

Server Room Guardian

Robu Tudor-Sebastian

Introducere

Server Room Guardian este un sistem embedded de monitorizare și control pentru infrastructura personală de acasă, construit în jurul unui microcontroller ATmega328P-XMINI și al unui modul WiFi ESP8266 pe placă NodeMCU V2.

Proiectul monitorizează periodic temperatura și umiditatea din carcasa/server rack-ul în care se află Raspberry Pi-ul și trimite aceste date către un VPS personal. În același timp, sistemul verifică dacă pe server există comenzi noi, cum ar fi pornirea sau oprirea Raspberry Pi-ului.

Feedback-ul local este oferit prin LED-uri, astfel încât starea sistemului să fie vizibilă imediat:

- Raspberry Pi online sau offline;
- temperatură normală sau critică;
- comunicare activă cu serverul;
- stare de eroare sau avertizare.

Ideea proiectului a pornit de la nevoia de a supraveghea de la distanță infrastructura personală de acasă, în special un Raspberry Pi folosit ca server. În utilizarea zilnică, pot apărea situații în care temperatura din carcasă crește prea mult, conexiunea se pierde sau Raspberry Pi-ul trebuie repornit fără acces fizic direct.

Sistemul este util deoarece permite:

- monitorizarea condițiilor de funcționare ale serverului;
- detectarea rapidă a unei temperaturi critice;
- controlul de la distanță al alimentării Raspberry Pi-ului;
- feedback vizual local prin LED-uri;
- integrarea cu un VPS personal pentru logare și comenzi remote.

