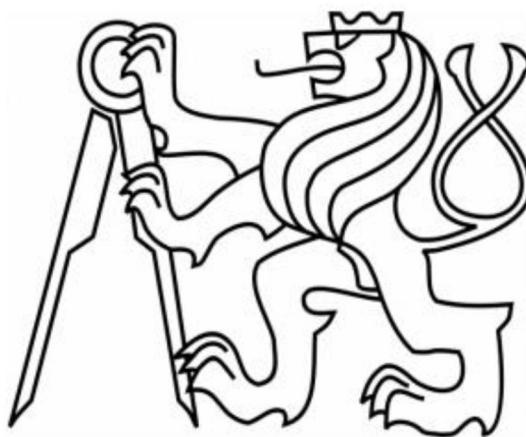


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STROJNÍ

Ústav přístrojové a řídicí techniky



RFID přenosný data kolektor
RFID mobile data collector

Ordokov Eldiir

10.02.2023

Studijní program: Strojírenství

Studijní obor: Informační a automatizační technika

Vedoucí práce: Ing. Matouš Cejnek Ph.D.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s využitím zdrojů, které jsou všechny v práci uvedeny.

Datum

Podpis.....

Poděkování

Rád bych poděkoval pana Ing. Matouš Cejnek, Ph.D. za rady, připomínky a pomoc při zpracování bakalářské práce.

Abstract

Cílem této diplomové práce je vývoj inovativního zařízení pro identifikaci návštěvníků, které je schopné číst kódy z karet, provádět výměnu dat se serverem a na dálku tato data spravovat. To je zvláště relevantní pro místa s vysokou frekvencí návštěvnosti, kde zastaralé metody identifikace již nezvládají úkoly zajištění bezpečnosti a efektivity evidence návštěvníků. V práci byla provedena důkladná analýza stávajících řešení, vybrány optimální komponenty, vyvinuty softwarové a hardwarové komponenty a také sestaven prototyp. Vyvinuté zařízení se vyznačuje vysokou přesností a rychlostí identifikace, zároveň zajišťuje spolehlivou ochranu osobních údajů. Výsledky výzkumu demonstrují významnou praktickou hodnotu vývoje, nabízejí efektivní řešení pro zvýšení úrovně bezpečnosti a evidence návštěvníků v různorodých objektech.

Klíčová slova

ESP8266MOD, RFID-RC522, I2C display LCD 1602, Fritzing, Převodník CP2102, Arduino, FileZilla Client, FTP server (android).

Abstract

The aim of this thesis is to develop a device capable of reading the code from the card, sending requests to the server, and remotely controlling and managing the data, which is relevant for providing control over the attendance of objects with a high frequency of people. Due to the obsolescence of traditional identification methods, the development of such a system represents an important step towards increasing the security and efficiency of visitor registration. The work includes the analysis of the current state in this area, the selection of components, the development of software and hardware, as well as the construction of the device itself. Using modern methods of development and analysis, a device was created that not only ensures high accuracy and speed of identification, but also guarantees the protection of visitor data from leakage. The results of the work have significant practical significance, as the designed device can be implemented in various objects with the aim of improving security control and visitor registration.

Keywords

ESP8266MOD, RFID-RC522, I2C display LCD 1602, Fritzing, Converter CP2102, Arduino, FTP client, FTP server (android).

Obsah

1. ANALÝZA SYSTÉMŮ ŘÍZENÍ A ŘÍZENÍ PŘÍSTUPU	3
1.1. Definice a hlavní možnosti systémů řízení a správy přístupu	3
2. Formulace cíle a úkolů projektu	5
3. Sekvenční diagram.....	5
3.1. Sekvenční diagram PlantUml	5
4. Výběr komponent zařízení.....	6
4.1. Čtečka RFID a RFID značka	6
4.2. Hardwarová platforma.....	8
4.3. Popis a účel vývodů NodeMCU V3 ESP8266	9
4.4. Pájecí prkénko	12
4.5. Propojovací kabely	13
4.6. I2C 1602 LCD Display Module 16X2	13
5. Napájení zařízení.....	14
6. Kompletní blokové schéma.....	15
6.1. Vypracování schématu elektrického obvodu zařízení:	16
6.2. Logika připojení zařízení:.....	17
6.3. Zařízení založené na platformě NodeMCU se skládá z následujících bloků:	17
7. Připojení Arduina k počítači	18
8. Arduino a sériový port UART	18
9. Softwarová část.....	19
9.1. Psaní programu pro mikrokontrolér.....	19
10. ESP8266 - mikrokontrolér s Wi-Fi rozhraním	21
11. Stanice a přístupový bod ESP8266.....	22
11.1. Zpracování HTTP požadavků v aplikaci webového serveru ESP8266.....	27
12. FTP client.....	29
13. FTP server	30
13.1. Výhody a nevýhody FTP serveru	31
13.1.1. Nevýhody FTP serveru.....	31
13.1.2. Výhody FTP serveru	31
14. Mobile remote server collector	34
15. Schéma činnosti zařízení.....	35
16. Závěr k oddílu	36
17. Výroba celého zařízení	36
17.1. Celý cyklus posloupnosti operací přenosu dat přes ESP8266.....	37
18. Závěr	38
19. Seznam použité literatury	39
20. Seznam obrázků	41
21. Seznam tabulek:	42

Úvod

Zajištění kontroly nad návštěvností ve vzdělávacích a dalších veřejných zařízeních, stejně jako ochrana osobních údajů před neoprávněným přístupem, se stávají stále naléhavějšími úkoly. Existující metody identifikace založené na používání dokumentů, jako jsou pasy a občanské průkazy, často nesplňují vysoké bezpečnostní standardy a mohou zpomalit proces průchodu kontrolními body.

V oblasti fyzické bezpečnosti je klíčovým aspektem kontrola přístupu, která zahrnuje opatření omezující přístup k určitým oblastem nebo prostorám za účelem zajištění přístupu pouze autorizovaným osobám. Pro fyzickou registraci přístupu mohou být použiti jak personál (například ochranka nebo administrátoři), tak mechanická zařízení (zámky a klíče), stejně jako moderní technologická řešení, včetně systémů založených na přístupových kartách a biometrických identifikačních systémech.

Moderní systémy kontroly přístupu, které integrují senzory pro rychlejší a přesnější identifikaci návštěvníků, přispívají k minimalizaci lidského faktoru a zajišťují efektivní prevenci neoprávněného přístupu. Takové systémy mohou fungovat jak autonomně, tak v synergií s dalšími bezpečnostními systémy.

V současné době jsou mnohé systémy kontroly přístupu vyvíjeny na základě specializovaných mikrokontrolérů, což často vyžaduje, aby zákazníci nakupovali zařízení a služby od konkrétního výrobce. Zároveň existují otevřené projekty vývoje mikrokontrolérů, jako je Arduino, které poskytují své zdroje na podmínkách svobodného softwaru. Dostupnost informací o charakteristikách a konstrukci Arduino umožňuje vývojářům volně používat tuto platformu pro tvorbu elektronických zařízení. S ohledem na možnosti přeprogramování a snížení nákladů na vývoj bylo Arduino vybráno jako základ pro vývoj systému kontroly přístupu.

1. ANALÝZA SYSTÉMŮ ŘÍZENÍ A ŘÍZENÍ PŘÍSTUPU

1.1. Definice a hlavní možnosti systémů řízení a správy přístupu

Předtím, než se ponoříme do analýzy existujících systémů kontroly přístupu, je klíčové pochopit důvod, proč je čtečka RFID nezbytná.

Systémy kontroly přístupu představují komplexní soubor zařízení a technologií, které jsou navzájem technicky, informačně a provozně kompatibilní. Tato definice však neposkytuje úplné pochopení zkoumaného konceptu. Rozšířený pohled na systémy kontroly přístupu odhaluje, že jde o integrovaný soubor hardwaru a softwaru navržený k omezení a evidenci přístupu osob, vozidel a dalších objektů do určitých prostor, budov, zón a areálů. [1]).

Většinou systémy registrace přístupu zahrnují:

- zařízení pro fyzické omezení přístupu (například turnikety, dveře s elektronickými zámky, brány, závory, šleuzy);
- čtecí zařízení, jako jsou RFID čtečky, daktyloskopické skenery, zařízení pro strojové vidění;
- kontrolní moduly pro řízení přístupu, které jsou elektronické mikroprocesorové jednotky zajišťující autentizaci a autorizaci přístupu do specifických oblastí a správu zařízení pro omezení přístupu;
- softwarové vybavení pro centralizované řízení a správu systému, vytváření reportů a další funkce;
- konvertory a další hardwarové komponenty pro propojení systémových modulů a jejich připojení k počítači;
- doplňkové vybavení, jako jsou napájecí zdroje, tlačítka pro nouzové otevření a kabeláž.

Zásadním prvkem pro funkčnost systémů kontroly přístupu je identifikátor - unikátní kód nebo znak, který umožňuje identifikaci subjektu nebo objektu. Identifikátory mohou být různého typu, od jednoduchých kódů, přes biometrické znaky, až po fyzické nosiče s kódovanými informacemi, jako jsou karty, elektronické klíče nebo přívěsky.

Při zavádění systémů kontroly přístupu je nutné brát v úvahu specifické požadavky a podmínky jednotlivých organizací. Analýza potřeb například bank, kanceláří odhaluje specifické aspekty, které mohou být pro průmyslové podniky nebo státní instituce méně relevantní. Přesto všechny systémy kontroly přístupu obsahují základní funkce, které jsou standardně integrovány do každého systému:

- vedení a udržování databází uživatelů a karet/identifikátorů.
- uchovávání fotografií uživatelů v databázi.
- zaznamenávání data a času průchodu v databázi.
- nastavení úrovně přístupu.
- autonomní provoz systémových ovladačů při zachování základních řídicích funkcí v případě výpadku komunikace s počítačem.
- registrace a ukládání informací o událostech v energeticky nezávislé paměti regulátorů ACS.
- ukládání identifikačních znaků do systémové paměti pro případ poruchy a výpadku napájení.
- otevření UPU při čtení identifikačního znaku registrovaného v paměti systému.
- zákaz otevření řídicí jednotky při čtení identifikačního znaku, který není registrován v systémové paměti.



Obrázek 1 Identifikační metody [1]

Existují mnohé další funkce, které jsou více či méně důležité pro všechny spotřebitele, ale budeme se zabývat pouze jednou funkcí - čtením karty Mifare pro evidenci pracovního času.

Tento proces umožňuje pomocí doplňkového softwaru pro systémy kontroly přístupu efektivně monitorovat časy příchodů a odchodů studentů. Díky tomu mohou administrátoři nebo jmenované osoby přesně sledovat, jak dlouho se studenti zdržují v daném zařízení. Na počátku každého dne umožňuje integrovaný systém pro evidenci studijního času v rámci systému kontroly přístupu generovat přehledy o studentech, kteří neprošli kontrolními body. Tato funkce usnadňuje identifikaci studentů, kteří přišli pozdě nebo se na výuku vůbec nedostavili, a zvyšuje efektivitu administrativních procesů [2]. Flexibilita softwaru pro systémy kontroly přístupu navíc umožňuje vyučujícím snadno vytvářet potřebné reporty.

Systémy kontroly a evidenci přístupu (SKEP) neplní pouze funkci sledování pracovního času, ale jsou také klíčovým prvkem zabezpečení budov a organizací. V Evropě působí řada renomovaných výrobců SKEP, mezi které patří:

- 1) Bosch Security Systems (Německo) - známá světová značka, která nabízí širokou škálu produktů pro bezpečnostní systémy, včetně SKEP [3].
- 2) Siemens Building Technologies (Německo) - další přední světový hráč, který nabízí integrovaná řešení pro správu budov, včetně systémů kontroly přístupu [4].
- 3) Honeywell Security (zastoupení po celé Evropě) - mezinárodní výrobce, který nabízí různorodá řešení v oblasti bezpečnosti, včetně SKEP [5].

Tyto společnosti nabízejí od jednoduchých až po komplexní integrované bezpečnostní systémy, včetně videomonitoringu, alarmů a dalších funkcí.

Nicméně, výzkum ukázal, že moderní systémy čelí výzvám, jako jsou vysoké náklady na hardware, základní software, omezení při ukládání dat v databázích výrobců a komplikace při integraci komponent od různých dodavatelů. Z toho vyplývá potřeba výběru cenově dostupných komponent, optimalizace ukládání dat a vývoje zařízení a softwaru, které by byly srovnatelné s funkcemi nabízenými výše zmíněnými systémy. [23].

2. Formulace cíle a úkolů projektu

Cílem práce je vývoj prototypu zařízení a nezbytného softwaru pro identifikaci návštěvníků jakéhokoli objektu. Hlavními výhodami zařízení budou nízké náklady na komponenty a ukládání dat uvnitř mikrokontroléru, což umožní eliminaci úniku informací. Pro ukládání a ověřování informací se předpokládá použití databáze, pro zobrazení informací o návštěvníkovi bude použit LCD displej, pro čtení čísla karty - RFID modul.

Pro dosažení cíle práce je nutné splnit následující úkoly:

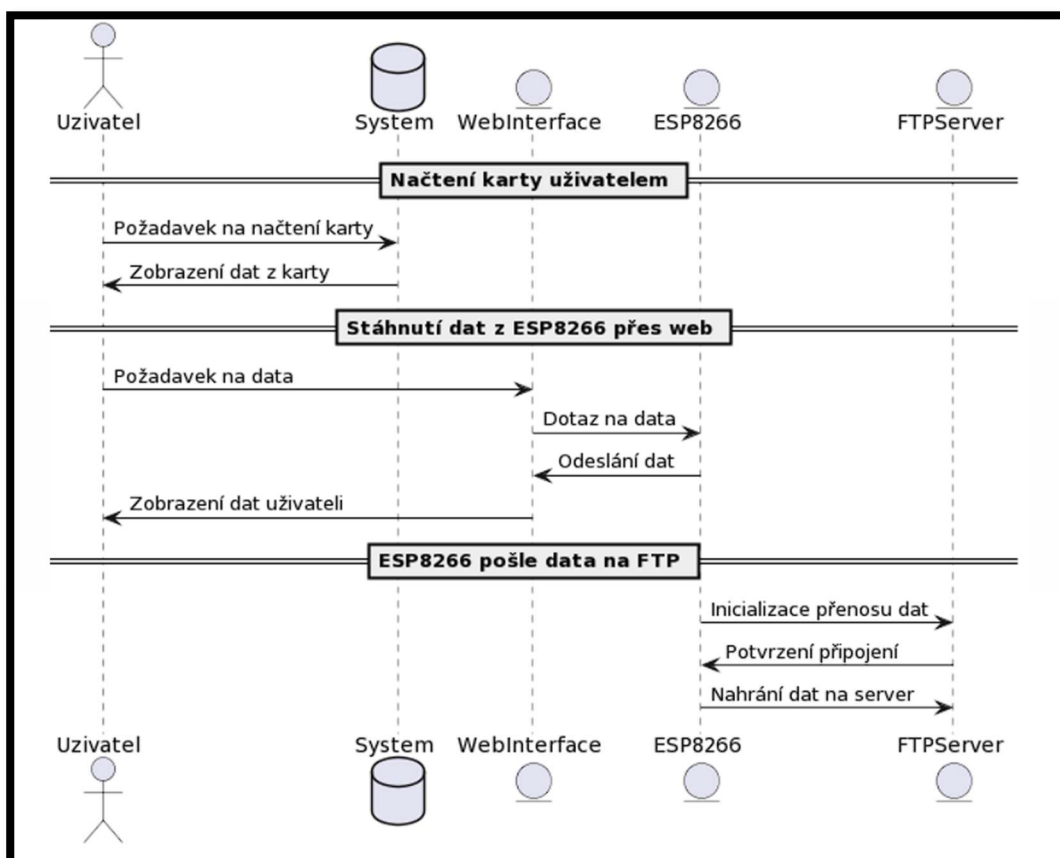
- 1) Přehled stávajících systémů kontroly a registrace přístupu
- 2) Vypracování sekvenčního diagramu
- 3) Výběr komponent
- 4) Vypracování schématu zapojení
- 5) Vývoj programů pro mikrokontrolér a PC
- 6) Montáž celého zařízení

3. Sekvenční diagram

3.1. Sekvenční diagram PlantUml

V souladu s předloženými cíli a požadavky bylo v programu plantuml vyvinuto sekvenční diagram systému, znázorněné na obrázku 2.

UML diagram popisuje interakci mezi uživatelem, systémem, webovým rozhraním, modulem ESP8266 a FTP serverem.



Obrázek 2: Kompletní blokové schéma PlantUML [autor]

PlantUML byl vybrán pro práci se čtečkou, protože poskytuje jednoduchý a pohodlný přístup k vytváření a úpravě sekvenčních diagramů. Diagramy jsou vytvářeny na základě textu, kde lze na rozdíl od grafických obrázků provádět změny, což na rozdíl od grafických editorů umožňuje snadné a rychlé úpravy. PlantUML chytře využívá jednoduchost vkládání textu k vytváření vizuálně popisných a přesných diagramů. Silná korelace mezi textem a grafickým výsledkem nejen zjednodušuje, ale také výrazně urychluje vývojový proces. Uživatelé těží z efektivnějšího a rychlejšího procesu, kdy je méně času stráveného přepracováním a nastavováním.

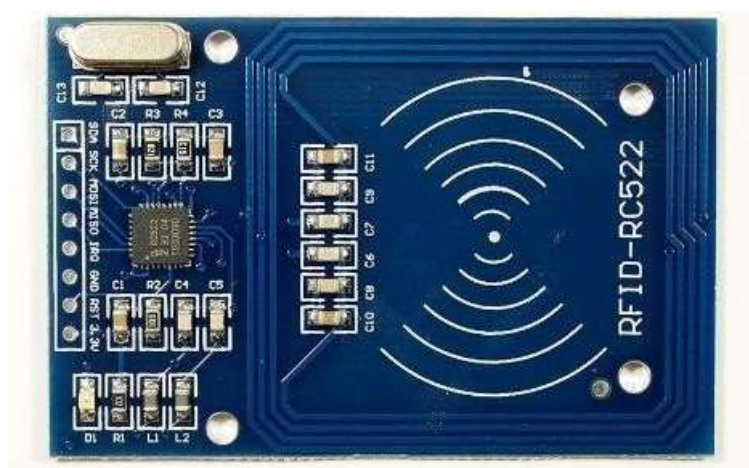
Diagram ukazuje architekturu systému, která podporuje získávání dat, interakci s uživatelem, zpracování dat v reálném čase a vzdálené ukládání nebo vyhledávání dat. To vše zdůrazňuje integraci hardwarových modulů se softwarovými rozhraními pro komplexní správu dat [6].

4. Výběr komponent zařízení

4.1. Čtečka RFID a RFID značka

Technologie RFID je technologie bezkontaktního přenosu dat, založená na využití rádiových frekvenčních elektromagnetických vln. RFID se využívá pro automatickou identifikaci a evidenci objektů. Typický RFID systém se skládá ze tří základních komponent: RFID značek, RFID čteček a softwaru [7].

Existují různé typy čteček, v tomto systému se používá portální RFID čtečka, která je určena pro registraci RFID značek v kontrolovaných průchodech. RFID-RC522 je vysoce integrovaná čtečka pro bezkontaktní komunikaci. Čtečka podporuje rozhraní SPI, UART a I2C, prostřednictvím kterých probíhá výměna dat s dalšími zařízeními [8]]. Na desce modulu je pro vývody čipu zvoleno rozhraní SPI. Základem modulu je čip MFRC522.



Obrázek 3: RFID čtečka-RC522 [autor]

Čtečka RFID RC522 reaguje při přiblížení značky. Výměna dat probíhá prostřednictvím rámcových antén, které se nacházejí v kartě (značce) a v modulu. Signál modulu poskytuje energii pro značku. Může zpracovávat informace současně z několika značek.

Technické specifikace:

1. Napájecí napětí: 3,3 V;
2. Proudová spotřeba v režimech:
 - 2.1 Pohotovostní 80 μ A;
 - 2.2 Čekání 12 mA;
 - 2.3 Maximální 30 mA;
3. Pracovní frekvence: 13,56 MHz;
4. Dosah čtení 0-25 mm;
5. Rychlost přenosu dat:
 - 5.1 106 kbit/s;
 - 5.2 212 kbit/s;
 - 5.3 424 kbit/s;
 - 5.4 848 kbit/s;

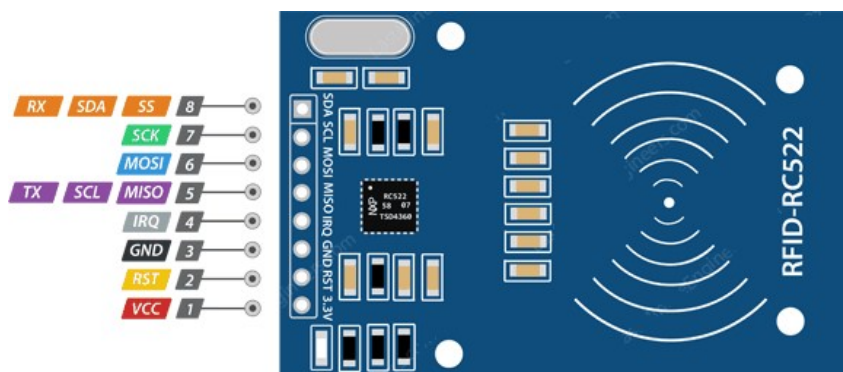
Tato čtečka je vybavena RFID štítkem v pouzdře nebo ve formě klíčové karty.

Klíčenky a karty fungují na stejné frekvenci jako čtečky. Uvnitř se nachází anténa a čip Mifare S50 obsahující paměť o velikosti 1 kilobajt typu EEPROM. Jedinečnost karty Mifare je zajištěna tím, že výrobce přiřadí číslo, které slouží jako indikátor.



Obrázek 4: RFID štítky v pouzdrech [autor]

RFID čtečka-RC522



Obrázek 5: Pinout čtečky RC522 [9]

Napájení pro čtečku se přivádí na pin VCC. Napětí napájení se pohybuje v rozmezí od 2,5 do 3,3 V. VCC by mělo být připojeno k pinu 3v3, který se nachází na desce NodeMCU. Pin RST slouží pro reset a vypnutí napájení. Při přijetí signálu LOW (logická nula), dojde k restartování čtečky.

Pin GND – je zemní vývod.

IRQ – vývod pro přerušení, určený k probuzení zařízení, když se v blízkosti nachází RFID značka.

Vývod **MISO/SCL/TX** – slouží pro komunikaci pomocí protokolu SPI, I2C nebo UART. Tento vývod funguje buď jako MISO (Master In Slave Out) při aktivním rozhraní SPI, nebo jako sériový taktový signál při aktivním rozhraní I2C, nebo jako výstup sériových dat při aktivním rozhraní UART.

MOSI (Master Out Slave In) - je vstupní SPI pin pro čtečku RC522. MOSI slouží k přenosu dat od hlavního zařízení k podřízenému.

MISO - slouží k přenosu dat od podřízeného zařízení k hlavnímu.

Pin **SCK** přijímá hodinové impulsy poskytované hlavním zařízením na sběrnici SPI. Slouží pro synchronizaci.

Vývod **SS/SDA/Rx** funguje buď jako vstup, když je aktivní rozhraní SPI, nebo jako linka sériových dat, když je aktivní rozhraní I2C, nebo jako vstup sériových dat, když je aktivní rozhraní UART. Tento vývod je obvykle označen obdélníkem, aby sloužil jako referenční bod pro identifikaci ostatních pinů [9]).

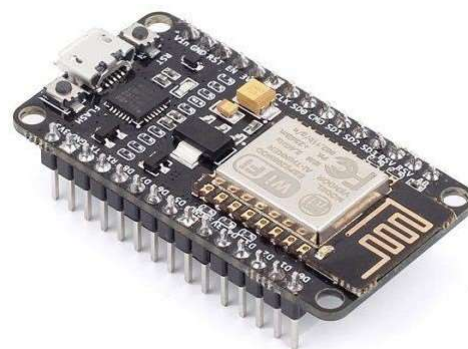
4.2. Hardwarová platforma

Z velké nabídky desek Arduino je možné vyzdvihnout dva modely, které by mohly být vhodné jako řídicí zařízení:

- Arduino Nano
- NodeMCU

NodeMCU je platforma založená na modulu ESP8266. Deska má prakticky podobné charakteristiky jako Arduino Nano, ale díky vestavěnému Wi-Fi modulu nabízí širší možnosti použití. Wi-Fi modul lze využít pro dálkové ovládání různých obvodů prostřednictvím přenosu signálu do lokální sítě nebo internetu.

Na přední straně desky se nachází konektor Micro USB, který se používá pro programování kontroleru. Tento konektor lze také použít pro napájení. Vedle konektoru se nacházejí dvě tlačítka: Flash a Reset. Tlačítko Flash se používá pro ladění, tlačítko Reset pro restartování desky.



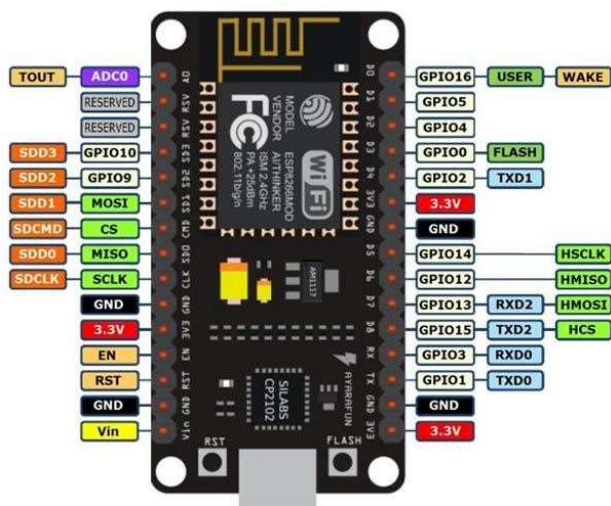
Obrázek 6: Hardwarová platforma NodeMCU [autor]

Technické specifikace modulu:

- Podpora Wi-Fi protokolu 802.11 b/g/n;
- Podporované režimy Wi-Fi – přístupový bod, klient;
- Vstupní napětí 3,7 V – 20 V;
- Pracovní napětí 3 V - 3,6 V;
- Maximální proud 220 mA;
- Vestavěný TCP/IP stack;
- Rozsah pracovních teplot od -40 °C do 125 °C;
- Frekvence 80 MHz, 32bitový procesor;
- Čas probuzení a odeslání paketů 22 ms;
- Vestavěný TR přepínač a PLL;

Přítomnost zesilovačů výkonu, regulátorů a systémů řízení napájení. Jako řídicí zařízení byly zváženy především desky Arduino Nano a NodeMCU, jelikož tyto desky mají kompaktní rozměry a splňují požadovanou funkcionalitu. Každá z desek disponuje dostatečným počtem vývodů pro připojení modulů. Nakonec byla vybrána deska NodeMCU, protože díky vestavěnému Wi-Fi modulu umožňuje přenos a příjem dat po síti a práci s ftp službou, a navíc nabízí rozšířené funkce [10]).

Hardwarová deska NodeMCU



Obrázek 7: Pinout hardwarové desky NodeMCU [11])

4.3. Popis a účel vývodů NodeMCU V3 ESP8266

Digitální vstupy/výstupy (GPIO):

GPIO (General Purpose Input Output) - kontakty pro vstup/výstup obecného účelu. Mohou být konfigurovány jako vstupy nebo výstupy a programově přiřazeny k různým funkcím.

Tabulka 1 kontakty GPIO:

Kontakty	Funkce vstupů/výstupů kontaktů
D0 (GPIO16):	může být použit pro probuzení čipu z hlubokého spánku.
D1 (GPIO5):	používá se pro I2C SCL.
D2 (GPIO4):	používá se pro I2C SDA.
D3 (GPIO0):	má integrovaný pull-up rezistor k napájení; používá se při spouštění pro programovací režim.
D4 (GPIO2):	má integrovaný pull-up rezistor k napájení a je spojen s integrovanou LED diodou na většině desek NodeMCU.
D5 (GPIO14):	používá se pro SPI SCLK.
D6 (GPIO12):	používá se pro SPI MISO.
D7 (GPIO13):	používá se pro SPI MOSI.
D8 (GPIO15):	má integrovaný pull-down rezistor k zemi; používá se pro SPI SS. Tento pin by měl být při zapnutí spojen se zemí pro normální spuštění.
D9 (RX, GPIO3):	přijímač pro UART. Může být použit jako GPIO.
D10 (TX, GPIO1):	vysílač pro UART. Může být použit jako GPIO.

Tabulka 2 Speciální piny:

Kontakty	Funkce vstupů/výstupů kontaktů
SD2, SD3 (GPIO9 a GPIO10):	používány v některých modelech ESP8266 pro připojení k interní flash paměti. Na většině desek NodeMCU nejsou tyto piny dostupné.
ADC0 / TOUT	pin integrovaného 10-bitového analogově-digitálního převodníku (ADC). Převezené hodnoty se pohybují v rozsahu 0-1023. Vývojová deska NodeMCU V2 má interní dělič napětí, vstupní rozsah ADC je 0 - 3,3 V.
GND	společný vodič, „zem“.

Tabulka 3 Napájecí vývody:

Kontakty	Funkce vstupů/výstupů kontaktů
Vin	pin pro připojení externího zdroje napájení 5V. Stabilizátor AMS1117-3.3 umožňuje připojení napájení na Vin v širokém rozsahu od 5 do 10 V. I když stabilizátor umožňuje vyšší napětí (až 15 V), bez dodatečného chlazení může dojít k přehřátí čipu.
3.3V	pin, na který je přivedeno výstupní napětí vnitřního stabilizátoru. Může být použit pro napájení senzorů připojených k desce. Celková maximální zátěž všech vývodů 3.3V by neměla přesáhnout 300mA.
USB	pin, na který je přivedeno napětí 5V z USB konektoru.

Tabulka 4 Řídící piny:

Kontakty	Funkce vstupů/výstupů kontaktů
RST (Reset)	pin používaný pro reset mikrokontroléru ESP8266.
EN (Chip Enable)	při přivedení signálu vysoké úrovně na tento pin, mikrokontrolér ESP8266 přechází do pracovního režimu, při signálu nízké úrovně do režimu úspory energie.
WAKE	pin používaný pro probuzení čipu ESP8266 z režimu hlubokého spánku (deep-sleep mode).

SPI (Serial Peripheral Interface) - sériové periferní rozhraní. NodeMCU má dva SPI (SPI a HSPI) v režimech master a slave.

