

Litiera Inteligenta

Autor:

Oprea Olivia Maria

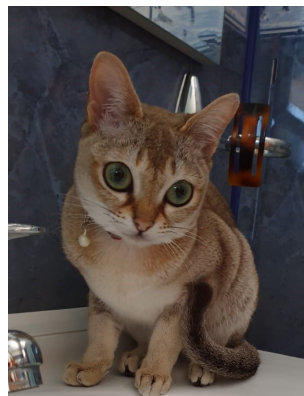
Introducere

Cei care detin pisici stiu cum este atunci cand acestea merg la toaleta si din pacate nu pot trage apa dupa ele, lasand in urma lor un parfum minunat.

Proiectul isi propune de a atentiona detinatorii de pisici atunci cand acestea intentioneaza sa foloseasca litiera, ajutandu-i sa fie pe faza sa "traga apa" dupa ele.

Idea acestui proiect a venit de la ghemotocul de blana care imi improspateaza casa de cel putin 2 ori pe zi fara sa stiu.

Acest proiect este util atat pentru mine cat si pentru altii deoarece ajuta orice detinator de pisica sa se bucure de un aer mai curat. De asemenea, pe mine ma ajuta acest proiect sa inteleg mai bine si sa pun in practica ceea ce am facut la laborator.

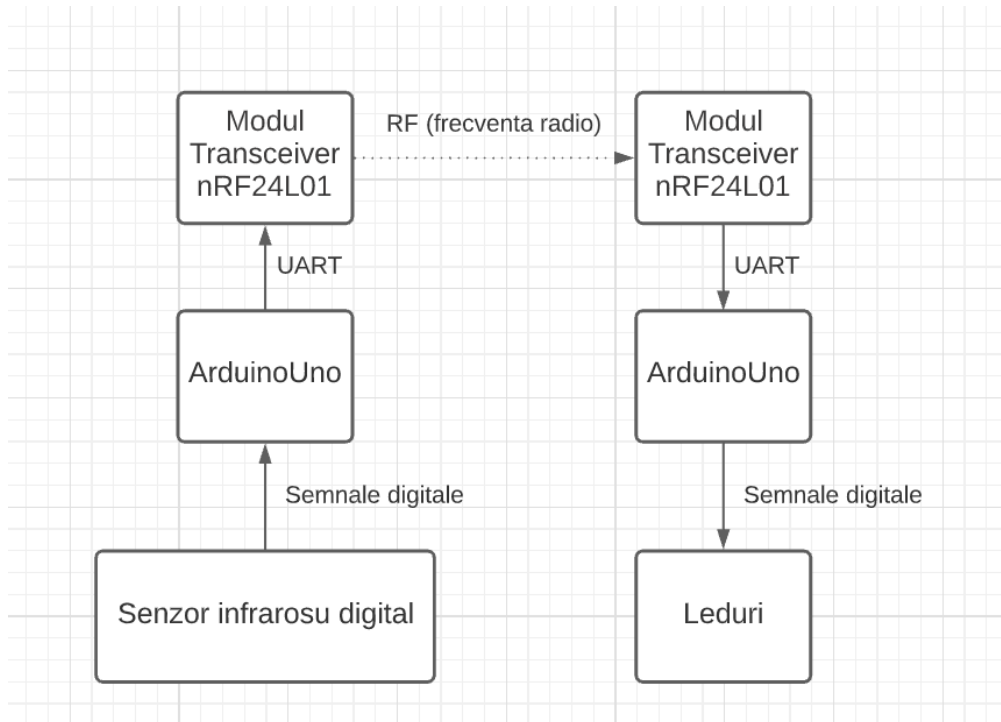


Descriere generală

Pentru realizarea acestui proiect voi folosi 2 placi [ArduinoUno](#), una aflandu-se pe biroul meu si alta la litiera pisicii. Acestea vor comunica cu ajutorul a doua [module transceiver nRF24L01](#). Placuta din litiera pisicii v-a fi **Master-ul** si v-a transmite datele primite de la [senzorul de distanta cu infrarosu](#) daca aceasta a intrat sau nu in litiera. Cealalta placuta o sa fie **Slave-ul**, iar atunci cand primeste date ca pisica a intrat in actiune o sa porneasca un set de [leduri](#) care ma vor atentiona de intentia

acesteia.

Diagrama bloc corespunzatoare proiectului:

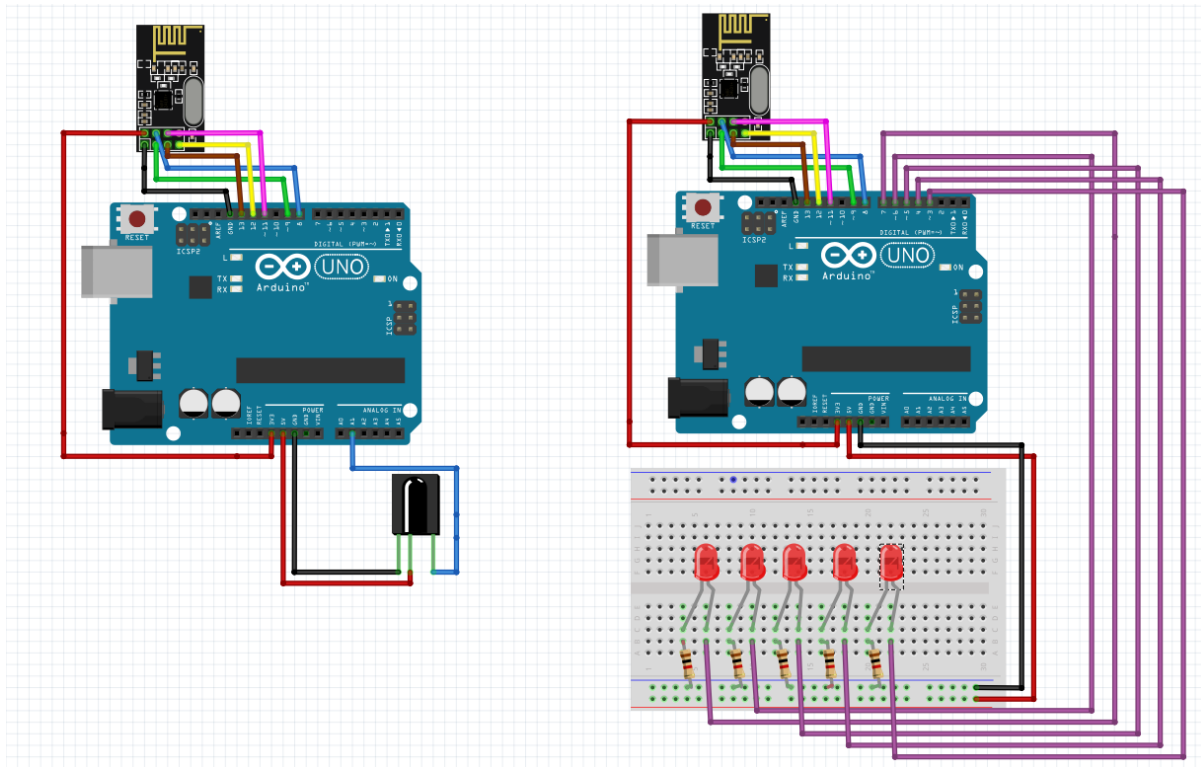


Hardware Design

Pentru a realiza acest proiect vom avea nevoie de:

Componente	Numar
ArduinoUno	2
Modul transceiver nRF24L01	2
Senzor infrarosu	1
Leduri rosi	5
Rezistente 1k	5
Breadboard	1 sau 2

Schema cablaj:



Software Design

Am ales sa folosesc drept mediu de dezvoltare **Arduino IDE**.

Librari necesare:

→ #include <SPI.h>
→ #include <nRF24L01.h>
→ #include <RF24.h>

Libraria NRF se poate gasi si aici: <https://github.com/nRF24/RF24>

```

COD MASTER
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
RF24 nrf(9, 8); // CE, CSN

const byte linkAddress[6] = "link1"; // address through which two modules
communicate.
int IRSensor = 2; // connect ir sensor to arduino pin 2
boolean state = 0;

void setup() {
  // put your setup code here, to run once:

```

```
pinMode (IRSensor, INPUT); // sensor pin INPUT

Serial.begin(9600);
Serial.println("MASTER: Starting");

nrf.begin();
nrf.openWritingPipe(linkAddress); // set the address
nrf.setPALevel(RF24_PA_HIGH);
nrf.stopListening(); // act as a transmitter
}

void loop() {
// put your main code here, to run repeatedly:
int statusSensor = digitalRead (IRSensor);

if (statusSensor == 1){
state = 1;
Serial.print("MASTER: Nu se intampla nimic");
nrf.write(&state, sizeof(state)); // send message to receiver
} else {
state = 0;
Serial.print("MASTER: A venit pisicuta master");
nrf.write(&state, sizeof(state)); // send message to receiver
}
delay(100);
}
```

```
COD SLAVE
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
RF24 nrf(9, 8); // CE, CSN

const byte linkAddress[6] = "link1"; // address through which two modules
communicate.
boolean state = 0;
const byte redLED1 = 7;
const byte redLED2 = 6;
const byte redLED3 = 5;
const byte redLED4 = 4;
const byte redLED5 = 3;

void setup() {
// put your setup code here, to run once:
pinMode(redLED1, OUTPUT);
pinMode(redLED2, OUTPUT);
pinMode(redLED3, OUTPUT);
pinMode(redLED4, OUTPUT);
pinMode(redLED5, OUTPUT);
```

```
Serial.begin(9600);
Serial.println("SLAVE: Starting");

nrf.begin();
nrf.openReadingPipe(0, linkAddress); // set the address
nrf.setPALevel(RF24_PA_HIGH);
nrf.startListening(); // set nrf as receiver
}

void loop() {
  // put your main code here, to run repeatedly:
  if (nrf.available()){ // read the data if available in buffer
    nrf.read(&state, sizeof(state));
    if (state == 1) {
      digitalWrite(redLED1, LOW);
      digitalWrite(redLED2, LOW);
      digitalWrite(redLED3, LOW);
      digitalWrite(redLED4, LOW);
      digitalWrite(redLED5, LOW);

      Serial.print("SLAVE: Nu se intampla nimic");
    } else {
      digitalWrite(redLED1, HIGH);
      delay(20);
      digitalWrite(redLED1, LOW);
      digitalWrite(redLED2, HIGH);
      delay(20);
      digitalWrite(redLED2, LOW);
      digitalWrite(redLED3, HIGH);
      delay(20);
      digitalWrite(redLED3, LOW);
      digitalWrite(redLED4, HIGH);
      delay(20);
      digitalWrite(redLED4, LOW);
      digitalWrite(redLED5, HIGH);
      delay(20);
      digitalWrite(redLED5, LOW);

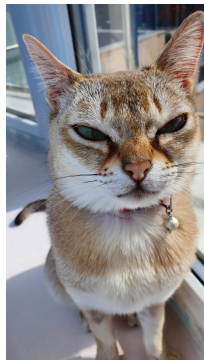
      Serial.print("SLAVE: A venit pisicuta");
    }
  }
  delay(100);
}
```

Rezultate Obținute

Se poate vedea o scurta demonstratie aici: <https://youtu.be/qD8kIzIj2Q0>

Concluzii

Dupa acest proiect pisica nu a fost prea fericita sa coopereze.



Cu toate acestea, litiera inteligenta este complet functionala si gata de utilizare.

Download

Documentatia in format PDF part1:

[litiera_inteligenta_ollie.pdf](#)

Link pagina: <https://ocw.cs.pub.ro/courses/pm/prj2021/abasoc/litierainteligenta>

Jurnal

24 Aprilie → alegere tema proiect

27 Aprilie → au sosit toate piesele necesare realizarii proiectului

01 Mai → implementare proiect

Bibliografie/Resurse

Diagrama bloc → <https://www.lucidchart.com>

Schema cablaj → <https://fritzing.org>

nRF24L01 datasheet →

https://www.mouser.com/datasheet/2/297/nRF24L01_Product_Specification_v2_0-9199.pdf

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/abasoc/litierainteligenta>

Last update: **2021/05/01 19:10**

