

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

Local Area Power Plugs

Controlul centralizat al alimentării cu energie electrică a consumatorilor este o problemă în permanentă căutare de noi soluții și de noi modalități de implementare a acestora. Chiar dacă vorbim de consumatori casnici – aplicații de automatizare a locuinței – și chiar dacă există soluții deja cunoscute de implementare a controlului centralizat la acest nivel, considerăm subiectul de interes suficient de mare încât să propunem o soluție nouă elegantă și extraordinar de simplu de implementat. Bazându-ne pe elemente de alimentare comandate în bandă ISM de 433MHz și pe o placă de dezvoltare Arduino Leonardo ETH vom dezvolta o aplicație de comandă centralizată de comandă prin intermediul rețelei locale.

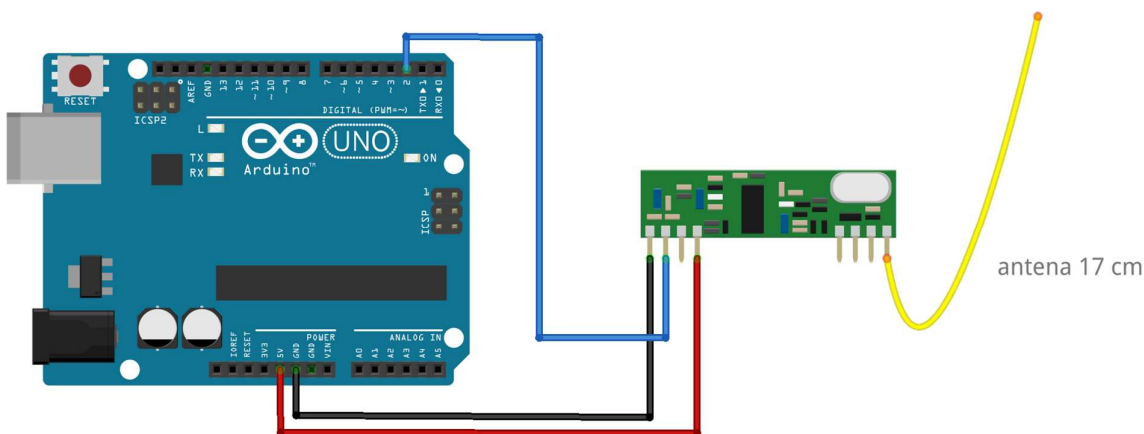


Având în vedere faptul că construirea de la zero a unui element de comandă de putere (220V) este riscantă (prezintă risc de electrocutare și de incendiu) materialul vine cu propunerea de utilizare a unor elemente prefabricate / comercializate ca dispozitive de sine stătătoare, și anume prizele intermediare telecomandate Conrad RSL comercializate la noi în țară de German Electronics SRL:



<https://www.germanelectronics.ro/casa-gradina/sistem-automatizare-casa-conrad-rsl/set-3-prize-intermediare-wireless-rsl-si-telecomanda-1168455.html>

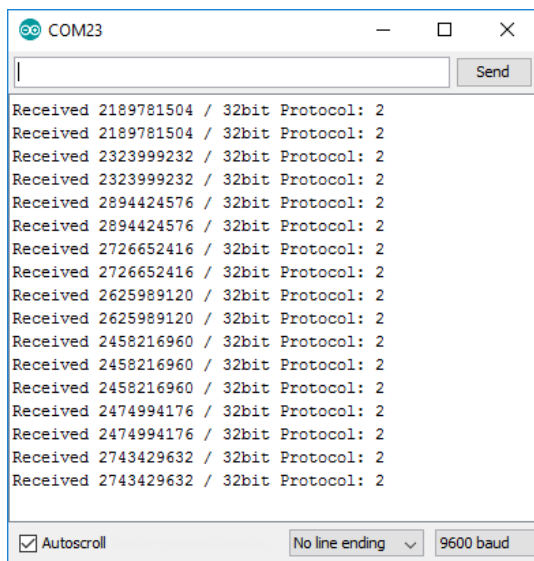
Aceste prize sunt controlate prin intermediul unei telecomenzi radio (bandă 433MHz) de la o distanță de maxim 30m (suficient pentru un apartament obișnuit). În cadrul sistemului nostru vom înlocui telecomanda furnizată de producător cu un sistem de rețea bazat pe placa Arduino Leonardo ETH (telecomanda va putea fi folosită în paralel cu sistemul de comandă propus). Pentru a putea implementa sistemul de comandă este necesară observarea codurilor emise de telecomanda oferită de producător, acest lucru se poate face cu ajutorul unui sistem simplu compus dintr-o placă Arduino Uno (sau orice versiune de placă Arduino compatibilă) și un receptor radio 433MHz:



https://www.robofun.ro/arduino/arduino_uno_v3

https://www.robofun.ro/wireless/wireless-433/receptor_radio

Rulând exemplul *ReceiveDemo_Simple* al bibliotecii software *rc-switch* vom putea vizualiza tipul de cod și codul emis la fiecare apăsare de buton al telecomenzii:



<https://github.com/sui77/rc-switch/>

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

Se observă că seria Conrad RSL folosește un cod de comandă pe 32 de biți. Din cei 32 de biți trimiși la fiecare apăsare de buton: primii **doi** biți sunt întotdeauna 10 (secvență de sincronizare), următorii **6** codează canalul (I-IV), numărul butonului (1-4) și comanda (On, Off, All-On, All-Off) iar ultimii **24** reprezintă un identificator unic al telecomenzii – cu alte cuvinte la apăsarea diverselor butoane ale aceleiași telecomandă vor varia doar șase biți (ce notați cu roșu) – 64 de combinații posibile.

10 00 00 10 100001010110101000000000

Pentru mai multe informații legate de protocolul folosit de dispozitivele Conrad RSL puteți consulta și discuția de la adresa:

Conrad RSL Switch

<https://forum.pilight.org/Thread-Fully-Supported-Conrad-RSL-Switch?page=7>

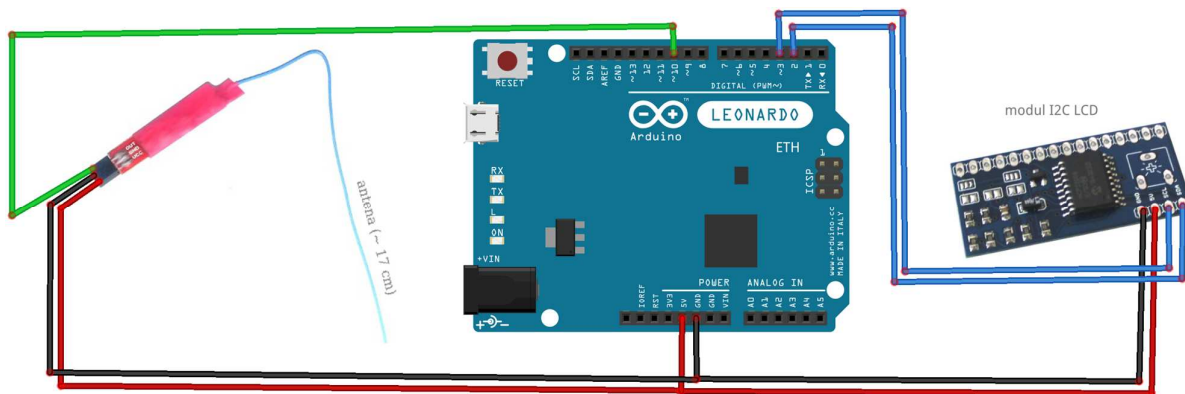
Spre deosebire de alte prize telecomandate în bandă ISM, prizele Conrad RSL permit învățarea codurilor emise de telecomandă în mod dinamic. Butonul roșu de pe priză folosește la deschiderea / închiderea manuală a prizei dar, prin apăsare prelungă, și la învățarea unui nou cod de comandă. Fiecare priză poate memora până la 8 coduri de comandă diferite. Priza memorează și în lipsa alimentării cu energie (deconectată) codurile învățate și starea (dacă este deconectată în starea deschis la reconectare va avea starea deschis). În plus fiecare priză integrează protecție la suprasarcină. La prima punere în funcțiune este necesară citirea cu atenție a documentației pusă la dispoziție de producător pentru înțelegerea exactă a modului de operare.

Comparativ, cu privire la protocolul radio de comunicație, se poate parcurge următorul material pentru a înțelege diferențele față de alte protocoale specifice unor produse similare:

Intertechno Code Berechnung

https://wiki.fhem.de/wiki/Intertechno_Code_Berechnung

După determinarea codurilor emise de telecomandă se poate trece la contruirea sistemului de comandă. Acesta se bazează pe o placă Arduino Leonardo ETH și o telecomandă radio 433MHz Arduino. Opțional, pentru a putea supraveghea sistemul, se poate adăuga un ecran LCD 20x4 I2C. Schema de interconectare este următoarea:



<https://www.robofun.ro/arduino/arduino-leonardo-eth>

<https://www.robofun.ro/wireless/wireless-433/telecomanda-prize-434MHz-arduino>

https://www.robofun.ro/lcd/lcd_20x4_i2c_negru_verde

Ambele componente ale sistemului se alimentează la 5V. Emițătorul radio se va conecta la pinul digital 10 al plăcii de dezvoltare iar ecranul LCD (modulul I2C) va utiliza magistrala I2C a plăcii de dezvoltare (pinii 2 și 3).

Programul sistemului va utiliza bibliotecile software rc-switch, LiquidTWI și Ethernet2 (biblioteca necesară noului chip Wiznet 5500 cu care este echipată placa Arduino Leonardo ETH):

```
#include <RCSwitch.h>
RCSwitch mySwitch = RCSwitch();

#include <Wire.h>
#include <LiquidTWI.h>
LiquidTWI lcd(0);

#include <SPI.h>
#include <Ethernet2.h>

EthernetServer server(80);
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
String HTTP_req;
```

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

```
boolean plug1_status = 0;
boolean plug2_status = 0;
boolean plug3_status = 0;
```

În cadrul secțiunii *setup()* vom inițializa componentele de lucru cu ecranul LCD, cu emițătorul radio și cu controlerul ethernet (sistemul va funcționa într-o rețea locală bazată pe configurare dinamică – DHCP):

```
void setup() {
  mySwitch.enableTransmit(10);
  mySwitch.setProtocol(2);
  mySwitch.setRepeatTransmit(3);
  lcd.begin(20, 4);
  lcd.clear();
  lcd.setCursor(0,0);
  if (Ethernet.begin(mac) == 0) { lcd.print(F("Failed
                                     DHCP")); while(1); }
  else
  { server.begin();
    lcd.print("SrvIP: ");
    lcd.print(Ethernet.localIP());
  }
  delay(5000); }
```

Secțiunea *loop()* implementează interfața web de comandă (captură de ecran) și se va ocupa cu trimiterea comenzilor radio provenite de la utilizator pe această cale:

Local Area Power Plugs

Click to switch plugs on and off.

Power Plug 1:	Turn On Plug1	Turn Off Plug1
Power Plug 2:	Turn On Plug2	Turn Off Plug2
Power Plug 3:	Turn On Plug3	Turn Off Plug3

All Power Plugs:	Turn On All Plugs	Turn Off All Plugs
------------------	-------------------	--------------------

```

void loop() {
  EthernetClient client = server.available();
  if (client) {
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        HTTP_req += c;
        if (c == '\n' && currentLineIsBlank) {
          client.println(F("HTTP/1.1 200 OK"));
          client.println(F("Content-Type: text/html"));
          client.println(F("Connection: close"));
          client.println();
          client.println(F("<!DOCTYPE HTML>"));
          client.println(F("<html>"));
          client.println(F("<head>"));
          client.println(F("<title>Local Area
                                PowerPlugs</title>"));
          client.println(F("</head>"));
          client.println(F("<body>"));
          client.println(F("<h1>Local Area Power
                                Plugs</h1>"));
          client.println(F("<p>Click to switch plugs on
                                and off.</p>"));
          client.println(F("Power Plug 1: "));
          client.println(F("<button
                                onclick=\"\"window.location.href='/?pluglon'\"\"
                                >Turn On Plug1</button>"));
          client.println(F("<button
                                onclick=\"\"window.location.href='/?plugloff'\"
                                >Turn Off Plug1</button><br>"));
          client.println(F("Power Plug 2: "));

```

```

        client.println(F("<button
            onclick=\"\"window.location.href='/?plug2on'\"\"
            >Turn On Plug2</button>"));
        client.println(F("<button
            onclick=\"\"window.location.href='/?plug2off'\"\"
            >Turn Off Plug2</button><br>"));
        client.println(F("Power Plug 3: "));
        client.println(F("<button
            onclick=\"\"window.location.href='/?plug3on'\"\"
            >Turn On Plug3</button>"));
        client.println(F("<button
            onclick=\"\"window.location.href='/?plug3off'\"\"
            >Turn Off Plug3</button><br><br>"));
        client.println(F("<b>All Power Plugs: </b>"));
        client.println(F("<button
            onclick=\"\"window.location.href='/?plugson'\"\"
            >Turn On All Plugs</button>"));
        client.println(F("<button
            onclick=\"\"window.location.href='/?plugsoff'\"\"
            >Turn Off All Plugs</button><br>"));
        client.println(F("</body>"));
        client.println(F("</html>"));
        break;
    }
    if (c == '\n') {
        currentLineIsBlank = true;
    }
    else if (c != '\r') {
        currentLineIsBlank = false;
    }
}
}
}

```



```
delay(1);
client.stop();
if (HTTP_req.indexOf("?plug1on") >0) {
    mySwitch.send(2189781504,32);
    lcd.setCursor(0,1);
    lcd.print("Plug1 On ");
}
else if (HTTP_req.indexOf("?plug1off") >0) {
    mySwitch.send(2323999232,32);
    lcd.setCursor(0,1);
    lcd.print("Plug1 Off");
}
else if (HTTP_req.indexOf("?plug2on") >0) {
    mySwitch.send(2424662528,32);
    lcd.setCursor(0,2);
    lcd.print("Plug2 On ");
}
else if (HTTP_req.indexOf("?plug2off") >0) {
    mySwitch.send(2558880256,32);
    lcd.setCursor(0,2);
    lcd.print("Plug2 Off");
}
else if (HTTP_req.indexOf("?plug3on") >0) {
    mySwitch.send(2625989120,32);
    lcd.setCursor(0,3);
    lcd.print("Plug3 On ");
}
else if (HTTP_req.indexOf("?plug3off") >0) {
    mySwitch.send(2458216960,32);
    lcd.setCursor(0,3);
    lcd.print("Plug3 Off");
}
```

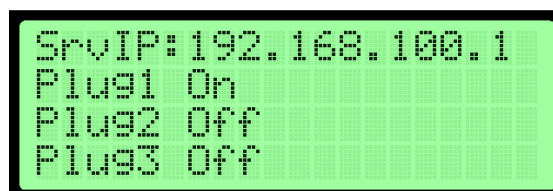
```

else if (HTTP_req.indexOf("?plugson") >0) {
    mySwitch.send(2474994176,32);
    lcd.setCursor(0,1);
    lcd.print("Plug1 On ");
    lcd.setCursor(0,2);
    lcd.print("Plug2 On ");
    lcd.setCursor(0,3);
    lcd.print("Plug3 On ");
}
else if (HTTP_req.indexOf("?plugsoff") >0) {
    mySwitch.send(2743429632,32);
    lcd.setCursor(0,1);
    lcd.print("Plug1 Off");
    lcd.setCursor(0,2);
    lcd.print("Plug2 Off");
    lcd.setCursor(0,3);
    lcd.print("Plug3 Off");
}
HTTP_req="";
}
}

```

Programul a fost dezvoltat și testat cu Arduino IDE 1.8.1 și bibliotecile rc-switch 2.6.2, Ethernet2 1.0.4 și LiquidTWI 1.5.1.

Neconectarea LCD-ului (neutilizarea acestuia) nu afectează în nici un fel funcționalitatea web a sistemului. Interfața web de comandă se va accesa din aceeași rețea locală cu sistemul de comandă utilizând un client web și accesând adresa de IP a sistemului. LCD-ul nu face altceva decât să refectione comenzile radio transmise prin intermediul interfeței web:



Comunicația între sistemul de comandă și prize este unidirecțională. Sistemul nu poate determina starea unei prize ci poate doar să transmită o comandă către aceasta – LCD-ul nu afișează starea prizelor ci ultima comandă transmisă, dacă starea prizei este modificată manual (din buton) sau din telecomandă acest lucru nu se va reflecta în informația afișată.

Seria de dispozitive de automatizare a locuinței Conrad RSL este compusă din dispozitive telecomandate radio perfect compatibile între ele. În cadrul sistemului de comandă construit se pot adăuga și alte elemente telecomandate din această serie. Codurile transmise de sistemul de comandă trebuie doar înregistrate în noile dispozitive adăugate. Vă recomandăm și priza intermediară de exterior (protecție IP44, funcționare identică cu a celor din setul de prize intermediare prezentat, memorie 10 coduri de comandă, 2000W, recepție comandă până la 70m):



<https://www.germanelectronics.ro/casa-gradina/sistem-automatizare-casa-conrad-rsl/priza-wireless-rslr2-2000-w-sarcina-rezistiva-640451.html>