

**Student: Mădălina-Maria Mitricioaei**

**Email:** [madalina.mitricioaei@outlook.com](mailto:madalina.mitricioaei@outlook.com), [madalina.mitricioaei@gmail.com](mailto:madalina.mitricioaei@gmail.com)



## Titlul proiectului: IOT Sensor Hub

### Rezumat

Proiectul de față are ca scop dezvoltarea unei sistem embedded care preia date de la senzori și le trimite, prin ethernet la un server PC. De aici datele sunt stocate în Cloud, folosind Docker, iar pe baza acestora se construiesc grafice care arată evoluția în timp a diferiților parametri (temperatura, umiditate, prezența/absența obiectelor etc.).

### Introducere

Industria automotive s-a dezvoltat foarte mult în ultimii ani și, în acest sens, a crescut mult numărul de dispozitive (ECU-uri, senzori, actuatori) care comunică într-o mașină. Poate între două hub-uri putem folosi CAN, LIN sau FlexRay, dar dacă dorim comunicație între mai multe astfel de hub-uri, avem nevoie de ceva ieftin, fiabil și simplu de folosit, cum este Ethernet-ul.

La momentul actual avem într-o mașină o rețea de tipul celei din imaginea de mai jos:

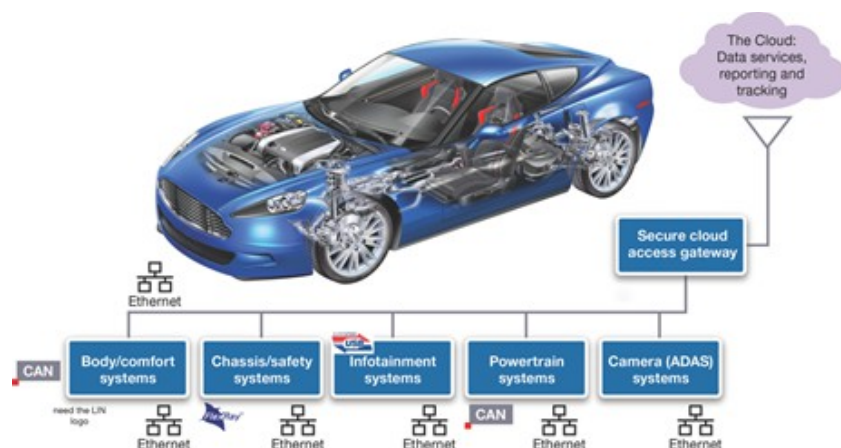


Figure 1. Ethernet in automotive [1]

Observăm în imagine că avem mai multe sisteme independente, cum sunt Body and

Comfort, Chassis and Safety, Infotainment, Powertrain, ADAS. Acestea sunt conectate prin ethernet la un BACKBONE central, care trimite datele securizat la Cloud.

### Componente hardware utilizate:

- Kit-ul de dezvoltare XMC4700 Relax Kit de la Infineon
- Senzor de înclinare SW-520D
- Senzor de detecție a apei
- Senzor de temperatură LM335Z
- Senzor optic cu reflexie LTH-209-01

### Componente software:

- Dave IDE
- Python
- Sockets library
- PlotLib pentru realizare grafice
- Docker pentru stocare CLOUD

## Descriere componente și detalii de implementare

**Senzorul de înclinare SW-520D**, digital, format în interior din două componente metalice. Când acestea intră în contact, se produce o tensiune la terminalele senzorului, permițând astfel trecerea curentului electric prin el.



Figure 2 Senzor de înclinare

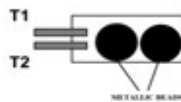


Figure 3 Senzor de înclinare interior

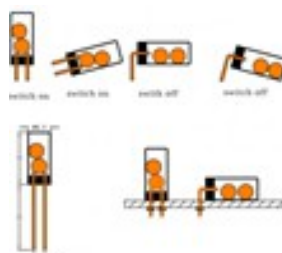


Figure 4 Senzor de înclinare – functionare

**Senzor de detecție a apei Arduino**, digital, având în componență mai multe trasee metalice. Acestea sunt conectate printr-o rezistență de PULL-UP la Vcc și determină un scurt-circuit când intră în contact cu apa. Printre proprietățile senzorului se numără consumul redus (mai mic de 20mA) și costul scăzut (1 \$).



