

CasaMonitor

Student: Justin-Marian Popescu

Grupa: 333CD

Îndrumător: Florin-Alexandru Stancu

Introducere

Inspirația pentru **“CasaMonitor”** a venit din nevoia de a-mi îmbunătăți confortul din cameră. Acest proiect îmi permite să ajustez condițiile ambientale direct de pe computer, ceea ce face munca de acasă mai eficient energetic. Proiectul ajustează consumul de energie \propto în funcție de necesitățile exacte în **timp real**.

Ideea Principală

Proiectul **CasaMonitor** are ca scop furnizarea unei soluții integrate pentru monitorizarea și controlul condițiilor ambientale și al consumului energetic într-un spațiu locuit sau des frecventat.

Sistemul utilizează senzori și un μC conectat la un dispozitiv prin intermediul unui cablu **USB 2.0 de tip A-B** pentru a colecta date în timp real despre temperatură, umiditate și consumul de energie al dispozitivelor.

Scopul Proiectului

Scopul proiectului este acela de a oferi utilizatorilor un control mai bun asupra mediului în care își desfășoară activitățile zilnice, optimizând **confortul și eficiența energetică**.

Utilitatea Monitorului

Utilizarea panoului de control **CasaMonitor** este multiplă:

- **îmbunătățirea confortului** prin ajustarea automată a temperaturii și umidității;
- **monitorizarea și gestionarea eficientă** a consumului pe locuință;
- **optimizarea** utilizării resurselor disponibile;

Descriere Generală

Sistemul folosește **Node-RED** pentru a gestiona logica și fluxul datelor între dispozitive. Acest instrument de programare conectează dispozitivele hardware. L-am ales pentru facilitățile pe care le oferă ca editor bazat pe browser pentru conectarea fluxurilor folosind noduri, care pot fi implementate în timpul execuției.



Sistemul captează datele primite de la senzori înainte de procesare. Acest lucru asigură că nu se pierd date în timpul perioadelor în care acestea sunt primite de la senzori. Sistemul comunică cu **Node-RED API** prin **USB**.

Control Sistem de Răcire (USART)

Senzorul de temperatură și umiditate **DHT22** măsoară valorile de temperatură și umiditate din încăpere. Acesta este conectat la placa Arduino și furnizează date digitale despre temperatura și umiditate. Pe baza acestor date, algoritmul de control va stabili dacă temperatura este în afara intervalului stabilit. **Ventilatorul** se pornește pe baza variației dintre temperatura reală și cea dorită.

Gestiunea Energiei (I2C, PWM și Timere)

Consumul de energie este gestionat prin **protocolul I2C**, această integrare va facilita implementarea funcțiilor de economisire a energiei, cum ar fi diminuarea și stingerea automată a luminilor atunci când nu este detectată nicio mișcare.

Luminile de bandă **LED** sunt controlate prin intermediul unui **întrerupător fizic ON-OFF**, detectarea mișcării sau a obstacolelor puse în fața senzorului infraroșu.

Modulul **IRF520**, controlat prin semnale **PWM**, gestionează alimentarea benzii **LED** în funcție de intrările de la **senzorul IR** sau de la **întrerupătorul fizic**.

Timerele sunt folosite pentru a controla durata în care luminile rămân aprinse după detectarea ultimei mișcări, asigurând o utilizare eficientă a energiei. Pentru măsurare a **curentului/tensiunii** în timp real folosesc un **timer** pentru a înregistra frecvența datelor, sistemul poate identifica rapid orice schimbare în consumul de energie, pentru a lua decizii rapide.

Transmitere/Primire Date (USB)

Comunicarea serială USB cu placa **Arduino UNO** asigură transferul de date fiabil. Această configurare mă va ajuta să actualizez și să manipulez ușor setările direct din interfața cu utilizatorul.

Toate datele sunt colectate de la senzori, apoi semnalele de control sunt trimise către un **port USB** al laptopului. Interfața laptopului va permite vizualizarea datelor înregistrate de senzori în timp real și ajustarea setărilor de sistem, cum ar fi **pragurile de temperatură și umiditate**.

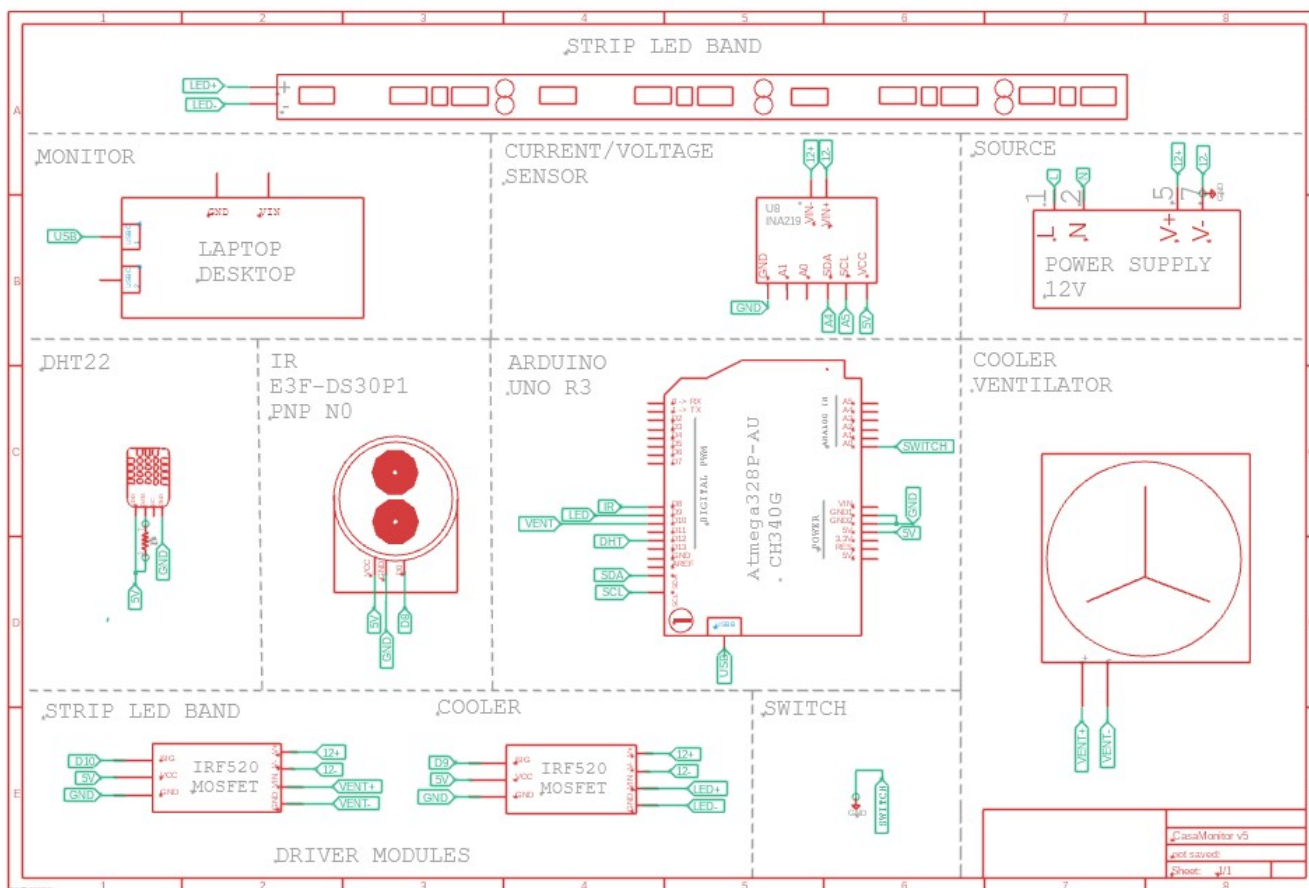
Schema bloc este disponibilă la următorul link, care ilustrează modul în care **μC** interacționează cu senzorii și restul dispozitivelor din sistem. Schema oferă o vedere completă asupra interacțiunii HW și SW din cadrul proiectului, prezentând fluxul de date și control între diferitele componente:

[CasaMonitor Flow](#)
Hardware Design

Schema Electrică

Pentru a fi mai ușor de urmărit cablajul, am încercat să realizez un design cât mai modular cu puțință pentru fiecare modul: de la sursa de alimentare, la senzori și **μC**. În plus, pentru a clarifica modul în care se vor distribui voltajele, întregul sistem va deveni mult mai accesibil pentru modificările ulterioare.

Toți **senzorii** vor fi alimentați la **12V** de la **sursa de tensiune**, comparativ cu **μC**, care va fi conectat la laptop prin **USB**, de unde primește **5V**.

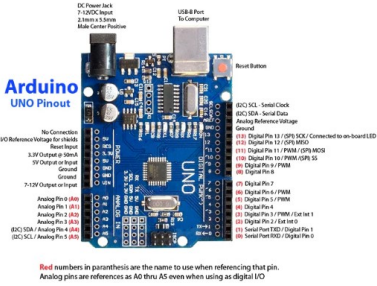



Componentele descrise în schematic au fost desenate manual. Am încercat să reproduc cât mai fidel modelele originale ale componentelor utilizate în proiect, pe baza datasheet-urilor.

Lista de Piese

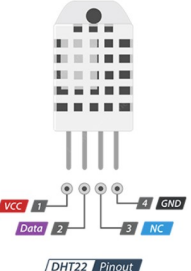
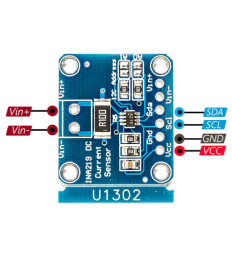

Nume Componenta	Preț	Site Cumparare
Arduino UNO R3 ATmega328P	30,76 RON	Arduino (SigmaNortec)
Module MOSFET IRF520 (2 buc)	15,68 RON	Mosfet (SigmaNortec)
Senzor Monitor Curent INA219	15,57 RON	INA219 (SigmaNortec)
Senzor Temperatura DHT22 AM302	33,12 RON	DHT22 (SigmaNortec)
Senzor Fotoelectric IR DS30P1	28,56 RON	IR (SigmaNortec)
Banda LED 12V, 120 smd	13,60 RON	LED (SigmaNortec)
Sursa de Alimentare + Prelungitor	36,30 RON	PSU (SigmaNortec)
Ventilator Spacer 12V 40mm	21,90 RON	Fan (eMag)
Breadboard	7,14 RON	Board (SigmaNortec)
Întreprupător ON-OFF	15,50 RON	Switch (Farnell)
Fire și rezistențe pentru conecția	25,00 RON	Wires (SigmaNortec)
Total	243,13 RON	X

Arduino UNO, Cablu USB, Sursă de Alimentare

		
<p>Arduino UNO R3 ATmega328P USB 2.0 de tip A-B</p>		<p>Power Source și Prelungitor 12V, 2A</p>

- **Timerele** sunt utilizate pentru a controla intervalele de citire a senzorilor și actualizările sistemului, pentru citirea datelor senzorului **DHT22** și **INA219**.
- **Înteruperile** asigură sincronizarea precisă și previn întreruperea secțiunilor critice care se ocupă cu primirea și trimiterea datelor în buffer pentru comunicarea cu **Node-RED**.
- Datele sunt colectate într-un buffer și trimise la un interval fix pentru a evita trimiterea continuă.
- **Cablul USB** conectează **μC** la laptop, facilitează alimentarea și transferul de date.
- Energie de **5V** către **Arduino** și permite comunicarea pentru monitorizare și control.
- Leșire constantă de **12V** pentru o funcționare fiabilă.
- Furnizează până la **2A**, potrivit pentru alimentarea mai multor componente: bandă **LED**, senzorii **IR**, **INA219**, **DHT22** și coolerul.

Senzori: Temperatură/Umiditate, Curent/Tensiune și Fotoelectric

		
<p>Temperatura-Umiditate DHT22 AM302 Senzor</p>	<p>Curent-Tensiune INA219 Senzor</p>	<p>Senzor Fotoelectric IR DS30P1</p>

- Conectat la **pinul digital 12** de pe Arduino, cu energie de la pinii de **Vcc** și **GND**.
- Pentru măsurarea temperaturii și umidității, folosim **USART**.
- Senzorul **DHT22** măsoară temperaturi între **-40°C și 80°C** și umiditate între **0% și 100%**.
- Senzorul **INA219** utilizează **I2C** pentru comunicare.
- Protocolul **I2C** permite transferul date pentru curent-tensiune prin 2 pini (**SCL** și **SDA**).
- Măsoară curent între **-3.2A și 3.2A** cu o rezoluție de **10μA**.
- Măsoară tensiunea de până la **26V** și curentul cu aceeași rezoluție de **10μA**.
- Înmulțind citirile de tensiune și curent ⇒ calculăm **puterea** obținută din (**U·I**)
- Senzorul **InfraRosu** aduce capabilități de automatizare, cum ar fi controlul luminii bazat pe mișcare.
- Controlează luminozitatea benzii **LED** prin semnale PWM pentru un control precis.
- Detectează mișcare până la o distanță de **30 cm** ⇒ aprinde luminile.
- Senzorul emite un semnal **PNP**, care se integrează cu ușurință în circuitele de control logic.

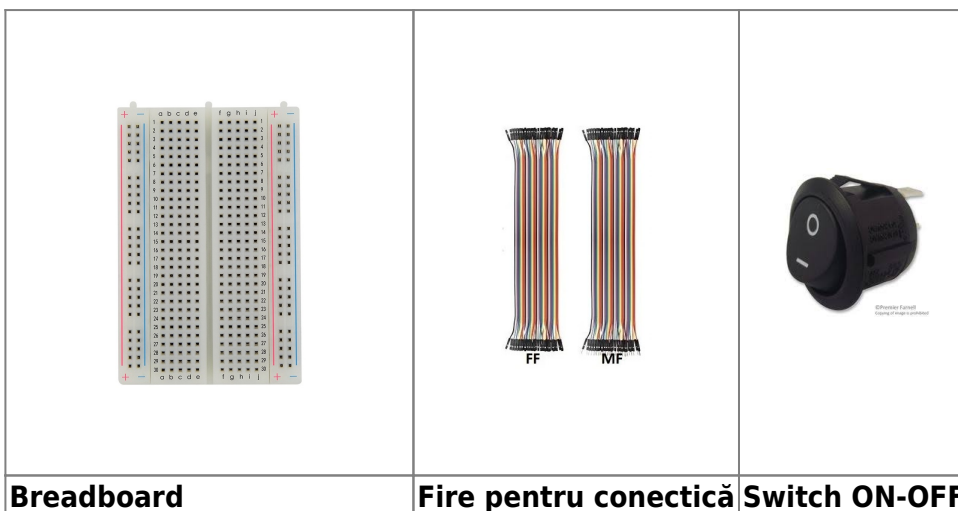
- Leșirea sensorului este conectată la pinul digital **8** de pe **Arduino**, tensiunea primită este de **12V**.

Control (Iluminat-Ventilator), Bandă LED și Ventilator



- **MOSFET**-ul controlează sarcini de curent ridicat, oferit de întregul ansamblu de senzori și actuatori.
- Comutarea rapidă și pierderi minime de putere. Ideale pentru controlul componentelor de mare putere.
- Suport pentru control *PWM*, care ajustează luminozitatea **LED-urilor** prin semnale *PWM*.
- Porțile (**G**) sunt conectate la pinul *PWM 9* pentru **LED-uri** și pinul digital **10** pentru ventilator.
- Drenajul (**D**) este conectat la sarcină (bandă **LED** și ventilator).
- Sursă (**S**) este conectată la masă.
- **LED-urile** asigură iluminare dinamică pentru sistem, controlată de **MOSFET** pentru atenuare lumina.
- Consumă puțină energie, oferind în același timp luminozitate ridicată.
- Bandă **LED** este alimentată de o sursă de **12V** și controlată de **MOSFET**, care este conectat la **μC**.
- Activat pe baza citirilor de temperatură pentru a răci eficient componentele.
- **Ventilatorul** se pornește sau oprește ventilatorul în funcție de temperatură.
- **Ventilatorul** este alimentat de o sursă de **12V** și controlat prin pinul digital **10**.

Comutator ON-OFF, Breadboard și Fire pentru Conectică



- **Breadboard-ul** și firele de conectare (**M-F, F-F**) sunt folosite pentru prototiparea și testarea

circuitului.

- **Breadboard-ul** permite testare, asamblare și flexibilitate pentru viitoare reconfigurări.
- Singurele lipituri necesare au fost pentru comutatorul **ON-OFF** și conectarea a două benzi **LED** în serie.
- **Înterupător ON-OFF** permite controlul manual al sistemului de iluminare.
- Pornește sau oprește sistemul, ușor accesibil pentru o operare rapidă.

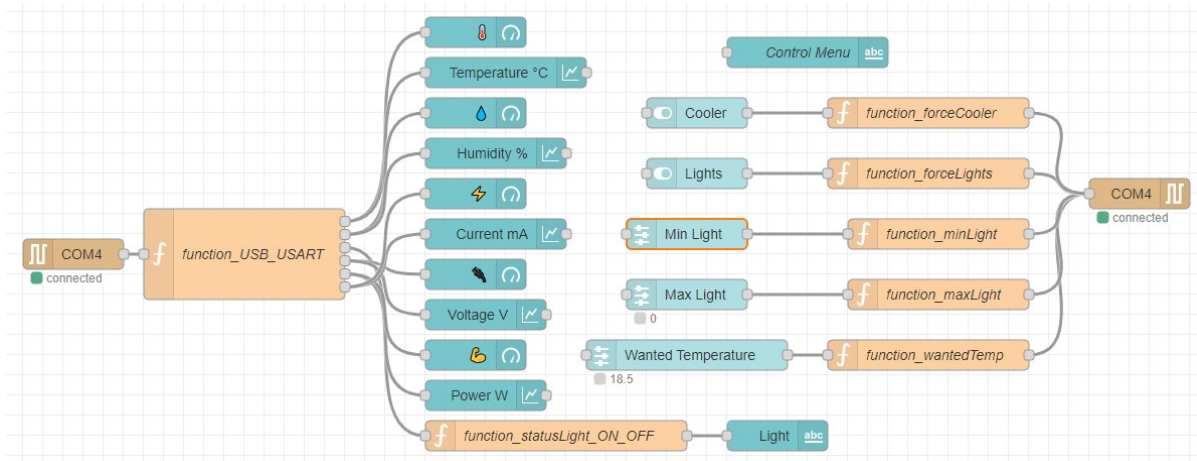
Software Design

- **Mediu de dezvoltare:** PlatformIO.
- **Trebuie să avem Node.js înainte de Node-RED!**
- **NPM packages for Node-RED:** node-red, node-red-dashboard, node-red-node-serialport.
- **Librării și surse 3rd-party:** DHT, Adafruit INA219, Adafruit Unified Sensor, Adafruit BusIO.
- **Algoritmi și structuri:** Implementarea algoritmilor pentru achiziția și procesarea datelor de la senzori, comunicarea cu interfața grafică și interacțiunea cu utilizatorul.
- **Functii implementate:** Funcții pentru inițializarea și configurarea senzorilor, citirea datelor de la senzori, gestionarea comunicațiilor prin USB și implementarea interfeței de utilizator grafice pentru ajustarea parametrilor.

Firmware-ul este organizat pentru a gestiona mai multe funcționalități, cum ar fi achiziția de date de la senzori, controlul **PWM** și comunicarea prin **USART**, folosind **GPIO**, **Timere** și **I2C**.

- **Butonul sau semnalul IR:** sunt folosite pentru controlul ventilatorului. Firmware-ul gestionează debouncing-ul butonului și decodarea semnalelor IR. După detectare, microcontrolorul schimbă starea pinului GPIO asociat ventilatorului pentru a-l activa sau dezactiva. Acțiunile sunt monitorizate în bucla principală a firmware-ului pentru funcționare corectă.
- **Timere și PWM:** Timerul1 generează intervale precise de timp pentru crearea întârzierilor și programarea sarcinilor. PWM-ul controlează luminozitatea LED-urilor, permițând tranziții fluide ale intensității luminii.
- **Comunicare I2C și UART:** I2C-ul facilitează comunicarea între microcontrolor și senzorul INA219 pentru măsurarea curentului, tensiunii și puterii electrice. UART-ul este folosit pentru comunicare serială, în special cu Node-RED.
- **GPIO:** Acestea sunt configurate pentru diverse funcții, inclusiv intrări digitale pentru senzori și ieșiri digitale pentru controlul dispozitivelor. Un pin GPIO controlează ventilatorul, reacționând la butonul de pornire/oprire sau semnalul IR.
- **Integrare a senzorilor:** Se citesc date periodice de la senzori precum DHT22 și INA219, și se transmit către Node-RED.
- **Bufere și operații atomice:** Se utilizează un buffer circular FIFO pentru stocarea datelor USART, asigurând o prelucrare sigură a datelor în medii cu întreruperi.

Rezultate Obținute



1. Comunicarea Serială

- Port COM4: Conectează senzori și actuatori prin USB.

2. Senzori

- Măsoară temperatura (°C),
- Umiditatea (%),
- Curentul (mA),
- Tensiunea (V) și
- Puterea (W).

