

Figure 1: Bok 2

#	Beskriving
5	Blink 4
6	Uppladdningsknapp
7	Potmeter och joystick
8	LDR

Contents

Förord	1
Lektion 5: Blink 4	2
Lektion 6: Laddningsknapp	11
Lektion 7: Potentiometer och joystick	24
8. LDR	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 5: Blink 4

I dessa läs använder vi 4 lysdioder och funktioner

5.1 Uppgift

- Gör en koppling med 2 LEDs, varje LED skulle ha en egen motstånd från 1000 Ohm
- Slut den 1:e LED på stift 2
- Slut den 2:e LED på stift 3
- Ladda upp denna kod:

```
const int stift_led_1 = 2;
const int stift_led_2 = 3;
const int vantetid = 1000;

void setup()
{
  pinMode(stift_led_1, OUTPUT);
  pinMode(stift_led_2, OUTPUT);
}

void loop()
{
  digitalWrite(stift_led_1, HIGH);
  digitalWrite(stift_led_2, LOW);
  delay(vantetid);
  digitalWrite(stift_led_1, LOW);
  digitalWrite(stift_led_2, HIGH);
  delay(vantetid);
}
```

5.2 Lösning

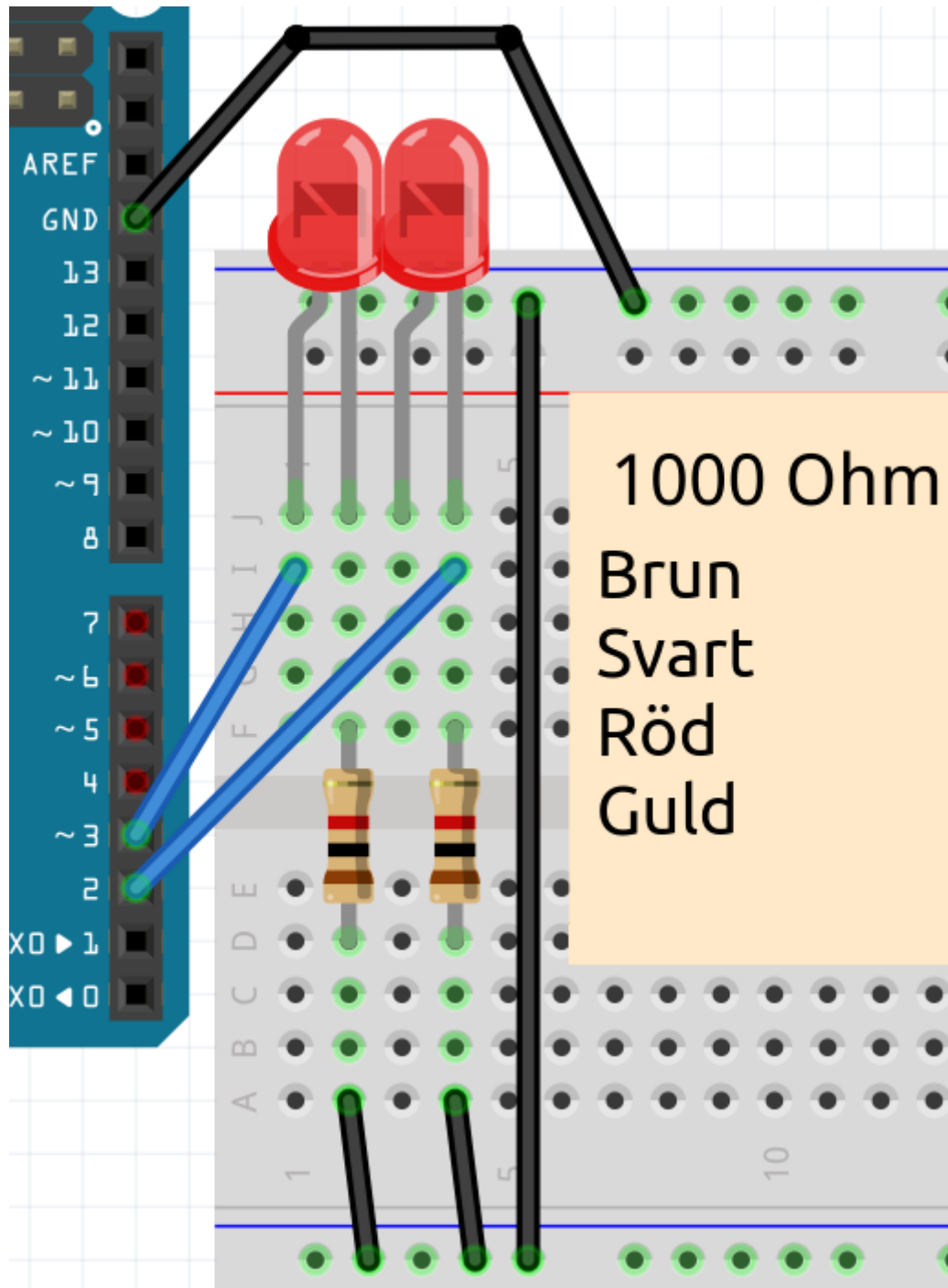


Figure 2: 5.2 Lösning

5.3 vanta_lite, uppgift

Vi skriver vår första funktion!

- Lägg till en kod tå, ovan loop:

```
void vanta_lite()
{
    delay(vantetid);
}
```

- I loop, ersätt två gånger `delay(vantetid);` genom `vanta_lite();`

5.4 vanta_lite, lösning

```
// ...

void setup()
{
    // ...
}

void vanta_lite()
{
    delay(vantetid);
}

void loop()
{
    // ... [ljuser bara LED 1]
    vanta_lite();
    // ... [ljuser bara LED 1]
    vanta_lite();
}
```

5.5 ljuser_bara_led_1, uppgift

- Skriv en funktion, `ljuser_bara_led_1`, för att säkerställa att enbart LED 1 märket (ofta: LED 2 måste ut)
- Använd `ljuser_bara_led_1` i loop

5.6 ljuser_bara_led_1, lösning

```
void ljuser_bara_led_1()
{
    digitalWrite(stift_led_1, HIGH);
    digitalWrite(stift_led_2, LOW);
}

void loop()
{
    ljuser_bara_led_1();
    // ...
}
```

5.7 ljuser_bara_led_2, uppgift

- Skriv en funktion, `ljuser_bara_led_2`, för att säkerställa att enbart LED 2 märket (ofta: LED 1 måste ut)
- Använd `ljuser_bara_led_2` i loop

5.8 ljuser_bara_led_2, lösning

```
void ljuser_bara_led_2()
{
    digitalWrite(stift_led_1, LOW);
    digitalWrite(stift_led_2, HIGH);
}

void loop()
{
    // ...
    ljuser_bara_led_2();
}
```

5.7 ljuser_bara_led_3, uppgift

- Bygg upp en tredje LED-lampa på stift 4
- Gör en ny variabel `stift_led_3` för denna LED
- Skriv en funktion, `ljuser_bara_led_3`, för att säkerställa att enbart LED 3 ljuser (på annan sätt: lysdioder 1 och 2 måste vara avstängad)
- Använd `ljuser_bara_led_3` i `loop`
- I `loop`, låt först enbart LED 1 brinna, vänta lite, låt bara LED 2 brinna, vänta lite, låt bara LED 3 brinna, vänta lite

5.8 ljuser_bara_led_3, lösning

```
// ...
const int stift_led_3 = 4;

void setup()
{
    // ...
    pinMode(stift_led_3, OUTPUT);
}

void ljuser_bara_led_1()
{
    // ...
    digitalWrite(stift_led_3, LOW);
}

void ljuser_bara_led_2()
{
    // ...
    digitalWrite(stift_led_3, LOW);
}

void ljuser_bara_led_3()
{
    digitalWrite(stift_led_1, LOW);
    digitalWrite(stift_led_2, LOW);
    digitalWrite(stift_led_3, HIGH);
}

void loop()
{
    // ...
    ljuser_bara_led_3();
    vanta_lite();
}
```

5.9 slutuppgift

- Slut en fyra LEDje till, på stift 5
- Gör en ny variabel `stift_led_4` för denna LED
- Skriv en funktion, `ljuser_bar_led_4`, för att säkerställa att enbart LED 4 märket (på annat sätt: Lysdioder 1 och 2 och 3 måste vara ut)
- Använd `ljuser_bar_led_4` i `loop`
- I `loop`, gör ett Nightrider-mönster: lamporna måste gå i 1-2-3-4-3-2. Det måste alltid finnas exakt ett ljus som brinner.

Lektion 6: Laddningsknapp

I den här lektionen kommer vi att använda en knapp, lysdioder och en funktion som returnerar ett värde.

6.1 Laddningsknapp: uppgift 1

- Du behöver inte ansluta någonting!
- Ladda upp denna kod:

```
const int vantetid = 1000;

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

void loop()
{
  Serial.println("Hej");
  delay(vantetid);
}
```

- Efter uppladdning, klicka på “Serial Monitor”

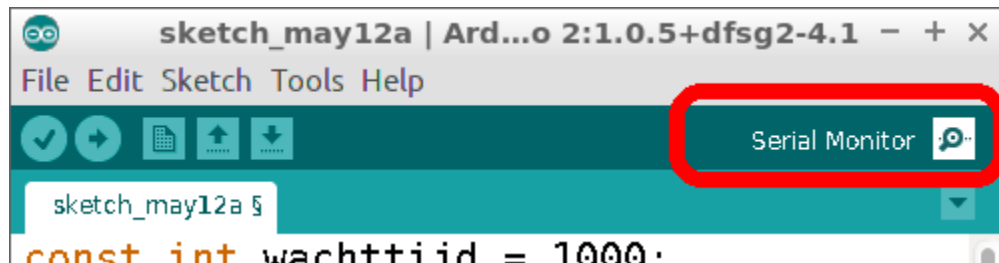
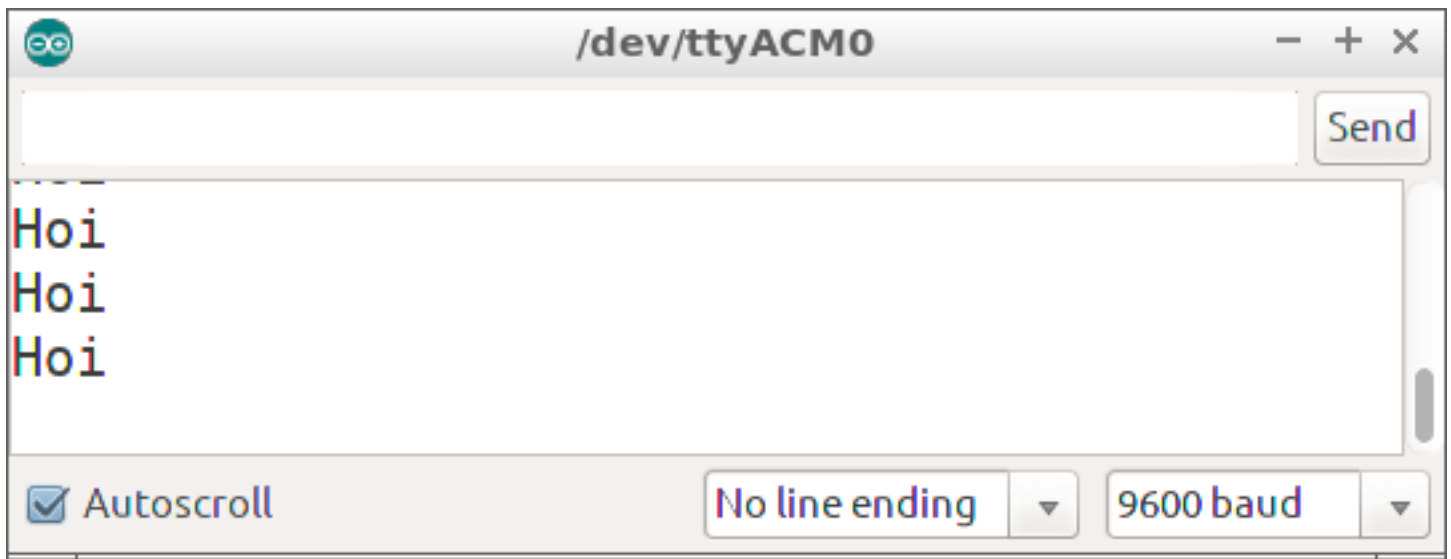


Figure 3: Klicka på ‘Serial Monitor’

Vad ser du?

6.2 Laddningsknapp: Lösning 1

Du ser Arduino säga "Hej"!



6.3 Ladda knapp: vanta_lite och visar_knapp_varde, uppgift

- Skriv en funktion `vanta_lite`: i denna funktion väntar Arduino "väntetid" millisekunder
- Skriv en funktion `visar_knapp_varde`: i denna funktion säger Arduino (fortfarande) "Hej"
- Använd `visar_knapp_varde` och sedan `vanta_lite` i loop

6.4 Laddningsknapp: vanta_lite och visar_knapp_varde, lösning

```
// ...

void setup()
{
    // ...
}

void visar_knapp_varde()
{
    Serial.println("Hej");
}

void vanta_lite()
{
    delay(vantetid);
}

void loop()
{
    visar_knapp_varde();
    vanta_lite();
}
```

6.5 Laddningsknapp: Knapp, Uppgift

- Anslut en knapp till stift 2
- Skapa variabel `knapp_stift`
- I `setup`, säg med `pinMode` att `knapp_stift` är en INPUT
- Ersätt `visar_knapp_varde` med denna kod:

```
void visar_knapp_varde()
{
  if (digitalRead(knapp_stift) == HIGH)
  {
    Serial.println("Knappen ar druckit");
  }
}
```

6.6 Laddningsknapp: Knapp, lösning

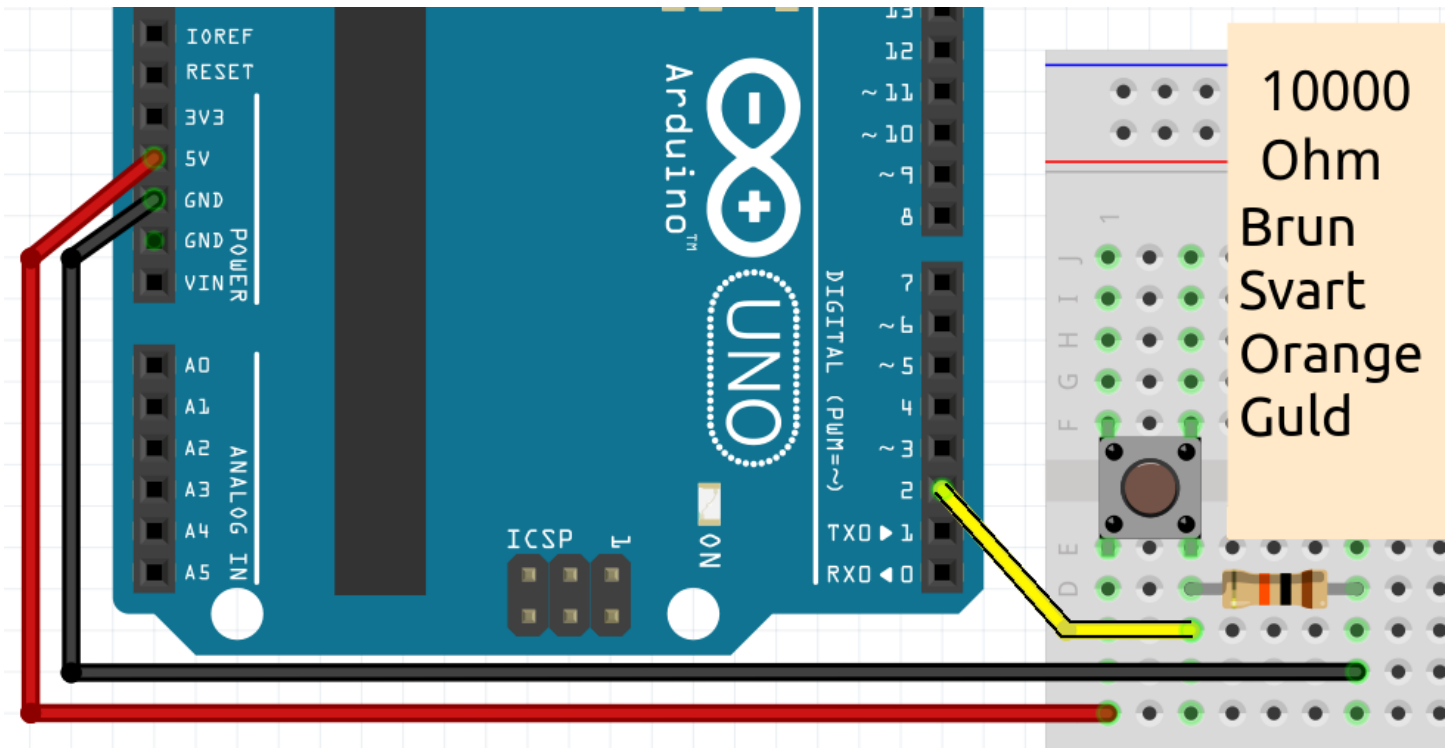


Figure 4: 6.6 Ladda knapp: knapp, lösning


```

// ...
const int knapp_stift = 2;

void setup()
{
    // ...
    pinMode(knapp_stift, INPUT);
}

void setup()
{
    // ...
}

void visar_knapp_varde()
{
    if (digitalRead(knapp_stift) == HIGH)
    {
        Serial.println("Knappen ar druckit");
    }
}

void loop()
{
    // ...
}

```

6.7 Laddningsknapp: släppknapp, uppgift

- I `visar_knapp_varde`, om knappen inte är nedtryckt, visa Arduino säga sedan “Knappen ar inte druckit”
- Ändra `vantetid` till 100 millisekunder

6.8 Laddningsknapp: släppknapp, lösning

```
const int vantetid = 100;
// ... [variabel knapp_stift]

// ...

void visar_knapp_varde()
{
  if (/* knappen är druckit */)
  {
    // ... [berättar knappen är druckit]
  }
  else
  {
    Serial.println("Knappen ar inte druckit");
  }
}
```

6.9 Ladda knapp: lastning, uppgift

- Skapa en variabel `lastning`. Detta är ett heltal som kan ändras, med initialt värde noll
- Skapa en ny funktion, `visar_lastning_display`. I denna funktion, värdet på `lastning` som skickas till den seriella monitorn. Du programmerar detta med:

```
Serial.println(lastning);
```

- Använd `visar_knapp_varde'`, `sedanvisar_lastning'` och sedan `vanta_lite` i loop

6.10 Ladda knapp: lastning, lösning

```
// ...
int lastning = 0;

void visar_lastning()
{
  Serial.println(lastning);
}

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

6.11 Uppladdningsknapp: `reagera_pa_knappen`, uppgift

- Skapa en ny funktion, `reagera_pa_knappen`. I `reagera_pa_knappen`: om knappen trycks ned, blir `lastning` 1 till. Du programmerar detta med:

```
lastning = lastning + 1;
```

- Använd `reagera_pa_knapp` mellan `visar_knapp_varde` och `visar_lastning` i loop

6.12 Lösning

```
void reagera_pa_knappen()
{
    if (digitalRead(knapp_stift) == HIGH)
    {
        lastning = lastning + 1;
    }
}

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

6.13: Slutuppgift

- I `reagera_pa_knappen`: om knappen släpps blir `lastning` noll igen
- Anslut en LED till stift 13
- LEDen tänds endast när `lastning` är mer än tio. Använd detta `if`-sats:

```
if (lastning > 10)
{
    // ...
}
```


Lektion 7: Potentiometer och joystick

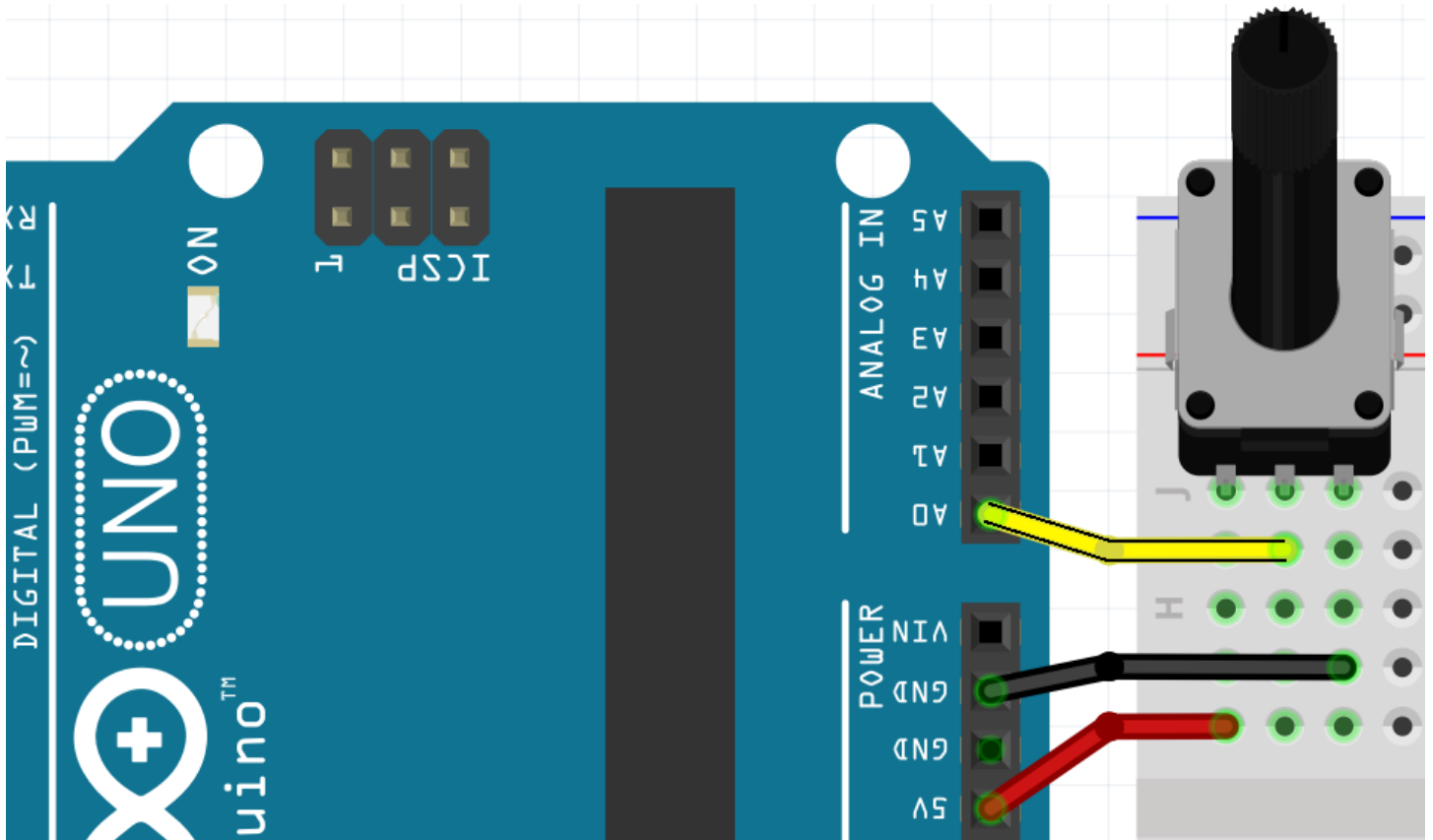
I den här lektionen kommer vi att använda en potentiometer, en joystick och en funktion som returnerar ett värde.

7.1 Potentiometer: anslut, uppgift

- Anslut en potentiometer. Gör följande anslutningar:

Potentiometer	Arduino
Vänster ben	5V
Mellanben	A0
Höger ben	GND

7.2 Potentiometer: anslut, lösning



7.3 Potentiometer: start, uppgift

- I `setup` startar den seriella monitorn vid 9600 baud
- Skapa en variabel `vantetid` med ett värde på 100
- Skapa en funktion `'effe_wait'` som programmerar 'väntetid' i millisekunder sen väntan
- Skapa en funktion `show_potentiometer_see`, som tar ordet `potentiometer` till seriella monitorkontroller
- I `loop`, använd `show potentiometer` och `effe_wait`

7.4 Potentiometer: uppstart, lösning

```
const int vantatid = 100; //milliseconden

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

void visar_potmeter()
{
  Serial.print("potmeter");
}

void vanta_lite()
{
  delay(vantatid);
}

void loop()
{
  visar_potmeter();
  vanta_lite();
}
```

7.5 Potentiometer: läs, uppgift

- Skapa en variabel 'potmeter_stift' med värdet 'A0'.
- I "setup", ställ in "pinMode" för "potmeter_stift" till "INPUT".
- Lägg till denna funktion:

```
int lasa_potmeter()  
{  
  return analogRead(potmeter_stift);  
}
```

- I show_potentiometer_see ersätt texten "potentiometer" med read_potentiometer()
- Ladda upp programmet och vrid potentiometern. Vilka siffror kommer ut?

7.6 Potentiometer: avläsning, lösning

```
const int potmeter_stift = A0;
//...

void setup()
{
    // ...
    pinMode(potmeter_stift, INPUT);
}

int lasa_potmeter()
{
    return analogRead(potmeter_stift);
}

void visar_potmeter()
{
    Serial.print(lasa_potmeter());
}
```

Siffrorna som kommer från `lasa_potmeter` är mellan noll och 1024.

7.7 Potentiometer: styrning, uppgift

- Anslut en LED till stift 11
- Skapa en variabel 'led_stift' med rätt värde
- I "setup", ställ in "pinMode" för "led_stift" till "OUTPUT".
- I show_potentiometer_see lägg till denna rad:

```
analogWrite(led_stift, lasa_potmeter());
```

- Ladda upp och vrid potentiometern. Vad ser du?

7.8 Potentiometer: styrning, lösning

```
// ...  
const int led_stift = 11;  
  
void setup()  
{  
  // ...  
  pinMode(led_stift, OUTPUT);  
}  
  
void visar_potmeter()  
{  
  // ...  
  analogWrite(led_stift, lasa_potmeter());  
}
```

Om du vrider på potentiometern ser du att LEDen lyser fyra gånger sätter på.

7.9 Potentiometer: bra styrning, uppgift

- Ändra följande kod...

```
analogWrite(led_stift, lasa_potmeter());
```

... till detta:

```
analogWrite(led_stift, lasa_potmeter() / 4);
```

- Vad ser du?
- Vad tror du att / betyder? Tips: var ser du sådana här ränder? i matte?

7.10 Potentiometer: bra styrning, lösning

- Du kan se att LEDen nu går från av till tänd när du slår på potentiometern vrider sig
- / betyder “delat med”. Detta är samma indelningslinje som med bråk och procent!

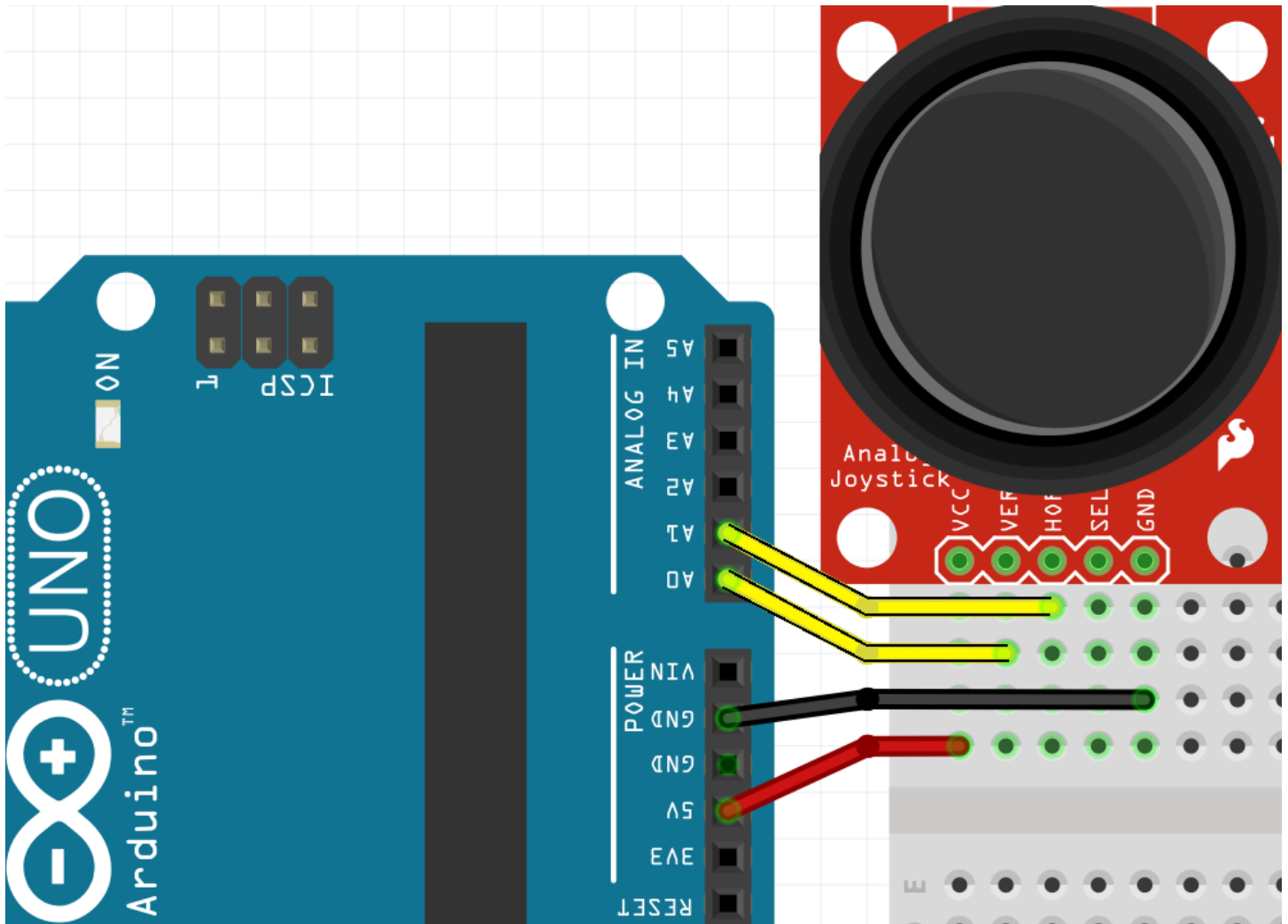
7.11 Potentiometer: anslut joystick, uppgift

- Byt ut potentiometern mot en joystick. Gör följande anslutningar:

Joystick	Arduino
VCC	5V
V	A0
H	A1
GND	GND

- Om du har anslutit denna rätt kan du nu styra LEDen med joysticken

7.11 Potentiometer: anslut joystick, lösning



7.12 Potentiometer: läs joystick, uppgift

I koden ersätter du texten `potentiometer` med `joystick_vertical`. För att göra detta, använd 'Find' (CTRL-F eller 'Edit | Find') och använd 'Replace All' ("Ersätt alla").

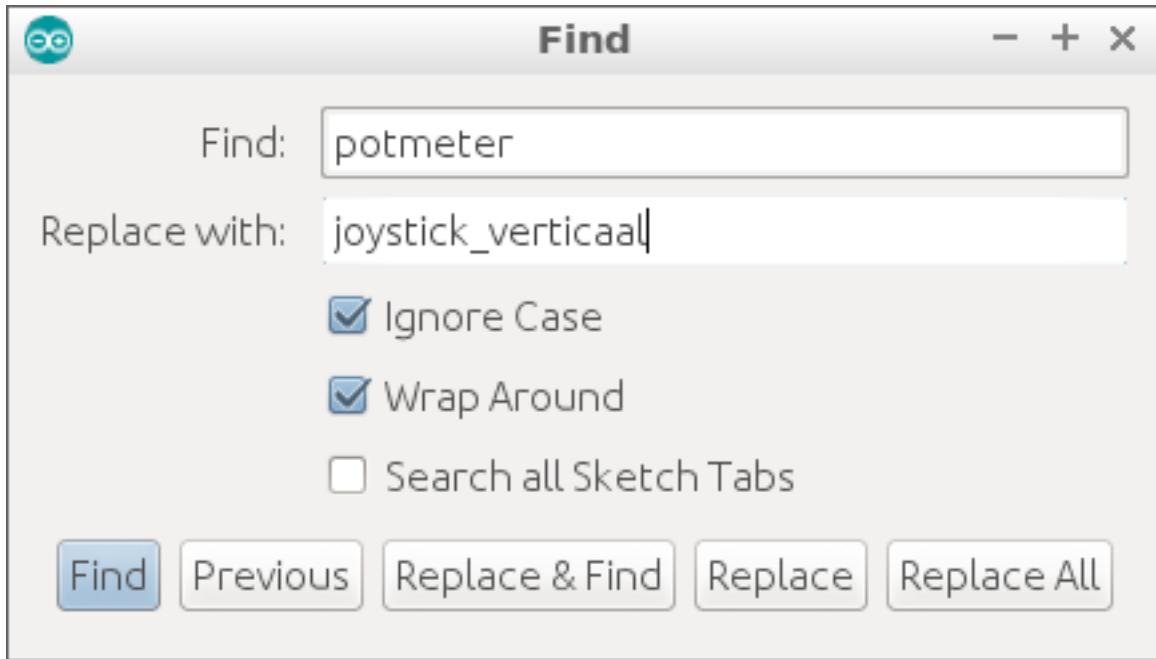


Figure 5: Sök, klicka här på 'Ersätt alla'

7.12 Potentiometer: läs joystick, lösning

```
const int vertikal_joystick_stift = A0;
// ...

void setup()
{
  pinMode(vertikal_joystick_stift, INPUT);
  // ...
}

int lasa_joystick_vertikalt()
{
  return analogRead(vertikal_joystick_stift);
}

void visar_joystick_vertikalt()
{
  Serial.print(lasa_joystick_vertikalt());
  analogWrite(led_stift, lasa_joystick_vertikalt() / 4);
}

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

7.12 Potentiometer: slutuppgift

- Anslut en andra LED
- Denna andra LED ska reagera som den första LEDen, men som joystickens flyttas horisontellt

8. LDR

Under den här lektionen använder vi en ljusberoende motstånd, kallas LDR.

8.1 Uppgift

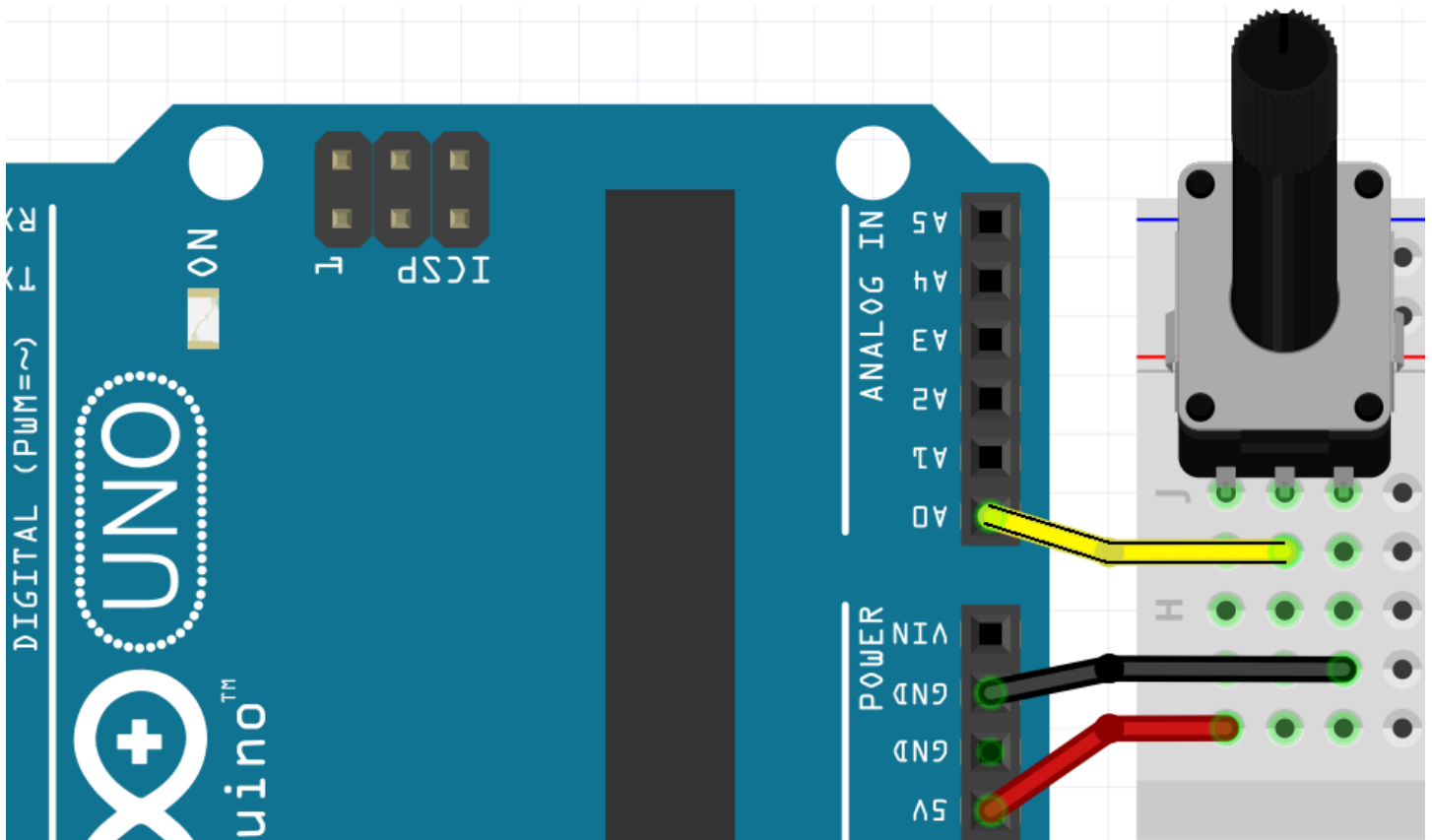
Bygg up:

- en potmeter, på A0

Programmerar:

- en `const` variabel `potmeter_stift` på riktigt stift
- att få värd av potmetern med `analogRead`
- att skriver värd av potmetern till Serial Monitor
- vänta 100 millisecond varje `loop`

8.2 Lösning



```
const int potmeter_stift = A0;

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

void loop()
{
  Serial.print(analogRead(potmeter_stift));
  delay(100);
}
```

8.3 Uppgift

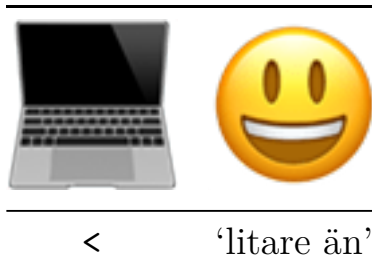
Skriv om programmet litegran för att ha en funktion som häter `visar_potmeter`.

`visar_potmeter` kan redan:

- att få värd av potmetern med `analogRead`
- att skriver värd av potmetern till Serial Monitor

Addera till `visar_potmeter`:

```
if (analogRead(potmeter_stift) < 512)
{
  Serial.print("Potmeter ar till vanster");
}
else
{
  Serial.print("Potmeter ar till hogger");
}
```



512 är bara i mitten från alla möjliga värder `analogRead` kan ger

8.4 Lösning

```
const int potmeter_stift = A0;

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

void loop()
{
  visar_potmeter();
  delay(100);
}

void visar_potmeter()
{
  Serial.print(analogRead(potmeter_stift));
  if (analogRead(potmeter_stift) < 512)
  {
    Serial.print("Potmeter ar till vanster");
  }
  else
  {
    Serial.print("Potmeter ar till hogre");
  }
}
```

8.5 Uppgift

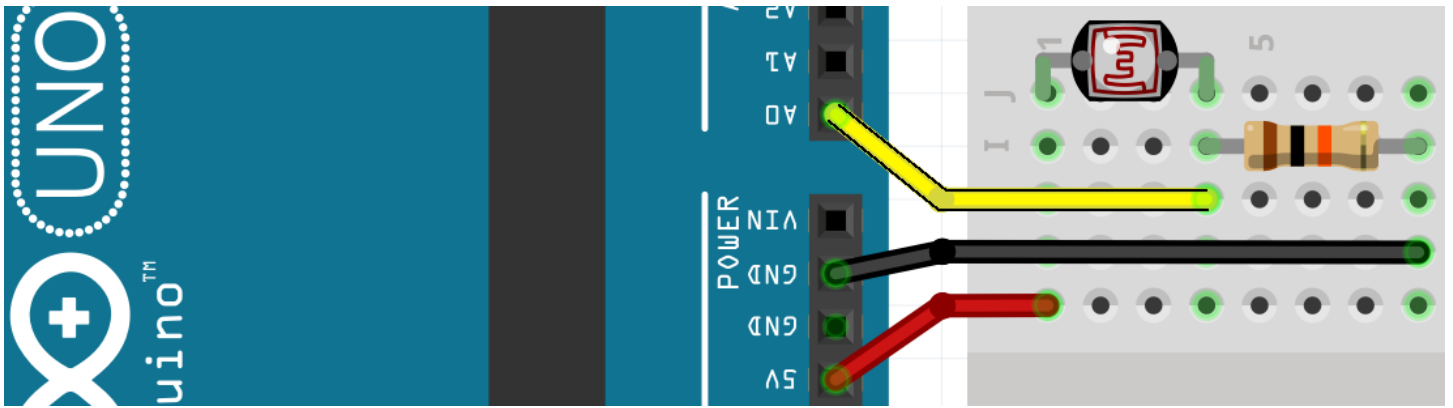
Byter potmeter om för en LDR. En LDR är ansluten sannalikt än en knapp:

- den första ben åker till 5V
- den andra ben går till en motstand från 10 kOhm, som går till GND
- den andra ben går till A0

Kör programmet med samma kod.

Vilken värder har LDR om du sta ljus iväg med din hand? Vilken värder har LDR om ljus är helt på den?

8.6 Lösning



8.7 Slutuppgift

Addera en LED på 13.

Om du håller din hand över LDRen:

- LEDen skulle ljuser upp
- Serial monitor skull säga 'Det ar morkt'

Om du inte håller din hand över LDRen:

- LEDen skulle vara släckt
- Serial monitor skull säga 'Det ar ljus'