

Rețea de senzori bazată pe standardul IEEE 802.15.1

Darius O. Firan, Andrei C. Haisiuc, Ioan A. Hedeș

Universitatea Politehnică din Timișoara, Facultatea de Automatică și Calculatoare

Sisteme Încorporate, an de studiu 3, licență

coordonatori Mircea Popa, Sergiu Nimară

Abstract

Sistemul construit constă în noduri autonome informațional și energetic, care colectează date de temperatură, presiune și concentrație gazoasă pentru monoxid de carbon și metan din mediu, și le trimite către un receptor dedicat, care va înmagazina datele primite, făcându-le disponibile pentru vizualizare. Comunicarea se va realiza prin standardul IEEE 802.15.1, Bluetooth. Nodurile vor fi alcătuite din câte o placă de dezvoltare Arduino Nano, cu microcontroller Atmega328P, un modul cu senzori BMP180, două module cu senzor pentru detecția gazelor - MQ-7 pentru monoxid de carbon, respectiv MQ-4 pentru metan și un modul pentru comunicație Bluetooth HC-05. Pe partea de recepție, se va folosi o aplicație proprie pentru telefoane Android, implementată folosind Java și Android Studio. Datele pot fi vizualizate sub formă de tabele folosind un script Python.

Keywords: arduino, bluetooth, android, sensors, network, java, C, embedded, python

Descrierea Hardware-ului folosit

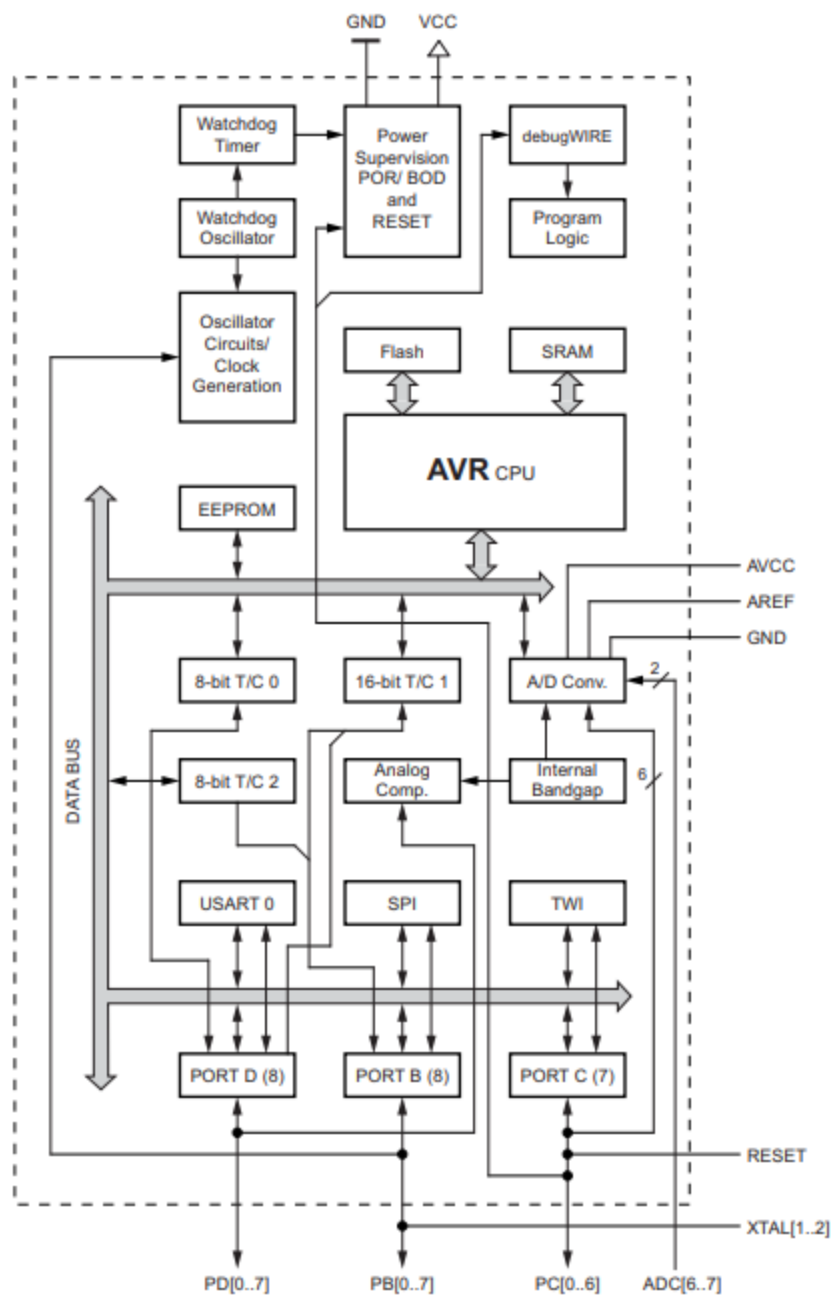
Arduino Nano

Arduino Nano este o placă de dezvoltare din familia Arduino, care folosește un microcontroller Atmega328P. Dispune de 8 terminale analogice pentru intrare și 22 de terminale digitale bidirecționale, dintre care 6 au capacități PWM.

ATmega328P

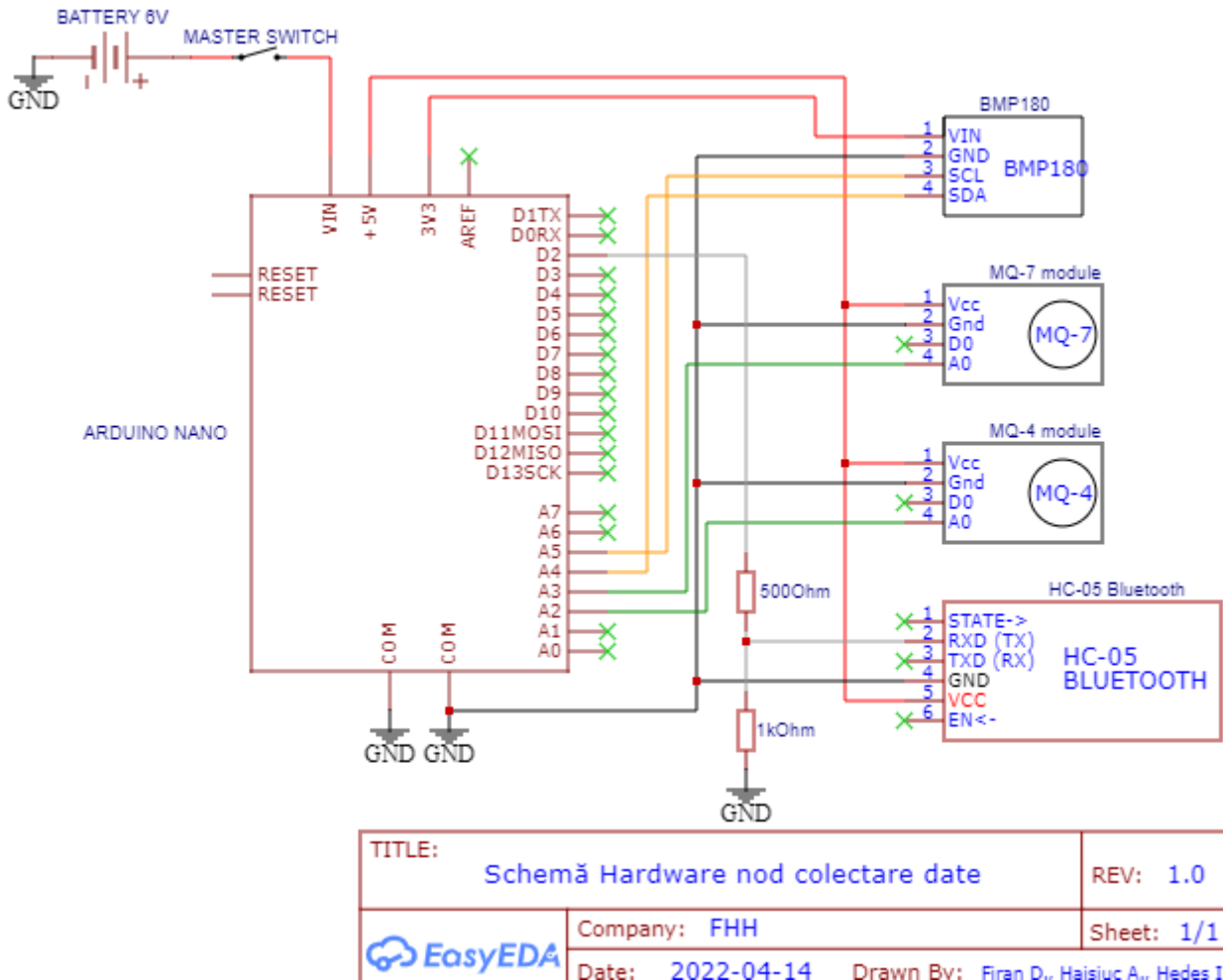
ATmega328P este un microcontroller din familia AVR, care folosește o arhitectură Harvard modificată, de tip RISC, pe 8 biți (1). Față de arhitectura Harvard clasică, cea modificată permite instrucțiunile din memoria corespunzătoare să fie interpretate drept date (1). Acest microcontroller dispune de o memorie Flash de 32 kiloocteți, o memorie EEPROM de 1 kilooctet și o memorie volatilă SRAM internă de 2 kiloocteți. Frecvența nominală de operare a microcontrollerului pentru tensiuni de alimentare între 4.5V și 5.5V este de 16MHz. Pentru tensiuni de alimentare între 2.7V și 4.5V, microcontrollerul poate funcționa la frecvența de 8MHz. Pentru periferie, circuitul dispune de module speciale pentru interfață serială USART, interfață serială cu 2 fire I²C, interfață serială SPI și module a lățimii impulsurilor (PWM). Din perspectivă electrică, circuitul este implementat în tehnologie CMOS. (2)

Schema bloc a lui ATmega328P



Arhitectura sistemului

Diagrama easyEDA



Detalierea conexiunilor

Alimentarea modulelor periferice

Deoarece modulul BMP180 lucrează la o tensiune de 3.3V (10), iar celălalte module lucrează la o tensiune de 5V (8) (9) (10), au fost conectate după cum urmează cele patru linii de alimentare ale breadboard-ului:

- Liniile de masă au fost conectate la o referință comună, preluată de la placa Arduino.

- Liniile + de pe breadboard sunt conectate la terminalul plăcii Arduino care oferă o tensiune de alimentare de 5V.

Conexiunile pentru BMP180

Primele două terminale ale lui BMP180, pornind de sus, SDA, respectiv SCL, sunt legate la terminalele analogice A4 și A5, folosite de placa Arduino Nano pentru emularea software a protocolului I²C. Nu este nevoie pentru rezistențe de ridicare pentru liniile SDA și SCL, deoarece modulul BMP180 are integrate aceste componente. Următoarele două terminale sunt folosite pentru alimentare. Pinul VCC este conectat direct la pinul 3V3 al Arduino-ului.

Conexiunile pentru HC-05

Pornind de sus în jos, primul terminal de interes al lui HC-05 este RXD. Pentru proiectul curent nu avem nevoie de pinul TXD al lui HC-05, deoarece modulul va fi folosit exclusiv pentru recepția datelor seriale de la microcontroller. Terminalul RXD va fi conectat printr-un divizor de tensiune la terminalul D2 al microcontrollerului. Divizorul de tensiune este folosit pentru a converti semnalele de 5V venite de la microcontroller în semnale de 3.3V (conform cu pagina produsului HC-05, acesta a fost testat, și funcționează și cu 5V pe acest pin).

Pinii dedicați pentru comunicații seriale ai plăcii Arduino, TXD și RXD nu pot fi întrebuințați, deoarece sunt folosiți pentru a încărca codul sursă. Pinii de alimentare sunt legați la liniile de alimentare ale breadboard-ului.

Conexiunile pentru MQ-4/MQ-7

Alimentarea modulelor MQ se face prin liniile de alimentare ale breadboard-ului. Datele de interes provenite de la senzor se transmit prin terminalul A0, care va trebui legat la un pin analogic de pe Arduino.

Module ale microcontrollerului implicate în sistem

I²C

Generalități

Inter-Integrated Circuit (I²C) este un protocol de comunicație serială sincron, multi-master/multi-slave, care folosește 2 linii pentru conectarea tuturor dispozitivelor implicate în transferuri:

- SDA – linia folosită pentru transferul datelor utile
- SCL – linia folosită pentru sincronizarea prin semnal de tact a dispozitivelor

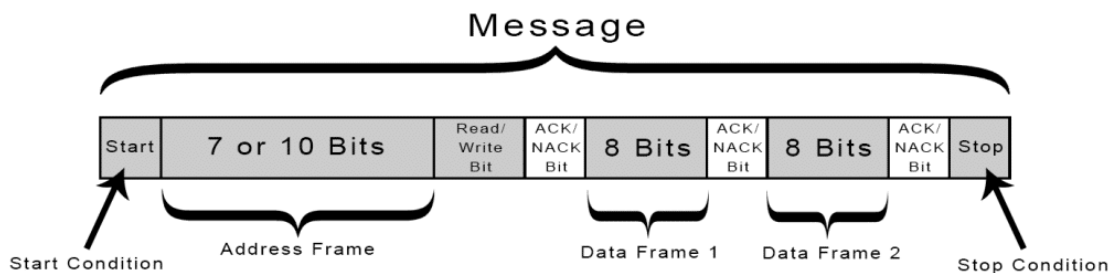
Pentru proiectul curent, I²C este folosit pentru interfațarea cu modulul BMP180.

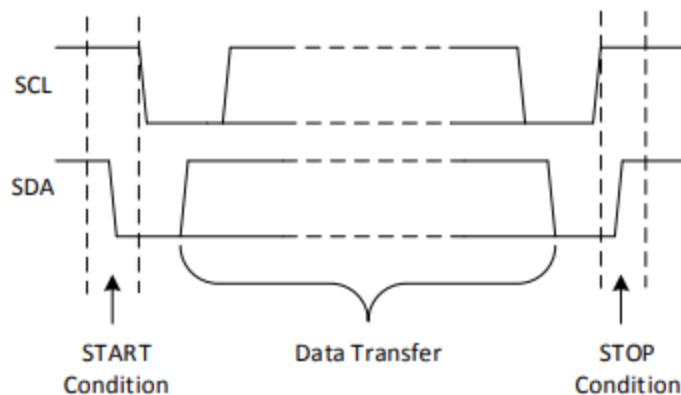
Identificarea circuitelor subordonate într-un sistem de comunicație bazat pe I²C se face folosind adrese pe 7 sau 10 biți. În cazul nostru, adresa I²C pentru BMP180 este 0x77 (11).

Liniile I²C sunt comandate de terminale cu drenă deschisă ale circuitelor. Din această cauză, circuitele nu pot plasa 1 logic pe cele 2 linii – pot plasa 0 logic, sau să lase linia deschisă (impedanță înaltă). Pentru a fi setat nivelul 1 logic pe cele două linii, se folosesc rezistențe ridicătoare. (3)

În cazul acestui proiect, interfața serială I²C este simulată în software folosind biblioteca Arduino Wire.h.

Transmisia mesajelor





Comunicarea în I²C se face prin mesaje, transmise pe linia SDA, fiecare mesaj transmițându-se în următoarele etape:

- *Condiția de start* – linia SDA este adusă de către master de la 1 logic la 0 logic, cât timp linia SCL este pe nivelul 1 logic
- masterul trimite 7 sau 10 biți reprezentând adresa slave-ului cu care se vrea realizarea unui transfer.
- masterul trimite un bit, în funcție de valoarea căruia slave-ul va ști dacă se va executa o scriere sau o citire.
- slave-ul selectat confirmă recepționarea corectă a cererii de transfer.
- este trimis un octet util, în funcție de sensul ales al transferului.
- se confirmă recepția corectă a octetului util
 - ultimii 2 pași se pot repeta pentru mai mulți octeți
- *Condiția de oprire* - linia SDA este adusă de către master de la 0 logic la 1 logic, cât timp linia SCL este pe nivelul 1 logic

