

Remote PC Control

Ștefan Silviu-Alexandru

Introducere

Scopul proiectului este de a permite executia unor comenzi predefinite pe PC, cu o telecomanda de la distanta, via un microcontroller.

Cea mai buna motivatie este lenea, si de la asta a venit si ideea pentru acest proiect. Intr-adevar, as putea sa opresc din a ma uita pe netflix la televizor ca sa opresc PCul, dar de ce sa fac asta, cand pot sa apas un buton si sa nu ma mai ridic?

Impreuna cu un daemon ce ruleaza pe PC, tot felul de actiuni mai mult sau mai putin privilegiate pot fi facute printr-o simpla apasare de buton.

Descriere generală

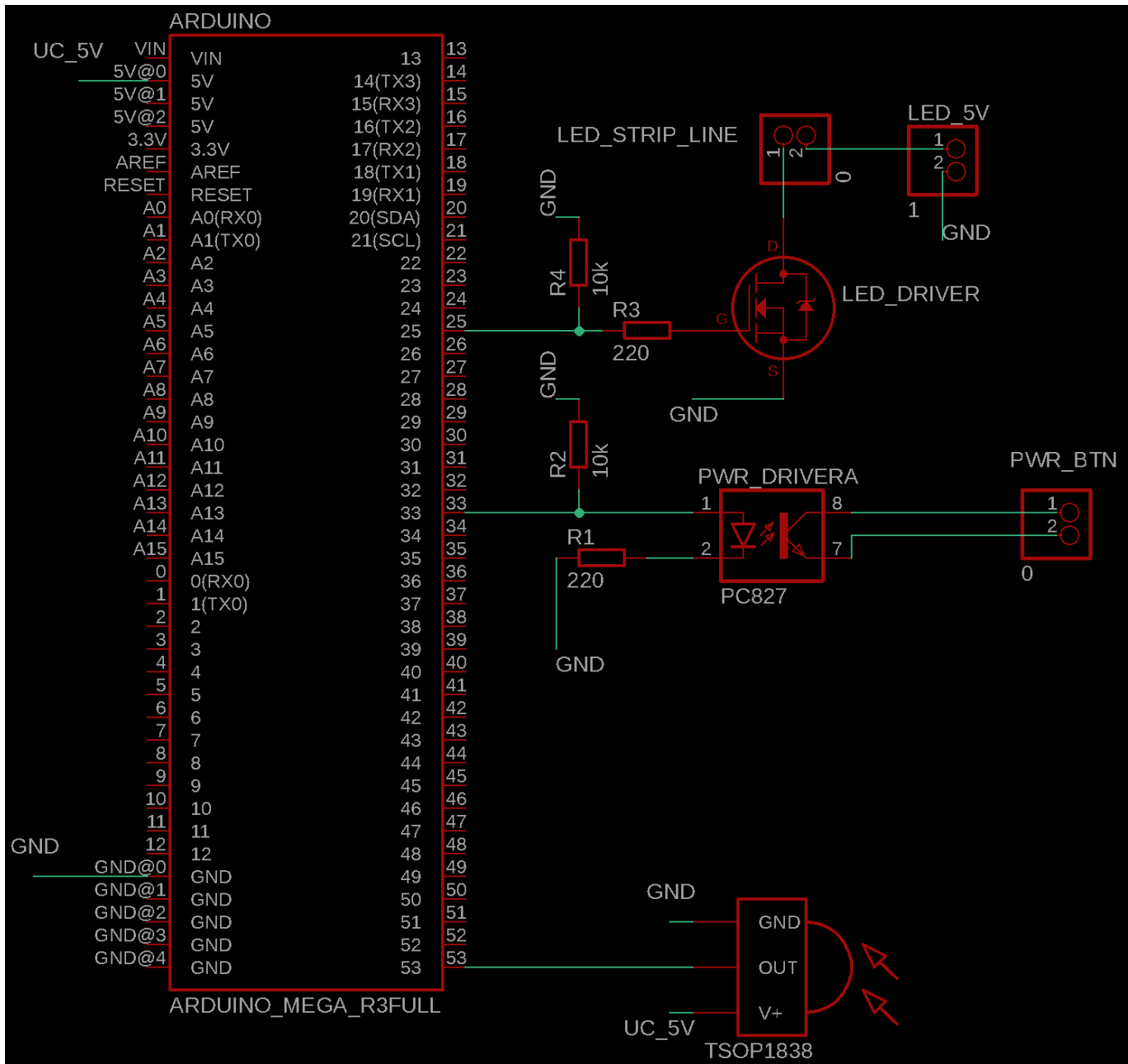
Rosu: alimentare

Albastru: elemente controlate de μ C



Hardware Design

- Arduino (Mega2560)
- 2N7000 N-MOSFET
- PC827 Optocoupler
- VS1838 IR Module (+telecomanda)
- On/Off switch
- alte piese de baza (rezistor, fire etc)
- alte chestii deja existente (PC-ul :), LED strip-ul, fire usb)



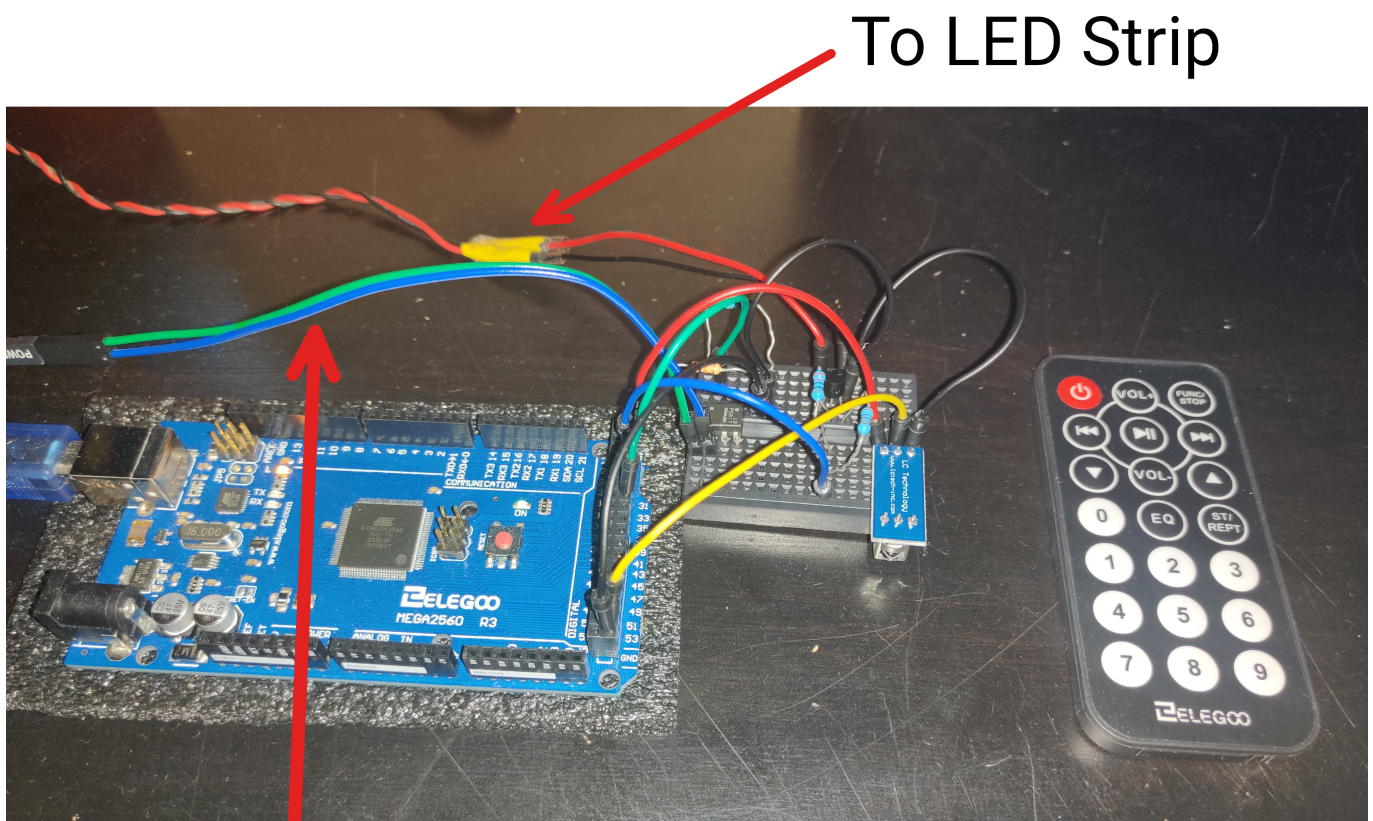
Motivatii pentru piese:

Initial am vrut sa folosesc un NPN in loc de MOSFET, dar ledurile au 300mA si nu se aprindeau cum trebuie.

Tot initial voiam sa folosesc un ATTiny85 in loc de ditamai Mega-ul, ca am nevoie doar de 3 pini digitali. Problema e ca tiny-ul nu are hardware pentru seriala, si software serial nu prea mergea.

Optocuplor - nu vreau sa leg arduino care costa \$ direct la PC care costa \$\$\$\$, deci tranzistor nu merge. Cealalta optiune e un releu, dar are 2 dezavantaje, anume ca e foarte mare (optocuplorul este mult mai mic), si ca am doar unul singur (optocuploare am o punguta). Una sau mai multe diode din optoculoare au fost arse pana sa-mi aduc aminte de ce nu legi diode direct la VCC.

Asa arata construita:



To Power Switch

Cele 2 fire pentru LED strip actioneaza ca un intrerupator, strip-ul fiind alimentat separat.

Firele pentru power switch sunt legate direct la pinii de pe motherboard din fpanel. Similar, sunt capetele unui intrerupator.

Software Design

Cod μ C

CLion + PlatformIO plugin a fost folosit pentru scrierea/uploadul codului.

Foloseste library-ul IRRemote pentru a interactiona cu modulul IR.

