Tehnički fakultet Rijeka

Računalom podržana mjerenja

Seminarski rad



Izradili: Deni Klen, Ani Perušić, Mihael Petranović, Mateo Srića

Mentor: prof. dr. sc. Saša Vlahinić

Rijeka, ožujak 2021

Sadržaj

[Uvod 3](#_Toc65772887)

[Razvojna okruženja i sklopovlje 3](#_Toc65772888)

[Arduino platforma 3](#_Toc65772889)

[Python 3](#_Toc65772890)

[Senzor 4](#_Toc65772891)

[Opis rada 4](#_Toc65772892)

[Programska podrška 6](#_Toc65772893)

[Praćenje i pohranjivanje podataka 6](#_Toc65772894)

[Grafičko sučelje 7](#_Toc65772895)

[Sažetak 10](#_Toc65772896)

[Literatura 10](#_Toc65772897)

# Uvod

Tema seminarskog rada je izrada grafičkog sučelja u Python-u koje će prikazivati podatke o temperaturi i vlažnosti pomoću senzora spojenog na Arduino mikročip, kao i izrada data loggera, uređaja koji prikuplja podatke sa senzora i sprema ih za kasnije korištenje.

# Razvojna okruženja i sklopovlje

## Arduino platforma

Mikročipska ploča korištena u izradi projekta je Arduino Micro. Ona je jedna od Arduinovih mikročipovskih ploča koje su napravljene za početnike; jednostavna je za upotrebu te pisanje koda, a služi kao dobra polazna točka za upoznavanje i učenje elektronike. Zasnonavana je na Atmegi32U4 i razvijena zajedno s Adafruitom. Ima 20 digitalnih ulazno/izlaznih pinova (od kojih se 7 mogu koristiti kao PWM izlazi, a 12 kao analogni ulazi), kristalni oscilator od 16 MHz, mikro USB vezu, ICSP zaglavlje i gumb za resetiranje. Sadrži sve potrebno za podršku mikrokontrolera i jednostavno se spaja na računalo uz pomoć mikro USB kabela. Ugrađena USB komunikacija uklanja potrebu za sekundarnim procesorom. To omogućuje mikročipskoj ploči da se na povezanom računalu prikaže kao miš i tipkovnica, uz virtualni (CDC) serijski / COM priključak.

Programsku podršku za mikročip pisali smo u Arduino IDE-u. To je softver otvorenog koda koji olakšava pisanje programske podrške i prijenos na bilo koju Arduino ploču. Aktivni razvoj se odvija na GitHub-u.

## Python

Za izradu grafičkog sučelja koristili smo programski jezik Python. Python je programski jezik visoke razine i opće namjene koji omogućava brzi rad i učinkovitu integraciju sustava. Njegove jezične konstrukcije i objektno orijentirani pristup imaju kao cilj pomoći programerima da napišu jasan i logičan kod bez obzira o veličini projekta. Python se dinamički upisuje i prikuplja smeće, stoga programer na mora misliti o alokaciji i brisanju memorije, podržava više paradigmi programiranja, uključujući strukturirano (posebno proceduralno), objektno orijentirano i funkcionalno programiranje. Python se često opisuje kao jezik "s uključenim baterijama" zbog svoje sveobuhvatne standardne biblioteke koja uvelike olakšava programiranje.

## Senzor

Senzor korišten za prikupljanje podataka o temperaturi i vlažnosti je DHT11. U njemu se nalazi senzor temperature – mali termistor zalemljen na pločicu te senzor vlage – mala tiskana pločica koja je nadolemljena na osnovnu. Što je više vlage u zraku, više vlage dolazi i na same vodove te je otpor među njima manji (ili počinje postojati kada se pojavljuje voda — to je razlog zašto senzor očitava vrijednosti od 20%, tek onda može očitati nekakav otpor). Osim navedene dvije komponente još nalazimo jedan integrirani krug koji analzira ulaze od prethodno navedenih senzora i komunicira s Arduinom.

|  |  |
| --- | --- |
| *Slika 1. VMA311 DHT11 senzor* | *Slika 2. Arduino Micro* |

# Opis rada

Prije otvaranja programa, potrebno je spojiti Arduino pločicu na računalo i senzor na Arduino pločicu. Senzor se na Arduino spaja na 5V, uzemljenje i analogni ulaz A0. Nakon pravilnog spajanja se može otvoriti program. Potrebno je odabrati port na koji je Arduino spojen nakon čega se može pokrenuti konekcija između računala i Arduina gumbom *Connect to Arduino*. U svakom trenutku se može ista konekcija zaustaviti gumbom *Disconnect from Arduino.*

|  |  |
| --- | --- |
| *Slika 3. DHT22/11 senzor spojen na Arduino Micro* | *Slika 4. Prikaz zaglavlja prozora* |

Sučelje se sastoji od dva gumba – *Show Last Minute* i *Show Last Hour* koji pritiskom pokazuju grafove i podatkovne vrijednosti u tablici.

|  |  |
| --- | --- |
| *Slika 5. Gumbi za prikaz podataka* | |
| *Slika 6. Graf vrijednosti za protekli sat* | *Slika 7. Sirovi prikaz vrijednosti* |

# Programska podrška

## Praćenje i pohranjivanje podataka

Arduino pločica služi za spremanje vrijednosti senzora kao i izračun srednjih vrijednosti i standardne devijacije. Programska podrška za Arduino uključuje blago izmjenjenu DHT biblioteku koja je obavljala čitanje vrijednosti, prebacivanje i spremanje u varijable te provjeru kontrolne sume. Na početku se sve vrijednosti postavljaju na tisuću što omogućuje da se ta vrijednost nikada ne dostigne. Koristi se funkcija *resetData()*. Da je *default* vrijednost postavljena na nula, ne bi se moglo razaznati kada je nula izmjerena vrijednost, a kada *default*. U glavnom kodu svake dvije sekunde izvršava se prihvaćanje novih vrijednosti senzora. Vrijednosti kraće od dvije sekunde mogle bi remetiti očitanja, no u ovom slučaju češća očitanja nisu potrebna. Nakon očitanja vrijednosti, temperatura i vlažnost se spremaju u zasebne nizove. Korištena je logika kružnih lista i slanje trenutnog indeksa. Time se zna koja je vrijednost najnovija, a starije su vrijednosti redom lijevo od nje.

|  |  |  |
| --- | --- | --- |
| *Slika 8. Glavni dio programa* | | *Slika 9. resetData()* |
| Slika 10. Dodavanje novih vrijednosti | | |
| *Slika 11. Poziv funkcija za dodavanje vrijednosti sata* | *Slika 12. Računjanje srednje vrijednosti tempetarure* | |
| *Slika 13. Dodavanje vrijednosti sata* | | |
| *Slika 14. Računanje standardne devijacije* | | |

Nakon spremanja vrijednosti u nizove, isti se python kodu šalju preko serijske komunikacije. Kada Arduino zaprimi vrijednost "1" šalje vrijednosti za predhodnu minutu, a kada zaprimi "2" šalje za predhodnih sat vremena. U oba slučaja na kraju šalje i trenutni indeks polja na kojemu se nalazi.

|  |  |
| --- | --- |
| *Slika 15. Funkcija za ispis vrijednosti* | *Slika 16. Ispis vrijednosti* |

## Grafičko sučelje

Python-ova programska podrška se temelji na bibliotekama *tkinter* (izrada prozora i objekata), *serial* (čitanje sa serijskog porta), *threading* (lakše upravljanje procesima i memorijom), *Matplotlib* (izračun i prikaz raznih grafova), *NumPy*, *pandas* i *re* (manipulacija podataka) te *time* i *datetime*(izračun i prikaz vremena).

Inicijalizacija serijskog porta kao i dretve koja obrađuje serijski ulaz i izlaz odrađuje se pritiskom na *Connect to Arduino* gumb. Za aktivaciju ulaza označujemo jedan od ponuđenih iz kombiniranog okvira. Istom logikom, pritiskom *Disonnect from Arduino* gumba zaustavlja se dretva, odnosno komunikacija.

|  |
| --- |
|  |
| *Slika 17. Funkcije za čitanje i slanje sa serijskog porta, kao i za dretve* |

Opisani će biti samo proces koji se izvršava pritiskom na gumb *Show Last Minute* jer je za prikaz podataka sata jako sličan. Poziva se izravno funkcija *showMinuteStats()* koja šalje *"1"* na serijski port označavajući da želi primiti podatke o predhodnoj minuti. U slučaju da serijski port nije otvoren, dolazi do iznimke i program skočnim prozorom javlja da komunikacija između Arduina i računala nije započeta. Kada dobije podatke, odvaja vrijednosti u listu delimiterima *","* i *";"*. Iz te liste odvaja podatke temperature i vlage u zasebne nizove.

|  |
| --- |
| *Slika 18. Dohvaćanje podataka* |

Poziva se funkcija *rearangeMinutes()* čiji je posao pronaći ispravan redoslijed podataka. Ista je krucijalna jer inače se nebi mogao odrediti ispravan poredak podataka, svaki bi mogao biti najnoviji. To radi uz pomoć funkcije *circularArray(a, l, ind)* koja, koristivši se pomoćnim poljem, reorganizira podatke. Nakon namještanja ispravnog polja, u drugo se dodaju vremena očitavanja podataka. Funkcija na kraju vraća indeks zadnjeg elementa. Zatim se zove funkcija *minuteStats(g)* koja stvara novi prozor i u tablicu stavlja podatke nizova. Na kraju se poziva funkcija za stvaranje grafa koji iste vrijednosti prikaže grafički linearnim grafom.

|  |  |
| --- | --- |
| *Slika 19. Manipulacija poljima* | |
| *Slika 20. Namještanje kružnog niza* | *Slika 21. Stvaranje grafa* |
| *Slika 22. Sve funkcije* | |

# Sažetak

U radu se upoznajemo načinom na koji smo realizirali mjerenje vrijednosti temperature i vlažnosti, koristeći se Arduinom i DHT11 senzorom, te grafičkim sučeljem napisanim u Python jeziku za vizualno prikazivanje istih. Dotičemo se razvojnog okruženja, sklopovlja i programske podrške korištene pri izradi projekta te detaljnije objašnjavamo neke dijelove koda i funkcije potrebne za ispravan rad projekta.

# Literatura

<https://store.arduino.cc/arduino-micro>

<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>

<https://www.arduino.cc/en/software>

<https://github.com/arduino/Arduino>

<https://docs.python.org/3/library/tkinter.html>

<https://matplotlib.org/stable/index.html>

<https://numpy.org>

<https://docs.python.org/3/library/time.html>