3. Control Dispozitive

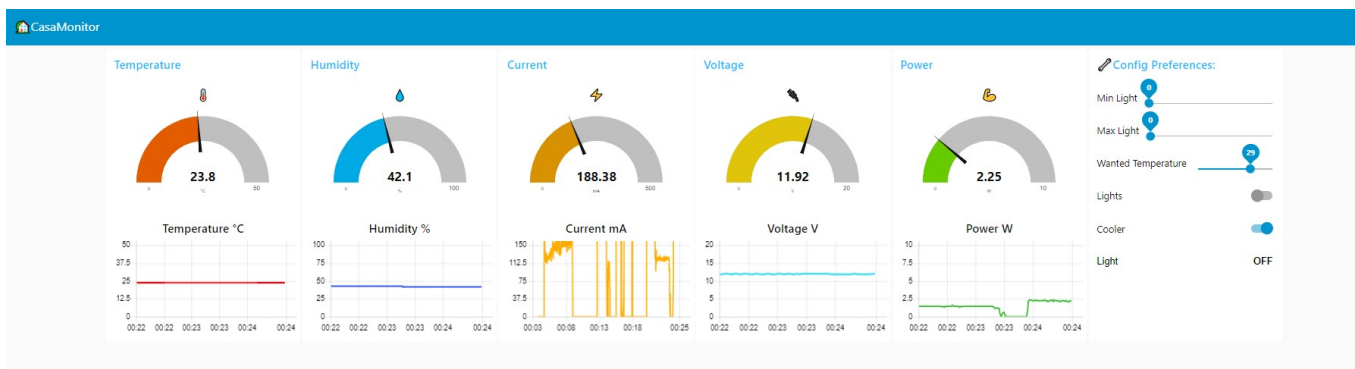
- Cooler: Activează ventilatorul.
- Lights: Controlează luminile.
- Min/Max Light: Setează intensitatea luminii.
- Wanted Temperature: Setează temperatura dorită.

4. Funcții de Procesare

- Controlează activarea ventilatorului și luminilor pe baza parametrilor setați și mășurați.

5. Interfață de Control

- Permite setarea și monitorizarea parametrilor de mediu și a stării dispozitivelor.



1. Slidere pentru Lumină

Max Light:

- Folosit pentru lumina IR sau butonul ON-OFF.
- Modifică intensitatea luminii și valoarea butonului ON-OFF.
- Utilizat dacă există mișcare detectată sau butonul ON este activ.
- Dacă niciuna dintre opțiuni nu este activă, se folosește slider-ul Min Light.
- Interval între minim și maxim; dacă min > max, sistemul funcționează normal cu valorile anterioare.

Min Light:

- Setează intensitatea minimă a luminii când nu se folosește IR sau butonul ON-OFF.
- Activează intensitatea minimă a luminii când celelalte opțiuni nu sunt active.

2. Butonul Cooler

- Activează ventilatorul pentru răcire suplimentară.

3. Interfața și Vizualizarea Datelor

- Temperature, Humidity, Current, Voltage, și Power:
- Afișează valori în timp real pentru temperatură, umiditate, curent, tensiune și putere.
- Graficele lineare arată variația în timp și monitorizează consumurile și valorile medii.

4. Interfața de Control și Configurare

- Setează preferințele de configurare, inclusiv intensitatea minimă și maximă a luminii și temperatura dorită.
- Butoane pentru activarea luminii și a ventilatorului.

5. Slider Temperatură Dorită

- Permite utilizatorului să seteze temperatura dorită.
- Compara temperatura setată cu temperatura reală.
- Temperatura reală este mai mare decât cea dorită, ventilatorul pornește automat pentru a răci aerul.

Download

Videoclipuri demonstrative, SW-HW, pentru a observa cum funcționează GUI (Node-RED) cu HW, vizualizați cele 2 videoclipuri în același timp, începeți videoclipul **pentru SW de la 0 și cel pentru HW de la secunda 25** de la începutul înregistrării.

- **Node-RED API Dashboard:** [SW-Interact](#)
- **Hardware Interfață:** [HW-Interact](#)
- Arhiva cu fișierele proiectului, inclusiv sursele, schemele de circuit și documentația, poate fi găsită pe [GitHub](<https://github.com/justin-marian/casamonitor>).

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/fstancu/justin.popescu1605>



Last update: **2024/05/27 07:34**