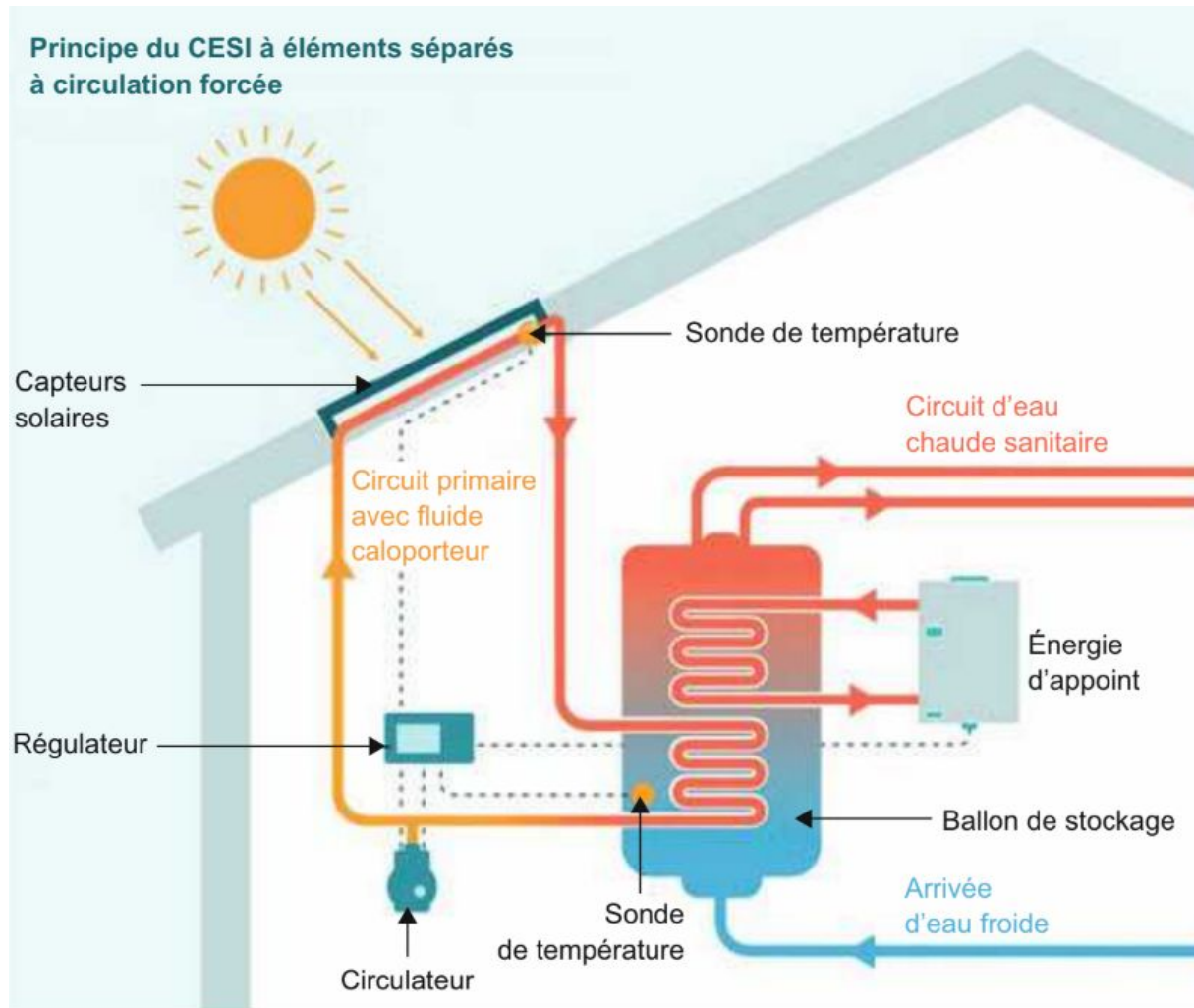


# De l'autoconsommation à l'autoproduction : optimisation des installations solaires thermiques



maquette chauffe-eau solaire individuel (CESI)

# La production d'eau chaude par chauffe-eau solaire individuel



# Comment optimiser et rendre plus accessible une installation solaire thermique ?

## Objectif :

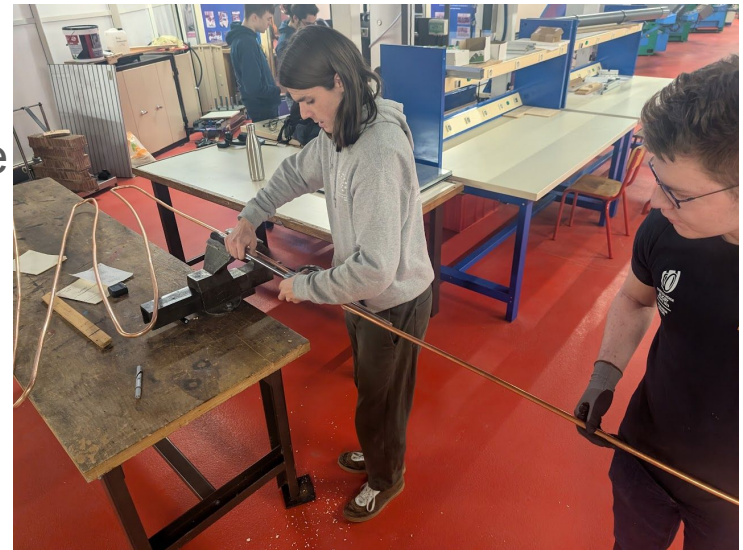
- modélisation/mesure puissance solaire.
- facteurs impactant le débit optimal.
- gain d'une pompe à débit variable.

## Plan :

1. Présentation de la démarche scientifique
2. Présentation de la maquette
3. Protocole de mesure, Expériences
4. Résultats
5. Conclusion

## Objectif binôme :

- impact de l'isolation sur le capteur solaire
- modélisation thermodynamique du capteur solaire



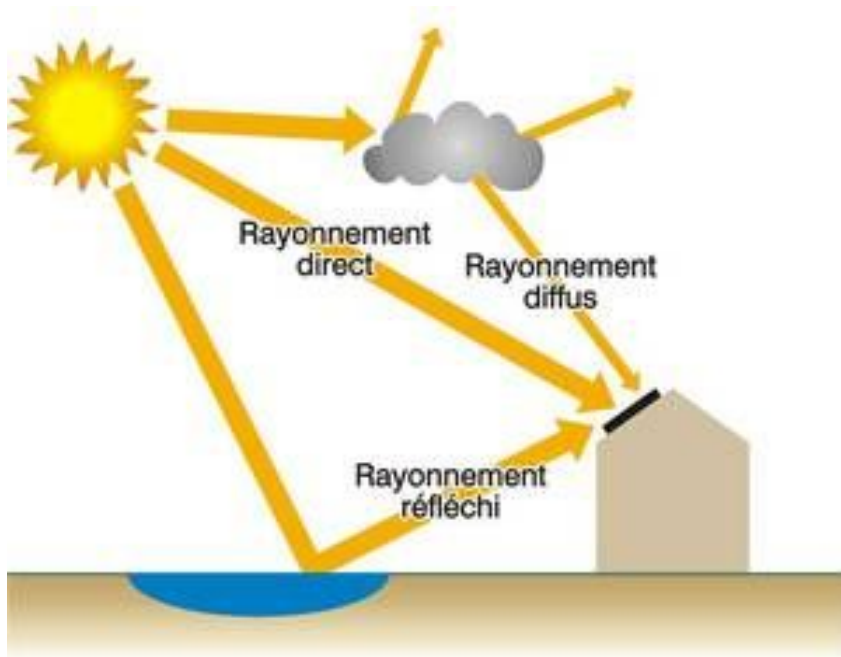
Fabrication du capteur solaire  
de mon binôme



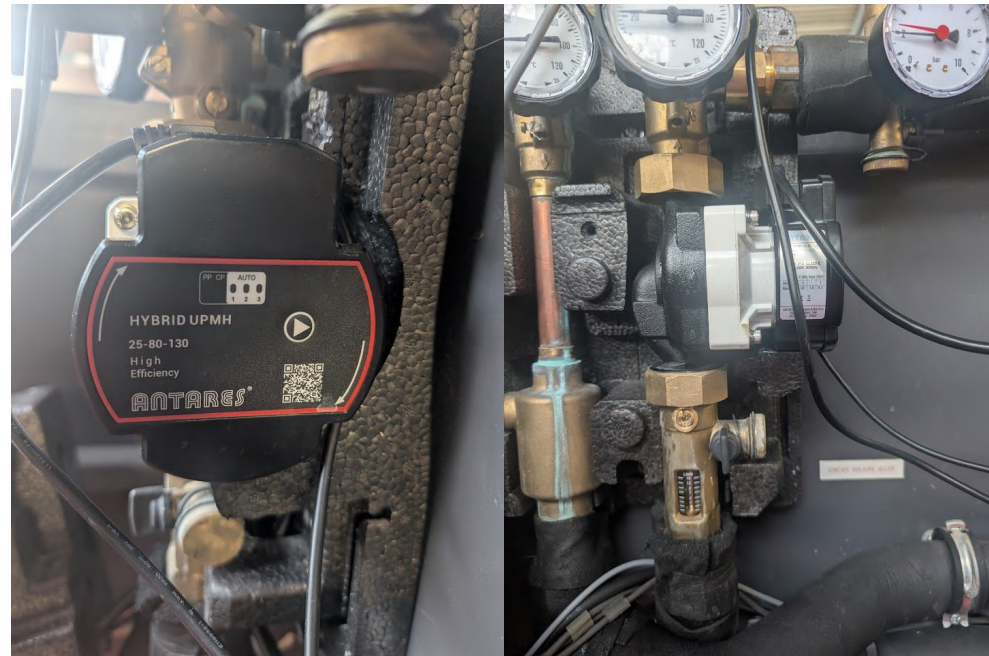
# 1. Présentation de la démarche scientifique

Objectif : obtenir la puissance produite par le panneau en fonction de :

- La puissance solaire reçue
- le débit de fluide



Ensoleillement, [Energie plus]



circulateur Antares UPM 25-80-130  
monté sur notre installation solaire

## 2. Présentation de la maquette



Maquette didactique de CESI

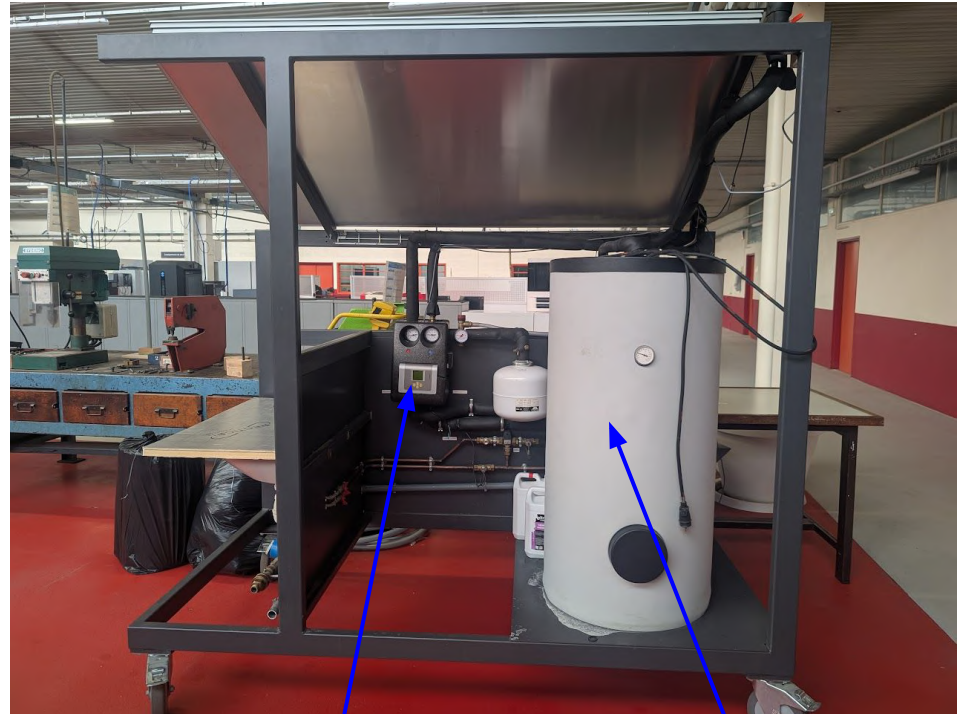


**Prototype → peu de ressource sur la maquette**

## 2- Présentation de la maquette



capteur solaire à plan vitré  
(objet du TIPE de mon  
binôme)



stockage eau chaude  
partie commande de l'installation (détaillée  
slide suivante)

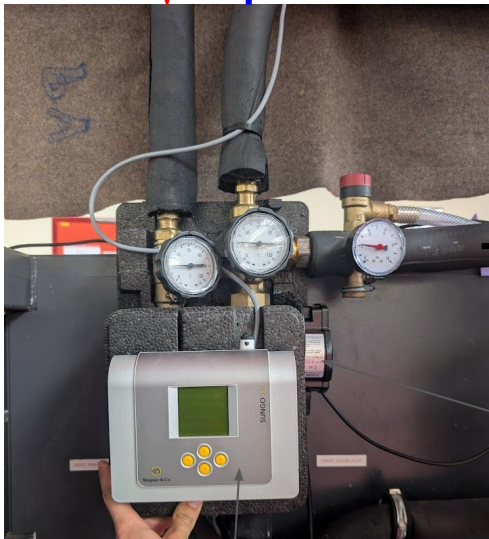


## 2- Présentation de la maquette

Partie commande et acquisition température

retour  
panneau

vers  
panneau



vers vase  
d'expansion

Circulateur WILO ST-25

régulation solaire  
SUNGO SXL

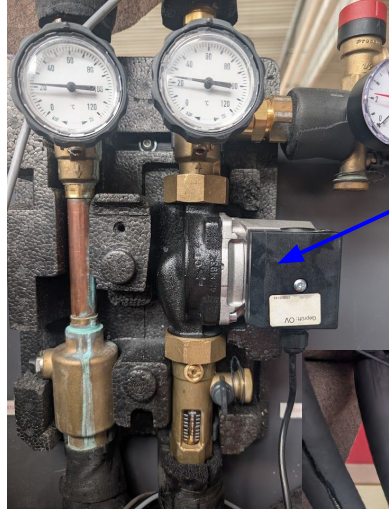
Problème :

- impossibilité de dialoguer avec la régulation d'origine
- Circulateur à débit constant



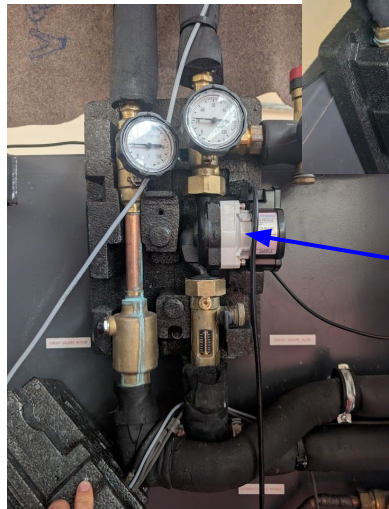
## 2- Présentation de la maquette

Partie commande et acquisition température



Circulateur WILO

Débit constant (0,2 kg/s)



Circulateur Antares

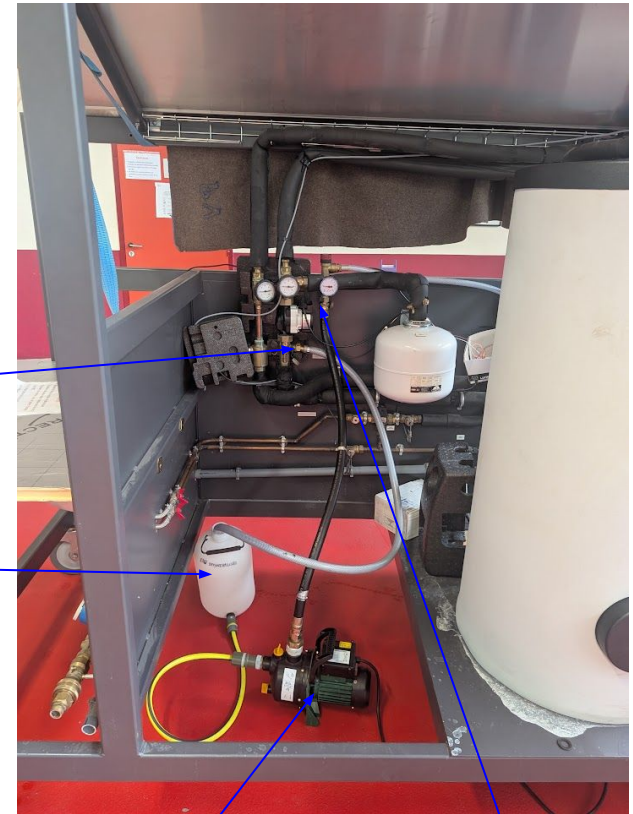
Débit variable (de 0 à 0,2 kg/s)



retour glycol

bac tampon

Réamorçage circuit glycole



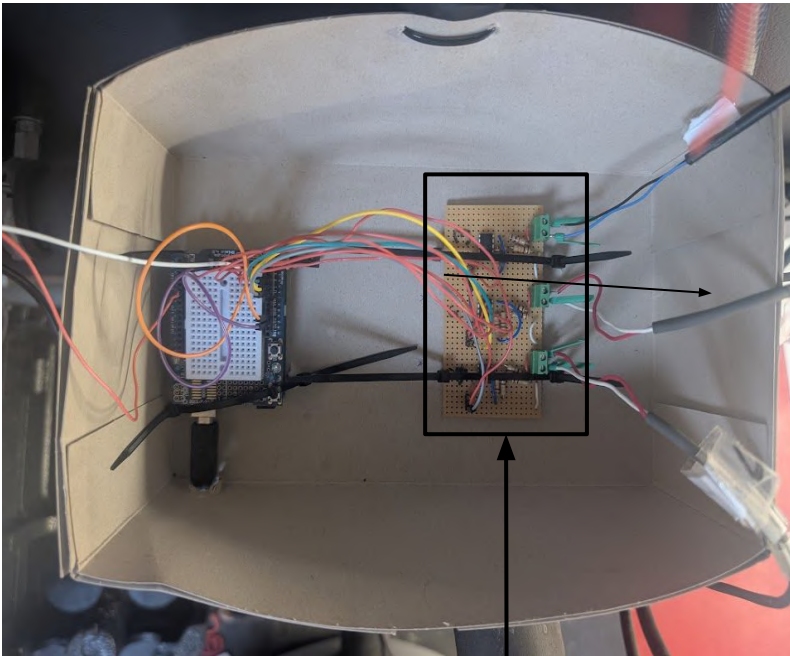
envoi glycol

pompe de circulation



## 2- Présentation de la maquette

Partie commande et acquisition température

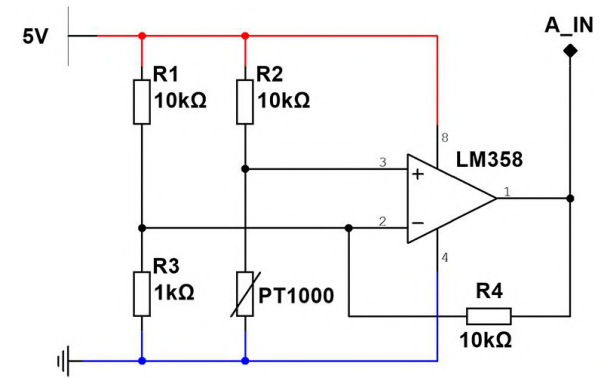


carte arduino

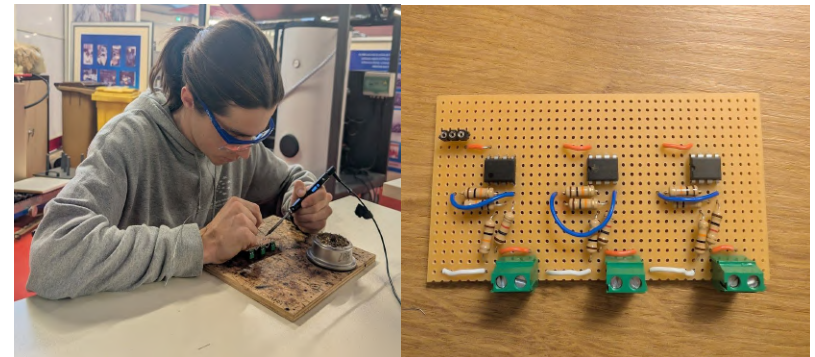
circuit d'acquisition  
sondes



Sondes  
PT1000



[AEQ-WEB]



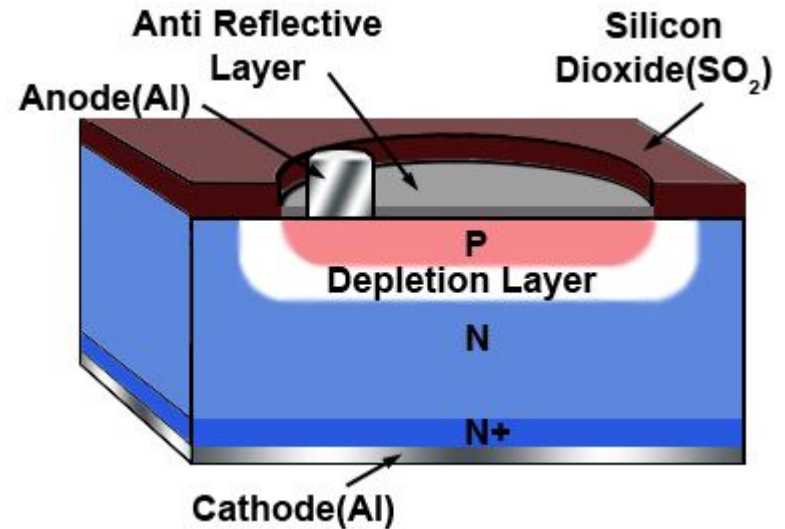
Fabrication des circuits d'acquisition

## 2- Présentation de la maquette

Acquisition puissance solaire reçue



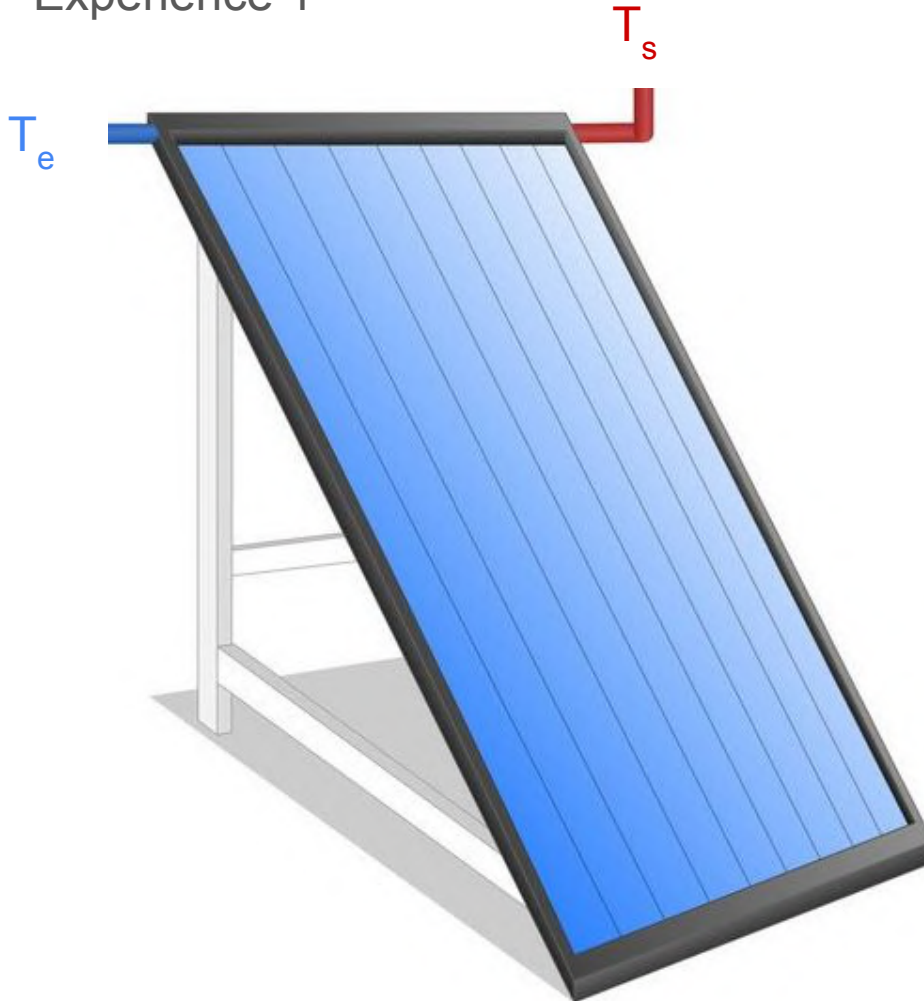
capteur RS pro IM-750  
(pyranomètre)



[[learnabout-electronics.org](http://learnabout-electronics.org)]

### 3- Expériences, mesures

#### Expérience 1



mesure des températures de sortie  
et d'entrée du panneau



Par application du premier principe  
industriel au fluide circulant entre l'entrée  
et la sortie du panneau, on obtient :

$$P_{th} = D_m c (T_s - T_e)$$

avec:

$P_{th}$  la puissance thermique  
récupérée par le panneau

$D_m$  le débit massique du fluide

$c$  la capacité thermique  
massique du fluide



# 3-Expériences, mesures

## Expérience 1

Choix orientation panneau :  
permet de “choisir” la  
puissance d'entrée

Pyranomètre orienté  
parallèlement au  
panneau solaire



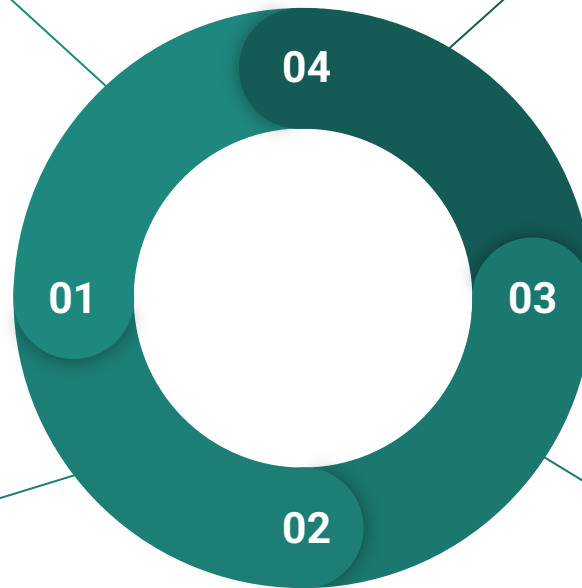
# 3-Expériences, mesures

## Expérience 1

### Orientation du panneau



Choix du débit de circulation.



Traitement de chaque mesure afin de tracer l'évolution de la puissance produite en fonction du débit de la pompe

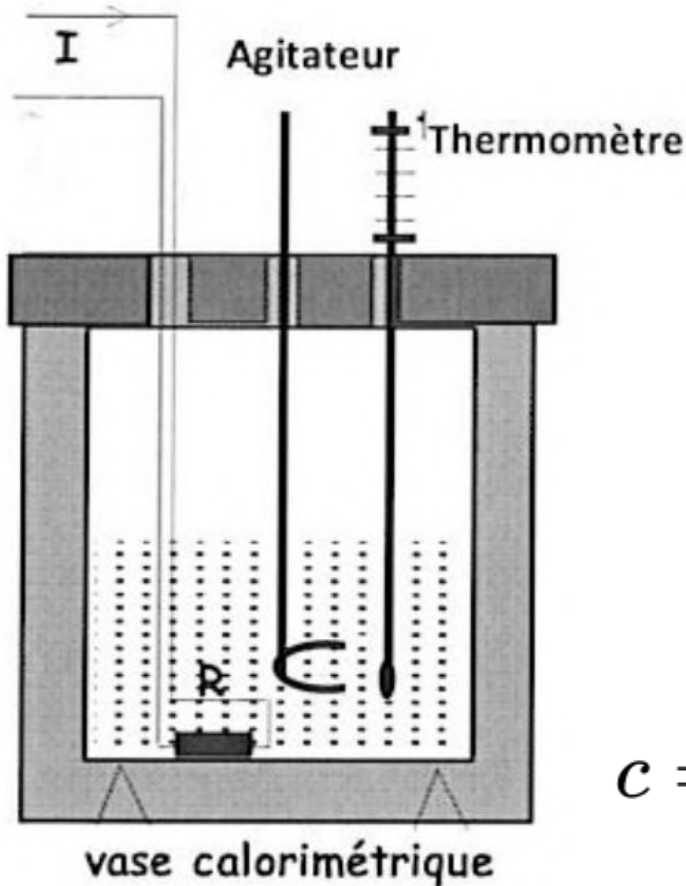


Relevé de la température du glycol à l'entrée et à la sortie du panneau toutes les 1/2 seconde pendant 2 minute



### 3- Expériences, mesures

#### Expérience 2



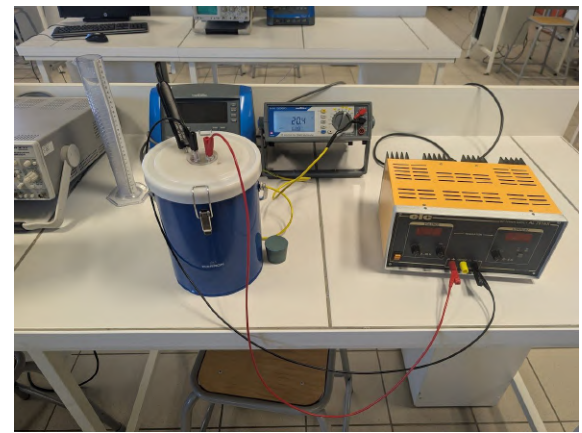
objectif :

- déterminer la capacité thermique massique du fluide caloporteur

outil :

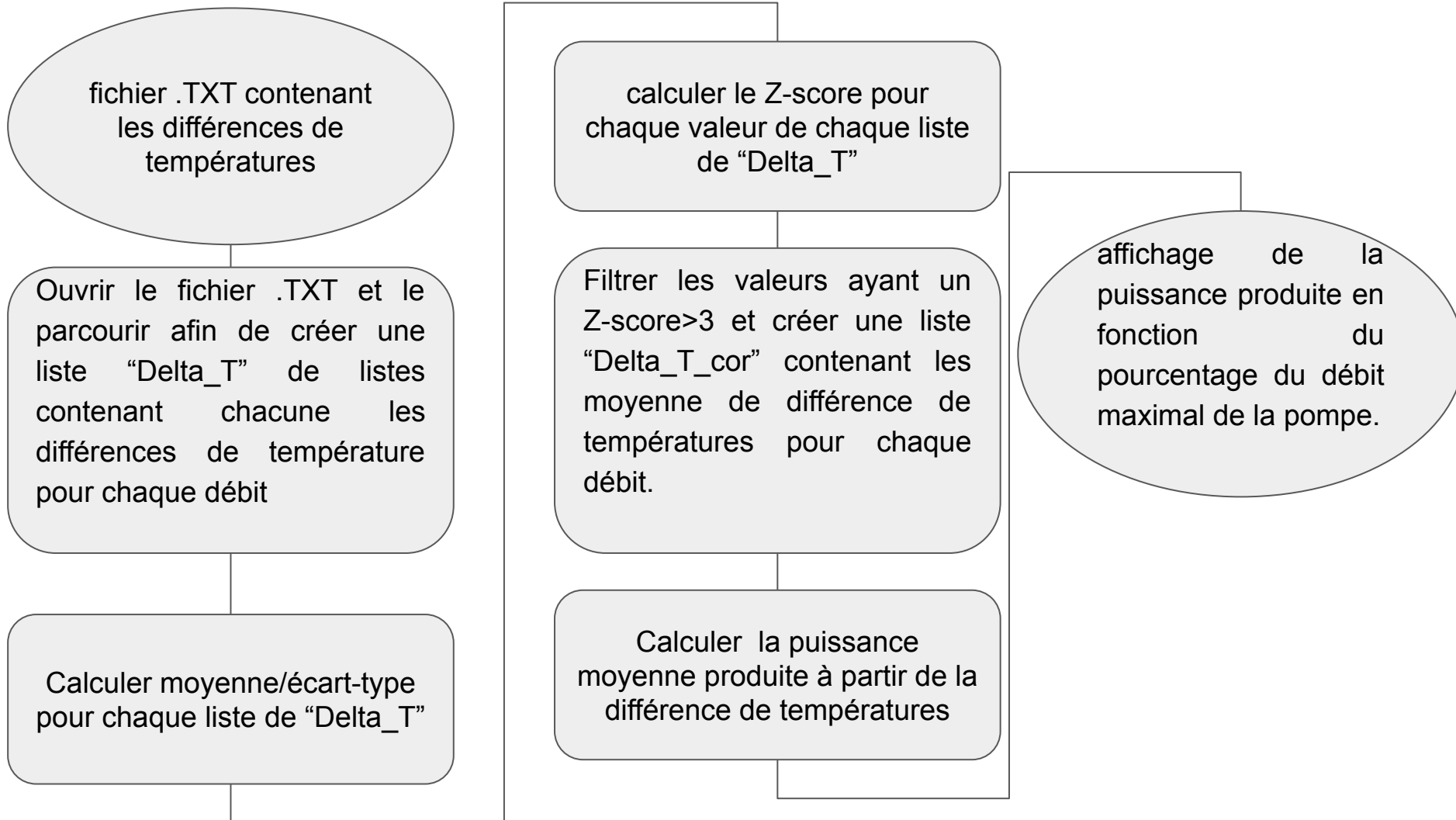
- application du premier principe enthalpique au vase et son contenant entre les instants  $t_f$  et  $t_i$

