FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Bezpečná vzdálená aktualizace firmwaru pro Arduino
Student:	Matyáš Rak
Vedoucí:	Ing. Jiří Dostál, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Důležitou součástí Internet of Things (IoT) zařízení je možnost aktualizace jeho firmwaru kvůli opravám bezpečnostních zranitelností či jiných chyb nebo vylepšení. Tato vzdálená (over-the-air, OTA) aktualizace musí být sama o sobě zabezpečená, a to s ohledem na omezené hardwarové prostředky. Jednou z možných platforem, na kterých lze IoT zařízení realizovat, je Arduino.

1. Prozkoumejte stávající řešení OTA aktualizace firmwaru IoT zařízení se zaměřením na platformu Arduino.
2. Navrhněte obecný model bezpečné OTA aktualizace firmwaru pro IoT zařízení.
3. Přizpůsobte navržený obecný model pro platformu Arduino a konkrétní použitý hardware.
4. Na základě tohoto přizpůsobeného modelu OTA aktualizaci na Arduinu implementujte.
5. Výsledné řešení otestujte.

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 15. února 2022 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Bezpečná vzdálená aktualizace firmwaru pro Arduino

Matyáš Rak

Katedra počítačových systémů
Vedoucí práce: Ing. Jiří Dostál, Ph.D.

23. června 2022

Poděkování

Na tomto místě bych rád poděkoval panu Ing. Jiřímu Dostálovi, Ph.D. za vedení bakalářské práce a cenné rady. Dále bych rád poděkoval své rodině a kamarádům za veškerou podporu při psaní této práce.

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, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 23. června 2022 . . . . . .

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2022 Matyáš Rak. 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

Rak, Matyáš. *Bezpečná vzdálená aktualizace firmwaru pro Arduino*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022. Dostupný také z WWW: ⟨`https://gitlab.fit.cvut.cz/rakmaty1/bp/`⟩.

Abstrakt

Tato bakalářská práce se zabývá možnostmi bezpečné vzdálené aktualizace pro platformu Arduino. Nejprve jsou představeny důležité pojmy použité v této práci a shrnuta stávající řešení vzdálené aktualizace pro Arduino. Dále je navržen obecný model aktualizace pro IoT zařízení. V poslední části je na základě tohoto modelu, upraveného pro Arduino, vzdálená aktualizace implementována.

Klíčová slova Vzdálená aktualizace, Internet věcí, Arduino

Abstract

This bachelor thesis deals with the possibilities of secure over-the-air updates for the Arduino platform. First, the important concepts used in this thesis are introduced and existing over-the-air update solutions for Arduino are summarized. Then, a general update model for IoT devices is proposed. In the last section, based on this model, adapted for Arduino, the over-the-air update is implemented.

Keywords Over-the-air update, Internet of things, Arduino

Obsah

Úvod **1**

1 Průzkum stávajících řešení **3**
1.1 Internet věcí . . . 3
1.1.1 Architekrura IoT . . . 3
1.2 Arduino . . . 4
1.2.1 Vývojové desky . . . 4
1.2.2 Připojení Arduina k internetu . . . 5
1.2.3 ESP8266 . . . 5
1.3 Vzdálená aktualizace . . . 6
1.4 Arduino IoT Cloud . . . 6
1.5 Další implementace OTA pro Arduino . . . 7

2 Návrh obecného modelu aktualizace **9**
2.1 Obecný pohled na aktualizaci firmwaru . . . 9
2.2 Kontrola nové verze . . . 9
2.2.1 Aktivní dotazování ze strany zařízení . . . 9
2.2.2 Řízení aktualizace serverem . . . 10
2.3 Stažení nové verze firmwaru . . . 10
2.4 Samotná fyzická aktualizace . . . 10

3 Implementace **13**
3.1 Komunikace s wi–fi čipem . . . 13
3.2 Implementace v Arduinu . . . 14
3.2.1 . . . 14
3.3 Implementace v ESP . . . 15
3.4 Serverová část . . . 15
3.4.1 Database.java . . . 16
3.4.2 Server.java . . . 16

3.5 Možná vylepšení 16

Závěr **19**

Bibliografie **21**

A Seznam použitých zkratek **23**

B Obsah přiloženého CD **25**

Seznam obrázků

1.1 Vývojová deska Arduino Uno, zdroj [5] 4
1.2 Vývojová deska Arduino Nano, zdroj [6] 5
1.3 Čip ESP8266 v provedení ESP-01, zdroj [8] 6

3.1 Zapojení čipu ESP–01 k Arduinu, zdroj autor 14

Úvod

Používání zařízení Internetu věcí, tedy všemožných „chytrých“ zařízení, od jednoduchých senzorů přes ovladače světel, různých kuchyňských spotřebičů až po automobily, která jsou připojena k internetu, se dnes stává čím dál častější.

Důležitou součástí těchto zařízení je možnost jejich vzdálené aktualizace, ať již z důvodu oprav možných bezpečnostních zranitelností, či vylepšení stávajících funkcionalit. Ovšem i samotný proces aktualizace může útočník napadnout, tedy tento musí být sám o sobě zabezpečený, a to co nejlépe s ohledem na omezené hardwarové prostředky.

Tato bakalářská práce se zabývá bezpečnou vzdálenou aktualizací takového zařízení, a to konkrétně postaveného na platformě Arduino.

Cílem práce je prozkoumání stávajících řešení problému bezpečné vzdálené aktualizace firmwaru zařízení internetu věcí se zaměřením na Arduino, návrh obecného modelu této aktualizace, adaptace takto navrženého modelu na konkrétní použitý hardware (Arduino) a na jeho základě implementace v hardwaru a její otestování.

Práce v první části obeznámí čtenáře s pojmy a technologiemi, které jsou v ní používány. Dále se zabývá průzkumem stávajících možností řešení problému. V další části je navržen obecný model vzdálené aktualizace. Tento model je poté upraven pro konkrétní použitý hardware a na základě něj je práce implementována.

Kapitola 1

Průzkum stávajících řešení

První kapitola této práce seznámí čtenáře s důležitými pojmy použitými v této práci. Dále je zde proveden průzkum stávajících možností OTA aktualizace na Arduinu.

1.1 Internet věcí

Pojem internetu věcí (anglicky *Internet of things*, zkráceně IoT) se objevil na přelomu tisíciletí v souvislosti s připojením RFID tagů k internetu pro automatické počítání a sledování zboží. Dnes se jedná o běžně užívaný pojem zahrnující všechny možné technologie připojení věcí k internetu a mezi sebou.

I přes rozšíření pojmu neexistuje jednotná definice a lze jich najít mnoho vzájemně různých, ovšem koncept vždy mají stejný. Jedná se o vzájemné propojení zařízení vestavěných do běžně využívaných věcí, jejich připojení k internetu či jiné síti za účelem sběru a sdílení dat z různých senzorů, a to bez nutnosti lidských zásahů. Tato data jsou sdílena mezi zařízeními navzájem a také odesílána pomocí připojení k internetu či jiným způsobem na servery, kde probíhá další jejich zpracování a vyhodnocování. [1, 2, 3]

1.1.1 Architekrura IoT

Architekturu IoT zařízení lze většinou rozdělit do tří vrstev, a to na vrstvu fyzickou, síťovou a aplikační.

Jako fyzickou vrstvu můžeme označit samotný hardware, povětšinou vestavěný v nějakém objektu, kterému dodává další funkcionalitu. Většinou obsahuje nějaký mikropočítač a senzory sloužící ke sběru dat.

Dále již přecházíme k síťové vrstvě, tedy způsobu přenosu dat ze zařízení na jiné zařízení či na server. Zde je využíváno například wi–fi připojení, protokolu Bluetooth či dalších technologií.

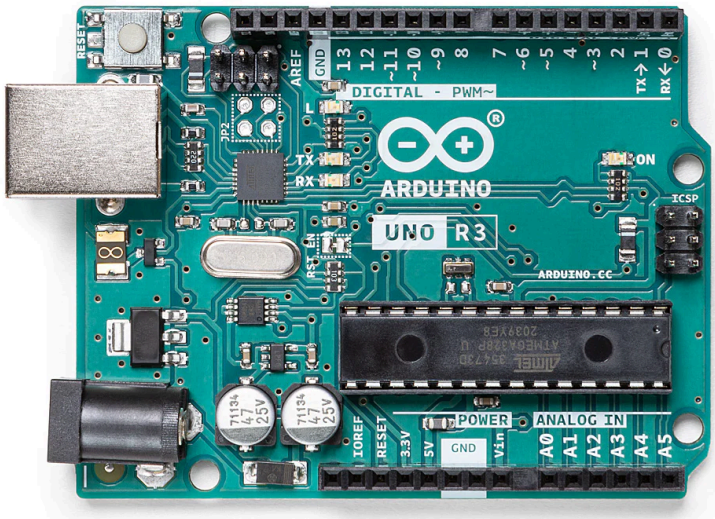
Aplikační vrstva zahrnuje server, se kterým zařízení komunikují, který data sbírá a vyhodnocuje. [1, 2]

1.2 Arduino

Jedná se o open-source hardwarovou platformu, která vznikla v Ivrea Interaction Design Institute v Itálii jako jednoduchá a levná platforma pro studenty bez zkušeností s elektronikou a programováním. Platforma je vyvíjena jako open-source, a to jak softwarová, tak hardwarová část. Díky tomu lze kromě oficiálních desek zakoupit i jejich levnější klony. Kromě původních 8bitových desek dnes existují i výkonnější produkty například pro IoT, 3D tisk či vestavné systémy.[4]

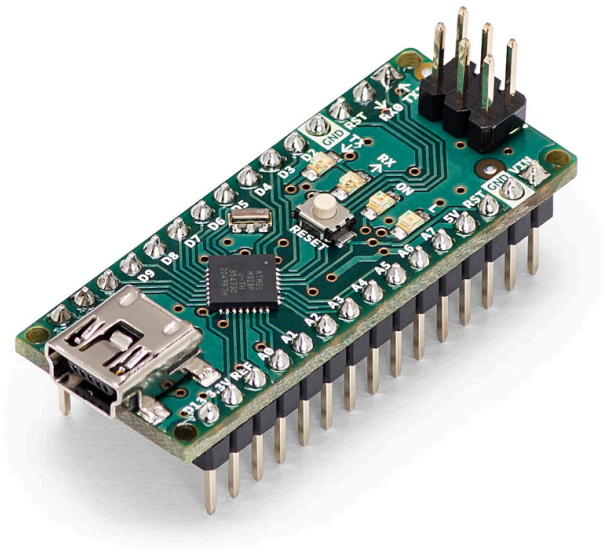
1.2.1 Vývojové desky

Jako první je nutné zmínit Arduino Uno. Jde o asi nejpoužívanější vývojovou desku platformy Arduino. V současnosti se prodává její třetí verze. Deska obsahuje procesor ATmega328P od firmy Atmel pracující na frekvenci 16 MHz. Deska obsahuje 6 analogových vstupních pinů a 14 digitálních vstupně-výstupních pinů. Dostupná paměť činí 32 kB FLASH, 2 kB SRAM a 1 kB EEPROM. K desce lze snadno přidat rozšiřující shield moduly. Existuje také verze Arduino Uno WiFi, která má zabudovanou podporu wi–fi a Bluetooth připojení.



Obrázek 1.1: Vývojová deska Arduino Uno, zdroj [5]

Vybavením podobnou deskou je Arduino Nano, které je oproti Unu menší a hodí se tedy spíše pro již vytvořený odladěný projekt. Také Arduino Nano má více oficiálních verzí.



Obrázek 1.2: Vývojová deska Arduino Nano, zdroj [6]

Jako další lze zmínit Arduino Micro, či větší a výkonnější desku Arduino Mega. Kromě těchto desek využívajících 8bitové AVR procesory existují i výkonnější verze desek Arduino založené například na procesorech ARM.

1.2.2 Připojení Arduina k internetu

Jak již bylo řečeno, existují desky Arduina se zabudovanou podporou wi–fi připojení, ať již oficiální či nikoliv. Ovšem pokud takovou desku nevlastníme, ale chceme si vytvořit z Arduina nějaké IoT zařízení, potřebujeme jinou možnost připojení k internetu.

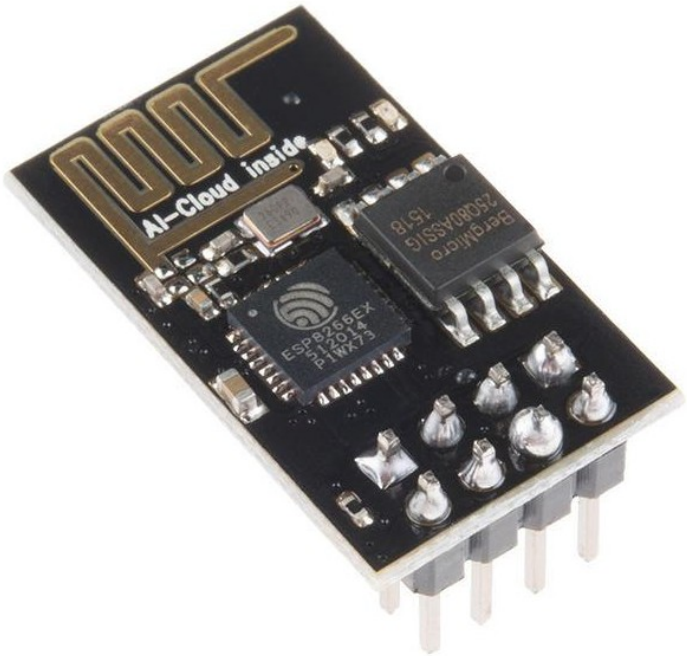
Dříve bylo možno zakoupit přímo wi–fi shield, ale ten již není k sehnání. Dnes je asi nejrozšířenější možností využití čipu ESP8266.

1.2.3 ESP8266

Jedná se o levný mikroprocesor od společnosti Espressif, který v sobě obsahuje podporu pro wi–fi připojení. Je potřeba si dát pozor, že tento čip využívá napájení 3.3V, ovšem Arduino je třeba využít převodník logické úrovně či nějaký adaptér.

IoT zařízení by šlo vytvořit pouze s použitím tohoto čipu, ovšem například ve své nejlevnější dostupné verzi, ESP–01, má vyvedeny pouze dva piny plus další dva pro sériovou komunikaci, což pro různé senzory bude málo.

V základní verzi je v zařízení nahrán firmware, který podporuje AT příkazy.[7]



Obrázek 1.3: Čip ESP8266 v provedení ESP-01, zdroj [8]

1.3 Vzdálená aktualizace

U IoT zařízení chce mít možnost aktualizovat firmware, který na něm běží. Ať už z důvodu oprav nějaké chyby ve firmwaru či opravy nějaké bezpečnostní hrozby, tak také z důvodu možného přidání nějaké funkcionality.

Protože zařízení je připojeno k internetu, nabízí se možnost tuto aktualizaci provést vzdáleně (Over-the-air) a nikoliv jen připojením zařízení k počítači.

1.4 Arduino IoT Cloud

Jedná se o oficiální cloudové řešení pro IoT zařízení od Arduina. Toto řešení umožňuje uživateli po zaregistrování se připojit do cloudu některou z oficiálních podporovaných Arduino desek. Také je možné přidat jakoukoliv jinou desku využívající wi–fi mikrokontrolér ESP32 či ESP8266.

Arduino IoT Cloud umožňuje monitorování dat ze senzorů připojených zařízení a jejich vizualizaci, komunikaci mezi více připojenými zařízeními pomocí sdílení proměnných, plánování spouštění různých úloh, integraci projektu s ostatními službami pomocí webhooks a v neposlední řadě také vzdálenou aktualizaci kódu připojených zařízení. Ta je pro tuto práci nejzajímavější.

Ovšem zde je nutno začít první nevýhodou, vzdálená aktualizace pomocí Arduino IoT Cloudu je dostupná pouze pro čtyři oficiální Arduino desky (konkrétně Arduino MKR WiFi 1010, Arduino Nano 33 IoT, Arduino Nano RP2040 Connect a Portenta H7) a pro žádné jiné, tedy hlavně desky třetích stran s přidaným wi–fi čipem, dostupná není.

Pokud tedy uživatel vlastní jednu z těchto podporovaných desek, může si po jejím připojení do Arduino IoT Cloudu aktivovat možnost vzdálené aktualizace. Tento proces probíhá tím způsobem, že uživatel nahraje nově napsaný kód na Cloud a zde zvolí možnost nahrání „Over-the-air“. Program je poté zkompilován a pomocí Amazon AWS S3 bucketu nahrán do připojeného Arduina.

Tento proces je tedy čistě manuální a vždy se pracuje jen s jedním konkrétním zařízením. Tato možnost se jeví jako vhodná pro osobní použití, například pro aktuální vývoj programu, případně pro možné aktualizace například senzoru umístěného venku na zahradě ve výšce. Ovšem například pro aktualizace více zařízení se stejným firmwarem příliš vhodný není. [9, 10]

1.5 Další implementace OTA pro Arduino

Jako další je vhodné zmínit tuto implementaci, neboť tímto postupem jsem se vcelku inspiroval. Jedná se o implementaci, v níž je k Arduinu připojen čip ESP8266 s výchozím firmwarem. Pomocí něj se Arduino připojí k aktualizačnímu serveru, který následně pomocí protokolu STK500 komunikuje s bootloaderem Arduina a nahraje do něj novou verzi firmwaru. [11]

Kapitola 2

Návrh obecného modelu aktualizace

Tato kapitola se zabývá návrhem obecného modelu aktualizace a diskutuje možná řešení.

2.1 Obecný pohled na aktualizaci firmwaru

Proces vzdálené aktualizace firmwaru IoT zařízení lze rozdělit do třech částí. Zařízení musí nejprve zjistit, zda má aktualizace proběhnout, tedy zda je k dispozici nová verze firmwaru. Pokud ano, tak dojde ke stažení této nové verze. Po ověření, že je stažený firmware v pořádku, nakonec dojde k vlastní aktualizaci v zařízení.

2.2 Kontrola nové verze

V prvním kroku musí zařízení zjistit, zda je k dispozici nová verze firmwaru. Tento postup může fungovat v zásadě dvěma způsoby. První možností je, že zařízení vystupuje jako klient a s nějakou periodou se dotazuje aktualizačního serveru na dostupnost nového firmwaru. Druhou možností je, že zařízení aktualizace aktivně nevyhledává, ale iniciativu zde přebírá server.

2.2.1 Aktivní dotazování ze strany zařízení

Zde se práce věnuje první z možností, tedy aktivnímu dotazování od zařízení. IoT zařízení musí mít v sobě uloženu adresu aktualizačního serveru. S nějakou předem určenou periodou se bude serveru dotazovat na dostupnost nové aktualizace. To, jaká tato perioda bude, je třeba určit s ohledem na spotřebu zařízení.

Zařízení naváže spojení se serverem. Tato komunikace by měla být šifrována, například pomocí protokolu TLS, aby se zabránilo možnému odposlechu komunikace útočníkem. Při tomto spojení je nutné, aby si zařízení ověřilo, že se připojuje ke správnému serveru. K tomu se využije ověření certifikátu poskytnutého serverem.

Pro vyšší zabezpečení může IoT zařízení serveru poskytnout svůj certifikát. Pokud ten server nezná, tak spojení odmítne a ukončí.

Po úspěšném ověření serveru následuje samotný dotaz na dostupnost nového firmwaru. Tuto může vyhodnocovat buďto server nebo samotné zařízení.

V prvním případě zašle zařízení serveru potřebné identifikační údaje jako je typ zařízení, aktuální firmware a číslo jeho verze. Server vyhodnotí, zda má pro toto zařízení novou verzi a zařízení toto sdělí.

Druhou možností je, že server zašle údaje o nejnovějším dostupném firmwaru zařízení, a to si samo vyhodnotí, zda je tato verze novější než ta, která v něm momentálně běží a zda tedy bude chtít firmware aktualizovat nebo ne.

2.2.2 Řízení aktualizace serverem

Druhou možností je, že si aktualizace bude řídit aktivně server. Tedy pokud má server k dispozici novou verzi firmwaru, tak jí začne nabízet všem jím řízeným zařízením. Tato verze přidává další možné vektory útoku pro útočníka, neboť zde musí existovat nějaká možnost připojit se k IoT zařízení, aby mu mohla být nabídnuta nová verze firmwaru.

2.3 Stažení nové verze firmwaru

Pokud si zařízení již zjistilo, že je k dispozici nová verze firmwaru (ať již samo nebo nabídnutím od serveru), tak dojde k jejímu stažení. Komunikace je samozřejmě nadále šifrována.

Samotný soubor firmwaru může být přenášen v běžné podobě. Dále může být kryptograficky podepsán, případně i zašifrován. Jak pro podpis, tak šifrování se zde hodí využití asymetrické kryptografie, kdy privátní klíč má server (resp. autor firmwaru) a podepíše/zašifruje tento firmware. Zařízení pak disponuje jeho veřejným klíčem, kterým si po stažení ověří, zda je firmware opravdu podepsán serverem, případně ho pomocí něj dešifruje. Tyto dva kroky lze samozřejmě zkombinovat a firmware jak šifrovat tak podepsat.

2.4 Samotná fyzická aktualizace

Tato část procesu aktualizace již probíhá v offline režimu bez připojení k aktualizačnímu serveru. Zařízení si již stáhlo nový firmware z aktualizačního serveru. To, jak proběhne samotná aktualizace, se u každého typu zařízení

bude lišit. V principu je nejprve nutné ověřit si integritu staženého firmware. K tomu lze využít například CRC či jiný kontrolní součet.

Pro vyšší zabezpečení může být firmware kryptograficky podepsán, v tomto případě si zařízení tento podpis ověří. Případně může být firmware zašifrován, tak dojde k jeho dešifrování.

Pokud při ověřování integrity či podpisu firmwaru nebo jeho dešifrování dojde k chybě, tak zařízení proces aktualizace ukončí a starý firmware zůstane zachován.

Jakmile zařízení disponuje novým firmwarem, u kterého má ověřenou integritu, může přistoupit k samotné aktualizaci. Tento proces se bude u každého zařízení lišit. Většinou se o něj bude starat bootloader zařízení.

Je vhodné myslet na to, že aktualizace může v průběhu selhat a v zařízení tedy bude firmware v nedefinovaném stavu. Pokud má zařízení dostatek paměti, může v ní mít uložen jak starý tak nový firmware. O to, aby se spustil aktuální firmware, se bude starat bootloader zařízení. Případně pro tyto případy může mít zařízení někde uloženo výchozí firmware, který bude při zjištění chyby v průběhu aktualizace nahrán místo nového.

Kapitola 3

Implementace

Pro implementaci bude využito Arduino Uno, respektive jeho klon. Jako wi–fi modul bude sloužit čip ESP8266, konkrétně v provedení ESP–01.

Z navrženého obecného modelu bude použita možnost aktivního dotazování se serveru. Pro ověření serveru bude využit předem uložený certifikát serveru. Veřejný a soukromý klíč certifikátu využívají eliptické křivky. Rozhodování, zda je firmware aktuální nebo není zde bude dělat server.

Firmware nebude prozatím podepisován, integrita bude kontrolována s využitím kontrolního součtu. Jako číslo verze firmwaru je využito prosté celé číslo, kdy každá další verze firmwaru má toto číslo vyšší než ta předchozí.

3.1 Komunikace s wi–fi čipem

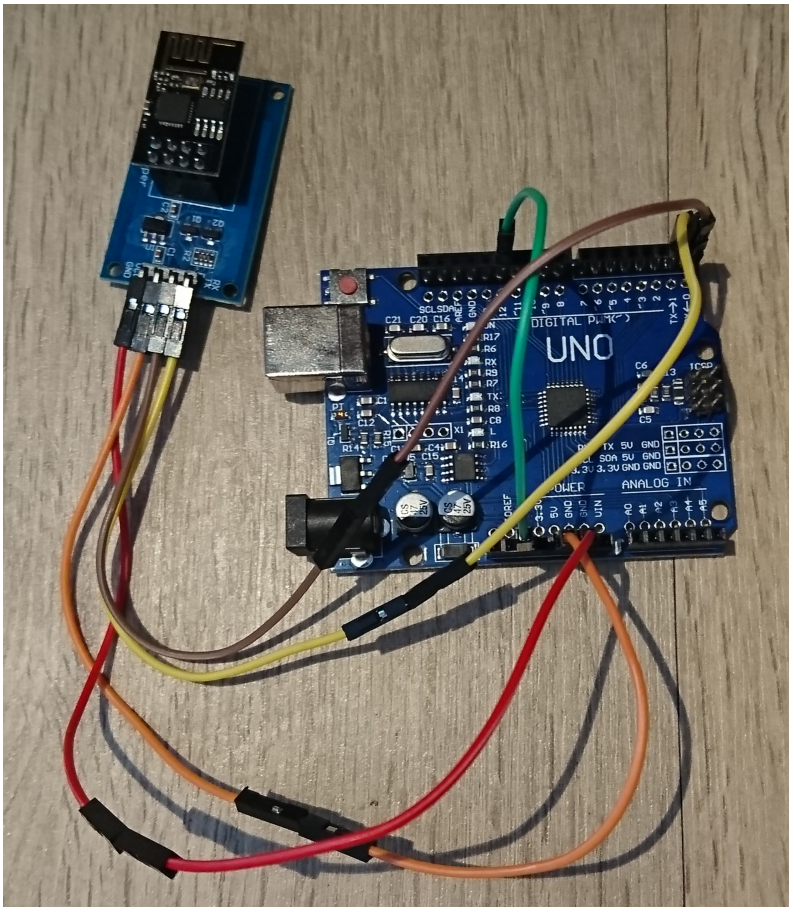
Prvotní myšlenka, tedy využít ESP8266 pouze jako wi–fi modul s výchozím firmwarem a veškerou potřebnou funkcionalitu naprogramovat přímo v Arduinu se nakonec ukázala jako spíše nereálná.

Jak bylo již dříve zmíněno, ESP8266 využívá ve výchozím stavu ke komunikaci AT příkazy. [7] Ovšem programování této komunikace tímto způsobem není zrovna jednoduché. Sice existují pro Arduino knihovny snažící se tuto komunikaci řešit, ovšem k uspokojivým výsledkům to nevedlo. I kdyby se zadařilo úspěšně do Arduina nový firmware stáhnout, tak by zde byla potíž v samotné aktualizaci. Na způsob, kterým by přímo Arduino samo aktualizovalo svůj vlastní běžící kód, jsem nepřišel.

Proto jsem po neúspěchu přistoupil k řešení, kdy nevyužiji výchozí firmware wi–fi modulu s AT příkazy, ale naprogramuji si pro něj vlastní. Čip ESP8266 má podporu pro Arduino IDE, tedy po instalaci potřebného balíčku je možné vyvíjet pro něj software stejným způsobem jako pro jiné desky Arduina. [12]

Pro komunikaci mezi Arduinem a ESP byl zvolen jednoduchý textový protokol. Komunikace probíhá pomocí sériové linky. Jednoduchý textový protokol

byl také zvolen pro komunikace mezi serverem a Arduinem.



Obrázek 3.1: Zapojení čipu ESP–01 k Arduinu, zdroj autor

3.2 Implementace v Arduinu

Implementace v Arduinu ve svém těle jednak bliká vestavěnou LED — což je v podstatě „simulace“ činnosti sběru nějakých dat ze senzorů. Pokud uplyne stanovená doba, tak dojde k připojení k serveru a ověření, zda není k dispozici nový firmware.

3.2.1

Arduino odešle wi–fi modulu zprávu „CONNECT“ a dále SSID a heslo k wi–fi síti, ke které se má připojit. V případě neúspěchu pokus o připojení zopakuje maximálně po stanovený počet pokusů. Po úspěšném připojením,

signalizovaném obdržením zprávy „CONNECTED“, odešle zprávu „CHECK UPDATE“ a jméno svého firmwaru a číslo jeho verze.

V případě neúspěchu toto opět pakuje po stanovený počet pokusů. Pokud obdrží od wi–fi modulu zprávu „NO UPDATE“, znamená to, že server vyhodnotil, že v Arduinu běží aktuální firmware a pokus o aktualizaci je ukončen. Zprávou „DISCONNECT“ signalizuje odpojení od wi–fi. Pokud přijde zpráva „UPDATE AVAILABLE“, tak server vyhodnotil, že aktualizace k dispozici je. V tomto případě Arduino odešle zprávu „PERFORM UPDATE“ a následně provede reset, čímž dojde ke spuštění bootloaderu Arduina, který zajistí nahrání nového firmwaru.

3.3 Implementace v ESP

Implementace v ESP čte příchozí zprávy na sériové lince. Při obdržení zprávy „CONNECT“ a následně SSID a hesla k wi–fi síti se k ní pokusí připojit a výsledek zašle po sériové lince zpět.

Při obdržení zprávy „DISCONNECT“ se od wi–fi sítě odpojí.

Při obdržení zpávy „PERFORM UPDATE“ se wi–fi modul připojí k aktualizačnímu serveru. Jeho adresa je napevno uložena v tomto modulu. K zabezpečení připojení je využita knihovna BearSSL. [13] Tato knihovna zatím nepodporuje TLS verze 1.3, tudíž je použita verze 1.2 (toto je nastaveno na straně serveru).

Při připojení dojde k ověření serveru, známý certifikát aktualizačního je uložen napevno ve wi–fi modulu. Pokud se certifikát změní, tak k připojení nedojde.

Následně je na server zaslán požadavek na kontrolu nové verze spolu se jménem a číslem firmwaru. Server z databáze zjistí, zda má k dispozici novější firmware, než běží v Arduinu a výsledek odešle zpět. Pokud nový firmware k dispozici není, ESP odešle na sériovou linku zprávu „NO UPDATE“.

Pokud je k dispozici nový firmware, tak se wi–fi modul znovu připojí k serveru, vyžádá a stáhne si tento nový firmware.

Firmware je uložen v souboru ve formátu hex tak, jak nám jej vygeneruje Arduino IDE. Po stažení z aktualizačního serveru je soubor převeden do binární podoby, přičemž je na každém řádku kontrolován kontrolní součet, čímž dojde k ověření integrity staženého firmwaru. V případě úspěchu je po sériové lince Arduinu zaslána zpráva „UPDATE AVAILABLE“.

Při obdržení zprávy „PERFORM UPDATE“ dojde k samotné aktualizaci. ESP naváže komunikaci s bootloaderem Arduina a pomocí protokolu STK500 odešle bootloaderu Arduina tento nový firmware. Správná implementace a zprovoznění této části nebyla vůbec jednoduchá.

3.4 Serverová část

Pro implementaci serverové části byl zvolen jazyk Java. Součástí implementace jsou dvě třídy.

3.4.1 Database.java

Třída *Database.java* se stará o komunikaci s databází. V implementaci je použita jednoduchá databáze SQLite. V databázi máme vytvořenu tabulku *firmware*, ve které evidujeme jméno a verzi firmwaru a cestu k němu. Přičemž jméno a verze firmwaru dohromady musí být unikátní.

Dále jsou zde implementovány metody pro přidání nového firmwaru do databáze a získání cesty k souboru pro konkrétní firmware (identifikovaný jménem a číslem verze). Poslední metoda vrací nejnovější verzi firmwaru nějakého jména. Tato projde databázi a vrátí nejvyšší číslo firmwaru se zadaným jménem — tedy nejnovější firmware tohoto jména, který má server k dispozici.

3.4.2 Server.java

Druhou třídou implementace serveru je třída *Server.java*. Tato třída se stará o zbytek funkcionality serveru.

Server je možno spustit z příkazové řádky spuštěním aplikace s argumentem *start*. Dalším možným argumentem je *add*, následovaný jménem firmwaru, číslem jeho verze a cestou k němu. Takovýto firmware je poté přidám do databáze.

Při spuštění serveru je načteno úložiště klíčů ve formátu PKCS12, kde je uložen certifikát serveru a k němu příslušící privátní klíč. Tento je použit k vytvoření SSL Socketu, který poslouchá na definovaném portu. Je nastaveno použití TLS verze 1.2. Při připojení klienta (metoda accept na socketu) je vytvořeno nové vlákno, které zajistí jeho obsluhu.

Pokud klient zašle příkaz „CHECK“, následovaný jménem a verzí firmwaru, tak server v databázi ověří, zda pro firmware klienta s tímto jménem nemá novější verzi. Pokud novější firmware má, odpoví zprávou „UPDATE AVAILABLE“, pokud ne, tak „NO UPDATE“. V případě neexistujícího jména firmwaru server odpoví chybou.

Druhý možný příkaz je „DOWNLOAD“, opět následovaný jménem firmwaru a číslem jeho verze. Pokud pro tento firmware existuje novější verze, tak ji server připojenému klientu zašle. Na prvním řádku je odeslána velikost souboru s firmwarem a poté jsou odeslána samotná data. Pokud novější verze neexistuje, tak server odpoví chybou.

V případě přijetí jiného příkazu server odpoví chybou a spojení ukončí.

3.5 Možná vylepšení

Práce byla v průběhu vývoje průběžně testována a nabízí nyní funkční základ OTA aktualizace pro Arduino. Ovšem nabízejí se možná vylepšení.

Jako první vylepšení by se nabízela také možnost aktualizace samotného wi–fi čipu. Podpora pro OTA aktualizace samotného wi–fi čipu existuje a tedy je možné jí doprogramovat.

Dalšími možnými vylepšeními by byla implementace podepisování či šifrování firmwaru.

Závěr

Cílem této bakalářské práce bylo navrhnout model bezpečné vzdálené aktualizace firmwaru pro zařízení internetu věcí, upravit tento model pro konkrétní použitý hardware a na jeho základě provést samotnou implementaci a otestování.

Nejprve byly prozkoumány stávající možnosti pro bezpečnou vzdálenou aktualizaci se zaměřením na platformu Arduino a byly shrnuty výhody a nevýhody jednotlivých dostupných řešení.

Dále se práce zabývala samotným modelem bezpečné aktualizace a diskutovala možná řešení, zde je výsledkem navržený obecný model pro bezpečnou vzdálenou aktualizaci.

Dále je tento model upraven pro konkrétní použitý hardware. K implementaci je využit klon Arduino Uno s wi–fi modulem ESP–01. Pro implementaci v hardwaru je použit jazyk C, serverová část je napsána v jazyce Java.

Bibliografie

1. ROSE, Karen; ELDRIDGE, Scott; CHAPIN, Lyman. The Internet of Things: An Overview: Understanding the Issues and Challenges of a More Connected World. *The Internet Society (ISOC)*. 2015, s. 7–18.
2. NORD, Jeretta Horn; KOOHANG, Alex; PALISZKIEWICZ, Joanna. The Internet of Things: Review and theoretical framework. *Expert Systems with Applications*. 2019, roč. 133, s. 98–103. ISSN 0957-4174. Dostupné z DOI: `https://doi.org/10.1016/j.eswa.2019.05.014`.
3. WORTMANN, Felix; FLÜCHTER, Kristina. Internet of Things. *Business & Information Systems Engineering*. 2015, roč. 57, č. 3, s. 221–224. ISSN 1867-0202. Dostupné z DOI: `10.1007/s12599-015-0383-3`.
4. *What is Arduino?* [Online] [cit. 2022-06-23]. Dostupné z: `https://www.arduino.cc/en/Guide/Introduction`.
5. ARDUINO. *Arduino UNO Rev3* [online] [cit. 2022-06-23]. Dostupné z: `https://store.arduino.cc/products/arduino-uno-rev3`.
6. ARDUINO. *Arduino Nano* [online] [cit. 2022-06-23]. Dostupné z: `https://store.arduino.cc/products/arduino-nano`.
7. ESPRESSIF. *AT Command Set* [online] [cit. 2022-06-23]. Dostupné z: `https://docs.espressif.com/projects/esp-at/en/latest/esp32/AT_Command_Set/index.html`.
8. DRÁTEK.CZ. *ESP8266 ESP-01 WIFI TCP/IP* [online] [cit. 2022-06-23]. Dostupné z: `https://dratek.cz/arduino/911-internet-veci-je-tady-tcp-ip-wifi-esp8266-esp-01.html`.
9. SÖDERBY, Karl. *Getting Started With the Arduino IoT Cloud* [online] [cit. 2022-06-23]. Dostupné z: `https://docs.arduino.cc/cloud/iot-cloud/tutorials/iot-cloud-getting-started`.

10. GARCIA, Jose. *Uploading sketches Over-The-Air (OTA)* [online] [cit. 2022-06-23]. Dostupné z: https://docs.arduino.cc/cloud/iot-cloud/tutorials/ota-getting-started.

11. *Programando un Arduino remotamente con el módulo ESP8266* [online] [cit. 2022-06-23]. Dostupné z: https://www.sistemasorp.es/2014/11/11/programando-un-arduino-remotamente-con-el-modulo-esp8266/.

12. GROKHOTKOV, Ivan. *ESP8266 Arduino Core's documentation* [online] [cit. 2022-06-23]. Dostupné z: https://arduino-esp8266.readthedocs.io/en/latest/index.html.

13. PORNIN, Thomas. *BearSSL* [online] [cit. 2022-06-23]. Dostupné z: https://bearssl.org/.

Příloha **A**

Seznam použitých zkratek

IoT Internet of things – internet věcí

OTA Over-the-air update – vzdálená aktualizace

Příloha **B**

Obsah přiloženého CD

```
src
  impl ..................................zdrojové kódy implementace
    arduino ..................zdrojové kódy programu pro Arduino
    esp ..........................zdrojové kódy programu pro ESP
    server ....................zdrojové kódy aktualizačního serveru
  thesis ......................zdrojová forma práce ve formátu LaTeX
text ..................................................text práce
  thesis.pdf ............................text práce ve formátu PDF
```