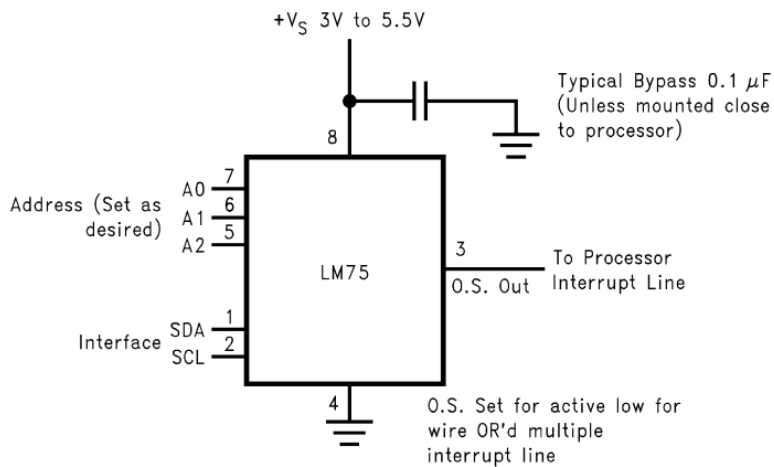


## Le schéma d'application typique

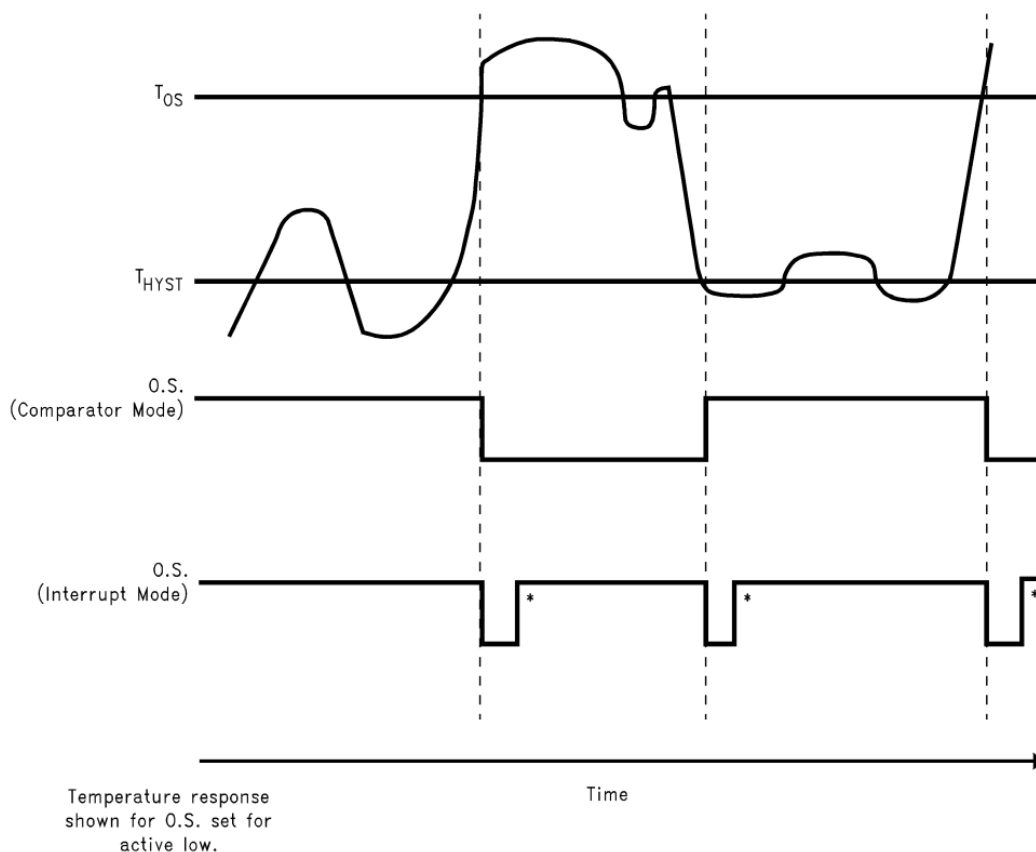


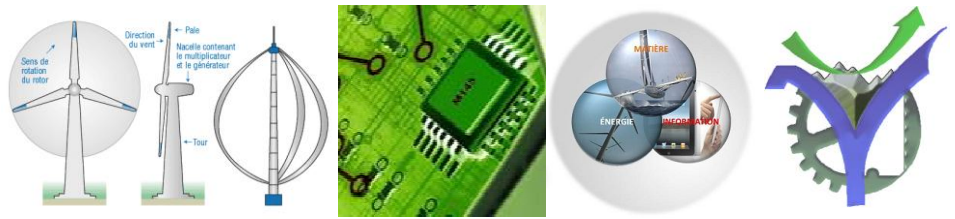
L'adresse :

1	0	0	1	A2	A1	A0
MSB			LSB			

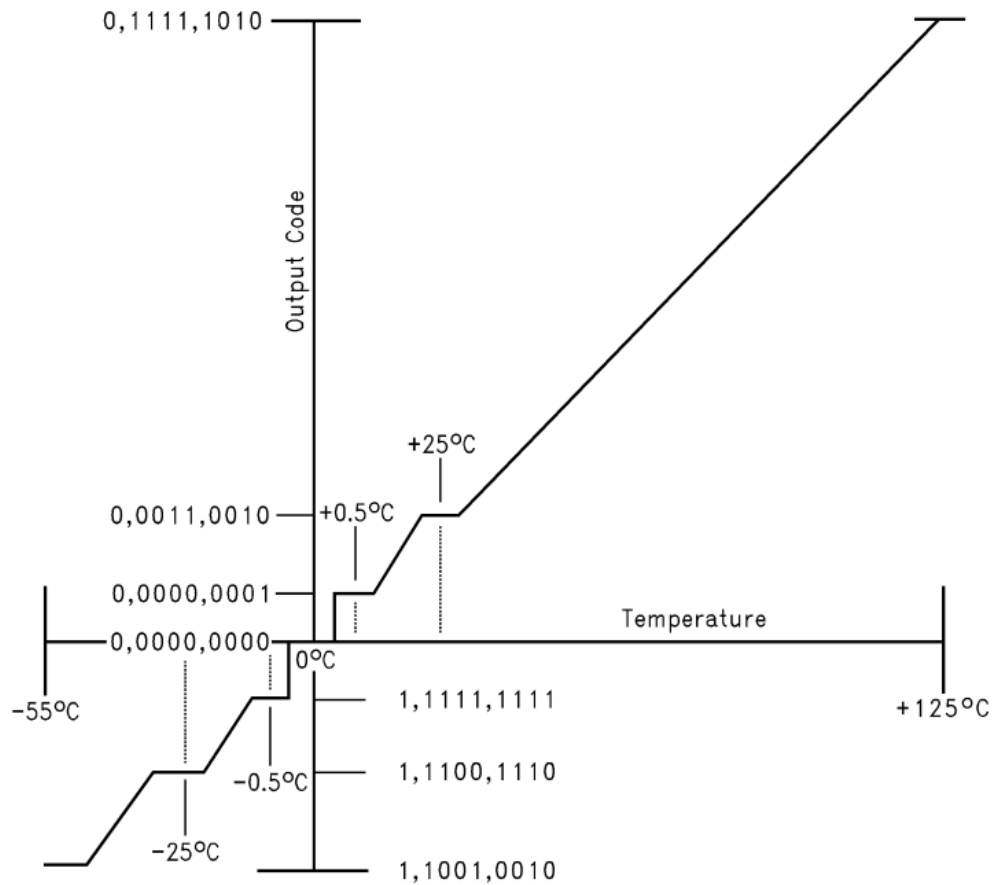
## Les deux modes de fonctionnement du capteur

Le LM75 permet deux modes de fonctionnement, le mode comparateur, et le mode Interrupt.





## La fonction de transfert



## Mise en œuvre du capteur

Le résultat de la mesure est donné sur deux octets :

P2–P7: Must be kept zero.

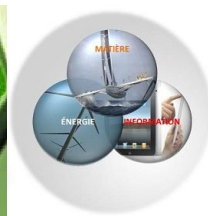
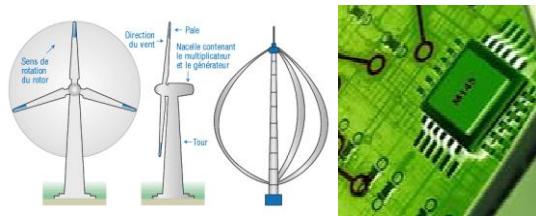
### 1.12 TEMPERATURE REGISTER

(Read Only):

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
MSB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	LSB	X	X	X	X	X	X	X

D0–D6: Undefined

D7–D15: Temperature Data. One LSB = 0.5°C. Two's complement format.



```
// Lecture de la température sur le capteur LM75 I2C Read mode complet
I2C_MasterReadBuf(Adresse_LM75, rbuffer,2, I2C_MODE_COMPLETE_XFER );
// wait until Transfer is complete
while((I2C_MasterStatus() & I2C_MSTAT_RD_CMPLT )==0);
```

Il faut retravailler le résultat obtenu pour pouvoir ensuite l'afficher :

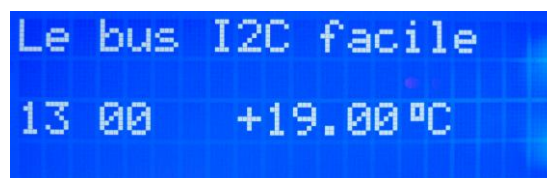
```
// Traitement de la donnée
// .Regroupement des deux octets dans un mot
// .Recadrage de 7 positions vers la droite ( voir documentation du LM75 )
// .Supression des bits non significatifs induits par le décalage
Temp = rbuffer[0]*256|rbuffer[1];
Temp = Temp >> 7;
Temp = Temp & 0x01FF;
```

Affichage de la valeur :

```
// Affichage de la valeur de la température
// .Positionnement du LCD en ligne 3 colonne 0
// .Préparation de la chaine de caractère float => Char
// .Affichage sur le LCD
// .Affichage du caractère utilisateur '\'' défini en Custom_0
CharLCD_Position(2,0);
CharLCD_PrintInt8(rbuffer[0]);CharLCD_PutChar(' ');
CharLCD_PrintInt8(rbuffer[1]);

CharLCD_Position(2,8);
sprintf(tstr, "%+5.2f", 0.5*Temp );
CharLCD_PrintString(tstr);
CharLCD_PutChar(CharLCD_CUSTOM_0);
CharLCD_PutChar('C');
```

Un exemple de résultat :



☞ réaliser la mise en œuvre du capteur LM75.