

Experience avec le capteurs de température et d'humidité DHT22

Ce document n'explique pas tous les détails pour interfacer le DHT22, décrits sous www.didel.com/diduino/DHT22.pdf.

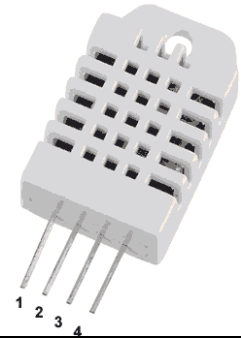
C'est un exemple simple d'utilisation de la librairie DHT22.h pour afficher avec OledPix.h

Les programmes sont sous www.didel.com/OledDHT22.zip

Le DHT22 est un capteur facile à se procurer à prix chinois. Il communique en série 1 fil avec un protocole spécial, et fournit une indication de température sur 16 bits et une indication d'humidité sur 16 bits également. On peut redemander une mesure toute les secondes environ avec une fonction bloquante de 5ms, non interruptible.

Pour mesurer une température, il existe des capteurs plus simple (TMP36, LM35)

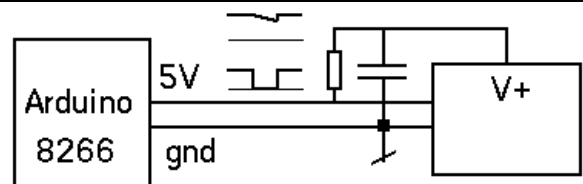
DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



Câblage et caractéristiques électriques

Le DHT22/Am2302 accepte une tension de 3.3 à 5.5V. Le courant est de 15 uA stand-by et 500 uA pendant la mesure (de 1 secondes?). La sortie peut absorber 8mA, une pull-up de 5k (donc un courant de 1mA) est recommandé. La pullup interne du microcontrôleur est suffisante pour des courtes distances.

Il doit être possible d'alimenter le circuit avec plusieurs mètres de câble à 2-fils, le signal étant 99% du temps à l'état un. 1kOhm et 500 uA donnent une tension de 4.5V sur le DHT22, avec une légère ondulation pendant les transferts série.



Librairie DHT22.h

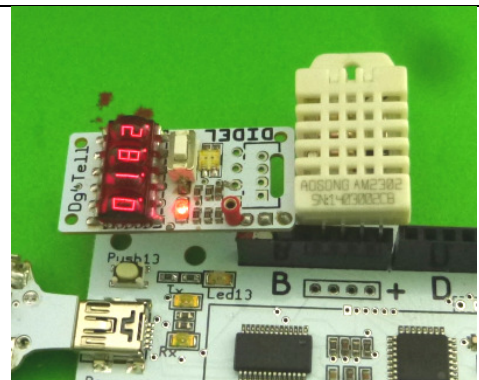
Cette librairie définit les variables globales

int humid, int temp (16 bits, 9-10 bits utilisés), int sign (vu comme un boolean)

L'appel MeasurementDHT22(); doit être fait au max toutes les secondes. et mets à jour ces 3 variables.

Pour nos tests, le DHT22 est inséré sur les pins Arduino 8 à 11. Le Tell sur la pin 13, le Oled sur les pins 4 à 7, comme dans www.didel.com/Oled1306.pdf

La définition du câblage est du set-up est faite dans DHT22.h



Programme type

On veut afficher sur le terminal Arduino et sur Tell

```
//TestDHT22Terminal
#define delMs delay
#include "DHT22.h" // declare temp, humid, sign
#include "Tell.h"

void setup() {
```

```

    SetupDHT22();
    SetupTell();
    Serial.begin(9600);
    Serial.println ("ok");
}

void loop() {
    Serial.println (".");
    Measure();
    Tell(temp);
    if () { S1On; Serial.println ("err");delMs(1000);}
    else { //error
//      S1On;
        Serial.print ("temp ");
        if (sign){ Serial.print ("-"); }
        Serial.print (temp/10); // converti le bin en décimal
        Serial.print (","); Serial.println (temp%10);
        Serial.print ("humid ");
        Serial.print (humid/10); // converti le bin en décimal
        Serial.print (","); Serial.println (humid%10);
        delMs (1000);
    }
}

```

Remarques

La librairie Y utilise usuellement delMs() programmé en C.
L'affichage sur Tell ne gêne pas s'il n'est pas câblé.

SetupDHT22();

Exemple Arduino sur pin 8

```

#define DthLow    digitalWrite (8,0); pinMode (8,1)
#define DthFlot  pinMode (8,0)
#define DthLevel  digitalRead(8)
void SetupDHT22() { // initialisation pin 8
    pinMode (8,1);
    delMs (1000); // power-up delay
}

```

Exemple Arduino/C et Oled sur 4 pins consécutives

Le circuit est inséré directement sur les pins Arduino 8 à 11, avec la pin8 (RB0) programmée à 0 (Gnd) et la pin 11 (RB3) à 1 (5V). On peut utiliser des sorties car le circuit consomme peu
L'entrée sur pin 10 utilise la pull-up interne.

```

// TestOledDHT22 170601
#define bDth 2 // PORTB
#define DthLow  bitClear(PORTB,bDth); bitSet (DDRB,bDth)
#define DthFlot bitSet (PORTB,bDth); bitClear (DDRB,bDth)
#define DthLevel (bitTest (PINB,bDth))
void SetupDHT22() {
    DDRB = 0b11111001 ;
    PORTB= 0b11111100 ; //V+ Pull-up nc Gnd
    delMs (1000); // power-up delay
}

byte tempH, tempL, humiH, humiL, check;
int sign,temp,humid;
byte cyLow, cyHigh, error;
    // on revient avec error = 0 1 2

byte GetByte() { // lit 8 bits
    for (byte i=0; i<8; i++) {
        dd <<=1;
        cyLow=0; cyHigh=0;
        while (DthLevel==0) { cyLow++; }
        while (DthLevel) { cyHigh++; }
        if (cyHigh > cyLow) {
            bitSet (dd,0);
        }
    }
}
return dd;

```

```

}

#define TimOut 100 //
byte Measure() { // imp et lit quittance
    DthLow; delMs(1); DthFlot;
    // réponse dans les 100 us ou erreur 1
    cyLow=0;
    while (DthLevel) {
        if (cyLow++ > TimOut) { error=1; return 0;}
    }
    // la ligne est à 0 on lit tout
    cli(); // no interrupt
    tempH=GetByte(); tempL=GetByte();
    //sign=tempH&0x80;
    temp=(tempH&0x7F)+tempL;
    humiH=GetByte(); humiL=GetByte();
    humid= humiH+humiL;
    check=GetByte();
    sei();
    // if (((tempH+tempL+humiH+humiL)&0xFF)-check)==0) { error=0;}
    // else { error=2;}
    error=0;
}

```

jdn 161122