Fyzická adresa I/O expanderu

Osobně jsem pin ADDR připojil k zemi. Z obrázku je tedy patrné, že adresa mého I/O expanderu je hexadecimálně reprezentovaná hodnota 0x20.

Další podstatnou informací je způsob přenosu dat, který lze popsat čtyřmi fázemi. Začátek přenosu určený adresou, za ním následuje některý z příkazů udávaných výrobcem, poté samotná data a na závěr ukončení přenosu [3].



(obr. 14) Komunikace s I/O expanderem po I2C sběrnici

Na základě těchto informací jsem vytvořil kód, který zaručí obecnou komunikaci s I/O expanderem. Nejprve bylo nezbytné inicializovat I2C sběrnici na Arduinu pomoci funkce *Wire.begin()*. Následný čtyř fázový přenos jsem byl schopen vykonat pomocí vybrané trojice funkcí z knihovny *Wire.h* [14]. Pro start přenosu jsem použil funkci *beginTransmission()*, která jako parametr přijímá adresu zařízení pro komunikaci. Předal jsem ji tedy fyzickou adresu mého I/O expanderu, konkrétně 0x20. Následuje funkce *write()*, která odešle data o velikosti jednoho bajtu jí předaná. Nejprve jsem tedy odeslal instrukci 0x02 pro umožnění nastavení pinů a následně jsem tyto piny nastavil jako výstupy dvojitým odesláním hodnoty 0x0. I/O expander disponuje horními a dolními 8 piny, tedy pro nastavení každé z obou osmic bylo zapotřebí odeslat jeden samostatný bajt [3]. Pro ukončení vysílání jsem použil funkci *endTransmission(),* jejíž návratová hodnota poskytuje informaci o úspěšnosti přenosu. Závěrem je třeba zmínit, že standardně jsou všechny výstupní piny ve stavu 1, což v mém případě znamená nesepnutý stav [3]. Nemělo by tedy docházet k neočekávaného chování výstupů.

Po dokončení nastavení pinů následovala část kódu pro změnu jejich logických úrovní. Jednalo se o identický kód, který se lišil pouze instrukcí a následnými daty. V tomto případě jsem použil instrukci 0x6, která umožnila nastavení pinů [3]. V této chvíli I/O expander očekával dva bajty, jejichž bitové hodnoty budou nastaveny na jednotlivé piny [3]. Vytvořená konstrukce modulu je navržena pro spínání výstupu pomocí logické 0. Pokud tedy odešlu dvakrát hodnotu 0x0, všechny výstupy budou sepnuty. Pro opačný případ použiji hodnotu 0xFF.

6.1.2 Modul ESP07

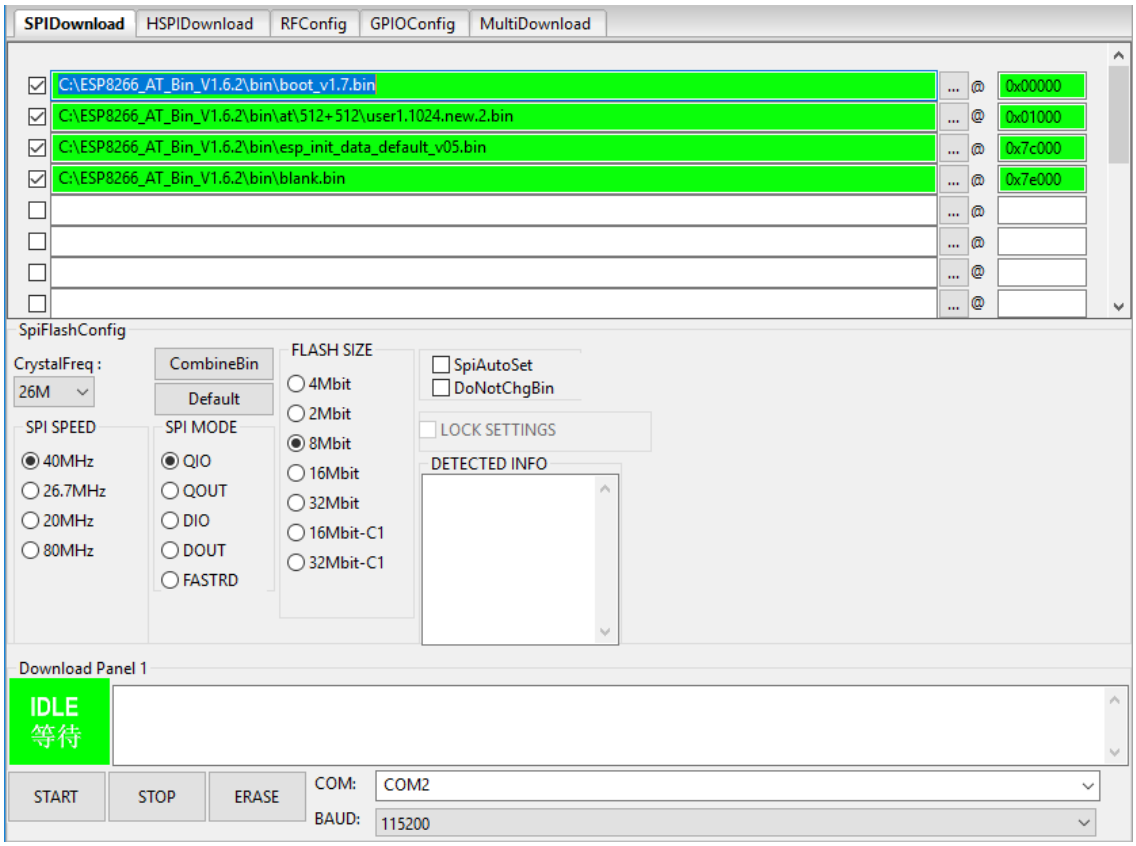
Modul ESP07 by měl vykonávat přijaté AT příkazy po sériové lince. Problémem je, že Arduino UNO disponuje pouze jedním rozhraním UART, který je využíván pro sériový monitor. Protože není možné provozovat obě tyto periferie na jedné sériové lince, použiji knihovnu *SoftwareSerial.h* pro nasimulování druhého UARTu. Již ze zapojení je patrné, že hardwarový UART bude používán pro komunikaci s modulem ESP07. Důvodem této volby je fakt, že na rozdíl od softwarové verze dosahuje několikanásobně větší přenosové rychlosti. Pro sériový

monitor budou vyhrazeny piny 4 a 5. K těmto pinům poté připojím počítač přes UART-USB převodník a jako zobrazovač použiji software PuTTY.

Základem při práci se sériovou komunikací je nastavení správné komunikační rychlosti označované jako baud rate. Modul ESP07 by měl pracovat na rychlosti 115200 [6]. Po nalezení správné komunikační rychlosti pomocí programu PuTTY jsem otestoval funkčnost modulu příkazy *AT, AT+RST* a *AT+GMR* [15]. Následně jsem se rozhodl zkusit změnit rychlost modulu z 115200 na 9600 pomocí příkazu *AT+IPR=9600*. Na základě přijaté odpovědi „OK“ jsem provedl restart a přenastavení komunikační rychlosti v terminálu. Od této chvíle modul přestal reagovat na veškeré příkazy a jeho jedinou aktivitou po startu byl výpis nečitelných znaků na všech otestovaných komunikačních rychlostech. Jako jedinou možností opravy se jevilo nahrání poslední verze AT firmwaru.

Nejprve jsem si na komunikační rychlosti 74880 nechal vypsat modulem informace o jeho konfiguraci. Konkrétně mě zajímala hodnota flash paměti, která je v mém případě 8 Mbit. Poté jsem z oficiálních stránek výrobce stáhnul poslední verzi AT firmwaru pro modul s touto velikostí flash paměti a následně také software Flash download tool pro samotnou instalaci firmwaru. Pro instalaci bylo nezbytné provést změnu bootovacího modu ESP07 připojením pinu GPIO0 na zem. Na závěr jsem modul ESP07 propojil s počítačem pomocí UART-USB převodníku [16].

Následná instalace vyžadovala podrobné nastavení a specifika. Mezi vyplňované položky patřila například frekvence krystalu, která je 26 MHz, velikost již zmíněné 8 Mbit flash paměti nebo konfigurace SPI sběrnice, kterou jsem pro využití maximálního výkonu nastavil na 40 MHz [6]. Zároveň jsem pro tuto sběrnici zvolil pracovní mód Quad Input Output, který by měl díky čtyřem pracovním linkám zaručit rychlejší komunikaci s flash pamětí. Nejvíce problematické bylo určení adres paměti společně v kombinaci s volbou správného souboru, který se do daného adresního prostoru měl nahrát. Informace o tomto nastavení se nachází v textovém dokumentu, který je přiložen u AT firmwaru. Bohužel s použitím těchto informací byla instalace neúspěšná. Avšak následným dohledáním a vyzkoušením jiných hodnot se mi po několika pokusech podařilo instalaci provést úspěšně [17]. Tato výsledná konfigurace je zachycena na obrázku níže. Poslední položkou byla volba baud rate 115200, společně se správným COM portem, přes který následně docházelo k samotné instalaci.



(obr. 15) Konfigurace pro instalaci firmwaru do ESP07

Po úspěšné instalaci jsem odpojil pin GPOI0 a modul restartoval. Následně jsem pomocí programu PuTTY přeinstalované ESP07 opět otestoval nejrůznějšími AT příkazy [15]. Nyní bylo již chování stabilní a spolehlivé.

K zajištění komunikace s mikrokontrolérem Arduino UNO jsem použil některé funkce z knihovny *Serial.h*. Nejdříve jsem aktivoval hardwarový UART voláním funkce *Serial.begin(115200)*, kde předaná hodnota udává komunikační rychlost modulu ESP07. Pro odesílání dat jsem použil funkci *Serial.print()*, které jsem jako parametr předal AT příkaz v podobě textového řetězce. Protože každý příkaz je specifikován dobou vykonání a typem odezvy, bylo nezbytné pro příjem dat implementovat samostatnou funkci. Té je předána očekávaná odpověď společně s maximálním časem, do kterého musí být daná odpověď přijata. Na základě těchto dvou parametrů je vrácena informace v podobě logické hodnoty o úspěšnosti provedení příkazu.

6.2 Bezdrátová komunikace

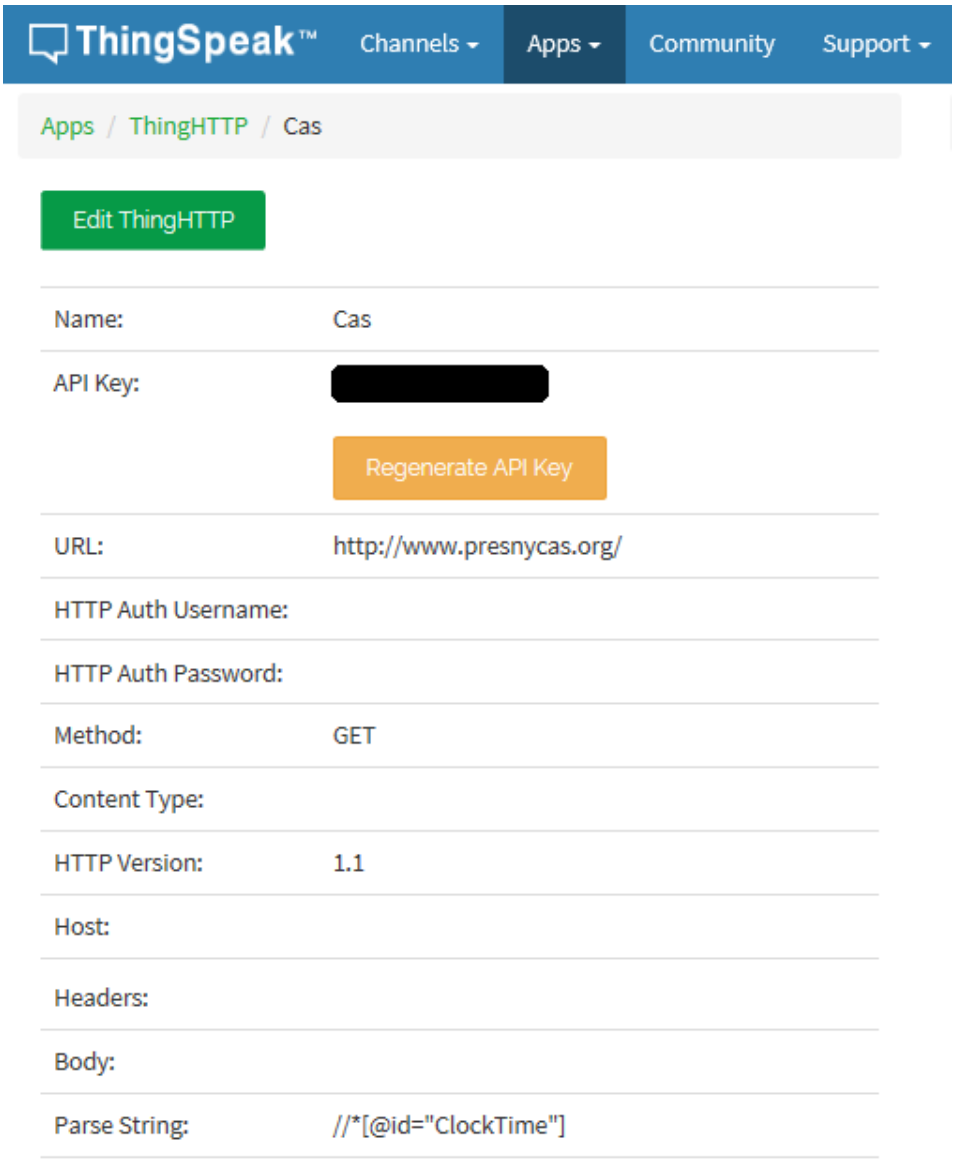
Klíčovou vlastností výsledného modulu by měla být bezdrátová komunikace přes WiFi. Protože ESP07 umožňuje práci jak v režimu klient, tak i v režimu přístupový bod, rozhodl jsem se tyto režimy využít oba [6]. První zmíněný bude použit pro předpokládaný případ, kdy modul bude mít možnost připojení k nějaké domácí síti. Výhodou bude možnost vzdáleného přístupu k modulu z libovolného zařízení přihlášeného do této sítě. Druhý pracovní režim bude používán v případě, kdy nebude možné se k místní WiFi síti připojit. Důvodem mohou být nesprávně zadané přihlašovací údaje, dočasná nedostupnost, či oblast bez již zabudované místní sítě. Díky této adaptaci bude modul neustále dostupný pro případnou obsluhu. Jistým omezením bude pouze nutnost připojení se přímo k tomuto zařízení, což je podmíněno fyzickou přítomností v jeho okolí.

Samotná implementace se skládá ze tří částí. První z nich je oblast s přihlašovacími údaji, kde musí uživatel ručně vyplnit jméno a heslo jak pro připojení k domácí síti, tak i pro případné vytvoření privátní sítě. Druhou částí je uvedení modulu do stavu připraveného k použití, protože po jeho spuštění by uvnitř mohla být některá předchozí nastavení, která by znemožňovala další práci. Jako první jsem tedy modul nastavil do režimu klient, následně mu odeslal příkaz pro odpojení od WiFi a poté pro zrušení automatického připojování. Nyní bylo možné přejít k poslední fázi, kterou je zajištění komunikace. Během té modul vyhledá všechny dostupné sítě v jeho okolí a porovnává jejich názvy s názvem, který vyplnil uživatel. V případě shody se pokusí připojit. Někdy se však stalo, že modulu se danou síť nepodařilo nalézt na první pokus, nebo měl problémy s autentizací. Z těchto důvodů jsou oba procesy při neúspěchu opakovány. Pokud se modul úspěšně připojí, nastaví se jeho automatické připojování, dále možnost vícenásobného připojení, spustí se TCP webový server na portu 80 a vypíše IP adresa, která mu byla přidělena. Tím je bezdrátová komunikace zaručena. V případě, kdy se modulu nepodaří během několika pokusů připojit, dojde k vytvoření vlastní WiFi sítě. Nejprve se změní pracovní režim ESP07 na přístupový bod a následně se vytvoří konfigurace samotné sítě, která je dána čtyřmi parametry. Jde o jméno, heslo, číslo kanálu a šifrování. Za první dva parametry se doplní ty, které zadal uživatel ručně. Číslo kanálu bude zvoleno 1 a typ šifrování WPA_WPA2_PSK. Po úspěšné konfiguraci se nastaví možnost vícenásobného připojení k této síti, které by mělo umožnit přístup až osmi zařízením. Nakonec se spustí TCP webový server na portu 80 a je vypsána IP adresa zařízení pro umožnění komunikace.

6.3 Stahování dat

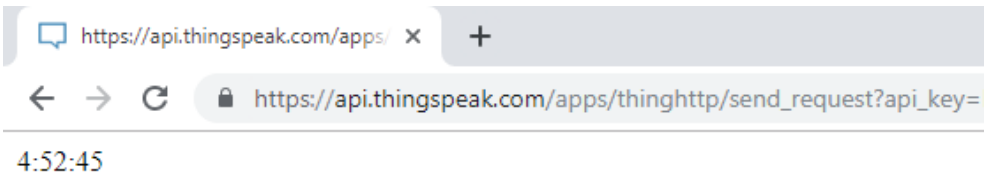
Zvolené použití modulu pro ovládání okenních žaluzií přináší závislost na několika informacích. Za hlavní z nich považuji dobu východu a západu slunce, aktuální předpověď počasí a čas, kde všechny tyto informace by bylo možné získat použitím nejrůznějších senzorů a výpočtů. Modul však disponuje připojením do internetu, což přidává možnost získání dat stažením. Osobně tuto možnost považuji za efektivnější z hlediska univerzálnosti, protože při případné změně použití stačí pouze upravit software. Pro moji práci tedy volím stažení dat.

K tomuto účelu jsem využil stránku thingspeak.com. Jedná se o službu pro internet věcí, která po bezplatné registraci nabízí ukládání dat, jejich následnou vizualizaci, analýzu a mnoho dalšího. Konkrétně jsem pracoval s aplikací ThingHTTP, pomocí které lze vytvořit žádost na stažení dat z libovolné http webové stránky. Po otevření této aplikace na stránkách thingspeak.com jsem vybral možnost vytvoření nové žádosti pomocí tlačítka New ThingHTTP. V otevřeném formuláři jsem zadal název žádosti, do URL zkopíroval dotazovanou stránku, vybral metodu GET a verzi HTTP nastavil na hodnotu 1.1. Poslední položkou k vyplnění byla Parse String. Tam jsem nakopíroval XPath konkrétních dat, které jsem chtěl z webové stránky získat.



(obr. 16) Konfigurace pro vygenerování žádosti

Po uložení žádosti mi aplikace vygenerovala URL pro získání chtěných dat. Jako kontrolu správnosti jsem tuto URL otestoval v prohlížeči.



(obr. 17) Testování vygenerované žádosti

Před použitím na modulu bylo nezbytné se nejprve k dané stránce připojit a vygenerovanou žádost upravit. Jednalo se o odstranění části, která již byla pro připojení použita, a následné přidání informací o verzi HTTP protokolu, odkazu na poskytovatele a doplnění příkazu pro ukončení spojení [18]. Výsledkem byla následující žádost:

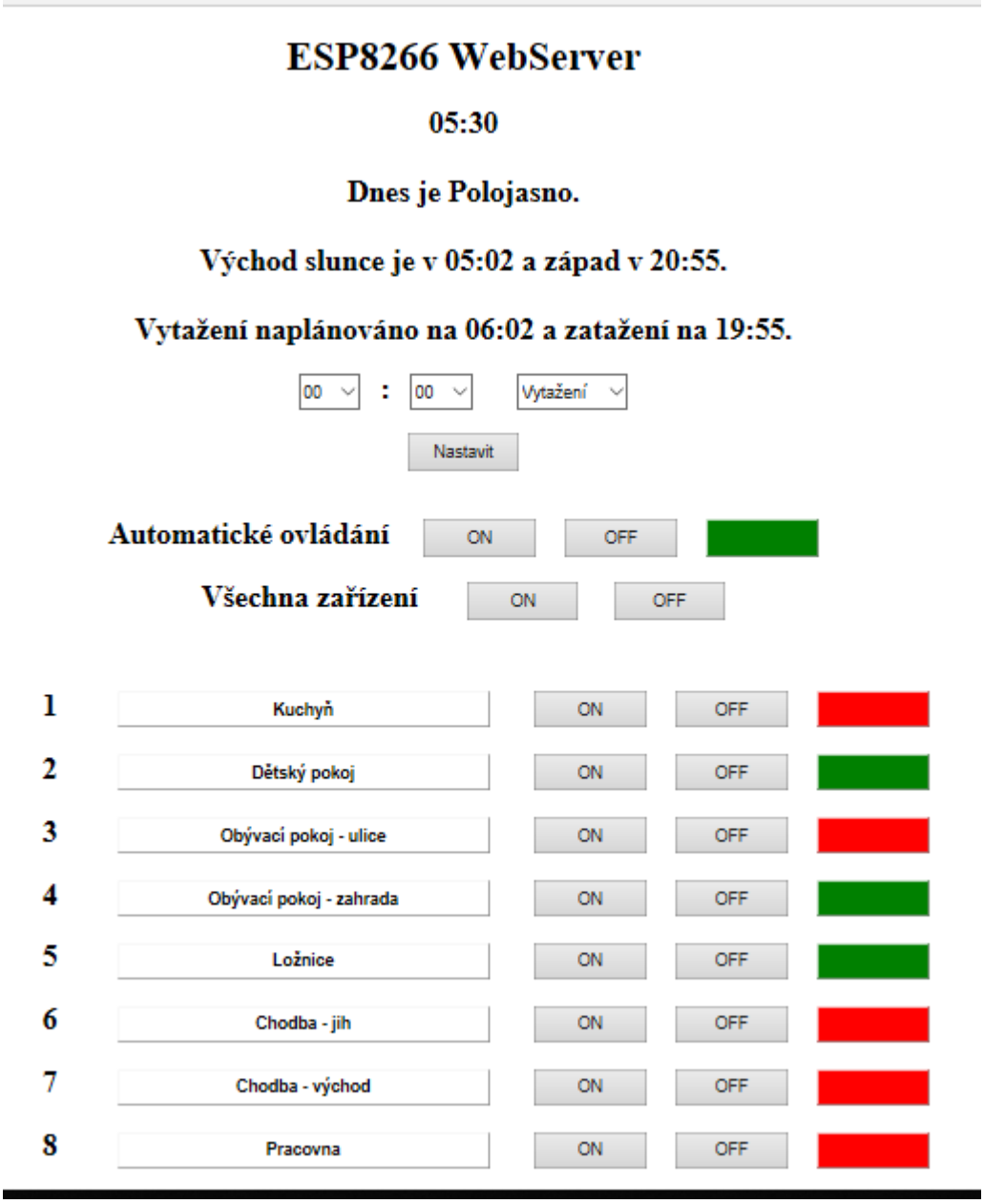
```
GET /apps/thinghttp/send_request?api_key="API klíč" HTTP/1.1
Host: "doména poskytovatele dat"
Connection: close
```

Upravenou žádost jsem odeslal pomocí dvojice příkazů. První z nich ohlásil modulu, na jaké síťové připojení budou, jak velká data poslána. Bezprostředně za ním již následovalo odeslání samotné žádosti, jejíž zpětnou vazbu tvořily požadované informace.

Aby byl zajištěn stálý chod v případě výpadku připojení k internetu či nedostupnosti stránky, je při každém úspěšném stažení dat ukládána návratová hodnota funkce *millis()*, která udává počet milisekund od startu procesoru. Modul tedy dokáže na základě poslední uložené informace jednoduše určit alespoň přibližný čas, který mu umožní pracovat bez jakýchkoliv omezení, než bude připojení obnoveno a data aktualizovány.

6.4 Webový server

Po připojení se k modulu z libovolného zařízení by se měla uživateli zobrazit webová stránka se všemi potřebnými informacemi a možnostmi vzdáleného ovládání a konfigurace. Nejedná se o zařízení s velkou přenosovou rychlostí, takže jsem kladl důraz na minimální množství dat. Vytvořil jsem tedy webovou stránku v jazyce HTML.



(obr. 18) Webová stránka

V horní části se nachází základní informace pro uživatele a bezprostředně pod ní začíná konfigurační a ovládací část doplněná textovým označením a barevnou signalizací polohy žaluzie. Důvod, proč webová stránka disponuje pouze 8 výstupy z celkových 16, je předpoklad, že elektrické žaluzie mají samostatný vodič pro ovládání každého směru pohybu. Jednotlivé názvy těchto výstupů budou vyplněny uživatelem na základě připojených zařízení.

Z hlediska implementace se jedná o periodickou kontrolu, zda nepřišlo „+IPD“, které značí data ze síťové komunikace. Pokud ano, tak se začne postupně přijímat zbývající část zprávy

odkud je zjištěno, zda došlo k nějaké změně. V případě změny dojde k aktualizaci dat a webové stránky. Celý princip zobrazení funguje tak, že na přijaté číslo síťového připojení se odešle HTML kód, který je prohlížečem připojeného zařízení zobrazen jako webová stránka. Pro úspěšný přenos bylo však nezbytné vyřešit některé problémy. Prvním bylo převedení HTML kódu na textový řetězec ohraničený uvozovkami. Tento problém jsem vyřešil použitím stránky *http://davidjwatts.com/youtube/esp8266/esp-convertHTM.html.* Dalším problémem byla nedostatečná paměť SRAM, jejíž kapacita v případě čipu ATmega328 je pouze 2 kB [2]. Řešením bylo takzvané *F()* makro, které uloží textový řetězec předaný jako parametr do flash paměti o kapacitě 32 kB [2]. Následně jsem toto makro využil i pro všechny ostatní textové řetězce, což mi ušetřilo dostatek paměti pro budoucí vývoj. Protože *F()* makro neumožňuje práci s proměnnými a funkce *Serial.print()* před odesláním kopíruje data do SRAM, byl jsem donucen moji HTML stránku rozdělit na menší datové bloky. Tím byly vyřešeny veškeré problémy s nedostatkem paměti a bylo možné tyto pakety odeslat. Po jejich úspěšném přenosu muselo dojít k ukončení spojení. Pokud by se tak nestalo, webová stránka by se neaktualizovala, ale vykreslovala za sebou.

6.5 Plánování

Samostatné rozhodování a vytváření scénářů pro následné vykonávání je zcela závislé na dostatku informací. Pro zvolené použití se bude jednat konkrétně o dobu východu a západu slunce, předpověď počasí a aktuální čas, kde všechny tyto hodnoty budou staženy z internetu.

Konkrétní čas pro stažení a vytažení žaluzií bude stanoven dobou východu a západu slunce, která bude posunuta o konstantu určenou kombinací mých zvyků a aktuální předpovědi počasí. Stránka, odkud bude tato předpověď stahována, obsahuje seznam všech predikcí, které lze očekávat. Pro usnadnění jsem si na základě klíčových slov celý tento seznam zúžil na 6 možností. Jako příklad zmíním, že při očekávané jasné obloze se žaluzie aktivují s východem a západem slunce. Naopak při snížené viditelnosti způsobené zataženou oblohou budou žaluzie vytaženy 2 hodiny po východu slunce a zataženy 2 hodiny před západem slunce. Pokud by to však uživateli nevyhovovalo, je možné obě hodnoty nastavit ručně na požadovaný čas.

6.6 Spínání výstupů

Modul disponuje 16 výstupy, kde prvních 8 je použito pro řízení vytažení žaluzií a druhá polovina pro jejich zatažení. Všechny jsou však spínány stejným způsobem pomocí hodnot nastavených na pinech I/O expanderu. Pokud tedy přijde požadavek na sepnutí výstupů, program na základě dostupných informací vygeneruje dvě hodnoty reprezentující logické stavy horních a dolních osmi pinů I/O expanderu. Tyto hodnoty jsou pak následně odeslány po I2C sběrnici a nastaveny. Pro úplnost ještě doplním, že k sepnutí dochází nastavením logické 0. Důležitou roli hraje také délka vygenerovaného řídícího impulzu, která je závislá na konkrétním spínaném produktu. Pro mé účely jsem tuto dobu stanovil na jednu sekundu, což zajistí dostatečnou dobu svitu signalizační LED diody pro optickou detekci.

7 Budoucí rozšíření

Modul v současné podobě disponuje konektivitou do internetu a 16 spínanými výstupy. Na základě těchto vlastností jsem implementoval softwarovou část se zaměřením na demonstraci ovládání okenních žaluzií. Během ní se mi podařilo výrobek obohatit o několik funkcionalit, které jsou v této oblasti schopny poskytnout uživateli potřebné služby.

Osobně již mám představy o budoucím rozšíření těchto služeb pro koncového uživatele. Konkrétně bych se rád zaměřil na rozšíření možností nastavení s důrazem na jednoduchost a přehlednost. Tím by bylo získáno více informací, což by mělo vést k lepším vyhodnocovacím závěrům, delšímu samostatnému běhu a snížení času k obsluze. Velkou výhodou pro tento budoucí vývoj je také možnost hardwarového rozšíření díky dostatečnému množství nevyužitých pinů uvnitř modulu.

Závěrem bych také zmínil, že modul může najít své využití i pro náročnější aplikace. V případě požadavku na velký výpočetní výkon by bylo možné přesunout tento požadavek na výkonnější zařízení prostřednictvím konektivity do internetu a na základě získaného výsledku provést požadovanou akci.

8 Závěr

Cílem práce bylo vytvoření modulu pro spínání spotřebičů napájených z rozvodné sítě 230 V. Hlavními požadavky byly možnost bezdrátového připojení do internetu, vzdálené ovládání a konfigurace, schopnost vytváření scénářů na základě získaných dat, implementace vhodné formy zabezpečení a umožnění případného hardwarového rozšíření.

V rámci hardwarového řešení jsem se rozhodl použít jako hlavní výpočetní prvek modulu populární vývojovou desku Arduino UNO. Tu jsem při výběru všech potřebných komponent mimo jiné obohatil o modul ESP07 pro síťovou komunikaci a o I/O expander k zajištění dostatku výstupů, které budou spínány optotriaky. Následovalo vytvoření schématu zapojení, během kterého bylo zapotřebí zajistit napěťovou kompatibilitu pro komunikaci mezi ESP07 a Arduino UNO, dále pak indikaci jednotlivých výstupů a přizpůsobení desky Arduino UNO pro napájení skrze jeho piny. Na základě schématu zapojení jsem navrhl desku plošných spojů, kterou jsem si poté nechal vyrobit. Následně jsem ji osadil všemi vybranými součástkami a umístil do předem připravené krabičky. Na závěr jsem byl donucen k provedení dočasné úpravy v oblasti napájení. Tím byl modul po hardwarové stránce úspěšně dokončen se všemi požadovanými vlastnostmi.

Softwarová část přidala modulu schopnost automatického připojení k lokální WiFi síti, či vytvoření vlastní sítě. Na základě této konektivity do internetu jsou stahována požadovaná data, která umožňují vytvářet různé scénáře. Další funkcionalitou je programové spínání jednotlivých výstupů. Modul také disponuje webovým serverem, který poskytuje uživateli vzdálené ovládání a konfiguraci přes webovou stránku. Při vývoji těchto základních bloků jsem kladl důraz na univerzálnost, protože modul by měl umožňovat široké spektrum použití. Přesto bylo nezbytné se zaměřit na konkrétní oblast, kterou bylo zvoleno ovládání okenních žaluzií.

