

Krepitev kompetenc strokovnih delavcev na področju vodenja inovativnega vzgojno-izobraževalnega zavoda v obdobju od 2018 do 2022

Izdelava digitalnega merilnika temperature s pomočjo termistorja in mikrokrmilnika Arduino

Opis delavnice, predvidene za izvedbo na srečanju srednješolskih profesorjev fizike 15. 5. 2019 na FMF, Jadranska 19, Ljubljana

Richard Feynman:

»What I cannot create, I do not understand.«

Vir: https://en.wikiquote.org/wiki/Richard_Feynman

Milenko Stiplovšek, Zavod RS za šolstvo

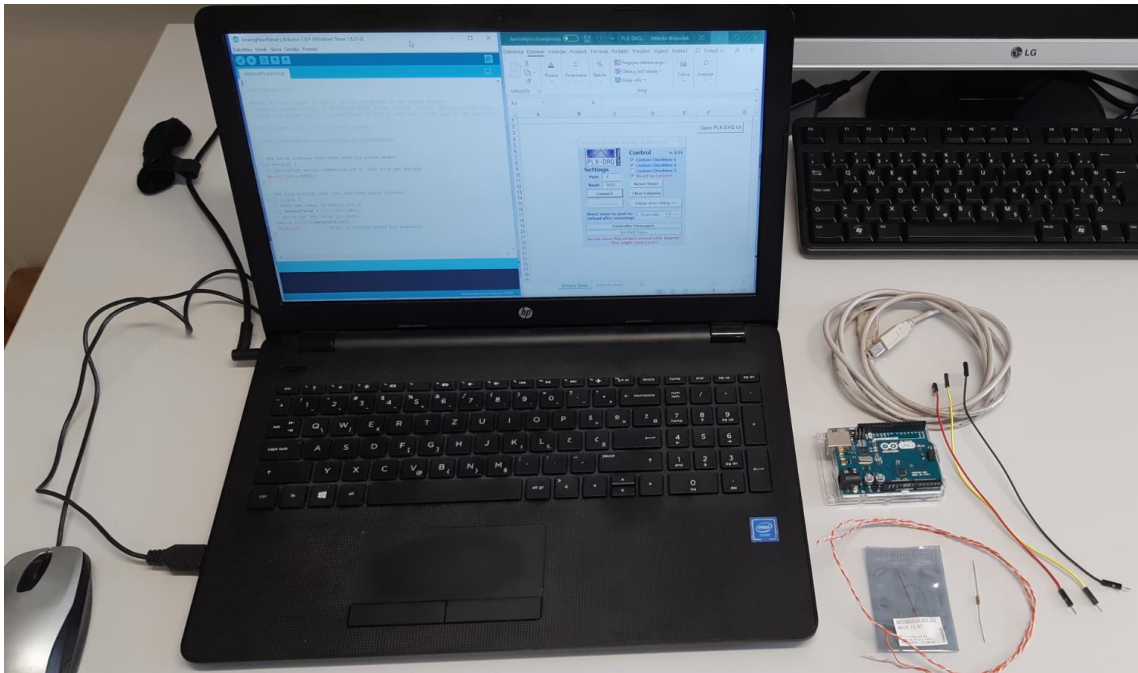
Namen delavnice: izdelava digitalnega merilnika temperature s pomočjo termistorja in mikrokrmilnika Arduino

Zgradba in opis merilnika:

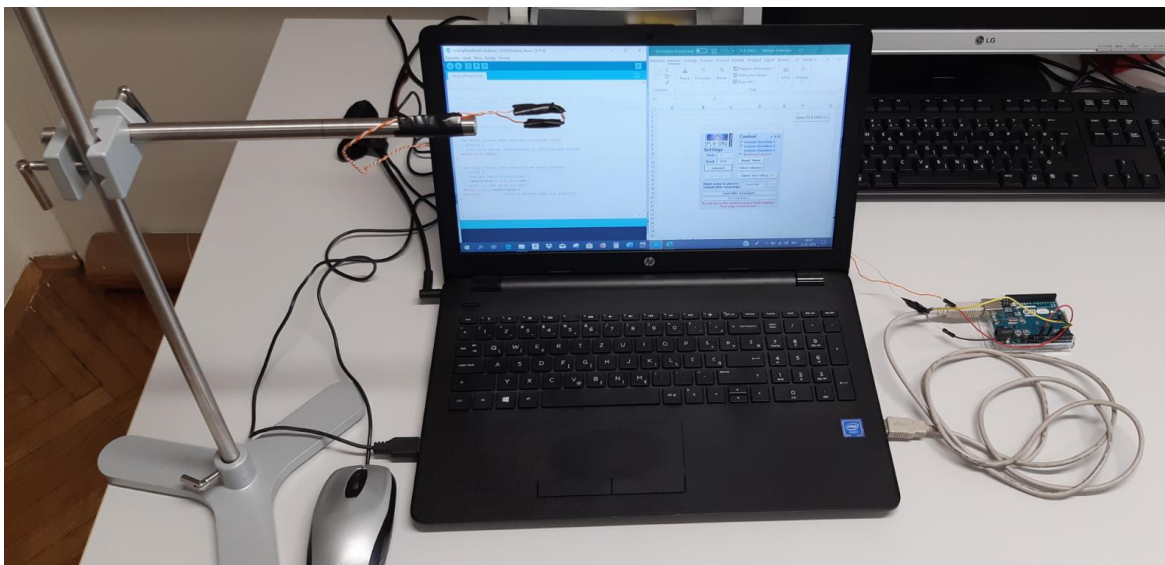
Merilnik sestavljajo:

- Sensor: zaporedno zvezana termistor in fiksni upornik ter vezne žice
- Vmesnik: mikrokrmilnik Arduino
- Računalnik: PC s prosto dostopnim programom Arduino IDE in s prosto dostopno razširitvijo PLX-DAQ za MS Excel

Opis aktivnosti: Vsak udeleženec sestavi senzor. Nato ta senzor priklopi na mikrokrmilnik Arduino, povezan s prenosnikom z ustrežno programsko opremo, ter ga umeri in preizkusi.

