

Video strežnik za detekcijo premika ter merjenje temperature

Avtor: Timotej Petrovčič

Mentorja: prof. dr. Tadej Tuma, as. dr. Jernej Olenšek

Predmet: Načrtovanje in programiranje vgrajenih sistemov

Datum: 20 Junij 2022

Kazalo:

1	Namen	1
2	Tehnične specifikacije	1
3	Komponente	1
4	Shema povezav	2
5	Programska izvedba	2
6	Diagrami poteka	3
7	Težave ob implementaciji	4
8	Možne izboljšave	4
9	Viri in literatura	4

Ključne besede: ESP32, RTOS, Jedro, Nit

1 Namen

Namen projekta je izdelava video strežnika, ki omogoča sočasno delovanje prenosa v živo ter branja senzorja temperature in premika. Strežnik omogoča sočasno povezljivost večih naprav ter ustvari dve ločeni lokalni spletni strani, prva za video prenos, druga za podatke o premiku in temperaturi.

Končni cilj je modularen sistem, ki ga povežemo na Home Assistant strežnik, ki omogoča integracijo večih ločenih pametnih naprav ter obdelavo podatkov (zaznavanje prihoda avtomobila ter branje registrskih tablic).

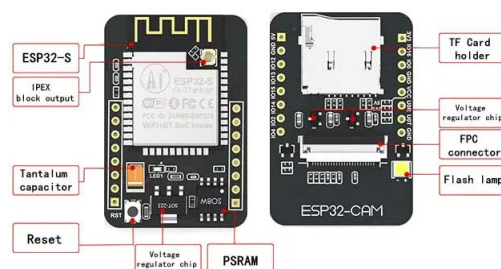
2 Tehnične specifikacije

- Video prenos do 25 FPS
- Merjenje temperature in zaznavanje premika
- Vklop svetlobne diode ob zaznanem premiku (Nočni način)
- Možnost povezljivosti 10 naprav hkrati (padec FPS)

3 Komponente

Mikrokrmilniški sistem s kamero

Uporabili smo mikrokrmilniški sistem ESP32, zaradi omogočanja različnih brezžičnih komunikacij in sicer različico ESP32-CAM, ki ima poleg osnovne različice sistema vgrajeno kamero ter svetlečo diodo.



Slika 1: Sprednja in zadnja stran ESP32-CAM mikrokrmilnika

PIR senzor

Uporabili smo že izdelan PIR senzor, ki omogoča spreminjanje občutljivosti tipala ter trajanja senzorkega pulza.



Slika 2: Sprednja in zadnja stran PIR senzorja

Temperaturni senzor

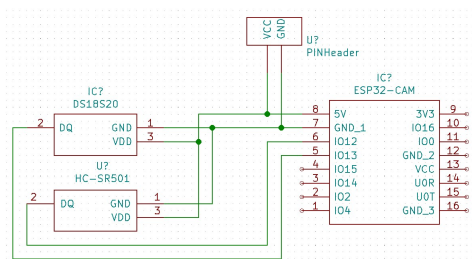
Uporabili smo digitalni temperaturni senzor Dallas DS18S20, ki omogoča hitro branje temperature ter ga lahko dobimo v pakiranju primernem za zunanjo namestitev.



Slika 3: Temperaturni senzor Dallas DS18S20

4 Shema povezav

Pri vezavi senzorjev na mikrokrmilniški sistem je bilo potrebno paziti predvsem na že uporabljene priključke mikrokrmilnika za vgrajeno kamero. Zaradi neuporabe modula z SD kartico smo porabili priključka 12 in 13, ki sta za to namenjena.



Slika 4: Preprosta vezava mikrokrmilnika in senzorjev

5 Programska izvedba

Prvotno je bilo potrebno zastaviti primerno obremenitev mikrokrmilniškega sistema, saj ima le-ta 2 jedra, procesno jedro ter aplikacijsko jedro. Izvajanje video strežnika smo dodelili aplikacijskem jedru, izvajanje podatkovnega strežnika, prikaz temperature in zaznavanje premika, pa smo dodelili procesnem jedru. Običajno se vsa opravila izvaja na aplikacijskem jedru, saj je le-ta za to namenjen, procesno jedro pa je prvotno namenjeno izvajanju protokolov, ki jih uporabimo, v našem primeru je to WIFI povezava, zato lahko na procesorsko jedro dodamo manjše opravilo.

Razdelitev opravil po jedrih:

- Procesno jedro:
 - *taskHandler_get_sensorTemp* (Vezan na funkcijo *sensorTemp*)
 - *taskHandler_get_sensorPIR* (Vezan na funkcijo *sensorPIR*)
- Aplikacijsko jedro:
 - *taskHandler_camera_scraper* (Vezan na funkcijo/nit *camCB*)
 - *taskHandler_stream_frames_to_client* (Vezan na funkcijo/nit *streamCB*)
 - *taskHandler_mjpeg_server* (Vezan na funkcijo/nit *mjpegCB*)

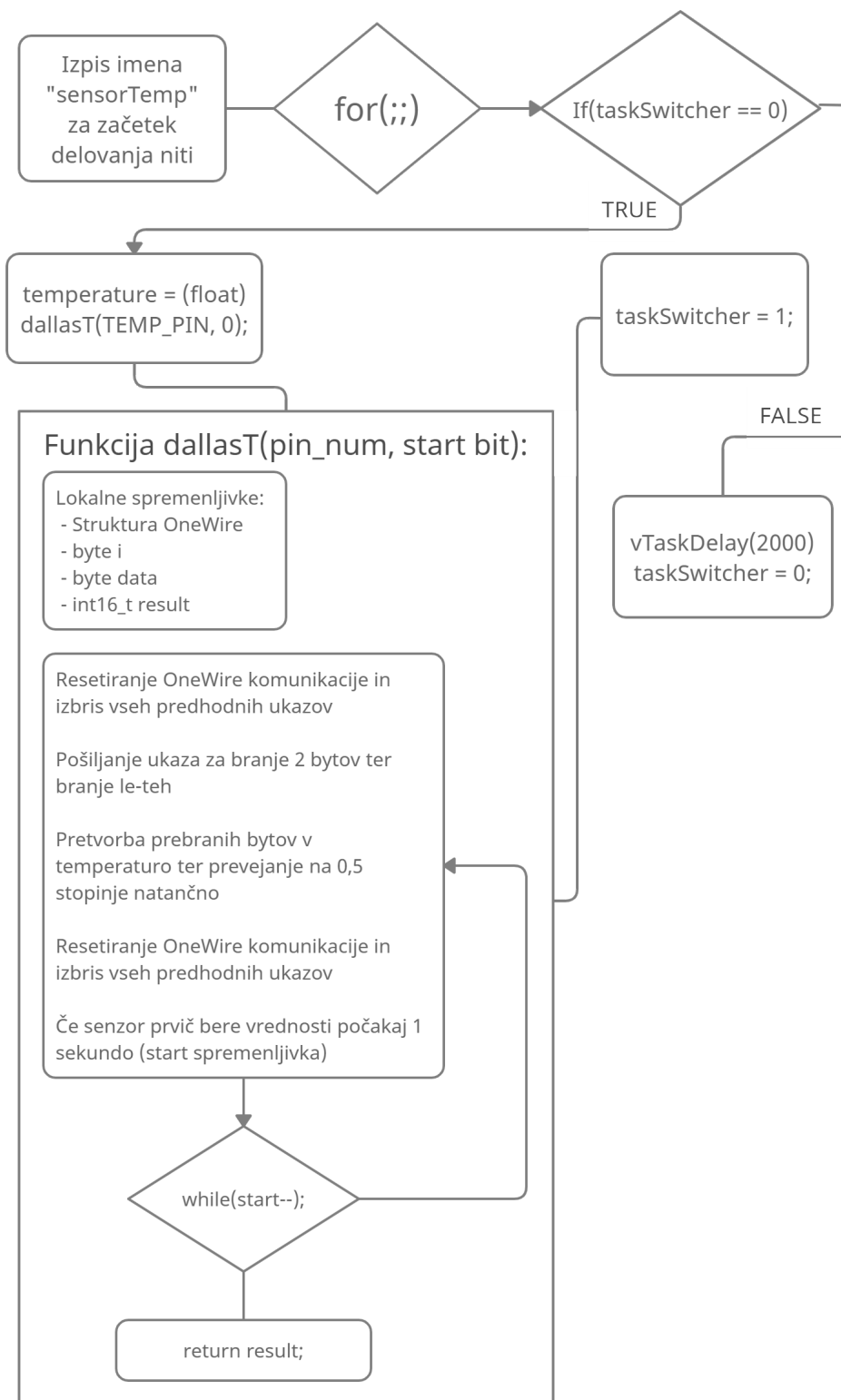
Poleg omenjenega je za pravilno delovanje potrebno dodati še dve ključni strukturi:

- Vrsto povezanih naprav (queue)
- Semafor za prekinitev opravil ob pisanju slike v začasni pomnilnik (Semaphore)

Za praktično izvedbo smo uporabili FreeRTOS v kombinaciji s sistemskimi knjižnicami.

