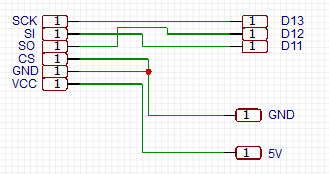
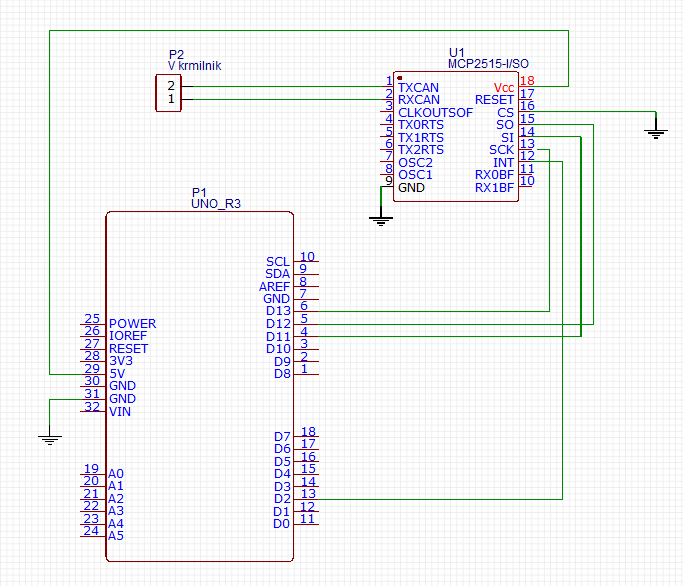
# **#rast** – maketa el. avtomobila - NALOGE

## Arduino

### Povezava med MCP2515 CAN/SPI modulom in Arduino Uno

1. Povežite pine Arduina Uno s pini modula MCP2515 CAN/SPI - izdelajte ustrezne priključke:

* 1x3p za priklop na Arduino,
* 1x6p ženski za MCP2515).
* 1x1p GND ženski,
* 1x1p 5V moški.

Vcc – 5 V; GND – GND; CS – GND; MISO – pin 12; MOSI – pin 11; SCK – pin 13

**Opravljeno**

1. Ustrezno podaljšajte CAN vodilo iz krmilnika motorja (uporabite dve žici in božirki) ter ga preko uvodnice vstavite v električno omarico. Na koncu namestite 2-pinski ženski konektor, ki paše na H-L moška pina modula MCP2515.

**Opravljeno**

1. Ustrezno sprogramirajte kodo Arduina, da bo lahko preko CAN vodila prejemal podatke iz krmilnika motorja. Primer inicializacija povezave SPI in CAN:  
   [1] <https://www.14core.com/wiring-the-mcp2515-controller-area-network-can-bus-diagnostics/>

[2] <https://www.electronicshub.org/arduino-mcp2515-can-bus-tutorial/a>

**Ni opravljeno**

### Vklop/izklop BMS-a z relejem z menjalnim stikalom na panelni plošči.

BMS se vklopi, ko je stikalo sklenjeno 3 sekunde, izključimo ga pa tako, če je stikalo sklenjeno 5 sekund. Za to bomo uporabili dvopoložajno stikalo: Arduino bo prebral stanje in krmilil rele, ki bo ustrezen čas sklenil »stikalo« BMS-a. Pozicija I – vklop BMS-a, pozicija II – izklop BMS-a. Stikalo ima tudi nevtralno pozicijo, katerega bomo vedno postavili takoj, ko bomo BMS vključili oz. izključili. Ker pa bi uporabnik na to pozabil, bi vključenost stikala na poziciji I ali II ne vplivala na ponovno aktivacijo releja za vklop/izklop BMS-a.

Na dig. vhoda (pin 6 in 7) Arduina povežemo skrajna priključka dvopoložajnega stikala (uporabite 2-polni moški konektor), sredinski priključek povežite na GND (0 V). Arduino pinout lastnost naj bo PULL\_UP, torej ga proži GND brez pull-up uporov. V programu ustrezno definirajte oba digitalna pina kot vhod.

IN vhod releja priključite na pin 4 Arduina in ga definirajte kot izhod. Priključek DC+ releja vežete na +12V, DC- pa na GND. Na drugi strani releja povežete COM ter N.O. na priključke tipke BMS-a. Izvorno tipko BMS-a ohranimo!

vklop GND izklop

**Opravljeno**

**Izdelava kode/programa:**

Ko uporabnik stikalo postavi na pozicijo I, Arduino vključi BMS – izhod postavi na visok nivo za 3 sekunde in ga nato postavi nazaj na nizki nivo. Ponovni vklop ni možen, dokler nista oba vhoda (pozicija I in II) na nizkem nivoju!

Ko uporabnik stikalo postavi na pozicijo II, Arduino izključi BMS – izhod postavi na visok nivo za 5 sekund in ga nato postavi spet na nizki nivo. Ponovni izklop ni možen, dokler nista oba vhoda (pozicija I in II) na nizkem nivoju!

**Opravljeno**

### C:\Users\vasja.markic\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\DB22575F.tmpVklop luči

Z ustrezno tipko na komandni plošči krmilimo prednjo in zadnjo luč makete avtomobila. Uporabimo ustrezni vhod na Arduino in dva izhoda za (direkten) vklop/izklop prednje ter zadnje LED luči. Na stikalu en priključek vežemo na GND, drugi pa na digitalni vhod Arduino.

**Opravljeno**

### Povezava do Nextion display na Arduino

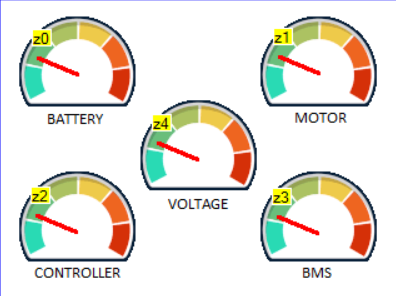
Povežite žice iz Nextiona v MCU omarico :

* ČRNA – GND (na vijačno zbiralko)
* RUMENA in MODRA – Rx pin 0 in Tx pin 1 Arduino, uporabite moški 2 polni kontakt
* RDEČA – na 5 V, uporabite moški kontakt in ga vstavite v že uporabljeni 4-p konektor

**Opravljeno**

### Programiranje Nextion display-a in Arduina

Vrednosti zajete iz temperaturnih senzorjev in optičnega linearnega ojačevalnika (napetost baterije) bomo prikazali kot grafične vrednosti – Gauges – na zaslonu, spodaj levo je primer:

**Pozor** – prikazovalnik za napetost obarvajte po sledečem vrstnem redu (z leve proti desni): RDEČA, ORANŽNA, RUMENA, ZELENA, MODRA. Dodajte še napisa HIGH in LOW.

Uporabimo tako program Slikar za ustvarjanje **png** datoteke kot orodje Gimp za pomanjševanje slik iz spleta (tako ne kvarimo kvalitete pri pretvorbi dobljene slike). Velikost slike ozadja mora biti točno takšno, kot je velikost strani Nextion zaslona, to je 480 x 320.

Postopek v Nextion:

1. Zaženemo Nextion Editor in ustvarimo nov projekt, izberemo ustrezni zaslon (BASIC) NX…035, pod DISPLAY spremenimo orientacijo na 90 Horizontal.
2. V levem oknu *Pictures* dodamo našo ustvarjeno sliko ozadja s petimi grafičnimi zasloni (kliknemo gumb +).
3. V desnem oknu *Page* kliknemo objekt page0 in v spodnjem oknu *Atribute* objekt **sta** spremenimo v **image.** Dvokliknemo v spodnje prazno polje poleg **pic** in izberemo našo naloženo sliko ozadja. Sedaj bi morali videti v glavnem oknu *Display* naloženo fotografijo na zaslonu.
4. V levem oknu *Toolbox* kliknemo **Gauge** in ustrezno postavimo kazalec (po potrebi pomanjšamo/povečamo) na prvi grafični prikazovalnik. V oknu *Atribute* mi spremenimo še barvo in debelino kazalca (**pco** in **wid**). Obvezno mu moramo še spremeniti ozadje: lastnost **sta** spremenimo v **crop image**, v prazno polje **picc** pa izberemo enako sliko ozadja kot pri točki c.
5. Ustvarjeni gauge copy/paste na še preostale grafične prikazovalnike. Zapišite si vrednost **id** in **objname** vsakega prikazovalnika/gauge, saj jih boste tako naslavljali v programski kodi Arduino.
6. Kliknite **Compile**.
7. Kliknite File 🡪 Open build Folder in prekopirajte nastalo binarno datoteko na mikro SD kartico (s pomočjo in adapterja). Na kartici mora biti samo ENA datoteka! Ustavite mikroSD kartico v (ugasnjen) Nextion display, priklopite napetost (5V, GND), počakajte, da Nextion naloži datoteko, izklopite napajanje, PRITISNITE mikroSD kartico navznoter, da jo lahko izvlečete ven in ponovno priključite napajanje Nextion-u. Sedaj bi morali videti vašo sliko na Nextion zaslonu.
8. Sedaj priključite tudi RX in TX pina iz Nextiona na Arduino pin 0 in 1 (Rx gre na Tx in Tx gre na Rx!).

Programiranje v Arduino:

1. Na spletu poiščite ITEADlib knjižnico in jo namestite v Arduino okolje.
2. Na začetku kode dodajte naslednje knjižnice:

#include <NexConfig.h>

#include <NexPicture.h>

#include <NexGauge.h>

#include <Nextion.h>

#include <NexTouch.h>

#include ''Nextion.h''

Nato deklarirate grafične prikazovalnike:

NexGauge z0 = NexGauge(0, 1, ''z0'');

NexGauge z1 = NexGauge(0, 2, ''z1'');

Itd ..

Kjer je prvi parameter v oklepaju številka strani, drugi parameter **ID** objekta, tretji pa **objname**.

1. V razdelki programa void setup() inicializirajte Nextion zaslon z ukazom:

nexInit();

1. V glavni zanki void loop() vrednosti preko UART prenašamo na Nextion z ukazom:

z0.setValue(T3);

Torej prvi prikazovalnik (z0) naj pokaže vrednost spremenljivke T3. Pozor – če je T3 na primer 60 °C, se bo prikazovalnik pomaknil na kot 60 ° 🡪 potrebo je preskalirati vrednost, na primer temperatura 90°C bo na prikazovalniku imela odklon 180°!

1. Podobno uredite prikaz za ostale temperature.
2. Pri prikazu napetosti (z4) želimo, da pri 42 V je kazalec na 0°, pri 58 V pa na 180°!

### Vklop ventilatorja ob prekoračitvi temperature

Dopolnite kodo tako, da ob prekoračitvi temperature 40 °C se vklopi ventilator (krmiljen preko rele FAN) za hlajenje baterij. Ko temperatura pade pod 40°C, se ventilator ugasne.

**Opravljeno**

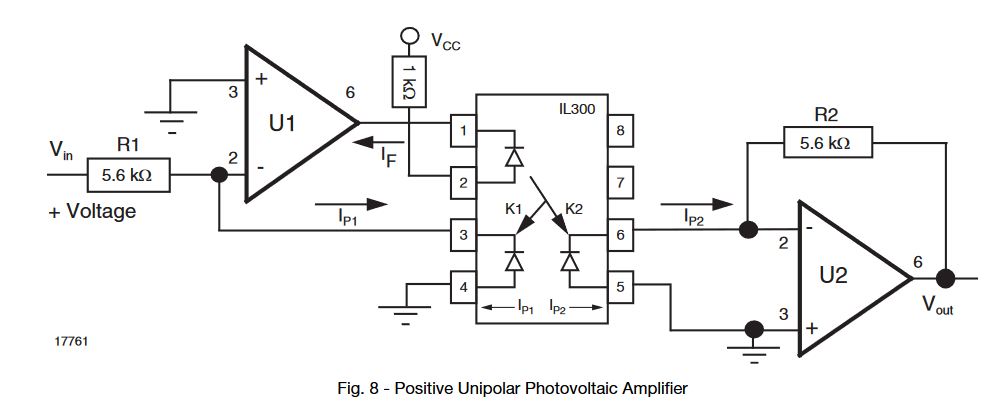
### Merjenje napetosti na baterijskem modulu

Baterijski Li-Ion modul mora biti galvansko ločen od ostalih elektronskih komponent (MCU, 12V sistem, Raspberry ..), tudi GND glavne baterije je galvansko ločen od ostalih ozemljitev!

[1] <https://www.digikey.com/en/articles/techzone/2012/dec/linear-optical-isolation-for-safe-sensor-operation>

[2] <https://www.vishay.com/docs/83708/appnote50.pdf>

Uporabimo linearni opto-izolator IL300 v povezavi z operacijskim ojačevalnikom LM393:



Vezje izdelajte na testni ploščici; uporabite ustrezne elemente in tudi konektorje za priklop pinov – za vhodne (Vin, Vcc, GND1) moške pine, za izhodne (Vout, GND2) pa ženske. Prejšnja sonda, ki vsebuje delilnik napetosti in tri žice je potrebno predelati – uporabimo le žico za GND (na GND2) ter žico, ki pelje na analogni vhod Arduina A0. Le tega boste priklopili na Vout (na shemi).

