

# **System pro řízení a monitorování kurníku**

Antonín Světinský

---

Bakalářská práce  
2020



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Antonín Světinský**  
Osobní číslo: **A17633**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Softwarové inženýrství**  
Forma studia: **Prezenční**  
Téma práce: **Systém pro řízení a monitorování kurníku**  
Téma práce anglicky: **A Henhouse Control and Monitoring System**

### Zásady pro vypracování

1. Popište existující systémy pro monitorování a řízení chovu hospodářských zvířat.
2. Navrhněte vlastní systém pro řízení a monitorování kurníku.
3. Zvolte snímače nutné pro získávání informací o požadovaných veličinách a prvky potřebné k ovládní výstupů.
4. Uvedený návrh hardwarově realizujte.
5. Implementujte software pro řídicí mikro počítač.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. BENEŠ, Pavel. Automatizace a automatizační technika: prostředky automatizační techniky. 5., rozš. a aktualiz. vyd. Brno: Computer Press, 2014. ISBN 9788025137475.
2. CATSOULIS, John. Designing embedded hardware. 2nd ed. Sebastopol, CA: O'Reilly, 2005, xvi, 377 p. ISBN 0596007558.
3. LADMAN, Josef. Elektronické konstrukce pro začátečníky. Praha: BEN – technická literatura, 2001. ISBN 80-730-0015-6.
4. MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, 2012, xx, 699 p. ISBN 1449313876.
5. PINKER, Jiří. Mikroprocesory a mikro počítače. Praha: BEN – technická literatura, 2004. ISBN 80-7300-110-1.

Vedoucí bakalářské práce:

**Ing. Jan Dolinay, Ph.D.**

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: 28. listopadu 2019  
Termín odevzdání bakalářské práce: 15. května 2020



---

doc. Mgr. Milan Adámek, Ph.D.  
děkan

---

prof. Mgr. Roman Jašek, Ph.D.  
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

**Jméno, příjmení: Antonín Světinský**

**Název bakalářské práce: Systém pro řízení a monitorování kurníku**

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 10. 8. 2020

Antonín Světinský v. r.

## **ABSTRAKT**

Tato Bakalářské práce se zabývá vytvořením systému pro řízení a monitorování kurníku. V teoretické části se zabývám analýzou současných řešení pro řízení a monitorování hospodářských zvířat. Dále se zabývám analýzou současných řešení, zvláště pak chovu slepic. V praktické části se pak zabývám požadavky, které jsou potřebné pro správný chod kurníku, analyzuji vhodné senzory a řídicí prvky. Na základě těchto požadavků a inspirací z jiných oblastí chovu bylo navrženo řešení kurníku. Pro monitorování pohybu slepic slouží RFID čipy a jako řídicí jednotka je využita mikropočítačová vývojová deska ESP32. Výsledkem práce je systém, který zajišťuje automatické otvírání a zavírání dvířek v nastavený čas, kontrolu počtu slepic v kurníku, detekci neoprávněného vstupu, zabezpečení proti škodné zvěři a počítání vajíček jednotlivých slepic.

Klíčová slova: mikropočítač, automatizace, kurník, ESP32, Arduino

## **ABSTRACT**

This bachelor thesis deals with the creation of a system for controlling and monitoring the chicken coop. In the theoretical part I deal with the analysis of current solutions for management and monitoring of livestock. I also deal with the analysis of current solutions, especially hen breeding. In the practical part, I deal with the requirements that are necessary for the proper operation of the chicken coop, analyze the appropriate sensors and controls. Based on these requirements and inspirations from other areas of breeding, a chicken coop solution was designed. RFID chips are used to monitor the movement of hens and the ESP32 microcomputer development board is used as a control unit. The result of the work is a system that ensures automatic opening and closing of the door at a set time, control of the number of hens in the chicken coop, protection against harmful game and counting the eggs of individual hens.

Keywords: microcomputer, automation, chicken coop, ESP32, Arduino

Chtěl bych zde velmi poděkovat své rodině za podporu během studia a dále vedoucímu mé bakalářské práce panu Ing. Janu Dolinayovi, Ph.D. za trpělivost, vstřícné jednání a odborné konzultace během vytváření této práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>1</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>3</b>
<b>1 SYSTÉMY PRO MONITOROVÁNÍ A ŘÍZENÍ CHOVU HOSPODÁŘSKÝCH ZVÍŘAT</b> .....	<b>4</b>
1.2 POTŘEBY ZVÍŘAT.....	4
1.3 POTŘEBY Z HLEDISKA KONKRÉTNÍHO TYPU ZVÍŘETE.....	4
1.4 EXISTUJÍCÍ ŘEŠENÍ PRO CHOV SLEPIC V DOMÁCÍCH PODMÍNKÁCH .....	6
<b>2 INTERNET VĚCÍ</b> .....	<b>9</b>
2.1 DOMÁCNOSTI .....	9
2.2 PRŮMYSL .....	9
2.3 ZÁKLADNÍ STAVEBNÍ PRVKY IOT .....	10
2.4 PERIFERIE.....	12
<b>II PRAKTICKÁ ČÁST</b> .....	<b>14</b>
<b>3 NÁVRH SYSTÉMU PRO ŘÍZENÍ A MONITOROVÁNÍ KURNÍKU</b> .....	<b>15</b>
3.1 NEDOSTATKY KONKURENČNÍCH ŘEŠENÍ.....	15
3.2 ZÁKLADNÍ POŽADAVKY NA SYSTÉM MONITOROVÁNÍ A ŘÍZENÍ KURNÍKU .....	15
3.3 REALIZACE ŘEŠENÍ.....	17
<b>ZÁVĚR</b> .....	<b>31</b>
<b>SEZNAM ZKRATEK</b> .....	<b>34</b>
<b>SEZNAM OBRÁZKŮ</b> .....	<b>35</b>
<b>SEZNAM TABULEK</b> .....	<b>36</b>
<b>SEZNAM PŘÍL</b> .....	<b>37</b>

## ÚVOD

Slepice jsou hospodářská zvířata, která jsou z pohledu chovu velmi nenáročná a přináší spoustu výhod, jako jsou čerstvá domácí vejce, chutné maso a neopomíjenou komoditou je pak i slepičí trus, které slouží jako cenné hnojivo [1]. Další výhodou může být i dobrý pocit, že jsme taky přiložili ruku k lepší ochraně krajiny a nejen nadávali na zemědělce, jak hospodaří, aniž bychom proto sami něco neudělali.

Dnes musí každý, kdo chová slepice a nemá to nějakým způsobem již vymyšlené, každé ráno vstát a kurník otevřít a večer ho jít zase zavřít. Existují sice řešení, které dokáží automaticky otevírat a zavírat dvířka kurníku, ale ve většině řešení se nedostává zpětné vazby uživateli, jestli nezůstala nějaká slepice venku i po zavření dvířek. Chovatel si tak nemůže být jistý zabezpečením slepic. Pro chovatele je dobré vědět i to, kolik vajíček snáší jednotlivé slepice. Přitom z hlediska dnešních technologií se jedná o velmi primitivní úkony, které lze pomocí technologií monitorovat a mnohé činnosti automatizovat.

Podle Organizace OSN pro výživu a zemědělství se přibližně 30 % veškerého jídla po celém světě vyhodí anebo nějakým způsobem znehodnotí [2]. Nevyužité potraviny a zbytky z domácností mohou mnohdy slepice spotřebovat, a navíc mohou vyprodukovat pro půdu organickou hmotu, která úzce souvisí s úrodností a půdními vlastnostmi. [3].

Slepice je možné mít za použití zabezpečovacích technologií dokonce i na loukách, polích a volných plochách, kde by nikoho neobtěžoval případný hluk a zápach slepic, který může být předmětem sousedských sporů. Slepice tak nemusí narušovat pohodlí v blízkosti domovů, a navíc částečně odpadá i starost s uklízením kuřinců.

Toto řešení má mít i edukativní charakter, který má lidi naučit se lépe starat o svoji krajinu, zpracovávat své zbytky z domácností a malé děti vychovávat k lásce k přírodě. Navíc díky automatizaci mohou odpadnout některé ne příliš příjemné činnosti a může to být i dobrá relaxace.

Myslím si, že dnešní technologie pokročily tak, že činnosti, které se zdály být dříve nepředstavitelné se dají v dnešní době pomocí moderních technologií, které se stávají čím dál více přesnějších, rychlejších, ale i dostupnějších, automatizovat. Dostáváme tak velmi silné nástroje, který dokáží řešit velmi sofistikované problémy. Mnoho úkonů lze tedy velmi jednoduše nechat na strojích, které mohou běžet 24 h denně a neunaví se.



Pro řešení jsem zvolil mikropočítač ESP32, který obsahuje dostatečný výpočetní výkon, velké množství pinů, paměti a požadovaných sběrnic. Dále je součástí tohoto mikropočítače i Wifi modul, Bluetooth a obsahuje technologie pro řízení spotřeby, které jsou pro řešení napájené z baterií taky velmi důležité. Dostáváme tak miniaturní zařízení, které dokáže obsloužit velmi velké množství požadavků, aniž bychom museli připojovat mnoho další periférií.

Pro systém počítání jsem využil technologie RFID (rádiově-frekvenční identifikace), která dokáže pomocí čtecích zařízení vzdáleně snímat hodnoty RFID čipů. Pokud tedy každou slepici tedy opatříme tímto čipem, který umístíme na nohy slepic, aby mohla být čtecí vzdálenost mezi čipem a čtecím zařízením co nejmenší, tak můžeme velice snadno slepici identifikovat.

Cílem této práce je vytvořit systém, který dokáže s minimálním úsilím chovat vlastní slepice, za dostupné ceny a s minimální počáteční investicí. Navržené řešení je rozděleno do několika samostatných modulů – automatické dvířka, RFID brána pro kontrolu počtu slepic v kurníku, zabezpečení proti škodné zvěři a snáška pro počítání vajíček jednotlivých slepic.

V případě, že dojde k nějakým nestandardním situacím, které systém nedokáže sám vyřešit, tak je chovatel upozorněn na mobilní telefon. Podle závažnosti pak může uživatel tuto komplikaci vyřešit vzdáleně anebo bude muset přijet osobně a situaci vyřešit.

