Doc LM35

À propos du LM35

Le LM35 est un capteur de température centigrade de précision peu coûteux fabriqué par Texas Instruments . Il fournit une tension de sortie qui est linéairement proportionnelle à la température Centigrade et est, par conséquent, très facile à utiliser avec l'Arduino.

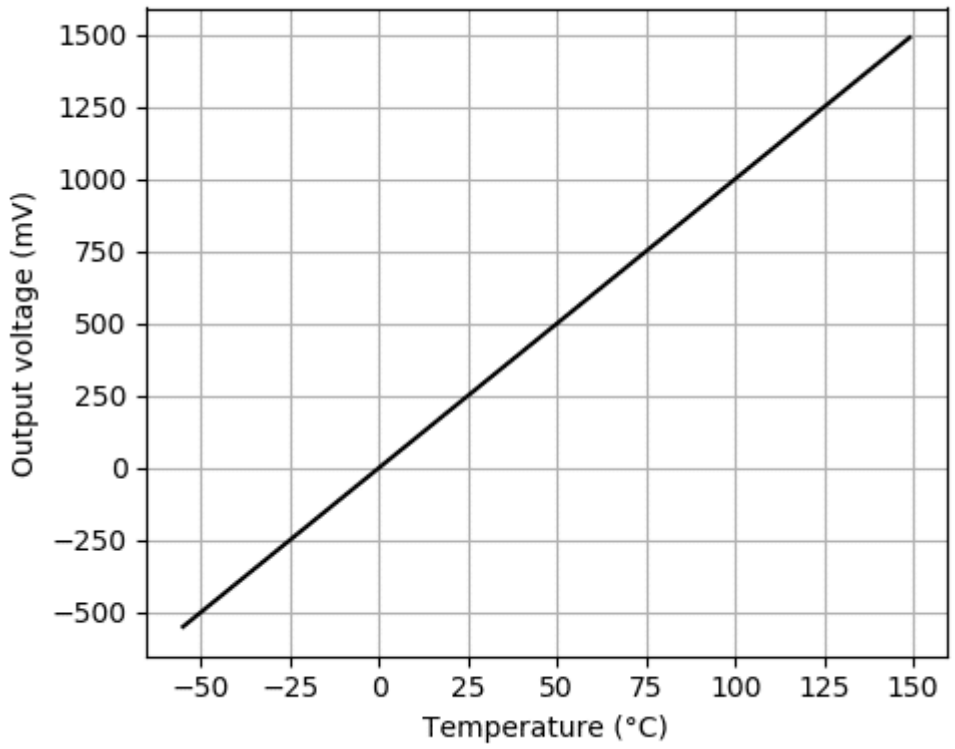
Le capteur ne nécessite aucun étalonnage ou ajustement externe pour fournir des précisions de ± 0,5 ° C à température ambiante et de ± 1 ° C sur la plage de température de −50 ° C à + 155 ° C.

L'un des inconvénients du capteur est qu'il nécessite une tension de polarisation négative pour lire les températures négatives. Donc, si cela est nécessaire pour votre projet, je recommande d'utiliser le DS18B20 ou TMP36 à la place. Le TMP36 d'Analog Devices est très similaire au LM35 et peut lire des températures de -40 ° C à 125 ° C sans aucun composant externe.

Vous pouvez trouver un tutoriel dédié pour le TMP36 et le DS18B20 ici:

- Capteur de température analogique TMP36 avec tutoriel Arduino
- Le guide complet des capteurs de température numériques DS18B20 avec Arduino

Le facteur d'échelle de sortie du LM35 est de 10 mV / ° C et il fournit une tension de sortie de 250 mV à 25 ° C (voir la figure ci-dessous).



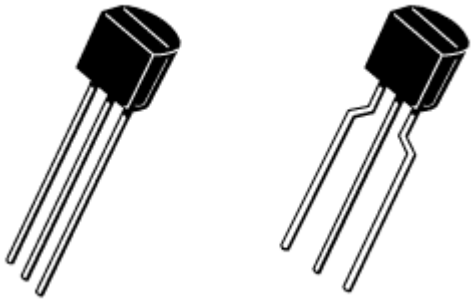
Notez que le capteur fonctionne sur une plage de tension de 4 à 30 V et que la tension de sortie est indépendante de la tension d'alimentation.

Le LM35 fait partie d'une série de capteurs de température analogiques vendus par Texas Instruments. Les autres membres de la série comprennent:

- LM335 - tension de sortie directement proportionnelle à la température absolue à 10 mV / ° K.
- LM34 - tension de sortie linéairement proportionnelle à la température Fahrenheit 10 mV / ° F.

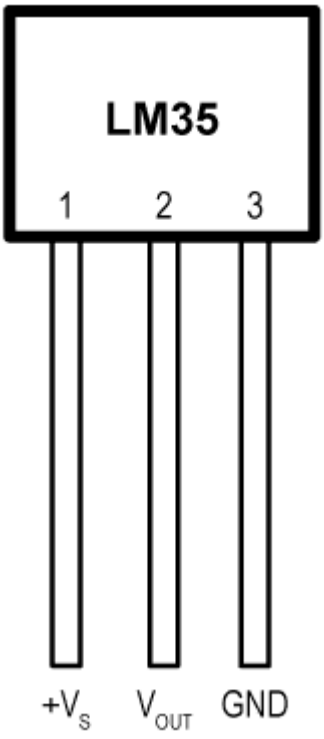
Brochage LM35

Le LM35 est disponible en 4 boîtiers différents, mais le type le plus courant est le boîtier à transistor TO-92 à 3 broches.



Paquet TO-92

Le brochage du capteur est le suivant:



Notez que la broche 1 (+ V_S) est la broche la plus à gauche lorsque le côté plat du capteur (avec le texte imprimé dessus) est tourné vers vous.

Nom	Épingle	Description
+ V_S	1	Broche d'alimentation positive (4 - 30 V)

Nom	Épingle	Description
V_{OUT}	2	Sortie analogique du capteur de température
GND	3	Broche de terre de l'appareil, connectez à la borne négative de l'alimentation

Vous pouvez trouver les spécifications du LM35 dans le tableau ci-dessous.

Spécifications du capteur de température analogique LM35

Tension d'alimentation	4 V à 30 V
Courant de fonctionnement	60 µA
Écart de température	-55 ° C à + 155 ° C
Précision garantie	± 0,5 ° C à + 25 ° C ± 1 ° C de -55 ° C à + 150 ° C
Facteur d'échelle de sortie	10 mV / ° C
Tension de sortie à 25 ° C	250 mV
Auto-chauffant	<0,1 ° C dans l'air calme
Paquet	TO-92 3 broches
Fabricant	Texas Instruments
Coût	Vérifiez le prix

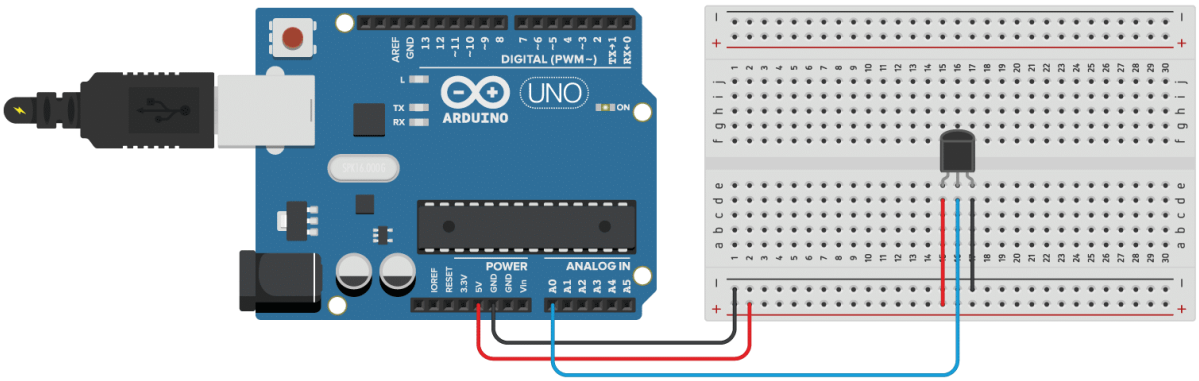
Pour plus d'informations, vous pouvez également consulter la fiche technique ici:

Fiche technique LM35

Câblage - Connexion du capteur de température analogique LM35 à Arduino

La connexion d'un LM35 à l'Arduino est très simple car il vous suffit de connecter 3 broches. Commencez par connecter la broche + V_S à la sortie 5 V de l'Arduino et la broche GND à la terre.

Ensuite, connectez la broche du milieu (V_{OUT}) à l'une des entrées analogiques de l'Arduino. Dans ce cas, j'ai utilisé la broche d'entrée analogique A0.



Capteur de température analogique LM35 avec schéma de câblage Arduino Uno

Les connexions sont également données dans le tableau ci-dessous:

Connexions du capteur de température analogique LM35

LM35	Arduino
Broche 1 (+ V_S)	5 V
Broche 2 (V_{OUT})	Broche A0
BROCHE 3 (GND)	GND

Conversion de la tension de sortie du LM35 en température

Pour convertir la tension de sortie du capteur en température en degrés Celsius, vous pouvez utiliser la formule suivante:

Temperature (°C) = V_{OUT} / 10

avec V_{OUT} en millivolt (mV). Donc, si la sortie du capteur est de 750 mV, la température est de 75 ° C.

Comme vous pouvez le voir dans le schéma de câblage ci-dessus, la sortie du LM35 est connectée à l'une des entrées analogiques de l'Arduino. La valeur de cette entrée analogique peut être lue avec la fonction analogRead () . Cependant, cette fonction ne retournera pas réellement la tension de sortie du capteur.

Les cartes Arduino contiennent un convertisseur analogique-numérique (ADC) 10 bits multicanal, qui mappera les tensions d'entrée entre 0 et la tension de fonctionnement (5 V ou 3,3 V) en valeurs entières comprises entre 0 et 1023. Sur un Arduino Uno, par exemple , cela donne une résolution entre les lectures de 5 volts / 1024 unités ou de 0,0049 volts (4,9 mV) par unité.

Donc, si vous utilisez analogRead () pour lire la tension sur l'une des entrées analogiques de l'Arduino, vous obtiendrez une valeur comprise entre 0 et 1023.

Pour reconvertir cette valeur en tension de sortie du capteur, vous pouvez utiliser:

V_{OUT} = lecture depuis ADC * (Vref / 1024)

Nous utiliserons ces formules dans les exemples de code ci-dessous.

Capteur de température analogique LM35 avec exemple de code Arduino

Avec l'exemple de code suivant, vous pouvez lire la température à partir d'un capteur LM35 et l'afficher dans le moniteur série.

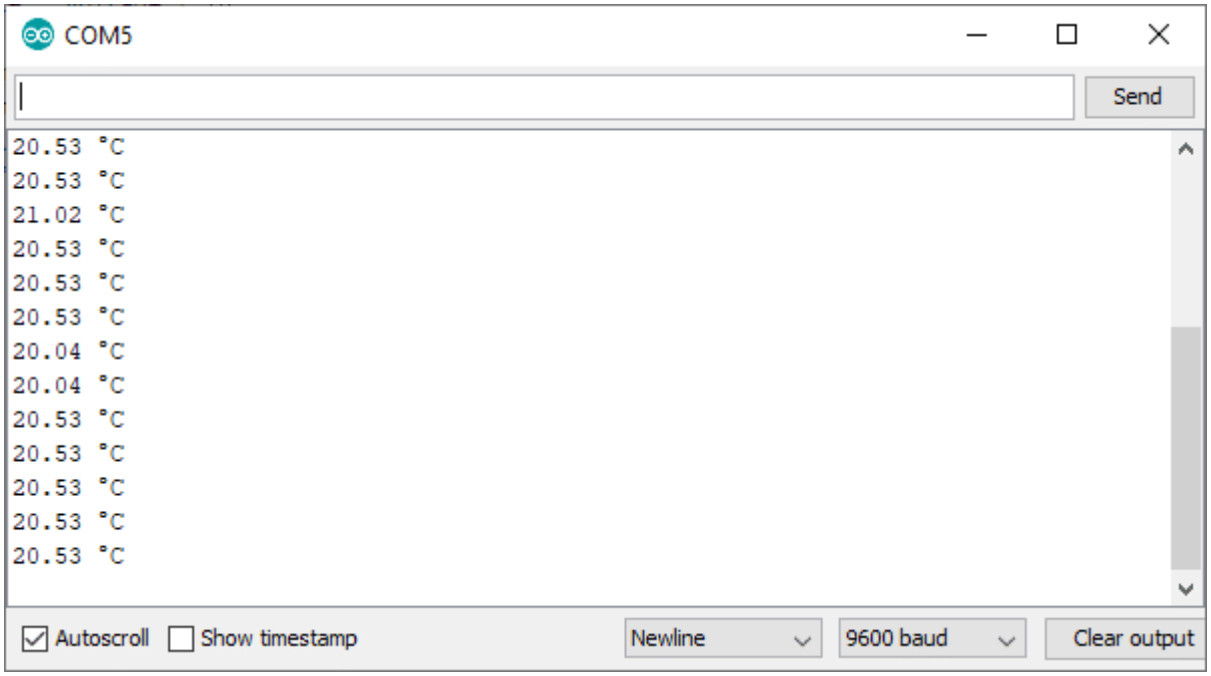
Vous pouvez télécharger l'exemple de code sur votre Arduino à l'aide de l' IDE Arduino .

Pour copier le code, cliquez sur le bouton dans le coin supérieur droit du champ de code.

/ * Capteur de température analogique LM35 avec exemple de code Arduino. Plus d'infos: https://www.makerguides.com * /
// Définit à quelle broche de l'Arduino la sortie du LM35 est connectée:

```
#define sensorPin A0
void setup () {
// Commencez la communication série à une vitesse de transmission de 9600:
En série. commencer ( 9600 ) ;
}
boucle void () {
// Obtenez une lecture du capteur de température:
lecture int = analogRead ( sensorPin ) ;
// Convertit la lecture en tension:
tension flottante = lecture * ( 5000 / 1024,0 ) ;
// Convertit la tension en température en degrés Celsius:
température du flotteur = tension / 10;
// Imprimer la température dans le moniteur série:
En série. impression ( température ) ;
En série. impression ( "\ xC2 \ xB0" ) ; // montre le symbole du degré
En série. println ( "C" ) ;
retard ( 1000 ) ; // attend une seconde entre les lectures
}
```

Vous devriez voir la sortie suivante dans le moniteur série:



Sortie moniteur série

Assurez-vous que le débit en bauds du moniteur série est également réglé sur 9600.

Comment fonctionne le code

Tout d'abord, j'ai défini à quelle broche de l'Arduino la broche V_{OUT} du capteur est connectée.

Dans ce cas, nous avons utilisé la broche analogique A0. L'instruction `#define` peut être utilisée pour donner un nom à une valeur constante. Le compilateur remplacera toutes les références à cette

constante par la valeur définie lors de la compilation du programme. Donc partout où vous parlez `sensorPin`, le compilateur le remplacera par A0 lorsque le programme sera compilé.

// Définit à quelle broche de l'Arduino la sortie du LM35 est connectée:
#define sensorPin A0

Dans la section de configuration du code, nous commençons la communication série à une vitesse de transmission de 9600.

void setup () {
// Commencez la communication série à une vitesse de transmission de 9600:
En série. commencer (9600) ;
}

Dans la section boucle du code, nous commençons par prendre une lecture du capteur avec la fonction `analogRead(pin)`.

// Obtenez une lecture du capteur de température:
lecture int = analogRead (sensorPin) ;

Ensuite, nous utilisons les formules que j'ai mentionnées plus tôt dans l'article pour convertir la lecture en tension puis en température.

// Convertit la lecture en tension:
tension flottante = lecture * (5000 / 1024,0) ;
// Convertit la tension en température en degrés Celsius:
température du flotteur = tension / 10;

Enfin, les résultats sont imprimés dans Serial Monitor:

// Imprimer la température dans le moniteur série:
En série. impression (température) ;
En série. impression ("\ xC2 \ xB0") ; // montre le symbole du degré
En série. println ("C") ;

Améliorer la précision des lectures

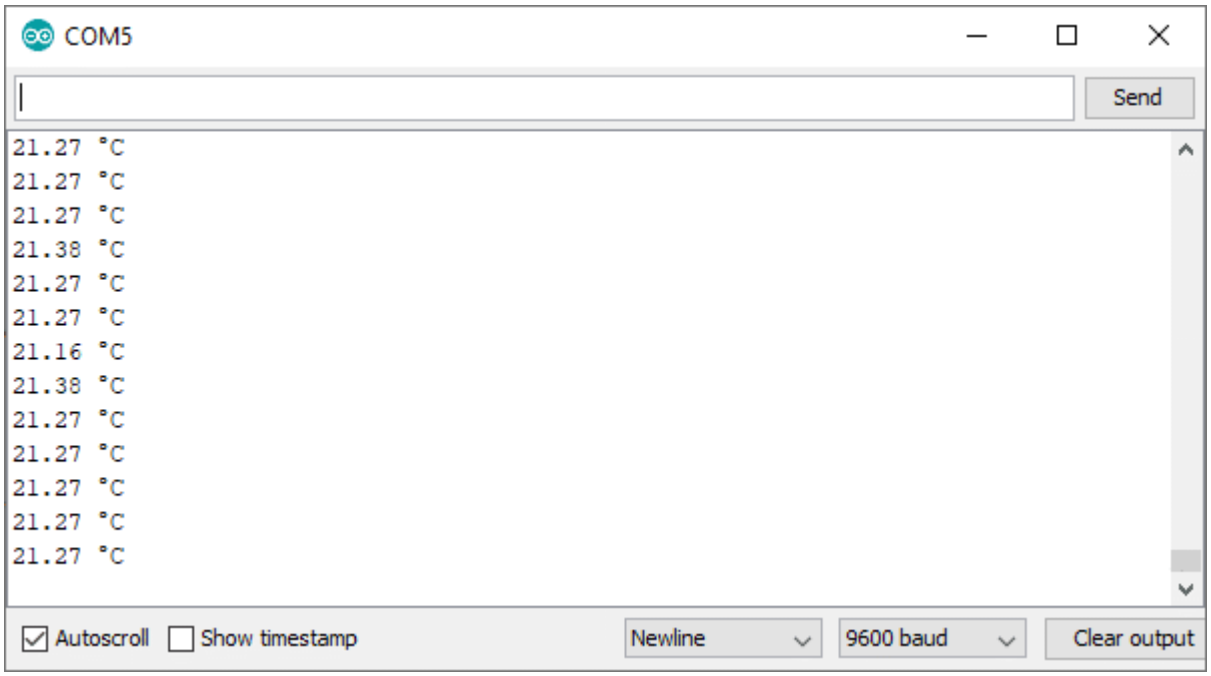
Parce que nous avons utilisé la tension de référence par défaut de l'Arduino pour l'entrée analogique (c'est-à-dire la valeur utilisée comme haut de la plage d'entrée), la résolution maximale que nous obtenons de l'ADC est 5000/1024 = 4,88 mV ou 0,49 ° C.

Si nous voulons une précision plus élevée, nous pouvons utiliser à la place la référence 1.1 V intégrée de l'Arduino. Cette tension de référence peut être modifiée à l'aide de la fonction analogReference () .

Avec 1,1 V comme tension de référence, nous obtenons une résolution de 1100/1024 = 1,07 mV ou 0,11 ° C. Notez que cela limite la plage de température que nous pouvons mesurer entre 0 et 110 degrés Celsius.

J'ai mis en évidence les lignes que vous devez ajouter / modifier dans le code ci-dessous:

```
/ * Capteur de température analogique LM35 avec exemple de code Arduino. Plus d'infos:
https://www.makerguides.com * /
// Définit à quelle broche de l'Arduino la sortie du LM35 est connectée:
#define sensorPin A0
void setup () {
// Commencez la communication série à une vitesse de transmission de 9600:
En série. commencer ( 9600 ) ;
// Réglez la tension de référence pour l'entrée analogique sur la référence intégrée de 1,1 V:
analogReference ( INTERNAL ) ;
}
boucle void () {
// Obtenez une lecture du capteur de température:
lecture int = analogRead ( sensorPin ) ;
// Convertit la lecture en tension:
tension flottante = lecture * ( 1100 / 1024.0 ) ;
// Convertit la tension en température en degrés Celsius:
température du flotteur = tension / 10;
// Imprimer la température dans le moniteur série:
En série. impression ( température ) ;
En série. impression ( "\ xC2 \ xB0" ) ; // montre le symbole du degré
En série. println ( "C" ) ;
retard ( 1000 ) ; // attend une seconde entre les lectures
}
```



Notez les plus petits incréments entre les lectures

LM35 avec I2C LCD et exemple de code Arduino

Si vous voulez créer un thermomètre autonome qui n'a pas besoin d'ordinateur, il peut être intéressant de savoir comment afficher les lectures de température sur un écran LCD.

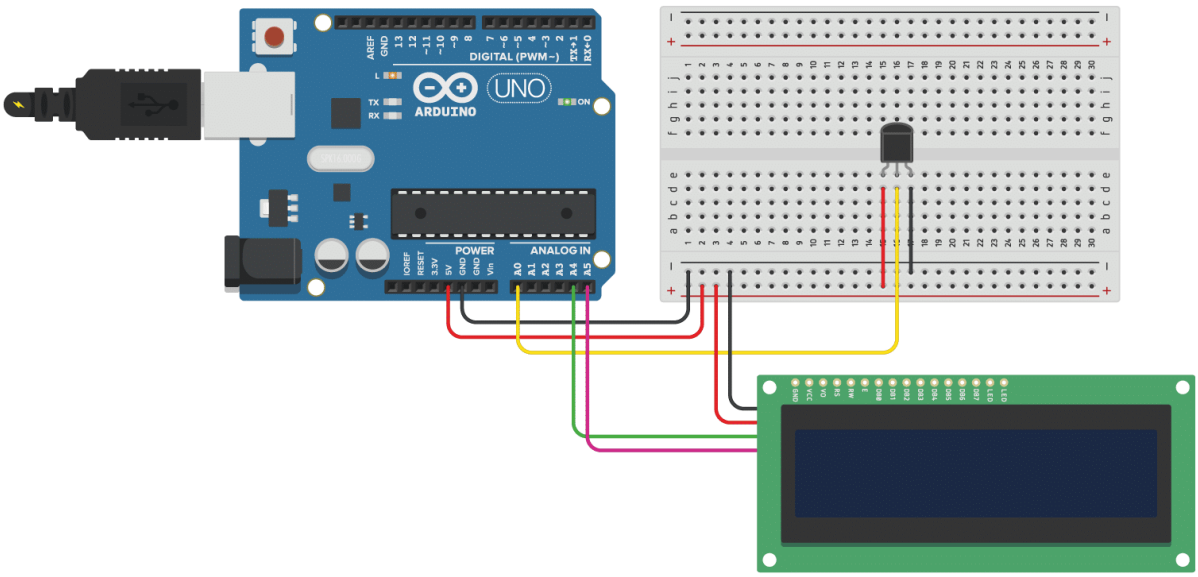
Avec l'exemple de code ci-dessous, vous pouvez afficher les lectures de température sur un écran LCD I2C 16 × 2 caractères .

La connexion de l'écran LCD I2C est assez simple comme vous pouvez le voir dans le schéma de câblage ci-dessous. Vous pouvez consulter mon tutoriel détaillé ci-dessous pour plus d'informations.

- Comment contrôler un écran LCD I2C avec Arduino

Si vous souhaitez utiliser un écran LCD non I2C standard à la place, consultez cet article:

- Comment utiliser un écran LCD 16 × 2 caractères avec Arduino



Capteur de température analogique LM35 avec schéma de câblage I2C LCD 16 × 2 caractères et Arduino.

Les connexions sont également données dans le tableau ci-dessous:

Connexions LCD I2C

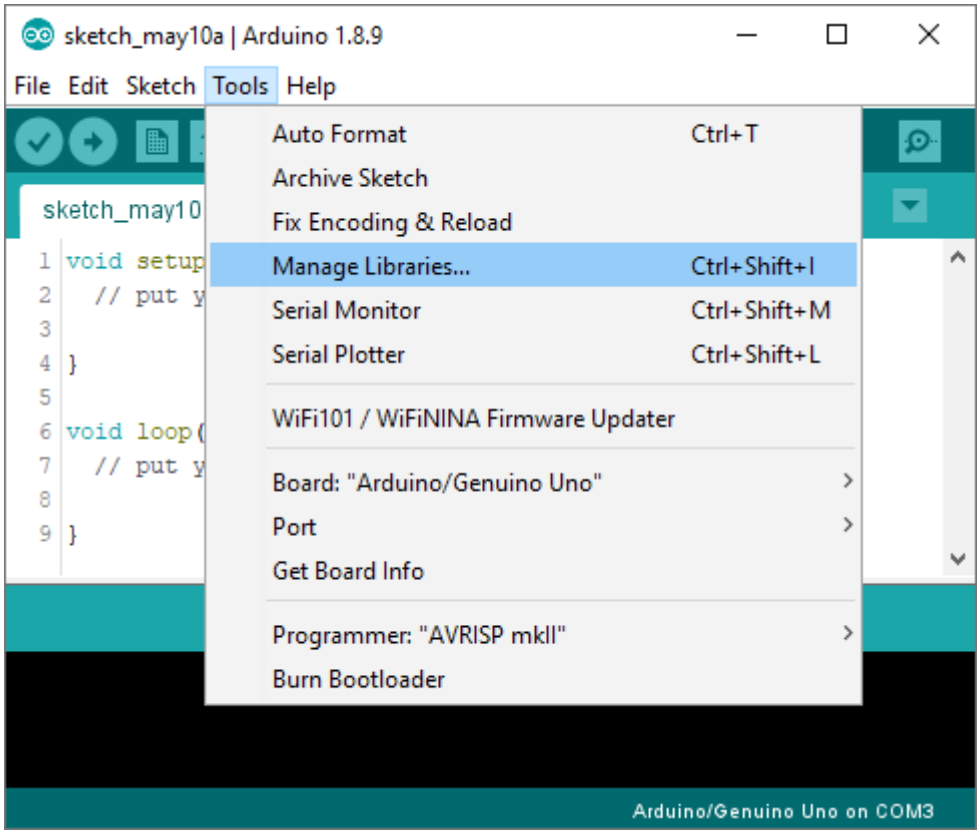
Ecran LCD à caractères I2C	Arduino
GND	GND
VCC	5 V
SDA	A4
SCL	A5

Notez que le capteur de température LM35 est connecté de la même manière que précédemment.

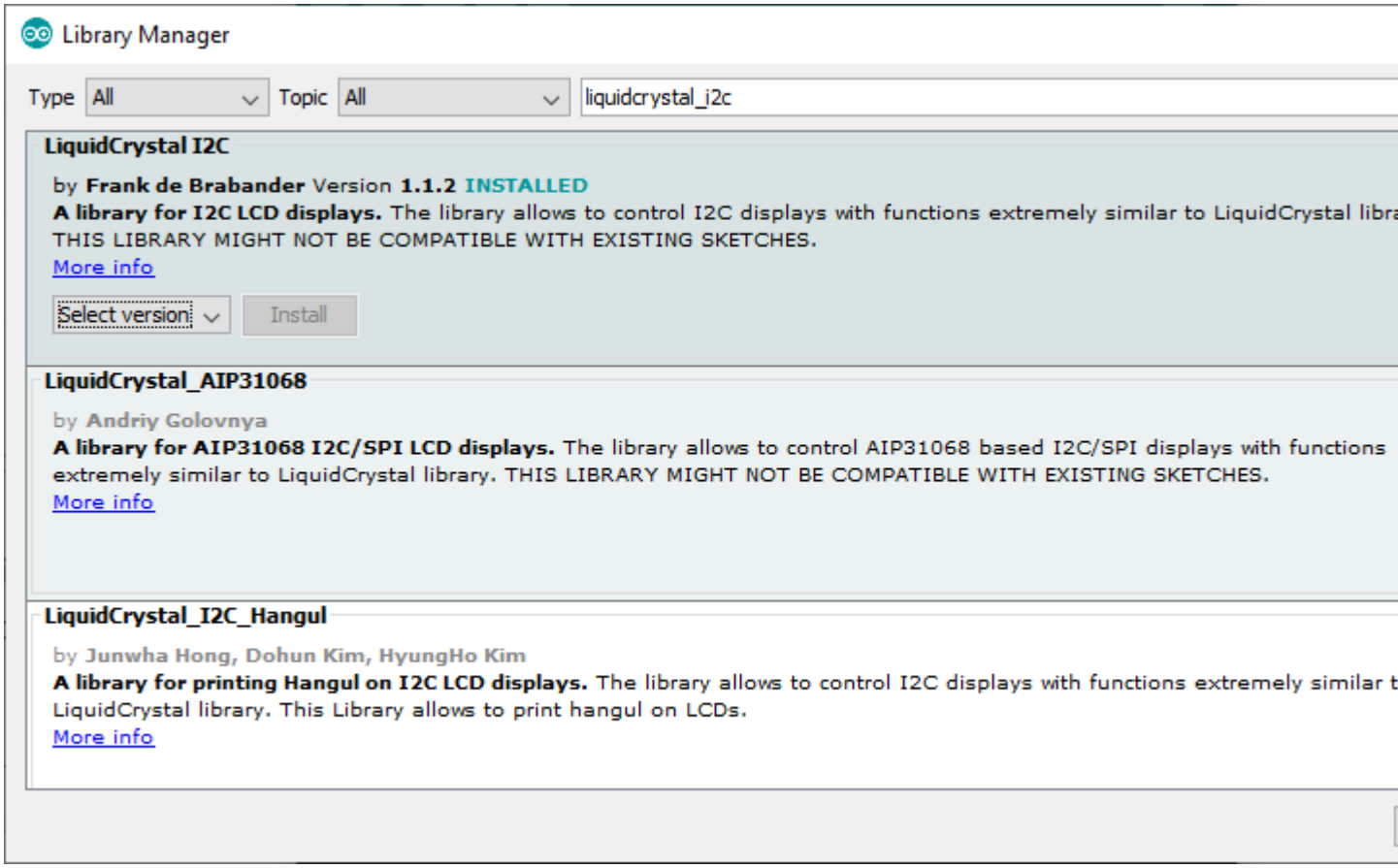
Installation des bibliothèques Arduino requises

