

ATOMATIZIRANJE SUSTAVA KONTROLE VENTILATORA S POMOĆU RASPBERRY PI RAČUNALA I TEMPERATURNOG SENZORA

Tolić, Mihael

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Karlovac
University of Applied Sciences / Veleučilište u Karlovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:128:703116>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-13**



VELEUČILIŠTE U KARLOVCU
Karlovac University of Applied Sciences

Repository / Repozitorij:

[Repository of Karlovac University of Applied
Sciences - Institutional Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

VELEUČILIŠTE U KARLOVCU

STROJARSKI ODJEL

PREDDIPLOMSKI STRUČNI STUDIJ MEHATRONIKA

MIHAEL TOLIĆ

**Automatizirani sustav kontrole ventilatora s
pomoću Raspberry Pi računala i
temperaturnog senzora**

ZAVRŠNI RAD

KARLOVAC, 2024.

VELEUČILIŠTE U KARLOVCU
STROJARSKI ODJEL
PREDDIPLOMSKI STRUČNI STUDIJ MEHATRONIKA

MIHAEL TOLIĆ

**Automatizirani sustav kontrole ventilatora s
pomoću Raspberry Pi računala i
temperaturnog senzora**

ZAVRŠNI RAD

Mentor: mr. sc. Vedran Vyroubal, v. pred.

KARLOVAC, 2024.

KARLOVAC UNIVERSITY OF APPLIED SCIENCES

DEPARTMENT OF MECHANICAL ENGINEERING

PROFESSIONAL STUDY OF MECHATRONICS

MIHAEL TOLIĆ

**Automated fan control system using
Raspberry Pi computer and temperature
sensor**

UNDERGRADUATE THESIS

Mentor: mr. sc. Vedran Vyroubal, senior lecturer

KARLOVAC, 2024.

IZJAVA

Izjavljujem da sam svoj rad izradio samostalno s pomoću stečenog znanja tijekom studija, stručne literature i interneta. Izrada ovog završnog rada omogućila mi je produblјivanje znanja o programiranju u programskom jeziku Python te spajanje elektroničkih elemenata u jednu cjelinu. Ovaj rad je bio izazovno, ali istovremeno i vrlo korisno iskustvo koje će mi zasigurno biti od velike pomoći u budućim radovima u ovom području.

Htio bih izraziti svoju zahvalnost svima koji su mi pomogli u izradi ovog završnog rada. Posebno se želim zahvaliti svom mentoru mr. sc. Vedran Vyrubal, v. pred. na strpljenju, vodstvu i korisnim savjetima tijekom cijelog procesa. Također želim zahvaliti svojoj obitelji koja me uvijek podržavala tijekom studiranja.

SAŽETAK

U ovom radu se prikazuje kako izraditi automatizirani sustav kontrole ventilatora s pomoću Raspberry Pi-a i temperaturnog senzora. Rad je podijeljen na uvodni, teoretski i praktični dio izrade sustava. U uvodnom dijelu se opisuje povijest automatizacije i razvoj Raspberry Pi-a. Teorijski dio sadrži opis korištenih dijelova te njihova primjena i također će se opisati dodatne biblioteke potrebne za ispravan rad komponenti. U praktičnom dijelu se prikazuje kako funkcionira izrađeni sustav te linije programa koje se izvode.

Ključne riječi: Raspberry Pi, automatizacija, senzor temperature i vlage, lcd zaslon

SUMMARY

In this thesis, was shown how to make automated fan control system using Raspberry Pi and temperature sensor. Thesis is divided into introductory, theoretical and practical part of system. The introductory describes history of automation and development of Raspberry Pi. Theoretical part describes used components and their use and also additional libraries required for proper functioning of components. Practical part shows how the system works and code used for programming.

Keywords: Raspberry Pi, automatization, temperature and humidity sensor, lcd screen

Sadržaj

1. UVOD1
2. TEORIJSKI DIO2
 - 2.1 Automatizacija2
 - 2.2 Raspberry Pi4
 - 2.2.1 Raspberry Pi Pico6
 - 2.3 DHT11 senzor temperature7
 - 2.4 Ventilator9
 - 2.5 LCD zaslon11
3. PRAKTIČNI DIO13
 - 3.1 Spajanje senzora DHT11 na Raspberry Pi13
 - 3.2 Spajanje LCD zaslona na Raspberry Pi14
 - 3.3 Spajanje ventilatora na Raspberry Pi15
 - 3.3.1 Povezivanje15
4. PROGRAMSKI DIO16
 - 4.1 Thonny program16
 - 4.2 Instalacija potrebnog softvera na Raspberry Pi18
 - 4.3 Programiranje u Thonny19
 - 4.3.1 Čitanje podataka sa senzora DHT1119
 - 4.3.2 Upravljanje ventilatorom21
 - 4.3.3 Upravljanje LED elementima22
 - 4.3.4 Upravljanje LCD zaslonom23
5. KONAČNI PROGRAM AUTOMATSKOG UPRAVLJANJA VENTILATORA24
6. ZAKLJUČAK26
7. LITERATURA27
8. PRILOZI29

1. UVOD

U modernim tehničkim sustavima i industrijskim procesima, regulacija temperature predstavlja ključni faktor za održavanje optimalnih uvjeta rada i produženje vijeka trajanja opreme. Automatizacija hlađenja postaje sve važnija zbog povećanih zahtjeva za učinkovitim korištenjem energije, stabilnošću rada sustava i smanjenjem rizika od pregrijavanja komponenata. U ovom završnom radu istražiti ćemo mogućnosti automatiziranog sustava hlađenja koristeći Raspberry Pi kao mikroupravljač, DHT11 temperaturni senzor za očitavanje temperature i ventilator kao izlazni uređaj za hlađenje. Implementacija će omogućiti učinkovitu kontrolu temperature u stvarnom vremenu, s ciljem sprječavanja pregrijavanja komponenata ili prostorija, te optimizaciju potrošnje energije.

Također, cilj je detaljno opisati proces dizajna, implementacije i testiranja sustava, uključujući povezivanje hardvera i razvoj softvera za upravljanje ventilatorom.

2. TEORIJSKI DIO

U ovom teorijskom dijelu fokusira se na automatizaciju i njen značaj u modernom dobu te na korištene komponente i njihov način rada.

2.1 Automatizacija

Automatizacija je proces primjene tehnologije za izvršavanje zadataka bez ili uz minimalnu ljudsku intervenciju. Cilj automatizacije je povećanje učinkovitosti, preciznosti i brzine izvedbe različitih zadataka, istovremeno smanjujući troškove i mogućnost ljudskih pogrešaka. Automatizacija se može primijeniti u različitim industrijama, uključujući proizvodnju, informacijske tehnologije, poljoprivredu, zdravstvo i mnoge druge sektore.

Automatizacija se može podijeliti u nekoliko glavnih vrsta, ovisno o složenosti i području primjene:

- a) Fizička Automatizacija: uključuje mehaničke uređaje i robote koji izvode fizičke zadatke. Primjeri uključuju industrijske robote u tvornicama koji obavljaju montažu, zavarivanje ili pakiranje proizvoda.
- b) Procesna Automatizacija: koristi se za automatizaciju složenih industrijskih procesa, kao što su rafinerije, kemijska postrojenja i elektrane. Procesna automatizacija koristi senzore, aktuatorne uređaje i sustave za nadzor kako bi osigurala stabilan i siguran rad.
- c) IT Automatizacija: koristi se za automatizaciju zadataka u informatičkom okruženju, poput upravljanja serverima, mrežnim postavkama i sigurnosnim procedurama.
- d) Automatizacija poslovnih procesa (BPA): fokusirana je na automatizaciju rutinskih administrativnih zadataka unutar organizacija, poput obrade podataka, fakturiranja i upravljanja dokumentima.
- e) Robotizirana automatizacija procesa (RPA): RPA koristi softverske robote ili "botove" za automatizaciju ponavljajućih zadataka unutar digitalnih sustava. Botovi mogu simulirati ljudsku interakciju s računalima i izvršavati zadatke poput unosa podataka, slanja e-pošte i pretraživanja baza podataka.

Prednosti Automatizacije:

- a) Povećana učinkovitost: automatizirani sustavi mogu izvršavati zadatke brže i točnije od ljudi, što rezultira većom produktivnošću.
- b) Smanjenje troškova: iako implementacija automatizacije može zahtijevati inicijalne troškove, dugoročno smanjuje operativne troškove smanjenjem potrebe za ljudskom radnom snagom.
- c) Povećana sigurnost: automatizacija može smanjiti izloženost ljudi opasnim uvjetima rada, primjerice u kemijskim postrojenjima ili na proizvodnim linijama.
- d) Konzistentnost i kvaliteta: automatizirani sustavi mogu održavati dosljednu kvalitetu proizvoda ili usluga, jer smanjuju varijacije koje nastaju zbog ljudske pogreške.

Nedostaci Automatizacije:

- a) Visoki inicijalni troškovi: implementacija automatizacije može biti skupa, što uključuje troškove opreme, softvera i obuke zaposlenika.
- b) Gubitak radnih mjesta: automatski sustavi mogu zamijeniti ljudsku radnu snagu, što može dovesti do smanjenja broja zaposlenih u nekim industrijama.
- c) Kompleksnost i održavanje: automatizirani sustavi mogu biti složeni za implementaciju i zahtijevaju redovito održavanje kako bi se osigurala njihova pouzdanost.

2.2 Raspberry Pi

Raspberry Pi je kompaktno i pristupačno računalo razvijeno od strane Raspberry Pi Foundationa u Ujedinjenom Kraljevstvu. Prvobitno je zamišljen kao edukacijski alat za podučavanje osnovnih koncepata računalstva, ali je brzo stekao popularnost među hobistima, inženjerima i istraživačima zbog svoje fleksibilnosti i širokog spektra mogućnosti primjene. Raspberry Pi dolazi u nekoliko različitih modela:

a) Raspberry Pi 1

- model A: prvi model izdan 2012. godine s 256 MB RAM-a, jednim USB portom i bez mrežnog priključka.
- model A+: poboljšana verzija s manjim dimenzijama i smanjenom potrošnjom energije.
- model B: ima 512 MB RAM-a, dva USB porta i mrežni priključak (Ethernet)
- model B+: poboljšana verzija s četiri USB porta, poboljšanim audio izlazom i smanjenom potrošnjom energije

b) Raspberry Pi 2

- model B: ima 4-jezgreni ARM Cortex-A7 procesor na 900 MHz i 1 GB RAM-a.

c) Raspberry Pi 3

- model B: ima 4-jezgreni ARM Cortex-A53 procesor na 1.2 GHz, 1 GB RAM-a, ugrađeni Wi-Fi i Bluetooth
- model B+: poboljšana verzija s bržim procesorom (1.4 GHz), poboljšanim Wi-Fi (802.11ac) i Bluetooth 4.2

d) Raspberry Pi 4

- model B: donosi značajna poboljšanja, uključujući 4-jezgreni ARM Cortex-A72 procesor na 1.5 GHz, podršku za do 8 GB RAM-a, dva Micro-HDMI porta za 4K video izlaz, USB 3.0 portove i gigabitni Ethernet

e) Raspberry Pi Zero

- Zero: manji i jeftiniji model, s jednim ARM11 jezgrenom procesorom na 1GHz i 512 MB RAM-a. Nema ugrađeni Wi-Fi ili Bluetooth.
- Zero W: verzija Pi Zero s ugrađenim Wi-Fi i Bluetooth
- Zero 2 W: donosi poboljšani 4-jezgreni ARM Cortex-A53 procesor i 512 MB RAM-a, uz zadržavanje ugrađenog Wi-Fi i Bluetootha.

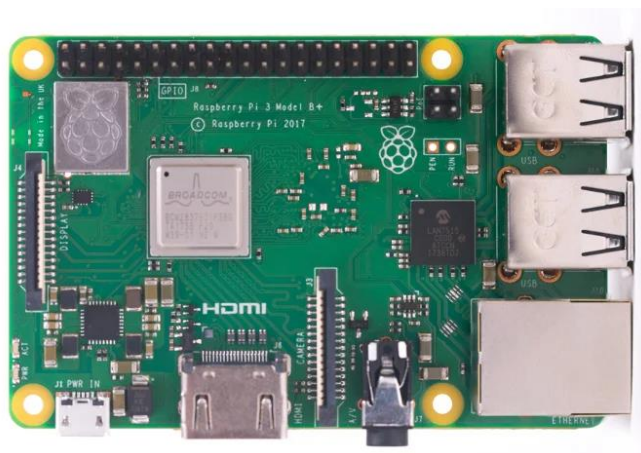
f) Raspberry Pi 400: ugrađen u tipkovnicu

g) Raspberry Pi Compute Module: dizajniran za industrijsku upotrebu

Raspberry Pi podržava različite operacijske sustave, najpopularniji od kojih je Raspberry Pi OS (prije poznat kao Raspbian), koji je prilagođen za rad na ovom uređaju. Tu su i druge opcije poput Ubuntu-a, LibreELEC-a, pa čak i Windows IoT Core. Fleksibilnost operacijskih sustava omogućuje korisnicima da prilagode uređaj za specifične potrebe, bilo da se radi o serveru, media centru, retro igraćoj konzoli ili platformi za učenje programiranja.

Neke od najčešćih primjena uključuju:

- a) Edukacija i učenje programiranja: Raspberry Pi je često korišten u školama i sveučilištima kao alat za učenje Pythona, Scratcha, te drugih programskih jezika.
- b) Internet of Things (IoT): zbog svoje male potrošnje energije i kompaktnih dimenzija, Raspberry Pi je idealan za IoT projekte, kao što su pametni kućni uređaji, senzorski sustavi i automatizacija.
- c) Medijski centri: uz korištenje softvera kao što je Kodi, Raspberry Pi može postati središte za reprodukciju multimedije.
- d) Retro igranje: Raspberry Pi se može pretvoriti u retro igraću konzolu s emulatorima za klasične igre putem platformi kao što su RetroPie.



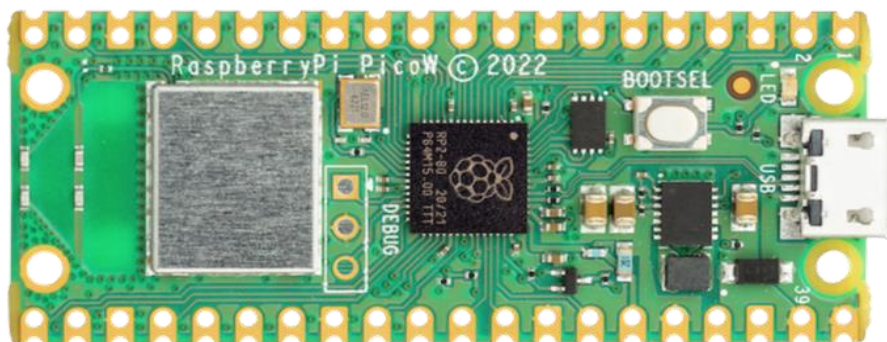
Slika 1. Raspberry Pi [1]

U izradi rada se koristi Raspberry Pi Pico koji ima 2 izvedbe:

- a) Pico
- b) Pico W koji ima ugrađen Wi-Fi modul

2.2.1 Raspberry Pi Pico

Raspberry Pi Pico pripada kategoriji mikroupravljača, a ne standardnih Raspberry Pi računala. To je prvi proizvod temeljen na vlastitom čipu Raspberry Pi Foundationa, nazvanom RP2040. Dok su tradicionalni Raspberry Pi modeli poput Raspberry Pi 3, 4 ili Zero mini računala sposobna za izvođenje punih operacijskih sustava, poput Raspberry Pi OS-a, Raspberry Pi Pico je mikroupravljač, što znači da je dizajniran za vrlo specifične zadatke i ne može pokretati operacijski sustav u klasičnom smislu. Pico je idealan za embedded projekte, gdje je potrebna mala potrošnja energije i jednostavno upravljanje hardverskim komponentama, poput senzora, motora, LED elementa i drugih elektroničkih komponenti. Programira se prvenstveno koristeći MicroPython ili C/C++. Raspberry Pi Pico je izvrstan za projekte koji zahtijevaju precizno upravljanje i nisku latenciju, poput automatizacije, robota ili DIY (uradi sam) elektronike.



Slika 2. Raspberry Pi Pico W [2]

Specifikacije Raspberry Pi Pico:

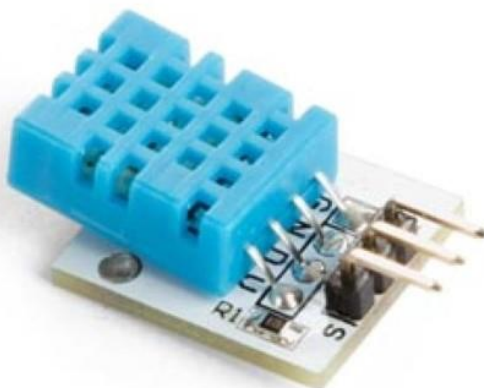
- a) Procesor: 2-jezgreni ARM Cortex-M0+ na 133 MHz
- b) RAM: 264 KB SRAM
- c) Flash memorija: 2 MB
- d) Ulazi/Izlazi (I/O): 26 višenamjenskih pinova (GPIO), uključujući I2C, SPI, UART, i PWM
- e) Napajanje: može se napajati putem USB-a ili vanjskog izvora napajanja (1,8-5,5 V)
- f) Veličina: 51 mm x 21 mm

2.3 DHT11 senzor temperature

DHT11 je popularan senzor za mjerenje temperature i vlažnosti zraka, koji se često koristi u DIY projektima, automatizaciji i sustavima za praćenje okolišnih uvjeta. Zbog svoje jednostavnosti, niske cijene i pouzdanosti, DHT11 je idealan za hobiste, studente i inženjere koji žele implementirati osnovne funkcionalnosti praćenja temperature i vlažnosti u svoje projekte.

Specifikacije senzora:

- a) Mjerni raspon temperature: 0 °C do 50 °C (s točnošću od ± 2 °C)
- b) Mjerni raspon vlažnosti: 20 % do 90 % relativne vlažnosti (s točnošću od ± 5 %)
- c) Napajanje: 3.3 V do 5 V
- d) Tip izlaza: Digitalni signal
- e) Veličina: 15.5 mm x 12 mm x 5.5 mm
- f) Potrošnja struje: Maksimalno 2.5 mA tijekom mjerenja



Slika 3. DHT11 senzor [3]

DHT11 senzor koristi kapacitivni senzor za mjerenje vlažnosti i termistor za mjerenje temperature. Električni signal koji se generira senzorom obrađuje se u internom mikroupravljaču senzora, a zatim se digitalni signal šalje na izlazni pin, gdje ga može očitati mikroprocesor, kao što je Raspberry Pi ili Arduino.

DHT11 se koristi u raznim slučajevima gdje je potrebno nadzirati uvjete okoliša. Neki primjeri uključuju:

- a) Automatizacija pametne kuće: praćenje temperature i vlažnosti za kontrolu sustava grijanja, ventilacije i klimatizacije (HVAC).
- b) Staklenici i poljoprivreda: održavanje optimalnih uvjeta za rast biljaka.
- c) Meteo stanice: praćenje i bilježenje okolišnih uvjeta za potrebe vremenskih prognoza ili znanstvenih istraživanja.
- d) Uređaji za kontrolu klime: integracija s ventilatorima, ovlaživačima ili odvlaživačima zraka za postizanje željene klime u zatvorenim prostorima.

Prednosti:

- a) Niska cijena: DHT11 je vrlo povoljan senzor, što ga čini pristupačnim za široki raspon korisnika.
- b) Jednostavnost korištenja: lako se integrira s popularnim mikroupravljačima poput Raspberry Pi i Arduino.
- c) Niska potrošnja energije: idealno za baterijski pogonjene projekte.

Nedostaci:

- a) Ograničen raspon mjerenja: mjerenje temperature do 50 °C i vlažnosti od 20 % do 90 % može biti nedovoljno za neke primjene.
- b) Manja preciznost: u usporedbi s naprednijim senzorima, DHT11 ima nižu preciznost i sporije reagira na promjene u uvjetima okoliša.
- c) Niska frekvencija uzorkovanja: DHT11 može slati podatke samo jednom svake dvije sekunde, što nije pogodno za primjene koje zahtijevaju brza očitavanja.

2.4 Ventilator

Korišteni ventilator MF40200V1-1000U-G99 je mali, visokoučinkoviti ventilator dizajniran za hlađenje elektroničkih komponenti u ograničenim prostorima. Ovaj ventilator proizvodi tvrtka Sunon, poznata po razvoju i proizvodnji visokokvalitetnih rješenja za hlađenje.

Tehničke Specifikacije:

- a) Dimenzije: 40 mm x 40 mm x 20 mm
- b) Nominalni napon: 12 V DC
- c) Potrošnja struje: Oko 0.15 A
- d) Brzina vrtnje: 7000 o/min.
- e) Razina buke: Oko 25 dBA
- f) Tip ležaja: Vapo ležaj (MagLev tehnologija)
- g) Radna temperatura: -10 °C do 70 °C
- h) Težina: 15 g



Slika 4. Ventilator [4]

Ventilator se može koristiti u različitim slučajevima gdje je potrebno osigurati učinkovito hlađenje u malim prostorima:

- a) Hlađenje malih elektroničkih uređaja: Ovaj ventilator je idealan za upotrebu u minijaturnim računalima, routerima, prekidačima i sličnim uređajima.
- b) Raspberry Pi i drugi mikroračunala: Zbog svojih dimenzija i performansi, često se koristi za hlađenje Raspberry Pi i drugih sličnih uređaja, gdje prostor i hlađenje predstavljaju izazov.
- c) 3D printeri: Koristi se za hlađenje elektronike i motora unutar 3D printera.
- d) LED svjetlosni paneli: Pomaže u raspršivanju topline iz napajanja ili LED elemenata koje proizvode puno topline tijekom rada.

2.5 LCD zaslon

LCD zaslon 16x2 je jedan od najčešće korištenih zaslona u elektroničkim projektima. Ovaj zaslon može prikazati do 32 znaka, raspoređena u dva reda s po 16 znakova. Koristi se za prikazivanje informacija kao što su poruke, podaci o senzoru, status uređaja i slično.

Tehničke Specifikacije:

- a) Dimenzije: oko 80 mm x 36 mm x 10 mm
- b) Broj znakova: 32 (16 znakova u 2 reda)
- c) Veličina znaka: oko 5.2 mm x 8 mm
- d) Tip zaslona: alfanumerički, matrica točaka (5x8 točaka po znaku)
- e) Napajanje: 5 V DC
- f) Potrošnja struje: oko 2 mA (bez pozadinskog osvjetljenja)
- g) Pozadinsko osvjetljenje: LED pozadinsko osvjetljenje (obično plavo ili zeleno)
- h) Kontrola kontrasta: podesiv s pomoću potenciometra (obično 10 k Ω)

LCD zaslon 16x2 temelji se na HD44780 kontroleru (ili kompatibilnom), koji je industrijski standard za LCD zaslone ove vrste. Ovaj kontroler omogućuje upravljanje prikazom znakova na zaslonu, koristeći set naredbi za pozicioniranje kursora, prikazivanje teksta, brisanje zaslona i druge funkcije.



Slika 5. LCD zaslon 16x2 [5]

Zaslon koristi 16-pinski konektor, pri čemu se najčešće koriste sljedeći pinovi:

- a) GND: uzemljenje (GND)
- b) VCC: napajanje (5 V)
- c) V0: kontrola kontrasta (povezano s potenciometrom)
- d) RS (Register Select): odabir između naredbe (0) ili podataka (1)
- e) RW (Read/Write): odabir između čitanja (1) ili pisanja (0)
- f) E (Enable): aktivira zapis podataka na zaslon
- g) D0-D7: podatkovni pinovi (8-bitni način rada); u 4-bitnom načinu koriste se samo D4-D7
- h) LED+: pozadinsko osvjetljenje (pozitivni terminal)
- i) LED-: pozadinsko osvjetljenje (negativni terminal)

Tablica 1. Naredbe za LCD instrukcijski registar

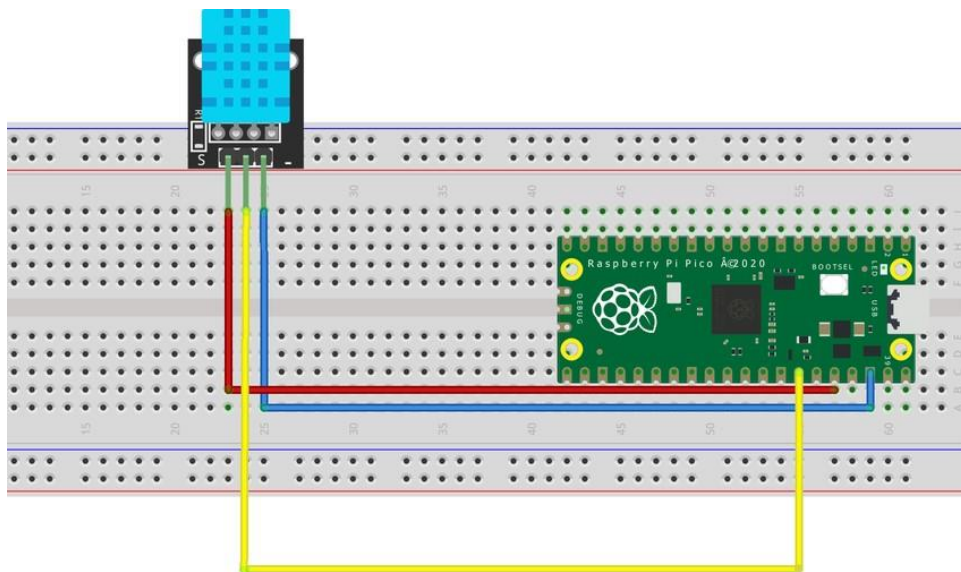
Heksadekadski kod	Naredba za LCD instrukcijski registar
0F	LCD uključen, kursor uključen
01	Čisti zaslon
02	Povratak kući
04	Pomak kursora ulijevo
06	Pomak kursora udesno
05	Pomakni prikaz udesno
07	Pomakni prikaz ulijevo
0E	Zaslon uključen, kursor treperi
80	Postavite kursor na početak 1. retka
C0	Postavite kursor na početak 2. retka
38	2 linije i matrica 5x8
83	Linija kursora 1 pozicija 3
3C	Aktivirajte drugu liniju
08	Zaslon isključen, kursor isključen
C1	Skoči na drugu liniju, pozicija 1
C2	Skoči na drugu liniju, pozicija 2
0C	Zaslon uključen, kursor isključen

3. PRAKTIČNI DIO

3.1 Spajanje senzora DHT11 na Raspberry Pi

Senzor DHT11 ima 3 pina:

- a) VCC pin
- b) GND pin
- c) Data pin



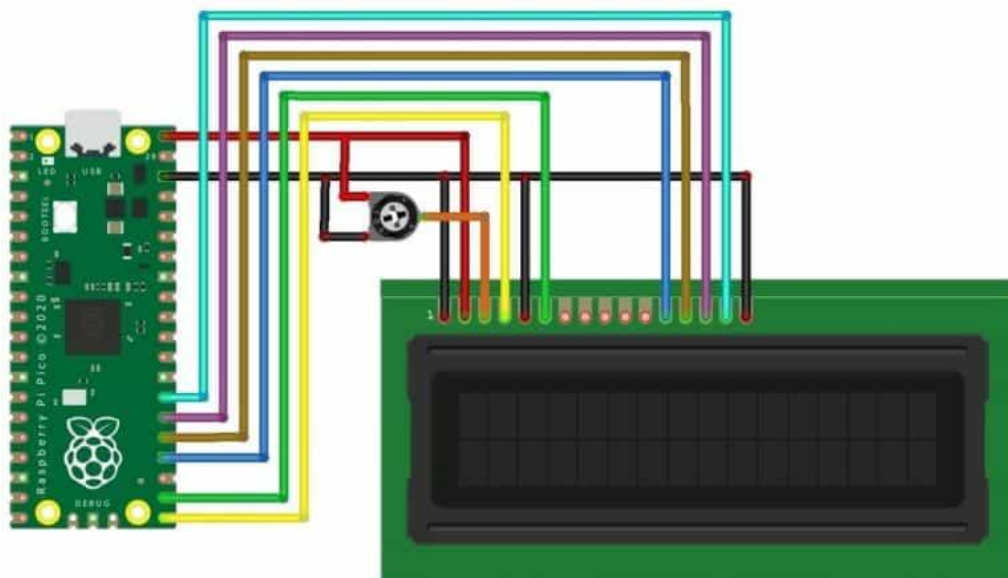
Slika 6. Spajanje DHT11 na Raspberry Pi [6]

Kod spajanja DHT11 senzora na Raspberry Pi potrebno je spojiti Data pin na pin 22 na Raspberry Pi, napajanje (VCC) na pin 5 V i uzemljenje (GND) na pin GND na Raspberry Pi.

3.2 Spajanje LCD zaslona na Raspberry Pi

LCD zaslon ima 16 pinova koji se spajaju na Raspberry Pi sljedećim redoslijedom:

- a) Pin 1 (GND) je uzemljenje i spaja se na GND pin mikroupravljača
- b) Pin 2 (VCC) je napajanje i spaja se na 5 V pin mikroupravljača
- c) Pin 3 (V0) je ulaz za podešavanje kontrasta. Pozitivan pin potencijometra (10 k Ω) se spaja na 5 V, negativan pin na GND i srednji pin na V0
- d) Pin 4 (Register select) se koristi za odabir između slanja podataka ili naredbi na LCD. Spaja se na GPIO pin 16
- e) Pin 5 (Read/write) se spaja GND i znači da je LCD u načinu pisanja
- f) Pin 6 (Enable) se spaja na pin 17 i omogućuje LCD-u da prihvati podatke s mikroupravljača
- g) Pin 7-10 (D0-D3) su niži 4-bitni podaci i ne koriste se u 4-bitnom načinu rada
- h) Pin 11 (D4) je peti bit podatka i spaja se na GPIO pin 18
- i) Pin 12 (D5) je šesti bit podatka i spaja se na GPIO pin 19
- j) Pin 13 (D6) je sedmi bit podatka i spaja se na GPIO pin 20
- k) Pin 14 (D7) je osmi bit podatka i spaja se na GPIO pin 21
- l) Pin 15 (A) je za pozadinsko osvjetljenje i spaja se na 5 V
- m) Pin 16 (K) je za pozadinsko osvjetljenje i spaja se na GND



Slika 7. Spajanje LCD zaslona na Raspberry Pi [7]

3.3 Spajanje ventilatora na Raspberry Pi

Ventilator ima 2 pina: 5 V i GND. Za kontrolu brzine ventilatora i njegovo sigurno upravljanje koristi se NPN tranzistor 2N2222, 1 k Ω otpornik i dioda 1N4007. Tranzistor se koristi kako ne bi došlo do preopterećenja pina zbog struje koju ventilator zahtjeva. Dioda se koristi za zaštitu tranzistora od povratne struje koja može nastati prilikom isključivanja ventilatora.

3.3.1 Povezivanje

Ventilator: [8]

- a) Pozitivni pin ventilatora spojiti na 5 V pin Raspberry Pi
- b) Negativni pin ventilatora spojiti na kolektor tranzistora

Tranzistor (NPN):

- a) Emiter tranzistora spojiti na GND
- b) Bazu tranzistora spojiti na GPIO pin 12 Raspberry Pi Pico preko 1 k Ω otpornika

Dioda:

- a) Katodu diode spojiti na 5 V
- b) Anodu diode spojiti na kolektor tranzistora

4. PROGRAMSKI DIO

4.1 Thonny program

Thonny je besplatni razvojni program za računala koji je izradio neovisni programer pod istim imenom. To je integrirano razvojno okruženje otvorenog koda (integrated development environment) koje se može koristiti za izradu različitih aplikacija s pomoću programskog jezika Python. [9]

Ključne Značajke Thonny IDE-a:

- a) Jednostavno sučelje: Thonny nudi minimalističko i jasno sučelje koje ne preopterećuje korisnika nepotrebnim funkcionalnostima. To je korisno za početnike, jer omogućuje fokusiranje na osnovne koncepte programiranja bez ometanja.
- b) Ugrađeni Python tumač: Thonny dolazi s ugrađenim Python tumačem, što znači da korisnici ne moraju zasebno instalirati Python na svoj sustav. Kada instalirate Thonny, Python tumač dolazi zajedno s njim, omogućujući vam da odmah započnete s radom.
- c) Korak-po-korak izvršavanje: Jedna od najmoćnijih značajki Thonny-a je mogućnost korak-po-korak izvršavanja koda. To omogućuje korisnicima da prate kako se kod izvodi liniju po liniju, što pomaže u razumijevanju tijeka programa i identifikaciji grešaka.
- d) Pregled varijabli: Thonny nudi jasan pregled svih varijabli koje su definirane u programu, zajedno s njihovim trenutnim vrijednostima. Ovo je izuzetno korisno pri učenju kako se promjenjive mijenjaju tijekom izvršavanja koda.
- e) Jednostavno debugiranje: Thonny ima ugrađen alat za otklanjanje grešaka (debugger) koji korisnicima omogućuje lako otkrivanje i ispravljanje grešaka u kodu. Moguće je postaviti prekidne točke i korak po korak pratiti što se događa tijekom izvršavanja.
- f) Podrška za MicroPython i Raspberry Pi: Thonny također podržava rad s MicroPythonom, što je korisno za rad s mikroupravljačima kao što su Raspberry Pi Pico, ESP8266, i ESP32. Ovo omogućava korisnicima da pišu, otklanjaju pogreške i prenose kod izravno na uređaj, sve unutar Thonny.

- g) Interaktivna konzola: Thonny uključuje Python REPL (Read-Eval-Print Loop) konzolu koja omogućava korisnicima da izvršavaju pojedinačne naredbe i odmah vide rezultate. Ovo je korisno za testiranje malih dijelova koda ili eksperimentiranje s različitim Python funkcijama.
- h) Jednostavno upravljanje modulima: Thonny olakšava instalaciju i upravljanje Python paketima i modulima putem ugrađenog alata za upravljanje paketima (pip). Korisnici mogu instalirati dodatne knjižnice izravno unutar Thonny sučelja.

Thonny je dostupan za više platformi uključujući Windows, macOS i Linux, a instalacija je vrlo jednostavna.

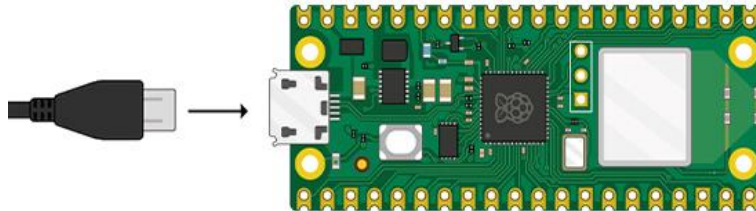
Thonny je posebno popularan među studentima, edukatorima i početnicima zbog svoje jednostavnosti i funkcionalnosti koja odgovara njihovim potrebama. S obzirom na to da je Thonny besplatan i otvorenog koda, dostupan je svima koji žele učiti Python bez potrebe za složenim konfiguracijama ili dodatnim troškovima.

4.2 Instalacija potrebnog softvera na Raspberry Pi

Za postavljanje Raspberry Pi potrebno je skinuti najnoviju verziju firmwarea te instalirati na Raspberry Pi. [10]

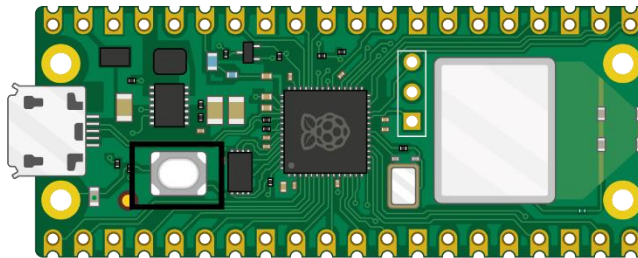
Postupak instalacije:

- a) Spojiti mikro usb kabel u Raspberry Pi



Slika 8. Spajanje mikro usb kabela u Rapsberry Pi

- b) Držati pritisnutu tipku BOOTSEL



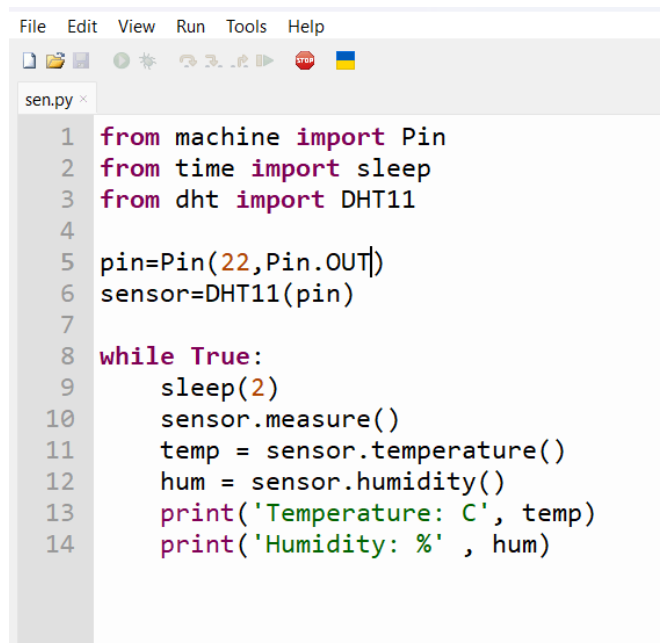
Slika 9. Pritisak tipke BOOTSEL

- c) Spojiti drugi kraj u računalo
- d) Otvori se upravitelj datoteka i Raspberry Pi Pico se prikaže kao vanjski pogon. Tad se prebaci firmware na Raspberry Pi i odspoji usb kabel
- e) Nakon toga se otvori Thonny i poveže se s Raspberry Pi
- f) Potrebno je još instalirati MicroPython biblioteku **picozero** i može se program koristiti

4.3 Programiranje u Thonny

4.3.1 Čitanje podataka sa senzora DHT11

Za korištenje senzora DHT11 potrebno je preuzeti pravilnu biblioteku koja definira funkcije potrebne za korištenje senzora (spremljena pod nazivom **dht.py**). Nakon preuzimanja biblioteke može se početi s pisanjem skripte u programu Thonny.



```
File Edit View Run Tools Help
sen.py x
1 from machine import Pin
2 from time import sleep
3 from dht import DHT11
4
5 pin=Pin(22,Pin.OUT)
6 sensor=DHT11(pin)
7
8 while True:
9     sleep(2)
10    sensor.measure()
11    temp = sensor.temperature()
12    hum = sensor.humidity()
13    print('Temperature: C', temp)
14    print('Humidity: %' , hum)
```

Slika 10. Program za upravljanje senzorom DHT11 [11]

Iz glavne biblioteke **machine** potrebno je pozvati funkciju Pin koja omogućava korištenje GPIO pinova Raspberry Pi te iz preuzete biblioteke **dht.py** pozvati funkcije senzora.

Objašnjenje programa:

- `pin = Pin(22, Pin.OUT)`: Postavlja pin broj 22 na mikroupravljaču kao izlazni pin. To je pin na koji se priključuje DHT11 senzor.
- `sensor = DHT11(pin)`: Inicijalizacija DHT11 senzora na odabranom pinu.
- `while True:`: Stvara beskonačnu petlju koja kontinuirano očitava podatke sa senzora.
- `sleep(2)`: Pauzira izvršavanje programa na 2 sekunde između svakog očitavanja. To je važno jer DHT11 senzor ne može očitavati podatke više od jednom svake sekunde.

- e) `sensor.measure()`: Inicira mjerenje temperature i vlažnosti.
- f) `temp = sensor.temperature()`: Dohvaća izmjerenu temperaturu u Celzijusima.
- g) `hum = sensor.humidity()`: Dohvaća izmjerenu vlažnost u postocima.
- h) `print('Temperature: °C', temp)`: Ispisuje izmjerenu temperaturu u konzolu.
- i) `print('Humidity: %', hum)`: Ispisuje izmjerenu vlažnost u konzolu.

4.3.2 Upravljanje ventilatorom

Za kontrolu ventilatora spojenog na Raspberry Pi potrebno je prvo pozvati iz biblioteke **machine** funkcije Pin i PWM. Promjena brzine ventilatora može se postići s pomoću PWM (Pulse Width Modulation) signala. PWM omogućuje kontrolu brzine ventilatora reguliranjem količine vremena tijekom kojeg je signal visok (uključen) u odnosu na ukupni ciklus. PWM radi tako što brzo uključuje i isključuje napon na izlazu, a brzina ventilatora zavisi od radnog ciklusa (duty cycle), tj. postotka vremena tijekom kojeg je signal visok u jednom PWM ciklusu:

- a) 0 % radnog ciklusa: ventilator je isključen.
- b) 50 % radnog ciklusa: ventilator radi na pola brzine.
- c) 100 % radnog ciklusa: ventilator radi punom brzinom.

Radni ciklus ima raspon od 0 do 65535 jer Raspberry Pi koristi 16-bitnu preciznost. Da bi se dobio postotak brzine ventilatora potrebno je radni ciklus podijeliti sa 100.

Potrebno je definirati pin na koji je spojena baza tranzistora (Pin 12) te definirati frekvenciju ventilatora (`fan.freq()`), ventilatori obično rade najbolje pri PWM frekvencijama između 20 kHz i 25 kHz). Nakon toga se postavlja while petlja u kojoj se poziva funkcija `set_fan_speed` i unosi postotak brzine ventilatora za temperaturni raspon koji želimo (npr. ispod 23 °C brzina ventilatora je na 50 % jer nije potrebno jako hlađenje).

```
File Edit View Run Tools Help
sen.py vent.py
1 from machine import Pin, PWM
2 import time
3
4 fan = PWM(Pin(12))
5 fan.freq(25000)
6
7 def set_fan_speed(duty_cycle):
8     fan.duty_u16(int(duty_cycle * 65535 / 100))
9
10 while True:
11     if temp <= 23:
12         set_fan_speed(50)
13     elif temp > 23 and temp<=26 :
14         set_fan_speed(75)
15     elif temp > 26:
16         set_fan_speed(100)
17
```

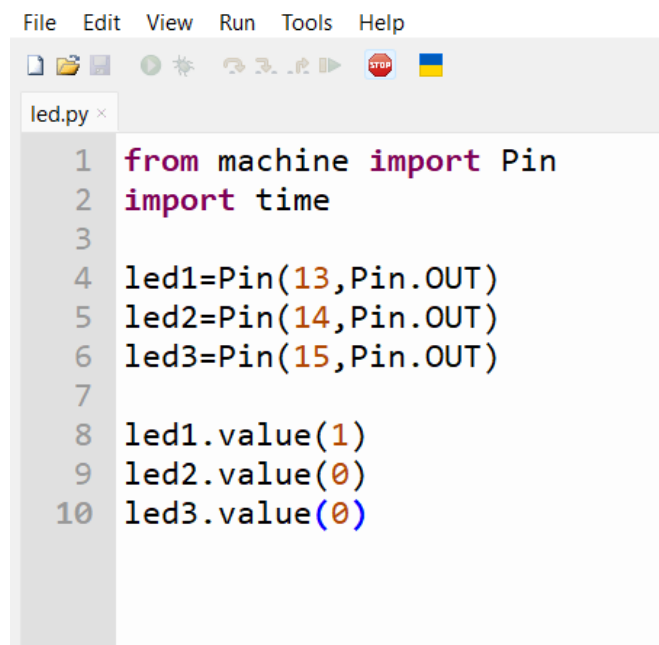
Slika 11. Program za kontrolu brzine ventilatora [12]

4.3.3 Upravljanje LED elementima

Za upravljanje LED elementima potrebno je pozvati funkciju Pin te definirati pinove na koje su spojene 3 LED elementa (zelena, žuta i crvena) te ih definirati kao izlaz (Pin.OUT). Svaka od LED elemenata predstavlja temperaturno područje:

- a) Zelena: ispod 23 °C
- b) Žuta: između 23 °C i 26 °C
- c) Crvena: iznad 26 °C

LED elementi se uključuju funkcijom `led.value()` i postavljanjem stanja 1 (ako želimo da se dioda uključi) ili stanje 0 (da se dioda isključi). Ovaj primjer prikazuje kako se mogu kontrolirati LED elementi na mikroupravljaču i kako ih možete koristiti za jednostavne ili složene signalizacije.

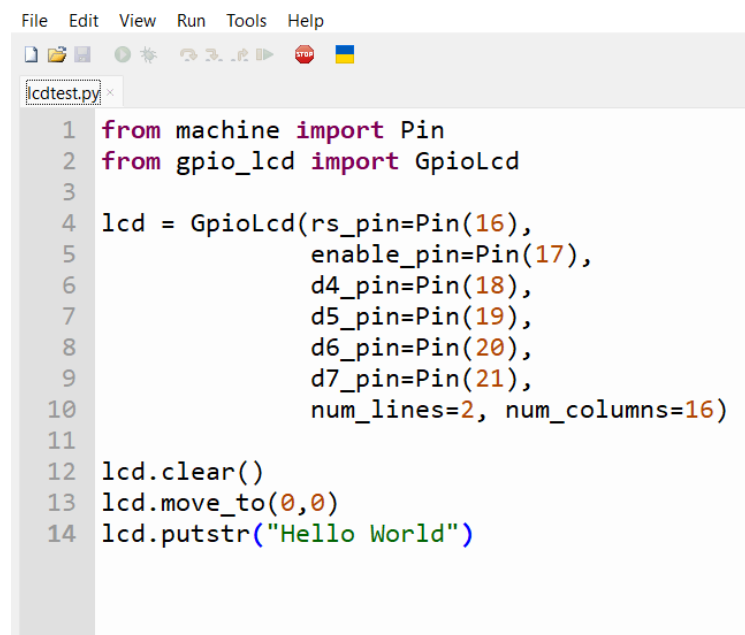


```
File Edit View Run Tools Help
led.py x
1 from machine import Pin
2 import time
3
4 led1=Pin(13,Pin.OUT)
5 led2=Pin(14,Pin.OUT)
6 led3=Pin(15,Pin.OUT)
7
8 led1.value(1)
9 led2.value(0)
10 led3.value(0)
```

Slika 12. Program za kontrolu LED elemenata [13]

4.3.4 Upravljanje LCD zaslonom

Za upravljanje LCD zaslonom prvo je potrebno preuzeti i instalirati 2 biblioteke koje definiraju funkcije LCD-a i omogućuju njihovo korištenje, a to su **gpio_lcd.py** i **lcd_api.py**. Nakon toga se definira sam LCD i pinovi spojeni na Raspberry Pi. Funkcija `lcd.clear()` briše LCD zaslon (uklanja sve prethodne znakove), a funkcija `lcd.move_to()` pomiče kursor na željeno mjesto (stupac, redak). Ovaj kod može biti osnova za razne projekte gdje je potrebno prikazivati informacije na LCD zaslonu, kao što su sustavi za praćenje temperature, vremena, kontroleri za kućne uređaje, ili drugi interaktivni projekti.