Cílovou skupinou tohoto systému budou pravděpodobně lidi na vesnicích a chataři, a dále pak lidi, kteří se chtějí podílet na udržování krajiny, minimalizovat odpad v domácnosti anebo chtějí být více potravinové soběstační.

V teoretické části jsem se nejprve zabýval již vytvořenými řešeními, které jsou využívána ve velkochovech drůbeže, skotu, prasat a dalších hospodářských zvířat. Dále jsem se zabýval technologiemi vhodnými řízení a monitorování kurníku.

V praktické části jsem navrhl seznam požadavků, a to především na co největší usnadnění, zabezpečení, ale i to, aby řešení bylo i cenově dostupné.

## **I. TEORETICKÁ ČÁST**

# 1 SYSTÉMY PRO MONITOROVÁNÍ A ŘÍZENÍ CHOVU HOSPODÁŘSKÝCH ZVÍŘAT

Zemědělství zažívá v posledních letech jak v rostlinné, tak i v živočišné výrobě technologický boom. Mnoho činností se zde automatizuje a pomocí různých senzorů se získává velké množství informací, které zajistit mnohem lepší efektivitu. Údaje lze sledovat i v reálném čase, dlouhodobě ukládat a vyhodnocovat.

Zemědělství není jen o tom umět něco vypěstovat anebo chovat, ale jelikož mezi hlavní rozhodovací faktory patří i cena, tak musí být brát zřetel i na co nejlepší produktivitu a celosvětový konkurenční boj.

Technologie dokáží zvýšit jak bezpečnost zvířat, tak i pracovníků. Tím, že se dbá na pohodlí zvířat se nepřímo ukazuje na kvalitě a výtěžnosti zvířat [4].

## 1.1 Co má smysl hlídat?

Člověk nemusí být zvěrolékař, aby mu bylo jasné, že různá zvířata mají ze své podstaty různé potřeby. Každému zvířeti je potřeba zajistit co možná nejvíce přirozené podmínky.

Hospodářská zvířata mohou být ve stájích anebo být ve volné přírodě. Z toho pak vyplývají potřebné technologie, kterými zvíře musíme vybavit.

## 1.2 Potřeby zvířat

Všichni biologičtí tvorové mají základní potřeby, které je potřeba dodržet. Je to především přísun vody, jídla a přirozené podmínky pro žití, které se liší podle druhu zvířat.

Různá zvířata potřebují ze své podstaty různou stravu, prostředí, výkyvy teplot, míru čistoty prostředí a vzduchu, vlhkosti a dalších parametrů. Každé zvíře potřebuje i různou míru „smyslu života“. Je jim potřeba tedy dát i prvky, které jim nasimulují přirozené prostředí, které ale může být i na bázi přírodního charakteru.

## 1.3 Potřeby z hlediska konkrétního typu zvířete

V této části uvádím příklady řešení pro určité typy zvířat.

### 1.3.1 Potřeby drůbeže

Existuje velká řada technologií, které dokáží usnadnit chov drůbeže jako jsou různí roboti, kteří dokáží vydezinfikovat prostředí, provzdušňovat podestýlku, řídit osvětlení, dodávky krmení a hlídat bezpečnost. [5]

### ***1.3.1.1 Hlídaní a přísun stravy***

Ve velkochovech je strava míchaná pomocí automatického míchače, který podle nastaveného poměru dokáže připravit vyváženou stravu. Sypké krmivo pro slepice se dá snadno přenášet pomocí potrubí až k miskám, ze kterých slepice zobou. [5]

### ***1.3.1.2 Rozeznávání uhynulých zvířat a dravců***

Pomocí kamer, termokamer a strojového je možné velmi rychle detekovat uhynulou drůbež, detekovat dravce a upozornit na to chovatele. Systém tak pomáhá lépe chránit slepice a rychle zamezit i případnému šíření nálezů, a tedy i případným ztrátám. [6]

## **1.3.2 Krávy**

### ***1.3.2.1 Pedometry***

Měření pohybové aktivity se stal v poslední době velký boom. Pomocí mobilních telefonů, které obsahují velmi sofistikované akcelerometry a chytrému vyhodnocování údajů lze přesně určit o jaký pohyb se jedná. Specifické opakující se pohyby se pak dokáží kategorizovat a můžete vědět, jak dlouho sedíte, ležíte, běháte apod.

Ukázalo se, že pomocí snímání hodnot z akcelerometrů lze vyčíslit, jak dlouho kráva leží, stojí, kolik udělá přežvyků a mnoho dalších činností. Tyto informace pak mohou sloužit k hlídání zdraví zvířete a ke vhodnému stanovování vhodné doby inseminace. Zařízení může obsahovat i GPS modul, který pak slouží k snadné lokalizaci konkrétního zvířete. [7]

### ***1.3.2.2 Systém dávkování a dopravy stravy***

Hlavní potravou krav ve stájích je především seno, které je mít oddělené od krav přepážkou, aby nedocházelo k jejímu znečištění pošlapáním. Jelikož forma stravy nelze jednoduše distribuovat pomocí trubek, jak je tomu u sypkých materiálů.

Společnost Lely Holding N.V [8], která pomáhá řešit automatizaci hospodářských zvířat to řeší naskladněním balíků na dopravníky automatické pily, který dokáže postupně odřezávat naskladněné balíky. Krmná směs je díky postupnému odkrajování čerstvá. Jakmile automat odřeže potřebné kusy, tak strava putuje do míchače, který rovnoměrně promíchá různé druhy krmiva, aby byly rovnoměrně rozprostřeny v celé

dávce. Vhodně namíchaná strava putuje do dopravníku, který krávám dopraví do stáje. [8]  
[9]

Další oblastí, kterou se společnost zabývá je dojení. Tato činnost lze rovněž automatizovat. Krávy mají kolem krku čip, kterým se identifikují. Kráva je motivovaná chodit k dojícímu stroji díky chutnějším granulím. Systém dokáže sám připnout přísavky na odsávání mléka. Při odběru mléka lze přesně analyzovat zdravotní stav krávy a její kvalitu mléka. Pokud by byla kráva nakažená, tak ji brána při odchodu z dojícího automatu pustí do samostatné ohrady a upozorní zemědělce, aby zkontroloval stav dané krávy. [8]  
[9]

### **1.3.2.3 Kartáče**

Další vítanou pomůckou jsou kartáče, kterým se může kráva podrbat na místech které ji svědí a zbaví se případných roztočů. Nemusí se tak nějak složitě převalovat, aby odehnala mouchy a další havěť. Jakmile kráva přijde pod kartáč, tak se zařízení aktivuje a začne se točit. To opět zlepšuje kvalitu pohodlí zvířat. [8]

## **1.3.3 Prase**

### **1.3.3.1 Simulování získávání potravy**

Prase má potřebu si získávat potravu samo. Systém dávkování pro prase je vhodné uzpůsobit tak, aby zvíře muselo samo vyvinout určitou aktivitu, aby jídlo získalo. Musí tedy čumákem narážet např. do dávkovače, který mu ale nedá celou dávku zároveň, ale postupně mu to dává. Aktivitou jim pak dále třeba nechřadnou kosti anebo jsou i v lepší psychické kondici [10].

## **1.4 Existující řešení pro chov slepic v domácích podmínkách**

Existuje velké množství kutilů a firem, které se snaží chov slepic usnadnit. Ať už jsou to automatické dvířka, které se sami otvírají a zavírají, tak i různé dávkovače krmení.

### **1.4.1 Systémy pro ovládání dvířek**

Existuje velké množství způsobů, jak se dají dvířka otevírat. Dvířka mohou být poloautomatická a pracovat čistě na mechanickém principu a nebo řízená pomocí elektronických prvků.

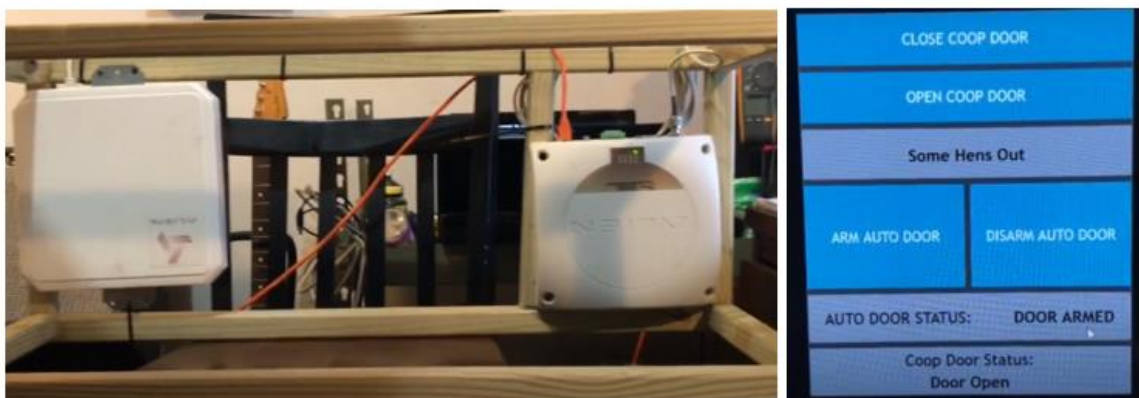
Dvířka pracující na mechanickém principu se při sešlápnutí dvířek zevnitř přes speciální zarážku otevřou. Večer ale musí chovatel slepice zavřít a zarážku zastrčit [11].

Existují i profesionální komerční řešení, které je možné si nastavit vlastní časy otevírání a zavírání anebo nechat řídit otevírání a zavírání na základě soumrakového senzoru, který vyhodnocuje světelné podmínky. Toto zařízení obsahuje motor s navijákem, který namotáváním a odmotáváním provázku pohybuje dvířky. Zařízení může být napájeno jak z baterie, tak z elektrické sítě [12].



