

SENZORI REZISTIVI

1. Fotorezistorul

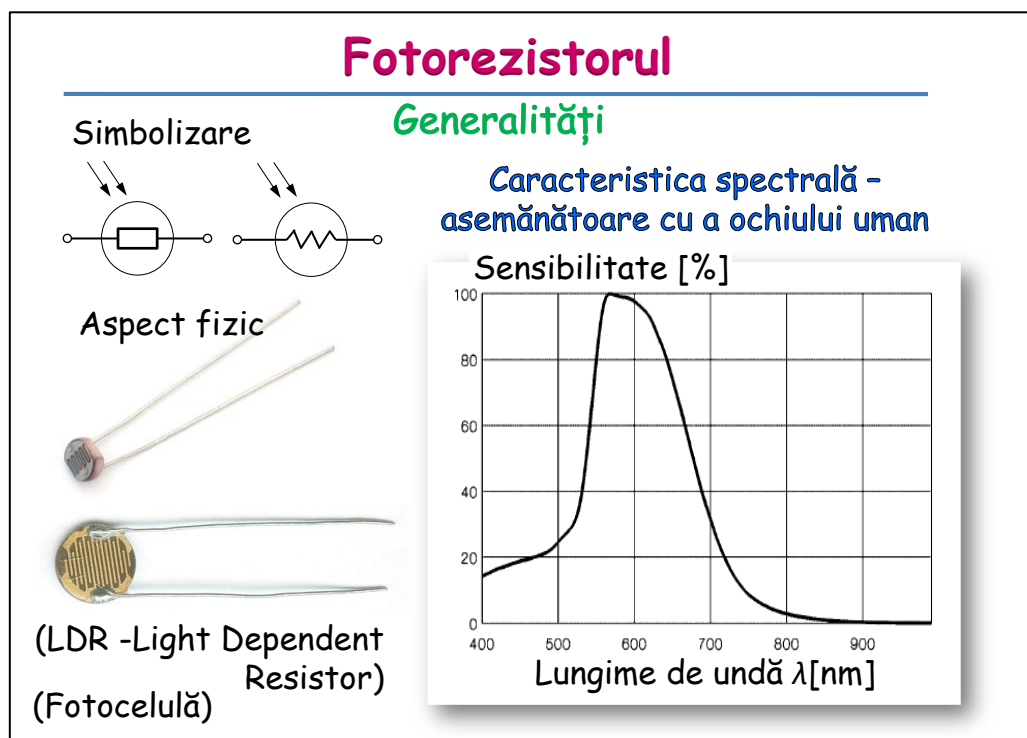
1.1 Fotorezistorul. Generalități

Cel mai simplu dispozitiv electronic folosit ca traductor de lumină este *fotorezistorul*. Acesta își modifică rezistența internă în funcție de intensitatea luminii recepționate. Se mai întâlnește și sub denumirea de *fotocelulă* sau prescurtarea *LDR* (Light Dependent Resistor).

Principiul său de funcționare are la bază efectul fotoelectric direct, prin care în materialul semiconductor se generează purtători mobili de sarcină electrică, sub acțiunea luminii incidente.

Cele mai cunoscute fotorezistoare sunt realizate din sulfură de cadmiu (CdS). Acestea au o sensibilitate spectrală comparabilă cu cea a ochiului uman. Putem astfel observa că ele acoperă o plajă largă de lungimi de undă din spectrul vizibil, și prezența un maxim al sensibilității spectrale la circa 540 nm, foarte aproape de cea a omului.

<slide LDR_general_RO.pdf>



1.2 Estimarea iluminării

Iluminarea (E_v) este o mărime fotometrică prin care se poate aprecia cantitatea de radiație luminoasă privită prin prisma observatorului uman. Unitatea de măsură a iluminării este *lux*-ul. Această mărime descrie puterea recepționată pe unitatea de arie, percepută de ochiul uman.

$$E_v = 683 \cdot V(\lambda) \cdot E_r$$

$$E_v = 683 \cdot V(\lambda) \cdot E_r$$

unde E_r este puterea recepționată pe unitatea de arie (W/m^2) iar $V(\lambda)$ este o funcție care exprimă sensibilitatea ochiului uman funcție de lungimea de undă λ a radiației luminoase.

Tabelul alăturat prezintă câteva valori tipice ale iluminării în diferite condiții de referință.

Pe de altă parte, se poate dovedi că legea de variație a rezistenței electrice R a unui fotorezistor cu iluminarea E_v este de forma:

$$R = A \cdot E_v^{-\alpha}$$

$$R = A \cdot (E_v)^{-\alpha}$$

unde A , α sunt constante de material, afectate de dispersia tehnologică (nu sunt exact aceleași pentru fiecare fotorezistor, ci iau valori într-o anumită plajă). Tipic, $\alpha = 0.7 - 0.9$ pentru fotorezistoarele din CdS.

Logaritmând ecuația de mai sus se obține o dependență liniară, descrescătoare

$$\lg R = \lg A - \alpha \cdot \lg E_v$$

$$\lg R = \lg A - \alpha \cdot \lg E_v$$

Din datele de catalog se observă că la întuneric rezistența R este cel puțin $200 \text{ k}\Omega$ și în jur de $10 \text{ k}\Omega$ la o iluminare de 10 lx .

De asemenea, fotorezistoarele *nu* au o precizie ridicată și sunt caracterizate printr-o *latență* a răspunsului (întârziere de ordinul a câteva zeci de milisecunde) la trecerea de la lumină la întuneric și invers, motiv pentru care ele nu pot detecta corect un semnal care variază rapid. Pe de altă parte, fotorezistoarele au avantajul că sunt ieftine, fiind utilizate pe scară largă la aparate foto, alarme, sisteme de iluminare automată, releu comandate cu lumină, jucării electronice, etc.

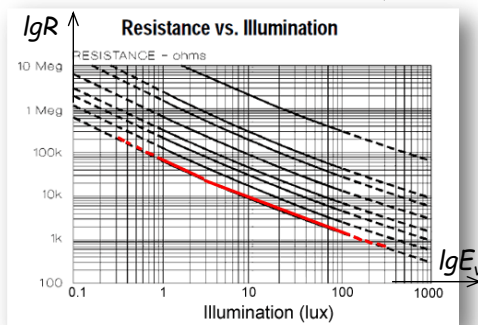
<slide LDR_R_vs_illumination_RO.pdf>

Estimarea iluminării

Rezistența fotorezistorului scade cu iluminarea

Efect fotoelectric direct

$$R = A \cdot E_v^{-\alpha} \Rightarrow \lg R = \lg A - \alpha \cdot \lg E_v$$



Condiția ambiantă	Iluminarea [lux]
Lună plină	1
Iluminare stradală	10
Locuință	30 - 300
Birou	100 - 1000
Sală de operații	10 000
Soare puternic	100 000

R - rezistența fotorezistorului

A, α - constante de material

E_v - iluminarea [lux]

(Puterea recepționată pe unitatea de arie percepută de ochiul uman)

1.3 Caracteristica experimentală

Fiind dat un fotorezistor din CdS (PDV-P8001), realizați montajul din figură. În acest caz se folosește ca sursă de lumină un circuit integrat specializat (WS2812B), numit și *neopixel*, în trei culori - roșu (R), verde (G) și albastru (B), a căror intensitate luminoasă poate fi reglată separat, în 256 de trepte. Prin combinarea celor trei culori rezultă că în total se pot obține $256^3 = 2^{24}$ nuanțe (număr reprezentat în sistemul binar pe 24 biți).

Această sursă este comandată cu ajutorul platformei Arduino, prin intermediul unui buton de acționare *PB*, de tip normal-deschis (*push-button*). Spre exemplu, în funcție de programul încărcat în micro-controller, prin apăsarea butonului *PB* se poate varia în trepte intensitatea luminoasă pentru o combinație *RGB* fixată; se poate selecta o anumită culoare, etc.

Pentru a simplifica circuitul, vom folosi aici opțiunea

`pinMode(2, INPUT_PULLUP);`

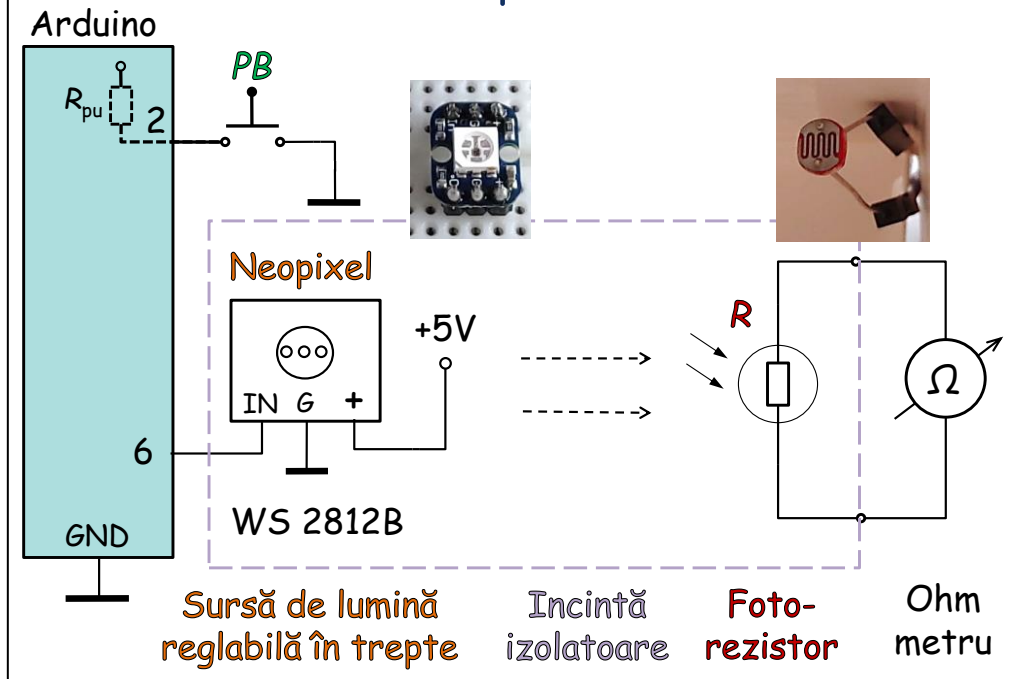
indicând astfel utilizarea rezistorului intern Arduino ($R_{pu} = 20k\Omega$ conectat la +5V) la pinul 2, nemaifiind nevoie de un rezistor extern, suplimentar pentru butonul *PB*.

Fotorezistorul, amplasat la o distanță r (de ex. 5...10 cm) pe aceeași axă cu sursa, este conectat la un ohmmetru. Întreg sistemul este introdus într-o incintă izolatoare (ex. cutie, tub cilindric tapetat la interior cu folie de aluminiu, etc.)

<slide Ardu_Ohmmeter_ImplemEx_1_RO.pdf>

Caracteristica fotorezistorului

Schema experimentală



1.4 Exemplu de program pentru implementarea sursei de lumină reglabile

Verificați încă o dată toate conexiunile, apoi conectați placa Arduino la portul USB și încărcați programul care implementează sursa reglabilă în trepte.

În acest caz se utilizează o combinație din cele trei culori R, G, B , în proporții egale, rezultând o rază de lumină de culoare albă. Intensitatea luminoasă este specificată de parametrul *brightness*, care se incrementează cu zece unități la fiecare apăsare a butonului PB. Porniți monitorul serial și acționați butonul de comandă. La fiecare apăsare, măsurați rezistența fotorezistorului. Înregistrați rezultatele într-un tabel. Reprezentați grafic dependența rezistenței de treapta de putere (exprimată în procente). Comentați rezultatele obținute.

<slide LDR_CarExp_SW0_RO.pdf>

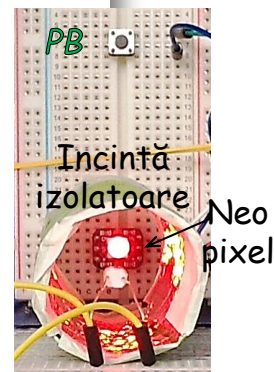
Fotorezistorul. Caracteristica experimentală

Program pentru implementarea sursei de lumină

```
#include <Adafruit_NeoPixel.h>
Adafruit_NeoPixel strip = Adafruit_NeoPixel(1, 6, NEO_GRB + NEO_KHZ800);
int buttonPin = 2;
int brightness = 0;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  strip.begin();
  strip.show();
  Serial.begin(9600);
  // "Light intensity level:"
  Serial.println("Treapta de intensitate luminoasa:");
}

void loop() {
  if (digitalRead(buttonPin) == LOW) {
    strip.setPixelColor(0, brightness, brightness, brightness);
    strip.show();
    Serial.println(brightness);
    brightness += 10;
    if (brightness > 255) {
      brightness = 0;
    }
  }
  delay(200);
}
```

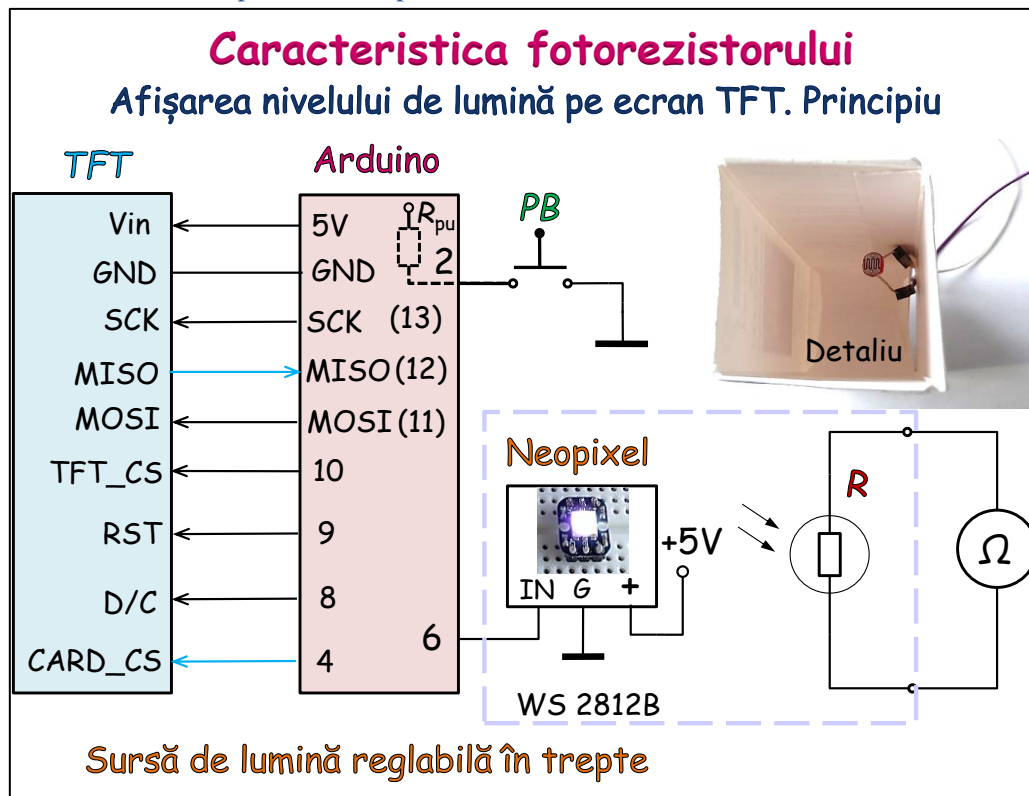


Fotorezistor

1.5 Adăugarea unui ecran *LCD* pentru afișarea nivelului de lumină

Conectați ecranul *TFT* ca în figură, pentru a putea afișa local nivelul de lumină de la sursă.

