

LM35 et Arduino - Mesure de température et affichage sur écran LCD

Dans cet article, nous présentons un tutoriel sur la manière d'**interfacer LM35 et Arduino** avec son programme. Une fois que nous avons réussi à interfacer arduino et lm35, nous construirons un ***affichage de la température en utilisant arduino et un module LCD 16 x 2*** qui surveille en permanence la température autour du champ / plage de mesure du LM35 et l'affiche sur le module LCD. Alors passons à la construction du projet!

Le LM35 est un capteur de température analogique linéaire dont la tension de sortie varie linéairement avec le changement de température. Le LM35 est un capteur de température linéaire à trois bornes de National Semiconductor. Il peut mesurer une température de ***-55 degrés Celsius à +150 degrés Celsius***. La tension de sortie du LM35 augmente de 10 mV par degré de hausse de température. Le LM35 peut être utilisé avec une alimentation de 5 V et le courant de veille est inférieur à 60uA. La broche en LM35 est illustrée dans la figure ci-dessous.

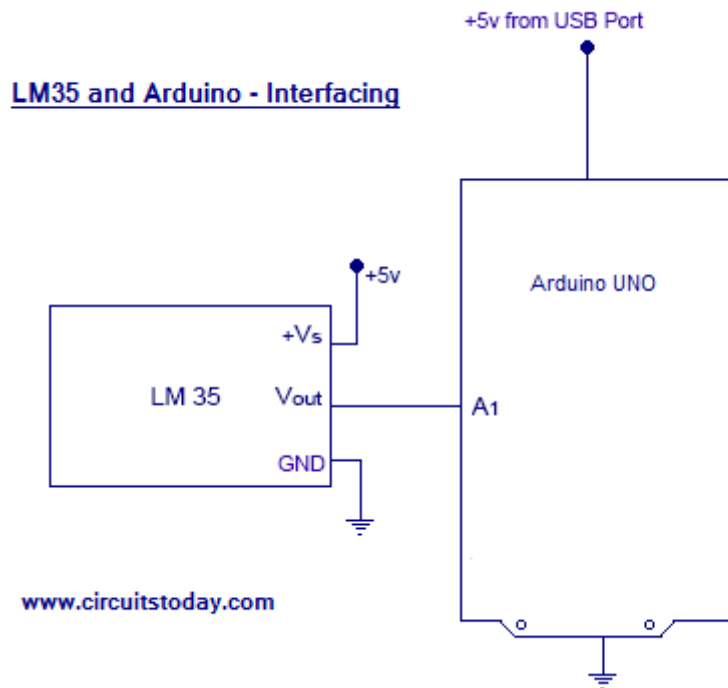
Voilà donc toutes les informations dont vous avez besoin sur le LM35 pour ce projet d'affichage de température particulier utilisant arduino uno. Passons donc au capteur de température LM35 en interface avec Arduino!

Nous utilisons Arduino Uno comme carte et le LM35 peut être connecté à arduino comme indiqué dans le schéma de circuit.

Remarque: - Le LM35 est un capteur de température analogique. Cela signifie que la sortie du LM35 est un signal analogique. Les microcontrôleurs n'acceptent pas directement les signaux analogiques comme entrée. Nous devons convertir ce signal de sortie analogique en signal numérique avant de pouvoir l'envoyer à l'entrée d'un microcontrôleur. Pour ce faire, nous pouvons utiliser un convertisseur analogique-numérique (ADC). Si nous utilisons un microcontrôleur de base tel que 8051, nous devons utiliser un convertisseur analogique-numérique externe pour convertir la sortie analogique du format LM35 en sortie numérique. Nous alimentons ensuite la sortie de l'ADC (valeur numérique convertie) en entrée de 8051. Cependant, les cartes modernes comme Arduino et la plupart des micro-contrôleurs modernes sont livrées avec un ADC intégré. Notre Arduino Uno a un ADC 10 bits intégré (6 canaux). Nous

pouvons utiliser cet ADC intégré d'arduino pour convertir la sortie analogique du LM35 en sortie numérique. Comme Arduino uno possède un CAN incorporé à 6 canaux, il existe 6 broches d'entrée analogique numérotées de A0 à A5. Connectez la sortie analogique du LM35 à l'une de ces broches d'entrée analogique d'arduino.

