

# Un exemple d'utilisation de la logique floue

## Intro: Objectif

### I/ Le Fonctionnement :

- 1) Fonctionnement général de l'algorithme
- 2) Dialogue entre les différents composants (interface)
- 3) Capteurs et conversion
- 4) Présence de logique floue (lois, fonction d'appartenance, obtention d'une décision)

### II/ Les travaux réalisés :

- 1) Programmation en Python
- 2) Programmation en C du microcontrôleur
- 3) L'enceinte expérimentale (protocole, photo)
- 4) Calcul des dépensements thermiques objectives.

### III/ Les résultats :

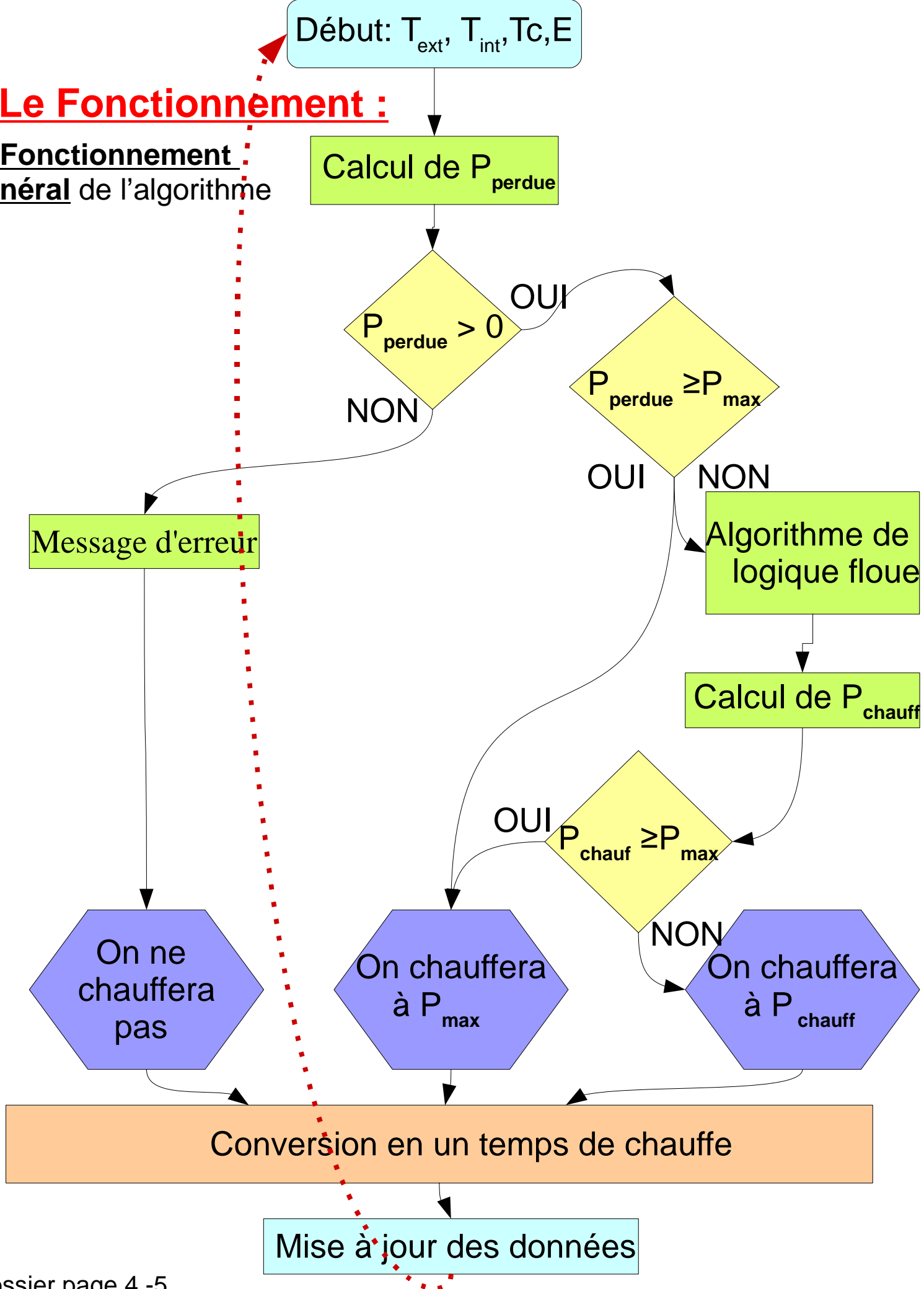
Les différentes courbes de régulations et tableau récapitulatif

## Conclusion

Début:  $T_{ext}$ ,  $T_{int}$ ,  $T_c$ ,  $E$

## I/ Le Fonctionnement :

### 1) Fonctionnement général de l'algorithme



# I/ Le Fonctionnement :

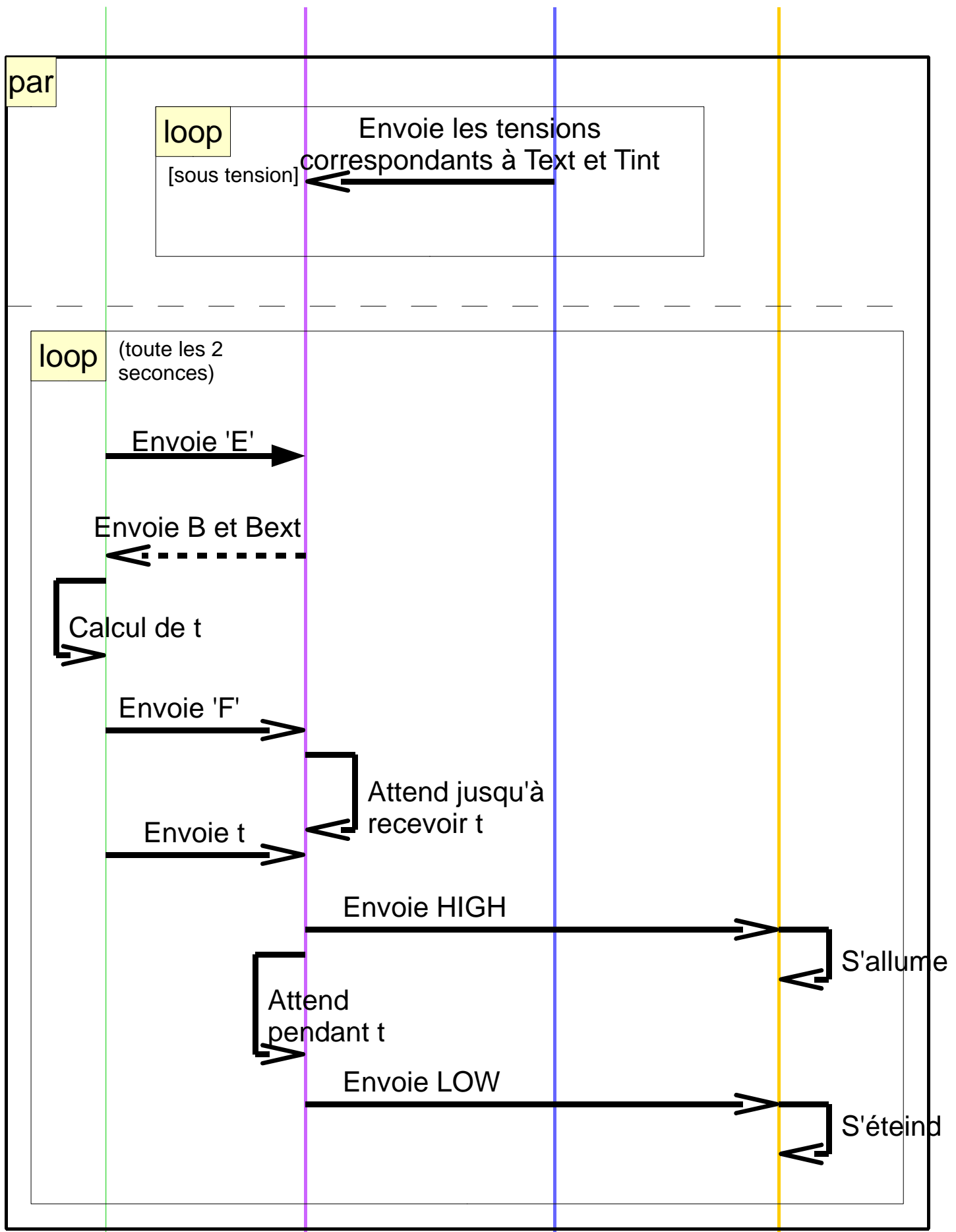
## 2) dialogue entre les différents composants