$$c = \frac{UI(t_f - t_i)}{m(\theta_f - \theta_i)}$$
$$U = 10V$$
$$I = 2A$$
$$m = 250g$$
$$c = 3,6 \text{ kJkg}^{-1}\text{K}^{-1}$$





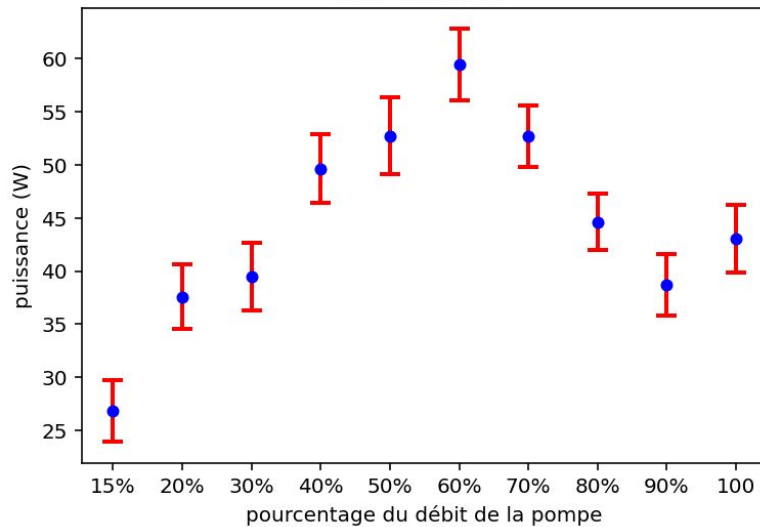
## 4-résultats



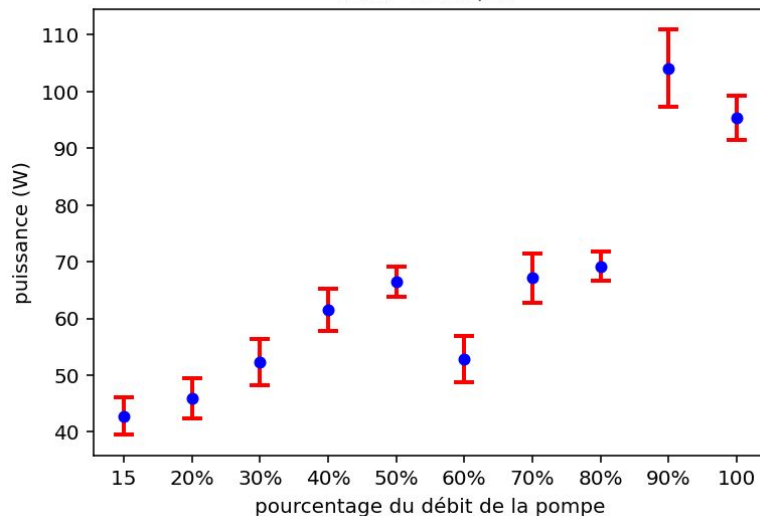
\*: méthode Z-score, codes, exemple .TXT en annexes

## 4-résultats

$P_{sol}=600W/m^2$



$P_{sol}=800W/m^2$

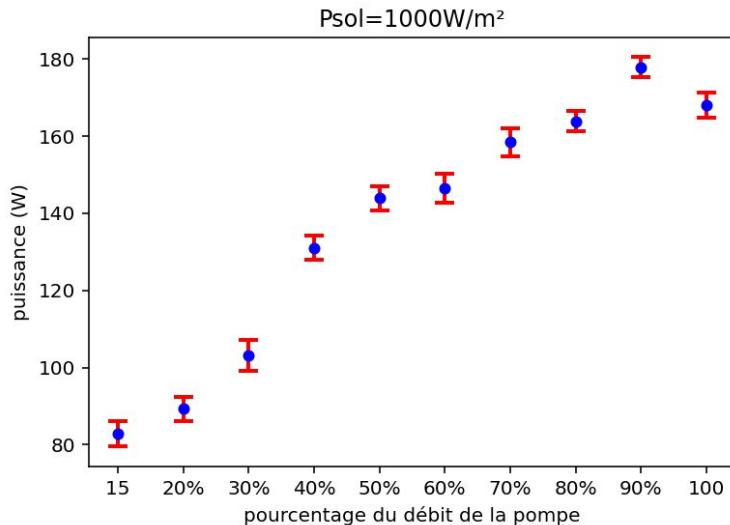
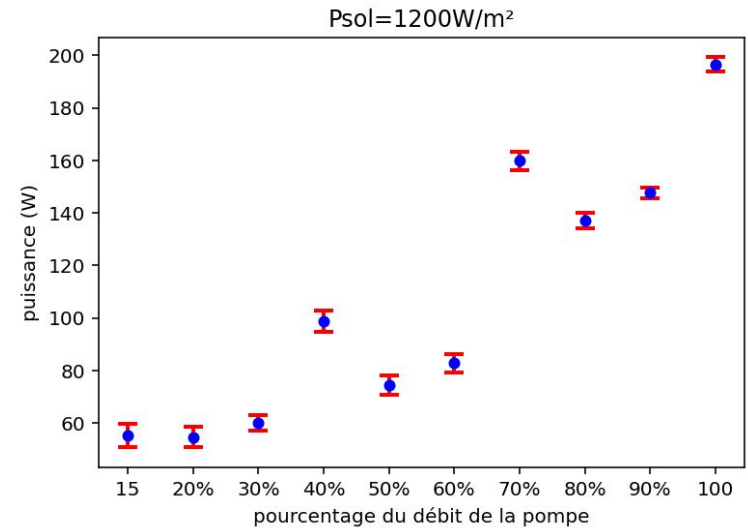
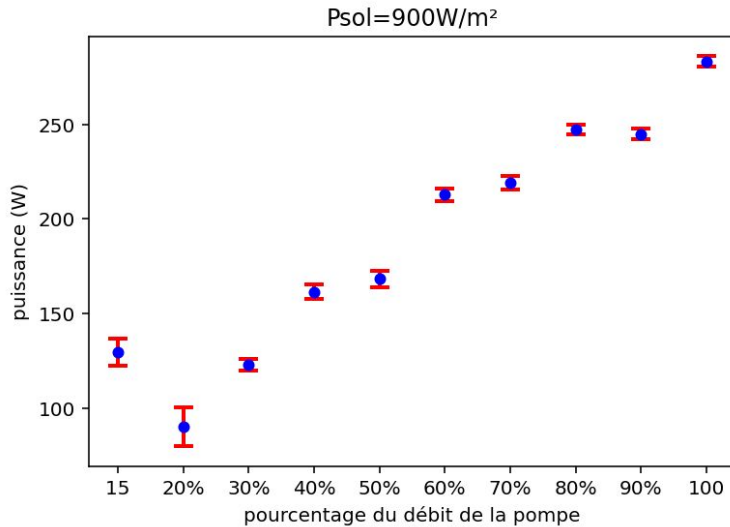


puissance en entrée basse : pic

- avant le pic, le fluide monte en température, mais ne circule pas assez vite
- après le pic, le fluide circule trop vite et n'a pas le temps de chauffer suffisamment

# 4-résultats

puissance en entrée haute



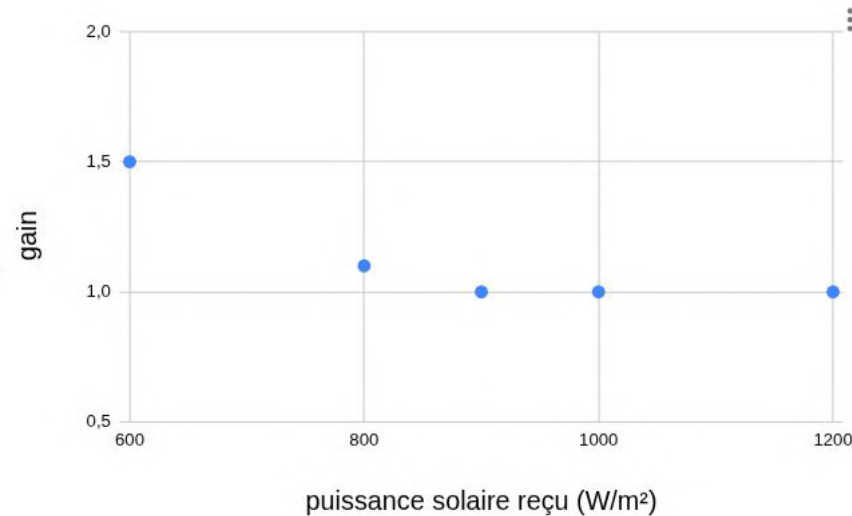
- pic non atteint, le meilleur fonctionnement est obtenu au débit maximum



## 6-Conclusions

- pompe à débit variable surtout intéressante lorsque les conditions ne sont pas optimales au fonctionnement de l'installation

$$G = \frac{P_{\max}}{P_{Dnom}}$$



- TIPE binôme : partie accessibilité

$$E = 100 \cdot \frac{\eta_{\text{panneau etude}}}{\eta_{\text{panneau wagner}}}$$

isolant	sans isolant	laine de verre	laine de roche	polystyrène extrudé
efficacité comparé au panneau Wagner	61%	65%	72%	106%

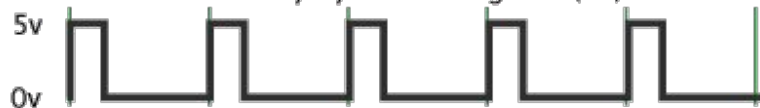
# Annexe 1 : commande PWM

## Pulse Width Modulation

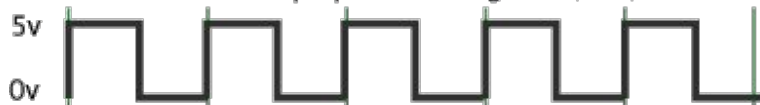
0% Duty Cycle - analogWrite(0)



25% Duty Cycle - analogWrite(64)



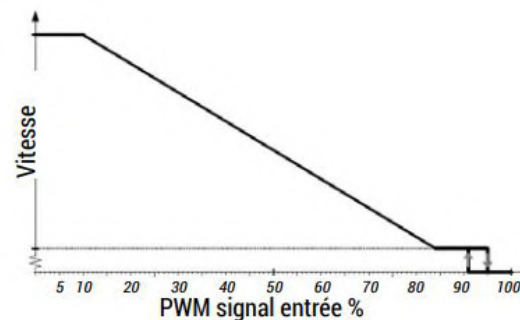
50% Duty Cycle - analogWrite(127)



75% Duty Cycle - analogWrite(191)



100% Duty Cycle - analogWrite(255)



Signal d'entrée PWM %	Etat de la pompe
0	Passer la pompe en mode non PWM (contrôle interne)
$0 < PWM \leq 10$	Vitesse maximum : max.
$10 < PWM \leq 84$	Vitesse variable : de max. à min.
$84 < PWM \leq 91$	Vitesse minimum : min.
$91 < PWM \leq 95$	Zone hystérésis : on/off
$95 < PWM \leq 100$	Mode standby : désactivée