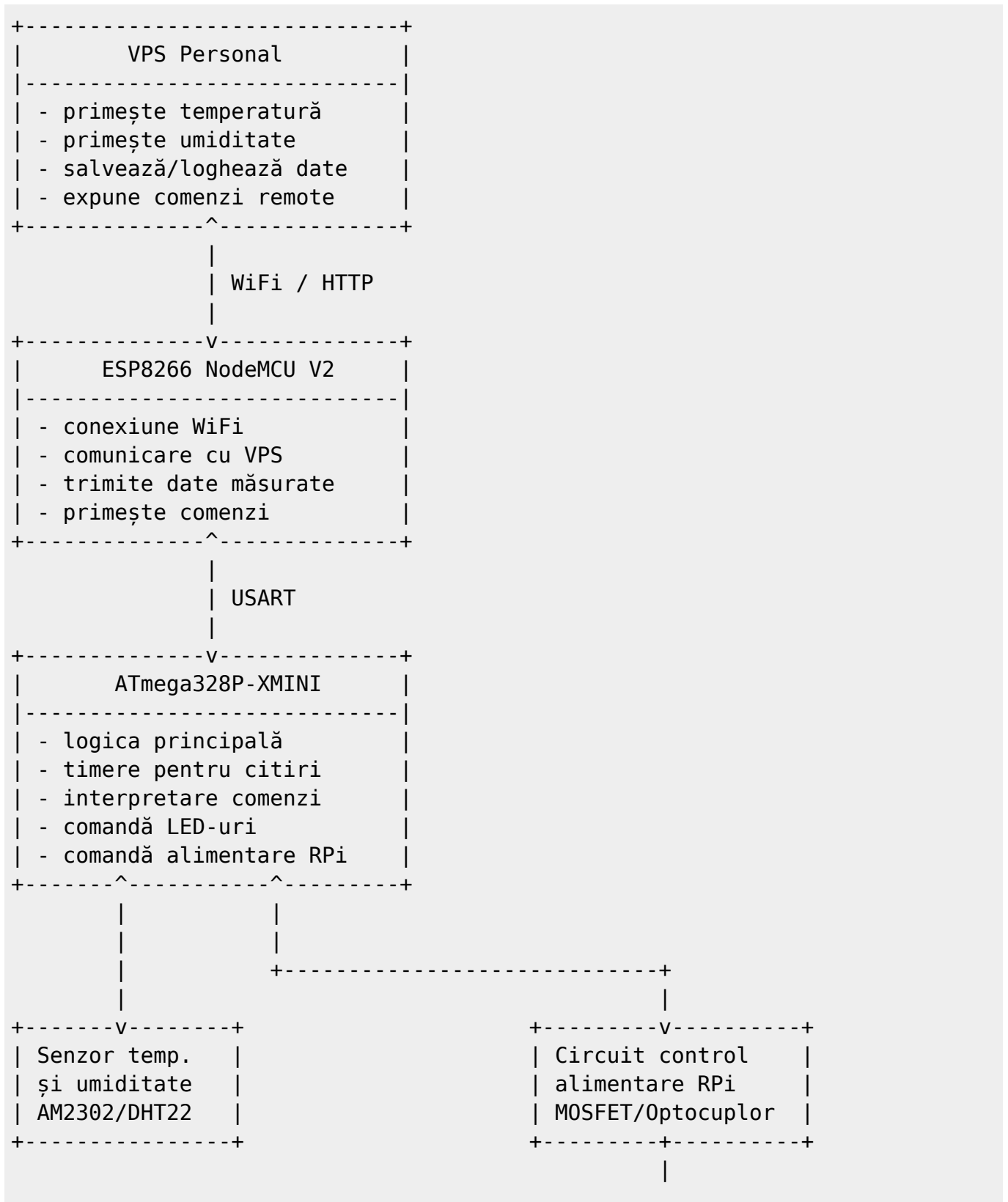
Descriere generală

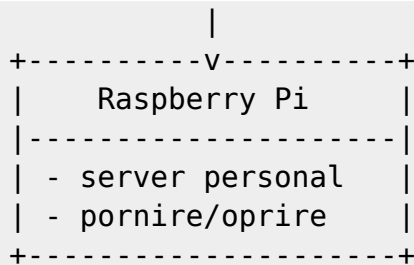
Sistemul este împărțit în două componente principale: partea hardware locală, aflată lângă Raspberry Pi, și partea software/server, aflată pe un VPS personal.

Partea locală este formată din microcontrollerul ATmega328P-XMINI, modulul ESP8266 NodeMCU V2,

senzorul de temperatură și umiditate, LED-urile de stare și circuitul de comandă pentru alimentarea Raspberry Pi-ului. Microcontrollerul citește periodic valorile de temperatură și umiditate, interpretează starea sistemului și comandă LED-urile. Pentru conectarea la Internet, acesta comunică prin USART cu modulul ESP8266.

Modulul ESP8266 are rolul de interfață WiFi. Acesta transmite datele către VPS și verifică dacă există comenzi noi. Comenzile primite de la server sunt trimise către ATmega328P, care decide dacă trebuie pornit sau oprit Raspberry Pi-ul.





Fluxul principal de funcționare este următorul:

1. ATmega328P pornește și inițializează perifericele: USART, timere, pini GPIO și eventual ADC/PWM.
2. Senzorul de temperatură și umiditate este citit periodic.
3. Valorile sunt trimise prin USART către ESP8266.
4. ESP8266 trimite datele către VPS prin WiFi.
5. VPS-ul salvează datele și poate furniza o comandă nouă.
6. ESP8266 primește comanda și o transmite către ATmega328P.
7. ATmega328P execută acțiunea cerută: pornire/oprire Raspberry Pi sau actualizare stare.
8. LED-urile sunt actualizate în funcție de starea Raspberry Pi-ului și de temperatura măsurată.

Hardware Design

Componentele hardware folosite în proiect sunt:

Nr.	Componentă	Rol în proiect
1	ATmega328P-XMINI	Microcontroller principal; controlează logica sistemului, citește senzorii și comandă ieșirile
2	ESP8266 NodeMCU V2	Modul WiFi folosit pentru comunicarea cu VPS-ul personal
3	Senzor temperatură și umiditate AM2302	Măsoară temperatura și umiditatea din carcasa serverului
4	LED-uri	Feedback vizual pentru starea Raspberry Pi-ului și temperatura sistemului
5	Rezistențe 220Ω	Limitarea curentului
6	MOSFET sau optocuplor	Controlul fizic al alimentării/pornirii/opririi Raspberry Pi-ului
7	Breadboard	
8	Fire jumper	
9	Sursă de alimentare (3.3V)	
10	Raspberry Pi	

Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)


- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/ciprian.popescu0411/tudor.robu>



Last update: **2026/05/09 21:12**