Obrázek 1: Ukázka automatických dvířek společnosti Chickenguard [12]

Na internetu existuje i řešení, které dokáže počítat slepice. Z uveřejněného videa lze vyzpozorovat, že systém funguje přes 2 velké RFID čtecí zařízení, které zaznamenává RFID čipy umístěných na slepicích. Pomocí aplikace lze sledovat stavy konkrétních slepic a ovládat dvířka na dálku.



Obrázek 2: Systém pro počítání slepic pomoc RFID čtecích zařízení [13]

Společnost GoGo Chicken rozpoznává slepice tak, že umístí gps moduly na nohy slepic a sleduje i jejich pohybovou aktivitu. Zákazník si může najít svoji přeoobjednanou slepici a sledovat její aktivitu. [14]

#### 1.4.1.1 Způsoby otevírání a zavírání dvířek

Na obrázcích níže je uvedeno několik způsobů možného mechanického a elektronického řešení otevírání dvířek. První řešení spočívá v namotávání provázku pomocí motoru na naviják. Dále pak existují řešení pomocí pístu anebo otáčení celé šroubovice. Řešení se jistě liší různou cenou, životností a spolehlivostí [15] [16] [17] .



Obrázek 3: Různé způsoby otevírání dvířek [15] [16] [17]

#### 1.4.2 Systémy pro zabezpečení proti škodné zvěři

Společnost PredatorGuard [18] řeší zabezpečení chovu slepic pomocí detekce pohybu a následných světelných záblesků. Na obrázcích níže můžete vidět, že řešení je energeticky soběstačné díky solárnímu panelu. Snímače pohybu jsou pod zkosenou hranou, aby systém fungoval i při zasněžení vrchní části [18].



Obrázek 4: Ukázka komerčního řešení k ochraně slepic [18]

## 2 INTERNET VĚCÍ

Internet věcí je označení pro síť fyzických zařízení, která jsou vybavena elektronikou, softwarem, senzory, pohyblivými částmi a síťovou konektivitou, která umožňuje těmto zařízením se propojit a vyměňovat si data. [19]

Experti odhadují, že do roku 2022 cena za domácí chytré zařízení dosáhne hodnoty více než 53 miliard dolarů. [20]

Tím, že mohou nejrůznější zařízení mezi sebou komunikovat, tak je možné mnoho činností lépe mezi sebou provázat.

### 2.1 Domácnosti

Internet věcí můžeme začít vidět v domácnostech, kde nám mohou pomoci s lepší kvalitou života, zabezpečením a automatizací nejrůznějších úkonů.

Příkladem může být chytrá domácnost, kde si můžeme např. nastavit budík na určitý čas a tím se vyvolá změna i na boileru, který zajistí, že se voda začne zahřívat tak, abychom se po probuzení osprchovali teplou vodou a nemuseli tak celou dobu udržovat zařízení zapnuté. Dále se nám mohou po probuzení spustit písničky, rozsvítit světla, vytáhnout rolety a spustit kávovar. Budík může být rovněž propojený se situací na silnicích a pokud zjistí, že bychom do práce nestihli přijít včas, tak nás může vzbudit dříve. Cestou do práce pak systém pohlídá, abychom například nezapomněli vypnout vařič, nechat odemknuté dveře a podobně. Cestou do práce může časem auto komunikovat s prvky na silnicích, se semaforey, ostatními auty. Dalšími zařízeními, které můžeme připojit do sítě mohou být např. lednička, pračka, kávovar, vysavač, trouba, osvětlení, okna, dveře a mnoho dalších zařízení [21].

### 2.2 Průmysl

Velké využití má technologie IoT i v průmyslu, kde se rozšířil pojem průmysl 4.0. Velký rozmach zajistilo především to, že se v poslední době spojilo velké množství různých technologií, které mají synergetický efekt. Jedná se především o čím dál lepší cenovou dostupnost elektroniky, vytvoření komunikačních technologií, strojového učení, cloudových služeb a spoustu dalších technologií.

Díky získávání velkého množství dat a technologií, které dokáží data uchovat a pracovat s nimi, tak lze např. volit lepší výrobní postupy, předvídat vývoj, předcházet haváriím a automatizovat výrobu.



## 2.3 Základní stavební prvky IoT

### 2.3.1 Mikropočítače

Mikropočítač je integrované zařízení, které obsahuje procesor a různé sběrnice ke kterým lze připojit periferie, díky kterým můžeme se zařízením komunikovat. Existuje velké množství zařízení, které se liší výkonem, počtem periférií, vstupně výstupními piny. [22]

V posledních letech se tyto zařízení stali díky projektu Arduino dostupnými i pro běžné lidi. Vznikla velká komunita lidí, kteří si s těmito zařízeními hrají a vytváří různé projekty. Je možné si za velmi malé peníze vytvořit něco funkčního. Původně byl projekt Arduino vymyšlen k rychlému prototypování pro studenty k výuce programovacích jazyků a jelikož výsledky práce můžete ihned vidět, tak je to pro začínající programátory motivující. Projekt Arduino nalézá uplatnění i v průmyslové sféře, jelikož lze velice rychle vytvořit prototyp na kterém se může odprezentovat řešení [23].

### 2.3.2 Arduino

Arduino je hardwarový a softwarový open-source ekosystém založený na mikropočítačích. Pod tímto pojmem si lze představit vývojové prostředí, programovací jazyk anebo různé druhy mikropočítačů, které se liší svými parametry jako je množství pinů, výkonem procesoru, velikostí paměti, sběrnicemi a řadou dalších parametrů. K mikrořadičům lze připojit různé periferie jako jsou např. senzory pohybu, teploty, plynů. [23]

Díky své jednoduchosti a rychlému vytvoření prototypu se projekt velmi rychle rozšířil po celém světě. Arduino se pak stalo pro výrobce IoT zařízení číslo jedna a předpokládá se, že do roku do roku 2021 prodá zařízení za 6 miliard dolarů. [23]

Díky své dostupnosti se velmi rychle rozšířil a vznikla velká komunita vývojářů, která vytvořila velmi velké množství knihoven.

Existují různé vývojové Arduino desky, které se liší velikostí, počtem sběrnic, počtem pinů a dalšími vlastnostmi. Mezi nejznámější patří vývojová deska Arduino UNO.

Zařízení Arduino lze programovat v prostředí i pomocí programovacího jazyka C++.

### 2.3.3 Espressif Systems

Společnost Espressif Systems je nadnárodní polovodičová společnost založená v roce 2008 se sídlem v Šanghaji a pobočkami ve Velké Číně, Singapuru, Indii, České republice a Brazílii. Společnost je složena z inženýrů a vědců z celého světa, zaměřený na vývoj špičkových řešení WiFi a Bluetooth s nízkým výkonem IoT. Mezi populárními produkty

patří zejména série modulů a vývojových desek založených na chipech – esp8266, ESP32 a ESP32-S. Společnost tedy poskytuje zelené, všestranné a nákladově efektivní čipové sady. Filozofií firmy, ke které se zavázali je nabízení IoT řešení, která jsou bezpečná, robustní a energeticky účinná. [24]

### 2.3.3.1 ESP8266

Integrovaný obvod byl uveden na trh v roce 2014. Původně sloužil jako UART-Wifi převodník. Cena byla pouhých 5 dolarů, což oproti konkurenčním čipům zanedbatelná částka. Vývojáři využívali toho, že je zařízení levné a dokáže se připojit k internetu a našli způsoby, jak toto zařízení využít ve svých projektech. Původně byla dokumentace pouze v čínštině, což znesnadňovalo vývoj. [25]

#### Technické parametry

- Zařízení využívá 32bitové jádro mikroprocesoru RISC pracujících na frekvenci 80 MHz
- IEEE 802.11 b / g / n Wi-Fi
- 16 GPIO pinů
- SPI, I2C, I2S a UART sběrnici
- 10bitový ADC
- 32 KiB instrukční RAM
- 32 KiB instrukční RAM RAM
- 80 kB uživatelských dat RAM
- 16 KiB ETS-data RAM

### 2.3.3.2 ESP32

ESP32 je nástupcem populárního obvodu ESP8266. Získal řadu vylepšení. Oproti staršímu čipu obsahuje Bluetooth. Procesor obsahuje dvoujádrový 32-bitový procesor pracující v 80-240 MHz a SRAM paměť. Bylo zvýšeno i množství GPIO pinů s podporou sběrnic SPI, I2C a UART. [26]

Mezi hlavní výhody patří robustní design, který nabízí schopnost spolehlivě fungovat i v extrémních podmínkách, a to v rozmezí od -40 do +125 stupňů Celsia. Zařízení je schopno se samo kalibrovat i při změnách okolních podmínek. [26]

ESP32 modul může fungovat jako samostatný plnohodnotný systém anebo jako podřízené zařízení. Pomocí různých rozhraní může být propojen i s jinými systémy, kterým zajistí připojení k Wifi a Bluetooth. [26]

Zařízení se vyrábí v různých provedeních, které se liší obsahem periférií, jako je kamera, GSM modul atd. Hlavní částí každého zařízení je modul WROOM nebo WROOVER, který zpracovává všechny operace. Tyto moduly pak mohou fungovat i plně samostatně, jelikož obsahují všechny potřebné komponenty nutné k provozu. [26]

Zde jsou uvedeny parametry modulu, které jsou nejdůležitější pro tento projekt.

- dvou-jádrový procesor Xtensa LX6, 240 MHz
- 4 mb
- Wi-Fi dle standardu 802.11 b/g/n, 2.4 GHz
- 34 GPIO pinů
- SPI, I2C, UART

### **2.3.3.3 ESP32 GSM**

Je zařízení, které obsahuje kromě ESP modulu periférii GSM, která je rovněž součástí integrovaného obvodu. Díky tomu je možné bez jakéhokoliv dalšího zařízení komunikovat skrze síť GSM, tedy zasílat zprávy a připojit se k internetu prostřednictvím protokolů HTTP a MQTT.

### **2.3.3.4 ESP32 CAM**

Esp32 CAM je zařízení, které nabízí kromě základního modulu ESP navíc připojení kamery k vývojové desce.

## **2.4 Periferie**

### **2.4.1 RFID čtecí zařízení**

Zařízení funguje na principu rozpoznání objektů na základě rádiových vln.

Počátky technologie můžeme najít ve druhé světové válce, kdy bylo potřeba rozeznat vlastní letadla od letadel protivníka. Na letadla se osadilo zařízení, které po detekci radarovým signálem vyslalo zpět signál a díky tomu se dokázal identifikovat letadlo. Technologie se postupem času vyvíjela a našla uplatnění i ve firmách, kde je potřeba taky identifikovat různé předměty a osoby na dálku [27].

Nyní má technologie velmi široké uplatnění. Můžeme ji najít v kamionech, které komunikují přes zařízení s mytími branami. Můžeme je najít rovněž k přidělování přístupu

do oblastí, lze jimi odemknout dveře místností, budov, identifikovat dobytek, zboží a třeba i platit.

Každý RFID čip obsahuje napájecí obvod, přijímací a vysílací, anténu, mikročip a paměť. Existuje velké množství těchto tagů, které dokáží vysílat od jednoduchých jednobitových informací až po datové bloky, které obsahují větší množství šifrovaných dat. RFID čipy i čtecí zařízení mohou mít různou podobu, velikost a použité technologie, které pak ovlivňují čtecí vzdálenost, spolehlivost a bezpečnost [28].

#### **2.4.1.1 Rozdělení**

RFID technologie se rozděluje na 2 kategorie – aktivní a pasivní.

Pasivní čipy dokáží pracovat s elektrickým polem, kde změnou náboje vygeneruje skrze anténu proud, který je dostatečný k napájení chipu. Signál se pak odrazí s hodnotou, která je uložena na čipu. Tím dochází k identifikaci. Oproti aktivním čipům mají mnohem nižší dosah a menší rychlost snímání. [28]

Aktivní čipy oproti pasivním navíc obsahují baterii, kterou jsou napájeny. Čip tak sám vysílá aktivně informace do okolí. Výhodou tohoto řešení je vyšší čtecí vzdálenost. Toto řešení je ale nákladnější a aby bylo možné používat zařízení i v nepříznivých podmínkách, tak se baterie usazuje do zařízení a celý čip po vybití je potřeba vyměnit. [28]

## **II. PRAKTICKÁ ČÁST**

### 3 NÁVRH SYSTÉMU PRO ŘÍZENÍ A MONITOROVÁNÍ KURNÍKU

V této části se zabývám analýzou požadavků a následně navrhuji řešení.

#### 3.1 Nedostatky konkurenčních řešení

Největší slabinou konkurenčních řešení vidím v tom, že vždy řeší jen dílčí části problému a jednotlivé technologie nejsou mezi sebou provázány komplexně. Na českém trhu jsem nenašel řešení, které by dávalo uživateli informace o tom, kolik je slepic v kurníku a zda nějaká slepice nezůstala venku po zavření dvířek.

Bohužel díky tomuto nedostatku uživatel si nemůže být jistý, jestli některá slepice nezůstala venku. Pro mé řešení, které je vhodné pro chataře anebo lidi, kteří si chtějí třeba jen zajet na dovolenou, tak je to nedostatečné. Proto jsem se rozhodl postavit takové řešení, které bude informovat uživatele v reálném čase a odkudkoliv, díky napojení zařízení k internetu.

Dále jsem nenašel řešení, které by řešilo kontrolu kurníku z hlediska krádeže.

#### 3.2 Základní požadavky na systém monitorování a řízení kurníku

Když jsem si dělal průzkum mezi potencionálními uživateli systému, tak mi vyšlo najevo, že většina populace je velmi zaneprázdněna a že na chov slepic nemá čas. Chtěl jsem tedy vytvořit řešení, které by práci co nejvíce usnadňovalo a stačilo navštívit kurníku jen jednou týdně. Některé postupy je vhodné automatizovat pomocí elektroniky a některé činnosti mohou být z počátku čistě mechanického charakteru, aby výsledné řešení nebylo příliš drahé anebo řešení pomocí elektroniky nemělo přidanou hodnotu.

##### 3.2.1 Sledování stavu slepic v kurníku

Základním prvkem automatizace kurníku je dle mého názoru sledování stavu počtu slepic v kurníku a počtu mimo něj. Díky tomuto řešení se výrazně ušetří množství potřebných návštěv. Aby bylo možné slepice počítat, tak bylo nutné vymyslet řešení, jak je slepice identifikovat.

##### 3.2.2 Sledování výtěžnosti jednotlivých slepic

Další výhodou řešení pomocí RFID technologie je i to, že můžeme detekovat slepici i ve snášecí místnosti. Můžeme tak měřit, jak pravidelně chodí snášet vajíčka, jak dlouho na nich sedí a malou úpravou snášecího hnízda, použitím snáškového hnízda, které má nakloněný

podstavec tak, aby se vajíčka odkutálela, a této vlastnosti využili a vajíčka nasměrovali přes senzor (tlačítko), které připočte slepici vejci. Ze získaných informací můžeme dělat různé statistiky, které nám pomohou v lepším opatření slepic jak z hlediska výtěžnosti, bezpečnosti, tak i kvality života zvířat.

### 3.2.3 Skladování vajíček

Aby člověk opravdu mohl přijít ke kurníku jen jednou za týden, tak je potřebné vyřešit i uskladnění vajíček. Řešení vidím v tom, že se vajíčka odkutálejí samospádovým systémem do tepelně izolovaného boxu, kde případné velké teplotní výkyvy anebo nedostatečnou izolaci lze dochladiť pomocí Peltierova článku, který má sice malou účinnost, ale pro občasné dochlazení může být dostatečný.

### 3.2.4 Předcházení zápachům, hluku a lepší využití nevyužitých ploch, zabezpečení

Lidé kromě toho, že jsou zaneprázdnění, tak jim vadí zápachy a hluk. Proto, aby se nenarušilo pohodlí a předcházelo se sporům se sousedy, tak se nabízí možnost, aby daný chov nebyl v blízkosti obydlí, ale přesunut na pole, kde by již neměl nikomu vadit.

Z osobního pohledu to vidím tak, že si jednou za týden zajdu na pole. Během toho, než okopu plevele, posbírám úrodu, tak slepice vypustím na volný výběh, který ohraničím nastavitelným plotem. Pokud bych vlastnil chatu, tak během svého pobytu bych dopřál slepicím co největší výběh. Během nepřítomnosti uživatele ale momentálně vidím řešení spíše v menším oploceném výběhu, ale asi záleží na uživateli, jak daleko bude mít svůj kurník a jeho možnostech případného vyřešení situace. Kdyby nastávaly situace, že slepice se zapomínají na zahradě a nechce se jim do kurníku, tak by bylo potřeba danou situaci vyřešit osobně.

Rizikem takového chovu je však zabezpečení. Podle mého názoru se díky moderním technologiím a cenové dostupnosti již nejedná o příliš komplikované záležitost a je dost pravděpodobné, že se chov podaří časem plně zabezpečit. Možná se může časem posunout i myšlení lidí, že je lepší, aby si místo vzájemného kradení slepic raději všichni pořídili tento způsob chovu a nebude potřeba řešit až tak moc zabezpečení.

### 3.2.5 Zabezpečení proti škodné zvěři

Dalšími situacemi, které je potřeba řešit je to, že slepice a krmivo jsou lákadlem pro potkany a kuny. I když jsou slepice v kurníku, tak to nezaručuje úplnou bezpečnost, jelikož potkání

a kuny jsou velmi vynalézaví tvorové a dokáží se prokousat pomalu i betonem a je tedy určitě lepší předcházet poškození kurníku.

### 3.3 Realizace řešení

Z analýzy požadavků jsem vybral ty, které jsem vyhodnotil jako nejlepší automatizovat a navrhl moduly, které se budou starat o dané problematiku.

System jsem rozdělil do modulů, které budou mezi sebou umět komunikovat anebo mohou fungovat i samostatně. V modulárního řešení vidím i výhodu, že je možné jednotlivé části vyvíjet samostatně, lépe se testoval a lépe se zařízení uvádělo na trh, jelikož uvedení produktu na trh mohlo brzdit některé dílčí řešení.

Pravděpodobně bude i jednodušší přepínat jednotlivé zařízení do energeticky úsporných režimů a tím neplýtvat energetickými zdroji.

#### 3.3.1 Výběr programovacího jazyka

Pro řešení jsem vybral programovací jazyk Arduino. Líbí se mi, že jednak je možné použít spoustu knihoven, existuje zde velká komunita lidí a díky tomu lze vytvořit prototyp mnohem rychleji a s větší funkcionalitou. Líbí se mi i to, že pokud by mi některé knihovny nevyhovovaly z hlediska výkonu, tak mohu si mohu vytvořit vlastní pomocí programovacího jazyka C++. Jako velkou výhodu spatřuji i to, že je možné programovat i vývojovou desku ESP32.

#### 3.3.2 Výběr vývojového prostředí

Z počátku jsem vyvíjel v Arduino IDE, které nabízelo jednoduché prostředí, které pro programování jednoduchých řešení plně dostačovalo. Bohužel jsem narazil na nevýhod, které znesnadňovali vývoj. Je to zejména problém se správou knihoven, absencí vytvoření adresářové struktury a přenášení projektů mezi různými počítači. Pokud programátor pracuje na více projektech a používá stejné knihovny, tak někdy potřebuje knihovny buď mírně upravit anebo mít i různé verze. Rovněž pokud je software vyvíjen na více zařízeních, tak není tak snadná přenositelnost, jelikož použité knihovny se ukládají globálně do počítače, a ne ke konkrétnímu projektu a mnohdy v tom vzniká velký zmatek.

Musel jsem tedy přejít na pokročilejší vývojové prostředí PlatformIO, které se dohrálo jako plugin do vývojového prostředí Visual Studio Code. PlatformIO umožňuje instalovat knihovny konkrétním projektům, a tedy je jasné, které knihovny a jaké verze programátor



používá. Může si rovněž nadefinovat jaký je u daného projektu použit vývojový modul a ukládá se i seznam nainstalovaných knihoven, který po přejítí na jiný počítač při první kompilaci sám stáhne. Je možné používat i adresářovou strukturu, která zpřehledňuje složitější projekty.

