

Textul și imaginile din acest document sunt licențiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursă din acest document este licențiat

Public-Domain

Ești liber să distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, tipărire, sau orice alt mijloc), atât timp cât nu aduci nici un fel de modificări acestuia. Codul sursă din acest document poate fi utilizat în orice fel de scop, de natură comercială sau nu, fără nici un fel de limitări dar autorii nu își asumă nici o răspundere pentru pagubele pricinuite de implementările realizate de utilizatori. Schemele și codul sursă au un rol educativ și nu sunt gândite pentru a fi utilizate în mediu de producție (industrial, casnic sau comercial).

Convertor ASK / USB

Multe dintre dispozitivele casnice fără fir funcționează prin comunicație radio în bandă ISM (1) de 433MHz modulație ASK (2). Cele mai des întâlnite astfel de dispozitive sunt sistemele de securitate fără fir, prizele sau soclurile de bec comandate de la distanță și chiar sistemele meteo cu senzori fără fir. Toate aceste sisteme oferă dispozitive fără fir gata construite, ieftine și foarte ușor de folosit în sisteme de comandă personalizate. Dacă sistemul de comandă este bazat pe o placă de dezvoltare de tipul Arduino nu este nici un fel de problemă deoarece există module



radio ASK în bandă de 433MHz cu un cost mic și ușor de folosit. Dacă sistemul de comandă are o complexitate mai mare și necesită utilizarea unui PC sau a unei plăci de tipul Raspberry Pi lucrurile se complică puțin. În cadrul acestei lecții ne propunem să realizăm un sistem ieftin și de mici dimensiuni care să realizeze legătura între comunicația radio în bandă de 433MHz modulație ASK și portul USB prezent în sistemele de calcul de uz general (PC-uri, laptop-uri, plăci de dezvoltare bazate pe microprocesoare). Astfel de dispozitive există deja dar au un preț destul de mare (a se vedea (3)).

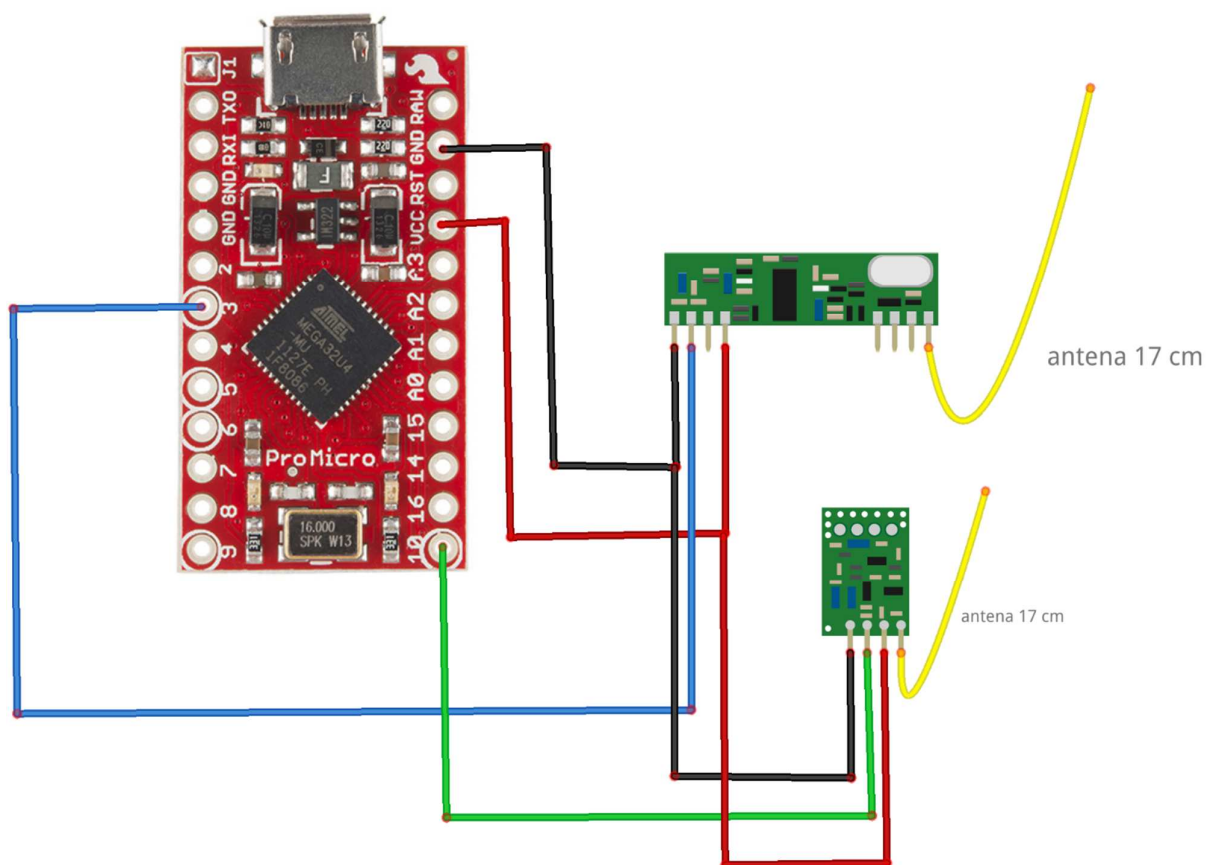
Sistemul nostru se va baza pe o placă de dezvoltare Arduino Pro Micro (4) cu un microcontroler ATmega 32U4 la 16MHz / 5V, un transmițător radio (5) și un receptor radio (6) – ambele în bandă de 433MHz modulație ASK. Componentele au fost alese astfel încât să asigure un cost total cât mai scăzut și posibilitatea de a fi integrate într-o carcasă de mici dimensiuni (aproximativ cât o baterie USB externă, imagine alăturată, astfel de carcase se găsesc de cumpărat pe Internet la prețuri de circa 2\$).



Dispozitivul rezultat va fi văzut de sistemul de calcul în cadrul căruia va fi utilizat ca un dispozitiv serial ce va transmite către sistem codurile radio primite și va transmite radio codurile primite de la sistem – va fi realiza conversia dintre comunicația serială (over USB) și comunicația radio ASK în bandă de 433MHz.

Interconectarea componentelor

Interconectarea dintre placa de dezvoltare și cele două module radio este reprezentată în diagrama următoare:



Transmițătorul radio va avea linia de date conectată la pinul 10 al plăcii de dezvoltare iar receptorul radio va avea linia de date conectată la pinul 4 al plăcii de dezvoltare (pinul de întrerupere 0). Ambele module se vor alimenta la 5V prin intermediul pinilor VCC / GND ai plăcii de dezvoltare.

Pentru mai multe detalii despre instalarea și operarea plăcii Sparkfun Pro Micro se recomandă parcurgerea materialului "Pro Micro & Fio V3 Hookup Guide" (7).

Programarea sistemului

Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.1 cu extensia Arduino AVR Boards 1.6.19 instalată și bibliotecă rc-switch 2.6.2.

```
#include <string.h>
#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  while (!Serial);
  Serial.begin(115200);
  mySwitch.enableTransmit(10);
  mySwitch.setRepeatTransmit(3);
  mySwitch.enableReceive(0);
}

void loop() {
  char inData[80];
  int index = 0;
  while(Serial.available() > 0)
  {
    char aChar = Serial.read();
    if(aChar == '\n')
    {
      inData[index] = NULL;
      index = -1;
    }
    else
    {
      inData[index] = aChar;
```

```

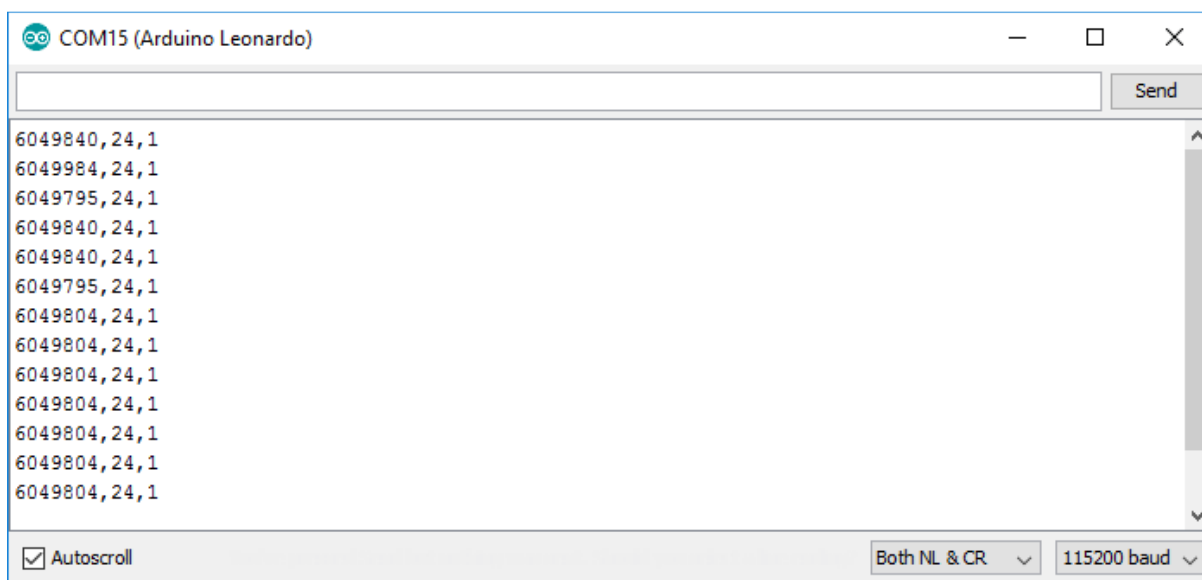
        index++;
        inData[index] = '\0';
    }
}
char * number;
unsigned long number1, number2, number3;
if (index== -1) {
    number = strtok(inData, ",");
    number1 = atol(number);
    number = strtok(NULL, ",");
    if(number!=NULL) { number2=atoi(number); }
    number = strtok(NULL, ",");
    if(number!=NULL) { number3=atoi(number); mySwitch.setProtocol(number3);}
    mySwitch.send(number1,number2);
}
if (mySwitch.available()) {
    int value = mySwitch.getReceivedValue();
    if (value == 0) {
        Serial.print("Unknown encoding");
    } else {
        Serial.print( mySwitch.getReceivedValue() );
        Serial.print(",");
        Serial.print( mySwitch.getReceivedBitlength() );
        Serial.print(",");
        Serial.println( mySwitch.getReceivedProtocol() );
    }
    mySwitch.resetAvailable();
}
}

```

Comunicația dintre sistemul de calcul și dispozitivul USB construit va avea mesajele de forma *COD,DIMENSIUNE,VERSIUNEPROTOCOL* .

De exemplu, în consola serială, apăsarea butoanelor unei telecomenzi radio ce face parte dintr-un sistem de securitate a locuinței va avea următorul efect (se observă codul butonului de dimensiune 24 biți, protocol radio versiunea 1).

Pentru mai multe informații legate de diverse dispozitive radio de securitate se poate parcurge și proiectul "HomeWatch" din cartea "10(zece) proiecte cu Arduino" (8).



În cazul în care dorim să dăm o comandă radio vom tasta în consola serială cei trei parametrii necesari transmisiei. De exemplu: 2474994176,32,2 (cod de deschidere pe 32 de biți, protocol versiunea 2, specific unei prize radio din gama Conrad RSL).



Pentru mai multe informații legate de dispozitivele din gama Conrad RSL se pot parcurge și materialele ” Cloud’s Lights” (9) și ” Local Area Power Plugs” (10).

Bineînțeles, dispozitivul prezentat nu își găsește utilitatea majoră în comanda sau supravegherea manuală în consola serială. Utilizarea lui trebuie integrată într-o aplicație specifică platformei la care este conectat (o aplicație Windows sau un script de automatizare sub platforma Linux) și sub care pot fi predefinite codurile de comandă pentru diverse dispozitive cunoscute.

O aplicație utilă poate fi utilizarea dispozitivului în conjuncție cu o platformă de automatizare a locuinței, de exemplu OpenHAB (11). În acest scop se pot vedea și materialele ” Realizarea unui sistem de tip Home Automation” (12) și ” Conectarea unor dispozitive antiefracție la platforma OpenHab” (13).

Referințe on-line

(1) ISM band

https://en.wikipedia.org/wiki/ISM_band

(2) Amplitude-shift keying

https://en.wikipedia.org/wiki/Amplitude-shift_keying

(3) easyRadio 433MHZ USB Transmitter for Raspberry Pi

<https://www.seeedstudio.com/easyRadio-433MHZ-USB-Transmitter-for-Raspberry-Pi-p-2325.html>

(4) Pro Micro 5V/16MHz - ATmega 32U4

https://www.robofun.ro/pro_micro_5v_16_mhz?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(5) Transmittor radio 434 Mhz

https://www.robofun.ro/wireless/wireless-433/transmittor_radio?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(6) Receptor radio 434 MHz

https://www.robofun.ro/wireless/wireless-433/receptor_radio?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(7) Pro Micro & Fio V3 Hookup Guide

<https://www.robofun.ro/forum/>

<https://www.sparkfun.com/products/12640>

(8) 10(zece) proiecte cu Arduino

<https://play.google.com/store/books/details?id=aMB8BgAAQBAJ>

(9) Cloud's Lights

<https://blog.robofun.ro/2017/03/28/proiect-clouds-lights/>

(10) Local Area Power Plugs

<https://blog.robofun.ro/2017/03/21/proiect-local-area-power-plugs/>

(11) openHAB

<https://www.openhab.org/>

(12) Realizarea unui sistem de tip Home Automation

<https://blog.robofun.ro/2016/07/01/realizarea-unui-sistem-de-tip-home-automation-partea-i/>

(13) Conectarea unor dispozitive antiefracție la platforma OpenHab

<https://blog.robofun.ro/2016/07/05/conectarea-unor-dispozitive-antiefracție-la-platforma-openhab/>