

AZ-Delivery

Bienvenue !

Nous vous remercions d'avoir acheté notre sonde de température AZ-Delivery DS18B20 avec un câble de 3m. Dans les pages qui suivent, nous allons vous expliquer comment utiliser et configurer cet appareil pratique.

Amusez-vous bien !



Zones d'application

Le capteur de température convient à la mesure des températures dans les liquides, l'air et d'autres fluides. Le capteur est spécifiquement destiné à être utilisé avec des plates-formes de microcontrôleurs telles que Arduino et Raspberry Pi.

Connaissances et compétences requises

L'utilisation du capteur nécessite des connaissances de base en électronique, en mesure de température et en programmation de microcontrôleurs. Les utilisateurs doivent pouvoir connecter correctement le capteur, effectuer la programmation nécessaire et évaluer les données de mesure.

Des conditions de fonctionnement

Le capteur de température DS18B20 doit être utilisé dans des environnements qui ne dépassent pas la plage de température spécifiée du capteur (-55°C à +125°C). Le capteur est conçu pour être utilisé dans des liquides ou autres milieux à condition de ne pas dépasser la profondeur d'immersion maximale (manchon en acier inoxydable)

Conditions environnementales

Le manchon en acier inoxydable du capteur est étanche et peut être immergé dans des liquides. Cependant, il ne doit pas être utilisé dans des milieux chimiquement agressifs ou extrêmement corrosifs afin d'éviter d'endommager le manchon en acier inoxydable.

Utilisation prévue

Le capteur est destiné à être utilisé pour une mesure précise de la température dans divers milieux. La connexion s'effectue sur des plateformes de microcontrôleurs telles que Arduino et Raspberry Pi.

Utilisation inappropriée et prévisible

Les utilisations inappropriées mais prévisibles incluent l'utilisation du capteur dans des milieux extrêmement corrosifs, à des températures en dehors de la plage spécifiée ou dans des applications susceptibles de provoquer des dommages mécaniques au capteur. De telles utilisations peuvent entraîner des dommages au capteur et des mesures inexactes.

Élimination

Ne pas jeter avec les ordures ménagères ! Votre produit est conforme à celui européen Directive relative aux déchets d'équipements électriques et électroniques devant être éliminés de manière respectueuse de l'environnement. Les matières premières précieuses qu'elles contiennent peuvent être recyclées. L'application de cette directive contribue à la protection de l'environnement et de la santé. Utilisez le point de collecte mis en place par votre commune pour rapporter et Recyclage des anciens appareils électriques et électroniques. N° d'enregistrement DEEE : DE 62624346

consignes de sécurité

Attention : Les fluides chimiquement agressifs peuvent endommager le manchon en acier inoxydable. Remarque : N'utilisez pas le capteur dans des milieux corrosifs ou chimiquement agressifs. Attention : Une élimination inappropriée des capteurs peut mettre en danger l'environnement et la santé. Remarque : Jetez le capteur conformément aux réglementations locales et utilisez les options de recyclage appropriées. Attention : Les fluides chimiquement agressifs peuvent endommager le manchon en acier inoxydable. Remarque : N'utilisez pas le capteur dans des milieux corrosifs ou chimiquement agressifs. Attention : Une élimination inappropriée des capteurs peut mettre en danger l'environnement et la santé. Remarque : Jetez le capteur conformément aux réglementations locales et utilisez les options de recyclage appropriées. Attention : Les chocs mécaniques ou les flexions peuvent endommager le manchon en acier inoxydable et le capteur. Remarque : évitez les contraintes mécaniques et protégez le capteur des influences physiques. Attention : Un endommagement de la gaine thermorétractable peut affecter son étanchéité. Remarque : Vérifiez régulièrement l'état de la gaine thermorétractable et remplacez-la si elle présente des dommages visibles. Attention : Une fixation insuffisante peut entraîner des mesures incorrectes. Remarque : Assurez-vous que le capteur est fermement placé dans le milieu de mesure. Attention : Une sollicitation continue du câble de raccordement peut entraîner une rupture du câble. Remarque : Posez le câble de raccordement de manière à ce qu'il ne soit pas soumis à une contrainte de traction constante et protégez-le des influences mécaniques. Attention : un câblage incorrect peut entraîner des dysfonctionnements et endommager le capteur ou le microcontrôleur. Remarque : Assurez-vous de connecter correctement le capteur à Arduino ou Raspberry Pi conformément aux instructions du fabricant. Attention : Une surtension peut endommager définitivement le capteur ou le microcontrôleur. Remarque : re-

spectez strictement les limites de tension spécifiées du capteur et du microcontrôleur. Attention : Une mauvaise manipulation des connexions électriques peut entraîner des courts-circuits. Remarque : Assurez-vous que toutes les connexions sont isolées et protégées. Attention : Une surchauffe peut endommager le capteur. Remarque : faites fonctionner le capteur uniquement dans la plage de température spécifiée (-55°C à $+125^{\circ}\text{C}$). Attention : des changements brusques de température peuvent endommager le capteur. Remarque : évitez les changements rapides de température et laissez le capteur s'adapter lentement aux nouvelles températures. Attention : Des températures élevées peuvent endommager l'isolation du câble de raccordement. Remarque : veillez à ce que le câble de raccordement ne soit pas posé à proximité de sources de chaleur.

Sommaire

Le Pinout	5
Comment configurer l'IDE Arduino	6
Comment configurer le Raspberry Pi et le Python	10
Connexion de la sonde avec l'Atmega328PBoard	11
Connexion de la sonde avec Raspberry Pi	20
Activation de l'interface One-Wire	21
Activation de plusieurs interfaces One-Wire	24
Script Python pour la lecture de plusieurs sondes DS18B20	25

Az-Delivery

Le capteur de température DS18B20 est enfermé hermétiquement dans la sonde. L'extrémité de la sonde est en acier inoxydable et convient parfaitement à la mesure de la température dans les environnements humides.

Le DS18B20 est un capteur de température numérique qui fournit des mesures de température numériques de 9 à 12 bits et dispose d'une fonction d'alarme avec des points de déclenchement supérieurs et inférieurs non volatils programmables par l'utilisateur. Le capteur communique sur un bus à un fil qui ne nécessite qu'une broche de données, une broche d'alimentation et une broche de masse pour communiquer avec un microcontrôleur.

Chaque capteur possède une adresse série unique de 64 bits, ce qui permet à plusieurs capteurs de fonctionner sur le même bus One-Wire. Ainsi, il est simple d'utiliser un microcontrôleur pour contrôler de nombreux capteurs répartis sur une grande surface. Les applications qui peuvent bénéficier de cette fonctionnalité comprennent les systèmes de surveillance de la température à l'intérieur des bâtiments, des équipements ou des machines, ainsi que les systèmes de surveillance et de contrôle des processus.

Le capteur ne nécessite pas d'alimentation en veille, ce qui signifie que lorsque les données de température ne sont pas lues, le capteur ne consomme pas d'énergie du tout.

La gamme de température de mesure va de -55°C à $+125^{\circ}\text{C}$ (67°F à 257°F), avec une précision de $\pm 0,5^{\circ}\text{C}$ (9 bits) ; $\pm 0,25^{\circ}\text{C}$ (10 bits) ; $\pm 0,125^{\circ}\text{C}$ (11 bits) ; et $\pm 0,0625^{\circ}\text{C}$ (12 bits).

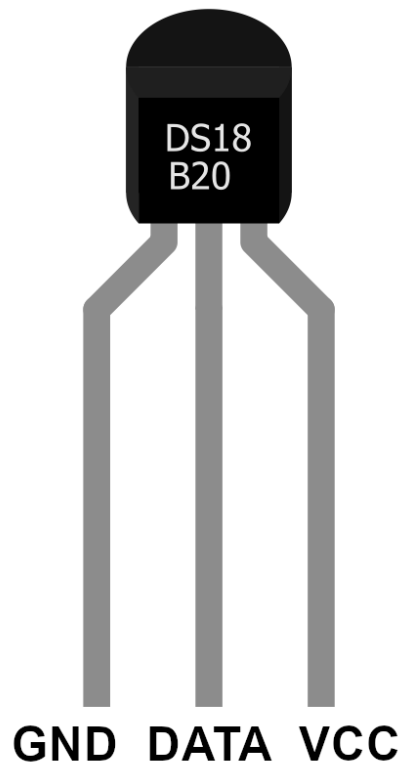
REMARQUE : si vous rencontrez des problèmes de communication, essayez d'ajouter une résistance élevée de $4,7k\Omega$ sur la broche *DATA*.

REMARQUE : Les données techniques relatives à l'interface *One-Wire* ne mentionnent pas le nombre maximum de sondes qui peuvent être reliées sur la même interface, mais dans les applications pratiques, ce nombre n'est pas aussi élevé, et vous devez y prêter attention.

REMARQUE : Il existe une limitation de la longueur des câbles qui doit être prise en considération lors de l'utilisation de communications à longue distance. Vous devez faire attention à la capacité et à la résistance distribuées par le câble.

REMARQUE : Le *DS18B20* et les transistors ordinaires se ressemblent, veuillez donc à ne pas le considérer comme un transistor pour éviter tout dommage !

Le pinout



"VCC" pin - fournit l'alimentation de la sonde. Bien que la tension d'alimentation puisse varier entre 3,3V et 5,5V, une alimentation de 5V est recommandée. Dans le cas d'une alimentation de 5V, vous pouvez utiliser un câble qui relie la sonde et le microcontrôleur jusqu'à 20 mètres. Cependant, avec une alimentation de 3,3V, la longueur du câble ne doit pas dépasser un mètre. Sinon, la chute de tension de la ligne entraînera des erreurs de mesure.

"DATA" pin - est la broche de données, et est utilisé pour la communication entre la sonde et le microcontrôleur (peut être connecté sur l'interface One-Wire).

"GND" pin - est la broche de masse et doit être connectée à la masse commune, ou 0V (sur Atmega328P Board ou Raspberry Pi).

Comment configurer l'IDE Arduino

Si vous n'avez pas encore installé Arduino IDE, allez sur le lien suivant :

<https://www.arduino.cc/en/Main/Software> et téléchargez le fichier

d'installation pour la plateforme de votre système d'exploitation.

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a large teal circle containing the Arduino logo (an infinity symbol with a minus and plus sign). To the right of the logo, the text reads: **ARDUINO 1.8.9**. Below this, it says: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." On the right side of the page, there is a teal sidebar with links for different operating systems: **Windows** (Installer, for Windows XP and up; ZIP file for non admin install), **Windows app** (Requires Win 8.1 or 10, with a 'Get' button), **Mac OS X** (10.8 Mountain Lion or newer), and **Linux** (32 bits, 64 bits, ARM 32 bits, ARM 64 bits). At the bottom of the sidebar, there are links for Release Notes, Source Code, and Checksums (sha512).

Pour Windows, les utilisateurs devront double-cliquer sur le fichier ".exe" téléchargé et suivre les instructions de la fenêtre d'installation.

Az-Delivery

Pour les utilisateurs de Linux, vous allez télécharger un fichier avec l'extension *".tar.xz"*, que vous devez extraire. Lorsque vous l'aurez extrait, allez dans le répertoire extrait, et ouvrez un terminal dans ce répertoire. Vous devez exécuter deux scripts *".sh"*, le premier appelé *"arduino-linux-setup.sh"* et le deuxième appelé *"install.sh"*.

Pour exécuter le premier script dans un terminal, exécutez la commande suivante :

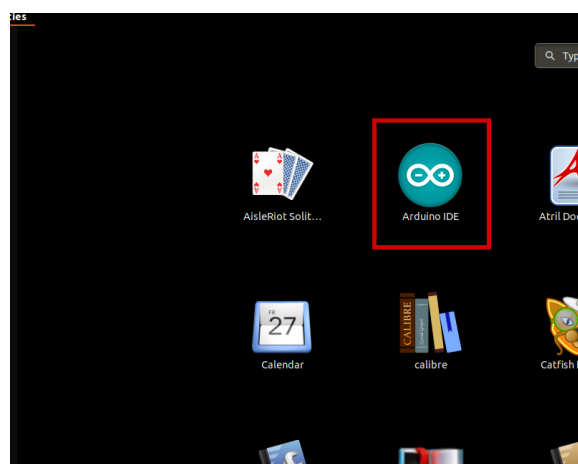
```
sh arduino-linux-setup.sh user_name
```

user_name - est le nom du super utilisateur dans le système d'exploitation Linux. Il vous sera demandé de fournir le mot de passe du super utilisateur. Attendez quelques minutes pour que le script complète tout.

Après l'installation du premier script, vous devez exécuter le second script appelé *"install.sh"*. Dans le terminal, exécutez la commande suivante :

```
sh install.sh
```

Après l'installation de ces scripts, allez dans le menu *"All Apps"*, où vous pouvez trouver l'IDE Arduino installé.



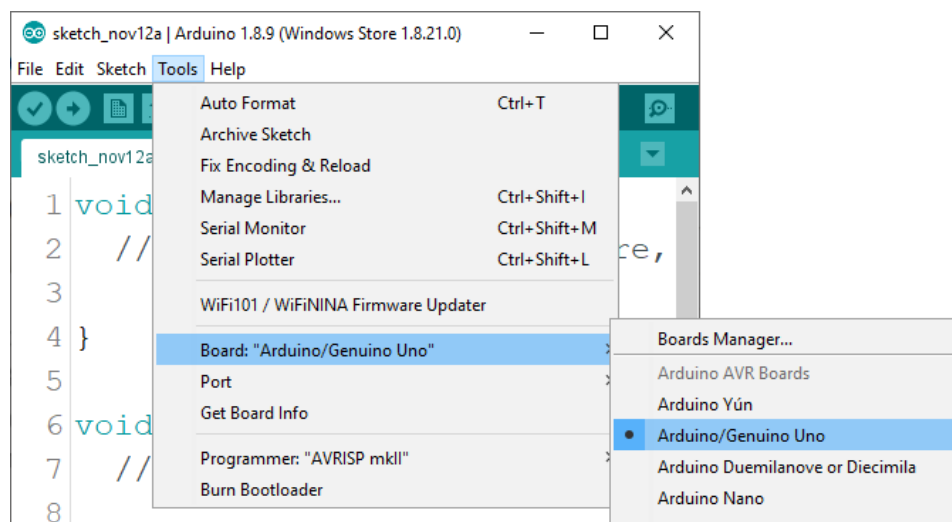
Az-Delivery

Presque tous les systèmes d'exploitation sont livrés avec un éditeur de texte préinstallé (Windows est livré avec le Bloc-notes, Linux Ubuntu est livré avec Gedit, Linux Raspbian est livré avec Leafpad, etc.) Tous ces éditeurs de texte sont parfaitement adaptés à l'objectif de l'eBook.

La prochaine étape est de vérifier si votre PC peut détecter votre carte microcontrôleur. Ouvrez l'IDE Arduino fraîchement installé, et allez dans :

Tools > Board > {votre nom de conseil ici}

{votre nom de conseil ici} devrait être l'Arduino/Genuino Uno, comme vous pouvez le voir sur l'image ci-dessous :



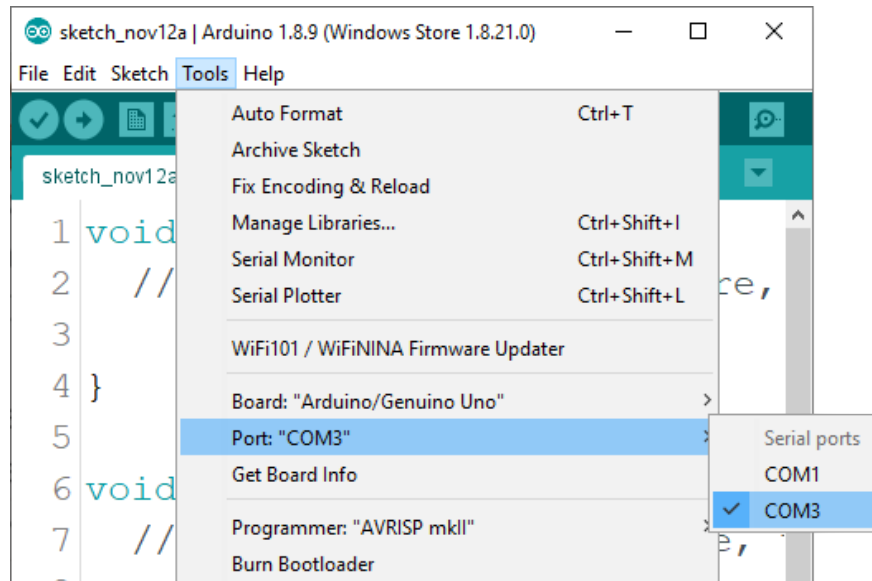
Vous devez sélectionner le port sur lequel la carte microcontrôleur est connectée. Allez à Tools > Port > {le nom du port va ici}

Az-Delivery

et si vous avez connecté la carte microcontrôleur sur le port usb, vous devriez voir un nom de port.

Az-Delivery

Si vous utilisez Arduino IDE sous Windows, les noms des ports sont les suivants :



Pour les utilisateurs de Linux, le nom du port est "/dev/ttyUSBx" par exemple, où "x" représente un nombre entier entre 0 et 9, par conséquent.



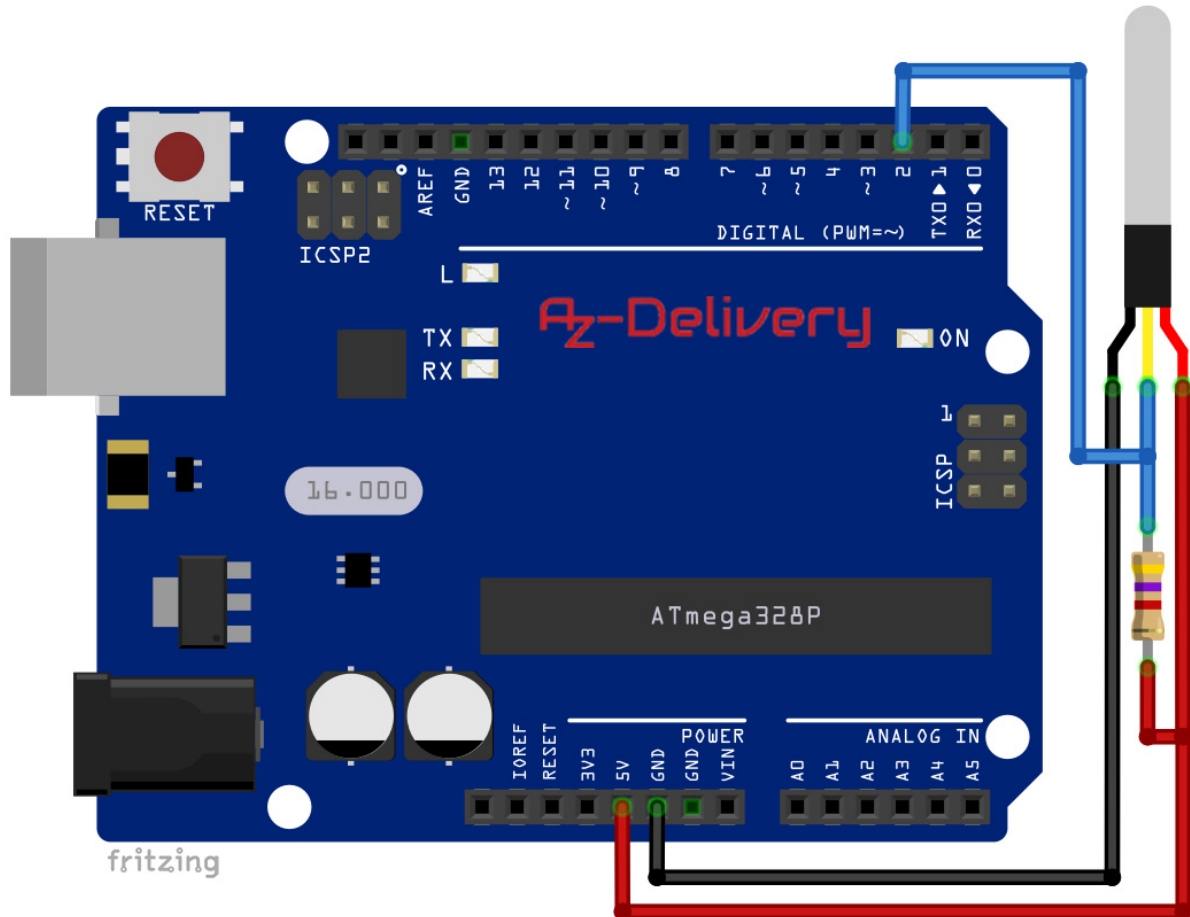
Comment configurer le Raspberry Pi et le Python

Pour le Raspberry Pi, vous devrez d'abord installer le système d'exploitation sur celui-ci, puis tout configurer pour pouvoir l'utiliser en mode *"Headless"*. Le mode *"Headless"* vous permet de vous connecter à distance au Raspberry Pi, sans avoir besoin de l'écran du PC, de la souris ou du clavier. Les seules choses dont vous avez besoin pour ce mode sont le Raspberry Pi, une alimentation électrique et une connexion Internet. Tout ceci est expliqué en détail dans le eBook gratuit *"Raspberry Pi Quick Startup Guide"*, que vous pouvez trouver sur notre site : <https://www.az-delivery.de/products/raspberry-pi-kostenfreies-e-book?ls=en>

Le système d'exploitation Rasbian est livré avec Python préinstallé.

Connexion de la sonde avec l'Atmega328P

Connectez la sonde DS18B20 avec la carte microcontrôleur comme indiqué sur le schéma de connexion suivant :



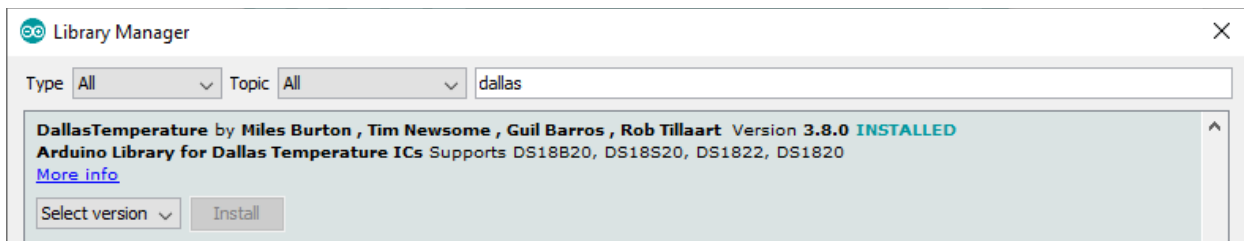
DS18B20 pin	>	Board pin
DATA	>	D2 Câble bleu
VCC	>	5V Câble Rouge
GND	>	GND Câble Noir

REMARQUE : une résistance de 4,7kΩ est connectée entre la broche DATA et VCC.

Az-Delivery

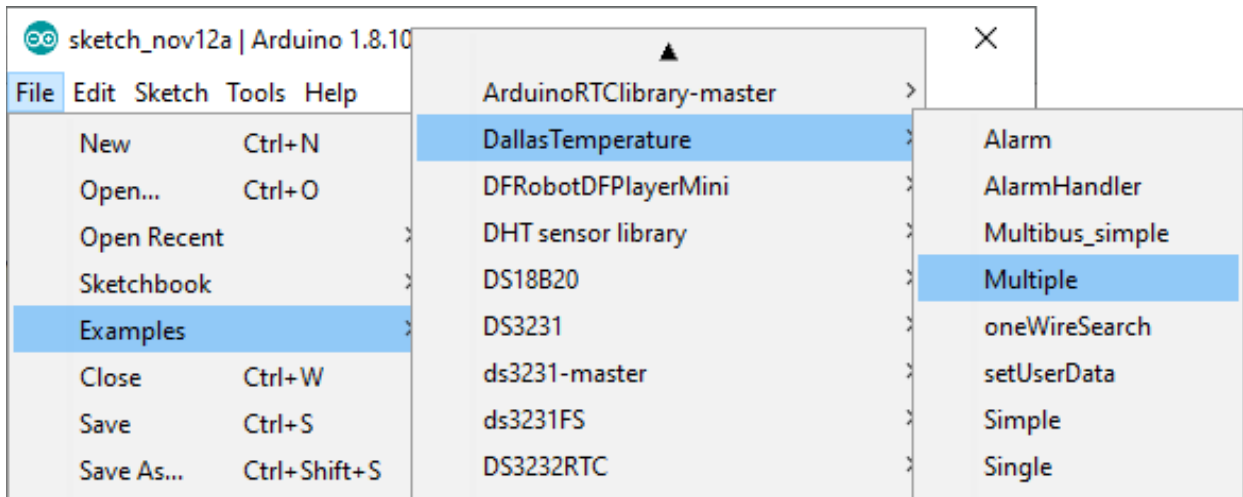
Pour utiliser la sonde DS18B20 avec les carte microcontrôleur, nous devons d'abord télécharger une bibliothèque correspondante. Allez sur : Tools > Manage Libraries, une nouvelle fenêtre apparaîtra. Tapez "Dallas" dans le champ de recherche et téléchargez la bibliothèque "DallasTemperature" de "Miles Burton, Tim Newsome Guil Barros et Rob Tillaart", comme le montre l'image suivante :

La version de la bibliothèque utilisée est 3.7.5



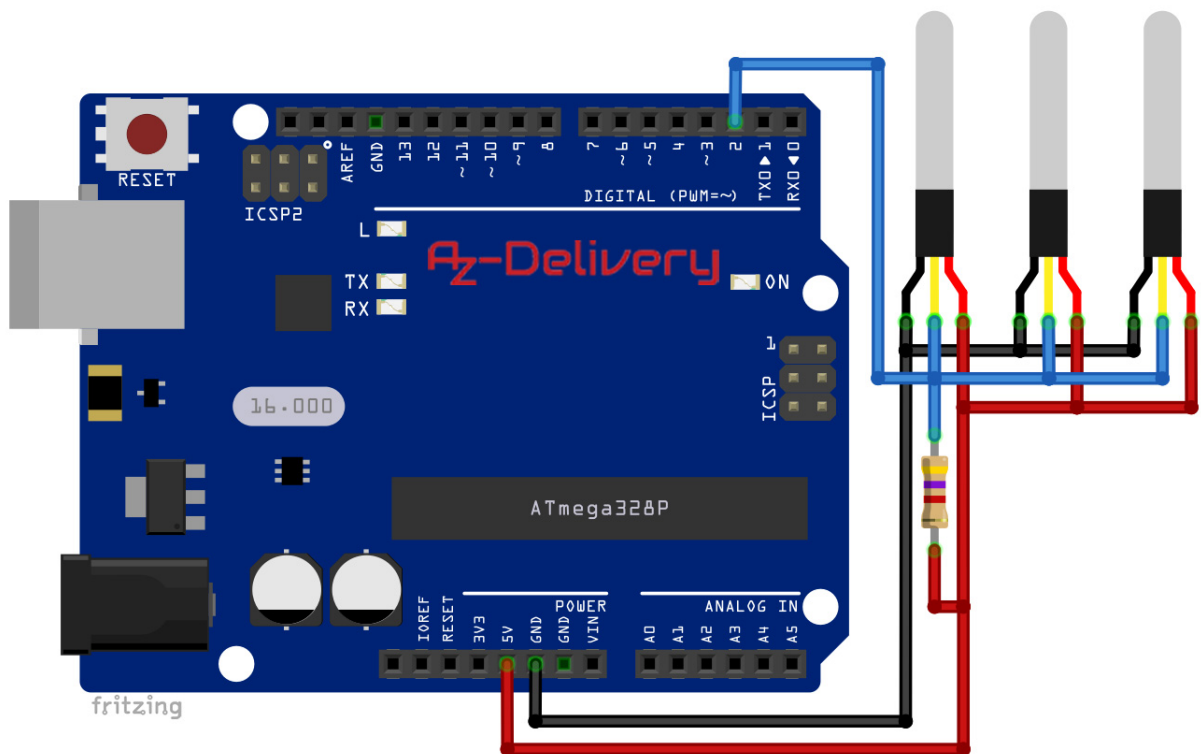
Maintenant, allez dans : *File > Examples > DallasTemperature > ...* et vous trouverez de nombreux exemples de sketchs. Nous allons utiliser et modifier un sketch appelé "Multiple" afin de lire les données de température de trois sondes DS18B20 différentes.

Az-Delivery



Si vous n'utilisez qu'une seule sonde DS18B20, le sketch appelé "Simple" est suffisant.

Connectez trois sondes DS18B20 avec la carte microcontrôleur comme indiqué sur le schéma de connexion suivant :



Voici l'exemple de croquis pour trois sondes DS18B20 sur la même interface à un fil :



```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2 // Data wire is plugged into D2 pin
#define TEMPERATURE_PRECISION 12
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
DeviceAddress one, two, three;

void setup() {
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");
  sensors.begin();
  Serial.print("Locating devices...");
  Serial.print("Found ");
  Serial.print(sensors.getDeviceCount(), DEC);
  Serial.println(" devices.");
  Serial.print("Parasite power is: ");
  if(sensors.isParasitePowerMode()) {
    Serial.println("ON");
  }
  else {
    Serial.println("OFF");
  }
  if(!sensors.getAddress(one, 0)) {
    Serial.println("Unable to find address for Device 0"); }
  if(!sensors.getAddress(two, 2)) {
    Serial.println("Unable to find address for Device 2"); }
  if(!sensors.getAddress(three, 1)) {
    Serial.println("Unable to find address for Device 1"); }
```

Az-Delivery

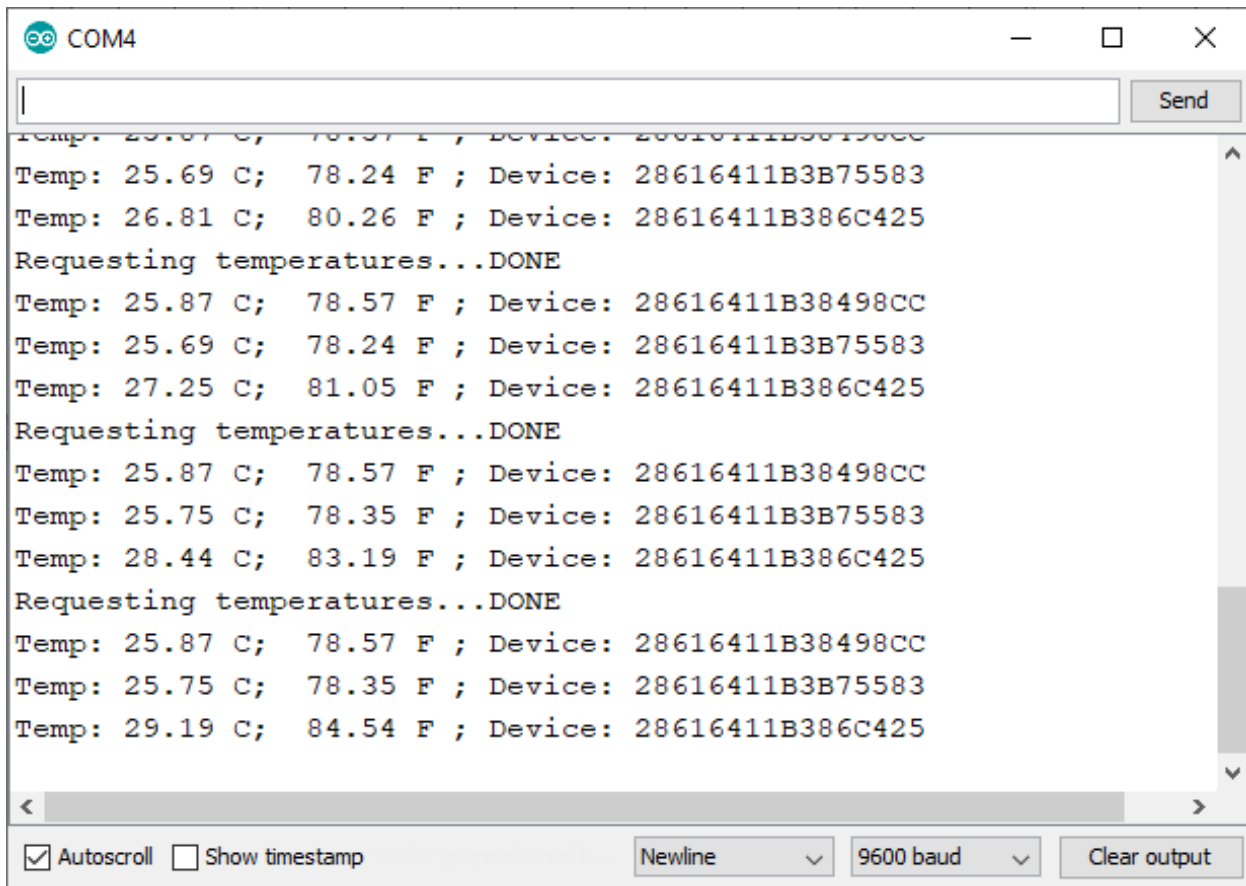
```
Serial.print("Device 0 Address: ");
  printAddress(one);
  Serial.println(); Serial.print("Device 1 Address: ");
  printAddress(two);
  Serial.println(); Serial.print("Device 2 Address: ");
  printAddress(three); Serial.println();
  sensors.setResolution(one, TEMPERATURE_PRECISION);
  sensors.setResolution(two, TEMPERATURE_PRECISION);
  sensors.setResolution(three, TEMPERATURE_PRECISION);
// one tab
  Serial.print("Device 0 Resolution: ");
  Serial.print(sensors.getResolution(one), DEC);
  Serial.println(); Serial.print("Device 1 Resolution: ");
  Serial.print(sensors.getResolution(two), DEC);
  Serial.println(); Serial.print("Device 2 Resolution: ");
  Serial.print(sensors.getResolution(three), DEC);
  Serial.println();
}
void printAddress(DeviceAddress deviceAddress) {
  for(uint8_t i = 0; i < 8; i++) {
    if(deviceAddress[i] < 16) Serial.print("0");
    Serial.print(deviceAddress[i], HEX);
  }
}
void printTemperature(DeviceAddress deviceAddress) {
  float tempC = sensors.getTempC(deviceAddress);
  Serial.print("Temp: "); Serial.print(tempC);
  Serial.print(" C; ");
  Serial.print(DallasTemperature::toFahrenheit(tempC));
  Serial.print(" F");
}
void printResolution(DeviceAddress deviceAddress) {
  Serial.print("Resolution: ");
  Serial.println(sensors.getResolution(deviceAddress));
}
void printData(DeviceAddress deviceAddress) {
  printTemperature(deviceAddress); Serial.print(" ; Device: ");
  printAddress(deviceAddress); Serial.println();
}
```

Az-Delivery

```
void loop() {  
  Serial.print("Requesting temperatures...");  
  sensors.requestTemperatures(); Serial.println("DONE");  
  printData(one);  
  printData(two);  
  printData(three);  
  delay(1000);  
}
```

Téléchargez le sketch sur l'Atmega328P Board et ouvrez Serial Monitor (Tools > Serial Monitor). La sortie devrait ressembler à celle montrée dans l'image ci-dessous :

Az-Delivery



```
Temp: 25.87 C; 78.57 F ; Device: 28616411B38498CC
Temp: 25.69 C; 78.24 F ; Device: 28616411B3B75583
Temp: 26.81 C; 80.26 F ; Device: 28616411B386C425
Requesting temperatures...DONE
Temp: 25.87 C; 78.57 F ; Device: 28616411B38498CC
Temp: 25.69 C; 78.24 F ; Device: 28616411B3B75583
Temp: 27.25 C; 81.05 F ; Device: 28616411B386C425
Requesting temperatures...DONE
Temp: 25.87 C; 78.57 F ; Device: 28616411B38498CC
Temp: 25.75 C; 78.35 F ; Device: 28616411B3B75583
Temp: 28.44 C; 83.19 F ; Device: 28616411B386C425
Requesting temperatures...DONE
Temp: 25.87 C; 78.57 F ; Device: 28616411B38498CC
Temp: 25.75 C; 78.35 F ; Device: 28616411B3B75583
Temp: 29.19 C; 84.54 F ; Device: 28616411B386C425
```

Voici une explication du croquis.

Nous utilisons la variable `ONE_WIRE_BUS` pour définir sur quelle broche numérique nous allons connecter l'interface *One-Wire*. Dans le cadre de cet eBook, la valeur de la variable `ONE_WIRE_BUS` est fixée à `D2`, mais vous pouvez utiliser n'importe quelle autre broche numérique de la carte microcontrôleur, à l'exception de celles utilisées dans l'interface série, `D0` et `D1` (c'est une recommandation, vous pouvez les utiliser, mais vous devez vous assurer que ces broches sont déconnectées lorsque vous téléchargez des croquis).

Az-Delivery

Nous avons utilisé la variable `TEMPERATURE_PRECISION` pour définir la précision des sondes `DS18B20`. Le nombre enregistré dans cette variable est un nombre de conversion numérique en bits et il peut être compris entre 9 et 12, tout autre nombre entraînera une erreur. Pour les besoins de cet eBook, nous l'avons fixé à la valeur maximale, 12.

Nous avons utilisé la ligne de code suivante :

```
DeviceAddress one, two, three
```

pour créer des variables pour les adresses des sondes, et dans notre exemple nous en avons créé trois.

Nous avons défini et créé l'objet `oneWire`, utilisé pour l'interface `OneWire` :

```
OneWire oneWire(ONE_WIRE_BUS);
```

Ensuite, nous avons utilisé l'objet `oneWire` pour définir et créer l'objet `sensors`, qui est utilisé pour toutes les sondes connectées :

```
DallasTemperature sensors(&oneWire)
```

Az-Delivery

Pour initialiser l'objet capteurs nous avons utilisé la ligne de code suivante : *sensors.begin()*

Avec cette ligne de code, l'objet *sensors* détecte toutes les sondes connectées sur l'interface *One-Wire*. Il détecte également toutes les adresses des sondes.

Maintenant on peut vérifier si les sondes fonctionnent correctement, en utilisant les lignes de code suivantes pour chaque sonde que nous connectons à l'interface *One-Wire* :

```
if(!sensors.getAddress(one, 0)) {  
    Serial.println("Unable to find address: Device  
0"); }
```

où *one* est l'adresse de la première sonde.

Pour configurer la précision de la conversion analogique-numérique de la sonde spécifique, nous avons utilisé la ligne de code suivante :

```
sensors.setResolution(one, TEMPERATURE_PRECISION)
```

Si vous souhaitez lire la précision de la conversion analogique-numérique d'une sonde spécifique, vous pouvez utiliser la ligne de code suivante :

```
sensors.getResolution(one)
```

La fonction renvoie une valeur hexadécimale, et pour la convertir en valeur décimale, nous utilisons la ligne de code suivante :

```
Serial.print(sensors.getResolution(one), DEC);
```

Afin de lire les données de température, nous devons d'abord demander toutes les données de toutes les sondes, en utilisant la ligne de code suivante :


```
sensors.requestTemperatures();
```

C'est seulement après cette ligne de code que nous pouvons lire les données d'une sonde particulière, en utilisant la ligne de code suivante :

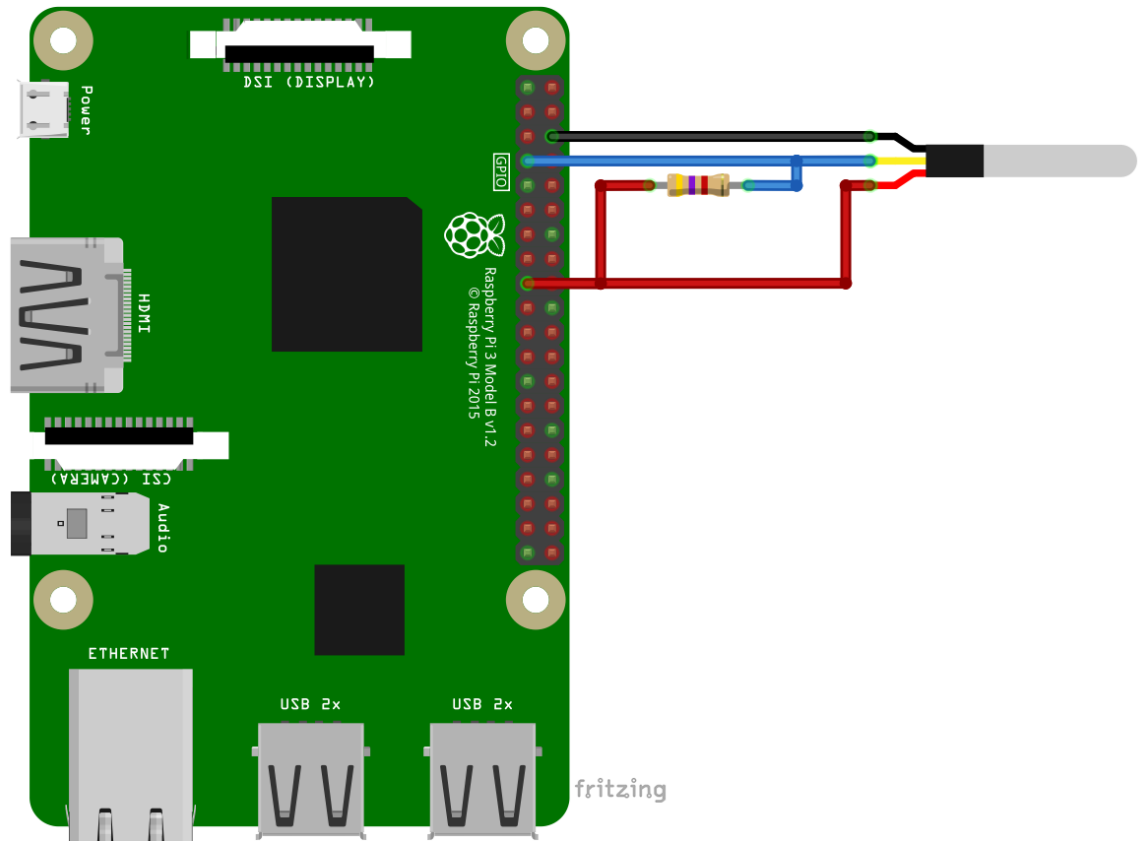
```
float tempC = sensors.getTempC(deviceAddress);
```

où nous passons l'argument *deviceAddress* à la fonction afin de lire les données de température d'une sonde spécifique. Ces données sont des valeurs de température en *Celsius*, et pour les convertir en *Fahrenheit*, nous avons utilisé la ligne de code suivante :

```
DallasTemperature::toFahrenheit(tempC)
```

Connexion de la sonde avec Raspberry Pi

Connectez la sonde DS18B20 avec le Raspberry Pi comme indiqué sur le schéma de connexion suivant :



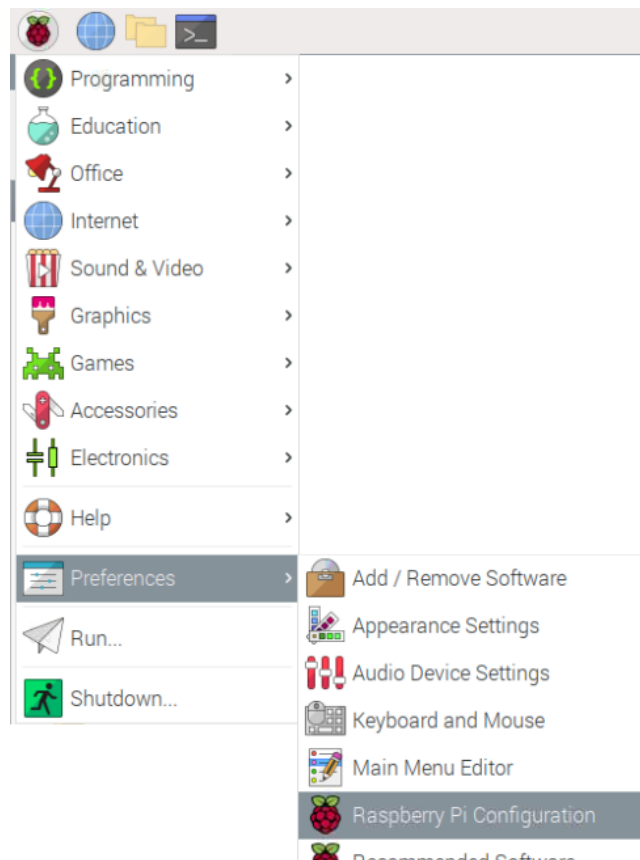
DS18B20 pin	>	Raspberry Pi pin
GND	>	GND [pin 4] Câble Noir
DATA	>	GPIO4 [pin 7] Câble Bleu
VCC	>	3V3 [pin 17] Câble Rouge

REMARQUE : Une résistance de 4,7kΩ est connectée entre la broche OUT et la broche 3V3.

Activation de l'interface One-Wire

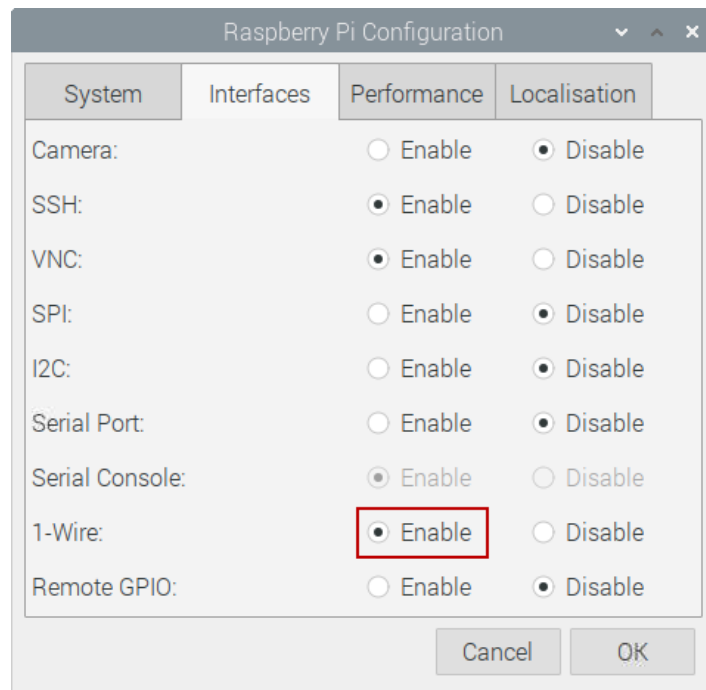
Avant de pouvoir utiliser la sonde DS18B20 avec le Raspberry Pi, nous devons d'abord activer l'interface One-Wire dans Raspbian. Par défaut, l'interface matérielle One-Wire est sur la broche GPIO4 (broche 7), mais nous devons d'abord l'activer. Afin d'activer l'interface One-Wire, ouvrez All Apps et allez dans :

Preferences > Raspberry Pi Configuration, comme indiqué ci-dessous :

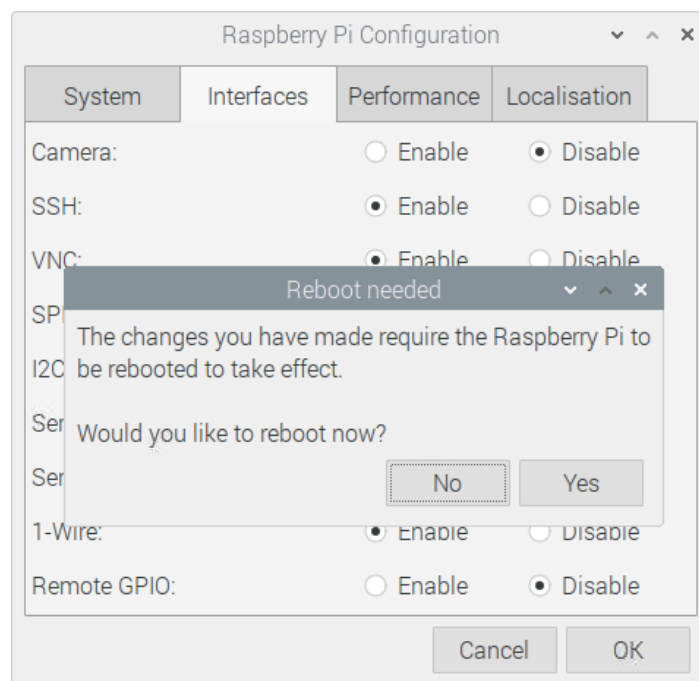


Az-Delivery

Lorsque la nouvelle fenêtre apparaît, ouvrez l'onglet *"Interfaces"* et recherchez les boutons radio appelés *"1-Wire"*, puis sélectionnez le bouton radio *"Enable"*, comme le montre l'image suivante :



Il vous sera demandé de redémarrer le système.



Az-Delivery

Lorsque le Raspbian est redémarré, ouvrez le terminal, et exécutez les commandes suivantes, une par une :

```
sudo modprobe w1-gpio
```

```
sudo modprobe w1-therm
```

```
cd /sys/bus/w1/devices/
```

et quand vous exécutez la commande suivante :

```
ls
```

la sortie dans le terminal devrait être la suivante :

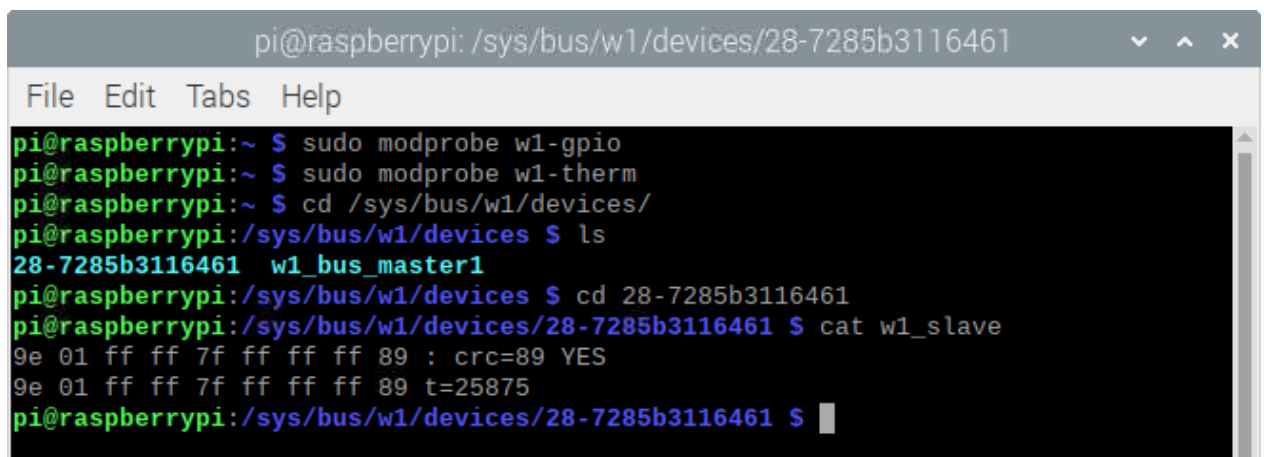
```
28-7285b3116461 and w1_bus_master1
```

le premier nombre (28-7285b3116461) sera différent pour vous, car il s'agit de l'adresse série de la sonde spécifique, et chaque sonde a sa propre adresse série unique. Maintenant pour tester si tout marche, exécutez ces deux commandes :

```
cd 28-7285b3116461 - un numéro ou une adresse de série de la dernière page
```

```
cat w1_slave
```

La sortie devrait ressembler à celle montrée dans



```
pi@raspberrypi: /sys/bus/w1/devices/28-7285b3116461
File Edit Tabs Help
pi@raspberrypi:~ $ sudo modprobe w1-gpio
pi@raspberrypi:~ $ sudo modprobe w1-therm
pi@raspberrypi:~ $ cd /sys/bus/w1/devices/
pi@raspberrypi:/sys/bus/w1/devices $ ls
28-7285b3116461 w1_bus_master1
pi@raspberrypi:/sys/bus/w1/devices $ cd 28-7285b3116461
pi@raspberrypi:/sys/bus/w1/devices/28-7285b3116461 $ cat w1_slave
9e 01 ff ff 7f ff ff ff 89 : crc=89 YES
9e 01 ff ff 7f ff ff ff 89 t=25875
pi@raspberrypi:/sys/bus/w1/devices/28-7285b3116461 $
```

l'image suivante :

t=25875 - il s'agit des données de température en °C (Celsius) = 25,875°C.

Activation de plusieurs interfaces One-Wire

Pour activer l'interface *One-Wire*, sans interface graphique utilisateur (*GUI*), avant de redémarrer votre Raspberry Pi, au fichier situé sur *"/boot/config.txt"* vous devez ajouter la ligne suivante :

```
dtoverlay=w1-gpio
```

ou

```
dtoverlay=w1-gpio,gpiopin=x
```

où x est une broche personnalisée, si vous souhaitez l'utiliser (par défaut, *GPIO4* [broche 7], comme nous l'avons mentionné dans le chapitre précédent).

Les noyaux plus récents (4.9.28 et plus) vous permettent d'utiliser le chargement dynamique de la superposition à la place, y compris la création de plusieurs interfaces One-Wire à utiliser en même temps :

```
sudo dtoverlay w1-gpio gpiopin=4 pullup=0 # header  
pin 7  
sudo dtoverlay w1-gpio gpiopin=17 pullup=0 # header pin  
11  
sudo dtoverlay w1-gpio gpiopin=27 pullup=0 # header pin  
13
```

Une fois que toutes les étapes ci-dessus ont été effectuées, et que la découverte est terminée, vous pouvez lister les périphériques que le Raspberry Pi a découvert via toutes les interfaces One-Wire en exécutant la commande suivante dans le terminal :

```
ls /sys/bus/w1/devices/
```

NOTE : L'utilisation de **w1-gpio** sur le Raspberry Pi nécessite généralement une résistance pull-up de

4,7k Ω connectée entre la broche GPIO et une alimentation de 3,3V.

Script Python pour la lecture de plusieurs sondes DS18B20

Nous avons choisi de séparer le code en deux scripts, pour une meilleure lisibilité. Ce qui suit est un code pour l'indice de classe :

```
import os
import glob
import time

class DS18B20:

    def __init__(self):
        os.system('modprobe w1-gpio')
        os.system('modprobe w1-therm')
        base_dir = '/sys/bus/w1/devices/'
        device_folder = glob.glob(base_dir + '28*')
        self._count_devices = len(device_folder)
        self._devices = list()
        i = 0
        while i < self._count_devices:
            self._devices.append(device_folder[i] +
'/w1_slave')
            i += 1

    def device_names(self):
        names = list()
        for i in range(self._count_devices):
            names.append(self._devices[i])
            temp = names[i][20:35]
```


Az-Delivery

```
names[i] = temp
```

Az-Delivery

```
return names
```

```
# (one tab)
```

```
def _read_temp(self, index):
    f = open(self._devices[index], 'r')
    lines = f.readlines()
    f.close()
    return lines

def tempC(self, index = 0):
    lines = self._read_temp(index)
    retries = 5
    while (lines[0].strip()[-3:] != 'YES') and (retries
> 0):
        time.sleep(0.1)
        lines = self._read_temp(index)
        retries -= 1

    if retries == 0:
        return 998

    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp = lines[1][equals_pos + 2:]
        return float(temp) / 1000
    else:
        return 999 # error

def device_count(self):
    return self._count_devices
```

(la plupart du code dans le script est modifié à partir du script sur le site adafruit)

Enregistrez le script sous le nom de *"DS18B20classfile.py"*.

Az-Delivery

Ce qui suit est un code pour le script principal :

```
import time
from DS18B20classFile import DS18B20

degree_sign = u'\xb0' # degree sign
devices = DS18B20()
count = devices.device_count()
names = devices.device_names()

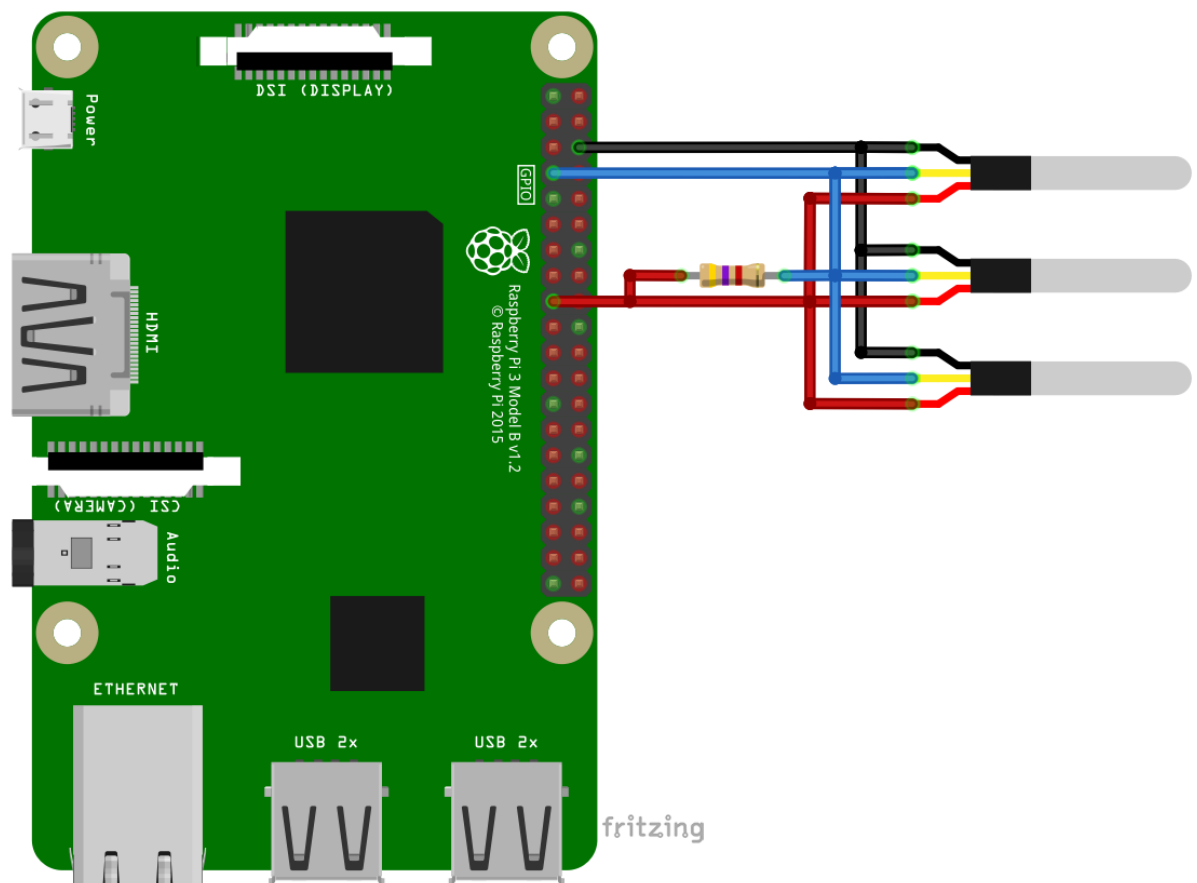
print('[press ctrl+c to end the script]')
try: # Main program loop
    while True:
        i = 0
        print('\nReading temperature, number of sensors: {}'.format(count))
        while i < count:
            container = devices.tempC(i)
            print('{} Temp: {:.3f}{}C, {:.3f}{}F of the device'.format(i+1, container, degree_sign, container * 9.0 / 5.0 + 32.0, degree_sign, names[i]))
            i = i + 1
        time.sleep(1)

# Scavenging work after the end of the program
except KeyboardInterrupt:
    print('Script end!')
```

Az-Delivery

Sauvegardez le script sous le nom *"DS18B20multiple.py"* dans le même répertoire où vous avez sauvegardé le premier script.

Par exemple, nous avons connecté trois sondes DS18B20 sur la même interface One-Wire du Raspberry Pi comme indiqué sur le schéma de connexion suivant :

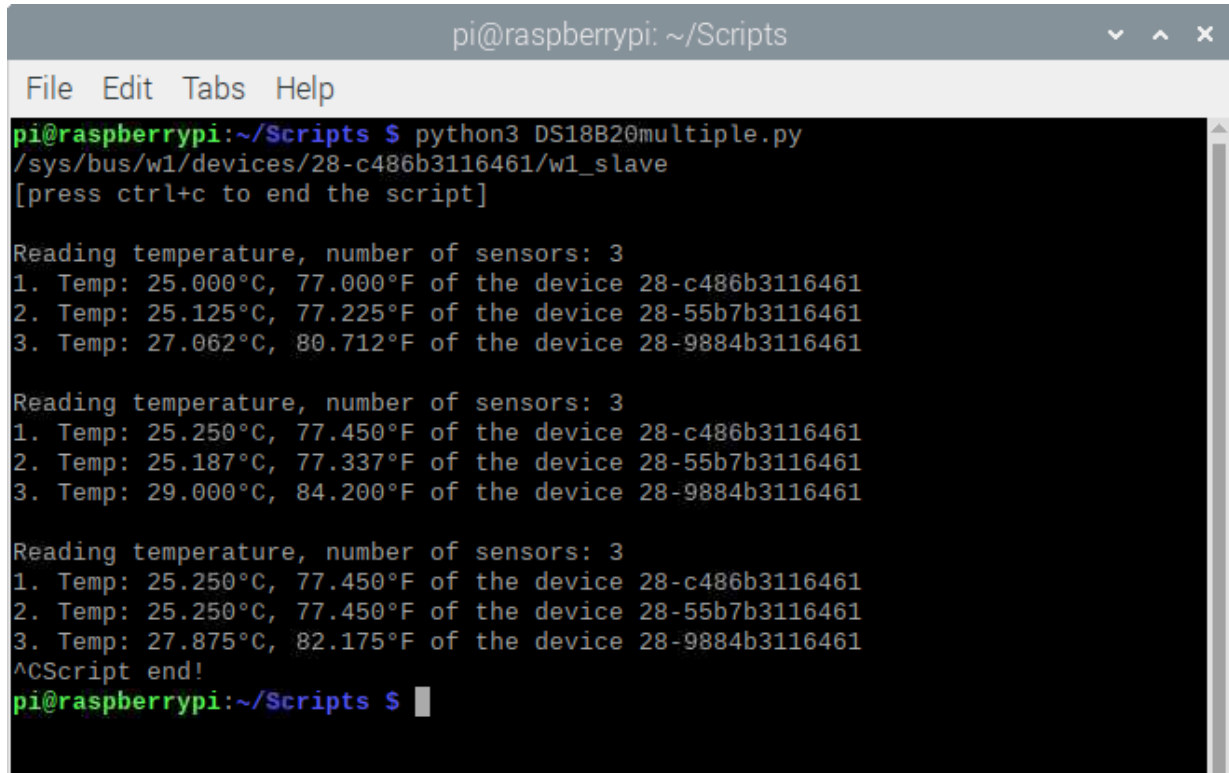


Az-Delivery

Pour exécuter le script principal, ouvrez un terminal dans le répertoire où vous avez enregistré les deux scripts, et exécutez la commande suivante :

python3 DS18B20multiple.py

La sortie devrait ressembler à celle montrée dans l'image ci-dessous :



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 DS18B20multiple.py
/sys/bus/w1/devices/28-c486b3116461/w1_slave
[press ctrl+c to end the script]

Reading temperature, number of sensors: 3
1. Temp: 25.000°C, 77.000°F of the device 28-c486b3116461
2. Temp: 25.125°C, 77.225°F of the device 28-55b7b3116461
3. Temp: 27.062°C, 80.712°F of the device 28-9884b3116461

Reading temperature, number of sensors: 3
1. Temp: 25.250°C, 77.450°F of the device 28-c486b3116461
2. Temp: 25.187°C, 77.337°F of the device 28-55b7b3116461
3. Temp: 29.000°C, 84.200°F of the device 28-9884b3116461

Reading temperature, number of sensors: 3
1. Temp: 25.250°C, 77.450°F of the device 28-c486b3116461
2. Temp: 25.250°C, 77.450°F of the device 28-55b7b3116461
3. Temp: 27.875°C, 82.175°F of the device 28-9884b3116461
^CScript end!
pi@raspberrypi:~/Scripts $
```

Pour arrêter le script, appuyez sur "CTRL + C" sur le clavier.

Vous pouvez facilement utiliser le script pour une ou plusieurs sondes DS18B20.

Ça y est, vous avez réussi !

Vous pouvez maintenant utiliser votre module pour divers projets.



Il est maintenant temps d'apprendre et de réaliser les projets par vous-même. Vous pouvez le faire à l'aide de nombreux exemples de scripts et d'autres didacticiels, que vous trouverez sur Internet.

Si vous recherchez des microélectronique et accessoires de haute qualité, AZ-Delivery Vertriebs GmbH est l'entreprise idéale pour vous les procurer. Vous recevrez de nombreux exemples d'application, des guides d'installation complets, des livres électroniques, des bibliothèques et l'assistance de nos experts techniques.

<https://az-delivery.de>

Amusez-vous !

Mentions légales

<https://az-delivery.de/pages/about-us>