USART



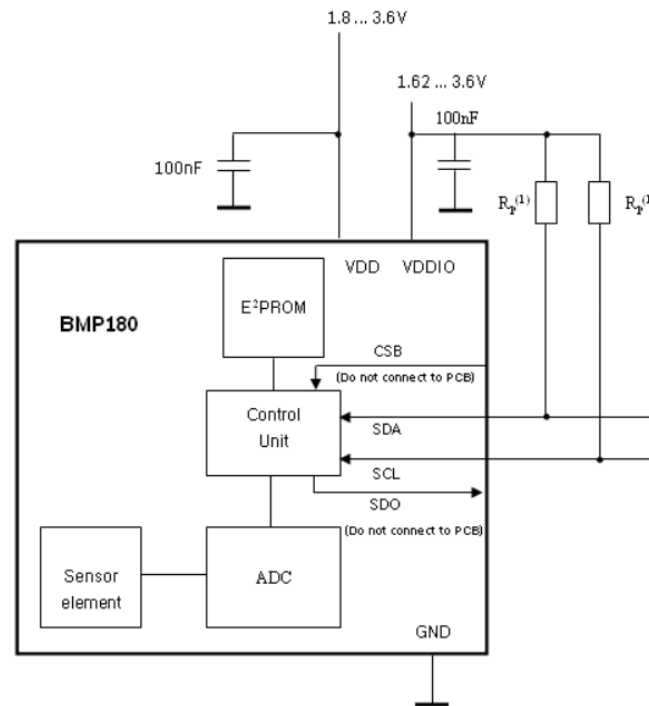
Universal Synchronous and Asynchronous Receiver-Transmitter (USART) este un tip de interfață serială, cu capacități full-duplex. În cadrul acestui proiect se folosește protocolul USART pentru comunicare serială asincronă. Comunicarea serială asincronă este un tip de comunicare în care terminalele nu sunt continuu sincronizate de un tact comun. Sincronizarea între capete se face prin semnale de start și de stop, reprezentate de biți specifici în structura mesajelor trimise. Avem un singur bit de start, dar putem avea mai mulți biți de stop. În cadrul acestui tip de transfer, datele se transmit la nivel de caracter – 8 biți. Pentru a se putea sincroniza, terminalele trebuie să funcționeze la aceeași rată de transfer, numită și baudrate. Valori uzuale pentru acest parametru sunt: 4800, 9600, 19200, 115200. Fiecare caracter transmis poate conține, pe lângă biții utili și biții de cadrare și un bit de paritate, pentru verificarea integrității transmisiei. Acest bit va ocupa unul din pozițiile dedicate biților utili. (5)

În cazul acestui proiect, interfața serială USART este simulată în software folosind biblioteca Arduino SoftwareSerial.h.

Descrierea tehnică a senzorilor și a modulelor folosite

BMP180

Schema bloc a modulului



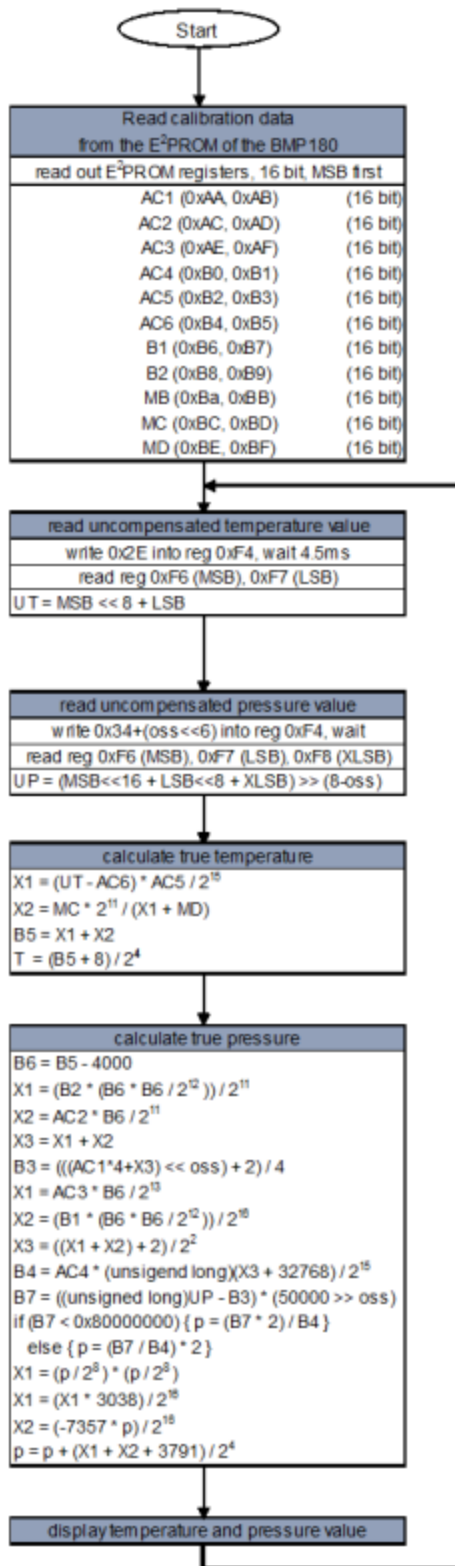
Generalități

Senzorul de presiune BMP180 constă într-un senzor [piezorezistiv](#), un convertor analogic-digital, o unitate de control cu E²PROM și o interfață serială I²C. (11)

Acesta este construit pentru a fi direct conectat la un microcontroller prin bus-ul I²C. Datele legate de temperatura și presiune sunt oferite de către senzor în forma lor necompensată. Memoria E²PROM conține 176 biți cu date, care trebuie încărcate în memoria microcontrollerului, și folosite pentru convertirea valorilor necompensate în valori utile.

Mod de obținere al datelor utile

Modul de obținere al datelor este explicat în detaliu în fișa de catalog a circuitului, din care extragem următorul algoritm. (11)

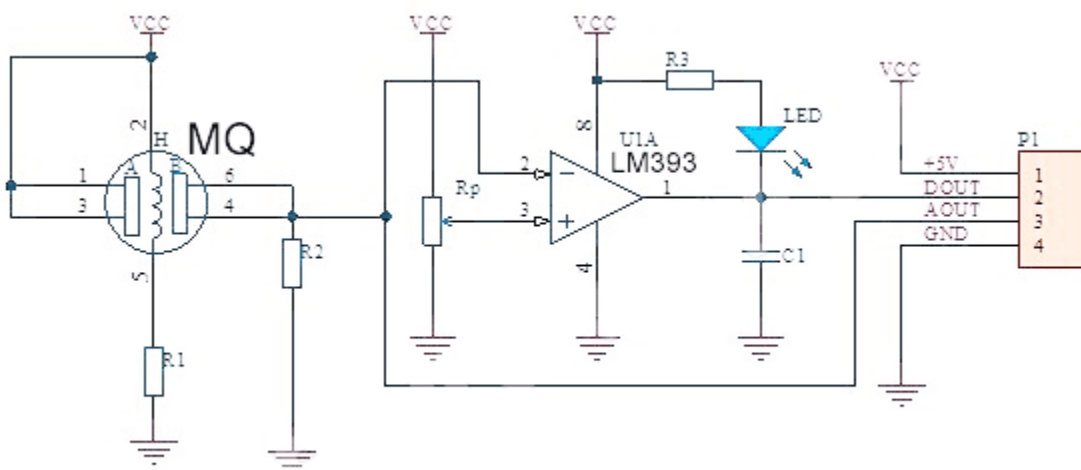


Conform algoritmului descris în poza alăturată, pentru a obține o valoare a temperaturii în grade Celsius precum și o valoare a presiunii în hPa procedăm în următorul mod:

1. Preîncărcăm datele de calcul de care avem nevoie din memoria E²PROM prin interfața I²C
2. Cerem modulului o valoare de temperatură prin registrul său de comandă
3. Citim o valoare necompensată pentru temperatură de la registrul de stare al modulului
4. Cerem modulului o valoare de presiune prin registrul său de comandă
5. Citim o valoare necompensată pentru presiune de la registrul de stare al modulului
6. Calculăm valoarea reală a temperaturii (în °C)
7. Calculăm valoarea reală a presiunii (în hPa)
8. Revenim la punctul 2.

MQ-7 / MQ-4

Schema bloc a modului



Diferențele între modulul cu senzor MQ-7 și modulul cu senzor MQ-4 constau în rezistențele chimice folosite în interiorul modulului cu senzor. Astfel, schema bloc și principiul de funcționare al senzorilor sunt identice. (12)

Generalități

Senzorii din familia MQ sunt dispozitive care măsoară concentrația diferitelor gaze din aer. În cazul lui MQ-4 vorbim despre metan (CH_4), iar în cazul lui MQ-7 vorbim despre monoxid de carbon (CO). În structura lor internă, senzorii au câte un radiator care ajută la calibrare și câte un senzor electrochimic cu oxid de staniu (SnO_2).

În cadrul acestui proiect, senzorii se folosesc în cadrul unor module care se ocupă automat de încălzirea și calibrarea lor. (12)

Funcționarea senzorilor se învârtă în jurul unei rezistențe chimice SnO_2 , care are electroni donori liberi. Moleculele din gazele aflate în jurul senzorului atrag și împing electronii liberi față de rezistență. În funcție de concentrația gazoasă, valoarea rezistenței senzorului se

modifică. Senzorii sunt sensibili la mai multe gaze, existând câte o ecuație caracteristică a valorii rezistenței în funcție de concentrația gazoasă pentru fiecare gaz detectat.

Mod de obținere al datelor

Rezistența de măsurare a senzorului se găsește într-un raport de divizare cu o rezistență de sarcină din modul, numită R_L . În cazul modulelor folosite în acest proiect, R_L a fost măsurat la 1kOhm. Astfel, R_S poate fi calculat în următorul fel folosind valoarea citită a tensiunii de ieșire a senzorului: (12)

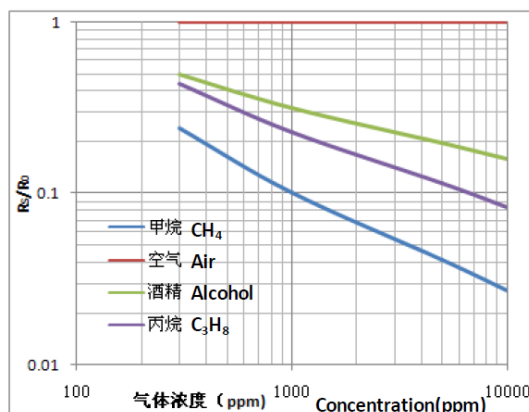
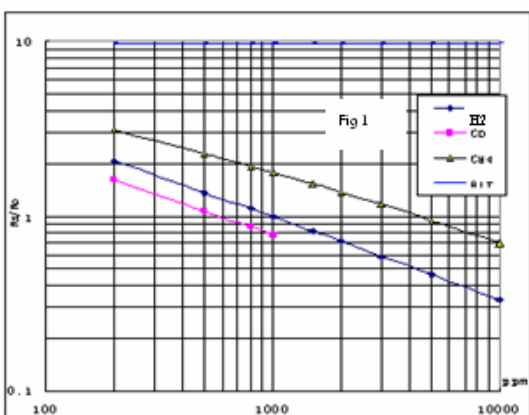
$$R_s = \left(\frac{V_{cc}}{V_{out}}\right) * R_L, \text{ unde } V_{cc} = 5V, R_L = 1kOhm \quad (1)$$

