|  |  |  |
| --- | --- | --- |
| Logo__SSPU_2016_Barva | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Čtečka otisků prstů**  Jan Švrček | | |
|  | | |
| VÃ½sledek obrÃ¡zku pro fingerprint reader arduino  Obr. 1 | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2018/2019 | |

#### Poděkování:

Chtěl bych poděkovat učitelům, kteří mi s projektem pomohli. Panu učiteli Mgr. Petrovi Grussmannovi za konzultaci a pomoc při řešení mnohých problémů. Dále panu učiteli Mgr. Marcelovi Godovskému za zapůjčení hardwarových součástí a rovněž za pomoc při řešení problémů.

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2016

*podpis autora práce*

**ANOTACE**

Čtečka otisků prstů je elektronické zařízení, které používá digitální obraz vzoru otisku prstu neboli „live scan“. Ten je digitálně zpracován k vytvoření biometrické šablony, která se používá pro srovnávání. Používá se pro zabezpečení objektů (domácnosti, přístupu do telefonu... ) nebo jako periferie k počítači.

Čtečka je ovládaná mikroprocesorem ESP8266. K jeho naprogramováni jsem použil Platformio a samotný kód je napsán v jazyce C++. „Dále jsem přidal LCD display pro obrazový výstup, spolu s relátkem pro simulaci zámku.“

**KLÍČOVÁ SLOVA**

Čtečka otisků prstů, zapezpečení, ESP, LCD

**ANOTATION**

Fingerprint reader is an eletronic device that is used to take a digital image of a fingerprint pattern, named live scan. It is digitally processed to create a biometric template that is used for comparison. Used to secure objects (home, access to your phone ...) or as a peripheral to a computer.

The reader is controlled by the ESP8266 microprocessor. I used Platformio to program it and the code itself is written in C ++. „Next I added an LCD display for image output, along with a relay for simulation of the lock“

**KEYWORDS**

Fingerprint reader, security, ESP, LCD

OBSAH

[Úvod 5](#_Toc370246085)

[1 Teoretická a metodická východiSKa 6](#_Toc370246086)

[2 Využité technologie 7](#_Toc370246087)

[3 Způsoby řešení a použité postupy 8](#_Toc370246088)

[4 Výsledky řešení, výstupy, uživatelský manuál 9](#_Toc370246089)

[Závěr 10](#_Toc370246090)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 11](#_Toc370246091)

[Seznam příloh 12](#_Toc370246092)

Úvod

Čtečka otisků prstů je elektronické zařízení, které používá digitální obraz vzoru otisku prstu neboli „live scan“. Ten je digitálně zpracován k vytvoření biometrické šablony, která se používá pro srovnávání. Používá se pro zabezpečení objektů (domácnosti, přístupu do telefonu... ) nebo jako periferie k počítači.

Cílem závěrečné práce bylo vytvořit čtečku otisků prstů, která by byla schopna odesílat snímaný otisk přes Wi-fi, který by byl následně analyzován, porovnán a zaznamenán do databáze otisků, spolu s doplňujícími údaji.

V dokumentaci se Vám pokusím přiblížit detaily tvoření projektu, jako jsou využité technologie a seznámit Vás se vzniklými problémy a jejich řešením. A jelikož jsou knihovny i samotné zařízení vytvořeny pro spolupráci s Arduinem, tak jsem se bohužel bez problému neobešel.

# VyuŽité technologie

## Visual Studio Code a Platformio

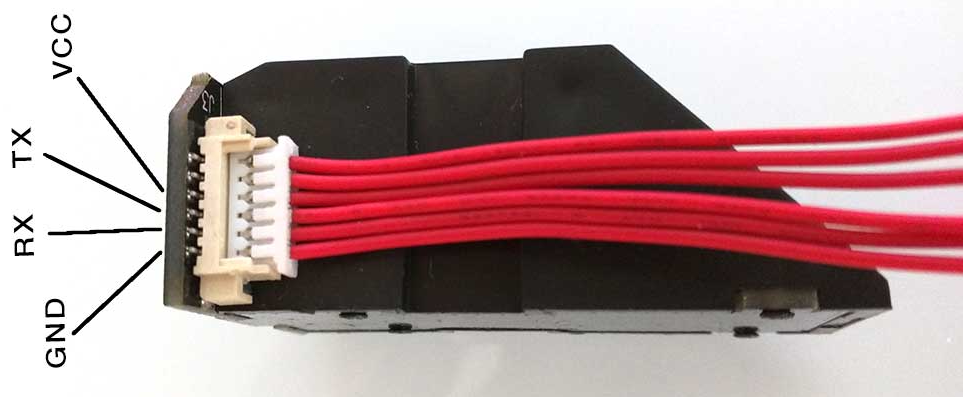
Visual Studio Code je editor zdrojového kódu, do kterého lze přidávat různá rozšíření. Použil jsem rozšíření Platformio, které je určeno k programováni mikroprocesorů. Programovacím jazykem je C/C++. Rozhodl jsem se použít právě Platformio z důvodu ulehčení práce s různými mikroprocesory.

## Snímač otisků prstů s pamětí DY50

Hlavní hardwarový komponent projektu je snímáč otisků prstů, který obsahuje flash pamět, do které lze uložit až 127 otisků. Je vytvořen pro spolupráci především s Arduinem, založeným na miktrokontroleru Atmega. Čtečka má 6 výstupů, z nichž potřebujeme jen 4.

**Parametry DY50:**

* *Režim shody: 1:1*
* *Porovnání vzorku: <1s*
* *Napájení: 3.6 až 6.0 V*
* *Pracovní proud: 120mA až 140mA*
* *Plocha okna: 14 x 18 mm*

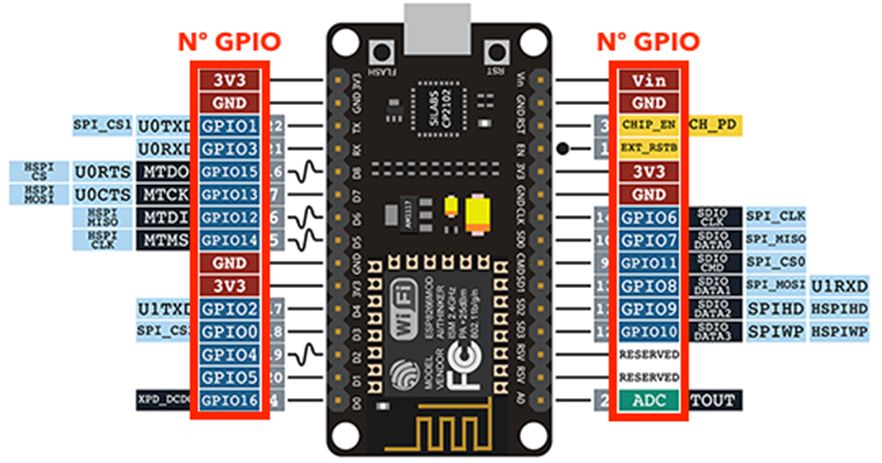


Obr. 2 DY50 Senzor

## Mikroprocesory

### ESP 8266

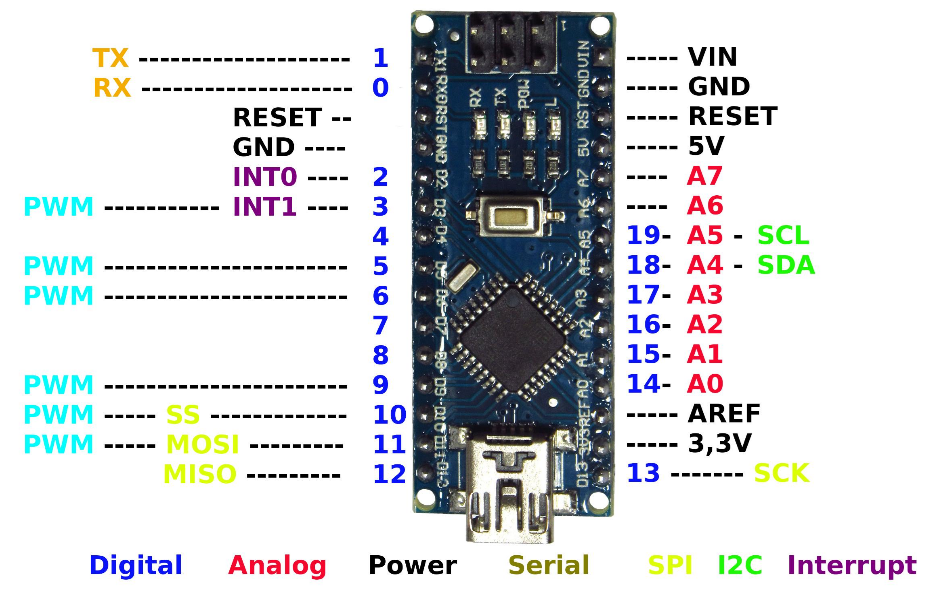
Hardwarovým základem je 32bit mikroprocesor ESP 8266, který jsem zvolil kvůli vestavěnému Wi-fi modulu. Avšak nevýhodou je, že pracuje pod napětím pouze 3.3V.



Obr. 3 Datasheet ESP 8266

### Arduino Nano

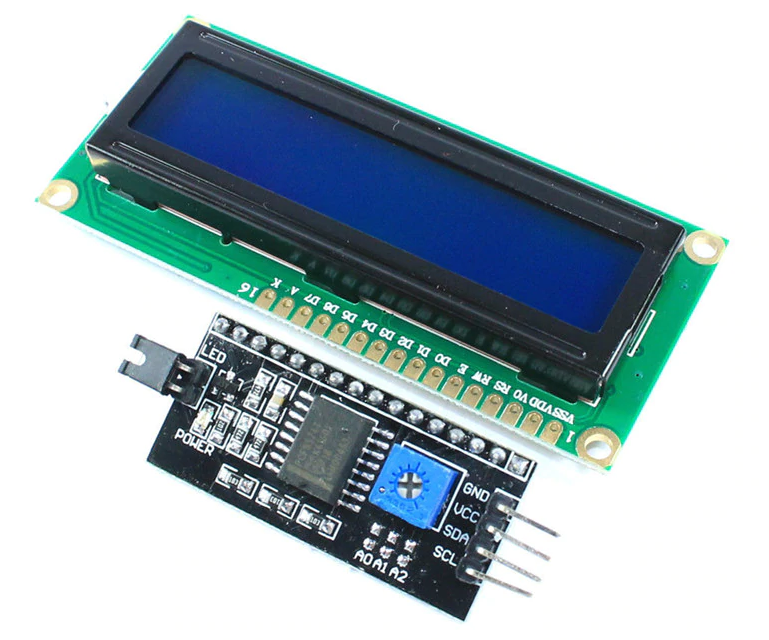
Později jsem pracoval také s 32bit mikroprocesorem Arduino, z důvodu sdíleného 5-ti voltového napětí mikroprocesoru a čtečky otisků prstů.



Obr. 4 Datasheet Arduino Nano

## LCD displej

Pro interaktivní komunikaci s uživatelem jsem použil 16x2 modrý LCD displej s řízením přes I2C sběrnici, pro ušetření datových pinů. Kontrast je nastavitelný trimrem.



Obr. 5 LCD displej

# realizace a problematika

Než mi přišly potřebné objednané součástky, tak jsem se zabýval studováním materiálů, tzv. „datasheetu“ o samotné čtečce, jejím zapojením a funkčností. Také jsem se zabýval připojením displeje a připojením ESP k Wi-fi pomocí Homie.

Po obdržení čtečky se vyskytli největší potíže. Ani ne tak se samotnou čtečkou, ale s mikroprocesorem ESP, který s ní dlouhou dobu nekomunikoval. Prvním z problému byla knihovna SoftwareSerial, která s mikroprocesorem nepracuje správně. Poté se vyskytl problém rovněž s knihovnou Adafruit fingerprint. Obě mají naštestí alternativu pro ESP. Dalším problémem při navázaní komunikace mezi zařízeními byla seriová rychlost obou zařízení. Což mělo za přičinu výpis nesmyslných znaků v debugovacím terminálu. Nakonec se zjistilo, že hlavním problémem bylo právě používané ESP (Lolin NodeMCU V3), které nedokázalo správně vypisovat na debugovací konzoli. Nahradil jsem ho ESP D1\_mini, na kterém vypisování fungovalo a přidal jsem k němu patřičné knihovny.

Napájení senzoru byl jeden z největších problémů. Podařil se vyřešit za pomoci zesílení výstupního napětí mikroprocesoru a redukci výstupního napětí čtečky.

I přes vyřešení těchto problému napsaných výše, se mi nepodařilo dosáhnout 100% funkčnosti čtečky s mikroprocesorem ESP. Bohužel čtečka někdy otisk snímá a někdy ne. Pokoušel jsem se pomocí USB TTL serial kabelu debugovat připojení. Zjištěním bylo, že se data neposílají celá.

# Způsoby řešení PROBLÉmŮ a použité postupy

## LCD displej

Ke sprovoznění dislpeje s mikroprocesorem je zapotřebí knihovna LiquidCrystal\_I2C, nastaveni adresy a inicializaci displeje. Viz ukázka kódů. Dále připojení přes I2C sběrnici pomocí 4 pinů –VCC, GND a dvou datových pinů.

#include <LiquidCrystal\_I2C.h> //Přidání knihovny

//Natavení adresy LCD, definování displeje na 16 znaků a 2 řádky

LiquidCrystal\_I2C lcd(0x27,16,2);

void setup(){

lcd.init(); //inicializace displeje

lcd.backlight(); //zapnutí/nastavení podsvícení

lcd.print("Prvni radek, "); //výpis hlášky

lcd.setCursor(0, 1); //nastaveni pozice

lcd.print("druhy radek"); //následný výpis na nastavené pozici

}

## Snímač otisků prstů s pamětí DY50

Ke správné funkčnosti potřebujeme 5 V napájení, připojení čtečky na pin GND a další dva piny s možností seriové komunikace. Pro připojení k ESP požívám piny D5 a D6 a pro připoejní k Arduinou piny 2 a 3. Je vyžadováno připojení dvou knihoven a to Adafruit fingerpritf a SoftwareSerial nebo nějakou alternativu těchto knihoven. Kód čtečky je rozdělen na dvě části. První je tzv. Enrollment nebo-li přidání nových otisků do paměti a druhá část je porovnání již uloženého otisku s otiskem právě prováděným. Čtečka pracuje se seriovou ryhlosti od 57600 do 115200.

## ESP 8266, knihovny, seriová komunikace a rychlosti

Knihovny doporučené firmou Adafruit k použití s mikroprocesorem Arduino se ukázaly jako nevhodné za předpokladu použití ESP . Při použití těchto knihoven čtečka s ESP nekomunikovala. Místo knihovny Adafruit fingerprint jsem použil alternativní verzi pro ESP. Avšak s knihovnou SoftwareSerial se objevily větší potíže. Než se mi podařilo najít vhodnou alternativu téeo knihovny, tak jsem vystřídal mnohé možnosti. Mezi které patřily například knihovny: EspSoftwareSerial, SomeSerial, ESPeasySerial a SoftwareSerial32. Bohužel žádná z uvedených nefungovala. Rozhodl jsem se tedy obejít knihovnu SoftwareSerial knihovnou HardwareSerial, která nebyla také řešením problému. Nakonec se ukázalo, že problém byl v momentálně používaném ESP (Lolin NodeMCU V3), což jsem zjistil, díky připojení debugovacího USB TTL kabelu. Nakonec jsem použil ESP 8266 D1\_mini, na kterém fungovala doporučená knihovna SoftwareSerial.

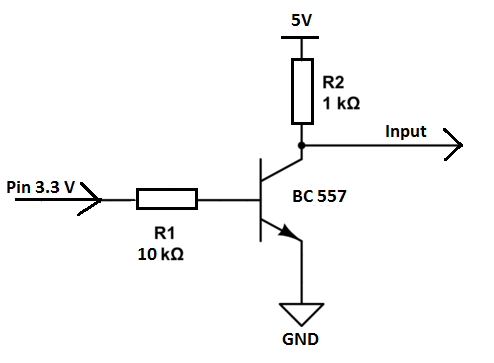
Čtečka funguje v rozmezí rychlostí 57600 až 115200 a ESP v rozmezí 9600 až 57600, což je tzv. baud rate. Jak jsem se později dozvěděl na různých stránkách a fórech, tak s rychlostmi u ESP je obecně veliký problém. Vyřešení tohoto problému bylo otázkou zkoušení. Dále pak nastavení rychlosti samotné frekvence procesoru ESP. Pro funkčnost se čtečkou je nutné rychlost nastavit na 80000000L. Než jsem tyto problémy vyřešil, tak se mi v debugovacím terminálu vypisovaly náhodné znaky a nebyl jsem schopen zjistít zda-li je čtečka připojena.

//obrazek

## Napájení Čtečky s ESP

### Zesílení napětí výstupu ESP na 5 V

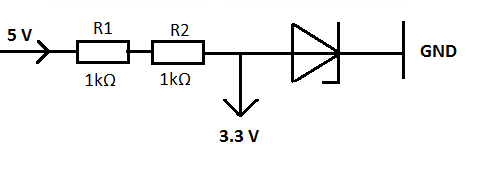
Jelikož je čip ESP napájen 3.3 V a čtečka ke správnému běhu potřebuje napětí v rozmezí 3.6 V až 6 V musel jsem posílit data z ESP pinu pomocí rezistorů a tranzistoru.



Obr. 6 Schéma zesílení

### Zeslabení napští na vstupu do ESP na 3.3 V

Aby čip nezhořel při 5 V vstupu dat na pin ESP bylo nutné zmenšit napětí. Použil jsem dva 1kΩ rezistory a pro vylepšení zenerovu diodu.



Obr. 7 Schéma zeslabení

**ZÁVĚR**

Cílem projektu bylo uvést do provozu čtečku otisků prstů s mikroprocesorem ESP 8266, připojit mikroprocesor na Wi.fi a odeslat otisk prstu. Dále otisk analyzovat, zaznamenat a uložit.

Tento projekt byl vymyšlen za účelem možného odstranění RFID čteček a nahrazení čtečkami otisků prstů. Dále má čtečka mnoho jiných praktických využití. Na rozdíl od čteček, které se dají zakoupit na internetu má centralizovanou databazi otisků a možnost monitorování např. vstupu a východu určitých osob z budovy.

Bohužel jsem nesplnil zadaný cíl projektu z důvodu nevhodného výběru mikroprocesoru a spoustu problémů s ním spojeným. Nicmémě se mi podařilo řadu problémů vyřešit a dostat se tak alespoň k částečné funkčnosti. Jelikož mě tento projekt velmi zaujal a spostu nových věcí jsem se naučil, tak bych chtěl tento projekt dotáhnout do konce. Ať už volbou jiných hardwarových součástí nebo vyřešením dosud nevyřešených problémů.

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] *Adafruit fingerprint datasheet* [online]. 2018 [cit. 2019-01-06]. Dostupné z: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf>

[2] *Adafruit fingerprint datasheet* [online]. 2016 [cit. 2019-01-06]. Dostupné z: <https://www.marutsu.co.jp/contents/shop/marutsu/ds/adafruit-optical-fingerprint-sensor.pdf>

[3] *Arduino fingerprint sensor tutorial*  [online]. 2017 [cit. 2019-01-06]. Dostupné z: <https://www.hackster.io/nickthegreek82/arduino-fingerprint-sensor-tutorial-103bb4>

[4] *The engineering projects* [online]. 2017 [cit. 2019-01-06]. Dostupné z: <https://www.theengineeringprojects.com/2017/01/use-arduino-software-serial.html>

[5] *Arduino forum: forum* [online]. 2015 [cit. 2019-01-06]. Dostupné z: <https://forum.arduino.cc/index.php?topic=354933.0>

[6] *Eletronics* [online]. [cit. 2019-01-06]. Dostupné z: https://electronics.stackexchange.com/questions/56093/how-to-use-a-3v-output-to-control-a-5v-relay

[7] *Espressif* [online]. 2018 [cit. 2019-01-06]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\_datasheet\_en.pdf

[8] *ESP serial comumnication* [online]. [cit. 2019-01-06]. Dostupné z: https://circuits4you.com/2016/12/16/esp8266-serial-communication/