

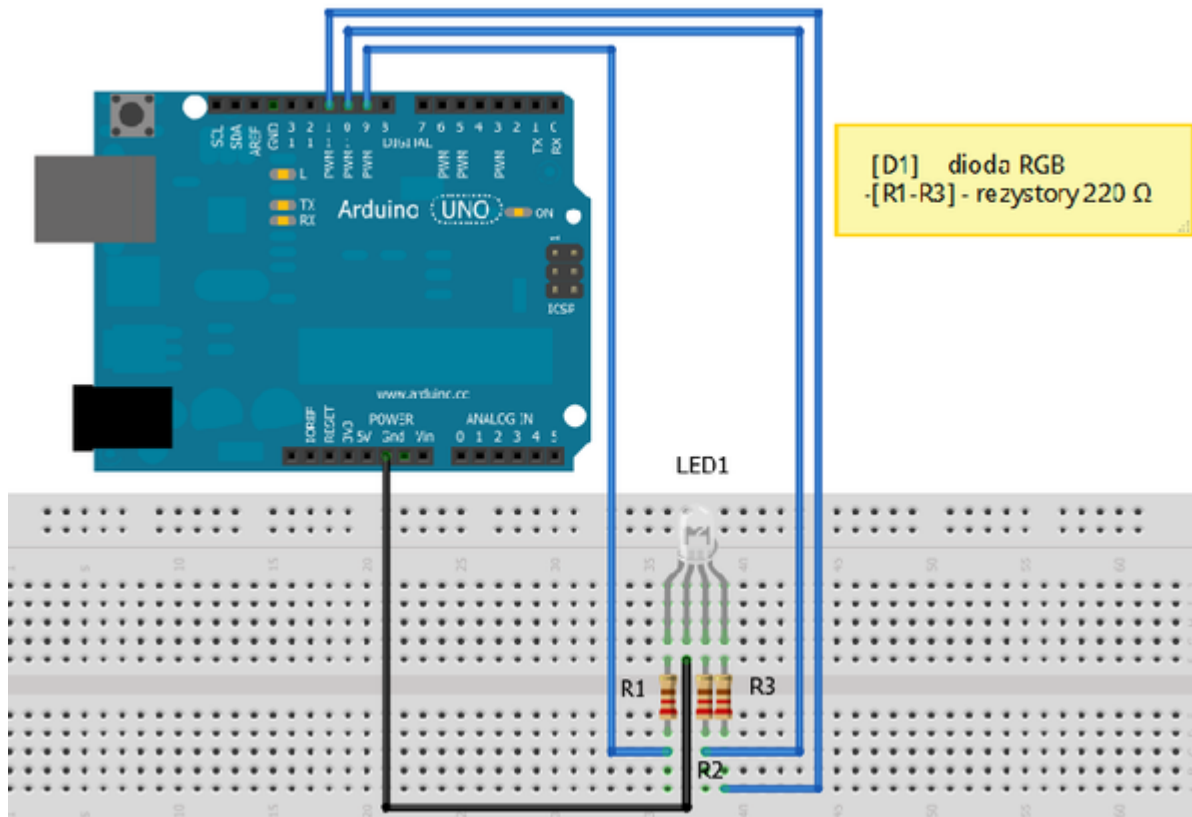


Nazwa implementacji: Sterowanie układem z interfejsu Arduino IDE - b

Autor: Krzysztof Bytow

Opis implementacji: Komunikacja modułu-interfejsu z komputerem PC na przykładzie sterowania jasnością i kolorami diody RGB.

Schemat połączeń – Sterowanie diodą RGB ze wspólną katodą z poziomu interfejsu Arduino IDE.



Uczeń/Uczennica po zestawieniu połączeń zgłasza nauczycielowi gotowość do sprawdzenia układu i wszystkich połączeń.



<- oznaczenie kodem barwnym rezystora 220 Ω



opis wyprowadzeń diody RGB ze wspólną katodą ->

Kod implementacji

```
char tablica[18];           // tworzymy tablicę przechowującą ciąg znaków

void setup()
{
  Serial.begin(9600);       // ustawienie prędkości komunikacji
  Serial.flush();          // czyści nam serial port ze wszystkich danych
  for(int i=9;i<=12;i++)   // w pętli ustawiamy piny od 9 do 12 jako wyjścia i ustawiamy stan niski
  {
    pinMode(i,OUTPUT);
    digitalWrite(i,LOW);
  }
}

void loop()
{
  if(Serial.available(>0) // wyrażenie warunkowe
  {
    int i=0;
    delay(100);           // czekaj wyznaczony czas
```

2





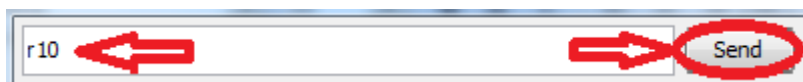
```
int znak=Serial.available();           // http://arduino.cc/en/Serial/Available
if (znak>15)                           // instrukcja warunkowa
{
znak=15;
}
while (znak-->0)                       // pętla while
{
tablica[i++] = Serial.read();          // zapis do tablicy wartości odczytanych
}
wyswietl(tablica);                     // wywołanie funkcji wyswietl
}
}
void wyswietl(char* k)
{
Serial.print ("Dioda rgb ustawiona na wartosci: "); //wyświetlenie tekstu
Serial.println(k);                     //wyświetlenie tekstu
char* parameter;
parameter = strtok (k, " ,");         // strtok - konwertuje wartość zapisaną w łańcuchu znaków w dowolnym
while (parameter != NULL)             // systemie liczbowym do postaci liczby typu całkowitego,
{
rgb(parameter);                       // wywołanie funkcji rgb
parameter = strtok (NULL, " ,");
}
for(int x=0;x<=15;x++)                // w pętli zwiększamy wartość x od 0 do 15
{
tablica[x]='0';
}
Serial.flush();                       //http://arduino.cc/en/Serial/Flush
}
void rgb(char* k)
{
if((k[0] == 'r') || (k[0] == 'R'))     // wyrażenie warunkowe
{
int wartosc = strtol(k+1, NULL, 10);   // http://www.thinkage.ca/english/gcos/expl/c/lib/strtol.html
wartosc = constrain(wartosc, 0, 255);  // ograniczenie liczby w przedziale
analogWrite(9,wartosc);                // PWM – modulacja szerokości impulsu, (pin, wartość od 0 do 255)
Serial.print("Czerwona ustawiona na: "); //wyświetlenie tekstu
Serial.println(wartosc);               //wyświetlenie tekstu
}
}
}
```

Następnie w Arduino IDE wybieramy ikonkę monitor portu szeregowego:

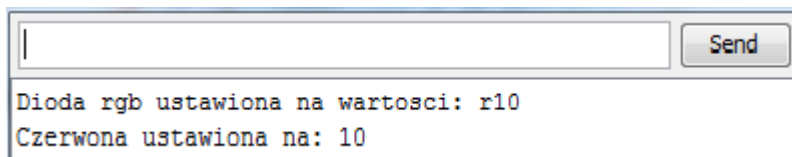


W terminalu wpisujemy wartości od 0 do 255, wskazując kolor diody RGB, który ma zaświecić (odpowiednie litery zostały zdefiniowane we fragmencie kodu `if((k[0] == 'r') || (k[0] == 'R'))`). Analizując dalej kod, widzimy, który pin Arduino będzie zmieniał wartość `analogWrite(9,wartosc)` :





Efekt obserwujemy w jasności świecenia diody i w oknie terminala:



Dodatkowe informacje dotyczące komunikacji dostępne na stronie:

<http://arduino.cc/en/Reference/Serial>

Aby sterować w pełni diodą RGB należy rozbudować kod o kolejne dwa kolory.

