



Středoškolská technika 2017

Setkání a prezentace prací středoškolských studentů na ČVUT

ArduMoto

Kaiser Miroslav

SPŠ A VOŠ PÍSEK
Karla Čapka 402, 397 11 Písek

Anotace

Tato práce se zabývá použitím mikrokontroléru Arduino, jakožto ovladače pro různé komponenty motocyklu. V mé práci tento mikrokontrolér nejenže ovládá běžné periferie motocyklu, ale má i mnoho dalších pomocných funkcí, které by v reálném provozu mohly uživateli poskytnout více informací o okolním dění, více komfortu a celkově více možností ovládání elektroniky. Projekt zahrnuje informace o mnoha vstupních senzorech, podle kterých se chová celý program. V praxi by pravděpodobně nebyl používán jako celek, ale využity by byly pouze ty části, které by si uživatel sám navolil.

Vše je usazeno do dřevěné desky, která umožňuje snadnou a přehlednou demonstraci jednotlivých funkcí.

Klíčová slova

Arduino, motocykl, mikrokontrolér, 3D tisk, bluetooth

Annotation

This work deals with using microcontroller Arduino, as the drivers for the various component of the motorcycle. To this work I also added many other auxiliary functions that would provide to the user in a real operation more information about the surrounding events, more convenience and more options overall of electronics control. The project also includes a lot of input sensors by which entire program behaves. The project is developed on a wooden board for easy demonstration. His real use would probably not use a whole but only the parts that the user selected.

Keywords

Arduino, motorcycle, microcontroller, 3D printing, bluetooth

Obsah

Úvodní stránka	1
Základní informace	1
Prohlášení	1
Poděkování	Chyba! Záložka není definována.
Anotace	1
Klíčová slova	2
Annotation	2
Keywords	2
Obsah	Chyba! Záložka není definována.
1 Úvod	4
2 Teoretický rozbor	4
Arduino	4
Mikrokontrolér	4
Arduino software (IDE)	Chyba! Záložka není definována.
Základní použité příkazy	5
Použité knihovny	6
Použité komponenty	7
3 Praktický rozbor	10
Popis fotografie práce	10
Uložení jednotlivých komponentů	10
Display	12
Vstupní periferie	13
Výstupní periferie	15
Napájení	17
Možnosti rozšíření programu v budoucnosti	18
Cenová kalkulace	20
3 Závěr	21
4 Použitá literatura	21
5 Seznam obrázků a tabulek	22
6 Příloha 1: Zdrojový kód programu	24

1 Úvod

Práce se zabývá návrhem funkčního modelu elektroniky pro motocykl, který je ovládán přes mikrokontrolér Arduino, dále jsou přidány další pokročilé funkce, které by uživateli usnadnily ovládání všech komponentů, i motocyklu jako celku, z rozhraní na řídicích, i pomocí mobilní aplikace přes bluetooth. Celý projekt je umístěn na dřevěné desce pro snadné přenášení. Display je uzavřen ve speciálním krytu vytištěném na 3D tiskárně. Projekt jsem si vybral jako nástroj, pomocí kterého jsem se učil s Arduinem od nejjednodušších částí programu až po poměrně pokročilé programy. Taktéž si myslím, že využití Arduina u motocyklu by, vzhledem k jeho opravdu nízké ceně, dostupnosti komponentů a možnostem, které uživateli přináší, bylo skvělé pro reálné využití v provozu i s jeho nastavbovými funkcemi. Pro uživatele se jeví jako hlavní výhoda možnost výběru pouze těch funkcí, které skutečně požaduje.

2 Teoretický rozbor

Arduino

Arduino je nástroj pro tvorbu malých počítačů, které mají vyšší kontrolu a cit pro fyzický svět, než běžné počítače, jak je známe. Arduino je otevřená elektronická platforma, založená na jednoduché počítačové desce a vývojovém prostředí, které slouží k tvorbě software. Pomocí Arduina můžeme vytvářet interaktivní objekty. Arduino deska získává údaje prostřednictvím různých snímačů a senzorů a na základě těchto údajů ovládá příslušné výstupy. Aby Arduino deska vykonávala to, co potřebujeme, musíme pro ni vytvořit program. Na to využijeme programovací jazyk Arduino (založený na jazyce Wiring) a **Arduino software** (IDE), založené na prostředí Processing. Více k tématu programování Arduina na oficiálních stránkách (<https://www.arduino.cc/en/Guide/Introduction>).

Mikrokontrolér

Mikrokontrolér je integrovaný obvod, který v sobě zahrnuje zpravidla vše potřebné k tomu, aby mohl obsáhnout celou aplikaci, aniž by potřeboval další podpůrné obvody. Především jde o paměť pro uložení programu (FLASH, EEPROM nebo ROM) a operační paměť RAM pro uložení aplikačních proměnných a zásobníku. Většina jednočipových počítačů také obsahuje rozsáhlou sadu podpůrných obvodů. Typicky jde o bloky pro logické a analogové vstupy/výstupy, pro komunikační linky, pro rozdělení strojového času a další aplikační logiku. Více k tématu mikrokontrolérů v odborných textech (MATOUŠEK David, odborný text 2002).

Arduino software (IDE)

Arduino IDE je software, ve kterém je program vytvářen a následně jím nahrán do mikrokontroléru. Aplikace kontroluje chyby v programu, je možné s ní sledovat dění

programu a ovládat jej. Používá se jazyk Arduino vycházející za jazyka C++. Do programu lze snadno přidávat knihovny, které usnadňují práci s různými rozšiřujícími moduly.

Základní použité příkazy

- `#include <>` - Do závorek jsou zapsány použité knihovny.
- `void setup(){}` - Do složených závorek je zapsána část programu, která se provede pouze jednou při spuštění Arduina. Slouží především k nastavení vstupních či výstupních pinů nebo k zahájení sériové komunikace.
- `void loop(){}` - Do složených závorek je zapsána část programu, která se opakuje stále dokola. V této části by jsme prováděli měření a ovládání výstupů.
- `const int` nebo `int` - Vytvoření veličiny nesoucí zapsanou hodnotu (LOW/HIGH, 1/2/3..).
- `unsigned long` - Zavedení bez znaménkové veličiny nesoucí číselnou hodnotu.
- `pinMode()` - Nastavení stavu pinu jakožto vstupního nebo výstupního.
- `Serial.begin()` - Začátek sériové komunikace, do závorek je zapsána rychlost přenosu.
- `Serial.read()` - Čtení hodnoty na sériovém portu.
- `Serial.print()` - Odeslání hodnoty na sériový port.
- `if(){}` - Funkce vyhodnocující pravdu nebo nepravdu v prvních závorkách - v případě pravdy provede příkaz ve složených závorkách, v opačném případě pokračuje program za složenou závorkou. Funkce `if()` může být doplněna funkcí `else{}`, která v sobě uchovává část programu, jež by proběhla v případě, že by byla podmínka ve funkci `if` vyhodnocena jako nepravda.
- `switch(){}` - V prvních závorkách je uchována porovnávaná proměnná. Ve složených závorkách jsou uchovány možnosti (`case`) o určených hodnotách, za kterými zapíšeme program spjatý s tímto případem, každá možnost je ukončena zvoláním "break".
- `digitalWrite()` - Do závorek zapíšeme pin, na který chceme zapisovat hodnotu a hodnotu, kterou chceme na pin zapsat. Lze zapsat hodnotu HIGH(5V) nebo hodnotu LOW(0V).
- `digitalRead()` - Čteme buď kladnou, nebo nulovou hodnotu na zapsaném pinu (formát zapisování jako u předchozí funkce).
- `analogRead()` - Čteme hodnotu od 0V do 5V, která je převedena na číslo o hodnotách od 0 do 1023.
- `tone()` - Vyšle signál o zadané frekvenci na připojený reproduktor. Do závorky za příkazem se nejdříve zapíše výstupní pin a za ním požadovaná frekvence. Zvuk se ukončí funkcí `noTone`.
- `map()` - Převod hodnot na požadovanou velikost v určeném rozsahu.
- `millis()` - Získání aktuální doby běhu programu.

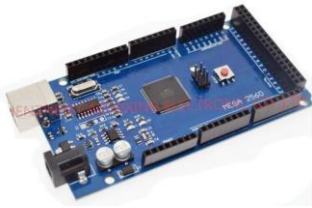
Použité knihovny

Použité knihovny je nutné zapsat na začátku programu před funkcí “void setup()” příkazem “#include <Jméno knihovny.h>”.

UTFT Multi-Platform Universal TFT display library - Knihovna vytvořená Henningem Karlsenem sloužící k jednoduchému ovládní připojeného LCD displeje. Obsahuje dva fonty pro psaní textu a znaků (malý a velký) a jeden ve stylu sedmissegmentového displeje, který však dovoluje zobrazovat pouze čísla. Dále umožňuje vykreslovat samostatně jakýkoliv pixel nebo použít základní geometrické tvary, které jsou dostupné buď s výplní, nebo pouze jako obrys. Knihovna umožňuje vybrat barvu buď z přednastavených barev, nebo lze ručně nastavit poměrem červené, modré a zelené. Příkazy definované v knihovně začínají jako “myGLCD.”, za tento text se zapíše požadovaný obrazec nebo příkaz. Knihovna je volně dostupná na: <http://www.rinkydinkelectronics.com/library.php?id=51> a je použitelná se širokou škálou displejů.

Wire Library - Knihovna povolující komunikaci s I2C / TWI zařízeními. Využívá pinů SDA (datový tok) a SCL (časový tok). Knihovna je zde zapotřebí kvůli modulu reálného času. Umožňuje číst i zapisovat přesný čas, datum a den v týdnu a číst jej. Příkazy, které využívají knihovnu začínají “Wire.”. Knihovnu není nutné nikde stahovat, byla již přidána vývojáři do Aplikace Arduino IDE.

Použité komponenty



Arduino Mega2560 - Řídící jednotka celého projektu. Jedná se o jednu z pokročilejších desek z rodiny Arduina. Má k dispozici 54 digitálních pinů (z nichž 15 umožňuje pulsní šířkovou modulaci), 16 analogových vstupů a 4 sériové porty. V našem případě je použita neoriginální verze Arduina kvůli její nízké ceně. Pro neoriginální verzi je nutné stáhnout speciální ovladač CH340.

Obr. 1: Arduino Mega2560



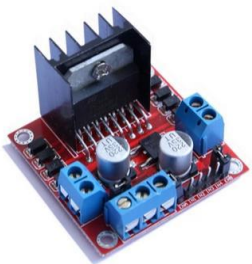
8-kanálový reléový modul - Jelikož má Arduino omezený výstupní proud i napětí, musíme použít spínací součástku, v našem případě je to tento reléový modul s osmi výstupy. Tranzistor spínající relé je napájen ze step-down convertoru nastaveného na 4V. Relé umožňuje průchod proudu až 10A a přepínání mezi dvěma polohami.

Obr. 2: Reléový modul



LCD display - Použitý 3,2 palcový display s rozlišením 320x480 pixelů. Display je dimenzován přímo na zasazení do Arduina Mega2560. V našem případě je však připojen k Arduinu na piny 22 až 53 pomocí kabelu z 32 bitové sběrnice. K programování displeje byla použita UTFT knihovna od Henninga Karlsena. Display má možnost připojení SD karty a nahrání obrázků přímo na ni.

Obr. 3: Lcd display



L298n shield - Rozšiřující modul sloužící k obousměrnému ovládní dvou motorů, v našem případě však ovládáme zadní brzdové světlo. Není použito relé vzhledem k jeho hlučnosti a rychlému opotřebení při vysoké frekvenci přepínání. Modul slouží také jako regulátor napětí pro napájení Arduina a všech modulů mimo reléového.

Obr. 4: L298n shield



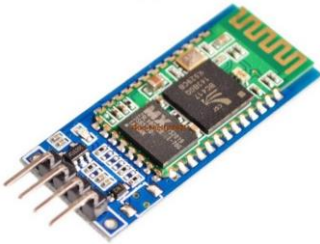
LM2596 - Modul určený ke snížení stejnosměrného napětí v rozsahu 4V až 40V na 1.3V až 37V. V našem případě máme nastavené při vstupním napětí 12V výstupní napětí 4V. Step down konvertor slouží k napájení cívek relé. Modul umožňuje napájení výstupu s proudem až 2A.

Obr. 5: LM2596



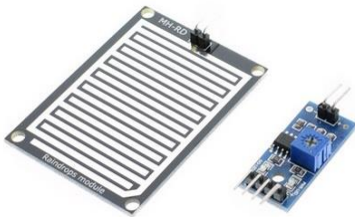
DS3231 - Modul měření reálného času. Modul je nezávislý na napájení z desky díky baterii CR2032 umístěné na desce. Je připojen na piny SCL a SDA a kromě aktuálního času je schopen zobrazit datum, rok, měsíc a den v týdnu. Pro jeho funkci je využívána knihovna "wire.h". Kód využívaný v programu pochází ze serveru Tronixstuff, který v češtině osvětlil Antonín Handl pro server <https://arduino.cz/>.

Obr. 6: DS3231



HC-06 - Modul umožňující sériovou komunikaci přes bluetooth. V našem případě je využíván prostřednictvím mobilního telefonu se systémem Android a aplikací Bluetooth Terminal. Modul dále umožňuje indikaci připojeného zařízení a změnu názvu zařízení nebo hesla. Modul je připojen na piny sériové komunikace. Rx pin Arduina je připojen na Tx pin modulu a Tx Arduina je připojen na Rx modulu.

Obr. 7: HC06



Srážkový senzor - Modul doplněný destičkou měnící svůj odpor v závislosti na množství vody, které se na ni dostane. Senzor je vybaven LM393 komparátorem, který převede naměřené hodnoty na signál od 0V do +5V a vyšle jej výstupem A0 do připojeného Arduina. Arduino tento signál přečte jako kladné číslo o hodnotě 0 až 1023 a dále s tímto číslem pracuje - v našem případě je převedeno na 3 stavy: "silný déšť", "déšť" nebo "žádné srážky".

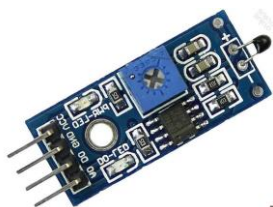
Obr. 8: Srážkový senzor Modul je napájen napětím 5V.



Senzor světla - Modul s fotorezistorem reagujícím na světlo. Senzor je opět doplněn komparátorem LM393. Má pouze jeden digitální výstup, který vysílá buď +5V nebo 0V, v závislosti na nastavené citlivosti na světlo. Citlivost senzoru lze měnit trimrem umístěným na

desce.

Obr. 9: Senzor světla



Teplotní senzor - Modul s thermistorem reagujícím na okolní teplotu, kterou převádí na analogový signál pomocí LM393 konvertoru. Dále má také k dispozici jeden digitální výstup, ze kterého vystupuje buď +5V nebo 0V. Citlivost tohoto výstupu je nastavitelná trimrem na desce.

Obr. 10: Teplotní senzor



Výhřev řídítek - Pro výhřev řídítek jsou použity topné ohebné podložky, které by byly umístěny pod rukojeti na řídítkách (v mém simulovaném provedení nejsou kryty rukojetí). Byly by napájeny přímo napětím z baterie a kvůli jejich vysokému proudovému odběru spínány relátkem.

Obr. 11: Výhřev řídítek



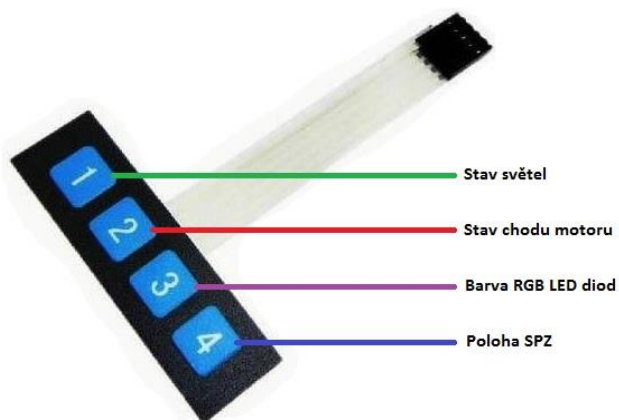
Zadní led světlo - Využito jednoduché ohebné zadní světlo osazené čtyřiceti osmi LED diodami. Je opět napájeno přímo baterií. Blinkry a koncové světlo jsou spínány relátkem, brzdové světlo je spínáno motorovým shieldem L298N z důvodu vyšší použité frekvence spínání.

Obr. 12: Zadní světlo

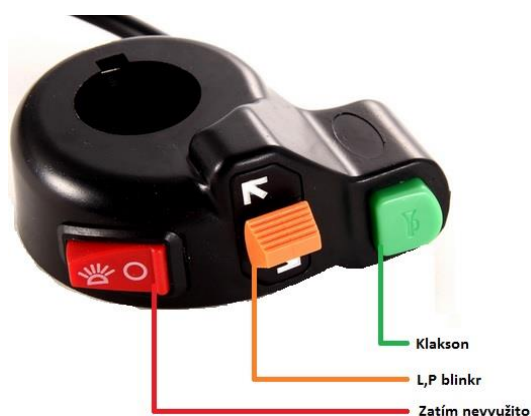
Membránová klávesnice - Jednoduchá membránová klávesnice se čtyřmi tlačítky.

Klávesnice má čtyři výstupní kontakty a jeden společný pro všechna tlačítka. Klávesnice je ohebná a proto ji lze snadno připevnit kamkoliv je třeba. Tlačítka ovládají v našem zapojení světla (1), stav chodu motoru (2), barvu RGB LED diod a displeje (3), Polohu SPZ (4).

Motocyklový přepínač - Ovladač na řídítka obsahující jeden vypínač, jedno tlačítko a přepínač blinkrů se třemi stavy (zapnutý levý blinkr, zapnutý pravý blinkr, oba vypnuté).



Obr. 13: Membránová klávesnice

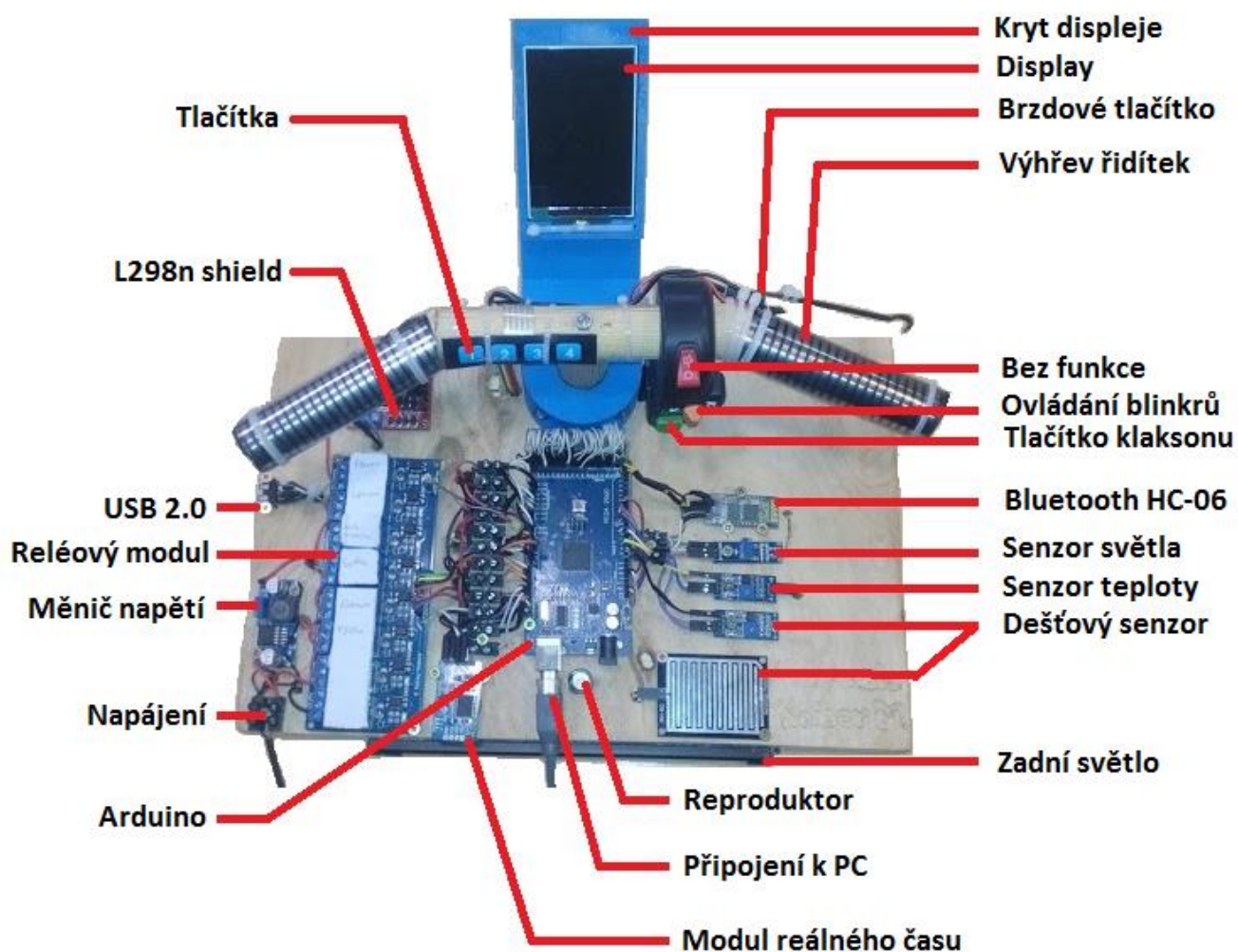


Obr. 14: Motocyklový přepínač

Ostatní komponenty - Dále byly použity v zapojení rezistory o hodnotách 10K a 4K7, RGB led diody, sluchátkový reproduktor transd.50ohm, svorkovnice, koncové tlačítko a propojovací kabely.

3 Praktický rozbor

Popis fotografie práce

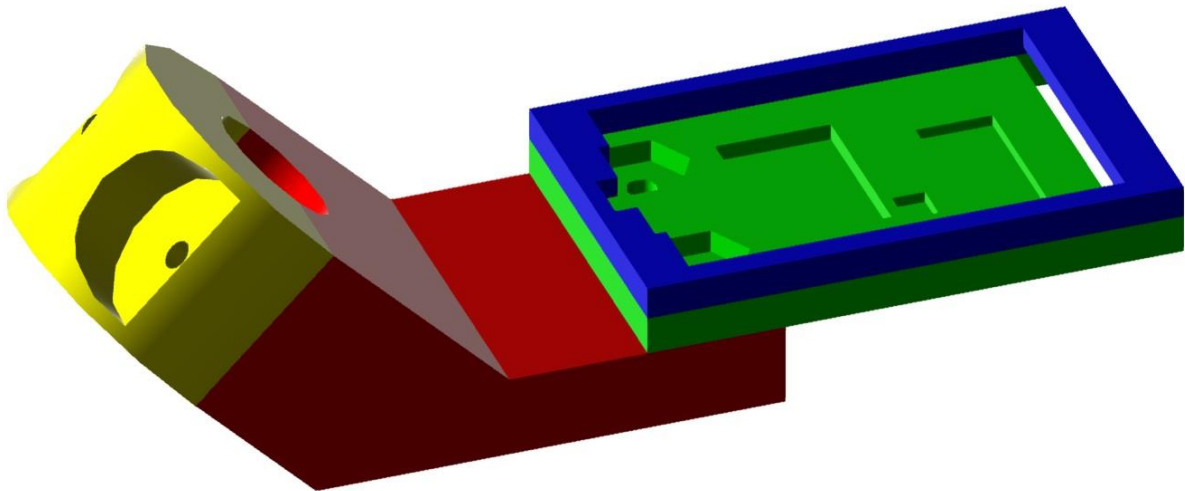


Obr. 15: Popis projektu

Uložení jednotlivých komponentů

Kryt na display - Kryt byl navržen v programu AutoCAD 2017 a vytištěn na 3D tiskárně Prusa i3 z ABS materiálu. Kryt je spolu s úchytem rozdělen na čtyři části, které jsou spojeny

šrouby nebo stahovacími páskami. Díky vyššímu množství dílů je možné nahradit různé části krytu a upravit systém uchopení.



Obr. 16: Kryt na display

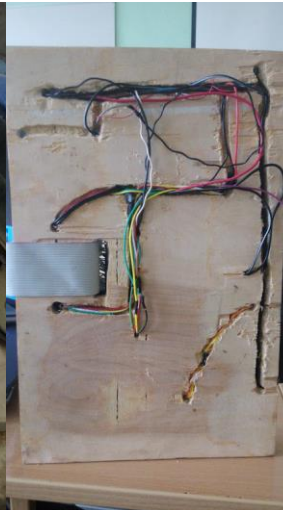
Podložka - Dřevěná deska, na niž jsou upevněny všechny části projektu, byla zhotovena z prkna, které bylo nejprve obroušeno, aby bylo dosaženo hladkého povrchu. Do jeho plochy jsou v pravém dolním rohu vyryty základní informace o autorovi a název práce. Deska je provrtaná a skrze ni jsou vedeny kabely propojující jednotlivé senzory a výstupy. Dále jsou na desku shora připevněna dřevěná říditka, kolem nichž je upnut kryt displeje.



Obr. 17: Výřez desky



Obr. 18: Obroušení desky



Obr. 19: Vyvrtání desky

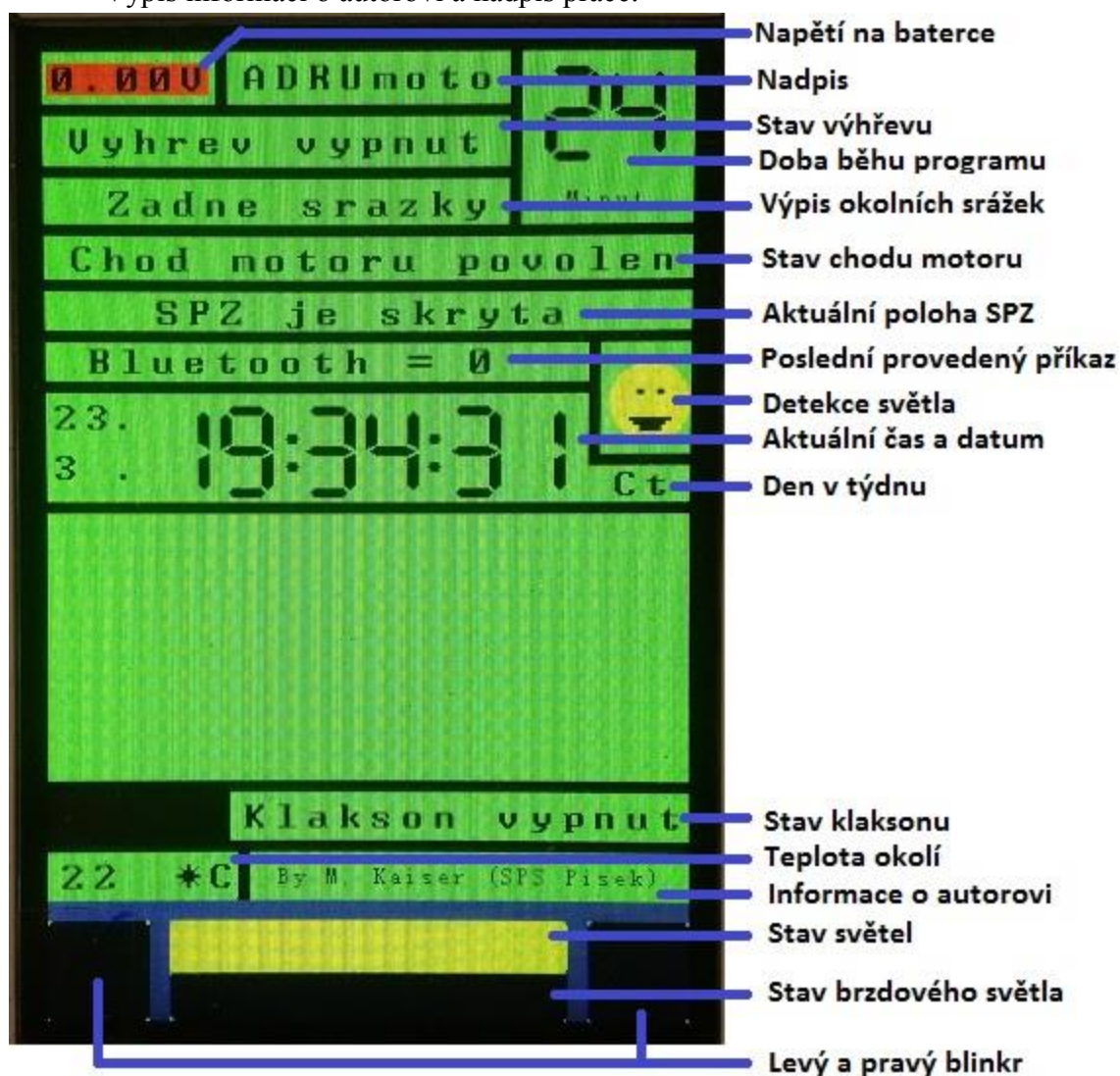


Obr. 20: Spodní díl

Display

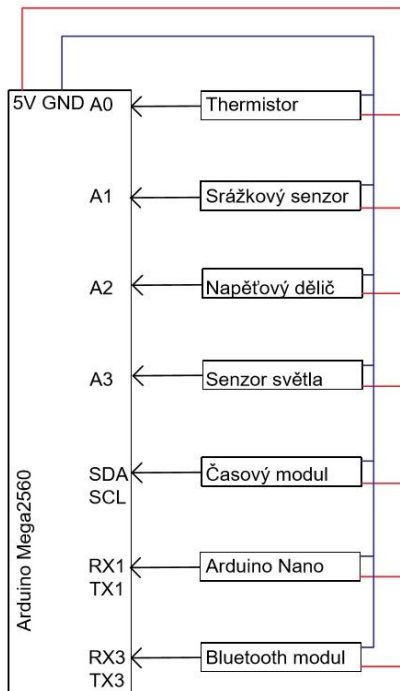
Funkce

- Zobrazení napětí - pokud klesne napětí pod kritickou hodnotu, je zbarveno pozadí za nápisem červeně.
- Zobrazení venkovní teploty, okolních srážek a okolního světla.
- Zobrazení reálného času, dne v týdnu, dne v měsíci, dále zobrazení roku, měsíce a doby chodu programu v minutách.
- Zobrazení posledního odeslaného příkazu přes bluetooth.
- Vypsání stavu polohy SPZ, chodu motoru, výhřevu řídicíků a klaksonu.
- Zabarvení částí displeje v závislosti na stavu světel, blinkrů a klaksonu.
- Zabarvení pozadí displeje podle aktuální barvy RGB led diody.
- Výpis informací o autorovi a nadpis práce.



