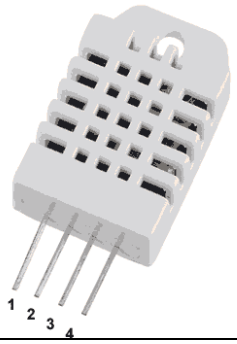
CH-1092 Belmont, Switzerland
Tel +41 21 728-6156, Fax 728-6157
Email info@didel.com, www.didel.com

Capteurs de température et d'humidité DHT22

Cette documentation affiche les résultats sur Tell 16bits. La documentation plus récente www.didel.com/OledDHT22.pdf ajoute l'affichage sur Oled.

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.
On trouve beaucoup d'exemples d'utilisation Arduino sur le Web. La doc du fabriquant "traduite google* est sous http://akizukidenshi.com/download/ds/aosong/AM2302.pdf

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND

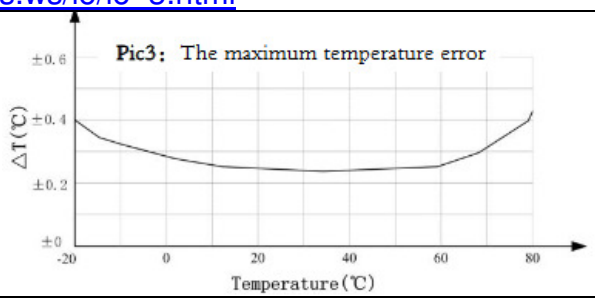


Si vous ne comprenez pas ce qui suit, une description avec le vocabulaire Arduino traditionnel est sous https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/

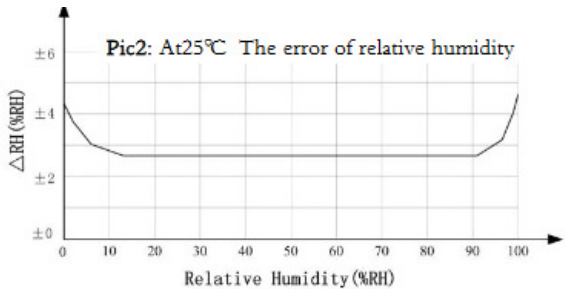
La technologie

Les détails interne du circuit ne sont pas connus. On trouve une information générale intéressante sous http://www.electronics-tutorials.ws/io/io_3.html

Les détails interne du circuit ne sont pas connus. On trouve une information générale sur les capteurs de température sous http://www.electronics-tutorials.ws/io/io_3.html
Le DHT22 mesure la température **Celsius** en dixièmes de degrés.



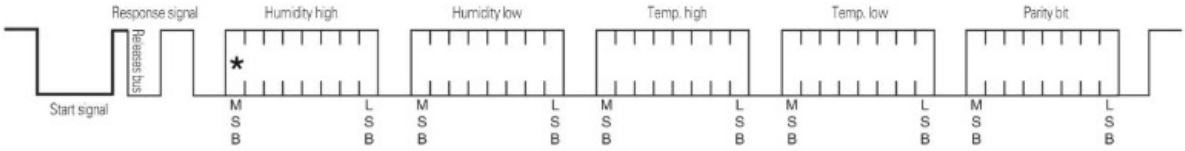
L'**humidité relative** de l'air, ou **degré hygrométrique**, correspond au rapport de la pression partielle de la vapeur d'eau contenue dans l'air sur la pression de vapeur saturante (ou tension de vapeur) à la même température. Elle est donc une mesure du rapport entre le contenu en vapeur d'eau de l'air et sa capacité maximale à en contenir dans ces conditions. (Wikipedia).
Le zone de confort est entre 30 et 50 %RH
Le DHT22 mesure en dixièmes de %RH



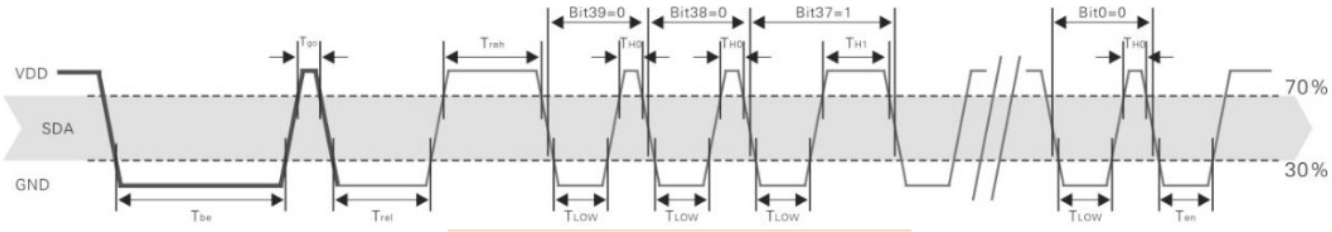
Sérialisation des mesures

Le DHT22 n'utilise pas le protocole 1-wire de Dallas. Le transfert de la mesure sur un fil est déclenché par une impulsion du maitre qui force un zero (<0.5V) pendant 70ms. Le maître se met en entrée et si le capteur est prêt, il envoie son flot de données. S'il n'a pas réagi dans les 50 microsecondes, c'est qu'il n'est pas prêt et il faudra redemander. Les mesures doivent être séparées de plus de 1 secondes. Si on attend 10 secondes, on obtient la mesure fait 9 secondes auparavant, donc on fait une double mesure.

Le capteur envoie 5 mots de 8 bits, le 5^{e} mot étant la somme des 4 premiers.



Chaque bit est envoyé sous forme d'une impulsion positive de 50 us pour un zero, de 26 us pour un un, avec un espace négatif de 50us. Avec naturellement des marges.



Pic 6: AM2302 Single-bus communication timing — Host signal — Sensor signal

Table 6: Single bus signal characteristics

Symbol	Parameter	min	typ	max	Unit
T_{be}	Host the start signal down time	0.8	1	20	ms
T_{go}	Bus master has released time	20	30	200	μs
T_{rel}	Response to low time	75	80	85	μs
T_{reh}	In response to high time	75	80	85	μs
T_{LOW}	Signal "0", "1" low time	48	50	55	μs
T_{H0}	Signal "0" high time	22	26	30	μs
T_{H1}	Signal "1" high time	68	70	75	μs
T_{en}	Sensor to release the bus time	45	50	55	μs

Traduction des mesures

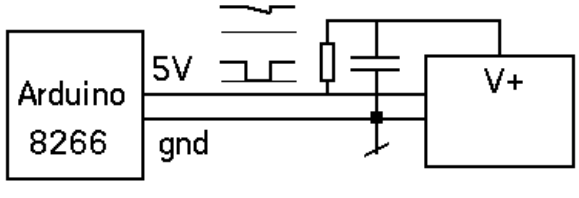
Le circuit donne 16 bits pour l'humidité relative, exprimée en dixièmes et codé en binaire.
Par exemple, si les 2 premiers bytes sont 0000 0010 1001 0010 = 0x0292
La conversion en décimal donne 658, l'humidité est 65.8%.
Un Serial.print () Arduino affichera 658 et c'est toujours comme cela que cela se passe avec les langages "évolués": les variables sont en binaires en mémoire, mais on les montre en décimal.

Si les 2 bytes suivant pour la température sont 1000 0000 0110 0101, le 1er bit à un indique que la température est négative et sa valeur absolue suit: 0000 0000 0110 0101 = 0x0065
La conversion en décimal donne 101, donc la température est de -10,1 degrés

Il semble important de vérifier que la somme des 4 premières mesures (en ignorant les dépassements) est égale à la somme de contrôle. Il faut alors recommencer la mesure.

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 DTH22, avec une légère ondulation pendant les transferts série.	

Librairie DTH22.h

Cette librairie définit les variables globales
int humid, int temp (16 bits, 9-10 bits utilisés), int sign (vu comme un booleen)

L'appel MesureDTH22(); doit être fait au max toutes les secondes.et mets à jour ces 3 variables.
La définition du câblage est du set-up est faite dans DTH22, voir les exemples.

Programme type

On veut afficher sur le terminal Arduino et sur Tell

```
#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 (#incluse Ybase.h).
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
}
```

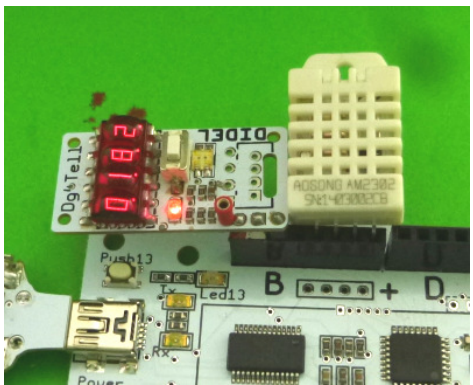
Exemple sur ESP8266

Idem JM?

Exemple Arduino/C sur 4 pins consécutives

Le circuit est inséré directement sur les pins Arduno 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.

```
#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
}
```



jdn 161122