Tabulka 5 Komunikační rozhraní:

Kontakty	Funkce vstupů/výstupů kontaktů
SPI	přes piny D5 (SCLK), D6 (MISO), D7 (MOSI), D8 (SS) pro hardwarové SPI.
SDIO	rozhraní pro bezpečné digitální vstupy/výstupy, určené pro komunikaci s externí flash pamětí standardu SD přes sériovou sběrnici.
Rozhraní I2C	sériová asymetrická sběrnice. I2C se používá pro připojení senzorů a periferních zařízení. NodeMCU ESP8266 nemá hardwarové piny pro I2C, ale rozhraní lze implementovat softwarově. Podporuje se jak I2C Master, tak I2C Slave. Obvykle se jako piny I2C používají následující vývody.

FLASH

Tlačítko Flash na NodeMCU připojuje GPIO0 k zemi. Může být použito jako běžné tlačítko. Pokud je programově připojen pin GPIO0 pomocí interního pull-up rezistoru k vysoké úrovni, nízká úroveň na tomto pinu znamená, že bylo tlačítko stisknuto [11]).

Tabulka 6 Tabulka rozhraní mikrokontroléru:

Kontakty	Funkce vstupů/výstupů kontaktů
SPI	přes piny D5 (SCLK), D6 (MISO), D7 (MOSI), D8 (SS) pro hardwarové SPI.
PWM (pulse-width modulation)	šířková impulzní modulace (PWM) řídí výkon metodou pulzujícího zapínání a vypínání pinu. NodeMCU podporuje softwarový PWM na pinech označených na obrázku zakřivenou čarou.
UART	asynchronní sériové rozhraní umožňuje komunikaci s jinými zařízeními přes sběrnici UART.

Stabilizátor napětí CP2102 přijímá vyšší vstupní napětí (například 5V z USB portu) a převádí ho na nižší stabilizované výstupní napětí (3.3V), které je potřebné pro ESP8266.

Použití stabilizátoru napětí chrání ESP8266 před poškozením, které by mohlo být způsobeno příliš vysokým napětím. Stabilizátor zajišťuje stabilní napájecí napětí i při kolísání napětí na vstupu, což je klíčové pro stabilní provoz mikrokontroléru.

Deska NodeMCU založená na ESP8266 obvykle obsahuje interní dělič napětí pro analogový vstup (ADC). To je dáno tím, že ESP8266 podle svých specifikací podporuje maximální napětí na analogovém vstupu přibližně 1V, zatímco mnoho uživatelských aplikací vyžaduje měření napětí v širším rozsahu [12)].

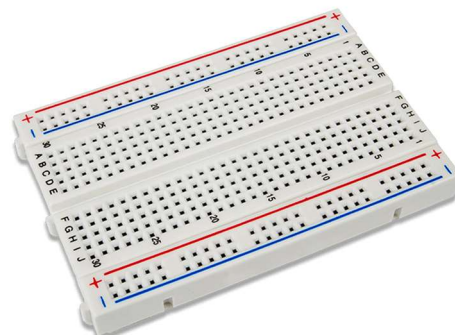


Obrázek 8: Převodník CP2102 (univerzální adaptéry pro připojení ESP modulů k počítači) [11)]

4.4. Pájecí prkénko

K sestavení prkénka na krájení zařízení bez použití páječky bylo řešením použít nepájivé prkénko. Takové desky se v anglicky mluvícím prostředí nazývají Breadboard.

Vývojová deska se skládá z ABS plastového pouzdra obsahujícího rozebíratelné spoje, které připomínají dvojité kovové tyče, mezi kterými je sevřen vodič. Na přední části pouzdra jsou očíslované a označené otvory, do kterých lze vložit vodiče, nožky mikroobvodů, tranzistory a další rádiové komponenty v pouzdrech s přívody.



Obrázek 9: Arduino Breadboard [autor]

Na uvažované desce plošných spojů jsou vnější dva sloupce otvorů na každé straně vertikálně kombinovány se společnými sběrnicemi, z nichž se obvykle tvoří kladná sběrnice zdroje a minus (společná sběrnice). Obvykle je označen červeným a modrým pruhem podél okraje desky, plus a minus. Střední část desky je rozdělena na dvě části, každá část je na této konkrétní desce spojena v řadě pěti otvorů v řadě.

Princip připojení je následující: do otvorů vložíme nožičky elektronických součástek a propojky, spojíme díly dohromady podél vodorovných linií a napájíme z nejkrajnějších svislých čar.

Rozhodnutí o použití nepájivých prkének padlo z důvodu jednoduchosti obvodu, který nekladal vysoké nároky na elektrické zapojení a přesnost, automatizaci a digitální obvody a také vysokonapěťový provoz [13)].

4.5. Propojovací kabely

Propojovací kabely Dupont jsou nezbytné pro prototypování elektronických projektů, včetně těch, které využívají platformu Arduino a další systémy mikrokontrolérů. Tyto kabely poskytují rychlé a flexibilní spojení mezi různými elektronickými součástmi, jako jsou senzory, přepínače, displeje a mezi kontaktními plochami a vývojovými deskami.

Kabely Dupont jsou k dispozici v mnoha stylech, včetně zástrček, zásuvek a kombinovaných konektorů, což vám umožní snadno připojit prakticky jakoukoli elektronickou součást bez nutnosti pájení.

V kontextu návrhu elektronických zařízení a prototypování zjednodušují kabely Dupont proces testování a ladění tím, že poskytují snadno odpojitelné připojení, které lze rychle změnit nebo upravit. Díky své všestrannosti a snadnému použití jsou kabely Dupont považovány za nedílnou součást laboratoří, výzkumných center a vzdělávacích institucí zabývajících se elektronikou a robotikou.



Obrázek 10: Connecting cables [autor]

4.6. I2C 1602 LCD Display Module 16X2

Displej I2C 1602 LCD je znakový displej z tekutých krystalů schopný zobrazit 16 znaků v každém ze dvou řádků. Použití rozhraní I2C (Inter-Integrated Circuit) umožňuje ovládat displej pomocí pouhých dvou vodičů, což výrazně zjednodušuje připojení k mikrokontrolérům jako je ESP8266 a minimalizuje počet použitých GPIO pinů.

Rozhraní I2C zahrnuje dvě hlavní linky: SDA (Serial Data Line) a SCL (Serial Clock Line). Linka SDA se používá k přenosu dat mezi zařízeními na sběrnici I2C, zatímco SCL poskytuje hodinové impulsy, které tento přenos dat koordinují. Každé zařízení na sběrnici I2C má jedinečnou adresu, což umožňuje mikrokontroléru komunikovat s konkrétním zařízením bez rušení ostatních připojených zařízení.



Obrázek 11: I2C displej LCD 1602 [14]

Displej I2C 1602 má obvykle modul I2C, buď již integrovaný na desce, nebo zapojený samostatně, který má na desce čip pro převod I2C signálů na paralelní pro řízení LCD displeje. To umožňuje vývojářům odesílat příkazy a data pro zobrazení textu pomocí standardních knihoven dostupných pro mnoho vývojových platforem.

V projektu diplomové práce je LCD displej použit pro zobrazení informací přijatých z mikrokontroléru ESP8266, jako je stav Wi-Fi připojení, výsledky čtení z RFID modulu MFRC522 nebo jiné důležité informace pro uživatele. Připojení displeje přes I2C výrazně

zjednodušuje zapojení obvodů a programování snížením počtu potřebných vodičů a zjednodušením kódu ovládání displeje.

Znakové LCD displeje, jako je I2C 1602, nabízejí nízkou spotřebu energie a vysokou spolehlivost, díky čemuž jsou ideální volbou pro různé aplikace, které vyžadují vizuální, jasné zobrazení informací v reálném čase [14]).

5. Napájení zařízení

Mikrokontrolér ESP8266 může být napájen z různých zdrojů v závislosti na konkrétní aplikaci a požadovaném výkonu. Zde jsou hlavní možnosti napájení:

USB napájení: ESP8266 je často napájeno přes USB při připojení k počítači nebo USB adaptéru. Napětí USB je obvykle 5 voltů.

Baterie: Mikrokontrolér může být napájen z baterií. To mohou být AA nebo AAA baterie, lithium-iontové nebo lithium-polymerové akumulátory. Je důležité zajistit správné napětí a stabilitu proudu. ESP8266 obvykle vyžaduje 3,3 volta.

Stabilizované zdroje napájení: Pokud máte zdroj napájení, který může poskytnout stabilní 3,3 volta, je to ideální volba pro ESP8266. Je nutné zajistit, aby napětí nepřekročilo 3,6 volta, protože to může zařízení poškodit.

Regulátory napětí: Pokud zdroj napájení poskytuje vyšší napětí, než je potřeba (například 5V z USB nebo 12V z jiných zdrojů), lze použít regulátor napětí pro snížení na 3,3 volta [8].

V rámci této diplomové práce byl jako zdroj napájení pro mikrokontrolér ESP8266 zvolen notebook. Toto rozhodnutí bylo motivováno několika klíčovými faktory, které jsou rozhodující pro úspěšnou realizaci projektu:



Obrázek 12: Baterie Samsung [15)]

Stabilita napětí a proudu: Notebook poskytuje stabilní výstupní proud přes USB port, což je velmi důležité pro spolehlivou funkci ESP8266. Vzhledem k tomu, že mikrokontrolér je citlivý na kolísání napětí, použití notebooku jako zdroje napájení zaručuje absence náhlých skoků napětí, které by mohly zařízení poškodit.

Pohodlí připojení a nastavení: Notebook nabízí pohodlný způsob, jak připojit ESP8266 přes USB port, což zjednodušuje proces programování a ladění mikrokontroléru. USB rozhraní umožňuje nejen napájet mikrokontrolér, ale také zajišťuje jeho komunikaci s vyvíjeným softwarem.

Firmware a programování: Důležitým aspektem použití notebooku je možnost firmware ESP8266. Proces programování vyžaduje nejen elektrické napájení, ale i data přenášena přes USB port. Notebook umožňuje snadno nahrávat firmware a provádět nezbytné testy, což zajišťuje efektivní vývojový proces.

Mobilita a dostupnost: Použití notebooku jako zdroje napájení poskytuje vysokou míru mobility. To umožňuje provádět vývoj a testování v různých podmínkách bez nutnosti připojení k pevným zdrojům napájení.

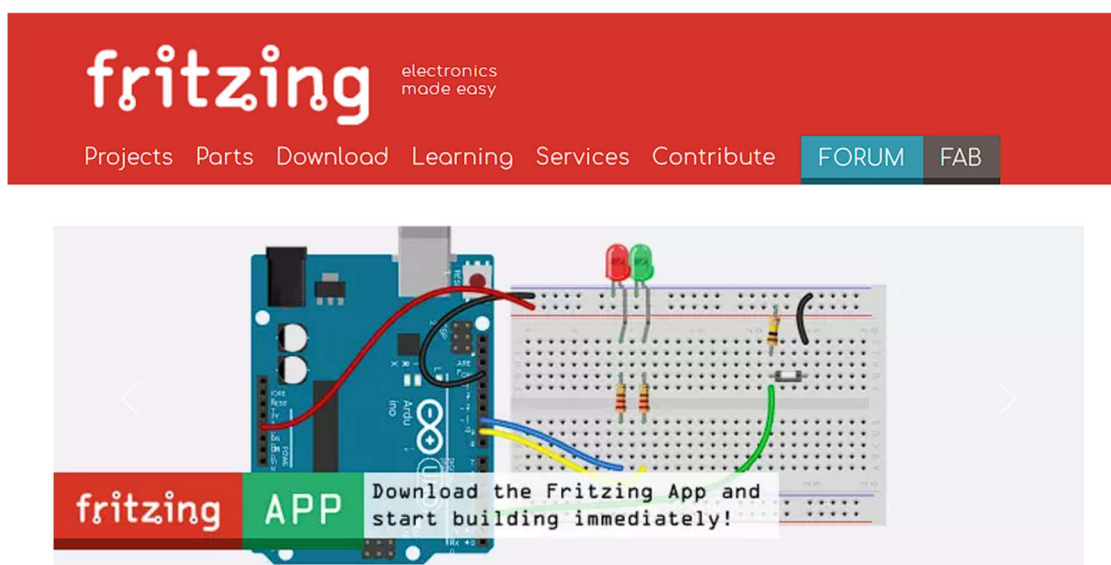
Závěrem, výběr notebooku jako zdroje napájení pro ESP8266 v této diplomové práci je odůvodněn potřebou zajištění stabilního napájení, pohodlí během vývoje a programování, stejně jako flexibilitou a mobilitou pracovního procesu. Toto řešení přispívá k efektivní realizaci projektu a dosažení stanovených cílů [15]).

6. Kompletní blokové schéma

Po porovnání a výběru komponent pro řídicí zařízení bylo vypracováno kompletní schéma zapojení zařízení.

Program Fritzing byl použit k vývoji a vizualizaci schématu zapojení pro komponenty projektu. Fritzing je open-source software pro návrh a tvorbu elektronických obvodů, který umožňuje inženýrům a designérům vizualizovat rozvržení prkénka, vytvářet schematické výkresy a navrhovat desky plošných spojů.

Fritzing byl vybrán pro své intuitivní rozhraní a rozsáhlou knihovnu elektronických součástek, které mu umožnily přesně zobrazit všechny prvky projektu, včetně mikrokontroléru ESP8266, RFID modulu MFRC522, převodníku USB-UART, vývojové desky a LCD displeje s I2C modulem. Schéma zapojení bylo navrženo s velkým důrazem na detail, aby bylo zajištěno, že všechny vodiče a připojení jsou jasné a snadno srozumitelné (viz. obrázek 13).



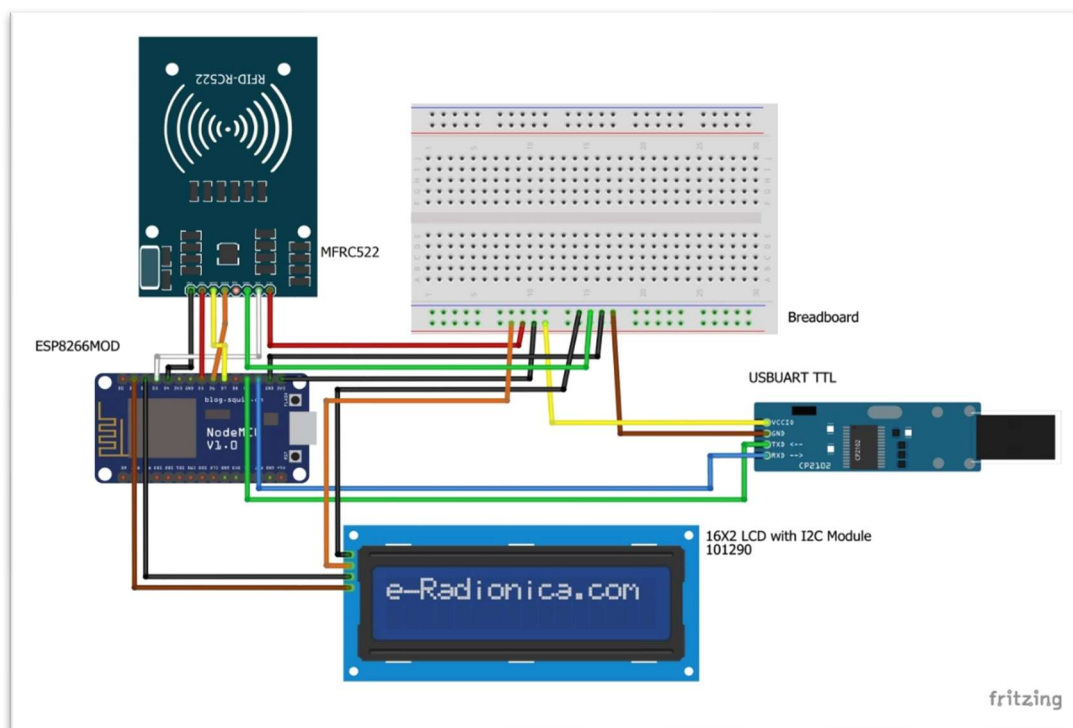
Obrázek 13: Fritzing free cross-platform software for projects Arduino [16])

Použití Fritzingu umožnilo nejen graficky znázornit konfiguraci návrhu, ale také zjednodušit proces montáže a ladění, protože každý komponent a jeho připojení bylo možné snadno identifikovat. Program tak usnadnil vývoj a poskytl vizuální pomůcku pro dokumentaci projektu v rámci diplomové práce.

Kromě vytvoření obvodu byl Fritzing použit k testování spojení a kolíků, které jsou kritické pro fungování mikrokontroléru a interakci s ostatními součástmi. Toto opatření zajistilo správnou funkci systému od prvního zapnutí a zamezilo možným chybám připojení.

Tato část práce představuje schéma zapojení mezi modulem NodeMCU ESP8266, modulem RFID MFRC522 a LCD displejem 1602. Schéma zapojení bylo implementováno tak, aby byla zajištěna komunikace mezi těmito zařízeními za účelem identifikace a zobrazení informací [16]).

6.1. Vypracování schématu elektrického obvodu zařízení:



Obrázek 14: Device connection diagram developed in the program Fritzing [autor]

Tabulka 7 Připojení NodeMCU ESP8266 k RFID MFRC522

Pin na NodeMCU	Pin na MFRC522	
D3	RST	používá se pro resetování RFID modulu.
D4	SDA/SS	slouží pro výběr aktivního zařízení na sběrnici SPI
D5	SCK	poskytují synchronizační signály hodin pro SPI spojení.
D6	MISO	umožňující přenos dat od RFID modulu k mikrokontroléru
D7	MOSI	zajišťující přenos dat od mikrokontroléru k RFID modulu
GND (zem)	GND (zem)	obou zařízení jsou spojeny, aby byla zajištěna společná zemní bod.
3V3	3.3V	pro napájení modulu

Tabulka 8 Připojení NodeMCU ESP8266 k LCD 1602

Pin na NodeMCU	Pin na LCD 1602	
D2	SDA	což slouží pro přenos dat na sběrnici I2C
D1	SCL	poskytuje synchronizaci přenosu dat přes I2C
GND	GND	obou zařízení jsou spojeny, zajišťující společný zemní obvod.
VCC	VCC	zajištění elektrického napájení.

Tabulka 9 Připojení USB TTL CP2102 k ESP8266:

Pin na USB TTL	Pin na ESP8266	
GND	GND	pro vytvoření společného zemního bodu.
3V3	3.3V	poskytující potřebné napájení pro mikrokontrolér
TX	RX	umožňující přenos dat z USB do mikrokontroléru.
RX	TX	zajišťující přenos dat od mikrokontroléru k USB.

Každé z těchto připojení je kriticky důležité pro fungování systému: zajišťují nejen elektrické napájení komponent, ale i obousměrný přenos dat mezi zařízeními. Správné provedení těchto připojení je klíčové pro fungování vyvíjeného systému.

6.2. Logika připojení zařízení:

NodeMCU ESP8266 to RFID MFRC522

D4 <————> **SDA/SS**

D5 <————> **SCK**

D7 <————> **MOSI**

D6 <————> **MISO**

GND <————> **GND**

D3 <————> **RST**

3V/3V3 <————> **3.3V**

NodeMCU ESP8266 to LCD 1602:

• **D2** <————> **SDA**

• **D1** <————> **SCL**

• **GND** <————> **GND**

• **VV or VCC** <————> **VCC**

USB TTL to esp8266:

• **GND** <————> **GND**

• **3V/3V3** <————> **3.3V**

TX <————> **RX**

RX <————> **TX**

6.3. Zařízení založené na platformě NodeMCU se skládá z následujících bloků:

- čtečka RFID-RC522
- Hardwarová platforma NodeMCU
- USB TTL CP2102
- LCD displej

7. Připojení Arduina k počítači

Arduino se k počítači připojuje pomocí USB. U modelu Arduino NodeMCU je to implementováno pomocí čipu FT232R nebo jeho ekvivalentu CH340. Tyto součástky nejsou mikrokontroléry, ale převodníky úrovní signálů, což usnadňuje vlastnoruční sestavení Arduino NodeMCU od základů.

Vzhledem k tomu, že deska ESP8266 pracuje na napětí 3,3 V a je napájena přes micro-USB připojení na 5 V prostřednictvím 3,3 V napěťového regulátoru nebo přímo z pinu 3,3 V, piny nevydrží externí napětí 5 V. Maximální proud dodávaný z jednoho pinu je 12 mA, což vedlo k rozhodnutí použít USB-UART jako převodník napětí. Nejjednodušší převodníky CH340, CPL2102 a FT232R jsou dostupné ve verzích pro 3,3 V a 5 V a často obsahují jumper pro přepnutí napájecího napětí. Pro zprovoznění bylo nutné nainstalovat ovladač CP210x USB-to-UART bridge pro desku NodeMCU ESP8266 [10].

8. Arduino a sériový port UART

Protokol komunikace přes sériový port UART umožňuje dvoustannou výměnu dat mezi dvěma zařízeními. Oba účastníci komunikace jsou na stejné úrovni, což znamená, že není stanoveno rozdělení na hlavní (master) a vedlejší (slave) zařízení. Pro přenos dat jsou potřeba jen dvě vodiče: jeden slouží k přenosu dat od prvního zařízení k druhému a druhý v opačném směru. Na mikrokontroléru jsou pro tento účel vyhrazeny dva piny, označované jako TX (transmit, vysílací) a RX (receive, přijímací). Přenosové linky spojují TX pin prvního zařízení s RX pinem druhého a naopak. Arduino Uno je vybaveno jedním hardwarovým sériovým portem připojeným k USB portu. Arduino Mega poskytuje tři další sériové porty [17].

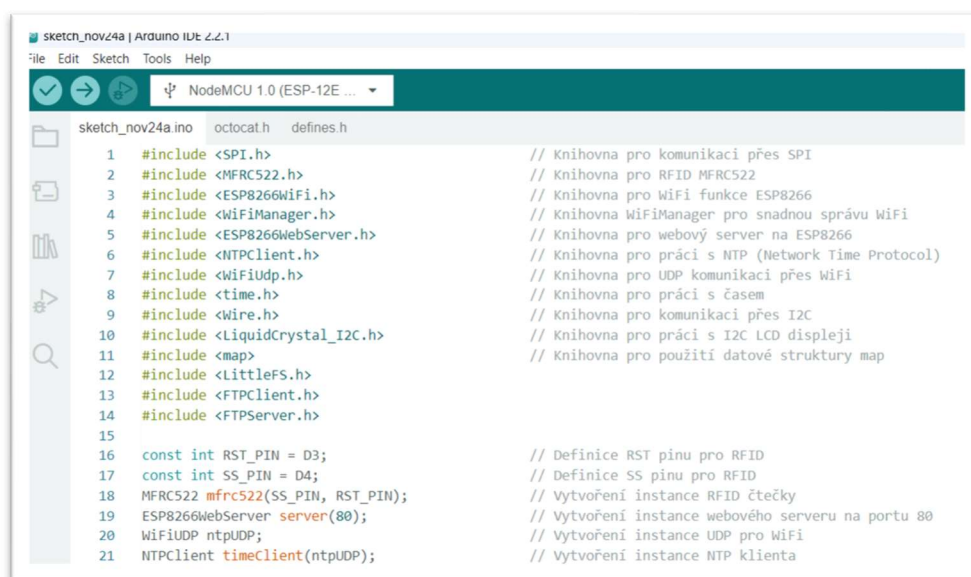
9. Softwarová část

9.1. Psaní programu pro mikrokontrolér

Vývoj vlastních aplikací založených na deskách kompatibilních s architekturou Arduino probíhá v bezplatném vývojovém prostředí Arduino IDE. Prostředí je navrženo pro zápis, kompilaci a nahrávání vlastních programů do paměti mikrokontroléru, instalovaného na desce kompatibilní s Arduino. Hlavním vývojovým prostředím je jazyk Processing/Wiring — jedná se v podstatě o běžný jazyk C++, doplněný o jednoduché a intuitivní funkce pro ovládání vstupu/výstupu na kontaktech. Existují verze prostředí pro operační systémy Windows, Mac OS a Linux. Pro napsání softwarové části bylo rozhodnuto použít integrované vývojové prostředí Arduino IDE 2.2.1.

Nová hlavní verze Arduino IDE 2.2.1 se stala rychlejší a ještě výkonnější. Kromě modernějšího editoru a citlivějšího rozhraní má automatické dokončování, navigaci v kódu a dokonce i živý debugger [18].

Nová hlavní verze Arduino IDE se stala rychlejší a ještě výkonnější. Kromě modernějšího editoru a citlivějšího rozhraní má automatické dokončování, navigaci v kódu a dokonce i živý debugger (viz. obrázek 15).



Obrázek 15: Arduino IDE interface [autor]

Možnosti programovacího prostředí Arduino lze výrazně rozšířit pomocí knihoven. Knihovny rozšiřují funkcionalitu programů a nesou další funkce - například pro práci s hardwarem, funkce zpracování dat apod. Řada knihoven se instaluje automaticky spolu s vývojovým prostředím, ale můžeme si stáhnout i další knihovny nebo si vytvořit vlastní. Používání knihoven výrazně zjednodušuje práci na projektech, protože umožňuje soustředit se na hlavní logiku programu bez plýtvání časem na mnoho detailů.

Pro implementaci softwarové části práce byly z platformy Arduino staženy potřebné knihovny.

Tabulka 10 Knihovny určené pro práci s Arduino

Knihovny	Implementace knihoven
SPI.h	Knihovna SPI umožňuje interakci se zařízeními SPI (sériové periferní rozhraní), jako jsou senzory a karty SD. Jedná se o digitální komunikační protokol používaný mikrokontroléry k rychlé komunikaci s jedním nebo více zařízeními na krátké vzdálenosti.
MFRC522.h	Tato knihovna se používá k propojení se čtečkou RFID MFRC522. Umožňuje číst a zapisovat RFID tagy, které se často používají v systémech kontroly přístupu.
ESP8266WiFi.h	Tato knihovna je specifická pro čipovou sadu ESP8266 a používá se ke správě integrované Wi-Fi. Poskytuje možnost připojení mikrokontroléru k síti Wi-Fi nebo vytvoření vlastní sítě.
WiFiManager.h	WiFiManager je knihovna, která umožňuje spravovat připojení Wi-Fi. Můžete jej použít ke správě uložených sítí, připojení k novým sítím nebo dokonce k nastavení zařízení jako hotspotu.
ESP8266WebServer.h	Tato knihovna umožňuje ESP8266 fungovat jako webový server. Můžete definovat koncové body a obsluhovat stránky HTML nebo jiné odpovědi HTTP.
NTPClient.h	Klient Network Time Protocol (NTP) se používá k synchronizaci času mikrokontroléru s internetovým časovým serverem. To je užitečné pro aplikace, které potřebují udržovat přesný čas.
WiFiUdp.h	Knihovna WiFiUDP se používá k odesílání a přijímání paketů UDP přes Wi-Fi. UDP je komunikační protokol, který umožňuje odesílat zprávy bez předchozího navázání spojení s příjemcem.
time.h	Toto je standardní knihovna C pro funkce související s časem. Umožňuje převádět formáty času, provádět aritmetiku času a získat aktuální čas a datum.
Wire.h	Knihovna Wire se používá pro komunikaci I2C (inter-integrated circuit). I2C je komunikační protokol běžně používaný mikrokontroléry pro komunikaci se senzory a dalšími zařízeními.
LiquidCrystal_I2C.h	Používá se k ovládání I2C LCD. Umožňuje zobrazovat text na displeji pomocí komunikačního protokolu I2C.
map	Knihovna C++, která poskytuje datovou strukturu pro ukládání párů klíč-hodnota. Často se používá k vytvoření vyhledávací tabulky nebo slovníku.
LittleFS.h	LittleFS je knihovna systému souborů pro práci se soubory ve flash paměti ESP8266. To je užitečné pro ukládání dat po restartování nebo pro poskytování souborů přes webový server.
FTPClient.h	Řídí FTP (File Transfer Protocol) komunikaci a umožňuje ESP8266 připojit se k FTP serveru a nahrávat nebo stahovat soubory.
FTPServer.h	Používá se k vytvoření FTP serveru na ESP8266, což umožňuje dalším zařízením připojit se k ESP8266 a nahrávat nebo stahovat soubory.

Společně tyto knihovny poskytují robustní základ pro vytvoření zařízení připojeného k síti, které dokáže číst RFID tagy, obsluhovat webové stránky, spravovat soubory a ukládat přesný čas. Kód funguje pro docházkový systém, který k identifikaci využívá RFID a pro připojení k síti využívá Wi-Fi [19].

Po připojení všech potřebných knihoven můžeme začít psát program. Psaní programu bylo pro pohodlí rozděleno do částí.

Pro mikrokontrolér je potřeba napsat program, který bude reagovat na RFID kartu obsahující unikátní 12místný kód přinesený do modulu. Mikrokontrolér odešle přijatý kód do PC pomocí WIFI modulu, který se připojí k routeru, a pokud není přístupový bod, vytvoří si vlastní přístupový bod.

10. ESP8266 - mikrokontrolér s Wi-Fi rozhraním

Wi-Fi moduly ESP8266 jsou plnohodnotné 32bitové mikrokontroléry s vlastními sadami GPIO, včetně podpory sběrnic SPI, UART a I²C. Tyto moduly se skládají z minimálního počtu součástek: samotného čipu ESP8266, flash paměti a křemíkového oscilátoru. Charakteristiky těchto modulů jsou prezentovány v tabulce charakteristiky modulů ESP8266.

V současnosti je vyráběno více než 12 modifikací ESP8266 modulových desek, které se liší počtem pinů a variantami provedení. Moduly jsou nabízeny s předinstalovaným firmwarem, který tvoří Wi-Fi - UART most pro připojení k jinému mikrokontroléru, včetně Arduino. Nastavení spojení a výměna dat jsou prováděny pomocí AT příkazů.

Existují dva způsoby práce s modulem ESP8266:

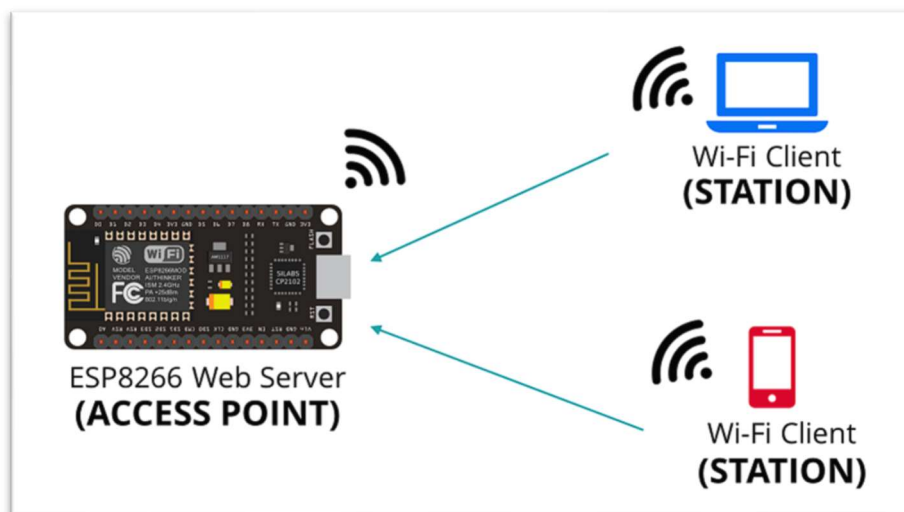
- Použití ve spojení s deskou Arduino, která bude modul ovládat přes UART.
- Vytvoření vlastního firmware pro modul ESP8266 a jeho použití jako samostatného zařízení.

Tabulka 11 **charakteristiky modulů ESP8266** [20]

Podporované typy šifrování	WEP, WPA, WPA2
Podporované provozní režimy	Klient (STA), přístupový bod (AP), klient+přístupový bod (STA+AP)
Napájecí napětí	1,7–3,6 V
Aktuální spotřeba	70 mA (špička 240 mA)
Počet dostupných pinů	4–10
Externí flash paměť	512 kB
Data RAM	80 kB
Pokyny pro RAM	32 kB
Teplotní režim	Od –40 do +70 °C

11. Stanice a přístupový bod ESP8266

Ve většině našich projektů webového serveru ESP8266 NodeMCU připojujeme ESP8266 k bezdrátovému routeru. V této konfiguraci můžeme přistupovat k ESP8266 přes LAN. V tomto



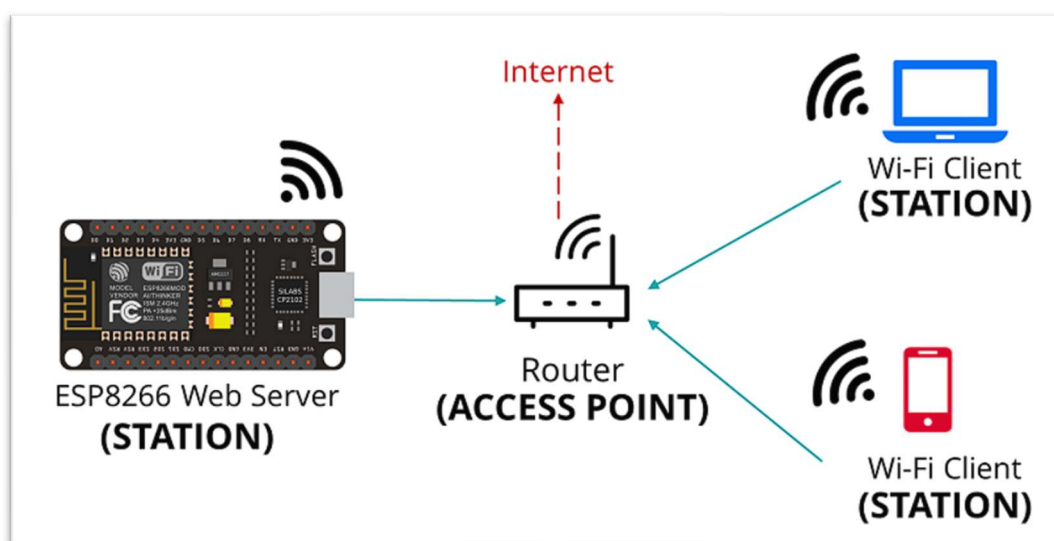
Obrázek 16: ESP8266 station and access point [21]]

scénáři router funguje jako přístupový bod a ESP8266 je nainstalován jako stanice. Chcete-li tedy ovládat ESP8266, musíte se připojit k routeru (LAN).

V některých případech to nemusí být nejlepší konfigurace (pokud v blízkosti není router). Pokud ale nakonfigurujeme ESP8266 jako hotspot (přístupový bod), můžeme se k ESP8266 připojit pomocí jakéhokoli zařízení s Wi-Fi možnostmi bez nutnosti připojení k routeru.

Jednoduše řečeno, když nainstalujeme ESP8266 jako přístupový bod, vytvoříme si vlastní Wi-Fi síť a mohou se k ní připojit blízka Wi-Fi zařízení (stanice) (například náš smartphone nebo počítač).

V této části práce nakonfigurujeme ESP8266 jako přístupový bod v našich projektech webového serveru. Tímto způsobem se k ovládání ESP8266 nepotřebujeme připojovat k routeru.



Obrázek 17: ESP8266 as an access point [21]]

Protože se ESP8266 dále nepřipojuje ke kabelové síti (jako je router), nazývá se soft-AP (soft access point). To znamená, že pokud se pokusíme stáhnout knihovny nebo použít firmware z internetu, nebude to fungovat. Nebude také fungovat, pokud se pokusíme odeslat požadavky HTTP službám na internetu, jako je například návštěva naší stránky a přiložení karty [21]).

Tato část zobrazuje kód, který je zodpovědný za připojení modulu sítě WIFI přímo k routeru. Chceme-li připojit síťový modul k routeru, nastavíme v programu název a heslo pro přístupový bod WIFI a použijeme funkce připojení [24]):

```
void connectToWiFiAndStartServer() {  
  WiFiManager wifiManager;  
  if (!wifiManager.autoConnect("AutoConnectAP")) {  
    Serial.println("Failed to connect to WiFi. Entering Access Point mode.");  
    WiFi.softAP("ESP8266-AP", "password");  
    Serial.print("AP IP address: ");    }
```

```
*wm:AutoConnect  
*wm:No wifi saved, skipping  
*wm:AutoConnect: FAILED for 2 ms  
*wm:StartAP with SSID: AutoConnectAP  
*wm:AP IP address: 192.168.4.1  
*wm:Starting Web Portal
```

```
*wm:AutoConnect  
*wm:No wifi saved, skipping  
*wm:AutoConnect: FAILED for 2 ms  
*wm:StartAP with SSID: AutoConnectAP  
*wm:AP IP address: 192.168.4.1  
*wm:Starting Web Portal  
*wm:5 networks found  
*wm:Connecting to NEW AP: Vodafone-FBAA  
*wm:connectTimeout not set, ESP waitForConnectResult...  
*wm:Connect to new AP [SUCCESS]  
*wm:Got IP Address:  
*wm:192.168.0.115  
*wm:config portal exiting  
Connected to WiFi network!  
HTTP server started.  
Time synchronized
```

WiFiManager

AutoConnectAP

Configure WiFi

Info

Exit

Update

No AP set

Obrázek 18: Výsledek funkce správce připojení Wi-Fi [autor].

Po navázání spojení musí mikrokontrolér počkat na předložení karty modulu RFID a načíst data z karty. K tomu používáme funkce `String readRFID ()`.

Čtení karet RFID Funkce čtení RFID () je zodpovědná za interakci se čtečkou RFID a získávání dat z karty.

```
String readRFID () {
    String content = "";
    if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
        for (byte i = 0; i < mfrc522.uid.size; i++) ...}
```

Tato funkce kontroluje přítomnost nové RFID karty a zda ji lze přečíst. Poté načte jedinečný identifikátor karty (UID) bajt po bajtu, převede jej na hexadecimální řetězec a připojí jej k řetězci obsahu. Pokud je některý bajt menší než 16 (0x10 v šestnáctkové soustavě), připojí se k němu "0", aby bylo formátování bajtů konzistentní.

- Home
- Export
- Export HTML to FTP

ID.No	Name	CardID	Department	Date	Time In	Time Out
1	Unknown	62ADA551	IAT	06.02.2024	12:47	12:47
2	Unknown	B8836C12	IAT	06.02.2024	12:47	12:47

Obrázek 19: Výsledkem práce je načtení karty a synchronizace data s časem při připojení k síti [autor].

Přidání registrace karty `addRegistration(String cardID)`: poté, co funkce `readRFID()` načte ID karty, přidá tato funkce nový registrační záznam s ID karty, aktuálním datem a časem. Zkontroluje, zda v systému existuje ID karty; pokud ano, aktualizuje se poslední časový limit registrace. V opačném případě se v systému vytvoří nový záznam.

```
void addRegistration(String cardID) {
    // Check if the card ID already exists in the registrations
    bool cardExists = false;
    for (auto& reg : registrations) {
        if (reg.cardID.equals(cardID)) {
            // Update the appropriate timestamps based on last scan
            cardExists = true; }
    }
```

Ve funkci programu kódu, která je zodpovědná za zpracování exportu souborů CSV při stisknutí tlačítka export, je to `handleExportCSV()`. Tato funkce generuje řetězec CSV s hlavičkami dat a iteruje vektor registrací, aby přidala data každé registrace do řetězce CSV. Nakonec odesílá data CSV jako odpověď HTTP s typem obsahu „text/csv“ (viz obrázek 20).

Funkce exportu pracuje následujícím způsobem:

```
- void handleExportCSV() {
- String csv = "ID No;Name;Card ID;Department;Date;Time In;Time Out\n"; // CSV
```

string initialization with headers

- **for (const auto& reg : registrations) { // Inicializace CSV řetězce s hlavičkami**
- **for (const auto& reg : registrace) { // Procházení každé registrace**
- **csv += reg.idNo + ";;"; // Přidání ID do řetězce CSV**
- **csv += reg.name + ";;"; // Přidání názvu do řetězce CSV**
- **csv += reg.cardID + ";;"; // Přidání ID karty do řetězce CSV**
- **csv += reg.oddělení + ";;"; // Přidání oddělení do řetězce CSV**
- **csv += reg.date + ";;"; // Přidání data do řetězce CSV**
- **csv += reg.timeIn + ";;"; // Přidání času přihlášení do řetězce CSV**
- **csv += reg.timeOut + "\\n"; // Přidání času odhlášení do řetězce CSV**
- **server.send(200, "text/csv", csv); // Odeslání řetězce CSV jako odpověď HTTP**

ID No	Name	Card ID	Departmer	Date	Time In	Time Out
1	Unknown	62ADA551	IAT	07.02.2024	12:47	12:47
2	Unknown	B8836C12	IAT	07.02.2024	12:47	12:47

Obrázek 20: Excel stažen ze serveru [autor].

„Funkce `handleExportCSV()` v kódu je určena k usnadnění exportu registračních dat do formátu souboru CSV. Po aktivaci, obvykle prostřednictvím tlačítka uživatelského rozhraní s nápisem 'Export', funkce kompiluje řetězec ve formátu CSV, který spojuje hlavičky pro identifikační číslo, jméno, identifikátor karty, oddělení, datum, čas příchodu a čas odchodu. Každý registrační záznam uložený v systému je postupně přidán do tohoto řetězce. Po dokončení kompilace funkce předává řetězec CSV klientovi jako stahovatelný soubor s MIME typem příslušně nastaveným na 'text/csv', aby označila formát souboru.

Funkce NTP (Network Time Protocol) se používá k synchronizaci času zařízení s přesným časem přes internet. ESP8266 se nejprve připojí k síti Wi-Fi pro přístup k internetu. To je nezbytné, protože pro získání správného času NTP je vyžadováno připojení k internetu.

Vytváříme instanci `NTPClient` a předáváme jí objekt `WiFiUdp`, adresu NTP serveru ("pool.ntp.org" je veřejně dostupný a široce používaný), posun časového pásma ve vteřinách.

Nastavení NTP klienta:

`configTime("CET-1CEST,M3.5.0,M10.5.0/3", "pool.ntp.org", "time.nist.gov");`

Funkce `configTime` je volána s parametry časového pásma a adresami NTP serverů ("pool.ntp.org" a "time.nist.gov"). To nastavuje časové pásmo a určuje, které servery NTP použít pro synchronizaci.

pool.ntp.org: je velký cluster časových serverů, který poskytuje spolehlivé a snadno použitelné služby NTP pro miliony klientů po celém světě. Když nastavíme naše zařízení na použití "pool.ntp.org", naše zařízení bude dotazovat pool a bude směřováno k dostupnému časovému serveru, který poskytne aktuální čas.

time.nist.gov: je jeden z referenčních časových serverů používaných zařízeními a sítěmi pro synchronizaci hodin podle přesného a přesného časového standardu. Servery NIST jsou přímo spojeny s atomovými hodinami a poskytují čas v různých formátech uživatelům a zařízením, která požadují synchronizaci.

Řetězec "**CET-1CEST,M3.5.0,M10.5.0/3**" je konfigurační parametr používaný pro nastavení časového pásma a změn letního času (DST) pro synchronizaci času NTP (protokol síťového času) v zařízeních, která podporují takové konfigurace (například ESP8266 nebo ESP32 při použití určitých knihoven NTP).

- CET: Středoevropský čas. To je standardní časové pásmo, které se používá v mnoha evropských zemích.
- -1: tato část řetězce určuje posun od UTC (koordinovaný světový čas). CET je o 1 hodinu napřed UTC (UTC+1).
- CEST: Středoevropský letní čas. To je letní čas pro stejnou oblast, který je obvykle o 1 hodinu napřed standardního času, tedy UTC+2.
- M3.5.0: Tato část určuje, kdy začíná přechod na letní čas. Čte se následovně:
 - M: Měsíc
 - 3: Březen (třetí měsíc roku)
 - 5: Pátá událost
- 0: neděle (0 - neděle, 1 - pondělí atd.), to znamená, že přechod na letní čas probíhá v poslední neděli v březnu.
- M10.5.0/3: určuje, kdy dochází k návratu na standardní čas:
 - M: Měsíc
 - 10: říjen (desátý měsíc)
 - 5: Pátá událost
 - 0: neděle
 - /3: V 3:00 ráno, to znamená, že přechod z letního času na standardní čas probíhá v poslední neděli v říjnu v 3:00 ráno.

Spojením všeho, "**CET-1CEST,M3.5.0,M10.5.0/3**" nastavuje na zařízení středoevropský čas, tj. UTC+1, přepíná na středoevropský letní čas (CEST), tj. UTC+2, v poslední neděli v březnu a přepíná zpět na středoevropský čas v poslední neděli v říjnu v 3:00 ráno.

Synchronizovaný čas lze v programu použít voláním `timeClient.getFormattedTime()` nebo `timeClient.getFormattedDate()`, které používají synchronizovaný čas NTP k vrácení aktuálního času nebo data ve snadno čitelném formátu.

Celkově synchronizace NTP zajišťuje, že na zařízení bude správný čas, což je klíčové pro časové označení událostí, koordinaci akcí mezi několika zařízeními.

Výstup popisuje informace o souborovém systému LittleFS na ESP8266, který je používán pro ukládání dat. Jedná se o přehled kapacity, využití, a dalších technických parametrů souborového systému, které jsou klíčové pro správu a optimalizaci úložiště na zařízení. Zde je vysvětlení jednotlivých položek:

Total space: Celkový dostupný prostor na souborovém systému je 2072576 bajtů, což přibližně odpovídá 2 MB. Toto číslo ukazuje, kolik paměti je celkově dostupné pro ukládání souborů.

Total space used: Celkový prostor využitý soubory je 24576 bajtů, což je přibližně 24 KB. To znamená, kolik paměti je aktuálně využito uloženými daty.

Block size: Velikost bloku souborového systému je 8192 bajtů, tedy 8 KB. Blok je základní jednotkou pro alokaci prostoru na souborovém systému.

Page size: Velikost stránky, zde uvedená stejná jako celkový prostor, což je pravděpodobně chyba v popisu. Obvykle se jedná o menší jednotku než velikost bloku a určuje minimální množství dat, které lze najednou zapsat nebo přečíst.

Max open files: Maximální počet souborů, které lze mít současně otevřené, je 5. Toto omezení je důležité pro správu paměti a výkon zařízení.

Max path length: Maximální délka cesty k souboru je 32 znaků. To omezuje, jak dlouhé mohou být názvy souborů a cesty k nim.

```
File sistem info.  
Total space:      2072576byte  
Total space used: 24576byte  
Block size:       8192byte  
Page size:        2072576byte  
Max open files:   5  
Max path lenght:  32  
  
export.html - 798
```

Obrázek 21: výstupní informace o úložišti LittleFS [autor].

Dále výstup uvádí informaci o konkrétním souboru export.html s velikostí 798 bajtů. Toto číslo ukazuje, kolik místa tento soubor zabírá na souborovém systému.

Tento přehled poskytuje užitečné informace o tom, jak je souborový systém LittleFS na ESP8266 využíván, což je důležité pro efektivní správu úložiště a optimalizaci výkonu aplikací běžících na tomto zařízení.

11.1. Zpracování HTTP požadavků v aplikaci webového serveru ESP8266

V této části je popsáno, jak systém zpracovává HTTP požadavky, s osobním zaměřením na funkce handleRoot a handleExportCSV, které obsluhují různé webové stránky a data jako odpověď na požadavky uživatelů.

Aplikace webového serveru ESP8266 je určena k odpovídání na HTTP požadavky prostřednictvím poskytování webových stránek a dat, které usnadňují interakci s uživatelem a export dat. Server je nakonfigurován k zpracování určitých tras, z nichž každá je spojena s jedinečnou funkcí zpracování požadavku a odpovědi na něj.

Funkce handleRoot je volána, když klient odešle HTTP požadavek GET na kořenovou URL adresu ("/") serveru. Tato funkce vytváří dynamickou HTML stránku, která představuje přehled stavu systému a dostupných akcí. Stránka obsahuje odkazy na další trasy serveru, nabízející takové funkce, jako je export dat a nastavení systému.

Obsah HTML je ve funkci vytvořen jako řetězec, obsahující HTML tagy pro formátování struktury stránky, včetně nadpisů, odstavců a hyperodkazů. Tento přístup umožňuje webovému serveru poskytovat data v reálném čase, odrážející aktuální stav systému přímo na webové stránce. Na požadavek server odešle tento dynamicky generovaný HTML obsah zpět klientovi, zobrazující informace v webovém prohlížeči uživatele.

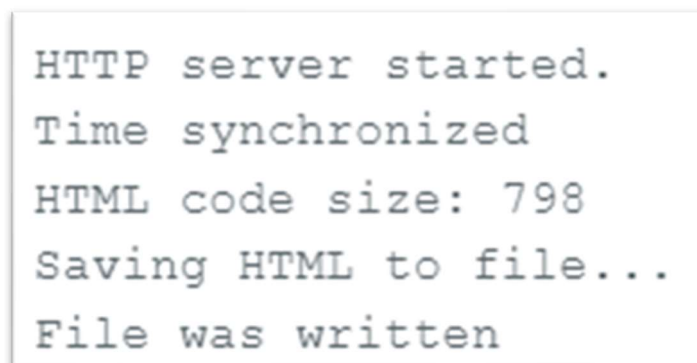
Funkce `handleExportCSV` je určena k exportu systémových dat ve formátu CSV (hodnoty oddělené čárkami). Tato funkce, aktivovaná přístupem k trase `"/export.csv"`, kompiluje záznamy dat do řetězce CSV, dodržující standardní strukturu souboru CSV. Každý záznam je oddělen čárkami a každý nový řádek představuje odlišný záznam, zahrnující taková data, jako je identifikační číslo, jméno, identifikátor karty, oddělení, datum, čas příchodu a čas odchodu.

Po přípravě řetězce CSV funkce nastaví pro hlavičku odpovědi HTTP hodnotu `"text/csv"`, aby informovala klienta, že odesílaný obsah je soubor CSV. To vede webový prohlížeč nebo klientskou aplikaci k příslušnému zpracování odpovědi, obvykle stahováním souboru nebo zobrazením jeho obsahu. Server poté odešle řetězec CSV jako tělo odpovědi, dokončující proces exportu dat.

Webový server ESP8266 je realizován pomocí knihovny `ESP8266WebServer`, která zjednodušuje vytváření HTTP koncových bodů a zpracování příchozích požadavků. Definováním tras a spojováním je s konkrétními funkcemi, aplikace efektivně spravuje různé typy požadavků: od poskytování webových stránek po export dat.

Tento přístup k zpracování HTTP požadavků nejen zvyšuje pohodlí uživatele tím, že poskytuje data a funkce v reálném čase prostřednictvím webového rozhraní, ale také využívá možnosti ESP8266 pro implementaci kompaktního a efektivního webového serveru přímo na zařízení.

Během provozu aplikace webového serveru ESP8266 jsou na sériovém monitoru Arduino zobrazeny různé stavové zprávy. Tyto zprávy poskytují zpětnou vazbu v reálném čase o stavu a činnosti systému. Tyto zprávy slouží také jako přímé rozhraní pro monitorování stavu aplikace a pokroku prostřednictvím sériového monitoru. Poskytují cenné informace o vnitřním fungování systému, pomáhají při ladění, optimalizaci výkonu a zajištění spolehlivosti aplikace webového serveru (viz. obrázek 22) .



```
HTTP server started.  
Time synchronized  
HTML code size: 798  
Saving HTML to file...  
File was written
```

Obrázek 22: výstupní informace o spuštění serveru HTTP [autor].

Na závěr, naše implementace využívá ESP8266 jako webový server pro zpracování HTTP požadavků a jako FTP klient pro nahrávání souborů na vzdálený FTP server, což zajišťuje efektivní správu a výměnu dat generovaných zařízením.

Odpověď serveru HTTP je realizována pomocí operátora `server.send(status-code, content-type, content)`. Kód stavu 200 nebo 404 indikuje, že HTTP požadavek byl úspěšný, nebo požadovaná URL adresa nebyla nalezena. Typ obsahu je obyčejný text, text HTML nebo JSON, který je označen jako `text/plain`, `text/html` a `text/json`. Když je typ obsahu `text/html`, HTML kód je prezentován ve formě řetězce nebo funkce (viz. obrázek 23) [10].

```

PS C:\Users\eordo> curl http://192.168.0.115/

StatusCode      : 200
StatusDescription : OK
Content         : <!DOCTYPE html><html><head><meta charset='UTF-8'><title>Attendance</title></head><body><style>body
                  { background-color: #e6ffe6; color: #333333; }h1 { text-align: center; }table { width: 100%; text-a
                  li...
RawContent      : HTTP/1.1 200 OK
                  Connection: close
                  Content-Length: 798
                  Content-Type: text/html

                  <!DOCTYPE html><html><head><meta charset='UTF-8'><title>Attendance</title></head><body><style>body
                  { background-colo...
Forms           : {}
Headers         : {[Connection, close], [Content-Length, 798], [Content-Type, text/html]}
Images          : {}
InputFields     : {}
Links           : @{innerHTML=Home; innerText=Home; outerHTML=<A href="/">Home</A>; outerText=Home; tagName=A; href=
                  /}, @{innerHTML=Export ; innerText=Export ; outerHTML=<A href="/export.csv">Export </A>; outerText=
                  Export ; tagName=A; href=/export.csv}, @{innerHTML=Export HTML to FTP; innerText=Export HTML to FTP
                  ; outerHTML=<A href="/export">Export HTML to FTP</A>; outerText=Export HTML to FTP; tagName=A; href
                  =/export}}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 798

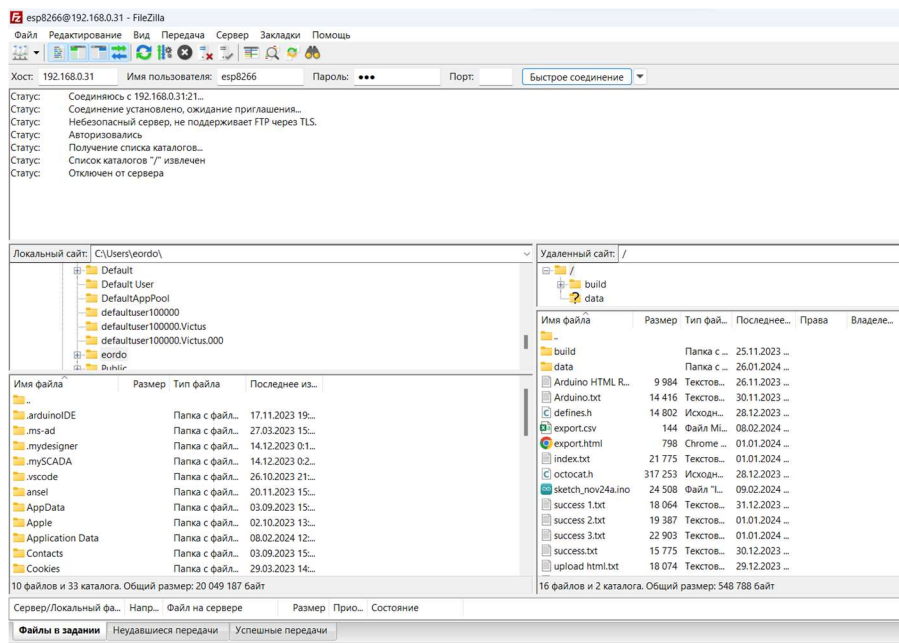
```

Obrázek 23: zobrazení informací PowerShell o úspěšném připojení Ftp client/server [autor].

12. FTP client

Kód používá FTP klienta ke zpracování přenosů souborů mezi modulem ESP8266 a FTP serverem. Funkčnost klienta FTP (File Transfer Protocol) je zapouzdřena třídou FTPClient a kód demonstruje několik operací, které lze provést pomocí tohoto klienta.

FTPClient se inicializuje pomocí informací o serveru, včetně IP adresy FTP serveru, uživatelského jména, hesla a čísla portu. Tyto informace jsou nezbytné pro navázání spojení s FTP serverem. Na obrázku 24 je vidět, jak FTP klient funguje.



Obrázek 24: FileZilla Client [autor]

Spuštění FTP klienta: `ftpClient.begin(ftpServerInfo);` volání spustí FTP klienta s poskytnutými informacemi o serveru. Některé funkce v kódu předpokládají schopnost přenášet soubory na a ze serveru FTP. Zejména funkce `ExportHTMLtoFTP()` a `ExportFileToFTP()` znamenají, že můžeme zahájit přenos souborů ze systému souborů ESP8266 na server FTP. Tyto

funkce používají režim `FTP_PUT_NONBLOCKING`, který předpokládá, že přenos bude zahájen a bude pokračovat na pozadí, což umožní spuštění jiného kódu, aniž by byl zablokován operací přenosu souboru.

Kontrola stavu FTP přenosu se provádí funkcí `ftpClient.check()`, která kontroluje stav aktuálního FTP přenosu. Vrací stavový objekt, který označuje, zda přenos probíhá, zda byl úspěšně dokončen nebo zda došlo k chybě.

Zpracování odpovědi FTP probíhá po spuštění přenosového kódu FTP, kde je zkontrolováno dokončení nebo chyba. Vytiskne kód odpovědi a popis, což může být užitečné pro ladění a potvrzení úspěšného přenosu. Časový limit přenosu FTP funguje takto, když kód implementuje jednoduchý mechanismus časového limitu pro přenosy FTP. Pokud se přenos nedokončí během zadaného časového období (30 sekund ve kódu), vytiskne se zpráva o vypršení časového limitu.

Použití neblokovaného FTP přenosu znamená, že ESP8266 může pokračovat v provádění dalších úkolů, jako je zpracování požadavků webového serveru nebo čtení RFID tagů, zatímco přenos souborů pokračuje.

Funkce přenosu FTP je ve smyčce `ftpClient.handleFTP()`, takže volání uvnitř funkce `loop()` je nezbytné, protože umožňuje pokračovat v neblokujícím přenosu FTP. Tato funkce musí být volána pravidelně, aby se zpracoval přenos a všechny příchozí nebo odchozí datové pakety spojené s relací FTP.

Proto je FTP klient v kódu nakonfigurován tak, aby usnadnil nahrávání souborů z ESP8266 na FTP server neblokujícím způsobem, což umožňuje asynchronní přenosy souborů, zatímco zařízení pokračuje v provádění jiných úkolů. Kód také obsahuje mechanismy pro kontrolu stavu přenosu a zpracování různých transakcí souvisejících s FTP. Na obrázku 25 je vidět princip fungování FTP serveru v programu FileZilla Serveru [1].

13. FTP server

V projektu je implementována funkce mobilního sběrače dat, který funguje jako server. To znamená, že sběrač nejen shromažďuje data z různých zdrojů, ale také umožňuje přístup k těmto datům na vyžádání z klientských aplikací nebo zařízení. To zajišťuje flexibilitu a rozšiřitelnost systému sběru dat a činí informace v reálném čase dostupné pro analýzu a rozhodování.

Pro implementaci serverové funkcionality v mobilním kolektoru se používají moderní síťové protokoly komunikace, které zajišťují stabilitu a bezpečnost při přenosu dat. Server je schopen zpracovávat příchozí požadavky na stažení dat, autentizovat uživatele a poskytovat omezení přístupu k informacím v závislosti na úrovni práv.

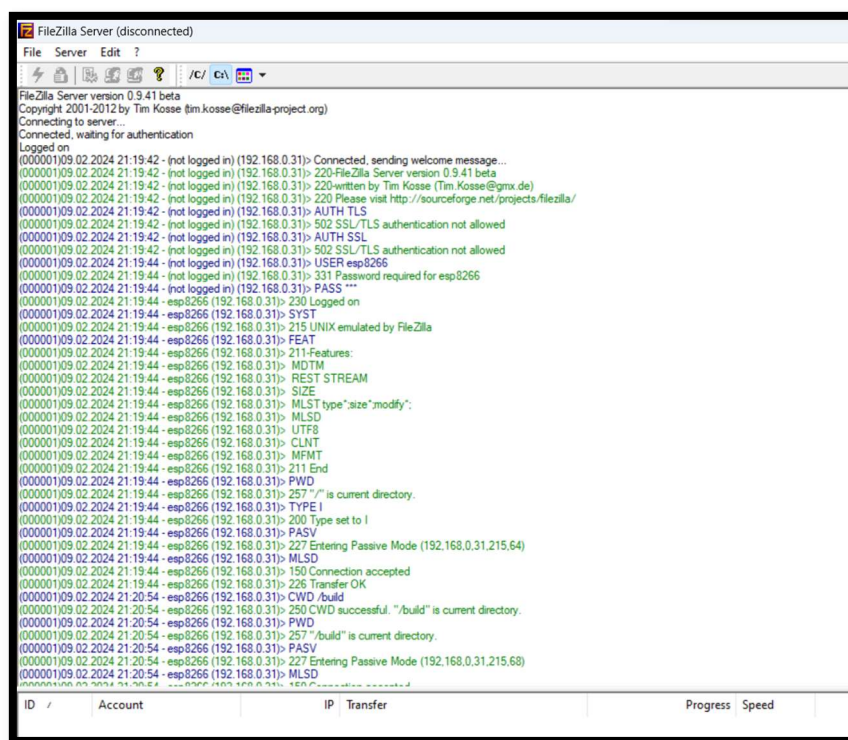
Integrace serverových funkcí do mobilního sběrače dat umožňuje

Vzdálený přístup: uživatelé mohou požadovat a získávat data od mobilního sběrače z jakéhokoli místa, kde je přístup k internetu, použitím standardních protokolů a API.

Škálovatelnost: systém lze snadno škálovat pro zpracování většího množství požadavků nebo objemů dat, což je důležité při zvyšování počtu připojených zařízení nebo bodů sběru dat.

Zajištění bezpečnosti: všechna data přenášená na mobilní jeřáb a zpět mohou být chráněna pomocí šifrování a bezpečných protokolů přenosu dat, jako je HTTPS, což eliminuje riziko neoprávněného přístupu.

Výkon a spolehlivost: mobilní sběrač může fungovat nepřetržitě, poskytovat data na vyžádání a zajišťovat stabilitu systému monitorování a analýzy (viz. obrázek 25).



Obrázek 25 FileZilla Server [autor]

V práci je podrobně zkoumána architektura mobilního kolektoru, protokoly výměny dat a metody zajištění bezpečnosti a důvěrnosti informací. Jsou také prezentovány scénáře použití mobilního kolektoru jako serveru v různých oblastech aplikace, demonstrující jeho univerzálnost a efektivitu.

13.1. Výhody a nevýhody FTP serveru

13.1.1. Nevýhody FTP serveru

- FTP nezajišťuje bezpečnost. Všechny soubory jsou přenášeny jako otevřený text a mohou být úplně přečteny útočníkem připojeným k kanálu.
- Servery pro přenos souborů neověřují pravost zdroje odesílání. To činí server zranitelným vůči hackerským útokům, jako je DDoS.
- Je obtížné sledovat činnosti uživatelů, nahrávané a stahované soubory ze serveru.

13.1.2. Výhody FTP serveru

- FTP umožňuje rychle přenášet velké objemy dat. Díky absenci šifrování a záznamu systémových dat probíhá přenos rychleji.
- Pomocí FTP lze přenášet několik různých souborů nebo adresářů současně.
- Přenos lze obnovit. Pokud byl přenos souboru přerušen, nemusíte se obávat ztráty pokroku. FTP podporuje plynulé obnovení pomocí příkazu rest.

Hlavní část kódu v naší práci, odpovědná za export souboru CSV přes FTP server v poskytnutém úryvku kódu, zahrnuje několik kroků a komponent, které spolupracují na přenosu souboru.

Generování dat CSV

Funkce `generateCSVData()` je zodpovědná za generování dat CSV z uložených registrací. Prochází vektor registrací, kombinuje detaily každé registrace do jednoho řádku formátovaného jako CSV.

```
String generateCSVData() {  
    String csv = "ID No;Name;Card ID;Department;Date;Time In;Time Out\n"; // Header  
    for (const auto& reg : registrations) {  
        csv += reg.idNo + ";" + reg.name + ";" + reg.cardID + ";" + reg.department + ";" +  
reg.date + ";" + reg.timeIn + ";" + reg.timeOut + "\n";  
    } return csv; ..... }  

```

Uložení CSV do souboru

Funkce `saveCSVToFile()` přijímá cestu, kam se má soubor CSV uložit, a data CSV jako argumenty. Snaží se otevřít (nebo vytvořit) soubor na určené cestě ve filesystému LittleFS ESP8266 a zapisuje do něj data CSV (viz. obrázek 26).

```
bool saveCSVToFile(const String& path, const String& data) {  
    File file = LittleFS.open(path, "w");  
    if (!file) {  
        Serial.println("Failed to open file for writing");  
        return false; ..... }  
}
```



ID No	Name	Card ID	Department	Date	Time In	Time Out
1	Unknown	62ADA551	IAT	12.02.2024	00:25	00:25
2	Unknown	B8836C12	IAT	12.02.2024	00:27	00:27

Obrázek 26: Excel stažený z telefonu

Export souboru na FTP

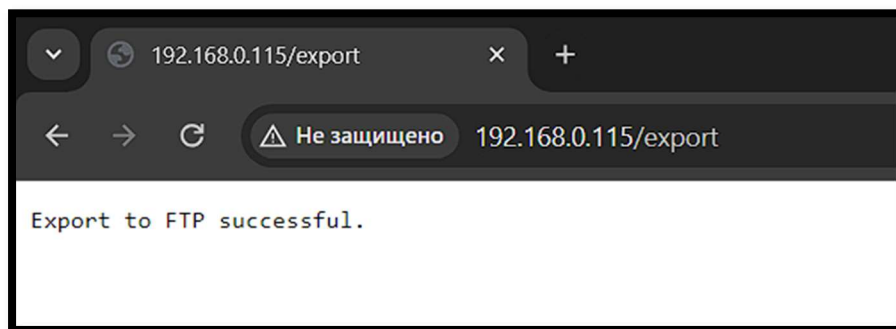
Funkce `exportFileToFTP()` je určena pro zpracování FTP přenosu lokálně uloženého souboru CSV na FTP server. Používá objekt `ftpClient` (instanci `FTPClient`, inicializovanou informacemi o serveru) k zahájení přenosu v neblokujícím režimu. Funkce kontroluje stav přenosu, dokud není dokončen nebo nepřerušen kvůli chybě nebo timeoutu.

```
bool exportFileToFTP(const String& localPath, const String& remotePath) {  
    Serial.println("Starting FTP transfer...");  
    ftpClient.transfer(localPath, remotePath, FTPClient::FTP_PUT_NONBLOCKING);  
}
```

```

uint32_t startTime = millis();
while (true) {
    ftpClient.handleFTP(); // Regularly call handleFTP to process the transfer
    FTPClient::Status status = ftpClient.check(); // Check the transfer status
    if (status.result == FTPClient::OK || status.result == FTPClient::ERROR) {
        if (status.result == FTPClient::OK) {
            Serial.println("FTP Transfer complete.");
            return true;
        } else {
            Serial.println("FTP Transfer failed: " + status.desc);
            return false;    }    }
    if (millis() - startTime > 30000) { // Timeout check
        Serial.println("FTP transfer timeout.");
        return false; ..... }
}

```



Obrázek 27: Funkce exportu z webové stránky mikrokontroleru

Zpracování požadavku na export

Funkce `handleExport()` je obslužná rutina cesty webového serveru, která zahajuje proces exportu při přístupu. Nejprve generuje data CSV, ukládá je do souboru a poté se pokouší exportovat tento soubor na FTP server. Odesílá HTTP odpověď na základě výsledku operace exportu (viz. obrázek 27).

```

void handleExport() {
    String csvData = generateCSVData();
    saveCSVToFile("/export.csv", csvData);
    bool exported = exportFileToFTP("/export.csv", "/remote/path/export.csv");
    if (exported) {
        server.send(200, "text/plain", "Export to FTP successful.");
    } else {
        server.send(500, "text/plain", "Export to FTP failed."); ..... }
}

```


Tato posloupnost operací zahrnuje základní funkce potřebné k vytvoření souboru CSV na základě registračních dat, jeho lokálnímu uložení a následnému exportu na vzdálený FTP server. Výsledkem celé práce programu je zobrazen na obrázku 28 výstupu sériového monitoru Arduina, kde probíhá sekvence inicializačních kroků a zpráv o stavu programu nahráného do zařízení. Po úspěšné inicializaci a synchronizaci času, stejně jako připojení k WiFi, což ukazuje, že zařízení je připraveno na plnohodnotnou práci, včetně obsluhy webového obsahu a interakce se soubory přes FTP.

```
Inizializing FS...done.
File sistem info.
Total space:      2072576byte
Total space used: 40960byte
Block size:       8192byte
Page size:        2072576byte
Max open files:   5
Max path lenght:  32

export.csv - 52
export.html - 798
*wm: AutoConnect: ESP Already Connected
*wm: AutoConnect: SUCCESS
*wm: STA IP Address: 192.168.0.115
Connected to WiFi network!
HTTP server started.
Time synchronized.
HTML code size: 798
Saving HTML to file...
File was written.
Starting FTP transfer...
FTP Transfer complete.
Export to FTP successful.
```

Obrázek 28: Výsledek úspěšného nahrání a práce programu

14. Mobile remote server collector

V rámci projektu je mobilní datový kolektor navržen tak, aby vykonával funkce serveru. Toto klíčové architektonické řešení umožňuje zařízení nejen shromažďovat informace ve vnitřní paměti, ale také tato data na vyžádání poskytovat dalším zařízením nebo službám. Mobilní kolektor tedy funguje jako uzel schopný pravidelně aktualizovat a poskytovat aktuální informace v reakci na externí požadavky.

Tato funkce je řízena potřebou rychlého přístupu k datům pro různé analytické, monitorovací a manažerské aplikace, což je důležitý aspekt v systémech, které vyžadují odezvu v reálném čase. Provoz mobilního kolektoru jako serveru poskytuje následující možnosti:

Mobilní faucet dokáže přijímat požadavky od klientů a reagovat na ně poskytnutím požadovaných dat. To lze implementovat prostřednictvím REST API, kde klienti odesílají HTTP požadavky na načtení dat.

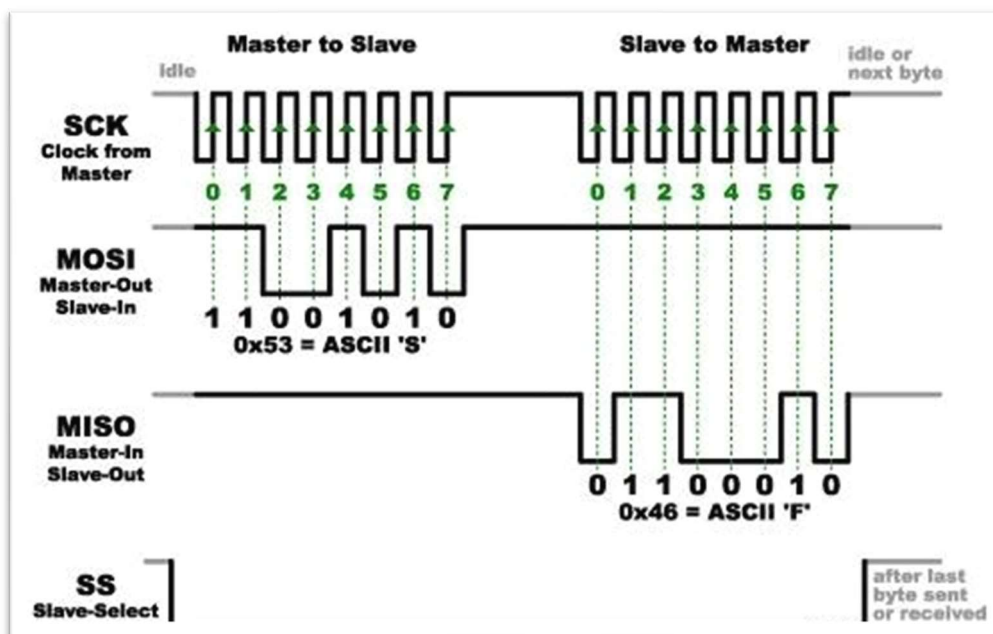
Návrh zahrnuje autentizační a přístupové mechanismy, které zajistí, že k datům budou mít

přístup pouze oprávnění uživatelé. Integrace funkcí na straně serveru do mobilního kolektoru výrazně zlepšuje dostupnost a užitečnost shromážděných dat a zpřístupňuje je širokému spektru spotřebitelů, od vestavěných systémů po cloudové aplikace.

Tento článek podrobně popisuje technické aspekty implementace funkcí serveru v mobilním kolektoru, včetně výběru vhodných softwarových a hardwarových komponent, algoritmů zpracování dat a principů pro konstrukci síťové interakce.

15. Schéma činnosti zařízení

Pro úplnější pochopení činnosti zařízení zvažte časový diagram čtečky RFID. Přenos dat z modulu RFID do mikrokontroléru probíhá pomocí rozhraní SPI. SPI je sériové periferní rozhraní, které umožňuje přenášet 1 bajt informací najednou mezi dvěma moduly. Protokol SPI je známý jako protokol master/slave. To znamená, že existuje pouze jedno ovládací zařízení a několik ovládaných zařízení. Toto rozhraní je skvělé pro rychlý přenos dat ze senzorů. Aby toto rozhraní fungovalo, jsou potřeba minimálně 4 piny: SCK, MOSI, MISO a SS (označené jako SDA na modulu RFID).



Obrázek 29: SPI komunikační časový diagram [22]

SCK je hodinový signál, který je potřebný k synchronizaci mezi dvěma zařízeními, například jak rychle mají být data odesílána a přijímána. Rychlost přenosu dat je známá jako baud. Bez SCK by zařízení od sebe nemohla přijímat data.

MOSI je zodpovědná za přenos dat z masteru na slave, **MISO** je zodpovědná za přenos informací z slave na master. Na časovém diagramu

Můžete vidět, jak hlavní zařízení posílá znak „S“ do podřízeného zařízení a přijímá jako odpověď znak „F“.

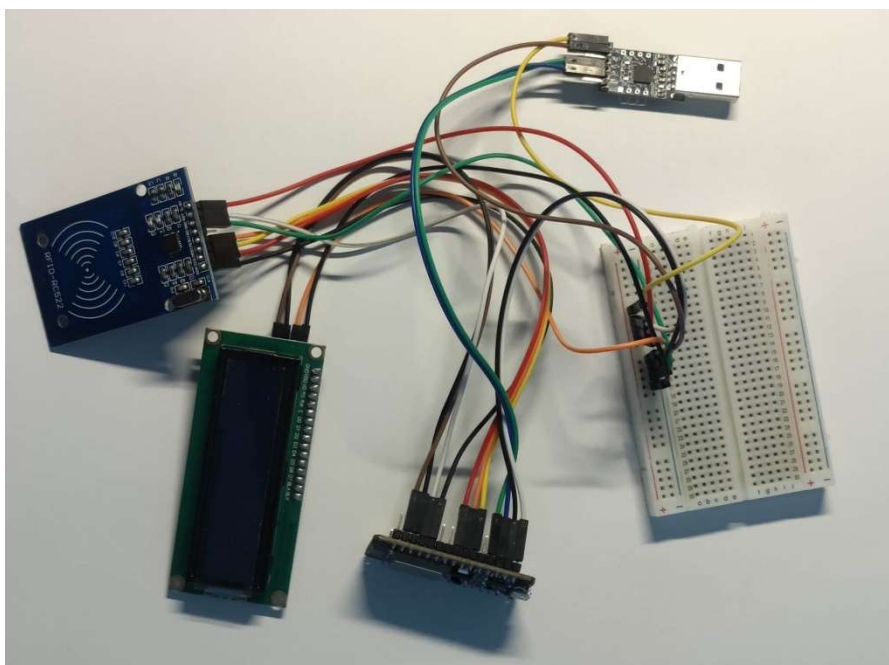
SS se používá k upozornění konkrétního slave, že se master chystá odeslat data. Poté, co master zahájí komunikaci mezi zařízeními, bude udržovat signál SS na nízké úrovni, dokud se přenos dat nezastaví. Když je signál na SS nízký, master začne přepínat hodinový pin SCK a řídit MOSI, aby poskytoval informace podřízenému. Čísla v diagramu představují každý bit přenášeného bytu [22].

16. Závěr k oddílu

Tato část zkoumala časový diagram čtečky RFID. Vzhledem k tomu, že existuje více typů protokolů pro přenos informací z modulu do mikrokontroléru, byl uvažován protokol SPI, na kterém RFID modul pracuje. Byly zvažovány výhody tohoto protokolu, jak probíhá synchronizace mezi moduly, na jakém principu a v jakém pořadí jsou informace přenášeny a přijímány. Rovněž v případě, že je nutné k mikrokontroléru připojit více RFID čteček, je možných několik typů připojení, což umožňuje rozšířit možnou funkčnost zařízení na požadovanou.

17. Výroba celého zařízení

V této fázi bylo sestaveno zařízení ze vstupních komponent pro mobilní sběr RFID dat na základě předem vybraných modulů a napsaného softwaru a byla také prověřena funkčnost zařízení a softwarové části mikrokontroléru.



Obrázek 30: Kompletní zařízení sestaveno [autor].

V této části bylo na základě vývojového schématu a vybraných komponent, které byly navrženy ve třetím oddílu, sestaveno zařízení pro čtení bezkontaktních karet a značek využívající technologii RFID (Radio-Frequency Identification), které je kompatibilní s kartami a značkami pracujícími na frekvenci 13,56 MHz. Do mikrokontroléru a PC byly nahrány programy se všemi knihovnami. Po nahrání programů byla nastavena IP adresa pro otevření html stránky se všemi definovanými funkcemi práce s exportem a FTP serverem. Poté, co se mikrokontrolér úspěšně připojil k Wi-Fi síti, byla ověřena funkčnost zařízení. Práce zařízení plně odpovídá stanoveným požadavkům. Výsledkem úspěšné kontroly je zobrazení UID (Unique Identifier) karty, který zobrazuje unikátní sériové číslo zabudované v každé RFID kartě s časem příchodu a odchodu [23]).

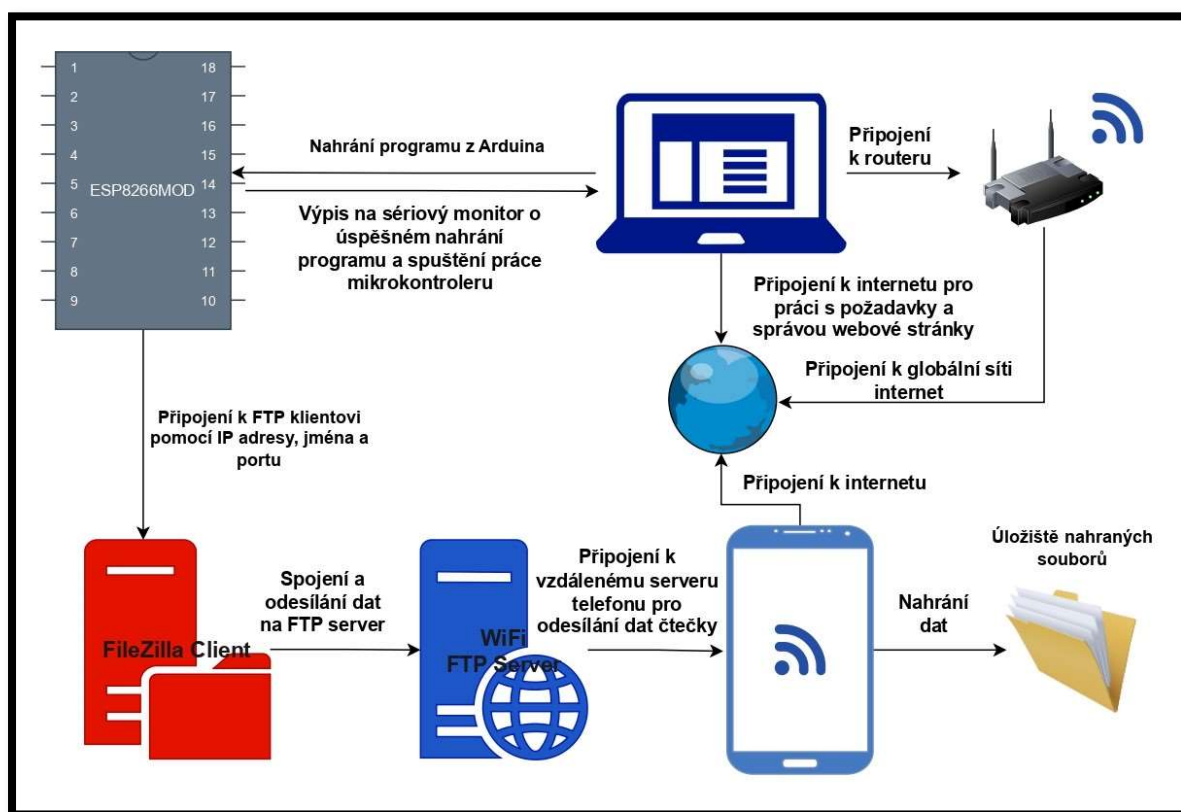
Po úspěšném nahrání programu se v sériovém monitoru objeví zpráva o stavu WiFi: Jakmile dojde k připojení k WiFi a vytvoření WiFi přístupového bodu, objeví se příslušné

zprávy, včetně IP adresy pro otevření html stránky. Pokud ESP8266 přejde do režimu přístupového bodu, IP adresa pro otevření html stránky bude nastavena lokálně prostřednictvím esp8266. Po úspěšném připojení k síti a synchronizaci času se objeví zprávy o tom, že čas byl synchronizován s systémem [24]).

Při skenování RFID karty se zobrazí zprávy o načtených ID karet, stejně jako informace o registraci (čas příchodu/odchodu) a na LCD se zobrazí unikátní identifikátor karty.

17.1. Celý cyklus posloupnosti operací přenosu dat přes ESP8266

Tento diagram ilustruje krok za krokem pracovní proces nahrávání dat z mikrokontroléru ESP8266 do vzdáleného úložiště přes FTP. Proces začíná programováním mikrokontroléru přes rozhraní Arduino, po kterém následuje ověřovací výstup na sériový monitor. Po naprogramování se ESP8266 připojí k Wi-Fi síti, získává přístup k internetu pro správu webových stránek a zpracování požadavků. Tok dat pokračuje, zatímco ESP8266 interaguje s FTP klientem, používá určenou IP adresu, uživatelské jméno a číslo portu. FTP klient, jako například FileZilla, iniciuje spojení s FTP serverem přes Wi-Fi, což vede k úspěšnému přenosu dat. Tato data jsou poté dostupná na vzdáleném serveru, odkud je možné je stáhnout do mobilního zařízení nebo uložit do úložiště digitálních souborů.



Obrázek 31: Schéma fungování mikrokontroléru s využitím FTP pro přenos dat

18. Závěr

V rámci bakalářské práce bylo vyvinuto zařízení založené na platformě Arduino pro identifikaci návštěvníků vzdělávacího zařízení.

V první části práce byla zdůvodněna potřeba vyvinout takové zařízení. Za tímto účelem byl proveden průzkum existujících systémů kontroly a řízení přístupu a byly identifikovány funkce, které jsou v současnosti realizovány v systémech kontroly přístupu. Ve druhé části byly na základě analýzy známých řešení stanoveny požadavky na vyvíjené zařízení a učiněn závěr o nutnosti vytvoření zařízení s co nejnižšími ekonomickými náklady. Ve třetí části byla navržena primární strukturální schéma zařízení, poté byly vybrány potřebné komponenty, analyzovány jejich technické specifikace a způsoby připojení. Výsledkem bylo získání elektrického schématu zařízení. Navržená strukturální schéma a schéma komponent umožnily další vývoj algoritmů fungování a realizaci softwarového prototypu. Ve čtvrté části byl vyvinut algoritmus fungování zařízení, byly vybrány prostředky pro psaní softwaru, po kterém byly napsány programy pro mikrokontrolér a počítač.

Výhody vyvinutého zařízení zahrnují ukládání informací ve vzdělávacím zařízení a práci s vzdáleným serverem programu, což zabraňuje možnosti získání přístupu k datům jinými osobami, možnost přidávání nových funkcí a nových komponent díky flexibilní struktuře a možnosti aktualizace kódu mikrokontroléru a PC.

19. Seznam použité literatury

- 1) GOST R 54831-2011 „Systémy kontroly a řízení přístupu. Řízená blokovací zařízení. Všeobecné technické požadavky. Zkušební metody“(schváleno a uvedeno v platnost nařízením Rosstandart ze dne 13. prosince 2011 N 1223-st).
- 2) Netypické funkce ACS [Elektronický zdroj]. – Režim přístupu: http://www.secuteck.ru/articles2/sys_ogr_dost/netipichnye-funktsii-skud/ (datum přístupu: 20.05.2016).
- 3) *Home*. (n.d.). Bosch Security and Safety Systems I Global. <https://www.boschsecurity.com/xc/en/>
- 4) Building technology. (n.d.). siemens.com Global Website. <https://www.siemens.com/global/en/products/buildings.html>
- 5) Security. (n.d.). Building Automation. <https://buildings.honeywell.com/us/en/brands/our-brands/security>
- 6) Open-source tool that uses simple textual descriptions to draw beautiful UML diagrams. (n.d.). PlantUML.com. <https://plantuml.com/en/>
- 7) Výrobce RFID štítků a RFID zařízení společnost RST-Invent [Elektronický zdroj]: webové stránky společnosti. URL: <http://www.rst-invent.ru/faq/>
- 8) Čtečky a štítky RFID – Režim přístupu [Elektronický zdroj]: vzdělávací blog. URL: <https://arduino-kit.ru/product/schi-tyivatel-rfid-rc522-1356mhz-2B-karta-2B-brelak>
- 9) Staff, L. E. (2018, July 30). In-Depth: What is RFID? How It Works? Interface RC522 with Arduino. Last Minute Engineers. <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>
- 10) Cameron, N. (2018). Arduino Applied: Comprehensive Projects for Everyday Electronics. Apress.
- 11) NodeMCU ESP8266. (n.d.). Components101. <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>
- 12) GM electronic | elektronické součástky, komponenty. (n.d.). GM electronic | elektronické součástky, komponenty. <https://www.gme.cz/v/1508310/prevodnik-usb-uart-reset-pin>
- 13) Half-Size Breadboard. (n.d.). Digilent. <https://digilent.com/shop/half-size-breadboard/>
- 14) Buy LCD display 2x16 characters blue + I2C. (n.d.). BOTLAND.

<https://botland.store/alphanumeric-and-graphic-displays/2351-lcd-display-2x16-characters-blue-i2c-lcm1602-5904422309244.html>

15) Samsung 22E 18650 Battery, 2150mAh, 4.4A, 3.7V, Grade A Lithium-ion (ICR18650-22E). (n.d.). Voltaplex Lithium Ion Battery Packs. <https://voltaplex.com/samsung-22e-18650-battery-icr18650-22e>

16) Fritzing is free cross-platform software, simplified CAD with a WYSIWYG interface for hobby projects in the Arduino ecosystem. (n.d.). Welcome to Fritzing. <https://fritzing.org/>

17) Belov, A.V. (2020). Control of the ARDUINO module via Wi-Fi from mobile devices. St. Petersburg: Science and Technology, 496 pages, ill.

18) Arduino - Home. (n.d.). Arduino - Home. <https://www.arduino.cc/>

19) Batrinu, C. (2017). ESP8266 Home Automation Projects: Leverage the power of this tiny WiFi chip to build exciting smart home projects. Packt Publishing.

20) Petin V.A. Projects using the Arduino controller. / V.A. Petin. – St. Petersburg. : BHV-Petersburg, 2015. – 464 p.

21) ESP8266 NodeMCU Access Point (AP) for Web Server | Random Nerd Tutorials. (n.d.). Random Nerd Tutorials. <https://randomnerdtutorials.com/esp8266-nodemcu-access-point-ap-web-server/>

22) Communication Protocols. // Documentation. URL: https://tessel.gitbooks.io/t2-docs/content/Tutorials/Communication_Protocols.html#spi

23) Bhuptani, Manish, and Shahram Moradpour. RFID field guide: deploying radio frequency identification systems. Prentice Hall PTR, 2005.

24) Kurniawan, Agus. Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32. Packt Publishing Ltd, 2019.

20. Seznam obrázků

Obrázek 1	Identifikační metody [1]	4
Obrázek 2:	Kompletní blokové schéma PlantUML [autor]	5
Obrázek 3:	RFID čtečka-RC522 [autor]	6
Obrázek 4:	RFID štítky v pouzdrech [autor]	7
Obrázek 5:	Pinout čtečky RC522 [9]	7
Obrázek 6:	Hardwarová platforma NodeMCU [autor]	8
Obrázek 7:	Pinout hardwarové desky NodeMCU [11]	9
Obrázek 8:	Převodník CP2102 (univerzální adaptéry pro připojení ESP modulů k počítači) [11]	12
Obrázek 9:	Arduino Breadboard [autor]	12
Obrázek 10:	Connecting cables [autor]	13
Obrázek 11:	I2C displej LCD 1602 [14]	13
Obrázek 12:	Baterie Samsung [15]	14
Obrázek 13:	Fritzing free cross-platform software for projects Arduino [16]	15
Obrázek 14:	Device connection diagram developed in the program Fritzing [autor]	16
Obrázek 15:	Arduino IDE interface [autor]	19
Obrázek 16:	ESP8266 station and access point [21]	22
Obrázek 17:	ESP8266 as an access point [21]	22
Obrázek 18:	Výsledek funkce správce připojení Wi-Fi [autor].	23
Obrázek 19:	Výsledkem práce je načtení karty a synchronizace data s časem při připojení k síti [autor].	24
Obrázek 20:	Excel stažen ze serveru [autor].	25
Obrázek 21:	výstupní informace o úložišti LittleFS [autor]	27
Obrázek 22:	výstupní informace o spuštění serveru HTTP [autor]	28
Obrázek 23:	zobrazení informací PowerShell o úspěšném připojení Ftp client/server [autor]	29
Obrázek 24:	FileZilla Client [autor]	29
Obrázek 25:	FileZilla Server [autor]	31
Obrázek 26:	Excel stažený z telefonu	32
Obrázek 27:	Funkce exportu z webové stránky mikrokontroleru	33
Obrázek 28:	Výsledek úspěšného nahrání a práce programu	34
Obrázek 29:	SPI komunikační časový diagram [22]	35
Obrázek 30:	Kompletní zařízení sestaveno [autor].	36
Obrázek 31:	Schéma fungování mikrokontroléru s využitím FTP pro přenos dat	37

21. Seznam tabulek:

Tabulka 1 kontakty GPIO:.....	10
Tabulka 2 Speciální piny:	10
Tabulka 3 Napájecí vývody:.....	10
Tabulka 4 Řídicí piny:	11
Tabulka 5 Komunikační rozhraní:	11
Tabulka 6 Tabulka rozhraní mikrokontroléru:	11
Tabulka 7 Připojení NodeMCU ESP8266 k RFID MFRC522	16
Tabulka 8 Připojení NodeMCU ESP8266 k LCD 1602.....	16
Tabulka 9 Připojení USB TTL CP2102 k ESP8266:	17
Tabulka 10 Knihovny určené pro práci s Arduino	20
Tabulka 11 charakteristiky modulů ESP8266 [20]	21