Obr. 21: Display

Vstupní periferie



Analogové vstupy - Na analogové piny je připojen teplotní modul, srážkový senzor, světelný senzor a dělič napětí měřící napětí na baterii. Hodnoty jsou čteny příkazem `analogRead()` a poté jsou převedeny na námi požadovanou hodnotu.

Sériové komunikace - Sériová komunikace je zahájena příkazem `Serial.begin(rychlost přenosu)`, hodnoty jsou čteny příkazem `Serial.read()`. Hodnoty lze i odesílat a to příkazem `Serial.print(odesílaná data)`.

Ukázka čtení hodnot

hodnot z analogového Pinu:

```
int Hodnota;
const int PinKteryCteme = A0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  Hodnota = analogRead(PinKteryCteme);
  Serial.print(Hodnota);
}
```

Obr. 22: Blokové schéma vstupů

Ukázka odesílání
na sériový port:

```
int Prijathodn;
void setup() {
  Serial.begin(9600);
}
void loop() {
  Prijathodn = Serial.read();
  if Prijathodn == 1 {};
}
```

Obr. 23: Čtení hodnot

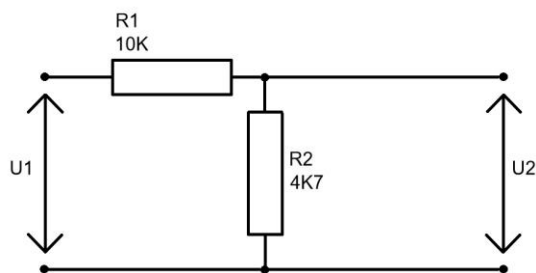
Obr. 24: Sériový přenos

Thermistor - Tento modul je připojen na analogový pin 0. Hodnota přečtená z LM393 převodníku je převedena pomocí vzorce $(1023 - \text{teplota} - 519) / 7$ na požadovanou hodnotu. Ve vzorci označuje "teplota" velikost přečteného signálu. Výsledné hodnoty bude ještě nutno zkalibrovat, momentálně neměří přesně v celém svém rozsahu. Výsledná hodnota je teplota okolí ve stupních Celsia. Velikost této hodnoty je jednou z podmínek pro sepnutí výhřevu a je také vyobrazena v levém horním rohu displeje. Modul je napájen napětím 5V a jeho zem je spojena se zemí Arduina.

Srážkový senzor - Modul je připojen na analogový pin 1. Opět využívá konvertor LM393, do kterého je zapojena destička měnící svůj odpor v závislosti na kapalinách, které na ni dopadnou. Hodnota, která je přečtena Arduinem, je rozdělena příkazem `map()` na tři možnosti, kterým příkaz `switch()` přiřazuje stavy (žádné srážky, dešť a silný dešť). Stav srážek také ovlivňuje výhřev řídiček a je vypisován na displeji. Senzor je napájen napětím 5V a jeho zem je spojena se zemí Arduina.

Senzor světla - Modul je připojen na analogový pin 3. Pro jeho funkci je použit fotorezistor, opět s komparátorem LM393. Na rozdíl od předchozích senzorů tento dokáže rozlišit pouze dva stavy (světlo/tma). Kritická hodnota, po které modul vyšle signál, je nastavitelná trimrem na komparátoru. Na okolním světle není zatím závislá žádná funkce, v budoucnosti se však počítá s ovládáním dálkových světel podle okolního světla (v tomto případě by však byly použity moduly ve větším počtu, s možností čtení přesné analogové hodnoty). Aktuální stav světla je zobrazen na displeji. K napájení modulu je potřeba napětí 5V a zem modulu musí být spojena se zemí Arduina.

Napěťový dělič - Napěťový dělič je realizován pomocí dvou rezistorů o hodnotách 10K a 4K7. Dokáže přeměnit napětí do 15.6 V na napětí do 5V, což je maximální hodnota, kterou dokáže arduino přečíst. Přečtená hodnota je převedena na reálné napětí pomocí vzorce



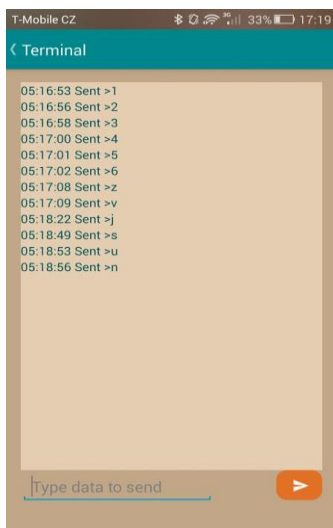
“(napeti*0.01528662)”, kde “napeti” označuje přečtenou hodnotu. Vstupní svorky děliče jsou
 Obr.25: Napěťový dělič
 připojeny na svorky baterie a výstupní na zem Arduino a na 2. analogový pin. Napětí je vypsáno v levém horním rohu obrazovky a v případě, že klesne pod nastavenou kritickou hodnotu (nyní 10V), zabarví se pozadí za hodnotou do červené

barvy.

Časový modul - Aktuální čas je čten z modulu DS3231, který je připojen na piny SDA a SCL. Modul je napájen napětím 5V a v případě odpojení je soběstačný díky baterii CR2032 umístěné přímo na modulu. Programy pro nastavení času a jeho čtení jsou převzaty ze serveru <https://arduino.cz> od Antonína Handla. Modul kromě času dokáže vypsát i datum, den v týdnu a aktuální rok. Všechny tyto hodnoty jsou vypsány na display.

Arduino Nano - Na desce je připraven USB 2.0 port (samice), do kterého by bylo připojováno druhé Arduino ovládající polohu SPZ a odesílající aktuální data do Arduina Mega2560. USB port je však využitelný i pro napájení/nabíjení libovolného zařízení. Piny konektoru D+ a D- jsou připojeny na RX1 a TX1 piny Arduina. Pokud je požadováno vyklopení SPZ, je vyslán z Arduina Mega2560 signál “1”, v opačném případě je odeslán příkaz “0”. Hodnoty, kterými bude Arduino Nano odpovídat, zatím nejsou stanoveny. Aktuální poloha SPZ je zobrazena na displeji.

Bluetooth komunikace - Ke komunikaci přes bluetooth slouží modul HC-06 připojený na RX3 a TX3 Arduina. Modul je opět napájen napětím 5V. K ovládání slouží mobilní telefon s podporou bluetooth a aplikací Bluetooth Terminal, nebo jinou aplikací podporující odesílání příkazů. Zařízení lze vyhledat pod jménem HC-06 a při potřebě hesla se zadá výchozí”1234”.

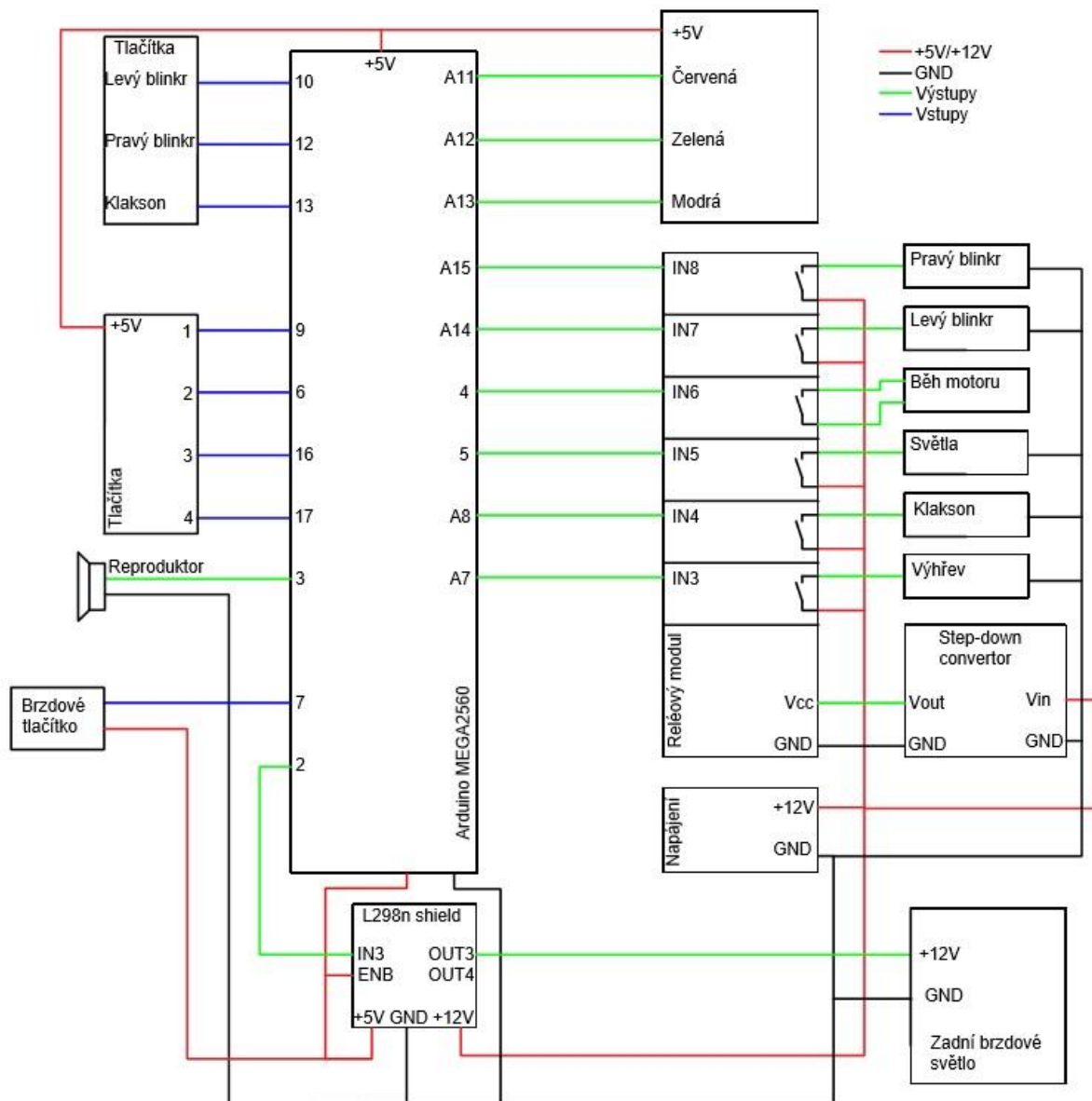


Příkazy:

- “1” - Změní barvu displeje a RGB led na modrou.
- “2” - Změní barvu displeje a RGB led na světle modrou (aqua).
- “3” - Změní barvu displeje a RGB led na zelenou.
- “4” - Změní barvu displeje a RGB led na žlutou.
- “5” - Změní barvu displeje a RGB led na červenou.
- “6” - Změní barvu displeje a RGB led na bílou.
- “z” - Zapne přední a zadní světlo.
- “v” - Vypne přední a zadní světlo.
- “j” - Povolí chod motoru.
- “s” - Zakáže chod motoru.
- “u” - Vysune SPZ.
- “n” - Schová SPZ.