LM35 et Arduino - Schéma de circuit



Connectez le LM35 à Arduino uno comme indiqué dans le schéma de circuit. Le + 5v pour le LM35 peut être extrait de la broche + 5v de arduino uno. La broche de terre du LM35 peut également être connectée à la broche GND de l'arduino uno. Connectez Vout (la sortie analogique du LM35) à n'importe quelle broche d'entrée analogique de arduino uno. Dans ce schéma de circuit, nous avons connecté Vout de LM35 à A1 de Arduino.

Remarque: - Le LM35 est disponible sur le marché en 3 variantes de série: les séries LM35A, LM35C et LM35D. La principale différence entre ces 3 versions du LM35 IC réside dans leur plage de mesures de température. La série LM35D est conçue pour mesurer de 0 degré Celsius à 100 degrés Celsius, tandis que la série LM35A est conçue pour mesurer une plage plus large de -55 degrés Celsius à 155 degrés

Celsius. La série LM35C est conçue pour mesurer de -40 degrés Celsius à 110 degrés Celsius.

Dans notre exemple arduino LM35, nous utilisons le capteur LM35Dz, qui relève de la série LM35D. Ainsi, notre plage de mesure de température min-max va de 0 degré Celsius à 100 degrés Celsius.

Le programme - Interfaçage LM35 et Arduino

```
const int capteur = A1; // Affectation de la broche analogique A1 à la variable 'capteur'

float tempc; // variable pour stocker la température en degrés Celsius

float tempf; // variable pour enregistrer la température en degrés Fahrenheit

float vout; // variable temporaire pour conserver la lecture du capteur

void setup()

{

  pinMode (capteur, INPUT); // Configuration de la broche A1 en entrée

  Serial.begin (9600);

}

void loop ()

{
```

```
vout = analogRead (capteur);

vout = (vout * 500) / 1023;

tempc = vout;  // Mémorisation de la valeur en degrés Celsius

tempf = (vout * 1.8) + 32;  // Conversion en Fahrenheit

Serial.print ("in DegreeC =");

Serial.print ("\ t");

Serial.print (tempc);

Serial.println ();

Serial.print ("in Fahrenheit =");

Serial.print ("\ t");

Serial.print (tempf);

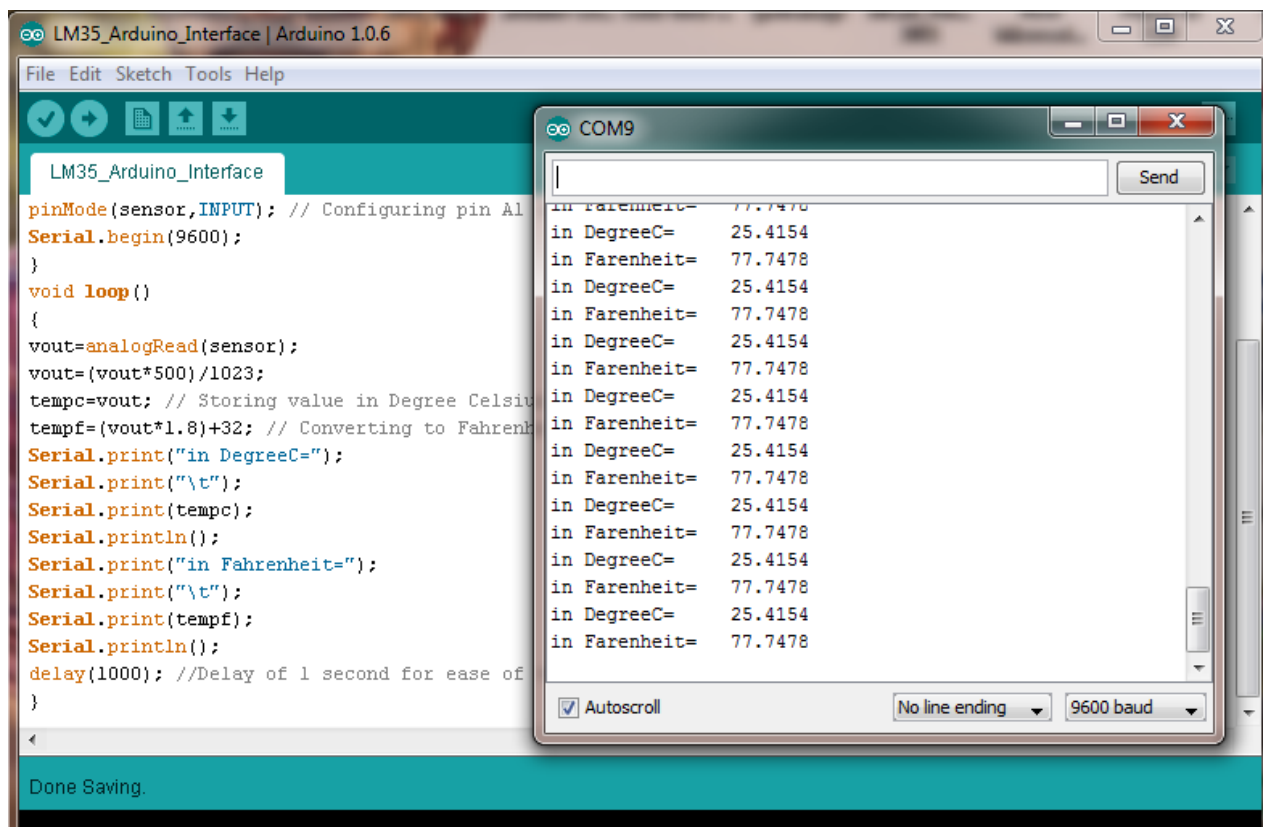
Serial.println ();

délai (1000);  // Retard d'une seconde pour faciliter la visualisation

}
```

Voilà donc le code arduino lm35 pour la lecture de la température et l’affichage en degrés Celsius et Fahrenheit. Le programme est explicite.

Capture d'écran de sortie du moniteur série



D'accord! Nous avons appris que le capteur de température LM35 s'interfaçait avec Arduino. Si vous avez des doutes concernant cette partie, demandez dans la section commentaires.

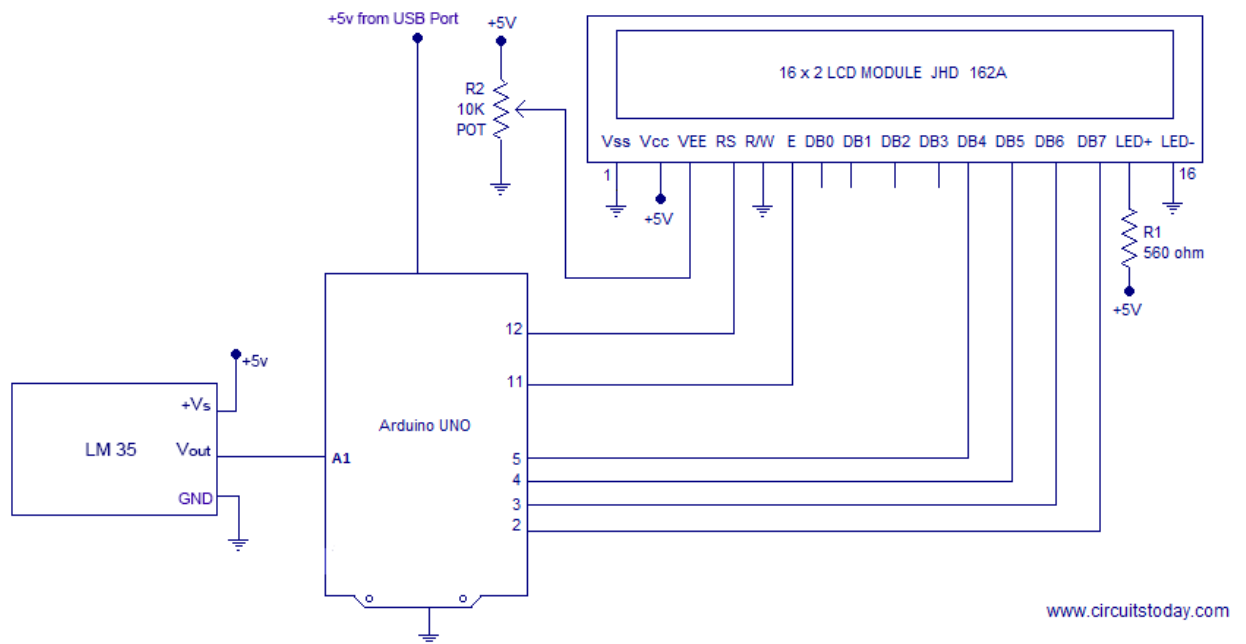
Affichage de la température sur un module LCD 16 × 2 - utilisant Arduino et LM35

Ajoutons maintenant un écran LCD 16 × 2 avec les interfaces LM35 et Arduino et affichons les valeurs de température sur cet écran LCD (au lieu d'un moniteur série). Nous allons donc construire nul autre qu'un affichage de température autonome utilisant arduino.

Remarque: - Si vous êtes débutant, lisez notre tutoriel sur le [module d'interface LCD pour Arduino](#) avant d'essayer ce projet de LM35 et d'Arduino avec un écran LCD 16 × 2.

Schéma de circuit - LM35 et Arduino - Affichage de la température sur un écran LCD 16 × 2

LM35 and Arduino - Temperature Display on 16x2 LCD Module



D'accord! Voilà donc le schéma de circuit pour construire un capteur de température Arduino LM35 avec écran LCD. Le programme Arduino pour le circuit est donné ci-dessous.

Le programme

```
#include <LiquidCrystal.h>

Liquid Crystal Crystal (12, 11, 5, 4, 3, 2);

const int capteur = A1; // Affectation de la broche analogique A1 à la variable 'capteur'

float tempc; // variable pour stocker la température en degrés Celsius

float tempf; // variable pour enregistrer la température en degrés Fahrenheit
```

```
    flottant vout; // variable temporaire pour conserver la lecture du capteur

void setup()

{

    pinMode (capteur, INPUT); // Configuration de la broche A1 en entrée

    Serial.begin (9600);

    lcd.begin (16,2);

    retard (500);

}

boucle vide ()

{

    vout = analogRead (capteur);

    vout = (vout * 500) / 1023;

    tempc = vout; // Mémorisation de la valeur en degrés Celsius

    tempf = (vout * 1.8) +32; // Conversion en Fahrenheit

    lcd.setCursor (0,0);
```

```
lcd.print ("en degré C =");

lcd.print (tempc);

lcd.setCursor (0,1);

lcd.print ("in Fahrenheit =");

lcd.print (tempf);

délai (1000); // Retard d'une seconde pour faciliter la visualisation sur
le moniteur série

}
```

D'accord! Le programme est très simple et explicite, si vous avez une idée de base des esquisses Arduino et que vous avez déjà appris comment interfacer le module [Arduino et le module LCD](#) .

Certaines personnes aimeraient utiliser l'affichage à 7 segments pour afficher la température mesurée à l'aide de LM35 et d'Arduino. Notre section suivante explique donc comment construire un système utilisant LM35 et Arduino pour afficher la température sur un affichage à 7 segments.

Tout d'abord, découvrez notre didacticiel sur l' [interfaçage de l'Arduino et de l'afficheur à 7 segments](#) - qui explique clairement l'interfaçage de différents types d'affichages à sept segments sur Arduino et comment y afficher différents chiffres / caractères numériques. Pour afficher la température jusqu'à 3 chiffres (100 degrés Celsius ou plus jusqu'à 999) avec l'unité correspondante (Celsius de Fahrenheit), il est bon de choisir une unité d'affichage à 4 chiffres à 7 chiffres.