A screenshot of a code editor window titled 'lcdtest.py'. The editor shows Python code for initializing and controlling an LCD display. The code includes imports for 'Pin' from 'machine' and 'GpioLcd' from 'gpio_lcd'. It then creates a 'GpioLcd' object with specific pin configurations and parameters. Finally, it calls 'clear()', 'move_to(0,0)', and 'putstr("Hello World")' on the LCD object.

```
1 from machine import Pin
2 from gpio_lcd import GpioLcd
3
4 lcd = GpioLcd(rs_pin=Pin(16),
5             enable_pin=Pin(17),
6             d4_pin=Pin(18),
7             d5_pin=Pin(19),
8             d6_pin=Pin(20),
9             d7_pin=Pin(21),
10            num_lines=2, num_columns=16)
11
12 lcd.clear()
13 lcd.move_to(0,0)
14 lcd.putstr("Hello World")
```

Slika 13. Program za upravljanje LCD zaslonom [14]

5. KONAČNI PROGRAM AUTOMATSKOG UPRAVLJANJA VENTILATORA

Nakon spajanja svih komponenti na eksperimentalnoj pločici i programa za svaku korištenu komponentu, završni program izgleda ovako:

```
File Edit View Run Tools Help
[ main.py ] * x
1 from machine import Pin,PWM
2 from time import sleep
3 from dht import DHT11
4 from gpio_lcd import GpioLcd
5
6 fan = PWM(Pin(12))
7 fan.freq(25000)
8
9 led1=Pin(13,Pin.OUT)
10 led2=Pin(14,Pin.OUT)
11 led3=Pin(15,Pin.OUT)
12
13 lcd = GpioLcd(rs_pin=Pin(16),
14             enable_pin=Pin(17),
15             d4_pin=Pin(18),
16             d5_pin=Pin(19),
17             d6_pin=Pin(20),
18             d7_pin=Pin(21),
19             num_lines=2, num_columns=16)
20
21 pin=Pin(22,Pin.OUT)
22 sensor=DHT11(pin)
23
24 lcd.clear()
25 def set_fan_speed(duty_cycle):
26     fan.duty_u16(int(duty_cycle * 65535 / 100))
27
```

Slika 14. Konačni program


```
File Edit View Run Tools Help
[ main.py ] x
27
28 while True:
29     sensor.measure()
30     temp = sensor.temperature()
31     hum = sensor.humidity()
32     if temp <= 23:
33         led1.value(1)
34         led2.value(0)
35         led3.value(0)
36         set_fan_speed(50)
37     elif temp > 23 and temp<=26 :
38         led1.value(0)
39         led2.value(1)
40         led3.value(0)
41         set_fan_speed(75)
42     elif temp > 26:
43         led1.value(0)
44         led2.value(0)
45         led3.value(1)
46         set_fan_speed(100)
47     lcd.move_to(0,0)
48     lcd.putstr("Temperatura:"+str(temp)+chr(223)+"C")
49     sleep(5)
50     lcd.clear()
51     lcd.move_to(0,0)
52     lcd.putstr("Vlaznost: "+str(hum)+"%")
53     sleep(5)
```

Slika 15. Konačni program

U programu se za ispis temperature još i koristi znak stupnja (naredba `chr(223)`). Namješteno je da se očitane vrijednosti temperature i vlažnosti prostoriije izmjenjuju svakih 5 sekundi. Kada je temperatura prostoriije veća od 26 stupnjeva, pali se crveni LED element i brzina ventilatora je na 100 %. Kada je temperatura prostoriije između 23 i 26 stupnjeva, tada se upali žuti LED element i brzina ventilatora je na 75 %, a kada je temperatura ispod 23 stupnja, upali se zeleni LED element i brzina ventilatora je na 50 %.

6. ZAKLJUČAK

Primjena Raspberry Pi kao upravljačkog sustava u automatizaciji hlađenja pruža jednostavno, fleksibilno i pristupačno rješenje za kontrolu temperature. Iako se Raspberry Pi koristi uglavnom za edukacijske i prototipske projekte, ovaj rad će pokazati njegovu primjenjivost u realnim uvjetima, posebice u manjim sustavima gdje su troškovna efikasnost i fleksibilnost ključni faktori.

Rad također naglašava potencijal automatiziranih sustava u svakodnevnim primjenama, gdje se jednostavni sustavi mogu koristiti za optimizaciju potrošnje energije i povećanje operativne sigurnosti uređaja.

7. LITERATURA

[1] Raspberry Pi, <https://www.raspberrypi.com/>

Pristupljeno 10.08.2024.

[2] Raspberry Pi, <https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html>

pristupljeno 10.08.2024.

[3] Velleman.eu, <https://www.velleman.eu/products/view/?id=459216>

pristupljeno 10.08.2024.

[4] Sunon.com,

[https://www.sunon.com/en/MANAGE/Docs/PRODUCT/286/481/SUNON%20DC%20Brushless%20Fan%20&%20Blower\(255-A\).pdf](https://www.sunon.com/en/MANAGE/Docs/PRODUCT/286/481/SUNON%20DC%20Brushless%20Fan%20&%20Blower(255-A).pdf)

pristupljeno 10.08.2024.

[5] Circuitdigest.com, <https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet>

pristupljeno 12.08.2024.

[6] Instructables.com, <https://www.instructables.com/DHT11-With-Raspberry-Pi-Pico/>

Pristupljeno 12.08.2024.

[7] How2electronics.com, <https://how2electronics.com/interfacing-16x2-lcd-display-with-raspberry-pi-pico/>

Pristupljeno 12.08.2024.

[8] Core Electronics Forum, <https://forum.core-electronics.com.au/t/controlling-a-pi-fan-on-the-pi-pico-w/15778>

Pristupljeno 15.08.2024.

[9] Thonny.org, <https://thonny.org/>

Pristupljeno 15.08.2024.

[10] Raspberry Pi, <https://projects.raspberrypi.org/en/projects/get-started-pico-w/1>

Pristupljeno 15.08.2024.

[11] Instructables.com, <https://www.instructables.com/DHT11-With-Raspberry-Pi-Pico/>

Pristupljeno 20.08.2024.

[12] Raspberry Pi Forums, <https://forums.raspberrypi.com/viewtopic.php?t=310374>

Pristupljeno 20.08.2024.

[13] Raspberry Pi Projects,

<https://projects.raspberrypi.org/en/projects/introduction-to-the-pico/7>

Pristupljeno 20.08.2024.

[14] Instructables.com,

<https://www.instructables.com/Raspberry-Pi-Pico-and-16x2-LCD/>

Pristupljeno 20.08.2024.

8. PRILOZI

POPIS SLIKA

Slika 1. Raspberry Pi

Slika 2. Raspberry Pi Pico W

Slika 3. DHT11 senzor

Slika 4. Ventilator

Slika 5. LCD zaslon 16x2

Slika 6. Spajanje DHT11 na Raspberry Pi

Slika 7. Spajanje LCD zaslona na Raspberry Pi

Slika 8. Spajanje mikro usb kabela u Raspberry Pi

Slika 9. Pritisak tipke BOOTSEL

Slika 10. Program za upravljanje senzorom DHT11

Slika 11. Program za kontrolu brzine ventilatora

Slika 12. Program za kontrolu LED elemenata

Slika 13. Program za upravljanje LCD zaslonom

Slika 14. Konačni program

Slika 15. Konačni program

POPIS TABLICA

Tablica 1. Naredbe za LCD instrukcijski registar