



STREDNÁ PRIEMYSELNÁ ŠKOLA TECHNICKÁ

Novomeského 5/24

036 36 Martin

Ročníkový projekt

Študijný odbor: Elektrotechnika

Vozidlo s Arduinom alebo Raspberry PI

Vypracoval: Matej Mokrý

Trieda: 4.D

Školský rok: 2018/2019

Konzultant: Ing. Eduard Jadroň

Čestné vyhlásenie

Prehlasujem, že prácu s názvom Vozidlo s Arduinom alebo Raspberry PI, som vypracoval samostatne s použitím uvedenej literatúry a súčasne, že som túto prácu neprihlásil a neprezentoval v žiadnej inej súťaži vyhlásenej – MŠV VaŠ SR.

V Turanoch, 13.2.2019

.....

Licencia

This program is free software: you can redistribute it and/or modify it under the terms of the GNU (General Public License) as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

ABSTRAKT V ŠTÁTNOM JAZYKU

Projekt sa zaoberá programovaním mikropočítačov a ich komunikáciou pomocou webového rozhrania alebo sériovej linky. Cieľom práce bolo zhotovenie multifunkčného vozidla, ktoré dokáže ovládať aj bežný užívateľ. K realizácii projektu boli potrebné dva mikropočítače - Arduino Uno a Rpi3 B. Ako programovací jazyk pre mikroprocesor Arduino sa použil C a pre Raspberry PI 3 node-red. Vývojové prostredie pre Arduino bolo Microsoft Visual Studio 2017 s addonom Arduino IDE for Visual Studio. Toto vozidlo je určené širokému publiku, ako začínajúcim programátorom alebo elektrotechnikom .

Kľúčové slová: ARDUINO. C. RPI3. VOZIDLO.

ABSTRACT IN ENGLISH

The project deals with programming of microcomputers and their communication via web interface or serial line. The aim of the work was to make a multifunctional vehicle that can be operated by a common user. Two microcomputers were needed to implement the project - Arduino Uno and Rpi3 B. For programming language for microprocessor Arduino was used a language C and for pre Raspberry PI 3 node-red. The Arduino development environment was Microsoft Visual Studio 2017 with the Arduino IDE addon for Visual Studio. This vehicle is intended for a wide audience, as for the beginner programmers or electricians.

Key words: ARDUINO. C. RPI3. VEHICLE.

Obsah

Zoznam obrázkov	6
Zoznam skratiek	7
Úvod.....	8
1 Mikropočítač Arduino Uno.....	9
1.1 Mikropočítač Raspberry Pi	10
1.3 Node-red	10
2 Konštrukcia vozidla a vzniknuté komplikácie.....	12
3 Rozbor vozidla	14
3.1 Rozbor programu v Arduino (LCD display)	14
3.2 Rozbor programu v Arduino (" Autoconfig ")	15
3.3 Rozbor programu v Arduino (Help)	17
3.4 Rozbor programu v Arduino (Kernel).....	17
4 Node-red na RPI (Status)	19
4.1 Node-red na RPI (Telegram Bot)	20
Záver.....	21
Zoznam použitej literatúry	22

Zoznam obrázkov

Obrázok č. 1 – Arduino uno

Obrázok č. 2 – Raspberry PI 3

Obrázok č. 3 – Node-red

Obrázok č. 4 – LCD program

Obrázok č. 5 – Poloautomatická inštalácia kód 1/2

Obrázok č. 6 – Poloautomatická inštalácia kód 2/2

Obrázok č. 7 – Help program

Obrázok č. 8 – Kernel

Obrázok č. 9 – Program zobrazujúci základne informácie RPI

Obrázok č. 10 – Telegram bot

Zoznam skratiek

RPI3 Raspberry Pi 3 Model B

OS Operating system

CFG Configuration

Úvod

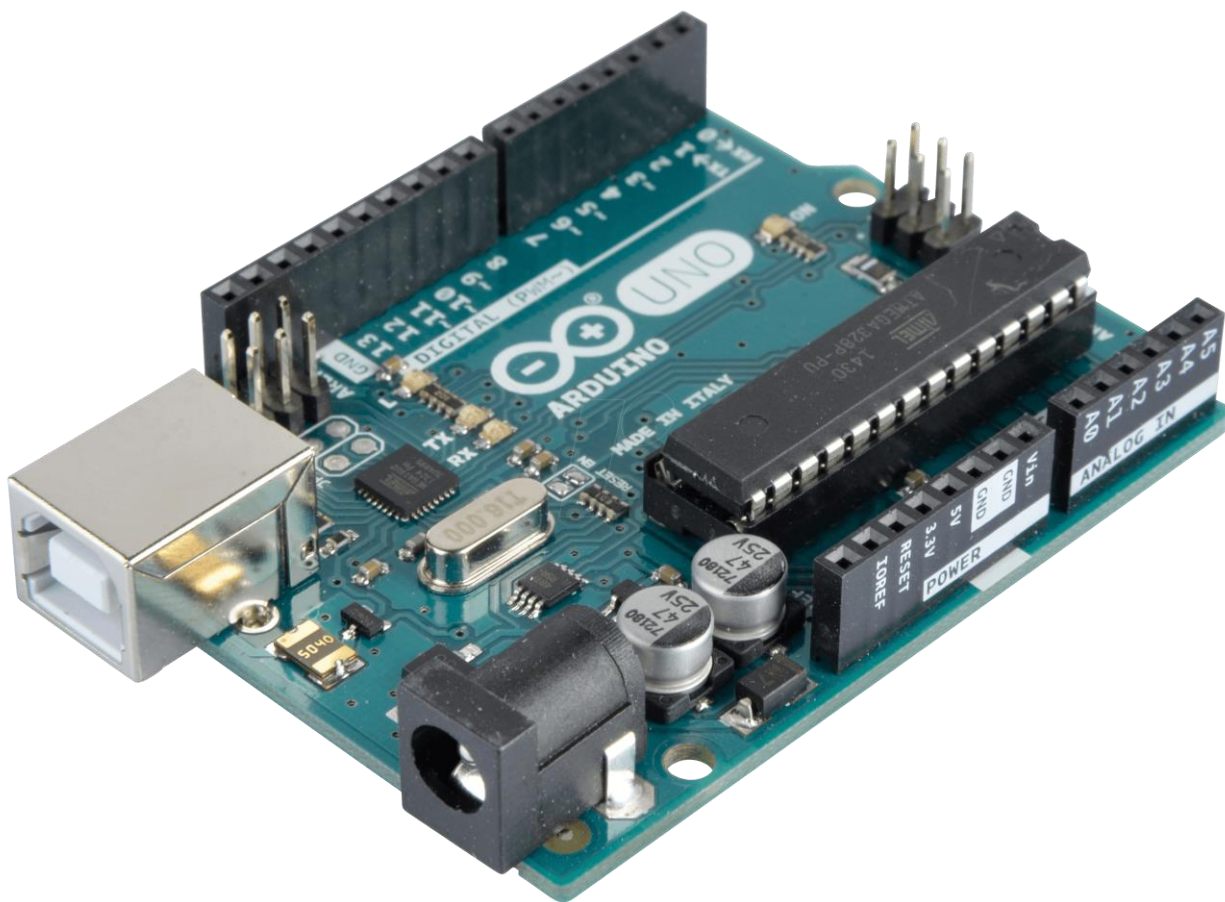
Tému *Vozidlo s Arduinom alebo Raspberry PI* som si zvolil kvôli zlepšeniu a aplikácií poznatkov v mikroprocesoroch, kde som sa sústredil najmä na ich funkciu a programovanie. Zároveň som chcel preskúmať funkcie senzorov.

Už v detstve som sa zaujímal o pohybujúce sa mechanizmy. To bol dôvod, prečo som sa rozhodol pre projekt Smart Auto. Po preskúmaní jednotlivých možností som zistil, že vozidlo poskladám sám, a to za pomoci tzv. kitu¹. Dôležitým faktorom bolo najprv samotnému projektu porozumieť a až následne ho vyhotoviť. Zistil som, že väčšina kitov sú na mikroprocesor Arduino, s ktorým som mal skúsenosti zo školy. Arduino je malý mikropočítač, ktorý je síce cenovo dostupný, ale má jednu nevýhodu – neschopnosť multitasking-u. Keďže už dlhšiu dobu sa zaoberám RPI3 a operačným systémom Linux (ubuntu) tak problém som vyriešil pridaným RPI3, ktorý má vlastný operačný OS, a teda bude schopný multitaskingu.

¹ Návrh vozidla už bol vypracovaný, zostáva ho len zhotoviť.

1 Mikropočítač Arduino Uno

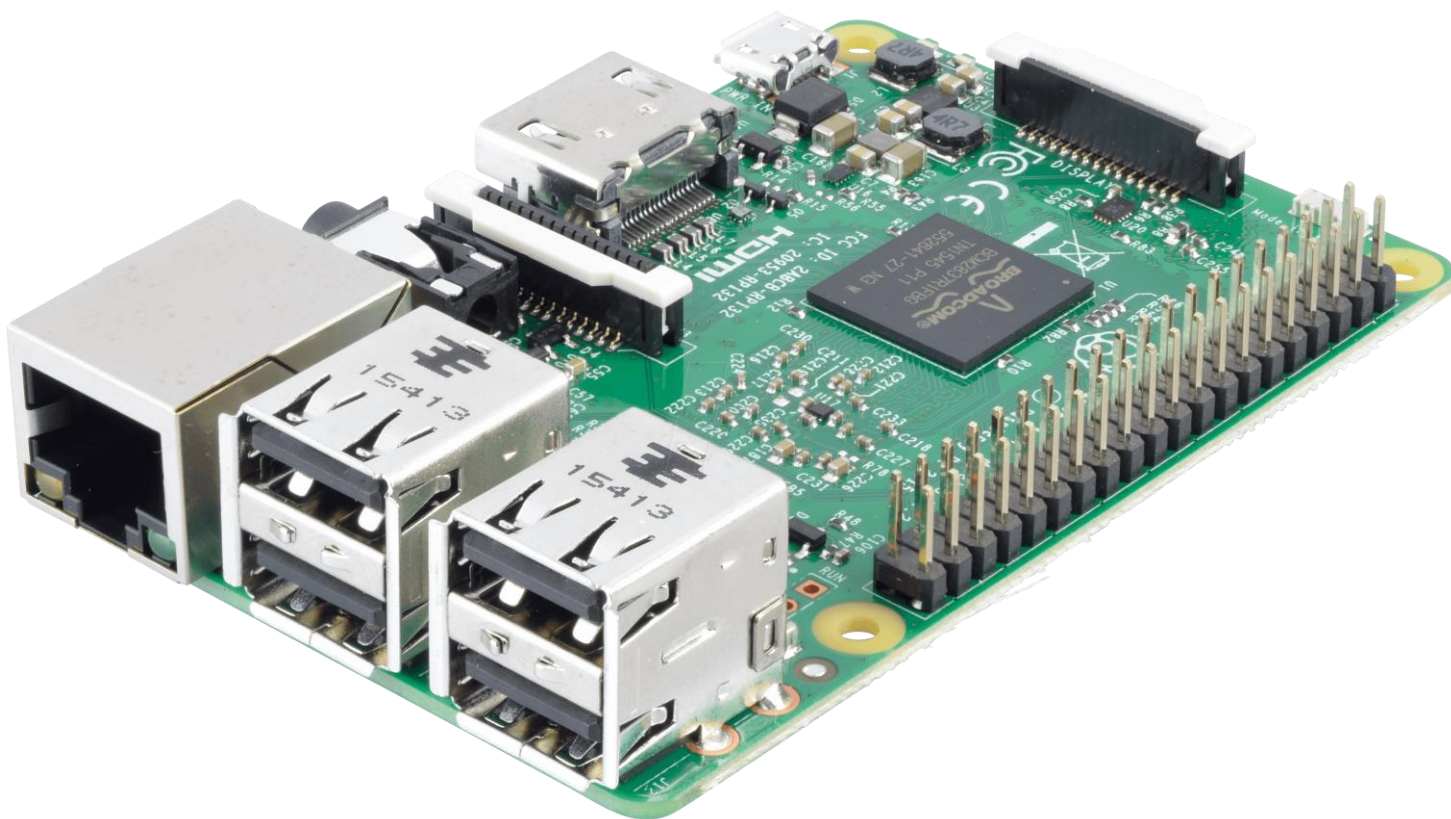
Arduino Uno je vývojová doska s mikropočítačem AVR Atmega 328. Ide o základnú verziu, ktorá nemá v sebe wifi modul. Má 14 digitálnych vstupov a 6 analógových. Existuje viacero druhov. Výhoda Arduina spočíva v jeho cene a jednoduchosti.



Obrázok č. 1 – Arduino uno

1.2 Mikropočítač Raspberry Pi

Raspberry Pi je jednodoskový mikropočítač založený na platforme ARM. Má 64 bitový procesor so štvoricou jadier Cortex-A53 s frekvenciou 1,2 GHz. Obsahuje 40 pinov, Bluetooth, čítačku pamäťových kariet, Wi-Fi, LAN, výstup pre slúchadlá/reproduktor a micro USB. Jeho výhodou je operačný systém. Pri CPU je zvýšená prevádzková teplota, a preto sa odporúča použitie ventilátora.

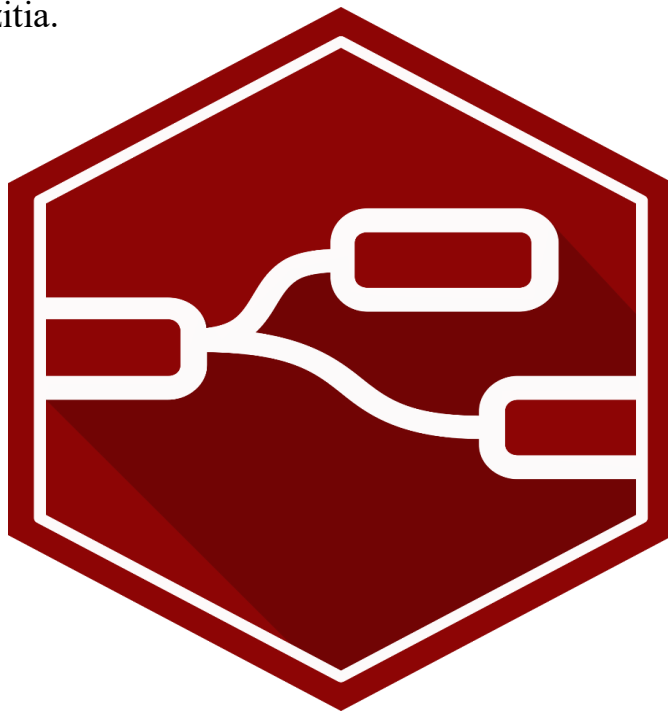


Obrázok č. 2 – Raspberry Pi 3

1.3 Node-red

Node-RED je programovací nástroj na zapojenie hardvérových zariadení, rozhraní API a služieb online inovatívnymi spôsobmi.

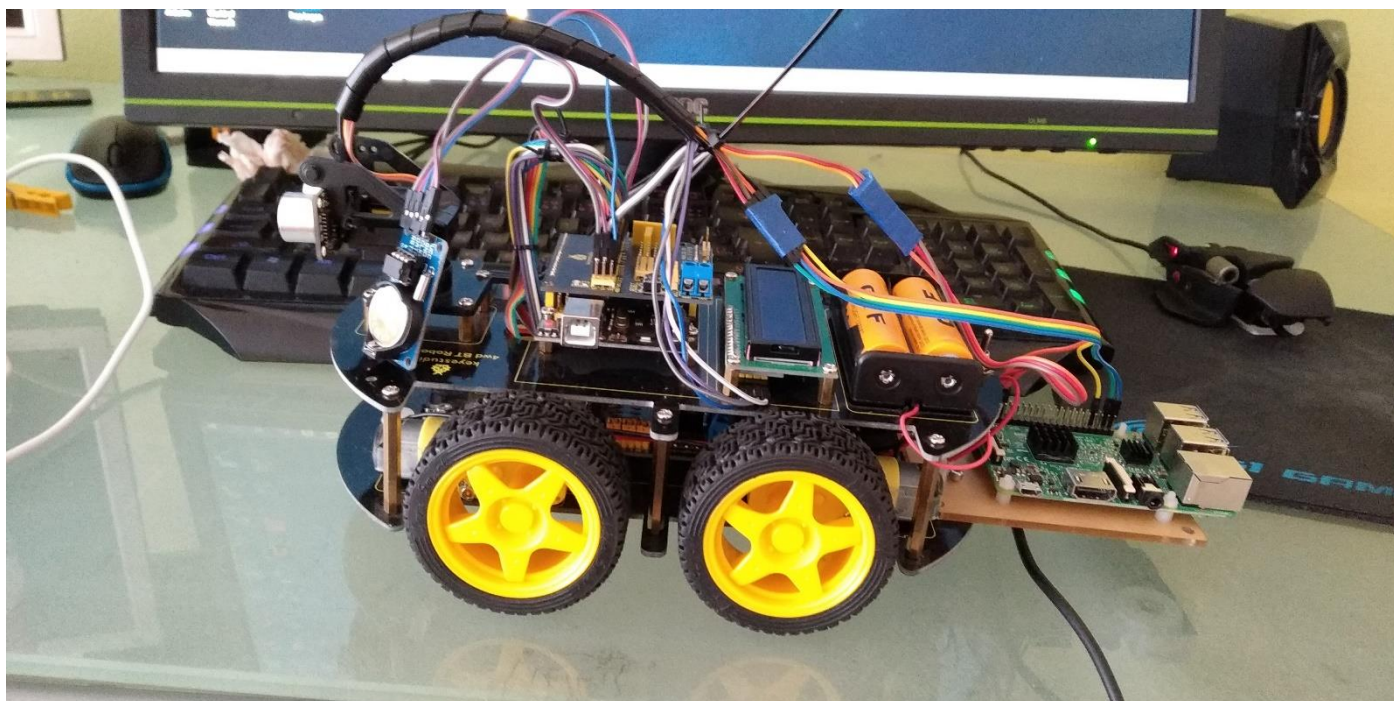
Poskytuje užívateľovi editor založený na prehliadači, ktorý umožňuje jednoduché spájanie flows pomocou širokej škály preddefinovaných uzlov. Uzly sa dajú pridať do projektu jediným kliknutím. Beží na Node.js, a preto sa môže spustiť aj na slabších zariadeniach napr. RPI. Obsahuje ako 225 000 modulov, preto má veľkú škálu využitia.



Obrázok č. 3 – Node-red Logo

2 Konštrukcia vozidla a vzniknuté komplikácie

Hlavným cieľom môjho projektu Vozidlo s Arduinom alebo Raspberry PI bola schopnosť prevádzky vozidla. Na účely zhotovenia projektu som si zvolil kit od dodávateľa Keyestudio 4WD, ktorý obsahuje LCD display a IR reciver. Vozidlo malo dobrú kvalitu. Vyhodenie kitu bolo jednoduché, nakoľko obsahoval veľmi podrobný a dobre spracovaný manuál spolu s PDF verziou. K dispozícii bolo aj video v balení. Prvá komplikácia vznikla pri káblovaní, ktoré sa robí na arduino shielde, a s tým som nemal ešte žiadne skúsenosti.



Samotné programovanie vozidla sa začalo komplikovať, keď bolo potrebné využiť multitasking, ktoré arduino nemá, ale čiastočne sa to dalo vyriešiť knižnicou FreeRTOS. Nesplnilo to moje očakávanie, keďže to bolo príliš pomalé, a preto som pridal RPI3. Ďalší problém sa vyskytol na I2C displayom kedy chyba v knižnici spôsobila to, že bolo vidieť iba prvé písmeno. Vzniknutý problém sa vyriešil v editovaní configu LiquidCrystal_I2C.cc kde `return 0;` prepíšeme na `return 1;`.

Na Raspberry som nainštaloval Raspbian, ktorý fungoval bezchybne. Na ovládanie auta pomocou webového GUI som použil už nainštalovaný software node-red. Node-red je často využívaný hlavne pre jeho jednoduchosť. Aby som auto dokázal ovládať cez web musel som stiahnuť knihovanú node-red-dashboard. Rozhodol som sa, že auto bude mať aj vlastnú meteorologickú stanicu, pretože jeden dht senzor som mal nevyužitý. Aby dht senzor fungoval, potreboval knižnicu s názvom node-red-contrib-dht-sensor. Ale túto knižnicu sa nepodarilo úspešne nainštalovať. Preto som musel skontrolovať log, v ktorom bola chyba uvedená. Oprava chyby sa uskutočnila prostredníctvom inštalácie ďalšej chýbajúcej knižnice.

3 Rozbor vozidla

Hlavným, ale zároveň najslabším, článkom vozidla je Arduino. Ako bolo už spomenuté v úvode, nie je schopný multitaskingu, ktorý je tu žiaduci. K Arduino je pripojený L298n shild, ktorý ovláda 4 TT motory a otáčky motora. Arduino je cez sL298n shild napájaný pomocou dvoch 3,7V batérií. Hlavnou vetvou je spojenie Arduina a RPI3 cez USB kábel. Arduino s RPI3 komunikuje cez Serial port.

3.1 Rozbor programu v Arduine (LCD display)

Pri prvotnom pustení sa ako prvé nastaví LCD display s dodatočným nastavením rýchlosti, dátumu atď’.

```

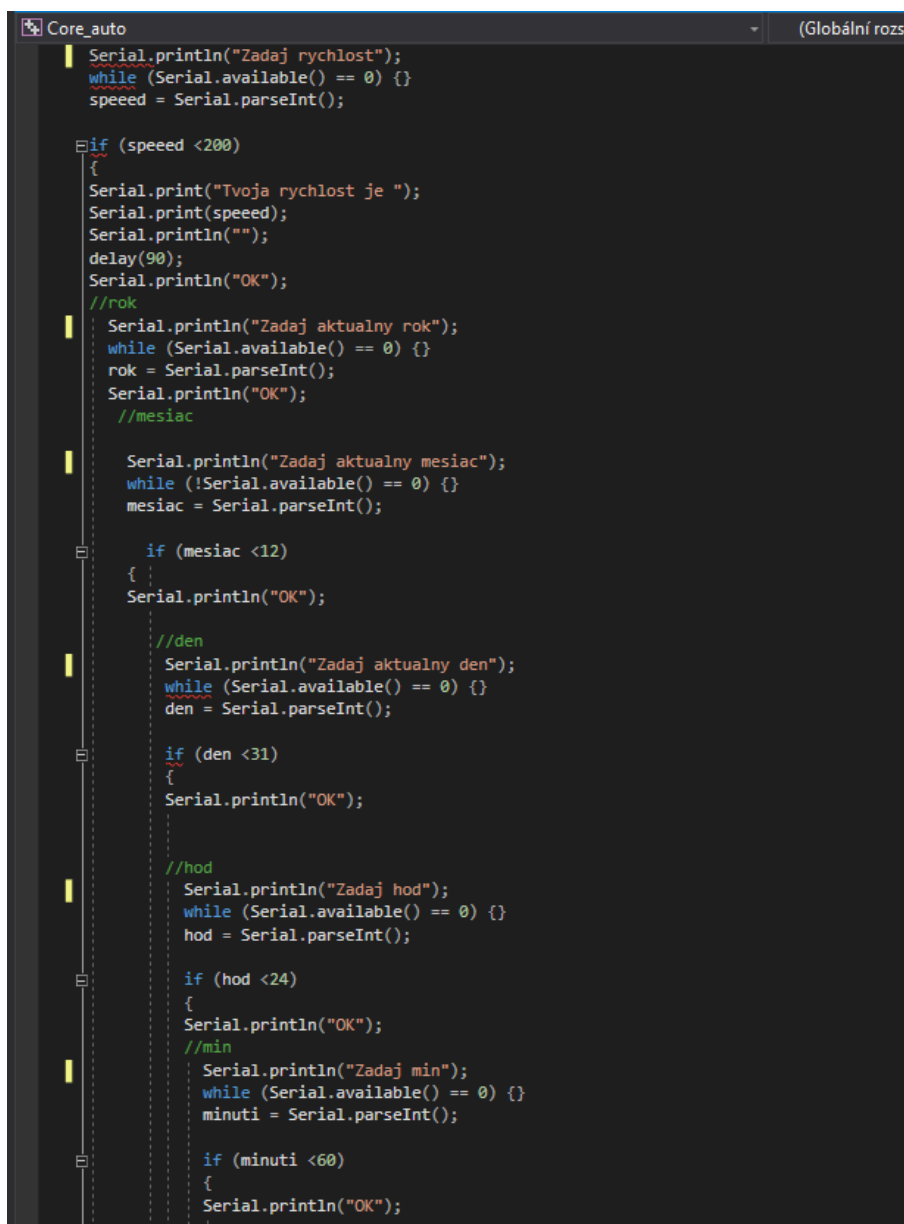
  /** *****
  **
  **                               LOGO
  **
  ** *****/
  LiquidCrystal_I2C lcd(0x27, 16, 2);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(2,0);
  lcd.print("Matej Mokry");
  delay (250);
  lcd.setCursor(1,1);
  lcd.print("BUILD 0.95 GNU ");
  delay (5000);
  /** *****
  **
  **                               Start
  **
  ** *****/
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Ip 192.168.0.17");
    lcd.setCursor(0, 1);
    lcd.print("port 1880 GPL");
  }

```

Obrázok č. 4 – LCD program

3.2 Rozbor programu v Arduino („Autoconfig“)

Auto nastaví defaultné nastavenia pre užívateľa. Taktiež má užívateľ priestor na vlastné nastavenie. Nazval som to poloautomatická konfigurácia. Pri poloautomatickej konfigurácii si užívateľ sám nastaví svoje parametre.



```
Core_auto (Globální rozsah)

Serial.println("Zadaj rychlost");
while (Serial.available() == 0) {}
speed = Serial.parseInt();

if (speed < 200)
{
  Serial.print("Tvoja rychlost je ");
  Serial.print(speed);
  Serial.println("");
  delay(90);
  Serial.println("OK");
  //rychlost

  Serial.println("Zadaj aktualny rok");
  while (Serial.available() == 0) {}
  rok = Serial.parseInt();
  Serial.println("OK");
  //rok

  Serial.println("Zadaj aktualny mesiac");
  while (!Serial.available() == 0) {}
  mesiac = Serial.parseInt();

  if (mesiac < 12)
  {
    Serial.println("OK");
    //mesiac

    Serial.println("Zadaj aktualny den");
    while (Serial.available() == 0) {}
    den = Serial.parseInt();

    if (den < 31)
    {
      Serial.println("OK");
      //den

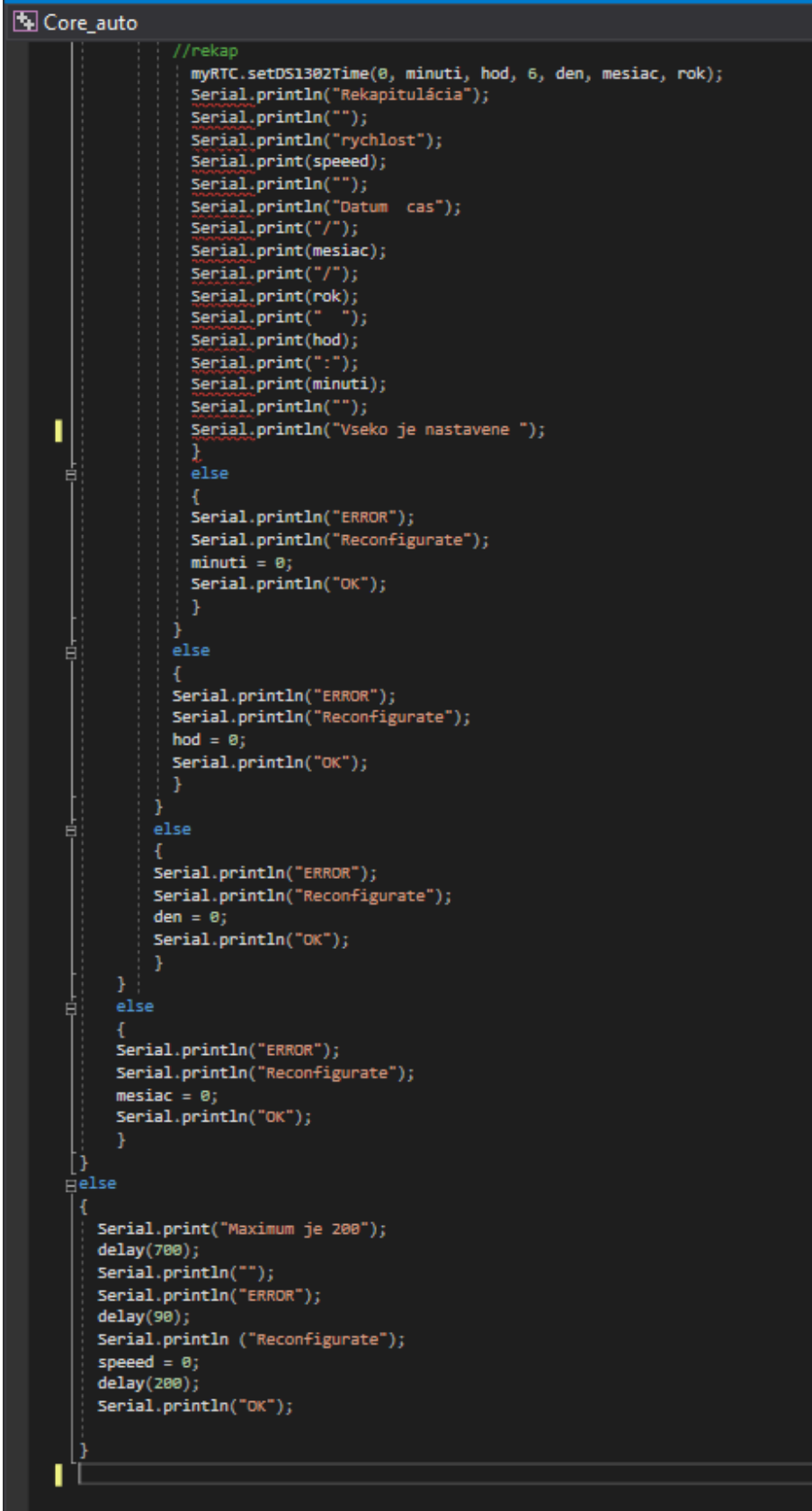
      Serial.println("Zadaj hod");
      while (Serial.available() == 0) {}
      hod = Serial.parseInt();

      if (hod < 24)
      {
        Serial.println("OK");
        //hod

        Serial.println("Zadaj min");
        while (Serial.available() == 0) {}
        minuti = Serial.parseInt();

        if (minuti < 60)
        {
          Serial.println("OK");
          //min
        }
      }
    }
  }
}
```

Obrázok č. 5 – poloautomatická inštalácia kód 1/2



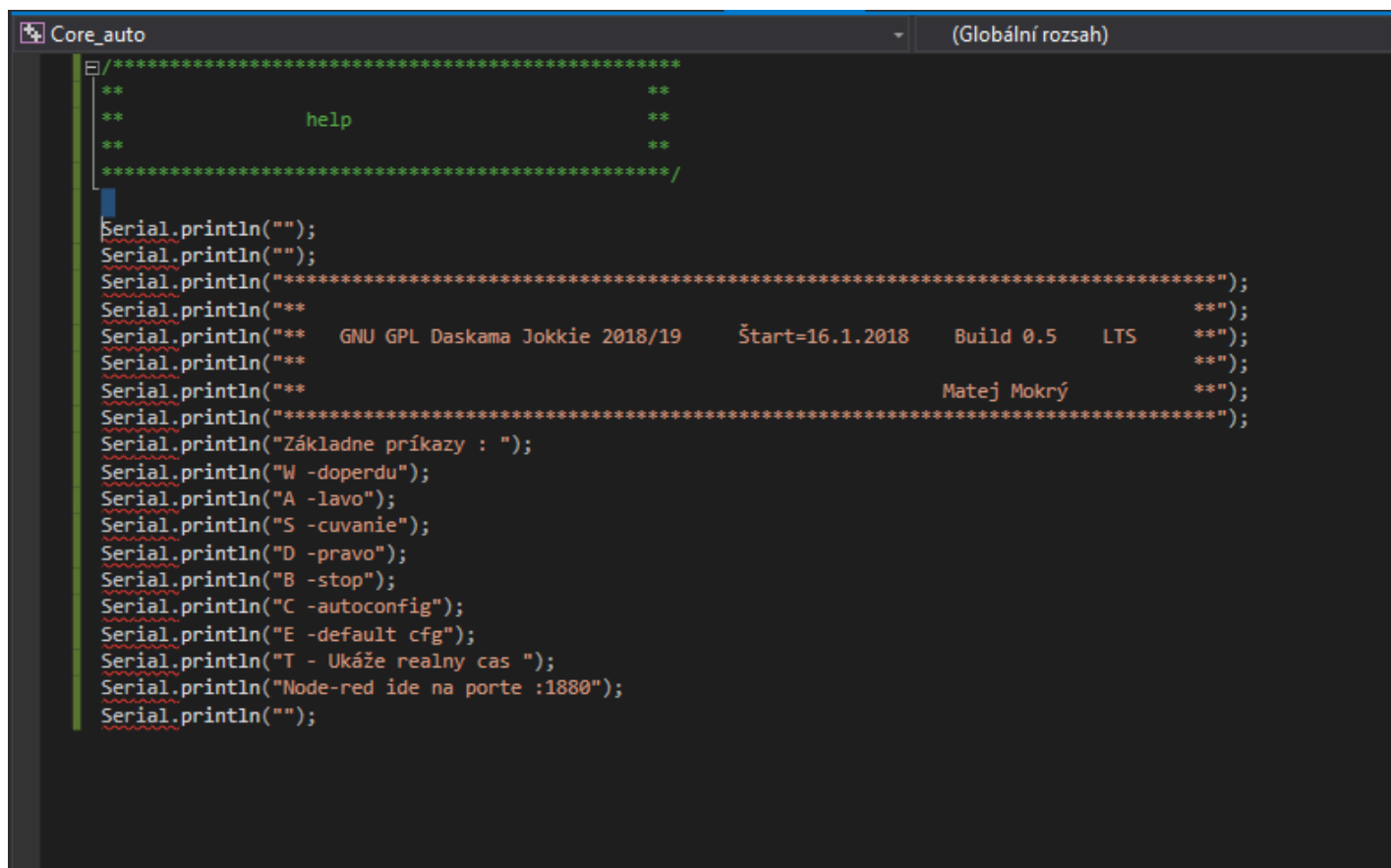
```
Core_auto

//rekap
myRTC.setDS1302Time(0, minuti, hod, 6, den, mesiac, rok);
Serial.println("Rekapitulácia");
Serial.println("");
Serial.println("rychlost");
Serial.print(speed);
Serial.println("");
Serial.println("Datum  cas");
Serial.print("/");
Serial.print(mesiac);
Serial.print("/");
Serial.print(rok);
Serial.print(" ");
Serial.print(hod);
Serial.print(":");
Serial.print(minuti);
Serial.println("");
Serial.println("Vseko je nastavene ");
}
else
{
    Serial.println("ERROR");
    Serial.println("Reconfigure");
    minuti = 0;
    Serial.println("OK");
}
}
else
{
    Serial.println("ERROR");
    Serial.println("Reconfigure");
    hod = 0;
    Serial.println("OK");
}
}
else
{
    Serial.println("ERROR");
    Serial.println("Reconfigure");
    den = 0;
    Serial.println("OK");
}
}
else
{
    Serial.println("ERROR");
    Serial.println("Reconfigure");
    mesiac = 0;
    Serial.println("OK");
}
}
else
{
    Serial.print("Maximum je 200");
    delay(700);
    Serial.println("");
    Serial.println("ERROR");
    delay(90);
    Serial.println("Reconfigure");
    speed = 0;
    delay(200);
    Serial.println("OK");
}
}
```

Obrázok č. 6 – poloautomatická inštalácia kód 2/2

3.3 Rozbor programu v Arduino (Help)

Tento príkaz je ako v bežnom programe. Vypíše príkazy, ktoré auto podporuje. Bežný užívateľ by to mal využívať iba v prípade debugu.

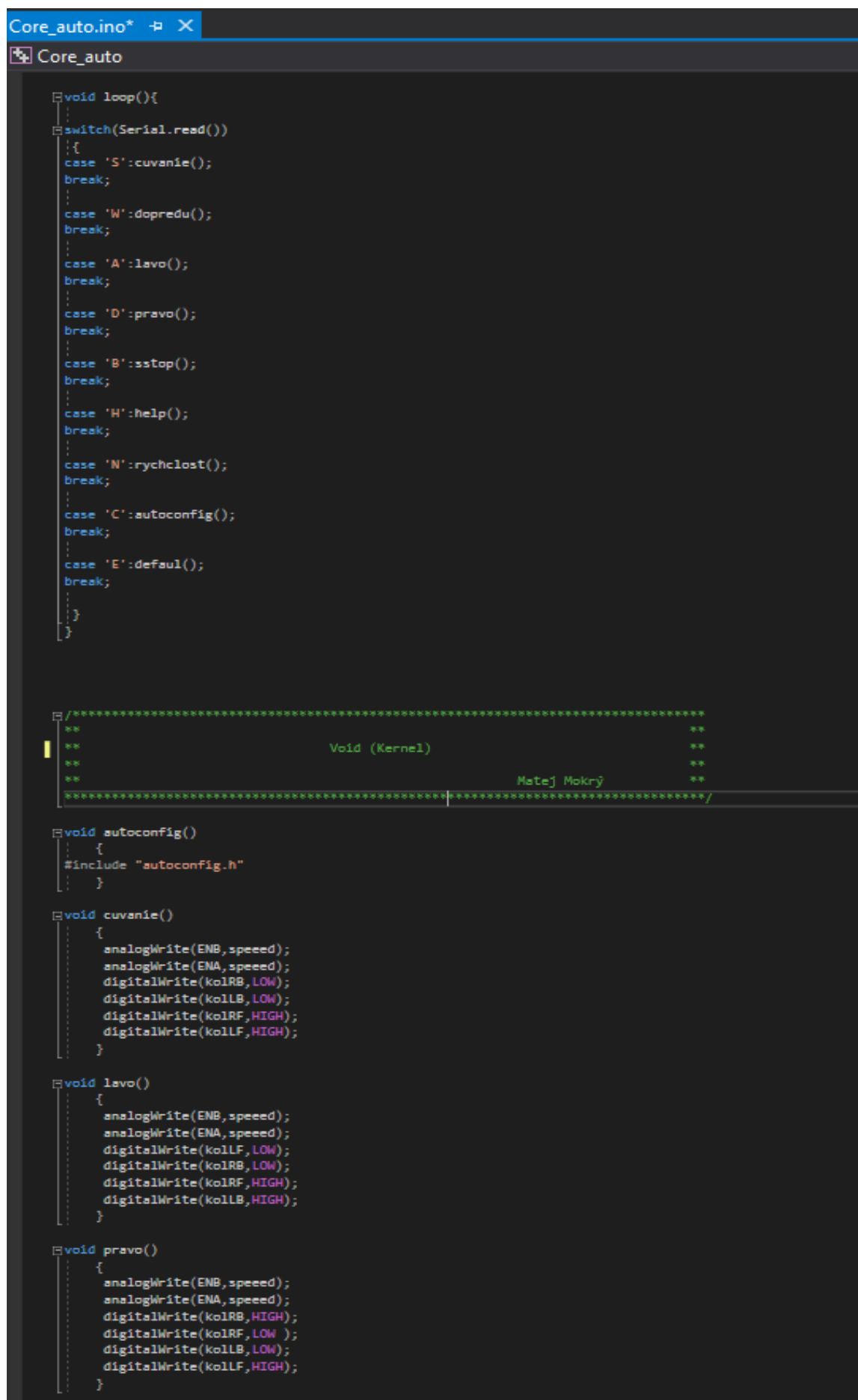


```
Core_auto (Globální rozsah)
/*****
**                                     **
**          help                     **
**                                     **
*****/
Serial.println("");
Serial.println("");
Serial.println("*****");
Serial.println("***");
Serial.println("***  GNU GPL Daskama Jokkie 2018/19    Štart=16.1.2018    Build 0.5    LTS    ***");
Serial.println("***");
Serial.println("***                                     Matej Mokry    ***");
Serial.println("*****");
Serial.println("Základne príkazy : ");
Serial.println("W -doperdu");
Serial.println("A -lavo");
Serial.println("S -cuvanie");
Serial.println("D -pravo");
Serial.println("B -stop");
Serial.println("C -autoconfig");
Serial.println("E -default cfg");
Serial.println("T - Ukáže realny cas ");
Serial.println("Node-red ide na porte :1880");
Serial.println("");
```

Obrázok č. 7 – Help program

3.4 Rozbor programu v Arduino (Kernel)

Ďalší obrázok nám ukazuje jadro kódu. Vozidlo je navrhnuté tak, že arduino počúva príkazy zo Serial portu, zistí o aké písmeno sa jedná a vykoná príslušný void.



```
Core_auto.ino* X
Core_auto

void loop(){
  switch(Serial.read()){
    case 'S':cuvanie();
    break;

    case 'W':dopredu();
    break;

    case 'A':lavo();
    break;

    case 'D':pravo();
    break;

    case 'B':stop();
    break;

    case 'H':help();
    break;

    case 'N':rychlost();
    break;

    case 'C':autoconfig();
    break;

    case 'E':defaul();
    break;

  }
}

/*****
**
**                               Void (Kernel)                               **
**                               Matej Mokry                               **
*****/

void autoconfig()
{
  #include "autoconfig.h"
}

void cuvanie()
{
  analogWrite(ENB,speed);
  analogWrite(ENA,speed);
  digitalWrite(kolRB,LOW);
  digitalWrite(kolLB,LOW);
  digitalWrite(kolRF,HIGH);
  digitalWrite(kolLF,HIGH);
}

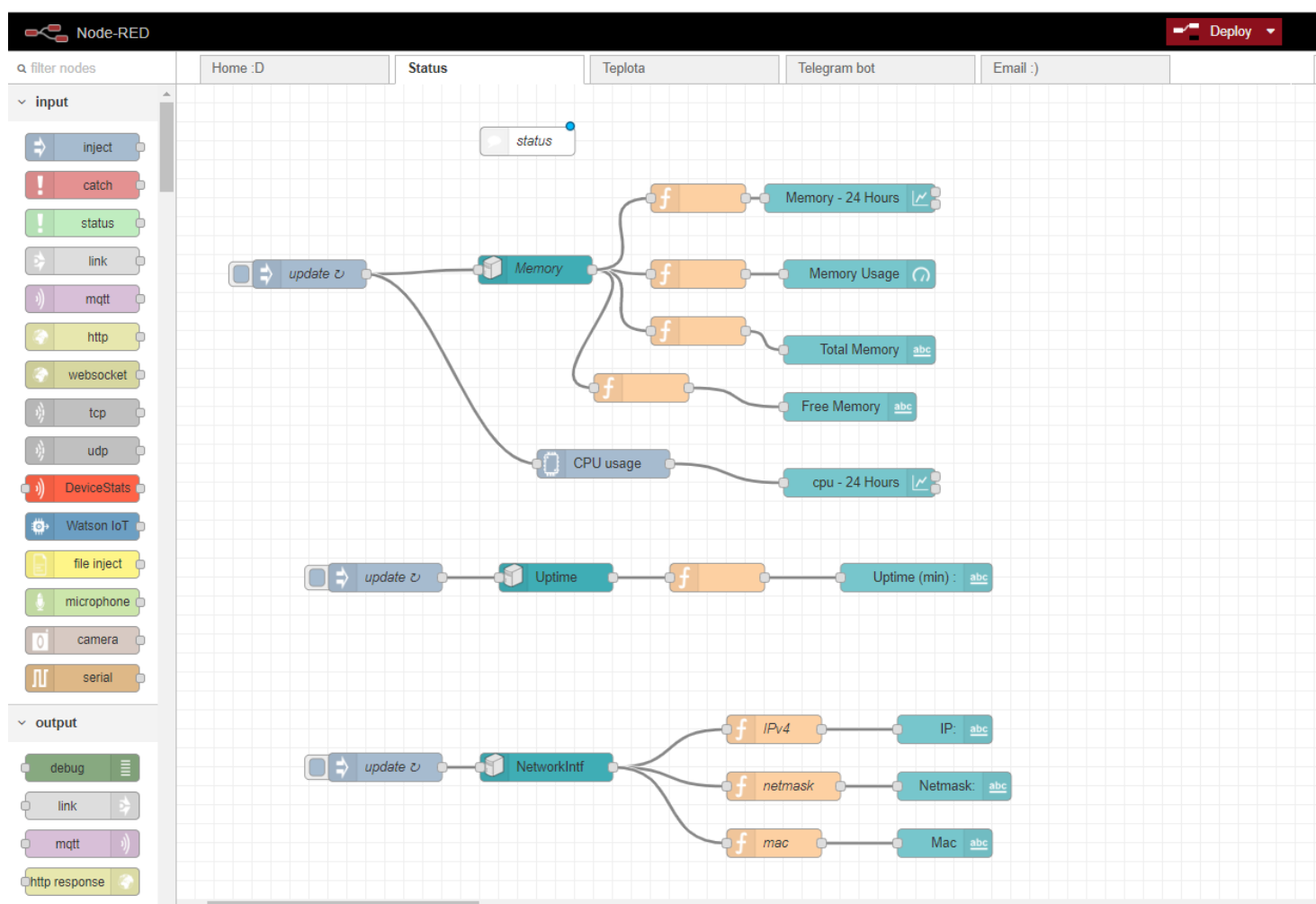
void lavo()
{
  analogWrite(ENB,speed);
  analogWrite(ENA,speed);
  digitalWrite(kolLF,LOW);
  digitalWrite(kolRB,LOW);
  digitalWrite(kolRF,HIGH);
  digitalWrite(kolLB,HIGH);
}

void pravo()
{
  analogWrite(ENB,speed);
  analogWrite(ENA,speed);
  digitalWrite(kolRB,HIGH);
  digitalWrite(kolRF,LOW);
  digitalWrite(kolLB,LOW);
  digitalWrite(kolLF,HIGH);
}
```

Obrázok č. 8 – Kernel

4 Node-red na RPI (Status)

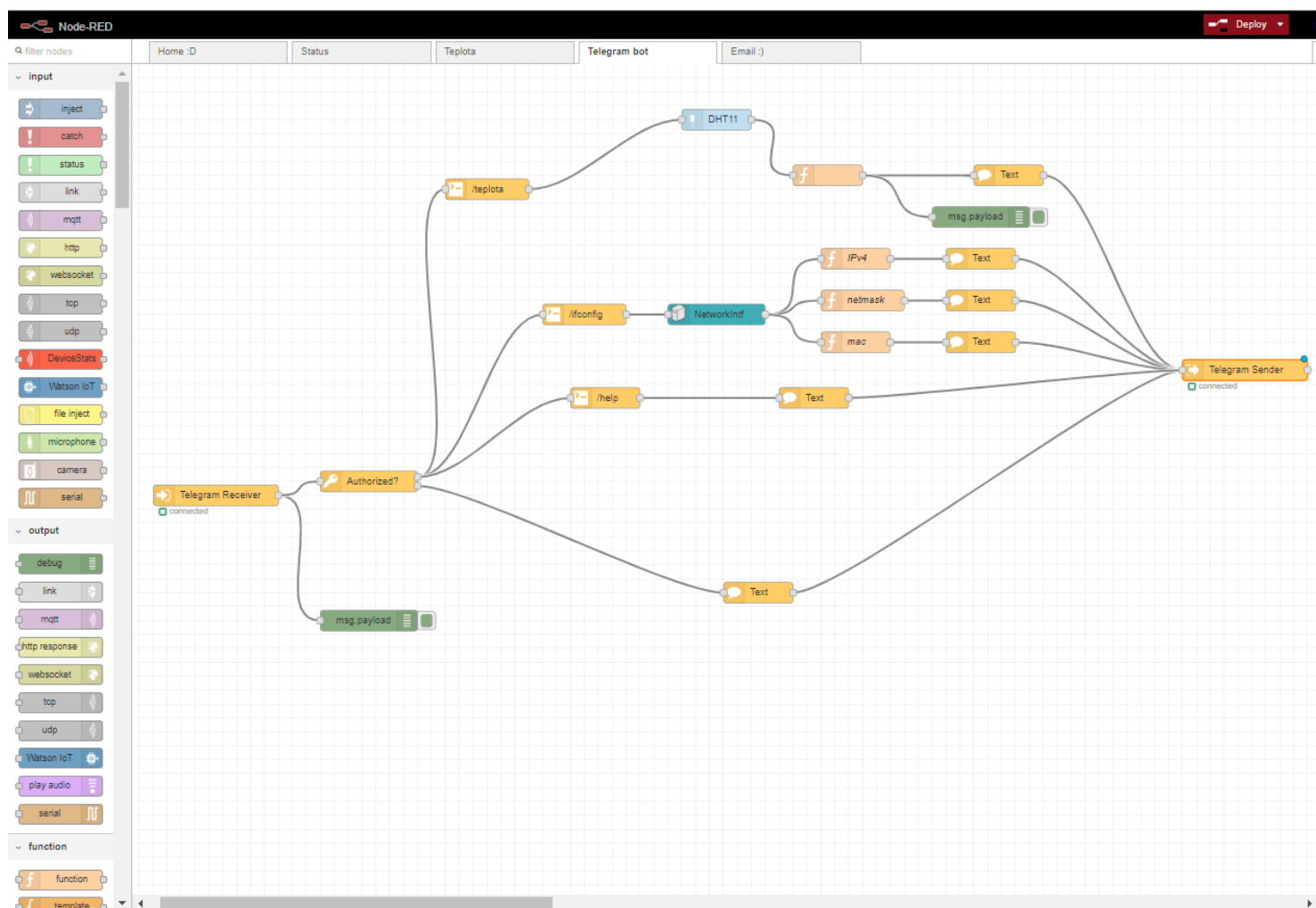
Pre ovládanie RPI3 som použil aplikáciu node-red. Programuje sa blokovo, tzn. programovací jazyk nie je potrebné poznať. Node-red obsahuje viac ako 225 000 modulov, čiže je veľmi variabilný. Pomáha tomu aj jednoduchosť importovania a exportovania svojho projektu. Používal som viacero modulov, ale hlavný je node-red-dashboard, ktorý zabezpečuje zobrazovanie na web. Obrázok nižšie ukazuje ako vyzerá program, ktorý ukazuje základne informácie RPI3.



Obrázok č. 9 – Program zobrazujúci základne informácie RPI

4.1 Node-red na RPI (Telegram bot)

Zmyslom projektu bolo zhotovenie takého vozidla, ktoré je možné ovládať aj prostredníctvom internetu, tak som aktívne využil príkazy cez telegram. Existuje veľa možností napr. Messenger ale najjednoduchším je Telegram. Telegram bot overuje užívateľov, ktorí sú na white liste. Preto ho nedokáže ovládať iný užívateľ.



Obrázok č. 10 – Telegram bot

Záver

Výsledkom môjho projektu je multifunkčné vozidlo, ktoré môžeme ovládať cez mobil, telegram, Web alebo aj cez inteligentný televízor. Vozidlo obsahuje 2 batérie a 1 power banku, ktorá dodáva dostatočnú energiu pre motor, arduino a raspberry. Môžem konštatovať, že projekt bol úspešný. Teraz už prechádzam na debugging a opravovanie chýb. Veľkosť celého programu je 497 kB. Pamäť arduina je vyplnená na 70%. To vyžaduje optimalizáciu programu. Obzvlášť vzniknutý problém, vyžadujúci si svoje vlastné riešenie, bol LCD display. LCD display mal Bug v knižnici, ktorá spôsobovala nesprávne znázornené písmená. Moje vozidlo, jeho konštrukcia a následná práca s ním, mi dali veľa nových skúseností, kde som mohol naplno rozvinúť a aplikovať doterajšie poznatky, nie len z oblasti arduina a ostatných systémov, ale aj vedomosti z elektrotechniky a techniky samotnej.

Zoznam použitej literatúry

<https://www.alza.sk/arduino-uno-rev3-d569244.htm> ; 12.11.2018

https://en.wikipedia.org/wiki/Raspberry_Pi ; 12.11.2018

<https://nodered.org/> ; 12.8.2018

<https://www.arduino.cc/en/Serial/Available> ; 5.8.2018

<http://forum.arduino.cc/index.php?topic=39965.0> ; 6.8.2018

<https://flows.nodered.org/node/node-red-contrib-chatbot> ; 10.11.2018

<https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/> ; 10.3.2019