Středoškolská technika 2019

Setkání a prezentace prací středoškolských studentů na ČVUT

Autonomní vozítko

Markéta Aghová, Pavel Stenchlý, Vojtěch Tomala

Střední průmyslová škola, Karviná, příspěvková organizace

Žižkova 1818, Karviná - Hranice

Anotace

Práce se zabývá návrhem a konstrukcí Autonomního vozítka pro edukační účely na naší škole. Cílem práce je navrhnout a sestrojit Autonomní vozítko ovládané mobilní aplikací prostřednictvím WIFI. Práce je rozdělena na teoretickou a praktickou část. V teoretické části jsou vysvětleny základní pojmy, které jsou nezbytné pro návrh a výrobu vozítka. V praktické části je detailně zachycen postup návrhu, výroby, popsány funkce, program a využití ve vzdělávání ve školách.

Klíčová slova

3D tisk; Arduino; Autonomní vozítko; vzdělávací účely

Annotation

The work is focused on a design and construction of Autonomous vehicle for education purpose at our school. The aim of the work is to suggest and construct an Autonomous vehicle which is controlled by an app via WIFI. The work is divided into the theoretical and practical part. The theoretical part explains the basic concepts which we needed to know to design and build such a vehicle. The practical part describes the scheme, production process, function, programe and usage in education at school.

Keywords

3D printing; Arduino; Autonomous vehicle; education function

OBSAH

ÚVOD

V naší práci se zabýváme návrhem a sestavením autonomního vozítka, které se pohybuje autonomně po černé čáře a dokáže překonat i překážku nebo se pohybuje manuálně pomocí mobilní aplikace. Hlavním cílem je navrhnout, sestavit vozítko a dát studentům, či žákům základních škol možnost vyzkoušet si, seznámit se s některými programy např. Solid Edge a Aruidno IDE. Dále se naučí navrhovat a pracovat s 3D tiskem, programovat platformu Arduino a vytvářet mobilní aplikace. Nejdůležitější však je fantazie, komunikace a práce v týmu.

Práce je rozdělena do dvou částí, a to teoretické a praktické. Teoretická část se věnuje práci s mikropočítačem Arduino a programováním v jazyce Wiring, dále zde naleznete informace o 3D tisku, programu Solid Edge a programech pro tvorbu mobilních aplikací, což jsou Visual studio 2015 a online Mit app convertor.

V praktické části se zabýváme detailně návrhem, sestavením vozítka, programování platformy Arduino a tvorbou aplikací přímo týkajících se našeho projektu. Vše je podrobně popsáno a vysvětleno. V praktické části se nachází i naše představa o možném vzdělávání ve školách.

Námětem naší práce je soutěž Students for Automotive, kde jsme za úkol dostali sestavit vozítko z dodaných komponentů. To nás inspirovalo, abychom si vyrobili vlastní vozítko, které můžeme libovolně upravit. Proto jsme se rozhodli udělat nějaké úpravy, jako je například vlastní karoserie a toto vozítko si sami sestavit a chtěli bychom, aby i ostatní studenti dostali možnost naučit se novým věcem.

TEORETICKÁ ČÁST

1 ARDUINO

Arduino je elektrická vývojová platforma, která je snadno programovatelná, a proto je vhodná i pro začátečníky. Platforma Arduino je hodně rozšířená, tato platforma může vnímat okolí pomocí vstupních senzorů nebo jej ovlivňovat pomocí výstupních periferií. Je mnoho desek, klonů a shieldů pro rozšíření možností práce s Arduinem.

Oproti jiným kitům jsou levnější, mají jednoduché zapojení a existuje mnoho návodů. Ve světě a ČR je široká komunita zabývající se Arduinem. Základní části většiny desek je mikroprocesor ATMega328. To je výkonný 8bitový mikroprocesor firmy ATMEL, deska dále obsahuje krystal, napájení na 5 V a převodník pro sériovou komunikaci s počítačem.

Využití je velmi široké, protože se vyrábí spousta doplňujících senzorů a výstupních periferií. Může sloužit pro zábavu, k výuce programování, rozšíření svých znalostí a dovedností. Záleží, co daný uživatel potřebuje a jak to naprogramuje. Arduino může sloužit například jako bezpečnostní systém nebo pro detekci kouře či plynu CO. [1, 2]

1.1 O jazyce Wiring

Aby Arduino deska vykonávala to, co potřebujeme nebo chceme, musíme ji naprogramovat. Nejrozšířenějším programovacím jazykem je Wiring a programátorské prostředí Arduino IDE. Tento software je volně dostupný a spolu s ním i velké množství knihoven pro správnou funkci připojených periferií. Prvním autorem jazyka Wiring je Hernando Barragán, který jej představil ve své diplomové práci v roce 2003.

Wiring patří k vyšším programovacím jazykům, byl vytvořený pro snadné programování mikropočítačů bez větších znalostí hardware. Tento jazyk je vyvíjen v C a C++. Software je funkční na běžných operačních systémech a dnes už i na mobilních zařízení. Nejčastěji využívané programátorské prostředí je již zmíněné Arduino IDE. Jedná se o přehledné a jednoduché prostředí.

Po napsání zdrojového textu a před nahráním strojového kódu do mikropočítače dojde ke kompilaci a poté pomocí převodníku k nahrání do čipu. Tímto způsobem je kód zapsán do paměti EEPROM mikropočítače, program běží neustále v nekonečné smyčce a plní své naprogramované funkce. [3, 5]

1.2 3D Tisk

3D tisk je proces, při kterém se z digitální předlohy (3D modelu) vytváří fyzický model. Jedná se o proces aditivní, to znamená, že se materiál přidává. Na rozdíl od obráběcích strojů, kde se z celistvého bloku materiál odebírá, až zůstane jen požadovaný tvar.

Nejpoužívanější technologie, FDM (fusion deposition modeling), funguje velice jednoduše. Objekt vzniká vrstvu po vrstvě natavováním tenkého proužku plastového materiálu. Dalšími používanými technologiemi jsou například SLA (stereolytografie), SLS (selective laser sintering) a DMLS (direct metal laser sintering). [6]

1.3 Solid Edge

Jedná se o 3D CAD (Computer aided design) software určený pro návrh strojírenských konstrukcí. Je postaven na jádru Parasolid. [7]

1.4 Mit app convertor

Mit app convertor je program, ve kterém si můžete naprogramovat aplikaci na mobilní platformu Google Android.

Přes webový prohlížeč máte možnost vytvořit si uživatelské rozhraní své mobilní aplikace, kterou sestavíte z různých grafických komponent – tlačítek, nabídek, zatrhovátek a podobně. K nim posléze za pomoci javovské aplikace doplníte funkcionalitu, vše otestujete v emulátoru nebo přímo mobilním telefonu, a nakonec si stáhnete balíček s hotovou aplikací. Ten můžete instalovat a šířit dále. [16]

1.5 Visual studio 2015

Visual Studio je vývojové prostředí od společnosti Microsoft, které je standardem pro programování nových aplikací pro Windows.

Visual Studio má ve znaku symbol nekonečna a jeho možnosti se tomu podobají. Ve výchozím nastavení Visual Studio podporuje několik programovacích jazyků a platforem. A díky různým doplňkům je možné si do Visual Studia doinstalovat několik dalších. Pokud jsou tyto doplňky zadarmo a jsou kvalitní, Microsoft se s vývojáři snaží domluvit a mnohdy se stanou výchozí součástí Visual Studia. Stalo se tak například s perfektními nástroji pro vývoj v Pythonu nebo TypeScriptu, které se ve verzi 2012 distribuovaly jako klasické doplňky a ve verzi 2013 již jsou předinstalované nebo je lze velmi jednoduše na pár kliknutí doinstalovat. [17]

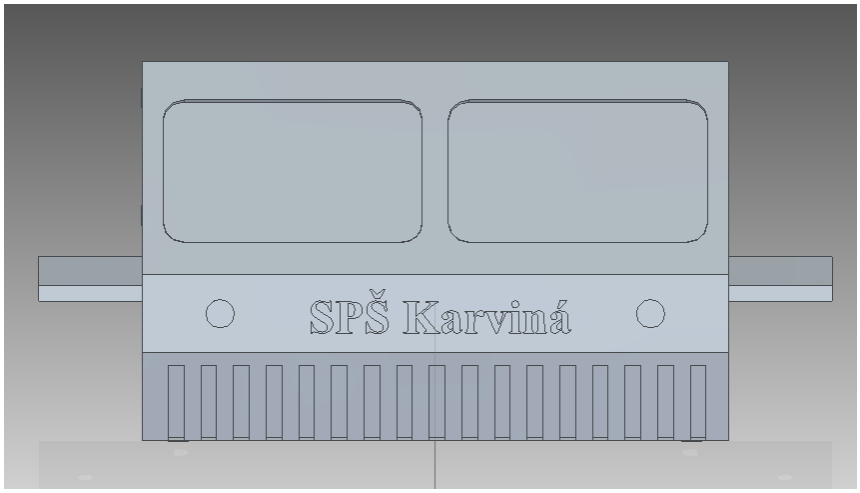
PRAKTICKÁ ČÁST

2 KONSTRUKCE

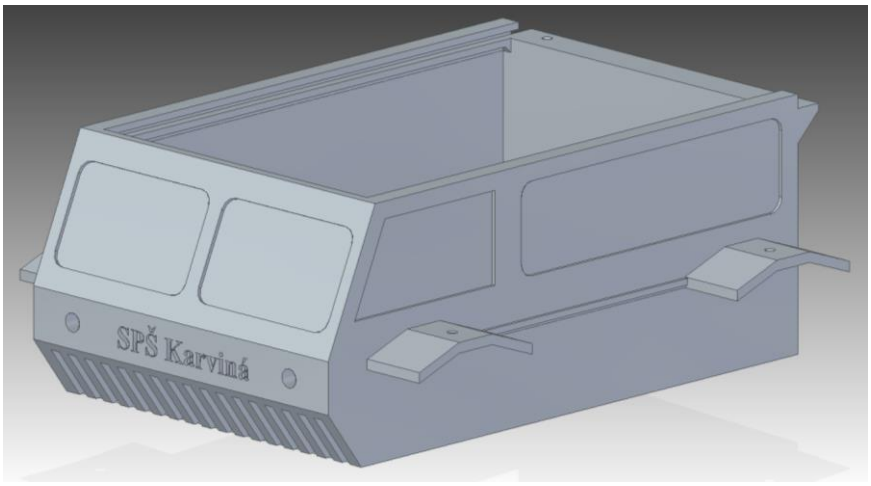
2.1 Návrh vozítka

Konstrukce vozítka je náš vlastní návrh, který je vytvořen v programu Solid Edge. Konstrukce vozítka je rozdělena na dvě hlavní části, a to podvozek a kabina. Dalšími částmi jsou střecha a kryt podvozku. Konstrukce je takto rozvržena proto, aby bylo možné tyto části vytisknout na 3D tiskárně a byla zajištěna případná manipulace v nitru vozítka.

V nitru vozítka se nachází i určité zarážky, či otvory pro uchycení komponentů. Tyto pomocné zarážky jsou umístěny podle rozložení komponentů a jejich rozměrů. Jednotlivé části jsou spojeny šrouby a distančními válečky do hromady, ale tak aby se daly kdykoli bez úsilí rozebrat. Střecha vozítka je zasouvací a zajištěna proti pohybu šrouby, na střeše je také název školy.



Obrázek 1 Kabina – čelní strana



Obrázek 2 Kabina

2.2 Použité komponenty

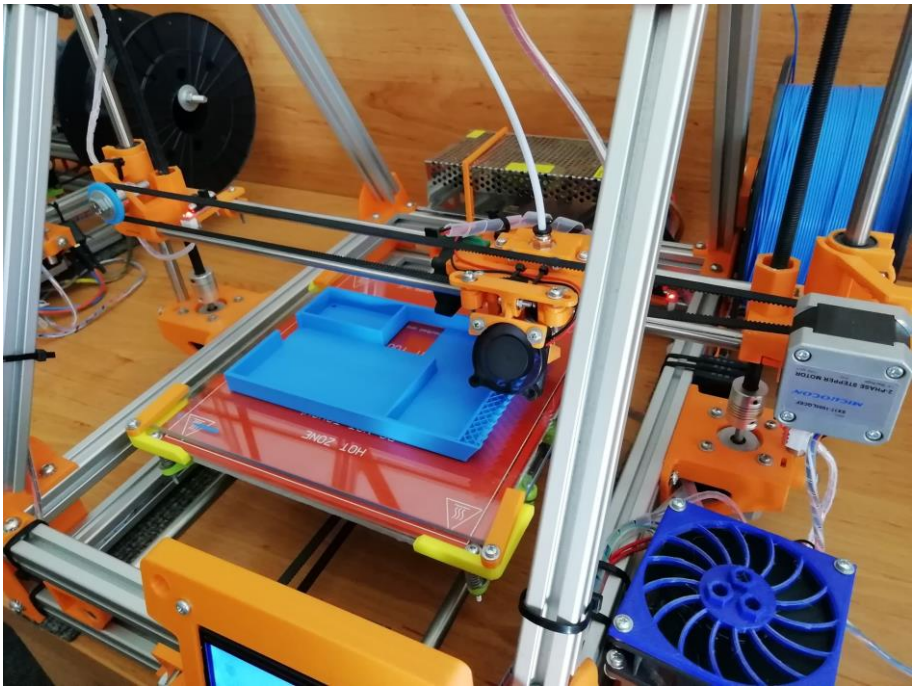
Základní části jsou již zmiňované podvozek a kabina, které tvoří vnější kostru. Základní komponenta pro ovládání vozítka je *Arduino Mega*, ve které je nahrán program pro řízení vozítka. Pohon je zajištěn čtyřmi DC motory, které jsou ovládány pomocí *Arduino motor shield,* který je propojen a řízen s *Aurduino Mega.* Motory a další části jsou napájeny monočlánky v držáku baterií.

Vozítko ovládáme mobilní aplikací, proto nezbytnou součástí je *ESP 8266,* což je prvek, který zajišťuje komunikaci mezi mobilní aplikací prostřednictvím WIFI a *Arduino Mega,* kde je komunikace zajištěna sériovým přenosem.

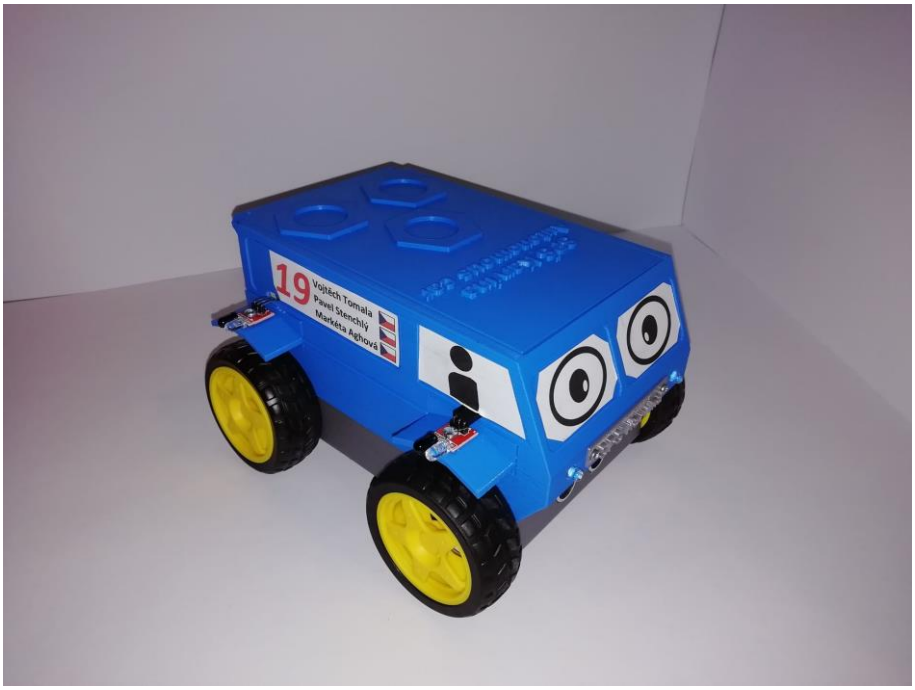
Pro autonomní pohyb vozítka je zapotřebí *IR sensor* pro sledování černé čáry, *Ultrasonic sensor* pro rozpoznání překážky a zamezení čelní kolizi. Pro překonání překážky je použit *Arduino senzor překážek,* jehož čidla jsou umístěna na stranách vozítka.

2.3 Sestavení

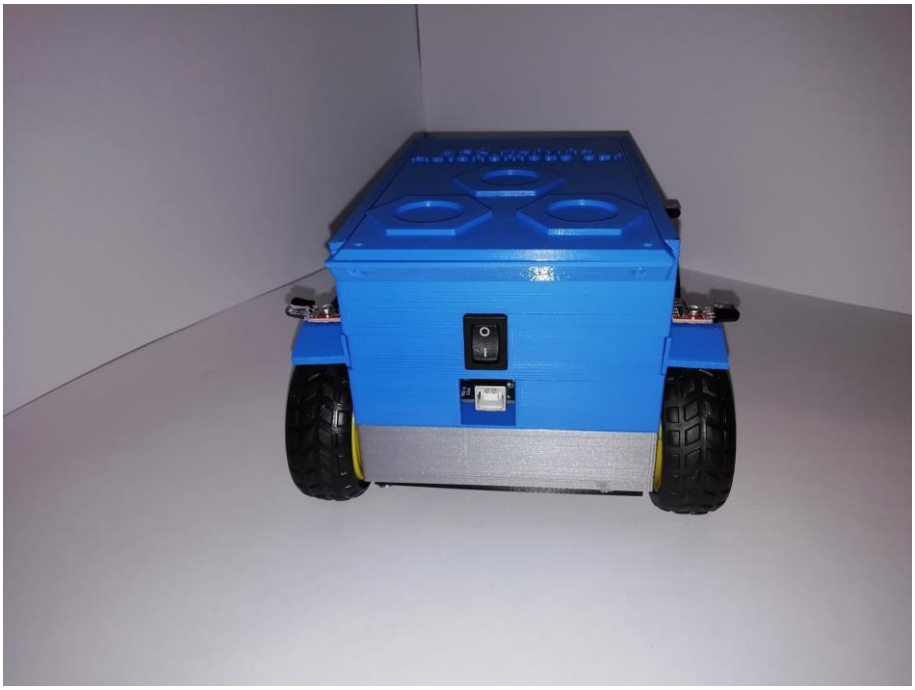
Sestavení vozítka bylo jednoduché, protože všechny části byly pečlivě a přesně navrženy. Prvním krokem bylo spojení podvozku a kabiny dohromady, tyto části jsme spojili šrouby. Poté jsme uchytili motory a další komponenty uvnitř vozítka jako jsou například *Arduino Mega* a *ESP8266.* Z vnější části jsou uchyceny čidla pro překonání překážek na bocích vozítka a v přední části v úrovni podvozku je *Ultrasonic sensor* pro detekci překážky a zabránění čelní kolizi. Dále jsme jen připevnili kryt podvozku a střechy.



Obrázek 3 Tisk komponenty



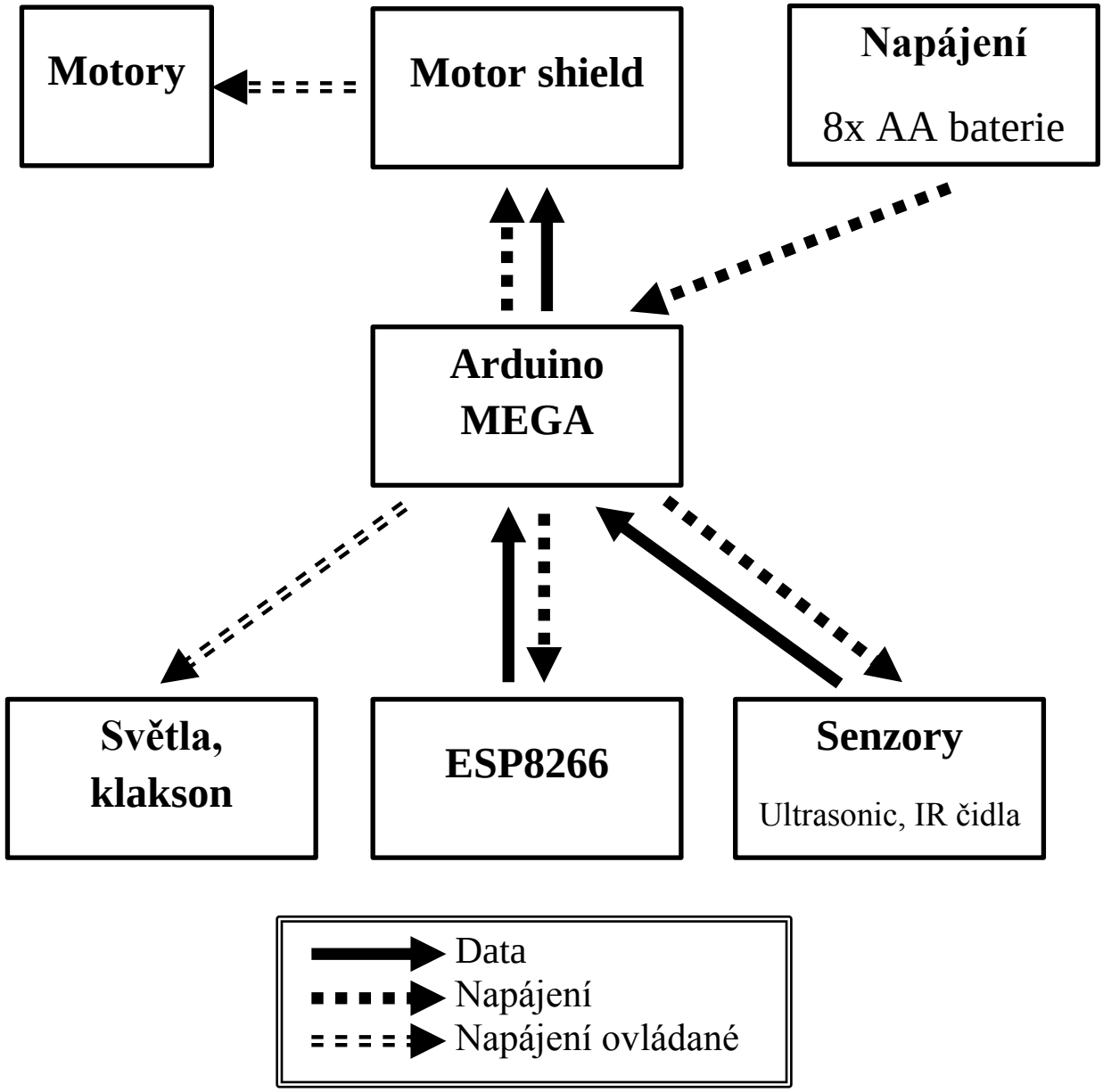
Obrázek 5 Autonomní vozítko 1



Obrázek 4 Autonomní vozítko 2

3 ELEKTRONIKA

3.1 Blokové schéma



3.2 Drobná elektronika

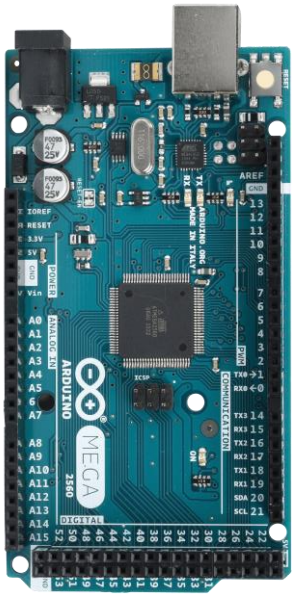
Použili jsme drobný elektrotechnický materiál, jako je rezistor, kondenzátor pro filtr napětí, shoty diodu pro zachycení zpětných napěťových špiček a nepájivé pole pro zapojení a propojovací vodiče.

3.3 Arduino MEGA 2560

Arduino Mega 2560 je deska založena na mikrokontroléru ATmega2560, tento vývojový kit obsahuje 54 digitalních pinů, z toho 15 PWM výstupu a 16 analogových vstupů. Tato vývojová deska je řízena krystalovým oscilátor o frekvenci 16 MHz. Arduino Mega je programovatelná v jazyce Wiring, ve vývojovém prostředí Arduino IDE.
[9]

Mikrokontrolér	ATmega2650
Architektura	AVR
Datová sběrnice	16-bit
Provozní napětí	5 V DC
Flash paměť	256 kB, 8kB použito pro bootloader
SRAM	8 kB
Taktovací frekvence	16 Mhz
Eeprom	4 kB
Stejnosměrný proud na pin	40 mA
vstupní napětí	5 ÷ 12 V DC
Odběr	38 mA

Tabulka 1 Arduino MEGA – parametry



Obrázek 6 Arduino MEGA

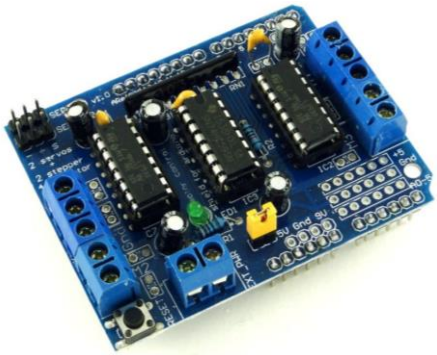
3.4 Motor shield L293D

Arduino motor shield L293D je přídavný modul pro platformu Arduino. Tento motor shield je určený pro jednoduché a bezpečné ovládání motorů prostřednictvím vývojových desek Arduino. Na tento shield je možné připojit tři druhy motorů, a to DC motory, servo motory a krokové motory a najednou lze připojit až šest motorů. Tento shield má čtyři konektory pro připojení motorů, jejich připojení lze i kombinovat.

Tento motor shield je také vybaven pro externí napájení, kvůli proudovému zatížení, proudové zatížená na jeden výstup je 0,6 A, špičkově až 1,2 A. Proto je tento shield také vybaven tepelnou ochranou řídicích obvodů. [9]

H-můstek	4krát
Proudový odběr na výstup	0,6 A (I_{max} = 1,2 A)
Napájení motorů	4,5–36 V DC
Ochrana proti přetížení	ANO
Čipy	2 × L293D, M74HC5981
Rozměry	70 × 53 × 20 mm

Tabulka 2 Motor shield – parametry



Obrázek 7 Arduino motor shield

3.5 Ultrazvukový senzor vzdálenosti HC-SR04

Senzor pro měření vzdálenosti pomocí ultrazvukových vln, přesnost senzoru až 3 mm a dosah až 4 m. Má čtyři piny: Vcc, GND, TRIG a ECHO. Nevýhodou tohoto senzoru je malý pracovní úhel a požadavky na rovnost překážky. [9]

Rozsah	20–4500 mm
Rozlišení	3 mm
Pracovní úhel	< 15°
Klidový proud	2 mA
Napájení	5 V DC
Rozměry	45 × 20 × 1,6 mm

Tabulka 3 Ultrasonic senzor – parametry

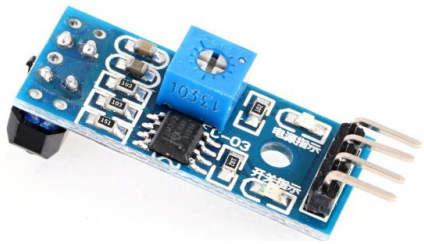
Obrázek 8 Ultrasonic senzor

3.6 Infračervený optický senzor TCRT5000

Tento IR snímač se používá k detekci barev a vzdálenosti, je založen na odrazu IR vln. Pro nastavení citlivosti je senzor vybaven potenciometrem. [9]

Napájení	3,3–5 V
Rozsah	1–8 mm
Čip	LM393
Rozměry	32 × 14 mm

Tabulka 4 IR senzor – parametry



Obrázek 9 IR senzor

3.7 Senzor překážek YL-70

Senzor překážek YL-70 se skládá z hlavní propojovací desky a čtyř detekčních modulů. Každý detekční modul obsahuje přijímací a vysílací diodu pro detekci překážky. Na hlavní desce je pro každý modul trimr pro nastavení citlivosti snímání a integrovaný obvod, který zajišťuje převod analogového signálu na digitální signál. [9]

V našem případě jsme raději detekční moduly připojili přímo k Arduino, a to proto, že získáme analogové hodnoty a ty zpracováváme přímo v programu. Je to výhoda hlavně při ladění obvodu. [9]

Napětí	3,3–5 V
Provozní teplota	-10° až +50 °C
Rozsah	10–600 mm
Hmotnost	21 g

Tabulka 5 Senzor překážek – parametry

Obrázek 10 Senzor překážek

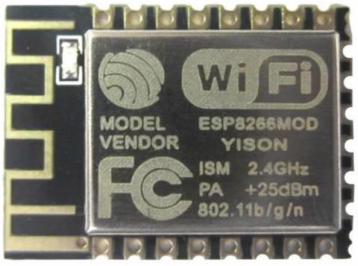
3.8 Motory

Použili jsme čtyři DC motory a každý motor ovládáme samostatně. Provozní napětí motorů je 3–12 V DC a zatěžovací proud činí 70 mA maximálně však 250 mA.

Obrázek 11 Motor

3.9 ESP 8266

Jedná se o WIFI čip pro komunikaci mezi Arduinem a okolím pomocí WIFI. Napájení je 3,3 V a provozní teplota je v rozmezí -40° až +125 °C. Průměrný pracovní proud je 80 mA, frekvenční rozsah je 2,4–2,5 GHz.



Obrázek 12 ESP8266

3.10 Napájení

Napájení je nezbytná součást vozítka pro jeho funkci a pohyb. Napájení vozítka je tvořeno osmi nabíjecími AA Ni-MH monočlánky, jejíž jmenovité napětí je 1,2 V a kapacita 2400 mAh. Tyto monočlánky jsou umístěny v držáku baterií, kde jsou zapojeny sériově.

Napájení není zcela ideální, protože kapacita monočlánku je poměrně malá a úroveň nabití ovlivňuje výstupní hodnoty z čidel, což se může projevit na chodu vozítka. Tyto monočlánky jsou měkký zdroj elektrického napětí, z toho důvody při nižší úrovni nabití dochází k velkému úbytku napětí při spuštění motorů. Tento úbytek napětí způsoboval vypadnutí napájení pro *ESP 8266* a došlo k zaseknutí vozítka. Tento problém jsme vyřešili tak, že jsme do obvodu přidali *Shottyho diodou* a kondenzátor pro vyhlazování úbytku napětí. V budoucnu bychom chtěli použít jiný způsob napájení, který by byl výhodnější.

4 PROGRAM

Komunikaci s mobilem zajišťuje mikropočítač ESP8266. Všechny ostatní části programu běží na Arduino Mega.

4.1 Komunikace s mobilní aplikací

ESP8266 po spuštění vytvoří vlastní šifrovanou wifi síť, na kterou se připojí mobil. Mobilní aplikace odesílá data pomocí argumentu GET, který následně ESP8266 přečte a odešle po sériové lince do Arduino Mega.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
const char* ssid = "Tatrovka";
const char* password = "12345678";
ESP8266WebServer server(80); //Server on port 80
void HTTP_handleRoot(void){
  if(server.args() != 0){
    Serial.println(server.arg(0));
  }
  server.send(200, "text/html","<html><head><title></title></head><body></body></html>");
}
void setup(void){
  delay(200);
  Serial.begin(115200);
  WiFi.mode(WIFI_AP);           //Only Access point
  WiFi.softAP(ssid, password);  //Start HOTspot removing password will disable security
  IPAddress myIP = WiFi.softAPIP(); //Get IP address
  server.on("/", HTTP_handleRoot);      //Which routine to handle at root location
  server.onNotFound(HTTP_handleRoot);
  server.begin();                  //Start server
}
void loop(void){
  server.handleClient();          //Handle client requests
}
```

Obrázek 13 Kód pro komunikaci mezi vozítkem a mobilem

4.2 Manuální ovládání

Poté, co Arduino mega po sériové lince přečte odeslána data z mobilu, může buď spustit autonomní pohyb vozítka nebo vozítko můžeme takto manuálně ovládat. V manuálním režimu ovládáme celý pohyb vozítka sami přes aplikaci. Arduino ovládá motory pomocí motor shieldu a knihovny AFMotor.h. Dalšími funkcemi manuálního režimu jsou uživatelem ovládané přední světlo a klakson nebo front assist, který podle údajů z ultrazvukového čidla dokáže automaticky zastavit vozítko před nárazem do překážky.

4.3 Sledování černé čáry

Program pro sledování černé čáry je autonomní program, který se spouští i vypíná přes mobilní aplikaci. Pro rozeznání černé čáry využíváme IR čidlo s analogovým výstupem, na kterém se mění napětí podle toho, jestli čidlo míří na černou plochu nebo na bílou. Toto napětí Arduino čte pomocí analogového vstupu. Pro sledování černé čáry vozítko v aktuální době využívá pouze jedno IR čidlo, ale do budoucna bychom chtěli tento program rozšířit na tři IR čidla, která již jsou ve vozítku implementovaná.

Program přímo ovládá motory podle toho, jestli je čidlo na černé čáře, či nikoli. Pokud je na černé čáře, zatáčí doprava, pokud na bílé, tak zatáčí doleva. Tímto algoritmem se vozítko pohybuje menší rychlostí, ale velmi spolehlivě po černé čáře pouze za pomocí jednoho čidlo. V pozdější fázi vývoje bychom tento algoritmus chtěli vylepšit na tří senzorový, u kterého by šlo teoreticky dosáhnout větších rychlostí i vyšší spolehlivosti. V aktuální fázi využíváme pouze jeden senzor, protože se nám zatím s vyšším počtem senzorů nepodařilo dosáhnout vyšší spolehlivosti, ani vyšší rychlosti.

```
1   if(analogRead(7) < 250){
2      PrekonejPrekazku();
3      if(cernaCara1 < 200){
4        motor2.run(FORWARD);
5        motor4.run(FORWARD);
6        motor1.run(RELEASE);
7        motor3.run(RELEASE);
8        motor2.setSpeed(105);
9        motor4.setSpeed(105);
10       motor1.setSpeed(105);
11       motor3.setSpeed(105);
12     }else if(cernaCara1 < 210){
13       motor2.run(FORWARD);
14       motor4.run(FORWARD);
15       motor1.run(BACKWARD);
16       motor3.run(BACKWARD);
17       motor2.setSpeed(255);
18       motor4.setSpeed(255);
19       motor1.setSpeed(255);
20       motor3.setSpeed(255);
21     }else{
22       motor2.setSpeed(195);
23       motor4.setSpeed(195);
24       motor1.setSpeed(195);
25       motor3.setSpeed(195);
26       cernaCara1 = 600;
27     }
28   }else{
28     }else{
29       cernaCara2++;
30       if(cernaCara2 > 5){
31         cernaCara1 = 0;
32       }
33       if(cernaCara2 < 15){
34         motor2.run(RELEASE);
35         motor4.run(RELEASE);
36         motor1.run(FORWARD);
37         motor3.run(FORWARD);
38         motor2.setSpeed(145);
39         motor4.setSpeed(145);
40         motor1.setSpeed(145);
41         motor3.setSpeed(145);
42       }else{
43         motor2.run(BACKWARD);
44         motor4.run(BACKWARD);
45         motor1.run(FORWARD);
46         motor3.run(FORWARD);
47         motor2.setSpeed(255);
48         motor4.setSpeed(255);
49         motor1.setSpeed(255);
50         motor3.setSpeed(255);
51         cernaCara2 = 100;
52       }
53     }
```

Obrázek 14 Program pro sledování černé čáry

4.4 Překonávání překážky

Překážka je definovaná jako krychle o rozměrech 120 × 120 × 120 mm. Script pro překonání překážky se sám spustí v autonomním módu sledování černé čáry, pokud na dráze nalezne dostatečně velkou překážku, kterou ultrazvukové čidlo zaregistruje. Pro překonávání překážky Arduino využívá již zmíněné ultrazvukové čidlo a boční IR senzory.

Poté co se program spustí, tak zastaví vozítko a začne se točit doleva, dokud bočními čidly nepozná, že stojí kolmo k překážce. Občas ale nastalo, že vozítko zastavilo o malý kousek dál od překážky a boční čidla ji následně nezaregistrovala a vozítko se točilo stále dokola. Tuto chybu jsme vyřešili pevným časovým intervalem, který nesmí otáčení přesáhnou (300ms).

Pevný časový interval jsme použili také u všech dalších kroků, které vozítko pro překonání překážky udělá. Po otočení kolmo k překážce se vozítko rozjede rovně, dokud boční čidla nezaregistrují, že se vozítko již nenachází u překážky. Poté se vozítko začne postupně stáčet doprava, tak aby skončilo cca 10 cm za překonávanou překážkou. Úspěšně překonání Arduino pozná pomocí čidla černé čáry, ale kvůli setrvačnosti vozítka i po vypnutí motoru se vozítko ve finální fázi nachází kousek za černou čárou a kolmo k ní. Proto dále vozítko začne pomalu couvat doprava, než opět narazí na černou čáru. Nakonec program opět spustí script pro sledování černé čáry a vozítko může dále pokračovat v jízdě.

```
void PrekonejPrekazku(){
  int poc = 0;
  if(vzdalenost < 7){
    int pr2 = PrumerA(15,3);
    int pr = PrumerA(15,3);
    while((PrumerA(15,3) > 680) and (poc < 300)){
      motor2.run(BACKWARD);
      motor4.run(BACKWARD);
      motor1.run(FORWARD);
      motor3.run(FORWARD);
      motor2.setSpeed(255);
      motor4.setSpeed(255);
      motor1.setSpeed(255);
      motor3.setSpeed(255);
      delay(1);
      poc++;
    }
    delay(100);
    motor2.run(RELEASE);
    motor4.run(RELEASE);
    motor1.run(RELEASE);
    motor3.run(RELEASE);
    delay(500);
    poc = 0;
      motor2.run(FORWARD);
      motor4.run(FORWARD);
      motor1.run(FORWARD);
      motor3.run(FORWARD);
      motor2.setSpeed(105);
      motor4.setSpeed(105);
      motor1.setSpeed(105);
      motor3.setSpeed(105);
      delay(400);
      delay(400);
    while(PrumerA(7,1) < 200){
      motor2.run(FORWARD);
      motor4.run(FORWARD);
      motor1.run(BACKWARD);
      motor3.run(BACKWARD);
      motor2.setSpeed(255);
      motor4.setSpeed(255);
      motor1.setSpeed(55);
      motor3.setSpeed(55);
    }
    motor2.run(RELEASE);
    motor4.run(RELEASE);
    motor1.run(RELEASE);
    motor3.run(RELEASE);
    delay(500);
    while(PrumerA(7,1) < 200){
      motor2.run(BACKWARD);
      motor4.run(BACKWARD);
      motor1.run(FORWARD);
      motor3.run(FORWARD);
      motor2.setSpeed(205);
      motor4.setSpeed(205);
      motor1.setSpeed(135);
      motor3.setSpeed(135);
    }
    motor2.run(RELEASE);
    motor4.run(RELEASE);
    motor1.run(RELEASE);
    motor3.run(RELEASE);
    delay(500);
  }
}
```

Obrázek 15 Script pro překonání překážky

5 APLIKACE

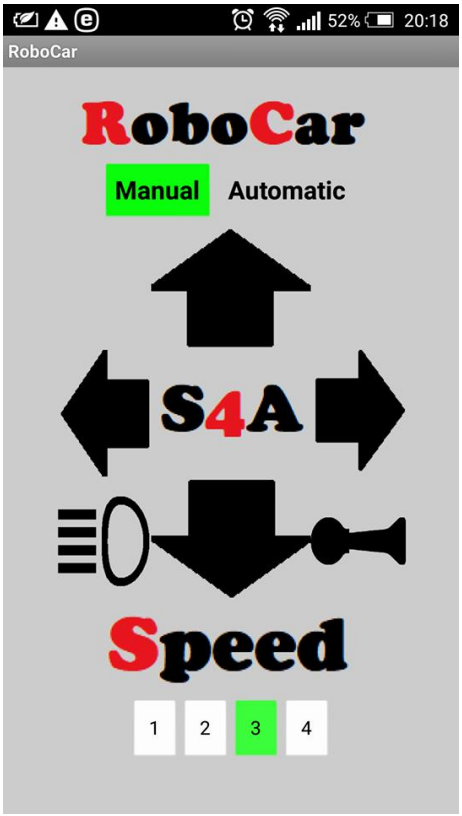
5.1 Aplikace pro Android

Aplikace pro Android je vytvořena v online vývojovém prostředí pro tvorbu aplikací Mit app invertor. Toto vývojové prostředí je grafické programování pomocí bloků a je rozděleno do dvou částí, a to *Designer* a *Blocks*.

V části *Designer* se navrhuje vzhled a rozložení akčních i pasivních členů. Je zde spousta možností, jakou funkci může mít daný prvek. Deklarujeme zde, zda se bude jednat o tlačítko, jenom nápis či obrázek. Také se zde implementují senzory, se kterými budeme chtít pracovat, čí komunikace s okolím. Jako komunikaci s okolím lze použít *Bluetooth* nebo *Web*. My jsme využili komunikaci *Web,* která pomocí WIFI odesílá data na nastavenou IP adresu. Pro odesílání dat je nastavena IP adresa ESP8266. Tyto čipy sice mají přednastavenou IP adresu, ale pro úspěšnou komunikaci musí být mobilní telefon připojen na WIFI síť daného *ESP*.

Blocks je část, kde se programuje, jak aplikace bude fungovat. Ke každému členu, který byl deklarován v části *Designer* jsou navrhnuty možnosti prvku, které s daným prvkem můžeme provádět. Mimo to je možno deklarovat proměnné, použít ovládací příkazy, či logické funkce.

Toto vývojové prostředí a tvorba aplikace byly pro nás nové, ale poradili jsme si s tím a stále poznáváme nové možnosti a funkce toho online prostředí pro tvorbu aplikací.

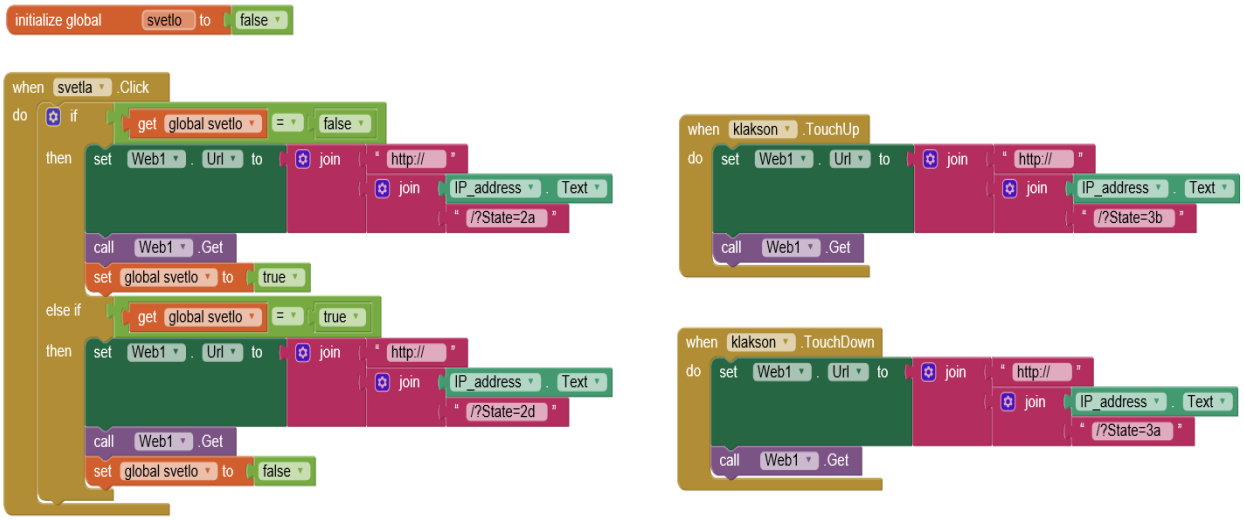


Obrázek 16 Aplikace pro Android

5.1.1 Popis kódu pro Aplikaci Android

Na obrázku níže je zobrazena ukázka kódu pro ovládání světel a klaksonu. Pravá část kódu je pro ovládání klaksonu, kde po zmáčknutí tlačítka *klakson* se nastaví kam chceme data posílat, v našem případě na *IP_address* a co chceme poslat. Ve všech případech se odesílá proměnná *State*, která nabývá určitých hodnot. Dále už dáme jen příkaz *call*, který nám odešle definovaná data na nastavenou IP adresu. Po uvolnění tlačítka *klakson* proměnná *State* nabude jiné hodnoty, data se odešlou a po přijetí *ESP* a převedení do *Arduina* se vykoná naprogramována funkce.

V levé části se jedná o ovládání světel. Protože chceme jedním tlačítkem zapínat a vypínat světla a nechceme jej držet, je zapotřebí deklarovat pomocnou proměnnou *svetlo*, která má dvě hodnoty, a to *true* nebo *false*. Když klikneme na tlačítko *svetla*, program porovnává proměnnou *svetlo* a podle toho jakou hodnotu nabývá, koná příkazy. Příkazy se liší v hodnotě proměnné *State* a proměna *svetlo* nabude hodnoty opačné. Takže při dalším kliknutí na tlačítko program splní druhou podmínku a vykoná příkazy, které jsou v ní uvedeny. Čili odešle jiná data, která zajistí zapnutí či vypnutí světel podle toho jaký byl předchozí stav.



Obrázek 17 Ukázka kódu – Aplikace pro Android

5.2 Aplikace pro Windows Phone

Pro ovládání našeho autonomního vozítka jsme vytvořili speciální aplikaci typu UWP (Universal Windows Platform, univerzální platforma Windows), která běží na všech zařízeních s Windows 10 (mobily, tablety, počítače). Primárně jsme ji vytvořili pro ovládání mobilním telefonem a tomu také odpovídá rozvržené prostředí.

5.2.1 Nastavení IDE Visual studio 2015

Pro programování UWP aplikace určenou převážně pro mobilní zařízení je potřeba vlastnit mobil s Windows 10 (v našem případě Lumia 640) a mít nainstalované Visual studio (2015) s rozšířením pro UWP aplikace. Programování je již velmi snadné, pokud programátor ovládá jazyk C# a základy značkovacího jazyku XML, který je velmi podobný známému značkovacímu jazyku HTML. Dále stačí pouze na zvoleném mobilním zařízení zvolit vývojářský režim a propojit ho USB kabelem s počítačem, na kterém vyvíjíme aplikaci.

Design i programování probíhá na počítači a po kliknutí na tlačítko *start* se nám program zkompiluje, odhalí případné chyby a v testovacím režimu spustí se na připojeném mobilu, kde můžeme aplikaci testovat. Po skončení testovacího režimu aplikace stále zůstává v paměti mobilního zařízení a nadále ji můžeme využívat.

5.2.2 Design ovládacího rozhraní

Design aplikace se vyvíjí ve značkovacím jazyce XML a grafickém prostředí Visual studio 2015, které umožňuje snadný grafický návrh aplikace i méně zkušenějším programátorům.

Prostředí aplikace jsme se snažili navrhnout snadno a intuitivně ovladatelné bez zbytečných ozdobných prvků. Pozadí, jak bývá zvykem v mobilních windows aplikací, je černé a prvky na něm bílé, tak jsme zachovali jednotné prostředí spolu s ovládacím rozhraním mobilního telefonu. V aplikaci se projevuje pouze málo barev (například označení zvolené rychlosti vozítka), pro co nejmenší narušení designu.

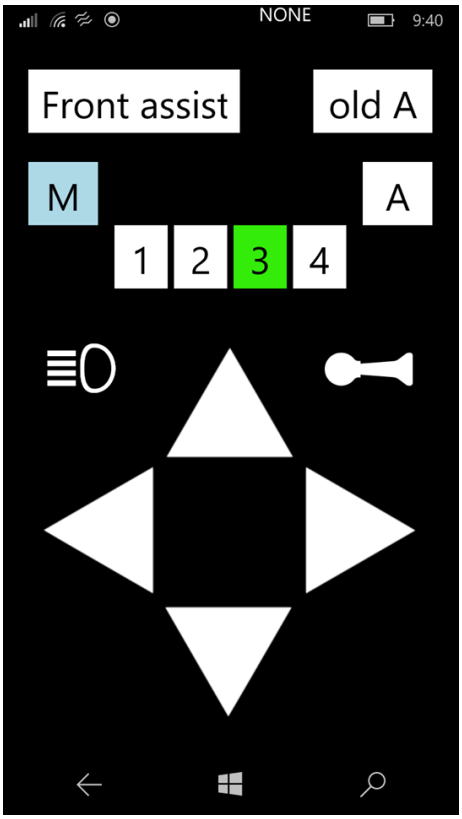
V horní části aplikace se nachází ovládání *front assist* (čelního senzoru), dále ovládání jednotlivých módů vozítka *manual, automotive, old automotive* (zmiňujeme se o nich v kapitole Funkce autonomního vozítka), které jsou na tlačítkách zastoupeny pouze zkratkami (M, A, old A). Zvolený mód se aktivuje stiskem tlačítka a tento zvolený mód se zbarví do světle modré barvy, kromě módu *automotive*, který po zvolení zůstává v manuálním módu, kdy pouze přepne světla do speciálního módu, který je rozsvítí pouze po najetí vozítka na černou čáru a až následným stisknutím tlačítka módu *automotive* se spustí autonomní režim vozítka. Tímto může uživatel velmi snadno spustit autonomní mód bez toho, aby musel odhadovat, jestli najel na černou čáru nebo nikoliv.

Pod módy se nachází ovládání rychlosti V dolní části se nachází směrové šipky pro manuální ovládání vozítka spolu s klaksonem a ovládáním světel vozítka. Výhodou dolního umístění ovládacích tlačítek manuálního módu je snadné ovládání vozítka pouze jednou rukou.

5.2.3 Programování ovládacího rozhraní

Funkční část UWP aplikace jsme naprogramovali v programovacím jazyce C#. Program je velmi jednoduchý neobsahuje žádné třídy, ale pouze jednotlivé funkce, které se spouští po stisku tlačítek. Tyto funkce posílají přes wifi síť data na vozítko přes HTML argument GET a využíváme na to předinstalovanou knihovnu *WebBrowser*. Každá funkce posílá jiná data, podle toho, jaké tlačítko spustíme.

Dále funkce obsahují jednoduché ovládání grafiky, které například zbarví šipku pro manuální ovládání do šeda, pokud ji stiskneme. Celý tento kód je zabalen do výjimkového bloku *try-catch*, který zajistí, že pokud nastane v programu nějaké chyba (například nepřipojení do wifi sítě vozítka), tak program pouze uživateli zahlásí chybu a dále pokračuje v běhu, nehrozí jeho spadnutí nebo zamrznutí. Zabalení celého kódu jsme si mohli dovolit pouze z důvodu jeho nenáročnosti, u náročnějšího kódu by toto řešení bylo velmi výpočetně náročné a museli bychom tak ošetřovat výjimky samostatně.



Obrázek 18 Aplikace pro Windows

6 FUNKCE AUTONOMNÍHO VOZÍTKA

6.1 Manuální ovládání

Vozítko je ovládané manuálně prostřednictvím WIFI skrze mobilní aplikace pro Windows a Android. Manuálně řídíme směr vozítka pomocí směrových šipek a rychlost vozítka je dána volbou rychlosti 1–4. Volí se zde také jednotlivé módy: manuální; autonomní a u aplikace pro Windows můžeme samostatně aktivovat či deaktivovat jízdního asistenta, který bude popsán níže. U aplikace pro Android je tento asistent přímo obsažen v jednotlivých módech.

6.2 Autonomní provoz

Při aktivaci autonomního provozu se vozítko pohybuje po černé čáře, kterou má pod sebou. Může jet po rovině, zatáčet pod různým úhlem ani 90° zatáčky nejsou problém. A to za pomocí naprogramovaného algoritmu, kterým se vozítko řídí.

6.3 Jízdní asistent

Vozítko je vybaveno *Ultrasonic senzorem,* který slouží jako přední jízdní asistent pro zabránění čelní kolize. V manuálním módu zabrání čelní kolizi, kdežto v autonomním módu slouží jako funkce pro překonání překážky.

Tento jízdní asistent zabrání čelní kolizi vůči rovné překážce, protože je zde jen jeden *ultrasonic sensor* uprostřed přední části a v případě, kdy je před vozítkem překážka s nepravidelným tvarem se ultrazvukové vlny se odrazí jiným směrem a *ultrasonic* jej nezachytí zpět.

Jízdní asistenty bychom chtěli vylepšit, a to přidáním čidel na přední část pro zachycení jakýchkoliv překážek a boční strany pro autonomní provoz vozítka mezi stěnami např. v bludišti. Toto jsou naše plány pro vylepšení, které jsme zatím nestihli realizovat.

6.4 Osvětlení a klakson

Dále je autíčko vybaveno předními světly a klaksonem. Jedná se o přední světla, která kdykoliv můžeme zapnout. Při zmáčknutí na klakson zazní bzučák a světla blikají. V plánu je rozšíření světel, a to přidáním zadních obrysových světel a směrových světel.

7 EDUKAČNÍ ÚČELY

7.1 Co se studenti, žáci mohou naučit

Studenti získají možnost naučit jsem novým věcem a podle stupně zkušeností mohou studenti pracovat na různých obtížnostech. Obtížnosti by se lišily počátečním stavem, tzn. úplným začátečníkům by byla dodána karoserie, základní kód a aplikace, které by společně dodělali a upravili. Chtěli bychom sepsat i manuál, popisující postup sestavení vozítka, aby byla větší škála použití, a za pomocí kterého by bylo možné vozítko sestavit i s minimálními zkušenostmi.

7.1.1 Týmová spolupráce

Určitě se zlepší v týmové práci, která je v dnešní době zapotřebí a určitě ji využijí v budoucnu. Podle nás je toto nejvýznamnější účel, protože i nás to velmi obohatilo v tomto směru, protože jsme si navzájem předávali své zkušenosti.

7.1.2 Návrh 3D komponentů

Naučí se v programu Solid Edge navrhovat 3D součásti, které se následně vytisknou na 3D tiskárně. Velkou výhodou je možnost návrhu jakékoliv karoserie, takže studenti můžou zapojit i svou fantazii.

7.1.3 Tvorba aplikace

Předpokládáme, že studenti budou vytvářet jen aplikaci pro Android, jelikož se jedná o jeden z nejrozšířenějších operačních systémů a tvorba aplikace v online vývojovém prostředí Mitt App Invertor je jednoduchá. Naučí se tedy vytvářet mobilní aplikace pro Android. V případě zkušeností či zájmu si mohou vytvořit i aplikaci pro Windows, což je ale obtížnější.

7.1.4 Programování platformy Arduino

Nedílnou součástí je naprogramovat Arduino, tak aby vozítko jezdilo. Naučí se tedy programovat tuto platformu, či si rozšíří své znalosti a dovednosti například použitím knihovny pro *Arduino motor shield.*

7.2 Využití na školách

Využití je především na středních technicky zaměřených školách, protože se zde spojuje více technických oborů. Pro strojní zaměření se jedná především o návrh karoserie a pro elektro a IT obory je programování a tvorba aplikace. Takže studenti si rozšíří své znalosti i v jiném technickém zaměření. Na základních školách by vozítko sloužilo jako ukázka, seznámení s komponenty a získání prvních dovedností v daných odvětvích.

7.2.1 Ukázka zadání

Naprogramujte vozítko tak, aby překonalo překážku před sebou. Vycházejte z programu pro sledování čáry a přidejte do něj prvky, které zajistí překonání vozítka.

Musíme si určit co po vozítku chceme a jak by to bylo možné realizovat. Vozítko jede po čáře, před ním je překážka, potřebujeme, ať ji rozpozná, k tomu použijeme Ultrasonic sensor. Když bude vzdálenost menší než cca 5 cm vozítko se zastaví, hodnoty jsou vždy relativní a je zapotřebí je otestovat a nastavit podle potřeby. Poté co vozítko zastaví, začne se otáčet, dokud na zadním senzoru nebude logická 1 v případě použití AD převodníku nebo hodnota detekující překážku, otáčení vozítka je vhodné provádět tak, že kola na každé straně se otáčejí opačným směrem, a to proto aby se vozítko otočilo na místě.

Poté co je identifikována log. 1 nebo analogová hodnota ze zadního čidla překážky, vozítko se rozjede dopředu, jede stále dopředu, dokud nebude z téhož čidla detekována log. 0, poté se vozítko otáčí do té doby, než na předním čidle překážky bude log. 1, následně vozítko jede dopředu opět tak dlouho, než na zadním čidle bude log. 0, pak se vozítko znovu otočí a jede tak dlouho dokud nenarazí na černou čáru. Poté, co čidlo identifikuje černou čáru, vozítko pokračuje v autonomním pohybu po čáře.

Bude se tedy jednat o podprogram, který se aktivuje, když bude čelní vzdálenost menší než nastavená hodnota a končí po vykonání jednotlivých kroků a detekování černé čáry. K překonání překážky je zapotřebí mít dobře nastavenou citlivost čidel nebo použít správné hodnoty, to je zapotřebí otestovat.

7.3 Porovnání s roboty Lego Mindstorms

Roboti Lego Mindstorms jsou velice dobrá a užitečná pomůcka, která rovněž rozvíjí technické dovednosti a programování. Oproti Lego Mindstroms hlavní výhodou toho vozítka je cena, protože cena robotů Lego Mindstorms může být i 10 499 Kč,[1]- a cena našeho vozítka je do 1000,- Kč, včetně nákladů na 3D tisk. Cena materiálu na 3D tisk činí cca 700,-/kg, hmotnost našich vytištěných komponentů je 320 g, náklady na tisk jsou tedy cca 224 Kč.

Cena vozítka je nízká, protože jednotlivé komponenty se dají nakoupit v Číně. Co se týče rozšiřování, tak jako je velké množství čidel pro Lego Mindstroms, tak je i spousta čidel platformy Arduino, možnosti jsou tedy v této části srovnatelné.

[1] Lego Mindstroms [online]. [cit. 2019-03-26]. Dostupné z: https://shop.lego.com/cs-CZ/product/LEGO-MINDSTORMS-EV3-31313

ZÁVĚR

Cílem bylo navrhnout a sestavit vozítko a dát tak studentům, či žákům základních škol možnost vyzkoušet si práci v Solid Edge a Arduino IDE, pracovat s 3D tiskem, programovat platformu Arduino a vytvářet mobilní aplikace.

Návrh a sestavení vozítka bylo úspěšné a nebylo to pro nás obtížné. Autonomní vozítko se autonomně pohybuje rychle a přesně, stejně tak i manuální režim funguje bez problému. Jediný problém je závislost úrovně nabití baterii na odečítaní hodnot z čidel.

Oproti vozítku S4A je vozítko trochu větší, ale lehčí, obzvláště proto, že podvozek je také vytisknut na 3D tiskárně, kdežto na S4A je kovový, díky čemu se lépe pohybuje. Sestavení a práce s ním byla snazší díky celkovému vlastnímu návrhu karoserie. Naše autonomní vozítko se pohybuje po černé čáře s vyšší přesností, jelikož máme k dispozici více čidel pro sledování černé čáry.

My sami jsme díky tomuto projektu nabyli nové zkušenosti a rozvinuli v dnešní době důležitou fantazii a práci v týmu. Doufáme, že se nám tento projekt podaří rozšířit a aplikovat nejen na naší škole, ale i na jiných školách, a tímto dát ostatním studentům možnost získat nové zkušenosti.

POUŽITÁ LITERATURA

1 *Arduino: Co je Arduino* [online]. [cit. 2018-03-26]. Dostupné z: https://arduino.cz/co-je-to-arduino/
2 *Arduino: co je to Arduino* [online]. [cit. 2018-03-26]. Dostupné z: http://czechduino.cz/?co-je-to-arduino,29
3 BARRAGÁN, Hernando. Wiring. *Wiring* [online]. [cit. 2018-03-26]. Dostupné z: http://wiring.org.co/
4 VODA, Zbyšek a TÝM HW KITCHEN. *PRŮVODCE SVĚTEM ARDUINA* [online]. [cit. 2018-03-26]. Dostupné z: https://arduino.cz/e-book-zdarma/
5 *Wiring: Programovací jazyk* [online]. [cit. 2018-03-26]. Dostupné z: https://cs.wikipedia.org/wiki/Wiring_(programovac%C3%AD_jazyk)
6 O 3D tisku [online]. [cit. 2019-03-17]. Dostupné z: https://josefprusa.cz/o-3d-tisku/
7 Solid Edge [online]. [cit. 2019-03-13]. Dostupné z: https://cs.wikipedia.org/wiki/Solid_Edge
8 *Arduino MEGA* [online]. [cit. 2019-03-21]. Dostupné z: https://cdn-reichelt.de/bilder/web/xxl_ws/B300/ARDUINO_MEGA_A03.png
9 ŠIMANÍK, Petr a KOLEKTIV. *Students for Automotive (S4A)* [online]. Ostrava, 2018 [cit. 2019-03-21].
10 Arduino motor shield [online]. [cit. 2019-03-25]. Dostupné z: https://shop.robotclass.ru/image/cache/data/Modules/Shields/SHLD-ARDU-MTR-L293D-1024x768_0.JPG
11 Ultrasonic sensor [online]. [cit. 2019-03-25]. Dostupné z: http://arduinolearning.com/wp-content/uploads/2014/12/HC-SR04-Ultrasonic-Sensor.jpg
12 IR sensor [online]. [cit. 2019-03-25]. Dostupné z: https://cdn.myshoptet.com/usr/www.hwkitchen.cz/user/shop/orig/2157-3.jpg?5ade35f8
13 IR sensor překážek [online]. [cit. 2019-03-25]. Dostupné z: https://cdn.myshoptet.com/usr/www.hwkitchen.cz/user/shop/orig/2163-3_4kanalovy-infra-senzor-prekazek-yl-70-pro-arduino.jpg?5ade35b1#a
14 TT Motor Smart Car Robot Gear Motor [online]. [cit. 2019-03-25]. Dostupné z: https://www.aliexpress.com/item/5pcs-lot-TT-Motor-Smart-Car-Robot-Gear-Motor-for-Arduino-Free-Shipping-Wholesale/1655276579.html?spm=a2g0s.9042311.0.0.2a3a4c4dNBT7mh&fbclid=IwAR2PPwpy9c5rl81sebvec6binVoNBFM97IZQS6VaqoJU3L7iWoIarCGRSX4
15 ESP8266 [online]. [cit. 2019-03-25]. Dostupné z: https://arduino-shop.cz/arduino/1352-esp8266-bezdratovy-modul-esp-12f-ap-sta.html
16 App Inventor: naklikejte si vlastní aplikaci pro Android [online]. [cit. 2019-03-27]. Dostupné z: https://www.maxiorel.cz/app-inventor-naklikejte-si-vlastni-aplikaci-pro-android
17 Lekce 1 - Visual Studio - Úvod do vývojového prostředí [online]. [cit. 2019-03-27]. Dostupné z: https://www.itnetwork.cz/csharp/visual-studio/tutorial-visual-studio-uvod

SEZNAM OBRÁZKŮ

SEZNAM TABULEK