

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Automatizovaný systém pro ekologický chov drůbeže

Bakalářská práce

Aleš Císař

Školitel: PhDr. Milan Novák, Ph.D.

České Budějovice 2022

Bibliografické údaje

A. Císař, 2022: Automatizovaný systém pro ekologický chov drůbeže [Automated system for organic poultry farming. Bc. Thesis, in Czech.] – 77 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic

Anotace

Bakalářská práce navrhuje konkrétní technické řešení automatizovaného systému pro ekologický chov drůbeže. Po provedené analýze problematiky, podmínek chovu a získání hlavních parametrů, které musí být splněny, byl navržen modulární systém. Systém zohledňuje základní požadavky ve formě parametrů, které podmiňují efektivní chov. Jedná se zejména o řízení oblasti bezpečnosti, mikroklimatu, potravy a výsledné produkce. Přestože klíčovou vlastností řídicího systému je jeho autonomní režim, je doplněn i o mobilní aplikaci, která umožňuje monitorování aktuálního stavu a manuální ovládání zejména akčních členů. Navržený systém by měl pomoci chovatelům snížit náklady a časovou náročnost malochovu drůbeže v mobilních kurnících.

Klíčová slova

automatizace, řízení, Arduino, RobotDyn, ESP82, malochov, drůbež

Annotation

This bachelor thesis proposes a specific technical solution of an automated system for organic poultry farming. After the analysis of the problematics, breeding conditions and obtaining the main parameters that must be met, a modular system was designed. The system takes into account the basic requirements in the form of parameters that condition effective breeding. It is mainly the management of safety, microclimate, food and the final production. Although a key feature of the control system is its autonomous mode, it is supplemented by a mobile application that allows monitoring of the current status and manual control of actuators in particular. The proposed system should help breeders to reduce the cost and time of small-scale poultry farming in mobile hen houses.

Key words

automation, control, Arduino, RobotDyn, ESP82, small farming, poultry

Prohlášení

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

V Českých Budějovicích dne 9.4.2022

Ales
Cisar



Digitálně
podepsal Ales
Cisar
Datum: 2022.04.09
12:57:48 +02'00'

Poděkování

Rád bych při této příležitosti poděkoval svému vedoucímu bakalářské práce panu PhDr. Milanovi Novákovi, Ph.D., za jeho cenné připomínky, rady a metodické vedení při zpracování zadané bakalářské práce.

Obsah

Seznam zkratek.....	1
1 Úvod	2
1.1 Cíle práce	3
1.2 Abstraktní popis řešení	4
2 Teoretická část	5
2.1 Podmínky pro ekologický chov.....	5
2.1.1 Teplota.....	7
2.1.2 Relativní vlhkost	7
2.1.3 Světelný režim.....	7
2.1.4 Přísun krmiva	8
2.1.5 Přísun vody.....	8
2.1.6 Sledování snášky	8
2.1.7 Bezpečnost	9
2.2 Návrh systému.....	9
2.2.1 Současný stav řešení.....	9
2.2.2 Klíčové veličiny	11
3 Praktická část.....	14
3.1 Blokové schéma systému	14
3.2 Mikrokontroler	15
3.3 Architektura	16
3.4 Proč nepoužít mikroprocesor?	18
3.5 Jak vybrat mikrokontroler	20
3.5.1 Aplikace	21
3.5.2 Hardwarové požadavky	21
3.5.3 Softwarové požadavky	21

3.5.4 Architektura MCU.....	21
3.5.5 Požadavky na paměť	22
3.6 Výběr čidel a akčních členů	22
3.6.1 Čidlo teploty a vlhkosti	22
3.6.2 Koncové spínače	23
3.6.3 Ovladač krokového motoru.....	23
3.6.4 Krokový motor	24
3.6.5 Fotorezistor.....	24
3.6.6 Senzor pohybu.....	25
3.6.7 Ovladač vyhřívání, větráku, LED	25
3.6.8 Senzory RFID.....	26
3.6.9 IR senzor	27
3.6.10 Čidlo množství kapaliny.....	27
3.6.11 Elektromagnetický ventil	28
3.6.12 RTC	28
3.6.13 Servomotor	29
3.6.14 Shrnutí	30
3.7 Volba mikrokontroleru	30
3.7.1 Platforma RobotDYN.....	31
3.8 Elektronické zapojení	33
3.8.1 Schéma zapojení.....	33
3.8.2 Návrh tištěného spoje	36
3.8.3 Výroba tištěného spoje	36
3.8.4 Osazení tištěného spoje	38
3.9 Programování.....	39
3.9.1 Příprava MCU pro programování	40
3.9.2 Webový server pro provoz aplikace.....	41

3.9.3 Nastavení ESP 8266 jako přístupový bod	43
3.9.4 Konstrukce serverové aplikace.....	44
3.9.5 Uživatelské rozhraní serverové aplikace.....	49
3.10 Programování čidel a akčních členů	49
3.10.1 Ovládání dveří	50
3.10.2 Evidence drůbeže	51
3.10.3 Snímání teploty a vlhkosti.....	53
3.10.4 Spuštění vytápění	54
3.10.5 Spuštění větrání	54
3.10.6 Detekce predátorů	55
3.10.7 Modul reálného času	55
3.10.8 Přisun vody.....	56
3.10.9 Přisun krmiva	57
3.10.10 Počítání snášky	57
3.11 Konstrukční zpracování	58
3.11.1 Bezpečnost	58
3.11.2 Produkce.....	59
3.11.3 Potrava.....	61
3.11.4 Řídící jednotka	61
3.12 Testování systému	62
3.12.1 Testování senzorů.....	62
3.12.2 Testování serveru	62
3.12.3 Testování mechanických částí.....	63
3.13 Úpravy systému pro další verzi	64
3.13.1 Čidla	64
3.13.2 Evidence drůbeže	64
3.13.3 Sběrný koš	64

3.13.4 Ovládání dvířek	64
3.13.5 Bezpečnost	65
3.13.6 Napájení systému	65
3.13.7 Dešťová voda	65
4 Závěr	66
Seznam literatury	68
Seznam obrázků.....	74
Seznam tabulek.....	76
Příloha A.....	77

Seznam zkratek

ADC	– Analog digital converter
CPU	– Central processing unit
CSS	– Cascading Style Sheets
DAC	– Digital analog converter
GPIO	– General Purpose Input Output
HTML	– Hypertext Markup Language
I/O	– Input/Output
IoT	– Internet of Things
MCU	– Mikrokontroler
MPU	– Mikroprocesor
PC	– Program counter
PCB	– Printed Circuit Board
PWM	– Pulse Width Modulation
RAM	– Random Access Memory
RFID	– Radio Frequency Identification
ROM	– Read Only Memory
RTC	– Real-time clock
UART	– Universal asynchronous receiver-transmitter
USB	– Universal Serial Bus
Wi-Fi	– Wireless Fidelity

1 Úvod

Produkci drůbeže tvoří maso různých druhů a konzumní vejce, která produkují slepice snáškového typu. V České republice tvoří chov drůbeže nejrozšířenější odvětví živočišné výroby založené na koncentrované produkci konzumních vajec, jatečné drůbeže a technologiích chovu srovnatelných s EU. Kromě toho je drůbežářská prvovýroba dodavatelem vstupních surovin pro průmyslové zpracování. Drůbeží výrobky (maso a vejce) mají nezastupitelné místo ve výživě obyvatelstva. Drůbeží maso nejvíce odpovídá svým složením doporučením, která vypracovaly týmy odborníků zabývajících se výživou. Z nutričního hlediska je drůbeží maso výhodné z důvodů vysokého obsahu bílkovin, především esenciálních aminokyselin, vysokého podílu esenciálních nenasycených mastných kyselin, minerálních látek a vitamínů, stejně jako nízkého podílu tuků především u hrabavé drůbeže. Splňuje většinu požadavků na surovinu pro racionální výživu a může pozitivně ovlivnit některé nepříznivé jevy vyplývající z nutričních výzkumů (nadměrná energetická dávka, vysoká spotřeba tuků, cukru a kuchyňské soli). Teoreticky mají drůbeží výrobky všechny předpoklady pozitivně ovlivnit realizaci racionální i léčebné výživy v rodině a ve společném stravování [1, s. 5].

V poslední době prošly systémy ustájení velkými změnami. Původní technologie chovu drůbeže byly založeny převážně na klecových chovech. Klecové chovy se vyznačují zejména tlakem na kvantitu na úkor kvality. Od této technologie se prostřednictvím směrnice EK 74/1999 začaly prosazovat tzv. alternativní systémy. Alternativní systémy ustájení slepic respektují volný pohyb nosnic, umožňují popelení, běhání a létání. Zajišťují možnosti hrabání, hřadování, snášky vajec v hnízdech, dostatek krmného a napájecího prostoru. Na druhou stranu jsou nosnice více stresovány sociálním složením hejna, přístupem ke krmivu, vodě [2, s. 7].

Přes neustálý tlak na zvyšování produkce se přeci jenom začíná hovořit o tzv. welfare (pohodě) v chovech drůbeže. Dodržování welfare požadované jednotlivými technologiemi v zemědělském sektoru je velmi přísně kontrolováno Státní veterinární správou ČR. V České republice je výroba vajec v zemědělském sektoru doplňována výrobou v domácích hospodářstvích (do 350 kusů nosnic), na které se povinnosti vyplývající ze směrnice nevztahují. Mohlo by se zdát, že by domácí sektor mohl nahradit zemědělský sektor. Ovšem

domácích podmínkách není možné vyrobit požadované množství ani kvalitu vajec pro celou obchodní síť [3]. I přes to, se do chovných systémů začíná prosazovat i systém ekologický.

Ekologický systém chovu drůbeže odráží veškeré požadavky na welfare chovy [4, s. 82]. Ekologické chovy musí být travnaté. Toho lze docílit jen častým střídáním výběhů [5, s. 83]. Největším problémem ekologických chovů bývá právě travnatá plocha, kdy za deštivého počasí bývá slepicemi devastována. Tento aspekt může být hlavní motivací k vytvoření efektivního systému pro co největší autonomizaci ekologického chovu drůbeže. Nelze předpokládat, že minimálně pro počáteční návrh bude takový systém hlavním konkurentem pro automatizační systémy velkochovů.

Tato práce se zabývá návrhem systému pro ekologický chov drůbeže v podobě malochovných objektů, u kterých se předpokládá časté přesouvání pro dodržení podmínky zelené plochy. Tyto objekty by měly v co největší šíři poskytnout bezobslužný provoz při dodržení kontroly nad stavem chovu drůbeže. Dnešní technologie pokročily tak, že činnosti spojené s bezobslužným a ekologickým chovem drůbeže poskytují silné nástroje pro jeho realizaci.

1.1 Cíle práce

Hlavním cílem této bakalářské práce je vytvořit autonomní řídicí systém, který pokryje požadavky na ekologický chov drůbeže. Tento systém bude vyvíjen jako otevřená platforma, která se bude snažit obsáhnout co nejvyšší modularitu tak, aby si případný uživatel mohl zvolit, které komponenty budou pro něho nejvhodnější, a to i s ohledem na případnou rozšiřitelnost v průběhu času. Pro naplnění hlavního cíle je nezbytné splnit několik dílčích úkolů:

- Analyzovat problematiku a podmínky ekologického chovu drůbeže s ohledem na získání parametrů, které musí být splněny.
- Na základě analýzy navrhnout model systému, který bude podporovat v co největší šíři podmínky modularity a autonomie.
- Podle návrhu obecného modulu systému vytvořit návrh hardwarových komponent, které budou nejvhodnější pro pokrytí stanovených podmínek.
- Otestovat skladbu zvolených elektronických komponent na prototypovém zařízení.

- Pokusit se implementovat celý systém do reálného provozu na modelovém malochovu drůbeže.
- Otestovat celý systém a na základě výsledků navrhnout jeho změny.

1.2 Abstraktní popis řešení

Hledané řešení je technicko-funkčním celkem, který obsahuje dvě základní složky. Jedná se především o hardwarovou platformu, která obsahuje systém řízení a mechanické komponenty, které jsou automatizačním systémem ovládány. Další částí je software, který tvoří komunikační rozhraní pro nastavování a ovládání dílčích komponent.

K řešení podobných úloh, které jsou v této práci definovány je přístupováno různým způsobem v závislosti na dostupné či zvolené technologii. Může se jednat o nativní aplikaci v osobním počítači, která tvoří komunikační rozhraní s ovládanými komponentami, nebo lze využít řešení prostřednictvím cloudových služeb.

V rámci této práce vznikne systém, který bude představovat primárně řešení pro co možná nejuniverzálnější použití v malochovu drůbeže. Je třeba hned na začátku brát v úvahu podmínky definované ekologickým chovem. Následovat bude kvalitní návrh architektury technického řešení, který bude obsahovat i softwarovou složku pro ovládání systému. Ovládací software bude univerzální, aby se snížily náklady a čas dalšího vývoje pro různé operační systémy a platformy.

2 Teoretická část

Moderní společnosti zabývající se chovem drůbeže mnohdy využívají kompletní systém řízení, který jim pomáhá řídit jejich každodenní výrobní činnosti. Systémy pokrývají širokou škálu požadavků od řízení personální kanceláře, řízení nákupu-prodeje zásob, monitoringu a kontroly prostředí v drůbežárnách až po sledování informací o jednotlivých kusech drůbeže. Zároveň zahrnují řízení dohledu nad produkty, diagnostiku a včasné varování před chorobami [6, s. 2].

S rozvojem rozsáhlého a intenzivního chovu drůbeže se uplatňují inteligentnější a automatizovanější technologie a metody [7, s. 102], jako je technologie radiofrekvenční identifikace [8, s. 183-186], technologie internetu věcí (IoT) a cloudové technologie. Existují i metody jako detekce nemocí drůbeže, sledování diety drůbeže [9, s. 1-6], sledování prostředí v drůbežárnách [6, s. 1-5], sledování produkce a abnormální detekce v drůbežárnách za účelem dosažení precizního chovu.

Systém precizního chovu drůbeže se v posledních letech dostává stále více do popředí [11, s. 614]. Někteří výzkumníci pomáhají farmářům kontrolovat a monitorovat zdravotní stav prostřednictvím IoT, analýzy snímků a dalších technologií [12, s. 19-20]. Jiní vytvářejí online platformy a využívají chytré senzory k záznamu a správě výrobních informací v reálném čase [13, s. 191-205]. Přestože technologie bezdrátového snímání a cloudové platformy jsou velmi pokročilé, neexistuje žádný kompletní systém, který by pokryl všechny funkce, které by uspokojily potřeby pro řízení chovu drůbeže. Mezi technické potíže patří jednotná konstrukce systému, rozumné rozdělení funkčních modulů, dobrá vzájemná spolupráce mezi moduly, souhra softwaru a hardwaru a inteligence systému.

Přestože výše uvedené odstavce naznačují existenci poměrně rozsáhlých systémů pro řízení chovu drůbeže, jedná se především o využití ve velkochovech. Pro potřeby malého chovu v ekologickém módu nebude potřeba takové rozsáhlé řešení, ale lze z uvedených hledisek vycházet.

2.1 Podmínky pro ekologický chov

Přijetí směrnice Rady č. 1999/74/ES z 19. června 1999 podnítilo rozvoj alternativních systémů. Většina těchto systémů jsou modifikace toho, co bylo poznané a rozvíjené před

mnoha lety. Ale díky novým poznatkům a technologiím byly představeny v podobě, která uspokojí současné legislativní, ekonomické a hlavně etické podmínky. Zvláštní postavení v chovech drůbeže mají alternativní chovy. Pod tímto názvem se ale skrývá celá řada systémů ustájení. Lze je rozdělit do několika kategorií: chov na hluboké podestýlce s možností roštové podlahy, voliérový chov a intenzivní výběhový chov [1, s. 27].

Podle uvedené směrnice určující podmínky pro ekologický chov drůbeže je zásadním faktorem volný pohyb. Tento fakt je klíčový z toho důvodu, že drůbež na omezeném prostoru značně degraduje venkovní výběh. Proto, aby mohl zregenerovat, poskytovat dostatek živin a opět plnil svůj účel, se musí drůbež i její příbytek stěhovat na různá místa a hovoří se o tzv. mobilních kurnících.

Při zajištění mobility, ale přichází další problémy spjaté s kontrolou nad drůbeží. Mnohdy mohou být kurníky umístěny v odlehlejších lokalitách a každodenní kontrola a servis jsou ekonomicky a konec konců i ekologicky nevýhodné. Proto vzniká požadavek na autonomní přístup k chovu drůbeže. K tomu lze využít současných technických prostředků, které při vhodném návrhu mohou principu autonomie pomoci.

Pro samotný výběr vhodných komponent, které zajistí automatický chod kurníku se musí stanovit parametry v podobě veličin, které se budou měřit, popřípadě regulovat. Lze částečně vycházet z welfare typu chovu. Ten je založen na volném pohybu a realizaci celého repertoáru přirozeného chování, neustálém přístupu ke krmivu a vodě, přirozeném osvětlení a přátelských vztazích mezi zvířaty [5]. Také lze vycházet z definice mikroklima ustájení, které je komplexním pojmem zahrnující vzájemnou interakci mezi různými faktory prostředí. Teplota prostředí je ovlivňována například příjmem krmiva a naopak. Veličiny, které definují mikroklima u velkochovů, jako je teplota, relativní vlhkost, větrání a objemy pohybu vzduchu jsou přesně korigovány, ale opět se vztahují k halovým velkochovům [14].

Protože pro malochovy existují jen oddělené systémy v podobě ovládání dílčích parametrů, je vhodné vyjít ze základních předpokladů, které jsou využívány v halových chovech. Také se musí přihlídnout ke specifickým ekologického chovu a praktičnosti, zda konkrétní parametr vůbec sledovat, a který je podstatně důležitější. Od těch se při potřebném nastavení pro ekologický chov provede návrh odpovídajícího systému.

2.1.1 Teplota

Sledování teploty je primární u ustájení drůbeže ve velkých lehkých halách. V případě nevhodného mikroklimatu mohou zvířata trpět stresem z vysokých teplot. Tato zátěž způsobuje nejen zhoršení zdravotního stavu, ale i snížení produkce, a tím vším se snižuje zisk výrobce [1, s. 27]. Mohlo by se zdát, že u mobilního ustájení drůbeže je tato veličina zbytečná, ale záleží na konkrétní poloze. Zda je ve stínu nebo na volném prostranství. V případě příliš teplého počasí nebo naopak deštivého je dobré zajistit odpovídající podmínky i uvnitř mobilního ustájení. Z hlediska teploty je optimální udržovat tzv. termoneutrální zónu, která odpovídá 13 až 24 °C [1, s. 27]. V horkých nebo naopak velmi studených dnech se tohoto teplotního rozmezí dá docílit pouze kvalitní klimatizací. Při zachování užitečnosti se dá tolerovat teplota mezi 24 až 29 °C [15, s. 2].

2.1.2 Relativní vlhkost

Sledování relativní vlhkosti souvisí s teplotou [16]. Ke zvýšení vlhkosti dochází především dýcháním, odpařením z výkalů a snesených vajec. Produkce vodních pár vydýcháváním se u slepic pohybuje mezi 3,6 – 5,8 g.h⁻¹ na kus při teplotě 20 °C. S vyššími teplotami výrazně narůstá a např. při teplotě 30 °C je produkce vodních par přibližně kolem 6,5 g.h⁻¹.ks⁻¹. Při chovu na podestýlce se odpařuje více vody než při chovu v klecích. Za jednu hodinu se z 1 m² podestýlky odpaří 15 – 30 g vody, ale podestýlka si zachovává svoji stabilní vlhkost. Vysoká vlhkost u podestýlky a v ovzduší zvyšuje uvolňování čpavku a sirovodíku a také počet znečištěných vajec [17]. Tyto údaje jsou opět důležité zejména pro velkochovy. Pro malá mobilní ustájení nemusí mít takový význam, ale je nutné o tomto faktoru vědět a případný systém navrhnout tak, aby i tuto veličinu umožňoval sledovat.

2.1.3 Světelný režim

Tento faktor jako prvek mikroklimatu při chovu drůbeže je velmi důležitý, zejména v halových velkochovech. Zásadně ovlivňuje reprodukční funkce, chování zvířat a jejich sociální interakce. V chovech bez vnějšího výběhu je rozdělení a regulace délky světelného dne na několik period světla a tmy a různou intenzitou osvětlení ovlivňuje např. dobu snášky a intenzitu páření [16]. U mobilních ustájení a k přihlédnutí k samotné povaze ekologického chovu, kdy má drůbež přirozený pohyb a není prioritou extrémní produkce, se lze domnívat, že tento faktor mikroklimatu nemá vysokou prioritu.

2.1.4 Přísun krmiva

Ve velkochovech je krmivo připravováno pomocí automatických míchačů a podle nastaveného poměru dokáže připravit odpovídající stravu. Sypké krmivo se dá snadno přenášet pomocí potrubí až k miskám. K tomuto účelu se využívá několik typů krmítek. Může se jednat o řetězová krmítka vhodná pro chov na hluboké podestýlce a roštových podlahách, kdy dochází k dopravě krmiva prostřednictvím řetězkových nosičů, umístěných ve žlábcích. Ty jsou v tzv. krmném okruhu a průběžně projíždí pod zásobníkem s krmivem [18]. Může se jednat také o zásobníková krmítka, která se využívají v podlahových chovech pro suché krmné směsi. Zásobníková krmítka využívají samočinného sesypávání obsahu krmiva do obvodového žlábků [18]. Z hlediska mobilního zařízení pro chov drůbeže je sice určující, že si drůbež potravu hledá přirozeně, ale je otázkou, v jakém množství ji dokáže omezený prostor výběhu poskytnout. Potom příkrm prostřednictvím krmítek nepozbývá úplně významu.

2.1.5 Přísun vody

Poskytnutí čisté a studené vody je zásadní předpoklad pro ochranu proti vysokým teplotám. Protože ptáci v období s vysokou teplotou ztrácejí minerály, musí být tyto látky přidávány do pitné vody [1, s. 30]. Voda musí být čistá, nezávadná a předkládaná v přebytku. Při použití kloboukových napáječek se doporučuje několikrát za den vodu vyměnit, napáječky čistit a podle potřeby dezinfikovat. Teplota vody by se v létě měla pohybovat v rozmezí 12 – 18 °C, v zimě 18 – 22 °C. Nejmenší kategorie drůbeže musí dostat vodu odstátou, o teplotě 20 – 24 °C. Studená voda způsobuje u malých kuřat průjmy [1, s. 20]. Podle způsobu ovládání přítoku vody lze napáječky rozdělit na napáječky s přímým ovládáním, kdy drůbež uvolňuje vodu přímým dotekem ventilu, a s nepřímým ovládáním, u nichž se voda automaticky doplňuje po snížení hladiny nebo zásoby [1, s. 21].

2.1.6 Sledování snášky

Přestože je kapitola pojmenována jako sledování snášky, jedná se spíš o technologii sběru vajec. Samozřejmě, že tyto dvě věci jsou spjaté. Jedná se především o to, jak detekovat, zda se ve snáškovém hnízdě nachází vejce a pokud ano, tak jak s ním dál pracovat. V malých chovech se uplatňují hnízda s ručním sběrem vajec. Mobilní zařízení pro chov drůbeže je sice zástupcem malého chovu, ale protože se předpokládá autonomní přístup, tak i tento parametr je nutné řešit.

Při větších koncentracích je základem sběru vajec samovolné vykutálení sneseného vejce ze snáškového hnízda a následný ruční nebo mechanizovaný sběr. Podmínkou pro vykutálení vajec je vhodný sklon dna hnízda, který je závislý na materiálu a úpravě dna; nesmí přitom vytvářet podmínky pro poškození vajec. Ke spolehlivému vykutálení dochází již při sklonech 8 – 10°, pokud se dno udržuje v čistém stavu. U menších sklonů se u vajec s menší hmotností spolehlivost snižuje [1, s. 22].

2.1.7 Bezpečnost

Pohled na bezpečnost v souvislosti s ekologickým chovem drůbeže lze pojmut dvěma způsoby. V první řadě se jedná o bezpečnost týkající se samostatného mikroklimatu. To znamená, zda teplota, vlhkost nebo koncentrace plynů nedosahuje rizikových hodnot, které by vedly k úhynu zvířat. Může se také jednat o bezpečnost v zálohových systémech. Jedná se hlavně o technickou složku bezpečnosti. Dále se může jednat o biologickou bezpečnost spojenou se zoohygienou [1, s. 57]. V rámci malochovu prostřednictvím mobilního příbytku, je ale důležitou částí bezpečnost fyzická. Myslí se tím především útok predátorů v podobě kun, či potkanů. Pokud je drůbež ve venkovním prostranství, tak se na noc vrací do příbytku. S tím je spojené fyzické zajištění mobilního příbytku, před vniknutím zmiňovaných predátorů.

2.2 Návrh systému

V předchozích kapitolách proběhla analýza podmínek pro ekologický chov drůbeže, ze které vzešly primární parametry, které by se měly zohlednit při návrhu systému. Důležitá a někdy opomíjená kontrola faktorů prostředí, jako je teplota, vlhkost, kvalita vzduchu, zajištění dostatečného přísunu světla a vody, ovlivňují drůbež a mohou vést k poruchám příjmu potravy, dýchání, trávení, změnám chování a vyšší pravděpodobnosti onemocnění [20, s. 187]. Dalším již zmiňovaným parametrem je přirozený výběh, mobilita kurníku a omezení zásahů obsluhy. Přestože se tento typ chovu nemůže rovnat velkokapacitním drůbežárnám, můžeme vhodným řízením zvětšit produkci a eliminovat finanční ztráty.

2.2.1 Současný stav řešení

V rámci šetření, zda existuje nějaký ucelený systém, který by pokrýval kompletní požadavky na ekologický chov drůbeže s přihlédnutím na již několikrát zmiňovanou podmínku mobility chovu, bylo možné zjistit, že pro zajištění potřeb drůbeže se hovoří o tzv. embedded systémech pro drůbeží farmy nebo také o embedded drůbežích farmách. Tento pojem lze chápat jako

systém, který je schopen automaticky regulovat proměnné, ovlivňující pohodu drůbežářské farmy. Pohoda drůbeže závisí do značné míry na několika environmentálních nebo klimatických parametrech, které ovlivňují výnosy a výkonnost produkce, jako je doba snášky, hmotnost vajec a průměrná hmotnost. Kurník by měl být schopen nejen měřit a monitorovat, ale měl by i řídit teplotu, vlhkost, příjem vody a dobu osvětlení [21]. Embedded drůbeží farma je tedy taková, která je plně automatizovaná za účelem zlepšení produktivity a environmentálních nebo klimatických podmínek [20, s. 2]. Rozšiřujícím pojmem, který se používá v kontextu moderních technologií současné doby je tzv. IoT a Smart zemědělství. Smart Farming je nově vznikající koncept, protože senzory IoT jsou schopny poskytovat informace o farmách a následně jednat na základě vstupu uživatele [26].

Některé přístupy jsou založeny na začlenění automatizace do různých aspektů chodu farmy. Většina funkcí je navržena tak, aby pokrývaly chod v průmyslové drůbežárně. Funkce systému automatizace farem, zahrnuje automatické spínání světla a použití obousměrných čítačů pro počítání počtu lidí vstupujících do farmy. U systému klimatizace se sleduje teplota, vlhkost a další parametry [22]. Systémy využívají senzory vlhkosti, který je zodpovědný za zvlhčování nebo odvlhčování drůbeže v závislosti na hodnotě vlhkosti. Dále se využívá vzdálené monitorování pomocí technologie GSM používané v mobilních telefonech. Lze se mnohdy dočíst, že parametry nejsou přesně regulovány podle specifikací drůbeže. To znamená, že poskytuje obecný přehled o automatizaci systému hospodaření jako celku. Nebývá dodržen doporučený rozsah teploty a vlhkosti. Dalším omezením bývá, že systém automatického spínání světla se používá pouze k počítání počtu lidí vstupujících nebo opouštějících místnost, což v případě potřeby nedává prostor pro stabilní osvětlení drůbeže [3]. Lze, ale také nalézt systémy, které uvedené nedostatky odstraňují. Uživatelé mohou získat informace o vnitřním prostředí drůbežářské farmy přijetím zprávy na registrované mobilní číslo. Systém může zahájit akci automaticky, aby nastavil parametry prostředí tehdy, když dojde k náhlé změně klimatu. Pomocí senzorů je monitorována a řízena i kontrola hladiny vody a mechanismus kontroly jídla. Jednotlivé přístupy se shodují na používaných senzorech. Příkladem mohou být např. LM35 teplotní senzor, SY-HS-220 senzor vlhkosti, MQ135 senzor používaný pro detekci plynného čpavku, MQ kouřový senzor, hladinový senzor pro monitorování hladiny vody. Signály ze senzorů jsou přiváděny do analogově-digitálního převodníku MCP3204, který je převádí do digitální podoby. Některé řízení embedded farem je založené na tzv. senzorových bezdrátových sítích. Opět se využívá uvedených senzorů, které ovšem odesílají měřená data bezdrátově do mikrokontroleru [24]. Některé zdroje také

uvádí, v jaké míře jsou jednotlivé senzory využívány co do počtu. Například se mnohdy jedná o odčítání teploty pouze z jednoho místa farmy a nemůže tak poskytnout průměrnou hodnotu [25]. V tomto ohledu jsou samozřejmě výhodné senzorické sítě, protože umístění jednotlivých snímacích prvků lze optimalizovat jednoduchým přemístěním na vhodnější místo.

Důležitou součástí automatizačního systému jsou monitorovací systémy. Existují různé přístupy v závislosti na tom, jaké komponenty jsou použity. Může se jednat pouze o vizualizaci aktuálních hodnot nebo u sofistikovanějších přístupů o systém se vzdáleným ovládáním. V aplikaci a využívání takovýchto monitorovacích systémů se v zásadě jedná o dva přístupy. Jeden vychází ze softwarového řešení v podobě řídicí aplikace umístěné na hlavním počítači, který provádí veškeré výpočty na základě získaných dat ze senzorů. K tomuto účelu je vhodné využít např. systém LabVIEW. Druhý přístup je založen na přístupu v podobě internetového rozhraní, které slouží pouze jako komunikátor s řídicím počítačem nebo mikrokontrolerem. Hojně se také využívá kontroly prostřednictvím SMS, resp. upozornění na změny nebo provedená nastavení jsou odesílány prostřednictvím textových zpráv. Ukazuje se, že většina existujících systémů, které nevyužívají implicitního řídicího systému v podobě systémů na klíč v rámci nějakého komplexního řešení, využívá jako řídicí jednotku buď technologie založené na mikrokontroléru Arduino, PIC nebo na Raspberry Pi. Ty pak slouží jako jádro pro zpracování odečítaných dat z připojených senzorů.

2.2.2 Klíčové veličiny

Pro lepší orientaci při návrhu řídicího systému je v této kapitole uveden přehled veličin, které je vhodné sledovat pro splnění specifických podmínek chovu drůbeže. Primárně lze vycházet z velkochovů:

- Dostatečný přísun vody.
- Přísun potravy.
- Sledování a regulace teploty uvnitř kurníku.
- Sledování vlhkosti uvnitř kurníku.
- Zajištění odebírání snášky.

Klíčovou podmínkou pro ekologický chov drůbeže je možnost volného pohybu zvířat. S tím souvisí některé odlišnosti od velkochovů:

- Zajištění kontroly nad pohybem drůbeže (myslí se, zda jsou zvířata uvnitř či venku kurníku).
- Zabezpečení před predátory.
- Možnost vzdálené kontroly (míní se tím např. počet zvířat apod.).
- Napájení ze solárních kolektorů.

Výše uvedené je zobrazeno v souhrnném schématu, které jednotlivé veličiny sdružuje do funkčních celků a poskytne tak základ pro přesné blokové schéma s konkrétními druhy senzoru a akčních členů v praktické části viz. Obrázek 1.



Obrázek 1: Souhrnné schéma sledovaných veličin

Ve schématu jsou rozděleny jednotlivé veličiny do celků, které jsou barevně odlišeny.

Mikroklima – sledování teploty, regulace teploty a vlhkosti, sledování vlhkosti.

Bezpečnost – ovládání vstupních dvířek, plašení predátorů, sledování počtu zvířat.

Potrava – přísun potravy, přísun vody.

Produkce – kontrola snášky.

Ovládání – napájení systému, vzdálená kontrola/řízení.

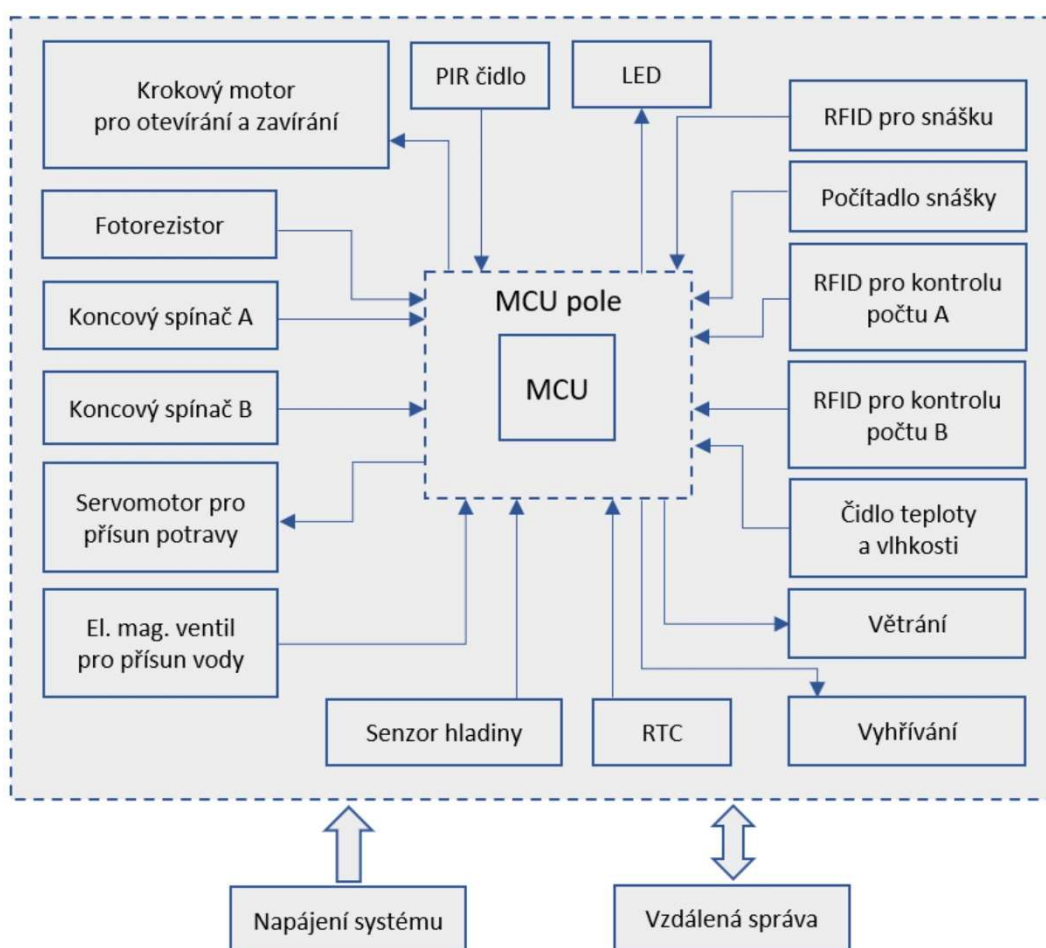
Řízení – všechny veličiny poskytují jednotce data nebo jsou jednotkou řízeny.

3 Praktická část

Praktická část předkládá kompletní řešení vycházející z analýzy realizované v teoretické části. Jedná se zde již o konkrétní návrh řídicího systému, definici konkrétních senzorů a akčních členů a výběr MCU jako řídicí jednotky. Na tyto fyzické části navazuje konkrétní návrh elektronického zapojení a programování.

3.1 Blokové schéma systému

Na základě identifikace sledovaných veličin, které vzešly z analýzy v teoretické části, lze navrhnout základní schéma systému – Obrázek 2. Zahrnuje bloky všech senzorů a akčních členů. Blokové schéma vychází ze schématu definovaného v kapitole 2.2.2.



Obrázek 2: Blokové schéma čidel a akčních členů

Mikroklima je zajištěno senzory teploty a vlhkosti které předávají MCU data pro zapnutí či vypnutí větrání v podobě stejnosměrného motoru s větrákem a vyhřívací fólie. Pro ovládání těchto akčních členů se využívá modul, který je připojen na digitální pin MCU.

Bezpečnost vyžaduje několik prvků. V první řadě je to ovládání dvířek, pro které je požadován senzor zjišťující aktuální denní světlo s analogovým výstupem. Dále dva koncové spínače pro limitní stavy otevřených a zavřených dvířek. Pohyb dvířek je realizován krokovým motorem ovládaným pomocí speciálního modulu, který vyžaduje digitální výstup z MCU. Ochrana proti predátorům je realizována prostřednictvím pohybového čidla a LED světla, které se rozsvítí v závislosti na pohybu před kurníkem v nočních hodinách. Pro kontrolu přítomnosti drůbeže v kurníku jsou k dispozici RFID čidla, která evidují počet kusů uvnitř a venku.

Pro kontrolu snášky se opět využívá RFID čidlo a pro počítání snášky čidlo, které může být například IR senzor. Tento senzor vytváří tzv. světelnou bránu. Po přerušení paprsku je indikován průchod vejce.

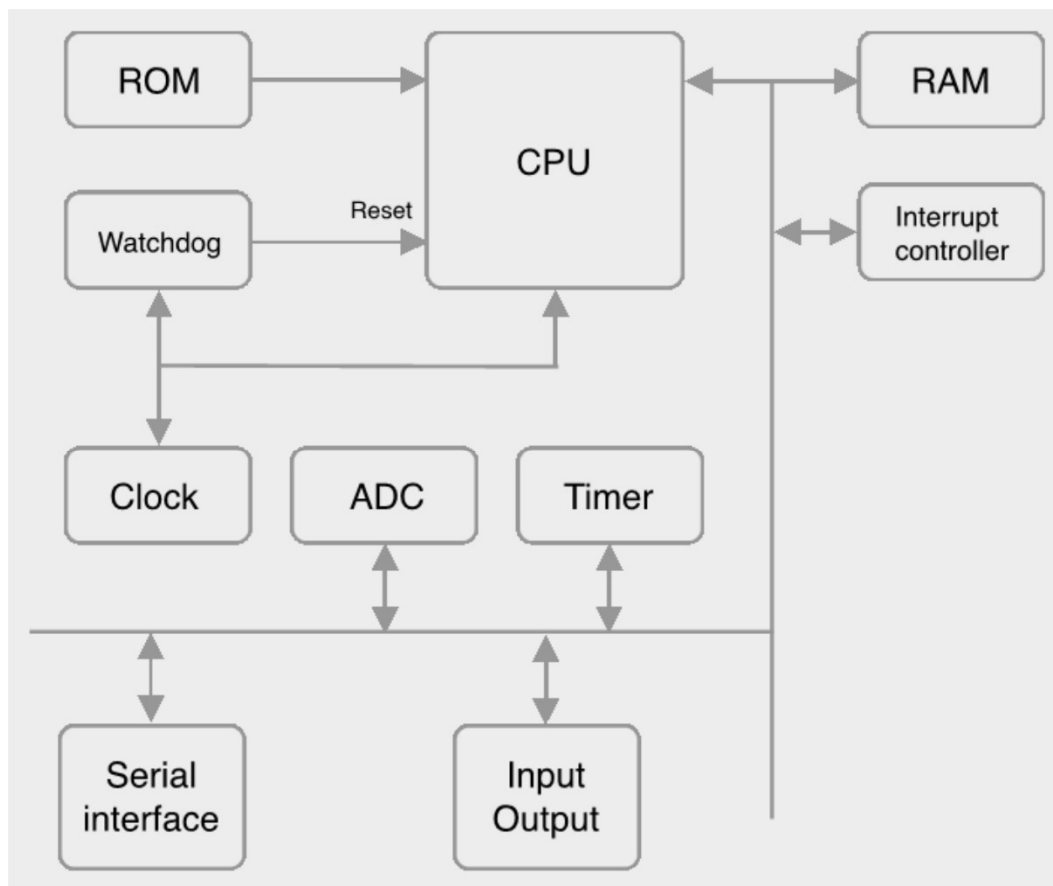
Pro přísun potravy je určen servomotor, který tvoří pohon pro automatickou záklopku, která provádí dávkování krmiva uloženého v zásobníku. Přísun vody probíhá prostřednictvím elektromagnetického ventilu ze zásobníku. Otevírání ventilu spolupracuje s čidlem množství vody.

3.2 Mikrokontroler

Mikrokontrolery jsou jedno-čipové počítače, které jsou určeny zejména pro řízení a automatizaci strojů a procesů [27, s. 4]. Mikrokontrolerem je malé zařízení, které provádí vyhrazené úkoly zadané uživatelem. Úkoly mohou souviset s normálními výpočty, jako je sčítání, odčítání, dělení a matematika s plovoucí desetinnou čárkou. V některých případech jsou často označovány jako „mikropočítače na čipu“. Stavebními kameny mikrokontroleru jsou procesor, paměť a periferní zařízení [28]. Jsou to v podstatě jednoduché miniaturní osobní počítače určené k ovládání malých funkcí větší komponenty bez složitého front-end operačního systému [29]. Pro řešení projektu autonomního kurníku, hrají mikrokontrolery zásadní roli. Tvoří jádro pro ovládání všech senzorů a aktualizátorů, proto je nutné věnovat jim větší pozornost. Na základě objasnění principů a součástí mikrokontrolerů bude zvolen odpovídající typ, který pokryje definované potřeby.

3.3 Architektura

MCU se skládá z podpůrného hardwaru, jako je CPU, hodiny, krystalový oscilátor, paměť a hardwarové periferie interně připojené – Obrázek 3. [28]



Obrázek 3: Blokové schéma MCU

Procesor – je v podstatě CPU centrální částí mikrokontroleru. Je to mozek, který následuje akci zadanou uživatelem. CPU načte instrukci z paměti (ROM), dekoduje ji a provede. Tato technika je známá jako pipelining. Zřetězení se provádí pomocí datové sběrnice a adresové sběrnice. CPU zpracovává instrukce rychlostí MegaHertz (MHz) nebo GigaHertz (GHz).

Hodiny – MCU vyžaduje hodiny k provedení jakékoli úlohy. Pro zajištění hodin mikrokontroleru je k CPU buzen krystalový oscilátor. Výběr krystalu určuje rychlost, se kterou chce mikrokontroler pracovat. Frekvenční rozsah krystalového oscilátoru je obecně v rozsahu MHz.

Registr – je prvek, který uchovává data. Ukládá binární slovo o délce 8 bitů. Mikrokontrolery jsou vybaveny různými obecnými a periferními registry. Mezi obecné registry patří PC a ukazatel zásobníku pro ukládání dat a instrukcí. Zatímco periferní registry jsou užitečné pro konfiguraci hardwaru v mikrokontroléru.

Vstupně-výstupní porty (I/O porty) - se běžně nazývají porty GPIO. To znamená, že tyto porty mohou být použity jako vstupní nebo výstupní. Některé mikro MCU podporují alternativní GPIO. Mohou být použity pro více funkcí. Jsou vyžadovány pro připojení externích zařízení, jako jsou rozhraní displeje (LED, LCD a dotyková obrazovka), motory atd. I/O porty se také používají pro účely snímání vstupu a přepínání.

Paměť – paměť je dvou typů (RAM a ROM). RAM je pro dočasné ukládání dat a ROM je pro trvalé ukládání. Pro uložení dat v mikrokontroleru je architektura procesoru rozdělena na architekturu Harvard a Von-Neumann. Harvard má samostatné paměťové sběrnice (adresová sběrnice a datová sběrnice) pro ukládání instrukcí a dat, zatímco Von-Neumann má společnou paměť pro instrukce i data [27, s. 9].

Časovače – jsou nejnovativnější periferie, které mohou produkovat přesné časové zpoždění. Může opakovat předem definovaný úkol ve známém časovém období. Některé z aplikací časovače zahrnují ovládání motoru, zapínání a vypínání reléového obvodu, GPS, ovládání domácích spotřebičů atd.

Čítač – je periferní zařízení, které počítá události probíhající mimo MCU. Aplikace čítačů zahrnují měření pulsů, detekci objektů atd. Používají se při měření frekvence, k počítání nárůstu nebo poklesu teploty v mikrovlnných troubách, k měření nastaveného času v pračkách a elektrických ohřívačích.

Sériová rozhraní – v elektronice spolu různá zařízení mluví pomocí sériové komunikace. K dosažení sériové komunikace se využívá rozhraní jako UART, I2C a SPI atd. V nejnovějších mikrokontrolerech se používají pokročilé protokoly jako Ethernet, USB.

Analogově digitální převodník – senzory převádějí fyzikální parametry, jako je teplota, vlhkost a tlak, na analogové signály. ADC převádí tento analogový signál na digitální bajty. Analogové signály mohou být ve formě napětí, proudu nebo odporu. Má vnitřní hodiny, které měří taktovací cykly napájené MCU a vzorkují vlastními hodinami. Počet hodinových cyklů představuje digitální reprezentaci analogového napětí.

Digitálně analogový převodník – je opakem ADC. DAC převádí digitální data na analogovou formu napětí. Některé z aplikací DAC zahrnují digitální zpracování signálu, ovládání motoru, hudební přehrávače, digitální potenciometr atd.

Přerušení – je událost, která přichází do hry, aby provedla více než jeden úkol. Když je voláno přerušení, mikrokontroler zastaví aktuální provádění úlohy (první úloha) a provede další úlohy (druhá úloha). Po dokončení druhého úkolu se vrátí k prvnímu úkolu a provede normální operace.

Kromě toho mikrokontroler řídí, zpracovává a ukládá informace do paměti. Velikost paměti se liší pro různé rodiny mikrokontrolerů.

3.4 Proč nepoužít mikroprocesor?

Rozdíl mezi mikrokontrolery a mikroprocesory se stal méně zřetelným, protože hustota a složitost čipu se stala relativně levnou na výrobu a mikrokontrolery tak integrovaly více funkcí „obecných počítačů“. Celkově však lze říct, že mikrokontrolery fungují užitečně samy o sobě, s přímým připojením k senzorům a akčním členům, kde jsou mikroprocesory navrženy tak, aby maximalizovaly výpočetní výkon na čipu, s interními sběrnici (spíše než přímými I/O) na podporu hardwaru, jako je RAM a sériové porty [29]. Přestože se mnohdy jeví, že mikrokontroler a mikroprocesor jsou stejným zařízením a někteří výrobci je i takto zaměňují, je vhodné uvést jejich rozdíly viz. Tabulka 1. [28].

Tabulka 1: Porovnání MCU a MPU

Mikrokontroler – MCU	Mikroprocesor – MPU
Využívá malý mikroprocesor, který běží na nižší frekvenci	Využívá univerzální centrální procesorovou jednotku (CPU), která běží na vyšší frekvenci
Pomalá rychlost provozu	Rychlejší v operacích.
Uvnitř jsou přítomny všechny periferie (časovač, čítač, rtc).	Některé součásti jsou mimo mikroprocesor.
Podporuje booleovské operace	Nepodporují booleovské funkce.
Přístup k registrům CPU vyžaduje méně času.	Přístup k registrům CPU vyžaduje více času.
Používá v aplikacích v reálném čase.	Používá se v osobních počítačích a noteboocích.
Podporuje RTOS (operační systém v reálném čase).	Podporuje RTOS a služby založené na jádře.
Příklady: 8051, PIC, MSP430, Renesas, mikrokontrolér STM atd.	Příklady: X86, Motorola, Broadcom, Pentium atd.
Vhodné pro provádění bitových operací.	Nepodporuje všechny typy bitových operací.
Ne vždy podporují multi-tasking.	Nejlépe se hodí pro multitasking.
Levnější	Vysoká cena

Na základě srovnání v tabulce je patrné, že hlavní výhody MCU jsou nižší cena a velikost. Propojení periferií je jednodušší pomocí I/O portů. Rychlost provádění požadovaných operací v rámci opakujících se instrukcí je rychlejší. Chceme-li ve své aplikaci použít jiný mikrokontrolér než procesor, potom klíčové výhody jsou:

- Ideální pro specializované aplikace.
- Jednorázově programovatelné.
- Minimalizace času pro vývoj aplikace.
- Praktická pozorování lze provádět pomocí simulátoru a emulátoru.
- Snadné navrhování a nasazení.

Na druhou stranu jejich velkou nevýhodou je, že nejsou schopny odolat vysokému napětí. Kromě výše uvedeného mají omezené množství paměti RAM a nejsou vhodné k provádění paralelních úloh současně.

Výběr mikrokontroleru je ovlivněno celou řadou parametrů. Odborníci uvádějí, že kritéria výběru zahrnují následující otázky:

- Jaké hardwarové periferie jsou vyžadovány?
- Je potřeba externí komunikace?
- Jaká architektura by měla být použita?
- Jaký druh komunity a zdrojů jsou pro mikrokontroler k dispozici?
- Jaká je dostupnost mikrokontroleru na trhu?

Také by se mělo vzít v úvahu rozhraní MCU a kolik paměti mají MCU, jeho rychlost a spotřebu energie [30].

3.5 Jak vybrat mikrokontroler

Výběr MCU je jedním z klíčových úkolů při návrhu řídicího systému pro autonomní kurník. Pro efektivní výběr se musí přihlídnout k různým kritériím, která se podílejí na výběru správného MCU. Není zcela vhodné začít s výběrem MCU dříve, než budou vyjasněny podrobnosti o systému [31].

3.5.1 Aplikace

Nezbytnou věcí, kterou je třeba udělat před výběrem MCU pro jakýkoli projekt, je důkladné pochopení úkolu, pro které má být MCU nasazen. Ve skutečnosti může být během tohoto procesu vytvořen list technických specifikací. Stejně tak pomáhá identifikovat konkrétní funkce MCU, které budou pro projekt použity. Může se také vypracovat blokové schéma, popřípadě vývojový diagram. Teprve potom bude k dispozici dostatek dat pro zahájení racionálního výběru MCU. Když je dosaženo tohoto bodu, existuje několik jednoduchých kroků, které lze provést, a ujistit se tak, že je vytvořen správný výběr MCU [32].

3.5.2 Hardwarové požadavky

Základní věcí, kterou je třeba zvážit při výběru MCU, je vědět, jaká hardwarová zařízení, popřípadě periferie budou s MCU propojeny. Je třeba si ujasnit, zda bude požadováno nějaké komunikační rozhraní, jako je UART. Zda bude potřeba Ethernet. Kolik bude využíváno sériových portů a zda bude také vyžadované USB. Zda se bude používat PWM. To jsou však jen některé základní otázky, které povedou k výběru odpovídajícího MCU z pohledu hardwaru.

3.5.3 Softwarové požadavky

Stejně jako je třeba zodpovědět otázky týkající se hardware, musí se zodpovědět některé otázky o software. Mezi ně například patří jaká je potřebná rychlost zpracování, jaké typy výpočtů jsou zahrnuty, jaký může být vyžadován výpočetní výkon, a jaká jsou časová omezení.

3.5.4 Architektura MCU

Pokud je již známá představa hardwarových a softwarových požadavků projektu, může se přistoupit k hledání různých typů MCU. MCU se dodávají v různých typech, které zahrnují 4bitové, 8bitové, 16bitové a 32bitové atd. Bitová velikost MCU představuje velikost „slova“ použitého v instrukční sadě MCU. To znamená, že v 8bitovém MCU je reprezentace každé instrukce, adresy, proměnné nebo registru 8bitová. Jedním z klíčových důsledků velikosti bitu je kapacita paměti MCU. Ve vestavných systémech jsou užitečné 32bitové MCU [32].

3.5.5 Požadavky na paměť

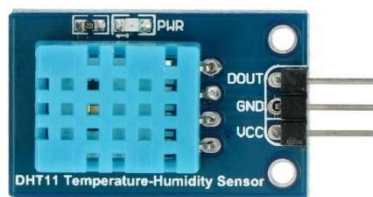
Tento požadavek je velmi důležitý. Musí se brát v úvahu, kolik paměti zabere programový kód a kolik se potřebuje RAM paměti. Je vždy lepší vybrat MCU s větší pamětí a mít rezervu, která se po konečném výběru může zredukovat volbou MCU s menší pamětí ve stejné rodině čipů. Pomocí softwarové architektury a komunikačních periférií obsažených v aplikaci lze odhadnout, kolik paměti flash a paměti RAM bude aplikace vyžadovat [33].

3.6 Výběr čidel a akčních členů

Přestože v kapitole 3.1 jsou již uvedeny předpokládané senzory a akční členy, které by měly být v systému použity, je nutné provést jejich výběr, aby se určilo množství a typ I/O pinů, které pomůže vybrat vhodný MCU.

3.6.1 Čidlo teploty a vlhkosti

Pro měření teploty a vlhkosti je vhodné použít senzor DHT11. Jedná se o základní, a levný digitální snímač teploty a vlhkosti. Využívá kapacitní snímač vlhkosti a termistor k měření okolního vzduchu. Disponuje digitálním datovým pinem. Jeho použití je poměrně jednoduché. Jedinou skutečnou nevýhodou tohoto senzoru je, že z něj lze získat nová data pouze jednou za 2 sekundy. To, ale v tomto projektu není nic omezujícího. Senzor DHT11 měří relativní vlhkost. Relativní vlhkost je množství vodní páry ve vzduchu vs. bod nasycení vodní páry ve vzduchu. V bodě nasycení začne vodní pára kondenzovat a hromadit se na površích za vzniku rosy. Bod nasycení se mění s teplotou vzduchu. Studený vzduch pojme méně vodní páry, než se nasytí, a horký vzduch pojme více vodní páry, než se nasytí. DHT11 detekuje vodní páru měřením elektrického odporu mezi dvěma elektrodami. Komponenta pro snímání vlhkosti je substrát zadržující vlhkost s elektrodami aplikovanými na povrch. Když je vodní pára absorbována substrátem, substrátem se uvolňují ionty, což zvyšuje vodivost mezi elektrodami. Změna odporu mezi dvěma elektrodami je úměrná relativní vlhkosti. Vyšší relativní vlhkost snižuje odpor mezi elektrodami, zatímco nižší relativní vlhkost odpor zvyšuje. Senzor měří teplotu pomocí NTC teplotního čidla (termistoru) zabudovaného do jednotky [34].



Obrázek 4: Čidlo teploty a vlhkosti

3.6.2 Koncové spínače

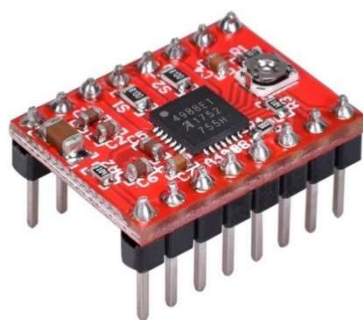
Koncové spínače vymezují limitní pozice při zavírání a otevírání dvířek kurníku. Jejich aktivací se vymezuje horní nebo dolní poloha. Při sepnutí vždy dojde k zastavení krokového motoru.



Obrázek 5: Koncový spínač

3.6.3 Ovladač krokového motoru

Pro ovládání krokového motoru je nutné použít speciální ovladač. To vychází z principu krokového motoru [35, s. 21]. Vhodným řešením je mikrokrokový ovladač A4988 pro ovládání bipolárních krokových motorů, který má vestavěný převodník. To znamená, že krokový motor lze ovládat pouhými 2 piny, nebo jedním pro ovládání směru otáčení a druhým pro ovládání kroků. Ovladač poskytuje pět různých rozlišení kroků: úplný krok, krok po ose, čtvrtkrok, osmikrok a šestnácti krok. Má také potenciometr pro nastavení proudového výstupu, tepelné vypnutí při přehřátí a ochranu proti překročení proudu.



Obrázek 6: Ovladač krokového motoru

3.6.4 Krokový motor

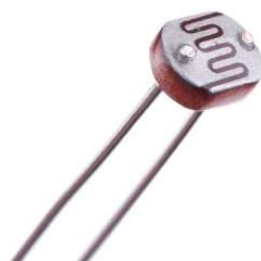
Přestože krokový motor je akčním členem v systému a není přímo připojen k MCU, ale prostřednictvím jeho ovladače, je nutné jej zde uvést, protože má pro tento projekt specifickou mechanickou konstrukci. Jedná se o motor s lineární osou se šroubovicí, která je přímo upevněna k motoru. Po šroubu se pohybuje upevňovací matice.



Obrázek 7: Krokový motor

3.6.5 Fotorezistor

Rezistor závislý na světle neboli fotorezistor, je snímač, jehož odpor klesá s rostoucím množstvím světla dopadajícího na něj. Když je tma, odpor fotorezistoru může být až několik M Ω . Když je však světlo, odpor fotorezistoru může být jen několik stovek ohmů. V závislosti na změně odporu bude docházet k automatickému ovládání dvířek kurníku. Vzhledem k tomu, že není požadována vysoká přesnost, lze využít například fotorezistor GM5539.



Obrázek 8: Fotorezistor

3.6.6 Senzor pohybu

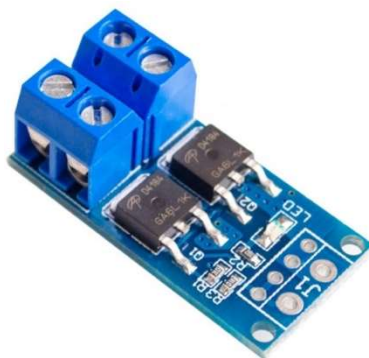
Pro senzor pohybu lze využít pasivní infračervené senzory, které dokážou detekovat pohyb objektů, vyzařující IR světlo. Proto je použití těchto senzorů pro detekci pohybu živočichů nebo jejich obsazení v bezpečnostních systémech velmi běžné. Infračervený senzor HC-SR501 je účinný, levný a nastavitelný modul pro detekci pohybu. Malá velikost a fyzický design tohoto modulu umožňuje jeho snadné použití. Výstup PIR senzoru pohybu lze připojit přímo k jednomu z digitálních pinů. Pokud je senzorem detekován jakýkoli pohyb, tato hodnota pinu bude nastavena na „1“. Dva potenciometry na desce umožňují nastavit citlivost a dobu zpoždění po detekci pohybu.



Obrázek 9: Senzor pohybu

3.6.7 Ovladač vyhřívání, větráku, LED

Přestože se jedná o tři zcela odlišné komponenty, lze pro jejich ovládání prostřednictvím MCU využít modul stejného typu, ale pro každý prvek zvlášť. Na ovládání těchto prvků se využívá PWM. Všechny uvedené komponenty vyžadují 12 V napájení. Nelze tedy použít přímé napojení z MCU, ale musí se využít rozhraní, které to umožňuje. Jako vhodný se jeví modul s obvodem MOSFET. Modul lze zapojit jako výkonový spínač nebo jako výkonový stupeň k regulátorům PWM. Zvoleným modulem je PWM MOS Driver 15 A 400 W. Jedná se o malý modul založený na dvou paralelně zapojených MOSFETech, které umožňují odesláním vstupního PWM signálu s maximální frekvencí 20 kHz ovládat stejnosměrné motory, stejnosměrná miničerpadla a další zátěže s napájením mezi 5 a 36 V DC a 15 A (max. 400 wattů).



Obrázek 10: Modul MOSFET PWM 15 A 400 W

3.6.8 Senzory RFID

Pro sledování přítomnosti drůbeže uvnitř nebo venku kurníku a kontrolu snášky, lze využít senzory RFID. RFID je identifikační systém, který k přenosu dat využívá elektromagnetické vlny na rádiové frekvenci. Jednoduchý RFID systém se skládá pouze ze dvou komponent: samotného štítku a zařízení pro čtení/zápis. Zařízení pro čtení/zápis se skládá z vysokofrekvenčního modulu a antény, která generuje vysokofrekvenční elektromagnetické pole, zatímco štítek je obvykle pasivní zařízení, které obsahuje mikročip, který ukládá a zpracovává informace. Pro identifikaci drůbeže se místo klasického štítku používají kroužky s čipem.

Jedním ze zařízení RFID je modul RDM6300 125 kHz EM4100. Je navržen pro čtení a zápis dat z tagů kompatibilních se 125 kHz. Bezkontaktní RFID deska řady RDM6300 využívá pokročilý obvod RF přijímače a vestavěnou architekturu MCU v kombinaci s vysoce účinným dekódovacím algoritmem pro čtení EM4100 a všech kompatibilních karet. K přenosu dat štítků RFID používá sériové TTL s přenosovou rychlostí 9600. Pro práci s touto RFID čtečkou lze použít jakýkoli MCU se sériovou periferií UART [36]. Výhodou tohoto modulu je externí anténa, takže aplikace tohoto modulu je variabilnější.



Obrázek 11: Modul RFID s tagy v podobě kroužků

3.6.9 IR senzor

Pro sledování snůšky je využíván kromě čipu RFID pro identifikaci drůbeže také IR senzor, který slouží jako počítadlo snůšky. Spíš, než o klasický IR senzor se jedná o senzor přerušení IR paprsku. Senzor reaguje na přerušení paprsku infračerveného světla vysílaného mezi vysílačem a přijímačem. Přijímač informuje o přerušení paprsku jiným objektem jeho osvětlením. Modul je vybaven 5 mm IR LED s vyzařovacím úhlem 20°. Používají se mimo jiné systémy přerušení paprsku v měřicích branách.



Obrázek 12: Senzor přerušení IR paprsku

3.6.10 Čidlo množství kapaliny

Pro zajištění pitného režimu drůbeže lze použít kombinaci snímače vody a elektromagnetického ventilu. Využít lze senzoru dešťové vody, který je tvořen deskou plošných spojů, na které se nachází detekční vodivé proužky. Pro napájení lze použít napětí v rozsahu 3,3 – 5 V. Proudový odběr se pohybuje v jednotkách miliampér, maximálně 20 mA.



Obrázek 13: Senzor snímání vodní hladiny

3.6.11 Elektromagnetický ventil

Pro doplňování vody na základě údajů senzoru kapaliny, je vhodné využít elektromagnetický ventil. Zde je jedno úskalí. Většina podobných zařízení vyžaduje pro svou funkčnost stálý tlak. Tento předpoklad vyžaduje čerpadlo, který stálý tlak udržuje. To přináší další konstrukční složitost a energetický výdaj. Vhodným řešením je přímý elektromagnetický ventil pro uzavírání průtoku kapalin a plynů s vnitřním průměrem 1/4. Tělo je vyrobeno z PVC odolávající teplotě 100 °C. Bez napětí cívky je ventil uzavřen. Vhodné pro rozpětí tlaku 0 ~ 0.8 MPa. Vzhledem k tomu, že pro funkčnost ventilu se vyžaduje napájení 12 V, je nutné využít MOSFET tranzistor, nebo modul, který je použit pro ovládání větráku, který je v tomto případě použit jako výkonový spínač.



Obrázek 14: Elektromagnetický ventil

3.6.12 RTC

Modul reálného času je doplňkem celého systému a slouží k časovým údajům v případě nedostupnosti internetového připojení. S časovými údaji se může propojit systém automatického ovládání dvířek nebo provázání s evidencí snášky. Standardem je modul hodin reálného času založený na DS3231. Má malou velikost, nízkou cenu a extrémně přesné hodiny reálného času sběrnice I2C. RTC poskytuje informace včetně sekund, minut, hodin, dne, data, měsíce a roku. Datum na konci měsíce se upraví automaticky pro měsíce kratší než 31 dní, včetně kalibrace na přestupný rok. Tento modul může pracovat buď ve 12hodinovém formátu s indikátorem AM/PM nebo ve 24hodinovém formátu.



Obrázek 15: Modul reálného času

3.6.13 Servomotor

Pro přísun potravy bude využit mechanismus, který bude založen na principu dávkování potravy ve stanovený čas v průběhu dne. Pro dávkování, resp. otevření záklopy pro přísun potravy bude využit servomotor.

Servomotor je lineární pohon nebo rotační pohon, který umožňuje přesné řízení lineární nebo úhlové polohy, zrychlení a rychlosti. Skládá se z motoru spojeného se snímačem pro zpětnou vazbu polohy. Vyžaduje také poměrně sofistikovaný ovladač, který bývá navržen speciálně pro použití se servomotory. Hlavním důvodem použití serva je to, že poskytuje úhlovou přesnost, tj. bude se otáčet pouze tolik, kolik je potřeba, a poté se zastaví a čeká na další signál, aby provedl další akci. Servomotor je na rozdíl od standardního elektromotoru, který se začne otáčet, po přivedení energie, a otáčení pokračuje, dokud se napájení nevypne. Nelze ovládat průběh otáčení elektromotoru, ale lze ovládat pouze rychlost otáčení, zapnutí a vypnutí [39].

Na trhu je nepřeberné množství servomotorů. Pro konkrétní použití v tomto projektu je, ale důležité, aby byl servomotor kvalitní. Tím se míní, že by minimálně jeho převody měly být kovové a ne plastové, které podléhají při frekventovaném použití degradaci. Dalším parametrem, který by se měl zohlednit je jeho síla, tj. kg na 1 cm. Vhodným zástupcem splňujícím požadavky tohoto projektu je např. Servo KONECT 21 kg Waterproof.



Obrázek 16: Servomotor

Tento modelářský servomotor využívá kombinované kovové a kompozitní pouzdro pro dosažení nižší váhy, zatímco stejnosměrný motor a ocelo/mosazné převody zajišťují vysoký výkon a trvanlivost. Navíc disponuje silou 21 kg/cm a je vodotěsné.

3.6.14 Shrnutí

Na základě výše uvedeného přehledu senzorů a akčních členů se identifikují požadavky na I/O vstupy. Přehled požadovaných typů vstupů a jejich množství je uveden v přehledové tabulce – Tabulka 2.

Tabulka 2: Přehled čidel a akčních členů s potřebným typem IO pinů

Komponenta	Typ vstupu
Čidlo teploty a vlhkosti	Digital
Koncové spínače	Analog – 2x
Ovladač krokového motoru	Digital
Senzor pohybu	Digital
Ovladač vyhřívání	Digital
Ovladač větráku	Digital
Ovladač LED	Digital
Fotorezistor	Analog
RFID	Digital – 3x
IR senzor	Digital
Elektromagnetický ventil	Digital
Čidlo množství kapaliny	Analog
RTC	Digital
Servomotor	Digital

3.7 Volba mikrokontroleru

Na základě definice blokového modelu celého systému a identifikaci všech senzorů a akčních členů lze provést výběr z několika platforem, které splňují definovaná kritéria. Z přehledové tabulky – Tabulka 2. je patrné, že je vyžadováno poměrně velké množství pinů. Celkem je zapotřebí 14 digitálních a 4 analogové piny. Při srovnání dostupných platforem, které se dají použít pro takto definovaný technický projekt se mohou využít dva přístupy. První by mohl být založen na kombinaci několika dílčích MCU, které by spolupracovaly prostřednictvím bezdrátové sítě a například cloudové technologie. Takovými MCU může být například

ESP8266, jehož součástí je i bezdrátové připojení prostřednictvím WiFi. Propojení jednotlivých modulů by muselo být realizováno prostřednictvím řídicího serveru. Toto řešení sebou přináší, poměrně vysoké finanční náklady a technické problémy týkající se zejména napájení.

Druhý přístup využívá platformu Arduino Mega, která poskytuje dostatečný výkon a počet digitálních a analogových I/O vstupů. Samotná platforma, ale nedisponuje možností bezdrátového připojení, a tudíž by musela být doplněna například o výše uvedený modul ESP8566. Kombinace těchto dvou platform vytváří odpovídající konfiguraci pro pokrytí definovaných požadavků.

Existuje, ale platforma, která je přímou kombinací uvedena v druhém přístupu. Jedná se o platformu RobotDYN, která sdružuje platformu ATmega 2560 a ESP8266 do jediné desky. Poskytuje tak dostatečný počet vstupů a přístup k ovládání prostřednictvím WiFi.

3.7.1 Platforma RobotDYN

Pokud se porovná platforma RobotDYN s klasickým ATmega 2560, tak kromě rozšiřujícího modulu ESP8266 má i další výhody, jak je patrné viz. Tabulka 3.

Tabulka 3: Porovnání platforem

Deska	Procesor	Rychlost CPU	Analogové In/Out	Digitální IO/PWM	Flash	USB	UART
ESP	ESP8266	80 MHz	1/0	17/17	512 kb	CH340G	1
MEGA	ATmega2560	16 MHz	16/0	56/15	256 kb	Klasické	4
RobotDYN	ATmega2560 ESP8266	16 MHz 80 MHz	16/0 1/0	54/15	32 Mb 8 Mb	CH340G	4 1/WiFi

Srovnání mezi Mega Arduinem s integrovaným ESP (Arduino Mega RobotDyn) a tradičním Mega Arduinem (Arduino Mega 2560) je vidět, že jsou si podobné, ale ve 2560 je klasické USB, což je velký konektor. V RobotDyn je k dispozici mini-USB. Je tedy především kompaktnější variantou, ale výkon je u obou stejný. Další rozdíl je, že ATmega má 32 MB paměti, nepočítaje ESP paměť. Tradiční Mega Arduino má pouze 256 kb paměti.

Napájení desky RobotDyn je 7 až 12 voltů a ESP8266 je již napájen přímo přes regulátor, takže není nutné dílčí napájení pro tento čip [37].



Obrázek 17: Deska Robot DYN

Na desce je umístěn DIP přepínač, díky kterému lze nastavit, v jakém vztahu budou jednotlivé mikroprocesory – Tabulka 4. Dále je k dispozici konektor pro PIG-TAIL anténu. Deska nabízí možnost vybrat, na které lince mají procesory komunikovat. Arduino Mega2560 nabízí až 3 UART. Deska má micro USB konektor pro programování, DC005 jack pro napájení a převodník CH340G.

Tabulka 4: Konfigurace DIP přepínačů

	1	2	3	4	5	6	7	8
CH340 připojit k ESP8266 (nahrát program)	OFF	OFF	OFF	OFF	ON	ON	ON	NOUSE
CH340 připojit k ESP8266 (připojit)	OFF	OFF	OFF	OFF	ON	ON	OFF	NOUSE
CH340 připojit k ATmega2560 (nahrát program)	OFF	OFF	ON	ON	OFF	OFF	OFF	NOUSE
CH340 připojit k Mega2560, COM3 připojit k ESP8266	ON	ON	ON	ON	OFF	OFF	OFF	NOUSE
Mega2560+ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF	NOUSE
Všechny moduly pracují samostatně	OFF	OFF	OFF	OFF	OFF	OFF	OFF	NOUSE

Vzhledem k tomu, že deska RobotDYN má dva nezávislé procesory, musí se programovat nezávisle, jako dva oddělené programy. Komunikace mezi ATmega a ESP8266 lze zajistit prostřednictvím linky Serial 3.

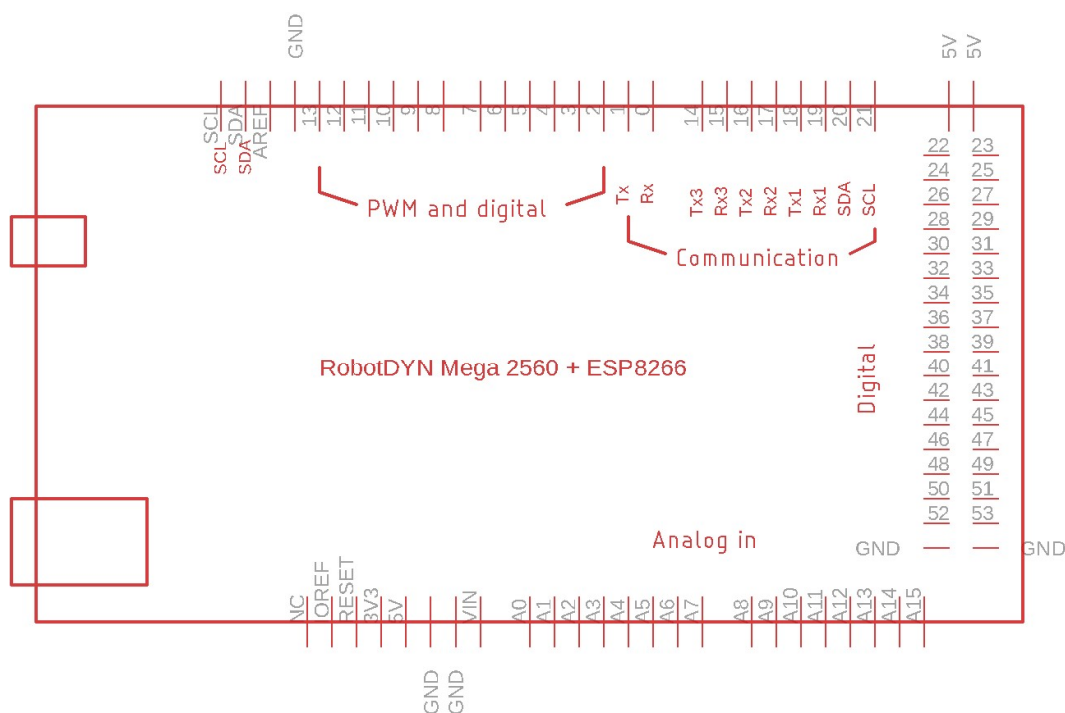
3.8 Elektronické zapojení

V předchozích kapitolách byly vybrány potřebné senzory, akční členy a MCU. Bude tedy využita deska založená na platformě Arduino. Jak je patrné z konstrukce desky RobotDYN, je k dispozici svorkovnice v podobě pinů, kterou lze využít k zapojení drátových propojek. Ty mohou být vyvedeny na nepájivé pole, které může obsahovat zapojené senzory, popřípadě vývody k akčním členům nebo vzdáleným sensorům. Tento přístup lze uplatnit v laboratorních podmínkách, pro prototypové zapojení. Pokud se má zohlednit umístění celého systému do externích podmínek, musí se jednat o řešení, které je kompaktní a odolné i hrubému zacházení. Tím je využití pinů desky pro napojení přídatného pole, které bude tvořit rozhraní mezi MCU a zapojenými senzory.

Pro vytvoření pole se musí nejdříve navrhnout schéma zapojení a následně vyrobit deska PCB. Pro tyto činnosti je využit program Autodesk Fusion 360 (Eagle) s modulem pro návrh elektronických schémat a tištěných spojů. Jedná se o profesionální program, který je pro studenty při prokázání studia zdarma.

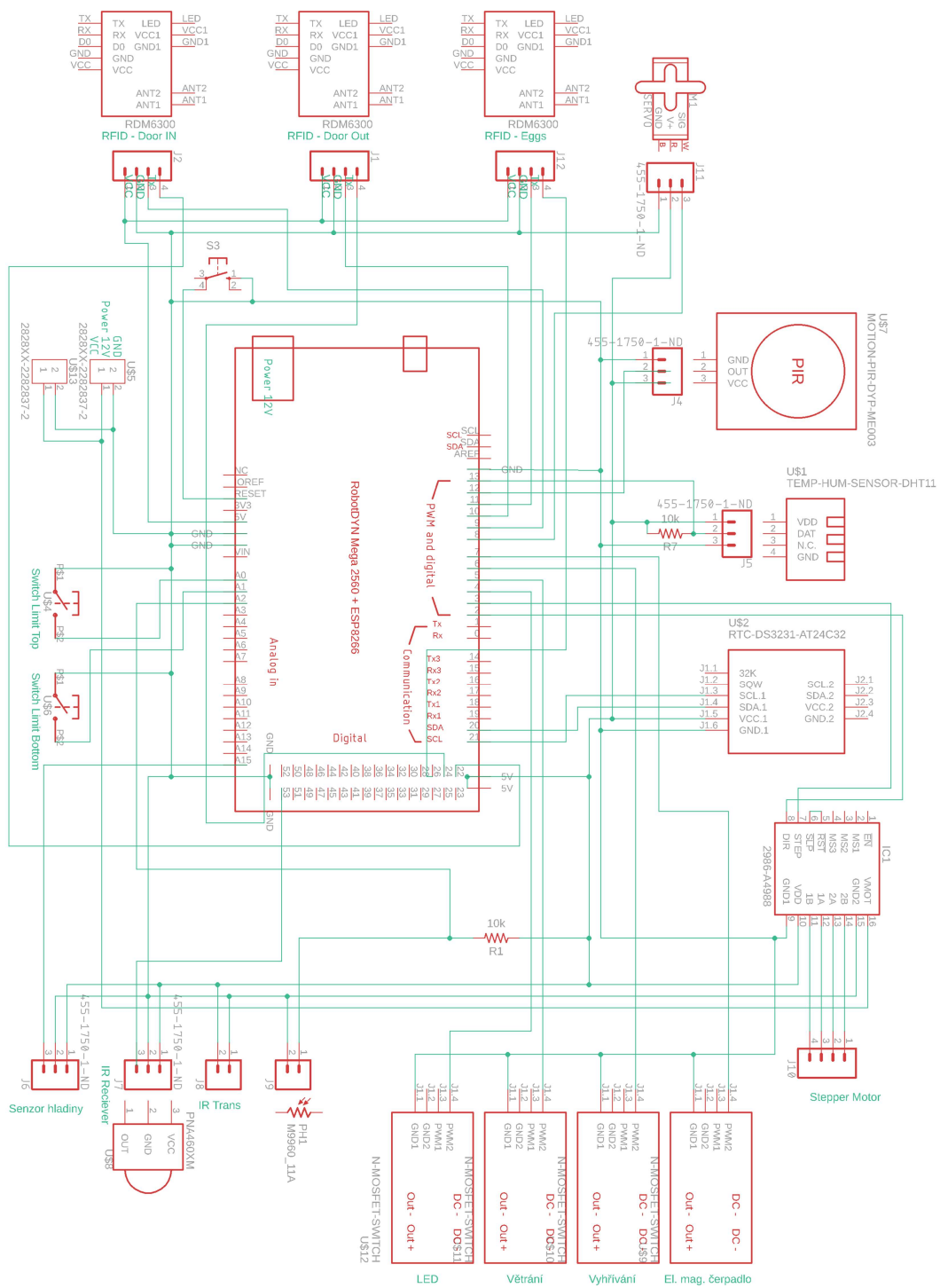
3.8.1 Schéma zapojení

Pro návrh schématu zapojení je nutné nainstalovat potřebné knihovny, které obsahují zvolené součástky a senzory. Pokud takové knihovny nejsou k dispozici je nutné tyto komponenty navrhnut přímo v nástroji pro plošné spoje. Takovou součástí je v první řadě samotný MCU. Existují některé knihovny, které obsahují desky platformy Arduino pro Eagle, ale deska RobotDYN mezi nimi není. Bylo tedy nutné tuto desku navrhnut jako novou komponentu pro návrhový software Eagle – Obrázek 18.



Obrázek 18: Návrh desky Robot DYN pro Eagle

Důležité bylo dodržet rozměry a rozložení pinů tak, aby finální podoba navrhovaného pole přesně odpovídala svorkovnici na desce RobotDYN. K této komponentě se následně připojily požadované senzory a akční členy. Přídavné pole nemusí odpovídat rozměrům, které má deska RobotDYN, což trochu usnadní návrh samotné PCB. Vzhledem k tomu, že většina senzorů je umístěna mimo pole, byly pro jejich napojení na pole použity konektory JST. Moduly pro ovládání LED, větrání, vyhřívání, elektromagnetické čerpadlo, RTC, řízení krokového motoru jsou zakomponovány přímo do pole, ale pro jejich zapojení je použita patice. Důvodem je snadná montáž a výměna v případě jejich poruchy. Předpokládá se, že napájení bude 12 V, což umožňuje využití akumulátoru, který může být dobíjen prostřednictvím fotovoltaiického panelu. Stejnou hodnotu napájení jsou také napájeny akční členy jako krokový motor, větrák, vyhřívání a ventil. Schéma elektronického zapojení je uvedeno na obrázku – Obrázek 19.

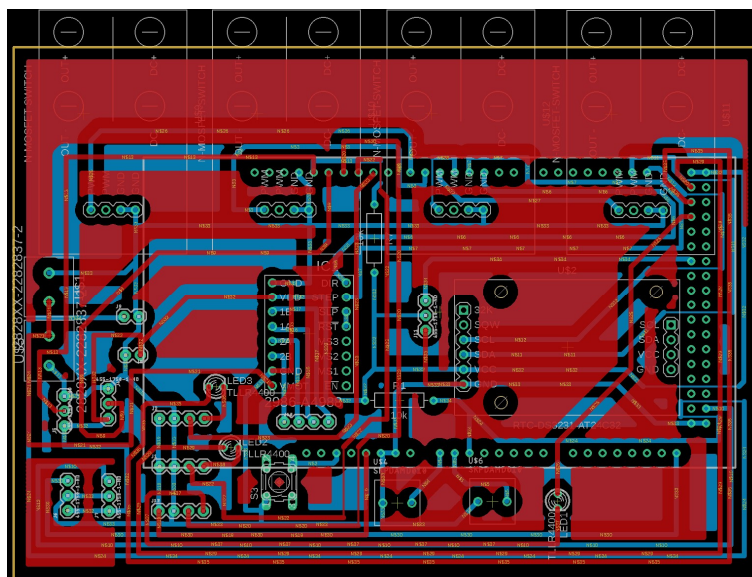


Obrázek 19: Schéma zapojení modulů pro řídicí jednotku

3.8.2 Návrh tištěného spoje

Po navržení elektronického schématu se v programu Eagle využije přepínače mezi schématem a deskou. Po prvotním přepnutí do módu návrhu plošného spoje jsou všechny součástky umístěny mimo návrhovou oblast. V první řadě se musí součástky rozmístit po návrhové oblasti. Velikost této oblasti lze měnit podle požadavků. Při návrhu PCB je logické, že jednotlivé cesty se nesmějí křížit. Složitost návrhu PCB se odvíjí od jeho typu cuprexitové desky. Míní se tím, zda se jedná o jednostrannou nebo oboustrannou desku, tj. zda se vodivá vrstva mědi nachází na jedné nebo na obou stránkách desky.

Pro návrh řídicího systému je využita oboustranná deska. Vytvoření spojů se realizuje pro dolní nebo horní vrstvu – Obrázek 20. Dolní vrstva je v návrhu reprezentována modrou barvou a horní vrstva červenou. Návrh plošného spoje se musí vygenerovat do formátu, který umožní jeho výrobu.



Obrázek 20: Deska plošného spoje

3.8.3 Výroba tištěného spoje

Pro výrobu plošných spojů, lze využít několik metod. Může se jednat o mokrou cestu, kdy se plošný spoj například leptá v chloridu železitém. Na vodivou plochu se nanese obraz plošného spoje. K tomu lze využít fotocestu prostřednictvím osvitové jednotky a následně se deska vloží do leptacího roztoku. Tam kde je spoj osvícen, zůstane vodivý materiál a ostatní se odleptá. Tato metoda vyžaduje velkou opatrnost, protože pracuje s chemikáliemi a někdy

může nastat tzv. podleptání. Zejména pokud je leptací roztok starší. Výroba mokrou metodou je také časově náročnější, protože obnáší několik technologických kroků.

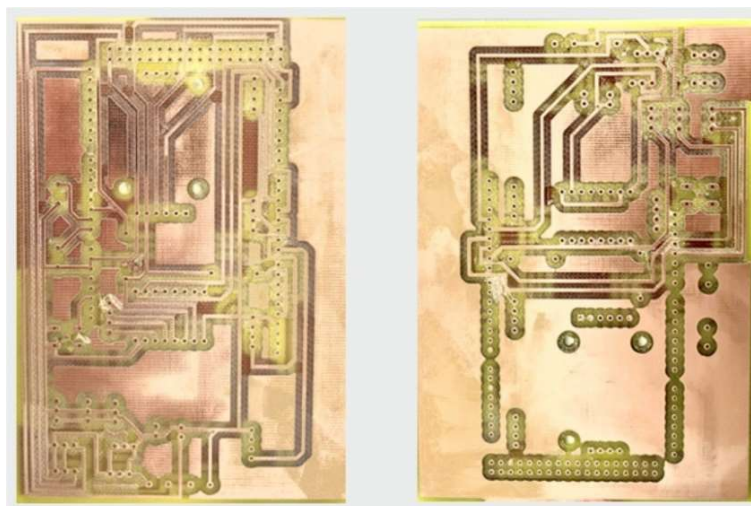
Druhá metoda je založena na mechanickém odstraňování nevodivých ploch plošného spoje prostřednictvím speciální CNC frézy. Tato metoda je výhodná pro prototypové konstrukce plošných spojů a s využitím moderních strojů je vysoce přesná. Kromě samotného frézování vodivých spojů, tyto stroje umožňují zcela automaticky realizovat další technologické fáze, jako je postprodukční ošetření, nanesení popisů apod. Pro výrobu plošného spoje řídicí jednotky bylo využito zařízení Protomat E 44 – Obrázek 21.

Hlavní oblastí využití je ve frézování struktur v poměděném materiálu desek plošných spojů, vrtání otvorů nebo frézování jednotlivých desek plošných spojů větších panelů. Toto zařízení má své přednosti zejména při výrobě menšího množství desek a při prototypové výrobě. Pro ovládání zařízení je dodáván speciální software - LPKF CircuitPro CAM, který nabízí rozsáhlý přístup ke všem parametrům procesu. Při výrobě PCB pracuje se standardním formátem soubor Gerber. Soubory se naimportují do programu a přiřadí se jim odpovídající vrstvy zpracování. Software pak provede uživatele všemi kroky výrobního procesu.



Obrázek 21: Zařízení pro výrobu plošného spoje Protomat E 44

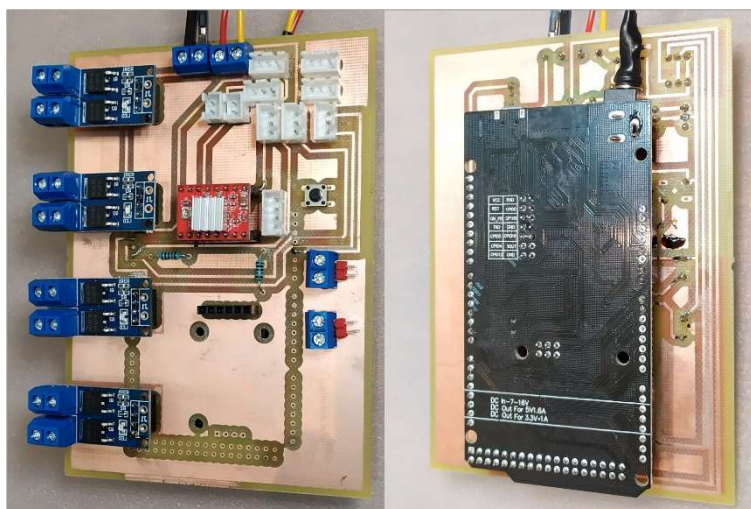
Zařízení disponuje polohovací kamerou, která je klíčová pro výrobu oboustranných PCB, protože zajišťuje přesné zaměření desky. Výsledek vyrobeného PCB je vidět na obrázku – Obrázek 22.



Obrázek 22: Vyrobená oboustranná PCB deska

3.8.4 Osazení tištěného spoje

Tištěný spoj byl osazen zejména konektory JST pro připojení externích čidel a paticemi, které umožňují rychlou výměnu modulů jako jsou PWM spínače, RTC, nebo řídicí jednotka pro krokový motor. Přestože zvolená technologie výroby PCB je velmi rychlá a pro účely této práce také dostupná, jako problém při osazování se ukázala volba oboustranného PCB. Důvodem byl problém s prokovením otvorů. Moderní linky pro výrobu PCB technologií prokovení disponují, ale při výrobě této desky ještě nebyla potřebná technologie dostupná. Nicméně tištěný spoj byl osazen a vhodným pájením se podařilo všechny komponenty propojit podle návrhu, jak byl definován v kapitole 3.8.1. Spíše, než funkční problémy jsou patrné designové nedokonalosti.



Obrázek 23: Osazený tištěný spoj

3.9 Programování

Programování desky RobotDYN probíhá pro každou platformu, tj. Mega a ESP odděleně. Pro přehlednější správu programového kódu obou platforem se využívá software od společnosti Microsoft – Visual Studio Code, který umožňuje doinstalovat modul PlatformIO. Pro programování uvedených MCU se používá jazyk C a C++. Jak je uvedeno obě platformy na desce se programují odděleně, ale musí mezi sebou vzájemně komunikovat. Platformy desky RobotDYN mohou mezi sebou komunikovat pomocí sériových linek. Tabulka 5 ukazuje srovnání platforem Arduino, které potvrzuje výběr MCU pro tento projekt.

Tabulka 5: Sériové linky desek Arduino

Deska	Serial	Serial1	Serial2	Serial3
Uno, Nano, Mini	0(RX), 1(TX)			
Mega	0(RX), 1(TX)	19(RX), 18(TX)	17(RX), 16(TX)	15(RX), 14(TX)
Leonardo, Micro, Yún		0(RX), 1(TX)		
Uno WiFi Rev.2	USB	0(RX), 1(TX)	NINA	
MKR boards		13(RX), 14(TX)		
Zero	Prog. port	0(RX), 1(TX)		
Due	0(RX), 1(TX)	19(RX), 18(TX)	17(RX), 16(TX)	15(RX), 14(TX)
101		0(RX), 1(TX)		

Pro sériovou komunikaci se používají piny TX/RX, tedy využívá se logické úrovně TTL (5 V nebo 3,3 V v závislosti na desce). Tyto piny by se neměly připojovat přímo k sériovému portu RS232, který pracuje s +/- 12 V a může tak poškodit desku. Sériové rozhraní se používá pro komunikaci mezi deskou Arduino a počítačem nebo jinými zařízeními. Všechny desky Arduino mají alespoň jeden sériový port (také známý jako UART nebo USART): Serial. Komunikuje na digitálních pinech 0 (RX) a 1 (TX) a s počítačem přes USB. Pokud se tedy využívá tato funkce, nelze zároveň použít piny 0 a 1 pro digitální vstup nebo výstup [40].

Ke komunikaci s deskou Arduino se využívá vestavěný sériový monitor, který je součástí IDE pro Arduino. Standardní Arduino má jeden hardwarový sériový port, ale sériová komunikace je také možná pomocí softwarových knihoven pro emulaci dalších portů (komunikačních kanálů) pro zajištění konektivity k více než jednomu zařízení. Softwarový sériový port vyžaduje velkou pomoc od řadiče Arduino k odesílání a přijímání dat, takže není tak rychlý

a efektivní jako hardwarový port. Tabulka 5 ukazuje, že Arduino Mega má čtyři hardwarové sériové porty, které mohou komunikovat až se čtyřmi různými sériovými zařízeními. Pouze jeden z nich má vestavěný USB adaptér (lze připojit USB-TTL adaptér k jakémukoli jinému sériovému portu). Tyto vlastnosti má i deska RobotDYN a pro předkládaný projekt se také využívají [41]. Projekt se tedy skládá ze dvou částí: za prvé programování ESP8266 a za druhé programování Mega2560.

3.9.1 Příprava MCU pro programování

Aby mohla probíhat komunikace mezi platformou Mega a ESP je nutné, aby druhá z uvedených platform umožňovala pracovat s AT příkazy. Způsob, jakým je toho dosaženo je flashováním čipu ESP8266 s firmwarem AT.

Pro flashování je určený nástroj *esptool*, který je spolu s *espefuse* a *espsecure* určen pro kompletní práci s čipy *Espressif*.

Pro začátek je nutné si stáhnout a rozbalit Firmware AT verze 1.7.4.0. Nyní je nutné uvedení desky RobotDYN do módu pro připojení ESP a nahrávání kódu. To se provede přepnutím DIP přepínače do následující konfigurace:

	1	2	3	4	5	6	7	8
CH340 připojit k ESP8266 (nahrát program)	OFF	OFF	OFF	OFF	ON	ON	ON	NOUSE

Deska se připojí k počítači prostřednictvím USB, čímž se zjistí, na jakém portu komunikuje a prostřednictvím příkazového řádku se provede příkaz pro flashování čipu:

```
esptool.py write_flash --flash_mode dio --flash_size 2MB-c1 0x0
bin/boot_v1.7.bin 0x01000 bin/at/1024+1024/user1.2048.new.5.bin
0x1fb000 bin/blank.bin 0x1fc000 bin/esp_init_data_default_v08.bin
0xfe000 bin/blank.bin 0x1fe000 bin/blank.bin
```

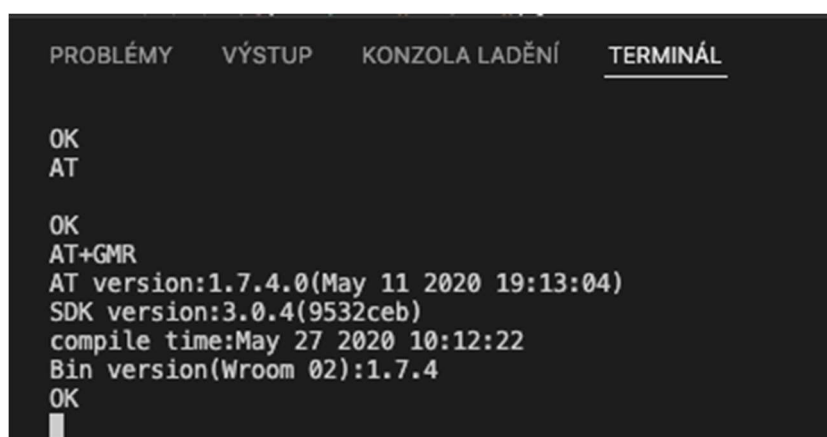
Po vykonání uvedeného příkazu se objeví text - *Hard resetting via RTS pin....* Po zobrazení, se deska musí odpojit a DIP přepínače nastavit do konfigurace pro připojení ESP:

	1	2	3	4	5	6	7	8
CH340 připojit k ESP8266 (připojit)	OFF	OFF	OFF	OFF	ON	ON	OFF	NOUSE

V sériovém monitoru by se měly pro ověření zadat testovací příkazy, zda opravdu proběhlo flashování:

1. AT (testuje spuštění)
2. AT + GMR (kontroluje informace o verzi)

Pokud proběhlo flashování v pořádku, na výstupu se objeví informace o úspěšném vykonání příkazu – Obrázek 24 [42]. Pro otestování lze použít i další AT příkazy [43].



```

PROBLÉMY  VÝSTUP  KONZOLA LADĚNÍ  TERMINÁL

OK
AT

OK
AT+GMR
AT version:1.7.4.0(May 11 2020 19:13:04)
SDK version:3.0.4(9532ceb)
compile time:May 27 2020 10:12:22
Bin version(Wroom 02):1.7.4
OK

```

Obrázek 24: Výstup AT příkazů

3.9.2 Webový server pro provoz aplikace

Přestože se hovoří o autonomním systému pro chov drůbeže, lze předpokládat, že vzhledem k velkému množství senzorů a akčních členů je nutností, aby existovala možnost manuálního ovládání a kontroly. Vzhledem ke konstrukci zvolené desky se dvěma nezávislými MCU se nabízí možnost využít vytvoření webového serveru na MCU ESP8266. Konstrukce programu webového serveru se odvíjí od typu přístupu k ovládací aplikaci. Pokud bude postačující pouze lokální přístup prostřednictvím interní sítě vytvořené ESP, pak se aplikuje kód pro vytvoření přístupového bodu. Pokud by mělo být připojení vzdálené, potom se vytvoří konektivita k existující síti. Základní programový kód pro zprovoznění webového serveru je následující:

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266mDNS.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>

const char* ssid = "PoultryApp";
const char* password = "12345678";

const char* type = "global"; // local nebo global

AsyncWebServer server(80);
MDNSResponder mdns;

const char index_html[] PROGMEM = R"rawliteral(
    Web Server běží.
)rawliteral";

void setup() {
    Serial1.begin(115200);
    Serial.begin(115200);
    if(type=="local"){
        WiFi.softAP(ssid, password);
        IPAddress IP = WiFi.softAPIP();
    }else{
        WiFi.begin(ssid, password);
        while (WiFi.status() != WL_CONNECTED) {
            delay(1000);
            Serial.print(".");
        }
        IPAddress IP = WiFi.localIP();
    }
    Serial.print("AP IP address: ");
    Serial.println(IP);

    if (mdns.begin("esp8266", WiFi.localIP())) {
        Serial.println("MDNS responder started");
    }
}

```

```

server.begin();
Serial.println("HTTP server started");

server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html);
});

}

void loop() {

}

```

Pro nahrání uvedeného programového kódu se musí nastavit DIP přepínač na desce pro komunikaci s ESP:

	1	2	3	4	5	6	7	8
CH340 připojit k ESP8266 (nahrát program)	OFF	OFF	OFF	OFF	ON	ON	ON	NOUSE

Jestliže je kód nahrán, musí se přepnout DIP přepínač do následující polohy:

	1	2	3	4	5	6	7	8
CH340 připojit k ESP8266 (nahrát program)	OFF	OFF	OFF	OFF	ON	ON	OFF	NOUSE

Po otevření sériového monitoru se vypíše IP adresa, přes kterou se lze připojit k webovému serveru na ESP. Pro přístup k aplikaci, musí být zařízení (telefon, počítač, tablet) připojen ke stejné síti, jako je připojena deska RobotDYN.

3.9.3 Nastavení ESP 8266 jako přístupový bod

Pro běh aplikace byl zvolen přístup asynchronního serveru, který umožňuje volání požadavků bez nutnosti obnovy aplikační stránky. Ve výše uvedeném kódu je implementována jak možnost vytvoření přístupového bodu, tak připojení k existující WiFi síti.

Pro zprovoznění serveru je důležité vytvoření objektu `AsyncWebServer` na portu 80.

```
AsyncWebServer server(80);
```

V sekci `setup()`, se využívá metoda `softAp()`, pro nastavení přístupového bodu, kde proměnné `ssid` je uveden název sítě, pod kterou je bod přístupný a `password` je přístupové heslo.

```
WiFi.softAP(ssid, password);
```

Pro získání IP adresy přístupového bodu se využívá `softAPIP()`, která se odešle na sériový monitor.

```
IPAddress IP = WiFi.softAPIP();  
Serial.print("AP IP address: ");  
Serial.println(IP);
```

3.9.4 Konstrukce serverové aplikace

Serverová aplikace je umístěna na platformě ESP a je ve formě html stránky. Tato stránka obsahuje kód javascriptu, který zajišťuje aktualizaci pozice tlačítek a hodnot čidel:

```
<script>  
function toggleCheckbox(element) {  
    var xhr = new XMLHttpRequest();  
    if(element.checked) {  
        xhr.open("GET",  
            "/update?action=update&output="+element.id+"&state=1",  
            true);  
    }else{  
        xhr.open("GET",  
            "/update?action=update&output="+element.id+"&state=0",  
            true);  
    }  
    xhr.send();  
}  
  
document.addEventListener('DOMContentLoaded', function (event) {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {
```

```

var lastStateArr = this.responseText.split("\n");
var stateArr = lastStateArr[lastStateArr.length-2].split("|");
for(var i=1;i<stateArr.length;i++){
    if(stateArr[i]==1){
        document.getElementById(i).checked = true;
    }else{
        document.getElementById(i).checked = false;
    }
}
};
xhttp.open("GET", "/state?action=getstate&output=0&state=0", true);
xhttp.send();
});
</script>

```

Funkce `toggleCheckbox()` vytvoří požadavek na různé adresy URL pro zapnutí nebo vypnutí zařízení. Odeslání požadavku se děje přes adresu `/update?action=update&output="+element.id+"&state=1`:

```

if(element.checked){
    xhr.open("GET",
"/update?action=update&output="+element.id+"&state=1", true);
}

```

V rámci adresy se odesílají tři parametry:

- `action` – identifikuje událost, podle které se rozhodne, co má v platformě Mega nastat,
- `output` – číslo, které identifikuje, jaké tlačítko bylo stisknuto a tím také ukazuje, jaké zařízení, resp. její stav bude aktualizován,
- `state` – logická hodnota, kde 1 - znamená zapnuto, 0 - vypnuto.

Aby byl stav výstupu na webovém serveru aktualizovaný, volá se druhá funkce, která při svém prvním spuštění nastaví tlačítka do stavu, který odpovídá uloženým hodnotám na platformě Mega. Je to důležité např. z důvodu výpadku spojení mezi aplikací a serverem. Tento skript

potom zajišťuje získání aktuálních hodnot. Následující programové kódy zajišťují aktivitu ESP8266, pokud obdrží požadavky na uvedené URL adresy. Když je požadavek přijat na root/URL, odešle se HTML stránka.

```
server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request) {  
  request->send_P(200, "text/html", index_html, processor);  
});
```

Následující řádky zkontrolují, zda byl obdržen požadavek na URL /update?action=update&output="+element.id+"&state=1 nebo /update?action=update&output="+element.id+"&state=0.

Na sériový monitor se zašle hodnota parametru v podobě inputMessage. Důležitá je skladba, resp. obsah proměnné inputMessage. Hodnoty jednotlivých parametrů bude číst ze sériového vstupu program v Mega 2560. Pokud by se nepoužívala druhá platforma, tak by bylo možné již rovnou poslat hodnotu na digitální vstup. Ale to by bylo možné pouze v případě, že je zařízení připojeno k ESP8266.

```
server.on("/update", HTTP_GET, [] (AsyncWebServerRequest *request) {  
  String inputMessage;  
  if (request->hasParam(PARAM_INPUT_1) && request->  
    >hasParam(PARAM_INPUT_2) && request->  
    >hasParam(PARAM_INPUT_3)) {  
    inputMessage = request->getParam(PARAM_INPUT_1)-  
      >value()+"|"+request->getParam(PARAM_INPUT_2)  
      >value()+"|"+request->getParam(PARAM_INPUT_3)-  
      >value();  
  }else{  
    inputMessage = "For UPDATE message no sent";  
  }  
  Serial.println(inputMessage);  
  request->send(200, "text/plain", "OK");  
});
```

Když je přijat požadavek na URL /state, odešle se aktuální stav výstupu:

```
server.on("/state", HTTP_GET, [] (AsyncWebServerRequest *request) {  
  request->send(200, "text/plain", String(Serial.read()). c_str());  
});
```

Jediný způsob jak přijímat data ze sériové linky z Mega2560, který na platformě RobotDYN spolehlivě funguje je využití funkce `serialEvent()`. Všechny ostatní způsoby příjmu dat, jejich šifrování a další zpracování selhaly.

```
void serialEvent() {
    while (Serial.available()) {
        char inChar = Serial.read();

        inString += inChar;
        if (inChar == '\n') {
            String output = getValue(inString, '|', 0);
        }
    }
}
```

Funkci `serialEvent()` přijímá jednotlivé hodnoty z link `Serial` až do té doby, kdy přijde znak nového řádku - `"\n"`. Ten signalizuje, že řetězec, který obsahuje identifikaci akce, stavu a hodnoty je celý a lze jej rozložit na dílčí hodnoty. Poslední důležitou částí je získání hodnot z řetězce. Při obousměrné sériové komunikaci vždy dochází k zaslání řetězce, který obsahuje kombinaci názvu události a stavů, oddělených symbolem roury - `"|"`. K rozdělení tohoto řetězce se využívá funkce `getValue`, která podle uvedeného znaku rozloží řetězec a vrátí hodnotu podle požadované pozice.

```
String getValue(String data, char separator, int index){
    int found = 0;
    int strIndex[] = {0, -1};
    int maxIndex = data.length()-1;
    for(int i=0; i<=maxIndex && found<=index; i++){
        if(data.charAt(i)==separator || i==maxIndex){
            found++;
            strIndex[0] = strIndex[1]+1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
    return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}
```

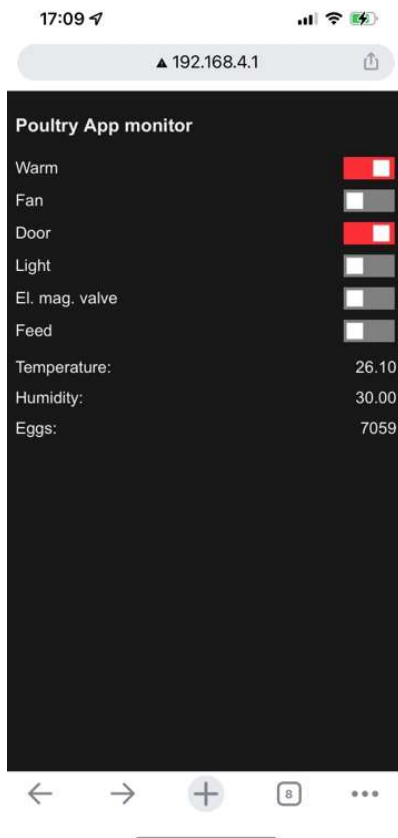
Pro přijímání aktuálních údajů ve směru od platformy Mega k ESP, například při získávání dat teploty, vlhkosti a počtu vajec je konstrukce programového kódu následující:

```
setInterval(function ( ) {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            var lastStateArr = this.responseText.split("\n");  
            var stateArr = lastStateArr[lastStateArr.length-  
                2].split("|");  
            if(stateArr[0]=="datafrommega" && stateArr[1]!="Err"){  
                if(stateArr[1]!="Err"){  
                    document.getElementById("temperature").innerHTML =  
                        stateArr[1];  
                    document.getElementById("humidity").innerHTML =  
                        stateArr[2];  
                }  
                document.getElementById("eggs").innerHTML =  
                    stateArr[3];  
            }  
        }  
    };  
    xhttp.open("GET", "/getdata?action=getdata", true);  
    xhttp.send();  
}, 10000 );
```

Odesílání požadavku na URL `getdata` se odehrává opakovaně podle definovaného časového intervalu. V uvedeném příkladu je to 10 sekund. Pokud je tedy zapnutá serverová aplikace například v mobilním telefonu, každých 10 sekund se odešle požadavek pro získání aktuálních dat. Struktura odesílaných a přijímaných dat mezi jednotlivými platformami je v podobě řetězce, kde jednotlivé hodnoty jsou odděleny oddělovníkem – rourou (`()`). Data jsou následně rozparsována pomocí funkce `getValue()`. Při ovládání akčních prvků je prvním parametrem řetězce identifikace, jaká událost se má vykonat. Následuje hodnota prvku, který má být ovládán a poslední je hodnota, na kterou se má prvek nastavit. V případě získávání dat je prvním parametrem identifikace a následuje seznam hodnot ze senzorů oddělených opět rourou.

3.9.5 Uživatelské rozhraní serverové aplikace

Jak bylo již zmíněno v předchozí kapitole, serverová aplikace má podobu HTML stránky, kde jednotlivé ovládací prvky zasílají asynchronní požadavky na konkrétní adresu. Vzhled stránky je uveden na obrázku – Obrázek 25.



Obrázek 25: Uživatelské rozhraní serverové aplikace

Aplikace je složena z přepínačů, které jsou určeny pro ovládání akčních prvků a oblastí pro zobrazení dat ze senzorů. Uživatelské rozhraní je pojato zcela funkcionálně a neřeší grafickou podobu. Vzhled lze měnit prostřednictvím jednoduchých úprav serverové aplikace, a to prostřednictvím kódu html a css. Přidávání dalších ovládacích prvků nebo informací ze senzorů vyžaduje vždy namapování na konkrétní funkci v Mega, která zajišťuje odpovídající akci.

3.10 Programování čidel a akčních členů

Prostřednictvím aplikace umístěné na webovém serveru, lze ovládat akční členy a zobrazovat stavy připojených senzorů. Realizace vstupně výstupních událostí zajišťuje platforma Mega. Ta jednotlivé události vyvolá automaticky na základě napsaných podmínek nebo manuálním vyvoláním prostřednictvím aplikace z platformy ESP.

3.10.1 Ovládání dveří

Ovládání otevírání a zavírání dveří by mělo být řízeno automaticky nebo ručně z aplikace. Kontrola automatického ovládání dveří je ovlivněna hodnotami fotorezistoru, který poskytuje hodnoty o aktuálních světelných podmínkách. Jestliže světelné podmínky dosáhnou hodnoty pro otevření dveří, aktivuje se krokový motor a dveře budou otevřeny. Krokový motor bude dveře otevírat do doby, než se aktivuje koncový spínač, který pohyb dveří zastaví. Tento postup je aplikovaný i pro zavírání dveří. Stejný princip je zajištěn i prostřednictvím aplikace, kde se otevírání a zavírání dveří děje pomocí odpovídajícího tlačítka.

Otevírání a zavírání dveří zajišťují dvě funkce:

`goStepperUp()` a `goStepperDown()`:

```
void goStepperUp() {
    digitalWrite(dirPin, HIGH);
    do{
        valEndStopUp = analogRead(endStopUp);
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }while (valEndStopUp<=500);
    isOpened=1;
}

void goStepperDown() {
    digitalWrite(dirPin, LOW);
    do{
        valEndStopDown = analogRead(endStopDown);
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }while (valEndStopDown<=500);
    isOpened=0;
}
```

Volání oboru funkcí se děje buď přímo, prostřednictvím aplikace, jejíž konstrukce je uvedena v kapitole 0 nebo automaticky podle hodnot světelných podmínek:

```
void loop() {  
    lightLevel = analogRead(photoResistorPin);  
    if(isOpened==0 && lightLevel>=dayTime){  
        goStepperUp();  
    }  
    if(isOpened==1 && lightLevel<=nightTime){  
        goStepperDown();  
    }  
    delay(200);  
}
```

3.10.2 Evidence drůbeže

Pro evidenci drůbeže byla zvolena technologie RFID čipů. Samotná implementace jedné vstupní brány je poměrně jasná, ale při implementaci více snímačů již problém nastane. Jedná se o to, že snímač komunikuje s řídící deskou přes sériové vstupy, ale ty v tomto projektu jsou využívány pro komunikaci mezi platformami Mega a ESP. Pro evidenci drůbeže nemusí být implementována obousměrná komunikace, ale stačí pouze komunikace ve směru od čidla k desce.

K tomu slouží jakýkoliv digitální vstup, protože se vytvoří softwarový port. Ten potom čte identifikační tag z čipu. Pokud se ale připojí dva senzory, tak nelze využívat stejné sériové rozhraní pro nezávislé čtení jednotlivých snímačů. Proto se muselo přistoupit k řešení, které spočívá v připojení LED na digitální vstupy. Z detekční desky je LED napájena 5 V. Pokud bylo provedeno čtení na RFID, pin na kterém je připojena LED přejde do úrovně LOW, tedy 0 V. Tato vlastnost je využívána pro přepínání mezi několika snímači. Takže se neustále čte v daný okamžik pouze z jediného snímače RFID. Důležitou součástí pro řešení výše uvedeného je knihovna:

```
#include <rdm6300.h>
```

Následně jsou definovány konstanty vstupních pinů snímačů a pomocné proměnné, které zajišťují hlídání směru pohybu drůbeže:

```
// RFID
#define RDM6300_IN_RX_PIN 10
#define RDM6300_IN_LED_PIN 24
#define RDM6300_OUT_RX_PIN 11
#define RDM6300_OUT_LED_PIN 22

Rdm6300 rdm6300_in;
Rdm6300 rdm6300_out;
Rdm6300 *current_rdm6300;

String smer;
int lastSmer, actSmer;
```

Ve funkci `loop()` je logika snímání a rozhodování o evidenci směru pohybu drůbeže a tím i evidence kolik kusů drůbeže se nachází uvnitř a venku:

```
// RFID
if (digitalRead(RDM6300_IN_LED_PIN) == LOW &&
    !rdm6300_in.is_listening()){
    rdm6300_in.listen();
    current_rdm6300 = &rdm6300_in;
}else if (digitalRead(RDM6300_OUT_LED_PIN) == LOW &&
    !rdm6300_out.is_listening()){
    rdm6300_out.listen();
    current_rdm6300 = &rdm6300_out;
}

if (current_rdm6300->update()){
    actSmer=current_rdm6300->get_tag_id();
    smer=(String)digitalRead(RDM6300_IN_LED_PIN)+(String)
    digitalRead(RDM6300_OUT_LED_PIN);
    if(smer=="01" && lastSmer!=actSmer){
        Serial.print("IN:");
        Serial.println(smer);
        lastSmer=actSmer;
    }else if(smer=="10" && lastSmer==actSmer){
        Serial.print("OUT:");
        Serial.println("Je venku");
        lastSmer=0;
    }
}
```

```

    }else if(smer=="10" && lastSmer!=actSmer){
        Serial.print("OUT:");
        Serial.println(smer);
        lastSmer=actSmer;
    }else if(smer=="01" && lastSmer==actSmer){
        Serial.print("IN:");
        Serial.println("Je ve uvnitr");
        lastSmer=0;
    }
}

```

3.10.3 Snímání teploty a vlhkosti

Snímání teploty a vlhkosti pro udržení optimálního klimatu probíhá pomocí čidla DHT11. Pro získání aktuálních hodnot jsou použity funkce `readDHTTemperature()`, `readDHTHumidity()`. Obě funkce snímají veličiny odděleně, ale pro zobrazení teploty a vlhkosti v serverové aplikaci jsou odesílány společně voláním funkce `readDHT()`. Pro korektní čtení hodnot musí být nainstalovány knihovny `Adafruit_Sensor.h` a `DHT.h`.

```

float readDHTTemperature(){
    float t = dht.readTemperature();
    if(isnan(t)){
        t=1000;
    }
    return t;
}

float readDHTHumidity(){
    float h = dht.readHumidity();
    if(isnan(h)){
        h=1000;
    }
    return h;
}

float readDHT(){
    resA="";
    float t = readDHTTemperature();
    float h = readDHTHumidity();

```

```

if(t==1000 || h==1000){
    resA="|Err|Err";
    t=0;
    h=0;
}else{
    resA=(String)t+"|"+(String)h;
}
Serial.println(resA);
Serial3.println(resA);
return t;
}

```

Ukazuje se, že čidlo DHT11 vykazuje poměrně velkou chybovost v čtení. Dochází k častým výpadkům. To je v kódu ošetřeno tak, že se proměnným, které obsahují teplotu a vlhkost přiřazuje hodnota 1000. Je to důležité pro další rozhodování při ovládání akčních členů.

3.10.4 Spuštění vytápění

Ovládání akčního členu v podobě větráku souvisí se získanými hodnotami čidla DHT11, z předchozí kapitoly. Na základě aktuální teploty se rozhodne, zda se zapne vytápění. Proměnná `manualRunWarm` definuje, zda má být větrání spuštěno manuálně pomocí serverové aplikace.

```

float tp=readDHTTemperature();
if(tp<limitTempRunWarm || manualRunWarm==1){
    analogWrite(PINWARM, 255);
}else{
    analogWrite(PINWARM, 0);
}

```

3.10.5 Spuštění větrání

Podobně jako vytápění pracuje i větrání. Spuštění větrání je podmíněno jednak hodnotou vlhkosti a dále teploty. Proměnná `manualRunFan` definuje, zda má být větrání spuštěno manuálně pomocí serverové aplikace.

```

float hp=readDHTHumidity();
if((hp>limitHumiRunFan && hp<1000) ||

```

```

        (tp>limitTempRunFan && tp<1000) ||
        manualRunFan==1){
            analogWrite(PINFAN, 255);
        }else{
            analogWrite(PINFAN, 0);
        }
    }

```

3.10.6 Detekce predátorů

Pro detekci pohybu v nočních hodinách se využívá PIR čidlo, které je popsáno v kapitole 3.6. Programování PIR čidla je velmi jednoduché. Výhodou zvoleného typu je, že lze nastavit přímo na čidle za jaké tmy bude čidlo aktivováno a na jak dlouho bude sepnuté. Při detekci pohybu se odešle na digitální pin informace o pohybu a následně bude přes PWM spínač zapnut LED pásek. PWM spínač je použit z toho důvodu, že napájení světelného zdroje přímo z desky nemusí být dostačující. Proto se použije externí zdroj napájení, kterým může být například 12 V. Ovládání světelného zdroje PIR čidlem zajišťuje funkce `setPirLed()`.

```

void setPirLed(){
    pirState = digitalRead(PINPIR);
    if(pirState == HIGH) {
        digitalWrite(PINLED, HIGH);
    }else{
        digitalWrite(PINLED, LOW);
    }
}

```

3.10.7 Modul reálného času

Modul RTC zajišťuje aktuální čas. Časové údaje jsou v řídicím systému důležité pro spouštění krmicího modulu. Modul RTC se musí v první řadě inicializovat:

```

DS3231 rtc;

```

Pokud není systém připojen k veřejné internetové síti musí se nastavit a do modulu RTC uložit aktuální čas. Funkce `initRTC()` se volá z inicializační funkce `setup()`.

```

void initRTC() {
    rtc.begin();
    /*
    rtc.setDOW(WEDNESDAY);
    rtc.setTime(10, 10, 00);
    rtc.setDate(20, 3, 2022);
    */
}

```

Přístup k časovým údajům je uveden následujícím způsobem:

```

rtc.getDateStr();
rtc.getTimeStr();

```

3.10.8 Přísun vody

Doplňování vody zajišťuje kombinace čidla množství vody a elektromagnetického ventilu. Logika doplňování vody je velmi jednoduchá. Čidlo vody hlídá, zda je dostatek vody v napájecí nádobě. Pokud je vody méně než definovaná minimální hodnota, otevře se ventil a voda se doplní. Otevření ventilu lze provést také ze serverové aplikace.

Funkce, které zajišťují doplňování vody, jsou `getWaterLevel()` a `conValve()`.

```

int getWaterLevel() {
    wLevel = analogRead(PINWSENSOR);
    return wLevel;
}

void conValve(int actValve) {
    if(actValve==1) {
        digitalWrite(PINVALVE, HIGH);
    } else {
        digitalWrite(PINVALVE, LOW);
    }
}

```

Ovládání ventilu závisí na návratové hodnotě funkce `getWaterLevel()`. Následně je testována limitní hodnota pro otevření ventilu, popřípadě, zda ventil není otevřen z aplikace.

```

if (getWaterLevel() < minWLevel || manualRunValve == 1) {
    conValve(1);
} else {
    conValve(0);
}

```

3.10.9 Přísun krmiva

Doplňkové krmení je realizováno prostřednictvím zásobníku, ze kterého je vysypáno krmení po dobu, co je otevřen uzávěr ovládaný servomotorem. Aktivace krmicího modulu probíhá v definovaný čas, který poskytuje modul RTC. Otevření zásobníku lze provést i ze serverové aplikace. Funkce, která zajišťuje řízený přísun krmiva je `feedingTime()`. Funkce `runServo()`, zajišťuje pouze otočení hřídele servomotoru na stanovenou pozici. Oddělení těchto dvou funkcí je z důvodu manuálního spuštění servomotoru ze serverové aplikace voláním funkce `runServo()`.

```

void runServo(int degree){
    servo.write(degree);
}

void feedingTime(){
    if(rtc.getHour(h12Flag, pmFlag) == feedHour && rtc.getMinute()
        == feedMin){
        runServo(180);
        delay(3000);
        runServo(0);
    }
}

```

3.10.10 Počítání snášky

Pro monitorování snášky byl využit IR snímač. Snímač se skládá ze dvou částí – vysílače a přijímače. Díky této konstrukci jej lze poměrně snadno zapojit, a to i s ohledem na mechanickou konstrukci. Programování probíhá pouze nad přijímačem, který neustále přijímá paprsek z vysílače. Vysílač je pouze napájen. Pokud dojde k přerušení paprsku, je zaevidován průchod objektu. V tomto případě průchod vejce. S průchodem vejce se také zaeviduje číslo živočicha, který snášku provedl. Počítání je zajištěno funkcí `countEggs()`.

```

int countEggs() {
    irSensorState = digitalRead(PINIR);
    if (irSensorState == HIGH) {
        countEggsNum++;
    }
    return countEggsNum;
}

```

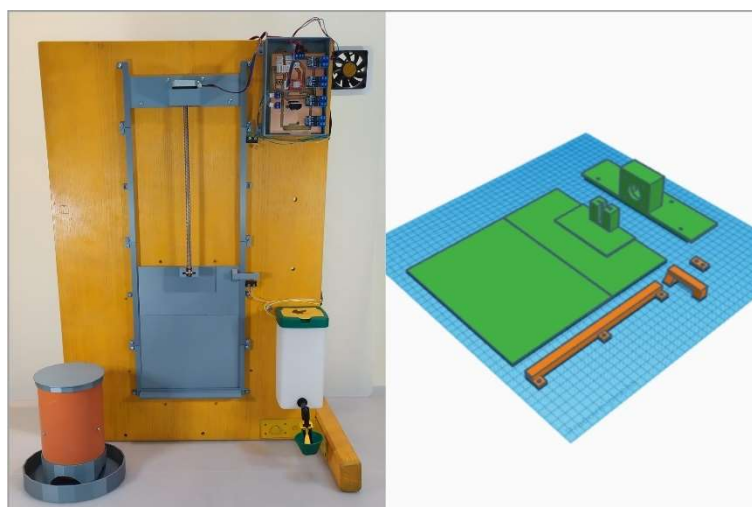
3.11 Konstrukční zpracování

Pro otestování všech komponentů bylo zapotřebí vytvořit prototypový model s osazením všech popsanych částí viz. Obrázek 1. Pro možnost úpravy jednotlivých mechanických dílů byla skupina komponentů vymodelována a vytisknuta pomocí 3D tiskárny. Výhodou je snadné přizpůsobení rozměrů pro již vyrobené kurníky. Dále možnost dostupné výroby a nahrazení opotřebovaných nebo poškozených dílů.

3.11.1 Bezpečnost

Ovládání vstupních dvířek

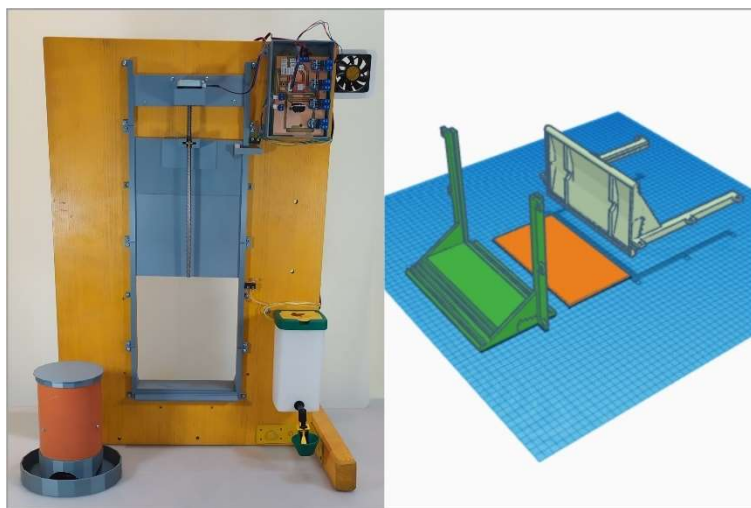
Patří k velmi namáhaným mechanickým částem. V horní části je připevněn krokový motor s lineární osou a se šroubovicí, po které se pohybuje matice pevně spojená s dvířky. Ta se pohybuje v lištách s vřítí pro bezproblémový chod i v případě vnějšího znečištění. Pro plné otevření a zavření jsou použity koncové spínače viz. Obrázek 26.



Obrázek 26: Vstupní dvířka v poloze zavřeno, dolní koncový spínač sepnutý

Sledování počtu drůbeže

Spodní část vstupu je konstrukčně nejsložitější a v provozu velmi zatěžována kontaktem drůbeže a vnějším znečištěním. V dutém prostoru jsou umístěna dvě RFID čidla pro sledování pohybu drůbeže, která má na pařátu tag v podobě kroužku. Podle detekce signálu vnitřní čidlo – vnější čidlo je drůbež venku, v opačném případě při detekci nejprve vnějším čidlem a následně vnitřním je drůbež bezpečně uvnitř. Obrázek 27 ukazuje umístění čidel, která jsou pak v běžném provozu zakryta spodním víkem.

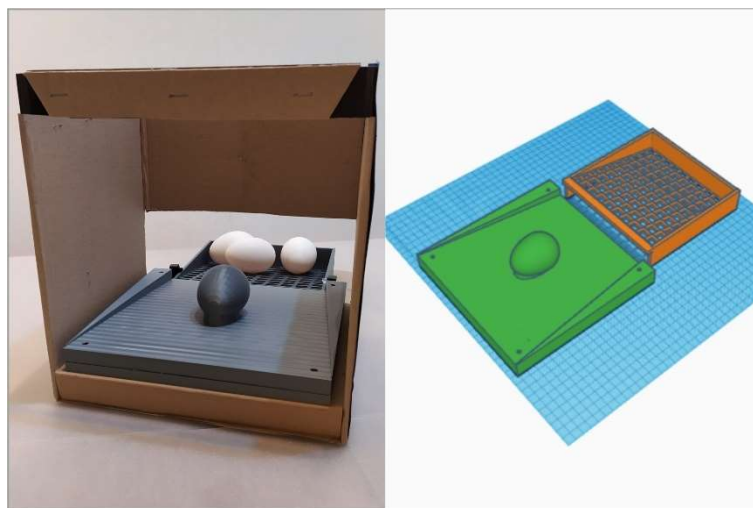


Obrázek 27: Spodní díl vchodu s RFID čidly, dvířka v poloze otevřeno

3.11.2 Produkce

Technologie sběru vajec

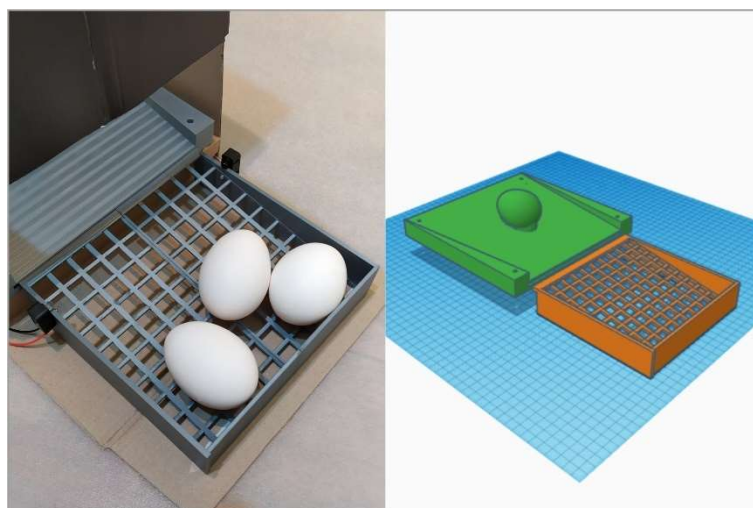
Snůškové místo pro naše účely musí splňovat několik základních parametrů. Je vhodné mít na snůškovém místě podkladek pro zvýšení produkce a obliby hnízda. Plocha má mít sklon pro samovolné vykutálení sneseného vejce do zásobníku. V podkladku (šedivé vejce na Obrázku 28) je umístěno další RFID čidlo pro identifikaci nosnice.



Obrázek 28: Snůškové místo s RFID čidlem v podkladku a sběrný koš s optickým čidlem

Kontrola snůšky

Zásobník je opatřen IR senzory pro detekci sneseného vejce, které se po snesení samovolně skutálí do zásobníku. Svým pohybem přeruší paprsek infračerveného světla, a tak lze tímto způsobem spárovat nosnici se sneseným vajíčkem. Údaj může chovatel využít pro identifikaci silných nebo naopak slabších a nemocných jedinců.

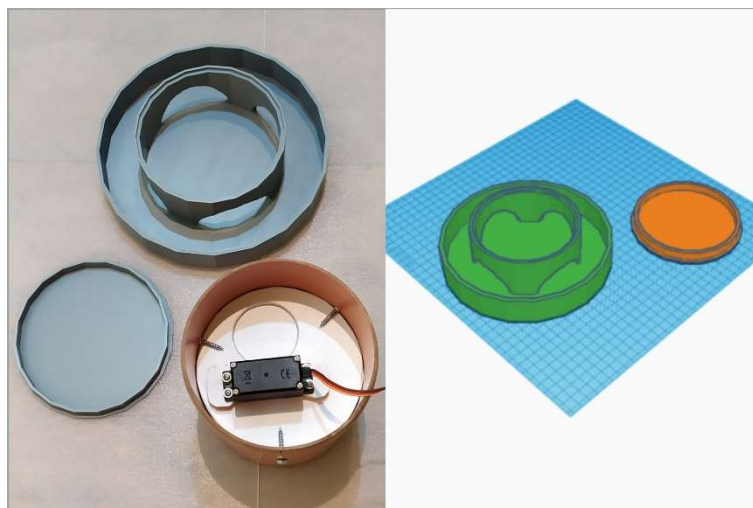


Obrázek 29: Sběrný koš s optickým čidlem

3.11.3 Potrava

Přísun potravy

Jednoduché provedení využívá obecně dostupných komponentů. Hlavní zásobník je tvořen oranžovou trubicou KGEM o průměru 110 mm osazenou ve spodní části klapkou ovládanou krokovým motorem. Ten má za úkol otevřít na krátký interval klapku a umožnit samočinné sesypání suché krmné směsi do krmítka, na kterém je zásobník nainstalovaný.

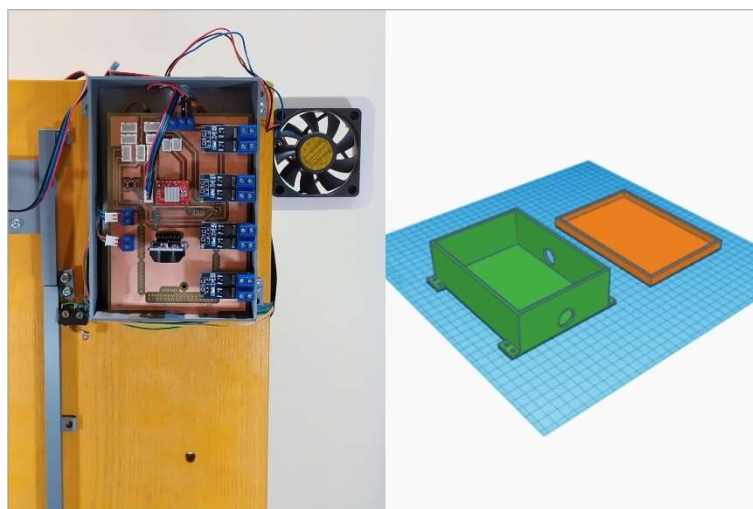


Obrázek 30: Model zásobníku pro sypké krmivo

3.11.4 Řídicí jednotka

Ochranná krabice

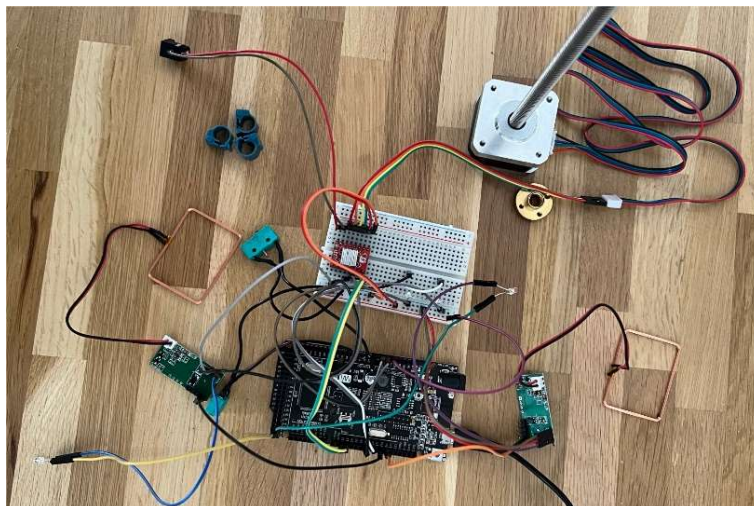
Pro ochranu před povětrnostními vlivy a snadnou montáž slouží krabice s víkem. V krabici jsou dvě průchodky pro kabeláž.



Obrázek 31: Ochranná krabice řídicí jednotky

3.12 Testování systému

Testování systému probíhalo v průběhu celého vývoje při postupném zapojování jednotlivých senzorů a akčních členů. Docházelo ke zpřesnění některých parametrů jako je automatické zavírání a otevírání dveří podle hodnot fotorezistoru, definice limitní hodnoty pro spuštění ventilu pro přísun vody, délka otevírací záklopky pro přísun dávky krmení.



Obrázek 32: Testování systému s využitím nepájivého pole

3.12.1 Testování senzorů

Testy jednotlivých senzorů a funkčnost akčních členů proběhla úspěšně. Jediné problémy přinesly senzory RFID, které zcela neposkytovaly parametry, které jsou uváděny výrobcem. Zejména ve spojení s identifikačními čipy vykazovaly velkou nepřesnost a docházelo k výpadkům při evidenci živočichů.

3.12.2 Testování serveru

Ze serverové aplikace jsou prostřednictvím odesílaných požadavků volány funkce, které zajišťují ovládání akčních členů nebo poskytují hodnoty ze senzorů. Limitujícím faktorem v komunikaci mezi platformou Mega a ESP je samotná sériová komunikace. Vzhledem k postupnému přidávání a testování jednotlivých čidel ovšem docházelo k postupné optimalizaci komunikace tak, že se postupně odstranily výpadky dat přeprogramováním funkce pro odesílání dat z Mega do ESP, čím tak již nedochází ke ztrátě dat. Důležitou funkcionalitou systému, resp. serverové aplikace je zajištění nastavení ovládacích prvků při reloadu stránky aplikace v prohlížeči. Při normálním pojetí html stránky by došlo

k výchozímu nastavení ovládacích prvků do pozice vypnuto. Zde jsou při prvním zobrazení aplikace nastaveny aktuální hodnoty, které se průběžně ukládají do paměti a pak je lze použít pro aktuální nastavení ovládacích prvků.

3.12.3 Testování mechanických částí

Modelování mechanických částí proběhlo v online verzi programu Tinkercad 3D a vzhledem k jednoduchosti dílů nebylo zapotřebí použít sofistikovanější programy typu Solidworks 3D CAD. Tvorba dílů byla postupná a dotvářena dle dílčích požadavků. Nejsložitější částí pro modelování byla spodní náslapná část rámu dvířek. Při modelování se muselo počítat s umístěním dvou RFID čidel. Celý díl je přístupný ze spodu a uzavřen víčkem proti vniknutí nečistot.

Tvar spodní části způsobil komplikace při samotném tisku, vše bylo nakonec vyřešeno rozdělením dílu před tiskem v softwaru PrusaSlicer. Obdobným způsobem byl vyřešen tisk dvířek. Jejich velikost totiž přesahovala tisknutelnou plochu tiskárny Prusa i3 MK3S+ viz. Obrázek 34. Díly, které nebylo vzhledem k jejich tvaru nebo velikosti možné vytisknout vcelku byly následně slepeny pomocí epoxidové pryskyřice. Důležitým výhodou je dostupnost a ekologická nezávadnost použitého PLA filamentu.



Obrázek 33: Tisk komponentů na 3D tiskárně

3.13 Úpravy systému pro další verzi

Při zpracovávání bakalářské práce se objevily nové oblasti otevírající dveře pro další zdokonalení systému. Dále i samotné testování ukázalo několik problémů, které by mohly být vyřešeny v některé z dalších verzí systému.

3.13.1 Čidla

V první řadě je to volba komponent RFID. Přestože RFID přijímače jsou velmi variabilní pro mechanické zpracování do konstrukce, vykazují velmi nízkou přesnost. Dosah čtení neodpovídá parametrům uváděným výrobcem a se zvolenými, resp. dostupnými čipy není evidence drůbeže zcela přesná. Vzhledem k metodě připojení jednotlivých senzorů prostřednictvím konektorů není náprava v podobě výměny a otestování jiného typu RFID snímačů nikterak problematická.

3.13.2 Evidence drůbeže

Pokud by byl klíčový požadavek na přesnou evidenci živočichů a jejich produkci, potom kromě zmíněných problémů by měl systém odesílat data na server do databáze, ze které by mohla být data lépe analyzována a uchovávána mimo paměť MCU. Tím by se vyřešily i problémy s evidencí snůšky, která má v tuto chvíli pouze informační charakter. Aby nemusel být použit vzdálený server, když je celý systém dostupný pouze v lokální síti lze využít některé z „lehkých“ databází umístěných na SD kartě.

3.13.3 Sběrný koš

Z dílů, které jsou nejvíce variabilní a budou se muset řešit individuálně v každé instalaci, je sběrný koš na vyprodukovaná vejce. Počítání vajec je pomocí čidel vyřešen, jejich instalace u snáškového místa také, ale technické provedení koše, velikost a systém sběru vajec (manuální, automatický) je velice variabilní.

3.13.4 Ovládání dvířek

Jedním z budoucích vylepšení samočinného otevírání a zavírání dvířek by mohlo být zastavení posuvu v případě identifikace živočicha stojícího v době zavírání ve vchodu. Důvodem je, aby nedošlo k jeho poranění. Možné by to mohlo být při vylepšení a spolehlivosti RFID čidel počítající průchody jedinců. Nebo využitím jiného typu čidla.

3.13.5 Bezpečnost

Přestože ochrana proti predátorům je řešena, je zde prostor pro další vylepšení hlavně v ochraně výběhu před dravými ptáky (ulovení drůbeže) a ostatním ptactvem (zavlečení nemoci). Dalším zvýšením bezpečnosti je vizuální kontrola pozemku a ochrana před zloději pomocí kamerového systému, alarmu apod.

3.13.6 Napájení systému

Hovoříme-li o splnění podmínky autonomie systému, musí se řešit i napájení celého systému. Předpokládá se, že systém bude napájen prostřednictvím akumulátoru, který bude nabíjen prostřednictvím solárního panelu. Přestože řízení nabíjení by se dalo realizovat přes solární regulátor a mohlo by být odděleno od samotného řídicího systému. Nicméně i tato součást by měla být zapracovaná do mobilní aplikace, aby bylo možné sledovat minimálně aktuální stav nabití akumulátoru, popřípadě proces nabíjení. Toto rozšíření ovšem obnáší další studium problematiky a úpravu systému.

3.13.7 Dešťová voda

Jedním ze zajímavých podnětů při zpracování práce je řešení zachytu a využívání dešťové vody na pozemku, její následná úprava a využití v pitném režimu. Je otázkou, zda by tato oblast pomohla či naopak zkomplikovala požadavek na autonomní přístup chovu.

4 Závěr

Předkládané řešení řídicího systému pro ekologický chov drůbeže poskytuje ucelený koncept. Snaží se pokrýt většinu požadovaných parametrů, které jsou důležité pro chov. Na základě provedené analýzy byl definován soubor parametrů, které by mohly ovlivňovat malochov drůbeže. Výsledek analýzy vyplývající z teoretické části poskytl základ pro návrh řídicího systému. Systém řeší oblast bezpečnosti, mikroklimatu, potravy, produkce a ovládání systému.

Pro splnění podmínek definovaných v cílech práce se jednotlivé kapitoly zabývají dílčí problematikou a na základě výsledků poskytují řešení. Základním problémem byla definice parametrů, které by mohly hrát klíčovou roli pro efektivní chov drůbeže v mobilním ustájení s ohledem na splnění podmínek ekologického chovu. Jak již bylo uvedeno, tyto parametry pomohla definovat rozsáhlá analýza současných řešení v oblasti velkochovů.

Pro pokrytí zpracování oblastí vzešlých z teoretické části práce byl navržen teoretický model systému. Tento model poskytl reálný pohled na množství proměnných, které je nutné řídit či sledovat. Klíčovým prvkem bylo nalezení vhodné platformy pro řízení celého systému a výběr odpovídajícího mikrokontroleru. Pro výběr MCU bylo nutné nejdříve definovat množství a typ senzorů a akčních prvků. Tím se zabývá zejména kapitola 3.6. Následně byla provedena volba vhodného mikrokontroleru. Byla zvolena platforma RobotDYN, která umožňuje připojení velkého množství senzorů a akčních prvků a připojení k internetu, které poskytuje možnost ovládání systému prostřednictvím mobilní aplikace.

Po výběru vhodných komponent se mohlo přistoupit k fyzickému návrhu systému. Zejména k výrobě tištěného spoje, který je navržen jako přídatné pole k vybranému MCU a umožňuje splnit podmínku modularity systému. Realizace fyzického návrhu byla podpořena postupným testováním funkčnosti jednotlivých komponent. Hovoříme-li o testování, tak jeho součástí je samozřejmě programování. Programování se týká zejména získávání hodnot veličin ze senzorů, které kromě informačního charakteru, poskytují data pro případně ovládání akčních členů. Programování probíhalo v prostředí Visual Studio Code s nadstavbou PlatformIO. Programovacím jazykem je kombinace wiring a C++.

Kromě testování komponent a jejich programování na úrovni pouhé funkcionality muselo být provedeno otestování ve fyzickém provedení. Proto bylo přistoupeno k návrhu a modelování mechanických prvků a jejich následné výrobě na 3D tiskárně. Při tvorbě prototypu se rozkryla

přidaná hodnota tohoto řešení. Vznikla možnost pro chovatele vytisknout si pouze komponenty, které chtějí použít a zároveň mít možnost díly upravit podle specifických požadavků.

Při realizaci této práce byl kladen důraz na návrh a realizaci řídicího systému. Testování probíhalo na modelové konstrukci, kde se ukázaly některé problematické části. Pro jejich vyřešení jsou poskytnuty návrhy úprav v dalších verzích. Práce svým rozsahem pokrývá rozsáhlou oblast a nejenom v tomto ohledu jsou definované cíle splněny.

Seznam literatury

- [1] BROUČEK, Jan. *Technologie a technika chovu drůbeže při splnění podmínek welfare: certifikovaná metodika*. V Českých Budějovicích: Jihočeská univerzita, Zemědělská fakulta, 2011. ISBN 978-80-7394-337-0.
- [2] TŮMOVÁ, Eva. *Vliv systému ustájení a výživy na kvalitu masa a vajec drůbeže*. Výzkumný ústav živočišné výroby, v.v.i. Praha: Vědecký výbor výživy zvířat, 2007.
- [3] JÍLEK, Arnošt. Požadavky na welfare chované drůbeže. *Zemědělec* [online]. Praha: Profi Press, 2012, 29.06.2012 [cit. 2022-01-17]. Dostupné z: <https://zemedelec.cz/pozadavky-na-welfare-chovane-drubeze/>
- [4] ZAPLETAL, David a Miroslav MACHÁČEK. *Chov hospodářských zvířat*. 1. Brno: Veterinární a farmaceutická univerzita Brno, 2015.
- [5] LEDVINKA, Zdeněk. *Chov drůbeže I*. V Praze: Česká zemědělská univerzita, 2011. ISBN 978-80-213-2174-8.
- [6] ZHENG, Haikun, Tiemin ZHANG, Cheng FANG, Jiayuan ZENG a Xiuli YANG. Design and Implementation of Poultry Farming Information Management System Based on Cloud Database. *Animals* [online]. 2021, 11(3) [cit. 2022-01-23]. ISSN 2076-2615. Dostupné z: doi:10.3390/ani11030900
- [7] FANG, Cheng, Junduan HUANG, Kaixuan CUAN, Xiaolin ZHUANG a Tiemin ZHANG. Comparative study on poultry target tracking algorithms based on a deep regression network. *Biosystems Engineering* [online]. 2020, 190, 176-183 [cit. 2022-01-23]. ISSN 15375110. Dostupné z: doi:10.1016/j.biosystemseng.2019.12.002
- [8] FANG, Cheng, Tiemin ZHANG, Haikun ZHENG, Junduan HUANG a Kaixuan CUAN. Pose estimation and behavior classification of broiler chickens based on deep neural networks. *Computers and Electronics in Agriculture* [online]. 2021, 180 [cit. 2022-01-23]. ISSN 01681699. Dostupné z: doi:10.1016/j.compag.2020.105863

- [9] AYDIN, A., C. BAHR a D. BERCKMANS. A real-time monitoring tool to automatically measure the feed intakes of multiple broiler chickens by sound analysis. *Computers and Electronics in Agriculture* [online]. 2015, 114, 1-6 [cit. 2022-01-23]. ISSN 01681699. Dostupné z: doi:10.1016/j.compag.2015.03.010
- [10] CHOUKIDAR, Geetanjali A. a N.A. DAWANDE. Smart Poultry Farm Automation and Monitoring System. In: *2017 International Conference on Computing, Communication, Control and Automation (ICCUBE)* [online]. IEEE, 2017, 2017, s. 1-5 [cit. 2022-01-23]. ISBN 978-1-5386-4008-1. Dostupné z: doi:10.1109/ICCUBE.2017.8463953
- [11] ROWE, DAWKINS a GEBHARDT-HENRICH. A Systematic Review of Precision Livestock Farming in the Poultry Sector: Is Technology Focussed on Improving Bird Welfare. *Animals* [online]. 2019, 9(9) [cit. 2022-01-23]. ISSN 2076-2615. Dostupné z: doi:10.3390/ani9090614
- [12] BALACHANDAR, S. a R. CHINNAIYAN. Internet of Things Based Reliable Real-Time Disease Monitoring of Poultry Farming Imagery Analytics. PANDIAN, A.Pasumpon, Tomonobu SENJYU, Syed Mohammed Shamsul ISLAM a Haoxiang WANG, ed. *Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI - 2018)* [online]. Cham: Springer International Publishing, 2020, 2020-08-01, s. 615-620 [cit. 2022-01-23]. Lecture Notes on Data Engineering and Communications Technologies. ISBN 978-3-030-24642-6. Dostupné z: doi:10.1007/978-3-030-24643-3_73
- [13] STEFANOVA, Magdalena. Precision Poultry Farming: Software Architecture Framework and Online Zootechnical Diary for Monitoring and Collaborating on Hens' Health. SALAMPASIS, Michail a Thomas BOURNARIS, ed. *Information and Communication Technologies in Modern Agricultural Development* [online]. Cham: Springer International Publishing, 2019, 2019-02-12, s. 191-205 [cit. 2022-01-23]. *Communications in Computer and Information Science*. ISBN 978-3-030-12997-2. Dostupné z: doi:10.1007/978-3-030-12998-9_14
- [14] KONOPÁSEK, V., (1994) *Stavby pro chov drůbeže z hlediska welfare*. Praha, Ústav zemědělských a potravinářských informací, ISSN 0862-3562

- [15] OLANREWAJU, H.A., J.L. PURSWELL, S.D. COLLIER a S.L. BRANTON. Effect of ambient temperature and light intensity on physiological reactions of heavy broiler chickens. *Poultry Science* [online]. 2010, 89(12), 2668-2677 [cit. 2022-01-25]. ISSN 00325791. Dostupné z: doi:10.3382/ps.2010-00806
- [16] LEDVINKA, Zdeněk. *Chov drůbeže I.* V Praze: Česká zemědělská univerzita, 2011. ISBN 80-7040-446-9.
- [17] MARSHALL, Ken. Temperature and relative humidity critical in climate control. *Poultry world* [online]. Amsterdam: Misset Uitgeverij B.V., 2017 [cit. 2022-01-25]. Dostupné z: <https://www.poultryworld.net/Health/Articles/2017/1/Temperature-and-relative-humidity-critical-in-climate-control-78825E/>
- [18] PŘIKRYL, Miroslav. *Technologická zařízení staveb živočišné výroby.* Praha: Tempo Press II, 1997. ISBN 80-901052-0-3.
- [19] ANDRT, Miroslav. *Technika a technologie pro chov zvířat.* V Praze: Česká zemědělská univerzita, 2011. ISBN 978-80-213-2164-9.
- [20] S. EZEMA, Longinus, Miracle C. NNABUKO a Chigozie BEN-OPARA. *Design and Implementation of an Embedded Poultry Farm.* In: Springer Nature, s. 187-192. ISBN 3030691438.
- [21] G. BILAL, A.-K. KHALED and C. K KHALED, A Poultry Farming Control System Using a ZigBee-based Wireless Sensor Network, *International Journal of Control and Automation*, vol. 10, no. 9, s. 191-198, 2017
- [22] DRISHTI, S., R. RAKHI and M. JIMMI, Smart Farm: Extending Automation To The Farm Level, *International Journal of Scientific & Technology Research*, vol. 3, no. 7, s. 109-112, 2014.
- [23] N. A. DAWANDE, A Survey on Smart Poultry Farm Automation and Monitoring System, *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 6, no. 3, s. 4806-4809, 2017
- [24] K.SINDUJE, S. JENIFER, M. ABISHEK, Automated Control System for Poultry Farm Based On Embedded System, *International Research Journal of Engineering and Technology*, vol. 3, no. 3, s. 620-623, 2016

- [25] MURAD, Mohsin, Khawaja Mohammad YAHYA a Ghulam Mubashar HASSAN. Web based poultry farm monitoring system using wireless sensor network. In: *Proceedings of the 6th International Conference on Frontiers of Information Technology - FIT '09* [online]. New York, New York, USA: ACM Press, 2009, 2009, s. 1- [cit. 2022-01-31]. ISBN 9781605586427. Dostupné z: doi:10.1145/1838002.1838010
- [26] MANSHOR, Noridayu, Amir Rizaan ABDUL RAHIMAN a Muhammad Kamil YAZED. IoT Based Poultry House Monitoring. In: *2019 2nd International Conference on Communication Engineering and Technology (ICCET)* [online]. IEEE, 2019, 2019, s. 72-75 [cit. 2022-01-31]. ISBN 978-1-7281-1439-2. Dostupné z: doi:10.1109/ICCET.2019.8726880
- [27] V DESHMUKH, Ajay. *Microcontrollers: theory and applications*. 1. New Delhi: Tata McGraw-Hill, 2005. ISBN 9780070585959.
- [28] Microcontroller Basics. *Codrey Electronic* [online]. Codrey Electronic, 2018 [cit. 2022-02-21]. Dostupné z: <https://www.codrey.com/microcontroller/microcontroller-basics/>
- [29] LUTKEVICH, Ben. DEFINITION microcontroller. *TechTarget* [online]. TechTarget, 2019 [cit. 2022-02-21]. Dostupné z: <https://internetofthingsagenda.techtarget.com/definition/microcontroller>
- [30] K. PRATT, Mary. Tips to select the right microcontrollers for IoT. *TechTarget* [online]. TechTarget, 2020 [cit. 2022-02-21]. Dostupné z: <https://internetofthingsagenda.techtarget.com/tip/Tips-to-select-the-right-microcontrollers-for-IoT>
- [31] Tricks to Select Perfect Microcontroller for Your Application. *Elysium Embedded School* [online]. Elysium Group of Companies, 2022 [cit. 2022-03-03]. Dostupné z: <https://embeddedschool.in/tricks-to-select-perfect-microcontroller/>
- [32] ODUNLADE, Emmanuel. Selecting the Right Microcontroller for your Embedded Systems Solution. *Medium* [online]. Nerd For Tech, 2022 [cit. 2022-03-03]. Dostupné z: <https://medium.com/nerd-for-tech/selecting-the-right-microcontroller-for-your-embedded-systems-solution-ce4ef7ecb906>

- [33] BENINGO, Jacob. 10 Steps to Selecting a Microcontroller. *Arm Community* [online]. Arm Limited, 2014 [cit. 2022-03-03]. Dostupné z: <https://community.arm.com/arm-community-blogs/b/embedded-blog/posts/10-steps-to-selecting-a-microcontroller>
- [34] How to connect DHT11 Sensor with Arduino UNO. *Project hub* [online]. 2020 [cit. 2022-03-04]. Dostupné z: <https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239>
- [35] FRASER, Charles J. Electrical and electronics principles. *Mechanical Engineer's Reference Book* [online]. Elsevier, 1994, 1994, s. 2-1-2-57 [cit. 2022-03-04]. ISBN 9780750611954. Dostupné z: doi:10.1016/B978-0-7506-1195-4.50006-3
- [36] CHOUDHARY, Ashish. Interfacing RDM6300 RFID Reader Module with Arduino. *Circuit Digest* [online]. 2021 [cit. 2022-03-05]. Dostupné z: <https://circuitdigest.com/microcontroller-projects/interfacing-rdm6300-rfid-reader-module-interfacing-with-arduino-nano>
- [37] Arduino MEGA 2560 With WiFi Built-in - ESP8266. *Trybotics* [online]. India: Trybotics, 2021 [cit. 2022-03-02]. Dostupné z: <https://trybotics.com/project/Arduino-MEGA-2560-With-WiFi-Built-in-ESP8266-99755>
- [38] NOVÁK, Milan. RobotDYN - Arduino MEGA 2560 s vestavěným WiFi - ESP8266. *Mindblog* [online]. Česká republika: VOXCAFE, 2022 [cit. 2022-03-02]. Dostupné z: <https://www.voxcafe.cz/mindblog/clanky/arduino/robotdyn---arduino-mega-2560-s-vestavenym-wifi---esp8266.html>
- [39] What is a Servomotor. *Electrical4U* [online]. 2020 [cit. 2022-03-12]. Dostupné z: <https://www.electrical4u.com/what-is-servo-motor/>
- [40] Serial. *Arduino* [online]. Arduino, 2022 [cit. 2022-02-21]. Dostupné z: <https://www.arduino.cc/en/reference/serial>
- [41] MARGOLIS, Michael. Serial Communications. *Arduino Cookbook* [online]. O'Reilly Media, 2021 [cit. 2022-02-21]. Dostupné z: <https://www.oreilly.com/library/view/arduino-cookbook/9781449399368/ch04.html>

[42] NOVÁK, Milan. RobotDYN - Arduino MEGA 2560 s vestavěným WiFi - ESP8266. *MindBlog* [online]. Česká Republika, 2022 [cit. 2022-03-13]. Dostupné z: <https://www.voxcafe.cz/mindblog/clanky/arduino/robotdyn---arduino-mega-2560-s-vestavenym-wifi---esp8266.html>

[43] ESP8266 *Non-OS: AT Instruction Set*. 3. Espressif Systems, 2021. Dostupné také z: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf

Seznam obrázků

Obrázek 1: Souhrnné schéma sledovaných veličin	12
Obrázek 2: Blokové schéma čidel a akčních členů	14
Obrázek 3: Blokové schéma MCU.....	16
Obrázek 4: Čidlo teploty a vlhkosti.....	22
Obrázek 5: Koncový spínač	23
Obrázek 6: Ovladač krokového motoru	23
Obrázek 7: Krokový motor	24
Obrázek 8: Fotorezistor	24
Obrázek 9: Senzor pohybu	25
Obrázek 10: Modul MOSFET PWM 15 A 400 W.....	26
Obrázek 11: Modul RFID s tagy v podobě kroužků	26
Obrázek 12: Senzor přerušení IR paprsku.....	27
Obrázek 13: Senzor snímání vodní hladiny	27
Obrázek 14: Elektromagnetický ventil.....	28
Obrázek 15: Modul reálného času.....	28
Obrázek 16: Servomotor	29
Obrázek 17: Deska Robot DYN.....	32
Obrázek 18: Návrh desky Robot DYN pro Eagle	34
Obrázek 19: Schéma zapojení modulů pro řídicí jednotku	35
Obrázek 20: Deska plošného spoje	36
Obrázek 21: Zařízení pro výrobu plošného spoje Protomat E 44	37
Obrázek 22: Vyrobená oboustranná PCB deska	38
Obrázek 23: Osazený tištěný spoj	38
Obrázek 24: Výstup AT příkazů	41
Obrázek 25: Uživatelské rozhraní serverové aplikace	49

Obrázek 26: Vstupní dvířka v poloze zavřeno, dolní koncový spínač sepnutý.....	58
Obrázek 27: Spodní díl vchodu s RFID čidly, dvířka v poloze otevřeno.....	59
Obrázek 28: Snůškové místo s RFID čidlem v podkladku a sběrný koš s optickým čidlem	60
Obrázek 29: Sběrný koš s optickým čidlem.....	60
Obrázek 30: Model zásobníku pro sypké krmivo	61
Obrázek 31: Ochranná krabička řídicí jednotky.....	61
Obrázek 32: Testování systému s využitím nepájivého pole	62
Obrázek 33: Tisk komponentů na 3D tiskárně.....	63

Seznam tabulek

Tabulka 1: Porovnání MCU a MPU.....	19
Tabulka 2: Přehled čidel a akčních členů s potřebným typem IO pinů.....	30
Tabulka 3: Porovnání platforem.....	31
Tabulka 4: Konfigurace DIP přepínačů.....	32
Tabulka 5: Sériové linky desek Arduino.....	39

Příloha A

Součástí práce jsou přílohy, které se nachází v komprimovaném souboru Cisar_Ales_2022_BP_prilohy.zip.

Soubor obsahuje následující složky:

3D_tisk – soubory modelů pro 3D tisk mechanických částí.

Eagle_kurnik – schéma řídicího systému a návrh PCB v programu Autodesk Eagle.

Mega_esp – zdrojové soubory programu řídicího systému.