

Realizarea unui VU-Metru cu led-uri Tower

Miron Cezar Andrei

Pentru realizarea proiectului, am folosit următoarele componente :

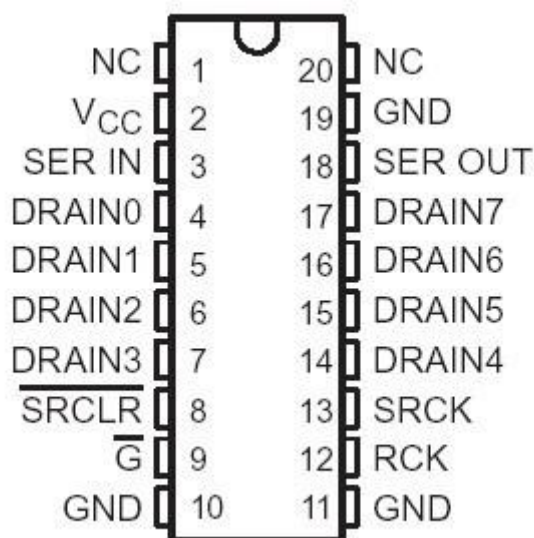
- Arduino Uno
- 8 leduri + 8 rezistori 220ohm
- Shift Register pe 8 biți TPIC6B595
- Breadboard + fire de interconectare
- Potențiometrul 10k
- Condensator de 10uF

Etape de proiectare :

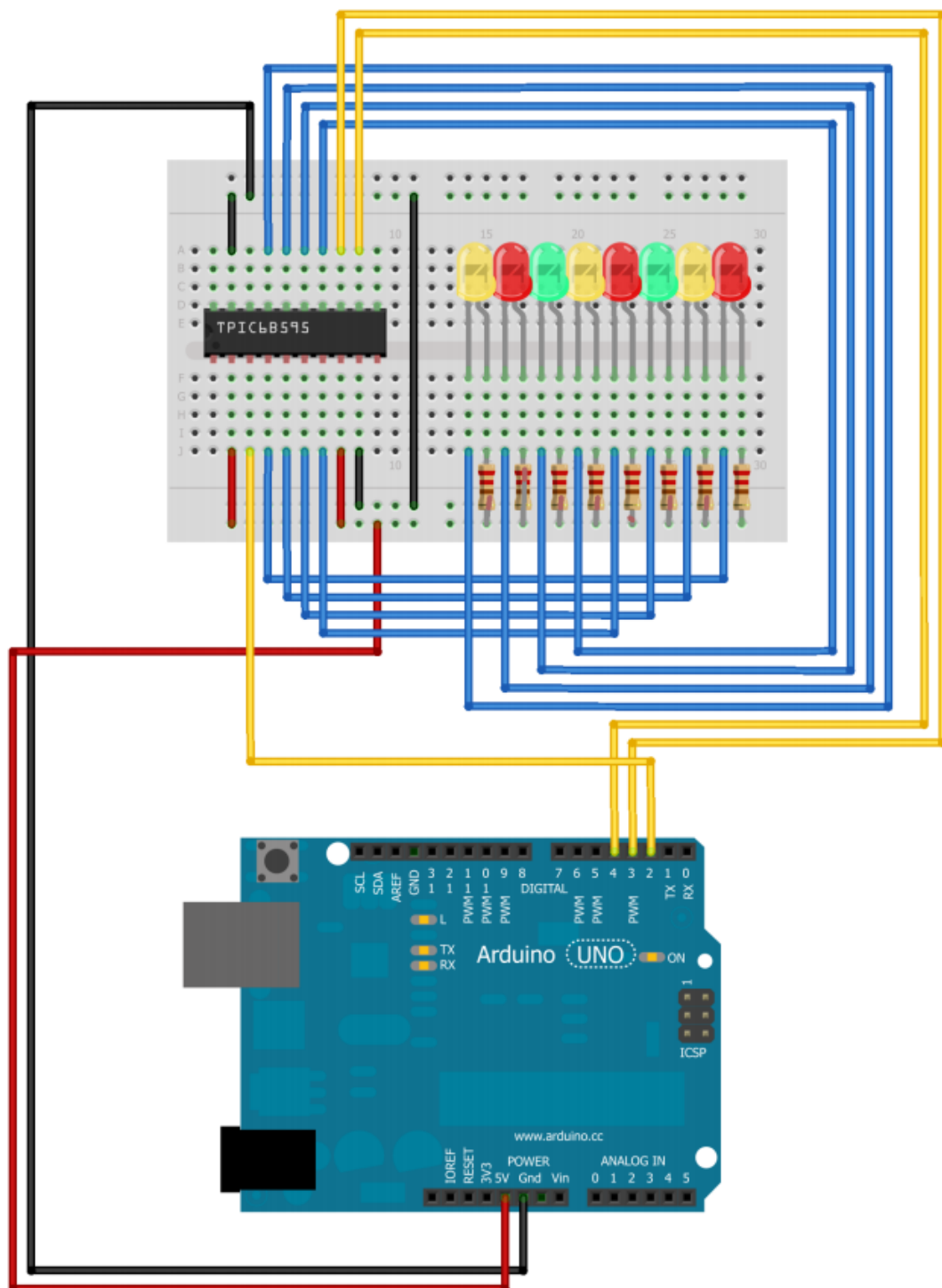
1. Realizarea comunicației între led-uri și placuța Arduino

Pentru a realiza această comunicare, am ales să folosim un registru de shiftare de 8 biți ce va fi utilizat ca un registru cu încărcare serială. Astfel liniile de comandă a ansamblului de 8 leduri va fi redus de la 8 la 3. Comanda de aprindere a ledurilor nu va fi dată de pinii plăcii de dezvoltare ci de liniile de ieșire a registrului de shiftare. Ca registru de shiftare am ales circuitul TPIC6B595. Liniile de ieșire a circuitului care vor comanda cele 8 leduri sunt următoarele :

- DRAIN0-7 -> linii de comandă pentru leduri
- SER IN -> dataPin
- SER OUT -> neconectat
- SRCK -> clockPin
- RCK -> latchPin
- /SRCLR -> inactiv – conectat la Vcc
- /G -> activat întotdeauna – la masă



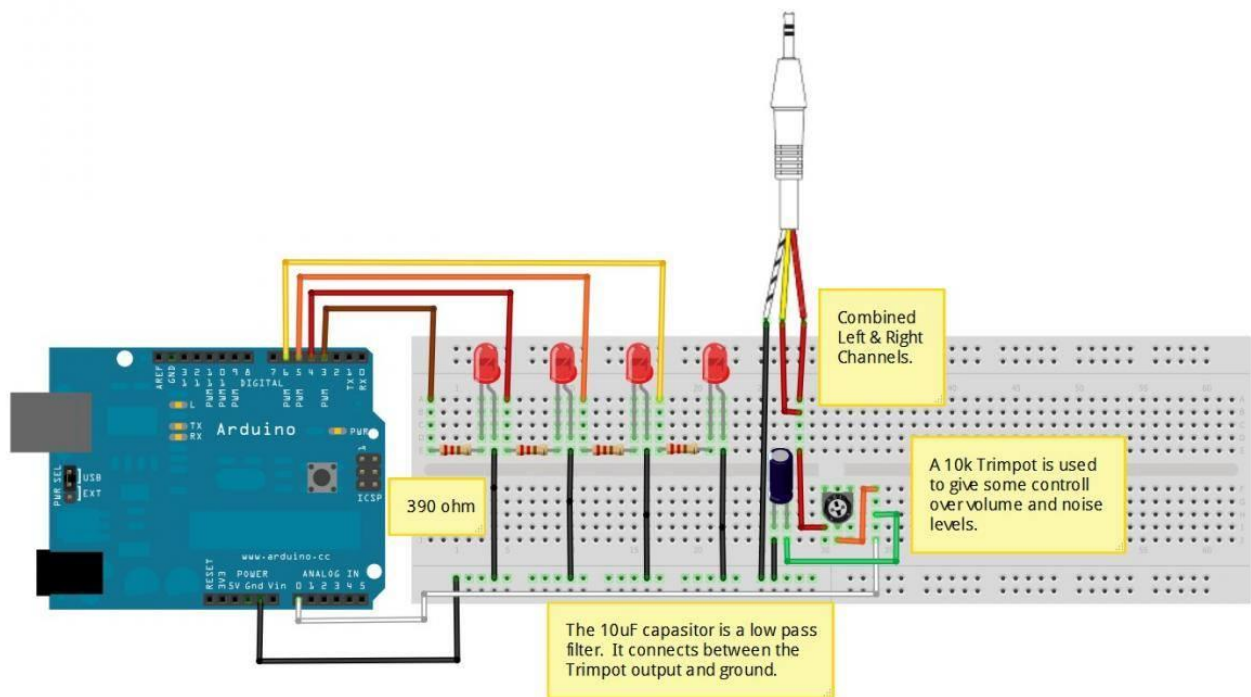
Schema de interconectare este următoarea :



2. Conectarea și comunicarea proiectului folosind ca intrare audio o mufă Jack

Deși initial am dorit să utilizăm un microfon pentru realizarea VU – metrului, am ales să folosim semnal (audio) primit printr-o mufă jack de la un device. Principalul motiv pentru care am făcut această alegere a fost costul ridicat al unui microfon sensibil.

Pentru conectarea mufei jack, am folosit următoarea schemă :



Pentru că semnalul audio primit prin mufa jack este sinusoidal, $\sim (-1.5\text{v} ; 1.5\text{v})$ pentru o bună funcționare a proiectului avem nevoie doar de tensiunile pozitive; de aceea, nu se putea citi direct valoarea semnalului pe un pin analogic. Pentru rezolva această problemă am creat un filtru trece jos. Astfel procesăm doar semnalul pozitiv. (0v – 1.5v)

Având în vedere faptul că ADC-ul plăcuței Arduino UNO interpretează tensiuni între 0 și 5V, pentru o precizie mai mare a VU-metrului se poate folosi un amplificator de semnal.

3. Programarea VU-metrului

```
const int latchPin = 4;
const int clockPin = 3;
const int dataPin = 2;

int analogPin = 5;
int val = 0;

void setup() {
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);

  Serial.begin(9600);
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, MSBFIRST, 0);
  digitalWrite(latchPin, HIGH);
}

void loop() {

  val = analogRead(analogPin);
  Serial.println(val);

  if( (val) > 90 )
  {
    digitalWrite(latchPin, LOW);

    shiftOut(dataPin, clockPin, MSBFIRST, 255);
    digitalWrite(latchPin, HIGH);
  }

  else if( (val) > 80 )
  {
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, 127);
    digitalWrite(latchPin, HIGH);
  }

  else if( (val) > 70 )
  {
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, 63);
    digitalWrite(latchPin, HIGH);
  }

  else if( (val) > 60 )
  {
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, 31);
    digitalWrite(latchPin, HIGH);
  }

  else if( (val) > 50 )
  {
    digitalWrite(latchPin, LOW);
```

```

    shiftOut(dataPin, clockPin, MSBFIRST, 15);    }
    digitalWrite(latchPin, HIGH);                delay(40);
}
                                                }

else if( (val) > 40 )
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, 7);
    digitalWrite(latchPin, HIGH);
}

else if( (val) > 30 )
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, 3);
    digitalWrite(latchPin, HIGH);
}

else if( (val) > 20 )
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, 1);
    digitalWrite(latchPin, HIGH);
}

else if( (val) > 3 )
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, 0);
    digitalWrite(latchPin, HIGH);
}

```