PC

Carte

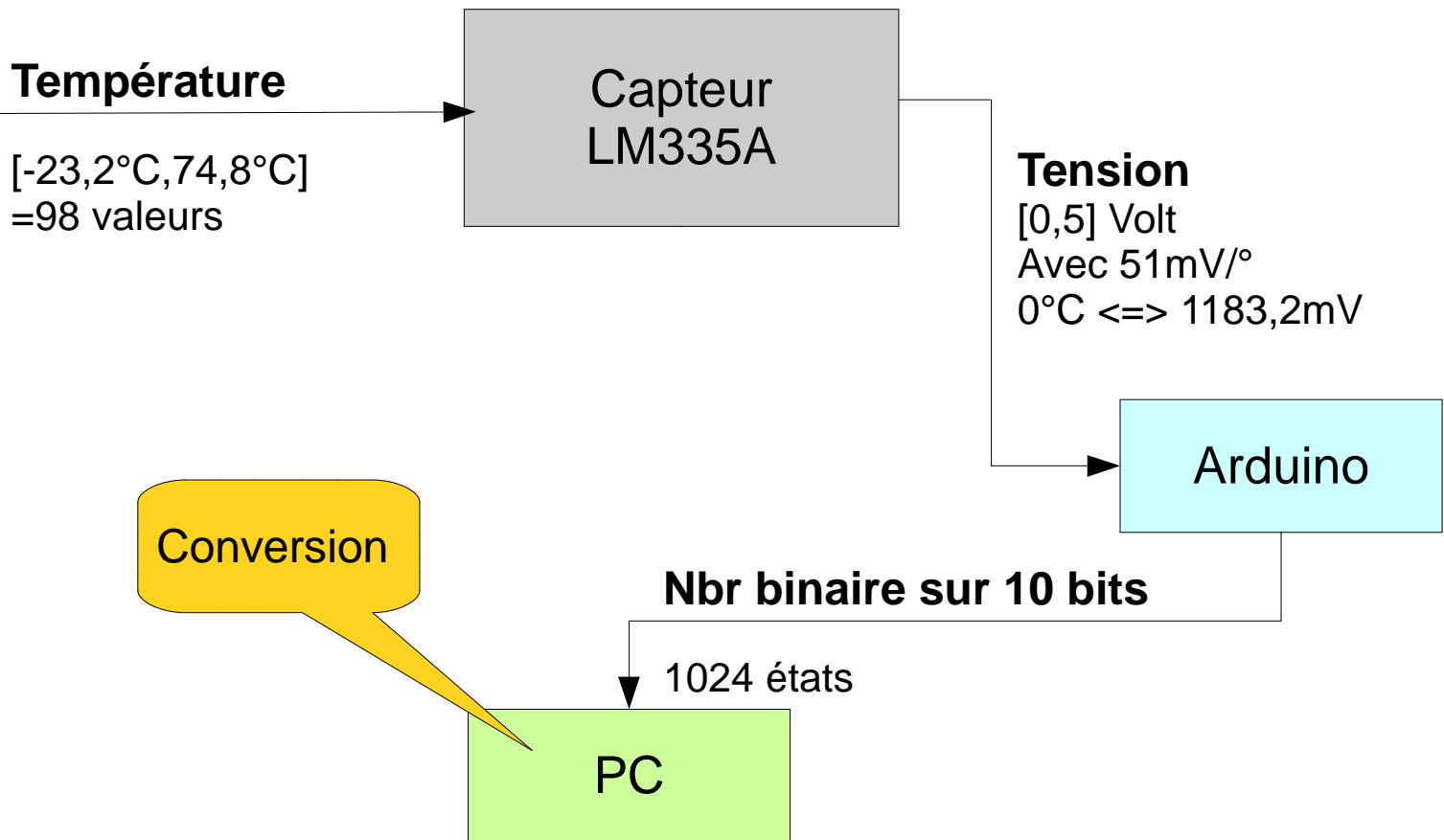
C1 et C2

Chauffage



## I/ Le Fonctionnement : 3) Capteurs et conversion

-Le LM335A dans son contexte



```
def conversion(B):  
    mvB=B*5000./1024.  
    celsiusB= (mvB -1183.2)/51.  
    return(celsiusB)
```

**Résolution de mesure:**  
 $98/1024 = 0,1^\circ$

I/ Le Fonctionnement : 4) Présence de logique floue (lois, fonction d'appartenance, obtention d'une décision)

\_\_\_ exemple de *lois qualitatives de régulation* :

Loi n°1: S'il [fait **bon** ]OU [qu'il fait **froid** et que l'utilisateur est **très économe**] *alors je ne compense que les pertes thermiques.*

\_\_\_ Les *ensembles flous* :



Caractérise la chaleur intérieure



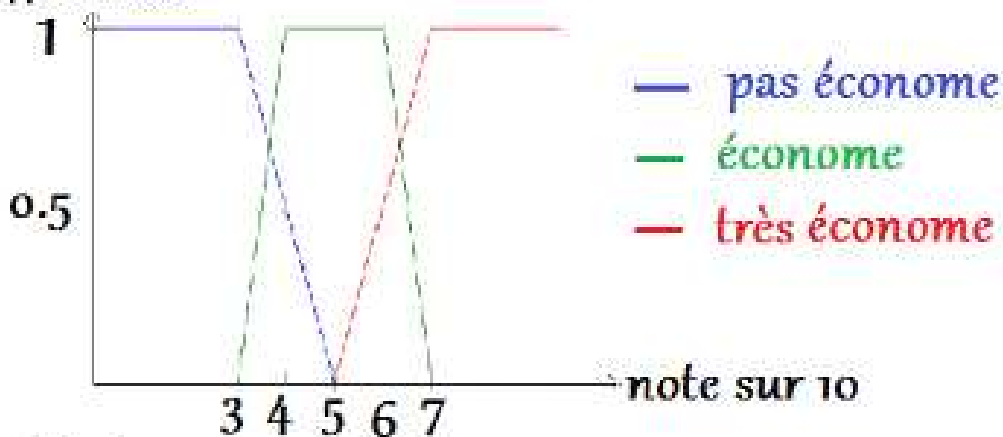
Correspond au caractère économe de l'utilisateur