Obr. 26: Rozhraní mobilní aplikace Bluetooth Terminal

Výstupní periferie



Obr. 27: Blokové schéma výstupních periférií

Výstupy - Na výstup jsou přes reléový modul připojena světla, blinkry, klakson, výhřev řídicítek a běh motoru. Přes motorový shield L298n je připojeno zadní brzdové světlo a přímo do Arduina je zapojen reproduktor a RGB led diody. Výstupy jsou ovládány buď automaticky, nebo stisknutím přiřazeného tlačítka. Hodnoty jsou na piny zapisovány příkazem `digitalWrite(pin,hodnota)`.

Tlačítka ovládající výstupy - K ovládní výstupů slouží membránová klávesnice se čtyřmi tlačítky, motocyklový přepínač a koncové tlačítko, zbylé výstupy jsou ovládány automaticky.

Ukázka čtení a zápisu hodnot:

Ukázka programu pro blikání bez

```
const int Tlacitko = 2;
const int LEDdioda = 3;
int StavTlacitka;
void setup() {
  pinMode(Tlacitko, INPUT);
  pinMode(LEDDioda, OUTPUT);
}
void loop() {
  StavTlacitka = digitalRead(Tlacitko);
  digitalWrite (LEDDioda, stavTlacitka);
}
```

funkce delay():

```
const int Tlacitko = 2;
const int LEDdioda = 3;
int StavTlacitka;
int StavDiody;
unsigned long UbehlyCasDiody = 0;
const long DobaBlikaniDiody = 1000;
void setup() {
  pinMode(Tlacitko, INPUT);
  pinMode(LEDDioda, OUTPUT);
}
void loop() {
  StavTlacitka = digitalRead(Tlacitko);
  if (StavTlacitka == HIGH) {
    if (millis() - UbehlyCasDiody > DobaBlikaniDiody) {
      UbehlyCasDiody = millis();
      if (StavDiody == LOW)
        StavDiody = HIGH;
      else
        StavDiody = LOW; }
    } else {StavDiody = LOW;}
  digitalWrite(LEDDioda, StavDiody);
}
```

Obr. 28: Ukázka čtení a zápisu hodnot

Obr. 29: Ukázka programu pro blikání

Blinky - Pravý blinkr je připojen na reléový modul (IN8) a je spínán signálem z pinu A14. Blinkr se rozblíká po přepnutí přepínače blinkrů do pravé polohy, ten je připojen na 12. pin. Levý blinkr je připojen na reléový modul (IN7) a je spínán signálem z pinu A15. Blinkr se rozblíká po přepnutí přepínače blinkrů do levé polohy, ten je připojen na 10. pin. Blinky jsou umístěny na led pásku spolu se zadním světlem. Stav blinkrů je vyobrazen na displeji v jeho dolní části. Nyní je doba sepnutí blinkrů nastavena na jednu sekundu. Doba sepnutí blinkrů lze měnit v kódu přepsáním hodnoty “const long DobaBlikaniBlinkru = 1000;”

Světla - Přední světlo zatím není připojeno, v budoucnu bude spojeno paralelně se zadním světlem. To je zatím připojené na reléový modul (IN5) a spínáno pinem 5. Tlačítkem “1” na membránové klávesnici, které je připojeno na pin 9, měníme stav světla. Světlo lze také ovládat přes bluetooth příkazem “z” pro zapnutí a “v” pro vypnutí. Stav světla je zobrazován v dolní části displeje.

Běh motoru - Relé běhu motoru (IN6) je připojeno na 4. pin Arduina. Zatím je připraveno pro simulaci pouze relé, které by rozpínalo napájení svíčky v motoru. Stav běhu motoru je přepínán tlačítkem “2” připojeným na 6. pin Arduina. Běh motoru lze také přepínat

bezdrátově přes bluetooth příkazem “j” pro jeho povolení a “s” pro vypnutí motoru. Aktuální stav je vypisován na displeji.

Klakson - Klakson by byl připojen na relé (IN4), které je ovládáno pinem A8. Klakson se sepne po stisknutí zeleného tlačítka připojeného na pin 13. Stav klaksonu je opět vypisován na displeji.

Výhřev řídicítek - Řídítka jsou vyhřívána topnými ploškami ovládanými reléovým modulem (IN3), který je spínán pinem A7. Tato funkce je plně samostatná a závisí na teplotě okolí nebo na okolních srážkách. Teplota okolí musí klesnout pod 15°C. Hodnotu lze měnit změnou “int spintempl = 15;” v programu. Pokud je teplota nad 15° a senzor zaznamená dešťové srážky, je výhřev také sepnut. Podmínky sepnutí lze jednoduše změnit dle uživatelských požadavků. Stav výhřevu je zobrazen na displeji.

Reproduktor - Reproduktor je připojen na pin 3 a je spínán signály o nastavené frekvenci. Zatím je reproduktor spínán pouze při stisknutí některého z tlačítek na membránové klávesnici signálem o frekvenci 8000Hz. Hlasitost reproduktoru lze upravit připojením rezistoru do série s ním.

Zadní brzdové světlo - Zadní brzdové světlo pro lepší viditelnost bliká a je spínáno přes motorový modul L298n. U relé by totiž docházelo k opotřebení a nadměrnému hluku. Frekvence spínání je nastavitelná změnou hodnoty “DobaBlikaniZadníhoSvetla = 400”. Zadní brzdové světlo je umístěno na LED pásku společně se zadním světlem a blinkry. Je ovládáno signálem z pinu 2. Zadní brzdové světlo začne blikat po stisknutí koncového tlačítka umístěného na řídicích a připojeného na pin 7.

RGB led - Použité diody mají společnou anodu, katody jsou připojeny na piny A11(červená), A12(zelená), A13(modrá). V Arduinu jsou nastaveny barvy: modrá, světle modrá, zelená, žlutá, červená a bílá. Barvy se mění stiskem tlačítka “3” připojeným na pin 16. Dále lze měnit aktuální barvu bezdrátově přes bluetooth příkazy “1”, “2”, “3”, “4”, “5” a “6”. Aktuální barva je zobrazena na displeji jako barva pozadí.

Napájení

Celá deska je napájena ze svorkovnice v levém dolním rohu, kam by se připojila libovolná motocyklová baterie. Pro prezentaci je však použit zdroj stejnosměrného napětí poskytující napětí 12V. Doporučené napětí pro správnou funkci programu je 9V až 16V. Přivedené napětí je přeměněno motorovým modulem L298n, který má integrovaný stabilizátor na 5V. Tento modul napájí veškeré moduly na desce kromě cívek reléového modulu. Cívky reléového modulu jsou napájeny kvůli vyššímu odběru proudu step down konvertorem nastaveným na 4V při 12V na vstupních svorkách.

Možnosti rozšíření programu v budoucnosti

Do budoucna se počítá s rozšířením možností této desky i s použitím v praxi na motocyklu.

Možnosti rozšíření desky

- Rozdělení řídicí jednotky (Arduina MEGA2560) na dvě desky, které by spolu komunikovaly. K rozdělení by došlo především kvůli LCD displeji, který zpomaluje chod programu.
- Ovládání polohy SPZ druhým Arduinem připojeným na již připravený USB port. Poloha by byla měněna krokovým motorem 28BYJ-48, který má dostatečnou sílu a dokáže skvěle držet svou pozici.
- Automatické vypnutí blinkrů po určeném čase.
- Ultrazvukový senzor reagující na překážky - na ty by upozorňoval uživatele vizuálně prostřednictvím displeje, při kritičtějších hodnotách i zvukovými signály.
- Kontrola dálkových světel v závislosti na okolním světle.
- Přidání stopek, které by uživatel ovládal doposud nevyužitým červeným přepínačem.
- Postupné rozsvěcování RGB diod pomocí pulsní šířkové modulace.
- Přidání grafiky, která by se zobrazila při načítání obrazovky.
- Zařazení minimálně jedné pojistky do obvodu.
- Použití dotykového displeje.

Možnosti rozšíření funkcí při použití na motorce

Při praktickém využití na motorce by se počítalo spíše s nějakou starší motorkou, kde ještě není elektronika tak složitá. Pravděpodobně by nebyly využity všechny funkce obsažené v tomto projektu, za to by se však počítalo s funkcemi, které by pouze na desce uskutečnit nešly.

Ideálně by byla přestavba provedena na motorce určené ke stuntridingu (akrobatickému manévrování s motocyklem) - poskytovala by větší ochranu pro jezdce i pro motorku.

- Měření otáček motoru, které by byly vypsané na displeji a vysvíceny na digitálním RGB led pásku (digitální LED pásy umožňují kontrolovat samostatně každou LED diodu).
- Měření rychlosti a to buď pomocí GPS, nebo čtením z náhonu tachometru.
- Zjištění zařazeného stupně.
- Zjištění stavu nádrže.
- Měření teploty motoru.
- Měření maximální rychlosti, akcelerace, průměrné rychlosti, spotřeby, dojezdu...
- Automatické podřazení při kriticky nízkých otáčkách.
- Automatické vypnutí motoru při pádu řidiče z motorky, přidání bezpečnostního tlačítka, které by vypnulo motor (bylo by ho možné “zakázat”, aby nedocházelo k jeho stisknutí nedopatřením).
- Integrovaní RGB led diod do světlometů.
- Senzor náklonu a uchování jím naměřených dat, případně chování programu v závislosti na těchto hodnotách.
- Snímání přítomnosti řidiče.

Cenová kalkulace

Součástka	Cena (Kč)
Arduino MEGA2560,3.2 LCD Display	333,-
Reléový modul	100,-
L298n	60,-
Step down converter	20,-
DS3231	30,-
HC-06	65,-
Srážkový senzor	15,-
Senzor světla	30,-
Teplotní senzor	30,-
Výhřev řídicíků	65,-
Zadní světlo	50,-
Klávesnice	40,-
Motocyklový přepínač	40,-
Ostatní součástky	20,-
Dřevo	0
ABS (materiál pro 3D tiskárnu)	0
Celková cena	898,-

Tab. 1: Cenová kalkulace

3 Závěr

Výsledkem mé práce je dle původního plánu funkční model mikrokontroléru, který ovládá periferie motocyklu. Model dobře ukazuje možnosti tohoto cenově velmi dostupného mikrokontroléru.

Při realizaci projektu jsem využil vědomostí získaných během výuky odborných předmětů a rovněž samostudiem. Díky tomuto projektu jsem pronikl více do světa Arduina i jeho programovacího jazyka. To mi nakonec pomohlo lépe porozumět látce hned v několika předmětech. Podobnou zkušenost doporučuji každému, kdo přemýšlí o nákupu mikrokontroléru. Pro mě je nejvyšším cílem využití výsledků této práce v praxi, na opravdovém motocyklu, Je také mnoho možností jak na tuto práci navázat a v mnohém ji rozšířit i vylepšit.

4 Použitá literatura

1. Co je to Arduino, *Creative Commons Attribution-ShareAlike 3.0*. Český Překlad z oficiálních stránek Arduina převzatý z českého webu: <https://arduino.cz/co-je-to-arduino/> . Původní text: <https://www.arduino.cc/en/Guide/Introduction>.
2. RYBIČKA, Jiří, Petra ČAČKOVÁ a Jan PŘICHYSTAL. Průvodce tvorbou dokumentů. 1. vyd. Bučovice: Martin Stříž, 2011. ISBN 978-80-87106-43-3.
3. Soubor návodů pro začátečníky v programování dostupný na oficiálních stránkách Arduina: <https://www.arduino.cc/en/Tutorial/BuiltInExamples>
4. UTFT Manual, Henning Karlsen. Manuál ke knihovně LCD displeje dostupný na: <http://www.rinkydinkelectronics.com/resource/UTFT/UTFT.pdf>
5. Užívání hodin reálného času DS1307 a DS3231 s Arduinem, Antonín Handl. Článek pojednávající o problematice hodin reálného času. Článek je volně dostupný na webu: <https://arduino.cz/tutorial-uzivani-hodin-realneho-casu-ds1307-a-ds3231-s-arduinem/>

5 Seznam obrázků a tabulek

- Obr. 1: Arduino Mega2560. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/Free-shipping-3-2-inch-TFT-LCD-screen-module-Ultra-HD-320X480-for-Arduino-MEGA-2560/32538935176.html?spm=2114.13010608.0.0.sRDzNe>
- Obr. 2: 8-kanálový reléový modul. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/With-optocoupler-8-channel-8-channel-relay-control-panel-PLC-relay-5V-module-for-arduino-hot/747770138.html?spm=2114.13010608.0.0.daPVAg>
- Obr. 3: LCD display. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/Free-shipping-3-2-inch-TFT-LCD-screen-module-Ultra-HD-320X480-for-Arduino-MEGA-2560/32538935176.html?spm=2114.13010608.0.0.sRDzNe>
- Obr. 4: L298n shield. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/New-Dual-H-Bridge-DC-Stepper-Motor-Drive-Controller-Board-Module-L298N-for-arduino/32581976321.html?spm=2114.13010608.0.0.1lowVy>
- Obr. 5: LM2596. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/86070-Free-shipping-DC-DC-Step-Down-Converter-Module-LM2596-DC-4-0-40-to-1/32354635261.html?spm=2114.13010608.0.0.9XTd1j>
- Obr. 6: DS3231. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/DS3231-AT24C32-IIC-High-Precision-RTC-Module-Clock-Timer-Memory-Module-for-Arduino/1859103686.html?spm=2114.13010208.99999999.278.OcBqM8>
- Obr. 7: HC06. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/HC-06-Bluetooth-serial-pass-through-module-wireless-serial-communication-from-machine-Wireless-HC06-for-arduino/1626637671.html?spm=2114.13010608.0.0.wZvxSB>
- Obr. 8: Srážkový senzor. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/1set-lot-Snow-Raindrops-Detection-Sensor-Module-Rain-Weather-Module-Humidity-For-Arduino/32711871234.html?spm=2114.13010608.0.0.bcFIVY>
- Obr. 9: Senzor světla. Obrázek z webu použitého prodejce. Web:<http://aukro.cz/ard2-11-a-senzor-svetla-pro-arduino-s-fotoodporem-i6757414959.html>
- Obr. 10: Teplotní senzor. Obrázek z webu použitého prodejce. Web:<http://aukro.cz/ard8-07-teplotni-senzor-s-termistorem-i6753815233.html>
- Obr. 11: Výhřev řídítek. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/Universal-12V-Motorcycle-Electric-Heating-Handle-Kit-Refit-Hand-Set-Universal-Electric-Heating-Insert-Handlebar-Heated/32780214162.html?spm=2114.13010208.99999999.263.OcBqM8>
- Obr. 12: Zadní světlo. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/1Pcs-48-Leds-Motorcycle-Brake-Turn-Signal-Light-Strip-License-Plate-Light-Flashing-Tail-Stop-Lights/32741332968.html?spm=2114.13010208.99999999.285.OcBqM8>
- Obr. 13: Membránová klávesnice. Obrázek z webu použitého prodejce. Web:<http://aukro.cz/ard4-73-klavesnice-membranova-4-pozice-i6752061714.html>
- Obr. 14: Motocyklový přepínač. Obrázek z webu použitého prodejce. Web:<https://www.aliexpress.com/item/BEST-12V-7-Pins-Motorcycle-ATV-Bike-Scooter-7-8-Switch-Horn-Turn-Signals-On-Off/32600538830.html?spm=2114.13010608.0.0.L0Dy6Z>
- Obr. 15: Popis projektu. Fotografie upravena v programu malování.
- Obr.16: Kryt na display. Obrázek byl vytvořen na stránce <https://a360.autodesk.com/> z výkresu, který jsem navrhl v programu AutoCAD.
- Obr. 17: Výřez desky. Fotka zobrazující vyřezávání podložky, na které jsou uloženy komponenty.
- Obr. 18: Obroušení desky. Fotka zobrazující broušení podložky.
- Obr. 19: Vyvrtání desky. Fotka zobrazující provrtanou spodní část podložky.
- Obr. 20: Spodní díl. Fotka zobrazující kryt kabelů pod podložkou.
- Obr. 21: Display. Fotka s popisem displeje.

- Obr. 22: Blokové schéma vstupů. Schéma bylo vytvořeno v programu AutoCAD 2017.
- Obr. 23: Čtení Hodnot. Kód zapsaný v softwaru Arduino IDE.
- Obr. 24: Sériový přenos. Kód zapsaný v softwaru Arduino IDE.
- Obr. 25: Napěťový dělič. Obrázek nakreslený v programu AutoCAD a obtažený v malování.
- Obr. 26: Rozhraní mobilní aplikace Bluetooth Terminal. Snímek displeje z mobilního telefonu.
- Obr. 27: Blokové schéma výstupních periférií. Vytvořeno v programu AutoCAD 2017.
- Obr. 28: Ukázka čtení a zápisu hodnot. Kód zapsaný v softwaru Arduino IDE.
- Obr. 29: Ukázka programu pro blikání bez delay(). Kód zapsaný v softwaru Arduino IDE.

Tab. 1: Cenová kalkulace

6 Příloha 1: Zdrojový kód programu

```
//konstanty
#include <UTFT.h>//knihovna LCD displeje
#include <SoftwareSerial.h>//knihovna seriove komunikace
#include "Wire.h"//knihovna casu

extern uint8_t BigFont[];
extern uint8_t SmallFont[];
extern uint8_t SevenSegNumFont[];
UTFT myGLCD(CTE32HR,38,39,40,41);

const int LevyBlinkr = A14 ;//levy blinkr-vystup
const int PravyBlinkr = A15;//pravy blinkr-vystup
const int LevyBlinkrTlacitko = 10;//levy blinkr-tlacitko
const int PravyBlinkrTlacitko = 12;//pravy blinkr-tlacitko
const int SvetlaPZ = 5;//predni + zadni svetlo-vystup
const int SvetlaPZTlacitko = 9;//predni + zadni svetlo-tlacitko
const int RGBtlacitko = 16;//tlacitko RGB ledek
const int Cervena = A11 ;//cervena barva
const int Modra = A12 ;//modra barva
const int Zelena = A13 ;//zelena barva
const int BehMotoruRele = 4;//rele behu motoru
const int BehMotoruTlacitko = 6;//tlacitko pro zmenu stavu behu motoru
const int KlaksonRele = A8;//klakson - vystup
const int KlaksonTlacitko = 13;//klakson - tlacitko
const int ZadniSvetloVyst = 2;//zadni svetlo - brzdoovy signal - vystup
const int zadnisvetlovyst = A6 ;
const int SignalZadníhoSvetla = 7;//zadni svetlo - brzdoovy signal - sensorove tlacitko!!!!!!!!!!
const int VyhrevRele = A7;//Rele vyhrevu riditek
const int SpzTlacitko = 17;// tlacitko menici polohu spz
const int Reproduktor = 3;// reproduktor

//promenne
int StavLevehoBlinkru = LOW;//stav leveho blinkru
int StavPravehoBlinkru = LOW;//stav praveho blinkru

int StavTlacitkaPravehoBlinkru = 0;//tlacitko leveho blinkru - stav
int StavTlacitkaLevehoBlinkru = 0;//tlacitko praveho blinkru - stav

int StavPZSvetla = LOW;//vystup svetel predni + zadni
int TlacitkoPZsvetlaWasOff = true;//stav tlacitka

int StavCervene = false;
int StavModre = false;
int StavZelene = false;
int RgbTlacitkoWasOff = true;
int KteraBarva = 1;
```



```

int StavChoduMotoru = HIGH;//stav vystupniho relatka
int TlacitkoMotoruWasOff = true;//tlacitko rele ovladani motoru

int StavKlaksonu = LOW;//stav klaksonu
int StavTlacitkaKlaksonu = 0;//tlacitko klaksonu - stav

int StavZadníhoSvetla = LOW;//vystup zadního brzdového svetla
int StavBrzdovéhoTlacitka = 0;//sensor brzd - stav

int StavVyhrevuRiditek = LOW;//Vyhrev riditek - stav
int spintep1 = 15;//kriticka teplota pro sepnuti vyhrevu

int StavTlacitkaPZsvetla;
int StavRGBTlacitka;
int StavTlacitkaChoduMotoru;
int StavSpzTlacitka;

int teplota;
int TeplPin = A0;//pin teplotního cidla
int cel;
float FloatCel;

int HodnotaSrazek;
int VstupSrazek = A1;//pin srazkového cidla
int StavSrazek;

int napeti;
int VstupNapeti = A2;//pin napetového cidla
float nap;

int jas;
int VstupJasu = A3;

int which1;//stav barev displeje
int which2;
int which3;
int which4;
int which5;
int which6;

unsigned long UbehlyCasLevehoBlinkru = 0;//blinkry-ubehly cas L strana
unsigned long UbehlyCasPravehoBlinkru = 0;//blinkry-ubehly cas P strana
unsigned long UbehlyCasZadníhoSvetla = 0;//zadni brzdove svetlo - ubehly cas
unsigned long UbehlyCasAutVypBlinkru = 0;//automaticke vypnuti blinkru
unsigned long CasOdPosledníhoMereni = 0;//Sensory na analogovem portu - cas od posledního mereni

const long DobaBlikaniBlinkru = 1000;//DobaBlikaniBlinkru sepnuti blinkru
const long DobaBlikaniZadníhoSvetla = 400;//DobaBlikaniBlinkru blikani zadního brzdového svetla
const long DobaAutVypBlinkru = 30000;//cas do automatickeho vypnuti blinkru
const long DobaDoDalsihoMereni = 400;//cas do dalsiho mereni

int Min=0;

```

```

int bluetooth = 0;
int SpzTlacitkoWasOff;
int StavSpz;

int kritNap = 10;

const int MinHodnotaSrazek = 0;//minimalni hodnota srazek
const int MaxHodnotaSrazek = 1024;//maximalni hodnota srazek

#define DS3231_I2C_ADDRESS 0x68//cas
byte decToBcd(byte val){
    return( (val/10*16) + (val% 10) );}
byte bcdToDec(byte val){
    return( (val/16*10) + (val% 16) );}

void setup() {

    myGLCD.InitLCD(PORTRAIT);//nastaveni orientace
    myGLCD.setFont(BigFont);//nastaveni fontu

    pinMode (LevyBlinkr,OUTPUT);//L blinkr
    pinMode (PravyBlinkr,OUTPUT);//P blinkr

    pinMode (LevyBlinkrTlacitko,INPUT);//L blinkr spinac
    pinMode (PravyBlinkrTlacitko,INPUT);//P blinkr spinac

    pinMode (SvetlaPZ,OUTPUT);//vystup predniho + zadniho svetla
    pinMode (SvetlaPZTlacitko,INPUT);//tlacitko zmeny stavu prediho + zadniho svetla

    pinMode (RGBtlacitko,INPUT);
    pinMode (Cervena,OUTPUT);
    pinMode (Modra,OUTPUT);
    pinMode (Zelena,OUTPUT);

    pinMode (BehMotoruRele,OUTPUT);//Rele behu motoru
    pinMode (BehMotoruTlacitko,INPUT);//Tlacitko zmeny stavu rele behu motoru
    digitalWrite(BehMotoruRele, HIGH);

    pinMode (KlaksonRele,OUTPUT);//vystup klaksonu
    pinMode (KlaksonTlacitko,INPUT);//tlacitko klaksonu

    pinMode (VyhrevRele,OUTPUT);//vystup vyhrevu riditek

    pinMode(SpzTlacitko, INPUT);

    pinMode(ZadniSvetloVyst, OUTPUT);
    pinMode(zadnisvetlovyst, OUTPUT);

    Serial.begin(9600);//seriova komunikace s PC
    Serial1.begin(9600);//seriova komunikace s druhým arduinem
    Serial3.begin(9600);//komunikace přes bluetooth

```

```

Wire.begin();//cas
}
// program cteni casu
void readDS3231time(byte *second, byte *minute, byte *hour, byte *dayOfWeek, byte *dayOfMonth, byte
*month, byte *year){
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set DS3231 register pointer to 00h
    Wire.endTransmission();
    Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
    // request seven bytes of data from DS3231 starting from register 00h
    *second = bcdToDec(Wire.read() & 0x7f);
    *minute = bcdToDec(Wire.read());
    *hour = bcdToDec(Wire.read() & 0x3f);
    *dayOfWeek = bcdToDec(Wire.read());
    *dayOfMonth = bcdToDec(Wire.read());
    *month = bcdToDec(Wire.read());
    *year = bcdToDec(Wire.read());
} //konec programu cteni casu

void loop() {

if (Serial3.available() > 0)
{
    bluetooth = Serial3.read();
    (bluetooth = bluetooth - 48);
    switch(bluetooth){
    case 1: KteraBarva=1;which1=false;break;
    case 2: KteraBarva=2;which2=false;break;
    case 3: KteraBarva=3;which3=false;break;
    case 4: KteraBarva=4;which4=false;break;
    case 5: KteraBarva=5;which5=false;break;
    case 6: KteraBarva=6;which6=false;break;
    case 70:StavPZSvetla=LOW;break;//vypnuti svetel
    case 74:StavPZSvetla=HIGH;break;//zapnuti svetel
    case 58:StavChoduMotoru=HIGH;break;
    case 67:StavChoduMotoru=LOW;break;
    default;;
    }
} //bluetooth

unsigned long Milisekundy = millis();//cas blinkry - blikani

```

```

StavTlacitkaPravehoBlinkru = digitalRead(LevyBlinkrTlacitko);//leva strana - blinkry
  if (StavTlacitkaPravehoBlinkru == HIGH) {

    if (Milisekundy - UbehlyCasLevehoBlinkru > DobaBlikaniBlinkru) {
      UbehlyCasLevehoBlinkru = Milisekundy;
      if (StavLevehoBlinkru == LOW)
        StavLevehoBlinkru = HIGH;
      else
        StavLevehoBlinkru = LOW;
      digitalWrite(LevyBlinkr, StavLevehoBlinkru);
    }
  } else {
    StavLevehoBlinkru =LOW,
    digitalWrite(LevyBlinkr, LOW);
  } // konnec leve strany blinkru

```

```

StavTlacitkaLevehoBlinkru = digitalRead(PravyBlinkrTlacitko);//Prava strana - blinkry
  if (StavTlacitkaLevehoBlinkru == HIGH) {

    if (Milisekundy - UbehlyCasPravehoBlinkru > DobaBlikaniBlinkru) {
      UbehlyCasPravehoBlinkru = Milisekundy;
      if (StavPravehoBlinkru == LOW)
        StavPravehoBlinkru = HIGH;
      else
        StavPravehoBlinkru = LOW;
      digitalWrite(PravyBlinkr, StavPravehoBlinkru);
    }
  } else {
    StavPravehoBlinkru=LOW,
    digitalWrite(PravyBlinkr, LOW);
  } // konec prave strany blinkru

```

```

StavTlacitkaPZsvetla = digitalRead(SvetlaPZTlacitko);//svetla predni + zadni
if(StavTlacitkaPZsvetla==LOW) TlacitkoPZsvetlaWasOff = true;
if(StavTlacitkaPZsvetla==HIGH && TlacitkoPZsvetlaWasOff == true){
  TlacitkoPZsvetlaWasOff = false;
  if(StavPZSvetla==LOW){StavPZSvetla=HIGH;}
  else{StavPZSvetla=LOW;}}
digitalWrite(SvetlaPZ,StavPZSvetla);
//konec predni + zadni svetlo

```

```

StavRGBTlacitka = digitalRead(GBtlacitko);
  if(StavRGBTlacitka==LOW)
    RgbTlacitkoWasOff = true;
  if(StavRGBTlacitka==HIGH && RgbTlacitkoWasOff == true){
    RgbTlacitkoWasOff = false;
    KteraBarva = KteraBarva +1;
    if(KteraBarva==7)
      KteraBarva=0;}
  switch(KteraBarva){

case 1 :StavCervene=true;StavZelene=true;StavModre=false;break;//modra
case 2 :StavCervene=true;StavZelene=false;StavModre=false;break;//aqua
case 3 :StavCervene=true;StavZelene=false;StavModre=true;break;//zelena
case 4 :StavCervene=false;StavZelene=false;StavModre=true;break;//zluta
case 5 :StavCervene=false;StavZelene=true;StavModre=true;break;//cervena
case 6 :StavCervene=false;StavZelene=false;StavModre=false;break;//bila
default:;}
digitalWrite(Cervena,StavCervene);
digitalWrite(Modra,StavModre);
digitalWrite(Zelena,StavZelene);

StavTlacitkaChoduMotoru = digitalRead(BehMotoruTlacitko);//chod motoru
if(StavTlacitkaChoduMotoru==LOW)
  TlacitkoMotoruWasOff = true;
if(StavTlacitkaChoduMotoru==HIGH && TlacitkoMotoruWasOff == true){
  TlacitkoMotoruWasOff = false;
  if(StavChoduMotoru==LOW) {StavChoduMotoru=HIGH;}
  else{StavChoduMotoru=LOW;}}
  digitalWrite(BehMotoruRele, StavChoduMotoru);

//konec rele chodu motoru
StavTlacitkaKlaksonu = digitalRead(KlaksonTlacitko);//klakson
if (StavTlacitkaKlaksonu == HIGH) {
  StavKlaksonu = HIGH;}
else {StavKlaksonu = LOW;}
digitalWrite(KlaksonRele, StavKlaksonu);
//konec klaksonu

  StavBrzdovehoTlacitka = digitalRead(SignalZadníhoSvetla);//zadni brzdove svetlo
if (StavBrzdovehoTlacitka == HIGH) {
  if (Milisekundy - UbehlyCasZadníhoSvetla > DobaBlikaniZadníhoSvetla) {
    UbehlyCasZadníhoSvetla = Milisekundy;
    if (StavZadníhoSvetla == LOW)
      StavZadníhoSvetla = HIGH;
    else
      StavZadníhoSvetla = LOW;
    digitalWrite(ZadniSvetloVyst, StavZadníhoSvetla);
  }
} else {
  StavZadníhoSvetla=LOW;}
digitalWrite(zadnisvetlovyst,StavBrzdovehoTlacitka); //konec zadního brzdoveho svetla

```

```

StavSpzTlacitka = digitalRead(SpzTlacitko);//SPZ
if(StavSpzTlacitka==LOW)
    SpzTlacitkoWasOff = true;
if(StavSpzTlacitka==HIGH && SpzTlacitkoWasOff == true){
    SpzTlacitkoWasOff = false;
    if(StavSpz==0) {StavSpz=1;}
    else{StavSpz=0;}
    Serial1.print(StavSpz); //konec ovdání SPZ

//ANALOGOVE VSTUPY

if (Milisekundy - CasOdPoslednihoMereni > DobaDoDalsihoMereni) {
    CasOdPoslednihoMereni=0;

    teplota = analogRead(TeplPin);//mereni teploty
    FloatCel = (1023-teplota-519)/7;
    cel = (FloatCel);//konec mereni teploty

    HodnotaSrazek = analogRead(VstupSrazek);//mereni srazek
    int range = map(HodnotaSrazek, MinHodnotaSrazek, MaxHodnotaSrazek, 0, 3);
    switch (range) {
    case 0: // Silny dest

        StavSrazek=0;
        break;
    case 1: // Dest

        StavSrazek=1;
        break;
    case 2: // Zadne srazky

        StavSrazek=2;
        break;
    }//konec mereni srazek

    napeti = digitalRead(VstupNapeti);//mereni napeti na baterce
    nap = (napeti*0.01528662); //konec mereni napeti

    jas = analogRead(VstupJasu);
    if ((StavSrazek==2) && (spintep < cel))//vyhrev riditek
    {StavVyhrevuRiditek=LOW;}

    if ((StavSrazek<2) || (spintep > cel))
    {StavVyhrevuRiditek=HIGH;}
    digitalWrite(VyhrevRele,StavVyhrevuRiditek);
}

```

```

//Tony
if (
    StavTlacitkaPZsvetla==HIGH||
    StavRGBTlacitka==HIGH||
    StavTlacitkaChoduMotoru==HIGH||
    StavSpzTlacitka==HIGH
)
    {tone(Reproduktor, 8000);}
else {noTone(Reproduktor);}
//SERIOVY PORT

Serial.print("Teplota = ");
Serial.print(temperatura);
Serial.print("C");
Serial.println(); //vypis teploty na seriový port

if (StavSrazek==0){Serial.println("Silny dest");}
if (StavSrazek==1){Serial.println("Dest");}
if (StavSrazek==2){Serial.println("Zadne srazky");} //Vypis srazek na seriový port

if(StavVyhrevuRiditek==HIGH){Serial.println("Vyhrev zapnut");}
if(StavVyhrevuRiditek==LOW){Serial.println("Vyhrev vypnut");}

if(StavChoduMotoru==HIGH){Serial.print("Chod motoru je povolen");}
if(StavChoduMotoru==LOW){Serial.print("Chod motoru je zakazan");}
Serial.println();

    Serial.print("Napeti = ");
Serial.print(nap);
Serial.print("V");
Serial.println(); //Vypis napeti na seriový port}

Serial.print(jas);
Serial.println();

Serial.print(blueTooth);
Serial.println();

//Display
myGLCD.setFont(BigFont);
if(KteraBarva==1 &&
which1==false){myGLCD.fillRect(VGA_BLUE,which6=false,which1=true,myGLCD.setBackgroundColor(VGA_BLUE));} //display se zabarvi do stejne barvy jako RGB ledky
if(KteraBarva==2 &&
which2==false){myGLCD.fillRect(VGA_AQUA,which1=false,which2=true,myGLCD.setBackgroundColor(VGA_AQUA));}
if(KteraBarva==3 &&
which3==false){myGLCD.fillRect(VGA_LIME,which2=false,which3=true,myGLCD.setBackgroundColor(VGA_LIME));}

```

```

if(KteraBarva==4 &&
which4==false){myGLCD.fillScr(VGA_YELLOW),which3=false,which4=true,myGLCD.setBackColor(VGA_
YELLOW);}
if(KteraBarva==5 &&
which5==false){myGLCD.fillScr(VGA_RED),which4=false,which5=true,myGLCD.setBackColor(VGA_RED)
;}
if(KteraBarva==6 &&
which6==false){myGLCD.fillScr(VGA_TEAL),which5=false,which6=true,myGLCD.setBackColor(VGA_TEA
L);}//konec nastaveni barvy displeje
int BackColor = myGLCD.getBackColor();
myGLCD.print("ADRUmoto", CENTER, 4);//Nadpis
myGLCD.printNumI(CEL,4,400);//Vypis teploty
myGLCD.print("*C",62,400);

myGLCD.setFont(SevenSegNumFont);
Min = (Milisekundy/60000);
myGLCD.printNumI(Min,245,5);

myGLCD.setFont(BigFont);
if (StavSrazek==0){myGLCD.print("Silny dest ",29,65);}//vypis srazek na display
if (StavSrazek==1){myGLCD.print("Dest ",29,65);}
if (StavSrazek==2){myGLCD.print("Zadne srazky",29,65);}

myGLCD.setFont(SmallFont);
if (Min == 0) {myGLCD.print("Minut ",258,65);}
if (Min == 1) {myGLCD.print("Minuta ",258,65);}
if (Min >=2 && Min <=4) {myGLCD.print("Minuty ",258,65);}//minuty
if (Min >=5) {myGLCD.print("Minut ",258,65);}

myGLCD.print("By M. Kaiser (SPS Pisek)",115,402);//Info o autorovi v pravem dolnim rohu

myGLCD.setFont(BigFont);

myGLCD.print("Bluetooth = ",18,142);
myGLCD.printNumI(blueetooth,208,142);
if(StavVyhrevuRiditek==HIGH){myGLCD.print("Vyhrev zapnut",10,35);}//vyhrev riditek
if(StavVyhrevuRiditek==LOW){myGLCD.print("Vyhrev vypnut",10,35);}

if(StavChoduMotoru==HIGH){myGLCD.print("Chod motoru povolen ",10,92);}//chod motoru
if(StavChoduMotoru==LOW){myGLCD.print("Chod motoru zakazan ",10,92);}

if(StavSpz==1) {myGLCD.print("SPZ je viditelna",20,118);}//SPZ
if(StavSpz==0) {myGLCD.print(" SPZ je skryta ",20,118);}

byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
myGLCD.setFont(SevenSegNumFont);
myGLCD.printNumI(second, 198, 170);//vypis sekund

```



```

if (second==1){ (myGLCD.clrScr());
switch (KteraBarva){
case 1: which1=false; break;
case 2: which2=false; break;
case 3: which3=false; break;
case 4: which4=false; break;
case 5: which5=false; break;
case 6: which6=false; break;
default:;} }
myGLCD.printNumI(minute, 125, 170);//vypis minut
myGLCD.printNumI(hour, 52, 170);//vypis hodin
myGLCD.setFont(BigFont);
myGLCD.printNumI(dayOfMonth,1,170);//vypis dne v mesici
myGLCD.print(".",31,170);
myGLCD.printNumI(month,1,195);//vypis mesice
myGLCD.print(".",31,195);
myGLCD.fillCircle(122,188,3);//dvoutecky
myGLCD.fillCircle(122,202,3);
myGLCD.fillCircle(193,188,3);
myGLCD.fillCircle(193,202,3);
switch (dayOfWeek) { //den v tydnu
case 1: myGLCD.print("Ne",280,203); break;
case 2: myGLCD.print("Po",280,203); break;
case 3: myGLCD.print("Ut",280,203); break;
case 4: myGLCD.print("St",280,203); break;
case 5: myGLCD.print("Ct",280,203); break;
case 6: myGLCD.print("Pa",280,203); break;
case 7: myGLCD.print("So",280,203); break;
default:;
}
if (nap < kritNap) { myGLCD.setBackColor(VGA_RED);}
if (nap >= kritNap) { myGLCD.setBackColor(BackColor);}
myGLCD.printNumF(nap,2,2,5);// vypis napeti
myGLCD.print("V",66,5);

myGLCD.setBackColor(BackColor);

if(StavKlaksonu==HIGH){myGLCD.print("Klakson zapnut",95,370),myGLCD.setColor(VGA_LIME);}//
Ukazatel u klaksonu
if(StavKlaksonu==LOW){myGLCD.print("Klakson vypnut",95,370),myGLCD.setColor(VGA_MAROON);}
myGLCD.fillRect(0,366,84,389);

if (StavPravehoBlinkru==HIGH){myGLCD.setColor(VGA_YELLOW);}//signalizace blinkru
if (StavPravehoBlinkru==LOW){myGLCD.setColor(VGA_BLACK);}
myGLCD.fillRoundRect(270,430,320,480);

if (StavLevehoBlinkru==HIGH){myGLCD.setColor(VGA_YELLOW);}//signalizace blinkru
if (StavLevehoBlinkru==LOW){myGLCD.setColor(VGA_BLACK);}
myGLCD.fillRoundRect(0,430,50,480);

```

```

myGLCD.setColor(VGA_SILVER);//ramecek kolem svetel a blinkru
myGLCD.fillRect(0,420,319,429);
myGLCD.fillRect(51,420,60,479);
myGLCD.fillRect(259,420,269,479);

if (StavZadnihoSvetla==LOW){ myGLCD.setColor(VGA_BLACK);}//zadni brzdo ve svetlo- signalizace
if (StavZadnihoSvetla==HIGH){ myGLCD.setColor(VGA_RED);}
myGLCD.fillRoundRect(61,455,259,479);

if (jas>500){ myGLCD.setColor(VGA_BLACK);}
if (jas<500){ myGLCD.setColor(VGA_YELLOW);}

myGLCD.fillCircle(300,170,20);
myGLCD.setColor(VGA_BLACK);
myGLCD.fillCircle(295,165,2);
myGLCD.fillCircle(305,165,2);
myGLCD.fillRect(295,182,305,184);
myGLCD.fillRect(292,180,307,181);
myGLCD.fillRect(290,178,310,179);

if(StavPZSvetla==LOW){ myGLCD.setColor(VGA_BLACK);}//predni svetlo+zadni svetlo- signalizace
if(StavPZSvetla==HIGH){ myGLCD.setColor(VGA_YELLOW);}
myGLCD.fillRoundRect(61,430,259,455);
myGLCD.setColor(VGA_BLACK);
myGLCD.fillRect(230,0,235,85);
myGLCD.fillRect(0,83,319,88);
myGLCD.fillRect(85,0,90,30);
myGLCD.fillRect(0,25,230,30);
myGLCD.fillRect(0,55,230,60);
myGLCD.fillRect(0,110,319,115);
myGLCD.fillRect(0,390,319,395);
myGLCD.fillRect(95,390,100,419);
myGLCD.fillRect(0,360,319,365);
myGLCD.fillRect(85,360,90,390);
myGLCD.fillRect(0,135,319,140);
myGLCD.fillRect(270,135,275,195);
myGLCD.fillRect(270,196,319,201);
myGLCD.fillRect(0,160,270,165);
myGLCD.fillRect(0,220,319,225);

}

```