Proiect ED 2023-2024

Șlincu Marius 325CB

Proiectul constă în realizarea unui sistem de monitorizare ambientală folosind o placă de dezvoltare compatibilă cu Arduino UNO. Sistemul utilizează mai mulți senzori pentru a colecta date despre luminozitate, presiune barometrică, temperatură și umiditate, iar rezultatele sunt afișate pe un ecran LCD cu interfață I2C.

Componente utilizate:

* Placă de dezvoltare compatibilă cu Arduino UNO (Atmega328p și CH340)
* Fotorezistor (tip 5528) împreună cu un rezistor de 10 kΩ
* Senzor de presiune barometrică HW-661 E/P 280 (BMP280)
* Senzor de temperatură și umiditate DHT22
* LCD 1602 cu interfață I2C și backlight galben-verde
* Breadboard HQ (400 puncte)

Am atasat poze cu proiectul la final, iar aici puteti gasi un **video** cu aceasta:

<https://youtu.be/w_BgVq_8As4>

**1. Identificarea microcontroller-ului existent pe placa de dezvoltare**

Placa de dezvoltare compatibilă cu **Arduino UNO** utilizează microcontroller-ul **ATmega328P**.

**2. Identificarea datasheet-ului complet al microcontroller-ului**

Poți găsi datasheet-ul complet al microcontroller-ului ATmega328P mai jos:

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\_Datasheet.pdf

**3. Identificarea parametrilor electrici ai microcontroller-ului**

**ATmega328P** are următorii parametri electrici principali:

* **Tensiunea de alimentare**: 1.8V - 5.5V
* **Frecvențe de lucru nominale**: până la 20 MHz (cu cristal extern)
* **Consum energetic**:
  + **IDLE mode**: ~0.75mA la 1MHz și 1.8V
  + **Active mode**: ~0.2mA/MHz la 1.8V, ~0.4mA/MHz la 3.3V, ~0.8mA/MHz la 5V

**4. Identificarea tensiunii la care este alimentat microcontroller-ul și a frecvenței semnalului de ceas**

Pe placa Arduino UNO, microcontroller-ul este alimentat la **5V**. Frecvența semnalului de ceas este de **16MHz**.

**5. Realizarea unei scheme electrice a sistemului**

Mai jos este descrierea conexiunilor:

* **Fotorezistorul (5528)** și rezistorul de 10 kΩ formează un divizor de tensiune conectat la unul dintre pinurile analogice (A0).
* **Senzorul de presiune barometric HW-661 E/P 280**: Conectat la pinii I2C (A4 - SDA, A5 - SCL).
* **Senzorul de temperatură și umiditate DHT22**: Conectat la unul dintre pinii digitali (D2).
* **LCD 1602 cu Interfață I2C**: Conectat la pinii I2C (A4 - SDA, A5 - SCL).

**6. Identificarea fișelor de catalog a senzorilor și a altor componentelor utilizate în realizarea proiectului**

* **Fotorezistor 5528**: GL5528 (https://datasheetspdf.com/datasheet/GL5528.html)
* **Senzor de presiune HW-661 E/P 280**: BMP280 (https://www.apogeeweb.net/circuitry/bmp280-barometric-pressure-sensor.html)
* **Senzor de temperatură și umiditate DHT22**: (https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf)
* **LCD 1602 cu Interfață I2C**: (https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/62-lcd-1602-cu-interfata-i2c-si-backlight-galben-verde.html?search\_query=lcd+1602&results=17)

**7. Extragerea parametrilor electrici ai senzorilor și compararea acestora cu cei ai microcontroller-ului**

* **Fotorezistor 5528**:
  + Tensiune de alimentare: 5V
  + Curent: foarte mic, depinde de lumină
* **Senzor de presiune BMP280**:
  + Tensiune de alimentare: 1.8V - 3.6V
  + Curent: 2.7µA (în modul standby), 700µA (în modul activ)
* **Senzor DHT22**:
  + Tensiune de alimentare: 3.3V - 6V
  + Curent: 0.3mA (în modul standby), 1.5mA (în modul activ)
* **LCD 1602 cu I2C**:
  + Tensiune de alimentare: 5V
  + Curent: ~20mA (fără backlight), ~80mA (cu backlight)