Přestože je modul připraven pro instalaci do provozu, tak stále nedisponuje velkým množstvím funkcí, kterými by uživateli poskytoval potřebné služby. V současné době již mám nějaké představy o dalších vylepšení, které budou implementovány v rámci budoucího vývoje.

9 Zdroje

[1] Arduino UNO R3 Clone [online]. [cit. 2019-03-26]. Dostupné z: https://www.robotistan.com/arduino-uno-r3-clone-with-usb-cable-usb-chip-ch340

[2] Arduino(TM) UNO Rev3 [online]. [cit. 2019-04-10]. Dostupné z: https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf

[3] PCAL6416APW,118 Datasheet: *Low-voltage translating 16-bit I2C-bus/SMBus I/O expander with interrupt output, reset and configuration registers* [online]. [cit. 2019-03-26]. Dostupné z: http://www.farnell.com/datasheets/2207632.pdf?_ga=2.124414403.832761920.1558640739-1545305308.1557137827

[4] TLP360J Datasheet*: TOSHIBA Photocoupler* [online]. [cit. 2019-03-26]. Dostupné z: http://www.farnell.com/datasheets/2017851.pdf?_ga=2.68920408.832761920.1558640739-1545305308.1557137827

[5] ESP8266 WIFI Modul ESP-07 Bezdrátový [online]. [cit. 2019-04-02]. Dostupné z: https://arduino-shop.cz/arduino/1455-esp8266-wifi-modul-esp-07-bezdratovy.html

[6] ESP8266EX Datasheet [online]. [cit. 2019-04-02]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[7] ESP8266EX FAQs: *What are the general power supply requirements of the ESP8266?* [online]. [cit. 2019-04-02]. Dostupné z: https://www.espressif.com/en/support/download/documents

[8] VTX-214-005-0503 Datasheet: *5 Watt Dual Output AC-DC Converters* [online]. [cit. 2019-03-26]. Dostupné z: http://www.farnell.com/datasheets/2117907.pdf?_ga=2.60657124.832761920.1558640739-1545305308.1557137827

[9] ESP8266 Community Wiki: *ESP8266 GPIO Pin Allocations* [online]. [cit. 2019-04-09]. Dostupné z: https://www.esp8266.com/wiki/doku.php?id=esp8266_gpio_pin_allocations

[10] Arduino: Defining Pin Levels: *HIGH and LOW* [online]. [cit. 2019-04-10]. Dostupné z: https://www.arduino.cc/reference/en/language/variables/constants/constants/

[11] ESP8266EX FAQs: *Are the GPIO pins 5 V compatible?* [online]. [cit. 2019-04-10]. Dostupné z: https://www.espressif.com/en/support/download/documents?keys=&field_type_tid%5B%5D=14

[12] DATASHEET-1--960-346: *LED 0805* [online]. [cit. 2019-03-26]. Dostupné z: https://www.gme.cz/data/attachments/dsh.960-346.1.pdf

[13] DATASHEET-1--627-002: *Plastová průmyslová krabička* [online]. [cit. 2019-04-14]. Dostupné z: https://www.gme.cz/data/attachments/dsh.627-002.1.pdf

[14] Wire Library [online]. [cit. 2019-04-20]. Dostupné z: https://www.arduino.cc/en/Reference/Wire

[15] ESP8266 Non-OS AT Instruction Set [online]. [cit. 2019-04-21]. Dostupné z: https://www.espressif.com/en/support/download/documents?keys=&field_type_tid%5B%5D=14

[16] How to Update Flash ESP8266 Firmware – Flashing Official AT Firmware [online]. [cit. 2019-04-21]. Dostupné z: https://www.electronicshub.org/update-flash-esp8266-firmware/

[17] ESP8266 Flash addresses [online]. [cit. 2019-04-21]. Dostupné z: https://github.com/espressif/ESP8266_AT/tree/master/bin

[18] HTTP - Requests [online]. [cit. 2019-05-11]. Dostupné z: https://www.tutorialspoint.com/http/http_requests.htm

10 Přílohy

Seznam požitých součástek

LED 0805	16x
I/O Expander - PCAL6416APW,118	1x
AC/DC Napájecí zdroj - VTX-214-005-0503	1x
Optočlen - LP360J	16x
Pájecí svorkovnice 5,0 mm	17x
Modul ESP07	1x
Arduino UNO	1x
Keramický kondenzátor CKS0805 68 nF	2x
Tantalový kondenzátor CTS 47 uF	2x
Plastová průmyslová krabička IP65 U-01-22	1x
Schottkyho dioda MBR0520LT1G SMD	2x
SMD Rezistor R0805 10 kΩ	9x
SMD Rezistor R0805 56 Ω	16x