

C++ Fokus på sproget #08

in [Arduino Projekter](#), [C++ Fokus](#) with 0 Comments

Class & object i praksis – PROJEKTET

Nu er det tid til at se på et konkret eksempel. Jeg ønsker at måle luftfugtighed og temperatur med en DHT11 sensor.

- Værdierne skal udlæses ved hjælp af to lysdioder.
- Der skal være mulighed for at se om der er en stigende eller faldende tendens.
- Denne mulighed skal kunne vælges til/fra med en kontakt.
- Der skal være en kontakt, som sætter systemet i standby.

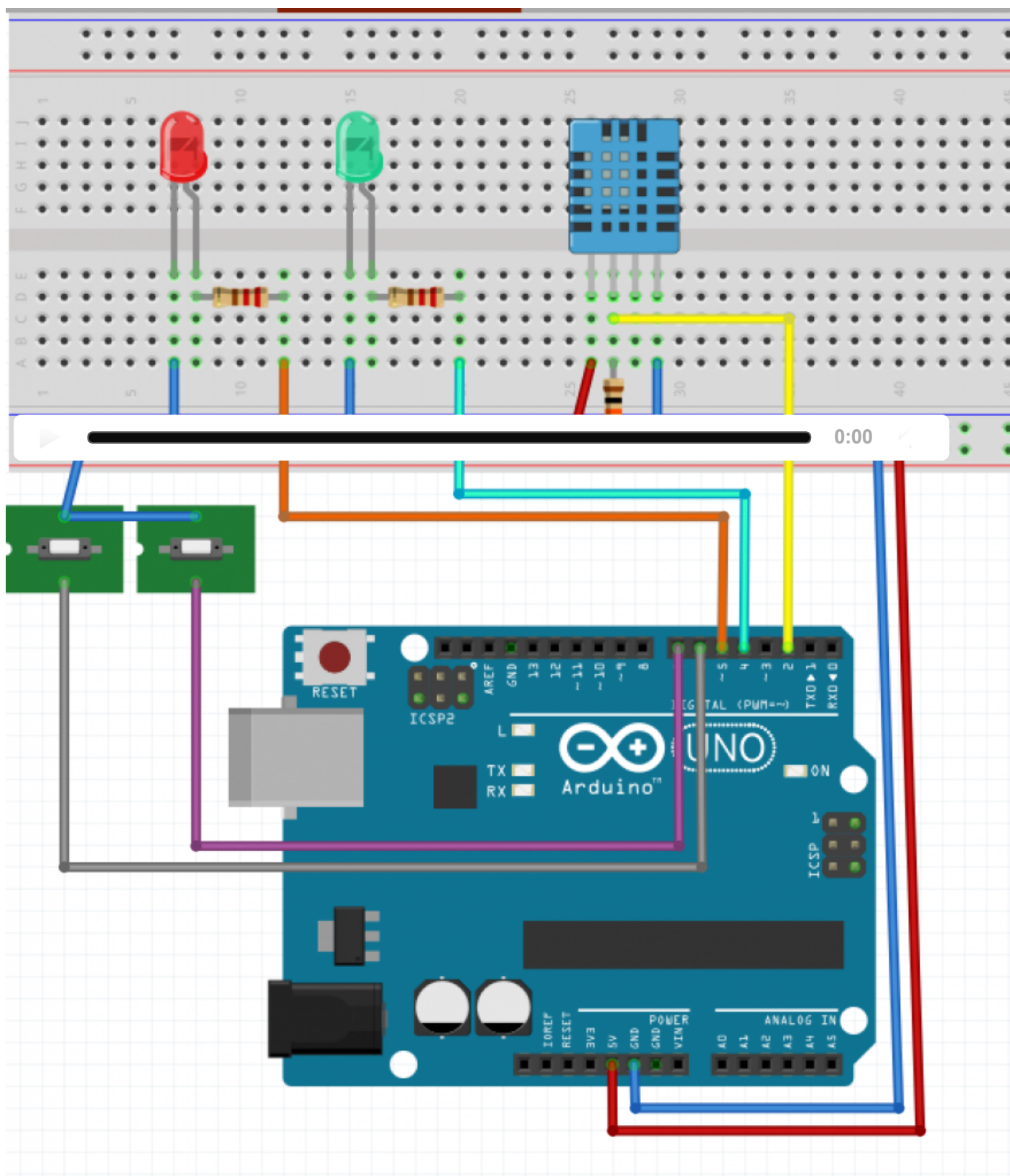
Det ville selvfølgelig være meget lettere at aflæse et display i stedet for at holde styr på at et blink betyder EN, to blink betyder TO osv. Men dels er det her jo blot en simpel demo og dels giver det mig anledning til at introducere en virkelig nyttig lille funktion ved navn modulus. Mere om den senere