**8. Estimarea întregului consum maxim al sistemului**

Să estimăm consumul maxim în mA și mW al întregului sistem:

* **ATmega328P**: ~13mA la 16MHz și 5V
* **Fotorezistor și rezistor de 10 kΩ**: neglijabil (0.5 mA(estimare medie)
* **Senzor BMP280**: 0.7mA
* **Senzor DHT22**: 1.5mA
* **LCD 1602 cu I2C și backlight**: 80mA

Consum total: 13 + 0.7 + 1.5 + 80 = **95.2mA**

Ptotal​=5V × 95.2mA = **476mW**

**9. Numărul de acumulatori necesari și durata de funcționare**

Dacă folosești acumulatori de 1.2V și 2500mAh:

* Pentru a obține 5V, ai nevoie de: 5V / 1.2V = 4.17 (aproximativ 5 acumulatori inseriați)
* Timpul de funcționare: T = 2500mAh / 95.2mA ≈ **26.2 ore**

**10. Propunere de scădere a consumului energetic al sistemului**

* **Modificări hardware**:
  + Utilizarea unui regulator de tensiune LDO cu pierderi mai mici.
  + Înlocuirea LCD-ului cu unul cu consum mai mic sau fără backlight.
* **Modificări software**:
  + Implementarea funcției de sleep pentru microcontroller atunci când nu se realizează măsurători sau afișări.
  + Setarea senzorilor în modul standby atunci când nu sunt folosiți.

https://youtu.be/w\_BgVq\_8As4A group of electronic devices

Description automatically generatedA group of electronic devices

Description automatically generated

A screen shot of a computer program

Description automatically generatedA circuit board with wires and wires

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

 **Compatibilitatea tensiunilor de operare**:

* Atât senzorul de temperatură și umiditate DHT22 cât și Arduino (MCU – Atmega328p) sunt alimentate la 5V, ceea ce asigură că semnalele de ieșire și intrare sunt compatibile. Ieșirile MCU nu vor genera tensiuni care să afecteze senzorul și viceversa, prevenind astfel probleme de compatibilitate electrică.

 **Interfața I2C și tensiunile de operare**:

* Senzorul de presiune barometrică HW-661 E/P 280 (BMP280) operează la o tensiune de 3.3V, dar are o interfață I2C compatibilă cu Arduino care poate opera și la 5V cu ajutorul unor rezistențe de pull-up corespunzătoare. Aceasta permite comunicația sigură între senzor și microcontroller fără riscul de a deteriora componentele.

 **Backlight-ul LCD-ului și consumul energetic**:

* LCD-ul 1602 cu interfață I2C și backlight galben-verde este alimentat la 5V, similar cu tensiunea de operare a Arduino. Deși backlight-ul crește consumul energetic, acest aspect poate fi controlat prin utilizarea unui tranzistor pentru a comuta backlight-ul pornit/oprit în funcție de necesitățile de utilizare, contribuind astfel la economisirea energiei.

 **Compatibilitatea fotorezistorului cu Arduino**:

* Fotorezistorul 5528 este utilizat într-un circuit divizor de tensiune împreună cu un rezistor de 10 kΩ, formând un semnal de ieșire care variază între 0 și 5V. Acest semnal este compatibil cu pinii analogici ai Arduino, permițând citirea corectă a nivelului de lumină ambientală fără a risca deteriorarea microcontroller-ului.

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

#include <Adafruit\_BMP280.h>

#include <Wire.h>

#define I2C\_ADDR    0x3F

#define LCD\_COLUMNS 16

#define LCD\_LINES   2

#define DHTPIN 2

#define DHTTYPE DHT22

#define PHOTORESISTOR\_PIN A0

LiquidCrystal\_I2C lcd(I2C\_ADDR, LCD\_COLUMNS, LCD\_LINES);

DHT dht(DHTPIN, DHTTYPE);

Adafruit\_BMP280 bmp;

Adafruit\_Sensor \*bmp\_temp = bmp.getTemperatureSensor();

Adafruit\_Sensor \*bmp\_pressure = bmp.getPressureSensor();

void setup() {

  Serial.begin(9600);

  lcd.init();

  lcd.backlight();

  dht.begin();

  pinMode(PHOTORESISTOR\_PIN, INPUT);

  lcd.setCursor(0, 0);

  lcd.print("Marius' project!");

  unsigned status;

  status = bmp.begin(BMP280\_ADDRESS\_ALT, BMP280\_CHIPID);

  if (!status) {

    Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "

                      "try a different address!"));

    Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);

    Serial.print("        ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");

    Serial.print("   ID of 0x56-0x58 represents a BMP 280,\n");

    Serial.print("        ID of 0x60 represents a BME 280.\n");

    Serial.print("        ID of 0x61 represents a BME 680.\n");

    while (1) delay(10);

  }

  /\* Default settings from datasheet. \*/

  bmp.setSampling(Adafruit\_BMP280::MODE\_NORMAL,     /\* Operating Mode. \*/

                  Adafruit\_BMP280::SAMPLING\_X2,     /\* Temp. oversampling \*/

                  Adafruit\_BMP280::SAMPLING\_X16,    /\* Pressure oversampling \*/

                  Adafruit\_BMP280::FILTER\_X16,      /\* Filtering. \*/

                  Adafruit\_BMP280::STANDBY\_MS\_500); /\* Standby time. \*/

  bmp\_temp->printSensorDetails();

}

void loop() {

  // Wait a few seconds between measurements.

  delay(2500);

  // FOTOREZISTOR PT LUMINA

  Serial.println(analogRead(PHOTORESISTOR\_PIN));

 int ldrValue = analogRead(PHOTORESISTOR\_PIN);

  // Conversia valorii brute într-o valoare de tensiune (0 - 5V)

  float voltage = ldrValue \* (5.0 / 1023.0);

  // Conversia tensiunii în lux

  float lux = voltageToLux(voltage);

  // DHT22 PT TEMP&UMIDITATE

  // Reading temperature or humidity takes about 250 milliseconds!

  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

  float h = dht.readHumidity();

  // Read temperature as Celsius (the default)

  float t = dht.readTemperature();

  // BMP280 PT TEMP&PRESIUNE

  sensors\_event\_t temp\_event, pressure\_event;

  bmp\_temp->getEvent(&temp\_event);

  bmp\_pressure->getEvent(&pressure\_event);

  // Check if any reads failed and exit early (to try again).

  if (isnan(h) || isnan(t)) {

    Serial.println(F("Failed to read from DHT sensor!"));

    return;

  }

  //AFISARE LDC:

  //Tura 1

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("  Temp: ");

  lcd.print(t);

  lcd.print("C");

  lcd.setCursor(0, 1);

  lcd.print("  Hum: ");

  lcd.print(h);

  lcd.print("%");

  //Tura 2

  delay(3000);

  lcd.clear();

  lcd.setCursor(3, 0);

  lcd.print("Pressure:");

  lcd.setCursor(2, 1);

  lcd.print(pressure\_event.pressure);

  lcd.print("hPa");

  //Tura 3

  delay(2500);

  lcd.clear();

  lcd.setCursor(1, 0);

  lcd.print("Outside: ");

  if (lux > 50) {

    lcd.print("Light!");

  } else {

    lcd.print("Dark!");

  }

  lcd.setCursor(2, 1);

  lcd.print(" Lux: ");

  lcd.print(lux);

  //Afisare Monitor

  Serial.print(F("Humidity: "));

  Serial.print(h);

  Serial.print(F("%  Temperature: "));

  Serial.print(t);

  Serial.println(F("°C "));

  Serial.print(F("T = "));

  Serial.print(temp\_event.temperature);

  Serial.println(" \*C");

  Serial.print(F("P = "));

  Serial.print(pressure\_event.pressure);

  Serial.println(" hPa");

  Serial.print("Lux: ");

  Serial.print(lux);

  Serial.println();

}

float voltageToLux(float voltage) {

  // Formula de conversie depinde de caracteristicile LDR și configurația circuitului

  // Exemplu simplificat: Lux = (voltage \* 100)

  return voltage \* 100.0;

}