



IFI6203 ROBOTIKA

Praktikum 4



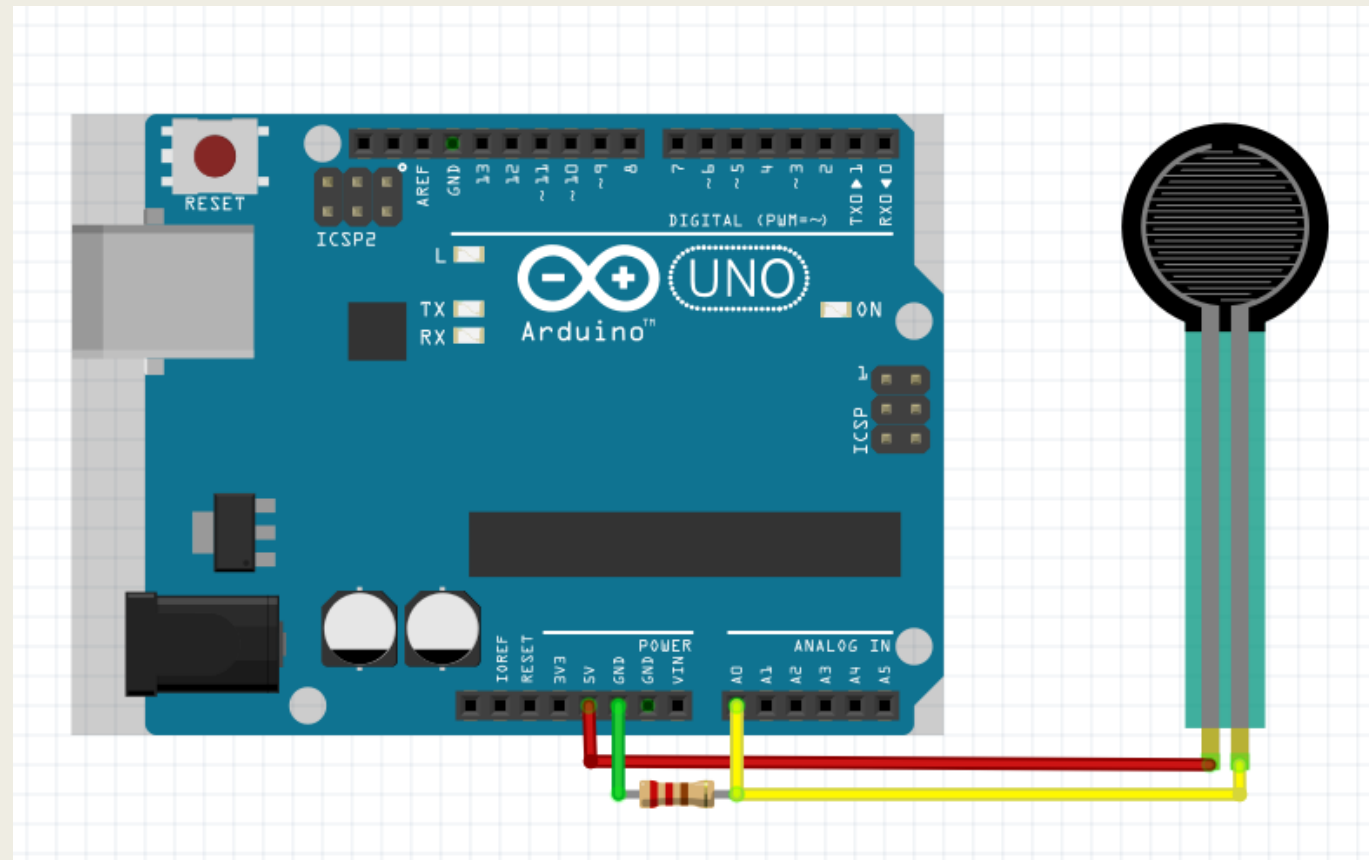
Analoogandurite kasutamine

- Analoogandurid tagastavad oma mõõtetulemuse voolupingena, mida saab lugeda kasutades kontrolleri analoogsisendit, mille küljes olev analoog-digital konverter tagastab tulemuse täisarvuna.
- Saadud väärtused tulevad vastavalt anduri andmelehel olevale informatsioonile teisendada mõõdetavateks ühikuteks.
- Arduino Uno puhul kasutame analoogviikuseid A0 - A5
- Viigu väärtust loeme funktsiooniga *analogRead()*

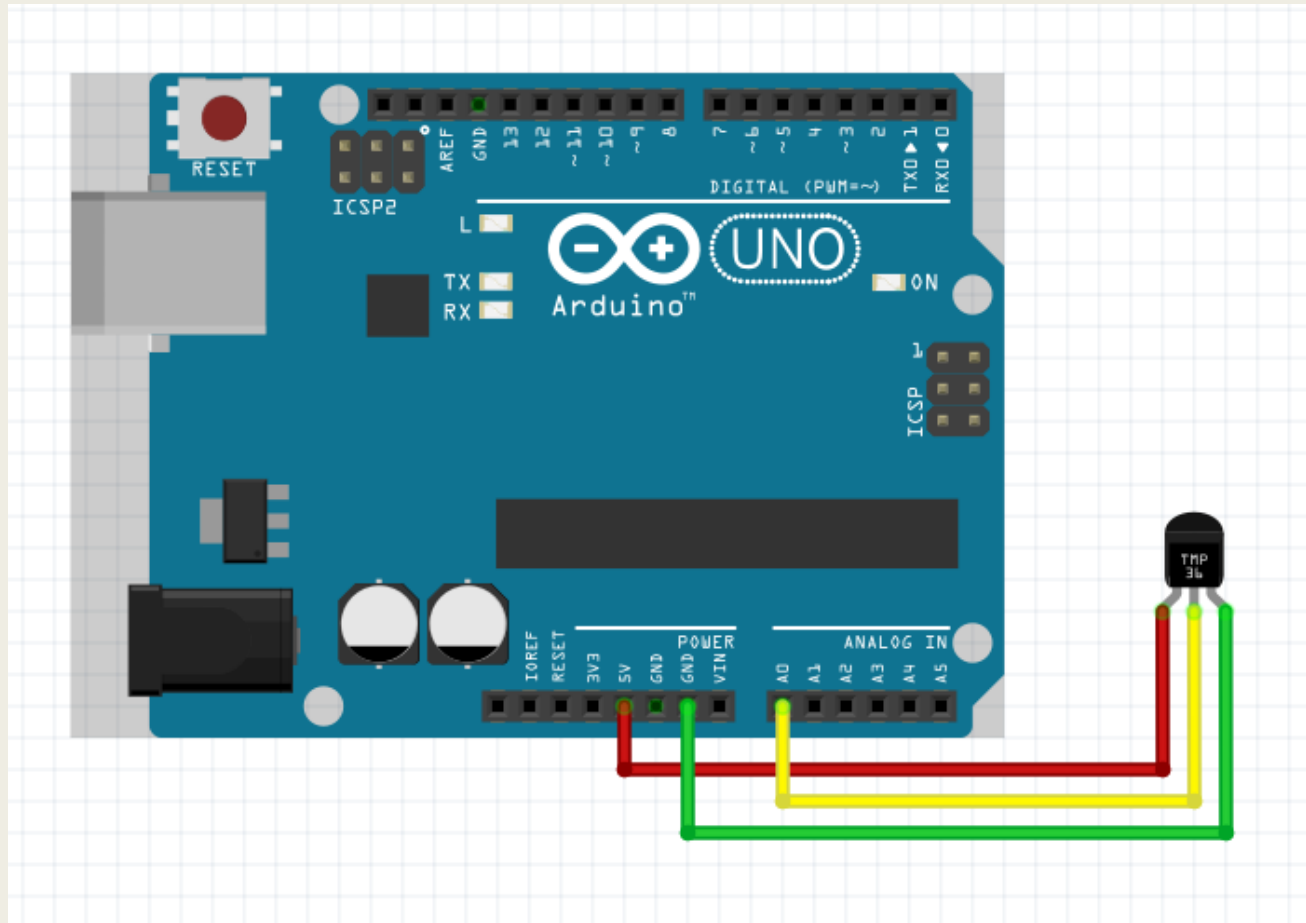
Analooganduri ühendamine

- Olenevalt anduri tüübist ja tootjast on anduril enamasti 2 või 3 ühendusviiku.
- Kui viikused on kaks, siis ühe külge neist läheb toide (olenevalt andurist 3V või 5V) ja teist viiku loeme Arduino analoog sisendiga. Sellise anduri puhul on enamasti vajalik kasutada *pulldown* takistit (takisti mis ühendab analoogviigu maandusega), et tagada korrektne mõõtetulemus.
- Kui viikused on kolm, siis on *pulldown* takisti ehitatud anduri sisse (või vajab andur muidu oma tööks eraldi maandust) ja lisaviik on ühendamiseks kontrolleri maandusega.

Surveandur koos *pulldown* takistiga



Temperatuuriandur tmp36



Levinud analoogandureid

- Surveandur
- Paindeandur
- Temperatuuriandur (tmp36)
- Valgusandur (fototakisti)
- Pinnase niiskusandur
- Vedeliku taseme andur (optiline või mehhaaniline)

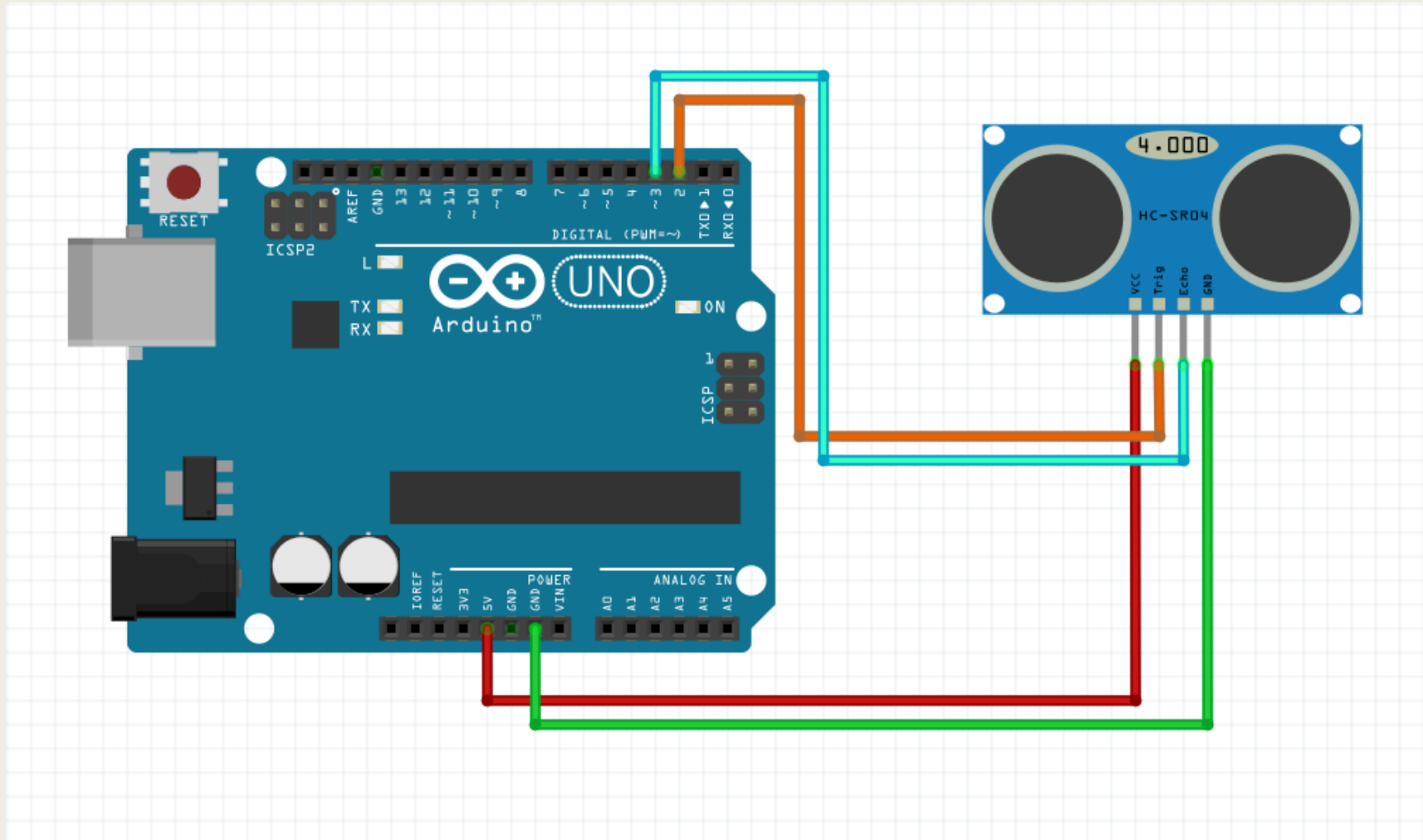
Digitaalandurite kasutamine

- Digitaalandurid tagastavad oma mõõtetulemused digitaalsignaalina, mille tõlgendamiseks tuleb enamasti kasutada andurile kirjutatud teeki.
- Sobiva teegi leiata suure tõenäosusega andurite tootja kodulehelt või tehes veebiotsingu “[andurinimi] Arduino library”

Digitaalanduri ühendamine

- Olenevalt anduri tüübist ja tootjast on anduril enamasti 3 kuni 6 ühendusviiku.
- 2 viiku on toite jaoks
 - *Toitepinge (enamasti 3.3V või 5V)*
 - *Maandus*
- Ülejäänud viigud on andmevahetuseks. Arduino toetab järgnevaid standardseid ühendusliideseid
 - *SPI (Serial Peripheral Interface) - 4 viiku andmevahetuseks*
 - *I2C (Two-Wire Serial Interface) - 2 viiku andmevahetuseks*
 - *Serial (Serial Interface) - 2 viiku andmevahetuseks*
- Mõnel digitaalanduril võib olla oma mittestandardne ühendusliides - loe anduri andmelehte

HC-SR04 ultrahelisensor kauguse mõõtmiseks



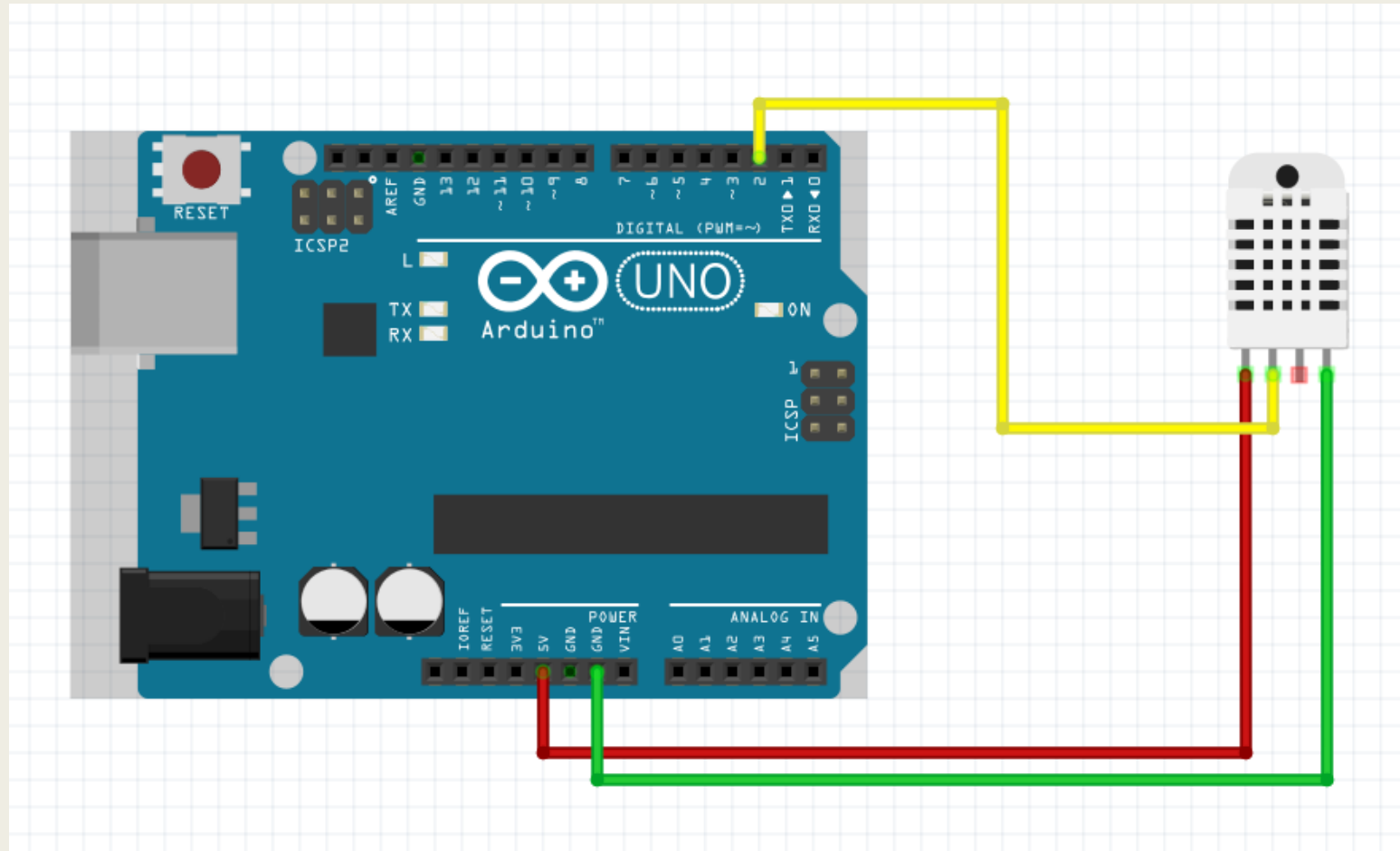
HC-SR04 näitekood (eraldi teeki pole vaja)

```
int trigPin = 2;
int echoPin = 3;
long duration;
int distance;

void setup() {
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
    Serial.begin(9600); // Starts the serial communication
}

void loop() {
    digitalWrite(trigPin, LOW); // Clears the trigPin
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); // Sets the trigPin on HIGH state for 10 micro seconds
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in ms
    distance= duration*0.034/2; // Calculating the distance
    Serial.print("Distance: "); // Prints the distance on the Serial Monitor
    Serial.println(distance);
}
```

DHT22 temperatuuri ja niiskussensor



DHT22 näitekood (vajab teeki SimpleDHT)

```
#include <SimpleDHT.h>
int pinDHT22 = 2;
SimpleDHT22 dht22(pinDHT22);

void setup() {
  Serial.begin(115200);
}

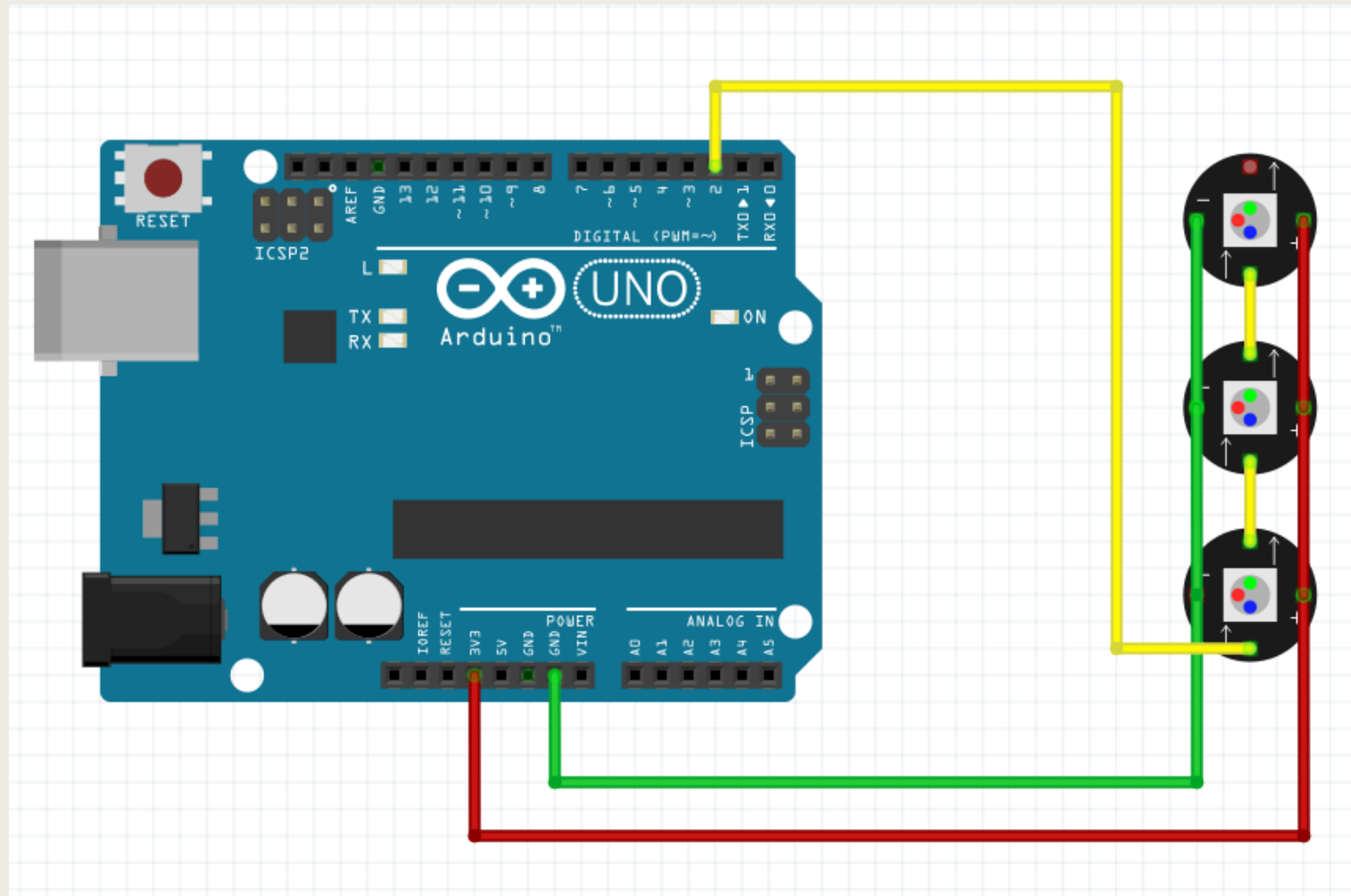
void loop() {
  float temperature = 0;
  float humidity = 0;
  int err = SimpleDHTErrSuccess;
  if ((err = dht22.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    Serial.print("Read DHT22 failed, err="); Serial.println(err);delay(2000);
    return;
  }

  Serial.print("sample OK: ");
  Serial.print((float)temperature); Serial.print(" *C, ");
  Serial.print((float)humidity); Serial.println(" RH%");
  delay(2500); // DHT22 sampling rate is 0.5HZ.
}
```

NeoPixel LED moodul

- NeoPixel on RGB LED moodul
- Ühe mooduli ühendamiseks on tarvis toidet 3.3V, maandust ja juhtsignaali
- Mooduleid võib jadamisi ühendada kuni 128 tükki
- Mooduli juhtimiseks on vaja Arduino IDE-s paigaldada teek “Adafruit NeoPixel” ning kasutada selle poolt pakutavaid funktsioone.

Arduino kolme NeoPixel mooduliga



NeoPixel teegi olulised funktsioonid (1)

- NeoPixel objekti nimega `np_led` defineerimine:

Adafruit_NeoPixel np_led(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

- *NUMPIXELS = järjestikku ühendatud NeoPixel moodulite arv*
- *PIN = Arduino viik, mille külge on ühendatud signaalikaabel.*
- *NEO_GRB + NEO_KHZ800 = see on mooduli tüüp*

- NeoPixel objekti initsialiseerimine: **`np_led.begin();`**

- NeoPixel objekti kõigi moodulite väljalülitamine: **`np_led.clear();`**

NeoPixel teegi olulised funktsioonid (2)

- NeoPixel mooduli värvi määramine:

`np_led.setPixelColor(i, np_led.Color(R, G, B));`

- *i = NeoPixel mooduli järjekorranumber - esimene moodul on nr. 0*
- *R = punase värvi väärtus RGB koodis (0-254)*
- *G = rohelse värvi väärtus RGB koodis (0-254)*
- *B = sinisi värvi väärtus RGB koodis (0-254)*

- Kõigi NeoPixel objekti moodulite oleku muutmise vastavalt eelnevalt määratud värvile: `np_led.show();`