

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Specializarea Informatică Economică

Casă inteligentă – Internet of Things

lucrare de licență

Coordonator

Prof. univ. dr. Cătălin Emilian BOJA

Absolvent

Flavia-Alice Pîrvu

București 2015

Declarație privind originalitatea continutului și asumarea răspunderii

Prin prezenta declar că rezultatele prezentate în această lucrare sunt în întregime rezultatul propriei mele creații cu excepția cazului în care se fac referiri la rezultatele altor autori. Confirm faptul că orice material folosit din alte surse (reviste, carti si site-uri de internet) este în mod clar referit in lucrare si este indicat în lista de referințe bibliografice.

Cuprins

[1. Introducere 1](#_Toc423722536)

[2. Problema abordata de lucrare [titlul capitolului este dependent de titlul lucrarii] 4](#_Toc423722537)

[2.1 Continut 4](#_Toc423722538)

[2.2 Studiu de piață pentru soluțiile existente 5](#_Toc423722539)

[2.3 Descrierea și analiza soluției 7](#_Toc423722540)

[3. Tehnologii/Metode utilizate 11](#_Toc423722541)

[3.1 Schița Arduino 11](#_Toc423722542)

[3.2 Server socket 12](#_Toc423722543)

[3.3 Servicii REST 13](#_Toc423722544)

[3.4 Client MQTT 13](#_Toc423722545)

[3.5 Client REST – Android 16](#_Toc423722546)

[4. Arhitectura soluției 17](#_Toc423722547)

[4.1 Diagrama de componente 17](#_Toc423722548)

[4.1 Diagrama de flux 18](#_Toc423722549)

[5. Implementarea solutiei 22](#_Toc423722550)

[5.1 Diagrama generală a cazurilor de utilizare – perspectivă MICROCONTROLLER 23](#_Toc423722551)

[5.2 Diagrama generală a cazurilor de utilizare - perspectivă UTILIZATOR 24](#_Toc423722552)

[6. Concluzii 27](#_Toc423722553)

# 1. Introducere

Frapant este faptul că o dată cu trecerea timpului, cu evoluția umanității și bineînțeles și a tehnologiei, unele lucruri au rămas neschimbate: nevoile. Sutele de ani care au trecut, generațiile care au adus câte un plus de valoare prin descoperirile și inovațiile lor, iar cu toate astea nevoile noastre au fost aceleași indiferent la ce perioada de timp ne raportăm. Ne dormim să fim iubiți, să ne simțim în siguranță, să ne simțim bine, să aducem un aport important în munca pe care o depunem, dar și în prieteniile cu cei din jur și totodată ne dorim să putem servi drept sprijin pentru familiile noastre și să jucăm un fel de rol – din păcate mic - în marea schemă a lucrurilor.

Așadar, ce s-ar întâmpla în momentul în care locuințele și orașele ar deveni inteligente?

Deși am încerca să urmărim toate noutățile, ar fi anevoios să prognozăm cum vor sta lucrurile în realitate, atunci când vom fi înconjurați de lucruri inteligente: case, birouri, străzi, orașe. Cert este că rezultatul va fi asemeni unui tsunami a ceva ce incepe cu pași mărunți, cu schimbări minore.

Sfera Internet of Things cumulează orice obiect sau ființă care poate avea asignată o adresă IP și are capacitatea de a transfera date într-o rețea. Putem spune că impresionanta creștere în spațiul adreselor a versiunii IPv6 reprezintă un factor important în dezvoltarea conceptului de Internet of Things. Potrivit lui Steve Leibson, putem afirma că expansiunea spațiului de adrese înseamna ca putem asigna o adresa IPv6 fiecărui atom de pe suprafața pământului, iar cu toate astea am avea suficiente adrese pentru a face aceste lucru pentru mai multe planete pămant. Cu alte cuvinte, oamenii pot atribui cu ușurință o adresă IP fiecărui “lucru” de pe Terra. Extinderea numărului de noduri inteligente, respectiv a multitudinii de date generate de aceste noduri, vor determina preocupări intense în domeniul datelor, cu privire la inegritatea și securitatea acestora (WhatIs, 2014).

Deși acest concept nu a fost numit până în 1999, Internet of Things a fost dezvoltat de-a lungul a multe decenii. Prima dată a fost aplicat la începutul anilor 1980, pe un dozator de Cola din Universitatea Carnegie Melon. Soluția implementantă le servea programatorilor pentru a se conecta prin intermediul Internetului la mașinărie, oferindu-le posibilitatea să verifice statusul mașinăriei, iar astfel i-ar scutit pe aceștia de un drum în zadar (WhatIs, 2014).

Din prim experiment putem vedea tendința oamenilor de a cunoaște, dar mai ales de a controla lucrurile pentru a-și facilita un grad ridicat de comoditate de care să beneficieze și în viața de zi cu zi. Soluțiile care ne sunt oferite astăzi încearcă să se plieze pe dorințele consumatorilor, dar în primul rând încearcă pe cât posibil să satisfacă nevoia de control pe care o au oamenii. Cum pentru fiecare activitate există un cost de oportunitate, oamenii platesc diverse prețuri pentru comoditate sau pentru un control cât mai mare asupra lucrurilor si pentru alte facilități de care mai pot dispune aceștia.

Nu doar prețul este ceea ce oamenii oferă în schimbul beneficiilor unei lumi cu obiecte intelgiente. Aceștia pot deveni dependenți de tehnologie, iar acest lucru s-ar putea răspândi de-a lungul a mai multe generații. În cazul unei defecțiuni a infrastructurii IoT sau a unei disfuncțiuni, dependenții de facilitățile obiectelor inteligente, ar putea întâmpina probleme.

În timp ce activitățile zilnice capătă cu timpul caracter automat, este de la sine înțeles că cererea de resursa umană va fi în scădere, iar primii vizați sunt muncitorii și personalul cu o educație precară. Acest fenomen poate deveni o mare problemă în societate.

Din moment ce sunt implicate mai multe dispozitive și tehnologii, există și o monitorizare din partea a multiple companii, iar acest fapt pune la îndoială securitatea și intimitatea. Depozitarea și regăsirea datelor devin un motiv important de îngrijorare pentr companii având în vedere că fiecare dintre ele sunt implicate în același timp. Informațiile disponibile în cadrul Internetului sunt o țintă către care hackeri înclină să lanseze atacuri. Ar fi un rezultat dezastruos dacă datele informațiile cofidențiale ar putea fi accesate de către intrușii neautorizați.

Cu toate că există aceste dezavantaje, oamenii preferă să-și asume riscul, dar să aleagă comoditatea și să fie in același pas cu trendul actual și caută să poată controla lucrurile cu care ei interacționează în viața de zi cu zi.

Deseori visăm și ne întrebăm: Cum ar fi dacă toate dispozitivele pe care le deținem ar fi conectate la Internet și ar comunica între ele, iar noi am putea să le dăm comenzi la care acestea să răspundă? Soluția pentru această dorință este automatizarea lucrurilor cu care aceștia vin cel mai des în contact, mai exact automatizarea locuințelor și a obiectelor din ea sau din jurul acesteia.

Ideea de automatizare a locuinței a fost un subiect care a existat de mai multe decenii în ceea ce privește iluminatul și controlul simplu, abia recent tehnologia a preluat idea unei lumi interconectatte care să permită controlul aboslut asupra locuinței, iar acest lucru tinde să devină cât mai aproape de realitate. Prin intermediul unei locuințe inteligente, utilizatorul poate să dicteze modul și momentul în care să reacționeze un dispozitiv și nu în ultimul rând motivul. Astfel, utilizatorul poate seta un orar, iar restul este automatizat, bazat pe preferințele personale ale utilizatorului respectiv, iar acest lucru îi oferă control, economie de bani și în plus, o casă inteligentă. Totodată o locuință automatizată poate să vină cu facilitatea de a avertiza utilizatorul cu privire la evenimentele pe care ar trebui să le afle nemijlocit, în timp ce acesta nu este acasă. De exemplu poate fi vorba de scurgeri de apă, de intruziuni nedorite, sau orice alte evenimente neplăcute. Utilizatorul poate în orice moment să modifice unele setări, sau să primească avertizări prin intermediul unui dispozitiv mobil.

Această nișă a sferei IoT a fost exploatată de diverse companii din lumea IT de-a lungul a mai multe decenii, iar printre aceștia se numără: IBM, Magnavox, Stanley, Sylvania etc. Tot în acest domeniu a investit și Google doar la inceputul anului trecut, cheltuind o sumă de 3,2 miliarde de dolari pentru a cumpăra Nest Labs, compania din spatele celor mai bine cotate profile de locuințe automatizate sau produse IoT de pe piață (Albanesius, 2014). În cadrul CES 2014, Samsung anunța inițiativa unei case inteligente, care va oferi utilizatorilor controlul asupra televizorului prin intermediul unui smartphone Galaxy. Tot la această expoziție IT, Archos anunță ca vor face un sistem pentru o casă conectată, cu opțiunea unei tablete care poate controla totul (Albanesius, 2014).

# 2. Problema abordata de lucrarea “Casă inteligentă – Internet of Things”

## 2.1 Conținut

Dispozitivele conectate sunt într-o continuă creștere, dar se poate spune că au ajuns în sfârșit în punctul în care este foarte posibil să se găsească cel puțin unul în fiecare casă. De la un pat care înregistrează bunăstarea din timpul somnului, la o furculiță care monitorizează viteza din timpul mesei, până la roboți care tund iarba, spală, aceste beneficii vor exista fie în interiorul sau în exteriorul lociunței.

Studii realizate în ultimii doi ani asupra populației spun că nici vârsta, genul sau nivelul venitului nu reușeau să anticipeze într-o mare măsură preferințele lor, ci mai degrabă nivelul de entuziasm pentru tehnologie. Aceștia au reușit să segmenteze populația intervevată, bazându-se pe entuziasmul oamenilor pentru tehnologie în 5 categorii, și anume: entuziaști convinși, ambivalenți, entuziaști, cei care nu sunt entuziasmați și moderații.

Automatizarea unei locuințe presupune introducerea unui nivel de control computerizat sau automatizat pentru anumite sisteme electrice și electronice dintr-o clădire, iar acestea pot include iluminatul, controlul temperaturii, sisteme de securitate, ușile garajului etc.

Modul în care obișnuim să controlăm televizorul utilizând telecomanda este banal, în comparație cu ceea ce am putea face dacă am dispune de o casă inteligentă. Fie că ne-am afla sau nu în împrejurimea locuinței, vom putea controla fiecare dispozitiv conectat din interiorul sau din exteriorul acesteia. Toate facilitățile de care am putea dispune pot să ne ofere un mod de viață mult mai comod într-un mediu simplu de controlat fie că ne aflăm sau nu la distanță.

O locuință automatizată poate acoperi la rândul ei nevoia de securitate și siguranță de care avem nevoie. Cel mai important este că spre deosebire de camerle obișnuite care pot să alerteze cu o întarziere ce poate fii suficent de costisitoare, o casă inteligentă ține casa constant sub supraveghere pentru ca utilizatorul să poată reacționa în momentul în care este notificat.

În timp ce costurile și sistemele variază de la locuință la locuință, toți prorietarii de case sunt ineteresați să întâlneascâ cele mai propice modalități pentru a-și ușura stilul de viață și totodată pentru a face ca respectivul sistem să se plieze pe stilul lor. Este dificil ca automatizarea locuinței să poată cuprinde exact ceea ce își pot dori clienții de la acest produs, tocmai de aceea există posibilitatea de a personaliza sistemul pe cât de mult posibil. Soluțiile existente pe piață par să fie cât mai complete, iar replica de marketing pecare o folosesc cel mai adesea pentru a promova sistemul de casă inteligentă este:”Poți controla totul!”. Aceasta este mai mult impresia pe care o lasă marketingul, deoarece există case care pot empatiza cu iuluminatul, altele sunt focusate pe securitate și unele dintre acestea sunt dedicate într-u totul distracției.

## 2.2 Studiu de piață pentru soluțiile existente

Multe dintre soluțiile pentru automatizarea locuințelor sunt disponibile pe piață, acestea fiind vândute predominant de companii care se axează pe dezvoltarea de astfel de produse. Alături de soluțiile oferite de companii, prin vânzarea lor, coexistă și soluțiile “open source” care oferă sub un spectru mult mai larg ce înseamnă și de ce este capabil acest domeniu. Realizând o comparație între produsele companiilor și cele care sunt “open source”, se poate observa că prima categorie poate fi încadrată într-un tipar, deoarece facilitățile cu care vin aceștia se află în aceeasi sferă și deja sunt predictibile prin soluțiile cu care vin pe piață. În schimb, cea de-a doua categorie, vine mereu cu un plus de valoare care fascinează prin forța continuă de inovație și reușește să surprindă.

A. Soluții “open source”

Printre cele mai intersante proiecte “open source” se numără o soluție a unui inginer software care vrea să elimine problema alterării din pricina temperaturii care poate alterna a loturilor de alcool. Până în momentul de față a reușit să monitorizeze temperatura constant prin intermediul unui senzor, iar acum lucrează la o souție eficientă pentru a modifica temperatura chiar și de la distanță (Dormehl, 2014).

O altă idee de acest tip este dedicată stăpânilor de câini pentru a fi mereu informați cu privire la ceea ce face, unde este, sau ce nevoi are animaulul de companie. În acest proiect dezoltatorul a folosit un sensor GPS, un senzor de temperatură, un senzor sensibil la lătrat. Câinele are atașat de zgardă un mic dispozitiv prin intermediul căruia el este monitorizat de către stăpân, care primește toate informațiile necesare în acest sens (Dormehl, 2014).

Din nevoia de reducere a consumului exagerat de apă, a apărut o soluție care implică dispozitive inteligente care măsura la fiecare moment consumul de apă. De asemenea, sunt implicați declanșatori în urma cărora proprietarul să fie anunțat atunci când sunt depășite anumite praguri.

Câteva alternative pentru automatizarea locuințelor ce aparțin de sfera proiectelor “open source” sunt OpenHAB și Home Assistant - o versiune mai nouă a primei soluții (Dormehl, 2014). Acestea au la bază aceleași principii și pe lângă faptul că le oferă utiizatorilor posibilitatea să creeze evenimente care pot fi activate individual, aduc un plus de valoare prin faptul că utilizatorii pot creea declanșatori în funcție de anumite stări la care soluția să se adapteze(de exemplu apusul și răsăritul soarelui). În topul liste de avantaje a primei alternative, OpenHAB este faptul că această soluție vine cu facilitatea de a funcționa și offline, iar acest lucru nu presupune fraptul că utilizatorul este nevoit să încarce datele ce provin de la această aplicație pe cloud. Dezvoltatorul acestei soluții spune că nu este vorba de Internet of Things, ci de Intranet of Things, adică deși toate dispozitivele ar trebui să interacționeze între ele, datele nu părăsesc casa utilizatorului.

B. Soluții destinate vânzării

Produsele existente pe piață sunt oferite de numeroase companii competente în domeniul automatizării locuințelor, iar serviciile oferite de aceștia variază în termeni de oferte și costuri de instalare. În vederea alegerii unei variante cât mai bune, ar trebui să se aibă în vedere câteva aspecte importante care țin de experiența în domeniul tehnic, de serviciul pentru clienți de care dispune compania respectivă, dar nu în ultimul rând și de garanția oferită pentru produs.

Analizând mai multe recenzii, dar și ceea ce oferă companiile am realizat un top al celor mai atractive soluții care se pot plia pe necesităților clienților.

Vivint, are facilități asemănătoare cu majoritatea altor sisteme automatizate pentru locuințe, dar se deosebește prin considerația pe care o primește în materie de sisteme de securitate și prin faptul că oferă și un pachet pentru energia solară (Costill, 2013) (Top Consumer Reviews, 2015).

SmartThings inovează piața prin faptul că oferă facilități de la transimiterea de notificări utilizatorului în legătură cu evenimente precum deschiderea bruscă a unei ferestre sau a unei uși, până la monitorizarea cheilor de la mașină (Costill, 2013).

Control 4, ce s-a autoproclamat o companie dedicată stilului de viață, oferă un sistem automatizat pentru locuințe de înaltă tehnologie, care este simplu de utilizat și are faciltăți flexibile. Mulți oameni preferă această companie datorită componentelor care nu necesită fire penru instalare. Această companie este recoandată și din prisma unui buget decent (Top Consumer Reviews, 2015).

## Descrierea și analiza soluției

Sistemele automatizate pentru locuințe întalnesc patru provocări importante, și anume costuri ridicate, inflexibilitate, gestiune la un nivel scăzut și dificultăți în ceea ce privește securitatea. Obiectivele pricipale ale acestei lucrări sunt de a crea și de a implementa un sistem automatizat pentru locuință care să fie ieftin și capabil să controleze și să automatizeze o parte din procese, printr-o interfață web ușor de gestionat care să permită ca sistemul să ruleze continuu și să se mențină în parametrii. De asemenea, soluția propusă vine cu o flexibilitate mare prin folosirea tehnologiei fără fir ce interconectează modulele distribuite la serverul locuinței automatizate. Acest fapt va duce la scăderea costurilor de implementare și va crește abilitatea de îmbunătățire și reconfigurare a sistemului.

Sistemul va folosi sistemul LAN de conexiune la rețea, fără fir pentru a realiza comunicarea dintre modulele hardware distribuite și server, iar comunicarea dintre useri si server se va realiza prin intermediul protocoalelor.

În comparație cu soluțiile existente pe piața fie că sunt “open source”, fie că sunt sisteme dezoltate de către companii, proiectul pe care îl dezvolt încearcă să îmbine managemenul de energie, costuri reduse, dar și o interfață placută și ușor de utilizat.

* Reducerea costurilor de instalare se datorează prin folosirea a cât mai multe componente care pot funcționa fără fir. Pentru componentele care folosesc fire pentru instalare, este nevoie de cabluri și alte materiale adiționale, dar și intervenția unui om specializat în acest domeniu.
* Ușor de implementat, instalat și poate fi folosit pe o arie mai largă. Punctele wireless pot fi montate aproape oriunde, chiar și în locuire greu accesibile pentru componentele conectate cu fire. Tehnologia wireless aduce ca beneficiu lărgirea zonei de acoperire.
* Scalabilitatea sistemului și ușurința de extindere.

Multe dintre soluțiile existente pe piață care deși la prima vedere par flexibile, în realitate constrâng utilizatorii și îi forțează să își adapteze sistemul în funcție de opțiunile pe care le oferă reespectiva companie. Acest fapt, restriționează destul de tare clienții și îi obligă într-un mod tacit să rămâna fideli companiei, așadar orice îmbunătățire a sistemului va depinde de compania care furnizează serviciile de automatizare a locuinței. Restricțiile care apar țin atât de compatibilitatea componentelor sistemului, cât și de o rețea bazată pe conexiunea prin cablu.

Implementarea unei rețele wireless este avantajoasă în primul rând în prisma faptului că ar putea să apară cerințe/specificații noi, sau acestea ar putea să se modifice, iar aceste schimbări vor atrage de la sine extinderea rețelei. În contrast cu instalațiile bazate pe conexiunea prin cablu, nodurile nou adăugate nu necesită cablu în plus, fapt ce ar face extensia trivială.

* Beneficii estetice. Nu doar că plasarea nodurilor este simplă și ca aria de acoperire a sistemului este mult mai mare, dar este vorba și de valențe estetice. Câteva exemple concludente pot fi reprezentate de construcțiile care au în componența arhitecturală doar sticlă, dar și clădirile istorice al căror design sau ale căror motive conservatoare nu permit instalarea unor cabluri adiționale.
* Integrarea dispozitivelor mobile. Asocierea de dispozitive mobbile precum PDA-uri sau smarphone-uri la un sistem automat este posibil oriunde și oricând atâta timp cât nu mai este o problemă prezența efectivă a dispozitivului, existând posibilitatea de control și de la distanță.
* Interfață “user friendly”. Posibiitatea ca utilizatorul să beneficieze de o interfață simplă, prin care să poate coordona un sistem complex se poate realize direct de pe PC sau de pe un dispozitiv mobil, accesând cu ușurință o interfață bazată pe servicii web, local sau la distanță.
* Securitatea și autentificarea.

Doar un utilizator autentificat se poate conecta la sistem – local sau la distanță – pentru a gestiona, controla și a monitoriza. În cazul în care sistemul detectează intruși, acesta trebuie imediat să transmită o notificare către utilizator prin care să-l atenționeze de acest incident și să blocheze posibilitatea de logare pentru o perioadă.

Acest aspect se mai poate adresa și utilizatorilor care nu doresc să folosească chei, ci mai degrabă preferă sa deschidă ușa conectându-se la sistem prin intermediul unui user și a unei parole. Astfel, pot evita problema pierderii cheilor și totodată îi pot ține pe cei mici departe de locurile în care aceștia nu au acces, de exemplu, dulapul cu medicamente, obiecte ascuțite etc.

* Economie de resurse.

Aceste resurse nu se referă doar la cablurile necesare instalării sistemului automatizat. Scopul acestei soluții este să ofere facilități, dar să reducă și costurile, consumul de curent, sau de căldură. Așadar utilizatorul poate avea acces la managementul resurselor și poate decide când să pornească sistemul de încălzire și la ce temperatură, sau când ar trebui stinse/aprinse luminile în funcție de necesitate, de oră sau chiar în funcție de senzorii care pot detecta mișcarea.

Nu doar utilizatorul direct poate beneficia de această economie de resurse, ci și mediul, prin reducerea de consum a energiei electrice, sau a căldurii.

A.Caracteristici ale soluției propuse

Soluția propusă reprezintă automatizarea unei locuințe sub forma unui sistem distruibuit care cuprinde serverul, microcontrolerul și componentele hardware care reacționează la schimbări. Serverul reprezintă de fapt un PC sau un laptop care se comportă ca un sever web la care se pot conecta utilizatorii, iar acesta coumnică microcontrolerului comenzile primite.

La rândul său, sistemul poate fi accesat prin intermediul unui browser web de pe orice PC local din aceeași rețea LAN folosind IP-ul serverului, sau de la distanță folosind orice dispozitiv care se poate conecta la internet.

Principalele funcții ale serverului constau în a gestiona, a controla și a monitoriza componentele sistemului care acționează în funcție de evenimentele care îi declanșează, sau de comenzile pe care le primesc.

B. Reprezentarea soluției



# 3. Tehnologii/Metode utilizate

Sistemul informatic implementează o serie de tehnologii ce contribuie la buna desfășurare a aplicației de automatizare a unei locuințe, aplicaţie ce are scopul de a oferi utilizatorului un confort sporit prin faptul că oferă posibilitatea de monitorizare şi control. Prin tehnologiile folosite se reuşeşte gestiunea mult mai bună a resurselor din întreaga locuinţă, acest proces putând fi realizat inclusiv de la distanţă. Sistemul poate fi cu ușurință divizat in subsiteme, sau module – cum vor fi ulterior referite – care dețin caracteristici concret definite și prin intermediul cărora pot fi ușor de recunoscut.

