

# **STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST**

**Obor: 10. Elektrotechnika, elektronika a telekomunikace**

## **Integrace do průmyslu 4.0**

**Jakub Andrýsek**

**Brno 2021**

# **STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST**

**Obor: 10. Elektrotechnika, elektronika a telekomunikace**

## **INTEGRACE DO PRŮMÝSLU 4.0**

**INTEGRATION INTO INDUSTRY 4.0**

**AUTOR** Jakub Andrýsek

**ŠKOLA** Gymnázium Brno, Vídeňská,  
příspěvková organizace

**KRAJ** Jihomoravský

**ŠKOLITEL** Mgr. Jaroslav Páral

**OBOR** 10. Elektrotechnika, elektronika  
a telekomunikace

**Brno 2021**

## **Prohlášení**

Prohlašuji, že svou práci na téma *Integrace do průmyslu 4.0* jsem vypracoval samostatně pod vedením Mgr. Jaroslava Párala a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Dále prohlašuji, že tištěná i elektronická verze práce SOČ jsou shodné a nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a změně některých zákonů (autorský zákon) v platném změně.

V Brně dne: \_\_\_\_\_

---

Jakub Andrýsek

## **Poděkování**

Děkuji svému školiteli Mgr. Jaroslavu Páralovi za obětavou pomoc, podnětné připomínky a hlavně nekonečnou trpělivost, kterou mi během práce poskytoval.

Tato práce byla provedena za finanční podpory Jihomoravského kraje.

**jihomoravský kraj**



## **Anotace**

Cílem mé práce bylo navrhnout ucelený automatický systém monitorující chod pletacích strojů a přizpůsobit ho co možná nejlépe potřebám výrobní firmy.

Systém jsem navrhoval na míru pro rodinnou firmu na pletení ponožek a v současnosti je schopen v reálném čase zaznamenávat a následně odesílat naměřená data ze strojů na server. Pro uživatele systém nabízí moderní webové stránky, kde si může naměřená data přehledně zobrazit a analyzovat.

Systém se skládá ze senzorové části, serveru a podpůrného serveru. Senzorová část je připojená k pletacímu stroji a odesílá naměřená data. Server zajišťuje veškeré zpracování dat a jejich zobrazení uživateli. Podpůrný server se stará o aktualizaci a o kontrolu správného chodu senzorů.

## **Klíčová slova**

IoT, ESP32, web, PHP, Nette, databáze, open-source, průmysl 4.0, automatizace, modernizace

## **Annotation**

The goal of my work was to design a comprehensive automatic system that monitors the operation of knitting machines according to the needs of a manufacturing company.

I designed a production monitoring system for a family sock knitting business that can record measured data and submit them from machines to a server via the wireless network in real-time. The system offers a modern web-based user interface where the user can clearly display and analyze the measured data in a form of a real-time dashboard, statistic tables and graphs.

The system consists of three subsystems - sensors, a main server, and a support server. The sensors are connected to the machines and send the data to the main server via WiFi. The main server processes all the data and displays it to the user. The support server takes care of updating and checking the correct operation of the sensors.

## **Keywords**

IoT, ESP32, web, PHP, Nette, database, open-source, industry 4.0, automation, modernization

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Integrace do průmyslu 4.0</b>	<b>3</b>
1.1 Práce operátorů . . . . .	3
1.2 Popis . . . . .	4
1.3 Řešení . . . . .	4
1.4 Nasazení . . . . .	4
<b>2 Porovnání automatických systémů</b>	<b>6</b>
2.1 Hardware . . . . .	6
2.1.1 PLC . . . . .	6
2.1.2 Controllino . . . . .	7
2.1.3 Hardwario . . . . .	7
2.1.4 Arduino . . . . .	7
2.1.5 Srovnání . . . . .	7
2.2 Software . . . . .	9
2.2.1 Node-RED . . . . .	9
2.2.2 Blynk . . . . .	9
2.2.3 Home Assistant . . . . .	9
2.2.4 Porovnání softwarové části . . . . .	9
<b>3 Senzory</b>	<b>10</b>
3.1 Procesor . . . . .	10

3.2	Vstupy . . . . .	11
3.3	Výstupy . . . . .	12
3.4	Vlastní PCB . . . . .	12
3.5	Verze 1 - univerzální senzorika . . . . .	13
3.5.1	Řídící deska . . . . .	13
3.5.2	Kryt a uchycení senzoru . . . . .	13
3.5.3	Program . . . . .	14
3.6	Verze 2 - speciální senzorika . . . . .	16
3.6.1	Řídící deska . . . . .	16
3.6.2	Uchycení . . . . .	17
3.6.3	Program . . . . .	17
<b>4</b>	<b>Webový server</b>	<b>19</b>
4.1	Frontend . . . . .	19
4.1.1	Bootstrap . . . . .	19
4.1.2	JavaScript . . . . .	20
4.2	Backend . . . . .	20
4.2.1	PHP . . . . .	20
4.2.2	Nette . . . . .	21
4.2.3	REST API . . . . .	21
4.3	Funkcionalita . . . . .	21
4.3.1	Podrobné statistiky . . . . .	21
4.3.2	Aktuální přehledy . . . . .	21
4.3.3	Responzivní design . . . . .	22
4.3.4	Chytrý výpočetní algoritmus . . . . .	22
4.3.5	Jednoduché grafy . . . . .	22
4.3.6	Kontrola běhu stroje . . . . .	22
4.3.7	Jednoduchý výběr dat . . . . .	22
4.3.8	Porovnávání směn . . . . .	22
4.3.9	Bezpečné zálohy . . . . .	23
4.3.10	Jednoduchý generátor zkušebních dat . . . . .	23
4.4	Webové rozhraní Pletačka IoT . . . . .	23

4.4.1	Úvodní stránka	23
4.4.2	Přehled ze senzoru	24
4.4.3	Správa senzorů	24
4.4.4	Nastavení směn	24
4.5	Databáze	27
<b>5</b>	<b>Podpůrný server</b>	<b>28</b>
5.1	Kontrola senzorů	28
5.2	Automatické aktualizace	28
<b>6</b>	<b>Princip fungování Pletačka IoT</b>	<b>29</b>
6.1	Sběr dat	29
6.2	Vyhodnocování dat	29
6.3	Zobrazování dat	30
6.4	Konektivita	30
<b>7</b>	<b>Vývoj</b>	<b>31</b>
7.1	Systém Pletačka IoT verze 1.0	31
7.1.1	Senzory	31
7.1.2	Web	32
7.2	Systém Pletačka IoT verze 2.0	32
7.2.1	Senzory	32
7.2.2	Web	32
<b>8</b>	<b>Testování</b>	<b>33</b>
8.1	Domácí testování	33
8.2	Testování ve firmě	33
<b>9</b>	<b>Zpětná vazba majitele firmy</b>	<b>35</b>
<b>Literatura</b>		<b>38</b>
<b>Přílohy</b>		<b>39</b>

# Seznam obrázků

1	Senzor na stroji . . . . .	2
1.1	Pletárna ponožek . . . . .	5
3.1	TTGO T-Display . . . . .	11
3.2	Senzor - 1. verze . . . . .	15
3.3	Senzor - 2. verze . . . . .	18
4.1	Domovská stránka . . . . .	24
4.2	Přehled ze senzoru . . . . .	25
4.3	Správa senzorů . . . . .	26
4.4	Nastavení směn . . . . .	26
4.5	Struktura databáze . . . . .	27
6.1	Zpracování dat . . . . .	30
8.1	Testování senzoru . . . . .	34
9.1	Senzory . . . . .	37
2	Schéma senzoru 1. verze . . . . .	39
3	Schéma senzoru 2. verze . . . . .	40
4	Testování funkčnosti zapojení . . . . .	40
5	Testovací verze senzoru . . . . .	41
6	Webový server Raspberry Pi . . . . .	41
7	Pletací stroj . . . . .	42

# **Seznam tabulek**

2.1 Tabulka srovnání hardwarové konkurence. . . . .	8
---	---

# Úvod

S nápadem vytvořit systém monitorující chod pletacích strojů přišel můj děda, zakladatel firmy na výrobu ponožek. Jeho snem vždy bylo mít systém, který by částečně zastal monotónní lidskou práci a nahradil ji efektivní automatizací. Tím myslел například automatické počítání vyrobených ponožek, hlášení poruch strojů a podobné monitorování výroby.

Můj systém jsem tedy navrhoval na míru pro rodinnou firmu na pletení ponožek, ve které je okolo 25 pletacích strojů.

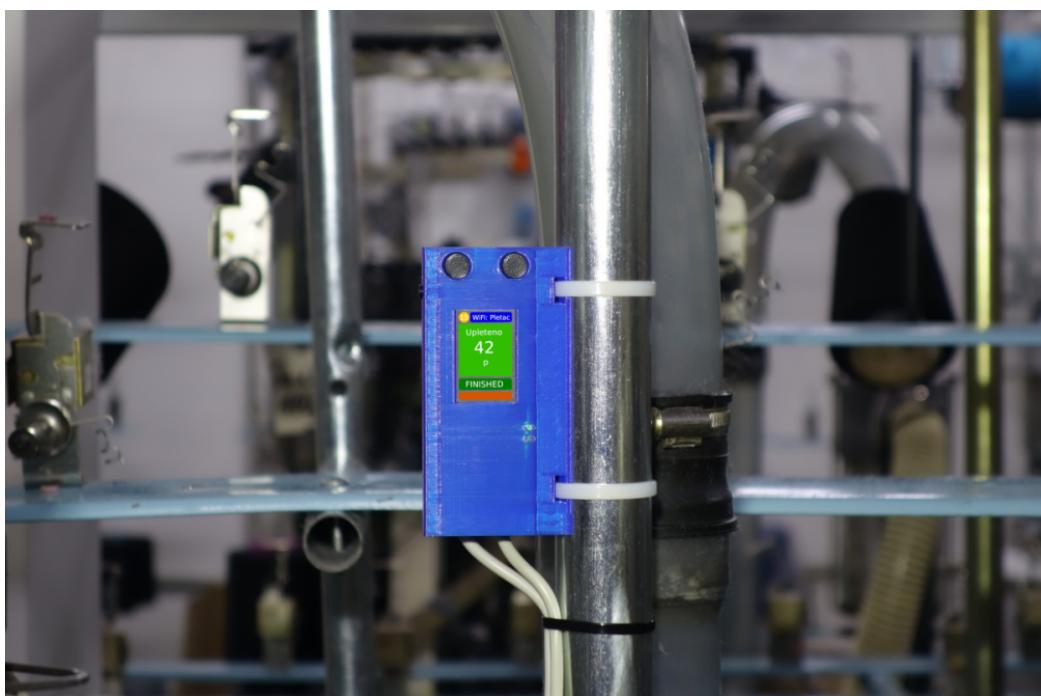
Podle pletacích strojů, na kterých tento systém běží, jsem projekt pojmenoval Pletačka IoT. Systém se skládá ze tří částí. Senzorová část, která je připojená k pletacímu stroji a odesílá data. Dále pak server, který veškerá data zpracovává a zobrazuje je uživateli. Poslední částí je podpůrný server, který se stará o aktualizaci a kontrolu správného chodu senzorů.

Při vytváření tohoto projektu jsem si dal za cíl:

- projekt s otevřeným zdrojovým kódem
- cenová dostupnost
- jednoduché přidání senzorů
- přehledné uživatelské rozhraní

Pro systém jsem si stanovil tyto požadavky

- Počítání upletených ponožek
- Zjišťování poruchovosti strojů
- Porovnání jednotlivých pracovních směn
- Monitorování průběhu výroby



Obrázek 1: Senzor na stroji

# Kapitola 1

## Integrace do průmyslu 4.0

Pojem Průmysl 4.0 se do České republiky dostal okolo roku 2013 a od té doby se stále více rozšiřuje v průmyslových firmách. Jedna z klíčových částí je IoT (Internet of Things), neboli internet věcí, který nám zajišťuje vzdálenou kontrolu a řízení strojů pomocí elektroniky, senzorů a různých softwarů. Další vlastností těchto systémů je zaznamenávání a následné ukládání dat do datových úložišť. Moderní IoT řídící systémy se snaží proniknout co nejvíce do hloubky a zpřesnit tak naměřená data, důležitá pro optimalizaci produkce.

### 1.1 Práce operátorů

Obě dvě pracovní směny se skládají ze tří operátorů. Jeden má na starosti pět až osm pletacích strojů. Z každého pletacího stroje vypadne každé čtyři minuty jedna upletené ponožka, kterou operátor musí zkontolovat a otočit naruby pro další zpracování. Dalším úkolem operátorů je doplnování materiálu a oprava strojů při poruše.

## 1.2 Popis

Při návrhu mého systému jsem se snažil řídit těmito zásadami a navrhnut tak co nejmodernější a provozně efektivní systém. Základem bylo zhodnocení stávající situace a navržení možného řešení.

Jednotlivé problémy neautomatizované výroby:

- dlouhá doba stání nečinných strojů při poruše i nedostatku materiálu
- neznalost doby nečinnosti stroje
- ruční počítání vyprodukovaného zboží
- absence historického přehledu produkce

## 1.3 Řešení

Mým řešením je návrh moderního systému, který by celý výrobní provoz monitoroval a zobrazoval operátorovi u stroje a zaměstnavateli. Systém umožňuje on-line zhodnocení jednotlivých směn a jejich vzájemné porovnání. Systém neustále vyvíjím a rozšiřuji podle potřeb firmy.

Současné funkce systému Pletačka IoT

- monitoring chodu každého pletacího stroje
- on-line hlášení závady na stroji
- monitoring provozu firmy
- porovnání pracovních směn
- počítání upletených ponožek

## 1.4 Nasazení

Tento systém je aktuálně nasazen ve firmě ROTEX Vysočina [1], která se věnuje výrobě ponožek. Firma pracuje ve dvousměnném provozu a týdně vyprodukuje v průměru 12000 párů ponožek.

Díky mému systému Pletačka IoT se ve firmě optimalizovala produkce a výkon strojů, a tím zefektivnila výroba. Toto se podařilo především díky minimizování času nečinnosti pletacích strojů z důvodu on-line hlášení poruch a úsporou času operátora výroby při dřívějším ručním počítáním upletených ponožek. vJako bonus navíc systém vyhodnocuje produktivitu jednotlivé směny.



Obrázek 1.1: Pletárna ponožek

# Kapitola 2

## Porovnání automatických systémů

Můj systém Pletačka IoT je velice specifický a nedá se srovnávat jako celek. Potenciální konkurenci tohoto systému jsem tedy rozdělil na dva celky.

- Hardwarová část
- Softwarová část

### 2.1 Hardware

#### 2.1.1 PLC

PLC neboli programovatelný logický automat je průmyslový počítač k řízení automatizovaných procesů. Automaty zpracovávají data v reálném čase a s co nejkratší odezvou. PLC jsou modulární a dají se skládat různě dohromady, podle potřeby uživatele.

### **2.1.2 Controllino**

Firma Controllino [4] se zabývá vývojem zařízení pro průmyslovou automatizaci založenou na platformě Arduino. Zařízení nabízí několik vstupních a výstupních pinů, pomocí kterých si uživatel může připojit své senzory a následně automatizovat některé procesy.

### **2.1.3 Hardwario**

Hardwario [3] je česká firma, která nabízí průmyslové IoT stavebnice. Cílem této firmy je nabídnout průmyslové IoT řešení, které si sami sestavíte podle svých představ. Firma se zaměřuje na nízkoenergetické moduly s výdrží několika let.

### **2.1.4 Arduino**

Arduino [2] je otevřený (open source) projekt, který se díky své nízké ceně a jednoduchosti na používání rozšířil po celém světě. Arduino má v nabídce přes deset různých modelů. Desky jsou univerzální a jsou často využívány na kutilské projekty. K Arduinu také existuje velké množství shieldů, které základním modulům dodávají další funkcionality. Desky Arduino se programují v jazyce Wiring. Jazyk Wiring je mírně upravený jazyk C++ a používá se k programování mikrokontrolérů.

### **2.1.5 Srovnání**

První tří zmíněné platformy jsou hojně využívány v průmyslu a řídí většinu automatizovaných procesů. Jejich nasazení je složité a celé systémy jsou velmi drahé.

<b>Hardware</b>	1	2	3	4	5	6
PLC	✓	✓	✗	✗	✗	✗
Controllino	✓	✓	✓	✗	✗	✗
Hardwario	✓	✗	✓	✗	✓	✗
Arduino	✗	✗	✓	✗	✗	✗
<b>Moje řešení</b>	✓	✓	✓	✓	✓	✓

Tabulka 2.1: Tabulka srovnání hardwarové konkurence.

Požadavky na hardwarovou část

1. Připraveno na montáž na zařízení
2. Průmyslové napětí 5-25 V
3. Open source
4. Barevný displej
5. Bezdrátová aktualizace senzorů
6. Moderní konektor USB-C

## 2.2 Software

### 2.2.1 Node-RED

Node-RED je jednoduché grafické prostředí k programování IoT zařízení. Hlavní výhodou této aplikace je, že celá běží jako webová stránka. Tím umožnuje uživateli rychlou práci bez nutnosti instalovat speciální aplikace. Node-RED programování stojí na principu propojování jednotlivých uzelů. Ve složitějších projektech mohou být ovšem bloky dosti nepřehledné a složité na úpravu.

### 2.2.2 Blynk

Blynk je platforma pro vzdálené ovládání IoT projektů. Základem platformy je jednoduchá mobilní aplikace pro nastavování a vyčítání dat. Aplikace nabízí velké množství widgetů, které se připínají na zobrazovací panel. Na osobní projekty do pěti zařízení je aplikace zdarma, jinak je nutné platit měsíční poplatky.

### 2.2.3 Home Assistant

Home Assistant je software pro řízení chytrých domácností. Systém dokáže pracovat s více než 1700 službami. Aplikace také dokáže integrovat mnoho rozšíření, například ESPHome. To slouží k ovládání mikrokontrolérů ESP které jsou hojně rozšířené v kutilské komunitě.

### 2.2.4 Porovnání softwarové části

Node-RED a Home Assistant jsou projekty s otevřeným zdrojovým kódem, utvářené komunitou. Díky tomu jsou tyto systémy modulární a rychle se rozvíjejí. Naopak Blynk je uzavřená platforma zaměřená na firmy a vývojáře. Můj systém spojuje užitečné vlastnosti ze všech těchto systémů a nabízí je jako celek v podobě systému Pletačka IoT.

# Kapitola 3

## Senzory

Senzory k projektu Pletačka IoT jsou postavené na mikročipu ESP32 a modulu TTGO T-Display. Celý tento systém je navržen tak, že na každém pletacím stroji je jeden senzor. Každý z těchto senzorů má svoje jedinečné číslo, pod kterým posílá naměřená data na server. Senzor je napájen z 5 nebo 24 V a má spotřebu 120 mA. Návrh senzorů i jejich software mám verzovaný nástrojem Git ve veřejném repozitáři na GitHubu.

GitHub: [Pletacka-board](#)[5]

### 3.1 Procesor

Jako řídící procesor jsem zvolil čip ESP32, protože je velice výkonný a disponuje konektivitou WiFi. Procesor obsahuje 4 MB flash paměti sloužící k ukládání programu.

Po prozkoumání trhu jsem našel modul TTGO T-Display, který kombinuje barevný displej s čipem ESP32. Tato kombinace mi vyšla jako nejlepší. Spojuje bezproblémovou komunikaci čipu s displejem a zároveň jednoduchou výměnu při nefunkčnosti modulu.



Obrázek 3.1: TTGO T-Display

## 3.2 Vstupy

Napájecí okruh pletacího stroje pracuje s napětím 24 V, proto jsem potřeboval, aby i moje elektronika, dokázala takovéto napětí zpracovávat.

Jeden vstup je připojen ke světlu, které signalizuje zastavení stroje a druhý k senzoru, který zaznamenává dopletenou ponožku. Tyto vstupní signály jsou připojeny na optočleny, které vodivě oddělí vstupní napájení. Na výstupu máme poté pouze tří voltový signál, který je zpracovatelný čipem.

Jako uživatelské vstupní periferie jsem využil jednoduchá tlačítka, pomocí kterých si uživatel upravuje nastavení senzoru.

### **3.3 Výstupy**

Jako hlavní zobrazovací článek jsem využil barevný TFT displej o velikosti 1,14 palce. Na displeji se zobrazuje ID zařízení, počet upletených ponožek a doba stání stroje. Spodní část je vyhrazena na logovací a chybové hlášky.

Druhým výstupním prvkem jsou barevné diody. Ty slouží pro signalizaci odesílání dat a k jednoduchému zjištění funkčnosti celé elektroniky.

### **3.4 Vlastní PCB**

K vytvoření vlastní řídící desky mě vedlo několik faktorů. První z nich byla velikost celé elektroniky. Nevýhodou elektroniky postavené z existujících modulů je právě jejich velikost, moduly se dají jednoduše skládat k sobě, ale zabírají spoustu místa.

Dalším důvodem je replikovatelnost. Těmito deskami plánuji osadit všechny pletací stroje ve firmě, což znamená vyrobit 25 stejných elektronických desek.

Desky jsem si tedy nechával vyrábět v čínské firmě JLCPCB. Jejich výroba je velmi precizní a dokáží desku také osadit vybranými součástkami.

## 3.5 Verze 1 - univerzální senzorika

První verzi jsem pojal jako testovací. Bylo třeba navrhnout univerzální desku a otestovat celý systém.

Při navrhování senzoru jsem si stanovil tyto body:

- ESP32 s barevným displejem
- vstup ze 4 periferií
- vstupní napětí od 10 do 25V
- teplotní čidlo
- tři barevné diody
- čtyři uživatelská tlačítka

### 3.5.1 Řídící deska

Návrh desky jsem vytvořil v aplikaci EAGLE od společnosti Autodesk. Deska má rozměry  $75 \times 60$  mm a v každém rohu má upevňovací otvory. Kabely se do desky připojují pomocí 5 mm svorkovnice. Na vstupu napájení je měnič napětí, který pracuje v rozsahu od 10 do 25 V a na výstupu dává 5V.

Řídící procesor celé desky je modul ESP32 TTGO T-Display. Tento čip také zajišťuje WiFi konektivitu s okolím a odesílá naměřená data na server. Pro univerzální detekování vstupů z periferií se využívají optočleny, které předávají signál do mikroprocesoru. K uživatelskému ovládání senzoru jsou zde čtyři programovatelná tlačítka a tři indikační diody. Aktuální naměřená data se zobrazují na displeji a informují obsluhu o zastavení stroje a počtu upletených párů ponožek. Senzor je schopen zaznamenávat data ze čtyř vstupů a teplotu z teplotního senzoru. Viz obrázek 3.2.

### 3.5.2 Kryt a uchycení senzoru

Kryt řídící desky jsem navrhl v aplikaci Fusion 360 a vyexportoval jsem ho do formátu STL. Tento model jsem následně vytisknul na 3D tiskárně

Průša z materiálu PETG. Na přední straně je průhled z plexiskla na barevný displej a okolo něj jsou rozmístěná uživatelská tlačítka. Na boční straně krytu jsou připravené dvě drážky na protažení stahovacích zip pásků pro uchycení na sloupek pletacího stroje. Signálové kabely jsou poté svedeny po konstrukci stroje až k periferiím.

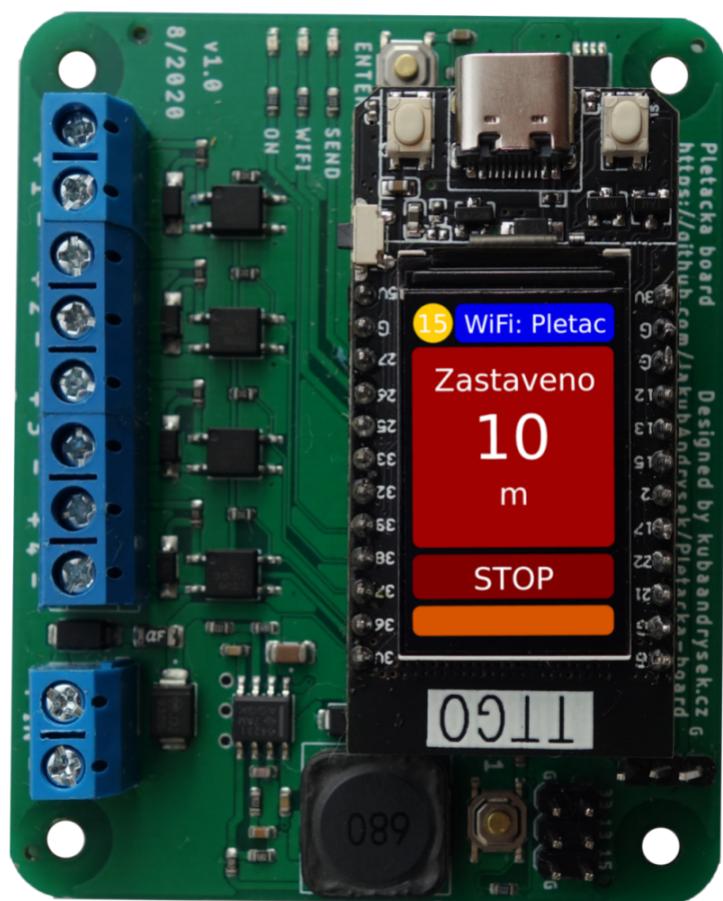
### 3.5.3 Program

K programování využívám aplikaci Visual Studio Code s rozšířením PlatformIO, která je navržena k programování mikrokontrolérů. Zdrojový kód mám napsaný v jazyce C++.

Program se skládá z několika vláken, které se pravidelně spouštějí a vykonávají. První a zároveň nejdůležitější vlákno je senzorové. Zde se periodicky kontroluje stav periferií a při změně se odešle událost na server. Další vlákno zajišťuje pravidelné vykreslování dat na displej a zbylá vlákna se starají o správný chod senzoru.

Software také obsahuje ladící mód, ve kterém si administrátor může zobrazit stav senzoru v mobilní aplikaci a jednodušeji tak hledat potenciální chybu.

Celý program jsem napsal objektově orientovaným programováním, díky čemuž je velmi jednoduché měnit například počet vstupních periferií. Každé nové periferii stačí nastavit správný typ, pin na který je připojena a její název. Poté už stačí zavolat například vyčítací metodu, která vrátí stav tlačítka.



Obrázek 3.2: Senzor - 1. verze

## 3.6 Verze 2 - speciální senzorika

Po měsíci testování jsem zhodnotil využití jednotlivých součástek a následně jsem vytvořil nový seznam požadavků, přizpůsobený pro lepší chod senzoru. Zařízení je díky tomu mnohem menší, levnější a softwarově rychlejší.

- vstup pouze ze 2 periferií
- vstupní napětí již od 5V
- zredukování rozměrů
- moderní USB-C konektor
- zredukování na dvě tlačítka a dvě indikační diody
- možnost přímého napájení senzoru bez měniče
- bezdrátová aktualizace senzorů

### 3.6.1 Řídící deska

Návrh druhé desky jsem se rozhodl udělat v open source aplikaci KiCad. Z uživatelského hlediska se s ní pracuje o něco rychleji díky jednoduchým klávesovým zkratkám. Dalším důvodem pro zvolení této aplikace bylo rozšíření KiKit, které razantně zjednodušuje export výrobních podkladů. Zatímco v dříve používané aplikaci Eagle jsem při každé změně musel projít zdlouhavým procesem exportu, nyní v linuxovém terminálu stačí zavolat příkaz ‘make’ a celý proces se vykoná automatizovaně bez nutnosti editace dat.

Rozšíření KiKit jsem také začal používat k automatizovanému generování dokumentace k plošným spojům. Rozvržení webové stránky si uživatel nastaví v konfiguračním souboru a následně při každé změně se stránka přegeneruje a aktualizuje.

V novém návrhu jsem se především zaměřoval na rozměr desky. Ten aktuálně činí  $32 \times 76$  mm, což je o 46 % menší plocha než u první verze.

Deska si zachovala stejný procesor ESP32 s displejem, ale přišla o dvě tlačítka a jednu indikační diodu. Ty jsem ani v předchozí verzi nepoužíval, proto

jsem se je rozhodl odstranit.

V senzoru se také změnilo zapojení měniče napětí. Nově dokáže pracovat již od 5V, které následně mění na 3,3V. Na bočních stranách desky vznikla také nová ”křidélka” pro zasunutí do vylepšeného krytu.

### **3.6.2 Uchycení**

Druhá verze využívá stejného principu uchycení, jako ta předchozí. Mění se zde však spojení krabičky se senzorovou deskou. V nové verzi jsem desku navrhl tak, aby se dala jednoduše zasunout do kolejnic, které jsou předtištěné v krabičce a následně zafixovat šroubkem ze zadní strany. To umožňuje jednoduchou montáž a rychlé připojení. Tento návrh už má také vyřešené zafixování kabelů ke konstrukci krabičky pomocí 3D tištěné svorky.

### **3.6.3 Program**

Program druhé verze vychází z minulé, ale přináší s sebou nové funkce a vylepšuje stávající. Novou funkcionalitou je například automatická aktualizace programu přes WiFi, kterou nadále zdokonaluji. Další vylepšení jsem provedl u displeje, který dokáže zobrazit více údajů a automaticky mezi nimi přepínat.



Obrázek 3.3: Senzor - 2. verze

# Kapitola 4

## Webový server

Webový server je nasazený na mikropočítači Raspberry Pi 4 Modelu B. Toto zařízení jsem zvolil hlavně kvůli nízké spotřebě elektrické energie a velké komunitě lidí, kteří tento mikropočítač využívají.

Na zařízení běží operační systém Raspberry Pi OS s grafickým rozhraním. Webové stránky běží na HTTP serveru Apache2 a PHP 8.0. Jako databázový systém využívám MariaDB. Server běží lokálně uvnitř firmy v zabezpečené síti, díky čemuž je systém rychlý a nezávislý na internetovém připojení. Celý webový server mám verzovaný také na GitHubu.

GitHub: [Pletacka-website](#)[6]

### 4.1 Frontend

Frontend je vizuální část webové stránky zobrazená uživatelem. Pomocí frontendu se na obrazovku vykresluje veškerý text a jednotlivé prvky stránky.

#### 4.1.1 Bootstrap

Bootstrap je knihovna sloužící k jednoduchému a rychlému vytvoření responsivních webových stránek. Díky této knihovně jsou stránky správně zobra-

zovány i na mobilních zařízeních. Tento nástroj se využívá od roku 2011 a je pod otevřenou licencí. Webový server využívá Bootstrap verze čtyři.

### 4.1.2 JavaScript

Na frontendu používám JavaScript společně s technologií AJAX pro aktualizaci částí stránek. AJAX umožňuje překreslovat jen určitou část obsahu stránky bez nutnosti načítat celou stránku znova. Tím se zásadně zrychluje načítání a interaktivita stránek. Dochází i ke značné úspoře přenesených dat. K tomuto efektivnímu překreslování slouží knihovna Naja[7], kterou napsal český vývojář Jiří Pudil. Knihovna také nabízí jednoduchou integraci do PHP frameworku Nette, o kterém budu psát dále.

## 4.2 Backend

Backend je serverová část webových stránek, neběží tedy u vás na počítači jako frontend, ale na webovém serveru. Celý backend systému Pletačka IoT jsem napsal v programovacím jazyce PHP a ve frameworku Nette[8], který nabízí ucelenou sadu nástrojů k tvorbě webu. Backend se stará o přijímání dotazů ze senzorů a následný zápis do databáze, pohání celý webový server a vytváří databázové výběry. Nejdříve zde popíšu použité technologie a následně rozberu jednotlivé stránky aplikace.

### 4.2.1 PHP

Webovou aplikaci programuji v PHP ve verzi 8.0. Jako programovací studio jsem zvolil studentskou verzi aplikace PHPStorm, která je velmi mocným nástrojem při tvorbě webu. Testovací verzi aplikace mám spuštěnou na svém počítači, kde také celý tento systém využívám.

Pro snadnější ladění chyb používám Xdebug, díky kterému si můžu krokovat jednotlivé řádky kódu a rychleji tak nalézt chybu.

Jako systém pro správu balíčků používám nástroj Composer, který se ovládá z terminálu pomocí jednoduchých příkazů. Umožňuje rychlou definici závislostí a aktualizaci všech modulů pomocí jednoho příkazu.

### 4.2.2 Nette

Nette je webový framework vyvíjený komunitou. Vznikl v České republice a jeho zakladatelem je David Grudl. Nabízí vlastní šablonovací jazyk, na jednoduché a efektivní vykreslování webových stránek. Nette disponuje obsáhlou a velmi dobře zpracovanou dokumentací, ale také velkou komunitou lidí, kteří s tímto frameworkm pracují a velmi dobře mu rozumí.

### 4.2.3 REST API

REST API je sada URL identifikátorů sloužících ke komunikaci s webovou stránkou. Webová stránka Pletačka IoT obsahuje základní sadu API. Primárně ji využívají senzory k odesílání naměřených dat a ke zpětnému posílání odpovědí do senzoru. Druhé využití API je k vytváření databázových výběrů. To je voláno nástrojem na automatizaci procesů v nastavený čas.

## 4.3 Funkcionalita

Níže je popis všech funkcionalit systému.

### 4.3.1 Podrobné statistiky

Díky tomuto systému má uživatel kompletní přehled o každém zastavení stroje a upletené ponožce.

### 4.3.2 Aktuální přehledy

Na úvodní stránce se vždy zobrazují aktuální přehledy o průběhu výroby (viz obrázek 4.1).

### **4.3.3 Responzivní design**

Stránka využívá moderní CSS styly, díky kterým se stránka správně zobrazuje na jakémkoliv zařízení.

### **4.3.4 Chytrý výpočetní algoritmus**

Veškerá naměřená data jsou analyzována mým výpočetním algoritmem. Algoritmus přijímá uložená data z databáze, ze kterých postupným procházením vypočítává pracovní statistiky, které následně ukládá do užších výběrů.

### **4.3.5 Jednoduché grafy**

Většina nasbíraných dat se dá přehledně zobrazit v grafech. Díky nim je porovnávání a procházení výběrů velmi jednoduché (viz obrázek 4.2).

### **4.3.6 Kontrola běhu stroje**

Prvotním a nejdůležitějším požadavkem systému bylo sledování běhu stroje. K tomu slouží úvodní stránka aplikace, kde uživatel přehledně vidí ikony všech strojů. Podle barvy ikony dokáže rozeznat, zda stroj běží, nebo je v poruše a následně si může zobrazit další podrobnosti.

### **4.3.7 Jednoduchý výběr dat**

Každý senzor nabízí jednoduchý výběr dat. Uživatel si zvolí požadovaný rozsah dat a aplikace mu tento výběr přehledně zobrazí v tabulce.

### **4.3.8 Porovnávání směn**

Systém jsem navrhoval tak, aby pracoval s dvousměnným provozem a správně přiřazoval data k jednotlivým směnám.

### **4.3.9 Bezpečné zálohy**

Systém se každý den automaticky zálohují a ukládá data na záložní disk, ze kterého se dají případně rychle obnovit.

### **4.3.10 Jednoduchý generátor zkušebních dat**

Pro otestování systému jsem připravil jednoduchý generátor dat, který dokáže simulovat reálné senzory.

## **4.4 Webové rozhraní Pletačka IoT**

Každá stránka je rozdělena na tři části. První je záhlaví, které obsahuje logo a odkazy na nejpoužívanější stránky. Druhou částí jsou samotné webové stránky, které budou popsány v dalších odstavcích. Poslední částí je minimalistické zápatí s copyright znakem.

Stránky Pletačky jsem navrhoval tak, aby splňovaly tyto parametry:

- jednoduché rozhraní pro uživatele
- přehledné zobrazení dat
- zobrazovat pouze užitečná dat
- rychlá editace senzorů
- využití číselných identifikátorů

### **4.4.1 Úvodní stránka**

V horní části úvodní stránky (obr 4.1) se vypisují tři nejpodstatnější údaje. Jde o celkový počet upletených párů za aktuální směnu. Dále pak úspěšnost vypočítávanou z času stání stroje a z času zapnutí stroje. Posledním údajem je průměrná doba stání jednoho stroje.

Pod těmito čísly se zobrazuje tabulka s barevnými obdélníky, kde každý představuje jeden stroj. Barva obdélníků udává aktuální stav stroje a text v pozadí tuto informaci doplňuje.



Obrázek 4.1: Domovská stránka

#### 4.4.2 Přehled ze senzoru

Po kliknutí na senzor na úvodní stránce se zobrazí data o právě vybraném stroji. Veškerá data jsou rozdělena do dvou sloupců podle pracovních směn. To umožňuje zaměstnavateli jednoduché porovnávání pracovních směn. V úvodu každého sloupce je obecný přehled naměřených dat za různá období. Pod nimi je přehled v grafech a porovnání nejdůležitější údajů.

#### 4.4.3 Správa senzorů

Pro vstup do této sekce je nutné uživatelské přihlášení do systému. Stránka pak nabízí přehled senzorů s jednotlivými možnostmi úpravy (viz obrázek 4.3).

#### 4.4.4 Nastavení směn

Jednoduchá stránka, na které se nastavuje pořadí směn. Střídání směn probíhá pravidelně po týdnech, proto je nastavení velmi jednoduché (viz obrázek 4.4)

[Domů](#)
**Senzor: 5**

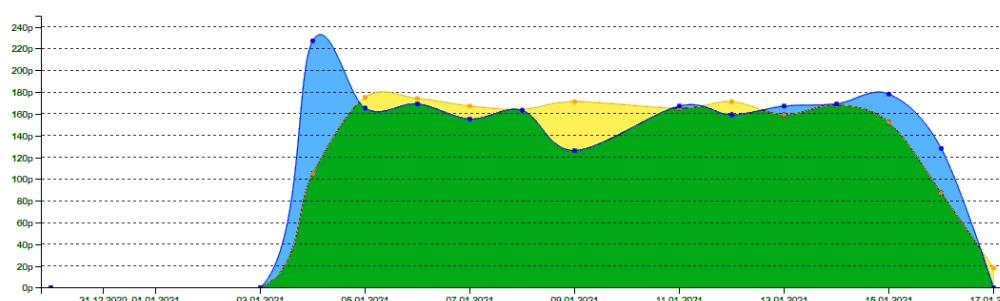
Dopolední směna

**Doba pletení**  
Poslední týden  
**52H 4M**
**Uplatoeno**  
Dnes  
**73 páru**
**Úspěšnost**  
Poslední měsíc  
**91%**
**Uplatoeno**  
Poslední měsíc  
**1011 páru**
**Počet zastavení**  
Poslední měsíc  
**337**
**Pr. doba stání**  
Poslední týden  
**1M 32S**

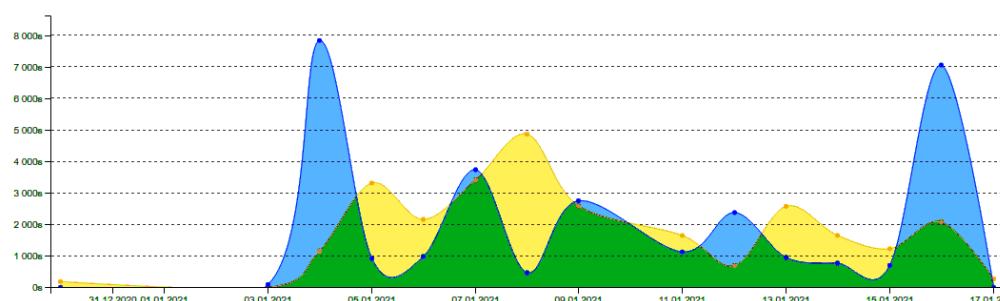
Odpolední směna

**Doba pletení**  
Poslední týden  
**55H 24M**
**Uplatoeno**  
Dnes  
**74 páru**
**Úspěšnost**  
Poslední měsíc  
**92%**
**Uplatoeno**  
Poslední měsíc  
**1060 páru**
**Počet zastavení**  
Poslední měsíc  
**314**
**Pr. doba stání**  
Poslední týden  
**1M 38S**
**Počet páru**

Posledních 30 dní


**Doba stání**

Posledních 30 dní



Obrázek 4.2: Přehled ze senzoru

## Výpis senzorů

Výpis senzorů					
Číslo	Popis	Datum změny	Zobrazit	Upravit	Smazat
1	Pletacka - 1	2020-11-08 00:14:59	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>
2	Pletacka - 2	2020-11-08 00:14:59	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>
3	Pletacka - 3	2020-11-08 00:15:03	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>
4	Pletacka - 4	2020-11-08 00:15:03	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>
5	Pletacka - 5	2020-11-08 00:15:04	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>
6	Pletacka - 6	2020-11-08 00:15:04	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>
7	Pletacka - 7	2020-11-08 00:15:05	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>
8	Pletacka - 8	2020-11-08 00:15:07	<a href="#">Zobrazit</a>	<a href="#">Upravit</a>	<a href="#">Smazat</a>

Obrázek 4.3: Správa senzorů

Rok	Lichá směna
2019	Cahovi
2020	Vaňkovi
2021	Cahovi
2022	Cahovi
2023	Vaňkovi

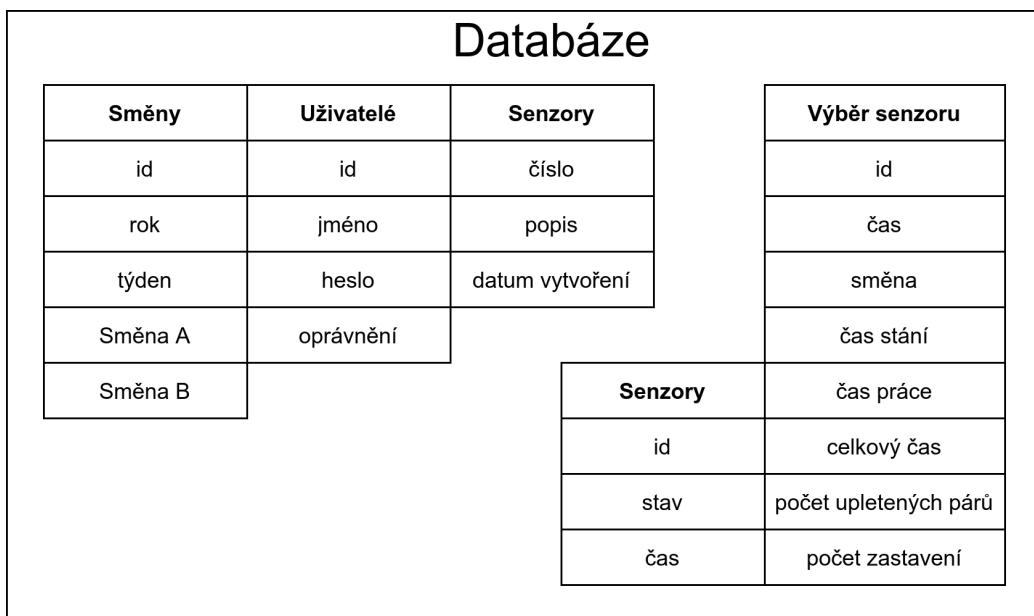
Obrázek 4.4: Nastavení směn

## 4.5 Databáze

Databáze je rozdělená do dvou skupin tabulek.

První skupina tabulek je nastavovací. Jedná se o tabulku s nastavením senzorů, nastavení směn a o tabulkou s uživateli a jejich oprávněním.

Druhá skupina je senzorová. Každý senzor zde má pět tabulek na ukládání svých dat. První senzorová tabulka ukládá čistá nezpracovaná data posílaná přímo ze senzoru. Zbylé čtyři tabulky jsou databázové výběry různých časových úseků. Jde o výběr hodinový, denní, měsíční a roční. Tyto tabulky se vytvářejí automaticky pomocí výběrového API. Struktura tabulek je vyobrazena ve schématu 4.5.



Obrázek 4.5: Struktura databáze

# Kapitola 5

## Podpůrný server

Podpůrný server vznikl jako rozšíření pro senzory. Server je naprogramovaný v Pythonu a běží na Raspberry Pi společně s webovým serverem.

Zdrojový kód na Githubu: [Pletacka-python-server](#)[9]

### 5.1 Kontrola senzorů

Hlavním úkolem tohoto serveru je detekce zapnutých senzorů. Na serveru běží takzvaný Watchdog. Jde o periodickou smyčku, která každé čtyři vteřiny čeká na zprávu ze senzoru. Touto zprávou se senzor nahlásí, že je zapnutý. Pokud takováto zpráva nedojde do deseti vteřin, je senzor prohlášen za vypnutý a v databázi se označí jako neaktivní.

### 5.2 Automatické aktualizace

Bezdrátová aktualizace senzorů je nová funkcionalita, kterou nadále vyvíjím a rozšiřuji. Senzory podporují rychlou aktualizaci přes WiFi ze vzdáleného počítače. V počítači stačí vybrat číslo senzoru a nová verze programu se pomocí WiFi připojení nahraje do senzoru.

# Kapitola 6

## Princip fungování Pletačka IoT

V předchozích kapitolách byly popsány části systému Pletačka IoT. V této kapitole bude celý systém popsán jako celek.

### 6.1 Sběr dat

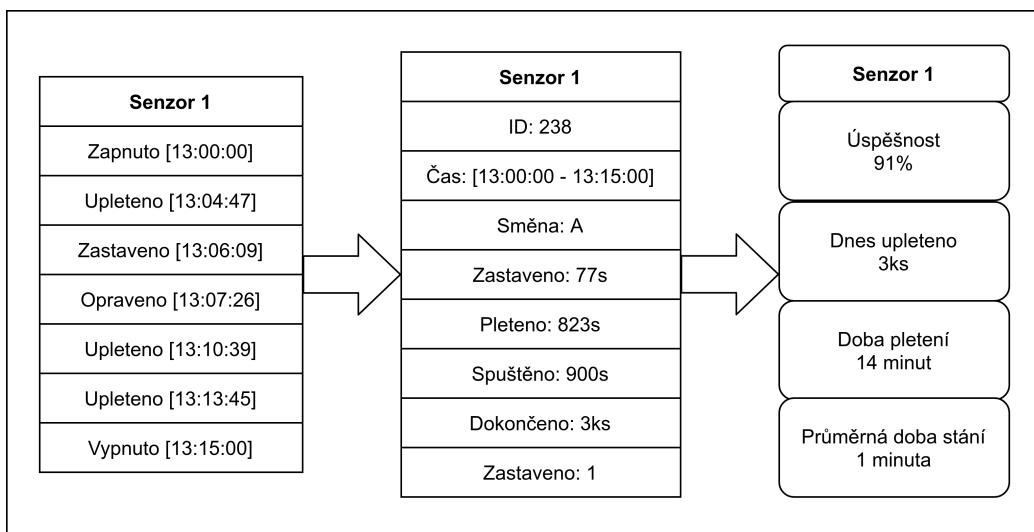
První, a tou nejdůležitější částí, je získávání dat pomocí senzorů. Jakmile senzor zaznamená jakoukoliv změnu, okamžitě tuto zprávu odesílá na server. Odesílání probíhá skrze senzorové API, kde se nejdříve senzor ověří a následně se stav zapíše do databáze k příslušnému senzoru. Po zapsání do databáze se vrátí do senzoru zpráva o provedení zápisu.

### 6.2 Vyhodnocování dat

Dalším krokem je zpracovávání surových dat z databáze. K tomuto účelu běží na serveru výběrové API, které je automaticky spouštěné v nastavený čas. Jde o generování širších výběrů dat, hodinové, denní, měsíční a roční výběry. Tyto výběry se následně ukládají do databáze k danému senzoru. Generování těchto dat probíhá převážně v noci, kdy je server nejméně vytížen.

## 6.3 Zobrazování dat

Posledním krokem je zobrazení dat uživateli. Je to jediná část, se kterou se běžný uživatel dostane do kontaktu. Proto je nutné, aby zobrazení bylo co nejrychlejší a pro uživatele co nejpříjemnější. K rychlému zobrazování se využívají před generované výběry, ke kterým se dopočítají dosud nezpracovaná data a celý výsledek se zobrazí uživateli.



Obrázek 6.1: Zpracování dat

## 6.4 Konektivita

Webové stránky se dají jednoduše zobrazit na počítači či notebooku. Stránky jsou responzivní a lze je používat i na mobilních zařízeních. Přístup k webu je pouze z vnitřní sítě firmy, což zajišťuje základní bezpečnost pro systém.

# Kapitola 7

## Vývoj

Na této práci jsem začal pracovat v únoru 2020, kdy jsem si jako úplný nováček četl dokumentaci k jazyku PHP. Původní verzi webového rozhraní jsem začal navrhovat v čistém PHP. Tento způsob byl však velmi zdlouhavý a neefektivní. Po měsíci práce v čistém PHP jsem přešel na framework Nette, který mi práci zjednodušil a posunul mě velmi rychle dál.

### 7.1 Systém Pletačka IoT verze 1.0

Tato verze byla vydána začátkem července 2020, kdy už systém uměl pracovat s virtuálními senzory.

#### 7.1.1 Senzory

Souběžně s programováním webu jsem pracoval na softwaru pro senzory. V této době byly senzory schopné posílat data na server, ale neměly žádný grafický výstup ani nepodporovaly interakci s uživatelem.

### **7.1.2 Web**

Vznikla základní kostra webu a postupně vznikaly první stránky. Data ze senzorů se pouze ukládala do databáze, ale web s nimi zatím neuměl pracovat. Začínal se vyvíjet systém na zpracovávání údajů ze senzorů.

## **7.2 Systém Pletačka IoT verze 2.0**

Druhá verze přinesla velké rozšíření systému. Tato verze je produkčně nasazena od půlky prosince 2020 a do teď běží bez větších problémů.

### **7.2.1 Senzory**

Propojení systému s novou verzí senzorů, které nově podporují nahrávání aktualizací přes WiFi, mají přehlednější zobrazování dat na displej a dokážou upozornit na výpadek sítě.

### **7.2.2 Web**

Největší proměnou prošlo webové rozhraní. Domovská stránka má přehledné zobrazování stavů senzorů. U senzorů se zobrazují důležitá data a pomocí grafů se dají data jednoduše porovnávat. Přibylo nastavování směn a hromadné přidávání senzorů.

# Kapitola 8

## Testování

Testování systému je jedna z nejdůležitějších částí navrhování jakýchkoliv systémů. Správným otestováním by se měla odladit většina potenciálních chyb.

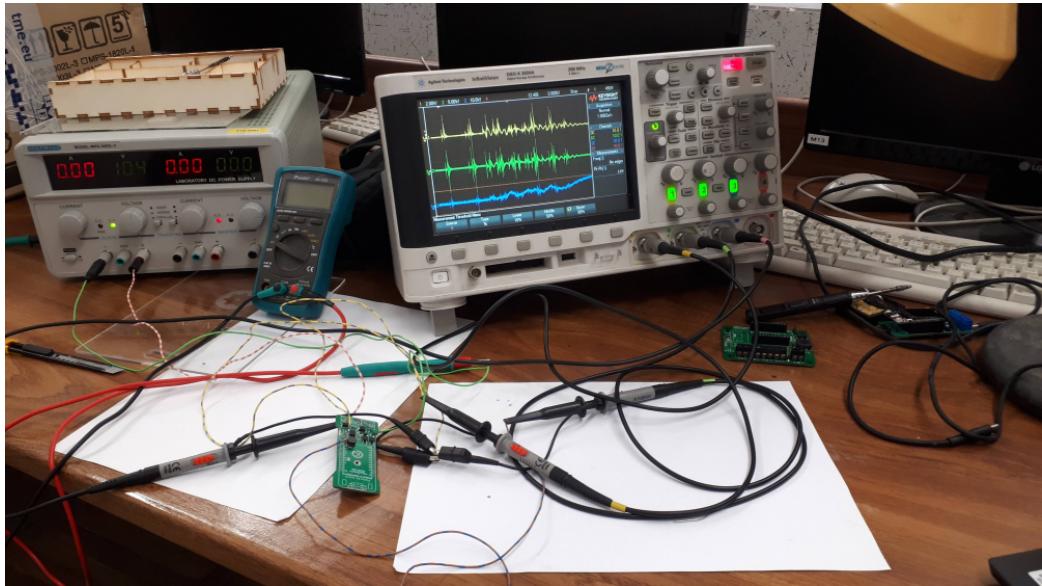
### 8.1 Domácí testování

Průběžné testování částí webu probíhalo již při vývoji a kontrolovalo správné fungování nových funkcí.

Později bylo nutné nachystat rozsáhlejší testy a připravit jim testovací databázi s fiktivními daty. Tímto způsobem jsem například kontroloval správnost běhu funkce pro výpočet času zastavení stroje.

### 8.2 Testování ve firmě

V květnu roku 2020, kdy byly odladěny chyby, jsem systém Pletačka IoT nasadil na dva pletací stroje. Nově nasbíraná data byla již reálná a dalo se na nich postavit nové testování. Senzory jsem nechal několik dní sbírat údaje o upletených ponožkách a následně jsem nad nimi spustil generování uživatelsky čitelných dat.



Obrázek 8.1: Testování senzoru

Nasazení dalších senzorů proběhlo koncem září, kdy byly osazeny další dva pletací stroje. Byly v provozu čtyři senzory a probíhal vývoj nových. V půlce prosince jsem připravil dalších šest senzorů a zahájil dlouhodobé testování bez zásahu do vygenerovaných dat. Naměřené údaje pravidelně stahuji a kontroluji jejich správnost.

# Kapitola 9

## Zpětná vazba majitele firmy

Systém Pletačka IoT funguje v naší firmě ROTEX Vysočina již 3 měsíce a pomáhá při každodenním provozu analyzovat chod strojů. Díky sběru dat on-line z pletacího stroje je vše velmi rychlé a efektivní. Hlavním přínosem tohoto systému je jednoduché porovnávání pracovních směn, což nám umožňuje rychle analyzovat průběh výroby u každého stroje. Systém také využíváme k vytváření výrobních statistik a ke kontrole poruchovosti strojů.

Za firmu Karel Krejčí

# Závěr

Cílem této práce bylo navrhnut ucelený systém, který dokáže:

- automaticky počítat upletené ponožky
- on-line hlásit poruchu na stroji a zjišťovat celkovou poruchovost strojů
- porovnávat výkonnost jednotlivých pracovních směn
- monitorovat průběh výroby
- nahradí část monotónní práce operátora
- zrychlí a zefektivní výrobu
- sníží chybovost

Všechny tyto vytýčené cíle se mi podařilo splnit. Systém nadále běží ve firmě ROTEX Vysočina s.r.o [1] a pomáhá v běžném provozu. Můj systém se stal nedílnou součástí výrobního procesu a analyzuje a zefektivňuje průběh výroby.

Systém je k 1. únoru 2021 nasazen na deseti pletacích strojích a po dobu provozu zaznamenal již přes padesát tisíc upletených ponožek bez závady na senzorech.

Velkým přínosem pro firmu je porovnávání pracovních směn, díky kterým zaměstnavatel ihned vidí rozdíly mezi produktivitou práce v daném čase.

Díky SOČ jsem se naučil navrhovat plošné spoje, rozšířil jsem si obzory v elektronice a při vývoji jsem si vyzkoušel práci s měřícími přístroji. Také jsem se naučil programovat v jazyce PHP a vytvářet komplexní webové systémy.

V budoucnu bych chtěl tento systém rozšířit na všechny pletací stroje a pokrýt tak celou výrobu. Taktéž pokračuji na vylepšování webové aplikace a plánuji ji rozšířit o další funkce. Jde například o export dat do tabulek.

Tuto práci můžete najít na adrese: <https://github.com/JakubAndrysek/SOC-Integrace-do-prumyslu-4.0/blob/master/text.pdf>.

Všechny zdrojové kódy a DPS k projektu jsou k dispozici na <https://github.com/Pletacka-IoT> pod MIT licencí.

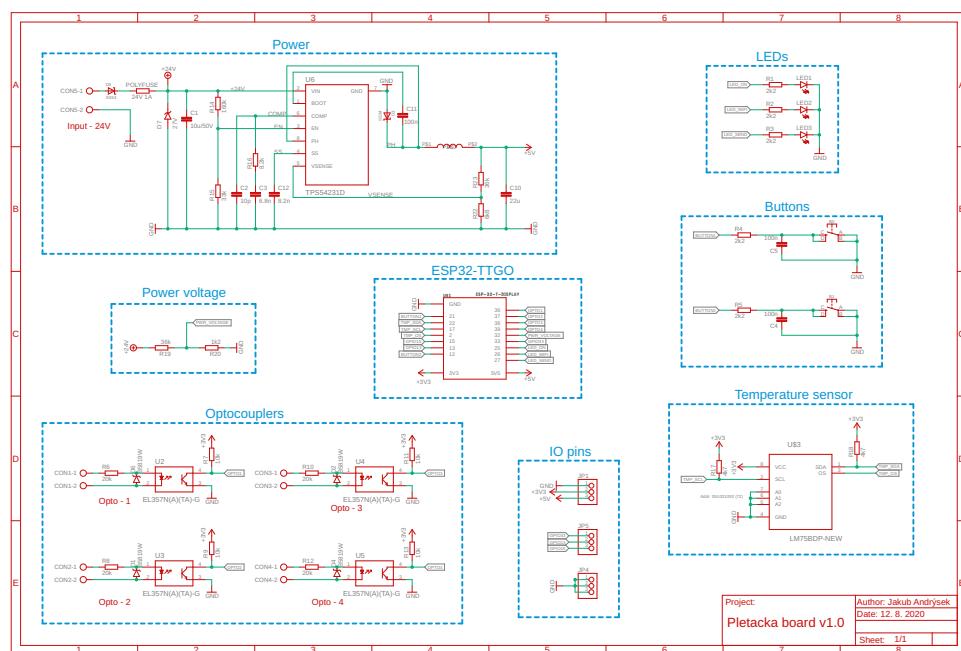


Obrázek 9.1: Senzory

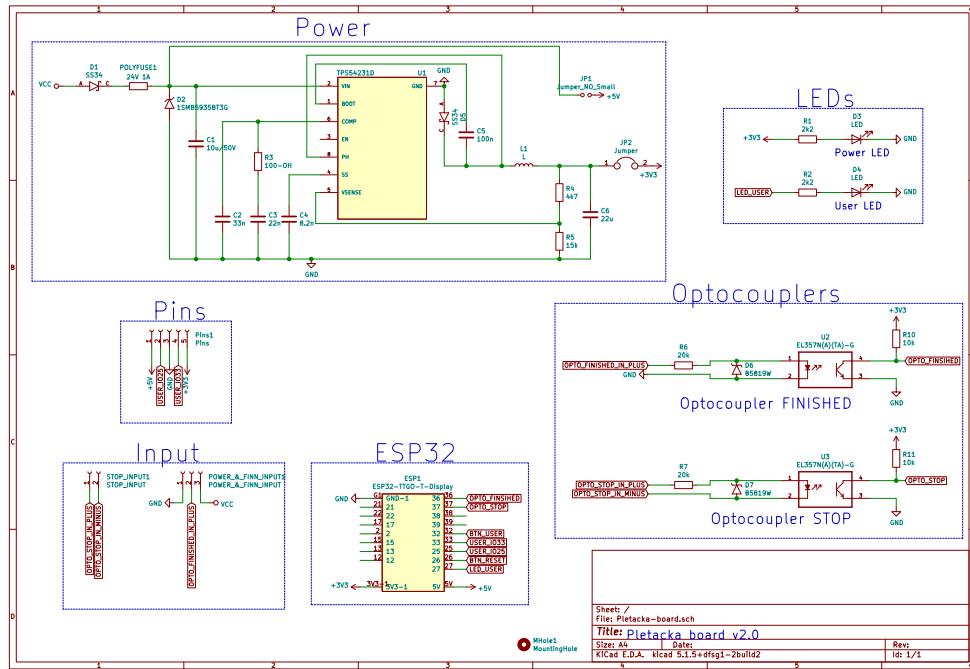
# Literatura

1. ROTEX VYSOČINA S.R.O. *Webové stránky firmy ROTEX Vysočina* [online] [cit. 2020-12-08]. Dostupné z: <https://www.rotexvysocina.cz>.
2. ARDUINO. *Webové stránky Arduino* [online] [cit. 2020-11-15]. Dostupné z: <https://www.arduino.cc>.
3. HARDWARIO. *Webové stránky Hardwario* [online] [cit. 2020-11-15]. Dostupné z: <https://www.hardwario.com>.
4. CONTROLLINO. *Webové stránky Controllino* [online] [cit. 2020-11-15]. Dostupné z: <https://www.controllino.com>.
5. JAKUB ANDRÝSEK. *Návrh a zdrojový kód senzorů* [online] [cit. 2020-12-08]. Dostupné z: <https://github.com/Pletacka-IoT/Pletacka-board>.
6. JAKUB ANDRÝSEK. *Zdrojový kód webového serveru* [online] [cit. 2020-12-08]. Dostupné z: <https://github.com/Pletacka-IoT/Pletacka-website>.
7. JIŘÍ PUDIL. *Webové stránky knihovny Naja* [online] [cit. 2020-12-01]. Dostupné z: <https://naja.js.org>.
8. DAVID GRUDL. *Webové stránky frameworku Nette* [online] [cit. 2020-12-01]. Dostupné z: <https://nette.org>.
9. JAKUB ANDRÝSEK. *Zdrojový kód podpůrného serveru* [online] [cit. 2020-12-08]. Dostupné z: <https://github.com/Pletacka-IoT/Pletacka-python-server>.

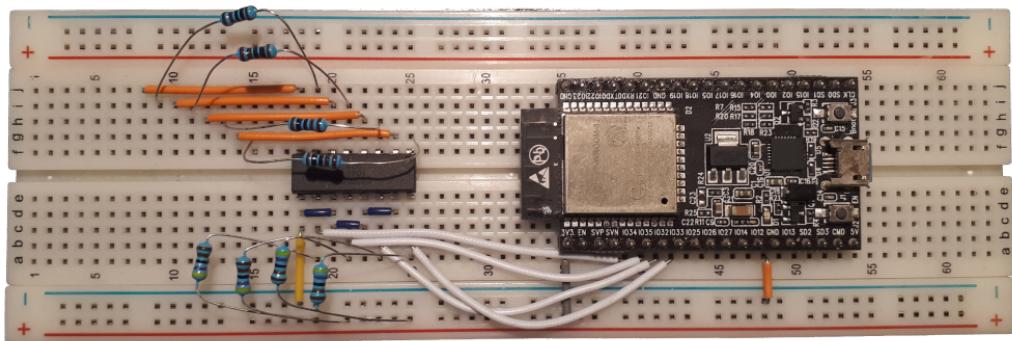
# Přílohy



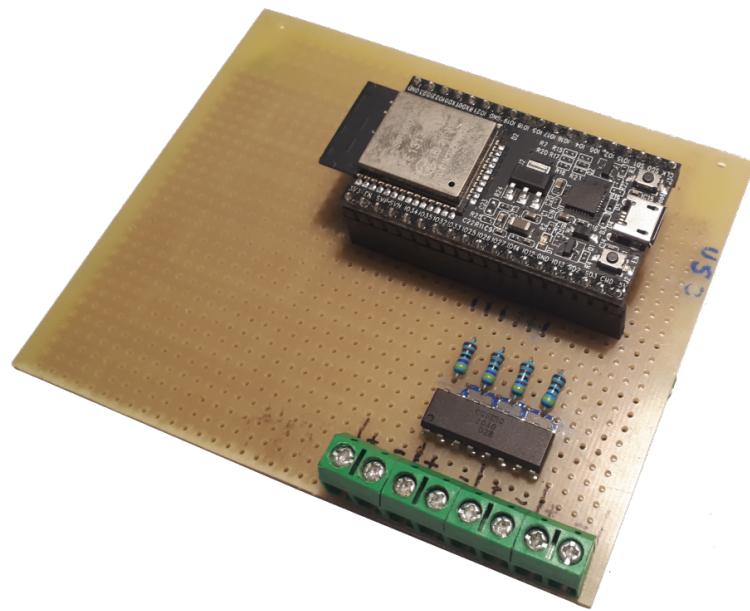
Obrázek 2: Schéma senzoru 1. verze



Obrázek 3: Schéma senzoru 2. verze



Obrázek 4: Testování funkčnosti zapojení



Obrázek 5: Testovací verze senzoru



Obrázek 6: Webový server Raspberry Pi



Obrázek 7: Pletací stroj