B+ glavne baterije (iz glavne varovalke) bomo priključili na Vin , za Vcc pa bomo izdelali uporovni delilnik, ki bo znižal napetost (med 10 V in 30 V). Prav tako mora levi operacijski ojačevalnik LM393 biti napajan z isto Vcc napetostjo.

Na pin 2 mora biti upor takšne vrednosti, da bo IF reda 10 mA (max 60 ma), torej je R=Vcc/10mA. Napetost VF na LED diodi je tako 1.25 V.

Postopek izračuna uporov R1 in R2 pa po sledečih korakih:

1. Ugotovite lastnosti OO (Iout) ter IL300 (K1, K2, K3, Vin, G – vaš željeni faktor ojačanja)
2. Izračunamo Ip1 = K1×Ioutmax
3. R1=Vin/Ip1
4. R2=(R1×G)/K3, če je G=Vout/Vin

**Ni opravljeno**

## Krmilnik in motor

### Osnovne lastnosti krmilnika

Izdelajte/prevedite dokumentacijo za nastavitve in prvi zagon motorja s krmilnikom – prav tako na PC namestite ustrezen program za diagnostiko motorja, ki ga boste upravljali preko USB/RS232 vodila.

**Ni opravljeno**

[1] <https://kellyev.com/shop/kls-mn/>

### Pedal za zavoro

Uporabite potenciometer ter pozicijsko tipko za simulacijo zaviranja. Analogni signal iz krmilnika je povezan na potenciometer poleg napajanja in GND, za pozicijsko stikalo, ki javlja pritisnjen pedal pa dve žici iz krmilnika (napajanje in digitalni vhod). Uporabite 5-polni IP67 konektor.

**Ni opravljeno**

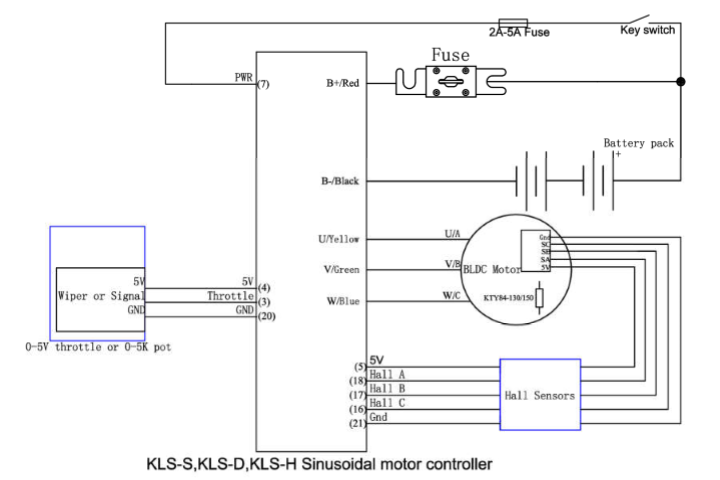
### Prvi zagon motorja

#### Identification angle operation

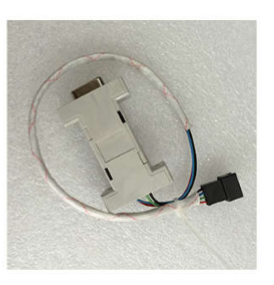
Na začetku moramo izvesti detekcijo kota preden zaženemo motor. To storimo s pomočjo programa na PC-ju ali ali aplikacije na pametnem telefonu/tablici. Privzeta izbira senzorjev motorja so halove sonde, izbira je 2 *(speed sensor type).*

Postopek:

1. Na motor ne montirajte nobenih gredi, menjalnika ali koles (bremena). Krmilnik je povezan na baterje, stikalo za vklop, motor in pedal za pospeševanje (glej shemo):



1. Povežite krmilnik z PC-jem (ali tablico) z USB na RS232 kablom. Med krmilnikom in RS232 kablom je vmesni adapter SM-4ADB9(RS232). Če uporabljate tablico ali telefon uporabite Z-TEK USB to RS232 kabel.

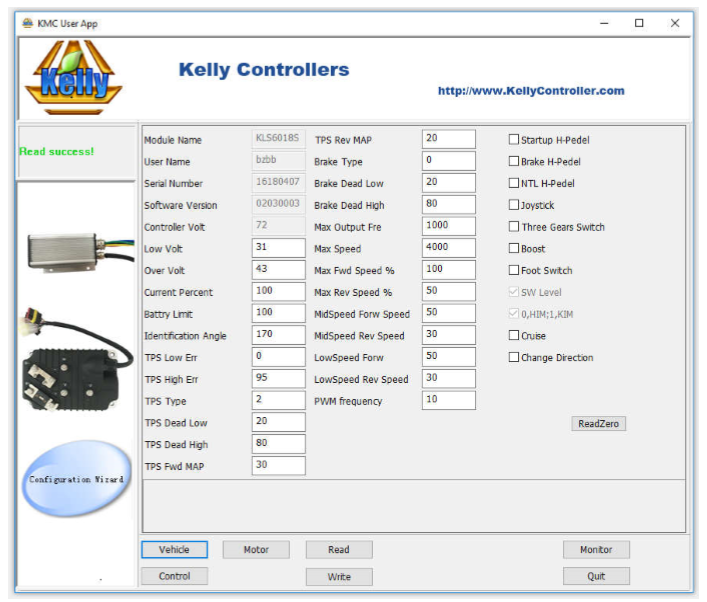




Z-tek (za telefon) USB na RS232 (za PC) SM-4ADB9(RS232) (na krmilnik)

1. Naložite ustrezen program na PC ali telefon/tablico za vaš krmilnik: [www.kellycontroller.com/support.php](http://www.kellycontroller.com/support.php)
2. Vključite krmilnik tako, da bo krmilnik vklopljen (pin 7 na napajalno napetost) ter priključka B- in B+ sta pod baterijsko napetostjo (vklopimo glavni kontakor).   
   **POZOR! Ali imata kontaktor in zasilno stikalo vzporedno vezana upora? Preverite!**

Zaženite program za diagnostiko krmilnika. Kliknite na gumb *Read.* Pojavila se bo prvo okno programa:



Preverite vrednost parametra *Identification angle*. Pravilna vrednost parametra je 85. ta vrednost pomeni, da je motor že opravil detekcijo smeri s krmilnikom, vendar je to tovarniška nastavitev. Vaš motor je drugačen od testnega, zato morate detekcijo ponovno opraviti.

1. V polje *Identification anlge* vpišite vrednost 170 in kliknite *Write.* Program bo javil okno z obvestilom, da je bilo zapisovanje uspešno (*Write operation successful*). Zaprite program z gumbom *Quit*. Izključite napajanje krmilnika.
2. Počakajte nekaj sekund in ponovno vključite napajanje. Piskač bo javljal neprekinjen zvok in motor se bo začel vrteti v različne smeri. Tako krmilnik opravi detekcijo smeri/kota.
3. To traja 2 do 3 minute, zato počakajte.
4. Ko je detekcija kota/smeri končana, piskač preneha s piskanjem. Nato bo piskalo v zaporedju, ki pomeni neko napako. Ponovno zaženite program – na zaslonu boste videli sporočila o napakah.
5. Ponovno ugasnite krmilnik. (Zaprite tudi program). Počakajte nekaj sekund, preden ponovno še enkrat prižgete krmilnik.
6. Ponovno vklopite krmilnik in program. V polju *Identification Angle* bi moralo ponovno pisati vrednost 85. V tem primeru je detekcija kota/smeri uspela. Krmilnik je sedaj pripravljen za uporabo in krmiljenje motorja.
7. V kolikor ni *reset error code*, bo vrednost ostala pri 170. Krmilnik še zmeraj ugotavlja detekcijo smeri.
8. Nikakor ne poskušajte sami zapisati vrednost 85 v to polje, ali resetirati napajanje, če ne vidite kakršne koli sporočila o napakah!
9. Če pisanje traja več kot 5 minut pri detekciji kota in se pojavijo druge napake v programu, preglejte še enkrat povezave (sploh pri senzorjih) in zapišite vrednost 85. ????  
   Vrednost 170 ne sme ostati v primeru, če želite izklopiti napajanje. ????

Sicer bo krmilnik ponovno poskušal opravljati identifikacijo kota/smeri.

Ko se pojavijo okna z napakami, kliknite *Quit* in ponovno začnite s točko 6.

1. Vrednost 170 ne sme biti zapisana preden ponovno vključimo napajanje po uspešno opravljeni identifikaciji.
2. V kolikor smer vrtenja in prava, enostavno omogočite *Change Direction* polje na prvi strani okna programa. Klinite *Write*. Ponovno resetirajte napajanje. Motor se bo nato vrtel v obratno smer od prejšnje.

**Uporabljajte funkcijo *Identify* za motor in za halove sonde v programu!**