

Proiect SMP
Seuleanu Mihnea-Radu
332AB

Banda de leduri RGB programata in Python

Facultatea de Automatica si Calculatoare
2022

Cuprins

Descrierea functionalitatii.....	2
Proiectare hardware	3
Proiectare software	4

Descrierea functionalitatii

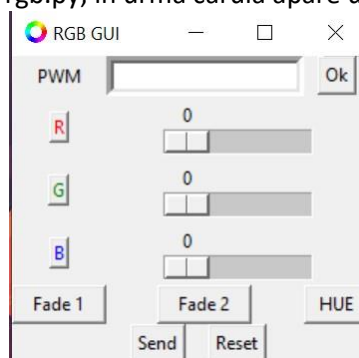
Proiectul “Banda de leduri RGB programata in Python” a avut rolul de a ma familiariza si de a ma ajuta sa dobandesc experienta in domenii diferite. Pentru acest proiect am avut nevoie de cunostinte hardware, dar si software.

Cunostintele hardware au necesitat sa cunosc urmatoarele lucruri : cum sa lipesc PCB-uri, cum sa interpretez schematici, cum sa folosesc tranzistori.

Cunostintele software au insemnat acumularea de experienta in Python si in mediul Arduino, dar si folosirea interfetei UART pentru comunicarea dintre interfata grafica in Python si codul de Arduino care dadea comanda tranzistorilor.

Aplicatia functioneaza in uratorul fel:

1. Se ruleaza scriptul in Python rgb.py, in urma caruia apare urmatoarea interfata grafica...



2. Caseta PWM reprezinta un mod prin care putem sa setam intensitatea maxima pe care vrem sa o aplicam ledurilor. Valoarea maxima pentru PWM este de 255, care corespunde aplicarii unui “1” logic continuu si reprezinta intensitatea maxima, unde ledurile sunt cele mai luminoase cu putinta. Atunci cand ramane necompletata aceasta caseta, cele 3 slidere pot fi setate cu orice

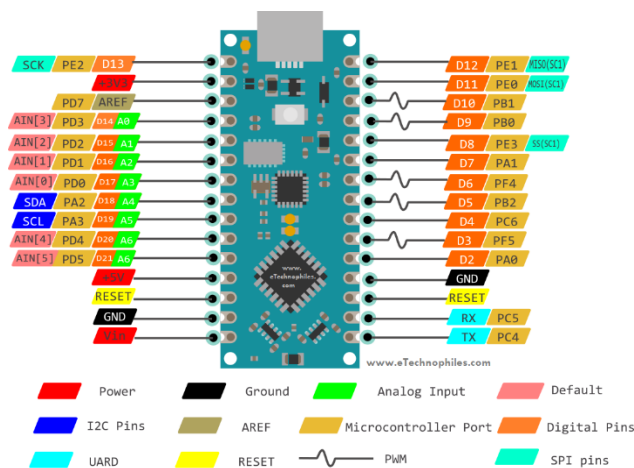
valoare între 0 și 255. Dacă setăm spre exemplu în caseta valoarea 128, atunci PWM-ul maxim aplicat pe cele 3 slidere va fi între 0 și 128. Această caracteristică este relevantă atunci când dorim o culoare statică.

3. Cele 3 butoane Fade 1, Fade 2 și HUE reprezintă 3 jocuri de lumini care ciclează prin culori, primele 2 folosindu-se strict de maparea RGB, pe când HUE se folosește și de anumite formule pentru a calcula saturarea, brightness-ul, etc.
4. Butoanul Send se folosește pentru a transmite valorile PWM de pe cele 3 slidere sau niste valori hardcodate care să corespundă celor 3 butoane.
5. Butoanul Reset se folosește pentru a reseta programul, lucru necesar deoarece ambele programe rulează într-o buclă infinită care la fiecare pas verifică valorile de pe slidere, butoane sau valorile primite pe serială (Arduino). După apăsarea lui Reset trebuie apăsat și Send.

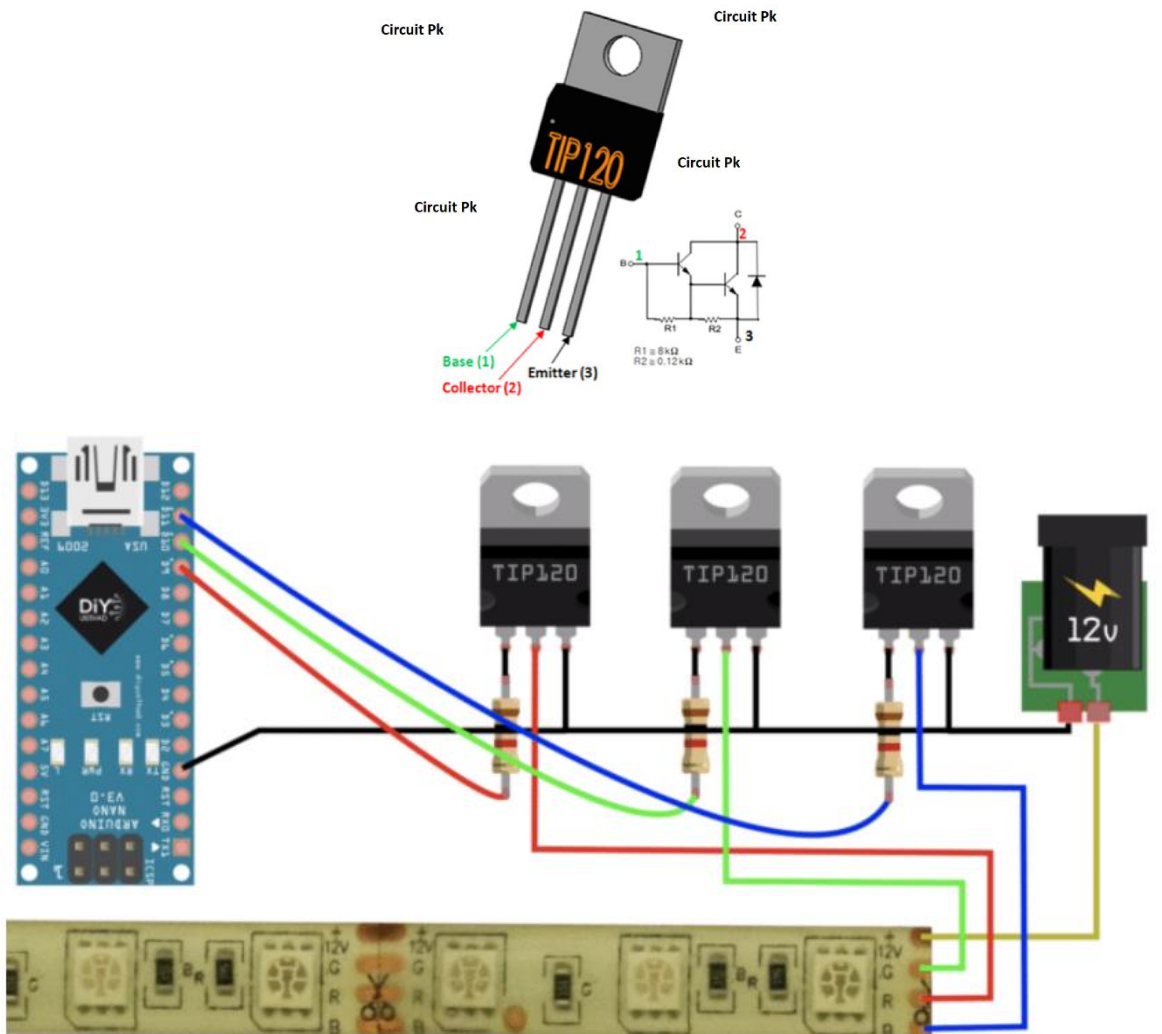
Proiectare hardware

Din punct de vedere hardware, am folosit următoarele componente:

1. Arduino Nano
2. 3 tranzistori TIP120
3. 3 rezistori
4. O bandă de leduri RGB
5. Mufa DC jack
6. Cabluri de conectare



Tranzistorul TIP120 este un tranzistor de tip NPN (adică trebuie scris 1 logic în baza), ce are particularitatea că este într-o configurație de tip Darlington, adică în aceeași carcasă avem 2 tranzistori, emitorul unuia fiind conectat la baza celuilalt. Această configurație are avantajul unui factor de amplificare mare.



Proiectare software

1. Cod Python

Mai intai se importa niste module pe care le-am folosit in aplicatie. Tkinter este pentru interfata grafica, serial pentru comunicatia seriala, iar time pentru contorizarea timpului.

Dupa aceea primul lucru pe care il facem este sa identificam un port liber, apelam o functie care ne introduce intr-un vector porturile libere si mai departe ne conectam la unul din ele.

Functia setvalue() este apelata la apasarea butonului Ok, si aceasta seteaza valoarea PWM setata in textbox-ul PWM.

Pentru valorile care se trimit pe seriala, avem un vector de 4 valori (R,G,B, respectiv buton). Fiecare slider are o functie care atribuie in acest vector valoarea PWM setata.

Slidele, butoanele si label-urile sunt facute cu functii specifice Tkinter.

Butoanele unde la apasare se scrie pe seriala sunt Fade1, Fade2, HUE si Send.

Atunci cand apasam pe unul din butoane, indiferent care, pe seriala se va trimite un string cu urmatoarea forma <commands[0], commands[1], commands[2], commands[3]>. Avem nevoie de separatori pentru a putea parsa cu usurinta in Arduino.

2. Cod Arduino

Pentru Arduino avem nevoie de biblioteca Wire, folosita pentru UART. De asemenea trebuie sa denumim si pinii folositi pentru controlul benzii de leduri, dar si diferiti vectori folositi la parsare. Functiile din loop :

1. `recvWithStartEndMarkers()` este folosita pentru a identifica punctul de start si de final al stringului mare cu ajutorul delimitatoarelor <,>. Fiecare componenta situata intre acestea este stocata intr-un vector de caractere.
2. Daca avem date noi primite pe seriala, stocam intr-un alt buffer temporar ce am stocat anterior.
3. Mai departe trebuie sa parsam datele. Folosim `strtok` si `atoi` pentru a atribui lui red, green, blue, actions
4. In functie de datele parsate, decidem ce valori PWM aplicam in baza tranzistorilor sau ce joc de lumini este folosit.