



Figure 1: Bok 3

#	Beskriving
9	RGB LED
10	Potmeter RGB
11	FSR RGB
12	LDR RGB

Contents

Förord	1
9: RGB	2
10. Potentiometer	10
12. LDR	28

Förord

Detta är en bok om Arduino för ungdomar. Arduino är ett mikrokontrollerkort du kan programmerar. Denna bok lär dig att göra det.

Om den här boken

Denna bok är licensierad av CC-BY-NC-SA.



Figure 1: Licensen för denna bok

(C) Richèl Bilderbeek och alla lärare och alla elever

Med det här häftet kan du göra vad du vill, så länge du hänvisar till originalversionen på denna webbplats: https://github.com/richelbilderbeek/arduino_foer_ungdomar. Detta häfte kommer alltid att förbli gratis, fritt och öppet.

Det är fortfarande en lite slarvig bok. Det finns stafvel och *layouten är inte alltid vacker*. Eftersom den här boken finns på en webbplats kan alla som tycker att den här boken är för slarvig göra den mindre slarvig.

9: RGB

Ibland tänker du “Kan det här göras smartare?”. En **for**-loop är ett sätt att göra något smartare!

9.1 Regnbåge

I den här lektionen ska vi programmera en regnbåge!

Så här fungerar en regnbåge:

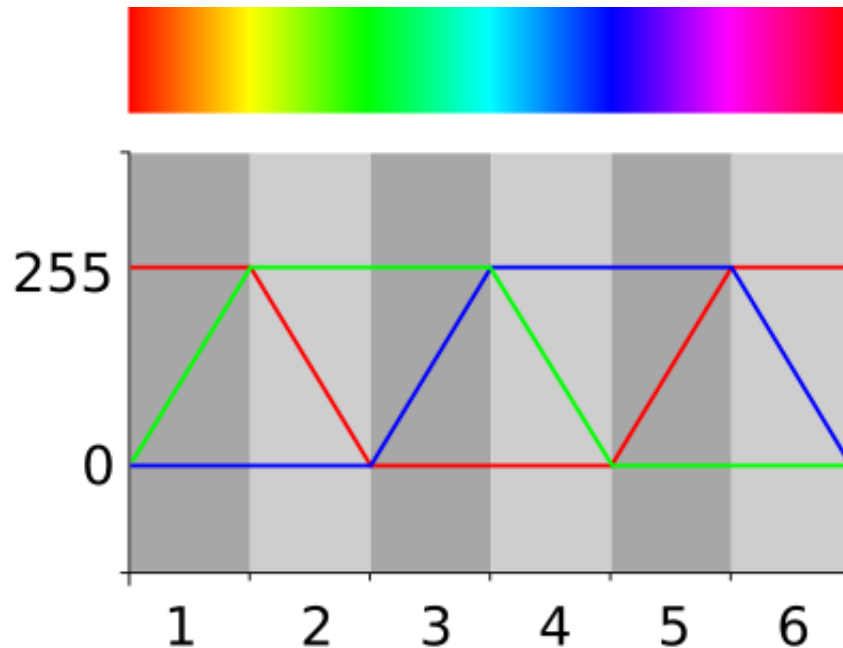


Figure 2: Rainbow



Gult ljus är rött och grönt ljus tillsammans.

9.2 Uppgift 1

Anslut diagrammet nedan:

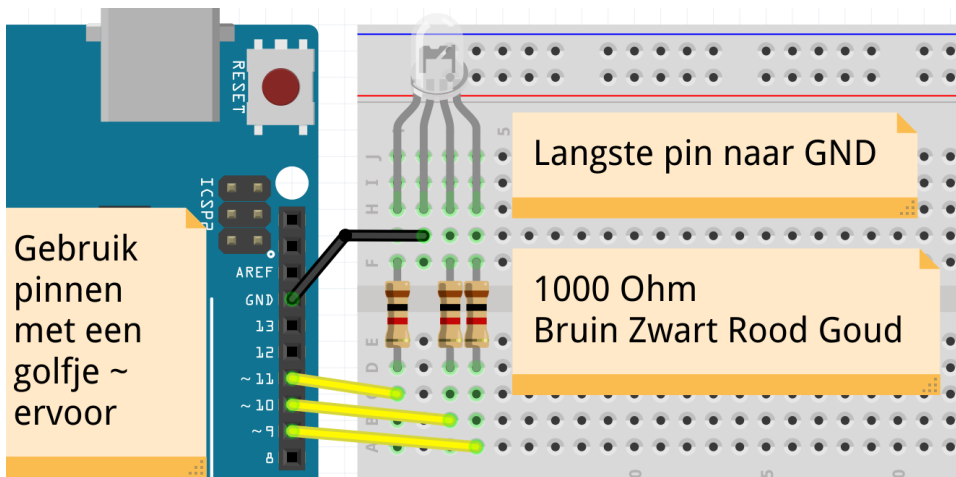


Figure 3: Schema

Använd denna kod:

```
const int stift_rod = 9;
const int stift_gron = 10;
const int stift_bla = 11;

void setup()
{
  pinMode(stift_rod, OUTPUT);
  pinMode(stift_gron, OUTPUT);
  pinMode(stift_bla, OUTPUT);
}

void loop()
{
  digitalWrite(stift_rod, HIGH);
  digitalWrite(stift_gron, HIGH);
  digitalWrite(stift_bla, HIGH);
}
```

Siffrorna bakom `stift_red`, `stift_green` och `stift_blue` är fel. Se till att dessa är rätt siffror. Hur? Koppla loss två ledningar och se vilken färg lysdioden får.

9.3 Lösning 1

Ha, det kan du ta reda på själv :-).

9.4 För loop

Vi börjar med denna kod:

```
// ...

void setup()
{
  // ...
}

void loop()
{
  for (int i=0; i<256; ++i)
  {
    analogWrite(stift_rod, i);
    analogWrite(stift_gron, 0);
    analogWrite(stift_bla, 255);
    delay(30);
  }
}
```



for (int i=0; i<256; ++i) { }	‘Bästa dator, låt ett tal i sträcka sig från 0 till 256 i steg om 1’
analogWrite(9, 255)	‘Bästa dator, slå på stift 9 fullt ut.’
analogWrite(9, 127)	‘Bästa dator, slå på stift 9 till hälften.’
analogWrite(9, 0)	‘Bästa dator, stäng av stift 9.’
analogWrite(9, i)	‘Bästa dator, låt stift 9 gå långsamt (måste ligga inom for-slingan ovan).’



Räknar till tre: ‘1, 2’. Räknar till tre: ‘1, 2, 3’

9.5 Övning 2

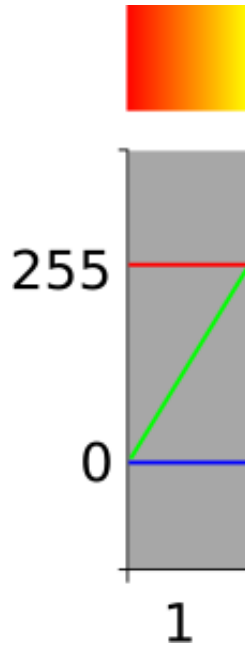


Figure 4: Första delen av regnbågen

Titta på den första delen av regnboken, figur ‘Första delen av regnbågen’. Den röda linjen är där vid 255, den gröna linjen är långsam från 0 till 255 och den blå linjen är noll.

Justera koden så att ljuset går från rött till gult (rött och grönt).



<code>for (int i=0;</code>	‘Bästa dator, låt ett tal i sträcka sig från 0 till 256 i steg om
<code>i<256; ++i) { }</code>	1’
<code> analogWrite(9,</code>	‘Bästa dator, slå på stift 9 fullt ut.’
<code> 255)</code>	
<code>analogWrite(9, 0)</code>	‘Bästa dator, stäng av stift 9.’
<code>analogWrite(9, i)</code>	‘Bästa dator, låt stift 9 gå långsamt (måste ligga inom
	for-slingan ovan).’

9.6 Lösning 2

```
// ...

void setup()
{
  // ...
}

void loop()
{
  for (int i=0; i<256; ++i)
  {
    analogWrite(stift_rod, 255);
    analogWrite(stift_gron, i);
    analogWrite(stift_bla, 0);
    delay(30);
  }
}
```

9.7 Uppgift 3

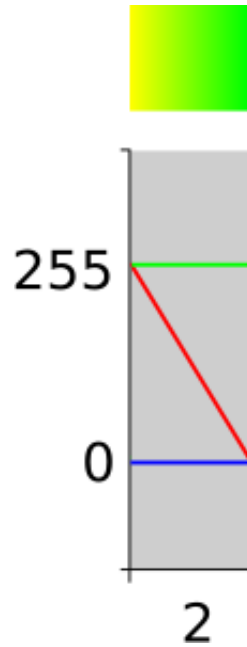


Figure 5: Andra delen av regnbågen

Titta på den andra delen av regnboken, figur “Andra delen av regnbågen”. Den röda linjen går från 255 till 0, den gröna linjen är 255 och den blå linjen är noll.

Justera koden så att ljuset går från rött till gult (rött och grönt).

Tips: lägg till denna kod:

```
for (int i=0; i<256; ++i)
{
  analogWrite(stift_rod, 255 - i);
  analogWrite(stift_gron, 255);
  analogWrite(stift_bla, 0);
  delay(30);
}
```



```
for (int i=0;
i<256; ++i) { }
  analogWrite(9,
    255)
analogWrite(9, 0)
```

‘Bästa dator, låt ett tal i sträcka sig från 0 till 256 i steg om 1’

‘Bästa dator, slå på stift 9 fullt ut.’

‘Bästa dator, stäng av stift 9.’



<code>analogWrite(9, 255</code>	‘Bästa dator, låt stift 9 gå ut långsamt (måste vara inom
<code>- i)</code>	for-slingan ovan).’

9.8 Lösning 3

```
// ...

void setup()
{
  // ...
}

void loop()
{
  // ...
  for (int i=0; i<256; ++i)
  {
    analogWrite(stift_rod, 255 - i);
    analogWrite(stift_gron, 255);
    analogWrite(stift_bla, 0);
    delay(30);
  }
}
```

9.9 Slutuppgift

Avsluta nu regnbågen.

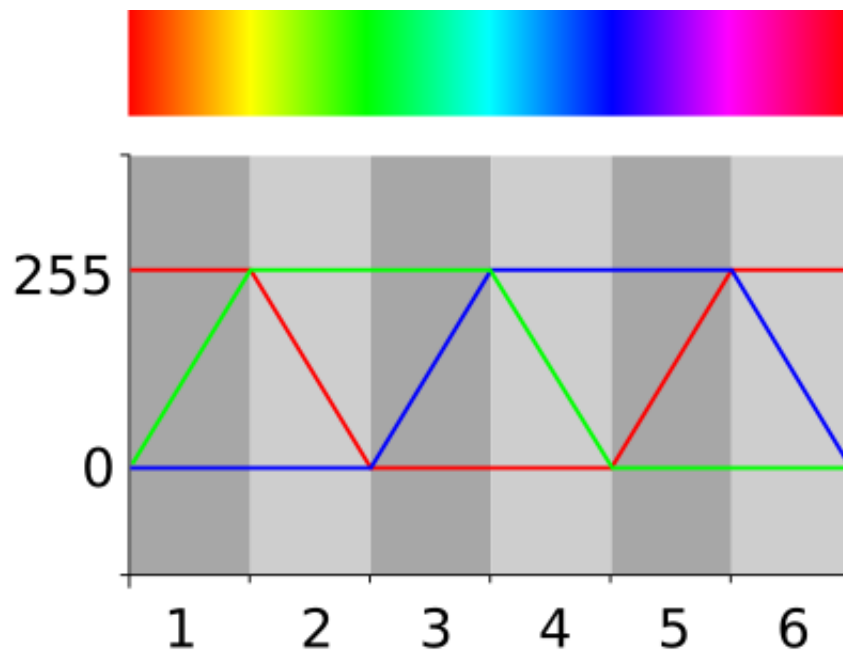


Figure 6: Rainbow

10. Potentiometer

Med en potentiometer kan du skicka en spänning till Arduino. Du kan använda denna som till exempel en volymratt.



Figure 7: En potentiometer

I den här lektionen kommer du att lära dig:

- Vad en potentiometer är
- Hur man använder en potentiometer

10.1 Anslut

Först ansluter vi bara en Arduino:

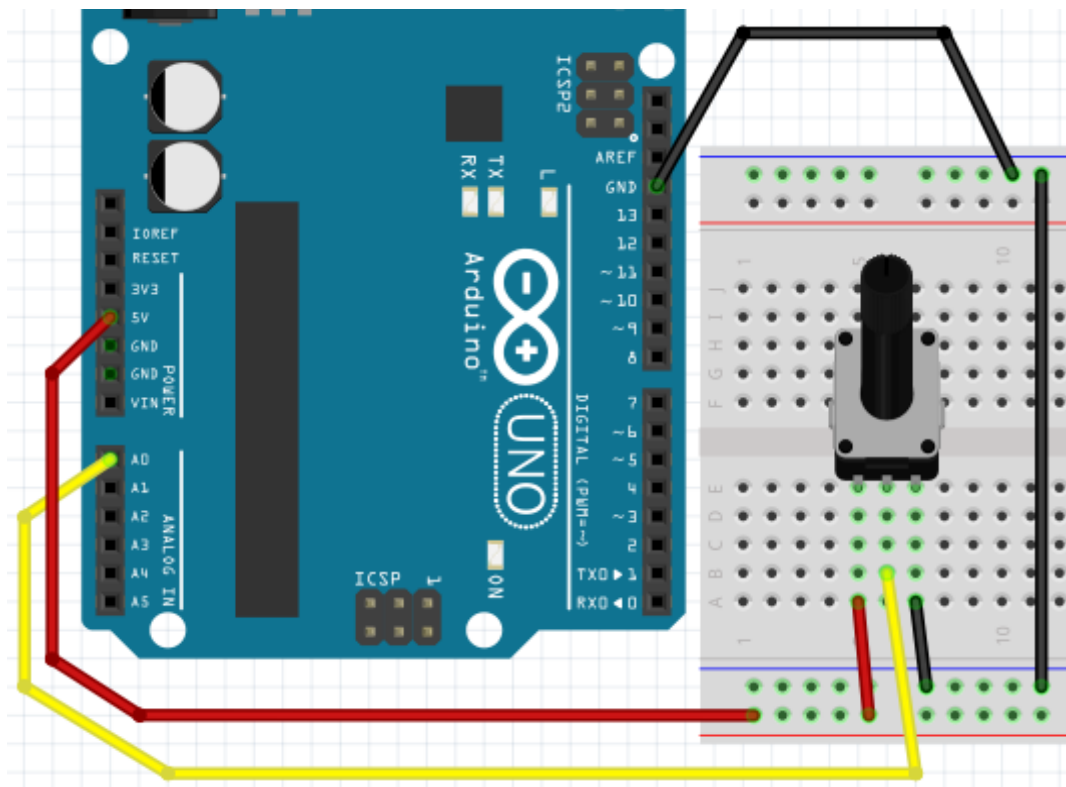


Figure 8: Anslut potentiometer

Jag tycker att det här borde fungera :-)

10.2 Kod: avläs potentiometer med seriell monitor

Vi mäter potentiometerns position med denna kod:

```
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  delay(100);
}
```

Detta är vad koden gör:

- Två saker händer i “setup”-funktionen
 - `pinMode(A0, INPUT)`: stiftet A0 är ett stift som läser, en ingång
 - `Serial.begin(9600)`: den seriella monitorn skickar 9600 bitar (‘nollor och ettor’) per sekund
- Två saker händer i ‘loop’-funktionen
 - `Serial.println(analogRead(A0))`: läs stift A0 och skriv det till seriell monitor
 - `delay(100)`: vänta hundra millisekunder

10.3 Uppgifter

1. Ladda upp programmet. I Arduino IDE, klicka på “Serial Monitor” uppe till höger. Vad ser du?
2. Vrid potentiometern hela vägen åt vänster (moturs). Vilket värde ser du på den seriella monitorn?
3. Vrid potentiometern hela vägen åt höger (medurs). Vilket värde ser du på den seriella monitorn?

10.4 Lösningar

1. Du ser ett tal från noll till 1024, beroende på potentiometerns position
2. Du ser siffran noll
3. Du ser numret 1023

10.5 Anslutningspotentiometer med LED

Nu ansluter vi även en LED:

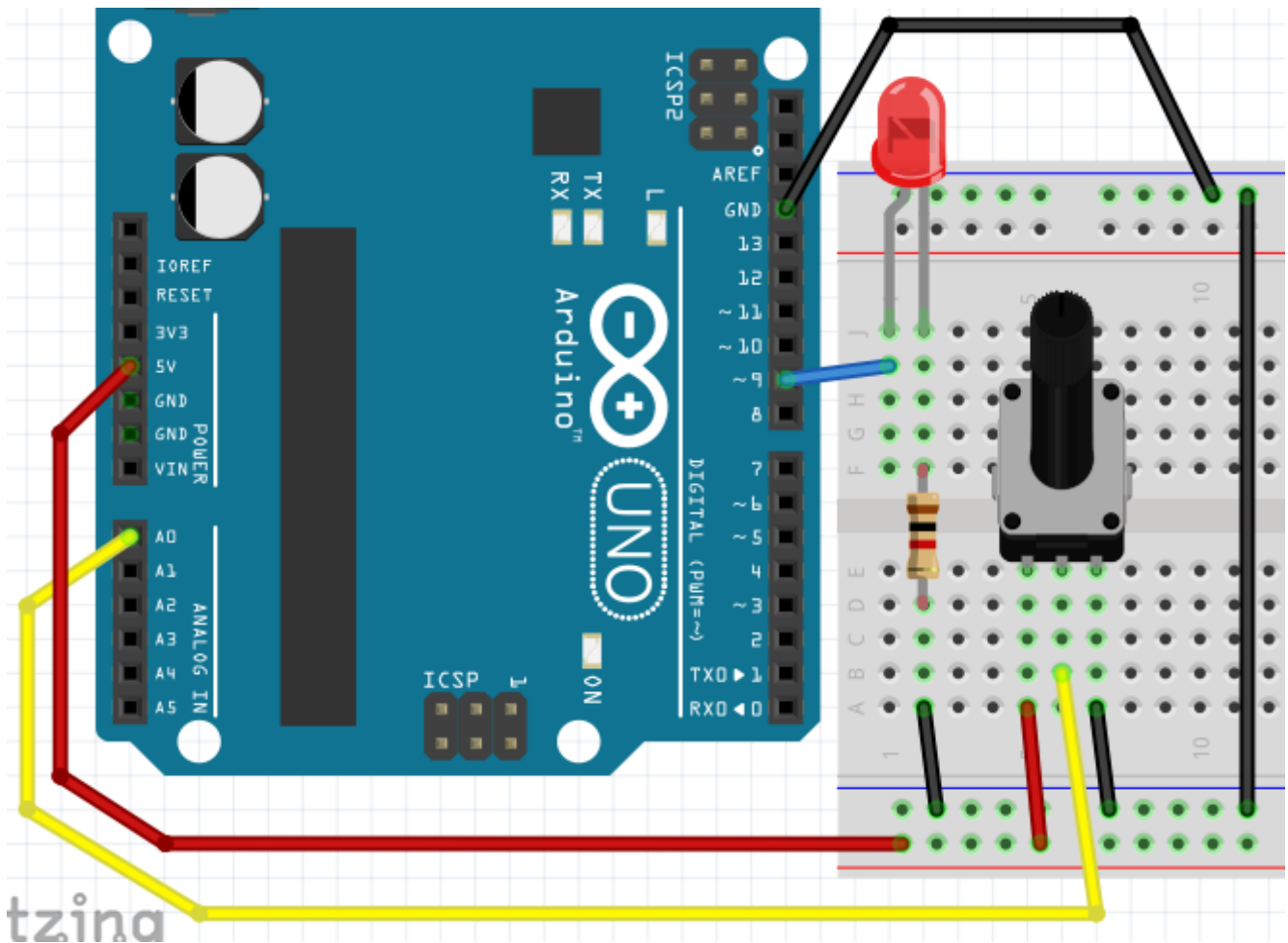


Figure 9: Potmeter med LED

OBS:

- resistorn på lysdioden är tusen ohm (brun-svart-röd-guld)
- LED-lampan måste vara på ett stift med PWM ('a wave')

Svara på potentiometern

Nu ska vi få lysdioden att reagera på potentiometern:

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(9, OUTPUT);
}
```

```
void loop()
```

```
{  
  analogWrite(9, analogRead(A0) / 8);  
  delay(100);  
}
```

Detta är vad koden gör

- Tre saker händer i “setup”-funktionen:
 - `pinMode(A0, INPUT)`: stiftet A0 är ett stift som läser, en ingång
 - `pinMode(9, OUTPUT)`: stift 9 är ett stift där ström kommer ut, en utgång
- Två saker händer i “loop”-funktionen:
 - `analogWrite(9, analogRead(A0) / 8)`: sätt en spänning på stift 9 `analogRead(A0) / 8`. `analogRead(A0) / 8` betyder: läs värdet från A0 dividerat med åtta (‘/’ är en divisionsstapel).
 - `delay(100)`: vänta hundra millisekunder

10.6 Uppdrag

- Vad händer om du ändrar divisionsnumret?
- Vilket är det bästa värdet av talet för division/

10.7 Lösningar

1. Med en högre siffra går ljuset mindre och mindre snabbt i fullt läge. Om siffran är lägre än fyra kommer ljuset att reagera konstigt: ljuset kommer först att gå snabbare, sedan av och högre igen osv
2. Bästa siffran är fyra. Eftersom från `analogRead` kommer som mest 1023, medan med `analogWrite` kan skriva högst 255. 1023 dividerat med 4 är 255 resterande 3. Ardiono gör 255 av det

10.8 Slutuppgift

- Anslut fyra lysdioder: en röd, gul, grön och blå
- Om potentiometern är ända till vänster ska ingen lysdiod lysa.
- När potentiometern vrids mer åt höger tänds den gröna lysdioden
- Om potentiometern vrids ännu mer åt höger, tänds den gula lysdioden
- Om potentiometern vrids ännu mer åt höger, tänds den röda lysdioden
- Den blå lysdioden lyser långsamt när du svänger mer åt höger

#4. FSR

Seriell monitor

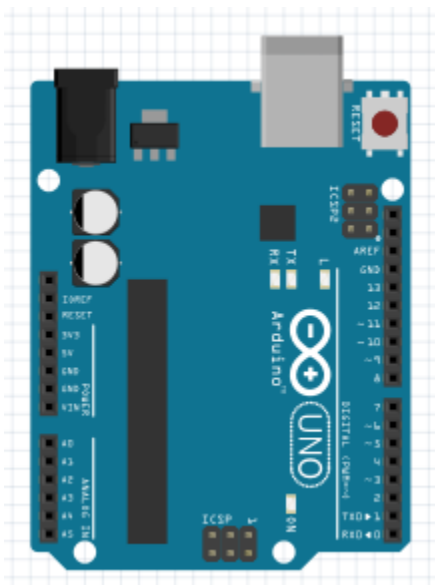


Seriell monitor: platsen där du kan få Arduino att prata genom en seriell port

Den seriella monitorn låter oss få Arduino att prata. Eller mer exakt: att denna text skickas till den seriella monitorn. Den seriella monitorn visar denna text på din dator.

Anslut endast Arduino

Först ansluter vi bara en Arduino:



Jag tycker att det här borde fungera :-)



Den seriella bildskärmen går via USB-kabeln mellan Arduino till dator

Kod: Serial Monitor

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Hallo");
  delay(1000);
}
```



`Serial.begin(9600);` ‘Bästa dator, låt Arduino prata med 9600 bitar per sekund’
`Serial.print("Hej");` ‘Bästa dator, låt Arduino säga ordet ’Hej’
`Serial.println("Hej");` ‘Bästa dator, låt Arduino säga ordet ’Hej’ och starta en ny rad’

Kommandon 1

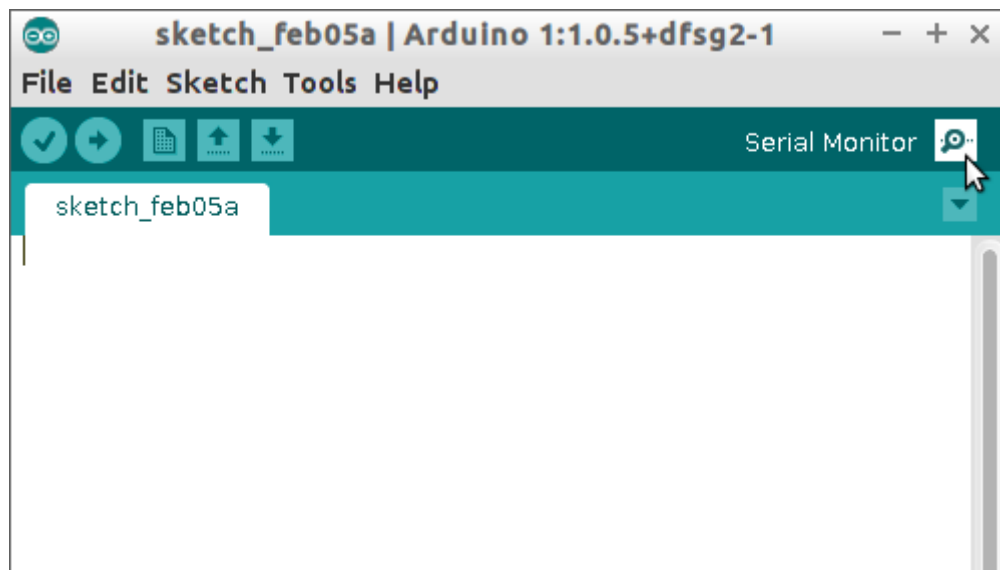


Figure 10: Den seriella monitorn är här

1. Ladda upp programmet. I Arduino IDE, klicka på “Serial Monitor” uppe till höger. Vad ser du?
2. Kan du ändra texten till “Hej Richel” (eller ditt eget namn?)

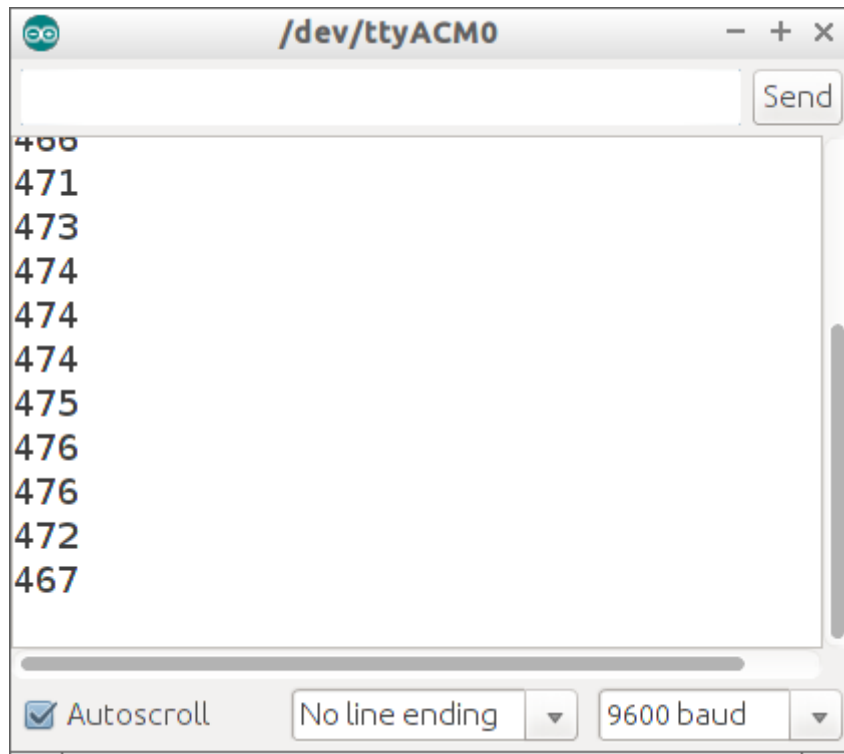


Figure 11: Den seriella monitorn med siffror

3. Ändra `Serial.println` till `Serial.print`. Vad ser du?
4. Ändra texten `Serial.begin(9600)` till `Serial.begin(4800)`. Vad ser du? Varför?

Svar 1

1. Den seriella monitorn visar en extra rad varje sekund, med texten "Hej"
2. Ändra raden `Serial.println("Hello");` till `Serial.println("Hello Ledge");`
3. Orden kommer en efter en, istället för en efter en
4. Nu visar den seriella monitorn oläsbar text. Detta beror på att Arduino är långsammare att texta skicka till din dator (4800), sedan läser din dator texten (9600)

Ansluter FSR utan LED



FSR betyder “Force Sensitive Resistance”

Först ansluter vi bara en FSR:

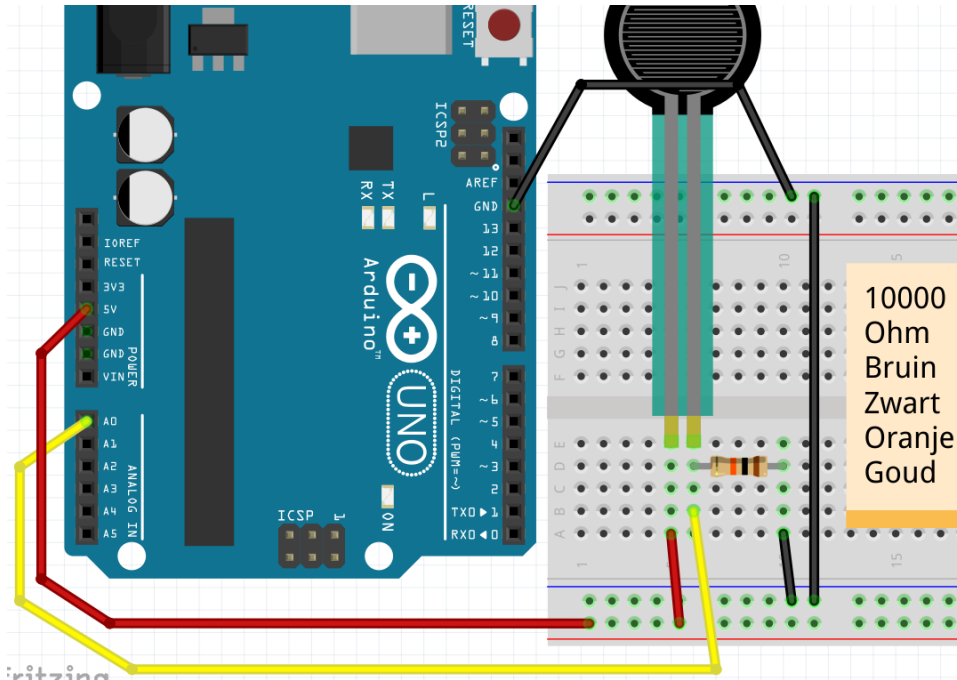


Figure 12: Flödesschema



Om det inte finns någon FSR, använd en LDR

Kod: läs FSR med seriell monitor

Med denna kod mäter vi värdet på FSR:

```
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  delay(100);
}
```



`Serial.println(analogRead(A0))` 'Bästa dator, visa värdet på stiftet A0 på den seriella monitorn'

Kommandon 2

1. Ladda upp programmet. I Arduino IDE, klicka på "Serial Monitor" uppe till höger. Vad ser du?
2. Tryck på FSR med fingrarna (eller, med en LDR: håll fingret över LDR) medan du tittar på den seriella monitorn. Vad ser du?
3. Ändra `Serial.println` till `Serial.print`. Vad ser du?
4. Ändra texten `Serial.begin(9600)` till `Serial.begin(4800)`. Vad ser du? Varför?
5. Ta bort kabeln till A0. Ja, ta bort kabeln mellan A0 och LDR. Titta på den seriella monitorn. Vad ser du?



Motståndet mellan A0 och LDR kallas ett 'Pull Down'-motstånd

Lösningar 2

1. Du kommer att se ett tal från noll till 1024, beroende på värdet på FSR
2. Du ändrar siffrorna
3. Alla nummer kommer efter varandra
4. Nu visar den seriella monitorn oläsbar text. Detta beror på att Arduino är långsammare att texta skicka till din dator (4800), sedan läser din dator texten (9600)
5. Nu kommer du att se siffran ändras slumpmässigt. Detta kallas en flytande ingång



Ett 'Pull Down'-motstånd förhindrar en flytande ingång

Ansluter FSR med LED, på/av



“Kraftkänsligt motstånd” betyder “Kraftberoende motstånd”

Nu ansluter vi även en LED:

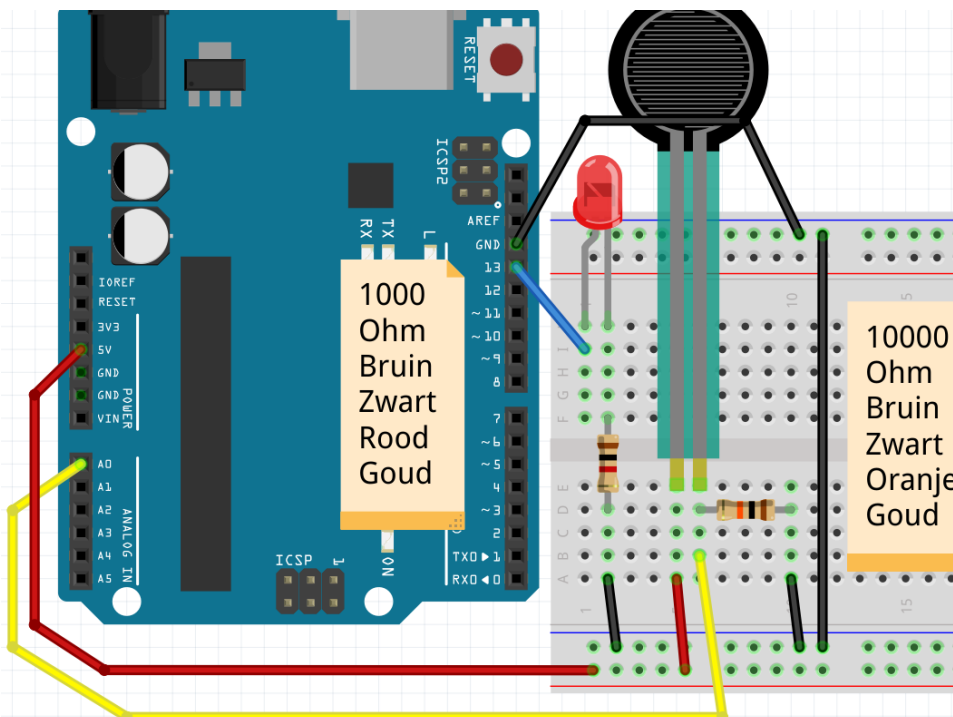


Figure 13: Flödesschema

Svara på FSR, på/av

Nu ska vi få lysdioden att reagera på lysdioden:

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(13, OUTPUT);
}

void loop()
{
  if (analogRead(A0) < 512)
  {
    digitalWrite(13, HIGH);
  }
  else
  {
    digitalWrite(13, LOW);
  }
  delay(100);
}
```



<code>if (analogRead(A0)</code>	'Bästa dator, om A0 läser mindre än 2,5 volt, sätt det inom
<code>< 512) {}</code>	parentes.

Uppdrag 3

1. Vad händer om du ökar 512? Vad händer om du sänker 512?
2. Se till att den seriella monitorn också mäter och visar "A0". Vilket nummer mäter FSR? i fred?
3. Se till att den seriella monitorn visar ordet "ON" när lysdioden tänds, och att ordet "OFF" när lysdioden är avstängd

Lösningar 3

1. Om 512 ändras till en siffra som är för hög, kommer lampan alltid att lysa oavsett hur hårt/mjukt du trycker. Om 512 ändras till ett för högt nummer kommer lampan alltid att vara släckt, oavsett hur hårt/mjukt du trycker
2. För detta använd koden från föregående kommando: lägg till i `setup`-funktionen `Serial.begin(9600);`, i `loop`-funktionen lägg till `Serial.println(analogRead(A0));`. Värdet du kommer att se är beroende på motstånd, FSR och situation
3. Detta kan göras genom att sätta `Serial.println("ON");` i den första delen av `if`-satsen. Sätt `Serial.println("OFF");` i den andra delen av `if`-satsen.

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  if (analogRead(A0) < 512)
  {
    digitalWrite(13, HIGH);
    Serial.println("AAN");
  }
  else
  {
    digitalWrite(13, LOW);
    Serial.println("UIT");
  }
  delay(100);
}
```

Svara på FSR, Dim

Nu ska vi få lysdioden att reagera på lysdioden. Denna gång *dimper* lysdioden.

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode( 9, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  const int fsr_waarde = analogRead(A0);
  Serial.print("FSR: ");
  Serial.println(fsr_waarde);
  const int led_waarde = map(fsr_waarde, 0, 1023, 0, 255);
  Serial.print("LED: ");
  Serial.println(led_waarde);
  analogWrite(led_waarde, 9);
  delay(100);
}
```



```
analogWrite( 0, 9)
analogWrite(128,
  9)
```

‘Bästa dator, stäng av stift 9’

‘Bästa dator, vrid stift 9 till halvfullt’

```
analogWrite(255,
  9)
```

‘Bästa dator, slå på stift 9 fullt ut’

`map(analogRead(A0), 0, 1023, 0, 255)` läser spänningen för A0. Detta är ett värde från 0 till 1023. Konvertera det avlästa värdet mellan 0 och 255.!

Uppdrag 4

1. Lysdioden är på ett annat stift. Titta i koden och anslut lysdioden till rätt stift
2. Vilka stift kan vi använda för att dämpa en lysdiod?

Lösningar 4

1. Lysdioden måste nu kopplas till stift 9
2. Alla stift med en våg (~) före numret. Dessa är 3, 5, 6, 9, 10, 11.

Uppgift 5

Anslut två lysdioder till stift 12 och 13. När FSR är i vila ska ingen lysdiod lysa. Om du trycker försiktigt på FSR, en LED tänds. Om du trycker hårt på FSR, två.



Tips: använd två `if`-satser

Lösning 5

Siffrorna i “if”-satsen måste vara korrekt inställda.

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  if (analogRead(A0) < 256)
  {
    digitalWrite(13, HIGH);
  }
  if (analogRead(A0) < 512)
  {
    digitalWrite(12, HIGH);
  }
  delay(100);
}
```

Uppgift 6

Du kan också få en LED att reagera på en FSR genom att bleka/dimma den

1. Vilket kommando gjorde du det med igen?
2. Är det möjligt med vilken stift som helst? Om inte, med vilket ja/nej?
3. Vilket är det högsta värdet som du kan tända en LED med?
4. Vilket är det högsta värdet FSR kan mäta?
5. Antag att du vill tända en lysdiod beroende på ett FSR-värde. Hur kunde du göra detta?
6. Hur får man kod att göra en division?
7. Slå på lysdioden beroende på FSR-värdet

Lösningar 6

1. Du kan tona en lysdiod med `analogWrite`, till exempel `analogWrite(11, 255);`
2. Du kan bara dimma en lysdiod med PWM-stift. Det här är stiften med en våg (~) bredvid deras nummer. På Arduino Uno är dessa stift 3, 5, 6, 9, 10 och 11
3. Med `analogWrite` kan du ge upp till 255, till exempel `analogWrite(11, 255);`
4. Med `analogRead` kan du mäta upp till 1023
5. Du läser ett värde, dividerar det med fyra (1024 dividerat med 256 är fyra) och låter lysdioden lysa så
6. Med divisionsfältet, `/`.
7. Se nedan. Glöm inte att sätta en lysdiod på stift 11

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(11, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  analogWrite(11, analogRead(A0) / 4);
  delay(100);
}
```



<code>analogWrite(11,</code>	‘Bästa dator, dämpa stift 11 till värdet av stift A0 (diva detta med fyra)’
<code>analogRead(A0) /</code>	
<code>4)</code>	

Einorder

- Anslut fyra lysdioder: en vit, en röd, en gul och en grön
- När FSR är i vila ska ingen lysdiod lysa.
- När du trycker lätt på FSR tänds den gröna lysdioden
- Om du trycker hårdare på FSR kommer de gröna och gula lysdioderna att tändas
- När du trycker hårt på FSR tänds de gröna, gula och röda lysdioderna
- Den vita lysdioden lyser starkare och mjukare beroende på FSR

Om du inte har en vit LED, använd en annan färg.

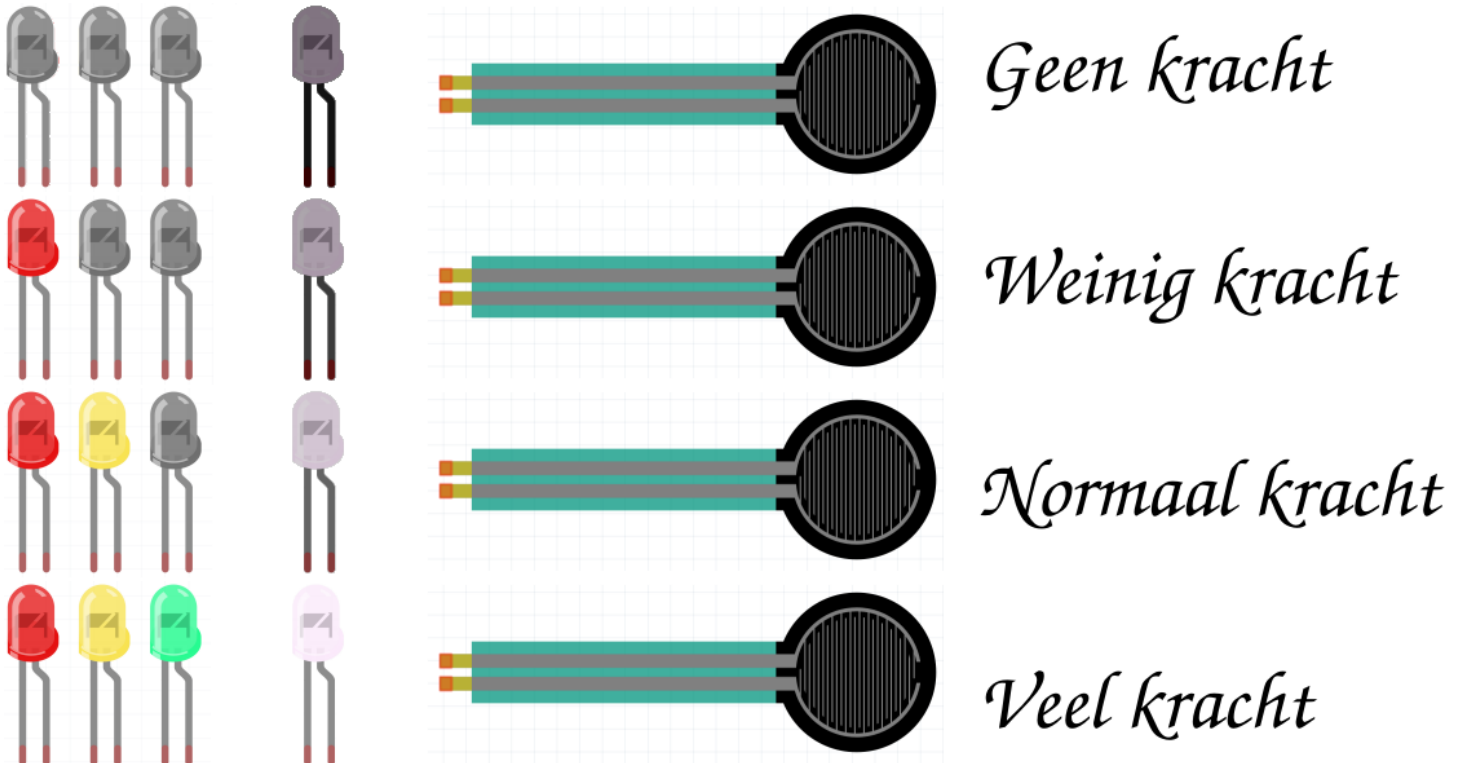
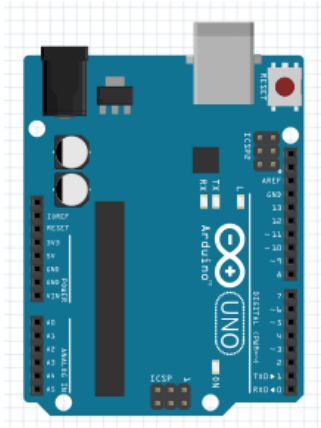


Figure 14: FSR final assignment

12. LDR

12.1 Anslut endast Arduino

Först ansluter vi bara en Arduino:



Jag tycker att det här borde fungera :-)

12.2 Kod: Serial Monitor

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Hallo");
  delay(1000);
}
```



Har du glömt vad det här är? Gå till lektion 4 ‘FSR’

12.3 Kommandon 1

1. Ladda upp programmet. I Arduino IDE, klicka på “Serial Monitor” uppe till höger. Vad ser du?
2. Kan du ändra texten till “Hej Richel” (eller ditt eget namn?)
3. Ändra `Serial.println` till `Serial.print`. Vad ser du?

4. Ändra texten `Serial.begin(9600)` till `Serial.begin(4800)`. Vad ser du? Varför?

12.4 Lösningar 1

1. Den seriella monitorn visar en extra rad varje sekund, med texten "Hej"
2. Ändra raden `Serial.println("Hello");` till `Serial.println("Hello Ledge");`
3. Orden kommer en efter en, istället för en efter en
4. Nu visar den seriella monitorn oläsbar text. Detta beror på att Arduino är långsammare att texta skicka till din dator (4800), sedan läser din dator texten (9600)

12.5 Ansluter LDR utan LED



LDR betyder "Ljusberoende motstånd"

Först ansluter vi bara en LDR:

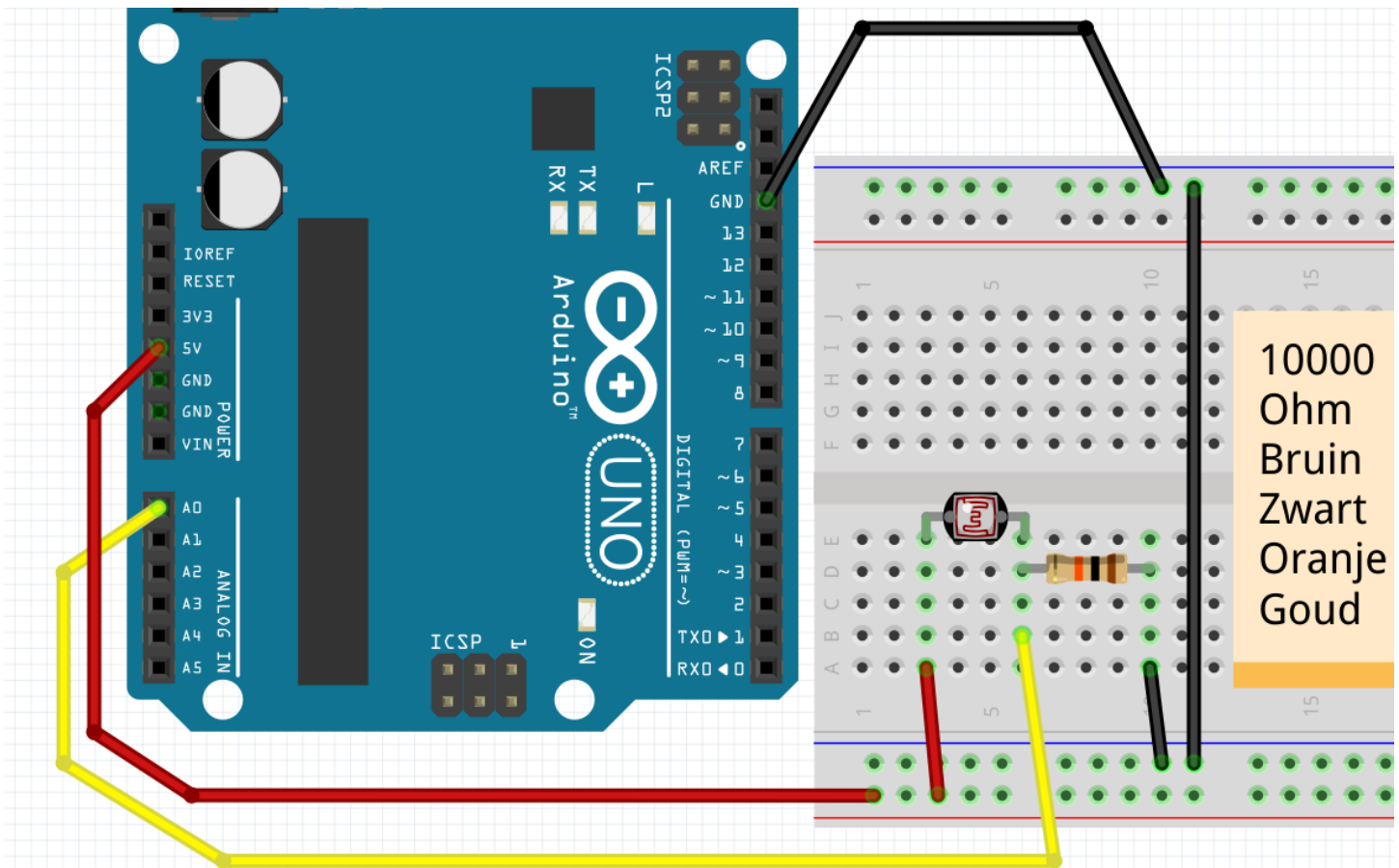


Figure 15: Flödesschema

12.6 Kod: Läs FSR med seriell monitor

Med den här koden mäter vi värdet på LDR:

```
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  delay(100);
}
```



Har du glömt vad det här är? Gå till lektion 4 'FSR'

12.7 Uppdrag 2

1. Ladda upp programmet. I Arduino IDE, klicka på "Serial Monitor" uppe till höger. Vad ser du?
2. Håll fingret över LDR medan du tittar på den seriella monitorn. Vad ser du?
3. Ändra `Serial.println` till `Serial.print`. Vad ser du?
4. Ändra texten `Serial.begin(9600)` till `Serial.begin(4800)`. Vad ser du? Varför?
5. Ta bort kabeln till A0. Ja, ta bort kabeln mellan A0 och LDR. Titta på den seriella monitorn. Vad ser du?

12.8 Lösningar 2

1. Du kommer att se ett tal från noll till 1024, beroende på värdet på LDR
2. Du ändrar siffrorna
3. Alla nummer kommer efter varandra
4. Nu visar den seriella monitorn oläsbar text. Detta beror på att Arduino är långsammare att texta skicka till din dator (4800), sedan läser din dator texten (9600)
5. Nu kommer du att se siffran ändras slumpmässigt. Detta kallas en flytande ingång



Har du en flytande ingång? Då har man ofta kopplat något fel

12.9 Ansluter LDR med LED



“Ljusberoende motstånd” betyder “ljusberoende motstånd”

Nu ansluter vi även en LED:

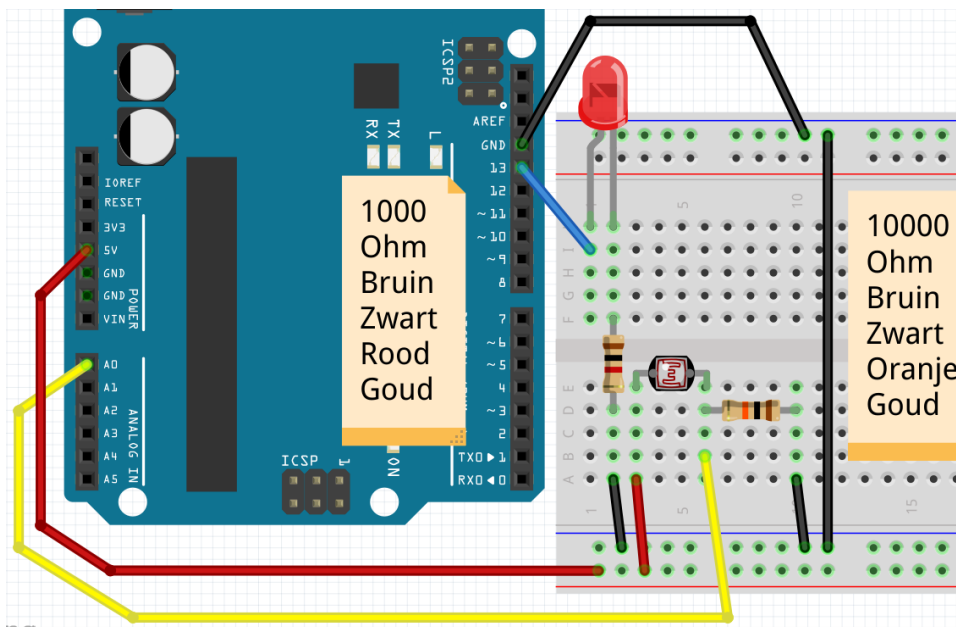


Figure 16: Flödesschema

12.10 Svar till LDR

Nu ska vi få lysdioden att reagera på LDR:

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(13, OUTPUT);
}

void loop()
{
  if (analogRead(A0) < 512)
  {
    digitalWrite(13, HIGH);
  }
  else
  {
    digitalWrite(13, LOW);
  }
  delay(100);
}
```



Har du glömt vad det här är? Gå till lektion 4 'FSR'

12.11 Uppgift 3

1. Vad händer om du ökar 512? Vad händer om du sänker 512?
2. Se till att den seriella monitorn också mäter och visar "A0". Vilket nummer mäter FSR? i fred?
3. Se till att den seriella monitorn visar ordet "ON" när lysdioden tänds, och att ordet "OFF" när lysdioden är avstängd

12.12 Lösningar 3

1. Om 512 ändras till en siffra som är för hög, kommer lampan alltid att lysa oavsett hur hårt/mjukt du trycker. Om 512 ändras till ett för högt nummer kommer lampan alltid att vara släckt, oavsett hur hårt/mjukt du trycker
2. För detta använd koden från föregående kommando: lägg till i `setup`-funktionen `Serial.begin(9600);`, i `loop`-funktionen lägg till `Serial.println(analogRead(A0));`. Värdet du kommer att se är beroende på resistans, LDR och ljusmängd
3. Detta kan göras genom att sätta `Serial.println("ON");` i den första delen av `if`-satsen. Sätt `Serial.println("OFF");` i den andra delen av `if`-satsen.

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  if (analogRead(A0) < 512)
  {
    digitalWrite(13, HIGH);
    Serial.println("AAN");
  }
  else
  {
    digitalWrite(13, LOW);
    Serial.println("UIT");
  }
  delay(100);
}
```

12.13 Uppgift 4

Anslut en extra LED. Om LDR är i normalt ljus bör ingen lysdiod lysa. Om du gör LDR lite mörkare med handen kommer en lysdiod att tändas. Om du gör LDR helt mörk två.



Tips: använd två “om”-satser

12.14 Lösning 4

Siffrorna i if-satsen måste vara korrekt inställda.

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  if (analogRead(A0) < 256)
  {
    digitalWrite(13, HIGH);
  }
  if (analogRead(A0) < 512)
  {
    digitalWrite(12, HIGH);
  }
  delay(100);
}
```

12.15 Slutuppgift

- Anslut tre lysdioder: en röd, gul och grön
- När LDR är i ljuset ska ingen lysdiod lysa.
- Om du mörkar LDR en aning kommer den gröna lysdioden att lysa
- Om du gör LDR mörkare mer kommer de gröna och gula lysdioderna att lysa
- Om du gör LDR helt mörkare kommer alla lysdioder att lysa



Figure 17: LDR final assignment



Tips: använd if tre eller fyra gånger
