



Figure 1: Bok 1

#	Beskriving
1	Blink
2	Blink Blink Blink
3	Knapp, <code>if</code> , <code>else</code>
4	Knapp flip

Contents

Förord	1
Lektion 1: Blink	2
Lektion 2: Blink Blink Blink	16
Lektion 3: Knapp if ... else	29
Lektion 4: Knapp flip	39

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.

Lektion 1: Blink

Under den här lektionen ska vi göra den enklaste kretsen som finns: Blink!



Vi börjar lätt!

1.1 Blink: Ansluta en Arduino

Anslut en Arduino så här:

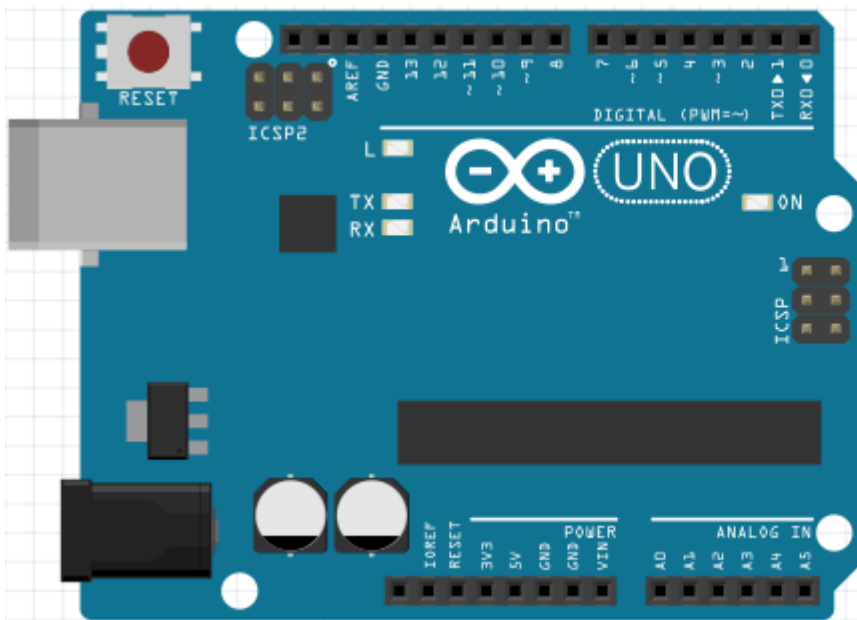


Figure 2: Blink utan lysdiod

Anslut en USB-kabel till din Arduino och till din bärbara dator.

1.2 Blink: Uppgift 1

Koppla in Arduino.



Arduino har redan ett ljus som du kan programmera.

1.3 Blink: Startar Arduino IDE



Vi programmerar Arduino med Arduino IDE



‘IDE’ uttalas som ‘ie-dee-ee’

Starta Arduino IDE genom att:

- Klicka på genvägen på skrivbordet
- Tryck på Windows-tangenten (nedre till vänster, mellan **Ctrl** och **Alt**). Typ sedan **arduino** (små bokstäver) och sedan Enter

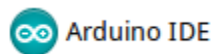


Figure 3: Logotyp för Arduino IDE



Figure 4: Win tangent

Du ser nu Arduino IDE:

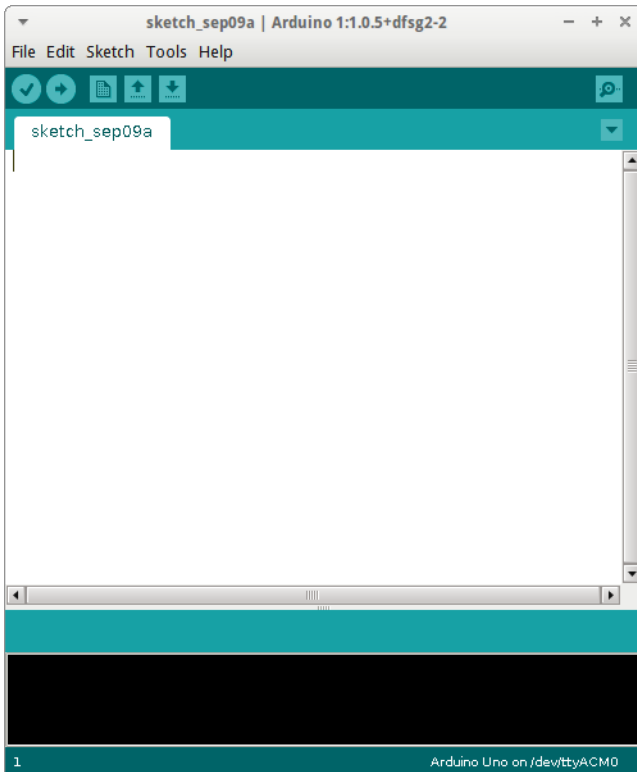


Figure 5: Arduino IDE



IDE betyder 'Integrated Development Environment',
programmet du använder för att programmera.

1.4 Blink: Vårt första program

Detta är (en version av) koden för Blink:

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

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```



```
void setup() { }  'Kära dator, gör vad som än står inom måsvingarna'
```



```
void loop() { }    'Kära dator, gör vad som än står inom måsvingarna hela  
                  tiden.'
```

Skriv in koden i Arduino IDE och klicka på 'Upload' ('Ladda up').



Figure 6: Här kan du klicka på 'Upload'

1.5 Blink: Uppgift 2

Få lysdioden att blinka 10 gånger så snabbt.

Tips: LED är nu 1000 millisekunder på och 1000 millisekunder av.

1.6 Blink: Lösning 2

```
void setup()
{
  // ...
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(100);
  digitalWrite(13, LOW);
  delay(100);
}
```



// ... betyder 'koden du redan har där'



pinMode(13, OUTPUT); 'Se till att spänning kan komma ut ur stift 13'



digitalWrite(13, HIGH); 'Sätt spänning på stift 13'



delay(1000); 'Vänta 1000 millisekunder'

1.7 Blink: Uppgift 3

Låt lysdioden blinka så snabbt du kan. Vad ser du?



Det blir antagligen nåt du inte förväntade!

1.8 Blink: Lösning 3

Det finns fler lösningar, det här är en:

```
void setup()
{
  // ...
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1);
  digitalWrite(13, LOW);
  delay(1);
}
```

Du kommer inte längre att se lampan blinka.



Du kommer inte längre att se lampan blinka!



Ljuset blinkar, men våra ögon kan inte titta tillräckligt snabbt för att se det.



```
digitalWrite(13, LOW); 'Ingen spänning på stift 13'
```



Också bra: `delay(0);` eller ta bort raderna med `delay` helt.

1.9 Blink: Anslut LED

Nu är det dags att ansluta Blink:

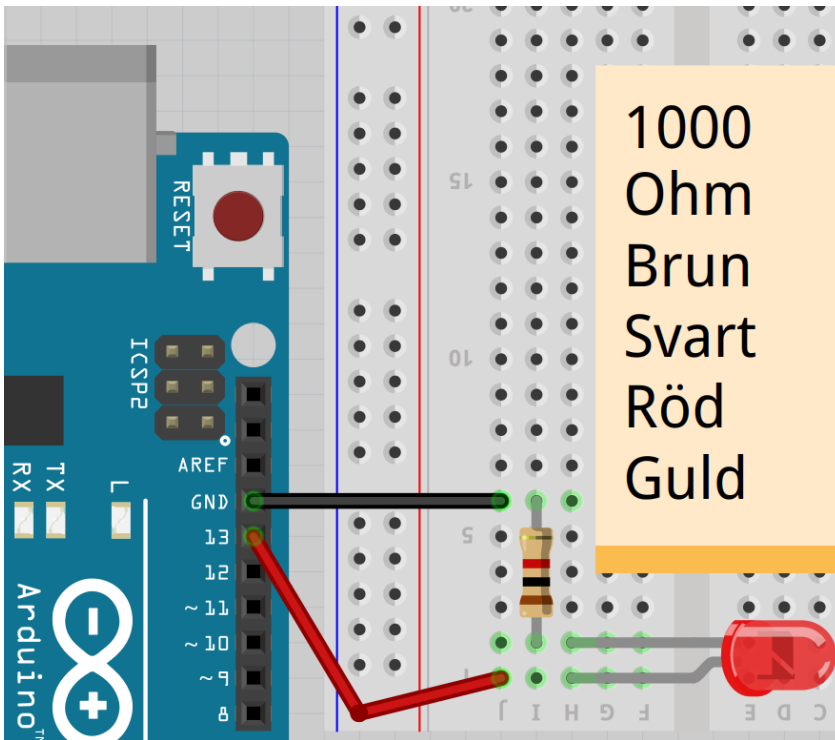


Figure 7: Blink

- Koppla ur USB-kabeln från datorn, så att Arduino inte längre har ström
- Anslut Blink enligt bilden
- Sätt tillbaka strömmen till Arduino
- Om allt är bra kommer nu den röda lysdioden att blinka tillsammans med lysdioden på Arduino.



Om lysdioden inte lyser, vänd på den

1.10 Blink: Uppgift 4

Anslut nu din LED till stift 12.



Det är coolt att ta tag i stift 12 efter stift 13



Detta beror också på att stift 0 och 1 är speciella

1.11 Blink: Lösning 4

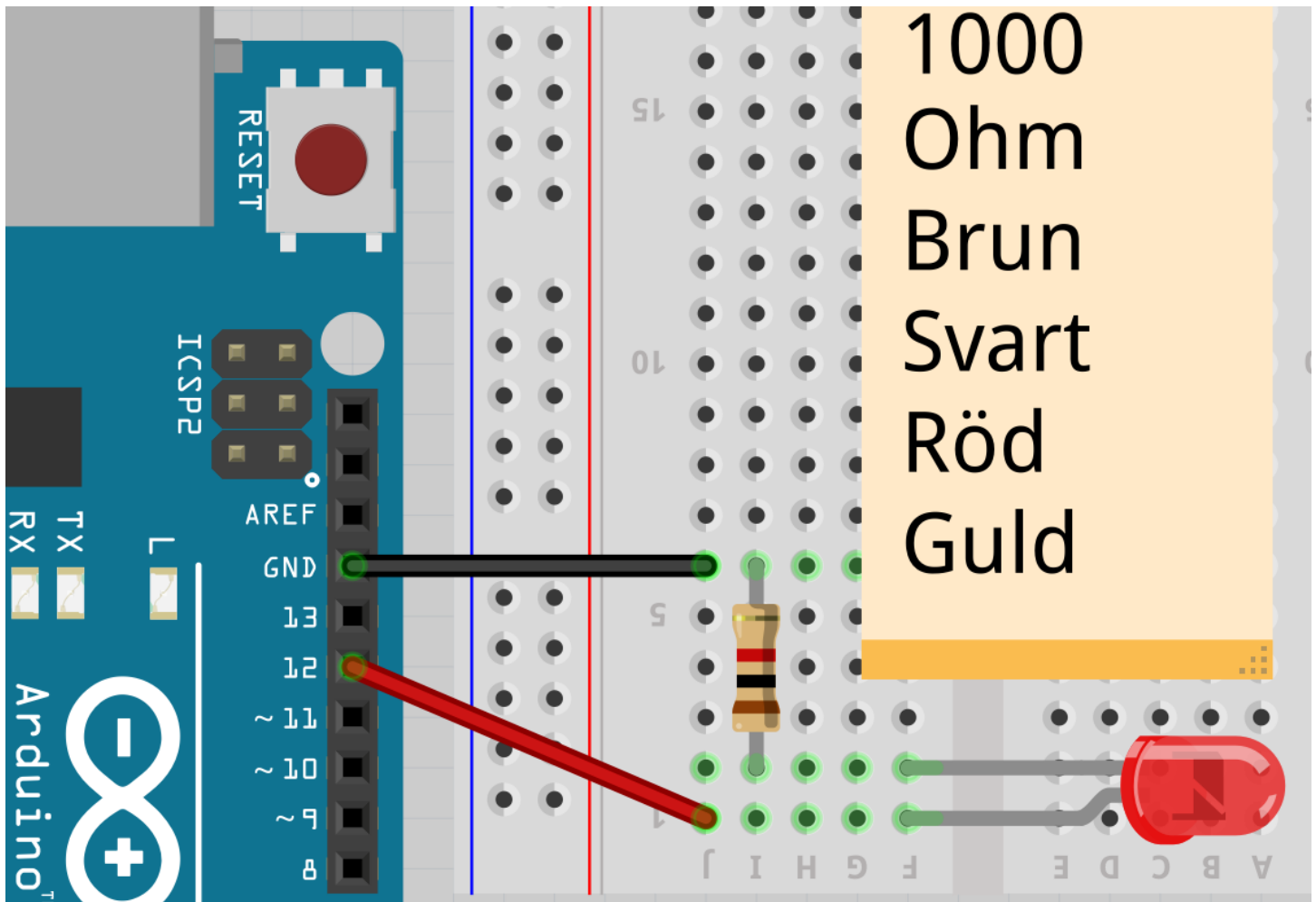


Figure 8: Blinka med LED på stift 12

1.12 Blink: Uppgift 5

Se nu till att lysdioden på stift 12 börjar blinka den andra.



LED är en förkortning av 'Light Emitting Diode'



Bra sagt! Bättre sent än aldrig!



LED menar på svensk: 'Ljus givande diod'

1.13 Blink: Lösning 5

```
void setup()
{
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(12, HIGH);
  delay(100);
  digitalWrite(12, LOW);
  delay(100);
}
```

1.14 Blink: Slutuppgift

Anslut lysdioden till stift 11 och få den att blinka 2 gånger per sekund.



Framgång? Visa detta för en vuxen för initialer!

Lektion 2: Blink Blink Blink

Den här lektionen kallas 'Blink Blink Blink', eftersom det är Blink lektionen, men med tre LEDs.

2.1: Blink Blink Blink: Blink

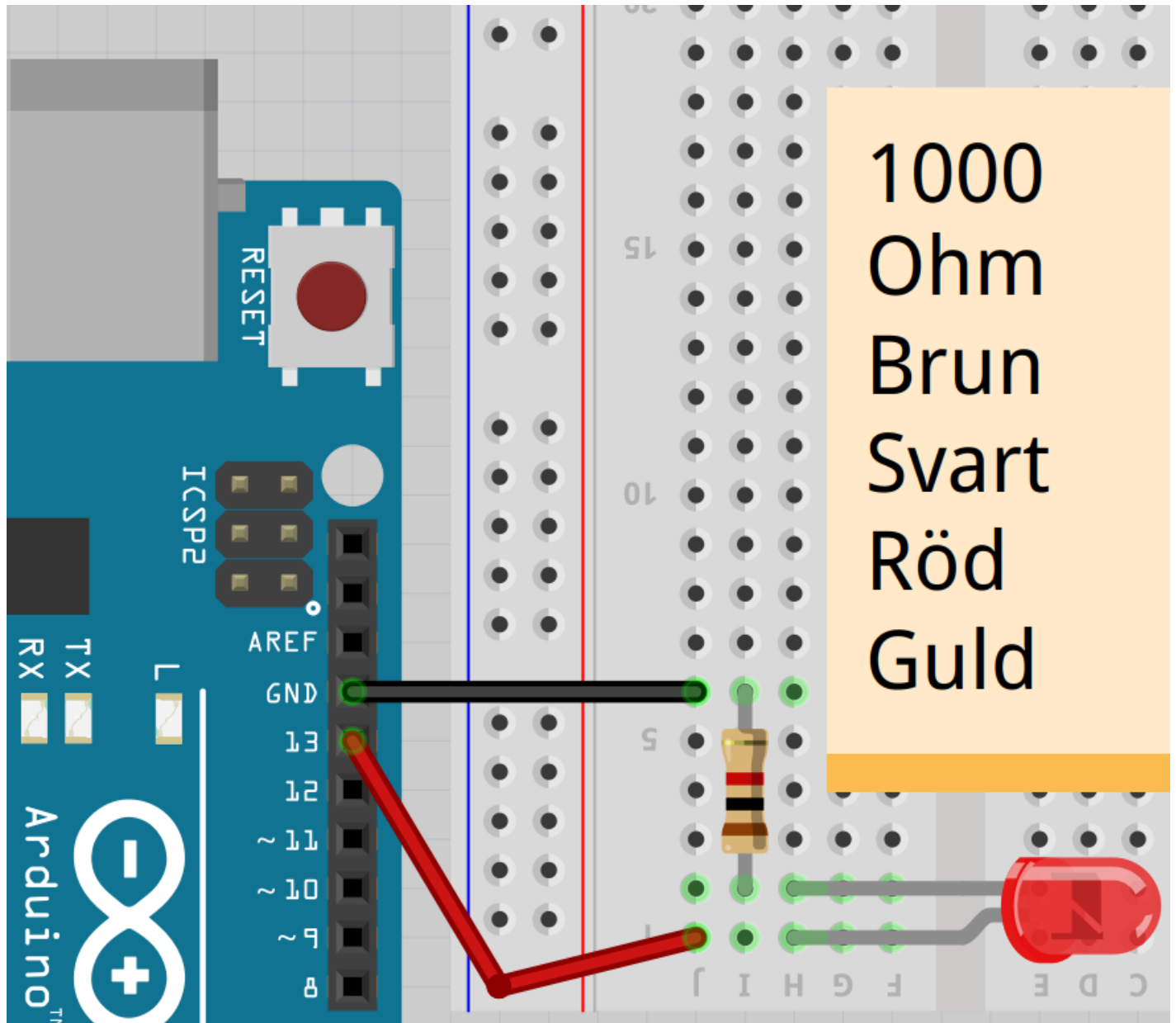


Figure 9: Blink

Här är Blinks kod på ett annat sätt:

```
const int led_stift = 13;

void setup()
{
  pinMode(led_stift, OUTPUT);
}

void loop()
{
  digitalWrite(led_stift, HIGH);
  delay(1000);
  digitalWrite(led_stift, LOW);
  delay(1000);
}
```

led_stift kallas en variabel: en bit datorminne med ett namn.



```
const int  
led_stift = 13;
```

‘Bästa dator, kom ihåg ett heltal som heter led_stift med initialvärdet 13.’

2.2: Blink Blink Blink: Uppgift 1

Anslut LEDen till stift 12 och ändra koden så att den blinkar.

2.3: Blink Blink Blink: Lösning 1

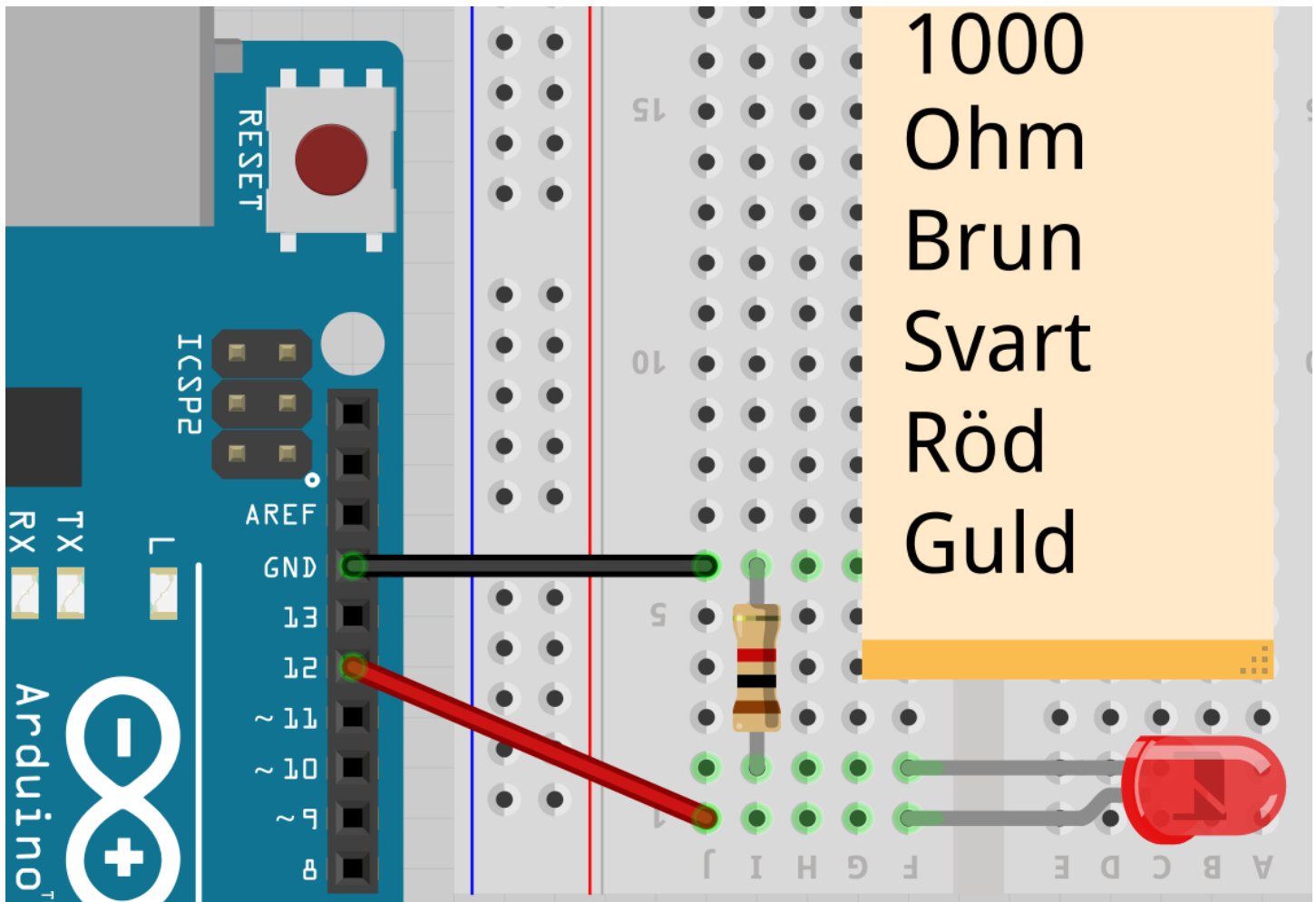


Figure 10: Blinka med LED på 12

Du behöver bara ändra en rad:

```
const int led_stift = 12;

void setup()
{
  // ...
}

void loop()
{
  // ...
}
```

2.4: Blink Blink Blink: Uppgift 2

Skapa en ny variabel för väntetiden, kallad `vantetid`, själv. `vantetid` är ett heltal med initialt värde 1000. Använd `vantetid` i raderna med `delay`.



Smart! Om du läser `vantetid` vet du vad det är till för. Vid 1000 vet du inte det

2.5: Blink Blink Blink: Lösning 2

```
// ... [gör led_stift att vara 12]
const int vantetid = 1000;

void setup()
{
  // ...
}

void loop()
{
  // ... [sätter spänning på led_stift]
  delay(vantetid);
  // ... [stanger av spänning från led_stift]
  delay(vantetid);
}
```



// ... [nåt] betyder ‘koden du redan har där som handlar om vad som står inom parentes’

2.6: Blink Blink Blink: Ansluter Blink Blink Blink

Nu är det dags att ansluta Blink Blink Blink:

- Koppla ur USB-kabeln från datorn, så att Arduino inte längre har ström
- Anslut delarna enligt bilden

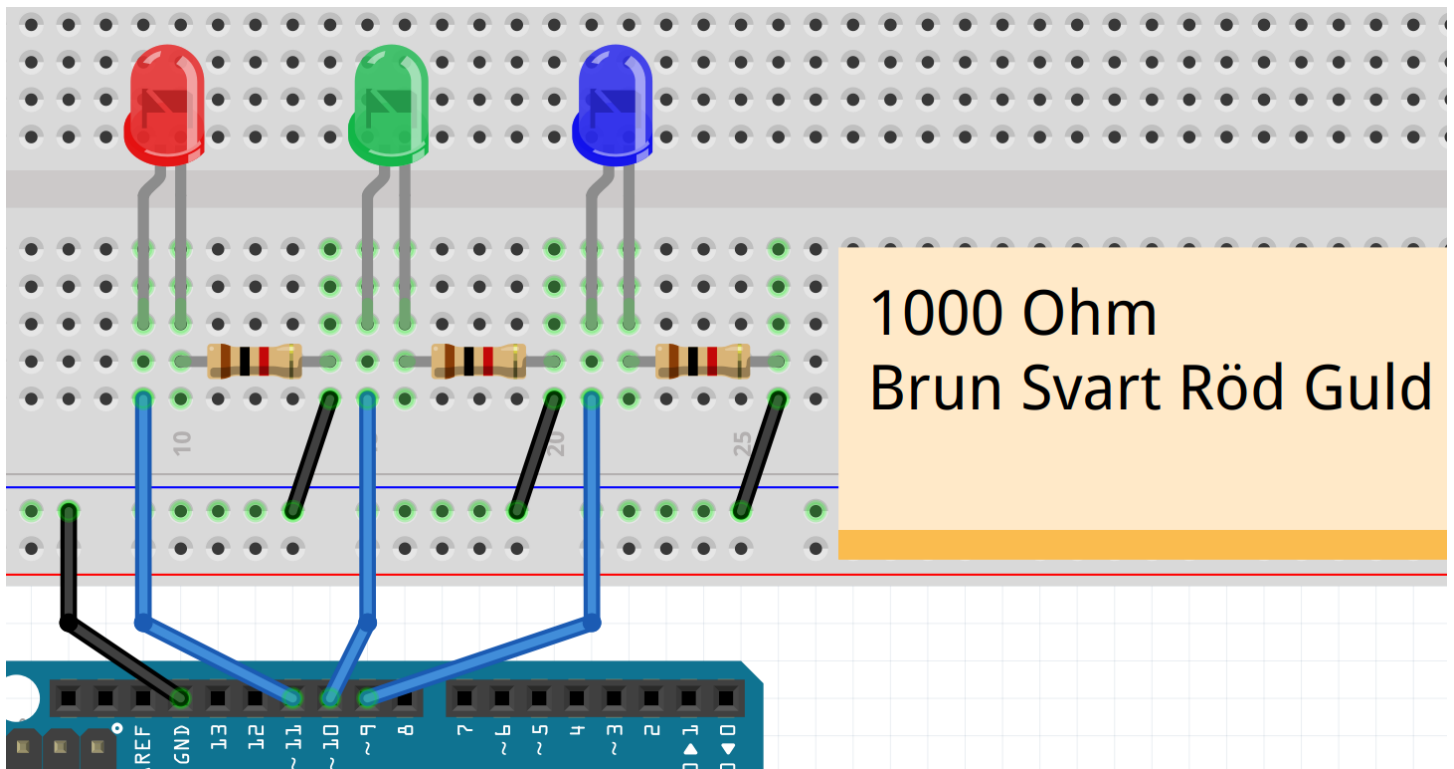


Figure 11: BlinkBlinkBlink

2.7: Blink Blink Blink: Uppgift 3

Koppla in Blink Blink Blink. Byt namn på variabeln `led_stift` till `led_stift_1`. och se till att det har rätt startvärde.

2.8: Lösning 3

```
const int led_stift_1 = 11;
// ... [min väntetid]

void setup()
{
  pinMode(led_stift_1, OUTPUT);
}

void loop()
{
  digitalWrite(led_stift_1, HIGH);
  // [vänta väntetid millisecond]
  digitalWrite(led_stift_1, LOW);
  // [vänta väntetid millisecond]
}
```



Programmerare använder många variabler eftersom det gör koden lättare att förstå.

2.9: Blink Blink Blink: Uppgift 4

Skapa en ny variabel `led_stift_2`. Låt först den första LEDen tändas och släckas, låt sedan den andra LEDen tändas och släckas.



Stavningen `'led_stift_2'` kallas `'snake case'`: `'snake'` är engelska för orm



Skulle du skriva `pinLed2`, heter det `'camel case'`: `'camel'` är engelska för kamel



Jag tänker båda är jämt bra

2.10: Blink Blink Blink: Lösning 4

```
const int led_stift_1 = 11;
const int led_stift_2 = 10;
// ...

void setup()
{
  pinMode(led_stift_1, OUTPUT);
  pinMode(led_stift_2, OUTPUT);
}

void loop()
{
  digitalWrite(led_stift_1, HIGH);
  // ... [vänta väntetid millisecond]
  digitalWrite(led_stift_1, LOW);
  // ... [vänta väntetid millisecond]
  digitalWrite(led_stift_2, HIGH);
  // ... [vänta väntetid millisecond]
  digitalWrite(led_stift_2, LOW);
  // ... [vänta väntetid millisecond]
}
```

2.11: Blink Blink Blink: Uppgift 5

Skapa en tredje variabel `led_stift_3`. Låt nu alla lampor blinka samtidigt: alla på, sedan alla av.



Det här är ganska mycket att skriva! Senare får du lära dig hur detta kan göras smartare



Korrekt! Arrays låter dig lagra flera tal i en variabel

2.12: Blink Blink Blink: Lösning 5

```
// ... [skapa vantetid, led_stift_1 en led_stift_2]
const int led_stift_3 = 9;

void setup()
{
  // ... [led_stift_1 en led_stift_2 ger spänning]
  pinMode(led_stift_3, OUTPUT);
}

void loop()
{
  // ... [sätter spänning på LED 1 och 2]
  digitalWrite(led_stift_3, HIGH);
  // ... [vänta vantetid millisecond]
  // ... [släck spanning av från LED 1 en 2]
  digitalWrite(led_stift_3, LOW);
  // ... [vänta vantetid millisecond]
}
```

2.13: Blink Blink Blink: Slutuppgift

Låt nu lamporna gå i ett 'Knight Rider-mönster': 1-2-3-2. Det måste alltid finnas exakt ett ljus som brinner.



Knight Rider var en TV-serie med en talande bil.

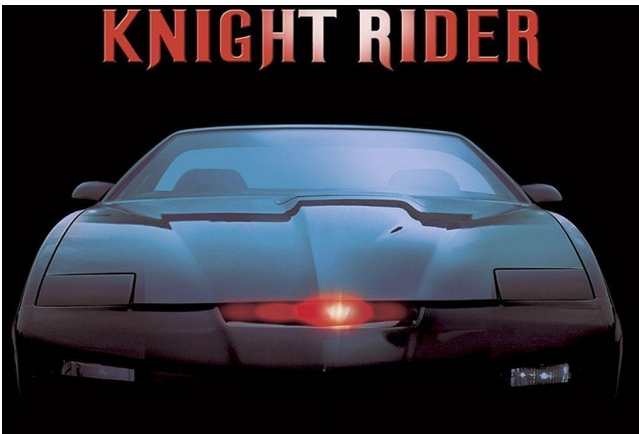


Figure 12: Knight Rider

Lektion 3: Knapp if ... else

I den här lektionen låter vi Arduino ställa frågor till sig själv.



Så Arduino blir en sorts filosof!

3.1. Knapp if ... else: Anslut

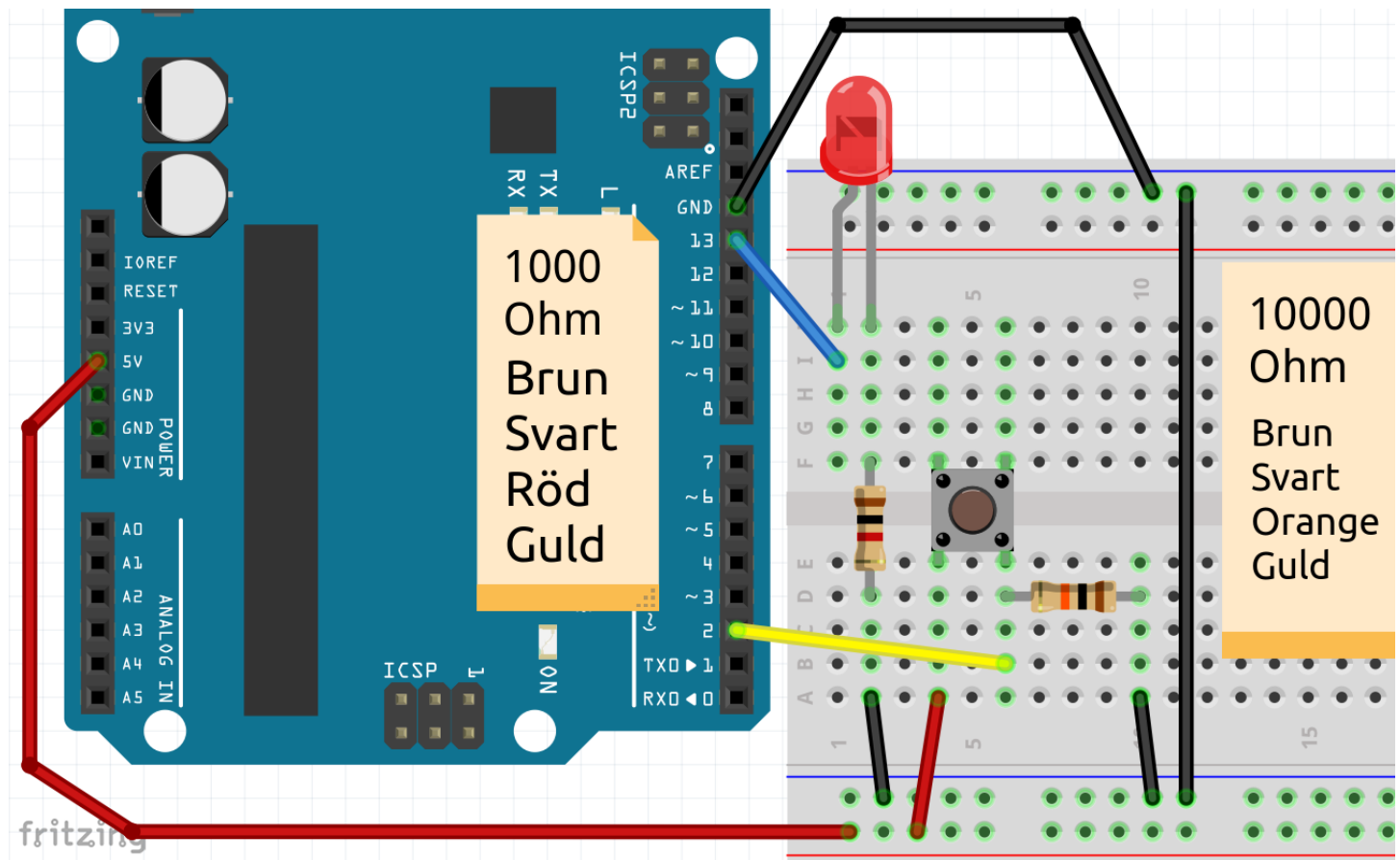


Figure 13: Flödesschema



Motståndet mot jord kallas ett 'Pull Down'-motstånd

3.2. Knapp if ... else: Kod

Här är koden för att få ljuset att tändas, när knappen trycks in:

```
const int led_stift = 13;
const int knapp_stift = 2;

void setup()
{
  pinMode(led_stift, OUTPUT);
  pinMode(knapp_stift, INPUT);
}

void loop()
{
  digitalWrite(led_stift, digitalRead(knapp_stift));
}
```



`digitalRead(knapp_stift)`



‘Ge mig HIGH om knapp_stift har spänning. Annars ge LOW’



`digitalWrite(led_stift, digitalRead(knapp_stift))`



‘Sätt spänningen till led_stift om knapp_stift har spänning’

3.3. Knapp if ... else: Fråga

När du trycker på knappen, släcks eller tänds LEDen?



‘Pull Down’-motståndet ser till att stift 2 är anslutet till GND när knappen inte trycks in

3.4. Knapp if ... else: Svara

Lysdioden tänds då.



Lysdioden tänds då

3.5. Knapp if ... else: if

Med if kan du få Arduino att göra något, om något så här:

```
if (digitalRead(knapp_stift) == HIGH)
{
    digitalWrite(led_stift_rod, HIGH);
}
else
{
    digitalWrite(led_stift_gron, LOW);
}
```

Denna kod kontrollerar om det finns spänning på `knapp_stift`. Om ja, lägg sedan Arduino spänning på `led_stift_red`. Annars (`else`) tar Arduino spänningen från `led_stift_gron`.



```
if (digitalRead(knapp_stift) == HIGH) {}
```



‘Kära dator, om det finns spänning på `knapp_stift`, gör detta inom parentes’



= kan uttalas som “sätta på”. “==” kan uttalas som “lika med”

3.6. Knapp if ... else: Uppgift 1

Detta är exempelkoden, som inte är helt färdig:

```
void setup()
{
  pinMode(4, INPUT );
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop()
{

  if (digitalRead(4) == HIGH)
  {
    digitalWrite(5, HIGH);
  }
  else
  {
    digitalWrite(6, LOW);
  }
}
```

Avsluta koden så att:

- När du trycker på knappen tänds lampan
- om du inte trycker på knappen slocknar lampan
- använd variabler för att namnge stiften



Det finns inget semikolon efter de runda parenteserna för `if`.

3.7. Knapp if ... else: Lösning 1

```
// ...

void setup()
{
  // ...
}

void loop()
{
  if (digitalRead(knapp_stift) == HIGH)
  {
    digitalWrite(led_stift, HIGH);
  }
  else
  {
    digitalWrite(led_stift, LOW);
  }
}
```



// ... betyder 'koden du redan har där'

3.8. Knapp if ... else: Uppgift 2

Anslut en andra grön LED. Gör koden så att:

- när du trycker på knappen tänds den första lampan och den andra LEDen släcks
- om du inte trycker på knappen slocknar den första lampan och den andra LEDen tänds
- Namnge variabeln för stiftet på den röda LEDen `led_stift_rod`, för grön LED `led_stift_gron`



Du kan skriva flera rader inom de krulliga parenteserna av ett `if`. Precis som mellan hängslen i `setup` och `loop`!



Glöm inte `else` ('annars')!

3.9. Knapp if ... else: Lösning 2

Bild Lösning för 'Knapp med två LEDer' visar hur man ansluter detta.

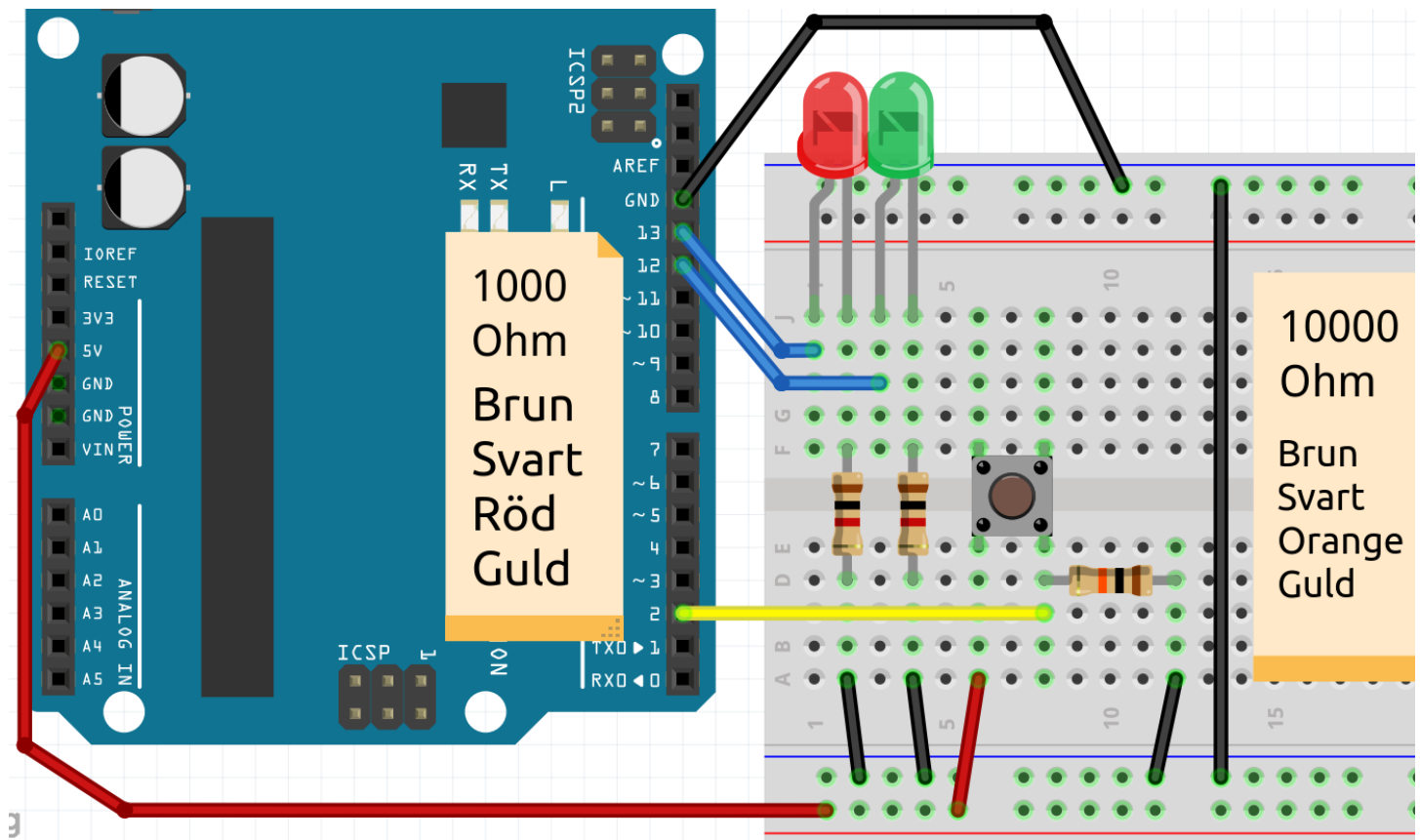


Figure 14: Lösning för 'Knapp med två LEDer'

Det här är koden:

```
const int led_stift_rod = 13;
const int led_stift_gron = 12;
// ...

void setup()
{
  pinMode(led_stift_rod, OUTPUT);
  pinMode(led_stift_gron, OUTPUT);
  // ...
}

void loop()
{
  if (/* ... */)
  {
    digitalWrite(led_stift_gron, LOW);
    digitalWrite(led_stift_rod, HIGH);
  }
  else
  {
    digitalWrite(led_stift_gron, HIGH);
    digitalWrite(led_stift_rod, LOW);
  }
  delay(10);
}
```



/* ... */ betyder också 'koden du redan har där'

3.10. Knapp if ... else: slutuppgift

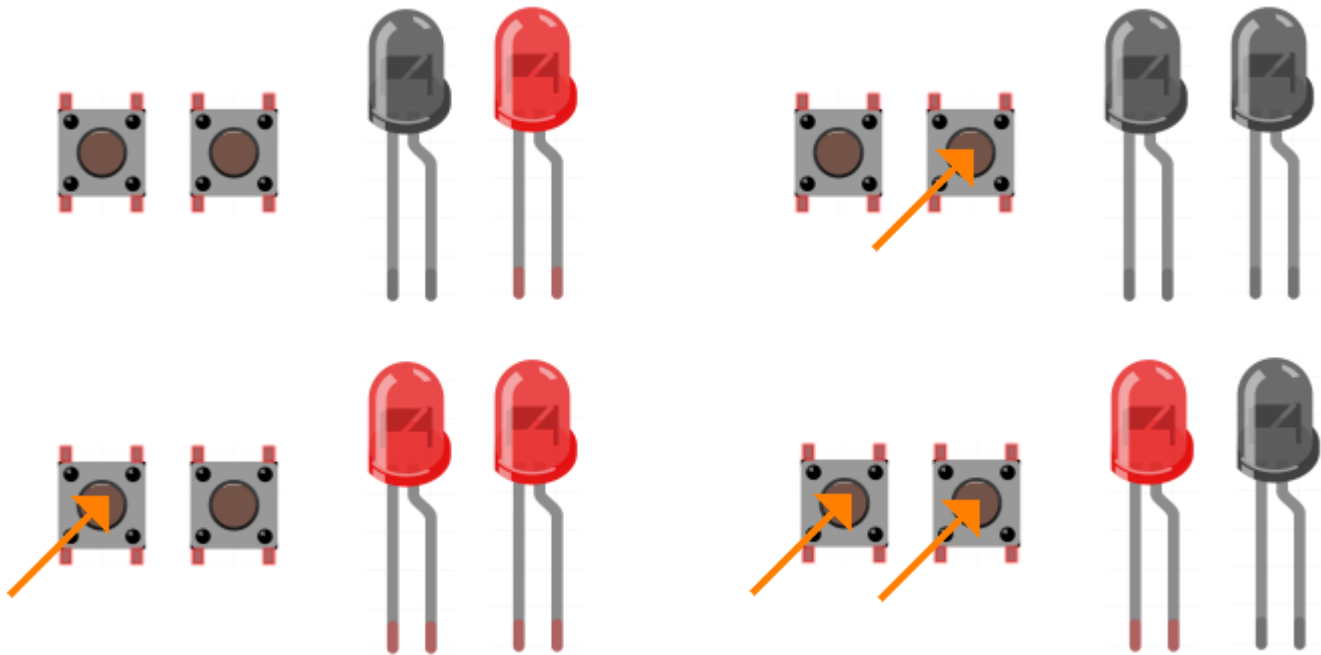


Figure 15: Avsluta uppdrag

Anslut en andra knapp. Göra koden så att

- om du trycker på den första knappen tänds den första lampan
- om du inte trycker på den första knappen slocknar den första lampan
- om du trycker på den andra knappen slocknar den andra lampan
- om du inte trycker på den andra knappen tänds den andra lampan



Du kan sätta `if` mer än en gång



För en andra knapp behöver du ett andra motstånd på tio tusen ohm

Lektion 4: Knapp flip

I den här lektionen kommer vi att lära oss hur man använder en variabel.



Lektionen kallas 'knappflip' eftersom du programmerar med en flip-flop

4.1. Knapp flip: Intro

Vi börjar med denna krets:

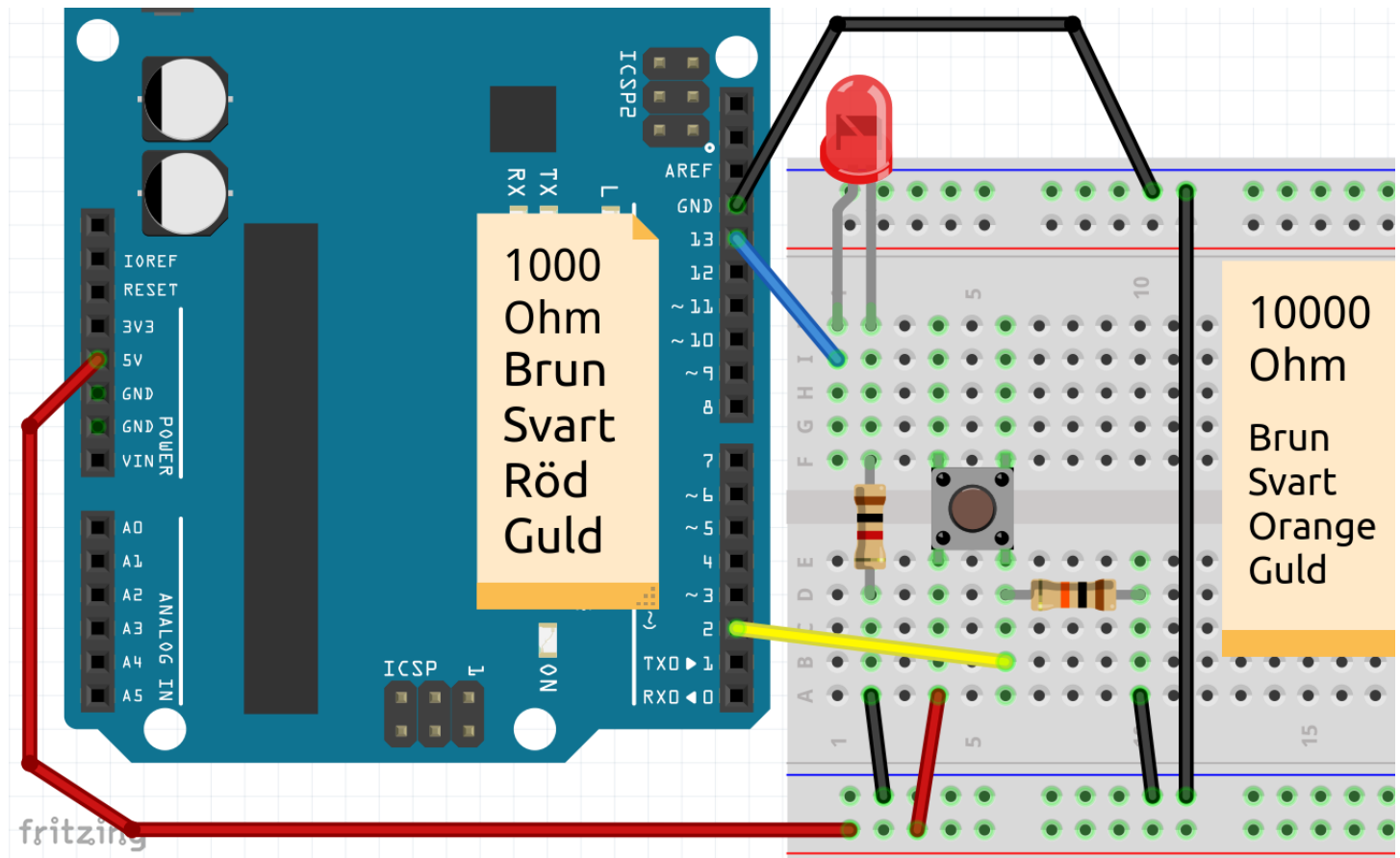


Figure 16: Flödesschema



Hej, jag känner igen den här kretsen!

Och vi använder den här koden:

```
const int led_stift = 13;
const int knapp_stift = 2;

void setup()
{
  pinMode(led_stift, OUTPUT);
  pinMode(knapp_stift, INPUT);
}

void loop()
{
  if (digitalRead(knapp_stift) == HIGH)
  {
    digitalWrite(led_stift, HIGH);
  }
  else
  {
    digitalWrite(led_stift, LOW);
  }
}
```



Hej, jag känner igen den här koden!

4.2. Knapp flip: Button forever on: uppgift 1

Överst i koden, ovanför `setup`, lägg:

```
boolean ar_pa = false;
```

I början av loop, använd ett `if` för att se om knappen är nedtryckt. Om så är fallet, ställ in `ar_pa` till `true`:

```
if (/* knappen är druckit */)  
{  
    ar_pa = true;  
}
```

Efter det första `if` i loop, använd ett `if` för att se om `ar_pa` är lika med `true`. Om ja, slå på LEDen. Om inte, stäng av LEDen.

```
if (ar_pa == true)  
{  
    // slå på LEDen  
}  
else  
{  
    // stäng av LEDen  
}
```

Vad ser du?

4.3. Knapp flip: Button forever on: lösning 2

```
// ...
boolean ar_pa = false;

// ...

void loop()
{
  if (digitalRead(knapp_stift) == HIGH)
  {
    ar_pa = true;
  }

  if (ar_pa == true)
  {
    digitalWrite(led_stift, HIGH);
  }
  else
  {
    digitalWrite(led_stift, LOW);
  }
}
```

När du trycker på knappen lyser LEDen för evig.

4.4. Knapp flip: lyser LEDen för evig: uppgift 2

Vi ska nu säga:

- När du trycker på knappen tänds LEDen och förblir tänd
- När du sedan trycker på knappen igen kommer LEDen att släckas och förbli släckt
- När du sedan trycker på knappen igen kommer LEDen att tändas och förbli tänd
- Och så vidare

Justera if som kontrollerar om knappen trycks ned:

```
if (digitalRead(knapp_stift) == HIGH)
{
  if (ar_pa == true)
  {
    ar_pa = false;
  }
  else
  {
    ar_pa = true;
  }
}
```

Ladda upp koden. Vad ser du? Om du ser något konstigt så stämmer det!



Vad ser du?



Om du ser något konstigt så stämmer det!

4.5. Knapp flip: lyser LEDen för evig: lösning 2

```
// ...

void loop()
{
  if (digitalRead(knapp_stift) == HIGH)
  {
    if (ar_pa == true)
    {
      ar_pa = false;
    }
    else
    {
      ar_pa = true;
    }
  }

  // ...
}
```



När du trycker på knappen dimmer LEDen. När du släpper knappen kommer LEDen antingen att vara tänd eller släckt för alltid



Detta beror på att Arduino snabbt och ofta tänder och släcker ljuset

4.6. Knapp flip: Knappen på för alltid med finkontroll: uppgift 2

Låt oss göra knappen mer lyhörd: släpp Arduino när knappen trycks ned 200 millisekunders väntan.

4.7. Knapp flip: Knappen på för alltid med finkontroll: lösning 2

I if när knappen trycks ned, lägg till en `delay(200);`-regel. Detta kan göras före eller efter if-satserna med `är_på`.

```
// ...

void loop()
{
  if (digitalRead(knapp_stift) == HIGH)
  {
    // ... [if (ar_pa == true) ...]

    delay(200);

    // ... [if (ar_pa == true) ...]
  }

  // ...
}
```


4.8. Knapp flip: styr två LEDer: uppgift

Vi kommer att ansluta en andra LED. Du kan byta genom att trycka på knappen mellan LEDer.

- Anslut en andra grön LED till stift 12
- I koden, ändra längst upp:

```
// led_stift är bort
const int led_stift_rod = /* stift numret */;
const int led_stift_gron = /* stift numret */;
// ... [knapp_stift stanner samma]
// ar_pa är tappat bort
int vilken_led_pa = 1;
```

- Se till att de två LEDerna och knappen hittas i koden, i `setup`.
- I koden, i loop, reagera annorlunda på knappen:

```
if (/* knappen är druckit */)
{
    vilken_led_pa = vilken_led_pa + 1;
    if (vilken_led_pa == 3)
    {
        vilken_led_pa = 1;
    }

    // ... [vänta 200 millisecond]
}
```

- I koden, i loop, reagerar nu på 'vilken_led_pa':

```
if (vilken_led_pa == 1)
{
    // Tänder på röd LEDen, släcker av gron LEDen
}
if (vilken_led_pa == 2)
{
    // Släcker av röd LEDen, tänder på gron LEDen,
}
```

4.9. Knapp flip: styr två LEDer: lösning

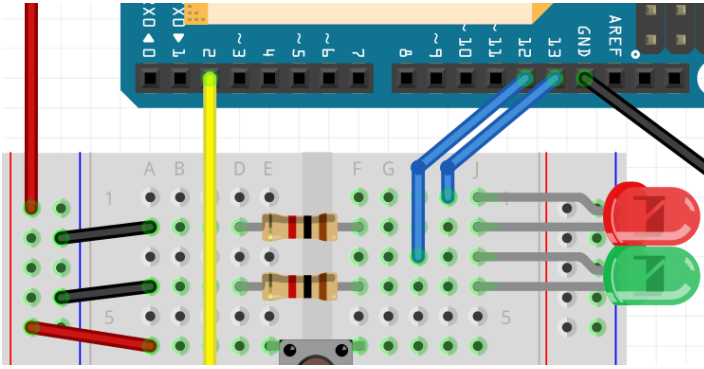


Figure 17: 4.9 Knapp flip: styr två LEDer: lösning

```

const int led_stift_rod = 13;
const int led_stift_gron = 12;
const int knapp_stift = 2;
int vilken_led_pa = 1;

void setup()
{
  pinMode(led_stift_rod, OUTPUT);
  pinMode(led_stift_gron, OUTPUT);
  pinMode(knapp_stift, INPUT);
}

void loop()
{
  if (/* knappen är druckit */)
  {
    vilken_led_pa = vilken_led_pa + 1;
    if (vilken_led_pa == 3)
    {
      vilken_led_pa = 1;
    }

    // ... [vänta 200 millisecond]
  }

  if (vilken_led_pa == 1)
  {
    digitalWrite(led_stift_rod, HIGH);
    digitalWrite(led_stift_gron, LOW);
  }
  if (vilken_led_pa == 2)
  {
    digitalWrite(led_stift_rod, LOW);
    digitalWrite(led_stift_gron, HIGH);
  }
}

```

4.10. Knapp flip: avsluta uppgift

Vi kommer att ansluta en tredje LED. Du kan byta genom att trycka på knappen från första, till andra, till tredje LED.