

Claudiu-Daniel MARINESCU (78546) - Smart Home

Autorul poate fi contactat la adresa: claudiu.marinescu95@stud.acs.upb.ro

Introducere

Proiectul își propune implementarea minimală a conceptului “Smart Home”.

Câteva funcționalități ale acestui proiect:

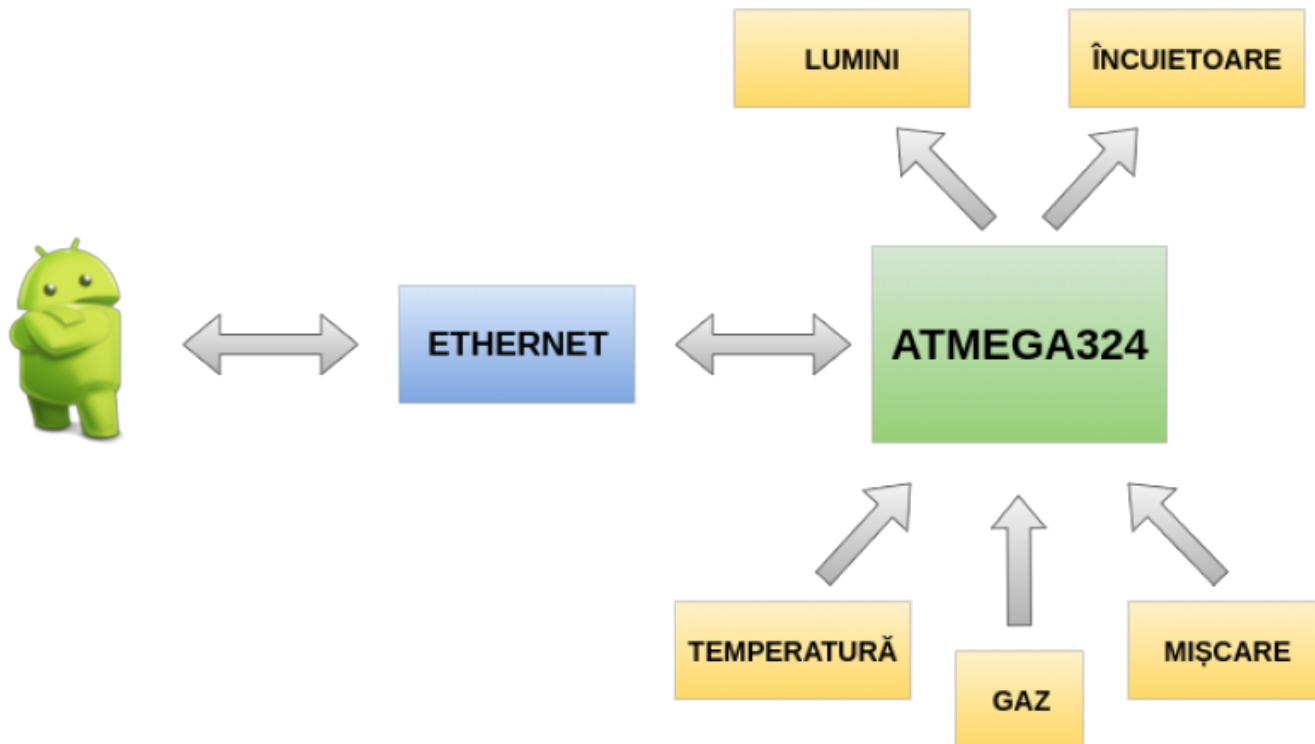
- monitorizarea temperaturii, scurgerilor de gaze, mișcărilor
- controlarea luminilor și a unei încuietori
- comunicarea prin Ethernet cu un dispozitiv Android

Consider că acest proiect este interesant și cu aplicație reală atât prin prisma faptului că proprietarul poate “comunica” cu locuința sa de la distanță prin internet, cât și prin controlarea, monitorizarea și centralizarea acestor date într-o singură aplicație Android. Interesul tot mai mare pentru concepte precum Smart Home, Smart City din ultimii ani m-a determinat să aleg această temă.

Descriere generală

Proiectul va funcționa astfel: cei 3 senzori (temperatură, gaze, mișcare) vor colecta date care vor fi trimise de microcontroller către aplicația Android prin modulul Ethernet la un anumit interval de timp. Utilizatorul va putea să trimită din aplicația Android comenzi precum deschiderea / închiderea unei încuietori, aprinderea / stingerea unei anumite lumini, comenzi ce vor fi preluate prin modulul Ethernet și apoi executate de către microcontroller.

Schema proiectului:

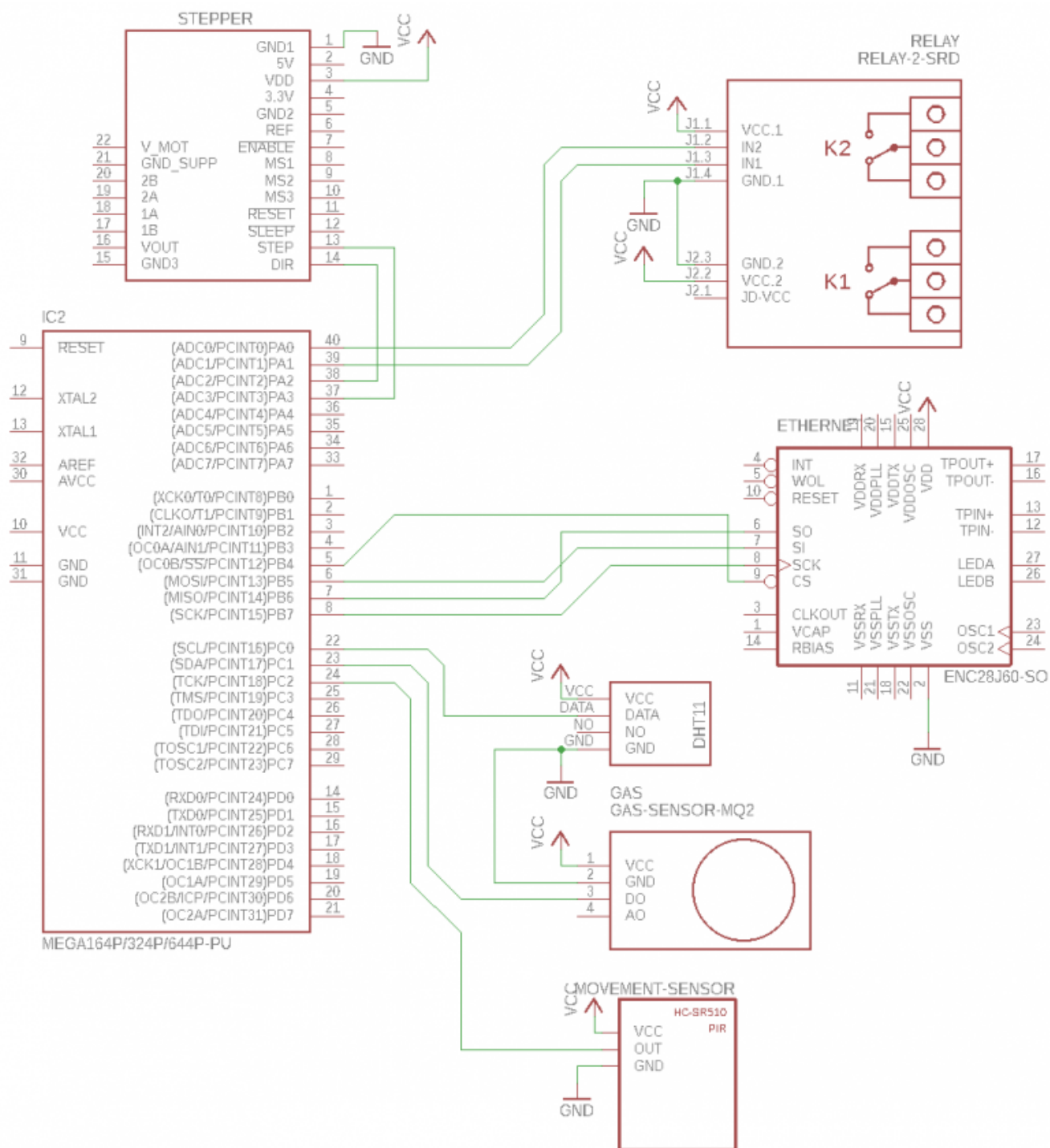


Hardware Design

Componente principale:

- ATmega 324
- Modul Ethernet
- Senzor temperatură
- Senzor gaz
- Senzor mișcare
- Motor încuietoare
- Releu pentru lumini

Schema electrică:



Software Design

Senzor de temperatura DHT11

Pentru a citi date de la senzorul de temperatura am folosit biblioteca dht disponibila online. Pentru aflarea temperaturii la un moment dat am apelat functia:

```
int8_t dht_gettemperature(int8_t *temperature)
```

Senzor de miscare PIR HC-SR501

Am folosit acest senzor pentru a detecta miscarea intr-o anumita camera. Se poate modifica sensibilitatea senzorului astfel incat sa detecteze miscarea la o distanta de 3 - 7 metri. Atunci cand se detecteaza miscare, pinul la care este conectat va avea valoarea HIGH, altfel va fi LOW:

```
if (PIR_PINBLOCK & (1 << PIR_PIN)) {  
    USART0_print("MOTION DETECTED!\r\n");  
} else {  
    USART0_print("NO MOTION DETECTED!\r\n");  
}
```

Senzor de gaz MQ-2

Senzorul MQ-2 ofera output atat digital, cat si analog. Am ales sa folosesc output-ul analog pentru ca am gasit o implementare facila de biblioteca in acest sens. Citirea datelor se face folosind pinul PA0. Initializarea canalului de ADC este realizata astfel (ca in laborator):

```
void ADC_init(void) {  
    ADMUX = (1 << REFS0);  
    ADCSRA = (1 << ADEN) | (7 << ADPS0);  
}
```

Pentru a citi efectiv date de la pinul PA0 am folosit urmatoarea functie:

```
int ADC_get(uint8_t channel)  
{  
    ADMUX = (ADMUX & ~(0x1f << MUX0));  
  
    ADCSRA |= (1 << ADSC);  
    while(ADCSRA & (1 << ADSC));  
  
    return ADC;  
}
```

Pentru a interpreta datele citite am folosit functia urmatoare din biblioteca gasita:

```
int GetGasPercentage(float rs_ro_ratio, int gas_id);
```

Releu

Releul poate fi controlat printr-ul pin setat ca output, LOW insemnand ca releul este inchis, iar HIGH deschis:

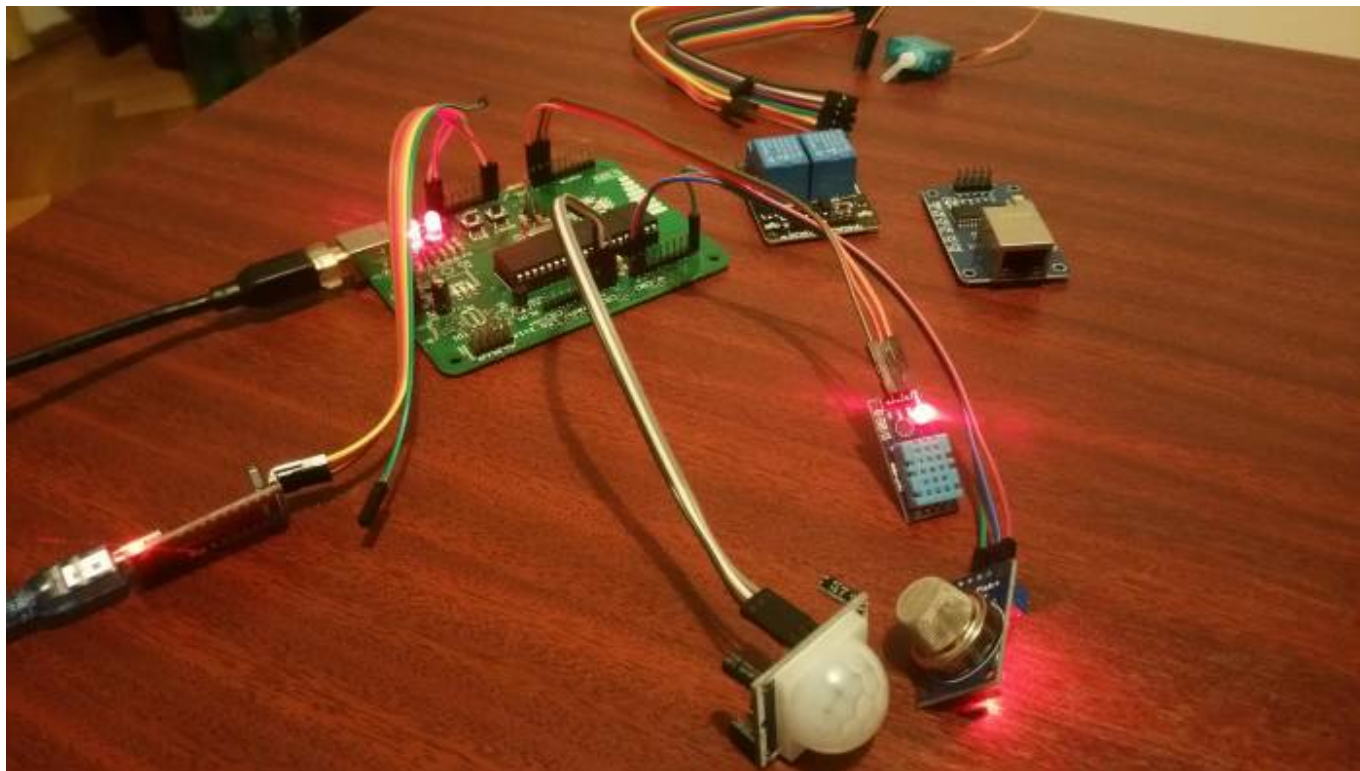
```
// toggle between LOW and HIGH  
RL_PORT ^= (1 << RL_PINBLOCK);
```

Modul Ethernet SPI ENC28J60

Desi inca nu am reusit sa folosesc acest modul (intentionez sa mai lucrez la el pana la PM Fair; voi actualiza aceasta sectiune in functie de rezultatele obtinute), pot spune ca voi folosi exemplul de webserver disponibil pe site-ul de PM si biblioteca disponibila pentru acest modul. Voi putea sa trimit o cerere GET catre placuta pentru a prelua date actualizate citite de la senzorii descrisi mai sus si, de asemenea, sa controlez un LED prin intermediul releului.

Rezultate Obținute

Am reusit sa citesc date de la toti senzorii si sa controlez releul. Din lipsa timpului, nu voi mai face incuietoarea, ma voi concentra in ultimele 2 zile pe modulul de Ethernet si sper ca pana la PM Fair sa fie functional.



Concluzii

Acest proiect a fost foarte interesant, mi-ar fi placut sa am mai mult timp pentru a lucra la el. Cred ca sistemele de case inteligente vor fi implementate in viitorul apropiat in multe locuinte, dat fiind faptul ca acestea imbunatatesc nivelul de trai al proprietarilor prin confortul si siguranta pe care le dobandesc. Voi mai lucra la acest proiect pana ce va arata ca in descrierea initiala.

Download

TBD

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).

Exemplu: Dumitru Alin, 331CC → **:pm:prj2017:avoinescu:dumitru_alin**.

Jurnal

25 aprilie 2018 - lipire placuta

02 mai 2018 - instalat bootloader

09 mai 2018 - achizitionare senzori si module

23 mai 2018 - proiect aproape finalizat

Bibliografie/Resurse

[Site-ul PM](#)

[DHT11 Library](#)

[MQ-2 Library](#)

[WebServer](#)

[Datasheet ATmega324](#)

- Documentația în format [PDF](#)

From:

<http://cs.curs.pub.ro/wiki/pm/> - **PM Wiki**

Permanent link:

<http://cs.curs.pub.ro/wiki/pm/prj2018/dghilinta/clauidumarinescu>

Last update: **2018/05/23 20:10**