Fik jeg nævnt, at den røde LED er tiere og den grønne er enere?

De to dioder blinker hurtigt for at markere, at der kommer et resultat af en temperaturmåling. De to dioder blinker lidt langsommere for at markere, at der kommer et resultat af en måling af luftfugtigheden. Og så er det bare med at tælle. Hvis der er valgt visning af tendens, vil begge dioder lyse i to sekunder, hvis den nye værdi er som den foregående. Lyser kun den røde, er tendensen stigende. Lyser kun den grønne, er tendensen faldende.

I videoen herunder indledes med 30% luftfugtighed (tendens: uændret) og fortsættes med en temperatur på 20° (tendens: uændret). Til sidst er temperaturen 19° (tendens: faldende)

Kredsløbet er bygget således op:



Bemærk at de to kontakter går direkte til GND. Når det kan lade sig gøre uden at kortslutte noget skyldes det, at man kan aktivere de indbyggede pullup-modstande ved at initialisere portene sådan her:

```
pinMode(ACTIVATE_TREND_PIN, INPUT_PULLUP);  
pinMode(RUN_PIN, INPUT_PULLUP);
```

Da jeg lavede projektet, begyndte jeg imidlertid IKKE med at bygge det op på mit breadboard. Jeg skrev koden først, så lad os se lidt på den. Såvel i en version skrevet med funktioner som i en version, hvor sketchen er bygget op omkring en klasse. I begge versioner skal man bruge et library, dht, som er skrevet af Rob Tillaart. Der er info på dette link: arduino.cc/playground/Main/DHTLib.

Du kan også downloade det her: kortlink.dk/kwad

For fleksibilitetens skyld har jeg gjort det muligt at vælge hvilke pins, der skal anvendes. Derfor begynder koden sådan her:

```
#include <dht.h>
const byte DHT11_PIN = 2;
const byte ONES_PIN = 4;
const byte TENTHS_PIN = 5 ;
const byte ACTIVATE_TREND_PIN = 6;
const byte RUN_PIN = 7;
```

Så skal der være nogle globale variable så vi kan få adgang til de foregående målinger og for at vi kan se, om der skal vises tendens eller om systemet er på standby:

```
// Global variables
boolean IsRunning = true;
boolean ShowTrend = true;
byte StoredTemperature = 0, StoredHumidity = 0;
```

At aflæse en DHT11 og få en lysdiode til at blinke er trivielt, så det går jeg ikke i detaljer med her, men lad os kigge lidt på, hvorledes man adskiller et tal i tiere og enere, F. eks. tallet 35. For at finde tierne skal man vide, hvor mange gange 10 går op i 35. Det kaldes heltalsdivision og hvis man i C++ dividerer to integers med hinanden, er det netop hvad der foregår. 35/10 giver med andre ord 3 og ikke 3,5. Decimalen bliver smidt væk. Enerne finder man ved at dividere med 10 og beholde det, som bliver til rest. Det finder man med funktionen 'modulo'. 12 modulo 10 = 2. 97 modulo 10 = 7. I C++ skrives modulo som % (og det har altså INTET med procent at gøre!).

Funktionen kommer derfor til at se således ud: