

Transistorstyrd RGB-lampa

**Innehållsförteckning**

[Uppgift: Skapa & Bygga ett Inbyggt system](#_7dc8p3x3td7m)

[Utförandet, Design & Utmaningar](#_io2zl3kzpe51)

[Design & schema](#_j8grfgig3jbd)

[Komponentlista](#_d128zlqzvuxt)

[Utförandet & byggnationen](#_bw4isbti1prv)

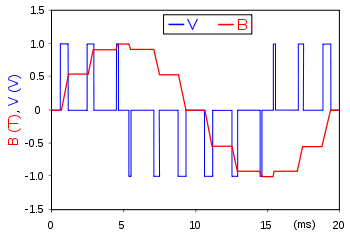
# Uppgift: Skapa & Bygga ett Inbyggt system

Jag valde att bygga en RGB-lampa styrd med hjälp av fototransistorer. Lampans färgvärde påverkas av ljusförhållanden i dess omgivning. Med av tre filter (Röd,grön och blå plastfilm) som man placerar ovanför transistorerna kan man “styra” färgen på lampan. Då plastfilmerna inte är heltäckande 360 grader filter så kan de inte blockera ljus med en viss våglängd till 100%

# Utförandet, Design & Utmaningar

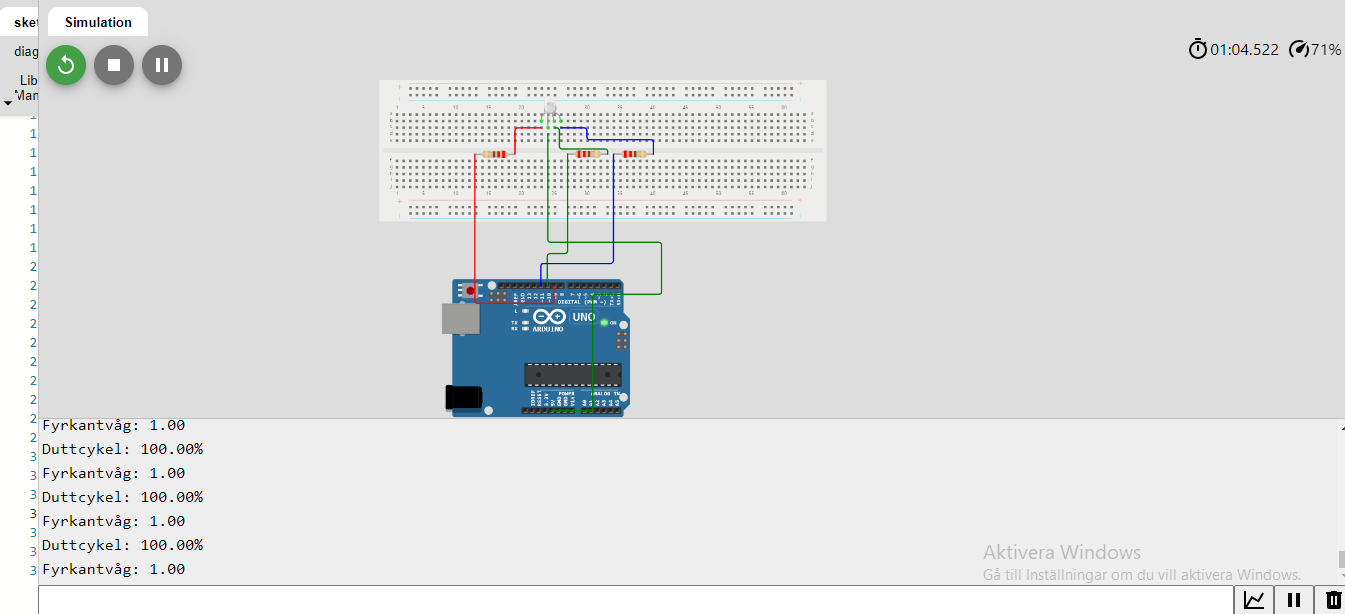
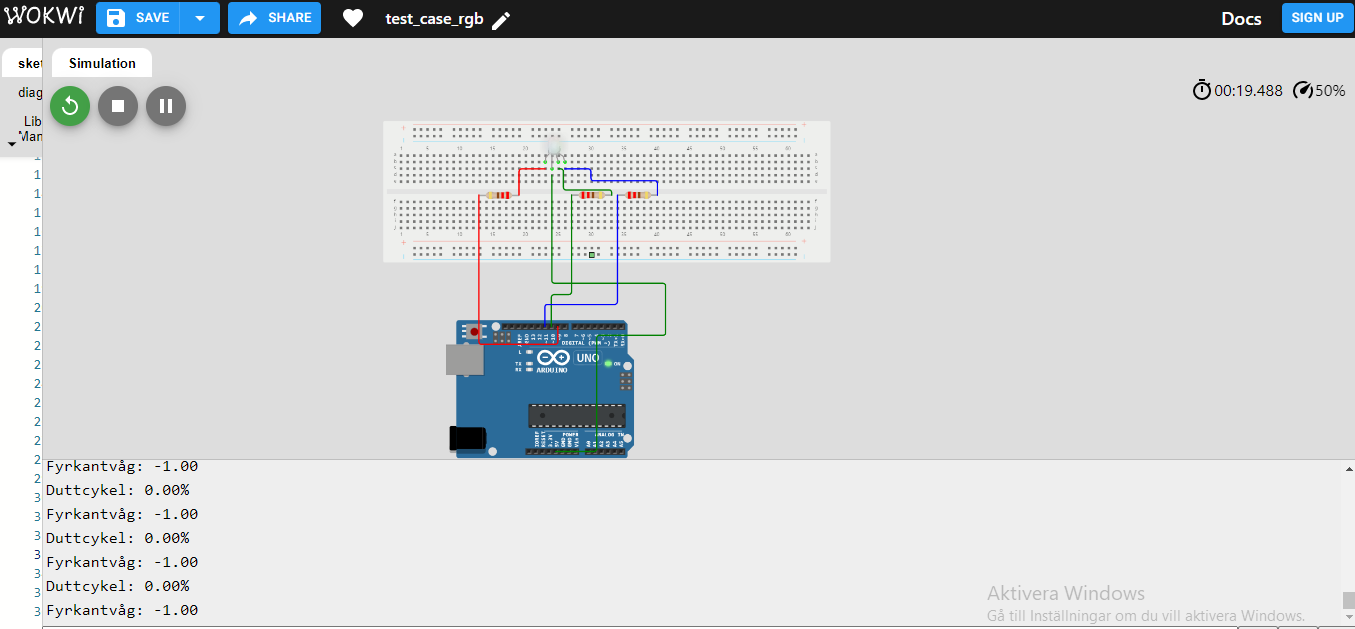
Eftersom en Arduino Uno inte kan ändra dess utgångsspänning  
Från 5V stiftet. Måste vi använda oss av en teknik som

heter “pulse-width-modulation” för att dämpa ljusstyrkan på LED lamporna.  
PWM omvandlar utgångsstiftet och gör så att spänningspulserna blir hög och låg under en bestämd tid. Vi kan förstå detta bättre genom att jämföra

det med hur det mänskliga ögat fungerar. När man tittar på en film så kollar du egentligen på flera sekvenser av tusentals bilder. Detta skapar illusionen av att du ser rörelser på en skärm. PWM har en liknande princip när den skiftar stiftet mellan hög (HIGH) och låg (LOW), då är det som om du skiftar spänningen så som en film visar dig tusentals sekvenser av flera bilder under n antal sekunder.   
  
Källa: [Pulse Width modulation](https://en.wikipedia.org/wiki/Pulse-width_modulation)

Den procentuella tiden som stiftet befinner sig i ett HÖGT läge kallas

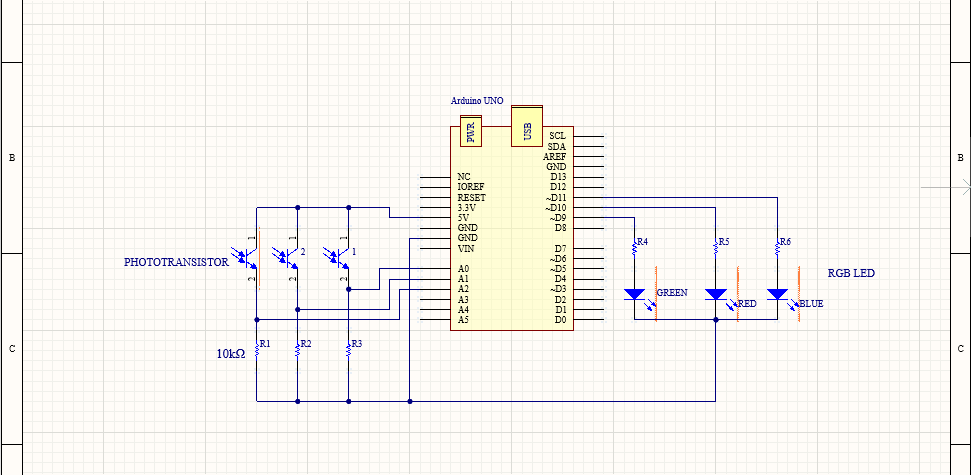
för arbetscykel. Då är det som om att du ändrar spänningen. När stiftet befinner sig 50% av tiden under HIGH hälften av tiden och LOW under andra hälften av tiden då är arbetscykeln 50%, en lägre arbetscykel innebär att ljusstyrkan på lampan dämpas.



Dessa två bilder beskriver en enkel simulering av konceptet arbetscykel. I första bilden är arbetscykeln 0% i den andra är arbetscykeln 100%.   
Anledningen till denna jämna skiftning i spänning är precis som diskuterats tidigare, i detta fall (Och med tanke på att det är ett mycket enklare projekt) har vi inte använt oss av PWM, därför får vi antingen 100 % eller 0%, då spänningen är i princip densamma under en fast period. Och inga yttre faktorer som t.ex. Ljusförhållanden påverkar späninngen.   
**Simulerings-exempel-av-arbetscykel-projektet består av**

* 1 Arduino Uno
* 3 Resistorer
* 1 breadboard.
* 1 RGB-lampa

## Design & schema

[](https://drive.google.com/file/d/1wTl8NiaCuCHZH0FQVpVCWgOSsozaAZ4X/view?usp=share_link)

### [Kretsschema PDF](https://drive.google.com/file/d/1KSQ5ikFvIcct4sEx1esHPI35zDY6R0iI/view?usp=share_link)

### Komponentlista

* [RGB-lampa](https://howtomechatronics.com/wp-content/uploads/2015/09/RGB-LED.png?ezimgfmt=ng:webp/ngcb2)
* [3 st Ohm resistorer](https://www.electrokit.com/uploads/productimage/40812/40812016.png) (220Ω)
* [3 st Ohm resistorer](https://www.electrokit.com/uploads/productimage/40810/40810410.png) (10KΩ)
* [3 st fototransistore](https://www.solarbotics.com/wp-content/uploads/17700_-_img_9808-1.jpg)r
* Breadboard
* Arduino Uno
* Plastfilm (Röd, grön, blå)
* Diverse kablar

## Utförandet & byggnationen

Det första steget som togs var att placera alla komponenter på breadboard, anledningen till att detta var att det skulle bli lättare att se exakt vart kablarna skall gå när alla elektroniska komponenter är utplacerade. Även om jag designade ett tydligt kretsschema så är teori & praktik två separata saker när man bygger utifrån ett kretsschema. Innan nästa steg utfördes dubbelkollade jag och såg till att alla komponenter var rätt utplacerade, utifrån hur jag hade tänkt att det skulle kopplas samman sedan med alla kablar. Man bör vara extra noggrann med fototransistorerna och RGB-lampan av följande anledningar:

1. En transistor är en så kallad polariserad komponent, vilket betyder att de enbart tillåter att elektricitet flödar åt ett håll. Det finns en anledning till att dessa fototransistorer har en lång och en kort ände, där den långa änden måste anslutas mot kraftkällan om vi skall få det resultat vi förväntade oss.
2. Eftersom vi vill att RGB-lampan skall skifta mellan röd, grön och blå är det viktigt att se till att vi inte kopplar varje ben på RGB-lampan hursomhelst. Det är viktigt att tänka på då vill att RGB-lampan skall reagera på de omkringliggande ljusförhållanden på ett korrekt sätt.

Nästa steg var att koppla samman alla komponenter och kablar till rätt punkt. Hänvisar till kretsschemat för en djupare insikt i hur det gick till[[1]](#footnote-0).

Det slutgilitga steget var sedan hur allting programmeras:

### Källkoden

Här deklarerar vi variablerna för de stift vi skall använda, på så vis kan vi hålla kolla på vilka stift korresponderar till de tre olika färgerna på RGB-lampan

// Definiera LED-pinnar

const int greenLEDPin = 9;

const int redLEDPin = 10;

const int blueLEDPin = 11;

Principen är densamma som ovan, vi deklarerar följande variabler så att stiften korresponderar till rätt färg

// Definiera sensorns pinnar

const int redSensorPin = A0;

const int greenSensorPin = A1;

const int blueSensorPin = A2;

Deklarerar variabler för de inkommande sensorvärden och för de utgångsvärden vi skall använda för dimmra RGB-lampan, då dessa inte kommer befinna sig ett konstant värde deklarerar vi endast med datatypen och tilldelar ett värde på 0.

// Definiera färgvärden

int redValue = 0;

int greenValue = 0;

int blueValue = 0;

// Definiera sensorvärden

int redSensorValue = 0;

int greenSensorValue = 0;

int blueSensorValue = 0;

I setup() funktionen så påbörjar vi vår seriella kommunikation på 9600 BPS, då vi har så mycket komponenter inkopplade samtidigt, och dessutom 3 känsliga fototransistorer finns det ingen konkret anledning till varför vi skall hålla på att experimentera för mycket med varken BPS eller vår baudrate. Därav anledningen använder vi 9600 som är standard i många fall. Dock om jag valt att titta djupare på den baudrate min nuvarande Arduino har, och konfigurerat detta på ett sådant sätt så att transistorerna reagerar bättre, så hade jag garanterat fått en mer finkänslig färgskiftning på RGB-lampan och bättre sensorvärden.   
Utöver detta så kommer vi kunna se de värden som våra sensorer genererar i vår seriella monitor. Tillagt också att vi kan se de kartlagda värdena som vi använder för att vår RGB-lampa att dimmra.

void setup(){

// Starta seriell kommunikation

Serial.begin(9600);

// Ange LED-pinnar som utgångar

pinMode(greenLEDPin, OUTPUT);

pinMode(redLEDPin, OUTPUT);

pinMode(blueLEDPin, OUTPUT);

}

I loop() funktionen läser analogRead() funktionerna av sensorvärdena från våra A0,A1 och A2 stift. (Här kan vi klart och tydligt se konceptet Pulse Width Modulation i arbete, hade vi inte använt oss av det så hade vi inte kunnat dimmra våra lampor eller för den delen fått ett korrekt värde i vår seriella monitor). Och lagrar dessa i de tidigare deklarerade variabler. Vår delay() funktion har vi också satt på 5 millisekunder (Anledningen till detta är för att om vi satt den högre hade inte vi inte fått korrekta värden då det tar vår ADC 1 millisekund att utföra sitt jobb)

void loop(){

// Läs sensorvärden för varje färg

redSensorValue = analogRead(redSensorPin);

delay(5);

greenSensorValue = analogRead(greenSensorPin);

delay(5);

blueSensorValue = analogRead(blueSensorPin);

Detta är vår rådata som vi hämtat ifrån våra fototransistorer. Vi väljer sedan att skriva ut dessa i vår seriella monitor.

// Skriver ut råa sensorvärden

Serial.print("Raw Sensor Values \t red: ");

Serial.print(redSensorValue);

Serial.print("\t green: ");

Serial.print(greenSensorValue);

Serial.print("\t blue: ");

Serial.print(blueSensorValue);

Funktionen för att förändra RGB-lampans ljusstyrka via PWM heter analogWrite(), den behöver två argument, stiftet som den skall skriva datan till, och ett värde mellan 0,255 (RGB-värde). Det andra värdet är den arbetscykel som vår Arduino skickar som utdata till det specificerade stiftet.

Ett värde på 255 räknas som **HIGH (**Under ett starkt och konsistent ljusförhållande**)** vilket resulterar i att RGB-lampan kommer lysa med högsta intensitet. Ett värde på 127 kommer att konfigurera stiftet **HIGH** hälften av tiden över en viss period, vilket resulterar i att RGB-lampan dimmrar **(**Under skiftande ljusförhållande**).** Om värdet befinner sig på 0 kommer stiftet att befinna sig i ett **LOW** stadie (Avsaknad av ljus)

//Kartlägger sensorvärden till färgvärden

redValue = redSensorValue/4;

greenValue = greenSensorValue/4;

blueValue = blueSensorValue/4;

Rapporterar kartlagda och beräknade värdena ifrån vår RGB-lampa och skriver ut de nya kartlagda värden till vår seriella monitor

//Kartlägger sensorvärden till färgvärden

redValue = redSensorValue/4;

greenValue = greenSensorValue/4;

blueValue = blueSensorValue/4;

Slutligen så skriver vi ut de kartlagda värdena på vardera rad

// Skriver ut kartlagda sensorvärden

Serial.print("Mapped Sensor Values \t red: ");

Serial.print(redValue);

Serial.print("\t green");

Serial.print(greenValue);

Serial.print("\t blue");

Serial.print(blueValue);

Och ställer in färgnivåerna för de stift som kontrollerar det röda, gröna & blåa “benen” på RGB-lampan

// Skriver ut färgvärden till LED-pinnar

analogWrite(redLEDPin,redValue);

analogWrite(greenLEDPin,greenValue);

analogWrite(blueLEDPin,blueValue);

}

## Utmaningar

Den största utmaningen var att koppla alla komponenter på ett bra sätt. Efter att ha designat kretsschemat och påbörjat byggnationen så började jag inse hur trångt det skulle bli när alla komponenter var på ett och samma bräde.   
Eftersom fototransisoterna skulle konstrueras med plastbitar och plastfilmer så tog de   
upp mycket plats på brädet, vilket gjorde att det blev svårare att koppla 10KΩ resistorna till samma terminalremsa. Ett annat problem som uppstod var det faktum att jag tyvärr aldrig fick uppleva några färgskiftningar. Sannolikt var detta för att plastfilmerna satt väldigt löst och att troligtvis kopplingarna sitter dåligt. Ett annat problem jag stötte på i efterhand var att jag tappade bort min gröna plastfilm, jag var tvungen att hämta en grön penna och “rita en ny” film för mitt projekt, inget större problem. Men det kom att påverka sensorvärdena .  
Överlag var utförandet inte så utmanande, att designa ett kretsschema och förstå begreppet Pulse Width Modulation var däremot det mest utmanande med detta projekt.  
För ytterligare dokumentation hänvisas man till denna video:

[Inlämningsuppgift-videodokumentation](https://youtu.be/HDF9UsM8a5w)  


1. [**Kretsschema PDF**](https://drive.google.com/file/d/1wTl8NiaCuCHZH0FQVpVCWgOSsozaAZ4X/view?usp=share_link) [↑](#footnote-ref-0)