

C++ Fokus på sproget #10

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

Klasser og objekter i praksis – THE LIBRARY

Har man nu fået skrevet kode til en rigtig effektiv og meget anvendelig klasse, kan den bruges igen og igen som blueprint for nye objekter eller som stamfader til nye klasser. Man skal blot lave lidt copy/paste får at få klassen flyttet over i en ny sketch. Sommetider har man skrevet noget, som kan være til stor hjælp for andres arbejde med Arduino og hvis man går ind for Open Source og tanken om at man også selv må bidrage til fællesskabet, kan man lagre sin Class som et Library. Så kan andre (og man selv 😊) på en meget let måde implementere alle de fine tanker i en sketch. Der er nogle få formkrav, som tidligere nævnt.

Det er god stil at navngive Class (og dermed Library) med ens egne forbogstaver for at forebygge at der forekommer libraries med samme navn. I dette tilfælde: jsControlDHT.

På den ene side må man sende en venlig tanke til de mennesker, som giver os lov til at benytte deres libraries og dermed dele resultatet af deres anstrengelser. På den anden side er det enormt irriterende, at der er så meget sjusk med navngivningen af libraries. Det sker alt for tit at forskellige libraries har samme navn og dermed giver anledning til en del bøvl.

Tilbage til konstruktionen af vores library!

Nu skal der laves

- en folder med navnet jsControlDHT
- en tekstfil med navnet jsControlDHT.h – Den skal ligge i folderen og rummer definitionen på vores Class.
- en tekstfil med navnet jsControlDHT.cpp – Den skal ligge i folderen og rummer implementationen af vores Class.
- en folder i folderen. Den skal hedde examples.
- en sketch som viser, hvorledes man bruger Library. Denne sketch gemmes i examples.

Header-filen (jsControlDHT.h) kommer til at se således ud:

```

1 ▢ /*
2  * jsControlDHT - a sample library for OZ6HR
3  * Version 1.01 - 2016 04 04
4  *
5  * File: jsControlDHT.cpp
6  * Author: Jørgen Sand
7  * URL: www.oz1js.net
8  *
9  */
10
11 #ifndef jsControlDHT_h
12 #define jsControlDHT_h
13
14 #include <arduino.h>
15 #include <dht.h>
16
17 ▢ class jsControlDHT {
18     public:
19         jsControlDHT(byte DHT11Pin, byte OnesPin,
20                     byte TenthsPin, byte ActivateTrendPin,
21                     byte RunPin);
22         void run(dht& MyDht);
23
24     private:
25         boolean _IsRunning = true, _ShowTrend = true;
26         byte _DHT11Pin = 0, _OnesPin = 0, _TenthsPin = 0;
27         byte _ActivateTrendPin = 0, _RunPin = 0;
28         byte _StoredTemperature = 0, _StoredHumidity = 0;
29
30         void makeAlert(unsigned int mseconds);
31         void doShowTrend(byte CurrentValue, byte &OldValue);
32         void showDhtValues(byte Value);
33         void processDhtValues(byte Temperature, byte Humidity);
34 };
35
36 #endif /* jsControlDHT */
37

```

- Linje 11 og 12 er med for at forhindre, at Library bliver loadet flere gange.
- Linje 14 er med fordi vi har brug for nogle Arduino metoder, eksempelvis digitalWrite, digitalRead.
- Linje 15 er med fordi vi skal kunne aflæse vores DHT11

Filen jsControlDHT.cpp indeholder implementeringen af Class. Der skal blot tilføjes lidt i starten af filen: