Stredná odborná škola Stará Turá

Športová 675, 916 01, Stará Turá

Študijný odbor: Mechanik počítačových sietí – 2682 K

Arduinom riadené autíčko

Stredoškolská odborná činnosť

Odbor : Elektrotechnika, hardware, mechatronika

Č. odboru : 12

riešiteľ: Jozef Matula

Mesto : Stará Turá

Šk. rok: 2021/2022

Trieda: štvrtá NP

Stredná odborná škola Stará Turá

Športová 675, 916 01, Stará Turá

Študijný odbor: Mechanik počítačových sietí – 2682 K

Arduinom riadené autíčko

Stredoškolská odborná činnosť

Odbor : Elektrotechnika, hardware, mechatronika

Č. odboru : 12

riešiteľ: Jozef Matula

školiteľ : Ing. Ján Koštial

Mesto : Stará Turá

Šk. rok: 2021/2022

Trieda: štvrtá NP

Vyhlasujem, že prácu stredoškolskej odbornej činnosti na tému „Arduinom riadené autíčko“ som vypracoval samostatne, s použitím uvedených zdrojov. Prácu som neprihlásil a ani neprezentoval v žiadnej inej súťaži, ktorá je pod gestorstvom MŠMVVaŠ SR. Som si vedomý dôsledkov, ak uvedené údaje nie sú pravdivé.

.....

Jozef Matula

Obsah

Úvod

Pred dvomi rokmi som videl video s danou prácou na internete. Najviac ma zaujal pohyb toho autíčka, dokázalo sa pohybovať ako klasické autíčko vpred a vzad, ale dokázalo sa pohybovať aj do bokov, šikmo a točiť sa okolo vlastnej osy. Autíčko bolo riadené diaľkovým ovládačom zostrojeného z arduina. Pomyslel som si že by to bolo zaujímavé si niečo také vytvoriť doma. Cieľom je aby sa autíčko funkčne pohybovalo do viacerých smerov pomocou diaľkového ovládania zostrojeného z arduina. Elektrické schémy boli navrhnuté v Eagle 7.7.0. Programy boli napísané v Arduino IDE 1.8.19. Kolieska a kryt na diaľkové ovládanie boli navrhnuté v Blender 3.0.0 a upravené pre tlačenie v Ultimate Cura 4.10.0.

1 Arduino

Arduino je open-source platforma, založená pôvodne na mikrokontroléri ATMega od firmy Atmel a grafickom vývojovom prostredí, ktoré vychádza z prostredia Wiring (podobný projekt ako Arduino, teda doska s mikrokontrolérom a IDE) a Processing (prostredie pre výuku programovania). Arduino môže byť použité k vytváraniu samostatných interaktívnych zapojení alebo môže byť pripojené k softvéri na počítači.[1]

1.1 Software

Vývojové prostredie (IDE) Arduina je viacplatformová aplikácia, naprogramovaná v Jave. Je navrhnuté tak, aby umožnilo programovať aj ľuďom, ktorí nemajú veľké skúsenosti s programovaním. Obsahuje editor kódu s bežnými vlastnosťami ako farebné označovanie syntaxe, automatické zarovnávanie a párovanie zátvoriek. Je schopné program skompilovať a nahrať do Arduina jedným kliknutím tlačidla. Program pre Arduino sa pomenúva anglickým slovom *sketch*. Programy pre Arduino sa píšu v programovacom jazyku C a C++ (Niekedy je nesprávne označovaný ako Wiring). IDE obsahuje knižnicu funkcií, ktoré uľahčujú písanie najzákladnejších operácii s hardvérom. Užívateľ musí definovať iba dve funkcie, aby sa získal spustiteľný program:

setup( ): funkcia, ktorá sa spúšťa iba raz na začiatku programu a používa sa na nastavenie parametrov
loop( ): funkcia, ktorá je periodicky spúšťaná, pokiaľ je mikrokontrolér pripojený ku zdroju elektrickej energie[2]

2 Ovládač

Autor videa ktorým som sa nechal inšpirovať mal video o tvorbe toho ovládača. Problémom bola malá informovanosť. Väčšinou vysvetľoval veci obsahovo, nie podrobne. Pozrel som si jeho webovú stánku pre viac informácií bohužiaľ, tam bolo len

[1] https://sk.wikipedia.org/wiki/Arduino prvé 3 riadky, [citované : 9.10.2021]

[2] https://sk.wikipedia.org/wiki/Arduino#Software [citované : 9.10.2021]

napísané to čo hovoril vo videu. Zostávalo mi zanalyzovať to video a snažiť sa z neho niečo vydedukovať. A po dňoch sa aj podarilo.

2.1 Elektronika

Ovládač je zložený z viacerých modulov, pre mňa sú tie najdôležitejšie: NRF24L01 PA/LNA 2.4GHz modul s anténou, adaptér pre modul NRF24L01, modul MPU6050 Gyroskop / akcelerometer a dva Joysticky. Namiesto arduina typu pro mini som použil Arduino UNO rev3. Už som ho totiž mal a je možné ho použiť v tomto projekte.

2.1.1 NRF24L01 PA/LNA 2.4GHz modul

Modul NRF24L01 je najnovší v RF moduloch od SparkFun. Tento modul využíva 2,4 GHz transceiver od Nordic Semiconductor. Táto doska obsahuje reverzne polarizovaný konektor SMA pre maximálny RF dosah. A na doske sú obvody PA a LNA, s externou anténou sa dostane na veľkú vzdialenosť ako bez týchto častí. Tento modul je dodávaný s 2,4G anténou (2DB), s prenosovou rýchlosťou 250 Kbps na otvorenom priestranstve môže dosiahnuť komunikačnú vzdialenosť 800 m až 1 km.[3]

Obrázok 1: NRF24L01 PA/LNA 2.4GHz modul

2.1.2 MPU6050 Gyroskop / akcelerometer

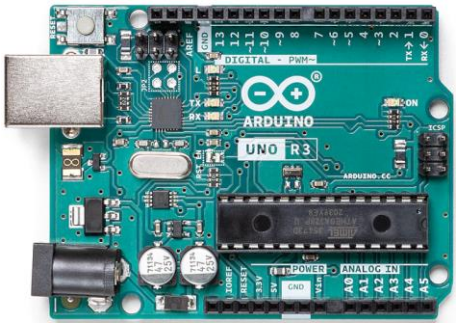
MPU6050 má zabudovaný 3-osový gyroskop, 3-osový akcelerometer. Tento modul je kompatibilný s úrovňou napätia 3,3V-5V.

[3] https://wiki.iteadstudio.com/NRF24L01_with_PA_and_LNA_Module [preložené, upravené : 11.10.2021]

Obrázok 2 : modul MPU6050

2.1.3 Arduino Uno rev3

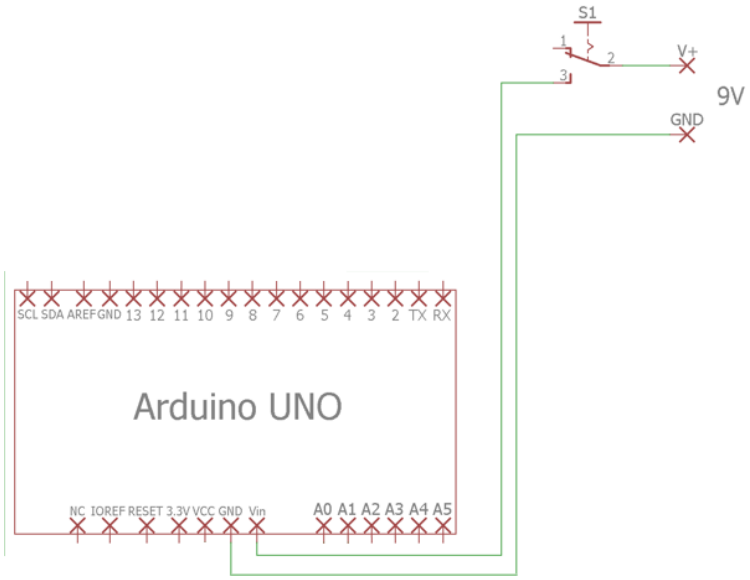
Arduino UNO je open-source mikrokontroléra doska založená na mikrokontroléri Microchip ATmega328P a vyvinutá spoločnosťou Arduino.cc. Doska je vybavená sadami digitálnych a analógových vstupno/výstupných (I/O) pinov, ktoré môžu byť prepojené s rôznymi rozširujúcimi doskami a inými obvodmi. Doska má 14 digitálnych I/O pinov (šesť dokáže PWM výstup), 6 analógových I/O pinov a je programovateľná pomocou Arduino IDE (Integrated Development Environment) cez USB kábel typu B. Môže byť napájaný káblom USB alebo externou 9-voltovou batériou, akceptuje napätie medzi 7 a 20 voltami.[4]



Obrázok 3 : Arduino UNO rev 3

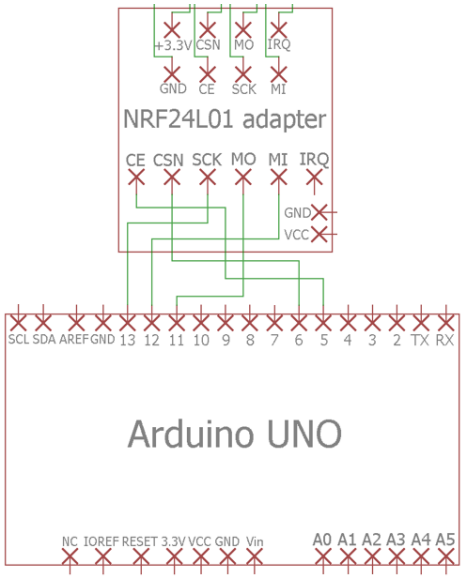
Ovládač je napájaný deväť voltovov batériou cez prepínač **S1** do **Vin** pinu ktorý si napätie upraví na päť volt. Zvyšok obvodu je napájaný piatimi voltami z arduina.

[4] https://en.wikipedia.org/wiki/Arduino_Uno prvých 4 a pól riadka, [preložené : 11.10.2021]



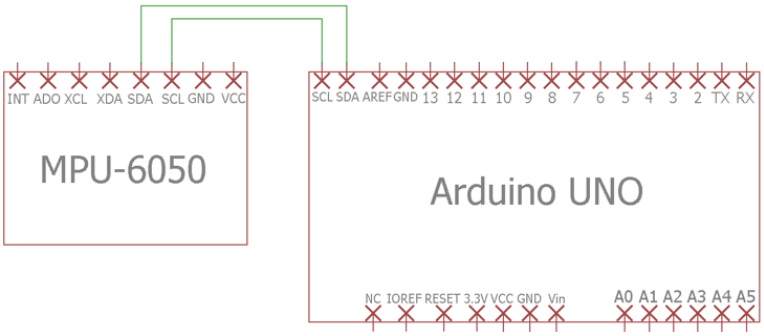
Obrázok 4 : Napájanie ovládača (vlastný), 11.10.2021

Rádiový modul NRF24L01 komunikuje s arduinom vďaka knižnici **SPI** cez nastavené **CE** a **CSN** piny a piny **13**, **12**, **11** čo sú piny **SCK**, **MISO** a **MOSI** . Tie sú pripojené z arduina do NRF24L01 adaptéra a z adaptéra do modulu. Modul NRF24L01 je prepojený cez adaptér s arduinom, pretože arduino má operačné napätie päť volt. Adaptér upraví týchto päť volt na 3,3 volt pre modul. Arduino má **3.3V** pin ale na modul NRF24L01 sa ťažšie spájkujú vodiče než na jeho adaptér.



Obrázok 5 : Zapojenie NRF24L01 adaptéra (vlastný), 14.10.2021

Modul **MPU6050** komunikuje s arduinom pomocou protokolu I2C cez piny **SCL** a **SDA**.



Obrázok 6 : Zapojenie MPU-6050 modulu (vlastný), 14.10.2021

Celá schéma sa nachádza v prílohe A , obrázok 1.

2.2 Program

Z videa sa o programe dalo zistiť dosť ale nie všetko. Program k ovládaču je verejne dostupný na web stránke[5] autora tak som si ho prečítal. Použil som z neho to čo som potreboval. Ďalším problémom boli chyby v programe. Po ich oprave bol hotový.

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Wire.h>

#define pl 7

const int MPU = 0x68;

float AccX, AccY, AccZ;
float GyroX, GyroY, GyroZ;
float AccUholX, AccUholY, GyroUholX, GyroUholY;
float uholX, uholY;
float AccErrorX, AccErrorY, GyroErrorX, GyroErrorY;
float Prejdcas, Teraz, Predcas;
int c = 0;

RF24 radio(5, 6);

const byte address[6] = "00001";

struct Data_Package
{
  byte j1PotX;
  byte j1PotY;
  byte j2PotX;
  byte pot1;
};

Data_Package data;
```

Obrázok 7 : 1.časť programu ovládača (vlastný), 14.10.2021

[5] https://howtomechatronics.com/projects/diy-arduino-rc-transmitter/ [použité : 14.10.2021]

Prvé štyri riadky v programe určujú knižnice ktoré bude program používať. Knižnicu **RF24.h** som musel stiahnuť[6], pretože sa nenachádza medzi knižnicami. V šiestom riadku je zadefinovaná hodnota **p1 (**prepínač) pre **digitálny pin 7**.V ôsmom riadku je zadefinovaná hodnota **MPU** ako 0x68 čo je I2C adresa modulu **MPU6050**. Od riadku 10 až po riadok 16 sú zadefinované hodnoty ktoré sa použijú pri prevedení IMU hodnôt modulu **MPU6050** na stupne. V riadku 18 sú zadefinované **digitálne piny 5** a **6** ako **komunikačné piny CE a CSN** pre modul **NRF24L01.** V riadku 20 je zadefinovaná **adresa** vďaka ktorej bude modul **NRF24L01** komunikovať z autíčkom. V riadku 22 je zadefinovaná **štruktúra Data_Package** v ktorej sú 4 Bajtové hodnoty. Tieto hodnoty budú preposlané autíčku. Maximálna veľkosť tejto štruktúry môže byť **32 bajtov**, pretože to je maximálny počet bajtov ktoré môže **NRF24L01** modul preposlať naraz. Príkaz v riadku 30 vytvorí jednotku z predošlej štruktúry.

```
32 void setup()
33 {
34   Serial.begin(9600);
35
36   initialize_MPU6050();
37
38   //calculate_IMU_error();
39
40   radio.begin();
41   radio.openWritingPipe(address);
42   radio.setAutoAck(false);
43   radio.setDataRate(RF24_250KBPS);
44   radio.setPALevel(RF24_PA_LOW);
45
46   pinMode(p1, INPUT_PULLUP);
47
48   data.j1PotX = 127;
49   data.j1PotY = 127;
50   data.j2PotX = 127;
51   data.pot1 = 1;
52 }
53
54 void loop()
55 {
56   data.j1PotY = map(analogRead(A0), 0, 1023, 0, 255);
57   data.j1PotX = map(analogRead(A1), 0, 1023, 255, 0);
58   data.j2PotX = map(analogRead(A2), 0, 1023, 0, 255);
59   data.pot1 = map(analogRead(A3), 0, 1023, 255, 0);
```

Obrázok 8 : 2.časť programu ovládača (vlastný), 15.10.2021

Od riadku 32 je otvorená funkcia **setup** v ktorej nastavíme parametre. V riadku 34 je príkaz ktorý umožní sériovú komunikáciu. Príkaz v riadku 36 inicializuje rozhranie k **MPU6050**. V riadku 38 je ako komentár napísaný príkaz **calculate_IMU_error()**,

[6] https://github.com/nRF24/RF24 tvorca : 2bndy5 [použité : 14.10.2021]

ktorý sa použije keď je treba zistiť IMU_error hodnoty. Riadky od 40 až po riadok 44 definujú rádiovú komunikáciu. V riadku 46 je nastavený pin do, ktorého je zapojení prepínač **p1**, ako vstupný a aktivujú sa interné pullup rezistory arduina. Od riadku 48 a až po 51 sú nastavené počiatočné hodnoty potenciometrov. Od 0 do 255. Keď sa Joystick nachádza v kľudovom stave, inak povedané v strede, tak hodnota je nastavená na 127. **Pot1** nie je Joystick preto je počiatočná nastavená na 1. Ukončili sme funkciu **setup**. Od riadku 54 je otvorená funkcia **loop** v ktorej napísaný program sa stále opakuje. Od riadku 56 až po riadok 59 je príkaz **analogRead** ktorý umožni arduinu čítať analógové hodnoty z analógových vstupov a príkaz **map** konvertuje rozhranie analógových hodnôt od 0 do 1023 na rozhranie bajtových hodnôt od 0 do 255 a tie priradí do predošlej štruktúry.

```
61   if (digitalRead(p1) == 0)
62   {
63     read_IMU();
64   }
65
66   radio.write(&data, sizeof(Data_Package));
67 }
68
69 void initialize_MPU6050()
70 {
71   Wire.begin();
72   Wire.beginTransmission(MPU);
73   Wire.write(0x6B);
74   Wire.write(0x00);
75   Wire.endTransmission(true);
76
77   Wire.beginTransmission(MPU);
78   Wire.write(0x1C);
79   Wire.write(0x10);
80   Wire.endTransmission(true);
81
82   Wire.beginTransmission(MPU);
83   Wire.write(0x1B);
84   Wire.write(0x10);
85   Wire.endTransmission(true);
86 }
```

Obrázok 9 : 3. časť programu ovládača (vlastný), 17.10.2021

Od riadku 61 až po riadok 64 je príkaz **if** v ktorom je podmienka: Ak na digitálnom vstupe **p1** je hodnota rovná 0, tak spusti funkciu **read_IMU**. Ak bude tento prepínač v zopnutej polohe tak namiesto Joystickov bude autíčko ovládať modul **MPU6050**. Pomocou príkazu riadku 66 arduino prepošle autíčku predošlú štruktúru z ktorej je teraz jednotka. Ukončili sme funkciu **loop**. Od riadku 69 po riadok 86 je funkcia

initialize_MPU6050. V tejto funkcií sa inicializuje komunikácia z modulom **MPU6050** a konfigurujú sa jeho registre.

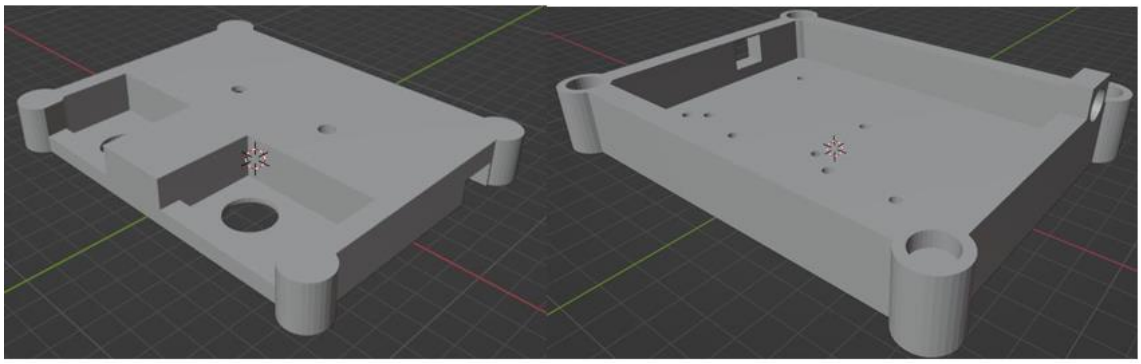
Od riadku 88 až po riadok 135 je funkcia **calculate_IMU_error** vďaka ktorej si arduino od modulu **MPU6050** vyžiada potrebné hodnoty, ktoré prepočíta na chybné. Tieto hodnoty následne vypíše na obrazovku a použijú pri kalibrácií ovládača. Celá funkcia **calculate_IMU_error** sa nachádza v prílohách A, obrázok 2 a B, obrázok 3 .

Od riadku 137 až po riadok 174 je funkcia **read_IMU** ktorá vezme hodnoty od **MPU6050** modulu premení ich na stupne a tie zmení na bajtové hodnoty. Tieto hodnoty dosadí namiesto hodnôt Joysticku 1. V riadkoch 148;149;163;164 sú vo výpočtoch hodnoty -1,46;+4,72;-0,49;-0,44 sú to záporné hodnoty IMU_error hodnôt. Slúžia na kalibráciu **MPU6050** modulu. Celá funkcia **read_IMU** sa nachádza v prílohách B, obrázok 4 a C, obrázok 5 .

V jednoduchosti, arduino príjme analógové a digitálne hodnoty z Joystickov, potenciometru (alebo prepnutím prepínača z MPU6050 modulu), konvertuje ich na hodnoty jedného bajtu, dosadí ich do štruktúry a tú pošle ako jednu jednotku autíčku.

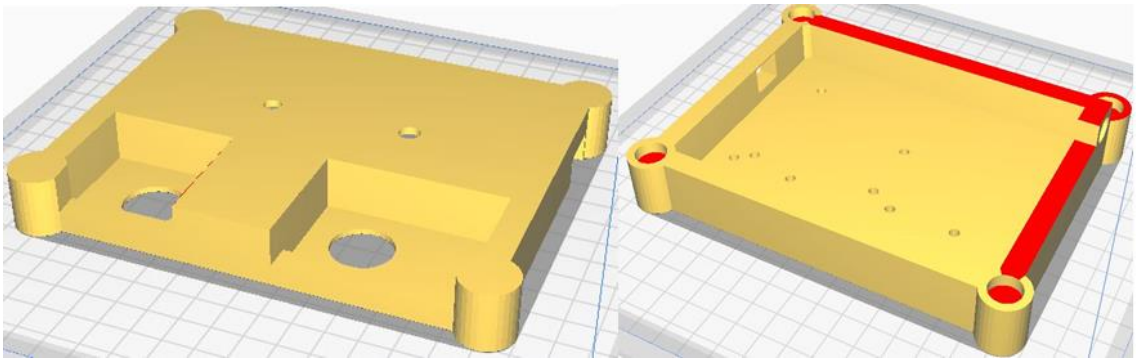
3 Dizajn a výroba

Autor vo videu mal pre svoj ovládač priehľadnú skrinku. Ja som si svoju navrhol v Blenderi a rozdelil som ju na 2 časti vrch a spodok.



Obrázok 10 : vrch a spodok ovládača (vlastný), 18.10.2021

Upravil som ju v Ultimate Cure a vtlačil som ju.



Obrázok 11 : vrch a spodok ovládača v cure (vlastný), 18.10.2021

4 Autíčko

Autor si vo videu navrhol plošný spoj. Ja som to radšej prepojil vodičmi pretože, ak sa v tom obvode niečo pokazí, bude sa to dať ľahšie vymeniť. Keby sa mu ten plošný spoj pokazil, napríklad by sa mu prepálila cestička, tak by ho musel znovu objednať. Mne stačí vymeniť vodič. Ako napájanie použil 12V batériu. Ja som použil držiak na dve 1,5V batérie a jeden konektor pre 9V. Prepojil som tieto dve časti do série a tým pádom keď tam teraz vložím batérie mi vznikol 12V zdroj pre autíčko.

4.1 Elektronika

Autíčko je tiež zložené z viacerých modulov, rádiový modul NRF24L01 2.4GHz, Arduino Mega rev 3, 4 riadiace jednotky DRV8825 pre 4 krokové motory NEMA 17 17HS8401.

4.1.1 Arduino Mega rev 3

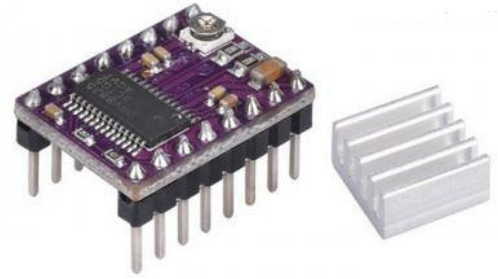
Arduino Mega 2560 je mikrokontrolérna doska založená na ATmega2560. Má 54 digitálnych vstupných / výstupných kolíkov (z ktorých 14 môže byť použitých ako výstupy PWM), 16 analógových vstupov, 4 UART (hardvérové sériové porty), kryštálový oscilátor 16 MHz, pripojenie USB, napájací konektor, hlavičku ICSP a tlačidlo resetu. Je napájaná 12V a operuje na 5V.[7]

[7] https://www.geeetech.com/wiki/index.php/Arduino_Mega_2560 prvých: 1,5, [preložené : 19.10.2021]

Obrázok 12 : Arduino mega rev 3

4.1.2 DRV8825

DRV8825 je doska vodiča krokového motora, ktorá má na DRV8825_chip, ktorá umožňuje ovládanie krokových motorov prostredníctvom firmvéru a softvéru programovania typu arduino. Doska DRV8825 môže dodávať až 2,5A vrchol alebo výstupný prúd 1,75A RMS (s riadnym zahrievaním pri 24 V a 25 °C).[8] Pri zapájaní bolo treba správne nastaviť referenčné napätie na trimery drivera.



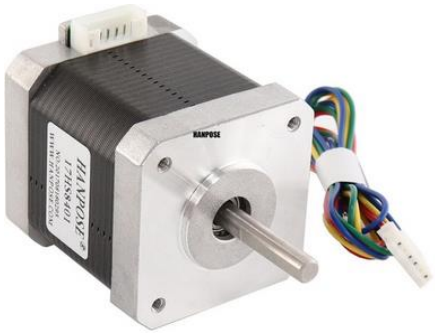
Obrázok 13: Riadiaca jednotka DRV8825 s chladičom

4.1.3 NEMA 17

Krokový motor NEMA 17 je krokový motor s čelnou doskou 1,7 x 1,7 palca (43,18 x 43,18 mm). NEMA 17 je väčší a vo všeobecnosti ťažší ako napríklad NEMA 14, ale to tiež znamená, že má viac priestoru na vyšší krútiaci moment. [9]

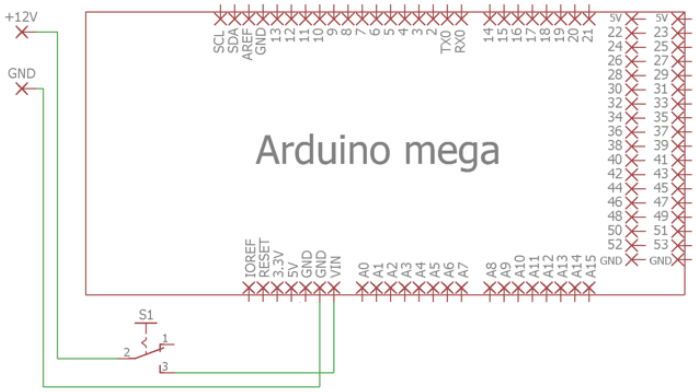
[8] https://reprap.com/wiki/DRV8825 prvé: 2, [preložené : 19.10.2021]

[9] https://reprap.com/wiki/NEMA_17_Stepper_motor prvých: 1,5, [preložené : 1.11.2021]



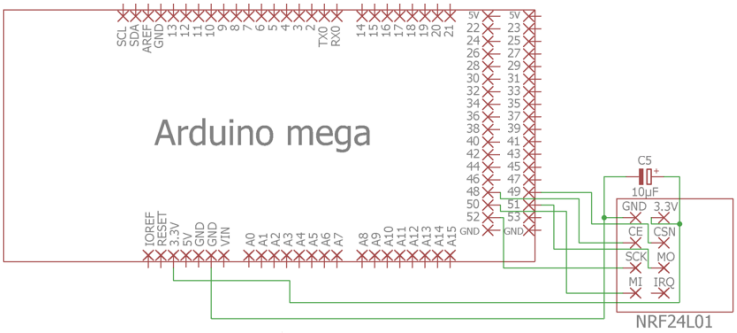
Obrázok 14 : Krokový motor NEMA 17

Arduino mega je napájané dvanástimi voltami cez prepínač **S1** do **Vin** pinu.



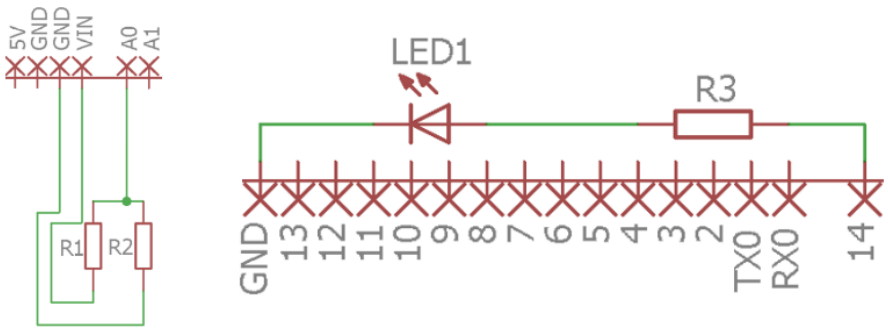
Obrázok 15 : Napájanie autíčka (vlastný), 1.11.2021

Modul NRF24L01 v tejto schéme je napájaný 3,3 voltami z pinu **3,3V** na arduine, tiež komunikuje cez **SPI** knižnicu.



Obrázok 16 : Zapojenie NRF24L01 modulu (vlastné), 2.11.2021

Delič napätia z rezistorov **R1** a **R2** a ledka s rezistorom **R3** slúžia na určenie stavu batérie.



Obrázok 17 : Určenia stavu batérie (vlastné), 2.11.2021

Dvanástimi voltami sú napájané aj krokové motory. Sú napájané cez **DRV8825** moduly. Celá schéma sa nachádza v prílohe C obrázok 6.

4.2 Program

```
1  #include <SPI.h>
2  #include <nRF24L01.h>
3  #include <RF24.h>
4  #include <AccelStepper.h>
5
6  RF24 radio(48,49);
7
8  const byte address[6] = "00001";
9  unsigned long CasPoslPrij = 0;
10 unsigned long CasTeraz = 0;
11
12 AccelStepper PravePredne(1, 32, 33);
13 AccelStepper PraveZadne(1, 34, 35);
14 AccelStepper LaveZadne(1, 36, 37);
15 AccelStepper LavePredne(1, 38, 39);
16
17 int rychlost = 1500;
18
19 struct Data_Package
20 {
21   byte j1PotX;
22   byte j1PotY;
23   byte j2PotX;
24   byte pot1;
25 };
26
27 Data_Package data;
```

Obrázok 18 : 1. časť programu autíčka (vlastný), 4.11.2021

Prvé štyri riadky v programe určujú knižnice ktoré bude program používať. V riadku 6 sú zadefinované **digitálne piny 48** a **49** ako **komunikačné piny CE a CSN** pre modul **NRF24L01.** V riadku 8 je zadefinovaná **adresa** vďaka ktorej bude modul **NRF24L01** komunikovať z ovládačom. V riadkoch 9 a 10 sú zadefinované hodnoty ktoré sa použijú

pri zistení komunikácie. Od riadku 12 až po 15 sú zadefinované piny ktoré budú používať drivery **DRV8825**.V riadku 17 je zadefinovaná hodnota pre jednotku **rychlost** ktorá určí rýchlosť akou autíčko pôjde. V riadku 19 je zadefinovaná rovnaká **štruktúra Data_Package** ktorá sa nachádza aj v programe ovládača. V riadku 27 je ten istý príkaz čo je v ovládači ktorý vytvorí rovnakú jednotku z predošlej štruktúry.

```
29 void setup()
30 {
31   LavePredne.setMaxSpeed(3000);
32   LaveZadne.setMaxSpeed(3000);
33   PraveZadne.setMaxSpeed(3000);
34   PravePredne.setMaxSpeed(3000);
35
36   radio.begin();
37   radio.openReadingPipe(0, address);
38   radio.setAutoAck(false);
39   radio.setDataRate (RF24_250KBPS);
40   radio.setPALevel (RF24_PA_LOW);
41   radio.startListening(); //Nastavuje túto jednotku ako prijímač
42
43   pinMode(14, OUTPUT);
44
45   Serial.begin(115200);
46 }
47
48 void loop()
49 {
50   CasTeraz = millis();
51   if (CasTeraz - CasPoslPrij > 1000 )
52   {
53     resetData();
54   }
```

Obrázok 19 : 2. časť programu autíčka (vlastný), 5.11.2021

Od riadku 29 je otvorená funkcia **setup** v ktorej nastavím maximálnu rýchlosť motorov. Riadky od 36 až po 40 definujú rádiovú komunikáciu a riadok 41 určuje že modul v autíčku je prijímač. V riadku 43 je zadefinovaný digitálny pin 14 ako výstup. V riadku 45 je zadefinovaná sériová komunikácia. Od riadku 48 je otvorená funkcia **loop** kde je na začiatku určená podmienka že pokiaľ čas posledných prijatých hodnôt je väčší ako sekunda (ak autíčko je mimo dosah) tak sa resetujú hodnoty. Autíčko zastaví.

```
56   if (radio.available())
57   {
58     radio.read(&data, sizeof(Data_Package));
59     CasPoslPrij = millis();
60   }
61
62   rychlost = map(data.potl, 0, 255, 100, 3000);
```

Obrázok 20 : 3. časť programu autíčka (vlastný), 7.11.2021

Od riadku 56 je určená podmienka že pokiaľ je dostupná rádiová komunikácia tak číta dáta z ovládača a počíta čas posledného prijatia. V riadku 62 je príkaz ktorý konvertuje bajtové hodnoty potenciometra **pot1** na rozhranie rýchlosti motorov. Od riadku 64 po riadok 117 je sekcia v ktorej sa nachádzajú príkazy **if else if** vďaka ktorým sa určí v ktorý smer sa motory budú točiť. Celá sekcia je v prílohe D .

```
119   LavePredne.runSpeed();
120   LaveZadne.runSpeed();
121   PraveZadne.runSpeed();
122   PravePredne.runSpeed();
```

Obrázok 21 : 4. časť programu autíčka (vlastný), 10.11.2021

Od riadku 119 po riadok 122 sú príkazy pre vykonanie jednotlivých krokov.

```
124   int sensorValue = analogRead(A0);
125   float voltage = sensorValue * (5.0 / 1023.00) * 3;
126
127   if (voltage < 11)
128   {
129     digitalWrite(14, HIGH);
130   }
131
132   else
133   {
134     digitalWrite(14, LOW);
135   }
136 }
```

Obrázok 22 : Kontrola stavu batérie (vlastný), 15.11.2021

Časť programu od riadku 124 po riadok 135 prijíma analógové hodnoty z pinu **A0** prepočíta ho na aktuálne napätie a vďaka časti **if else** vlastne kontroluje stav batérie a podľa neho sa rozsvieti ledka. Od riadku 139 sú funkcie ku ktorým sú priradené jednotlivé rýchlosti pre jednotlivé motory (Prílohy E a F). V riadku 227 je funkcia **reset_Data** ktorá zmení hodnoty prijaté z ovládača na základné

```
226 void resetData()
227 {
228   data.j1PotX = 129;
229   data.j1PotY = 129;
230   data.j2PotX = 129;
231   data.pot1 = 1;
232 }
```

Obrázok 23 : Resetovanie dát (vlastný), 20.11.2021

V jednoduchosti, arduino príjme štvor bajtovú jednotku z ovládača, rozbalí ju, zoberie si z nej jednotlivé bajtové hodnoty. Podľa nich určí ako sa má autíčko pohybovať a vykoná pohyb v danom smere.

4.3 Dizajn a výroba

Skrinku autíčka som vyrezal z centimeter hrubej drevenej dosky. Spodná a horná doska majú rozmery 30 x 15cm, predná a zadná majú 13 x 5,5cm a bočné dosky majú rozmery 30 x 5,5cm.

Obrázok 24 : Časti skrinky autíčka (vlastný), 10.12.2021

Na bočných doskách sú vyvŕtané otvory. Menšie majú priemer 3mm a slúžia na uchytenie motorov o dosky. Tie väčšie majú priemer 22mm a sú pre osky motorov. Jednotlivé dosky okrem hornej som prilepil lepidlom na drevo. Pre hornú dosku som do dvoch protiľahlých rohov stien a hornej dosky vyvŕtal diery. Do dier v stenách som prilepil dve závitové tyče. Hornú dosku som nasunul na dané závity a pripevnil matkami. Do prednej dosky som vyvŕtal dieru o priemery 3mm pre ledku a otvor pre vypínač. Skrinka je zložená pomocou pár skrutiek a lepidla na drevo.

Toto autíčko má schopnosť sa pohybovať do viacerých smerov vďaka individuálnemu pohybu motorov ale aj vďaka kolieskam. Tieto kolieska sa nazývajú mecanum alebo aj všesmerové kolieska.

Obrázok 25 : všesmerové koliesko

4.3.1 Mecanum koliesko

Základom kolesa Mecanum je koleso so sériou pogumovaných vonkajších valčekov, ktoré sú šikmo pripevnené po celom obvode jeho ráfika. Každý z týchto valčekov má zvyčajne os rotácie 45° k rovine kolesa a 45° k línii nápravy. Každé koleso Mecanum je nezávisle riaditeľné hnacie koleso s vlastným hnacím ústrojenstvom, ktoré pri otáčaní generuje hnaciu silu kolmú na os valca, ktorá sa dá vektorovať na pozdĺžnu a priečnu zložku vo vzťahu k vozidlu.

Typickým dizajnom Mecanum je štvorkolesová konfigurácia, so striedaním s ľavo- a pravotočivých valčekov, ktorých osi v hornej časti kolesa sú rovnobežné s uhlopriečkou rámu vozidla (a teda kolmé na uhlopriečku, keď sa spodná časť kolesa dotýka zeme). Takýmto spôsobom bude každé koleso generovať ťah približne rovnobežný s príslušnou uhlopriečkou rámu. Zmenou rýchlosti a smeru otáčania každého kolesa vytvorí súčet vektorov sily každého z kolies lineárne pohyby a/alebo rotácie vozidla, čo mu umožní manévrovať s minimálnou potrebou miesta.

Napríklad:

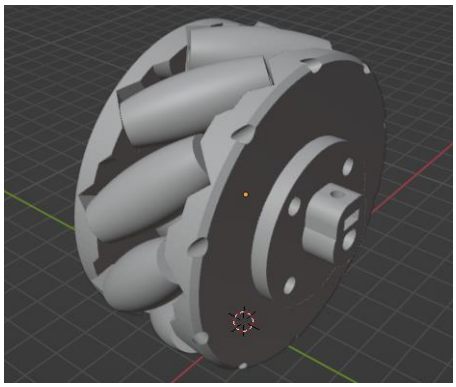
Pohyb všetkých štyroch kolies v rovnakom smere pri rovnakej rýchlosti bude mať za následok pohyb dopredu/dozadu, pretože vektory pozdĺžnej sily sa sčítavajú, ale priečne vektory sa navzájom rušia;

Pohyb (všetky pri rovnakej rýchlosti) oboch kolies na jednej strane v jednom smere, zatiaľ čo na druhej strane v opačnom smere bude mať za následok stacionárne otáčanie vozidla, pretože priečne vektory sa vyrušia, ale pozdĺžne vektory sa spoja, aby vytvorili krútiaci moment. okolo stredovej vertikálnej osi vozidla;

Pohyb (všetky pri rovnakej rýchlosti) diagonálnych kolies v jednom smere, zatiaľ čo ostatné diagonálne v opačnom smere bude mať za následok pohyb do strán, pretože priečne vektory sa sčítavajú, ale pozdĺžne vektory sa rušia.[10]

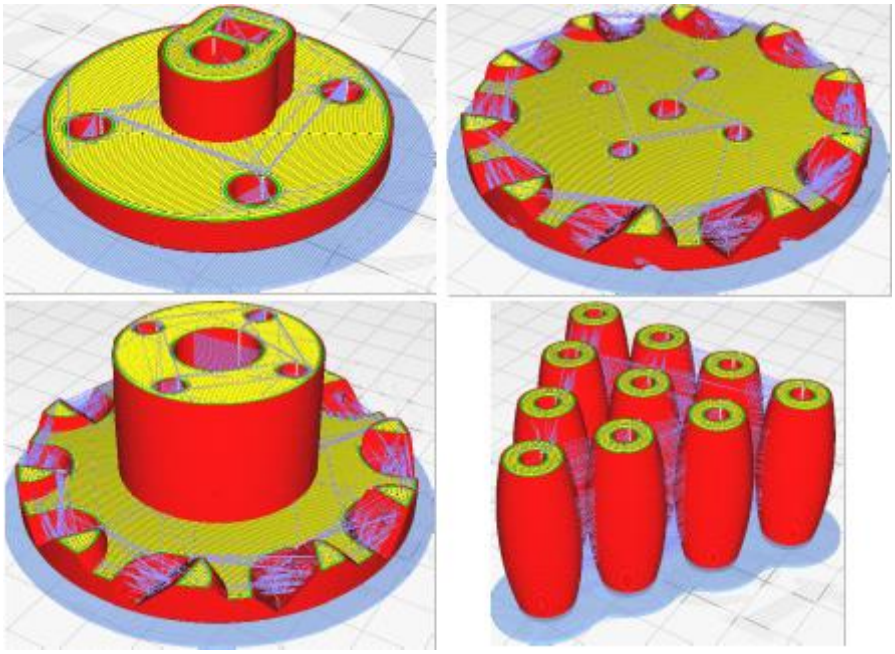
V Blenderi som sa pokúsil o navrhnutie vlastných koliesok podľa originálu.

[10] https://en.wikipedia.org/wiki/Mecanum_wheel#Design [preložené : 2.12.2021]

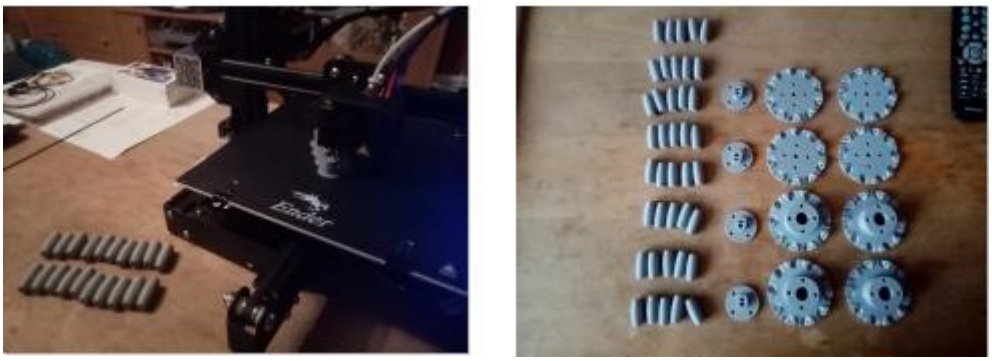


Obrázok 26 : model všesmerového kolieska (vlastný), 2.12.2021

Koliesko som rozdelil na 4 časti ktoré som upravil v slicery Ultimate Cura a vytlačil na 3D tlačiarni.



Obrázok 27 : Rozdelenie a úprava kolieska pre vytlačenie (vlastný). 4.12.2021



Obrázok 28 : Tlačenie častí kolieska (vlastný). 8.12.2021

Kolieska som zoskrutkoval pomocou 4mm skrutiek s dĺžkou 45mm a matiek.

Do ráfika som zasunul desať kusov 3mm zváračského drôtu, ktoré boli narezané na dĺžku 4cm. Na ne som nasunul desať valčekov a vonkajšie konce drôtov som prilepil tavnou pištoľou. V úchytke je malý otvor pre 3mm matku do ktorej som zaskrutkoval 3mm skrutku. Táto skrutka slúži pre uchytenie na osku z motora.

Obrázok 29 : autíčko (vlastný), 2.1.2022

Záver

Ovládač úspešne zbiera a mení analógové hodnoty z potenciometrov (z gyroskopu) na bajtové hodnoty. Úspešne ich preposiela autíčku a autíčko na základe nich úspešne vykonáva pohyby. Na moje prekvapenie to bola dosť zaujímavá ale aj dosť namáhavá a trochu komplikovaná práca. Dosť vecí som s vďaka nej naučil a som prekvapený sám zo seba že som bol schopný niečo takéto vytvoriť. Takýto návrh autíčka alebo celkovo podvozku má ale viacero využití. Najlepším príkladom je zahraničie kde sa takéto podvozky používajú v niektorých skladoch pre transport tovaru po sklade. Už ale aj existujú vysokozdvižné vozíky s takýmto podvozkom.

Zoznam použitej literatúry

Originálne video autíčka : https://youtu.be/83tVkgT89dM How To Mechatronics 28.5.2019

Originálne video ovládača : https://youtu.be/-BDCmwNssiw How To Mechatronics 13.1.2019

Arduino : https://sk.wikipedia.org/wiki/Arduino [citované : 9.10.2021]

Software : https://sk.wikipedia.org/wiki/Arduino#Software [citované : 9.10.2021]

NRF24L01 : https://wiki.iteadstudio.com/NRF24L01_with_PA_and_LNA_Module [preložené, upravené : 11.10.2021]

Obrázok 1 : https://www.google.com/search?q=NRF24L01+PA/LNA+2.4GHz+modul&client=firefox-b-d&sxsrf=AOaemvLh84EE0l094JPvmVfO3QmdsupNIg:1641374231925&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiry6XAo5r1AhWCCewKHW9ACS4Q_AUoAXoECAEQAw&biw=681&bih=283&dpr=1#imgrc=0tVX-RjOfVnhuM

Obrázok 2 : https://www.google.com/search?q=MPU6050&tbm=isch&ved=2ahUKEwj78ZKZpJr1AhU5gP0HHTFVB7YQ2-cCegQIABAA&oq=MPU6050&gs_lcp=CgNpbWcQAzIHCCMQ7wMQJzIFCAAQgAQyBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB5Q_hNY_hNg7hVoAHAAeACAAZICiAGEA5IBBTAuMS4xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=0mLVYfvAD7mA9u8PsaqdsAs&bih=283&biw=681&client=firefox-b-d#imgrc=tSF12nz8NxeKEM

Arduino UNO : https://en.wikipedia.org/wiki/Arduino_Uno prvých 4 a pól riadka, [preložené : 11.10.2021]

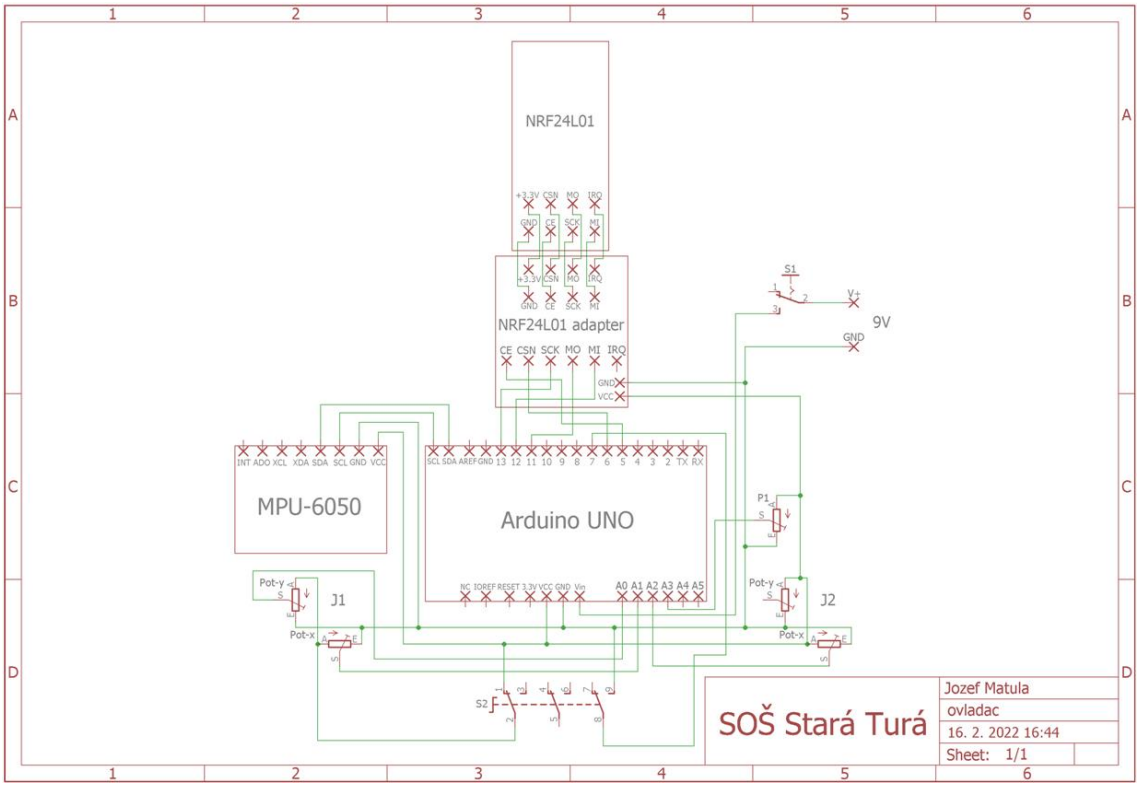
Obrázok 3 : https://www.google.com/search?client=firefox-b-d&sxsrf=APq-WBsNlcFPg8qMHKFTLdcJ_UBB8J47aQ:1645078859212&q=What+is+Arduino+Rev3?&tbm=isch&source=iu&ictx=1&vet=1&fir=PRVQ_oisht5imM%252Cdrxk3Yauoqw3YM%252C_&usg=AI4_-kThhLJkmlZZMU1WcXY15dfaV0apSQ&sa=X&ved=2ahUKEwj4356ojIb2AhWC7rsIHTGjBvMQ9QF6BAg3EAE&biw=1366&bih=653&dpr=1#imgrc=PRVQ_oisht5imM

Webová stránka autora – ovládač : https://howtomechatronics.com/projects/diy-arduino-rc-transmitter/ [použité : 14.10.2021]

Knižnica RF24.h : https://github.com/nRF24/RF24 tvorca : 2bndy5 [použité : 14.10.2021]
Arduino Mega : https://www.geeetech.com/wiki/index.php/Arduino_Mega_2560 prvých: 1,5, [preložené : 19.10.2021]
Obrázok 12 : https://www.google.com/search?q=arduino+mega+2560+rev+3&client=firefox-b-d&sxsrf=AOaemvIAanCca5lkiP6xzHbk3Yq8HnMXEQ:1641375567638&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiInJu9qJr1AhUHKuwKHfQ2ATgQ_AUoAXoECAIQAw&biw=681&bih=283&dpr=1#imgrc=FLg_pMyDZ4GBHM
DRV8825 : https://reprap.com/wiki/DRV8825 prvé: 2, [preložené : 19.10.2021]
Obrázok 13 : https://www.google.com/search?q=DRV8825&client=firefox-b-d&sxsrf=APq-WBvUXqMkuS7w9vdxTykT_KpwI1YTMQ:1645391062766&source=lnms&tbm=isch&sa=X&ved=2ahUKEwitsr-ul4_2AhUTh_0HHe03DDoQ_AUoAXoECAIQAw&biw=1366&bih=653&dpr=1#imgrc=M1tLZ1TuhKKpcM
NEMA 17 : https://reprap.com/wiki/NEMA_17_Stepper_motor prvých: 1,5, [preložené : 1.11.2021]
Obrázok 14 : https://www.google.com/search?q=nema+17+17hs8401&tbm=isch&ved=2ahUKEwjOveGDmY_2AhUI_qQKHRu9CaoQ2-cCegQIABAA&oq=nema+17+17hs8401&gs_lcp=CgNpbWcQAzIHCCMQ7wMQJzIHCCMQ7wMQJzIECAAQGFAAWOoTYKgYaAFwAHgAgAGCAogB-QuSAQUwLjUuM5gBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=lq4SYs4aiPyTBZv6ptAK&bih=653&biw=1366&client=firefox-b-d#imgrc=N6MgA0vGz3GSFM
Obrázok 24 : https://sc01.alicdn.com/kf/HTB1RoXQXkWE3KVjSZSyq6xocXXa8.jpg
Všesmerové koliesko : https://en.wikipedia.org/wiki/Mecanum_wheel#Design [preložené : 2.12.2021]

Prílohy

Príloha A



Obrázok 1 : Schéma ovládača (vlastný), 4.1.2021

```
void calculate_IMU_error()
{
  while (c < 200)
  {
    Wire.beginTransmission(MPU);
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU, 6, true);

    AccX = (Wire.read() << 8 | Wire.read()) / 4096.0 ;
    AccY = (Wire.read() << 8 | Wire.read()) / 4096.0 ;
    AccZ = (Wire.read() << 8 | Wire.read()) / 4096.0 ;

    AccErrorX = AccErrorX + ((atan((AccY) / sqrt(pow((AccX), 2) + pow((AccZ), 2))) * 180 / PI));
    AccErrorY = AccErrorY + ((atan(-1 * (AccX) / sqrt(pow((AccY), 2) + pow((AccZ), 2))) * 180 / PI));
    c++;
  }

  AccErrorX = AccErrorX / 200;
  AccErrorY = AccErrorY / 200;
  c = 0;
```

Obrázok 2 : 1. časť IMU error (vlastný), 4.1.2021

Príloha B

```
  while (c < 200)
  {
    Wire.beginTransmission(MPU);
    Wire.write(0x43);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU, 4, true);
    GyroX = Wire.read() << 8 | Wire.read();
    GyroY = Wire.read() << 8 | Wire.read();

    GyroErrorX = GyroErrorX + (GyroX / 32.8);
    GyroErrorY = GyroErrorY + (GyroY / 32.8);
    c++;
  }

  GyroErrorX = GyroErrorX / 200;
  GyroErrorY = GyroErrorY / 200;

  Serial.print("AccErrorX: ");
  Serial.println(AccErrorX);
  Serial.print("AccErrorY: ");
  Serial.println(AccErrorY);
  Serial.print("GyroErrorX: ");
  Serial.println(GyroErrorX);
  Serial.print("GyroErrorY: ");
  Serial.println(GyroErrorY);
}
```

Obrázok 3 : 2. časť IMU error (vlastný), 4.1.2021

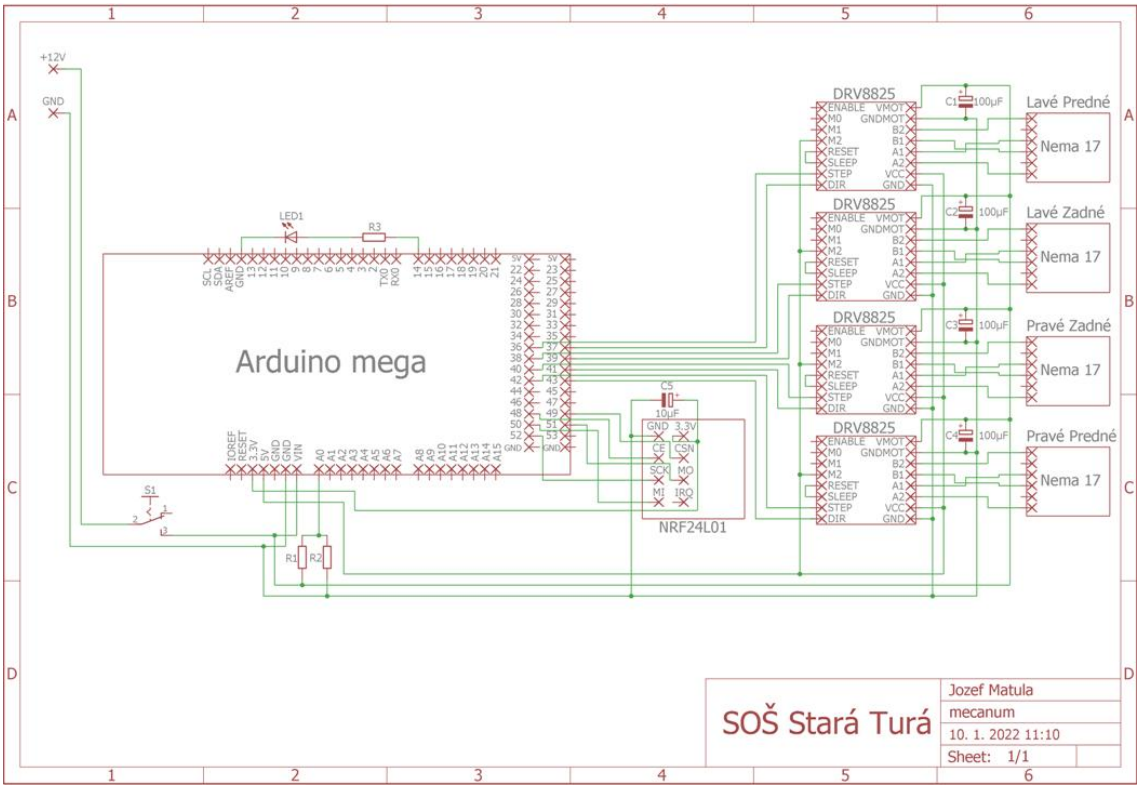
```
void read_IMU()
{
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 6, true);

  AccX = (Wire.read() << 8 | Wire.read()) / 4096.0;
  AccY = (Wire.read() << 8 | Wire.read()) / 4096.0;
  AccZ = (Wire.read() << 8 | Wire.read()) / 4096.0;

  AccUholX = (atan(AccY / sqrt(pow(AccX, 2) + pow(AccZ, 2))) * 180 / PI) - 1.46;
  AccUholY = (atan(-1 * AccX / sqrt(pow(AccY, 2) + pow(AccZ, 2))) * 180 / PI) + 4.72;

  Predcas = Teraz;
  Teraz = millis();
  Prejdcas = (Teraz - Predcas) / 1000;

  Wire.beginTransmission(MPU);
  Wire.write(0x43);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 4, true);

  GyroX = (Wire.read() << 8 | Wire.read()) / 32.8;
  GyroY = (Wire.read() << 8 | Wire.read()) / 32.8;

  GyroX = GyroX - 0.49;
  GyroY = GyroY - 0.44;

```

Obrázok 4 : 1. read IMU (vlastný), 4.1.2021

Príloha C

```
166   GyroUholX = GyroX * Prejdcas;
167   GyroUholY = GyroY * Prejdcas;
168
169   uholX = 0.98 * (uholX + GyroUholX) + 0.02 * AccUholX;
170   uholY = 0.98 * (uholY + GyroUholY) + 0.02 * AccUholY;
171
172   data.j1PotX = map(uholX, -90, +90, 0, 255);
173   data.j1PotY = map(uholY, -90, +90, 0, 255);
174 }
```

Obrázok 5 : 2. read IMU (vlastný), 4.1.2021



Obrázok 6 : Schéma autíčka (vlastný), 4.1.2021

Príloha D

```
64  if (data.j1PotY > 140 & data.j1PotX == 129 )
65  {
66    vpred();
67  }
68
69  else if (data.j1PotY < 121 & data.j1PotX == 129 )
70  {
71    vzad();
72  }
73
74  else if (data.j1PotX > 140 & data.j1PotY == 129 )
75  {
76    vpravo();
77  }
78
79  else if (data.j1PotX < 121 & data.j1PotY == 129 )
80  {
81    vlavo();
82  }
83
84  else if (data.j1PotX < 121 & data.j1PotY > 140)
85  {
86    vlavoVpred();
87  }
88
89  else if (data.j1PotX > 140 & data.j1PotY > 140)
90  {
91    vpravoVpred();
92  }
```

Obrázok 7 : 1. časť určenie smeru jazdy autíčka (vlastný), 4.1.2021

```
94  else if (data.j1PotX < 121 & data.j1PotY < 121 )
95  {
96    vlavoVzad();
97  }
98
99  else if (data.j1PotX > 140 & data.j1PotY < 121 )
100 {
101   vpravoVzad();
102 }
103
104 else if (data.j2PotX < 121)
105 {
106   otocVlavo();
107 }
108
109 else if (data.j2PotX > 140)
110 {
111   otocVpravo();
112 }
113
114 else
115 {
116   stoj();
117 }
```

Obrázok 8 : 2. časť určenie smeru jazdy autíčka (vlastný), 4.1.2021

Príloha E

```
140 void vpred()
141 {
142   LavePredne.setSpeed(-rychlost);
143   LaveZadne.setSpeed(-rychlost);
144   PraveZadne.setSpeed(rychlost);
145   PravePredne.setSpeed(rychlost);
146 }
147
148 void vzad()
149 {
150   LavePredne.setSpeed(rychlost);
151   LaveZadne.setSpeed(rychlost);
152   PraveZadne.setSpeed(-rychlost);
153   PravePredne.setSpeed(-rychlost),
154 }
155
156 void vpravo()
157 {
158   LavePredne.setSpeed(-rychlost);
159   LaveZadne.setSpeed(rychlost);
160   PraveZadne.setSpeed(rychlost);
161   PravePredne.setSpeed(-rychlost),
162 }
163
164 void vlavo()
165 {
166   LavePredne.setSpeed(rychlost);
167   LaveZadne.setSpeed(-rychlost);
168   PraveZadne.setSpeed(-rychlost);
169   PravePredne.setSpeed(rychlost);
```

Obrázok 9 : 1. časť určenie smeru otáčania kolies (vlastný), 4.1.2021

```
172 void vpravoVpred()
173 {
174   LavePredne.setSpeed(-rychlost);
175   LaveZadne.setSpeed(0);
176   PraveZadne.setSpeed(rychlost);
177   PravePredne.setSpeed(0);
178 }
179
180 void vpravoVzad()
181 {
182   LavePredne.setSpeed(0);
183   LaveZadne.setSpeed(rychlost);
184   PraveZadne.setSpeed(0);
185   PravePredne.setSpeed(-rychlost);
186 }
187
188 void vlavoVpred()
189 {
190   LavePredne.setSpeed(0);
191   LaveZadne.setSpeed(-rychlost);
192   PraveZadne.setSpeed(0);
193   PravePredne.setSpeed(rychlost);
194 }
195
196 void vlavoVzad()
197 {
198   LavePredne.setSpeed(rychlost);
199   LaveZadne.setSpeed(0);
200   PraveZadne.setSpeed(-rychlost);
201   PravePredne.setSpeed(0);
```

Obrázok 10 : 2. časť určenie smeru otáčania kolies (vlastný), 4.1.2021

Príloha F

```
204 void otocVlavo()
205 {
206   LavePredne.setSpeed(rychlost);
207   LaveZadne.setSpeed(rychlost);
208   PraveZadne.setSpeed(rychlost);
209   PravePredne.setSpeed(rychlost);
210 }
211
212 void otocVpravo()
213 {
214   LavePredne.setSpeed(-rychlost);
215   LaveZadne.setSpeed(-rychlost);
216   PraveZadne.setSpeed(-rychlost);
217   PravePredne.setSpeed(-rychlost);
218 }
219
220 void stoj()
221 {
222   LavePredne.setSpeed(0);
223   LaveZadne.setSpeed(0);
224   PraveZadne.setSpeed(0);
225   PravePredne.setSpeed(0);
226 }
```

Obrázok 11 : 3. časť určenie smeru otáčania kolies (vlastný), 4.1.2021