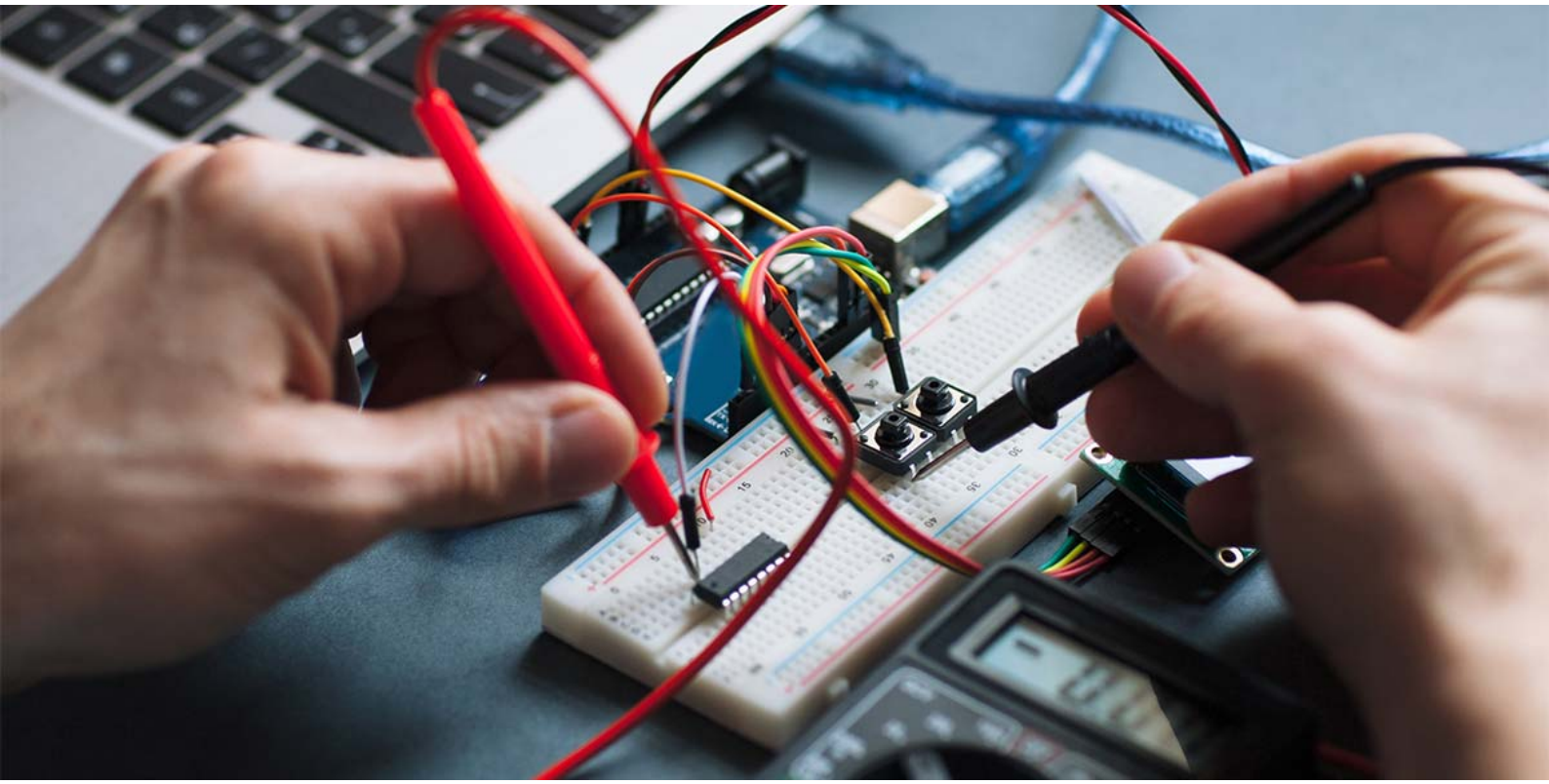
Elektronika z robotiko - VAJE

Skripta za učence

Avthor: Tomaž Kušar

januar, 2022



1 Mikrokrmilnik

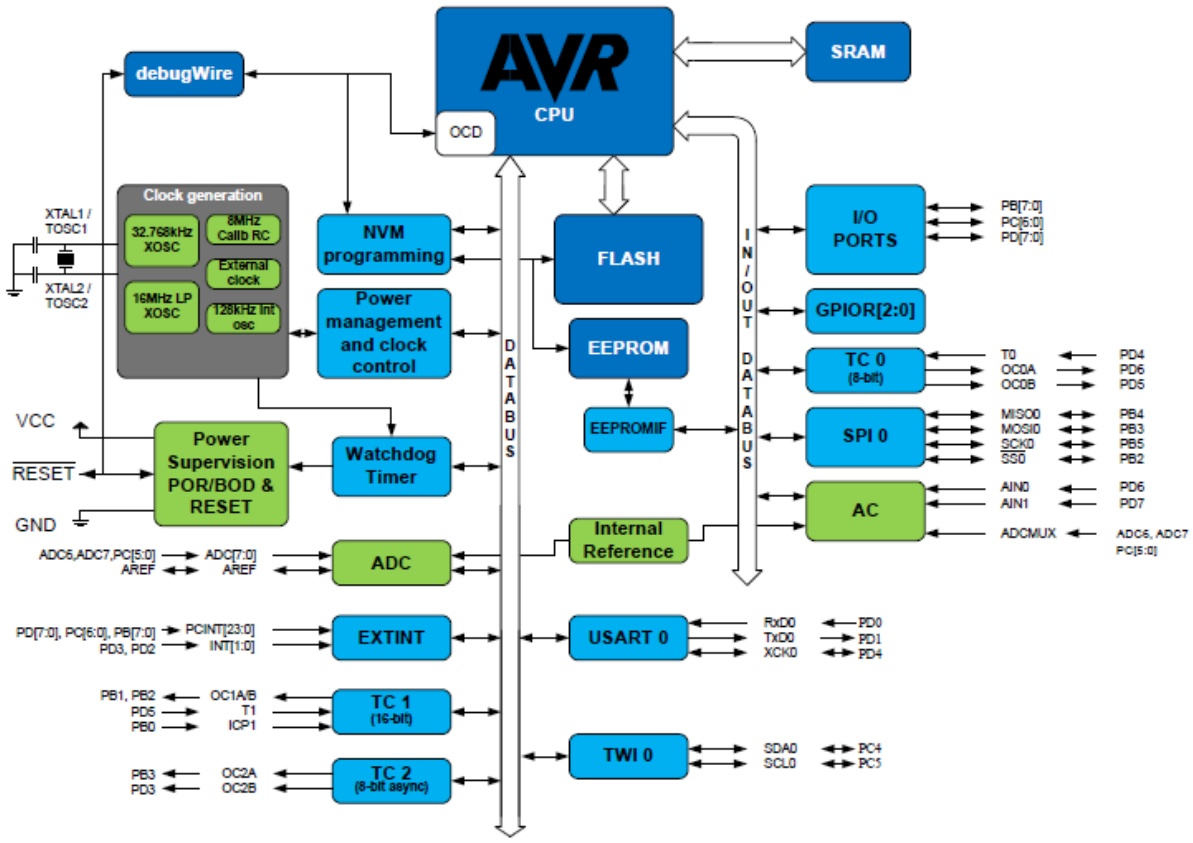
Nepogrešljivi del elektronike dandanes pa je tudi mikrokrmilnik. Je splošno namensko integrirano elektronsko vezje – praktično računalnik v malem. Kljub temu, da je manjši kot računalnik, ima vse enote kot računalnik. V njem so procesor (ALU), začasni spomin za podatke (RAM), spominska enota za program (ROM) itd.

Mikrokrmilnik sestavlja vhodna enota, procesna enota in kontrolna enota. Z njim zaznavamo svet okoli sebe. Signale in fizikalne količine, kot so hitrost, temperatura, zvok, razdalja, itd., zajemamo s posebnimi senzorji (nekatere smo spoznali tekom vaj), ki pretvorijo signale v električne signale, ki jih z mikrokrmilniki lahko beremo, analiziramo.

Izhodni signal, na primer senzorja temperature, je napetost, ki jo preberemo z vhodno enoto mikrokrmilnika. Pri tem uporabimo ukaze za delo z vhodnimi enotami. Nato pišemo program, katerega delovanje je odvisno od temperature (vhodne napetosti). Pri tem uporabljamo ukaze, ki jih uK razume. Podatke shranjujemo v spomin in pri tem uporabljamo spremenljivke. Recimo, da vklopimo lučko, če je temperatura previsoka. Pri tem uporabimo ukaze za delo z izhodnimi enotami.

Slika 1: Mikrokmilnik ATmega32

Shematično notranjost mikrokmilnika prikažemo na naslednji način (Slika 5). Na tej shemi



Slika 2: Shema mikrokrmilnika ATmega328p

lahko opazimo vse enote, ki smo jih omenili že zgoraj, torej centralno procesno enoto (CPU - Central Process Unit), SRAM, FLASH, EPROM, I/O ports (I/O = input/output = vhodne/izhodne enote), AC (Analog Comparator = komparator napetosti), SPI (Serial Peripheral Interface = vmesnik za serijsko komunikacijo), ADC (Analog digital converter = analogno digitalni pretvornik) in še mnoge druge. Torej, mikrokmilnik je integrirano vezje, ki ima zelo odprte možnosti uporabe. Z njim lahko nadomestimo zelo veliko namenskih integriranih vezij, kot so recimo števci, operacijski ojačevalniki, logična vrata itd. Poleg tega pa nam mikrokrmilniki omogočajo lažji razvoj vezij in marsikatera naloga z njim postane zelo trivialna.

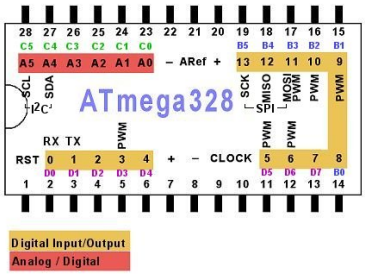
2 Priključki mikrokmilnika

Poglejmo, kako uporabljamo priključke mikrokrmilnika kot izhode ali kot vhode. Preden napišemo program, moramo poznati zgradbo mikrokmilnika. Torej, vedeti moramo, kako so priključki označeni. Namreč, vsak priključek ima svojo oznako in kasneje v programu za vsak tak priključek

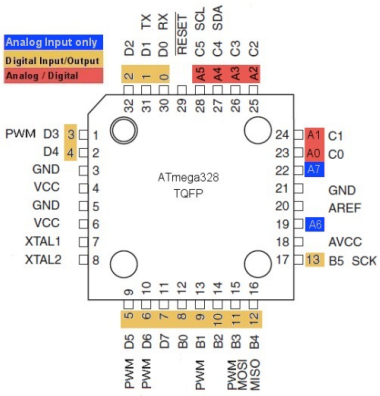
določimo, na kakšen način ga bomo uporabljali.

Mikrokmilniki so seveda v različnih ohišjih. Najbolj priročna ohišja so seveda PDIP (Slika 3), ki jih lahko vstavimo direktno v preizkusno ploščo.

Seveda vsak mikrokmilnik potrebuje za svoje delovanje tudi zunanje elemente, kot so kondenzatorji, kristali, in seveda napajanje ter enoto, s katero lahko mikrokrmilnik sprogramiramo. Da se tem težavam izognemo, bomo mi uporabljali kar vmesnik Arduino NANO, ki ima vse te dodatne elemente že nameščene. Opazimo pa lahko, da je ta čip na Arduinotu tudi v drugačnem ohišju in sicer v ohišju TQFP (Slika 4). Za nas je torej na koncu pomembno, da vemo, kako je ta čip

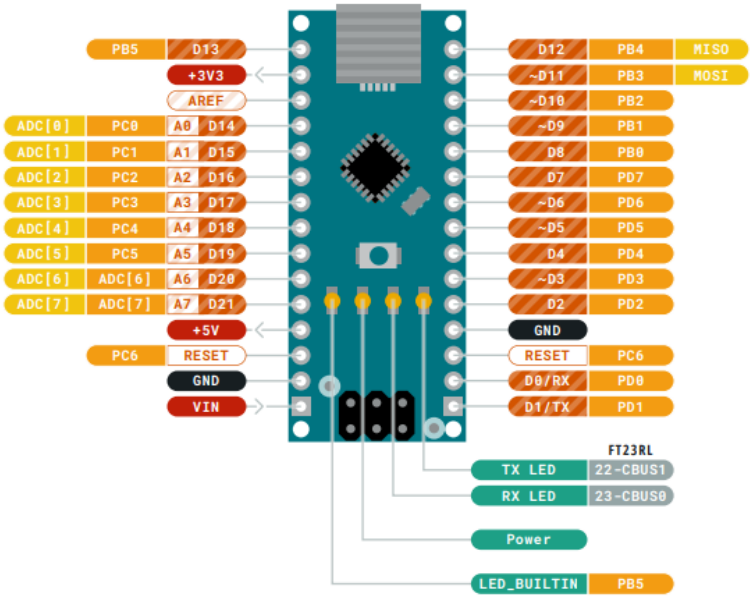


Slika 3: ATmega328p v ohišju PDIP



Slika 4: ATmega328p v ohišju TQFP

povezan s priključki vmesnika Arduino, ki jih lahko potem povezujemo na preizkusni pološči. To je shema, ki bo za nas pomembna pri vsaki vaji (Slika 5).



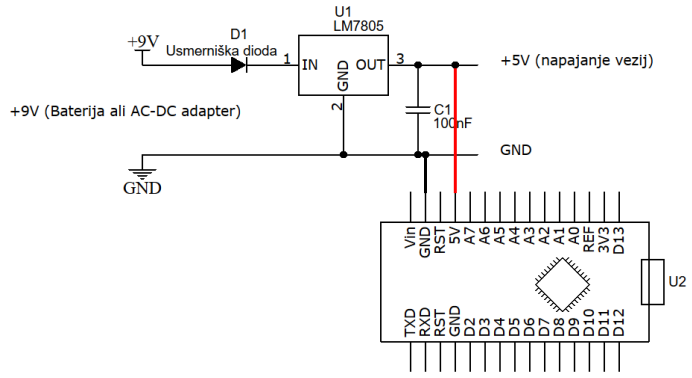
Slika 5: Shema mikrokrmilnika ATmega328p

3 Napajanje mikrokmilnika

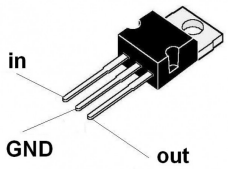
Digitalna vezja običajno priključimo na napetost +5V, sodobna vezja pa delujejo že pri 3.6V ali 3.3V. Tudi naš vmesnik Arduino NANO bomo napajali z napetostjo +5V, ki jo lahko zagotovimo že z USB povezavo z računalnikom. Ker pa bodo naša vezja v posameznih primerih potrebovala napetost 9 V in da bi zmanjšali možnost uničenja krmilnika oziroma računalnika zaradi napak pri vezavah, bomo za začetek sestavili vezje, ki zagotavlja stabilno napetost 5 V.

POTREBUJEMO:

- Prototipno ploščico
- AC-DC adapter 9V
- Napetostni regulator 7805
- Keramični kondenzator C = 100nF
- Usmerniško diodo
- Digitalni voltmeter

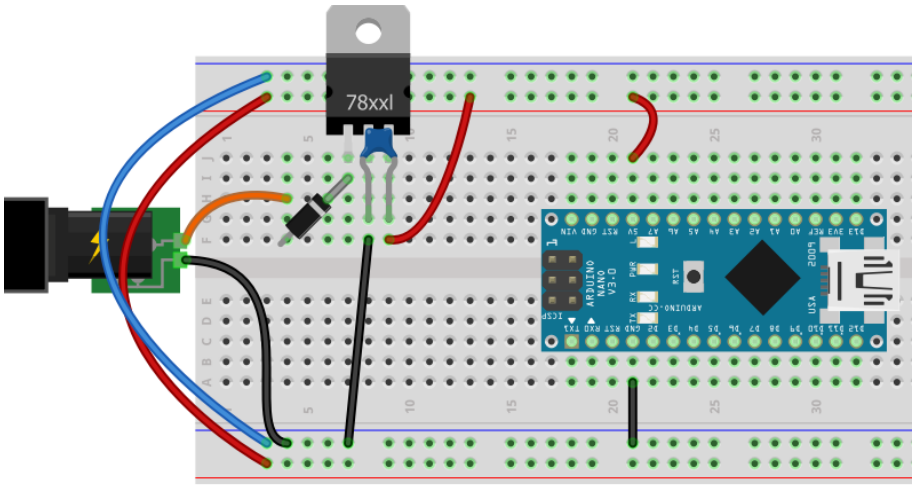


Slika 6: Stabilizator napetosti LM7805 in Arduino NANO



Slika 7: Stabilizator napetosti 7805

Na shemi je vezava stabilizatorja napetosti, ki ga napajaš z okoli 9 V enosmerne napetosti. Uporabiš lahko tudi 9 V baterijo in ne le AC/DC pretvornik. Dioda D1 na priključku stabilizatorja napetosti preprečuje napačni priklop polaritet in s tem uničenje stabilizatorja napetosti. Kondenzator C1 na priključku 3 (OUT) pa zmanjšuje šum napetosti. Na spodnji shemi (slika 8) je izvedba na prototipni ploščici (PP).



Slika 8: Pravilna vezava napajanja mikrokrmilnika preko stabilizatorja napetosti.

3.1 Vklop in izklop svetleče diode

Poglejmo, kako začnemo s programiranjem. Najprej v programu določimo, kakšen mikrokrmilnik programiramo in s kakšnim taktom deluje (vrstici 1 in 2). Odločimo se, kateri priključki bodo izhodi (v našem primeru priključki B) in kateri vhodi (priključki C). Če upoštevamo oznake na vmesniku Arduino NANO, bodo priključki od A0 do A5 vhodni, priključki od D0 do D13 pa izhodni. Torej, za naslavljanje pinov bomo uporabljali oznake, ki so na vmesniku. V ta anmen bomo uporalbjali posebno knjižnico ArduinoNANO.bas Če se v vrnemo na sliko 5 vidimo, da ima vmesnik Arduino NANO LED diodo, ki je povezana s priključkom z oznako B5 oziroma D13. Običajno priključek D13 uporabimo kot izhodni, in z njim vklapljamo in izklapljamo svetlečo diodo, ki je že na vmesniku. Poglejmo kako.

programi/izhodni_prikljucki.bas

```
1  $regfile ="m328pdef.dat"
2  $crystal =16000000 '16Mhz
3  $baud =115200
4  $include "ArduinoNANO.bas"
5
6  utripaj:
7     do
8        D13 =1
9        waitms 500
10       D13 =0
11       waitms 500
12    loop
13
14 end
```

Vse kar vpišemo med besedama `Do-Loop`, se ponavlja v neskončnost. To programsko strukturo imenujemo neskončna zanka. Običajno je glavni program mikrokrmilnika vedno napisan v

neskončni zanki. Zkaz `Waitms x` zadrži izvajanje programa za x milisekund. Lahko pa uporabimo tudi ukaz `Wait x` (za sekunde) ali `Waitus x` (za mikrosekunde). Ukaz `end` v programu določa konec programa. To vedno zapišemo na začetku pisanja programa.

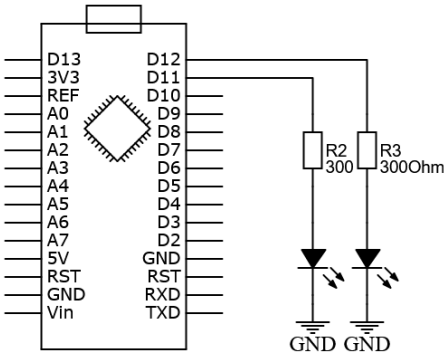
NALOGA: Preizkusi zgornji program in opiši kaj se dogaja. Potem spreminjaj vrednosti pri ukazih `Waitms` in opazuj kaj se zgodi, če je pri enem od ukazov `Waitms` vrednost 0.

Ponovimo. Z ukazom

```
1 D13 =1
2 D13 =0
```

torej določimo stanje na digitalnem izhodu priključka D13. Če Zapišemo vrednost 1, bo na priključku D13 (PORTB.5) visok napetostni potencial (+5V, ali +3,3V, odvisno od napajalne napetosti), če pa napišemo vrednost 0, bo na izhodu D13 ničelni napetostni potencial (0V).

NALOGA: Sestavi vezje, kot ga prikazuje spodnja shema. Zgornji program spremeni in dopolni tako, da bosta ledici izmenično utripali na vsako sekundo. Ko končaš, pokliči učitelja.



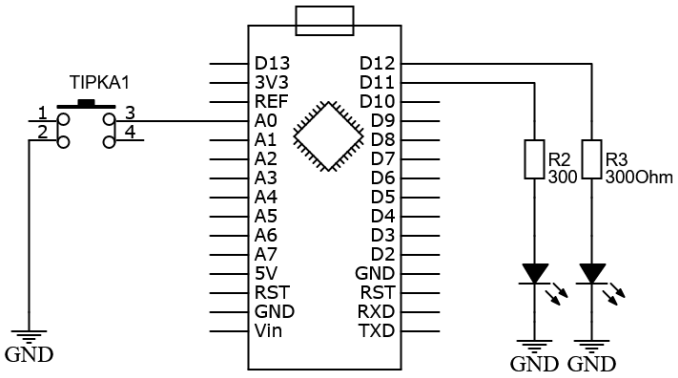
Slika 9: Ledici povežemo s priključkoma D11 in D12

3.2 Branje tipke - digitalni vhod

Tipko povežemo z vhodnim priključkom. Naši vhodni priključki imajo oznke od A0 do A5. Preden torej napišemo program za branje tipke, potrebujemo vezje.

NALOGA: Sestavi vezje, kot ga prikazuje spodnja shema oziroma, dopolni vezje iz prejšnje naloge.

NALOGA: Ko si vezje sestavil, preizkusi spodnji program. Torej, dopolniti moraš prejšnji program. Opazuj, kaj se dogaja, ko pritisneš tipko.

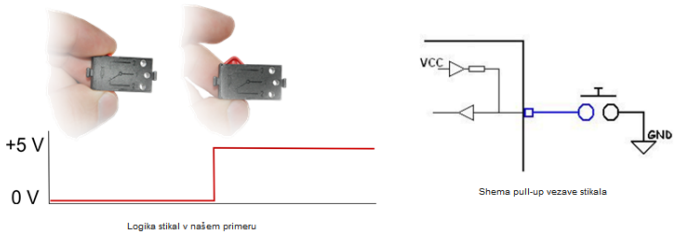


Slika 10: Ledici povežemo s priključkoma D11 in D12

programi/digitalni_vhod.bas

```
$regfile ="m328pdef.dat"
$crystal =16000000 '16Mhz
$include "ArduinoNANO.bas"
pullupa0 =1
   do
      if A0 =0 then
         D13 =1
      else
         D13 =0
      end if
   loop
   end
```

Ukaz A0, nam vrne vrednost 0 ali 1, odvisno od napetosti ki jo imamo na vhodnem priključku A0. Če dobro pogledamo shemo na sliki 10 vidimo, da ko tipko sklenemo, bo na vhodu ničelni napetostni potencial. Kakšno stanje pa imamo na vhodu, če tipka ni sklenjena? Ja, z vezavo pullup smo zagotovili logično stanje 1,torej visok napetostni potencial (+5v). To zagotovimo s stavkom `puulupa0 = 1`. Če želimo na enak način vezati tpke na ostalih vhodih od A0 do A5, uporabimo torej podobne ukaze (`pullupA1 = 1`, `pullupa2 = 1` itd). Praktično to pomeni, da se priključek preko upora poveže z napetostnim potencialom +5V (slika 11). Poglejmo si še



Slika 11: Pull-up vezava

programsko strukturo pogojnega oziroma odločitvenega stavka, ki smo ga uporabili v zgornjem

programu. Ta je lahko zapisan na več načinov:

```
Do
  If nek_pogoj then
  'naredi to, ce je pogoj izpolnjen
  else
  'ce pogoj ni izpolnjen naredi to
  end if
Loop
```

Drugi način je, če nimamo alternativne možnosti pri neizpolnjenem pogoju in želimo na primer, program preusmeriti drugam. V tem primeru je odvisni stavek v eni vrsti in ga ne potrebujemo zaključit z `End if` ukazom.

```
Do
  If nek_pogoj then goto utripaj
Loop

utripaj: 'podprogram utripaj, na koncu imena je dvopicje
Do
   D13 =not D13
   Waitms 200
Loop
```

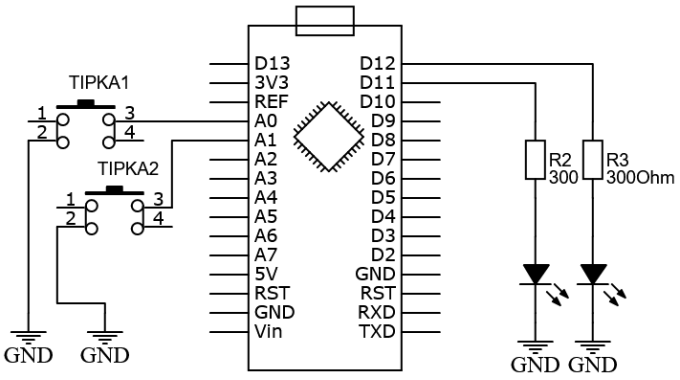
Imamo pa tudi možnost preverjanja več pogojev. Torej, preverimo če je izpolnjen prvi pogoj, če ta ni izpolnjen, preverimo drugega, ali tretjega in če noben od teh ni izpolnjen, zapišemo, kaj se zgodi v primeru, ko noben pogoj ni izpolnjen.

```
Do
  If pogoj_1 then
  'naredi, ce je izpolnjen pogoj 1
  elseif pogoj2 then
  'naredi ce je izpolnjen pogoj 2
  else
   'naredi, ce noben od pogojev ni izpolnjen
  End if
Loop
```

NALOGA: Zgornji program spremeni in dopolni tako, da bo svetila LED na priključku D12, ko boš tipko pritisnil, in LED na priključku D11, ko boš tipko spustil.

NALOGA*: Vezju dodaj še eno tipko in jo priključi na vhod A1, kot kaže shema na sliki 12. Dopolni in spremeni program tako, da bo svetila led na priključku D13, če bosta sklenjeni obe tipki hkrati, če bo sklenjena samo tipka na vhodu A0, bo svetila LED na D12, če pa bo sklenjena samo tipka na A1, bo svetila samo LED na D11.

NAMIG k nalogi: pogoje lahko tudi združuješ na naslednji način:



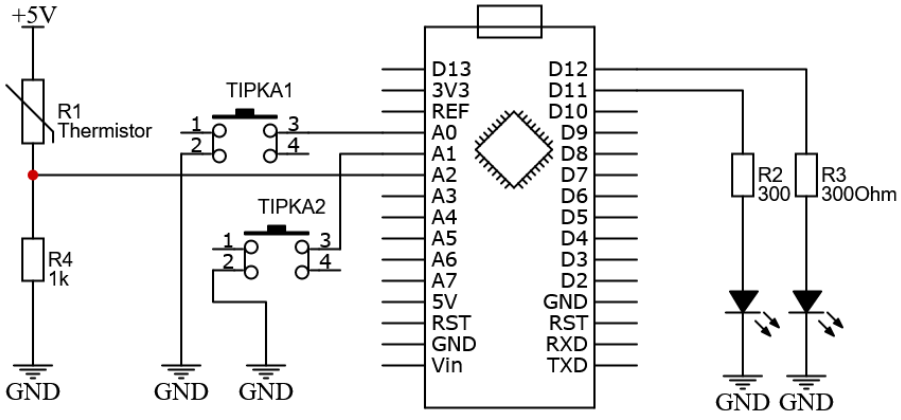
Slika 12: Dve tipki na vhodih

```
If A0 =0 and A1 =0 then
```

4 Temperaturni senzor - analogni vhod

Poleg digitalnih vhodov in izhodov ima mikrokmilnik tudi ANALOGANE VHODE. To so vhodi, na katerih lahko zaznavamo različne vrednosti napetostnih potencialov od 0 do +5V. Analogni vhod pretvori napetost v številčno vrednost in sicer od 0 do 1023. Napetostni potencial +5V torej predstavlja vrednost 1023, napetostni potencial 2,5V vrednost 512 itd....

NALOGA*: Sestavi vezje kot ga prikazuje shema na sliki 13.



Slika 13: Napetostni delilnik s termistorjem povezan na analogni vhod

POMNI: Na vhodih, ki jih uporabljaš kot analogne, ne smeš imeti nastavljene pull-up vezave!

Sedaj pa preizkusi spodnji program. V terminalno okno izpisuj vrednosti analognega vhoda

A2 in opazuj kaj se dogaja, če trmistor segrevaš z roko

programi/analog_vhod_temperatura.bas

```
$regfile ="m328pdef.dat"
$crystal =16000000 '16Mhz
$baud =9600
$include "ArduinoNANO.bas"

pullupa0 =1

dim povisana_temp as byte 'uvedemo novo spremenljivko
povisana_temp =0 'njeno vrednost nastavimo na 0

   bitwait A0,  reset
   do
      If AnA2 >500 then povisana_temp =1
      if povisana_temp =1 then
       d11 =1
      else
       d11 =0
      end if
   loop
   end
```

Pred uporabo terminala moraš najprej nastaviti določene parametre. V meniju kliknemo ikono (označena s puščico na sliki 14). Odpre se modro okno, kjer kliknemo v meniju na terminal in izberemo nastavitve. Potrebno je nastaviti ustrezna vrata (COM port) in hitrost prenosa. Pokliči učitelja, da ti pokaže. Program preizkusiš tako, da imaš odprt terminal, in klikneš na tipko povezano na vhod A0. V terminalnem oknu se ti začnejo izpisovati vrednosti.

Pojasnimo še dva ukaza, ki smo jih uporabili na novo. Prvi je:

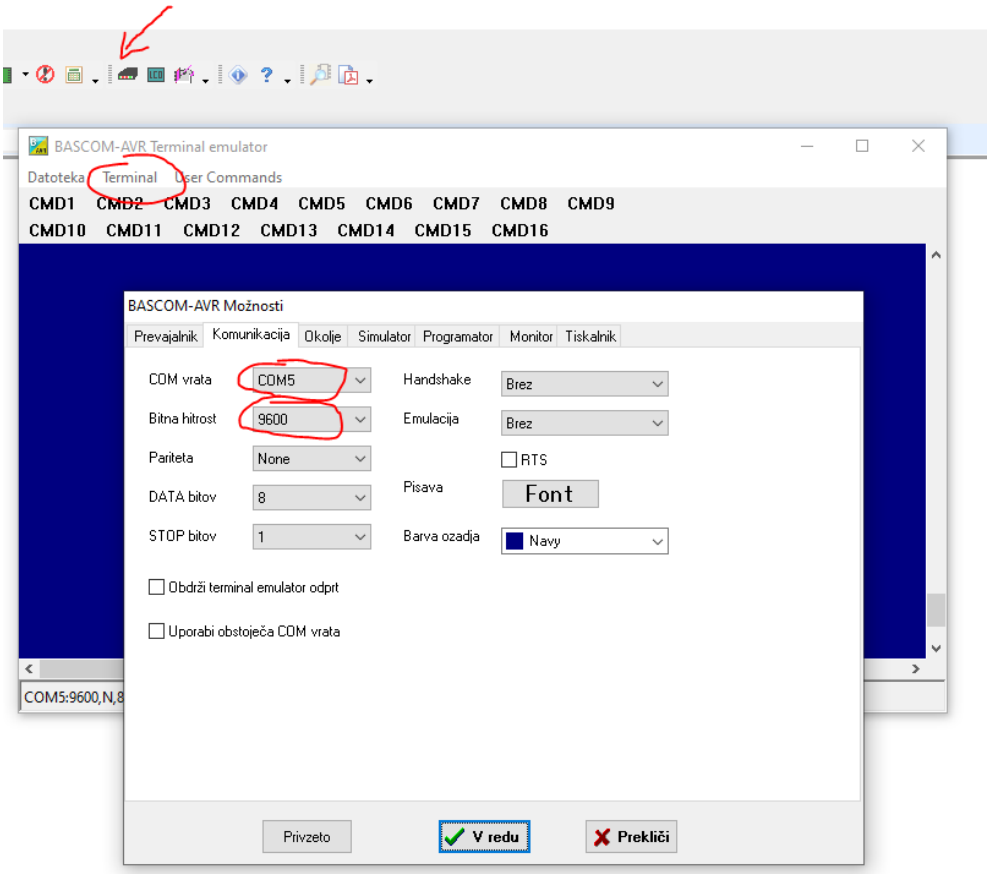
```
Bitwait A0,  reset 'cakamo da nekdo pritisne tipko
```

S tem ukazom čakamo na mestu toliko časa, da bo na vhodu A0 napetostni potencial 0V, torej, da nekdo stisne tipko A0. Če bi želeli, da čakamo tako dolgo, da se tipka razklene (da nekdo tipko spusti), bi zapisali:

```
Bitwait A0,  set 'cakamo da nekdo tipko spusti
```

Drugi ukaz je za branje analognega vhoda. Analogni vhod preberemo z ukazom `AnA0`. Ta ukaz nam sicer vrne številčno vrednost od 0 do 1023, ki jo lahko sicer shranimo v spremenljivko. Mi pa smo jo tokrat izpisali na zaslon z ukazom `Print`.

```
Print AnA2
```



Slika 14: Napetostni delilnik s termistorjem povezan na analogni vhod

NALOGA: Zapiši program, da se bo vklopila LED dioda na izhodu D11, če boš približno 2s držal termistor z roko. Ko se termistor zopet ohladi, naj se LED dioda izklopi.

NALOGA*: **a)** Zapiši program, da bo stvetila modra led dioda na izhodu D11, ko bo termistor ohlajen. **b)** Ko boš termistor stisnil s prsti za 4s, naj zasveti zelena dioda na izhodu d12, modra pa naj se izklopi. **c)** Ko pa boš držal termistor več kot 5s, naj zasveti rdeča led dioda na izhodu D13, ostali dve diodi pa naj bosta izklopljeni.

5 Kako si določeno stanje zapomnimo?

V prejšnjih vajah smo sestavili RS flip-flop vezje, s katerim smo si zapomnili določeno stanje. Torej, če je kadarkoli temperatura presegla določeno vrednost, smo si to zapomnili oziroma nas je na to opozorila lučka, tudi če se je med tem časom temperatura že znižala.

NALOGA*: Uporabi enako vezje iz prejšnje vaje (shema na sliki 13) in preizkusi spodnji program.

programi/alarm_povisane_temp.bas

```
$regfile ="m328pdef.dat"
$crystal =16000000 '16Mhz
$baud =9600
$include "ArduinoNANO.bas"

pullupa0 =1

dim povisana_temp as byte 'uvedemo novo spremenljivko
povisana_temp =0 'njeno vrednost nastavimo na 0

   bitwait A0,  reset
   do
      print Ana2
      If AnA2 >480 then povisana_temp =1
      if povisana_temp =1 then
       d11 =1
      else
       d11 =0
      end if
      waitms 200
   loop
   end
```

Vrednost v pogojnem stavku (480) nastavi glede na svoje vezje. To je vrednost ki jo preberemo na analogne vhodu, ko nekaj časa držiš oziroma segrevaš termistor. V terminalnem oknu lahko spremljaš kako se vrednost povečuje. Ko je ta vrednost presežena, zasveti LED-ica. Vse skupaj seveda sprožiš s tipko na vhodu A0.

Torej, spremenljivka povisana_temp nosi informacijo o tem, ali je temeratura kadarkoli presegla mejo 480. Takrat vrednost spremenljivke nastavimo na 1. Tudi če se temperatura kasneje zniža, LED-ica še vedno sveti.

NALOGA*: Dopolni in spremeni program tako, da boš lahko ta alarm resetiral s tipko na vhodu A1. Torej, vezje zazna povišano temperaturo. Seveda pa vezje lahko resetiramo, ampak samo kadar je temperatura pod mejno vrednostjo.