Pour utiliser un écran LCD I2C, vous devez installer la bibliothèque Arduino **LiquidCrystal_I2C** .

Pour installer cette bibliothèque, accédez à Outils> Gérer les bibliothèques (Ctrl + Maj + I sous Windows) dans l' IDE Arduino . Le gestionnaire de bibliothèque ouvrira et mettra à jour la liste des bibliothèques installées.



Recherchez maintenant 'liquidcrystal_i2c' et recherchez la bibliothèque de **Frank de Brabander** . Sélectionnez la dernière version, puis cliquez sur Installer.



Installation de la bibliothèque Arduino LiquidCrystal_I2C

LM35 avec exemple de code LCD I2C

/ * Capteur de température analogique LM35 avec exemple de code LCD I2C et Arduino. Plus d'infos: https://www.makerguides.com * /
// Inclut les bibliothèques Arduino requises:
#include <LiquidCrystal_I2C.h>
// Créer une nouvelle instance de la classe LiquidCrystal_I2C:
LCD LiquidCrystal_I2C (0x27 , 16, 2) ;
// Symbole de degré:
octet Degré [] = {
B00111,
B00101,
B00111,
B00000,
B00000,
B00000,
B00000,
B00000
};
// Définit à quelle broche de l'Arduino la sortie du LM35 est connectée:
#define sensorPin A0
void setup () {
// Démarre l'écran LCD et allume le rétroéclairage:

```
lcd. init () ;
lcd. rétroéclairage () ;
// Créer un personnage personnalisé:
lcd. createChar ( 0, degré ) ;
}
boucle void () {
// Obtenez une lecture du capteur de température:
lecture int = analogRead ( sensorPin ) ;
// Convertit la lecture en tension:
tension flottante = lecture * ( 5000 / 1024,0 ) ;
// Convertit la tension en température en degrés Celsius:
température du flotteur = tension / 10;
// Imprime la température sur l'écran LCD;
lcd. setCursor ( 0, 0 ) ;
lcd. print ( "Température:" ) ;
lcd. setCursor ( 0, 1 ) ;
lcd. impression ( température ) ;
lcd. écrire ( 0 ) ; // imprime le caractère personnalisé
lcd. imprimer ( "C" ) ;
retard ( 1000 ) ; // attend une seconde entre les lectures
}
```

Vous devriez voir la sortie suivante sur l'écran LCD: