

NYBÖRJARKURS · 5 TRÄFFAR

# Elektronik & Programmering med Arduino

Fem träffar. Fem moduler. Ett fungerande larm.

KURSÖVERSIKT

# Fem träffar. Fem moduler.



MODUL 01

## LED & krets

Digital output, Ohms lag



MODUL 02

## PWM & RGB

analogWrite, färgblandning



MODUL 03

## Digital input

Knapp, buzzer, logik



MODUL 04

## Analog input

Sensorer, Serial Monitor



MODUL 05

## Integration

Hackathon: bygg larmet

Varje träff bygger vidare på den föregående.

TRÄFF 1 AV 5

# LED & krets

Modul 1: **Digital output, Ohms lag**

DAGENS UPPLÄGG

# Idag — två timmar.

00:00 – 00:15

## Slutmålet & Arduinon

Vi tittar på det färdiga larmet och ramar in vad ett Arduino-kort egentligen är.

00:35 – 01:05

## Bygg kretsen & ladda upp första programmet

Praktiskt: LED + resistor på breadboarden, första uppladdningen från IDE:n.

01:10 – 02:00

## Din egen rytm & delresultat

Fri övning: ändra delay-värdena till ett eget mönster. Alla visar upp.

00:15 – 00:35

## Krets, spänning, ström

Kort genomgång av vad en krets är och varför LED:en behöver en resistor.

01:05 – 01:10

## Paus

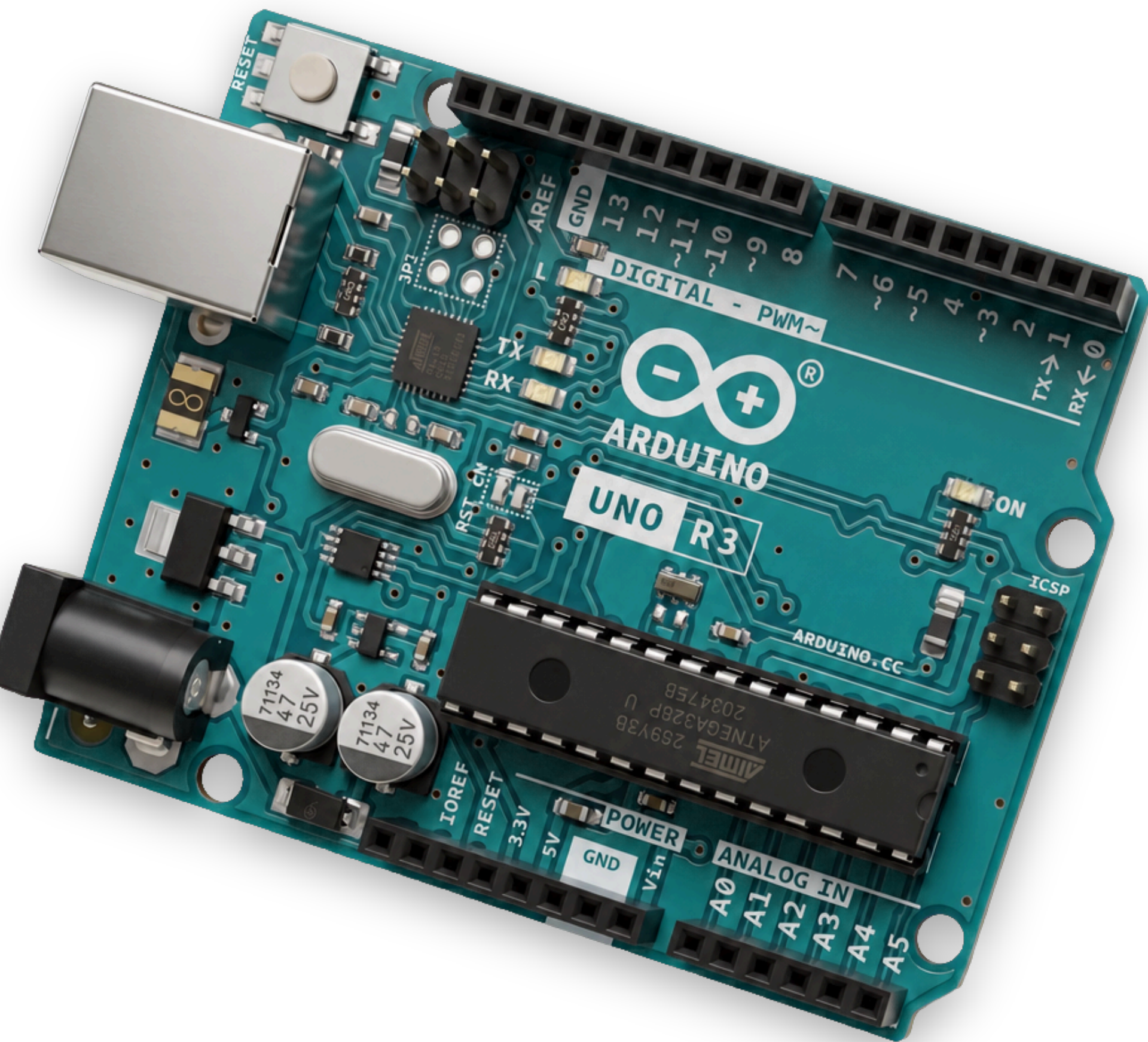
Vatten, WC, sträck. Tillbaka fem minuter senare.



SLUTMÅLET

## **Det här ska vi bygga.**

Ett larm som känner av mörker, lyser upp som stämningssljus och tjuter när någon rör det.



ARDUINO UNO R3

## Mikrokontrollern.

En **mikrocontroller** — processor, minne och in-/utgångar på ett chip.  
Kör ett enda program, helt förutsägbart.

Arduino UNO R3 • ATmega328P  
16 MHz • 32 KB flash • 2 KB SRAM

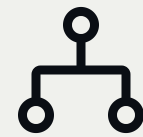
TRE GRUNDBEGREPP

# Spänning, ström, GND.



SPÄNNING

**Volt · V**



STRÖM

**Ampere · A**

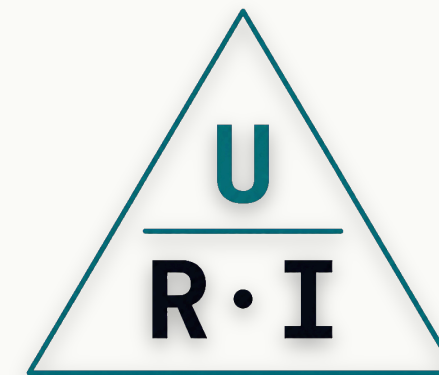
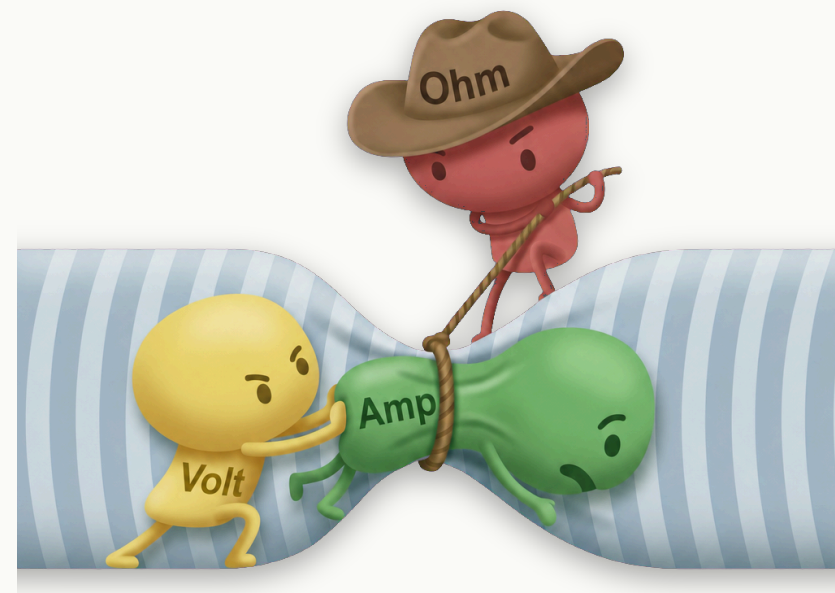


GND

**0 V · referens**

OHMS LAG

$$U = R \cdot I$$



täck variabeln ni söker

Volt trycker · Ohm stryper · Amp flödar

Räkneexempel och framspänningsfall: → [Bilaga F i kompendiet](#)

KRETSEN

# Kretsen — som ett vattensystem.

Strömmen går ut från en **utgångspinne** (t.ex. pin 13 när den är HIGH), genom komponenterna, och tillbaka till **GND**.

Utan bromsning: **LED:en brinner upp**.

Lösningen: en **strypventil** — en resistor på  $220\ \Omega$  i serie.

## MINNESREGEL

Ingen krets utan väg tillbaka till GND. Strömmen måste kunna fullborda varvet — annars händer ingenting.

# LED & resistor.

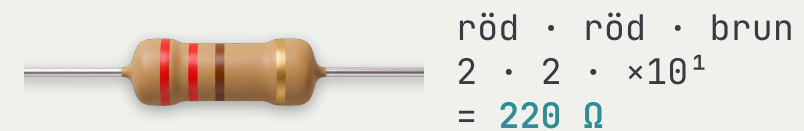
## LED:en

- **Har polaritet.** Kör du den baklänges lyser den inte.
- **Långt ben = anod (+)** → till signalpinnen (pin 13).
- **Kort ben + platt kant = katod (-)** → till GND.
- **Tål ca 20 mA.** Mer → den brinner upp permanent.

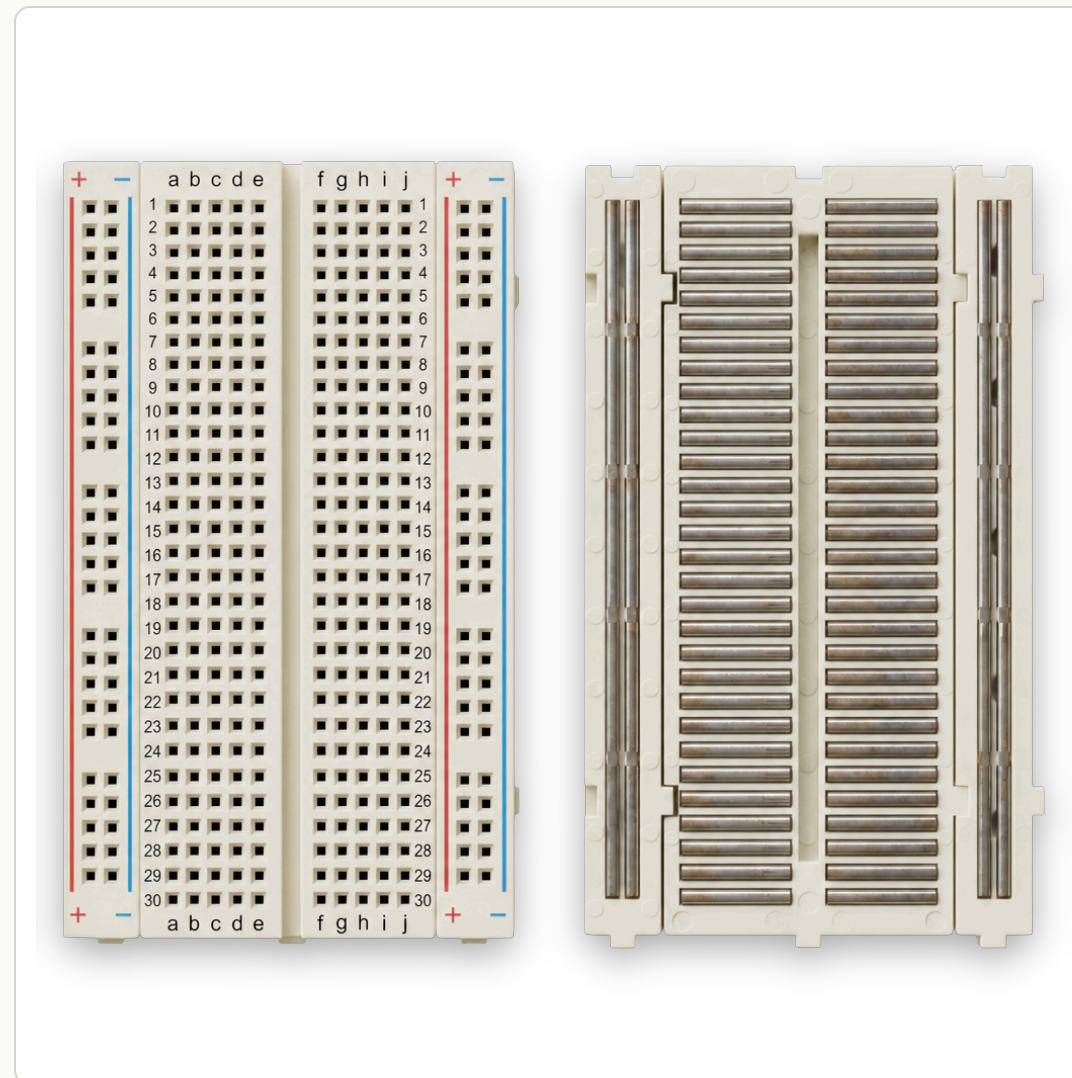
*Kör du den baklänges går den inte sönder — den lyser bara inte. Bara att vända.*

## Resistorn

- **Passiv komponent.** Ingen polaritet — koppla hur du vill.
- Värdet läses på **färgringarna**.
- **220 Ω (4-band)** = röd · röd · brun + guld.
- **220 Ω (5-band)** = röd · röd · svart · svart + brun.



# Fem hål = en nod.



- Fem hål i rad = en **nod**
- Raden bredvid = **egen** nod

- Gapet i mitten **bryter**
- **+/-**-skenorna går hela vägen

FELSÖKNING · REGEL #1

# Kontrollera raden först.

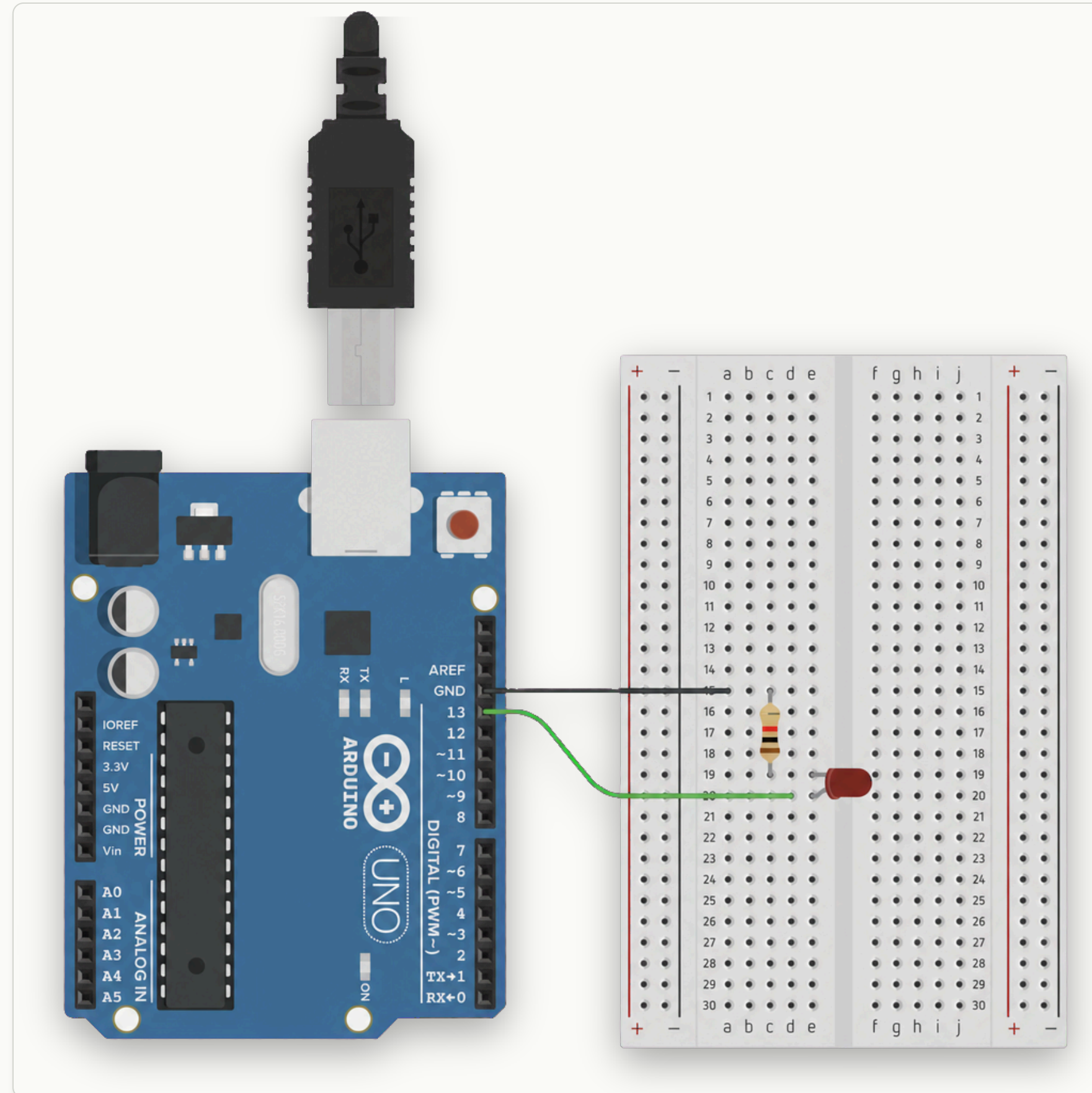
**90 %** av alla nybörjarfel  
är ett hål fel.

Kabeln och komponentbenet måste  
sitta på **samma rad**.

# Bygg kretsen.

- 01 Sätt LED:en på breadboarden. **Långt ben** i en ledig rad, **kort ben** i raden bredvid.
- 02 Kabel från **pin 13** på Arduinon → samma rad som långa benet.
- 03 **220  $\Omega$ -resistor** (4-band: röd-röd-brun; 5-band: röd-röd-svart-svart-brun). Ett ben i samma rad som korta LED-benet. Andra benet i valfri ledig rad.
- 04 Kabel från resistorns andra ben → **GND** på Arduinon.
- 05 Ladda upp [File](#) → [Examples](#) → [01.Basics](#) → [Blink](#). Lampan ska börja blinka.

# Så här ska det se ut.



D13 → LED långt ben · LED kort ben → 220 Ω → GND

# setup() och loop() .

## void setup()

Körs **en gång** när Arduinon startar eller när ni laddar upp. Används för konfiguration: vilka pinnar är utgångar, hastighet på Serial, etc.

## void loop()

Körs **om och om igen**, så länge kortet har ström. Här bor logiken: läs, reagera, vänta, upprepa.

Regel: exakt en setup() och en loop() per Arduino-kod. Aldrig fler.

```
1 void setup() {
2   pinMode(LED_BUILTIN, OUTPUT); // körs en gång vid uppstart
3 }
4
5 void loop() {
6   digitalWrite(LED_BUILTIN, HIGH); // körs om och om igen
7   delay(1000);
8   digitalWrite(LED_BUILTIN, LOW);
9   delay(1000);
10 }
```

TRE KOMMANDON

## Hela Blink — på tre rader.

```
1 pinMode(LED_BUILTIN, OUTPUT);
```

"LED\_BUILTIN är en [utgång](#)." Arduinos alias för pin 13 — där den inbyggda LED:en sitter. Körs i setup().

```
1 digitalWrite(LED_BUILTIN, HIGH);
```

"Sätt pinnen till [5 V](#)." HIGH = tänd (5 V), LOW = släckt (0 V). Körs i loop().

```
1 delay(1000);
```

"Vänta [1000 millisekunder](#)." 1000 ms = 1 sekund. Arduinon pausar allt annat under tiden.

Med bara dessa tre kan ni få en lysdiod att blinka i vilken rytm som helst.

# Din egen rytm.

Ändra `Blink` till ett mönster du själv väljer.

- **SOS-rytm:** `... — — — ...`
- **Ditt eget tempo** — snabb eller långsam, ojämn
- **Hjärtslag** — två snabba, sen paus

## TIPS FÖR BYGGET

Lägg tider i `const int`-variabler — då slipper du ändra många siffror när du justerar tempot. SOS växlar mellan kort blink (200 ms) och lång blink (600 ms), separerade av paus.

DELRESULTAT · TRÄFF 1

KLART

# Ni har byggt en blinkare.

*Er första rad kod som styr den fysiska världen.*

## Er första krets

LED + resistor + GND

## Er första kod

setup · loop · tre rader

## Er egen rytm

kod som styr världen

Ett varv runt rummet — visa upp ditt mönster  
för grannen.

NÄSTA TRÄFF



# Nästa gång...

Ni kan tända en lampa i en egen rytm.  
Nästa steg: blanda **färg** ur rött, grönt och blått.