Pentru a obține date valabile, avem nevoie în primul rând de valoarea rezistenței R_S a senzorului în aer curat, notată R_0 . Datele finale obținute de la senzor vor fi calculate ca funcție de variabilă adimensională R_S/R_0 . Principal, în aer curat este valabilă egalitatea $R_S/R_0 = 1$. Rezistența senzorului se află într-un raport de divizare cu o rezistență de sarcină R_L .

De asemenea, mai avem nevoie de coeficienții funcției caracteristice

$$ppm = f\left(\frac{R_s}{R_0}\right) = coef_a * \frac{R_s}{R_0}^{coef_b}$$

care pot fi aflați consultând caracteristicile de senzitivitate, prezentate în fișele de catalog ale senzorilor, redate mai jos pentru MQ-7, respectiv pentru MQ-4.



Conform wikipedia, ecuațiile prezentate în grafice logaritmice se pot afla pe baza formulei:

$$F(x) = F_0 \left(\frac{x}{x_0} \right)^{\frac{\log(F_1/F_0)}{\log(x_1/x_0)}},$$

unde (x_0, F_0) și (x_1, F_1) sunt două puncte de pe grafic. Această formulă ne dă funcția

$\frac{R_s}{R_0} = f^{-1}(ppm)$, care prin inversare devine funcția căutată. Prin egalarea coeficienților, obținem

următoarele valori de coeficienți valabile pentru proiectul nostru:

1. MQ-7 (CO)

a. $coef_a = 611.082$

b. $coef_b = -2.105$

2. MQ-4 (CH4)

a. $coef_a = 38.9835$

b. $coef_b = -1.409$

Plecând de la considerațiile precedente, pașii întreprinși pentru a obține date valabile ale concentrațiilor gazoase sunt:

1. Calcularea valorilor R_0 , în aer curat imediat după pornirea modulului.

- a. citim o valoare de la pinul A0 al modulului, folosind funcția `analogRead(PIN_NO)`; valoarea va reprezenta folosind valori întregi pe 10 biți fără semn (0-1023) o tensiune de ieșire între 0 și 5V.
- b. convertim, folosind regula de 3-simplă valoarea pe 0 biți într-o valoare în virgulă flotantă între 0 și 5V.
- c. folosim formula (1), derivată din formula divizorului de tensiune, pentru a obține valoarea rezistenței $RS=R_0$

2. Citirea continuă a valorilor concentrației de gaze în părți per milion (ppm).

a. aceeași pași ca la pasul 1, care rezultă valoarea lui R_S

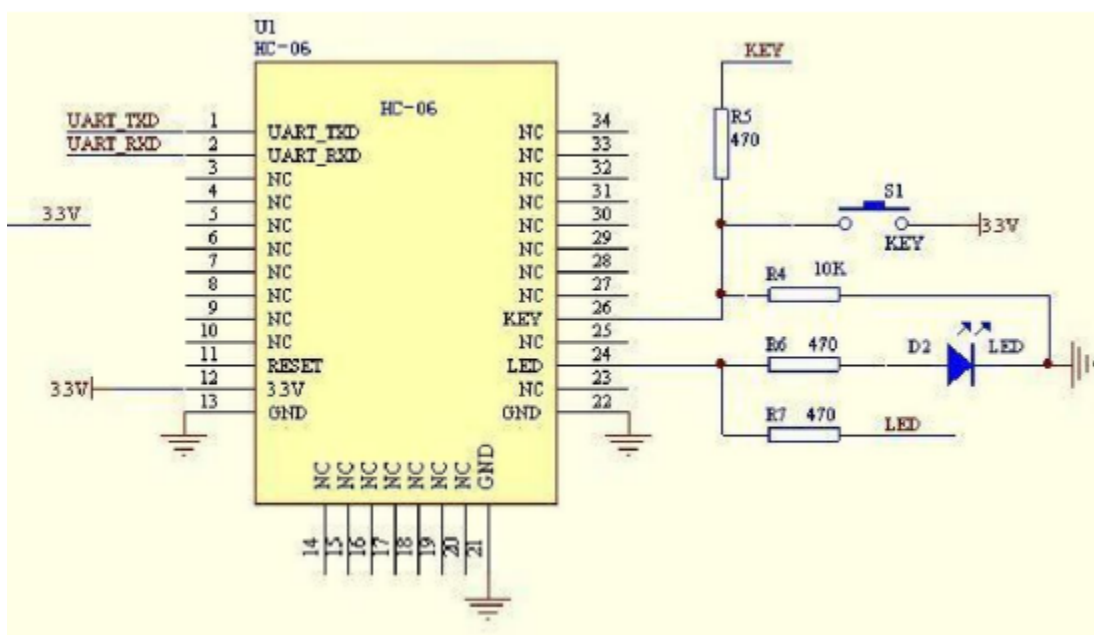
b. calcularea raportului $\frac{R_s}{R_0}$

c. calcularea $ppm = f(\frac{R_s}{R_0})$, folosind coeficienții obținuți

Pentru a elimina citiri false cauzate de erori transiente, codul Arduino asociat acestui proiect raportează media ultimelor 10 valori citite de la senzor.

HC-05 (ZS-040)

Schema bloc a modulului



Alimentarea modulului acceptă tensiuni între 3.3V și 6V, deci vom putea alimenta modulul și cu 5V. (10)

Generalități

Modulul pentru comunicații Bluetooth HC-05 poate lua atât rol de master, cât și de slave într-o legătură de date Bluetooth. Configurarea modulului se poate face prin Arduino accesând

modul AT al modulului și folosind comunicare serială între microcontroller și modul. Accesarea modulului AT se face menținerea valorii „1” logic la terminalul EN (sau KEY) al modulului în timpul pornirii acestuia. O listă cu comenzile de configurare disponibile poate fi găsită aici (bagă link).

O dată configurat modulul (parolă + vizibilitate + nume), dispozitivele Bluetooth se pot împerechea cu el normal. Doar un singur dispozitiv poate fi conectat la un HC-05 la un moment dat.

Descrierea Codului Sistemului

Întreg codul aplicației poate fi găsit în următorul repository de GitHub:

<https://github.com/ionhedes/bluetooth-sensors>

Cod C/C++ pentru Arduino

Codul Arduino se găsește în folder-ul [/data transmitter](#) din repositoryul mai sus menționat. Acest cod este împărțit în blocuri caracteristice fiecărui dispozitiv folosit, iar toate funcțiile sunt comentate.

Optimizări speciale cod Arduino:

1. pentru a nu apela **pow()** pentru calculul puterilor lui 2, au fost definite macro-uri speciale cu aceste valori
2. funcția **get_averaged_ppm()** va apela **pow()** o singură dată pentru 10 valori citite de la senzor, în loc să apeleze o dată pentru fiecare valoare

Cod aplicație Android

Codul Java pentru Android se găsește în folder-ul [/app/src/main](#) din repositoryul mai sus menționat.

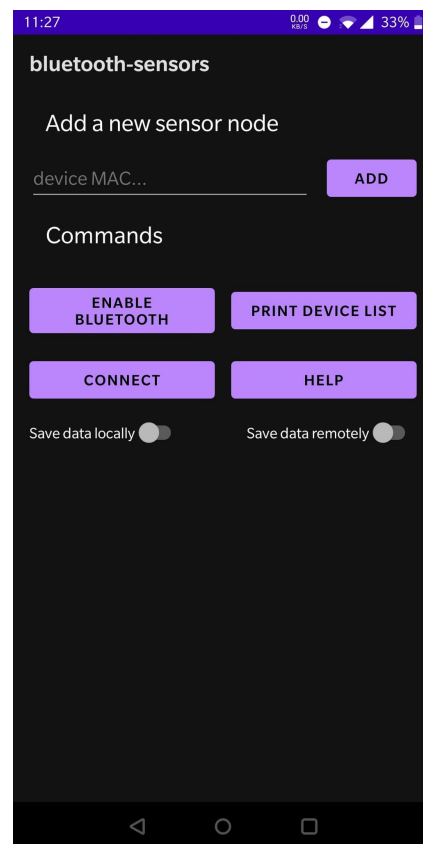
Aplicația Android conține 2 [componente activități](#): o componentă de gestiune, care pornește la deschiderea aplicației, și o componentă de transfer date, în cadrul căreia utilizatorul poate observa datele colectate.

Componenta de gestiune

Aceasta se folosește pentru adăugarea sau ștergerea nodurilor de senzori în aplicație.

Nodurile sunt identificate prin intermediul adreselor MAC.

- „Add” va adăuga în lista de noduri din aplicație adresa MAC furnizată în câmpul de text, dacă aceasta este validă (se fac verificări asupra formatului adresei folosind expresii regulate)
- „Enable Bluetooth” va porni Bluetooth pe dispozitiv.
- „Connect” va porni componenta transferului de date.
- „Help” redirectionează utilizatorul către repoziitoriul de GitHub al aplicației.
- Butoanele glisante se activează înainte de apăsarea „Connect” dacă utilizatorul vrea să salveze datele. Local, datele se salvează într-un fișier .json. Pentru salvare în cloud, se folosește Google Firebase.



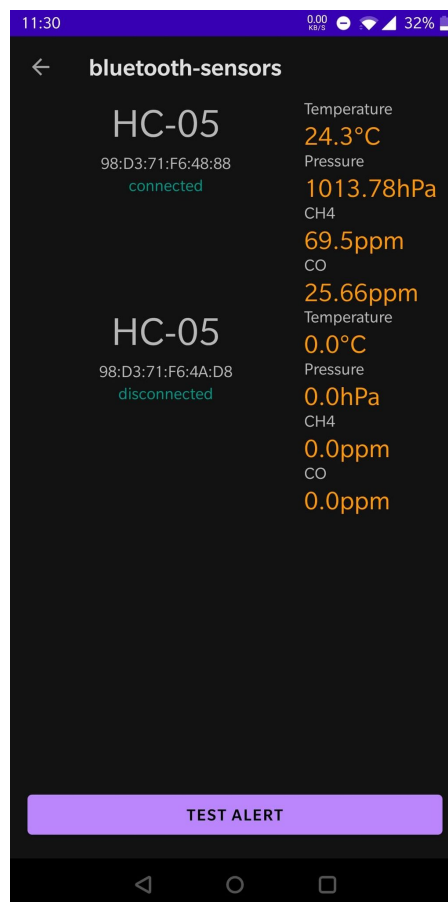
Componenta de transfer de date

Aceasta se folosește pentru recepția și vizualizarea în timp real a datelor provenite de la dispozitivele disponibile și înregistrate în telefon și aplicație.

