IoT Simulator

Cambose Alexandru

UAIC

1 Introducere

IoT este un concept ce presupune folosirea Internetului pentru a conecta între ele diferite dispozitive, servicii şi sisteme automate, formând astfel o reţea de obiecte. IoT aduce numeroase avantaje printre care se numara:

1) Colectarea de date
2) Abilitatea de a urmări şi monitoriza lucrurile
3) Automatizarea si eficientizarea muncii
4) O calitate mai bună a vieţii

IoT Simulator este un proiect ce are la baza conceptul de a conecta dispozitivele cotidiene la internet si controlul acestora de la distanta. Clientul, adica dispozitivul de pe care vom controla alte dispozitive contectate, va contine o interfata grafica si va juca rolul de un client al unui server. Dispozitivele controlate vor fi servere concurente ce vor primi comenzi de la clienti.

2 Tehnologiile utilizate

Interfata grafica

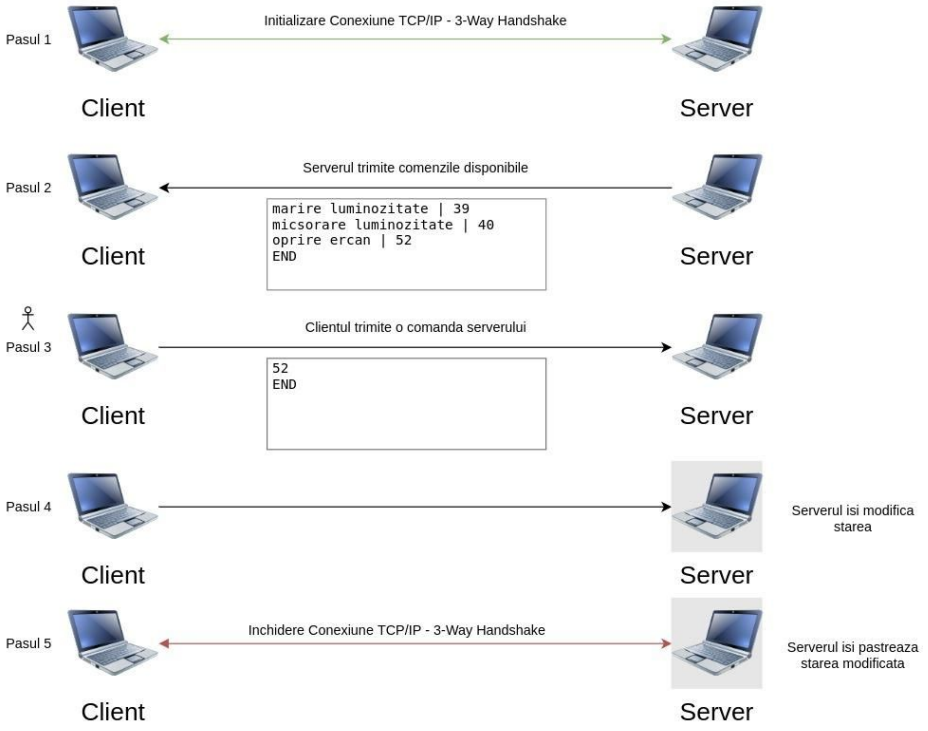
Pentru realizarea unei interfete grafice atractive vom folosi QT, o biblioteca cu widgeturi ce permite crearea unui program care sa ruleze atat pe Windows cat si pe Linux, Mac sau chiar dispozitivele mobile. Printre motivele alegerii acestei biblioteci se numara si: documentatia foarte bine realizata, suportul din partea comunitatii, licenta open-source si usurinta prin care putem realiza o interfata complexa. De asemena, QT este folosit si in aplicatiile enterprise, fiind astfel extrem de stabila.

Protocolul de comunicare

Pentru realizarea comunicarii dintre dispozitivele ce vor controla(clienti) cu cele ce vor primi comenzi si vor fi controlate(server) avem nevoie de un protocol de comunicare. Vom folosi TCP intrucat, comparativ cu UDP acesta garanteaza trimiterea, corectitudinea si ordinea datelor trimise catre destinatie. Aceasta caracteristica este extrem de utila in cazul in care vom dori sa controlam dispozitive

ce au o deosebita importanta din punct de vedere al modului in care acestea actioneaza, (exemplu: senzor de fum).

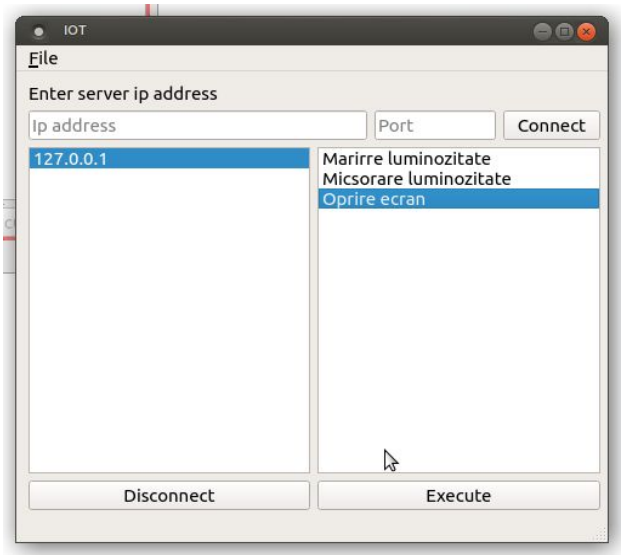
Arhitectura aplicatiei



Pasul 1: Initializare conexiune TCP

Pasul 2: Serverul va trimite comenzile disponibile. Modul in care acestea sunt formatate este urmatorul: fiecare comanda este de forma “<label> <cod>” unde label este o secventa de text care va fi afisata clientului pentru a descrie actiunea, iar codul va fi folosit pentru a trimite comanda respectiva catre server. Comenzile sunt separate prin newline iar la sfarsit va fi secventa “END”.

Pasul 3: Clientul selecteaza comanda si o trimite codul acesteia catre server. De asemenea, precum in pasul 2, mesajul trebuie sa fie precedat de secventa “END”. Serverul primeste codul si executa instructiunile asociate acestuia.



Pasul 4: Incheiere conexiune TCP

Detalii de implementare

In cazul intreruperii neasteptate a conexiunii, afisam utilizatorului un mesaj de eroare si schimbam starea aplicatiei client si a interfetei grafice pentru a suporta conectarea din nou a adresa ip si portul care s-a deconectat.

```
<item>
  <widget class="QPushButton" name="connectButton">
    <property name="enabled">
      <bool>false</bool>
    </property>
    <property name="text">
      <string>Connect</string>
    </property>
  </widget>
</item>
```

Elementele interfetei grafice sunt definite in format XML

```
#include "mainwindow.h"
#include <string>
#include <cstring>


using namespace std;
struct Command {
    string label;
    int code;
    void (*func) ();
};
Command availableCommands[10];

void incBrightness() {

}
void decBrightness() {

}
int main(int argc, char *argv[])
{
    availableCommands[0].label = "Marire luminozitate";
    availableCommands[0].code = 41;
    availableCommands[0].func = incBrightness;

    availableCommands[0].label = "Micsorare luminozitate";
    availableCommands[0].code = 42;
    availableCommands[0].func = decBrightness;

}
```

Fiecare server va contine un serie de comnezi disponibile pe care clientii le pot executa, aceste comennzi vor contine valorile: **label** - numele/descrierea comenzii, **code** - codul pe care clientul il va trimite pentru a executa comanda asociata cu acesta,

func - o functie ce va fi executaata atunci cand serverul primeste codul asociat acesteia.

Concluzii

In concluzie, prin utilizarea unei interfete grafice si al protocolului de comunicare TCP, avem o aplicatie nu doar cu aplicabilitate teoretica, dar si practica, care poate comunica in retea in mod robust si care este user-friendly.

O serie de imbunatatiri ar putea fi aduse. Printre acestea se numara, in mod special, modificari aduse protocolului intern de comunicare intre client si server pentru a trimite comenzi. Pe viitor acesta ar putea suporta comenzi cu unul sau mai multi parametri, confirmarea executarii cu succes al comenzii de catre server si un mod de a raporta eventualele erori la executarea comenzii. O alta imbunatatarie ar putea fi interfata grafica, pentru aceasta ar fi utila posibilitatea salvarii a adreselor ip ale serverelor si abilitatea de a asigna un alias text a adresei ip pentru o vizualizare intuitiva in cazul in care sunt mai multe servere conectate.

O alta imbunatatire ar putea fi a serverului, si mai exact al modului in care acesta stocheaza comenzile disponibile. In loc sa le stocam in codul efectiv al serverului, o alternativa mai flexibila si scalabila ar putea fi stocarea acestora intr-un fisier de configuratie, iar functia care in trecut ar fi fost executata si definita in interiorul executabilului va fi stocata ca un path la un executatbil extern. Aceasta abordare ar putea fi extrem de utila in cazul in care distribuim serverul, utilizatorul nu ar mai trebui sa recompileze codul de fiecare data cand doreste o schimbare a unei comenzi ci modifica doar fisierul de configuratie iar serverul trimite si executa dinamic comanda.

Bilbiografie

1. https://en.wikipedia.org/wiki/Internet_of_things
2. https://medium.com/zeux/the-internet-of-things-iot-5-reasons-why-the-world-needs-it-125fe71195cc
3. https://en.wikipedia.org/wiki/Qt_(software)
4. https://www.geeksforgeeks.org/differences-between-tcp-and-udp/
5. https://www.guru99.com/tcp-3-way-handshake.html
6. https://www.qt.io/
7. https://en.wikipedia.org/wiki/User_Datagram_Protocol
8. https://chrome.google.com/webstore/detail/marmoset/npkfpddkpefnmkflhhligbkofhnafieb?hl=en