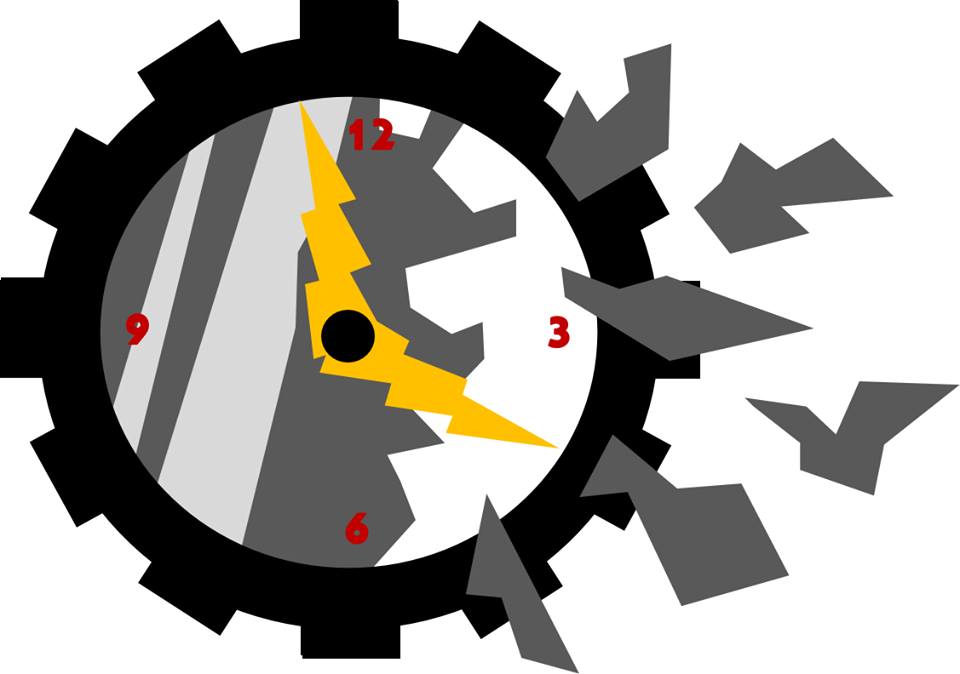
**Interfete om-masina**

**Energy monitoring system**



Cuprins

[Energy monitoring system 1](#_Toc469345993)

[1. Scopul documentului 4](#_Toc469345994)

[2. Continutul documentului 4](#_Toc469345995)

[3. Descriere generala a produsului 4](#_Toc469345996)

[a) Descrierea produsului 4](#_Toc469345997)

[b) Situatia curenta de pe piata 4](#_Toc469345998)

[c) Beneficii 5](#_Toc469345999)

[4. Diagrama de pachete de activtati (WBS) 6](#_Toc469346000)

[5. Arhitectura sistemului 7](#_Toc469346001)

[6. Diagramele UML 8](#_Toc469346002)

7. Principalele secvente de cod **.................................................................................................**9

8. Wireframe (interfata grafica)  **...............................................................................................**11

# Scopul documentului

In aceasta documentatie vom descrie functiile pe care aplicatia "**Energy monitoring system**" le pune la dispozitia utilizatorilor. Scopul acestui document este de a ajuta utilizatorii sa inteleaga mai bine functionalitatea produsului si modul in care il pot ei folosi. Cu ajutorul acestei documentatii, orice utilizator va putea folosi cu usurinta produsul, respectiv aplicatia.

# Continutul documentului

Acest document contine toate informatiile legate de produs si aplicatie, incepand

de la descrierea generala, scopul si obiectivul produsului, pana la descrierea modului de functionare si modul de proiectare al aplicatie. Incercand astfel familiarizarea utilizatorilor cu produsul final. Am incercat sa cuprindem intr-un mod cat mai folositor si usor de inteles, toate detaliile, de la componentele produsului pana la accesarea interfetei WEB.

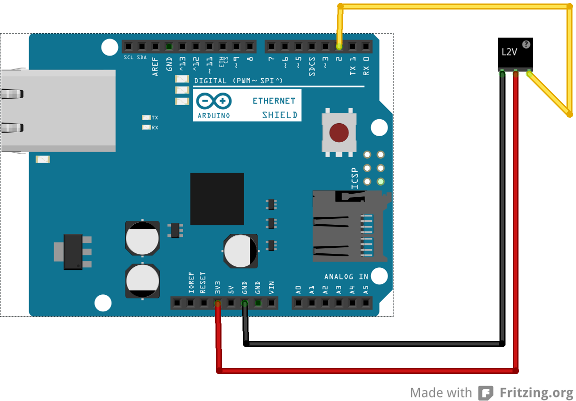
# Descriere generala a produsului

## Descrierea produsului

Produsul este format dintr-o placuta Intel Galileo, un multimetru si un senzor de masurare a curentului. Acesta are ca obiectiv masurarea consumului de curent si afisarea acestuia in cadrul unei interfete web. In aceasta interfata se afiseaza valorile citite de catre senzor, valori care se stocheaza intr-o baza de date. Scopul produsul este de a monitoriza consumul de energie electrica dintr-o incapare si de a aduce masuri de economisire a acestui consum cu ajutorul unor grafice, mai pe scurt, solutii de optimizare a consumului de energie electrica.

## Situatia curenta de pe piata

In momentul de fata pe piata nu exista multe produse pentru masurarea si afisarea energiei electrice, respectiv monitorizarea ei. Noi am folosit acest senzor deoarece este mult mai practic, nu am reusit sa-l integram intr-un dispozitiv pentru a fi cat mai practic si usor de folosit. In momentul de fata, ne gandim cum am putea multiplica acest produs si a monitoriza intreaga incapere fara foarte multe costuri. Asadar produsul nostru se incadreaza la mijloc, avand o precizie buna fara sa fie scump.



## Beneficii

Produsul ofera o modalitate de monitorizare a energiei electrice pe o perioada de timp, stabilita de catre utilizator. Este foarte usor de citit si de utilizat. Scopul acestui produs este de a economisi in timp energie electrica prin monitorizarea ei. De exemplu: Daca lasam un incarcator in priza fara sa il folosim, il monitorizam o zi, vom ajunge la concluzia ca el consuma chiar daca nu este folosit.

Un alt scop al acestui produs este indepartarea folosirii inutile a energiei electrice, in scopul protejarii planetei, dar si a „buzunarului”.

# Diagrama *de pachete de activtati (WBS)*

**Diagrama de pachete de activitati, e necesara tuturor proiectelor, este orientata spre rezultate, incluzind toate rezultatele proiectului, ea face, pana la un anumit nivel stabilit in proiect, rafinarea pe elemente de activitate necesare finalizarii proiectului, nu trebuie confundata cu o diagrama cu bare sau cu listele de proiect.**



# ****Arhitectura sistemului****

**Arhitectura unui sistem este o reprezentarea grafica a intregului sistem. Acesta ne ajuta sa distingem componentele sistemului si sa intelegem modul de functionare al acestuia.**

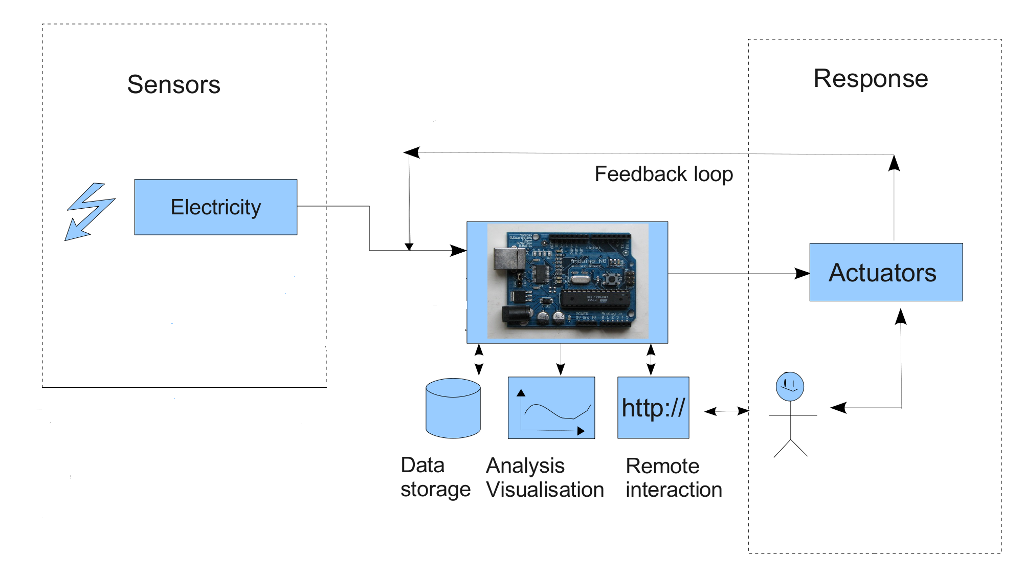


Figure 2: Arhitectura Sistemului

# Diagramele UML

Unified Modeling Language (prescurtat UML) este un limbaj standard pentru descrierea de modele și specificații pentru software. Aceste diagrame ne ajuta sa intelegem modul de functionare a partii software din intregul sistem.

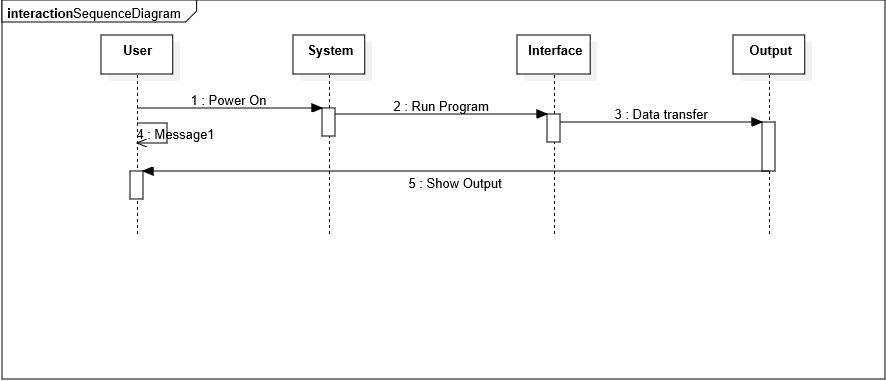


Figure 3: Diagrama secventiala

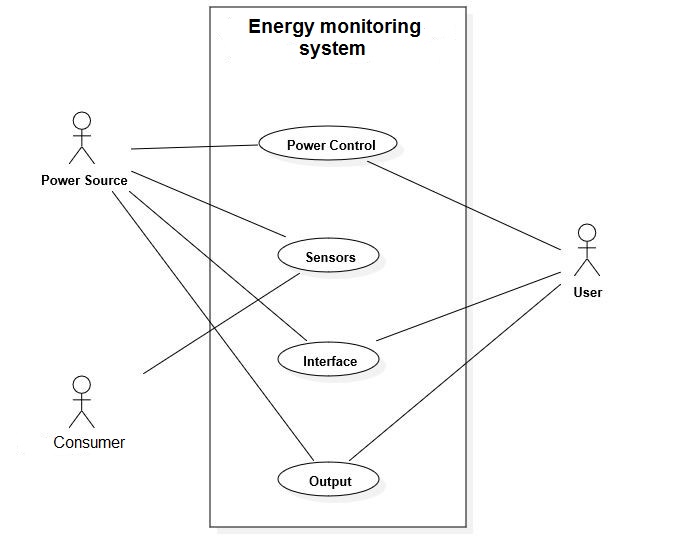


Figure 4: Diagrama use-case

1. **Principalele secvente de cod**

# COD ARDUINO

# #include <SPI.h>

# #include <Ethernet.h>

# byte mac[] = {

# 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

# 

# // Enter the IP address for Arduino, as mentioned we will use 192.168.0.16

# // Be careful to use , insetead of . when you enter the address here

# IPAddress ip(192,168,0,17); // asta e adresa asignata

# float sensorIn = A5; // Analog input pin on Arduino we connected the SIG pin from sensor

# float photocellReading; // Here we will place our reading

# double mVperAmp = 66; // use 100 for 20A Module and 185 for 5 Module

# double Voltage;

# double Voltage2;

# double Power;

# double VRMS;

# double AmpsRMS;

# char server[] = "192.168.0.102"; // IMPORTANT: If you are using XAMPP you will have to find out the IP address of your computer and put it here (it is explained in previous article). If you have a web page, enter its address (ie. "www.yourwebpage.com")

# // Initialize the Ethernet server library

# EthernetClient client;

# void setup() {

# 

# // Serial.begin starts the serial connection between computer and Arduino

# Serial.begin(9600);

# 

# // start the Ethernet connection

# Ethernet.begin(mac, ip);

# 

# }

# void loop() {

# Voltage = getVPP();

# Voltage2 = Voltage \* 80; // asta numai ca sa printez si voltajul pe serial monitor

# VRMS = (Voltage/1.0) \*0.707; //era initial /2.0

# AmpsRMS = (VRMS \* 1000)/mVperAmp;

# // Fill the sensorReading with the information from sensor

# 

# // Connect to the server (your computer or web page)

# if (client.connect(server, 80)) {

# client.print("GET /write\_data.php?"); // This

# client.print("value="); // This

# client.print(AmpsRMS); //We are making a GET request just like we would from our browser but now with live data from the sensor

# //client.print("GET /write\_data.php?");

# client.print("&value2="); //am pus & pt a doua citire

# client.print(Voltage2);

# Serial.print(AmpsRMS);

# Serial.print(Voltage2);

# 

# client.println(" HTTP/1.1"); // Part of the GET request

# client.println("Host: 192.168.0.102"); // IMPORTANT: find out the IP address of your computer and put it here

# client.println("Connection: close"); // Part of the GET request telling the server that we are over transmitting the message

# client.println(); // Empty line

# client.println(); // Empty line

# client.stop(); // Closing connection to server

# }

# else {

# // If Arduino can't connect to the server

# Serial.println("--> connection failed\n");

# }

# 

# // Give the server some time to receive the data and store it. I used 10 seconds here. Be advised when delaying. If u use a short delay, the server might not capture data because of Arduino transmitting new data too soon.

# delay(4000);

# }

# //functia de citire voltaj

# float getVPP()

# {

# float result;

# int readValue; //value read from the sensor

# int maxValue = 0; // store max value here

# int minValue = 1023; // store min value here

# uint32\_t start\_time = millis();

# while((millis()-start\_time) < 1000) //sample for 1 Sec

# {

# readValue = analogRead(sensorIn);

# // see if you have a new maxValue

# if (readValue > maxValue)

# {

# /\*record the maximum sensor value\*/

# maxValue = readValue;

# }

# if (readValue < minValue)

# {

# /\*record the maximum sensor value\*/

# minValue = readValue;

# }

# }

# // Subtract min from max

# result = ((maxValue - minValue) \* 4.50)/1023.0; // era imultit cu 5.0

# return result;

# }

COD PHP WRITE DATA

<?php

$servername = "localhost";

$username = "root";

$password = "password";

// Create connection

$conn = new mysqli($servername, $username, $password);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully";

$sql = "INSERT INTO testare.sensor (value, value2) VALUES ('".$\_GET["value"]."', '".$\_GET["value2"]."')";

if ($conn->query($sql) === TRUE) {

echo "New record created successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

?>

COD PHP GET DATA

<?php

$url=$\_SERVER['REQUEST\_URI'];

header("Refresh: 5; URL=$url"); // Refresh the webpage every 5 seconds

?>

<html>

<head>

<title>Current Sensor</title>

</head>

<body>

<h1>Current sensor readings</h1>

<table border="2" cellspacing="3" cellpadding="4">

<tr>

<td>ID</td>

<td>TIME</td>

<td>Curent</td>

<td>Tensiune</td>

</tr>

<?php

// Connect to database

// IMPORTANT: If you are using XAMPP you will have to enter your computer IP address here, if you are using webpage enter webpage address (ie. "www.yourwebpage.com")

$con=mysqli\_connect("localhost","root","password");

// Retrieve all records and display them

$result = mysqli\_query($con,'SELECT id, time, value, value2 FROM testare.sensor ORDER BY id DESC');

// Process every record

while($row = mysqli\_fetch\_array($result))

{

echo "<tr>";

echo "<td>" . $row['id'] . "</td>";

echo "<td>" . $row['time'] . "</td>";

echo "<td>" . $row['value'] . "</td>";

echo "<td>" . $row['value2'] . "</td>";

echo "</tr>";

}

// Close the connection

mysqli\_close($con);

?>

</table>

</body>

</html>

# 8. Wireframe (interfata grafica)

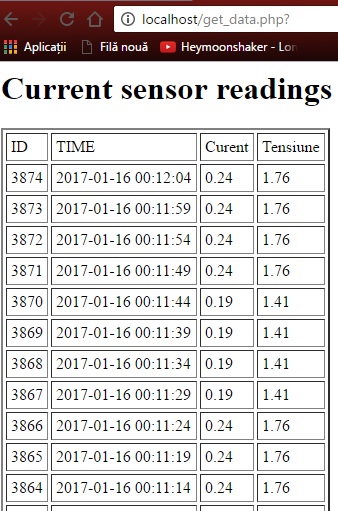


Figure 5: Interfata grafica

Figure 6: Sistemul implementat

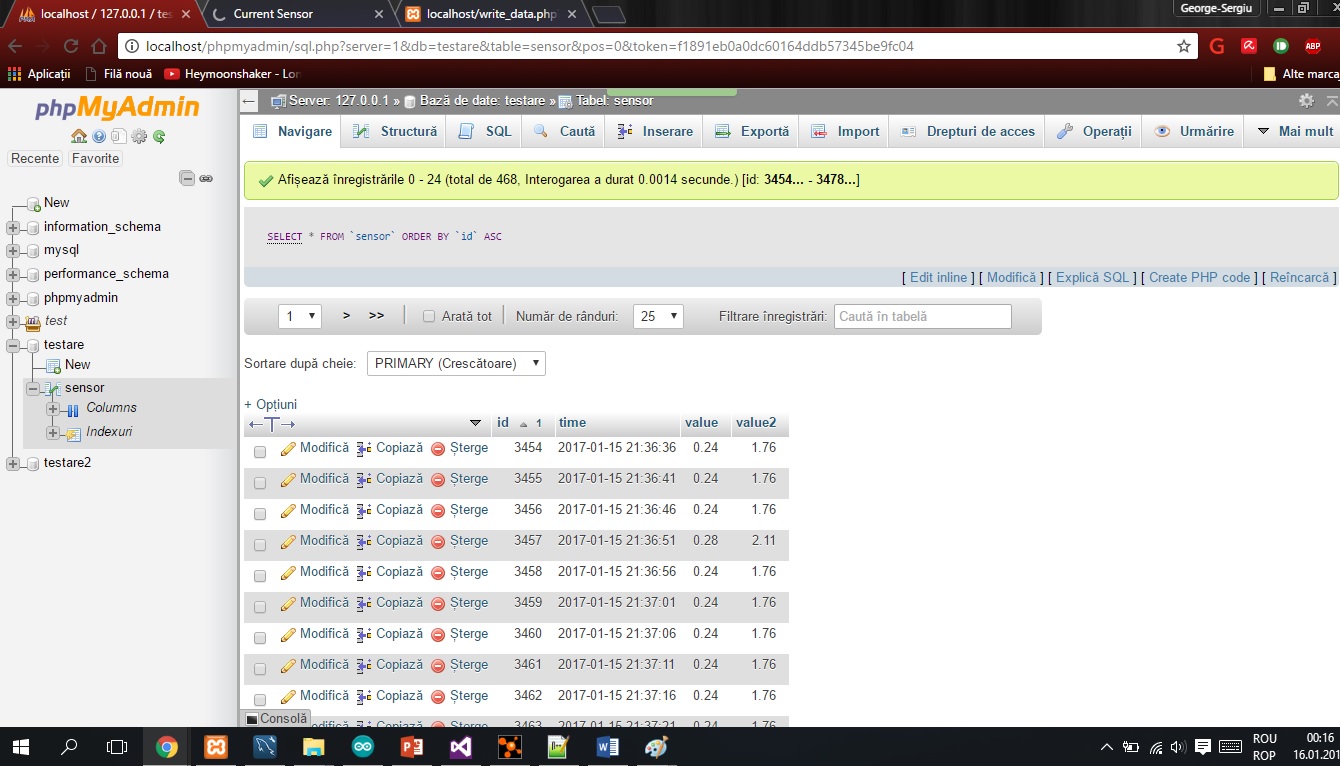


Figure 7: Baza de date

Bibliografie

1. <http://www.w3schools.com/php/php_mysql_insert.asp>

2. <http://www.instructables.com/id/Save-data-of-temperature-and-humidity-on-MySQL-wit/?ALLSTEPS>

3. <http://www.tweaking4all.com/hardware/arduino/arduino-ethernet-data-push/>

4. <http://www.icreateproject.info/uncategorized/arduino-save-data-to-database/>

5. <http://www.icreateproject.info/uncategorized/arduino-display-data-over-local-network/>