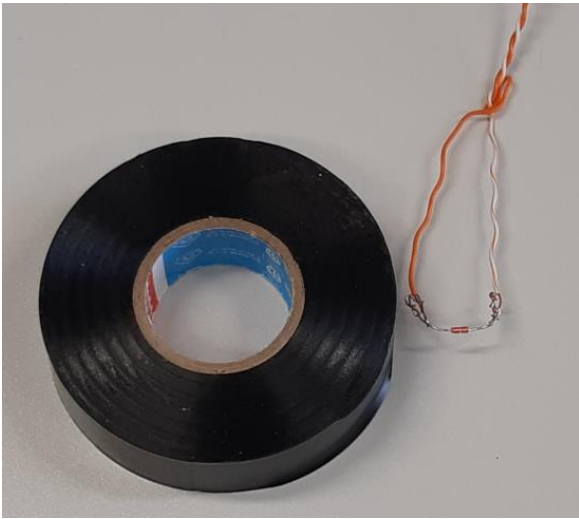


Slika 1: Sestavni deli merilnika

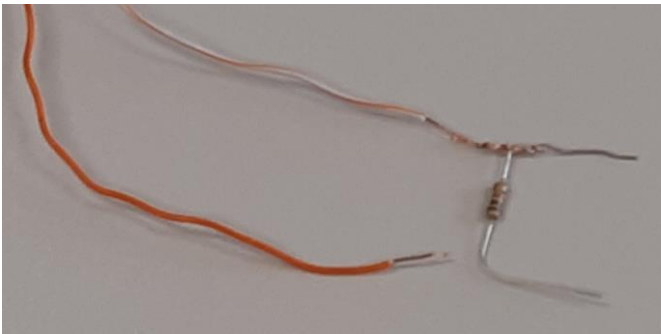


Slika 2: Sestavljen merilnik

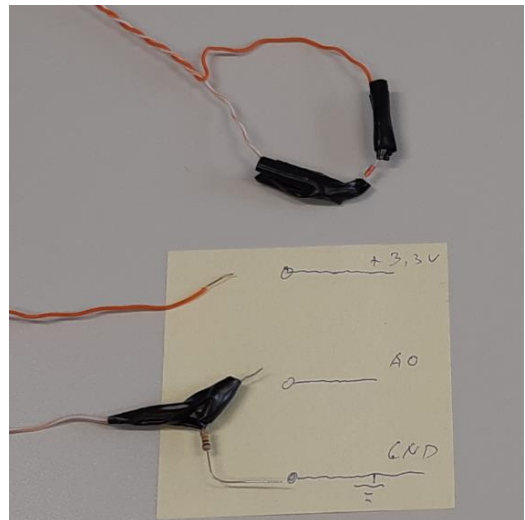
Izdelava senzorja:



Sliki 3 in 4: priklop termistorja na parico – provizorična/razstavljiva verzija. Za izvedbo zanesljivega in trajnega senzorja je seveda potrebno spajkanje in nato zaščita s termično skrčko.

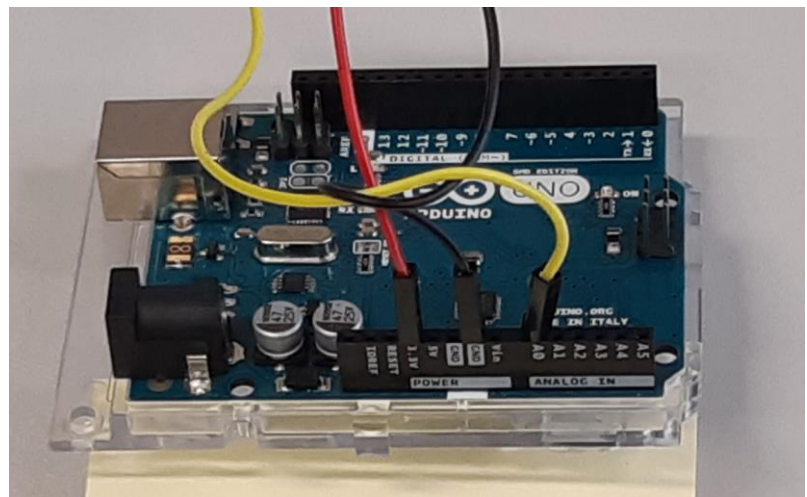
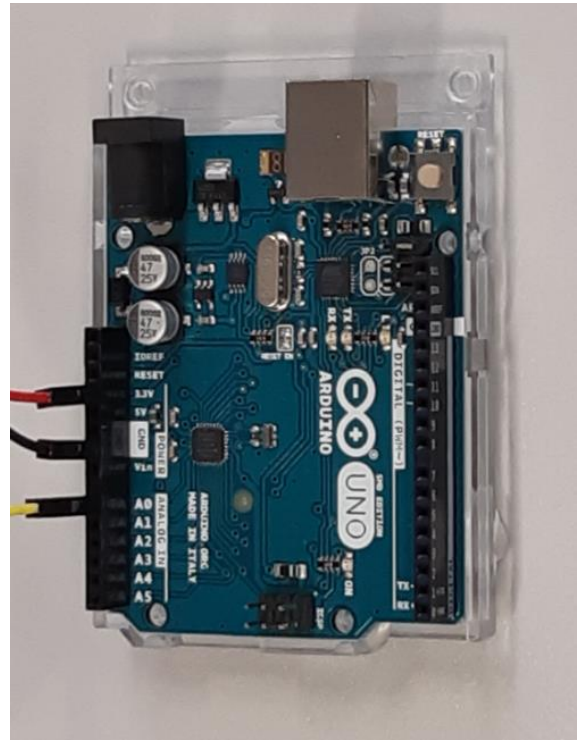
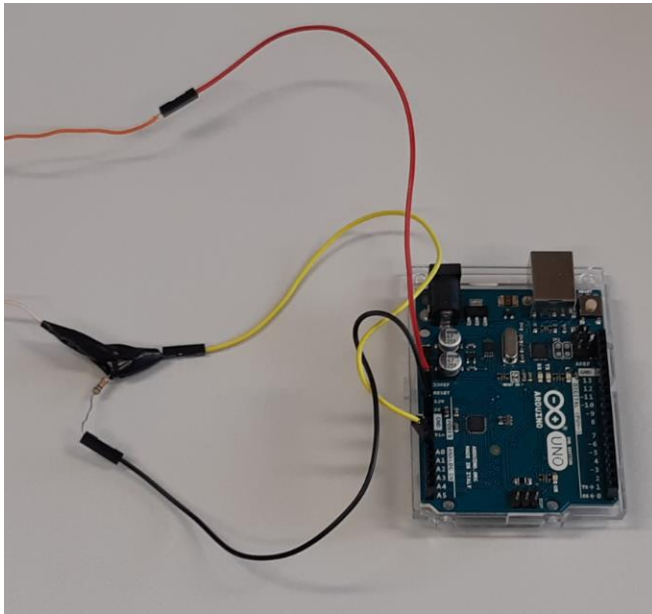


Slika 5: Provizorični priklop upornika na parico

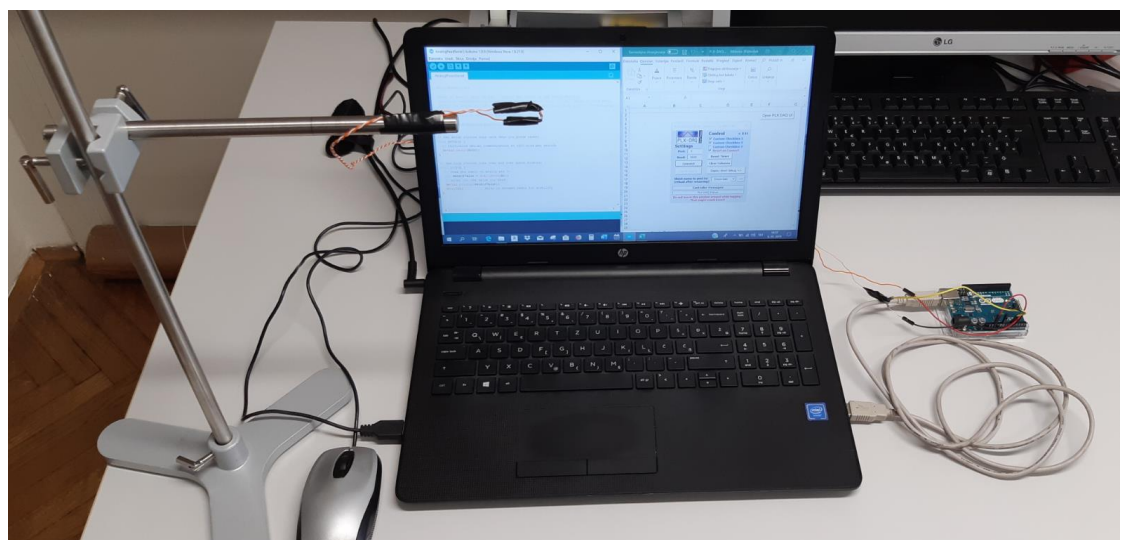


Slika 6: Sestavljen senzor z oznakami priklopa na Arduino

Sestava merilnika:



Slike 7, 8, 9: Priklop senzorja na mikrokrmilnik Arduino

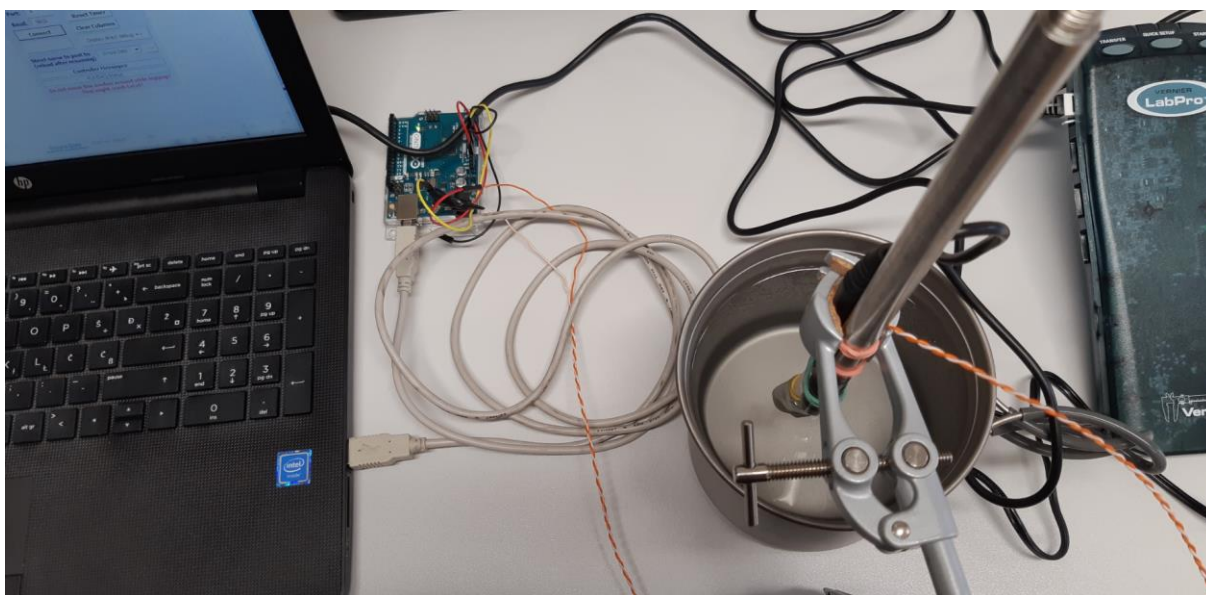


Slika 10:
Sestavljen
merilnik za
merjenje
temperature
zraka

Umerjanje in preizkušanje merilnika:



Slika 11: Postavitev za umerjanje in za preizkušanje merjenja temperature zraka – primerjava z rezultati merilnika Vernier



Slika 12: Postavitev za umerjanje in za preizkušanje merjenja temperature vode – primerjava z rezultati merilnika Vernier

Prvi način umerjanja – s funkcijo map():