Codul e relativ simplu, si scurt:

```
#include <Arduino.h>
#include <IRremote.h>

#define IR_PIN 53
#define PWR_BTN_PIN 33
```

```

#define LIGHTS_PIN 25

IRrecv irrecv(IR_PIN);
decode_results results;

void setup() {
  pinMode(LIGHTS_PIN, OUTPUT);
  pinMode(PWR_BTN_PIN, OUTPUT);
  Serial.begin(9600);
  irrecv.enableIRIn();
}

int lightsPower = LOW;

void loop() {
  if (irrecv.decode(&results)) {
    switch (results.value) {
      case 0xFFA25D:
        Serial.println("POWER");
        digitalWrite(PWR_BTN_PIN, HIGH);
        delay(100);
        digitalWrite(PWR_BTN_PIN, LOW);
        break;
      case 0xFFE21D:
        Serial.println("FUNC/STOP");
        lightsPower = !lightsPower;
        digitalWrite(LIGHTS_PIN, lightsPower);
        break;
      case 0xFF629D:
        Serial.println("VOL+");
        break;
      case 0xFF22DD:
        Serial.println("FAST BACK");
        break;
      /* snip snip restu butoanelor aici */
      case 0xFFFFFFFF:
        Serial.println("REPEAT LAST");
        break;
      default:
        /* other button */;
    }
    delay(500);
    irrecv.resume();
  }
}

```

Constantele magice se refera la comenzi NEC:

<https://github.com/Arduino-IRremote/Arduino-IRremote/issues/631#issuecomment-503101884>

Cum sunt folosite fiecare o singura data, nu le-am mai pus intr-un enum.

Cod PC

Command runner-ul de pe PC foloseste Typescript/Node.

Link git: <https://github.com/slak44/proiect-pm>

Pentru comunicarea cu microcontrollerul, am folosit library-ul `serialport`. Calea catre device-ul serial e configurabila printr-un env variable, `TTY_PATH` (`/dev/ttyACM0` in cazul meu). Library-ul citeste linie cu linie (tehnica pana la `\r\n`), si livreaza continutul printr-un event de date.

Avand evenuri cu comenzi de la arduino, partea de rulat este simpla. Un switch pe lista de comenzi posibile + actiunea relevanta (`handleSerialData` in `index.ts`).

Pentru controalele de media, am folosit standardul MPRIS, care specifica un API standard pe Linux pentru playere de audio/video. MPRIS este implementat pe Dbus. Fiecare player expune un serviciu Dbus care implementeaza metodele si proprietatile din standard. Command runner-ul listeaza toate serviciile din session bus, si le alege pe cele de tip MPRIS. De asemenea, asculta schimbari pe Dbus, si isi modifica lista interna de playere (daca se deschide un player nou, sau daca se inchide unul existent). Runnerul permite controlul a mai multor playere simultan, in mod dinamic.

DBus este relativ low-level, chiar si folosind un library care se ocupa de conexiunea cu el. Am scris un mic wrapper peste o fractiune din MPRIS (play, pause, volume, comenzi de baza). Se poate gasi in `mpris-proxy.ts`. Codul principal doar foloseste acest wrapper direct, de exemplu:

```
case SerialMessage.VOLUME_PLUS:
    await accessor.player.setVolume(await accessor.player.volume() +
    volumeDelta);
    break;
```

In afara de comenzile media, runner-ul accepta si scripturi custom. Butoanele 0..9 de pe telecomanda sunt mapate la 10 scripturi din folder-ul runnerului, `script${nr}.sh`. Astfel se pot executa comenzi arbitrare folosind telecomanda.

Rezultate Obținute

<https://www.youtube.com/watch?v=AwqmG-MQOOQ>

Concluzii

Proiectul e destul de fun, si are potential sa-mi fie chiar util. O posibila miniaturizare + imbunatatirea distantei la care merge telecomanda l-ar face mai practic si mai usor de utilizat. Chiar si in starea lui curenta, merge lasat legat la PC :)

Download

Cod: <https://github.com/slak44/proiect-pm/archive/refs/heads/master.zip>

Jurnal

- 25 aprilie - completare idee proiect
- 14 mai - schema
- 23 mai - schema v2
- 24 mai - documentatie finala
- 25 mai - video cu rezultat + concluzie
- zile random din mai - implementarea in sine

Bibliografie/Resurse

PDF: https://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/remote-pc-control?do=export_pdf

Cod: <https://github.com/slak44/proiect-pm>

Hardware:

- <https://github.com/Arduino-IRremote/Arduino-IRremote>
- https://www.electronics-tutorials.ws/transistor/tran_7.html
- <https://www.onsemi.com/pdf/datasheet/nds7002a-d.pdf>
- <https://datasheetspdf.com/pdf/701590/LEN/VS1838/1>
- <https://learnabout-electronics.org/Downloads/PC817%20optocoupler.pdf>
- <https://ocw.cs.pub.ro/courses/icalc>, bine ca am ales ic pt eagle

Software:

- <https://github.com/serialport/node-serialport>
- <https://github.com/dbusjs/node-dbus-next>
- <https://specifications.freedesktop.org/mpris-spec/latest/>
- <https://wiki.archlinux.org/index.php/MPRIS>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/remote-pc-control>



Last update: **2021/05/25 17:55**