Figure 5 Senzor de detecție a apei

**Senzorul de temperatură LM335Z**, analogic, cu o rezoluție pe 12 biți. Acesta funcționează ca o diodă Zener. Nu necesită calibrare. La ieșire avem 10mV/ grad Kelvin. Aici tensiunea de străpungere a diodei este egală cu temperatura.



Figure 6 Senzor de temperature

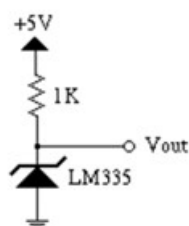


Figure 7 Senzor de temperatură – schematic [2]

**Senzor optic cu reflexie LTH-209-01**, format dintr-un LED și o fotodiodă. Practic mecanismul e simplu, LED emițând un fascicul, iar dacă acesta întâlnește o suprafață reflectă fasciculul luminos, care ajunge înapoi la fotodiodă, care îl recepționează.



Figure 8 Senzor optic

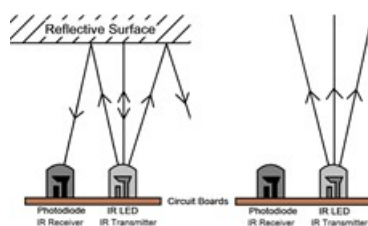


Figure 9 Senzor optic - principiu de funcționare

Schema bloc a sistemului este ilustrată în figura de mai jos:

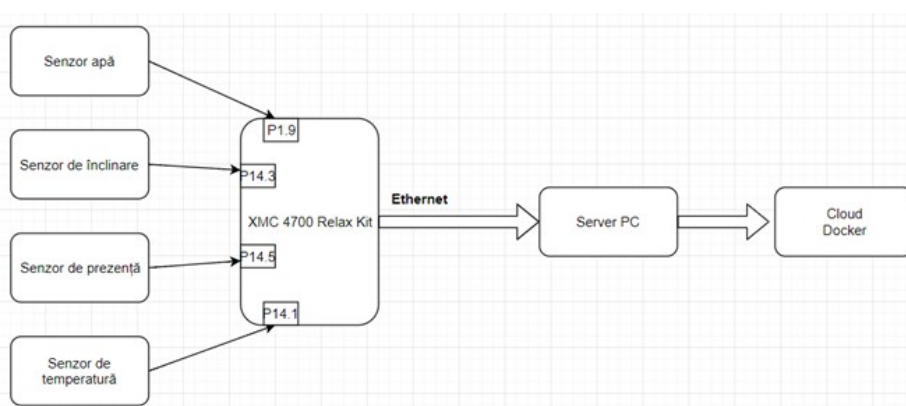


Figure 10 Schema bloc system

Observăm în figura 10, cei patru senzori sunt conectați la pinii de pe placa de dezvoltare XMC, care procesează informațiile trimise de aceștia, le trimite mai departe prin Ethernet la PC, iar de aici acestea sunt trimise pe Cloud.

Pentru dezvoltarea programului software s-a folosit mediul de programare Dave, iar pentru partea de server s-a folosit Python și librăria sockets. Pentru grafice s-a utilizat librăria Plotly.