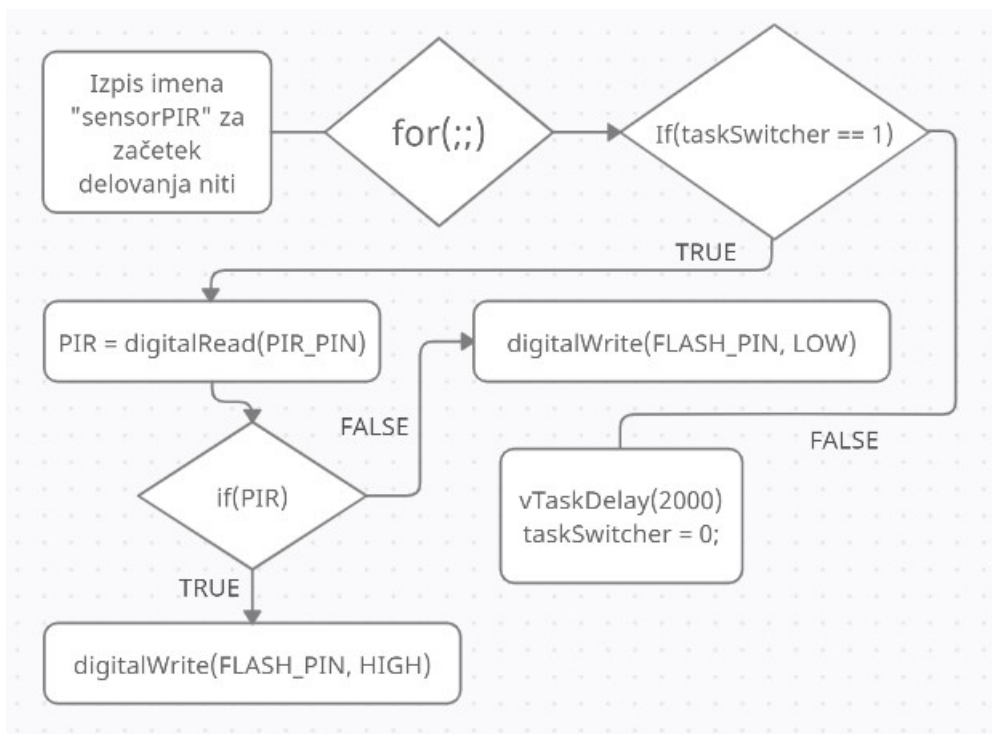
6 Diagrami poteka

Funkcija/nit sensorTemp



Slika 5: Diagram poteka sensorTemp

Funkcija/nit sensorPIR



Slika 6: Diagram poteka sensorPIR

7 Težave ob implementaciji

Glavno težavo implementacije je povzročalo izvajanje branja senzorjev na procesnem jedru, saj se le-to relativno hitro nasiči in strežnik prekine z normalnim delovanjem, saj ne more izvajati vseh operacij hkrati. Prvotno smo preizkusili implementacijo z DHT11 senzorjem temperature in vlage, a se je pri prepočasni frekvenci vzorčenja meritev izkazalo, da senzor ni primeren. Zamenjali smo ga z Dallas DS18S20 senzorjem, ki omogoča hitrejša merite, a smo naleteli na težavo implementacije hitrejšega algoritma, saj standardna readTemperature funkcija ni dovolj hitra. Poiskali smo alternativno implementacijo, ki meritev izvede hitreje.

8 Možne izboljšave

Zagotovo je mogoča optimizacija izvajanja opravil na procesnem jedru z dodatnim semaforjem, ki zaznava zasedenost WIFI povezave ter prilagodi pogostost branja snzorjev premika temperature. Kljub temu mora branje ostati dovolj pogosto, da premik pravočasno zazna. Posledično lahko tudi ločimo branje temperature in zaznavnaje premika na različne prioritete izvajanja, kar nam FreeRTOS tudi omogoča.

9 Viri in literatura

- ESP32 FreeRTOS documentation, *Pridobljeno dne: 20.6.2022*
- ESP32-CAM mjpeg streaming multiclient, *Pridobljeno dne: 20.6.2022*
- Dallas DS18S20 faster reading times, *Pridobljeno dne: 20.6.2022*