### 3.3.3 Výběr mikropočítače

Pro svůj záměr jsem si vybral mikropočítač ESP32 od společnosti Espressif, který je vhodný i pro průmyslové využití a je dostatečně cenově dostupný. Tím, že cena použitého čipu – ESP32 je velmi nízká, tak vícenásobným použitím zařízení v jednotlivých modulech se výsledné řešení příliš nezdraží a získáme tím řadu výhod. Při sestavování kurníku tak můžeme mít větší svobodu, jelikož všechny moduly nemusí být propojené dráty, ale je možné komunikovat bezdrátově. Lépe se tak řešení hodí i pro uživatele, kteří již kurník vlastní a chtějí si do něj doplnit pouze technologie (může snadněji umístit moduly).

Pro otestování jsem zvolil vývojový modul ESP32, kterým obsahuje USB-UART převodník a je možné do něj jednoduše nahrávat programy skrze USB kabel. Dále je možné zařízení přes tento konektor i napájet.

V průmyslovém řešení, kdy, již budou odladěny všechny neduhy je lepší použít pouze samostatný modul (ESP WROVER / ESP WROOM) a ten integrovat do konkrétně navržené integrované desky plošného spoje, který bude využívat jen specifickou část modulu. Výsledné zařízení je tak možné miniaturizovat a dostat se i na lepší výrobní náklady daných zařízení.

### 3.3.4 Volba způsobu komunikace

Pro mé řešení jsem volil mezi komunikačním protokolem MQTT a HTTP. Protokol MQTT je uzpůsoben pro komunikaci IoT zařízení, jelikož působ komunikace nepotřebuje tak velkou režii jako protokol http a lze u něj nastavit doručitelnost. Nicméně jsem ale nakonec zvolil řešení pomocí HTTP, jelikož jsem zvolil řešení, kde navržené moduly jsou víceméně samostatné a nepotřebují těch zpráv zasílat a přijímat nějak moc. Při užití protokolu http stačí mít jednoduchý server a odpadájí nutnost mít MQTT broukera. Nevýhodou http ale je zbytečně velká hlavička, a tedy posílání zbytečný dat navíc. Jelikož ale těch zpráv není potřeba zasílat tolik, tak si myslím, že se tato nevýhoda můžeme zanedbat.

### 3.3.5 Počítání slepic pomocí RFID brány

Rozhodl jsem se ověřit řešení v podobě očipování každé slepice pomocí pasivního RFID čipu, který se připne na nohy slepic. Toto řešení jsem zvolil, jelikož nepožaduji za podstatné, abychom museli znát přesnou lokalizaci slepice pomocí GPS modulu a mohli např. sledovat i to kolik slepice naběhá a kde se zrovna nachází, ale pouze monitorovat, zda jsou všechny slepice uvnitř kurníku před zavřením automatizovaných dvířek. Toto řešení pak může být zajímavé především z cenové dostupnosti. Nevýhodou pasivního RFID čipu je ale nízká sledovací vzdálenost, která ale pro požadované účely stačí a nemusíme navíc řešit stav baterie na případných aktivních RFID čípech.

#### 3.3.5.1 Princip řešení počítání slepic

Základním prvkem jsou 2 RFID čtecí zařízení z kterých je vytvořena pomyslná brána. Všechny slepice jsou označeny RFID čipy. Abychom dokázali identifikovat směr průchodu slepic, tak každá slepice musí mít unikátní označení. Tím, že zjistíme, které čtecí zařízení (RFID brána je složena z 2 čtecích zařízení) čip dříve přečetlo, tak dokážeme určit směr průchodu. Mohou nastat ale situace, kdy slepice neprojde skrze celé zařízení a bude zaznamenána detekce jen na jednom čtecím zařízení. Tyto situace je pak potřeba ošetřit v programu.



Obrázek 5: Ukázka řešení modulu RFID brány

Pro slepice se nejvíce hodí kroužky, které se mohou připnout slepicím k noze a tím, jak budou čtecí zařízení překračovat, tak je zajištěna dostupná vzdálenost.

Po zaznamenání detekce nad určitým zařízením se uloží konkrétní slepici čas a čtecí zařízení které slepici detekovalo. Pomocí časů, které se zaznamenávají v moment detekce pak určujeme stavy, kde se slepice nachází. A když víme, kde se slepice nachází, tak máme i přehled, kolik je slepic uvnitř a kolik vně.

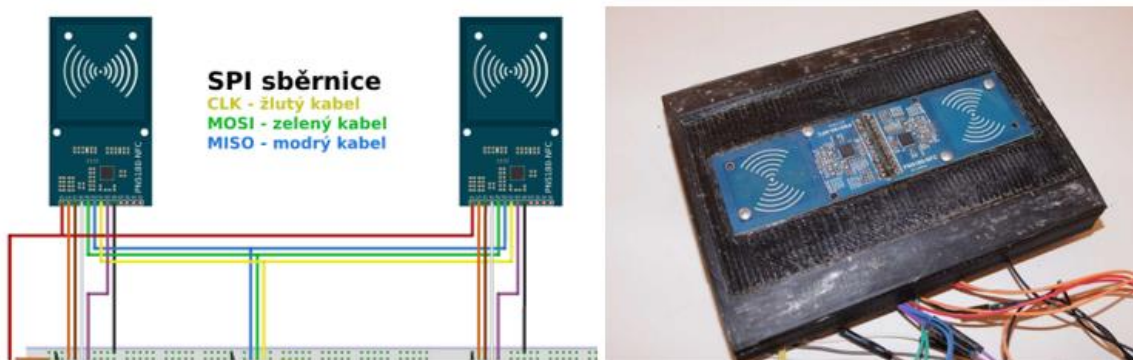
Jelikož správa údajů je pro udržení přehlednosti komplikovanější a aby bylo možné funkcionalitu i dále do budoucna jednoduše rozvíjet, tak jsem pro modul počítání slepic vytvořil 3 třídy – Coop, Chicken a State.

```
class Coop {  
private:  
    uint8_t numChicken;  
    uint8_t numChickenInCoop;  
    uint8_t lastUId[2][8];  
    String timeToOpen;  
    String timeToClose;  
    bool allChickenIsAlive;  
    bool requestToSendNotification;  
    Chicken2 chickens[];  
};  
  
class Chicken {  
private:  
    uint8_t tagId[8];  
    State state;  
    unsigned long rearder1Time;  
    unsigned long rearder2Time;  
};
```

Obrázek 6: Záhloví třídy Coop a Chicken

Ve třídě Coop jsou uchovány informace o celkovém počtu slepic, počtu slepic uvnitř, posledně načtený tag a obsahuje pole hodnot třídy Chicken a další informace.

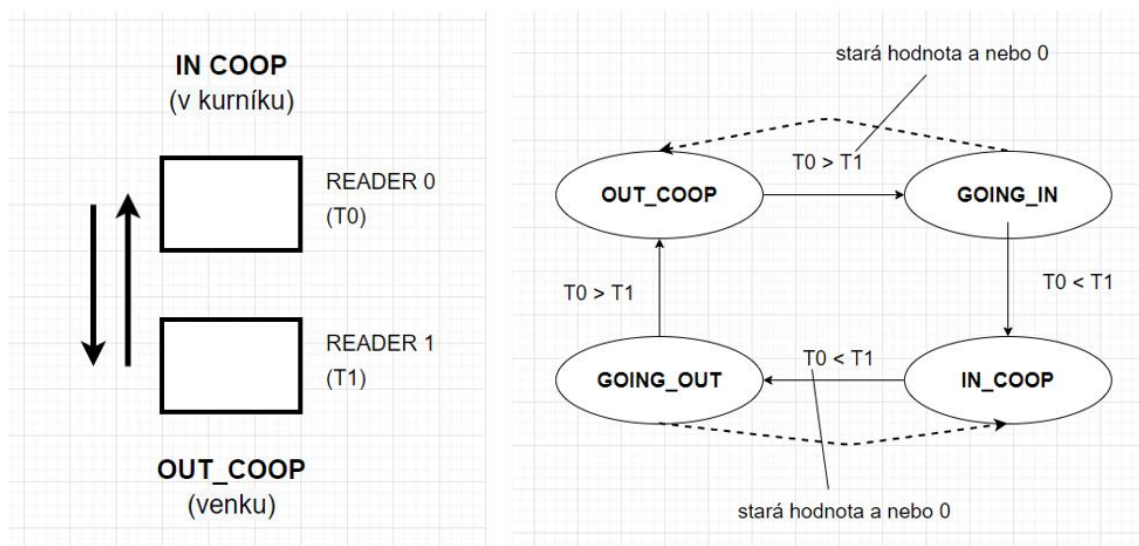
Ve třídě Chicken uchováváme informace hodnoty čipu kterým se slepice identifikuje, aktuální stav a časy, které byly zaznamenány při detekci nad čtecím zařízením.



Obrázek 7: Realizace RFID brány

Při sestavování brány jsem vycházel z knihovny Arduino-rfid-PN5180 [29], kde bylo již vyřešené čtení čipů. ESP32 pracuje stejně jako RFID čtecí zařízení PN5180 na 3,3V TTL logice. Komunikace přes SPI sběrnici.

Řešení počítání slepic funguje na principu konečného automatu. Všechny slepice mají na začátku nastavený stav na hodnotu OUT\_COOP. Každé čtecí zařízení uloží po průchodu do proměnné aktuální čas. Porovnáváním časů a stavu ve kterém se slepice aktuálně nachází můžeme určit její nový stav. Přechody mezi stavy a zobrazení čtecích zařízení a provázanost hodnot můžete vidět na diagramech níže. Je potřeba ochránit i stavy, kdy se načte pouze jedno zařízení, tedy stav, kdy se slepice vrátí odkud přišla, aniž by překročila druhé čtecí zařízení. K tomu jsou využity přechodové stavy – GOING\_IN a GOING\_OUT. Jakmile slepice získá stav IN\_COOP, tak se hodnota počtu slepic v kurníku zvýší a pokud dojde ke změně na stav OUT\_COOP, tak zase dojde ke snížení počtu slepic v kurníku.



Obrázek 8: Ukázka konečného automatu, který je základní část pro počítání slepic

V programu čtecí zařízení fungují tak, že se střídá čtení z čtecích zařízení. A aby čtení probíhalo co nejrychleji, tak to v programu funguje tak, že se uloží posledně načtený tag pokud je při dalším čtení shodný, tak dojde k přeskočení na čtení z druhého zařízení.

Nejprve se uloží posledně načtený čip do proměnné lastUid, uložíme záznam času přečtení. Hodnota proměnné „i“ uchovává informaci o jaké čtecí zařízení se jedná. Hodnota posledně načteného tagu se pak uloží ve třídě CoopPřirazení času konkrétní slepice probíhá tak, že se posledně načtený tag prohledá s tagy všech slepic a při shodě se uloží dané slepici uloží nová

hodnota zaznamenaného času. Po uložení nové hodnoty se zkontroluje, zdali nedošlo k změně stavu.

Výhodou mikropočítače ESP32 je to, že obsahuje 2 výpočetní jádra a sdílenou paměť. Jedno jádro využívám k sledování změn stavů, které pak ovlivňují proměnnou počtu slepic. Na druhém jádře pak sleduji, akorát, jestli se proměnná počtu slepic v kurníku nezměnila a pokud ano, tak pošlu POST request na server a ten informace uloží v databázi.

```
// Update the array of the last ID
memcpy(lastUid[i], thisUid[i], sizeof(lastUid[i][0])*8);
currentMillis = millis();

coop2.updateTimeChicken(lastUid, currentMillis, i);
```

Obrázek 9: uložení posledně načteného tagu do proměnné lastUid a předání času do funkce

```
bool Coop::updateTimeChicken(uint8_t tagId[8], unsigned long currentTime, uint8_t numReader){
    memcpy(this->lastUid[numReader], tagId, sizeof(this->lastUid[0])*8);

    uint8_t * tempTagId;
    for (int i=0; i<this->numChicken; i++) {
        tempTagId = this->chickens[i].getTagId();

        if(memcmp(tempTagId, this->lastUid, 8) == 0){
            //set time in readers
            Serial.println("Tag is EAQUAL ----- ");
            this->chickens[i].setReaderTime(numReader, currentTime);
        }
    }
}
```

Obrázek 10: vyhledání slepice podle tagu a času záznamu

```
State Chicken::updateState()
{
//OUT_COOP -> GOING_IN -> IN_COOP -> GOING_OUT -> OUT_COOP
//chicken has initial state GOING_OUT

switch(this->state){
  case OUT_COOP:
    if((this->rearder1Time < this->rearder2Time ))
    {
      this->state = GOING_IN;
      Serial.println("GOING_IN");
    }

    else if((this->rearder1Time < this->rearder2Time) && (rearder1Time==0)) {
      this->state = GOING_OUT;
      Serial.println("GOING_OUT");
    }
    else {
      Serial.println("debug - OUT_COOP");
    }
    break;

  case GOING_IN:
    if(this->rearder1Time < this->rearder2Time)
```

Obrázek 11: aktualizace stavu

```
switch (changedState)
{
  case IN_COOP:
    this->numChicken++;
    break;
  case OUT_COOP:
    this->numChicken--;
    break;

  default:
    break;
}
```

Obrázek 12: úprava hodnoty počtu slepic

```
//Task2code: vjeck num every 10 s
void Task2code( void * pvParameters ){
    Serial.print("Task2 running on core ");
    Serial.println(xPortGetCoreID());

    for(;;){

        if(num_chickens_in_coop_last != num_chickens_in_coop) {
            num_chickens_in_coop_last = num_chickens_in_coop;
            Serial.println(num_chickens_in_coop);
            httpPOSTRequest();
        }
        delay(10000);
    }
}
```

Obrázek 13: Pokud se změní hodnota počtu slepic v kurníku, tak dojde k odeslání dat na server

### 3.3.6 Automatické zavírání a otevírání dvířek kurníku

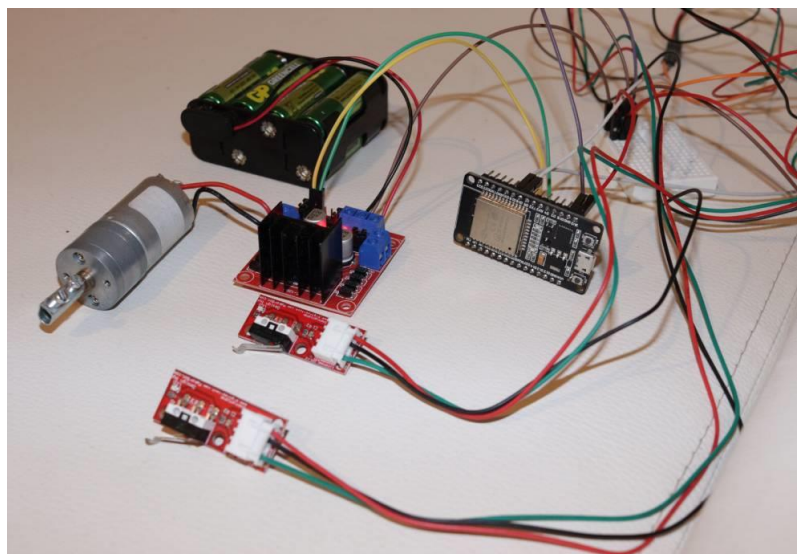
Dvířka slouží k zabránění vniknutí škodné zvěře do kurníku. Dvířka je potřeba ráno otevřít, aby z něj mohly vylézt slepice a večer zase zavřít. Existuje více způsobů, jak dát dvířkám impuls k otevření a nebo zavření. Některá řešení jsou postavena na snímání hladiny osvětlení a pokud se setmí podle úroveň nějakého limitu, tak se dvířka zavřou a ráno jakmile se rozední a dojde k překročení limitu hladiny světla, tak se zase otevřou. Uživatel si pak může částečně posouvat zpoždění pomocí potenciometru. Myslím si ale, že je toto řešení zbytečně komplikované, jelikož kdy se rozední a kdy zajde slunce lze zjistit např. na stránce [Meteogram.cz](http://meteogram.cz) a je tedy zbytečné snímat tuto hodnotu. Řešení vidím spíš v uložení hodnot západů a východů slunce po celý rok do proměnných. Tím že známe časy, tak není potřeba neustále snímat osvětlení. Pokud jsou hodnoty uloženy v proměnné, tak je snazší i tyto hodnoty upravovat. Lze tedy toto řešení pak aplikovat na větší množství lidí, kteří si mohou upravit nastavení, které jim konkrétně nejvíce vyhovuje.



Čvn	Čvc			Srp		Zář		Řij		
	☀	☀	☀	☀	☀	☀	☀	☀	☀	☀
2	20:50	04:52	21:02	05:26	20:33	06:10	19:36	06:53	18:32	06:41
1	20:51	04:53	21:01	05:27	20:31	06:11	19:34	06:55	18:30	06:43
1	20:52	04:54	21:01	05:29	20:30	06:13	19:32	06:56	18:27	06:45
0	20:53	04:54	21:01	05:30	20:28	06:14	19:29	06:58	18:25	06:46
0	20:53	04:55	21:00	05:31	20:26	06:16	19:27	06:59	18:23	06:48
9	20:54	04:56	21:00	05:33	20:25	06:17	19:25	07:01	18:21	06:49
9	20:55	04:57	20:59	05:34	20:23	06:19	19:23	07:02	18:19	06:51
8	20:56	04:58	20:59	05:36	20:22	06:20	19:21	07:04	18:17	06:53
8	20:57	04:59	20:58	05:37	20:20	06:21	19:19	07:05	18:15	06:54
8	20:57	05:00	20:57	05:38	20:18	06:23	19:17	07:07	18:13	06:56
8	20:58	05:01	20:57	05:40	20:16	06:24	19:15	07:08	18:11	06:57

Obrázek 14 - Tabulka s časy východu a západu slunce [30]

Jelikož není žádoucí, aby nějaká slepice zůstala mimo kurník po uzavření dvířek, tak je potřeba, aby systém dvířek znal hodnoty počtu slepic, které zaznamenává RFID brána. Zařízení fungují samostatně. RFID brána zasílá informace na server a ukládá počty slepic. Chytré dvířka si pak stáhnou hodnotu počtu slepic ze serveru, zavře dvířka a pokud některá zůstala venku, tak dojde k zaslání upozornění, že slepice je mimo kurník. Jelikož zařízení dostane před zvržením dvířek informaci o počtu slepic uvnitř a venku, tak máme přehled o tom, jestli nezůstala nějaká slepice vně kurníku. Systém je nastaven tak, že i když slepice zůstane v čas otevírání venku, tak dvířka se zavřou.



Obrázek 15: Ukázka modulu pro automatizované dvířka



Pro uživatele může být tento systém velmi usnadňující záležitostí, jelikož nemusí řešit každý den, jestli jsou všechny slepice uvnitř, ale jen v případech, kdy některá zůstane venku. Další výhodou je to, že se může pokusit situaci vyřešit nejdříve na dálku tím, že může ovládat dvířka skrze aplikaci. Dvířka se prioritně zavírají i když zůstane slepice venku. Je to z důvodu, aby během té doby, kdy by se na slepici čekalo, nevybila škodná zvěř celý kurník. Nicméně uživatel je na tuto situaci upozorněn a je na něm, jak danou situaci bude chtít řešit.

Modulu dvířek lze nastavit otevírání a zavírání v předem nastavených časech, úpravu těchto hodnot a ovládání dvířek přes webové rozhraní.

Chytrá dvířka jsou složena z 2 koncových spínačů, které dají informaci o tom, že dvířka se skutečně zavřela anebo otevřela. Z hlediska bezpečnosti je i nastaven čas otevírání a zavírání a pokud do stanoveného limitu dvířka neotevřou anebo nezavřou, tak dojde k vypnutí motoru a upozornění uživatele na danou událost.

Existuje velké množství řešení, jakým způsobem se mohou dvířek otevírat. Z programátorského hlediska je celkem jedno, který způsob se zvolí, jelikož se všechny řešení velmi podobají v tom, že vždy kontrolujeme, sepnutí koncových spínačů a nebo nastavení času sepnutí. Z technického hlediska je pak celkem jedno, jestli napojíme kontakty na pístový motor, na motor s navijákem anebo jakákoliv další řešení. Jistě by se hodilo otestovat všechny způsoby a je určitě na zvážení, které řešení je nejvhodnější z hlediska spolehlivosti, bezpečnosti, bezúdržbovosti a cenové dostupnosti.

Seznam použitých součástí: mikropočítač ESP32, H-můstek, dva koncové spínače, DC motor s převodovkou, napájení

### 3.3.7 Chytrá snášková místnost

Pro zjištění, jakým způsobem by se mohla počítat snesená vajíčka mne napadly 2 možné řešení, a to pomocí vyfocení a následné detekce obrazu vajíčka spočítat anebo pomocí jednoduchého tlačítka, které se stiskne, jakmile se vajíčko přes něj překutálí. Já zvolil jednodušší řešení na realizaci a to, že využiji toho, že je možné, aby se vajíčko odkutálelo pomocí nakloněného dna. Výhoda je ta, že stejně potřebujeme dostat vajíčko mimo dosah slepic a teoreticky dokážeme vajíčka nasměrovat až do tepelně izolovaného zásobníku nebo nějakého chladícího boxu.

Pro počítání vajíček potřebujeme několik komponent – RFID čtecí zařízení, které je umístěné pod snáškovým hnízdem a které dokáže na základě RFID tagu umístěného na nohách slepic identifikovat konkrétní slepici. Umístění může být při vstupu do snáškového hnízda nebo přímo v místě na kterém bude slepice sedět.



Obrázek 16: Ukázka modulu pro počítání vajíček

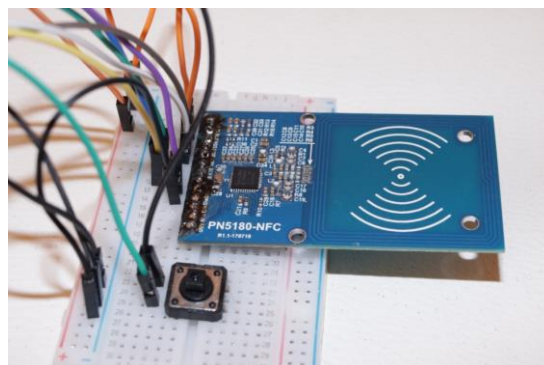
Můžeme sledovat pak jak dlouho je slepice na snášce, jak často tam chodí a kolik snese vajíček. Počítání a přesun vajíček je realizován tím, že snášecí zařízení je nakloněné a jakmile slepice odejde, tak se samospádem vajíčka odkutálejí. Pro tento účel jsem vytvořil pomocí 3D tiskárny jednoduchý model, který zajistí, aby vajíčka prošly postupně přes mikrosínač.

Díky tomuto řešení můžeme sledovat užítkovost slepic. Sledováním různých hodnot pak teoreticky můžeme zjistit, jestli třeba není slepice nemocná. Snáškové hnízdo může být časem uzpůsobeno tak, aby dokázalo slepici i zvážit.

Pokud by pak u dané slepice klesla užítkovost pod stanovenou normu, tak by mohlo být uživateli doporučeno, aby zvažil, jestli by slepice nebyla lepší v polévce.

Seznam použitých součástí: mikropočítač ESP32, tlačítko, RFID čtecí zařízení, RFID čipy

Princip počítání vychází z toho, že dokážeme určit pomocí RFID čtečky konkrétní slepici. Stisky se pak posledně detekované slepici v daném snášecím hnízdě započítají snesená vajíčka.



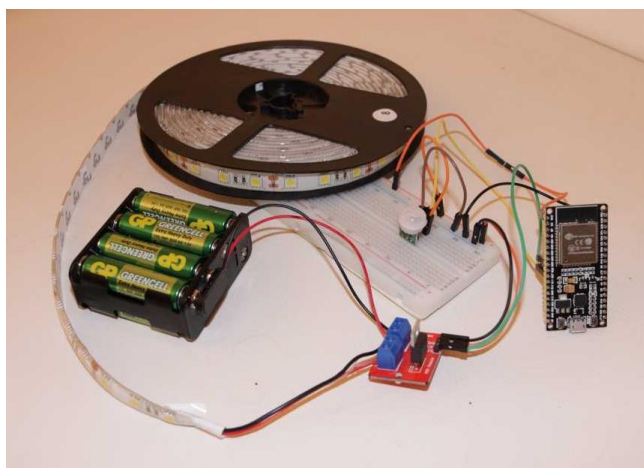
Obrázek 17: Testovací zapojení pro počítání slepic

### 3.3.8 Zabezpečení kurníku proti krádeži a škodné zvěři

Rozhodl jsem se pro jednoduché řešení složeného z několika senzorů pohybu, které po detekci rozsvítit led pás, kterým by byl pokryt kurník. Zvěř je většinou plachá na světlo a toto velmi jednoduché řešení by mohlo být účinné.

Zkoumal jsem 2 možnosti detekce, a to pomocí PIR senzoru, který sleduje změnu pomocí infračerveného světla anebo pomocí mikrovlnných vln, kde se sleduje změna elektromagnetického pole. Každý senzor má své klady a zápory. Pokud by se podařilo dobře odstínit pohyb slepic, tak by se dal použít jeden mikrovlnný senzor, který detekuje pohyb v 360 stupních přibližně kolem 7 m, záleží na senzoru. Senzor ale nejde snadně namířit jen na určitou oblast. Zkoušel jsem řešení, kdy jsem se snažil určitou rovinu useknout tak, že jsem před senzor dal železnou desku, přes kterou by mikrovlny neměly projít. Zatím jsem ale zkoumal tuto funkci jen velice povrchně a nedokáži říct, jestli by to bylo realizovatelné. U PIR senzorů je výhoda, že můžeme nasměrovat detekci tak, aby směřovala vně kurníku. Nevýhodou je, že musíme použít více senzorů, abychom dokázali pokrýt všechny směry, jelikož zařízení snímá jen určitý výsek.

Hlavní částí je zařízení je malý PIR senzor, který se zaznamená na mikropočítači a ten zase pouští do tranzistoru proud a tím se do led diod dostává externí napájení, které je potřebné k rozsvícení světla. Samotné piny mikropočítače totiž nedokáží napájet zařízení s větším odběrem.



Obrázek 18: Ukázka modulu pro zabezpečení

Zabezpečit kurník můžeme i pomocí chytrých pastí, které by měli výhodu v tom, že po detekci vibrace (zvěř pravděpodobně žere návnadu) zavře klec a upozorní uživatele. Pokud

by chovatel navštěvovat chov jen jednou týdně, tak není vhodné držet chycené zvíře v pasti dlouhou dobu a případně mít i možnost zamknutou klec otevřít na dálku.

### 3.3.9 Zaznamenání a ukládání fotek do vzdáleného úložiště

Hlavním prvkem tohoto modulu je vývojový kit ESP32 CAM, který obsahuje kameru. Zařízení zasílá data na vzdálené úložiště v pravidelných intervalech anebo při detekci pohybu. Do budoucna by zasláné fotografie mohl dokonce vyhodnotit systém na rozpoznávání obrazu, který by dokázal určit riziko a podle toho informoval uživatele. Možná by mohl být kurník vybaven i zvukovým záznamovým zařízením, které by dokázalo pomocí zvuku vyhodnotit riziko krádeže.

Pro realizaci funkce byl využit ukázkový program, V programu bylo potřeba upravit informace k připojení do wifi sítě a nastavit koncový bod kam se mají fotografie zasílat.

Aby se mohli bylo potřeba připravit api pro zpracování požadavku a následného přeposlání na cloudové úložiště Amazonu, které jsem zvolil.

Ukládání probíhá tak, že zaznamenaná fotografie se pošle na server, který zpracuje požadavek a následně pomocí získaných oprávnění a dohraných balíčků pro práci s cloudovými službami Amazonu - aws, multer, multer-s3.

V současném stavu lze fotografie zobrazit přes administrační rozhraní úložiště.

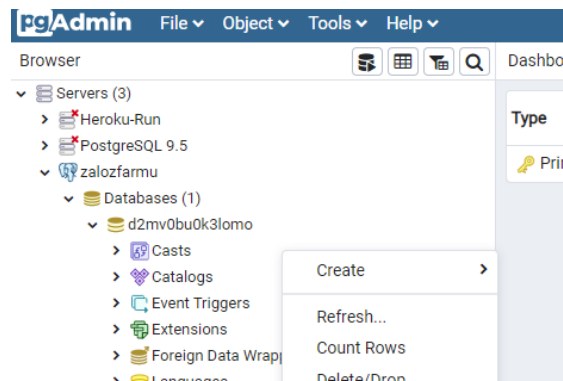


Obrázek 19: Mikropočítač ESP CAM, který obsahuje kameru

### 3.3.10 Databáze

Pro ukládání dat jsem si vybral databázi PostgreSQL. Pro svou funkcionalitu mi stačilo vytvořit pouze 2 jednoduché tabulky. S databází pracuji přes serverové rozhraní NodeJs. Používal jsem pouze základní operace, které jsou pro většinu relačních databází stejné. Pro

vytvoření tabulek a zobrazování hodnot jsem používal volně dostupné administrační rozhraní pgAdmin.



Obrázek 20: Rozhraní pgAdmin

### 3.3.11 NodeJS

Komunikace mezi mikropočítači a databází PostgreSQL probíhá skrz NodeJs, kde byl použitý framework Express pro usnadnění vytváření REST dotazů. Toto rozhraní jsem rozčlenil do několika sekcí, které se zabývají dílčími úkoly jako je správa dvířek, zabezpečení a ukládání fotek.

```
const express = require("express"),
      app = express(),
      bodyParser = require("body-parser"),
      path = require("path");

const doorRoutes = require("./server/routes/door"),
      fileRoutes = require("./server/routes/file-upload"),
      securityRoutes = require("./server/routes/security");

const { Pool } = require('pg');
const pool = new Pool({
  connectionString: 'postgres://eawppcqttxenab:d20a0f812e1255d0c9f4...
  ssl: {
    rejectUnauthorized: false
  }
});
```

Obrázek 21: NodeJs

## ZÁVĚR

Cílem bylo navrhnout systém pro řízení a monitorování kurníku. Podařilo se mi vytvořit prototypy modulů na kterých jsem simuloval navržené požadavky.

Pro identifikaci a určení pohybu slepic byla využita technologie RFID. Jako řídicí jednotky jsem využil mikropočítačové vývojové desky ESP32.

Výsledek práce ověřuje myšlenku možného usnadnění domácího chovu slepic pomocí moderních technologií. Systém zajišťuje automatické otvírání a zavírání dvířek v nastavený čas, kontrolu počtu slepic v kurníku, detekci neoprávněného vstupu, zabezpečení proti škodné zvěři a počítání vajíček jednotlivých slepic.

.

## 4 SEZNAM POUŽITÝCH ZDROJŮ

- [1] SMĚTÁKOVÁ, Kateřina. 20 důvodů, proč si (ne)pořizovat slepice. *Vše pro zvířátka* [online]. Hliník: Daniel Hanuš, 2019 [cit. 2020-08-09]. Dostupné z: <https://vse-pro-zviratka.cz/slepice/>
- [2] Food Loss and Food Waste. *Food and Agriculture Organization of the United Nations* [online]. Řím: Food and Agriculture Organization of the United Nations, 2019 [cit. 2020-08-09]. Dostupné z: <http://www.fao.org/policy-support/policy-themes/food-loss-food-waste/en/>
- [3] Organická hmota v půdě, její obsah, složky a význam. *Agromanual.cz* [online]. České Budějovice: Kurent, s.r.o., 2019 [cit. 2020-08-09]. Dostupné z: <https://www.agromanual.cz/cz/clanky/vyziva-a-stimulace/hnojeni/organicka-hmota-v-pude-jeji-obsah-slozky-a-vyznam>
- [4] MALÁ, G. a P. NOVÁK. Vliv welfare na zdraví hospodářských zvířat. *Česká technologická platforma pro zemědělství* [online]. Praha: Česká technologická platforma pro zemědělství, 2019 [cit. 2020-08-09]. Dostupné z: <https://www.ctpz.cz/vyzkum/vliv-welfare-na-zdravi-hospodarskych-zvirat-860>
- [5] *POTRAVINY Z DOMOVINY 23 - Rabbit Trhový Štěpánov. a.s. / Chov kuřat* [online]. [cit. 2020-08-10]. Dostupné z: <https://youtu.be/SiQmn-NivU4>
- [6] Chytrý chov drůbeže už v blízké budoucnosti. *Náš chov* [online]. Praha: Profi Press s.r.o., 2019 [cit. 2020-08-09]. Dostupné z: <https://www.naschov.cz/chytry-chov-drubeze-uz-v-blizke-budoucnosti/>
- [7] NOVOTNÁ, Iveta, Petr SMOLÍK a Luboš SMUTNÝ. *Sledování pohybové aktivity hospodářských zvířat*. Tábor.
- [8] *LELY - Chytré farmaření je vaší volbou* [online]. Łochowo: Lely East Sp. z o.o. [cit. 2020-08-09]. Dostupné z: <https://www.lely.com/cz/>
- [9] FESS, Ing. Ohaře, 2017.
- [10] *POTRAVINY Z DOMOVINY 25 - SPV Pelhřimov, a.s. / Chov prasat*. 2020. Dostupné také z: <https://www.youtube.com/watch?v=meaKQKX9g3w>
- [11] VAŇOUČEK, Jaroslav. *Polo - automatická dvířka kurníku*. 2015. Dostupné také z: <https://youtu.be/HiN-yhAImzo>
- [12] LTD, ChickenGuard. *ChickenGuard – Automatic Chicken Coop Door Opener*. Cambridge. Dostupné také z: <https://www.chickenguard.com/>
- [13] GARRET, Desmond. *RFID Chicken Coop Door Design*. 2018. Dostupné také z: <https://youtu.be/CQEznC6SyTs>
- [14] *V Číně sledují kuřata pomocí GPS náramků. Zjišťují, co jedí a jak se hýbou* [online]. In: . [cit. 2020-08-05]. Dostupné z: <https://vtm.zive.cz/clanky/v-cine-sleduji-kurata-pomoci-gps-naramku-zjistuji-co-jedi-a-jak-se-hybou/sc-870-a-197394/default.aspx>
- [15] TAJDUŠ, Pavel. *Otevírání kurníku (arduino)*. Dostupné také z: <https://www.youtube.com/watch?v=4dzdbR8q9G0>
- [16] HEATHER. *Hcarlsonlewiss Chicken Coop*. Whitesburg. Dostupné také z: <https://www.backyardchickens.com/articles/hcarlsonlewiss-chicken-coop.48076/>
- [17] DESTAR. *DEStar 12V DC Automatic Timer Controlled Indoor Outdoor Chicken Coop Door Opener Kit with Infrared Sensor and Bonus Remote Control*. Dostupné

- také z: <https://www.amazon.com/DEStar-Automatic-Outdoor-Controller-Infrared/dp/B0824JJYSR>
- [18] PREDATORGUARD. *Solar Powered Predator Deterrent Light Scares Nocturnal Pest Animals Away*. Dostupné také z: <https://www.amazon.com/PREDATORGUARD-Deterrent-Nocturnal-Repellent-Accessories/dp/B00E51X8Q8>
- [19] Internet věcí. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 6. 6. 2020 [cit. 2020-08-04]. Dostupné z: [https://cs.wikipedia.org/wiki/Internet\\_v%C4%9Bc%C3%AD](https://cs.wikipedia.org/wiki/Internet_v%C4%9Bc%C3%AD)
- [20] Home Smart IoT Home: Domesticating the Internet of Things. *Designers* [online]. San Francisco, CA [cit. 2020-08-04]. Dostupné z: <https://www.toptal.com/designers/interactive/smart-home-domestic-internet-of-things>
- [21] 5 CHYTRÝCH OTÁZEK NA TÉMA CHYTRÁ DOMÁCNOST. *Pěkné bydlení* [online]. Praha: Časopisy pro volný čas s.r.o. [cit. 2020-08-09]. Dostupné z: <https://www.peknebydleni.cz/5-chytrych-otazek-na-tema-chytra-domacnost/>
- [22] Mikropočítač. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2020-08-11]. Dostupné z: <https://cs.wikipedia.org/wiki/Mikropo%C4%8D%C3%ADta%C4%8D>
- [23] O nás. *Arduino* [online]. [cit. 2020-08-05]. Dostupné z: <https://www.arduino.cc/en/Main/AboutUs>
- [24] About Espressif. *Espressif* [online]. [cit. 2020-08-05]. Dostupné z: <https://www.espressif.com/en/company/about-us/who-we-are>
- [25] NEW CHIP ALERT: THE ESP8266 WIFI MODULE (IT'S \$5) [online]. In: BENCHOFF, Brian. 2014 [cit. 2020-08-05]. Dostupné z: NEW CHIP ALERT: THE ESP8266 WIFI MODULE (IT'S \$5)
- [26] Vývojová deska ESP32. *Arduino návody* [online]. [cit. 2020-08-05]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/vyvojova-deska-esp32.html>
- [27] HISTORIE RFID. *RFID-EPC.cz* [online]. Praha [cit. 2020-08-10]. Dostupné z: <https://www.rfid-epc.cz/co-je-rfid/historie-rfid>
- [28] RFID. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2020-08-11]. Dostupné z: <https://cs.wikipedia.org/wiki/RFID>
- [29] Arduino-rfid-PN5180. *Github.com* [online]. 2018 [cit. 2020-08-10]. Dostupné z: <https://github.com/playfultechnology/arduino-rfid-PN5180>
- [30] V kolik se rozednívá a stmívá?. *Meteogram.cz* [online]. [cit. 2020-08-10]. Dostupné z: <https://www.meteogram.cz/vychod-zapad-slunce/>



## **SEZNAM ZKRATEK**

REST – způsob komunikace přes HTTP

MQTT broker – komunikační komponenta IoT

Bastlír – označení počítačového kutila

RFID – rádiově-frekvenční identifikace

IoT – Internet věcí

**SEZNAM OBRÁZKŮ**

Obrázek 1: Ukázka automatických dvířek společnosti Chickenguard [12].....	7
Obrázek 2: Systém pro počítání slepic pomocí RFID čtecích zařízení [13].....	7
Obrázek 3: Různé způsoby otevírání dvířek [15] [16] [17].....	8
Obrázek 4: Ukázka komerčního řešení k ochraně slepic [18] .....	8
Obrázek 5: Ukázka řešení modulu RFID brány.....	19
Obrázek 6: Záhlaví třídy Coop a Chicken .....	20
Obrázek 7: Realizace RFID brány .....	20
Obrázek 8: Ukázka konečného automatu, který je základní část pro počítání slepic .....	21
Obrázek 9: uložení posledně načteného tagu do proměnné lastUid a předání času do funkce .....	22
Obrázek 10: vyhledání slepice podle tagu a času záznamu .....	22
Obrázek 11: aktualizace stavu .....	23
Obrázek 12: úprava hodnoty počtu slepic.....	23
Obrázek 13: Pokud se změní hodnota počtu slepic v kurníku, .....	24
Obrázek 14 - Tabulka s časy východu a západu slunce [30] .....	25
Obrázek 15: Ukázka modulu pro automatizované dvířka .....	25
Obrázek 16: Ukázka modulu pro počítání vajíček.....	27
Obrázek 17: Testovací zapojení pro počítání slepic .....	27
Obrázek 18: Ukázka modulu pro zabezpečení .....	28
Obrázek 19: Mikropočítač ESP CAM, který obsahuje kameru.....	29
Obrázek 20: Rozhraní pgAdmin .....	30
Obrázek 21: NodeJs .....	30

## SEZNAM TABULEK

## SEZNAM PŘÍLOH