ATENȚIE: înainte de a folosi aplicația, nodurile Bluetooth trebuie împerecheate prin Bluetooth cu telefonul din setările sistemului Android, și adăugate prin prima componentă a aplicației.

Butonul „Test Alert” declanșează forțat alarma pentru concentrații prea mari de gaze. Este folosit exclusiv pentru testare.

Lista afișată conține o intrare pentru fiecare nod înregistrat. Fiecare nod din listă are un fir de execuție dedicat, în care aplicația va încerca să se conecteze la modulul Bluetooth al acelui nod. Odată conectată, aplicația va primi date și le va afișa pe ecran. Dacă este cazul, datele primite vor fi și salvate.



Cod de generare grafice

Codul de generare grafice este scris în Python și se găsește în repositoryul indicat mai sus, pe calea [/data_fetcher](#). Scriptul primește ca argumente din linia de comandă:

- adresa MAC a nodului a căror date se vor folosi pentru trasarea graficelor
- un cuvânt din lista: cycle (10 minute), hour, day, week, month care să desemneze intervalul de timp pe care se face graficul

Datele folosite la grafice provin de pe Firebase.

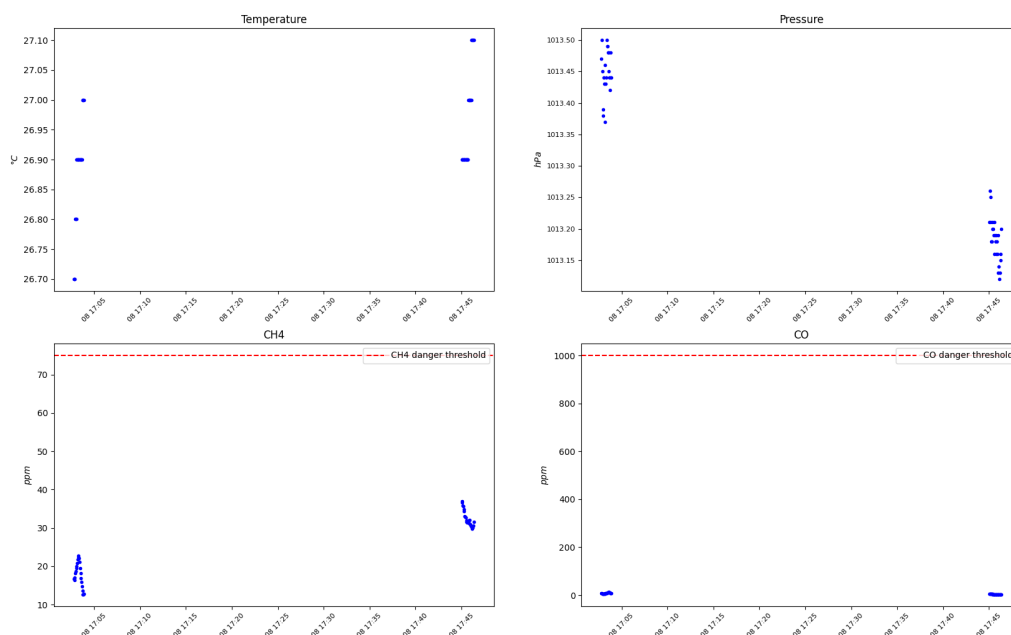
Codul folosește următoarele dependențe, care trebuie instalate pe calculatorul utilizatorului:

- matplotlib - pentru grafice
- firebase-admin - pentru legătura la baza de date

De asemenea, utilizatorul are nevoie de o cheie de access la baza de date.

În urma rulării script-ului va fi generat un fișier .png care conține graficele celor 4 mărimi cuantizate, numit în funcție de momentul apelului.

Readings for last day on 2022/05/08, 20:44:36



Bibliografie

- (1) Wikipedia, [*AVR microcontrollers*](#), link accesat 18.03.2022
- (2) Atmel Inc., [*ATmega328P Datasheet*](#), link accesat 18.03.2022
- (3) Texas Instruments, [*Understanding the I2C Bus*](#), link accesat 01.04.2022
- (4) circuitbasics.coms, [*Basics of the I2C Communication*](#), link accesat 02.04.2022
- (5) techtarget.com, [*What is USART?*](#), link accesat 02.04.2022
- (6) IBM Inc., [*Parity Bits*](#), link accesat 05.04.2022
- (7) IBM Inc., [*Start, stop, and mark bits*](#), link accesat 05.04.2022
- (8) Henan Hanwei Electronics Co., Ltd., [*MQ-7 Semiconductor Sensor for Carbon Monoxide Datasheet*](#) (descărcare PDF)
- (9) Winsen Co., Ltd., [*MQ-4 Flammable Gas Sensor Datasheet*](#)
- (10) unknown, [*ZS-040 Module with HC-05 Datasheet*](#)
- (11) Bosch Sensortec, [*BMP180 Digital Pressure Sensor*](#)
- (12) Roland Pelayo, [*How To Use the MQ-7 Carbon Monoxide Sensor*](#), link accesat 25.04.2022
- (13) Google Inc., [*Android Development Documentation*](#), link accesat 13.03.2022
- (14) Wikipedia, [*Log-log plot*](#), link accesat 20.03.2022