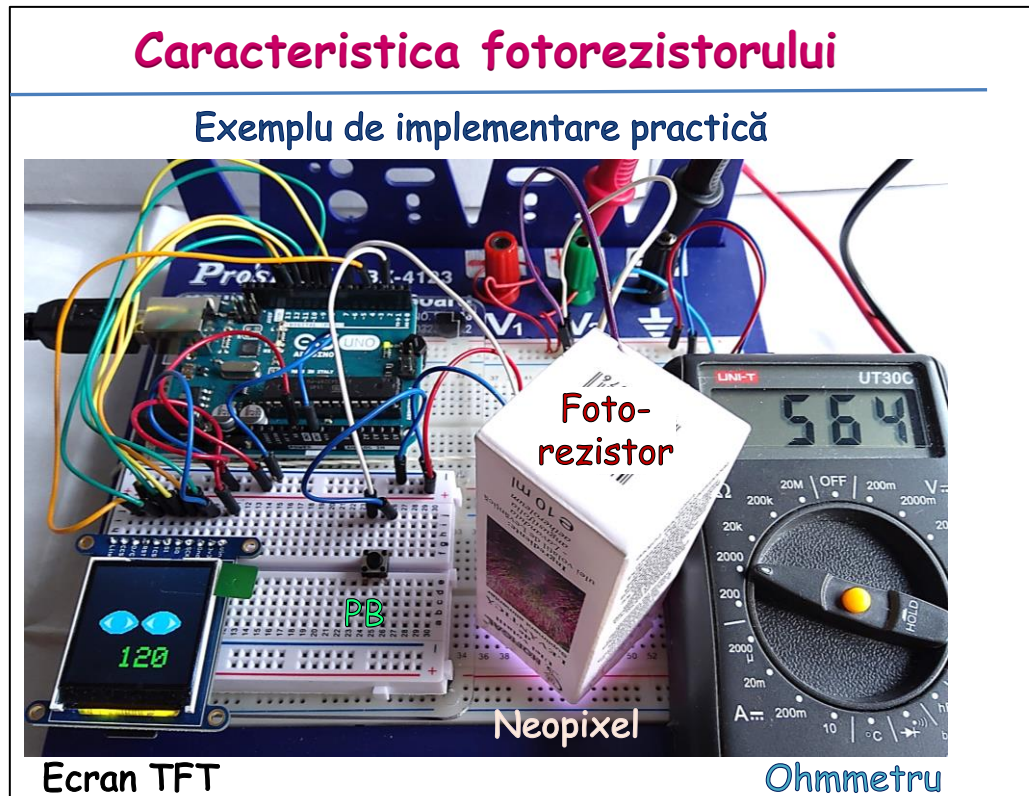
<slide LDR_CarExp_TFT1_RO.pdf>



1.6 Caracteristica fotorezistorului. Exemplu de implementare practică

Aici se prezintă un exemplu de implementare practică, care include afișarea treptei de lumină.

<slide LDR_CarExp_TFT2_RO.pdf>



1.7 Caracteristica fotorezistorului. Exemplu de program cu afișare pe ecran TFT

Programul pentru acest experiment este prezentat în următoarele două slide-uri.

Întrucât se folosește dispozitivul TFT, pentru simplificarea programului, vom construi (ex. cu "Notepad"-ul) un fișier separat în care vom specifica toate configurările (librăriile utilizate, definițiile, respectiv obiectele software create). Acest fișier îl vom salva cu extensia *.h (ex. configdevs.h) în același director cu programul principal Arduino. El va apare ca o filă suplimentară la deschiderea programului și va trebui specificat prin instrucțiunea #include configdevs.h la începutul acestuia.

<slide LDR_CarExp_SW1a_RO.pdf>

Fotorezistorul. Caracteristica experimentală

Program pentru implementarea sursei de lumină cu afișare TFT

Configurări ecran TFT, neopixel

```

sursa_neopixel_TFT_s  configdevs.h
#include <Adafruit_GFX.h>          // Core graphics library
#include <Adafruit_ST7735.h>        // Hardware-specific library
#include <SdFat.h>                  // SD card & FAT filesystem library
#include <Adafruit_SPIFlash.h>      // SPI / QSPI flash library
#include <Adafruit_ImageReader.h>  // Image-reading functions
#include <Adafruit_NeoPixel.h>

#define USE_SD_CARD
#define SD_CS 4 // SD card select pin
#define TFT_CS 10 // TFT select pin
#define TFT_DC 8 // TFT display/command pin
#define TFT_RST 9 // Or set to -1 and connect to Arduino RESET pin
SdFat SD; // SD card filesystem
Adafruit_ImageReader reader(SD); // Image-reader object, pass in SD filesys
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
Adafruit_NeoPixel strip = Adafruit_NeoPixel(1,6,NEO_GRB + NEO_KHZ800);
int32_t width = 0, // BMP image dimensions
height = 0;

```

1.8 Caracteristica fotorezistorului. Exemplu de program cu afișare pe ecran TFT

În figura alăturată se prezintă un exemplu de program principal, care include afișarea locală a treptei de intensitate luminoasă.

Verificați încă o dată toate conexiunile, apoi conectați placa Arduino la portul USB și încărcați programul.

Ce concluzie desprindeți din acest experiment? Cum credeți că s-ar putea îmbunătăți montajul de mai sus astfel încât să se reducă erorile de măsurare?

<slide LDR_CarExp_SW1b_RO.pdf>

Fotorezistorul. Caracteristica experimentală

Program pentru implementarea sursei de lumină cu afișare TFT

```
#include "configdevs.h"
int buttonPin = 2;
int brightness = 0;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  strip.begin(); strip.show();
  tft.initR(INITR_144GREENTAB);
  if(!SD.begin(SD_CS, SD_SCK_MHZ(10))) {
    for(;;);
  }
  Serial.println(F("OK!"));
  tft.fillScreen(0);
  tft.setCursor(0, 20);
  tft.setTextSize(2);
  tft.setTextWrap(false);
  tft.setTextColor(ST7735_CYAN);
  tft.print("Treapta de"); // Light intensity
  tft.setTextColor(ST7735_MAGENTA);
  tft.setCursor(0, 60);
  tft.print("intensitate");
  tft.setTextColor(ST7735_GREEN);
  tft.setCursor(0, 90);
  tft.println(" luminoasa");
  delay(5000);
  tft.fillScreen(0); //BLACK
}

void loop() {
  if (digitalRead(buttonPin) == LOW) {
    strip.setPixelColor(0, brightness, brightness, brightness);
    strip.show();
    tft.setRotation(0);
    tft.fillScreen(0);
    tft.setCursor(50, 90);
    tft.setTextSize(3);
    tft.setTextColor(ST7735_GREEN);
    tft.print(brightness);
    brightness += 10;
    if (brightness > 255) {
      brightness = 0;
    }
  }
  else {
    tft.setRotation(2);
    reader.bmpDimensions("/eyes2.bmp", &width, &height);
    tft.fillRect(64-(width/2), 64-(height/2), width, height, ST7735_BLACK);
    reader.drawBMP("/eyes2.bmp", tft, 64-(width/2), 64-(height/2));
  }
  delay(300);
}
```





1.9 Videoclip 1 - A photoresistor test

*Wonders by Alex-Productions | <https://onsound.eu/>
Music promoted by <https://www.free-stock-music.com>
Creative Commons Attribution 3.0 Unported License
https://creativecommons.org/licenses/by/3.0/deed.en_US*

YouTube

<https://youtu.be/OnfjBsVVKsY>

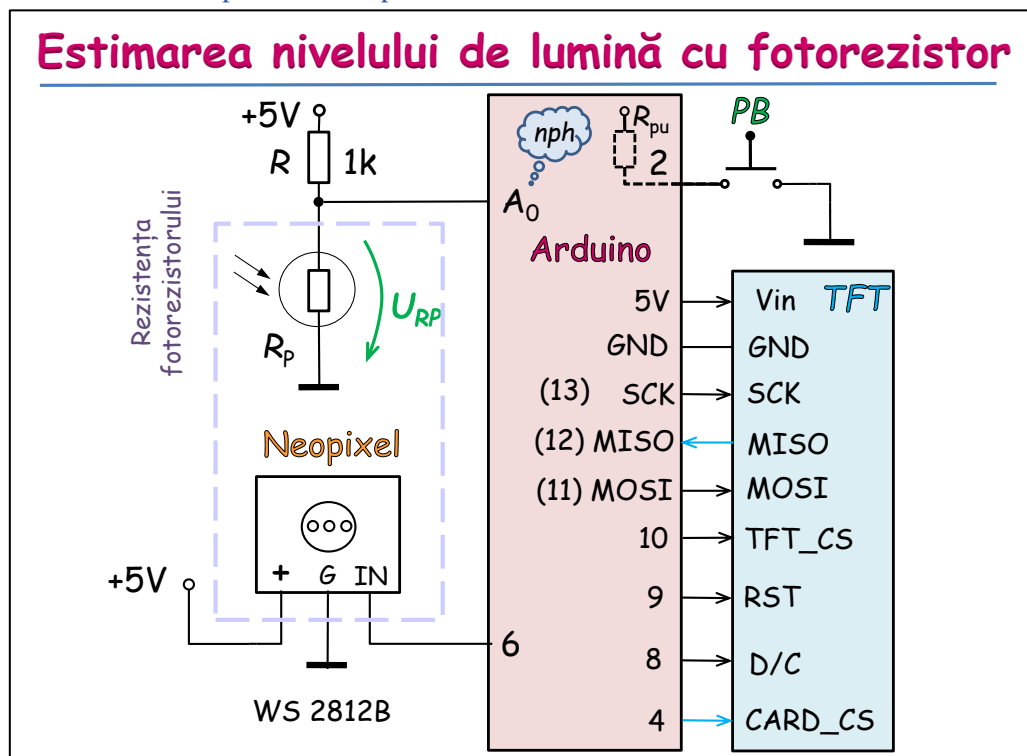
Embed

```
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/OnfjBsVVKsY" title="YouTube video  
player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;  
encrypted-media; gyroscope; picture-in-picture; web-share"  
allowfullscreen></iframe>
```

1.10 Aplicație. Estimarea nivelului de lumină ambiantă cu fotorezistor. Schema de principiu

Conectați fotorezistorul printr-un divizor rezistiv la pinul A0 Arduino, ca în figură. Vom scrie un program prin care vom măsura tensiunea la bornele fotorezistorului (care depinde de iluminare) și vom afișa în timp real calitativ starea luminozității ambiante (ex. "întuneric", "lumină slabă", etc.)

<slide LDR_Ap_TFT1_RO.pdf>



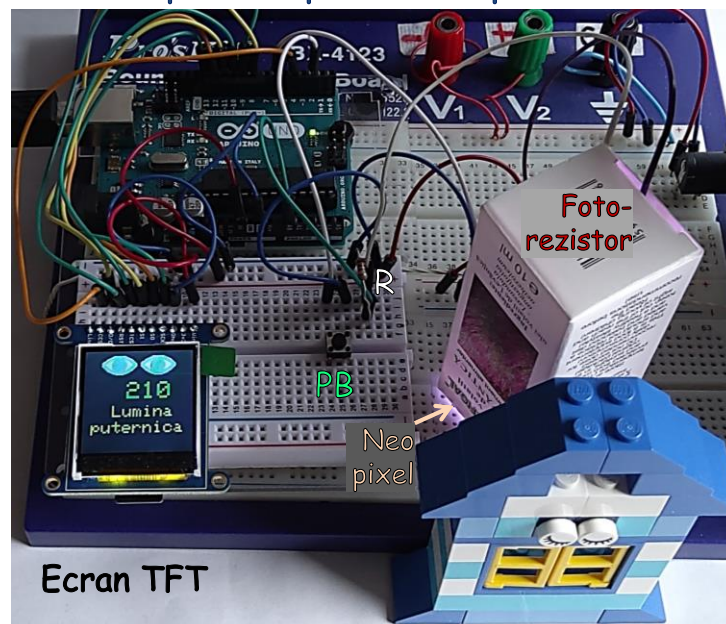
1.11 Aplicație. Estimarea nivelului de lumină ambiantă cu fotorezistor. Exemplu de implementare

Un exemplu de amplasare a elementelor circuitului este prezentat în figura de mai sus.

<slide LDR_Ap_TFT2_RO.pdf>

Estimarea nivelului de lumină cu fotorezistor

Exemplu de implementare practică



1.12 Aplicație. Estimarea nivelului de lumină ambiantă cu fotorezistor. Exemplu de program. Configurări și setup()

Verificați încă o dată toate conexiunile, apoi conectați placa Arduino la portul USB și încărcați programul care implementează această aplicație.

<slide LDR_TFT_Ap_SW1_RO.pdf>

Fotorezistorul. Estimarea nivelului de lumină

Exemplu de program Configurări și setup()

```

#include <Adafruit_GFX.h>           // Core graphics library
#include <Adafruit_ST7735.h>        // Hardware-specific library
#include <SdFat.h>                  // SD card & FAT filesystem library
#include <Adafruit_SPIFlash.h>      // SPI / QSPI flash library
#include <Adafruit_ImageReader.h>   // Image-reading functions
#include <Adafruit_NeoPixel.h>

#define USE_SD_CARD
#define SD_CS_ 4 // SD card select pin
#define TFT_CS 10 // TFT select pin
#define TFT_DC 8 // TFT display/command pin
#define TFT_RST 9 // Or set to -1 and connect to Arduino RESET pin
SdFat SD; // SD card filesystem
Adafruit_ImageReader reader(SD); // Image-reader object, pass in SD filesystem
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
Adafruit_NeoPixel strip = Adafruit_NeoPixel(1, 6, NEO_GRB + NEO_KHZ800);
int32_t width = 0, // BMP image dimensions
height = 0;

```



```

#include "configdevs.h"
#define SAMPLES 2500
int buttonPin = 2;
int brightness = 0;
int nph;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  strip.begin();
  strip.show();
  tft.initR(INITR_144GREENTAB);
  if(!SD.begin(SD_CS, SD_SCK_MHZ(10))) {
    for(;;);
  }
  Serial.println(F("OK!"));
  tft.fillScreen(0);
  tft.setCursor(0, 20);
  tft.setTextSize(2);
  tft.setTextWrap(false);
  tft.setTextColor(ST7735_CYAN);
  //tft.print(" Lumina");
  tft.print(" Ambient");
  tft.setTextColor(ST7735_MAGENTA);
  tft.setCursor(0, 60);
  //tft.print(" ambianta");
  tft.print(" light");
  tft.setTextColor(ST7735_GREEN);
  tft.setCursor(0, 90);
  //tft.println("estimata...");
  tft.println("estimate...");
  delay(5000);
  tft.fillScreen(0);
}

```

1.13 Aplicație. Estimarea nivelului de lumină ambiantă cu fotorezistor. Exemplu de secvență loop()

Pentru "netezirea" datelor măsurate la intrarea A0 vom utiliza și aici medierea mai multor eșantioane.

Ce puteți spune despre limitele domeniilor de iluminare? Propuneți soluții pentru îmbunătățirea experimentului.

<slide LDR_TFT_Ap_SW2_RO.pdf>

Fotorezistorul.Estimarea nivelului de lumină

Exemplu de program loop()

```
void loop(){
  unsigned long n = 0;
  unsigned int i;
  tft.setRotation(0);
  if (digitalRead(buttonPin)==LOW) {
    strip.setPixelColor(0,brightness,brightness,brightness);
    strip.show();
    tft.fillRect(0, 70,128,88,ST7735_BLACK);
    tft.setCursor(50, 40);
    tft.setTextSize(3);
    tft.setTextColor(ST7735_GREEN);
    tft.print(brightness);
    brightness += 10;
    if(brightness > 255){
      brightness=0;
    }
  }
  else{
    for (i=0; i<SAMPLES; i++){
      n += analogRead(A0);
    }
    np = n/SAMPLES;
    tft.setCursor(10, 70);
    tft.setTextSize(2);
    if(np > 700){
      Serial.println(" - Dark");
      tft.fillRect(0, 70,128,88,ST7735_BLACK);
      tft.setTextColor(ST7735_BLUE);
      //tft.print("Intuneric");
      tft.print(" Dark");
    }
    else if ((np <= 700) && (np >600)){
      tft.fillRect(0, 70,128,88,ST7735_BLACK);
      tft.setTextColor(ST7735_MAGENTA);
      tft.print(" Dim");//tft.print(" Lumina");
      tft.setCursor(40, 90);tft.print("light");//tft.print("slaba");
    }
    else if ((np <= 600) && (np >400)){
      tft.fillRect(0, 70,128,88,ST7735_BLACK);
      tft.setTextColor(ST7735_CYAN);
      tft.print(" Optimal");//tft.print(" Lumina");
      //tft.setCursor(40, 90);//tft.print("optima");
      tft.setCursor(10, 90);tft.print("lighting");
    }
    else if ((np <= 400) && (np >300)){
      tft.fillRect(0, 70,128,88,ST7735_BLACK);
      tft.setTextColor(ST7735_YELLOW);
      tft.print(" Bright");//tft.print(" Lumina");
      tft.setCursor(10, 90);
      tft.print(" light");//tft.print("puternica");
    }
    else{
      tft.fillRect(0, 70,128,88,ST7735_BLACK);
      tft.setTextColor(ST7735_RED);
      tft.println(" Very");//tft.println(" Lumina");
      tft.println(" bright");//tft.println(" foarte");
      tft.print(" light");//tft.print(" puternica");
    }
  }
  tft.setRotation(2);
  reader.bmpDimensions("/eyes2.bmp", &width, &height);// efect de clipire
  tft.fillRect(64-(width/2), 110-(height/2),width,height,ST7735_BLACK);
  reader.drawBMP("/eyes2.bmp", tft, 64-(width/2), 110-(height/2));
}
delay(200);
```



1.14 Videoclip 2 - A photoresistor application

Soundtrack by Alex-Productions | <https://onsound.eu/>
Music promoted by <https://www.free-stock-music.com>
Creative Commons Attribution 3.0 Unported License
https://creativecommons.org/licenses/by/3.0/deed.en_US

YouTube

<https://youtu.be/1D-JekiGLto>

Embed

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/1D-JekiGLto" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
```