

Mesure de Température avec LM35 et ESP32

Houssem-eddine LAHMER

4 mai 2025

Plan du Cours

- 1 Introduction
- 2 Le capteur LM35
- 3 Le microcontrôleur ESP32
- 4 Connexion LM35-ESP32
- 5 Connexion LM35-ESP32
- 6 Programmation
- 7 Simulation avec Wokwi
- 8 Applications Pratiques
- 9 Conclusion

Introduction

- Le LM35 est un capteur de température de précision dont la tension de sortie est linéairement proportionnelle à la température en degrés Celsius
- L'ESP32 est un microcontrôleur avec Wi-Fi et Bluetooth intégrés
- Wokwi est un simulateur en ligne pour tester des projets électroniques

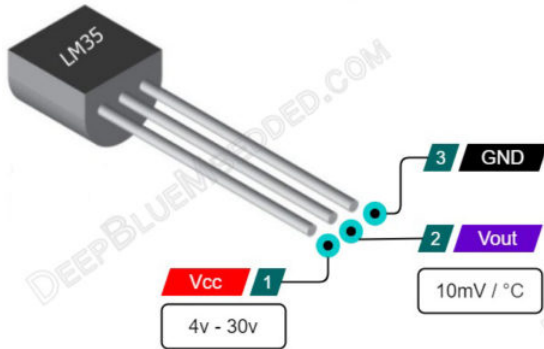
Objectifs du cours

- Comprendre le fonctionnement du capteur LM35
- Connecter le LM35 à l'ESP32
- Programmer l'ESP32 pour lire et interpréter les données du capteur
- Simuler le circuit dans Wokwi

Brochage du LM35

LM35 Temperature Sensor

Pinout



Brochage du LM35

Broche	Fonction
1	+Vs (Alimentation positive)
2	Sortie (10mV/°C)
3	GND (Masse)

Formule de conversion

$$T(^{\circ}C) = \frac{V_{out}}{10mV/^{\circ}C} \quad (1)$$

Avec V_{out} la tension mesurée en millivolts.

ESP32 - Caractéristiques

- Microcontrôleur dual-core 32-bit avec Wi-Fi et Bluetooth
- Fréquence CPU : jusqu'à 240 MHz
- Mémoire : 520 KB SRAM
- Convertisseur Analogique-Numérique (ADC) 12 bits
- Nombreuses GPIO (General Purpose Input/Output)
- Programmable via Arduino IDE, ESP-IDF, MicroPython, etc.

Avantages pour notre projet

- ADC intégré pour lire la tension du LM35
- Connectivité sans fil pour transmettre les données
- Programmation simple via l'environnement Arduino

ESP32 - Broches ADC

- L'ESP32 dispose de deux ADC :
 - ADC1 : connecté aux GPIO 32-39
 - ADC2 : connecté aux GPIO 0, 2, 4, 12-15, 25-27
- Résolution configurable : 9-12 bits
- L'ADC de l'ESP32 n'est pas parfaitement linéaire
- Nécessite une calibration pour des mesures précises

Attention

ADC2 n'est pas disponible lorsque le Wi-Fi est actif. Utilisez ADC1 (broches 32-39) pour notre projet.

Schéma de Connexion

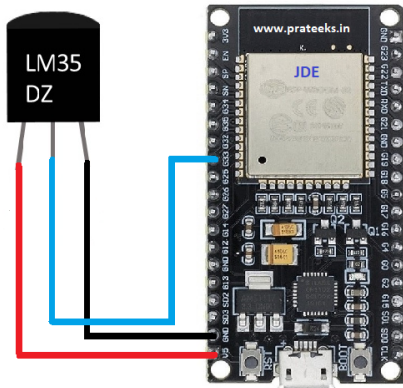


Schéma de Branchement LM35 - ESP32

LM35 Pin 1 (Vs) → ESP32 3.3V
LM35 Pin 2 (Output) → ESP32 GPIO 36 (ADC1_0)
LM35 Pin 3 (GND) → ESP32 GND

Matériel nécessaire

- 1 × ESP32 DevKit
- 1 × Capteur LM35
- Fils de connexion
- Breadboard (optionnel)

Considérations pour la Connexion

- Le LM35 fonctionne avec une alimentation de 4-30V, mais l'ESP32 fournit 3.3V
 - Le LM35 peut fonctionner à 3.3V avec une précision légèrement réduite
- L'ADC de l'ESP32 mesure des tensions de 0-3.3V (ou 0-1.1V selon la configuration)
- Conversion de tension en température :
 - 10mV/°C signifie que 25°C produiront 250mV
 - La résolution de l'ADC détermine la précision de mesure

Améliorations possibles

- Ajout d'un condensateur de 0.1μF entre la sortie et GND pour stabiliser le signal
- Utilisation d'un amplificateur opérationnel pour améliorer la précision

Code – Lecture basique (1/4)

```
1 // Programme basic pour lire la temp rature avec LM35
2 const int lm35Pin = 36; // GPIO36 (ADC1_0)
```

Code – Lecture basique (2/4)

```
1 void setup() {  
2     Serial.begin(115200);  
3     delay(1000);  
4     Serial.println("D marrage du programme LM35");  
5 }
```

Code – Lecture basique (3/4)

```
1 void loop() {  
2     // Lire la valeur analogique  
3     int adcValue = analogRead(lm35Pin);  
4  
5     // Convertir en voltage (mV)  
6     float voltage = adcValue * (3300.0 / 4095.0);  
7  
8     // Convertir en temperature ( C )  
9     float temperature = voltage / 10.0;  
10 }
```

Code – Lecture basique (4/4)

```
1  Serial.print("ADC: ");
2  Serial.print(adcValue);
3  Serial.print(" | Tension: ");
4  Serial.print(voltage);
5  Serial.print(" mV | Temp rature: ");
6  Serial.print(temperature);
7  Serial.println("  C ");
8
9  delay(1000);
10 }
```

Code Arduino – Version améliorée (1/5)

```
1 // Programme amélioré avec moyenne des mesures
2 const int lm35Pin = 36;          // GPIO36 (ADC1_0)
3 const int numReadings = 10;     // nombre de mesures pour la moyenne
```


Code Arduino – Version améliorée (2/5)

```
1 void setup() {  
2   Serial.begin(115200);  
3   delay(1000);  
4   Serial.println("D marrage du programme LM35 am lior ");  
5  
6   // Configuration de l'ADC  
7   analogReadResolution(12);           // 12 bits (0 4095 )  
8   analogSetAttenuation(ADC_11db); // Plage de tension: 0 3 .3 V  
9 }
```

Code Arduino – Version améliorée (3/5)

```
1 void loop() {  
2     // Prendre plusieurs mesures et calculer la somme  
3     int adcSum = 0;  
4     for (int i = 0; i < numReadings; i++) {  
5         adcSum += analogRead(lm35Pin);  
6         delay(10);  
7     }
```

Code Arduino – Version améliorée (4/5)

```
1 // Moyenne des lectures
2 int adcAvg = adcSum / numReadings;
3
4 // Conversion en tension (mV)
5 float voltage = adcAvg * (3300.0 / 4095.0);
6
7 // Conversion en température ( C )
8 float temperature = voltage / 10.0;
```

Code Arduino – Version améliorée (5/5)

```
1 Serial.printf(  
2     "ADC: %d | Tension: %.2f mV | Temp rature: %.2f  C \n",  
3     adcAvg, voltage, temperature  
4 );  
5  
6 delay(1000);  
7 }
```

Simulation avec Wokwi

- Wokwi est un simulateur en ligne pour l'électronique et les microcontrôleurs
- Permet de tester des projets sans matériel physique
- Supporte ESP32, Arduino, et de nombreux composants

Configuration de Wokwi pour notre projet

- 1 Créer un nouveau projet ESP32
- 2 Ajouter un composant LM35 depuis la bibliothèque
- 3 Connecter selon le schéma précédent
- 4 Copier le code Arduino et l'exécuter
- 5 Observer les résultats dans le moniteur série

Avantages

- Rapide à mettre en place
- Aucun risque d'endommager du matériel
- Facilite les tests et le débogage
- Accessible depuis n'importe quel navigateur

Inconvénients

- Ne reproduit pas parfaitement le comportement réel
- Ne simule pas les interférences, le bruit électronique
- Certains composants ou fonctionnalités avancées peuvent ne pas être disponibles

Applications Pratiques

- Station météo domestique
- Système de surveillance de température
- Contrôle de chauffage ou climatisation
- Incubateurs ou systèmes de culture hydroponique
- Moniteurs de température pour serveurs ou équipements électroniques

Extensions possibles du projet

- Ajout d'un écran OLED/LCD pour afficher la température
- Communication des données via Wi-Fi vers un serveur
- Mise en place d'alertes par email ou notification
- Stockage des données dans une base de données
- Interface web pour visualiser l'historique des températures

Conclusion

- Le LM35 est un capteur analogique simple et fiable pour mesurer la température
- L'ESP32 offre une plateforme puissante avec de nombreuses possibilités
- Wokwi permet de tester et valider notre projet avant de passer au matériel physique

Points clés à retenir

- Conversion : $10\text{mV}/^{\circ}\text{C}$ pour le LM35
- Utilisez l'ADC1 (GPIO 32-39) si vous utilisez le Wi-Fi
- Implémentez des moyennes de mesures pour améliorer la précision
- Calibrez l'ADC si une précision élevée est requise

Pour aller plus loin

Explorer d'autres capteurs (DHT22, BME280) qui offrent plus de mesures (humidité, pression) dans un seul composant.

- Fiche technique LM35 : <https://www.ti.com/lit/ds/symlink/lm35.pdf>
- Documentation ESP32 :
<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>
- Arduino IDE : <https://www.arduino.cc/en/software>
- Wokwi : <https://wokwi.com>
- ESP32 ADC : <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/adc.html>

Merci de votre attention !

Des questions ?