

Automatizované kŕmidlo pre zvieratá

Slavomír Štucka

Martin Tarasovič

Ivana Hojdová

Oleksandr Senych

Architektúry počítačových systémov

01/11/2022

**Obsah**

[Formulácia úlohy](#_heading=h.gjdgxs) 1

[Zoznam obrázkov](#_heading=h.3znysh7) 3

[Zoznam tabuliek](#_heading=h.i7d4b4u15cy5) 4

[Zoznam schém](#_heading=h.m3jqfx5dh0sj) 4

[Súpis súčiastok](#_heading=h.3dy6vkm) 5

[Riešenie](#_heading=h.2s8eyo1) 8

[Hlavné časti projektu](#_heading=h.17dp8vu) 8

[Celková schéma projektu](#_heading=h.3rdcrjn) 9

[Pripojenie potrebných knižníc](#_heading=h.lnxbz9) 9

[Dávkovanie jedla pomocou servo motora](#_heading=h.1ksv4uv) 10

[Zisťovanie prítomnosti zvieraťa pomocou senzora pohybu](#_heading=h.3j2qqm3) 11

[Komunikácia s používateľom](#_heading=h.2xcytpi) 12

[Ovládanie kŕmidla pomocou diaľkového ovládača a IR prijímača](#_heading=h.1pxezwc) 13

[Funkcia void loop()](#_heading=h.3fwokq0) 16

[Funkcia void nasypanie()](#_heading=h.3tbugp1) 19

[Funkcia void nasypanie\_manual()](#_heading=h.nmf14n) 20

[Postup ovládania](#_heading=h.1egqt2p) 24

[Nedostatky riešenia](#_heading=h.kgcv8k) 33

[Fotodokumentácia k projektu](#_heading=h.34g0dwd) 33

[Možné rozšírenia](#_heading=h.4h042r0) **38**

[Kontakty](#_heading=h.1baon6m) **39**

# Formulácia úlohy

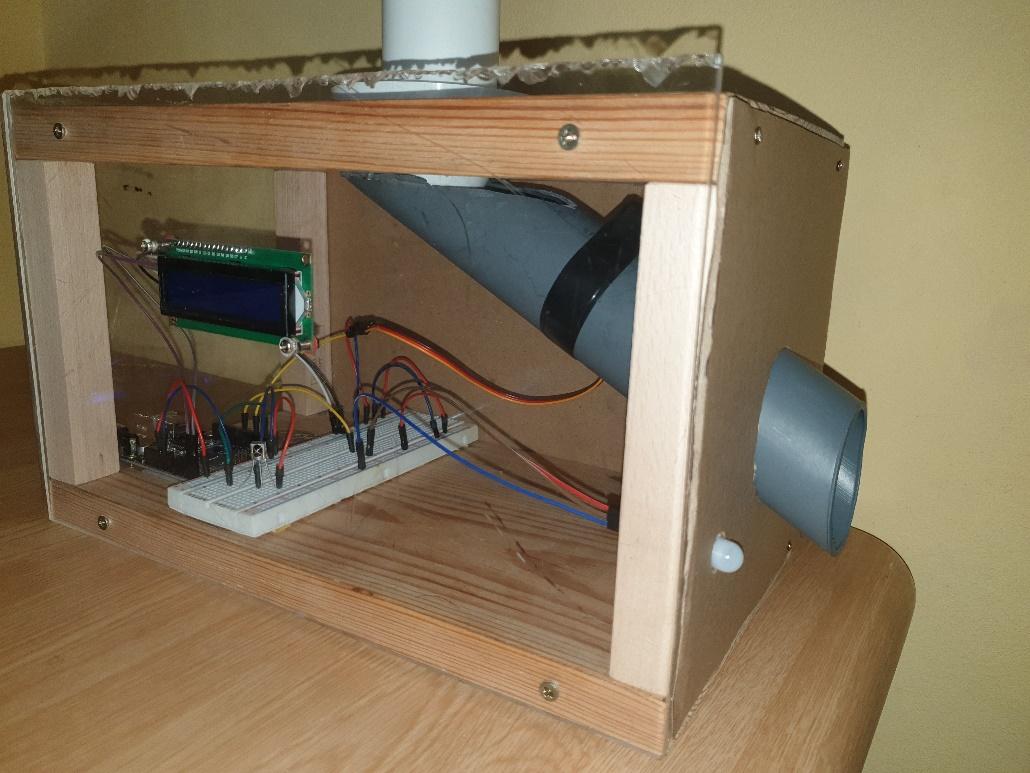
Zadanie má za úlohu zostrojiť multifunkčné kŕmidlo pre zvieratá, ktoré môžeme využiť v každej domácnosti, či už máme doma mačku alebo napríklad psa. Keďže je automatizované, umožňuje kŕmenie aj v našej neprítomnosti. Môžeme odísť na dlhší čas z domu a zvieratko bude dobre zásobené.. Nastaviť sa dá veľkosť dávky ale aj časové intervaly kŕmenia (môže to byť od 15 minút do troch hodín).

Najdôležitejšou súčasťou kŕmidla sú jeho komponenty, bez ktorých by zariadenie nemohlo fungovať. Základný komponent je vývojová doska Arduino, pomocou ktorej bude kŕmidlo ovládané. Dávkovanie je zabezpečené cez klapku napojenú na servo motor. Čím dlhšie je klapka zdvihnutá, tým bude dávka väčšia. Kŕmidlo môžeme ovládať manuálne pomocou ovládača - prepínanie módov a nastavenie veľkosti dávky. Pri stlačení tlačidla na manuálne nasypanie, bude krmivo okamžite vysypané. Pri nastavení automatického kŕmenia, bude krmivo vysypané až po uplynutí nastaveného času a zaregistrovaní zvieratka pomocou senzora pohybu. Je zabezpečené, že sa krmivo nevysype, pokiaľ zvieratko neprejde pred kŕmidlom. Všetko čo nastavujeme ovládačom a odpočítavanie časovača bude zobrazené na LCD displeji, čím zariadenie komunikuje s používateľom.

Úlohou zadania je pokúsiť sa vytvoriť takéto kŕmidlo fungujúce na vývojovej doske Arduino. Malo by spĺňať vyššie uvedené podmienky aby bolo dobre využiteľné v praxi.



*Obrázok 1 Kŕmidlo spredu*



*Obrázok 2 Kŕmidlo z boku*

# Zoznam obrázkov

[Obrázok 1 Kŕmidlo spredu 1](#_heading=h.30j0zll)

[Obrázok 2 Kŕmidlo z boku 2](#_heading=h.1fob9te)

[Obrázok 3 Umiestnenie servo motorčeka a tvar klapky 9](#_heading=h.44sinio)

[Obrázok 4 Konštrukcia rámu 19](#_heading=h.1mrcu09)

[Obrázok 5 Pripevnenie dosky Arduino a zadná strana kŕmidla 20](#_heading=h.46r0co2)

[Obrázok 6 Umiestnenie nepájivého poľa 20](#_heading=h.2lwamvv)

[Obrázok 7 Uchytenie LCD displeja na plexiskle 20](#_heading=h.111kx3o)

[Obrázok 8 Prichytenie naklonenej rúrky 21](#_heading=h.3l18frh)

[Obrázok 9 Prichytenie držiaka zásobníka 21](#_heading=h.206ipza)

[Obrázok 10 Rám s prichytenými súčiastkami 22](#_heading=h.4k668n3)

[Obrázok 11 Prichytenie stien z kartónu 22](#_heading=h.2zbgiuw)

[Obrázok 12 Počiatočný výpis po zapojení kŕmidla 23](#_heading=h.3ygebqi)

[Obrázok 13 Príklad výpisu nastaveného času po pridaní alebo odobraní 24](#_heading=h.2dlolyb)

[Obrázok 14 Výpis vyžadujúci nastavenie dávky od používateľa 25](#_heading=h.sqyw64)

[Obrázok 15 Výpis nastavenej hodnoty dávky 26](#_heading=h.3cqmetx)

[Obrázok 16 Výpis odpočítavania časovača 27](#_heading=h.1rvwp1q)

[Obrázok 17 Oznam o čakaní na pohyb 28](#_heading=h.4bvk7pj)

[Obrázok 18 Oznam o nasypaní krmiva 28](#_heading=h.2r0uhxc)

[Obrázok 19 Stav klapky pred nasypaním 29](#_heading=h.1664s55)

[Obrázok 20 Stav klapky počas nasypania 29](#_heading=h.3q5sasy)

[Obrázok 21 Výpis po stlačení tlačidla „PLAY/PAUSE“ 30](#_heading=h.25b2l0r)

[Obrázok 22 Krok 1 - pripojenie vodičov k LCD displeji s I2C modulom 32](#_heading=h.43ky6rz)

[Obrázok 23 Krok 2 - zapojenie IR prijímača na kontaktné pole 32](#_heading=h.2iq8gzs)

[Obrázok 24 Krok 3 – zapojenie senzoru pohybu 33](#_heading=h.xvir7l)

