

Nume student: Purice Rebeca Ligia

Gmail: rebeca.ligiap@gmail.com



Denumire/titlu subiect: Detectarea precipitațiilor, umidității și temperaturii

Rezumat:

În zilele noastre se dorește ca autovehiculele să aibă autonomie totală (să nu depindă deloc de factorul uman). Acest lucru presupune ca acestea să aibă aceleași “simțuri” ca oamenii.

Imitarea acestor simțuri se poate realiza cu ajutorul senzorilor. Proiectul de față își propune ca prin intermediul acestora să creeze posibilitatea ca un autovehicul să reacționeze la diverși factori meteorologici precum: ploaie, temperatură și umiditate.

Utilitate: Cu ce scop am dori ca autovehiculele să aibă aceste simțuri? În cazul unor factori meteorologici nefavorabili am dori ca viteza de deplasare să se reducă sau chiar autovehiculul să facă un popas până când vremea se îmbunătățește. În același timp știm că o problemă a șoferilor este căldura insuportabilă în timpul verii și frigul în timpul iernii din interiorul mașinii.

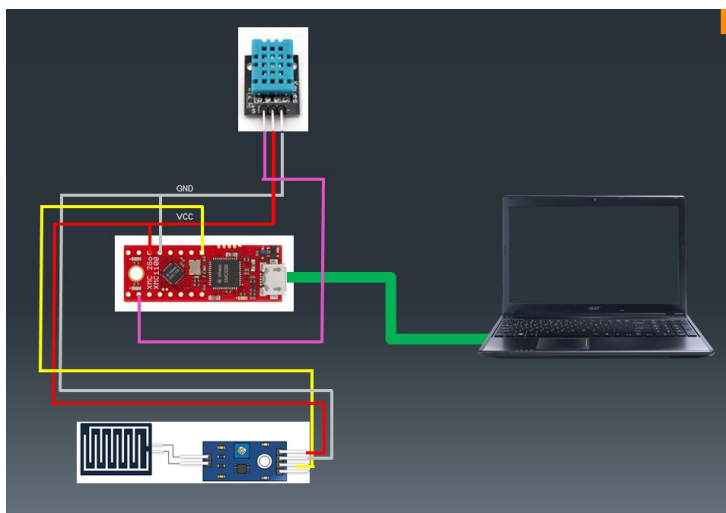
Resurse:

❖ **Hardware:**

- Placa de dezvoltare XMC2GO
- Senzor de umiditate și temperatură
- Senzor de ploaie
- Breadboard
- Fire

❖ **Software:**

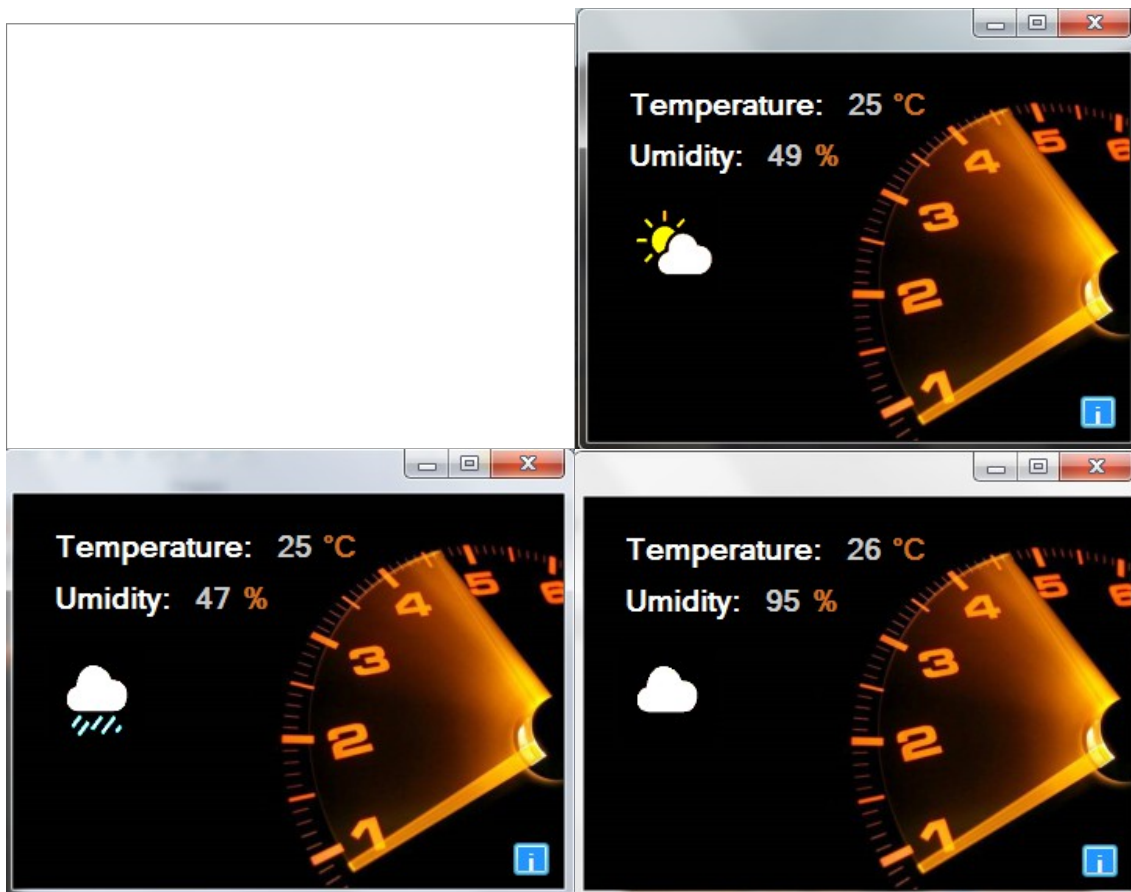
- Arduino IDE
- Visual Studio



Descriere soluție:

Legătura între componente este ilustrată în imaginea de mai sus. Informații au fost preluate de la senzori folosind XMC2GO. Pentru fiecare senzor s-a folosit câte un pin analogic (pentru senzorul de ploaie s-a folosit pinul A0, respectiv pinul A1 pentru senzorul de umiditate și temperatură). Senzorii au fost alimentați cu 3.3V de la pinul VCC de la plăcuța XMC2GO și conectați la pinul de masă al plăcuței. Informațiile preluate de la senzori sunt transmise mai departe serial către calculator. Pentru acest lucru s-a folosit un cablu usb. Pe calculator informațiile sunt citite de către o aplicație realizată în C# care, după ce prelucrează informațiile le afișează fie în format numeric, fie sub forma unor imagini sugestive.

Secvența demonstrativă:



Cod aplicație Arduino:

```
#include <dht.h>
#define dht_dpın A0 //seteaza pinul pentru senzorul de temperatura si umiditate
dht DHT;

//Declararea variabilelor folosite pentru senzorul de ploaie
int val_an_ploaie = A1; //valoarea pe care o preia placuta de la senzorul de ploaie

void setup( )
{
  Serial.begin(9600);
}
void loop( )
{
  DHT.read11(dht_dpın);
  if(analogRead(val_an_ploaie)<650){
    Serial.print(10000+((int)DHT.humidity)*100+(int)DHT.temperature);
  }
  else{
    Serial.print(20000+((int)DHT.humidity)*100+(int)DHT.temperature);
  }
  delay(1000);
}
```

Cod aplicație Visual Studio (C#):

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;
```

```
namespace Tema
```

```

{
public partial class Form1 : Form
{
private SerialPort port = new SerialPort("COM3", 9600, Parity.None, 8, StopBits.One);
string val = string.Empty; //Variabila in care se citeste valoarea transmisa serial

private delegate void SetTextDeleg(string text);

public Form1()
{
InitializeComponent();
try
{
port.DataReceived += new SerialDataReceivedEventHandler(port_DataReceived);
port.Open();
}
catch
{
MessageBox.Show("ERROR!!! Conexiunea nu a fost creata.");
}
}

private void port_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
try
{
Thread.Sleep(100);
// Show all the incoming data in the port's buffer
string x = port.ReadExisting();
this.BeginInvoke(new SetTextDeleg(val_sal), new object[] { x });
}
catch { }
}

private void val_sal(string data)
{
val = data.Trim();
int val_int = Convert.ToInt32(val);

//Extragerea valorii temperaturii
int temp = val_int % 100;
labelValTemp.Text = temp.ToString();
val_int = val_int / 100;

//Extragerea valorii umiditatii
int umidity = val_int % 100;
labelValUm.Text = umidity.ToString();

//Extragerea valorii ce stabilste daca afara ploua sau nu
int ploaie = val_int / 100;

if(ploaie==1)
labelImg.Image = imageListVreme.Images[3]; //Ploaie
else if(umidity<20)
labelImg.Image = imageListVreme.Images[0]; //Soare
else if(umidity >= 20 && umidity < 50)
labelImg.Image = imageListVreme.Images[1]; //Soare cu nori
else
labelImg.Image = imageListVreme.Images[2]; //Innourat
}

private void buttonInfo_Click(object sender, EventArgs e)
{
MessageBox.Show("Aplicatie creata de:\nPurice Rebeca\nCI - IA\n\n Avand in vedere faptul ca am ajuns intr-un punct al istoriei\nin care se
doreste ca un autovehicul sa aiba autonomie totala,\nproiectul de fata are ca scop imitarea anumitor stimuli externi.\nAstfel autovehiculul poate
reactiona la diversi factori.");
}
}
}

```

Bibliografie:

https://profs.info.uaic.ro/~arduino/index.php/Temperatura_%26_Umiditate
<http://roboromania.ro/produs/senzor-de-ploaie-raindrop-detectare-meteo-arduino/>

<https://github.com/Infineon/XMC-for-Arduino/issues/23>