**Raspberry Pi si Arduino in implementarea unui modul de**

**Casa Inteligenta**

**Schita de proiectare a modulului**

Modulul va avea doua componente principale si un numar variabil de componente auxiliare. Componentele principale sunt reprezentate de o placa de dezvoltare Raspberry Pi v3 B pe care ruleaza sistemul de operare RaspbianOS si o placa de dezvoltare de micro comanda Arduino UNO r3. Componentele auxiliare sunt reprezentate de un set de senzori si de module de interactionare cu Arduino UNO r3.

Setul de senzori cuprinde urmatorii senzori:

* Sensor de temperatura
* Sensor de umiditate
* Sensor de detectare a intensitatii luminii
* Sensor de miscare
* Sensor ultrasonic pentru detectarea distantei pana la anumite obiecte
* Microfon
* Sensor MQ-2 pentru detectarea gazelor

Modulele de interactionare sunt urmatoarele:

* Butoane activate prin apasare fizica
* Buzzer pentru alarme
* LED-uri pentru indicarea setarilor activate/dezactivate
* LED-uri pentru alarme

**Roluri componente si strategie de comunicare**

Placa de dezvoltare Raspberry Pi v3 B va avea rolul de computer central al modulului de casa inteligenta, iar de acum in documentatie se va referi asupra lui sub denumirea de computer central. Placa de dezvoltare Arduino UNO r3 va avea rolul de colector de date din partea fiecarui sensor conectat la acesta, iar de acum in documentatie se va referi asupra lui sub denumirea de microcontroller.

Asadar, microcontrollerul va avea ca sarcina sa mentina toti senzorii activi si sa colecteze datele returnate de fiecare in parte, date care vor fi trimise mai departe catre computerul central prin intermediul serialei. Seriala va avea rol de alimentare a micro controllerului, dar si de cale de comunicare intre acesta si computerul central, asadar va fi o conexiune directa, deci nu are sens sa ne pronuntam asupra securitatii transferului de date.

De asemenea, un alt rol al microcontrolerului va consta intr-un management de interactionare a datelor cu indicatorii si modulele de contact conectate la acesta. Mai precis, microcontrolerul se va folosi de led-uri, module de alarma, microfoane, difuzoare pentru a facilita interactionarea acestuia cu utilizatorul. Spre exemplu se vor aprinde led-uri de culoare rosie si se vor activa modulele de alarmare in cazul in care senzorul de gaz observa ca s-a depasit limita maxima admisa.

Programul software care va rula pe microcontroller va avea nevoie de o instalare initiala, dupa care nu va mai fi nevoie de mentenanta asupra acestuia din partea utilizatorului. Microcontrollerul va salva executabilul intr-o memorie locala si o va rula incontinuu, deci utilizatorul trebuie doar sa se asigure ca microcontrolerul este conectat la sursa de alimentare si anumite cablul serialei conectat la computerul central.

Computerul central va avea rolul de a citi toate aceste date de pe seriala si sa le incarce intr-o baza de date stocata local pe acesta. Mai departe, acesta va afisa datele intr-o interfata destinata utilizatorului si va da optiune acestuia sa faca anumite configurari in ceea ce privesc componentele auxiliare ale modulului. Se va incerca o implementare care sa permita o varietate de modalitati de configurare a modulului.

Vor exista 3 modalitati de interactionare a utilizatorului cu modulul:

1. O aplicatie care va rula direct de pe computerul central si va avea ca si mediu de interactionare cu utilizatorul un display capacitive touchscreen, iar atunci datele colectate vor beneficia de un nivel ridicat de securitate.
2. Un server HTTPS in care va fi incarcata o pagina web pentru reprezentarea interfetei de control a modulului. Serverul va putea fi accesat doar din interiorul retelei locale prin autentificarea utilizatorului cu un cont creat in momentul instalarii modulului. Pentru accesarea din exteriorul retelei, utilizatorul se poate folosi de o conexiune VPN.
3. O aplicatie Android, asemanatoare in design si mod de dezvoltare precum modalitatea numarul 2. S-a ales si aceasta varianta pentru a oferi confort utilizatorului.

**Tehnologii, implementari si medii de programare folosite**

Vor fi prezentate cateva dintre tehnologiile la care s-a recurs in dezvoltarea modulului precum si o descriere a utilizarii acestora in cadrul modulului. Se va incerca o enuntare a acestora intr-o ordine cat mai fireasca posibil.

Ca si inceput ne vom raporta la mediile de programare prezente in cadrul modulului si anume: Sistemul de operare RaspbianOS al computerului de control, Arduino IDE pentru implementarea si configurarea microcontrollerului, Android Studio pentru dezvoltarea aplicatiei de comunicare cu utilizatorul, Qt pentru implementarea interfetei de control de pe computerul central.

RaspbianOS in combinatie cu Raspberry Pi v3 (computerul central) ne ofera suportul si resursele mai mult decat suficiente pentru mentinerea si de asemenea imbunatatirea modulului. Acesta este capabil sa intretina baze de date, sa mentina servere HTTPS active fara a intampina dificultati si sa ruleze un numar relativ mare de aplicatii in acelasi timp.

Arduino IDE reprezinta “puntea” de transfer din cadrul modulului si o unealta pe atat de vitala, pe atat de sigura. Dezvoltarea programului software de colectare si transmitere a datelor de catremicrocontroller a avut loc folosind Arduino IDE. Deasemenea si testarea functionalitatii programului a avut loc tot datorita acestui IDE.

Tehnologiile si implementarile vor fi prezentate impreuna deoarece se afla in stransa legatura in cadrul modulului.

Pentru stocarea datelor pe computerul central s-a recurs la libraria SQLite3 deoarece complexitatea datelor si nivelul modului de prelucrare a acestora nu sunt unele crescute, iar o platforma precum aceasta este o solutie optima pentru managementul memoriei, dar si a timpului de prelucrare a datelor.

Pentru implementarea interfetei de control prezente pe computerul central am ales crearea unei aplicatii in limbajul de programare C/C++, ajutandu-ne de capacitatile software-ului Qt pentru crearea designului visual al aplicatiei.

**Securitatea datelor**

Datele colectate de microcontroller si stocate in baza de date a computerului central sunt prelucrate direct din baza de date in cadrul aplicatiei ce ruleaza de pe computerul central, dar pentru celelalte doua modalitati de interactionare cu modulul, transferul datelor pe server se va face prin intermediul tehnologiei SFTP (secure file transferring protocol) care va face o criptare a datelor inainte de a le incarca, ceea ce va spori confidentialitatea datelor chiar si in cazuri extreme pentru a evita penetrarile la nivel hardware a acestuia.

Porturile de acces ale computerului central vor fi securizate si accesate doar dupa introducerea datelor de inregistrare ale utilizatorului (cont si parola). Desigur in cazul unui atac la nivel hardware, cea mai mare responsabilitate o va avea utilizatorul insusi.

**Big Data Challenge?**

Apare intrebarea fireasca: Ce facem cu atatea date? Se vor aduna prea multe date? Simplu! Datele colectate de catre microcontroller nu reprezinta un impediment care sa conduca la o lipsa de memorie a modulului. Spre exemplu, pentru o zi intreaga, datele colectate vor putea fi stocate intr-un fisier .db de 100 KB, ceea ce inseamna ca datele aferente unui an intreg vor putea fi stocat in doar 35.6 MB, avand in vedere ca memoria de baza a modului este de 50 GB acest aspect nu reprezinta un motiv de ingrijorare.

Platforma de stocare a datelor personale (poze, fisiere) din interiorul modulului va avea limitarile mentionate in prezentarea initiala a acestuia pentru utilizator, asta inseamna ca nu vor exista dubii pentru utilizator sau pentru dezvoltatori in momentul epuizarii memoriei. Se va incerca o stocare a datelor cat mai eficienta posibil prin recurgerea, spre exemplu, la comprimari de fisiere, asta desigur cu accordul utilizatorului.