[Obrázok 25 Krok 4 – zapojenie servo motora a prepojenie plus a mínus riadkov poľa 33](#_heading=h.3hv69ve)

[Obrázok 26 Krok 5 – zapojenie vodičov na piny vývojovej dosky Arduino 34](#_heading=h.1x0gk37)

# Zoznam tabuliek

[Tabuľka 1 Zoznam súčiastok 5](#_heading=h.1t3h5sf)

[Tabuľka 2 Parametre súčiastok 6](#_heading=h.4d34og8)

# Zoznam schém

[Schéma 1 Celková schéma projektu 8](#_heading=h.26in1rg)

[Schéma 2 Schéma zapojenia projektu 31](#_heading=h.1jlao46)

# Súpis súčiastok

*Tabuľka 1 Zoznam súčiastok*

|  |  |  |  |
| --- | --- | --- | --- |
| **Názov (funkcia) súčiastky** | **Typ súčiastky** | **Dostupné v e-shope** | **Cena (1.11.2022) v EUR s DPH** |
| **Vývojová doska** | Arduino UNO SMD edition | [Arduino UNO SMD edícia precízny klon – Techfun.sk – Naj Arduino shop](https://techfun.sk/produkt/arduino-uno-smd-edicia-precizny-klon/) | 7,95 |
| **Kontaktné pole** | 830 Tie-Points Breadboard | [Nepájivé pole 830 bodov – Techfun.sk – Naj Arduino shop](https://techfun.sk/produkt/nepajive-pole-830-bodov/) | 2,50 |
| **LCD display** | LCD I2C 1602  16x2 display | [LED display 16×2 modré podsvietenie + spájkovaný I2C modul – Techfun.sk – Naj Arduino shop](https://techfun.sk/produkt/led-display-16x2-modre-podsvietenie-spajkovany-i2c-modul/) | 4,60 |
| **Servo motor** | Servo Motor(SG90) | [Servo motorček SG90 – Techfun.sk – Naj Arduino shop](https://techfun.sk/produkt/servo-motorcek-sg90/) | 3,10 |
| **Senzor pohybu** | HC-SR505 PIR Motion Sensor | [Infračervený mini senzor pohybu HC-SR505 – Techfun.sk – Naj Arduino shop](https://techfun.sk/produkt/infracerveny-mini-senzor-pohybu-hc-sr505/?fbclid=IwAR297IzPWz1-l2wcDcc3wCwq9dg-VVhIS8HJL96KRGOI0OruTN8ShRPOL8A) | 2,30 |
| **IR prijímač** | IR receiver(TSOP1738) | [IR přijímač s tvarovačem 38kHz Vishay TSOP4838 od 1 € - Heureka.sk](https://prislusenstvo-k-rc-modelom.heureka.sk/ir-prijimac-s-tvarovacem-38khz-vishay-tsop4838/#prehlad/) | 1,00 |
| **Diaľkový ovládač** | Infrared remote controller | [38khz Mcu Learning Board Ir Remote Control Infrared Decoder For Protocol Remote Control For Arduino .mp3 .mp4 - Integrated Circuits - AliExpress](https://www.aliexpress.com/item/1005003696255556.html?spm=a2g0o.productlist.0.0.1ee83429YufB3F&algo_pvid=5e3db3c2-f7d7-46fb-9d0f-53e7d2484f2c&algo_exp_id=5e3db3c2-f7d7-46fb-9d0f-53e7d2484f2c-8&pdp_ext_f=%7b%22sku_id%22%3A%2212000026839298765%22%7d&pdp_npi=2@dis!EUR!1.06!0.97!!!!!@2103143616693111095724104ee887!12000026839298765!sea&curPageLogUid=jBqmjgdbR7hS&fbclid=IwAR2DYDdvuYiK2Oq5uDR7hGqwBVDX0ZtVa2fziIIfAmUzG7SOAv4aT_nOq1M) | 0,97 |
| **Vodiče** | Jumper Wire | [Kábliky 65 kusov M-M rôzne veľkosti – Techfun.sk – Naj Arduino shop](https://techfun.sk/produkt/kabliky-65-kusov-m-m-rozne-velkosti/) | 2.90 |

*Tabuľka 2 Parametre súčiastok*

|  |  |
| --- | --- |
| **Typ súčiastky** | **Parametre / obmedzenia** |
| **Arduino UNO R3** | Čip: ATmega328  Frekvencia: 16 MHz  Pamäť: 32 KB, 0.5 KB použité bootloaderom  SRAM: 2 kB  EEPROM: 1 kB  Vstupy a výstupy:  14x digitálny vstup/výstup (Sériový port, externé prerušenie, PWM)  6x analógový vstup  Rozhranie:  USB 2.0 (prevodník na sériový port)  SPI konektor  Prevodník: CH340G |
| **LCD I2C display 16x2** | Napájacie napätie 5V DC  Rozhranie: I2C(SDA, SCL)  Adresa: 0x3F |
| **Servo motor** | Rozmery: 22mm x 11.5mm x 22.5mm  Operačná rýchlosť: 0.12sec/60° (4.8V bez nákladu)  Krútiaci moment(4.8V): 17.50 oz/in (1 kg/cm)  Odporúčaný teplotný rozsah: -30 až 60  Dead band width: 7usec  Operačné napätie: 3.0V~7.2V |
| **HC-SR505 PIR Motion Sensor** | Pracovné napätie: 4.5 – 20V  Čas zopnutia: 8s  Prúd: <60uA  Rozmery: 10 x 23 mm  Dosah: 3m  Zorný uhol: 110° |
| **IR receiver(TSOP1738)** | Frekvencia: 38kHz  Napájanie: 2.5 – 5.5 V  Púzdro: 3PIN THT |

# Riešenie

Realizácii projektu predchádzal prieskum na internete. Vyhľadali sme si rôzne spôsoby ako možno riešiť zadanú úlohu a čerpali inšpiráciu z podobných projektov riešiacich podomácky vyrobené kŕmidlá. Drvivá väčšina projektov bola založená na využití vývojovej dosky Arduino a klapky ovládanej servo motorčekom pre realizáciu funkcie automatického dávkovania krmiva. Na komunikáciu kŕmidla s používateľom sme sa rozhodli využiť LCD displej. Na zaznamenanie prítomnosti zvieraťa sme sa rozhodli použiť PIR senzor, ktorý zároveň zaručuje, že vysypané krmivo nebude čakať na to, kým ho zviera objaví. Ovládanie kŕmidla je možné realizovať viacerými spôsobmi.

Spôsoby realizácie ovládania

*Tlačidlá*

Použitie jednotlivých tlačidiel na ovládanie kŕmidla je založené na pripojení tlačidiel na jednotlivé piny vývojovej dosky. Toto riešenie vyžaduje využitie narastajúceho počtu pinov vývojovej dosky s narastajúcim počtom tlačidiel, ktorý taktiež narastá s narastajúcim počtom ovládaných funkcionalít kŕmidla. Taktiež je toto riešenie nepohodlné pre používateľa, ktorý by sa musel skloniť ku kŕmidlu, ktoré je v prevádzke a položené na zemi.

*Diaľkový ovládač*

Toto riešenie je založené na použití diaľkového ovládača a IR prijímača, umožnujúce použitie vysokého počtu tlačidiel s využitím len jedného pinu vývojovej dosky. Takisto umožňuje ovládanie kŕmidla z vzdialenosti, výrazne zvyšujúc komfort ovládania. Vzhľadom na dostupné knižnice vývojovej dosky je toto riešenie jednoduché na implementáciu.

*Mobilná aplikácia*

Ako riešenie sa ponúkal aj vývoj mobilnej aplikácie a využitie WiFi alebo Bluetooth modulu s vývojovou doskou Arduino na ovládanie kŕmidla. Toto riešenie je najkomfortnejšie pre používateľa a vysoko rozšíriteľné. Vzhľadom na časový rámec projektu, dostupnosť a cenu komponentov naň potrebných je toto riešenie drahé a náročné na realizáciu.

Naše riešenie

Vzhľadom na časový rámec projektu a cenu komponentov sme si vybrali najpraktickejšie riešenie pomocou vývojovej dosky Arduino, servo motorčeka, diaľkového ovládača a IR prijímača a PIR senzora.

## 

## Hlavné časti projektu

1 Nastavenie knižníc

2 Dávkovanie jedla pomocou servo motora

3 Senzor pohybu

4 Komunikácia s používateľom – využitie displeja

5 Ovládanie kŕmidla pomocou IR ovládača

6 Funkcia void loop()

7 Funkcia void nasypanie()

8 Funkcia void nasypanie\_manual()

## Celková schéma projektu

### 

*Schéma 1 Celková schéma projektu*

### Pripojenie potrebných knižníc

Na ovládanie serva využívame knižnicu na ovládanie serva <Servo.h>, na prácu s displejom využívame knižnicu <LiquidCrystal\_I2C.h> a na dekódovanie infračerveného signálu stlačeného tlačidla na diaľkovom ovládači využívame externú knižnicu <IRremote.h>.

#include <Servo.h> //kniznice

#include <LiquidCrystal\_I2C.h>

#include <IRremote.h>

*Výpis 1 Nastavenie knižníc*

### Dávkovanie jedla pomocou servo motora

 Ako zabezpečiť vysypanie jedla zvieratku a možné nastavenie veľkosti dávky bol asi najväčší konštrukčný problém. Rozhodli sme sa to vyriešiť pomocou serva a klapky. Klapka je priskrutkovaná k servo motorčeku, ktorý je pripevnený k naklonenej plastovej rúrke tak, aby klapka zapadla do zárezu v rúrke. Tvar klapky zabezpečí uzavretie rúrky. Vzhľadom na nápor vyvíjaný padajúcim krmivom bola klapka vyrobená z plechu, aby sa nepokrivila alebo nepretrhla. Pohyb krmiva pri otvorenej klapke je zabezpečený využitím naklonenej plastovej rúrky a gravitácie.



*Obrázok 3 Umiestnenie servo motorčeka a tvar klapky*

Náš program vytvorí objekt Servo s menom Myservo. Funkcia Myservo.attach(pin) povie programu na ktorom pine je servo pripojené, v našom prípade je to pin 5.

Funkcia Myservo.write(hodnota) natočí motor na aktuálny uhol v rozmedzí od nula až po 180 stupňov. V našom riešení využívame uhol otočenia nula stupňov na zatvorenie klapky - nechceme sypať žiadne jedlo a uhol 90 stupňov na otvorenie klapky. Využíva sa vo funkciách loop() a nasypanie\_manual().

Príklady použitia:

#include <Servo.h>

Servo Myservo; //servo premenna

void setup() { //setup

lcd\_init();

pinMode(2,INPUT);

Myservo.attach(5);

Myservo.write(0);

IrReceiver.begin(IR\_RECEIVE\_PIN);

pinMode(SIGNAL\_PIN, INPUT);

}

*Výpis 2 Pripojenie knižnice, vytvorenie objektu Servo a funkcia setup()*

Myservo.write(90);

delay(1000 \* charge);

Myservo.write(0);

delay(5000);

*Výpis 3 Natočenie motora na aktuálny uhol*

### Zisťovanie prítomnosti zvieraťa pomocou senzora pohybu

Ako fyzický model senzora sme zvolili senzor HC-SR505. Tento malý model senzora dostatočne postačuje na naše riešenie. Jeho malý dosah (3 metre) nie je nedostatkom vzhľadom na využitie kŕmidlom (detekovanie pohybu zvieraťa v tesnej blízkosti). Senzor pohybu pri zachytení pohybu vyšle signál reprezentovaný výrazným nárastom napätia na jeho pine OUT.

Po uplynutí nami stanoveného času (ako často sa nasype dávka), bude jedlo vysypané zvieratku len po splnení podmienky, že senzor zaregistruje pohyb. Senzor pohybu je umiestnený v dolnej časti tej strany fyzickej konštrukcie kŕmidla, z ktorej vypadáva krmivo. Vďaka tomu je senzor schopný zaregistrovať aj malé zviera. Vzhľadom na uhol a maximálnu vzdialenosť, v ktorej je senzor schopný zaznamenať pohyb je ale senzor schopný zaznamenať aj pohyb človeka, čo je nežiadúce a nedostatkom nášho riešenia. Pri optimálnych podmienkach (zvieratko je samo doma) táto schopnosť nie je nedostatkom.

Náš program funguje nasledovne: zadefinovali sme pin 8 ako pin signálu zo senzoru pohybu. Tento pin sme nastavili ako vstupný vo funkcii setup(). Vo funkcii nasypanie() načítame informáciu z pripojeného vstupu, a ak je rovná makru HIGH (ktoré symbolizuje logickú 1) senzor zaregistroval pohyb. Vo funkcii nasypanie() sa premenná nasypane nastaví na true čo spôsobí koniec cyklu, ktorý číta informáciu zo vstupu a zavolá sa funkcia nasypanie\_manual().

Príklady použitia:

#define SIGNAL\_PIN 8 //definovanie pinov

void setup() { //setup

lcd\_init();

pinMode(2,INPUT);

Myservo.attach(5);

Myservo.write(0);

IrReceiver.begin(IR\_RECEIVE\_PIN);

pinMode(SIGNAL\_PIN, INPUT);

}

*Výpis 4 Definovanie pinu a funkcia setup()*

while(nasypane==false){

if(digitalRead(SIGNAL\_PIN)==HIGH) {

nasypanie\_manual();

nasypane=true;

}

}

*Výpis 5 Využitie senzora vo funkcii nasypanie()*

### Komunikácia s používateľom

Na zabezpečenie komunikácie kŕmidla s používateľom sme sa rozhodli použiť LCD displej is I2C modulom pre jednoduchšie pripojenie než samotný displej s dvoma riadkami a 16-timi stĺpcami.

Pomocou príkazu LiquidCrystal\_I2C lcd(0x27,16,2) vytvoríme nový objekt LCD pre prácu s displejom. Displej bude pripojený na adrese 39 (27 v hexadecimálnej sústave), bude mať 2 riadky a 16 znakov v riadku, keďže toto sú rozmery našeho fyzického LCD displeja. Knižnica LiquidCrystal\_I2C.h využíva analógový pin 4 pre pin SDA I2C modulu a analógový pin 5 pre pin SCL I2C modulu.

#include <LiquidCrystal\_I2C.h>

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

void lcd\_init(){ //inicializacia displeja

lcd.clear();

lcd.init();

lcd.backlight();

}

void setup() { //setup

lcd\_init();

pinMode(2,INPUT);

Myservo.attach(5);

Myservo.write(0);

IrReceiver.begin(IR\_RECEIVE\_PIN);

pinMode(SIGNAL\_PIN, INPUT);

}

*Výpis 6 Pripojenie knižnice, vytvorenie objektu lcd , funkcia lcd\_init() a setup()*

Vo funkcii lcd\_init() inicializujeme LCD displej reprezentovaný objektom lcd pomocou metódy init() objektu lcd. Po inicializácii LCD displeja je zapnuté podsvietenie metódou backlight() objektu lcd. Metóda print() objektu lcd vypíše na displej zadaný text alebo číslo. Metóda setCursor() objektu lcd nastaví kurzor na zadanú pozíciu stĺpca (prvý parameter) a riadku (druhý parameter). Metóda clear() objektu lcd zmaže všetky znaky zobrazené na displeji a kurzor nastaví do ľavého horného rohu.

Výpisy na displeji slúžia na komunikáciu s používateľom - vypisujú mu hodnoty aké aktuálne nastavuje (môže to byť čas alebo veľkosť dávky). Okrem toho, po potvrdení všetkých nastavení displej zobrazí odčítavanie do najbližšieho nasypania dávky. Komunikuje s používateľom o čakaní na zvieratko alebo o úspešnom nasypaní. Výpisy na displej využívame v hlavnej funkcii loop(), aj vo funkciách nasypanie() a nasypanie\_manual().

Príklady výpisu:

lcd.setCursor(2,0);

lcd.print("Nastav cas");

if(command==IR\_BUTTON\_PLUS){ //nastavenie casu + 15 minut

minutes = minutes + 15;

if(minutes > 180){

minutes = 15;

}

lcd.setCursor(1,1);

sprintf(timer,"%02d hod %03d min",minutes / 60, minutes % 60);

lcd.print(timer);

delay(400); //caka 400ms

continue;

}

*Výpis 7 Útržok kódu funkcie loop() - príklad využitia displeja*

lcd.setCursor(2,0);

lcd.print("Nastav davku");

if(command==IR\_BUTTON\_PLUS){ //davka sa prida

charge = charge + 1;

if(charge > 3){

charge = 1;

}

lcd.setCursor(2,1); //printovanie aktualne nastavenej davky

sprintf(timer,"%01d nasobna",charge);

lcd.print(timer);

delay(400);

continue;

}

*Výpis 8 Útržok kódu funkcie loop() - príklad využitia displeja 2*

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Cakam na teba");

lcd.setCursor(2,1);

lcd.print("moj milacik!");

*Výpis 9 Výpis na displej vo funkcii nasypanie()*

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Nech sa paci!");

lcd.setCursor(2,1);

lcd.print("Dobru chut!");

*Výpis 10 Výpis na displej vo funkcii nasypanie\_manual()*

### Ovládanie kŕmidla pomocou diaľkového ovládača a IR prijímača

Okrem displeja komunikáciu medzi používateľom a kŕmidlom sprostredkuje aj ovládač. Konkrétne umožňuje ovládať kŕmidlo, teda nastavovať časovač, veľkosť dávky a spustenie manuálneho dávkovania. Pre neho je kľúčový IR prijímač, keďže umožňuje detekovať signál z IR ovládača, ktorý je naňho namierený. Pomocou Arduina sú tieto signály dekódované a podľa nich sú vykonávané funkcie kŕmidla.

Náš kód funguje nasledovne: ak chceme čítať infračervené lúče z diaľkového ovládania Arduino doskou, potrebujeme pripojiť externú knižnicu IRremote. Ďalej nastavíme číslo pinu pre prepojenie s Arduinom (v našom prípade je to pin 2). Vo funkcii setup() začneme pomocou IrReceiver.begin(IR\_RECEIVE\_PIN) komunikáciu s IR prijímačom.

#include <IRremote.h>

#define SIGNAL\_PIN 8 //definovanie pinov

#define IR\_RECEIVE\_PIN 2

#define IR\_BUTTON\_100 25

#define IR\_BUTTON\_PLUS 21

#define IR\_BUTTON\_MINUS 7

#define IR\_BUTTON\_PLAY 67

#define IR\_BUTTON\_EQ 9

#define IR\_BUTTON\_CH 70

int command; //navratova hodnota z receiveru

void setup() { //setup

lcd\_init();

pinMode(2,INPUT);

Myservo.attach(5);

Myservo.write(0);

IrReceiver.begin(IR\_RECEIVE\_PIN);

pinMode(SIGNAL\_PIN, INPUT);

}

*Výpis 11 Pripojenie knižnice, definovanie pinov a funkcia setup()*

Metóda decode() objektu IrReceiver vracia true ak je dostupný infračervený signál, ktorý bol dekódovaný, a uloží jeho dekódovanú hodnotu do premennej command patriacemu objektu decodedIRData patriacemu objektu IrReceiver. Pri splnení podmienky, že je signál dostupný si uložíme premennú command objektu reprezentujúcu rozpoznaný príkaz, ktorý reprezentuje stlačené tlačidlo, do pomocnej premennej command. Metóda resume() objektu IrReceiver zresetuje jeho stav, čo umožní prijímanie ďalších signálov.

if(IrReceiver.decode()){

IrReceiver.resume();

command=IrReceiver.decodedIRData.command;

}

*Výpis 12 Príklad načítania dát IRreceiverom a dekódovanie do pomocnej premennej command*

Máme rôzne možnosti, aké tlačidlá môže používateľ stlačiť (pozri výpis 11). Názvy definovaných makier používajú značenie obdobné tomu, ktoré je použité na diaľkovom ovládači. Napríklad IR\_BUTTON\_PLUS reprezentuje hodnotu premennej command, keď je stlačené tlačidlo „+“ na diaľkovom ovládači. Všetky tlačidlá sú použité vo funkcii loop() a pri každom tlačidle sa vykoná iný kód. Stlačenie tlačidla IR\_BUTTON\_EQ vymaže nastavený čas, čo spôsobí, že pri vykonaní ďalšieho cyklu funkcie loop() je znovu vykonaný kód umožňujúci nastavenie času. Stlačenie tlačidla IR\_BUTTON\_CH vymaže nastavenú dávku, čo umožní znovu nastaviť veľkosť dávky. Stlačenie tlačidla IR\_BUTTON\_PLUS slúži na zvyšovanie času medzi kŕmeniami o 15 minút alebo zväčšovanie dávky. A naopak stlačenie tlačidla IR\_BUTTON\_MINUS slúži na znižovanie času medzi kŕmeniami o 15 minút alebo zmenšovanie dávky. Stlačenie tlačidla IR\_BUTTON\_100 slúži na potvrdenie nastaveného času alebo dávky. Stlačenie posledného možného tlačidla IR\_BUTTON\_PLAY, zavolá funkciu nasypanie\_manual(). Tá spôsobí, že krmivo bude okamžite vysypané bez čakania na senzor pohybu.

if(command==IR\_BUTTON\_EQ){ //skoci na zaciatok programu, kde sa nastavuje cas

pom\_minutes=0;

minutes = 0;

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Nastav cas");

break;

}

*Výpis 13 Správanie kódu vo funkcii loop() po stlačení tlačidla „EQ“*

else if(command == IR\_BUTTON\_CH){ //skoci na zaciatok programu kde sa nastavuje davka

setcharge = 0;

charge = 0;

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Nastav davku");

break;

}

*Výpis 14 Správanie kódu vo funkcii loop() po stlačení tlačidla „CH“*

if(command==IR\_BUTTON\_PLUS){ //nastavenie casu + 15 minut

minutes = minutes + 15;

if(minutes > 180){

minutes = 15;

}

lcd.setCursor(1,1);

sprintf(timer,"%02d hod %03d min",minutes / 60, minutes % 60);

lcd.print(timer);

delay(400); //caka 400ms

continue;

}

*Výpis 15 Správanie kódu vo funkcii loop() po stlačení tlačidla „+“ pri nastavovaní času*

if(command==IR\_BUTTON\_PLUS){ //davka sa prida

charge = charge + 1;

if(charge > 3){

charge = 1;

}

lcd.setCursor(2,1); //printovanie aktualne nastavenej davky

sprintf(timer,"%01d nasobna",charge);

lcd.print(timer);

delay(400);

continue;

}

*Výpis 16 Správanie kódu vo funkcii loop() po stlačení tlačidla „+“ pri nastavovaní dávky*

else if(command==IR\_BUTTON\_MINUS){ //nastavenie casu -15minut

minutes = minutes - 15;

if(minutes < 15){

minutes = 180;

}

lcd.setCursor(1,1);

sprintf(timer,"%02d hod %03d min",minutes / 60, minutes % 60);

lcd.print(timer);

delay(400);

continue;

}

*Výpis 17 Správanie kódu vo funkcii loop() po stlačení tlačidla „-“ pri nastavovaní času*

else if(command==IR\_BUTTON\_MINUS){ //davka sa odoberie

charge = charge - 1;

if(charge < 1){

charge = 3;

}

lcd.setCursor(2,1);

sprintf(timer,"%01d nasobna",charge);

lcd.print(timer);

delay(400);

continue;

}

*Výpis 18 Správanie kódu vo funkcii loop() po stlačení tlačidla „-“ pri nastavovaní dávky*

else if(command == IR\_BUTTON\_100 && minutes > 0){ //potvrdenie casu

pom\_minutes = minutes;

break;

}

*Výpis 19 Správanie kódu vo funkcii loop() po stlačení tlačidla „100+“ pri nastavovaní času*

else if(command == IR\_BUTTON\_100 && charge > 0){ //potvrdenie davky

setcharge = charge;

break;

}

*Výpis 20Správanie kódu vo funkcii loop() po stlačení tlačidla „100+“ pri nastavovaní dávky*

if(command==IR\_BUTTON\_PLAY){ //manualne nasypanie

nasypanie\_manual();

}

*Výpis 21 Správanie kódu vo funkcii loop() po stlačení tlačidla „PLAY/PAUSE“*

Ako sme už vyššie spomenuli, IR prijímač a ovládač sú pre náš projekt kľúčový, keďže bez nich by nebolo možné nastavovať veľkosť dávky či časové rozostupy medzi sypaním krmiva.

### Funkcia void loop()

Funkcia loop je hlavná funkcia, ktorá sa dookola vykonáva. Jej neustále vykonávanie je zaručené funkcionalitou funkcie main() vytvorenej vývojovým prostredím Arduino IDE. Neustále prijíma dáta z prijímača a dekóduje ich. Na začiatku tejto funkcie sa pomocné premenné minúty (minutes) a dávka (charge) nastavia na nulu. Hneď potom (prípadne po stlačení tlačidla EQ) sa dá nastaviť čas, ako často má byť nasypaná dávka. Môže to byť od každých 15 minút až do 3 hodín. Používateľ zvyšuje/znižuje čas pomocou tlačidiel „+“ a „-“ diaľkového ovládača. Používateľ svoju voľbu potvrdí tlačidlom „100+“. Potom (prípadne po stlačení tlačidla „CH“) sa dá nastaviť veľkosť dávky. Používateľ má na výber 3 veľkosti. Tak ako pri čase, dávku zvyšuje/znižuje pomocou „+“/„-“ a svoju voľbu potvrdí pomocou tlačidla „100+“. Na displeji sa nastaví používateľom vybratý čas a začne odpočítavanie. Každú sekundu sa displej aktualizuje až kým nevyprší čas(t.j. pomocné premenné minutes a seconds budú rovné nule). Hneď potom sa zavolá sa funkcia nasypanie. Dokončenie tejto funkcie spúšťa časovač odznova vo funkcii loop(). Funkcia loop() okrem toho kontroluje či nebolo stlačené tlačidlo „EQ“, „CH“ alebo „PLAY/PAUSE“ ktorá volá funkciu nasypanie\_manual().

int minutes; //minuty

int pom\_minutes=0; //pomocna pre minuty

int charge; //premenna pre davku

int setcharge=0; //pomocna premenna pre davku

int seconds; //sekundy

char timer[16]; //pole na print casu

int command; //navratova hodnota z receiveru

*Výpis 22 Nastavenie všetkých pomocných premenných*

void loop()

{

minutes = pom\_minutes;

charge = setcharge;

lcd.clear();

while(pom\_minutes==0){ //tu vojde prvy krat, aj ked je stlacene tlacitko EQ

lcd.setCursor(2,0);

lcd.print("Nastav cas");

if(IrReceiver.decode()){

IrReceiver.resume();

command=IrReceiver.decodedIRData.command;

lcd.setCursor(2,0);

lcd.print("Nastav cas");

if(command==IR\_BUTTON\_PLUS){ //nastavenie casu + 15 minut

minutes = minutes + 15;

if(minutes > 180){

minutes = 15;

}

lcd.setCursor(1,1);

sprintf(timer,"%02d hod %03d min",minutes / 60, minutes % 60);

lcd.print(timer);

delay(400); //caka 400ms

continue;

}

else if(command==IR\_BUTTON\_MINUS){ //nastavenie casu -15minut

minutes = minutes - 15;

if(minutes < 15){

minutes = 180;

}

lcd.setCursor(1,1);

sprintf(timer,"%02d hod %03d min",minutes / 60, minutes % 60);

lcd.print(timer);

delay(400);

continue;

}

else if(command == IR\_BUTTON\_100 && minutes > 0){ //potvrdenie casu

pom\_minutes = minutes;

break;

}

}

*Výpis 23 Funkcia void loop()*

if(command==IR\_BUTTON\_PLAY){ //manualne nasypanie

nasypanie\_manual();

}

}

lcd.clear();

while(setcharge == 0){ //tu vojde prvy krat, aj ked je stlacene tlacitko CH

lcd.setCursor(2,0);

lcd.print("Nastav davku");

if(IrReceiver.decode()){

IrReceiver.resume();

command=IrReceiver.decodedIRData.command;

lcd.setCursor(2,0);

lcd.print("Nastav davku");

if(command==IR\_BUTTON\_PLUS){ //davka sa prida

charge = charge + 1;

if(charge > 3){

charge = 1;

}

lcd.setCursor(2,1); //printovanie aktualne nastavenej davky

sprintf(timer,"%01d nasobna",charge);

lcd.print(timer);

delay(400);

continue;

}

else if(command==IR\_BUTTON\_MINUS){ //davka sa odoberie

charge = charge - 1;

if(charge < 1){

charge = 3;

}

lcd.setCursor(2,1);

sprintf(timer,"%01d nasobna",charge);

lcd.print(timer);

delay(400);

continue;

}

else if(command == IR\_BUTTON\_100 && charge > 0){ //potvrdenie davky

setcharge = charge;

break;

}

}

if(command==IR\_BUTTON\_PLAY){ //manualne nasypanie

nasypanie\_manual();

}

}

seconds = 1;

lcd.clear();

while(minutes>=0){ //po spravnom nastaveni casu tu vojde

lcd.setCursor(2,0);

lcd.print("Dalsia davka");

lcd.setCursor(1,1); //aktualizacia printovaneho casu na displeji

sprintf(timer,"%01d h %02d m %02d s",minutes / 60, minutes % 60 ,seconds);

lcd.print(timer);

delay(1000);

*Výpis 24 Funkcia void loop() – pokračovanie*

seconds--; //odpocitanie sekund

if(seconds==0&&minutes>0){ //odpocitanie minuty a nastavnie na 60 sek

seconds=60;

minutes--;

}

if(seconds==0&&minutes==0){ //ak vyprsal cas tak nasype

nasypanie();

break;

}

if(IrReceiver.decode()) {

IrReceiver.resume();

command=IrReceiver.decodedIRData.command;

if(command==IR\_BUTTON\_EQ){ //skoci na zaciatok programu, kde sa nastavuje cas

pom\_minutes=0;

minutes = 0;

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Nastav cas");

break;

}

else if(command == IR\_BUTTON\_CH){ //skoci na zaciatok programu kde sa nastavuje davka

setcharge = 0;

charge = 0;

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Nastav davku");

break;

}

else if(command==IR\_BUTTON\_PLAY){ //manualne nasypanie

nasypanie\_manual();

break;

}

}

Myservo.write(0);

}

}

*Výpis 25 Funkcia void loop() - pokračovanie 2*

### Funkcia void nasypanie()

Po zavolaní funkcie nasypanie() sa do pomocnej premennej bool nasypane uloží hodnota false. Na displeji sa vypíše text: „Cakam na teba moj milacik!“. V cykle podmienenom tým, že je premenná nasypane false, sa kontroluje či neprišiel signál zo senzora. Až keď senzor pohybu zaregistruje pohyb je zavolaná funkcia nasypanie\_manual a hodnota premennej nasypane sa zmení na true, čo ukončí cyklus ako aj celú funkciu nasypanie(). Funkcionalita funkcie nasypanie\_manual vo funkcii nasypanie() je použitá za účelom neduplikovania kódu a zmenšením úložného priestoru Arduina, ktorý je zabraný kódom.

void nasypanie(){ //funkcia pre printovanie a osetrenie stavu serva pri nasypani

bool nasypane=false;

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Cakam na teba");

lcd.setCursor(2,1);

lcd.print("moj milacik!");

while(nasypane==false){

if(digitalRead(SIGNAL\_PIN)==HIGH) {

nasypanie\_manual();

nasypane=true;

}

}

}

*Výpis 26 Funkcia nasypanie()*

### Funkcia void nasypanie\_manual()

Po zavolaní funkcie nasypanie\_manual() sa na displeji vypíše text: „Nech sa paci! Dobru chut!“ . Servo je otočené na uhol 90 stupňov. Takto otočené zotrvá od jednej až do troch sekúnd v závislosti od veľkosti dávky. Po uplynutí tohto času je servo otočené naspäť na uhol 0 stupňov. V tomto stave kŕmidlo čaká 5 sekúnd.

void nasypanie\_manual(){ //samotne nasypanie pomocou serva

lcd.clear();

lcd.setCursor(2,0);

lcd.print("Nech sa paci!");

lcd.setCursor(2,1);

lcd.print("Dobru chut!");

Myservo.write(90);

delay(1000 \* charge);

Myservo.write(0);

delay(5000);

}

*Výpis 27 Funkcia nasypanie\_manual*

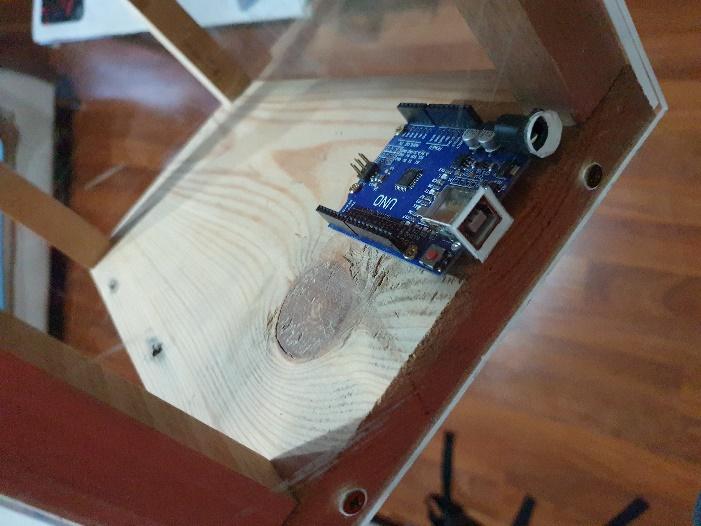
Zhotovenie nosnej konštrukcie

Pri zhotovovaní nosnej konštrukcie sme sa rozhodli pre materiály, ktoré sú ako bežne dostupné, tak aj odolné. Rám celej konštrukcie je vyrobený z drevených odrezkov spojených skrutkami. Okrem toho, že je tento rám ekologický a recyklovaný, je aj dostatočne pevný a ťažký, čo mu poskytuje odolnosť voči prevrhnutiu zvieraťom.

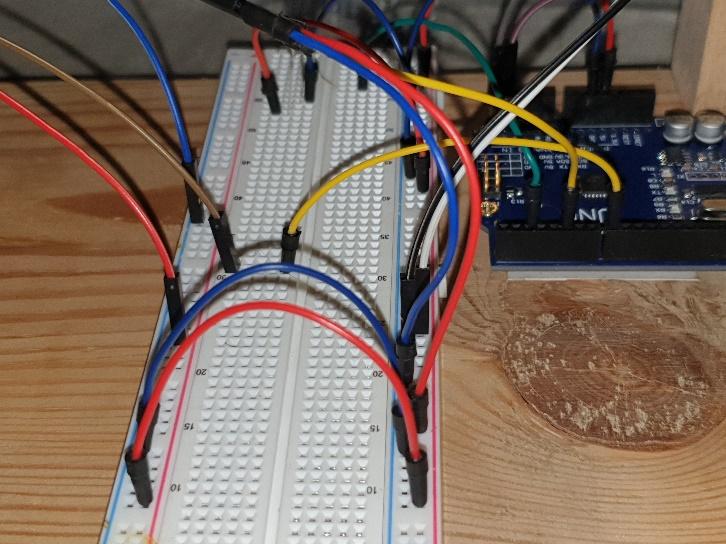


*Obrázok 4 Konštrukcia rámu*

Vývojová doska Arduino je prichytená k rámu skrutkami. Nepájivé pole je k rámu nalepené obojstrannou lepiacou páskou. Zadná a jedna z bočných strán kŕmidla je zhotovená z odrezkov z plexiskla. Do zadnej strany je vyvŕtaný otvor na napojenie zdroja elektrickej energie a USB kábla do dosky Arduino. K bočnej strane z plexiskla je prichytený LCD displej. Využitie plexiskla umožňuje prichytenie displeja k pevnému materiálu, ktorý je zároveň priehľadný. Senzor pohybu je pripevnený k rámu kúskom plastu, ktorý je k rámu veľmi pevno priskrutkovaný. Toto konštrukčné prevedenie poskytuje odolnosť voči prevrhnutiu alebo nárazom, ktoré sú nežiadúce pre korektné fungovanie jeho elektronických súčiastok, a výskyt ktorých je predpokladaný v prípade použitia so zvieratami.



*Obrázok 5 Pripevnenie dosky Arduino a zadná strana kŕmidla*



*Obrázok 6 Umiestnenie nepájivého poľa*



*Obrázok 7 Uchytenie LCD displeja na plexiskle*

Plastová rúrka, ku ktorej je prichytené servo s klapkou, je modifikovaným odrezkom plastovej rúrky s priemerom 50mm. Táto rúrka je prichytená k vrchnej strane rámu skrutkami. Držiak zásobníka je realizovaný využitím krytu na plastové rúrky s priemerom 50mm priskrutkovaným k otvoru vyvŕtanému do hornej časti rámu, do ktorého je možné zasunúť takúto rúrku. Zasunutá rúrka je rovnakého priemeru ako výdajná rúrka, čo zabezpečí, že všetko krmivo zo zásobníka prepadne len do nej. Toto prevedenie umožňuje konštrukciu viacerých zásobníkov krmiva z plastových rúr. Pre naše riešenie bol skonštruovaný jeden zásobník prilepením PET fľaše k odrezku plastovej rúrky, ako menší zásobník slúži ďalší kus rúrky.



*Obrázok 8 Prichytenie naklonenej rúrky*



*Obrázok 9 Prichytenie držiaka zásobníka*



*Obrázok 10 Rám s prichytenými súčiastkami*

Predná a druhá bočná strana sú z kartónu, keďže je doň jednoduché vyrezať otvor pre naklonenú rúrku a senzor pohybu a keďže tieto strany nenesú žiadne elektronické súčiastky. Výsledné kŕmidlo má šírku 32 centimetrov, výšku 20 centimetrov a hĺbku 18 centimetrov.



*Obrázok 11 Prichytenie stien z kartónu*

# Postup ovládania

V tejto kapitole je postupne rozobratý postup ovládania zostrojeného kŕmidla

Pri prvotnom zapojení kŕmidla k zdroju elektrickej energie sa rozsvieti LCD displej a zobrazí text „Nastav cas“, ktorým žiada používateľa o nastavenie časovača. Na tento krok v nastavení je možné sa dostať kedykoľvek za odpočítavania stlačením tlačidla „EQ“.



*Obrázok 12 Počiatočný výpis po zapojení kŕmidla*

Po stlačení tlačidla „+“ na diaľkovom ovládači sa pridá 15 minút k časovaču alebo pri presiahnutí hodnoty 3 hodiny sa časovač nastaví naspäť na 15 minút. Po stlačení tlačidla „-“ sa z časovača uberie 15 minút alebo sa časovač nastaví na 3 hodiny, ak by sa jeho hodnota mala nastaviť na nulu. Práve nastavená hodnota časovača sa vypíše na LCD displej.



*Obrázok 13 Príklad výpisu nastaveného času po pridaní alebo odobraní*

Potvrdenie nastaveného času a jeho zapamätanie sa realizuje tlačidlom „100+“. Následne kŕmidlo vyzve používateľa na zadanie veľkosti dávky krmiva, ak táto ešte nebola nastavená. Na tento krok v nastavení je možné sa opätovne dostať stlačením tlačidla „CH“ na diaľkovom ovládači pri odpočítavaní časovača.



*Obrázok 14 Výpis vyžadujúci nastavenie dávky od používateľa*

Rovnako ako aj pri časovači sa dávka pridáva tlačidlom „+“ a uberá tlačidlom „-“. Jej maximálna hodnota je trojnásobná (veľká) dávka a jej minimálna hodnota je jednonásobná (malá) dávka. Jej nastavená hodnota je následne vypísaná na LCD displej.



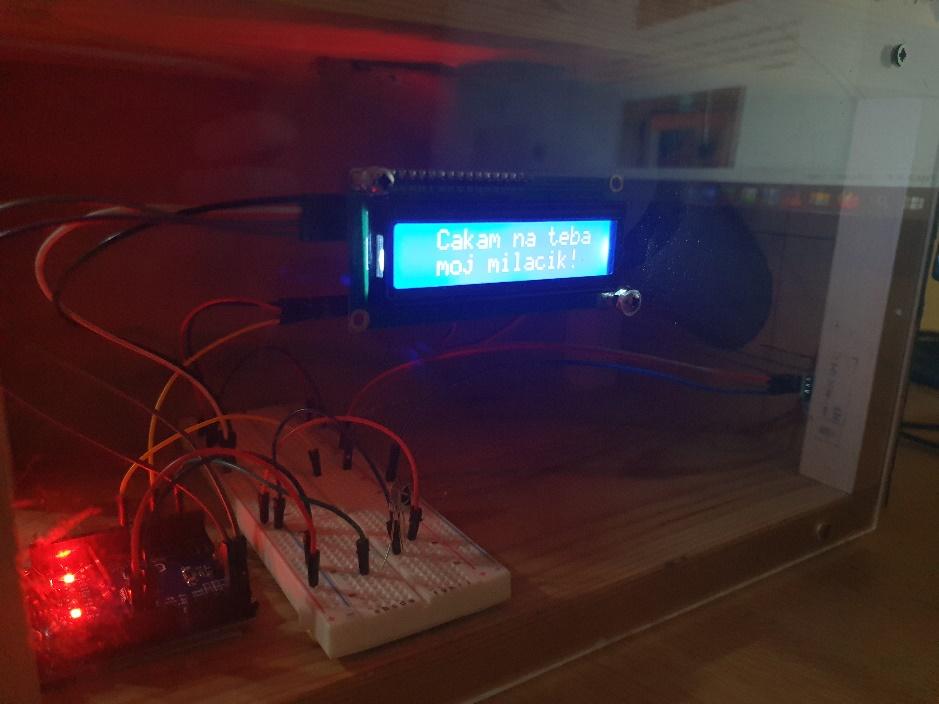
*Obrázok 15 Výpis nastavenej hodnoty dávky*

Jej uloženie sa taktiež potvrdzuje tlačidlom „100+“. Po jeho stlačení začína odpočet nastaveného času časovača. Kŕmidlo aktualizuje výpis zakaždým, čo sa z časovača odpočíta jedna sekunda.



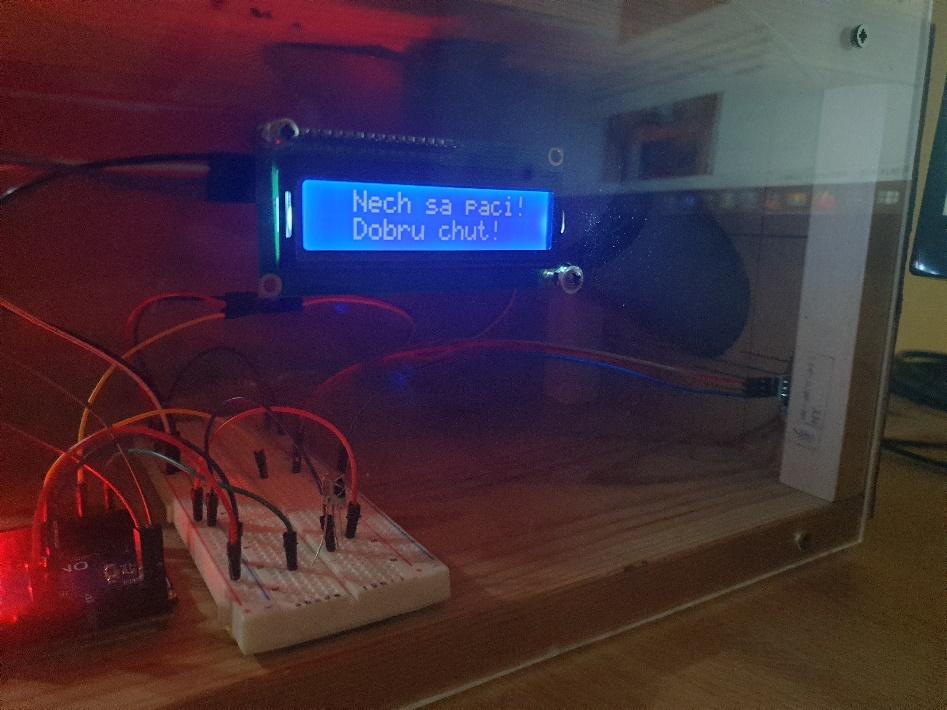
*Obrázok 16 Výpis odpočítavania časovača*

Po uplynutí časovača, kedy je jeho hodnota 0 minút a 0 sekúnd kŕmidlo čaká na signál zo senzora pohybu. Túto skutočnosť dáva používateľovi najavo výpisom informácie „Cakam na teba moj milacik!“ na displej.



*Obrázok 17 Oznam o čakaní na pohyb*

Po zaregistrovaní pohybu v tomto stave senzor pohybu vyšle signál. Tento signál spôsobí otvorenie klapky a nasypanie krmiva. Oznam o tom, že bolo krmivo vysypané, je vypísaný na displej.



*Obrázok 18 Oznam o nasypaní krmiva*



*Obrázok 19 Stav klapky pred nasypaním*

*Obrázok 20 Stav klapky počas nasypania*

Nasypanie krmiva je možné dosiahnuť aj stlačením tlačidla „PLAY/PAUSE“ pri odpočítavaní časovača. Výpis je v tomto prípade rovnaký ako po vypršaní časovača.



*Obrázok 21 Výpis po stlačení tlačidla „PLAY/PAUSE“*

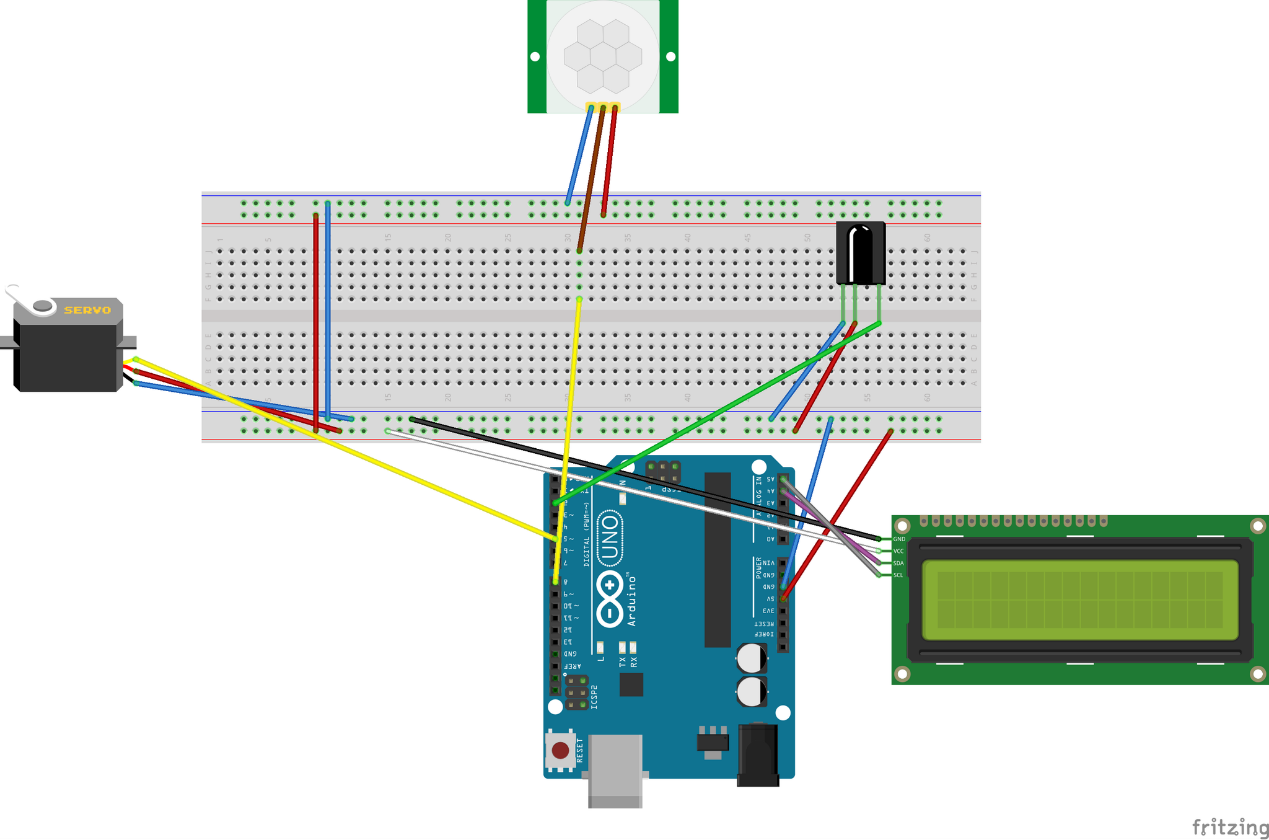
Názorné ukážky použitia sú taktiež prezentované v priloženom dokumentačnom videu.

# Nedostatky riešenia

Riešenie sa potýka s viacerými nedostatkami. Aj keď je jeho rám dostatočne pevný, neposkytuje úplnú odolnosť voči nárazom alebo prevráteniu. Použitie rúrok, ktoré nie sú spojené ako prechod medzi zásobníkom a výdajnou rúrkou môže spôsobiť uviaznutie krmiva, ak je toto zložené z granulátu dostatočnej veľkosti. Uchytenie servo motora a prichytenie klapky môže spôsobiť zaseknutie klapky v rúrke. Tieto chyby môžu spôsobiť nevypadnutie žiadneho krmiva v momente, kedy by malo byť vydané. Vypadávanie krmiva do misky je relatívne hlučné, čo môže vystrašiť niektoré zvieratá. Pri testovaní bolo zistené, že nami použitý diaľkový ovládač a IR prijímač nevykazujú dostatočnú spoľahlivosť na bežnú prevádzku, niektoré stlačenia sa nemusia zaznamenať, alebo sa naopak zaznamenajú dvakrát. Ako už bolo vyššie spomenuté, senzor pohybu nie je schopný rozoznať, či pohyb vykonalo zviera alebo človek.

# Fotodokumentácia k projektu

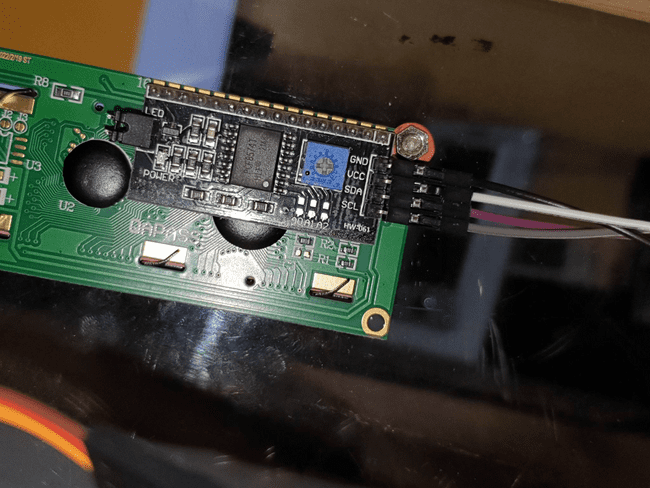
V tejto kapitole rozoberieme celkové zapojenie systému - znázornené v nasledujúcej schéme, vytvorenej v programe Fritzing.



*Schéma 2 Schéma zapojenia projektu*

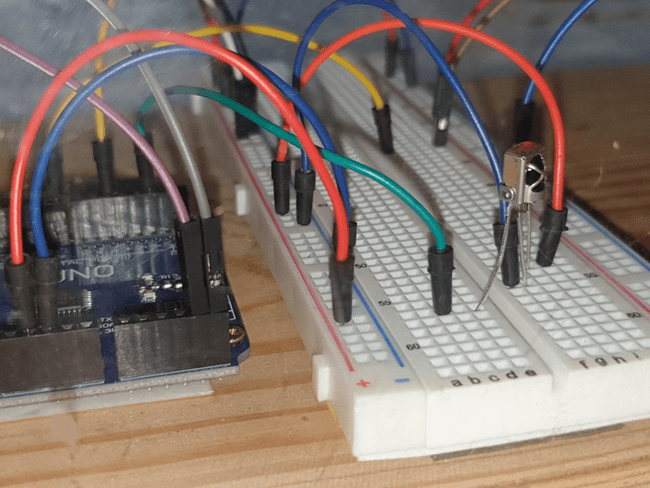
Riadiacou jednotkou nášho projektu je programovateľná doska Arduino Uno. Jednotlivé komponenty sme prepojili pomocou kontaktného poľa.

Ako prvý sme zapojili LCD displej s I2C modulom. Ground pin bol pripojený do mínusu (modrá) na kontaktnom poli čiernym vodičom. VCC pin bol pripojený do plusu (červená) na kontaktnom poli bielym vodičom . SDA pin(data signal) bol napojený na fialový vodič a SCL pin (clock signal) bol pripojený na sivý vodič.

**

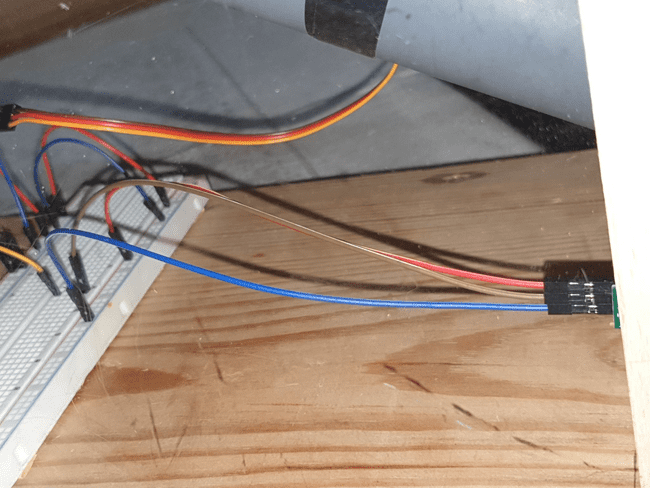
*Obrázok 22 Krok 1 - pripojenie vodičov k LCD displeji s I2C modulom*

Druhý sme zapojili IR prijímač ktorý má 3 piny. Jeho OUT pin (ľavý) bol zapojený na riadok 57 kontaktného poľa zeleným vodičom. GND (stredný) pin bol zapojený na riadok 55 kontaktného poľa a spojený s mínusom kontaktného poľa modrým vodičom. VCC (pravý) pin bol zapojený na riadok 53 kontaktného poľa a spojený s plusom kontaktného poľa červeným vodičom.



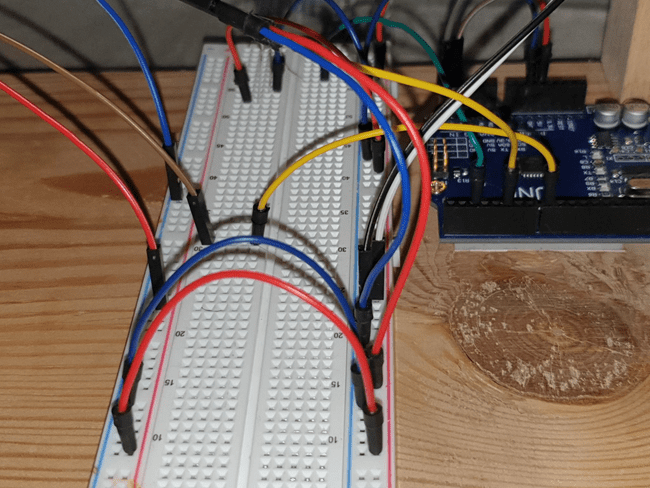
*Obrázok 23 Krok 2 - zapojenie IR prijímača na kontaktné pole*

Senzor pohybu má také isté označenie pinov ako IR prijímač. Pin VCC bol pripojený červeným vodičom k plusu kontaktného poľa a pin GND do mínusu kontaktného poľa modrým vodičom. Pin OUT bol pripojený na riadok 21 kontaktného poľa hnedým vodičom.



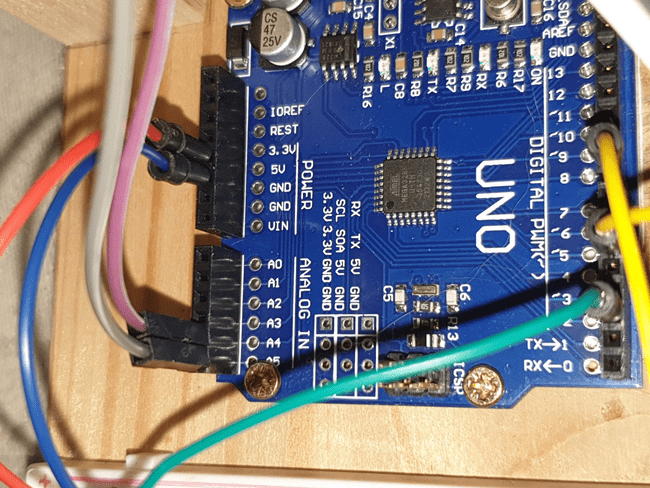
*Obrázok 24 Krok 3 – zapojenie senzoru pohybu*

Následne bol pripojený servo motor, ktorý má 3 piny. VCC pin bol pripojený k plusu kontaktného poľa červeným vodičom. GND pin bol pripojený k mínusu kontaktného poľa modrým vodičom. Signal pin musel byť pripojený k pinu s možnosťou PWM modulácie, a preto bol pripojený priamo k pinu 5 vývojovej dosky Arduino žltým vodičom, ktorý túto moduláciu podporuje.



*Obrázok 25 Krok 4 – zapojenie servo motora a prepojenie plus a mínus riadkov poľa*

V poslednom kroku boli všetky ešte nezapojené súčiastky pripojené na vývojovú dosku Arduino. Riadok plus kontaktného poľa bol napojený na pin 5V vývojovej dosky a riadok mínus na pin GND vývojovej dosky. Fialový vodič (SDA LCD displeja) bol napojený na pin A4 a sivý vodič (SCL LCD displeja) na pin A5. Zelený vodič (OUT pin IR prijímača) bol pripojený k riadku 57 kontaktného poľa a pinu 2 vývojovej dosky. Nakoniec bol zostávajúci žltý vodič pripojený na riadok 21 kontaktného poľa a pin 8, čím pripojil OUT pin senzora pohybu k vývojovej doske. Celkové znázornenie zapojenia je taktiež znázornené v Schéme 2.

**

*Obrázok 26 Krok 5 – zapojenie vodičov na piny vývojovej dosky Arduino*

# Možné rozšírenia

Konštrukcia kŕmidla by sa dala vylepšiť použitím plastových dielcov namiesto kartónových a použitím pevnejšieho systému na úchyt zásobníka. Rozmery konštrukcie je možné zmenšiť bez straty miesta na uloženie súčiastok. Zásobníky je možné skonštruovať z pevnejšieho plastu, aby boli odolnejšie a schopné poňať väčšie množstvo krmiva. K spodnej časti rámu by bolo možné pripevniť držiak na misku, ktorý by nedovolil zvieraťu prevrhnúť alebo posunúť misku s krmivom. Pri zavedení kŕmidla do výroby by bolo možné navrhnúť nový dizajn masovo produkovaného plastového rámu, ktorý by zahrnul všetky vyššie spomenuté rozšírenia.

Odstránenie možného elektrického šumu senzora pohybu a infračerveného prijímača by sa dalo odstrániť použitím kondenzátorov pripojených k týmto súčiastkam. Namiesto infračerveného prijímača by sa dalo možné implementovať ovládanie pomocou mobilnej aplikácie a WiFi alebo Bluetooth modulu, ako bolo uvedené na začiatku kapitoly Riešenie. Kŕmidlo by bolo možné rozšíriť o detekciu prázdneho zásobníka a prázdnej misky. Takéto riešenie by využívalo senzor hmotnosti na zistenie, či je zásobník alebo miska prázdna. Pri prázdnom zásobníku by kŕmidlo signalizovalo LED diódou alebo push notifikáciou, že je jeho zásobník prázdny. Naopak, kŕmidlo by bolo schopné nasypať krmivo len ak uplynul čas, bol detekovaný pohyb a miska s krmivom je prázdna, čo by spôsobilo, že sa krmivo nebude zbytočne hromadiť v miske. Pri tomto riešení by sa navyše vyskytla možnosť dávkovať krmivo podľa hmotnosti.

Keďže senzor pohybu nerozlišuje medzi mačkou a človekom (a ani žiadnym iným pohybom), táto funkcionalita by sa dala realizovať spárovaním senzora pohybu s rozpoznávaním objektu, ktorý sa pred kŕmidlom nachádza. Takáto funkcionalita by ale s najväčšou pravdepodobnosťou využívala oveľa viac výpočtových prostriedkov, ako má vývojová doska Arduino k dispozícii.

Pre celkové uvedenie kŕmidla do hromadného používania je potrebné použiť drahšie a kvalitnejšie elektronické súčiastky na zvýšenie ich spoľahlivosti pri používaní.

# Kontakty

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Slavomír štucka Vedúci tímu, programátor | |  |  | |
|  | |  |  | |  |  |
|  | |  | slavomir.stucka  @student.tuke.sk | |  |  |
| Martin Tarasovič programátor, konštrukcia, dokumentácia |  | ivana hojdová Dokumentácia | |  | oleksandr senych schéma zapojenia,video - dokumentácia | |
|  |  | Related image | |  |  | |
| martin.tarasovic  @student.tuke.sk | |  | ivana.hojdova  @student.tuke.sk | |  | oleksandr.senych  @student.tuke.sk |