

TINKEREN MET DE ARDUINO

Workshop 3. Spanningsverschillen, communicatie, beweging en
fun

VERVOLG TTT

Na de zomer

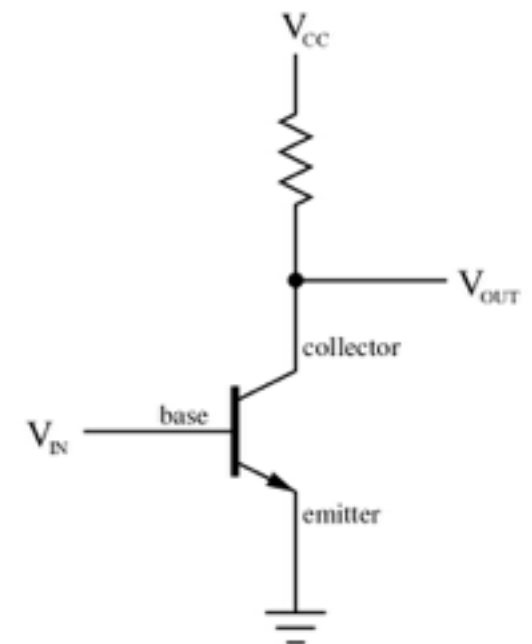
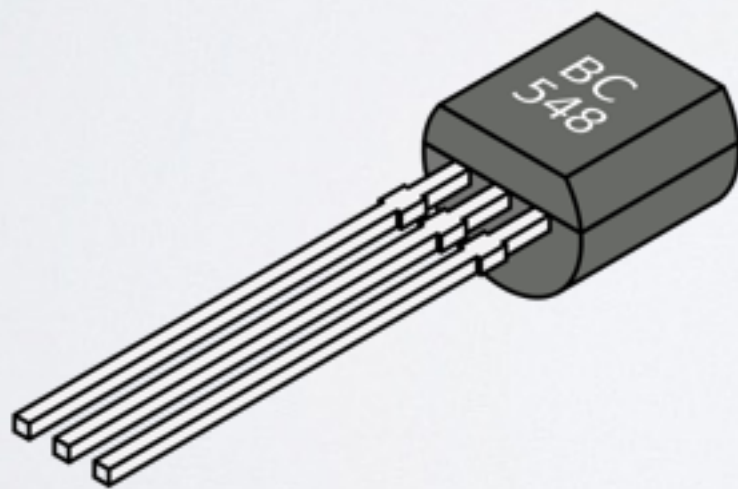
- geavanceerde technieken (int en milds)
- MySensors project (Bijv. weerstation) - extra onderdelen nodig
- MySensors Gateway bouwen
- Solderen
- De multimeter (hoe zit het met de gezamenlijke inkoopactie)
- Smart Devices met de Raspberry Pi

PROGRAMMA

- Enkele basis elektronica componenten
- Het schakelen van hogere spanningen
- Zet eens iets in beweging
- Communicatie (met elektronica en/of Arduino of Pi)

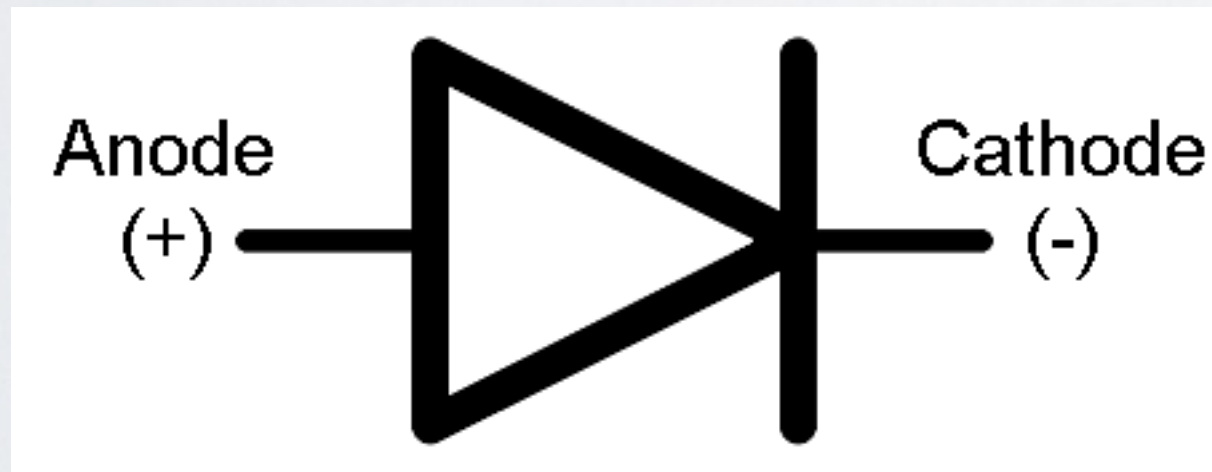
TRANSISTOREN

- Met behulp van een transistor, kun je de stroom van een schakeling aanzetten, hoger/lager schakelen en uitzetten, maar ook versterken. (NPN)
- Een transistor is in feite een elektronische schakelaar
- Dankzij de transistor hebben wij computers. (FlipFlop)
- Kunnen maar beperkte hoeveelheid stroom doorlaten (500mA). Behalve darlingtons (TIP120), maar die worden heel warm, waardoor ze niet efficiënt zijn.



DIODE

- Laat stroom maar in een richting door spert als de stroom omdraait
- Voorbeeld heb je al gezien (LED)

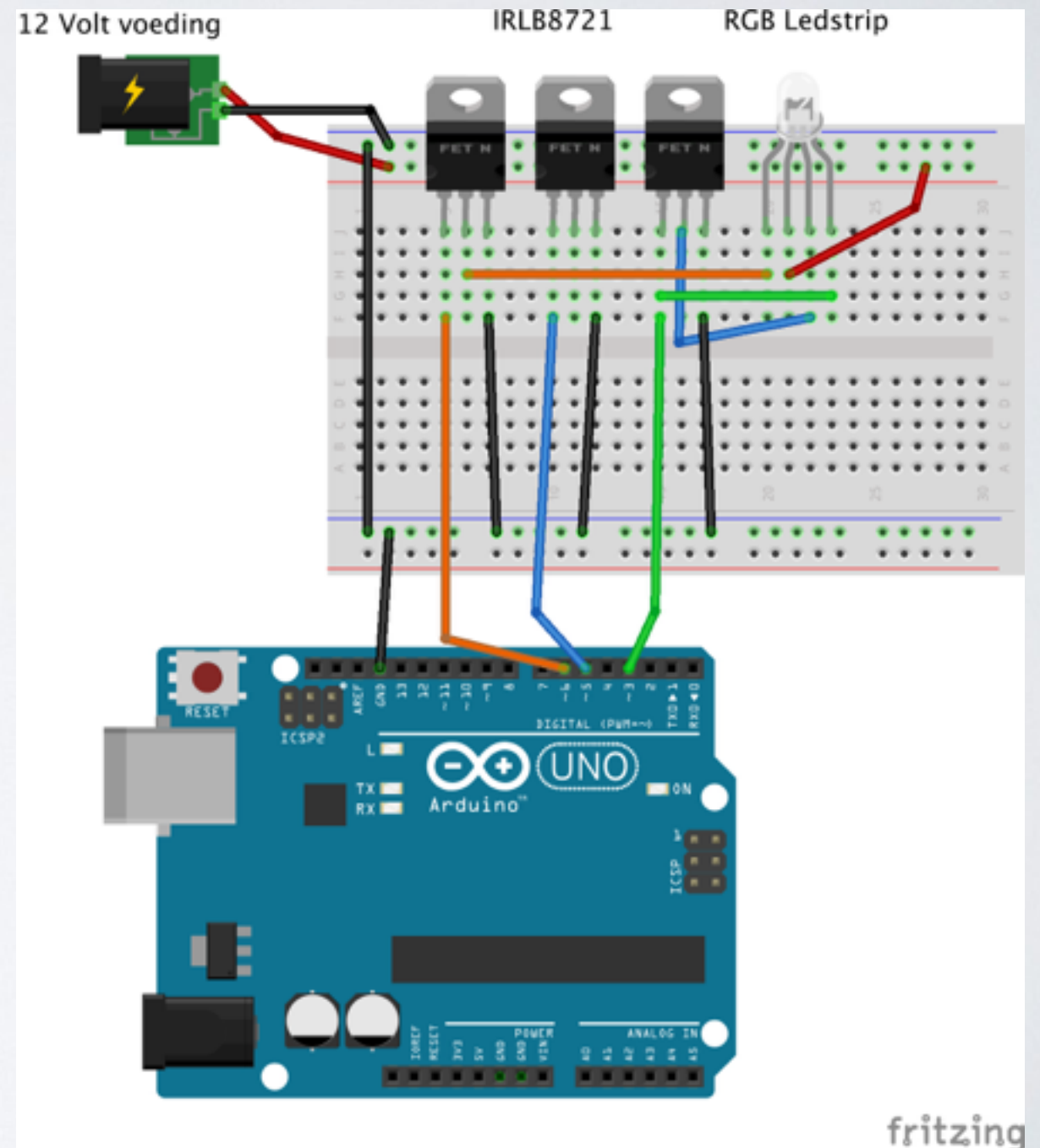


HET SCHAKELEN VAN HOGERE SPANNINGEN

- Je moet een 9V motor aansturen met een Arduino. Hoe doe je dat?
- Je moet een 12V ledstrip aansturen met een Arduino, hoe doe je dat?
- Je wilt een 220V apparaat aan of uitzetten met een Arduino? Hoe doe je dat?
- Schakelingen kunnen aan elkaar gekoppeld worden, mits je de grond van elke schakeling aan elkaar koppelt.

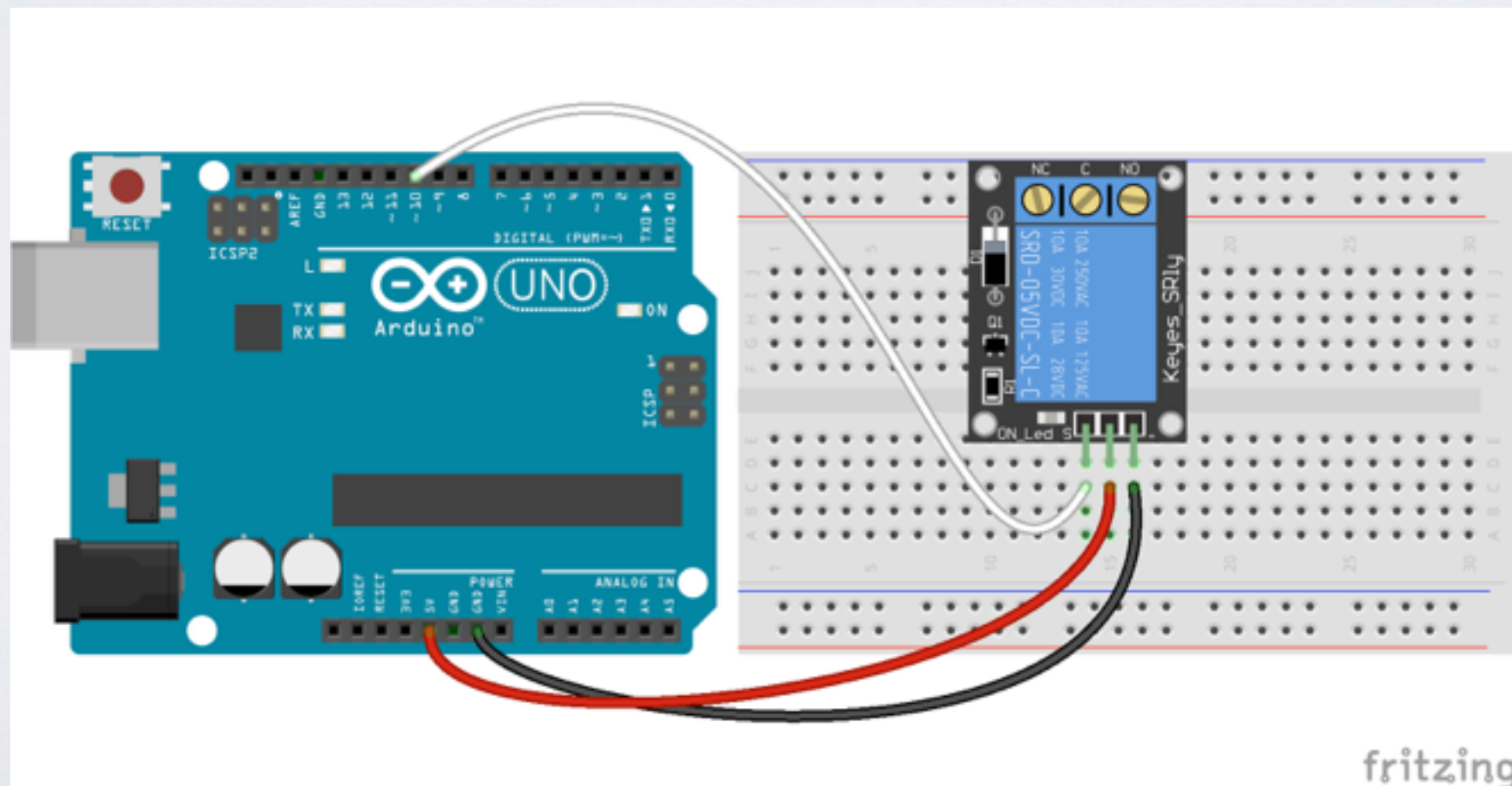
EEN LEDSTRIP AANSTUREN

Een ledstrip heeft relatief veel stroom nodig om te kunnen branden. En de ledstrip draait op 12V. Voor het aansturen van leestrips gebruik je een MOSFET (“moderne” transistor)



220V APPARATEN SCHAKELLEN

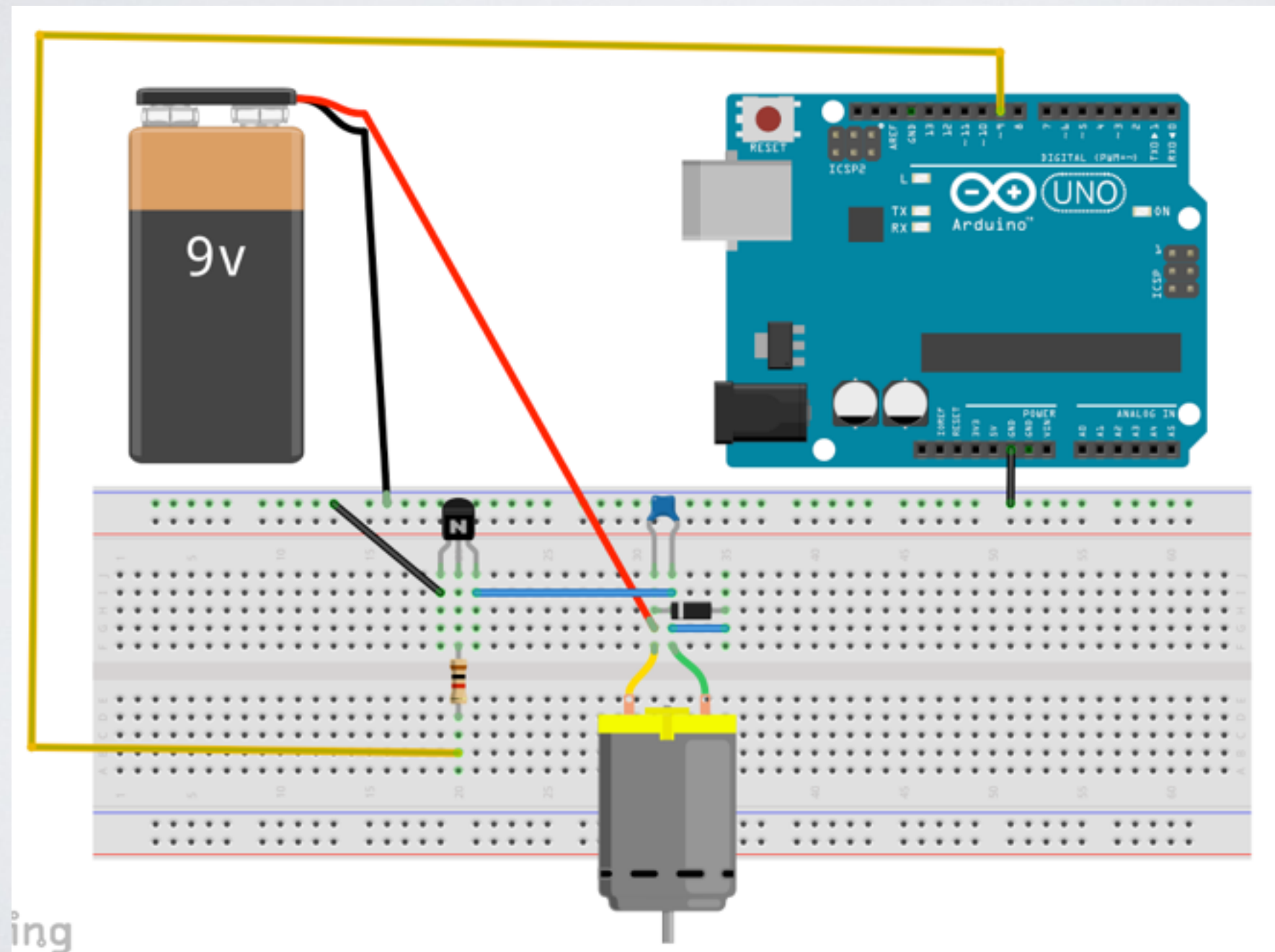
Hoe schakel je een 220V aan of uit, met een microcontroller die op 5V draait? Hiervoor kun je een relais gebruiken (een relais kun je overigens ook gewoon voor lagere spanningen gebruiken). Nadeel van een relais, is dat deze mechanisch is. Mechanische onderdelen gaan sneller stuk.



MOTOREN EN SERVO'S

- Als je met de arduino iets wilt laten bewegen
- Standaard motor (2 draden)
- Stappenmotor
- Servo (180 360 graden)

ELECTRO MOTOR



ing

VOORBEELD CODE (PWM)

```
int motorPin = 3;

void setup() {

    pinMode(motorPin, OUTPUT);

    Serial.begin(9600);

    while (! Serial);

    Serial.println("Speed 0 to 255");

}

void loop() {

    if (Serial.available() ) {

        int speed = Serial.parseInt();

        if (speed >= 0 && speed <= 255) {

            analogWrite(motorPin, speed);

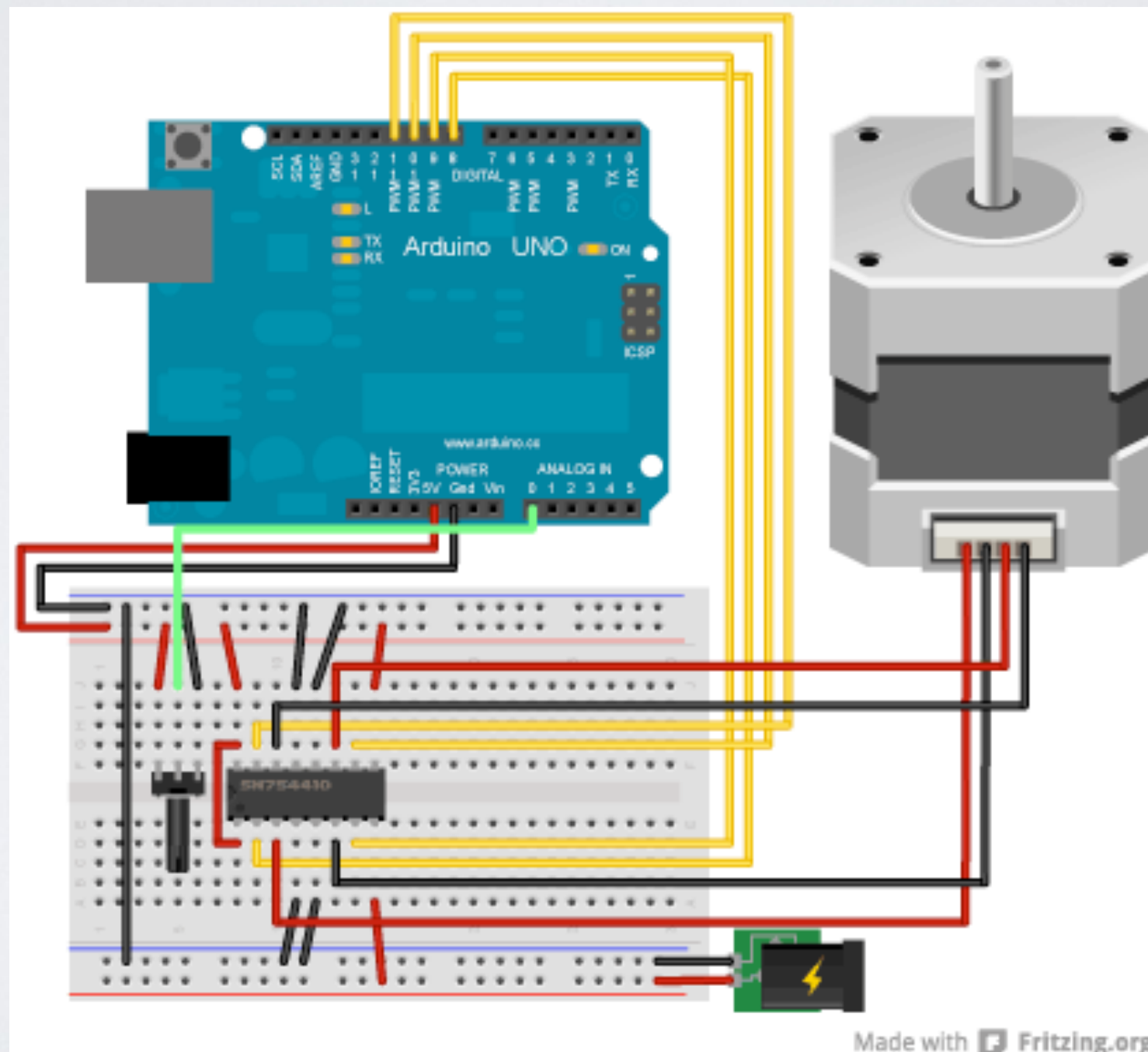
        }

    }

}
```

STAPPENMOTOR

Er zijn ook speciale stappenmotor printjes te koop voor een paar cent



STAPPEN MOTOR CODE

```
#include <Stepper.h>
```

```
const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution for your motor
```

```
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11); // initialize the stepper library on pins 8 through 11:
```

```
int stepCount = 0; // number of steps the motor has taken
```

```
void setup() {
```

```
    ; // nothing to do inside the setup
```

```
}
```

```
void loop() {
```

```
    int sensorReading = analogRead(A0); // read the sensor value:
```

```
    int motorSpeed = map(sensorReading, 0, 1023, 0, 100); // map it to a range from 0 to 100:
```

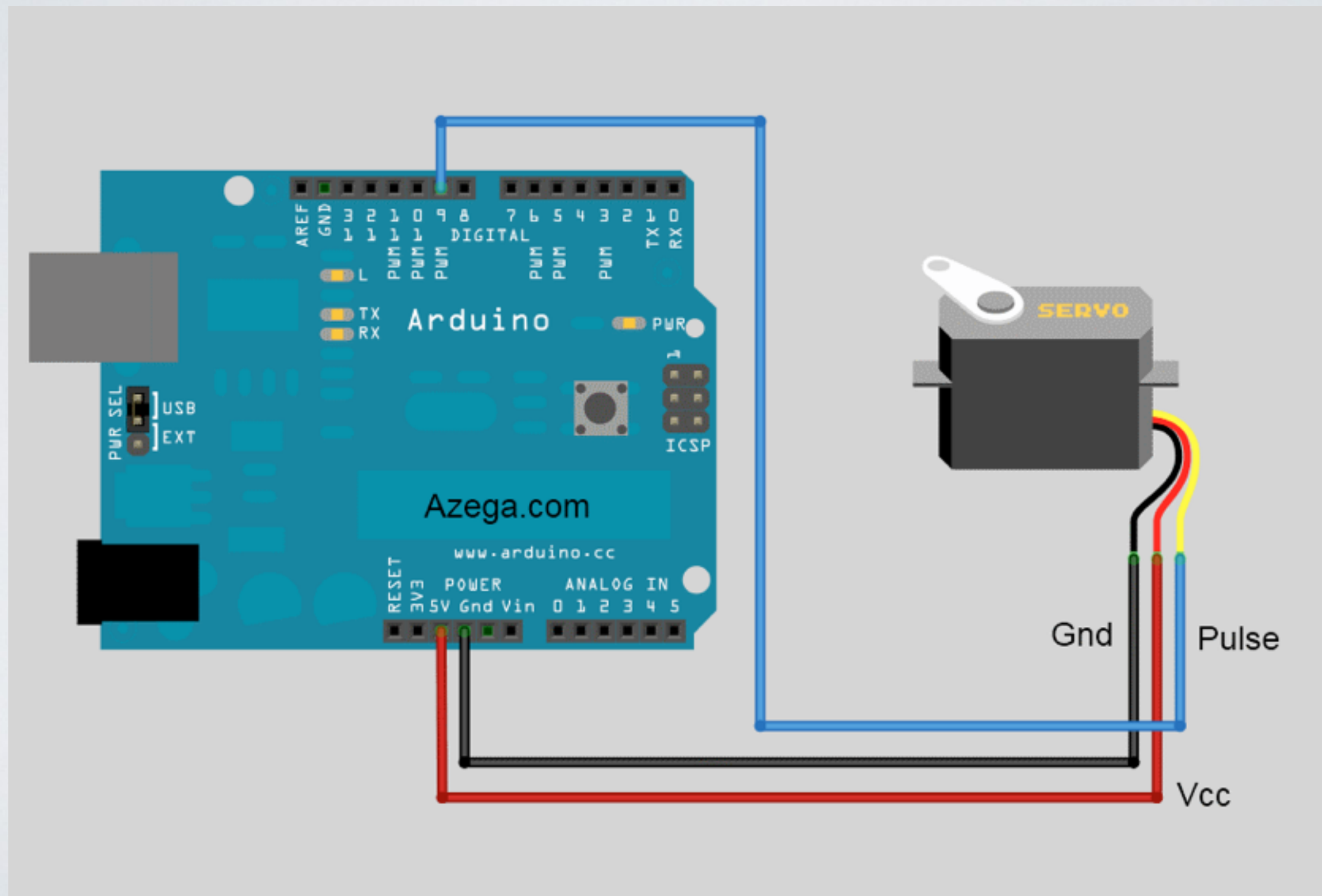
```
    if (motorSpeed > 0) { // set the motor speed:
```

```
        myStepper.setSpeed(motorSpeed); // step 1/100 of a revolution:
```

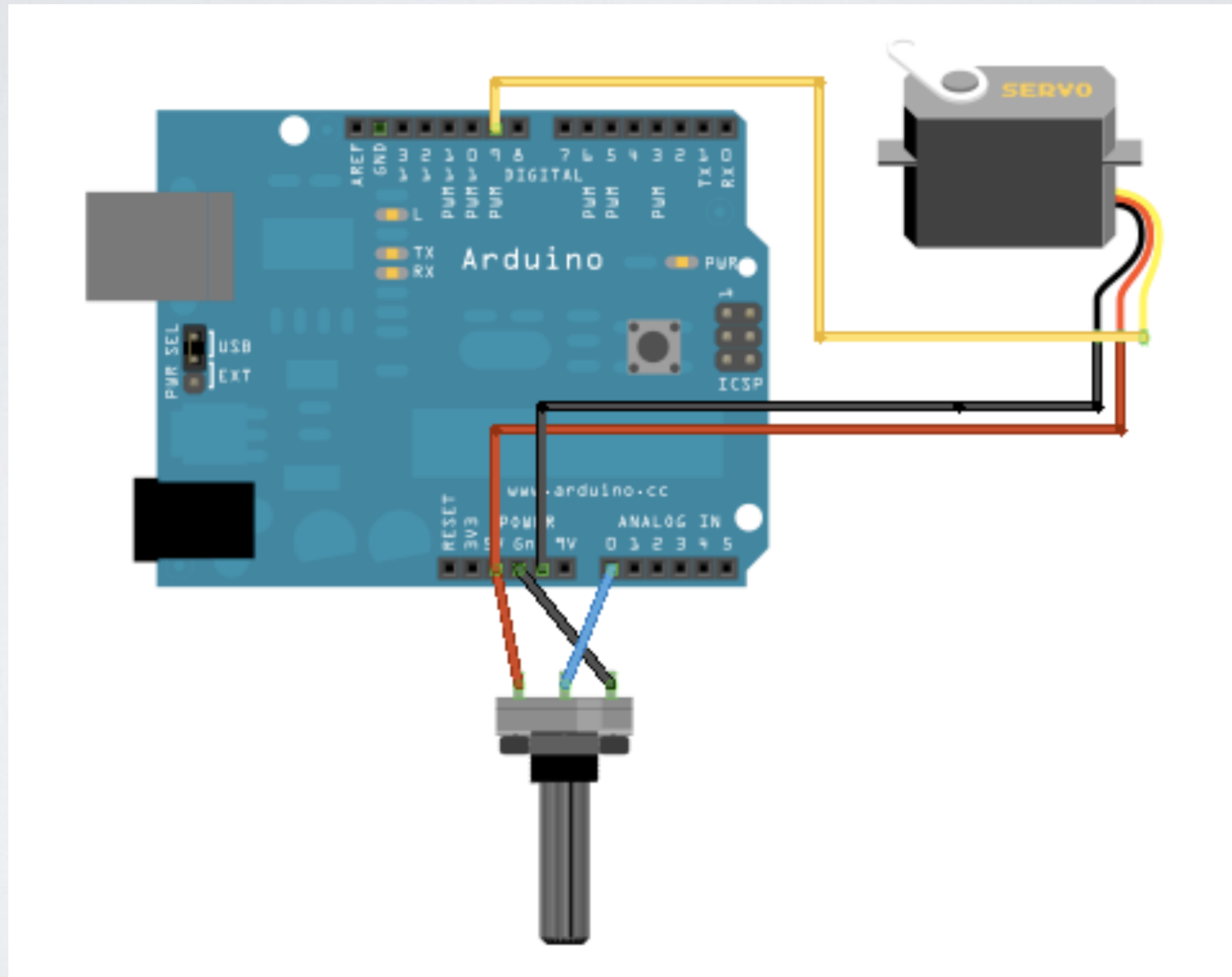
```
        myStepper.step(stepsPerRevolution / 100);
```

```
    }
```

SERVO AANSLUITEN



SERVO VOORBEELD



SERVO VOORBEELD

```
#include <Servo.h>
```

```
Servo myservo; // create servo object to control a servo
```

```
int potpin = 0; // analog pin used to connect the potentiometer
```

```
int val; // variable to read the value from the analog pin
```

```
void setup() {
```

```
    myservo.attach(9); // attaches the servo on pin 9 to the servo object
```

```
}
```

```
void loop() {
```

```
    val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
```

```
    val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
```

```
    myservo.write(val); // sets the servo position according to the scaled value
```

```
    delay(15); // waits for the servo to get there
```

```
}
```


TINKER TIME

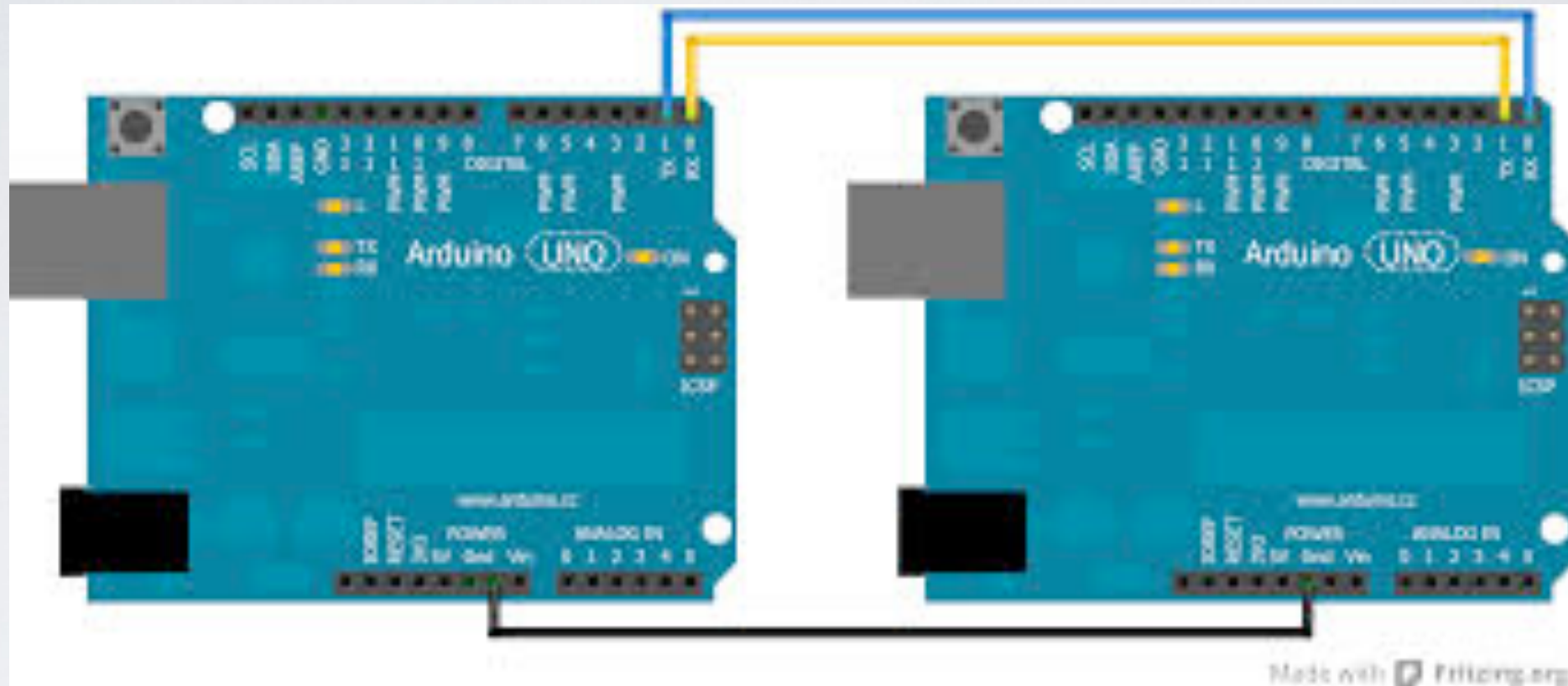
- sluit naast de servo ook een potmeter aan
- als je aan de potmeter draait, dan moet de servo hier op reageren
- onze servo kan maar max 180 graden aan gebruik dus de map functie

ps. sommige servo's kunnen minder dan 180 graden aan als hij kraakt dan moet je een mindere max aantal graden pakken

SOORTEN COMMUNICATIE

- Serieel (pin 0 en 1, Serial library)
- One-wire aansluiting afhankelijk van situatie
- i2c (ook i-kwadraat-c), aansluiten op i2c bus

ARDUINO NAAR ARDUINO



VOORBEELD SERIËLE COMMUNICATIE

```
void setup() {
```

```
    Serial.begin(9600); // start serial port at 9600 bps:
```

```
    while (!Serial) {} ; // wait for serial port to connect.
```

```
    pinMode(2, INPUT);
```

```
    establishContact();
```

```
}
```

```
void establishContact() {
```

```
    while (Serial.available() <= 0) {
```

```
        Serial.print('A'); // send a capital A
```

```
        delay(300);
```

```
    }
```

```
}
```

```
void loop() {
```

```
    // if we get a valid byte, read analog ins:
```

```
    if (Serial.available() > 0) {
```

```
        // get incoming byte:
```

```
        inByte = Serial.read();
```

```
        // read first analog input, divide by 4 to make the range 0-255:
```

```
        firstSensor = analogRead(A0) / 4;
```

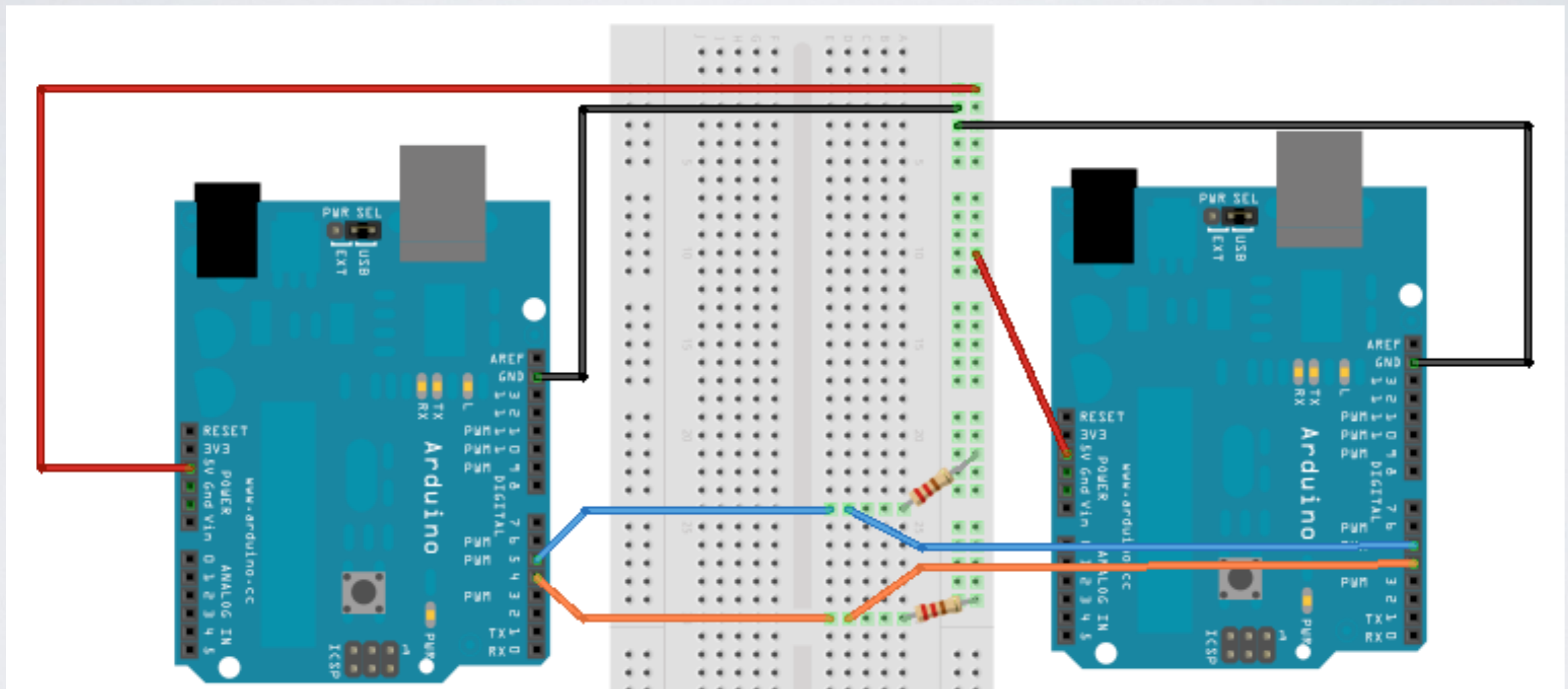
```
        // send sensor values:
```

```
        Serial.write(firstSensor);
```

```
    }
```

```
}
```


I2C COMMUNICATIE



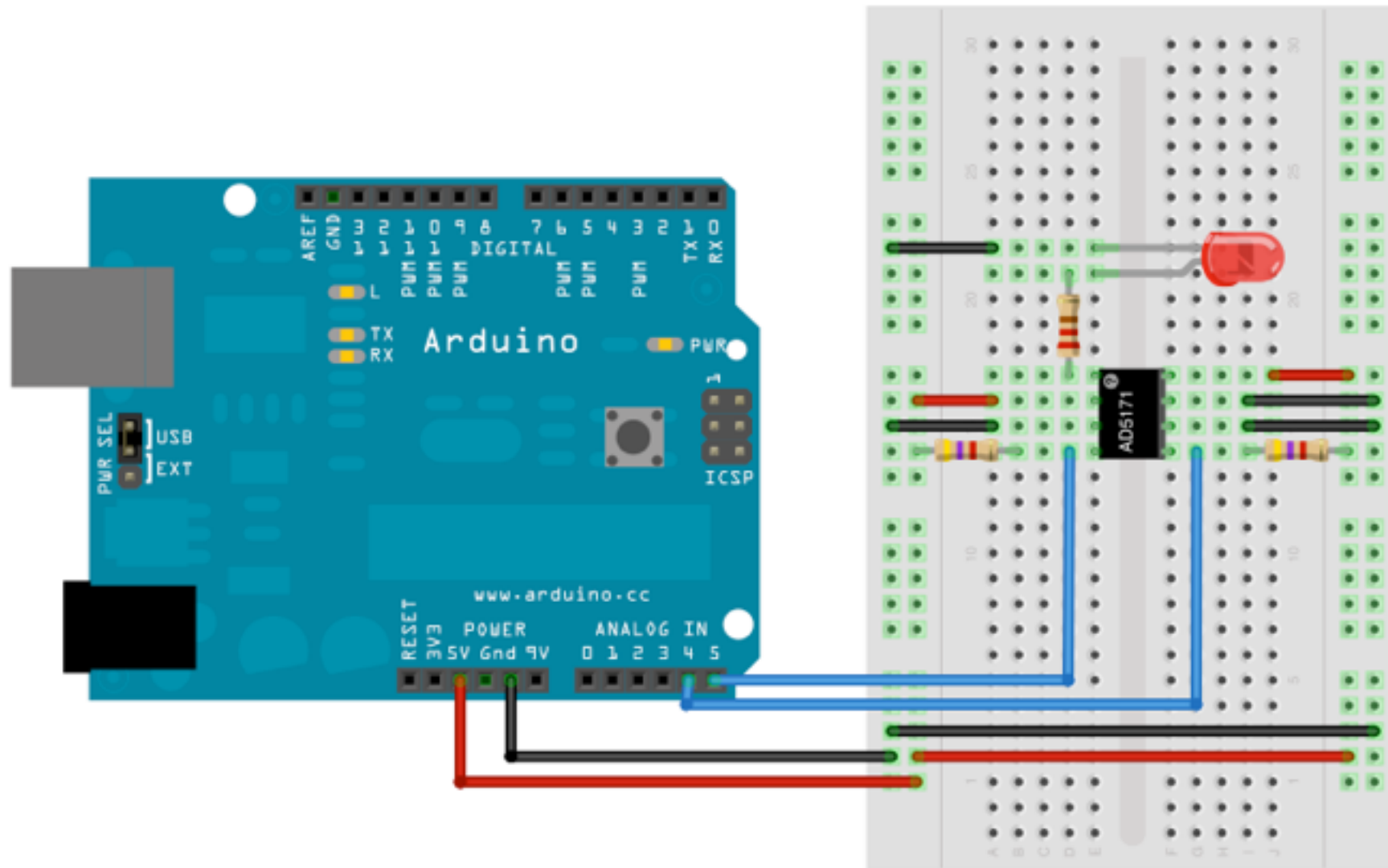
I2C COMMUNICATIE

- i2c is een bus waarop meerdere apparaten aangesloten kunnen worden
- Er is altijd een master met meerder slaves
- Elke slave heeft een uniek adres
- Pull up weerstanden zijn nodig voor stabiliteit
- Standaard i2c bibliotheek is Wire.h

I2C GEBRUIKEN OP DE ARDUINO

- I2C bus heeft twee draden:
 - clock (pin A5)
 - data (pin A4)

I2C VOORBEELD



I2C VOORBEELD CODE

```
#include <Wire.h>

byte val = 0;

void setup() {

    Wire.begin(); // join i2c bus (address optional for master)

}

void loop() {

    Wire.beginTransmission(44); // transmit to device #44 (0x2c) device address is specified in datasheet

    Wire.write(byte(0x00));      // sends instruction byte

    Wire.write(val);           // sends potentiometer value byte

    Wire.endTransmission();    // stop transmitting

    val++;    // increment value

    if (val == 64) { // if reached 64th position (max)

        val = 0; // start over from lowest value

    }

    delay(500);
```

ONE WIRE COMMUNICATIE

- Communicatie met - veelal - sensoren die met slechts een draad aan de Arduino worden verbonden
- Voor de meeste van deze sensoren zijn libraries te vinden

TINKEREN

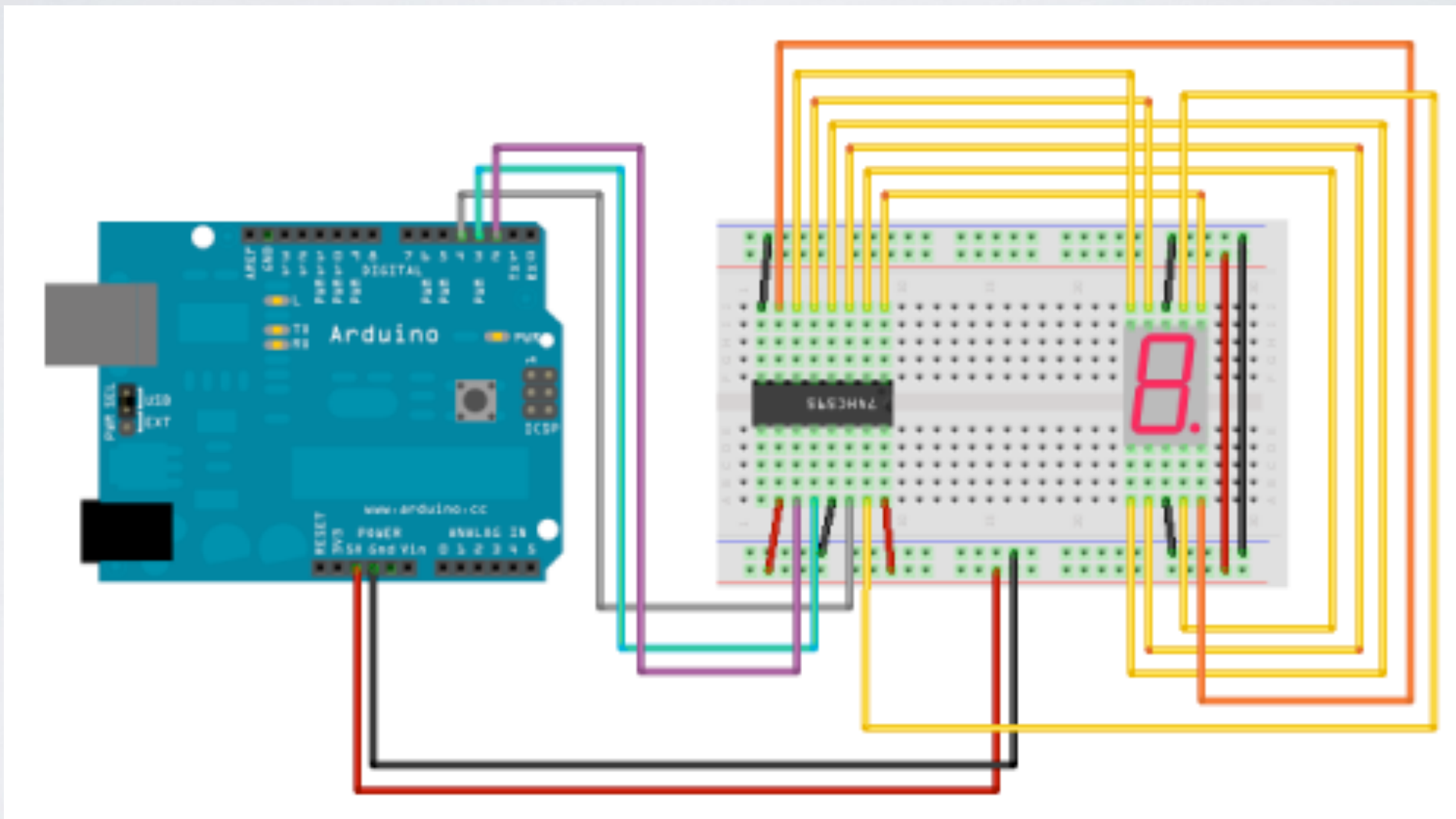
We gaan nu een dobbelsteen maken

1. Electronica:

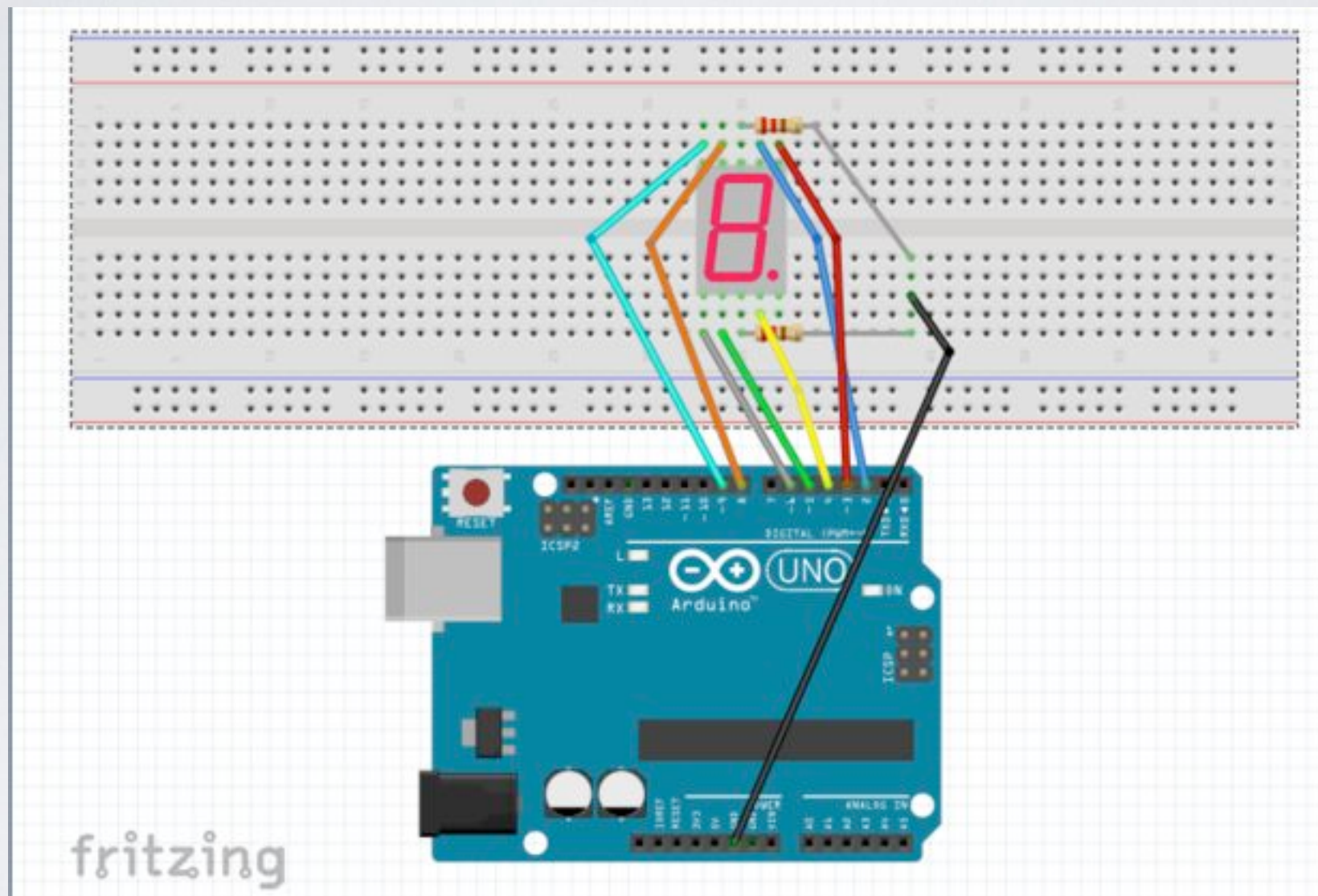
1. 7-segment

2. Tiltschakelaar (voor ontwikkeling is een normale schakelaar handiger)

7 SEGMENT MET SHIFT REGISTER



7 SEGMENT MET 8 PINNEN



AANWIJZINGEN

- Begin eenvoudig eerst de 7 segment aan de praat krijgen
- Begin met het genereren van 1 getal in de setup() van je sketch
- Gebruik random(1, 6) voor een willekeurige waarde
- Sluit een schakelaar aan (zie sheets eerste workshop)
- Schrijf nu de loop():
 - Begin met een donker 7 segment
 - Als er op de schakelaar gedrukt wordt genereer je een willekeurige waarde (1-6)
 - Toon de gegenereerde waarde
 - Vervang de schakelaar door de tilt schakelaar - schudden voor gebruik