Extrait documentation technique  
circulateur Antares

# Annexe 2 : Z-score

Objectif : filtrage des *outliers*

une valeur d'un ensemble

Z-score associé à la valeur X

$$Z = \frac{X - \mu}{\sigma}$$

moyenne de l'ensemble

écart type dans l'ensemble

- si  $Z=0$  la valeur est sur la moyenne
- Si  $Z<0$  la valeur est sous la moyenne
- Si  $Z>0$  la valeur est au-dessus de la moyenne

avec  $Z=3$ , entre 0 et 3% de valeur filtrée sur chaque mesure (valeur couramment utilisée en statistique, conserve 97% des valeurs pour une loi normal)



# Annexe 3 : code arduino

```
code type S
void setup() {
  Serial.begin(9600);
  // =====
  // Timer 0 (sorties PWM : D5 et D6)
  // =====

  // Déclaration des pins D5
  pinMode(5, OUTPUT);

  bitClear(TCCR0B, WGM02);    // Mise de WGM02 à 0
  bitSet(TCCR0A, WGM01);      // Mise de WGM01 à 1
  bitSet(TCCR0A, WGM00);      // Mise de WGM00 à 1
  // Remarque : les bits WGM01 et WGM00 sont dans le registre TCCR0A, tandis que le bit WGM02, quant

  moteur100pourcent();
}

void loop() {

  float V4 = analogRead(A0) * (5.0 / 1023.0);
  float V8 = analogRead(A1) * (5.0 / 1023.0);
  float V3 = analogRead(A2) * (5.0 / 1023.0);

  float T4 = 53.818*V4- 27.605;
  float T8 = 53.507*V8- 27.538;
  float T3 = 53.507*V3- 27.538;

  Serial.print(T3, 1);
  Serial.print(",");
  Serial.println(T8, 1);
  delay(1000);
}

void moteur100pourcent(){
  // Génération d'un signal PWM sur la sortie D5 (connectée à la pin 0C0B du µC), avec rapport cyclique
  // analogWrite(5, 85);
  // Génération d'un signal PWM sur la sortie D5 (connectée à la pin 0C0B du µC), avec rapport cyclique
  analogWrite(5, 13);
}
```

15% 800

16:59:41.248 -> 27.4,29.0

[...]

17:04:15.910 -> 29.2,25.8

30% 800

17:04:43.321 -> 27.9,26.9

[...]

17:06:43.552 -> 29.0,22.7

40% 800

17:07:21.691 -> 29.5,24.0

[...]

17:09:21.920 -> 28.4,26.3

50% 800

17:10:01.324 -> 28.7,26.6

[...]

17:12:01.556 -> 28.4,26.3

60% 800

17:12:48.319 -> 28.2,29.0

[...]

# Annexe 4 : code python

```
analyse donnée.py x
12
13 def lecture(nomfichier) :
14     fichier=open('This is a string','r')
15
16     lignes=fichier.readlines()
17
18     fichier.close()
19     j=0
20     pourcentage=[]
21     puissance_sol=[]
22     Delta_T=[[],[],[],[],[],[],[],[],[],[]]
23
24     time=[[],[],[],[],[],[],[],[],[],[]]
25     for i in range (len(lignes)) :
26         if lignes[i]==None :
27             vide=lignes[i].pop
28             elif len(lignes[i]) <=15 and len(lignes[i])>=3 :
29                 pourcentage.append(lignes[i][0:3])
30                 puissance_sol.append(float(lignes[i][4:8]))
31                 print(lignes[i][4:8])
32
33             j+=1
34
35             elif len(lignes[i])>=16 and len(lignes[i])<=28 :
36                 time[j-1].append(lignes[i][0:8])
37                 if len(lignes[i])==25 and is_float(lignes[i][15:19]) and is_float(lignes[i][20:24]):
38                     Delta_T[j-1].append((float(lignes[i][15:19])-float(lignes[i][20:24])))
39                 else :
40                     Delta_T[j-1].append((float(lignes[i][16:20])-float(lignes[i][21:25])))
41
42     return(pourcentage,puissance_sol,Delta_T)
43
44
45 def traitement(pourcentage,puissance_sol,Delta_T) :
46     puissance_moyenne=[]
47     Delta_T_moy=[]
48     rendement=[]
49     erreur=[]
50     for i in range (len(Delta_T)) :
51
52         moy_non_cor=np.mean(Delta_T[i])
53         ecart_type=np.std(Delta_T[i])
54         z_schore=[(element-moy_non_cor)/ecart_type for element in Delta_T[i]]
55         moy_cor=0
56         compteur=0
57         erreur.append(2*ecart_type)
58         for j in range(len(Delta_T[i])) :
59             if abs(z_schore[j])<=3:
60
61                 if abs(z_schore[j])<=3:
62                     moy_cor+=Delta_T[i][j]
63                     compteur+=1
64                     print(100*(100*(Delta_T[i]-compteur)/len(Delta_T[i])), '%')
65                     This is an integer (moy_cor/compteur)
66                 for i in range(len(Delta_T_moy)) :
67                     if i==0 :
68                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*200/60))
69                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
70
71                     if i==1 :
72                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*247/60))
73                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
74                     if i==2 :
75                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*341/60))
76                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
77                     if i==3 :
78                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*435/60))
79                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
80                     if i==4 :
81                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*530/60))
82                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
83                     if i==5 :
84                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*623/60))
85                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
86                     if i==6 :
87                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*710/60))
88                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
89                     if i==7 :
90                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*811/60))
91                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
92                     if i==8 :
93                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*905/60))
94                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
95                     if i==9 :
96                         puissance_moyenne.append(3.6*Delta_T_moy[i]*(1.048*1000/60))
97                         rendement.append(100*puissance_moyenne[i]/(2*puissance_sol[i]))
98
99                 return (puissance_moyenne,rendement,erreur)
100
101     pourcentage1,puissance_sol1,Delta_T1=lecture('MESURE 1200W.txt')
102     puissance_moyenne1,rendement1,erreur1=traitement(pourcentage1,puissance_sol1,Delta_T1)
103
104     plt.title("Psol=1200W/m²")
105     plt.errorbar(range(len(puissance_moyenne1)), puissance_moyenne1, yerr = erreur1,
106                 fmt = 'none', capsize = 5, ecolor = 'red', elinewidth = 2, capthick = 2)
107     plt.plot(pourcentage1,puissance_moyenne1,linestyle = 'none', marker = 'o', c = 'blue',
108             markersize = 5)
109     plt.xlabel("pourcentage du débit de la pompe")
110     plt.ylabel("puissance (W)")
111     plt.show()
```

# Annexe 5 : Etalonnage des PT1000

- Réalisation de L'étalonnage:
  - Protocole:
    - Chauffer de l'eau dans une bouilloire
    - réaliser des mesures de la résistance à des température fixées (tous les 5°)



$$T = 0,25 \times R - 250$$

