

APLICAȚIE PE SISTEM DE OPERARE ANDROID PENTRU MONITORIZAREA ȘI COMANDA DISPOZITIVELOR EXTERNE

Alexandru MOLDOVAN^{1*},
Alin GOANȚĂ¹,
Grațian COJOCARIU¹

¹Universitatea Politehnica Timișoara, Facultatea de Inginerie Hunedoara, Departamentul de Inginerie Electrică și Informatică Industrială, grupa Inginerie Electrică și Calculatoare, an II, Hunedoara, Romania

*Autorul corespondent: Alexandru MOLDOVAN, moldovanalexandru2000@gmail.com

Rezumat. *Lucrarea de față prezintă realizarea unei aplicații software care se încarcă pe telefonului mobil și poate monitoriza și controla de la distanță senzori, respectiv elemente de execuție. Concret, aplicația este programată pe platforma App Inventor și monitorizează prin comunicație bluetooth senzori de temperatură, gaz, iluminare și poate comanda un sistem de iluminare și funcționarea unei pompe de aspirație / refulare apă.*

Cuvinte cheie: *App Inventor, Android, Arduino, transmisie bluetooth, senzori.*

Introducere

MIT App Inventor este o aplicație web open-source dezvoltată de Google ce permite crearea de programe pentru două tipuri de sisteme de operare: Android și iOS. Platforma utilizează o interfață grafică utilizator similară cu limbajul de programare Scratch, în care programatorul poate crea obiecte prin drag-and-drop. În urma compilării se crează o aplicație ce poate fi transferată pe dispozitivul mobil.

Sistemul pe care această aplicație îl monitorizează reprezintă o machetă de tip smart-home echipată cu senzori de monitorizare controlați de o placă de dezvoltare Arduino Mega.

Descrierea sistemului cu senzori

Sistemul cu senzori de tip smart-home se alimentează de la rețeaua electrică 230V, 50Hz prin intermediul unei surse cu transformator și stabilizator de tensiune care generează diferite nivele de tensiune disponibile prin intermediul unor barete: 3,3V, 5V, 12V. Placa de dezvoltare de tip Arduino Mega este programată pentru achiziția în timp real a mai multor senzori de monitorizare pe diverse magistrale de comunicație (Figura 1).

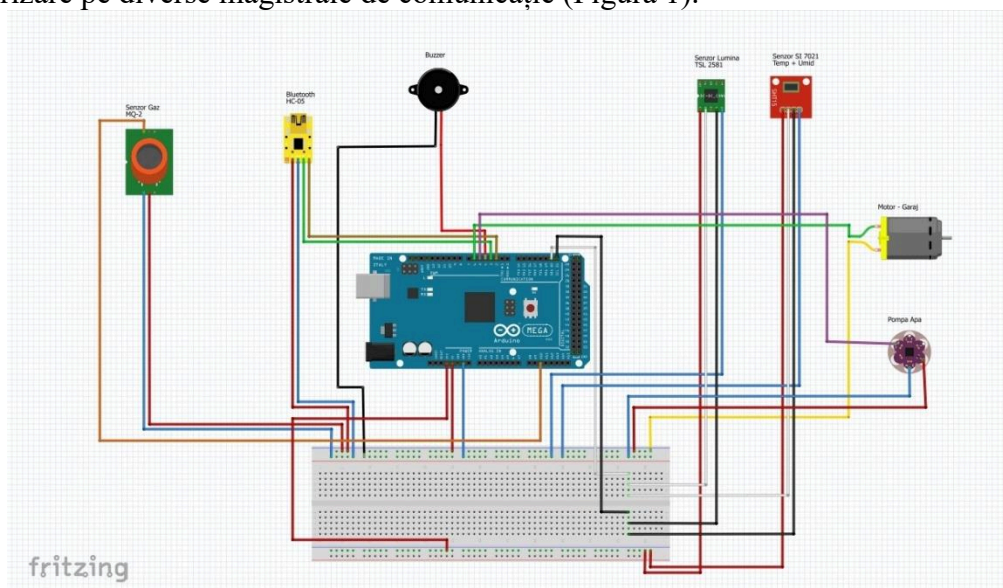


Figura 1. Schema de conectare a dispozitivelor.

Crearea interfeței cu utilizatorul

Interfața cu utilizatorul se poate proiecta utilizând o multitudine de opțiuni de încadrare în spațiul ecranului telefonului mobil, existând astfel diferite aranjamente pentru amplasarea butoanelor, a slider-elor, a listelor. Pentru comunicația cu bluetooth-ul este necesară generarea unei liste cu dispozitivele bluetooth disponibile la un moment dat și din care utilizatorul să aleagă dispozitivul HC-05 conectat la Arduino Mega. De asemenea, se mai alege un indicator pentru urmărirea status-ului comunicației. Astfel, indicatorul va afișa textul “*Conectat*” sau “*Deconectat*”, în funcție de situație (Figura 2).

Interfața are o componentă ce nu este vizibilă utilizatorului, este vorba despre rata de achiziție a semnalelor. Funcția are denumirea *Clock* și poate fi setată la milisecunde.

În vederea monitorizării senzorilor de temperatură, umiditate, gaz și iluminare s-au creat indicatoare de tip numeric în care sunt afișate valorile citite în timp real de senzori și care sunt trimise prin comunicație serială Bluetooth în format șir de caractere. Textul ce apare pe indicatoare poate fi editat.

Pentru variația gradului de iluminare în cele trei încăperi ale machetei smart-home, se proiectează trei slidere ce vor comanda Arduino să varieze factorul de umplere al iluminării cu benzi LED. Cele trei slidere vor genera valori numerice cuprinse între 0 și 255, corespunzătoare valorii factorului de umplere dorit. Informația de comandă se preia prin date de tip șir de caractere. Acestea vor fi comparate cu etichetele cu care au fost programate butoanele, astfel încât aplicația Arduino să știe în orice moment care slider a fost acționat.

Comanda de umplere a piscinei se face cu ajutorul unei pompe peristaltice de curent continuu alimentată la 12V. Butonul “Umplere piscină” va comanda pornirea pompei la turație maximă, iar butonul “Golire piscină” generează inversarea polarității motorului pompei, astfel încât pompa va aspira apa din piscină.

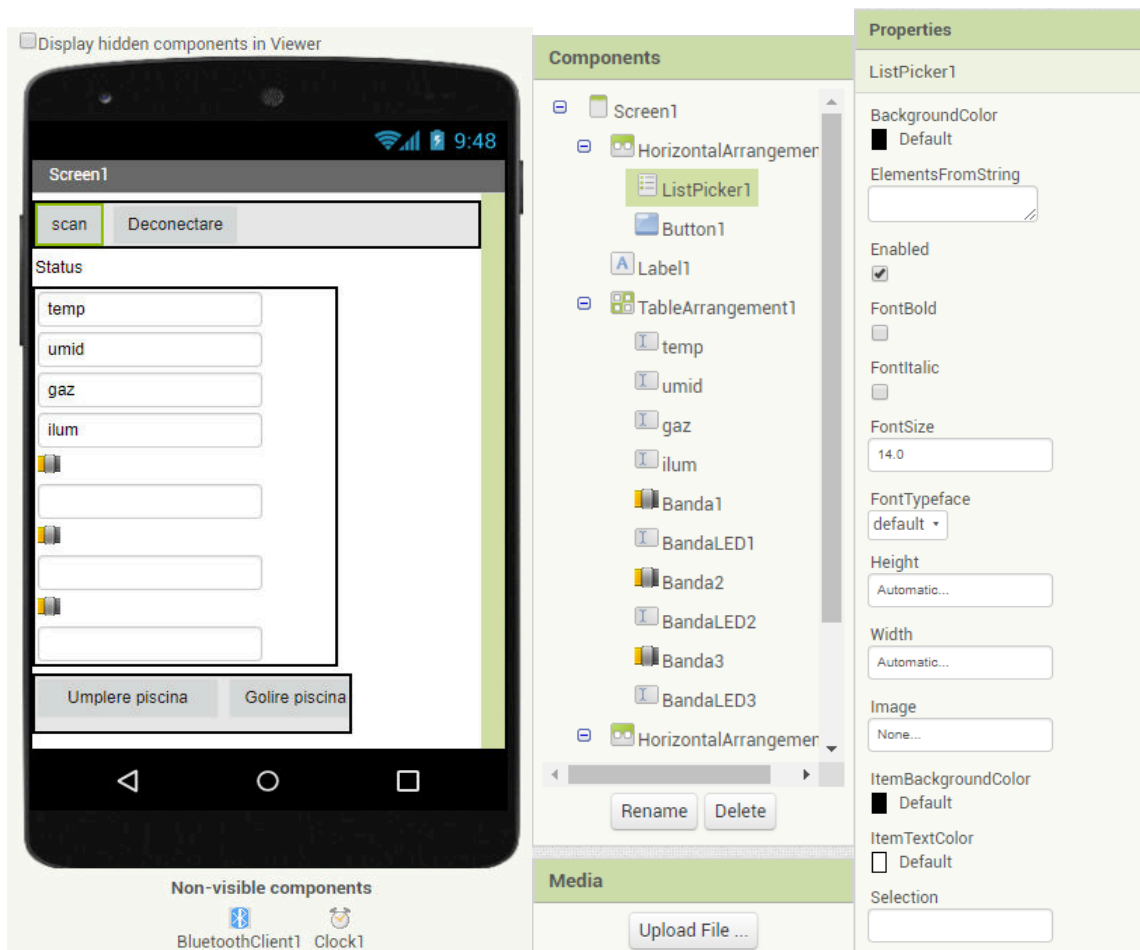


Figura 2. Crearea interfeței grafice cu utilizatorul.

Codul sursă al aplicației

S-au utilizat funcții nerepetitive pentru căutarea dispozitivului bluetooth conectat la Arduino și pentru afișarea status-ului comunicației.

```

when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames

when ListPicker1 .AfterPicking
do set ListPicker1 .Selection to call BluetoothClient1 .Connect
address ListPicker1 .Selection
set ListPicker1 .Text to "Conectat"
    
```

În vederea achiziționării datelor pe aplicația de pe telefon, s-a creat o variabilă globală.

```

initialize global data to create empty list
    
```

Funcția repetitivă principală este cu condiția la început și este temporizată de funcția *Clock* programată pe interfața cu utilizatorul.

Datele privind valorile generate de senzorii de gaz, iluminare, temperatură și umiditate sunt concatenate într-un șir de caractere la nivelul programului implementat pe Arduino. La fiecare iterație a structurii repetitive a aplicației App Inventor se preia câte un șir de caractere de la Arduino. Acest șir este despărțit prin spații, astfel că structura repetitivă folosește funcția *Split at spaces* pentru ca informația să poată fi afișată pe fiecare indicator în parte. Astfel informația despre umiditate este prima din șirul de caractere, astfel că primește indexul 1. Temperatura are indexul 2, temperatura are indexul 3, iar iluminarea are indexul 4 din șirul de caractere trimis prin bluetooth. Cu ajutorul funcției *Join*, s-a atașat fiecărei mărimi achiziționate unitatea de măsură corespunzătoare.

```

when Clock1 .Timer
do if BluetoothClient1 .IsConnected and call BluetoothClient1 .BytesAvailableToReceive > 0
then set global data to split at spaces call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set umid .Text to join join "umid"
select list item list get global data
index 1
"%"
set temp .Text to join join "temp"
select list item list get global data
index 2
"grd C"
set gaz .Text to join join "gaz"
select list item list get global data
index 3
" "
set ilum .Text to join join "ilum"
select list item list get global data
index 4
"Lux"
    
```

Comunicația pe bluetooth se realizează bidirecțional. Comanda și controlul sistemului de iluminat al căsuței se face prin 3 slidere programate pe interfața aplicației pe mobil. Astfel, când variabila atașată sliderului se modifică la dorința utilizatorului, valoarea întreagă a acesteia (cuprinsă între 0 și 255) se transmite prin bluetooth către Arduino împreună cu textul BandaLED pentru ca Arduino să știe care din cele 3 benzi LED este apelată la un moment dat.

```
when Banda1 .PositionChanged
  thumbPosition
do
  call BluetoothClient1 .SendText
  text join " BandaLED1 "
  Banda1 . ThumbPosition
  set BandaLED1 . Text to Banda1 . ThumbPosition

when Banda2 .PositionChanged
  thumbPosition
do
  call BluetoothClient1 .SendText
  text join " BandaLED2 "
  Banda2 . ThumbPosition
  set BandaLED2 . Text to Banda2 . ThumbPosition

when Banda3 .PositionChanged
  thumbPosition
do
  call BluetoothClient1 .SendText
  text join " BandaLED3 "
  Banda3 . ThumbPosition
  set BandaLED3 . Text to Banda3 . ThumbPosition
```

Concluzii

MIT App Inventor este o modalitate de programare de aplicații pe mobil extrem de eficientă, necesită doar un cont personal de Google. Comunicația wireless se poate face și prin wi-fi.

Referințe

1. <https://create.arduino.cc/projecthub/Aritro/smoke-detection-using-mq-2-gas-sensor-79c54a>
2. <https://www.mouser.se/ds/2/588/TSL2581%20TSL2583-A-553251.pdf>
3. <http://ww1.microchip.com/downloads/en/devicedoc/usart.pdf>
4. <https://appinventor.mit.edu/explore/ai2/tutorials>