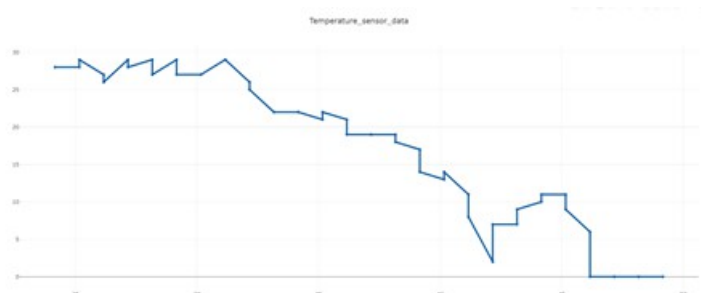


Figure 11 Grafic Temperatura

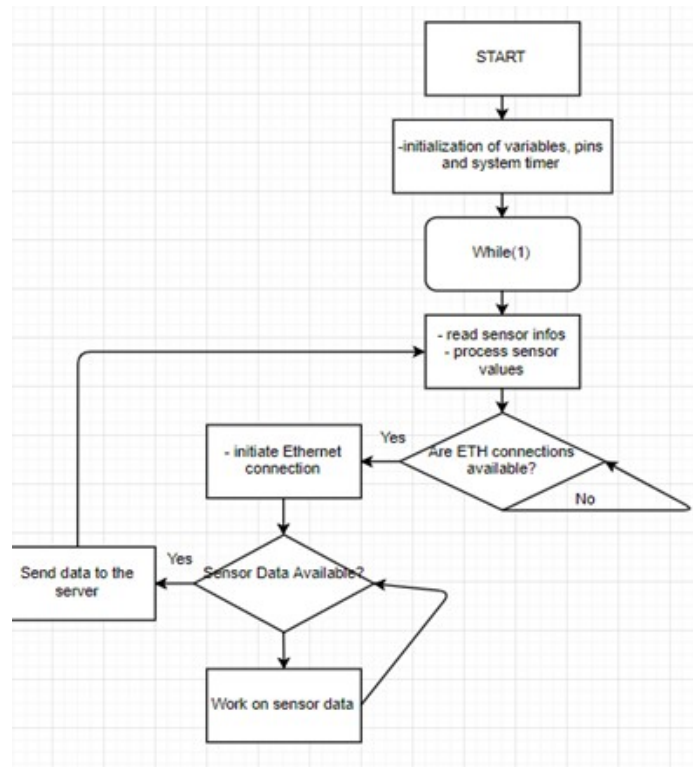


Figure 12 Diagramă Software

### Configurare Ethernet în Dave:

General Settings	Network Interface	IP Settings	Protocol Settings	Memory Settings
<input type="checkbox"/> Use DHCP				
IP address:		<input type="text" value="192.168.0.10"/>		
Subnet mask:		<input type="text" value="255.255.255.0"/>		
Gateway address:		<input type="text" value="192.168.0.10"/>		
<input type="checkbox"/> Enable IPv6 support				
<input type="checkbox"/> Enable IP options				
<input type="checkbox"/> Enable IP fragmentation				
<input type="checkbox"/> Enable IP reassembly				
Max. transmission unit (MTU):		<input type="text" value="1500"/>		
Default time to live (TTL):		<input type="text" value="255"/>		

Figure 13 Ethernet Configuration in **Dave**

În figura 13 este prezentată configurația Ethernet care trebuie setată în Dave Studio. Avem nevoie de adresa IP la care vrem să trimitem datele, masca de rețea și gateway-ul (adică calea) pe care vrem să trimitem datele.

Default time to live reprezintă perioada de timp în care mesajul rămâne disponibil în rețea (după care acesta se șterge, fără a reîncerca retransmiterea lui).

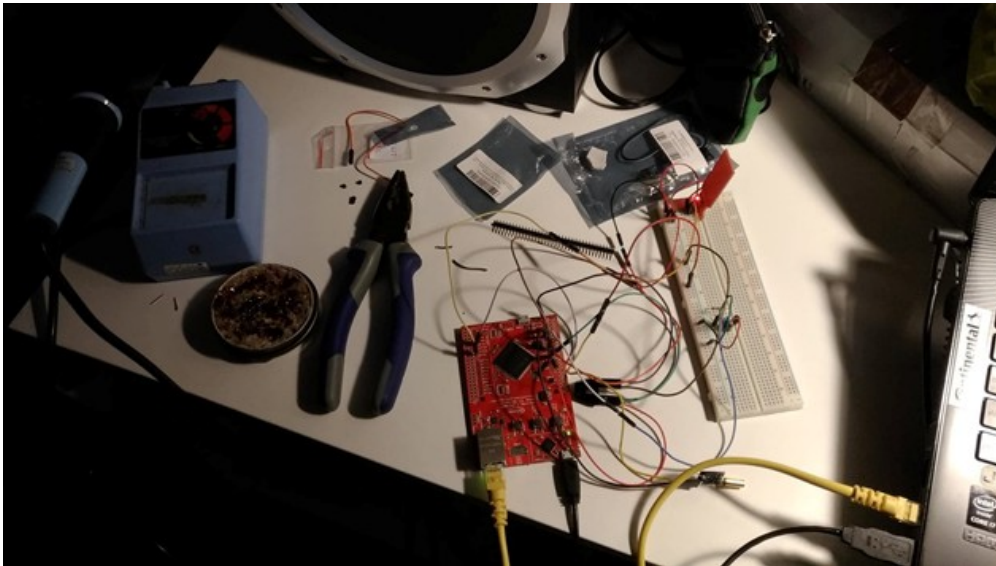


Figure 13 Realizare proiect

#### **Bibliografie:**

[1] Pagină web: <http://www.techdesignforums.com/practice/technique/using-ethernet-automotive-networks/>

[2] LM335Z Temperature Sensor Datasheet <http://www.ti.com/lit/ds/symlink/lm235.pdf>