Predpostavimo premo sorazmernost med vrednostjo na krmilnikovem analognem vhodu A0 in temperaturo termistorja. Poiščemo dva para vrednosti A0 in T ter uporabimo funkcijo map() v okolju Arduino IDE. Operiramo s celoštevilčnimi vrednostmi, program (skica) za Arduino je kratek in enostaven:

```
// deklaracija in inicializacija spremenljivk
int T = 0; // izračunana temperatura - s funkcijo map()
int pA0 = 0; // prebrana vrednost na A0
int T1 = 0; // izmerjena temperatura termistoja v trenutku, ko smo prebrali A01
int T2 = 0; // izmerjena temperatura termistoja v trenutku, ko smo prebrali A02
int A01 = 0; // prebrana vrednost na A0 v trenutku, ko je bila temperatura termistorja T1
int A02 = 0; // prebrana vrednost na A0 v trenutku, ko je bila temperatura termistorja T2

void setup() {
  Serial.begin(9600);
  pinMode(0, INPUT);

  // vnesite vrednosti T in A0, ki ustrezajo vsi kombinaciji termistroja (NTC DHT0B03F3953SY) in upornika (R=10kohmov):
  T1 = 23; // izmerjena temperatura termistoja v trenutku, ko smo prebrali A01
  A01 = 157; // prebrana vrednost na A0 v trenutku, ko je bila temperatura termistorja T1
  T2 = 30; // izmerjena temperatura termistoja v trenutku, ko smo prebrali A02
  A02 = 217; // prebrana vrednost na A0 v trenutku, ko je bila temperatura termistorja T2
}

void loop()
{
  pA0=analogRead(0); // branje vrednosti na A0
  T = map(pA0, A01, A02, T1, T2); // prirejanje T na osnovi prebranega A0 in referenčnih vrednosti
  Serial.println(T);
  delay(1000); // program čaka 1000 milisekund in nato nadaljuje na začetku funkcije loop()
}
```

Slika 13: koda programa (skice) za Arduino pri umerjanju s funkcijo map().

Sistematična napaka meritve zaradi uporabljenega teoretičnega modela pri tako umerjenem merilniku je odvisna od širine in lege intervala, ki ga omejujeta referenčni temperaturi in od oddaljenosti izmerka od referenčnih temperatur. Ožji kot je temperaturni interval in bližje kot smo referenčnim temperaturam – manjša je sistematična napaka meritve.

Drugi način umerjanja – s parametrom B oziroma β :

Vir: <https://en.wikipedia.org/wiki/Thermistor>

B or β parameter equation [\[edit\]](#)

NTC thermistors can also be characterised with the *B* (or β) parameter equation, which is essentially the [Steinhart–Hart equation](#) with $a = (1/T_0) - (1/B) \ln(R_0)$, $b = 1/B$ and $c = 0$.

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln\left(\frac{R}{R_0}\right),$$

where the temperatures are in [Kelvin](#), and R_0 is the resistance at temperature T_0 (25 °C = 298.15 K). Solving for R yields:

$$R = R_0 e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

Za izračun temperature termistorja na osnovi njegovega upora moramo torej poznati vrednost parametra B in eno vrednost upora termistorja pri dani temperaturi. Oboje nam praviloma poda proizvajalec. Standardno je podana vrednost upora pri temperaturi 25 °C.

Primer podatkov proizvajalca:

- NTC termistor DHT0B303F3953SY -40 - +200 °C stekleno ohišje

Tehnični podatki

| | |
|--------------------------------------|--------------------------------|
| Izvedba (senzori) | NTC |
| Material ohišja | Steklo |
| Mera, dolžina | 4 mm |
| Merilno območje temperature | -40 do +200 °C |
| Natančnost | 1 % |
| Oblika (uporniki/kondenzatorji) | Aksialno ožičen |
| Območje merjenja temperature (maks.) | +200 °C |
| Območje merjenja temperature (min.) | -40 °C |
| Premer, \varnothing | 2 mm |
| Priključek (senzor) | Spajkalni zatič |
| Referenčna osnovna upornost | +25 °C |
| Skladnost z rohs | Da |
| Talna odpornost | 30 kΩ |
| Temperaturni koeficient | 3950 K |
| Tip (tip proizvajalca) | DHT0B303F3953SY |
| Tip senzorja | Temperaturni senzor |

Program (skica) za Arduino za izračun T iz vrednosti na vhodu A0 je v tem primeru daljši in zahtevnejši, vendar je zaradi ustreznega znanja fizike in ob poznavanju lastnosti mikrokrmilnika še vedno obvladljiv z »amaterskim programiranjem«, ki smo ga spoznali med študijem.

Za konec še primer rezultatov meritev – umerjeno s funkcijo map() in z realizirano povezavo Arduino – MS Excel:

