



# CAPTEUR DE TEMPÉRATURE

---

Projet Tutoré S2



# Cahier Des Charges

---

Mesure de la température ambiante avec une précision de  $\pm 0.5^{\circ}\text{C}$ .

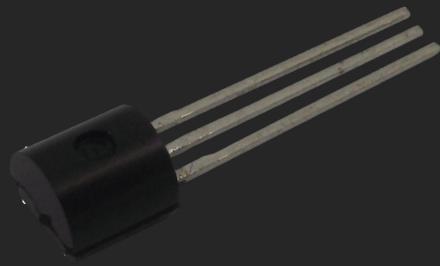
Affichage local de la température (écran LCD 2x16).

Actualisation de la mesure au moins toutes les secondes.

Alimentation par USB ou batterie rechargeable (faible consommation).

S'adapte à une carte mère « KIT JIPADA » et à un microcontrôleur de type **STM32 Nucléo F303K8**.

# Composants nécessaires :



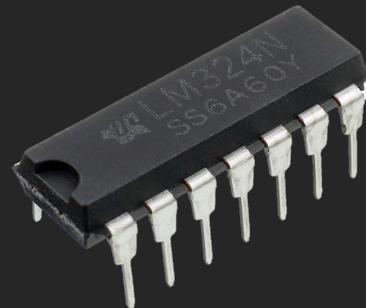
## LM335

Précision maximale (non calibré) : **±2°C (de -40°C à +100°C)**

Précision après étalonnage : **±0.5°C**

Coefficient de température : **10 mV/K**

$$V_{OUT} = T(K) \times 10 \text{ mV}$$

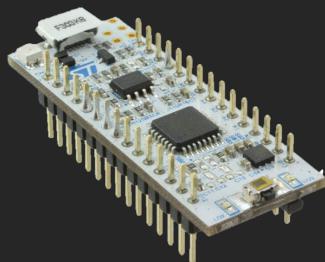


## LM324N

Embarque 4 AOPs

Pas de tension de déchet sur 0V, 1V sur VCC

Fonctionne entre 1 et 5V



## NUCLEO F303K8

Horloge cadencée à 72Mhz

Embarque 1 DAC et 2 ADC

GPIOs en 3,3V

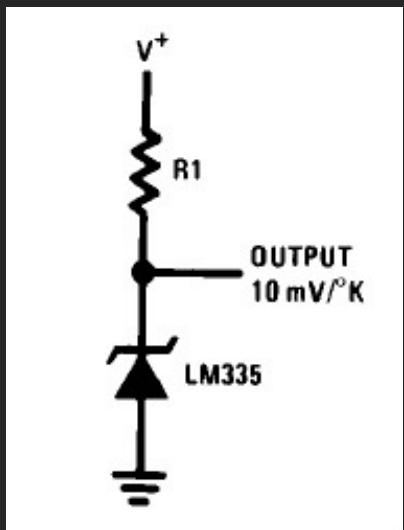
# Fonctionnement du capteur :

$$\Theta = F(V_{\text{capteur}})$$

$$0^{\circ}\text{C} < \Theta < 50^{\circ}\text{C}$$

0°C	273.15 K	2.7315 V
25°C	298.15 K	2.9815 V
50°C	323.15 K	3.2315 V
100°C	373.15 K	3.7315 V

$$\Leftrightarrow 2,73\text{V} < V_{\text{capt}} < 3,23\text{V}$$



D'après la doc. :  $400\mu\text{A} < I < 5\text{mA}$

On fait le choix de prendre  $R = 3,3\text{k}\Omega$

$$R_{min} = \frac{5-2,73}{5 \cdot 10^{-3}} = 454 \Omega$$

$$I_{max} = \frac{5-2,73}{3,3 \cdot 10^3} = 688\mu\text{A}$$

$$R_{max} = \frac{5-3,23}{400 \cdot 10^{-6}} = 4,42 \text{ K}\Omega$$

$$I_{min} = \frac{5-3,23}{3,3 \cdot 10^3} = 536\mu\text{A}$$

# Mise en équation du fonctionnement attendu :

On souhaite une température en °C :

$$V_{capt} = \Theta \text{ K} * 10 * 10^{-3}$$

$$V_{capt} = (\Theta^{\circ}\text{C} + 273,15) * 10 * 10^{-3}$$

On veut que la tension à 0° soit de 0V :

$$V_{capt} - 2,73$$

Changement de la plage de tension

$$\Delta V_{capt} = 3,23 - 2,73 = 0,5\text{V}$$

$$\Delta V_S = 3,3 - 0 = 3,3 \text{ V}$$

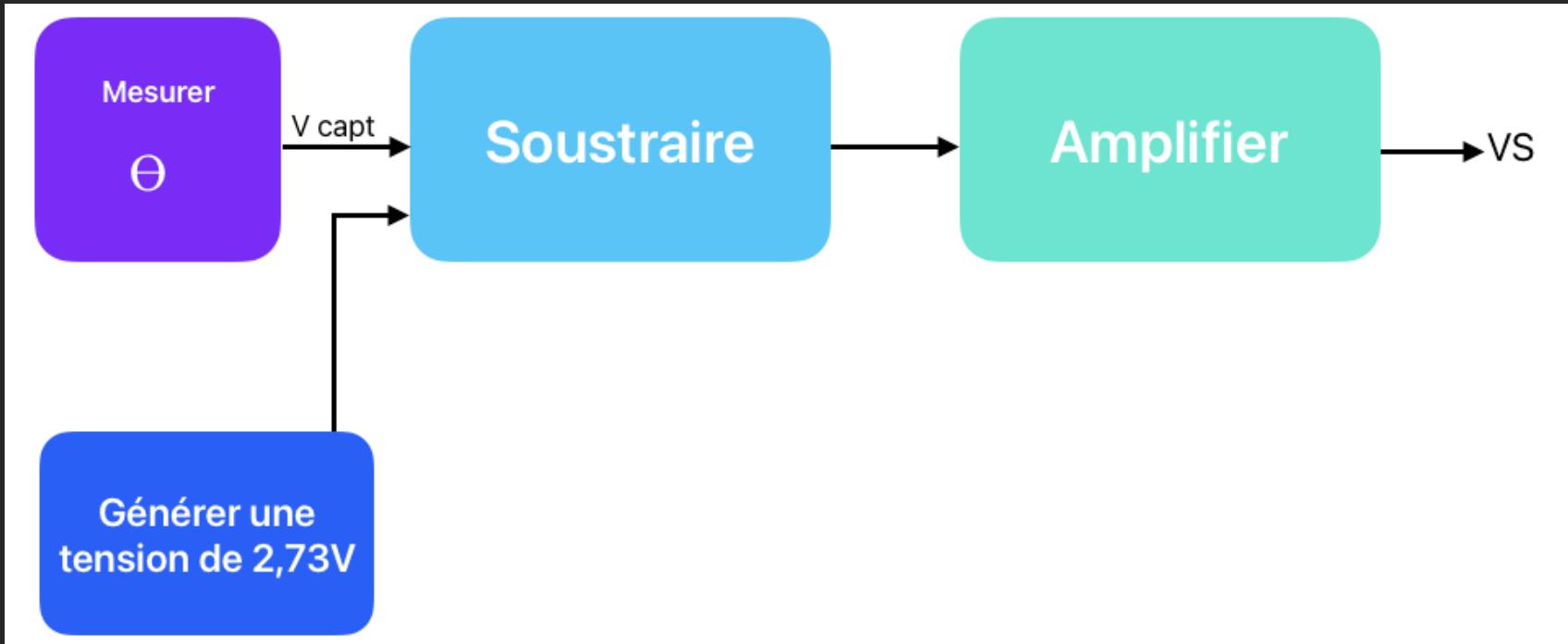
$$\text{Gain} = \frac{\Delta V_S}{\Delta V_{capt}} = \frac{3,3}{0,5} = 6,6$$

Mise à l'échelle :

$$V_S = f(V_{capt})$$

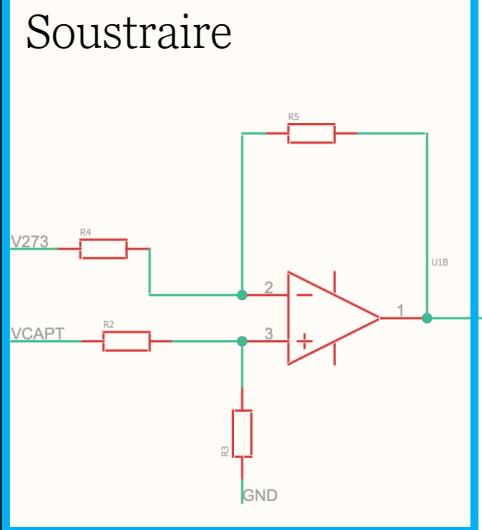
$$V_S = (V_{capt} - 2,73) * 6,6$$

# Schéma fonctionnel du système :

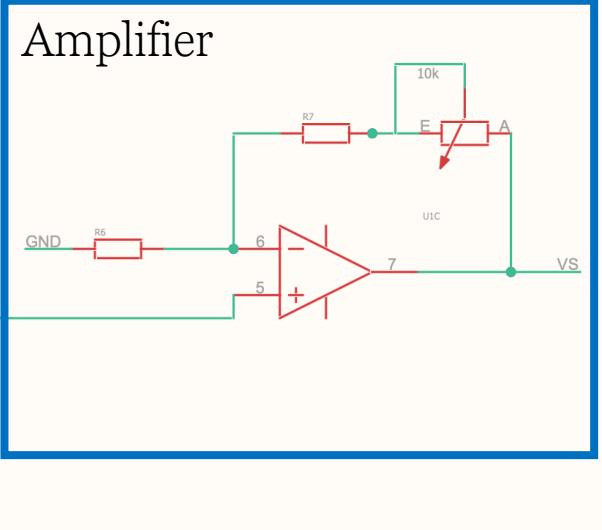


# Mise en œuvre électronique de VS = f(Vcapt):

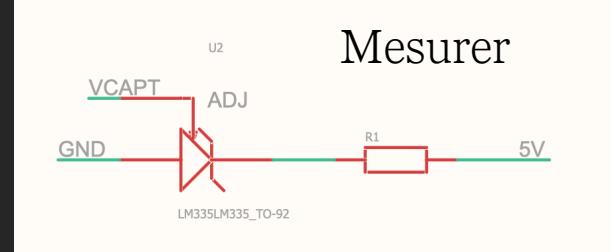
Soustraire



Amplifier



Mesurer



Montage soustracteur :

$$V2 = \frac{R}{R+R} * Vcapt * \left(1 + \frac{R}{R}\right) - \frac{R}{R} * V273$$

$$V2 = Vcapt - V273$$

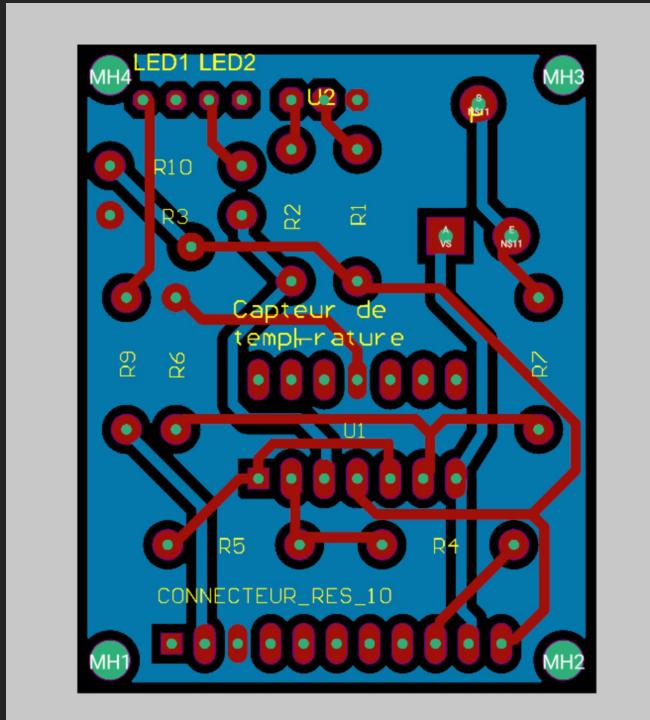
Montage AOP non inverseur (amplificateur) :

$$VS = 6,6 * V2 \Leftrightarrow \left(1 + \frac{R2}{R1}\right) * V2$$

$$\frac{R2}{R1} = 5,6 \text{ (doit être réglable pour calibrer le capteur)}$$

R1 fixé à 10kΩ R2 = R+P = 47kΩ + 10kΩ(potentiomètre)

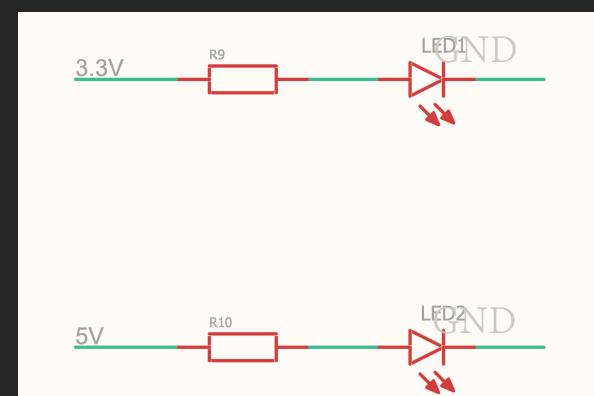
# Implantation et réalisation du système sur PCB:



Routage sur le Logiciel  
Autodesk Eagle



Ajout de 2 LED afin de contrôler la présence de tension 3,3V et 5V.



# Générer une tension de 2,73V (ajustable) :

```
HAL_ADCEx_Calibration_Start(&hadc1, ADC_SINGLE_ENDED);
HAL_DAC_Start(&hdac1, DAC_CHANNEL_1);
HAL_DAC_SetValue(&hdac1, DAC_CHANNEL_1, DAC_ALIGN_12B_R, 0);

HAL_ADC_Start(&hadc1);
```

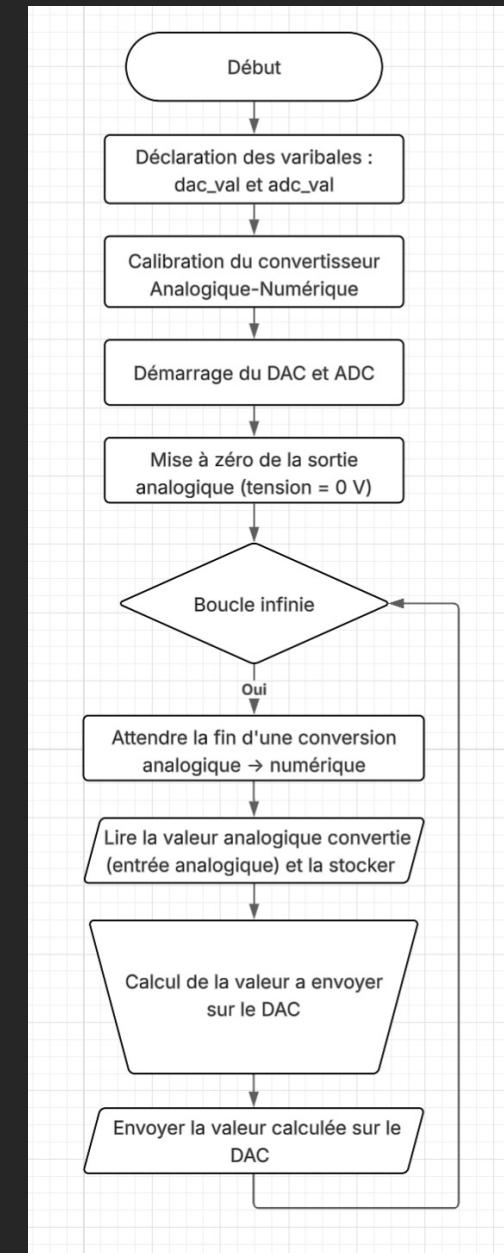
```
void generer_V273(void){

    HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
    adc_val = HAL_ADC_GetValue(&hadc1);

    dac_val = 3103 + (uint32_t)((adc_val / 4096.0f) * (0.5f / 3.3f) * 4095);

    HAL_DAC_SetValue(&hdac1, DAC_CHANNEL_1, DAC_ALIGN_12B_R, dac_val);

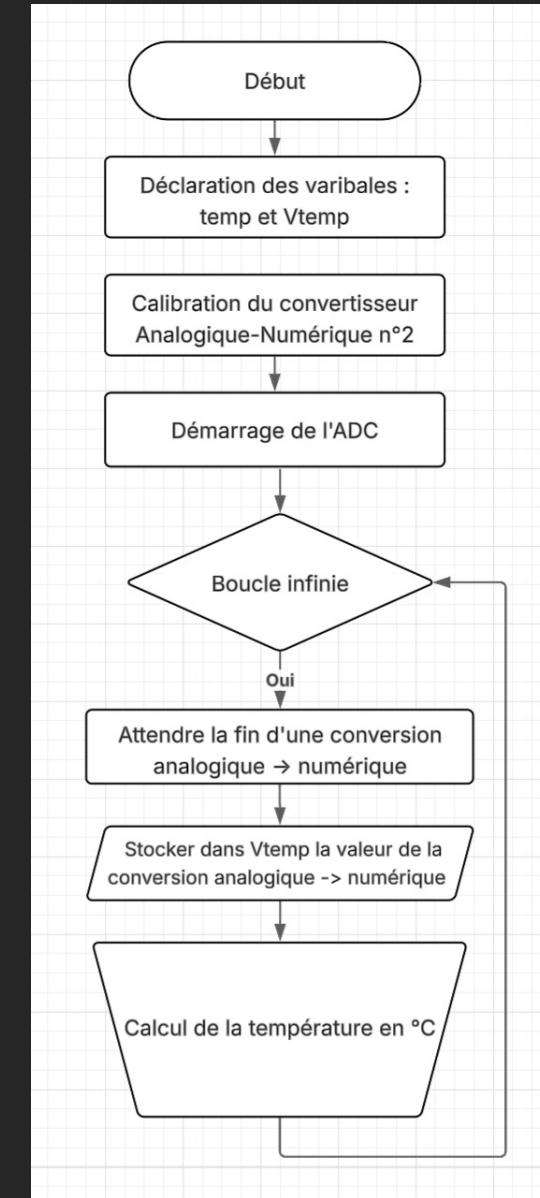
}
```



# Traiter et conditionner Vcapt :

```
HAL_ADCEx_Calibration_Start(&hadc2, ADC_SINGLE_ENDED);  
HAL_ADC_Start(&hadc2);
```

```
void recuperer_temperature(void){  
  
    HAL_ADC_PollForConversion(&hadc2, HAL_MAX_DELAY);  
    Vtemp = HAL_ADC_GetValue(&hadc2);  
  
    temp = Vtemp * (50.0f / 4096.0f);  
  
}
```



# Afficher la température toute les 100ms :

```
while (1)
{
    generer_V273();
    recuperer_temperature();
    lcd_affiche_temperature(temp);
    HAL_Delay(100);

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
```

Appel des fonctions et rafraîchissement de l'affichage périodiquement toutes les 100ms

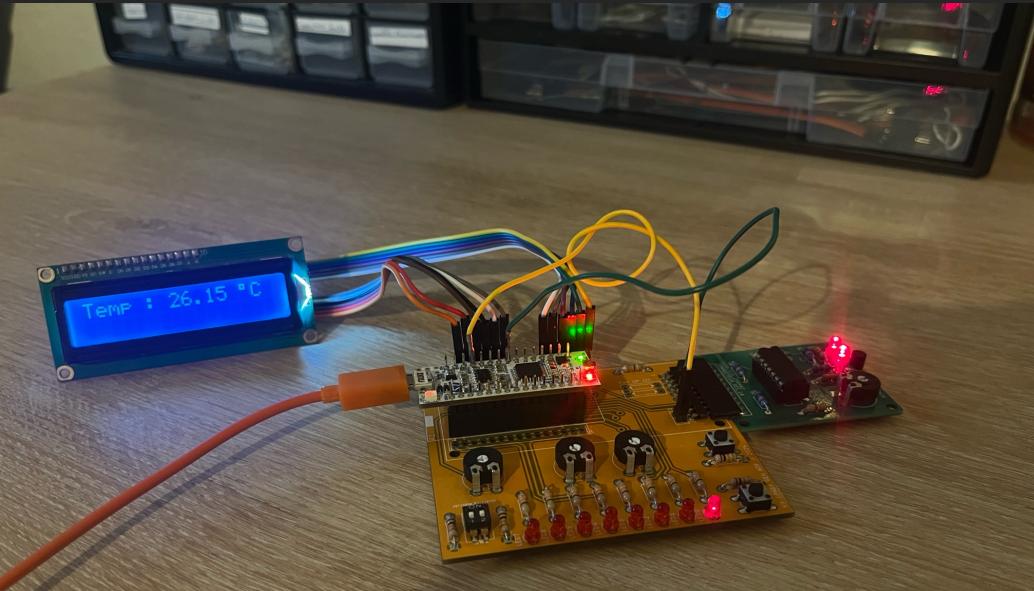
Dans le fichier LCD.c de la librairie LCD\_STM

```
void lcd_affiche_temperature(float temperature)
{
    char buffer[17]; // Max 16 caractères + caractère de fin de ligne

    // Formate le texte avec 2 décimales
    sprintf(buffer, sizeof(buffer), "Temp : %.2f \xDF""C", temperature);

    // Affiche à la ligne 1, position 0
    affiche_chaine(1, 0, buffer);
}
```

# Câblage et test du système :



- ➔ Alimenté sous 5V
- ➔ Ecran LCD commandé en 4bits
- ➔ Affichage de la température de façon lisible et actualisée toute les 100ms.
- ➔ Précision de la température de l'ordre de  $\pm 0.5^{\circ}\text{C}$  après calibration du gain.

