



Středoškolská technika 2018

Setkání a prezentace prací středoškolských studentů na ČVUT

Inteligentní semafor pro řízení kyvadlové dopravy

Jiří Michálek

Střední průmyslová škola elektrotechnická a Vyšší odborná škola Pardubice
Karla IV. 13, 530 02 Pardubice

Prohlášení autora:

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.

V Pardubicích, dne

podpis

ZADÁNÍ

Navrhňte, sestrojte a naprogramujte model inteligentního semaforu pro řízení kyvadlové dopravy. Zařízení bude samo vyhodnocovat množství projíždějících aut a podle toho bude automaticky prodlužovat nebo zkracovat jednotlivé intervaly cyklu pro zvýšení propustnosti kyvadlové dopravy. Zařízení bude řízeno vlastním programem na platformě Arduino.

ANOTACE

Někdy se mohou řidiči dostat do situací, kdy zůstanou stát v kolonách vozidel kvůli rekonstrukcím a opravám komunikací, kde je průjezd řízen jen pomocí obousměrné kyvadlové dopravy. Snahou této práce je pokusit se takovéto situace řešit tak, aby se zvýšil a zrychlil průjezd vozidel v zatíženém směru. Tím můžeme získat menší kolony vozidel v takovýchto oblastech a snížení spotřeby pohonných hmot zapříčiněných stojícími vozidly v dlouhých kolonách. Zlepší se tím plynulost provozu, ušetříme pohonné hmoty a sníží se tím emise výfukových plynů.

Klíčová slova:

Arduino, vozidlo, kyvadlová doprava, ultrazvukový senzor, bezdrátová komunikace, řízení dopravy,

ANOTATION

Drivers sometimes get into situations when they get stuck in a traffic jam due to road reconstructions. In these situations there is often two-way shuffle. The objective of my work is to solve such situations by increasing the rate of passing vehicles in the direction which is fuller. Results can be smaller traffic jams and decreased air pollution emissions caused by standing cars. Fluency on the roads is going to improve and we are going to save more fuel along with decreased pollution.

Key words:

Arduino, vehicle, shuttle, ultrasonic sensor, wireless communication, traffic control,

OBSAH

OBSAH.....	- 6 -
1. ÚVOD.....	- 7 -
2. ZÁKLADNÍ PROBLEMATIKA PRÁCE.....	- 8 -
2.1. Semafor, normy a jeho základní rozdělení.....	- 8 -
2.2. Pojem havarijní semafor	- 9 -
2.3. Rozsah modelu	- 9 -
2.4. Možná rozšíření modelu.....	- 10 -
2.5. Cílová podoba projektu	- 10 -
3. PRAKTICKÁ ČÁST	- 11 -
3.1. Arduino	- 11 -
3.2. Počítání projíždějících vozidel	- 12 -
3.3. Bezdrátová komunikace	- 13 -
3.4. Signalizační část.....	- 14 -
3.5. Program a jeho logika	- 14 -
3.6. Ukázka programu	- 16 -
3.7. Výpočty v programu	- 19 -
3.8. Kryt na jednotku.....	- 20 -
3.9. Realizace modelu a vzhled.....	- 21 -
4. ZÁVĚR	- 22 -
5. ZDROJE A PŘÍLOHY	- 23 -
5.1. Zdroje a citace	- 23 -
5.2. Přílohy	- 24 -
5.3. Seznam obrázků	- 31 -
5.4. Přílohy na disku.....	- 32 -

1. ÚVOD

První impulz k této práci vznikl na základě cesty autem s mým otcem. Jakožto každý řidič jsem se jednou dostal do kolony vozidel čekající na zelenou v kyvadlové dopravě. Kolona byla dlouhá na několik časových cyklů a v protisměru nikdo nejel. To mne vedlo k tomu, abych se pokusil zabývat tímto problémem a vytvořil systém, který umí počítat projíždějící vozidla, zpracovat tyto data, porovnat je s předešlými a zregulovat tím délku jednotlivých časových intervalů. Za cíl je zrychlit celkovou dopravu, snížit náklady v takovýchto situacích a snížit tím stres, protože některé typy řidičů se tímto dostanou do stresu, začnou předjíždět a ohrožovat sebe i všechny ostatní účastníky provozu.

2. ZÁKLADNÍ PROBLEMATIKA PRÁCE

2.1. Semafor, normy a jeho základní rozdělení

Semafor je světelné signalizační zařízení určené k řízení dopravy. U nás se první semaforey objevily v roce 1911 v rámci řízení tramvajového provozu v Praze. V té době se ještě jednalo o řízení mechanické. První elektronická křižovatka v Evropě se objevila v Berlíně v roce 1924. Během vývoje došlo k začlenění světelných křižovatek do norem.

- ČSN EN 12368 Řízení dopravy na pozemních komunikacích - Návěstidla,
- ČSN EN 12675 Řízení dopravy na pozemních komunikacích – Řadiče světelných signalizačních zařízení – Funkčně bezpečnostní požadavky
- ČSN EN 50556 Systémy silniční dopravní signalizace
- „Zbytková“ ČSN 36 5601-1 Světelná signalizační zařízení. Technické a funkční požadavky. Část 1: Světelná signalizační zařízení pro řízení silničního provozu včetně základních požadavků na signální plán.

V této práci se pohybujeme v rámci normy ČSN EN 12675 a ČSN EN 50556. Světelné signály dále pak spadají do §70 až §74 zákona č. 361/2000 Sb., o silničním provozu. A dále pak podle §24 přílohy vyhlášky 30/2001 Sb. jsou užívány tyto signály:

- S 1 Tříbarevná soustava s plnými signály
- S 2 Tříbarevná soustava se směrovými signály
- S 3 Tříbarevná soustava s kombinovanými směrovými signály
- S 4 Signál žlutého světla ve tvaru chodce (umístěná vedle signálu se zeleným směrovým signálem pro odbočení, případně i před přechodem, jehož se týká) upozorňuje řidiče, že křížuje směr chůze přecházejících chodců. Nepřerušované i přerušované svítící signál má stejný význam.
- S 5 Doplnková zelená šipka (umístěná vedle signálu s červeným světlem *Stůj!*) – svítí-li současně s červeným nebo žlutým světlem, umožňuje pokračovat v jízdě příslušným směrem, řidič však musí dát přednost vozidlům ve volných směrech a nesmí omezit ani ohrozit přecházející chodce.
- S 6 Signál pro opuštění křižovatky (tzv. vyklizovací šipka, umístěná v protilehlém rohu křižovatky)
- S 7 Přerušované žluté světlo
 - Signál s červeným světlem *Stůj!* znamená povinnost zastavit vozidlo před příčnou čarou, a kde taková čára není, před signalizačním zařízením. V případě použití dvoubarevné soustavy (bez žlutého světla) má červené světlo stejný význam jako samostatně svítící žluté světlo.
 - Signál se současně svítícím červeným světlem a žlutým světlem *Pozor!* znamená povinnost připravit se k jízdě.
 - Signál se zeleným světlem *Volno* znamená možnost pokračovat v jízdě. Jde-li o signál se směrovým signálem nebo svítí-li pro odbočení vlevo Signál pro opuštění

křižovatky, není třeba při příslušném odbočování dávat přednost protijedoucím vozidlům a souběžným tramvajím. Jde-li o signál *Volno!* se směrovým signálem a nesvítlí-li současně signál žlutého světla ve tvaru chodce, není třeba při odbočování dávat přednost chodcům v kolizním směru.

- Signál s nepřerušovaně svítícím žlutým světlem *Pozor!* má obdobný význam jako signál *Stůj!*. Je-li však při rozsvícení signálu vozidlo tak blízko, že by řidič nestačil zastavit, smí pokračovat v jízdě.
 - Svítí-li žluté světlo *Pozor!* přerušovaně, nejde momentálně o křižovatku s provozem řízeným světelnými signály. Je-li signál užit společně s dopravní značkou nebo dopravním zařízením, zdůrazňuje jejich význam. Je-li použit samostatně, upozorňuje na nutnost dbát zvýšené opatrnosti.
 - K zabezpečení vjezdu na pozemní komunikaci se někdy používá návěstidlo pouze s červeným a žlutým světlem (bez zeleného).
- S 8a Zakázaný vjezd vozidel do jízdního pruhu
 - S 8b Volný vjezd vozidel do jízdního pruhu
 - S 8c Světelná šipka vlevo, S 8d Světelná šipka vpravo – příkazují opuštění jízdního pruhu nebo objetí překážky příslušným směrem
 - S 8e Světelný kříž – označuje překážku provozu vedle vozovky
 - S 12 Rychlostní signály s proměnným signálním znakem nebo s více signálními znaky, vyznačují doporučenou rychlost v km/h.
 - S 13 Signál dvou vedle sebe umístěných přerušovaných červených světél – má obdobný význam jako červené světlo dvoubarevné soustavy.
 - S 14 Signály přejezdového zabezpečovacího zařízení

2.2. Pojem havarijní semafor

Jako havarijní semafor zde definujeme světelný signalizační systém určený pro řízení kyvadlové dopravy. Tento systém je ve většině případů realizován jako jednoduché reléové zařízení propojené vodičem s druhým identickým zařízením, který má nastavený čas pro periodické střídání světelných signálů, které řídí běžnou dopravu narušenou nějakou neobvyklou situací v jednom z dopravních pruhů.

2.3. Rozsah modelu

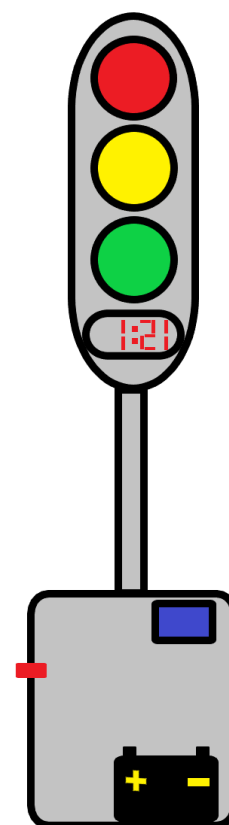
Model má znázorňovat reálnou dopravní situaci, kdy došlo k zapojení kyvadlové dopravy do provozu. Pro vyšší názornost je k modelu přidána i autodráha, na níž jde názorně demonstrovat průběžně se měnící hustotu dopravy v obou směrech. Rozsah modelu sahá do světelné signalizace, bezdrátové spolupráce mezi jednotkami pro oba směry, inteligentního řízení signalizace v závislosti na hustotě dopravy a snímání množství projíždějících vozidel.

2.4. Možná rozšíření modelu

Pro další práci je za cíl model rozšířit o několik prvků. Prvním by bylo jisté uživatelské rozhraní, ve kterém by bylo možno při zapnutí, ještě v části setup, nastavovat jednotlivé parametry, jako je doba pro bezpečné projetí úsekem, minimální a maximální doba pro zelenou, jednotlivé koeficienty a podobně. Toto by se nastavovalo jen u jedné z jednotek a tyto data by si obě jednotky mezi sebou bezdrátově přenesly. Realizace tohoto by bylo zpodobněna pravděpodobně na dvouřádkovém textovém displeji a ovládání by bylo čtveřicí tlačítek. Dalším nápadem na vylepšení modelu je časový displej, jež by odpočítával při červené ještě zbývající čas do konce této fáze. Tento krok by měl zvýšit trpělivost řidičů při čekání a umožňuje vypnout motor, pokud řidič vidí, že zbývá ještě velké množství času, což by šetřilo další pohonné hmoty a výfukové plyny. Mimo jiné by zde mohla být i možnost zpracovaná data odesílat na server, kdy by mohlo docházet k jejich vyhodnocení a další práci s nimi. Tedy v případě tvoření se dlouhých kolon v jednom směru by mohla být automaticky na základě software upozorňována média, která by dále upozorňovala řidiče a ti by měli prostor pro včasnou změnu trasy. Výhodou toho by byl obraz situace v reálném čase.

2.5. Cílová podoba projektu

Tento projekt cílí na zefektivnění kyvadlové dopravy. Kyvadlová doprava je omezením, se kterým se řidiči denně setkávají. Zvláště pak na frekventovaných místech se může stát velmi nepříjemnou skutečností. Stěžejním by mohla být spolupráce s některou stavební firmou, která se stará o komunikace ve větším až velmi velkém měřítku. Jedná se o projekt, jenž by měl být přijat s povděkem od velké skupiny lidí. Tedy vedení firem, které neustále hledá čas. Běžní lidé a zaměstnanci, ale i ekologové z důvodů ohleduplnosti vůči spotřebě paliva. Jedná se sice o nepatrná množství u jednoho dopravního prostředku, ale s ohledem na veškerá vozidla, jejichž množství se stále zvyšuje, se toto stává uváženíhodným. V cílové podobě by projekt měl disponovat řídicí jednotkou pro zpracování dat, baterii pro udržení systému při životě po dobu alespoň týdne, světelný signalizační systém, uživatelským prostředím pro možnost nastavení podmínek daného úseku, časovačem pro odpočet času do konce červené a senzor schopný snímat projíždějící vozidla za každého počasí. V případě zájmu zadávající firmy lze začlenit i systém pro odesílání statistik, nebo připojit GPS systém proti odcizení.



obrázek č. 1 – návrh semaforu

3. PRAKTICKÁ ČÁST

3.1. Arduino

Pro práci byl použit klon Arduina UNO, model CH340G. Jedná se o plnohodnotnou náhradu originální verze se standardním mikroprocesorem ATmega328P, jež je nahrazen kompaktnějším modelem mikroprocesoru se stejnou vnitřní stavbou. Výhodou tohoto modelu je zavedení pájitelných otvorů v tištěném spoji a rozšíření o několik napájecích a zemních otvorů, proti licencovanému standardu. Tato výhoda je protiváhou za znemožnění vyjmutí samotného procesoru z vývojové desky a umístění ho do vlastního tištěného spoje. Pro práci se jedná pravděpodobně o nejlepší možnost s ohledem na potřebné množství zemnicích a napájecích vstupů. Možnou zvažovanou alternativou byla vývojová deska ESP12E a ESP32, jejichž velkou nezpochybnitelnou výhodou byl koprocesor ovládající Wi-Fi komunikaci integrovanou v samotné desce. Nespornou výhodou byla však zkušenost s výrobky společnosti Arduino a s tím spojeno jejich programování. HW specifikace samotného užitého modelu, mimo již zmíněné vstupy pro napájení vodičů, jsou totožné s originálním Arduinem UNO, tedy:

Pracovní napětí: 5 V

Vstupní napětí: 7-12 V

Vstupní nap max.: 6-20 V

I/O Piny: 14 (6 použitelných jako PWM výstup)

Analog. vstupy: 6

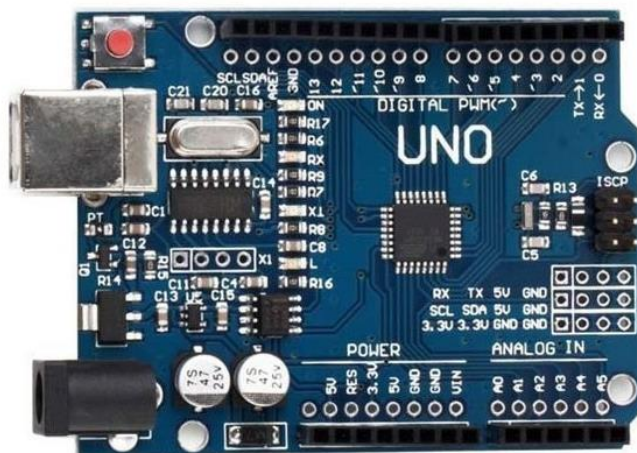
DC Proud na pin: 40 mA

Flash: 32 KB (ATmega328) 0.5 KB použito pro bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Krystal: 16 MHz

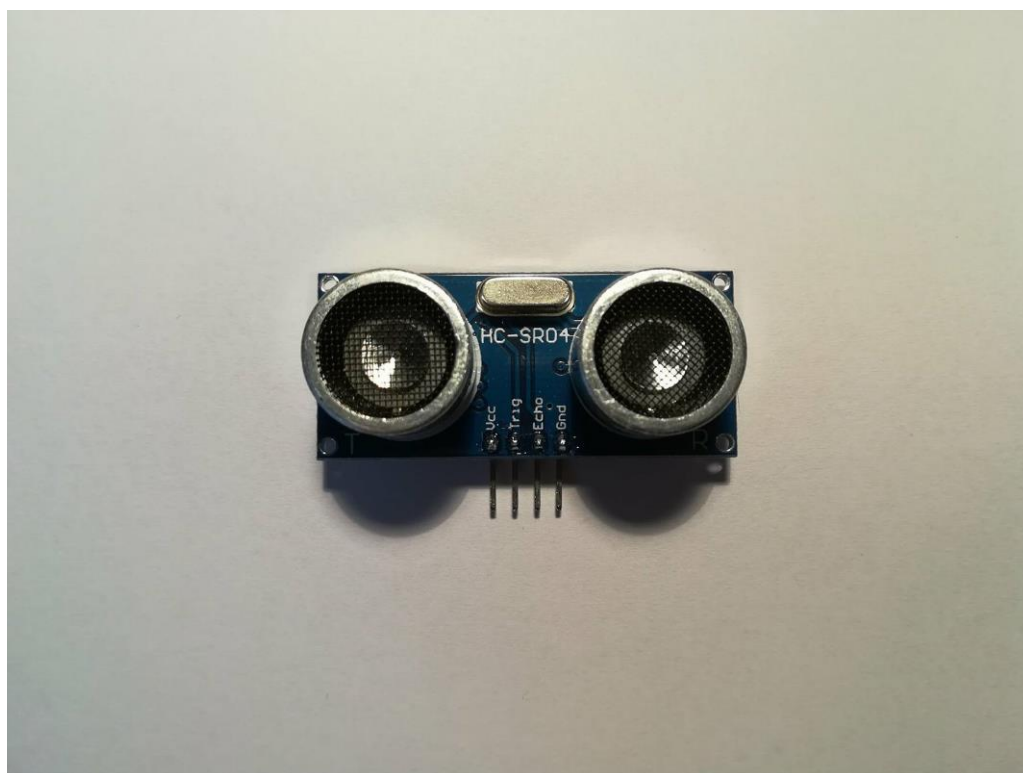


obrázek č. 2 – Arduino UNO CH340G

Pro napájení byl použit zdroj V0114 od společnosti EMOS. Tento model je vybaven dvěma USB výstupy. To nahrazuje napájecí zdroj, který by byl v běžné praxi nahrazen akumulátorem. Pro možnost vypnutí mikroprocesorů semaforů i jiným způsobem, než je vytažení ze zdroje, byl USB kabel typu B přerušen v napájecím vodiči a byl napojen na dvoupolohový přepínač.

3.2. Počítání projíždějících vozidel

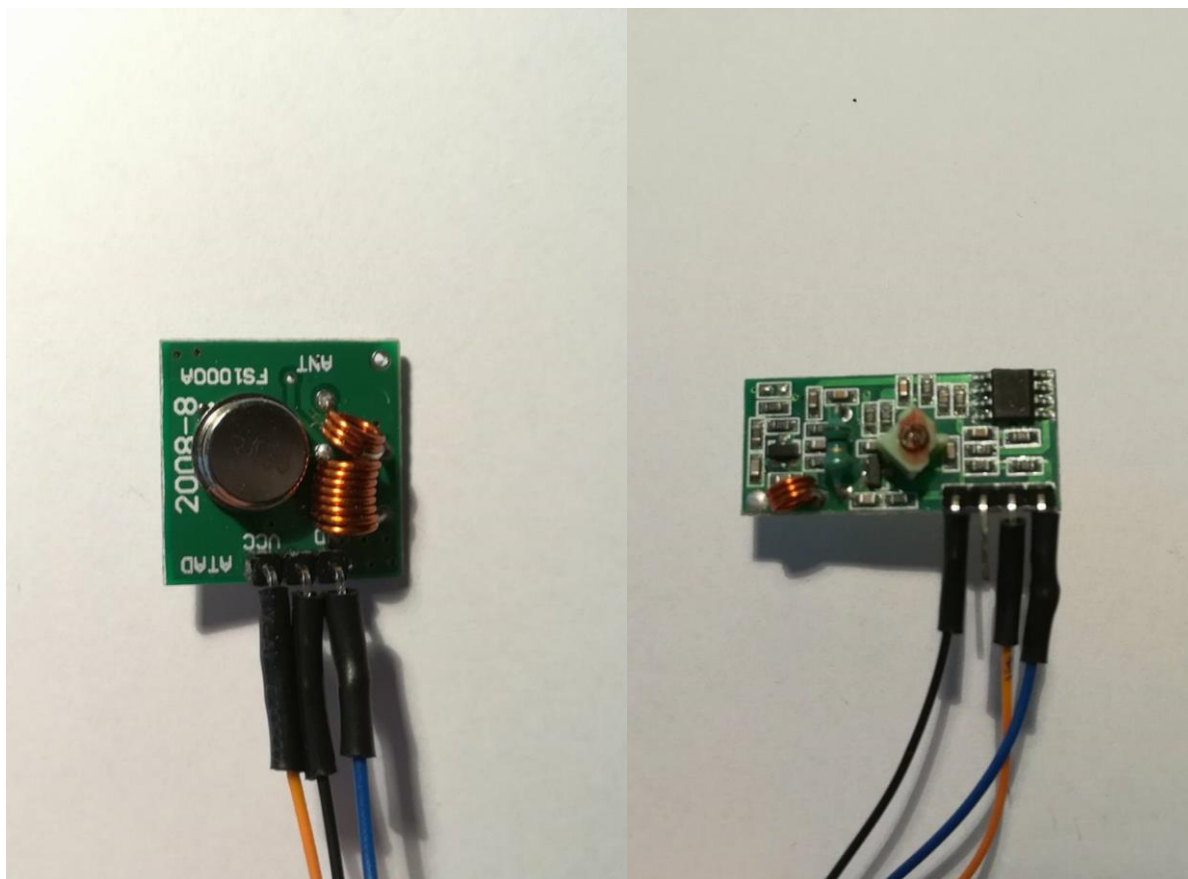
Pro počítání množství projíždějících vozidel byl zvolen ultrazvukový měřič vzdálenosti HC-SR04. Tento senzor funguje na jednoduchém principu, kdy vysílač vyšle krátký ultrazvukový pulz, který se odrazí od případného objektu, vrátí se a my díky znalosti rychlosti zvuku můžeme dopočítat vzdálenost. Toto zařízení zvládá pracovat až na vzdálenost 4 m s přesností v řádech milimetrů. Těmito parametry se stal senzor pro model nejlepší možností. Pro skutečnou realizaci by byl použitelný jen teoreticky. Navzdory tomu dosah by měl splnit potřebné minimum, neboli do dvou třetin jízdního pruhu. Bylo by však nutné provést testování ve skutečných podmínkách, zvláště pak za deště, nebo silného větru. V případě že by se nenalezla lepší alternativa a došlo by ke stavu, kdy by hustý déšť byl považován za jedno auto, nebo velmi hustý provoz, pak je program postaven tak, aby pracoval v dosavadním, nebo výchozím režimu. V dřívějších dobách byly užívány tlakové senzory napojeny na hadici nataženou přes silnici, například při měření rychlosti. Toto řešení je však již dávno překonáno a dnešní technologie nám umožňují užívat v běžné praxi lepší senzory, zvláště pak ty, které nezasahují do vozovky.



obrázek č. 3 – ultrazvukový senzor

3.3. Bezdrátová komunikace

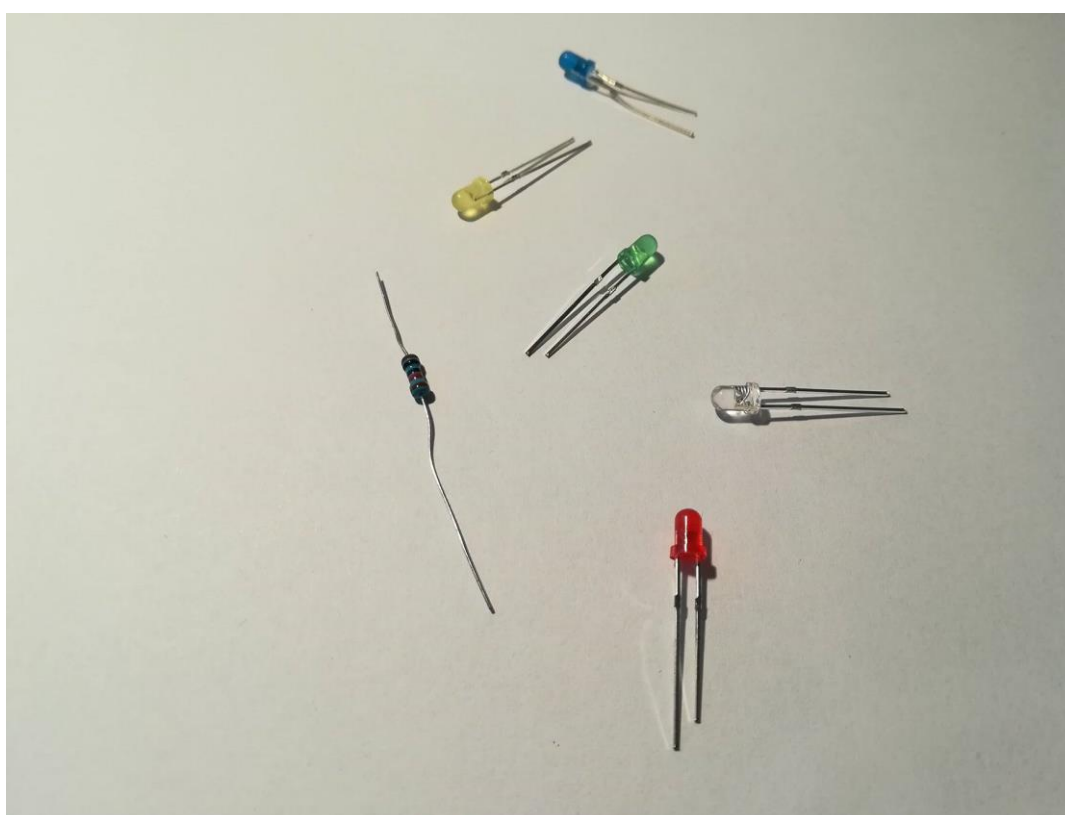
Pro bezdrátovou komunikaci byla užitá dvojice tvořena přijímačem a vysílačem pracující na frekvenci 433 MHz. Tabulkové hodnoty udávají schopnost pracovat na vzdálenost 20 až 200 m, což pro realizaci modelu je více než postačující. Dalším plusem pak je relativně kompaktní knihovna VirtualWire. Pro praktickou realizaci bychom se však mohli potýkat s problémy schopnosti komunikace na dostatečnou vzdálenost. To by muselo být vyřešeno rozšířením o výkonnější anténu do přístupných slotů, nebo tyto senzory nahradit výkonnějšími alternativami. Již pro model byla zvažována možnost Wi-Fi modulu určeného pro komunikaci mezi dvojicí platforem Arduino. To ale nakonec prohrálo kvůli zbytečné složitosti zápisu programu a náročnosti knihovny na úložiště. Proto vyhrála komunikace na frekvenci 433 MHz spolu s knihovnou VirtualWire, která umí rozložit data na hexadecimální kód, odeslat jej na druhou stranu a tam ho opět vrátit do původní podoby. Bohužel zde není umožněno odeslat zpět informativní bit o přijetí zprávy a tím je znemožněn takzvaný handshaking. Tuto skutečnost bylo nutno vyřešit opakovaným odesláním informace, čímž se pak limitně blížíme jistotě k přijetí zprávy.



obrázek č. 4 – přijímač a vysílač 433 MHz

3.4. Signalizační část

Signalizační část pro model byla realizována za pomoci čtveřice 5 mm LED. Nejprve pak červená, žlutá a zelená pro ukázkou signalizace pro řízení dopravy a jako čtvrtá je použita bílá, nebo modrá dioda. Tato čtvrtá dioda slouží ke dvěma účelům. Nejprve pak pro informování o probíhající bezdrátové komunikaci a dále pak pro rozlišení Master a Slave zařízení. Přičemž Master je označen modrou LED a Slave bílou. Pro běžnou skutečnou realizaci by musela být trojice řídicích diod nahrazena standardizovanými semaforovými světly. Z důvodů vysokého napětí pro tyto diody, které Arduino má na I/O výstupech, je nutno k diodám připojit i odpory. Pro udržení dostatečné svítivosti v poměru s životností byly zvoleny odpory 3k3 ohmu.

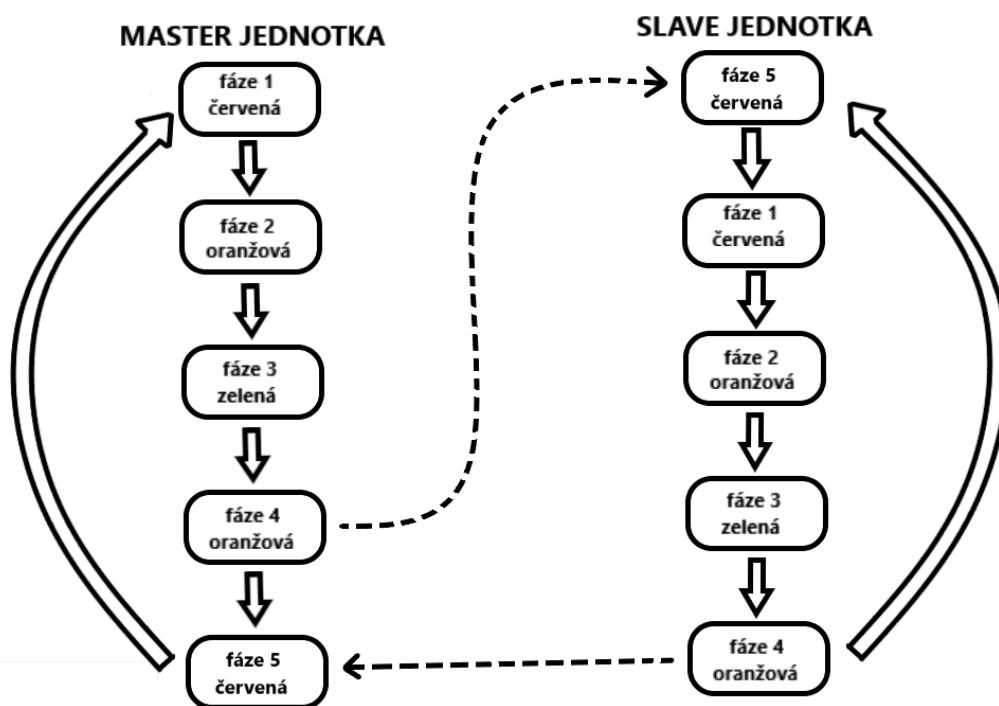


obrázek č. 5 – diody 5 mm a odpor 3k3 ohmu

3.5. Program a jeho logika

Program byl naprogramován v prostředí Arduino IDE verze 1.8.6. Program je řešen poměrně jednoduše. Dvojice jednotek je dělena na Master a Slave. Nejprve je nutno zapnout jednotku Slave. Ta se spustí, projde nastavovací protokolem a přejde do režimu, kdy čeká na impuls od Master jednotky. Když spustíme master jednotku prochází v rámci programu pěti fázemi. Jednotka Slave má totožný program, jen jednotlivé fáze má posunutě. Fáze jedna-červená. Jedná se o bezpečnostní fázi, kdy svítí na jednotce červené světlo. Je hlavně uplatnitelná až při několikatém opakování programu, složí k bezpečnému projetí vozidel,

přestože někdo vyjel ještě na oranžovou. Fáze dvě-oranžová. Tato fáze má rozsvícené červené a oranžové světlo a slouží k analyzování dat z předešlého cyklu této jednotky a z právě ukončeného cyklu jednotky v protisměru. Po zhodnocení dat dojde k nadefinování délky časového úseku pro třetí fázi. Při zhodnocování probíhá tak, že se porovnají obě hodnoty. Pro toto existuje pět možností. První, že došlo k značnému převýšení aut z protisměru a délka zelené bude zkrácena o nadefinovaný čas. Druhá, že je číslo srovnatelné časy se nezmění. Třetí možností je větší průjezd aut v tomto směru, tedy dojde o prodloužení doby o stejný časový úsek jako ve fázi jedna. Čtvrtou možností je výrazné převýšení množství aut v tomto směru, než protisměru, tedy protažení úseku o dvojnásobnou časovou hodnotu. A poslední, pátá možnost je, že v obou směrech je minimální, téměř nulová průjezdnost aut, pak dojde k restartování intervalu do základního režimu. Fáze tři-zelená. V průběhu této fáze je rozsvíceno zelené světlo a dochází k počítání projíždějících vozidel a k ukládání dat do proměnné. Fáze čtyři-oranžová. Čtvrtá fáze je určena k odeslání informace o množství projetých vozidel do protisměru. Během tohoto je rozsvíceno oranžové světlo a dojde k opakovanému odeslání informace. To je zapříčiněno nedostatečně vysokou spolehlivostí komunikátorů a chybějící zpětnou vazbou. Během této fáze dochází ke spuštění první fáze u jednotky v protisměru. Pátá fáze-červená, je fází poslední. Během ní svítí červené světlo a dojde k zacyklení, během kterého program čeká, dokud nepřijde pokyn od jednotky z protisměru. Díky takovému postavení programu v případě chyby nikdy nedojde k tomu, aby v obou směrech byla zelená fáze, jedinou možností je obousměrné zacyklení v poslední červené fázi.



obrázek č. 6 – fázové rozvržení běhu programu

3.6. Ukázka programu

```
//program pro master jednotku
#include <VirtualWire.h> //začlenění knihovny pro bezdrátovou komunikaci
#define modra 11 //přiřazení vývodů a označení pro udržení přehlednosti kódu
#define cervena 7
#define zluta 8
#define zelena 9
#define pTrig 4 //vývody k vysílači a přijímači ultrazvukového senzoru
#define pEcho 5

//hodnoty do kterých se uloží hodnoty vzálenostního senzoru
long odezva;
long vzdalenost;

int TX_PIN = 10; //nastavení PINu pro odesílání dat
int TX_ID = 3; // ID adresa pro odesílání
int RX_PIN = A0; // nastavení PINu pro přijímání dat
int RX_ID = 3; // ID adresa pro přijímání

typedef struct roverRemoteData // datová komunikační struktura proměnných
{
int TX_ID;
int Sensor1Data;
};

int doba1 = 10000; //minimální délka cyklu (v ms)
int doba2 = 35000; // maximální délka cyklu (v ms)
int doba3 = 6000; // doba pro přechody (čas pro bezpečné projetí auta úsekem) (v ms)
int d = 10; //vzdálenost od senzoru do dvou třetin pruhu (v cm)
int a = 3; //množství aut který zmení čas cyklu
int b = 10; //množství aut které výrazně zmení čas cyklu (prodloužení o dvě konstanty)
int p = 0; //počet aut, která projela v protisměru
int c = 0; //počet aut z minulého cyklu
int cas = 20000; //čas cyklu (v ms)
int x = 3000; //konstanta rozdílu času (v ms)
int zaklcas = 20000; //neměnný základní časový úsek cyklu (v ms)
unsigned long cas1; //časy pro millis
unsigned long cas2;
int m = 0; // proměnná pro opakované odeslání hodnoty projetých aut

void setup() {
Serial.begin(9600); //rychlost komunikace
pinMode(modra, OUTPUT); //nastavení vstupů a výstupů jednotlivých PINů
pinMode(cervena, OUTPUT);
pinMode(zelena, OUTPUT);
```



```

pinMode(zluta, OUTPUT);
pinMode(pTrig, OUTPUT);
pinMode(pEcho, INPUT);

vw_setup(2000); //nastavení vysílače
vw_set_tx_pin(TX_PIN);

vw_set_rx_pin(RX_PIN); //nastavení přijímače
vw_rx_start();

digitalWrite(cervena, HIGH); //nastavení výchozího stavu semaforu, tak aby se zabránilo
kolizi
digitalWrite(zluta, LOW);
digitalWrite(zelena, LOW);

digitalWrite (modra, LOW); // nastavení indikační komunikační diody
}

void loop() {
//fáze 1 - červená
digitalWrite (cervena, HIGH);
delay (doba3);
if (cas<doba1) {cas = doba1 + x;}
if (cas>doba2) {cas = doba2 - x;}
//fáze 2 - oranžová (porovnání hodnot projetých aut, nadefinování délky cyklu)
digitalWrite (zluta, HIGH);
if ((p+b)<c) {cas = cas + (2*x); }
else if ((p+a)<c) {cas = cas + x;}
else if (p>(c+a)) {cas = cas - x;}
else if (p<3 && c<3) {cas = zaklcas;}
else {cas = cas;}
delay (3000);
//fáze 3 - zelená (počítání projetých aut po danou dobu)
c=0; //vynulování množství projetých aut cyklu v paměti
digitalWrite (cervena, LOW);
digitalWrite (zluta, LOW);
digitalWrite (zelena, HIGH);
cas2 = millis(); //cyklus pro počítání aut vypočítaný časový úsek
while (cas1 < (cas2 + cas)) {
cas1 = millis();
//vypuštění zvukového pulzu
digitalWrite(pTrig, LOW);
delayMicroseconds(2);
digitalWrite(pTrig, HIGH);
delayMicroseconds(5);
digitalWrite(pTrig, LOW);
}
}

```

```

//změření odezvy a přepočet (za pomoci konstanty) na vzdálenost (v cm)
odezva = pulseIn(pEcho, HIGH);
vzdalenost = odezva / 58.31;
if (vzdalenost < d) { //pokud by byla naměřena dostatečně krátká vzdálenost, pak se přičte
1 do množství projetých aut
    c=c+1;
    while (vzdalenost < d) { // vyčkání dokud auto celé neprojede, aby se nestalo opakované
zaznamenání stejného auta
        digitalWrite(pTrig, LOW);
        delayMicroseconds(2);
        digitalWrite(pTrig, HIGH);
        delayMicroseconds(5);
        digitalWrite(pTrig, LOW);
        odezva = pulseIn(pEcho, HIGH);
        vzdalenost = odezva / 58.31;
    }
}
}
//fáze 4 - oranžová (odeslání informace o počtu projetých aut)
digitalWrite (zluta, HIGH);
digitalWrite (zelena, LOW);
while (m<50){
    struct roverRemoteData payload;
    payload.TX_ID = TX_ID;
    payload.Sensor1Data = c;
    vw_send((uint8_t *)&payload, sizeof(payload)); // odešli data
    vw_wait_tx();// vyčkej na odeslání všech dat
    m=m+1;
}
delay (2500);
m=0;
//fáze 5 - červená (čekání na příchozí data z protisměru)
digitalWrite (zluta, LOW);
digitalWrite (cervena, HIGH);
p=5000; // nastavení aut v protisměru na 5000, tak aby každá libovolná reálná hodnota
mohla být zaznamenána
while (p==5000) { // čekání na příchozí data a nastavení proměnných a kódování přenosu
    struct roverRemoteData receivedData;
    uint8_t rcvdSize = sizeof(receivedData); //změření velikosti příchozích dat
    vw_rx_start();// start příjmu dat

    if (vw_get_message((uint8_t *)&receivedData, &rcvdSize)) // kontrola, zda-li jsou
přístupná nějaká data
    {
        if (receivedData.TX_ID == RX_ID) //kontrola, zda-li odpovídá ID přenosu
        {

```

```

        digitalWrite( modra, HIGH); //rozsvícení komunikační diody pro indikaci
přenosu
        p = receivedData.Sensor1Data; //příjem dat
        }
        delay (100);
        digitalWrite(modra, LOW);
        if ( p < 5000) {
            vw_rx_stop();
        }
    }
}
}
}

```

3.7. Výpočty v programu

V rámci programu je užito několik specifických výpočtů. Prvním je výpočet pro zabezpečení toho, aby se čas pro zelenou fázi držel v jistých minimálních až maximálních mezích.

```

if (cas<doaba1) {cas = doaba1 + x;}
if (cas>doaba2) {cas = doaba2 - x;}

```

Další výpočet slouží k nadefinování příbytku, velkého příbytku a úbytku času v rámci porovnávání dat pro výpočet délky zelené fáze.

```

if ((p+b)<c) {cas = cas + (2*x); }
else if ((p+a)<c) {cas = cas + x;}
else if (p>(c+a)) {cas = cas - x;}
else if (p<3 && c<3) {cas = zaklcas;}
else {cas = cas;}

```

Jako třetí zásadní výpočet v rámci programu figuruje výpočet vzdálenosti u ultrazvukového čidla. Toto funguje na principu vyslání krátkého pulzu. Ve chvíli kdy se nám vrátí, jej vydělíme fyzikálně známou časovou konstantou a tím získáme vzdálenost v cm.

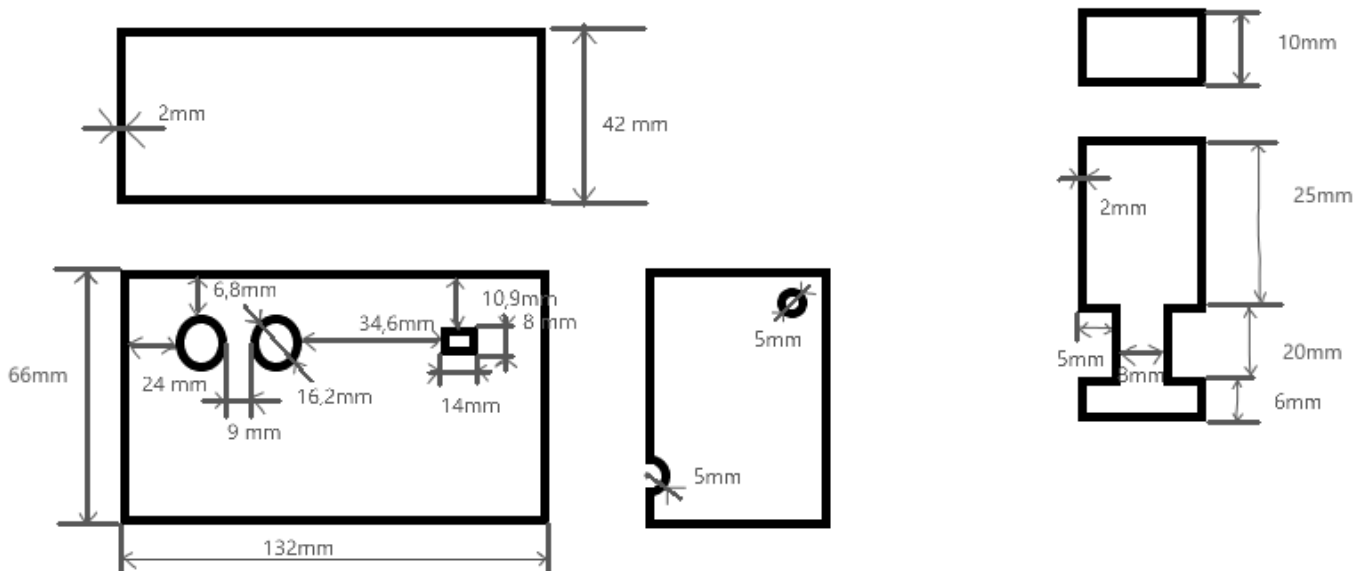
```

digitalWrite(pTrig, LOW);
delayMicroseconds(2);
digitalWrite(pTrig, HIGH);
delayMicroseconds(5);
digitalWrite(pTrig, LOW);
odezva = pulseIn(pEcho, HIGH);
vzdalenost = odezva / 58.31;

```

3.8. Kryt na jednotku

Obal na jednotlivé díly modelu je vytisknutý 3D tiskárně. Skládá se z části pro logickou jednotku a z části pro indikační část. První část je zpodobněna dvoudílnou krabičkou, oba díly jsou totožné, jen jeden je vybaven krom půlkruhového otvoru ještě dvěma otvory na ultrazvukový senzor, otvorem pro přepínač a drobnou dírou, kterou je vidět na indikační diodu. Druhá, indikační část, je tvořena opět ze dvou dílů. Tato část má připomínat samotný semafor, přičemž jeden z dílů je opatřen šesticí malých děr, do kterých je možno pohodlně, ale při tom pevně vložit trojici diod pro znázornění světel. Obě tyto části budou vsazeny do desky určené pro model, tudíž nebude hrozit rozevření dvou dílů od sebe, jediné co bude třeba je za aretovat jednotky aby nedošlo k jejich samovolnému vysunutí. Upevnění jednotlivých dílů v rámci jednotky je za pomoci vteřinového lepidla a montážních podložek k tomu určených. Samotný obal je tisknut standardním černým ABS plastem určeným pro 3D tiskárny. Návrh dílů tisknutých na 3D tiskárně je z programu Inventor. Ilustrativní fotografie přiloženy v rámci příloh.



obrázek č. 7 – rozvržení krytů na jednotky

3.9. Realizace modelu a vzhled

Model samotný tvoří několik částí. První část je tvořena řídicí jednotkou, jedná se plastovou krabičku obsahující senzor, komunikaci a celou řídicí část, tato jednotka je pro každý směr. V rámci desky je jednotka zapuštěna tak, že na druhé straně jsou mimo desku 2 cm. Druhou část tvoří indikační jednotka. Zpodobňuje semafor a je v ní obsažena trojice diod. I toto je ve dvojici. Jednotka je zapuštěna do desky do hloubky 6mm. Třetí část je dřevěná OSB deska jejíž rozměry jsou k nalezení ve fotografii v rámci příloh. Čtvrtou částí je čtveřice dutých plastových válců vytisknutých na 3D tiskárně. Průměr těchto válců jsou 3 cm a 4 cm. Přesný návrh všech plastových částí tištěných na 3D tiskárně je na disku přiloženému k této práci. Rozmístění válců je na spodní straně desky, vždy v polovině strany 1 cm od hrany. Pátou část tvoří síťový napájecí zdroj pro dvojici napájecích kabelů typu USB-B. Veškeré vnitřní obvody v rámci modelu jsou taženy třemi barvami vodičů. Černá znázorňující napájení, modrá znázorňující zem a oranžová, která je požitá jako signálový vodič. Ve všech třech případech se jedná o měděný drát o průřezu 0,4 mm² s povrchovým pocínováním. Uprostřed modelu jsou umístěny dva díly z autodráhy, za jejíž pomocí lze efektivně a názorně demonstrovat funkčnost tohoto výrobku.



obrázek č. 8 – výsledný vzhled modelu

4. ZÁVĚR

Jako autor této práce v ní vidím velký potenciál. Ne jen že by v případě úspěšné realizace mohla ročně ušetřit stovky litrů pohonných hmot. Ale hlavně mnoho nervů řidičům, kteří pak často z takovýchto důvodů jsou nervózní, na silnici spěchají a ohrožují sami sebe i všechny ostatní účastníky silničního provozu. A každý takto zachráněný lidský život se počítá. Dle mého pozorování, jakožto účastníka silničního provozu, je v tomto směru prostor pro zlepšení. Věřím, že mou práci ocení většina členů společnosti, a to jak běžní řidiči, vedení firem, tak i ochránci přírody a životního prostředí. Práce jako taková byla pro realizaci vcelku příjemná, jednalo se o zajímavé téma, které má smysl, a to dokáže člověka poměrně dost pohánět ve směru poznání a vzdělávání, tak aby si člověk doplnil mezery ve svém vzdělání a danou práci dokázal realizovat. Na tomto tématu je stále ještě co dělat. Uživatelské rozhraní, odpočítávání času, silnější a lepší komunikační rozhraní, nebo náhrada jednotky Arduino za jinou stabilnější a výkonnější, která je vhodnější pro průmyslové využití. Samozřejmě je cílem pro praktickou realizaci udržet rozumnou cenu, tak aby se to stalo lukrativním pro stavební firmy. Nepřiplatí si výrazně větší peníze proti dosavadním systémům, ale při tom, aby ohlas na ně se napříč společnostmi výrazně zvýšil. Doufám, že pro mne je tato práce jen začátkem a v budoucím životě se stejnou tematikou budu i nadále zabývat a povede se mi danou myšlenku uvést do praxe. Největší výzvou bude pravděpodobně nalézt investora, který by byl ochotný uvést výrobek na trh a do masové výroby. Projekt bych zhodnotil jako smysluplný, do praxe realizovatelný a práce samotná mne jako studenta střední školy posunula ve zkušenostech a schopnostech dál, tzn. po stránce programátorské a realizační, po stránce osobního projevu, mluveného díky soutěžím, do kterých jsem ji přihlásil, i psaného kvůli dokumentacím. Domnívám se, že na žáka čtvrtého ročníku střední školy jsem se zadání zhostil uspokojivě.

5. ZDROJE A PŘÍLOHY

5.1. Zdroje a citace

Klon Arduino UNO CH340G: Arduino UNO CH340G. *ARDUINO-SHOP.CZ* [online]. Havlíčkův Brod: ECLIPSE s.r.o, 2018 [cit. 2018-03-10]. Dostupné z: <https://arduino-shop.cz/arduino/1258-klon-arduino-uno-r3-atmega328p-ch340g-usb-typ-b-kabel-1459967190.html>

Ultrazvukový měřič vzdáleností HC-SR04: Měřič vzdálenosti ultrazvukový pro Arduino Raspberry. *ARDUINO-SHOP.CZ* [online]. Havlíčkův Brod: ECLIPSE s.r.o, 2018 [cit. 2018-03-11]. Dostupné z: <https://arduino-shop.cz/arduino/846-arduino-meric-vzdalenosti-ultrazvukovy-1500636000.html>

433 MHz vysílač + přijímač. *ARDUINO-SHOP.CZ* [online]. Havlíčkův Brod: ECLIPSE s.r.o, 2018 [cit. 2018-03-11]. Dostupné z: <https://arduino-shop.cz/arduino/1003-arduino-433mhz-vysilac-prijimac-1427821401.html>

VirtualWire Library. *PJRC Electronic Projects Components Available Worldwide* [online]. Sherwood, Oregon, USA: PJRC.COM, LLC., 2018 [cit. 2018-03-11]. Dostupné z: https://www.pjrc.com/teensy/td_libs_VirtualWire.html

JELÍNEK, Radek. *Mikroprocesorová technika: výklad a zápisky z hodin*. SPŠE a VOŠ Pardubice, 2016-17.

MACHAČOVÁ, Kateřina. *Technické kreslení: výklad a zápisky z hodin*. SPŠE a VOŠ Pardubice, 2016-17.

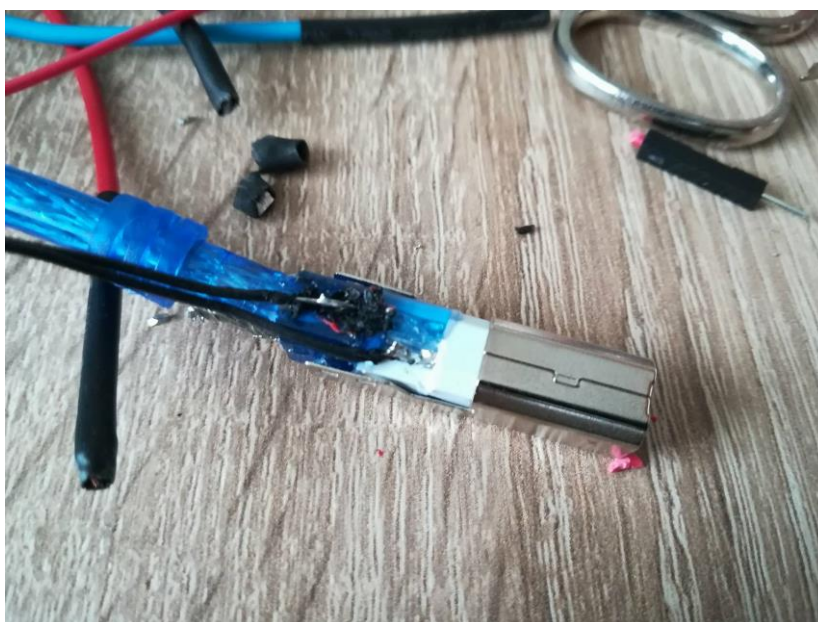
Světelné signalizační zařízení. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2018-03-14]. Dostupné z: https://cs.wikipedia.org/wiki/Světelné_signalizační_zařízení

5.2. Přílohy

obrázek č. 9 – USB zdroj společnosti EMOS



obrázek č. 10 – řešení přerušení napájecího vodiče v USB - B



Poté došlo k očištění, zaizolování a zpětnému zakrytí původním kovovým krytem.

obrázek č. 11 – kryt na řídicí jednotku



obrázek č. 12 – kryt na indikační jednotku



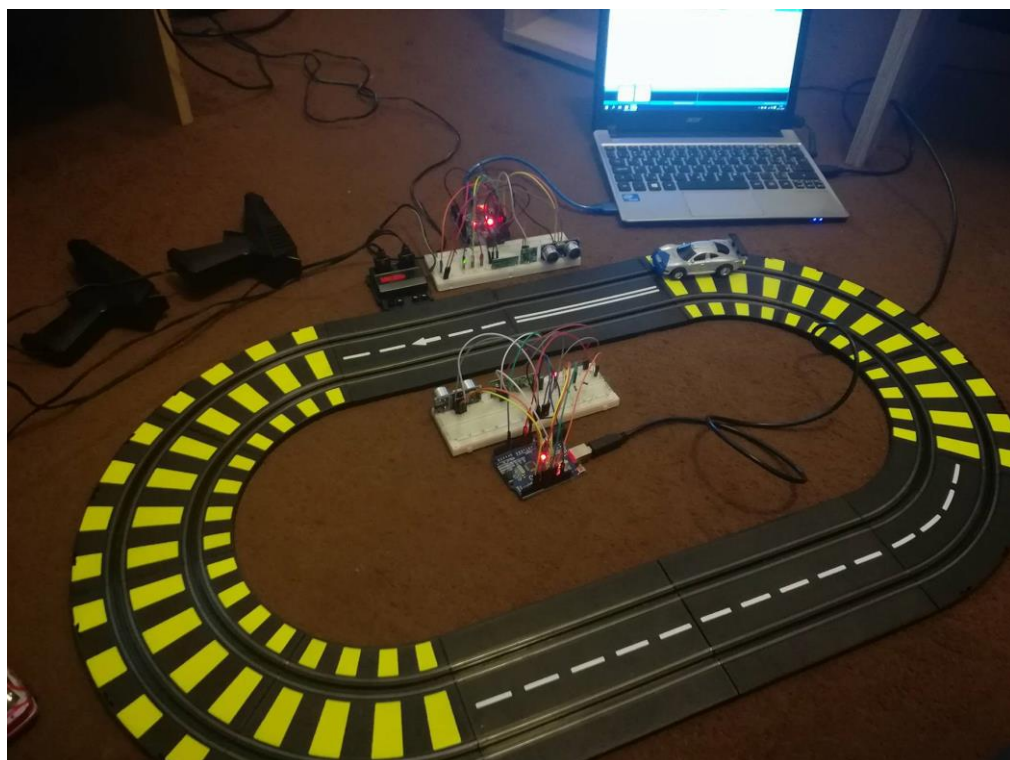
obrázek č. 13 – nožičky pod model



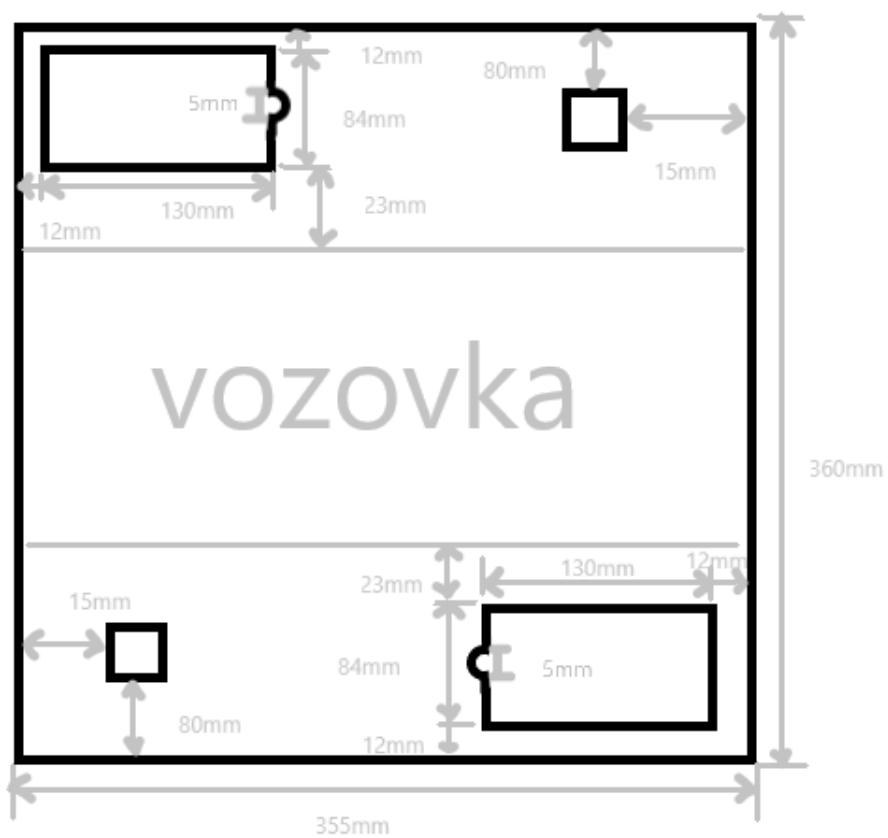
obrázek č. 14 – ukázka vyřezané 12 mm OSB desky



obrázek č. 15 – ukázka prvního testování výrobku



obrázek č. 16 – rozměření OSB desky



Ukázka programu jednotky Slave

```
//program pro slave jednotku
#include <VirtualWire.h> //začlenění knihovny pro bezdrátovou komunikaci
#define bila 11 //přiřazení vývodů a označení pro udržení přehlednosti kódu
#define cervena 7
#define zluta 8
#define zelena 9
#define pTrig 4 //vývody k vysílači a přijímači ultrazvukového senzoru
#define pEcho 5

//hodnoty do kterých se uloží hodnoty vzálenostního senzoru
long odezva;
long vzdalenost;

int TX_PIN = 10; //nastavení PINu pro odesílání dat
int TX_ID = 3; // ID adresa pro odesílání
int RX_PIN = A0; // nastavení PINu pro přijímání dat
int RX_ID = 3; // ID adresa pro přijímání

typedef struct roverRemoteData // datová komunikační struktura proměnných
{
int TX_ID;
int Sensor1Data;
};

int doba1 = 10000; //minimální délka cyklu (v ms)
int doba2 = 35000; // maximální délka cyklu (v ms)
int doba3 = 6000; // doba pro přechody (čas pro bezpečné projetí auta úsekem) (v ms)
int d = 10; //vzdálenost od senzoru do dvou třetin pruhu (v cm)
int a = 3; //množství aut který zmení čas cyklu
int b = 10; //množství aut které výrazně zmení čas cyklu (prodloužení o dvě konstanty)
int p = 0; //počet aut, která projela v protisměru
int c = 0; //počet aut z minulého cyklu
int cas = 20000; //čas cyklu (v ms)
int x = 3000; //konstanta rozdílu času (v ms)
int zaklcas = 20000; //neměnný základní časový úsek cyklu (v ms)
unsigned long cas1; //časy pro millis
unsigned long cas2;
int m = 0; // proměnná pro opakované odeslání hodnoty projetých aut

void setup() {
Serial.begin(9600); //rychlost komunikace
pinMode(bila, OUTPUT); //nastavení vstupů a výstupů jednotlivých PINů
pinMode(cervena, OUTPUT);
pinMode(zelena, OUTPUT);
```

```

pinMode(zluta, OUTPUT);
pinMode(pTrig, OUTPUT);
pinMode(pEcho, INPUT);

vw_setup(2000); //nastavení vysílače
vw_set_tx_pin(TX_PIN);

vw_set_rx_pin(RX_PIN); //nastavení přijímače
vw_rx_start();

digitalWrite(cervena, HIGH); //nastavení výchozího stavu semaforu, tak aby se zabránilo
kolizi
digitalWrite(zluta, LOW);
digitalWrite(zelena, LOW);

digitalWrite (bila, LOW); // nastavení indikační komunikační diody
}

void loop() {
//fáze 1 - červená (čekání na příchozí data z protisměru)
digitalWrite (zluta, LOW);
digitalWrite (cervena, HIGH);
p=5000; // nastavení aut v protisměru na 5000, tak aby každá libovolná reálná hodnota
mohla být zaznamenána
while (p==5000){ // čekání na příchozí data a nastavení proměnných a kódování přenosu
    struct roverRemoteData receivedData;
    uint8_t rcvdSize = sizeof(receivedData); //změření velikosti příchozích dat
    vw_rx_start(); // start příjmu dat

    if (vw_get_message((uint8_t *)&receivedData, &rcvdSize)) // kontrola, zda-li jsou
přístupná nějaká data
    {
        if (receivedData.TX_ID == RX_ID) //kontrola, zda-li odpovídá ID přenosu
        {
            digitalWrite( bila, HIGH); //rozsvícení komunikační diody pro indikaci přenosu
            p = receivedData.Sensor1Data; //příjem dat
        }
        delay (100);
        digitalWrite(bila, LOW);
        if ( p < 5000) {
            vw_rx_stop();
        }
    }
}
//fáze 2 - červená

```

```

digitalWrite (cervena, HIGH);
delay (doba3);
if (cas<doba1) { cas = doba1 + x;}
if (cas>doba2) { cas = doba2 - x;}
//fáze 3 - oranžová (porovnání hodnot projetých aut, nadefinování délky cyklu)
digitalWrite (zluta, HIGH);
if ((p+b)<c) { cas = cas + (2*x); }
else if ((p+a)<c) { cas = cas + x;}
else if (p>(c+a)) { cas = cas - x;}
else if (p<3 && c<3) { cas = zaklcas;}
else { cas = cas;}
delay (3000);
//fáze 4 - zelená (počítání projetých aut po danou dobu)
c=0; //vynulování množství projetých aut cyklu v paměti
digitalWrite (cervena, LOW);
digitalWrite (zluta, LOW);
digitalWrite (zelená, HIGH);
cas2 = millis(); //cyklus pro počítání aut vypočítaný časový úsek
while (cas1 < (cas2 + cas)) {
  cas1 = millis();
  //vypuštění zvukového pulzu
  digitalWrite(pTrig, LOW);
  delayMicroseconds(2);
  digitalWrite(pTrig, HIGH);
  delayMicroseconds(5);
  digitalWrite(pTrig, LOW);
  //změření odezvy a přepočet (za pomocí konstanty) na vzdálenost (v cm)
  odezva = pulseIn(pEcho, HIGH);
  vzdalenost = odezva / 58.31;
  if (vzdalenost < d){ //pokud by byla naměřena dostatečně krátká vzdálenost, pak se přičte
  1 do množství projetých aut
    c=c+1;
    while (vzdalenost < d) { // vyčkání dokud auto celé nepojede, aby se nestalo opakované
    zaznamenání stejného auta
      digitalWrite(pTrig, LOW);
      delayMicroseconds(2);
      digitalWrite(pTrig, HIGH);
      delayMicroseconds(5);
      digitalWrite(pTrig, LOW);
      odezva = pulseIn(pEcho, HIGH);
      vzdalenost = odezva / 58.31;
    }
  }
}
//fáze 5 - oranžová (odeslání informace o počtu projetých aut)
digitalWrite (zluta, HIGH);

```

```

digitalWrite (zelena, LOW);
while (m<50){
  struct roverRemoteData payload;
  payload.TX_ID = TX_ID;
  payload.Sensor1Data = c;
  vw_send((uint8_t *)&payload, sizeof(payload)); // odešli data
  vw_wait_tx();// vyčkej na odeslání všech dat
  m=m+1;
}
delay (2500);
m=0;
}

```

5.3. Seznam obrázků

obrázek č. 1 – návrh semaforu
 obrázek č. 2 – Arduino UNO CH340G
 obrázek č. 3 – ultrazvukový senzor
 obrázek č. 4 – přijímač a vysílač 433 MHz
 obrázek č. 5 – diody 5 mm a odpor 3k3 ohmu
 obrázek č. 6 – fázové rozvržení běhu programu
 obrázek č. 7 – rozvržení krytů na jednotky
 obrázek č. 8 – výsledný vzhled modelu
 obrázek č. 9 – USB zdroj společnosti EMOS
 obrázek č. 10 – řešení přerušení napájecího vodiče v USB – B
 obrázek č. 11 – kryt na řídicí jednotku
 obrázek č. 12 – kryt na indikační jednotku
 obrázek č. 13 – nožičky pod model
 obrázek č. 14 – ukázka vyřezané 12 mm OSB desky
 obrázek č. 15 – ukázka prvního testování výrobku
 obrázek č. 16 – rozměření OSB desky

5.4. Přílohy na disku

- 3Dtisk:
 - nozicka.ipt
 - programovaciJednotkaPredniDil.ipt
 - programovaciJednotkaZadnidil.ipt
 - semaforPredniDil.ipt
 - semaforZadniDil.ipt
- infrakomunikační modul.pdf
- ultrazvukový senzor HC – SRO4.pdf
- VirtualWire Library, for very cheap wireless communication.html
- master01.ino
- slave01.ino
- virtualwire_1.27.rar
- vnitrnizapojeni.jpg