## 3.1 Schița Arduino

Nucleul soluţiei este reprezentat de o placă Intel Galileo Gen 2, aceasta fiind prima care este bazată pe arhitectura Intel, construită astfel încât să existe o compatibilitate din punctul de vedere al pinilor, atât hardware cât şi software cu scuturile specific plăcii Arduino Uno. Intel Galileo permite implementarea de algoritmi în limbajul de programare Arduino în mediul software de dezvoltare Arduino.

În plus faţă de aceste caracteristici, placa Intel Galileo se diferenţiază de componentele din mediul Arduino prin faptul că deţine din producţie elemente care extind capacităţile native, şi anume: un port Ethernet de 100 Mb, un slot Micro-SD, un port USB gazdă şi unul de tip client. Comparativ cu alte plăci, care rămân constant fără memorie, implicit au probleme în a folosi mai multe librări în același timp, placa Galileo sipune de mult mai multă memorie decât acestea. Compatibilitaea microcontroller-ului Intel Galileo cu produsele Arduino poate constitui un avantaj având în vedere că, teoretic vorbind, soluţiile implementate pentru o placă Arduino sunt implicit funcţionale atunci când sunt implementate pe o placă Intel Galileo.

Bineînţeles, există şi puncte slabe, iar unul dintre cele mai semnificative este faptul că în momentul folosirii anumitor algoritmi şi librării create pentru produsele Arduino, acestea nu pot fi folosite şi la nivelul microcontroller-ului Intel Galileo din pricina arhitecturii diferite de care aceasta dispune. Având în vedere faptul că există o varietate de produse Arduino, fiecare dintre plăci are carcteristici specifice, iar diferenţele există atât la nivel hardware cât şi software, dar sunt categoric sesizabile în momentul implementării unor soluţii care sunt dezvoltate pentru un set de produse ce au anumite caracteristici comune.

Desigur, există mai multe alternative pentru a implementa soluţii care să ruleze pe Intel Galileo, dar analizând cu atenţie punctele forte şi pe cele slabe, am realizat că cea mai potrivită opţiune este folosirea platformei open-source furnizată de către Arduino. Această platformă se bazează pe o placă de microcontroller și pe un mediu de dezvoltare pentru scrierea de software pentru microcontroller. Arduino poate fi folosit pentru a dezvolta obiecte interactive , preluând intrări dintr-o varietate de comutatoare sau senzori și controlează o varietate de lumini , motoare , și alte componente fizice . Proiectele Arduino pot fi de sine stătătoare , sau pot comunica cu software-ul care rulează pe computer (Barret, 2013).

Având în vedere capcităţile de care dispune placa Intel Galileo şi de limitările impuse, conexiunea se realizează prin intermediul portului Ethernet la reţeaua locală, iar acest fapt permite comunicarea cu un server bazat pe conexiunea prin socket-uri. Faptul că microcontroller-ul folosit reprezintă un client care se conectează la un server cu care este într-o comunicare continuă, impune conectarea la rețeaua locală, rețea în care se găsește și serverul, acesta fiind reprezentat de un PC sau un laptop.

Schița, programul care este încărcat pentru a rula pe microcontroller, conține mereu aceleași două funcții setup() și loop(), iar includerea lor în program este necesară, chiar dacă nu sunt utilizate.fiecare dintre acestea având caracteristicile și utilitățile ei, dupa cum urmează (Barret, 2013):

* setup() – această funcție este apelată o singură dată atunci când schița rulează, aici se inițializează librării sau se setează anumite informații referitoare la pini.
* loop() – această funcție reprezintă nucleul multor schițe prin proprietatea sa de a fi apelată continuu.

Schița utilizată în aplicația de automatizare conține în implementarea funcției loop() algoritmi folosiți în preularea datelor de la senzori, în prelucrarea lor și mai apoi în transmiterea acestora către server (Barret, 2013).

## 3.2 Server socket

Serverul socket, o altă componentă a sistemului informatic dezvoltat, este realizat prin utilizarea limbajului de programare Java în mediul de dezvoltare Netbeans. Acesta are un rol foarte important, fiind un bun intermediar între microcontroller și mediul extern ( utilizatori care primesc notificări; utilizatori care pot controla resursele din locuință de la distanță).

A. Server socket

Din punctul de vedere al metodelor utilizate și al algoritmilor implementați, serverul cu conexiune prin socket-uri conține un fir de execuție în care este configurat serverul care așteaptă să primească date și să le prelucreze din nou. În urma prelucrării acestea devin informații, iar în funcție de caracteristicile lor se vor stabili topicurile la care vor fi subscrise în urma transmiterii către un server MQTT.

B. Client REST – Java

Un alt fir de execuție implementat în acest modul al aplicației are rolul de client care se conectează la un server care utilizează servicii de tip REST, acestea fiind susținute de către un server web, open-source dezvoltat de către Apache Software Foundation, și anume, Apache Tomcat. Clientul are acces la resurse prin intermediul metodelor GET și POST, fiind folosite în scopul de a vizualiza și a modifica. Acest fir de execuție conține un algoritm care folosește operația GET pentru a vizualiza lista de comenzi, urmând să fie verficate elementele listei conform unor reguli pentru a afla dacă există sau nu comenzi noi. Dacă acestea există, informațiile sunt preluate și prelucrate pentru a fi transmise către placa Galileo care va transmite comenzi către resursele din locuință. Elementele noi ale listei vor suferi modificări care să ateste faptul că respectivele comenzi au fost gestionate. Clientul folosește Jersey care permite implementarea de servicii JAX-RS și de către clienți (Oracle, 2013).

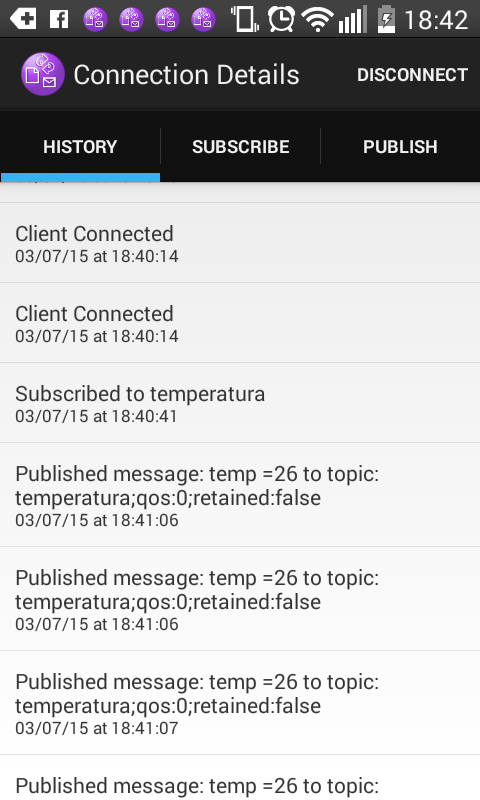
## 3.3 Servicii REST

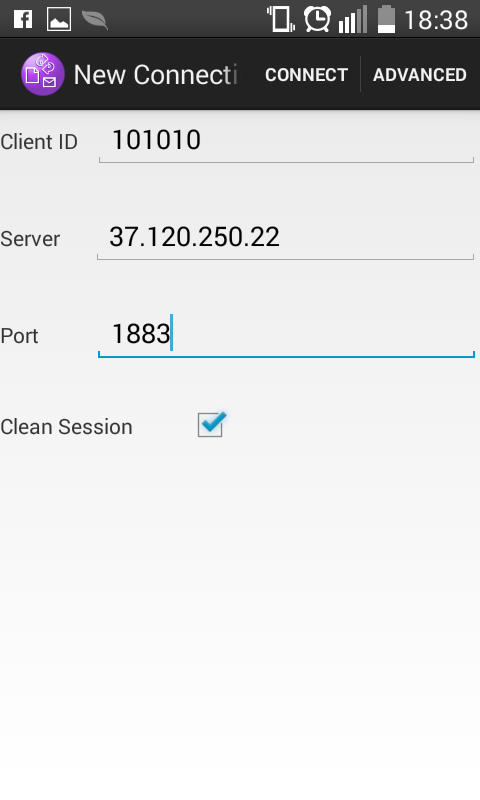
Serverul care implementează servicii REST este realizat prin utilizarea limbajului de programare Java, în cadrul mediului de dezvoltare Eclipse, folosind facilitățile pe care acesta le oferă. Pentru crearea serviciilor Web de tip REST, am folosit API-ul Java JAX-RS, definit în JSR 311 – Java Specification Request - si defineste in principal o serie de adnotari Java utilizabile în acest scop. Așadar, adontările au rolul de a realiza asocieri dintre metodele clasei și cererile HTTP, acestea generând clasele Java care prelucrează cererile HTTP primate. De reținut este faptul că JAX-RS reprezintă un API doar pentru furnizorii de servicii REST, nu și pentru clienții REST (Oracle, 2013).

## 3.4 Client MQTT

Publicarea datelor în cadrul unui anumit topic, cât și abonarea la un topic se realizează prin intermediul unui protocol de conectivitate destinat comunicări din sfera Internet of Things, acesta fiind implementat atât în limbajul de programare Java cât și Android. Pentru acest modul am folosit Eclipse Paho care se focusează pe implementări al căror scop este de a integra o gamă cât mai largă de modele de programare și de mesagerie. Implementările open-source de protocoale de mesagerie oferă soluții pentru aplicațiile în curs de dezvoltare din mediul Machine-to-Machine(M2M) și Internet of Things (IoT). Există constrângeri atât la nivel fizic, cât și la nivelul costurilor de conectivitate ale dispozitivelor, iar în acest sens, proiectul Paho are în vedere folosirea unor nivele efective de decuplare între dispositive și aplicații, având ca ținte păstrarea deschisă a piețelor și încurajarea unei evoluții rapide în rândul aplicațiilor Enterprise și Web (Paho).

Aplicația în cauză integrează în modulele sale atât publicarea unor mesaje la anumite topicuri specifice informațiilor ce se transmit, cât și abonarea la aceste topicuri care presupune implicit recepționarea mesajelor postate.

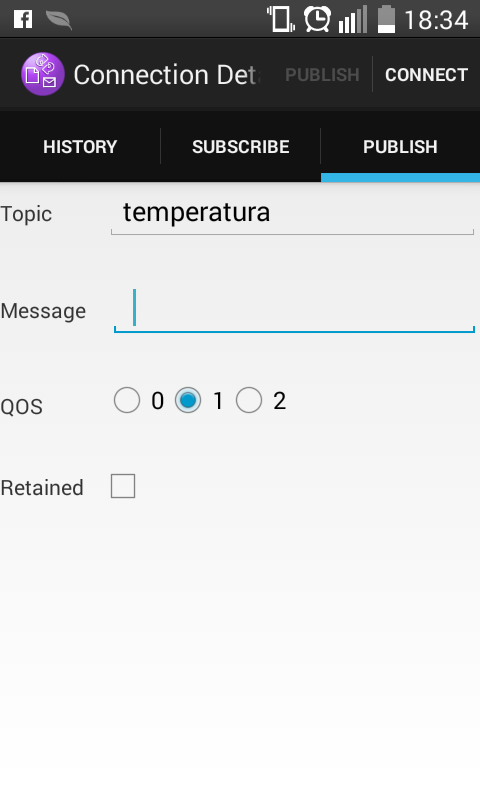




*Figura 1 - Interfață client MQTT*

A. Publicare

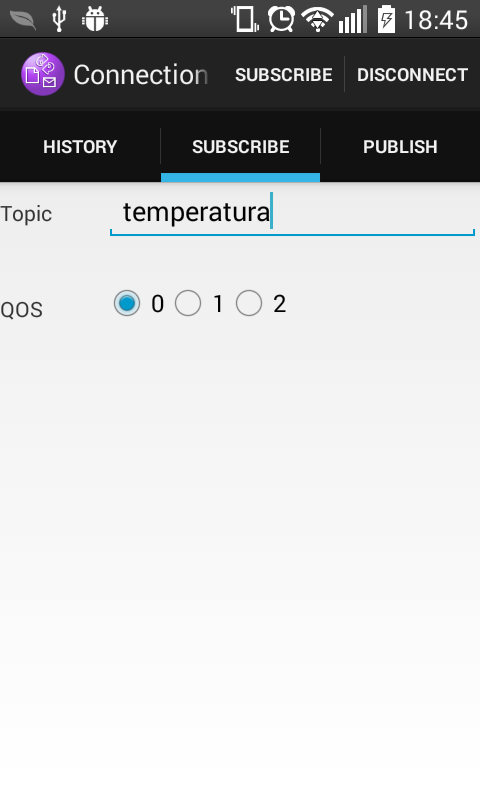
Clientul care este desemnat să publice mesajele este parte integrată a modulului de server socket despre care am detaliat anterior. Informațiile sunt transmise într-un flux continuu, imediat ce sunt procesate.



*Figura 2 - Interfață client MQTT - Publicare*

B.Abonare

În ceea ce privește clientul MQTT care este abonat la topicurile prestabilite, poate fi caracterizat ca un sistem de notificare al utilizatorului care permite monitorizarea ambientului din locuință. În funcție de cât de des sunt preluate informațiile de la senzori, utilizatorul este informat în legătură cu statusul utilităților care sunt supravegheate prin culegerea de probe de către senzorii instalați în locuință și care comunică prin fire de conexiune cu placa Galileo. Modulul care integrează acest client MQTT este unul dintre subsistemele care interacționează în mod direct cu utilizatorul prin intermediul unei aplicații Android pentru dispozitivele mobile care folosesc Android ca sistem de operare, fie prin intermediul unei aplicații desktop.



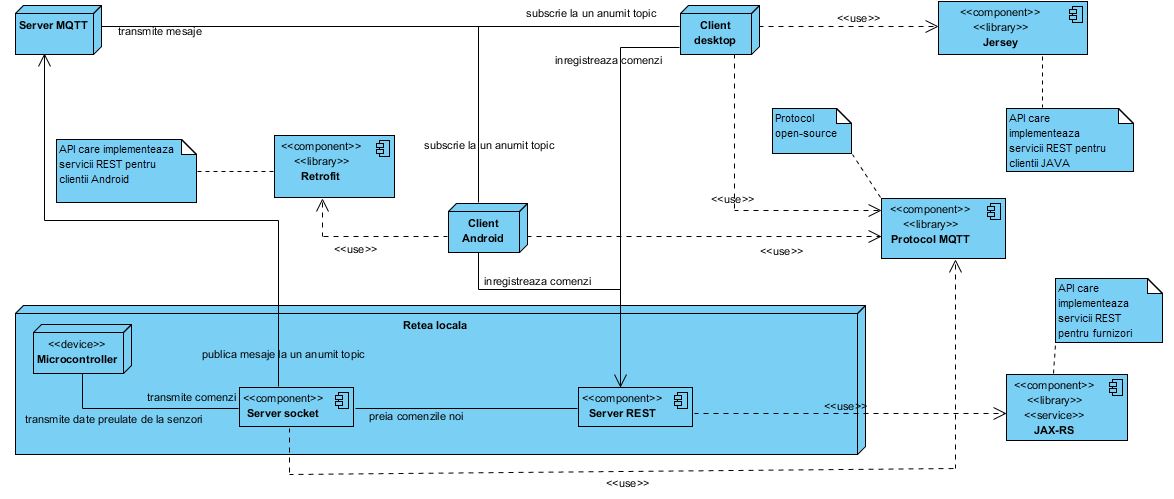
*Figura 3 - Interfață client MQTT - Abonare*

## 3.5 Client REST – Android

Acest modul implementează același algoritm folosit și de Clientul REST – Java, care vizualizează lista de comenzi, verifică elementele listei conform unor reguli pentru a decide dacă există sau nu comenzi noi. Dacă răspunsul este pozitiv, deci acestea există, informațiile sunt preluate și prelucrate pentru a fi transmise către placa Galileo care va transmite comenzi către resursele din locuință. Elementele noi ale listei vor suferi modificări care să ateste faptul că respectivele comenzi au fost gestionate. Clientul implementează Retorfit.

# 4. Arhitectura soluției

## 4.1 Diagrama de componente



*Figura 1 - Diagrama de componente*

Aplicația dezvoltată este alcăuită din mai multe module, așa cum este menționat și în capitolele anterioare, fiecare dintre acesta contribuind activ la buna desfășurare a procesului de automatizare a locuinței. Subsistemele integrate sunt într-o comunicare continuă, caracterizată printr-un transfer de date și de informații activ.

Subliniam faptul că aplicația are ca scopuri principale monitorizarea, controlul și gestiunea datelor și a informațiilor pe care aplicația le primește și pe care ulterior le prelucrează. Așadar aspectul referitor la monitorizarea locuinței automatizate se remarcă prin faptul că datele sunt colectate și transmise ulterior utilizatorului care va acționa în funcție de scenariul dat. Controlul are ca scop menținerea în parametrii normali a valorilor utilitaților automatizate din cadrul casei inteligente. Nu în ultimul rând, gestiunea datelor și a informațiilor vizează toate procesele la care sunt supuse datele, respectiv informațiile pentru a genera valoare în cadrul sistemului informatic.

A. Monitorizarea

Microcontroller-ul colectează datele care provin de la senzorii plasați în cadrul locuinței, fluxul de date fiind continuu. În circumnstanța în care datele care ajung de la senzori la placa Galileo prezintă abateri de la limitele normale, vor fi transmise semnale vizuale în interiorul locuinței – de exemplu aprinderea unui led de culoare roșie sau galbenă, care să sugereze faptul că există o anomalie. Aceste date sunt transmise către un server care le prelucrează și astfel sunt generate informațiile care urmează să fie publicate la câte un topic specific, în funcție de senzorul de la care acestea provin. Informațiile publicate vor fi transmise clienților care sunt abonați la topicurile respective și vor fi implicit notificați în legătură cu aceste mesaje.

Se poate remarca faptul că monitorizarea locuinței nu este accesibilă doar din incinta acesteia, ci și de la distanță prin faptul că utilizatorii se abonează pentru a fi notificați.

B. Controlul

Automatizarea locuinței presupune și posibilitatea de control a resurselor și a utilităților de care aceasta dispune, mai ales din perspectiva că o dată notificat, utilizatorul decide la ce măsuri este necesar să recurgă pentru a realiza ajustările necesare, astfel încât valorile să revină în parametrii normali. Totodată, poate fi vorba și de controlul de la distanță, care urmărește ajustarea valorilor consumatorilor în funcție de preferințele unui utilizator. Nu în ultimul rând sistemul informatic poate implementa un sistem de autocontrol, respectiv autoreglare în funcție de diversi factori care pot fi manipulați de către aplicație, fără a fi necesară intervenția unui agent extern, utilizatorul uman.

C. Gestiunea datelor și a informațiilor

Această caracteristică se referă în primul rând la transferul de date și informații dintre modulele aplicației, și mai apoi la folosirea acestora în scopul îmbunătățirii ambientului locuinței. În al doilea rând este vorba și de prelucrarea, respectiv stocarea datelor, a informațiilor implicate in fluxul de activități. Nu în ultimul rând, gestiunea se referă și la procesul de adaptare al consumatorilor la noile condiții.

## 4.1 Diagrama de flux

Sistemul informatic poate fi corelat și cu ceea ce numim logica afacerii – business logic – , acest aspect fiind unul dintre elementele definitorii pentru o afacere aflată în process de modelare și de automatizare. În logica afacerii, amintită anterior în contextul dezvoltării sistemului informatic, sunt cuprinse atât regulile afacerii, cât și fluxul de lucru din cadrul afacerii prin intermediul căruia este descris modul de transfer al datelor în interiorul aplicației de la un modul la altul.

Exemplificarea acestor fluxuri din interiorul aplicației se poate face prin intermediul diagramei de activitate, care are scopul de a evidenția fluxurile de lucru prin accentuarea acțiunilor întreprinse, dar și a rezultatelor acestora. Așadar, rolul diagramei de activitate este de a descrie fluxul de lucru dintr-un punctul de pornire până într-un punct de finalizare, respectiv de a detalia căile de decizie care pot apărea într-o activitate.

A.Publicare date folosind clientul MQTT

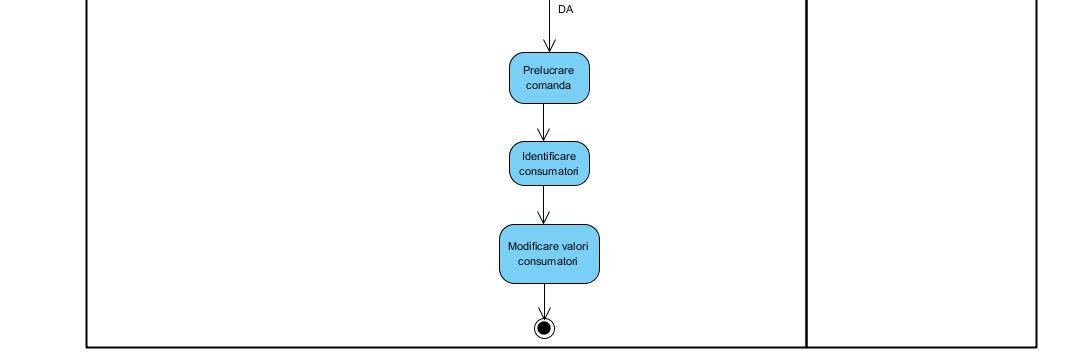
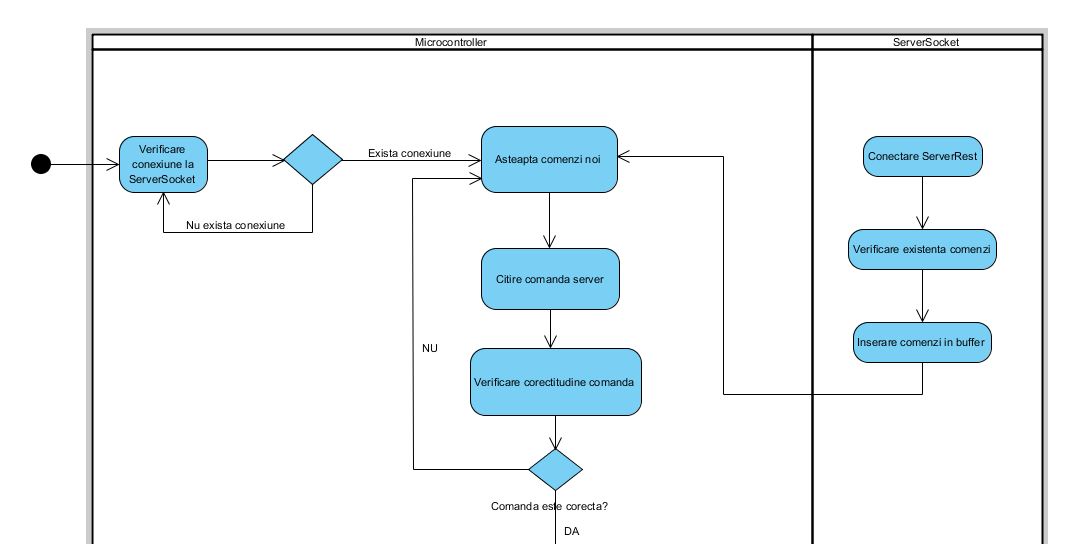


*Figura 2 - Diagrama de activitate: Publicarea datelor folosind clientul MQTT*

În *Figura 1* este reprezentat fluxul de lucru care descrie activitatea de publicare a datelor – preluate de la senzori folosind microcontroller-ul – prin utlizarea clientului MQTT. Din punctul de start pornește activitatea de conectare a plăcii Galileo, la finalul căreia apare un nod de bifurcație care generează alte două activități care se desfășoară simultan, și anume citirea datelor preluate de la senzori și conectarea la server socket. Fluxurile generate de aceste două activități converg către un nod de joncțiune înainte ca procesarea să continue, rezultând un singur flux care transmite controlul către activitatea de verificare a existenței unei conexiuni cu serverul. Urmează un nod decizional în care intră un flux și ies două, și anume unul care neagă existența conexiuni și revine la activitatea inițială, iar cel de-al doilea flux confirmă existența unei conexiuni cu server-ul și face referire la o nouă activitate care are rolul de a transmite date către senzor. Ceea ce a fost descris până acum reprezintă fluxul de lucru pe care îl realizează microcontrollerul.

Activitățile succesoare sunt executate de către server și cum sunt datele transmise începând cu alegerea topicului, urmând conectarea la server-ul MQTT și finalizând întreg procesul cu activitatea de publicare a datelor la topicul ales.

B. Modificare valori consumatori



*Figura 3 - Diagrama de activitate: modificarea valorilor consumatorilor*

*Figura 2* prezintă procesul de lucru care presupune modificarea valorilor consumatorilor din locuință, adică ajustarea nivelului de folosire al utilităților la un nivel optim în funcție de comenzile pe care sistemul informatic le primește de la utilizator. Analizând această diagramă observăm faptul că există un flux principal în care executantul activităților este microcontroller-ul Intel Galileo. În vederea argumentării acestei observații, doresc să evidențiez faptul că nodul de start dincare este invocată prima activitate, cât și nodul de finalizare în care intră fluxul ultimei activități se află în raza de procesare a plăcii Galileo. Fluxul secundar este reprezentat de activitățile desfășurate de server socket și se îmbină la finalulproceselor executate de către server cu fluxul principal.

Din prima activitate, verificarea existenței conexiunii cu server-ul, executată de către microcontroller iese un singur flux de control care intră într-un nod decizional din care vor rezulta două fluxuri, unul care neagă premisa și care se întoarce la activitatea inițială, iar cel de-al doilea susține premisa formulată în prima activitate. Fluxul acesta trimite către o altă activitate, care pune modulul in starea de așteptare a unei noi comenzi de la server. În paralel server-ul apelează serviciile REST și verifică existența unor comenzi noi pe care le va salva într-un buffer și le va transmite microcontroller-ului spre procesare. Revenind la procesul principal, microcontroller-ul citește comenzile primite în prealabil cercetează corectitudinea lor, iar în cazul în care acestea nu sunt corecte, placa Galileo revine în starea de așteptare, altfel prelucrează comanda, realizează identificarea consumatorilor, și apoi inițiază și execută modificarea valorilor astfel încăt să fie respectate comenzile primite.

# 5. Implementarea solutiei

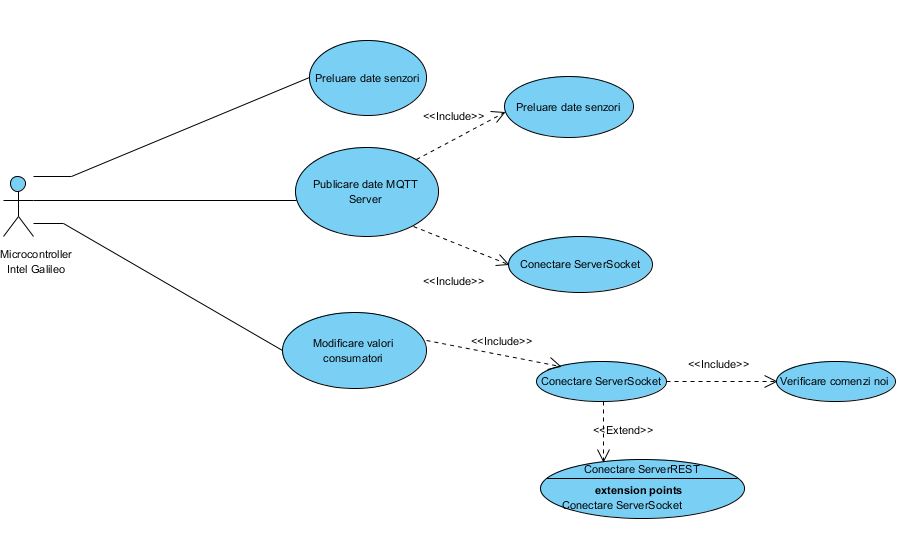
Soluția propusă reprezintă automatizarea unei locuințe sub forma unui sistem distruibuit care cuprinde serverul, microcontrolerul și componentele hardware care reacționează la schimbări. Serverul reprezintă de fapt un PC sau un laptop care se comportă ca un sever web la care se pot conecta utilizatorii, iar acesta coumnică microcontrolerului comenzile primite.

La rândul său, sistemul poate fi accesat prin intermediul unui browser web de pe orice PC local din aceeași rețea LAN folosind IP-ul serverului, sau de la distanță folosind orice dispozitiv care se poate conecta la internet.

Principalele funcții ale serverului constau în a gestioona, a controla și a monitoriza componentele sistemului care acționează în funcție de evenimentele care îi declanșează, sau de comenzile pe care le primesc.

În cadrul acestui capitol vor fi detaliate cerințele funcționale pe care sistemul trebuie să le îndeplinească astfel încât obiectivele anterior menționate să fie atinse, în acest sens urmează să fie prezentate diagramele cazurilor de utilizare completate de explicațiile aferente. Aceste reprezentări grafice sunt definite ca proiecții ale funcționalităților pe care sistemul trebuie să le îndeplinească sistemul informatic în faza finală a acestuia și sunt alcătuite din actori și cazuri de utilizare, respectiv din relațiile dintre acestea. Pentru fiecare dintre cazurile de utilizare ce caracterizează sistemul informatic, va fi prezentat câte un model al cerințelor care presupune modelul realizat de diagramele cazurilor de utilizare alături de documentele de descriere succintă a fiecărui caz de utilizare determinat. De remarcat este faptul că scopul cazurilor de utilizare este orientat către ceea ce sistemul trebuie să facă și nu cum. Așadar, se poate afirma faptul ca acestestea sunt neutre din punct de vedere tehnologic, existând posibilitatea să fie utilizate în orice proces sau arhitectură de aplicație.

## 5.1 Diagrama generală a cazurilor de utilizare – perspectivă MICROCONTROLLER



*Figura 4 - Diagrama generală a cazurilor de utilizare – perspectivă MICROCONTROLLER*

*Figura 3* are scopul de a evidenția cazurile de utilizare specifice microcontroller-ului Intel Galileo, adică este accentuată modelarea dialogurilor între actor, în cazul de față placa Galileo, și sistemul informatic. Din această perspectivă sistemul poate fi folosit în trei moduri care diferă între ele prin secvențele de tranzacții cuprinse:

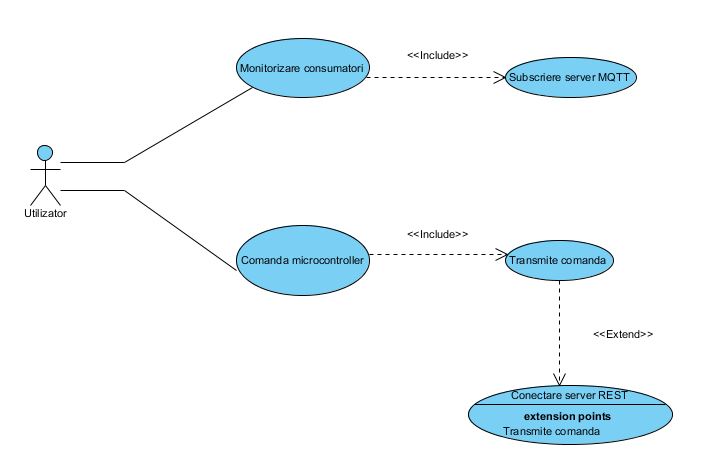
* Preluare date senzori
* Publicare date folosind clientul MQTT
* Modificare valori consumatori

Tabelul de mai jos reprezintă documentul de descriere al cazurilor de utilizare din prespectiva microcontroller-ului.

*Tabel 1 - Documentul de descriere al cazurilor de utilizare - Microcontroller*

|  |  |
| --- | --- |
| Element al cazului de utilizare | Descriere |
| Cod | CU01 |
| Stare | Schiță |
| Scop | Publicare date citite |
| Nume | Publicarea datelor |
| Actor principal | Microcontroller |
| Descriere | Presupune publicarea datelor în cadrul unui topic specific, date pe care microcontrollerul le preia de la senzorii ce compun sistemul informatic. |
| Precondiții | Microncontrollerul este conectat la rețeaua de internet și are acces la serverul cu care acesta comunică.  Senzorii sunt conectați la microconttroller și sunt funcționali. |
| Postcondiții | - |
| Declanșator | - |
| Fluz de bază | 1. Microcontrollerul este conectat.  2. Preia date de la senzori și încearcă să se conecteze la ServerSocket  3. Verifică dacă există conexiunea cu serverul, dacă da, transmite datele preluate către server, care le va publica la topicul specific. |
| Fluxuri alternative | - |
| Relații | - |
| Frecvența utilizării | Constant |
| Reguli ale afacerii | - |

## 5.2 Diagrama generală a cazurilor de utilizare - perspectivă UTILIZATOR



*Figura 5 - Diagrama generală a cazurilor de utilizare - perspectivă UTILIZATOR*

*Figura 4* are scopul de a evidenția cazurile de utilizare specifice utilizatorului, adică este accentuată modelarea dialogurilor între actor, în cazul de față placa Galileo, și sistemul informatic. Din această perspectivă sistemul poate fi folosit în două moduri care diferă între ele prin secvențele de tranzacții cuprinse:

* Monitorizare consumatori
* Comanda microcontroller

Tabelul de mai jos reprezintă documentul de descriere al cazurilor de utilizare din prespectiva utilizatorului.

Tabel - *Documentul de descriere al cazurilor de utilizare - Utilizator*

|  |  |
| --- | --- |
| Element al cazului de utilizare | Descriere |
| Cod | CU02 |
| Stare | Schiță |
| Scop | Modificarea unor valori |
| Nume | Modificare valori consumatori |
| Actor principal | Microcontroller |
| Descriere | Presupune automatizarea unei locuințe prin modificarea valorilor consumatorilor în funcție de comenzile transmise de la distanță. |
| Precondiții | Microncontrollerul este conectat la rețeaua de internet și are acces la serverul cu care acesta comunică.  Consumatorii sunt conectați la microcontroller și sunt funcționali. |
| Postcondiții | - |
| Declanșator | - |
| Fluz de bază | 1. Microcontrollerul verifică existența unei conexiuni stabilită cu ServerSocket-ul, în caz favorabil așteaptă comenzi de la acesta.  2. La primirea comenzilor, acestea sunt verificate și apoi prelucrate.  3. Se identifică consumatorii.  4. Microcontrollerul trasnimite catre consumatori pentru a ajusta statusul prezent al acestora. |
| Fluxuri alternative | - |
| Relații | - |
| Frecvența utilizării | Din oră în oră |
| Reguli ale afacerii | - |

# 6. Concluzii

Rezultatul actual este în conformitate cu soluția propusă, totuși au apărut anumite deviații de la traseul inițial din pricina restricțiilor de nivel hardware și software apărute. Alături de acești factori ce dețin o conotație ușor negativă, au contribuit la apariția aplicației actuale și factori cu valențe pozitive care au constat în descoperirea unor tehnologii mult mai avantajoase decât cele propuse inițial.

Deși unul dintre avantajele sistemului era reprezentat de către eliminarea firelor pentru conexiunea la internet a microcontroller-ului Galileo, respectiv a firelor care conectau senzorii la placă, am renunțat la această idee din mai multe motive. Prin implementarea unui sistem de automatizare al locuinței care să coumunice wireless, datele și informațiile care circulau în rețea ar fi fost mult mai ușor de accesat decât în cazul unei conexiuni prin fire, care nu ar fi expus către mediul extern informații private care ar putea fi accesate de oricine este interesat de acest tip de activitate. Prin această decizie de a folosi conexiunea prin cablu în detrimentul unei rețele wireless, am evitat ca aplicația să devină o țintă sigură pentru posibilii infractori.

Acest subiect merită o atenție mai mare deoarece folosirea unei rețele fără fir necesită folosirea unui microcontroller care să suporte librării pentru acest gen de conexiune, sau adaptarea plăcii prin adăugarea unor scuturi care să îmbunătățească radical caracteristicile microcontroller-ului folosit. De asemenea este nevoie de folosirea unor senzor wireless compatibili și care să permită implementarea cu ușurință a algoritmilor necesari, pentru ca datele să poată fi transmise cu ușurință, fără a se pierde sau fără a fi degradate în urma acestui transfer. Bineînțeles, accentul este pus în primul rând pe securitatea sistemului care poate fi ținta unor atacuri informatice, așa cum am menționat și anterior.

Aspectele prezentate în paragrafele precedente pot reprezenta o nouă pistă de cercetare, respectiv de dezvoltare în vederea îmbunătățirii soluției actuale. În plus față de ceea ce am evidențiat în prealabil, ar putea fi conturate mai multe direcții de dezvoltare a aplicației, și anume folosirea unui sistem de autentificare care să implementeze algoritmi de criptare a datelor.

Pe parcursul implementării soluției propuse am descoperit faptul că automatizarea locuințelor reprezintă o afacere încă prematură, dar care se dezvoltă rapid, în primul rând datorită faptului că tehnologiile existente la ora actuală le oferă dezvoltatorilor suficiente resurse în scopul de a crea soluții de acest fel. De asemenea, am putut observa și faptul că se preferă dezvoltarea de soluții în detrimentul cumpărării lor, iar acest lucru se datorează mai multor factori. În primul rând este vorba despre costuri, care evident vor fi mai mari în contextul achiziției sistemului, decât în cazul în care un iubitor de tehnologie preferă să își creeze propria soluție conform nevoilor. Așadar, se poate remarca faptul că raportul calitate preț nu este tocmai satisfăcător din punctul de vedere al facilităților oferite de către un produs plasat pe piață spre vânzare, iar acest lucru se datorează faptului că se oferă un spectru restrâns de opțiuni de care clientul poate dispune. De cele mai multe ori, serviciile oferite de către companiile care dezvoltă aplicații în domeniul automatizărilor, sunt în mare parte asemănătoare.

Bibliografie

Albanesius, C. (2014). *pcmag*. Retrieved 2015, from http://www.pcmag.com/

Barret, S. F. (2013). *Arduino Microcontroller Processing for Everyone!Third Edition.*

Costill, A. (2013, noiembrie). *Search Engine Journal*. Retrieved marti 2015, from http://www.searchenginejournal.com/top-6-home-automation-systems-apple-releases-version/76387/

Dormehl, L. (2014, decembrie). *FastCompany*. Retrieved martie 2015, from http://www.fastcompany.com/3038442/elasticity/5-open-source-home-automation-projects-we-love

Oracle. (2013). The Java EE 6 Tutorial.

*Paho*. (n.d.). Retrieved aprilie 2015, from http://www.eclipse.org/paho/

*Top Consumer Reviews*. (2015). Retrieved martie 2015, from http://www.topconsumerreviews.com/home-automation/

*WhatIs*. (2014, iunie). Retrieved 2015, from http://whatis.techtarget.com/definition/Internet-of-Things

Anexa 1 – Codul sursa al rutinei ...