Caractérise l'action permettant de réguler

\_\_\_ Les *fonctions d'appartenance* de chaque variable linguistique (**température**, **caractère économe**, **puissance**) aux ensembles (**chaud**, **très économe**, **ne pas chauffer**)

● Variable linguistique: Caractère de l'utilisateur  
degré d'appartenance



\_\_\_ En fonction des lois de régulation et de degré d'appartenance obtenu, l'algorithme de logique floue va retourner une *décision quantitative* **Pchauff**.

## II/ Les travaux réalisés :

### 1) Programmation en Python

**Orange / Marron:** affichage des données en temps réel

**Violet:** Interface Ordinateur-Arduino

**Vert:** Logique floue

**Rose:** Conversion (Binaire → Température en °C )

**Cyan:** Déperdition thermique et autres valeurs

**Rouge:** Sauvegarde des données

**Bleu:** Conversion de la puissance de chauffe en un temps de chauffe

---

### 2) Programmation en C (Arduino)

**Gris:** Définition des broches

**Violet:** Interface Arduino-Ordinateur

**Bleu:** Interface Arduino-Commande de chauffage

**Cyan:** Interface Arduino-Capteur

## II/ Les travaux réalisés:

### 3) L'enceinte expérimentale

Dossier page: 24

#### Matériel:

une règle

4 Panneaux de polystyrène extrudé URSA XPS NW1 de taille officiel 1250x600x20.

Tounevis

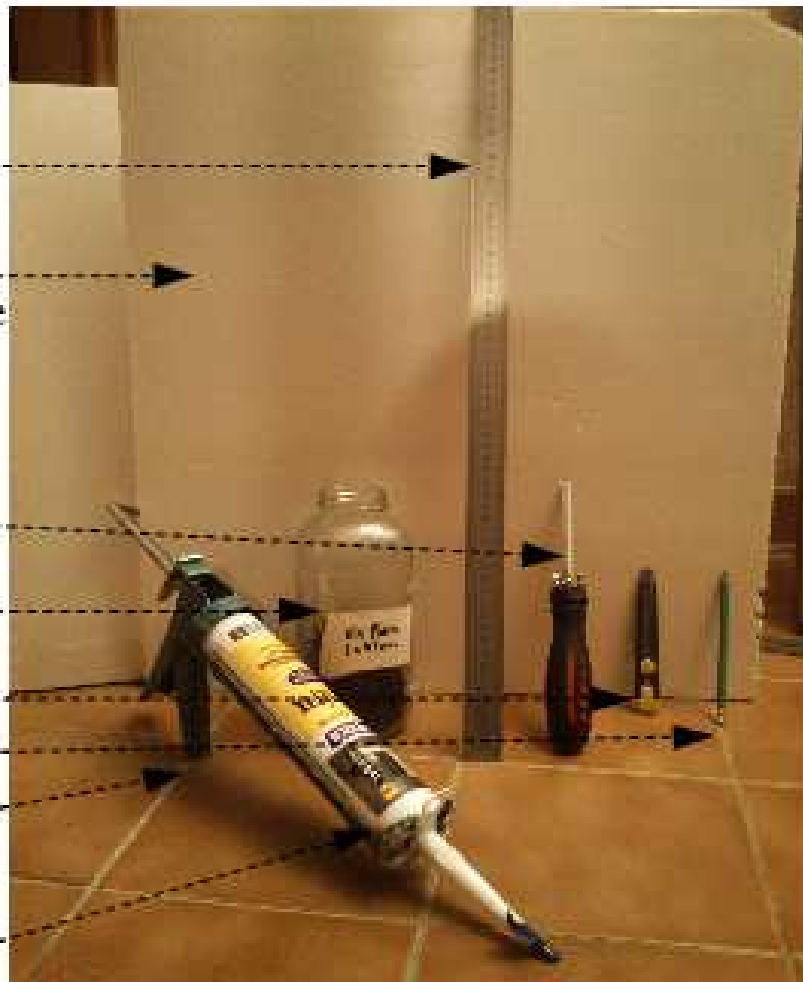
24 vis placo de longueur 45mm

un couteur

un crayon

un pistolet à colle

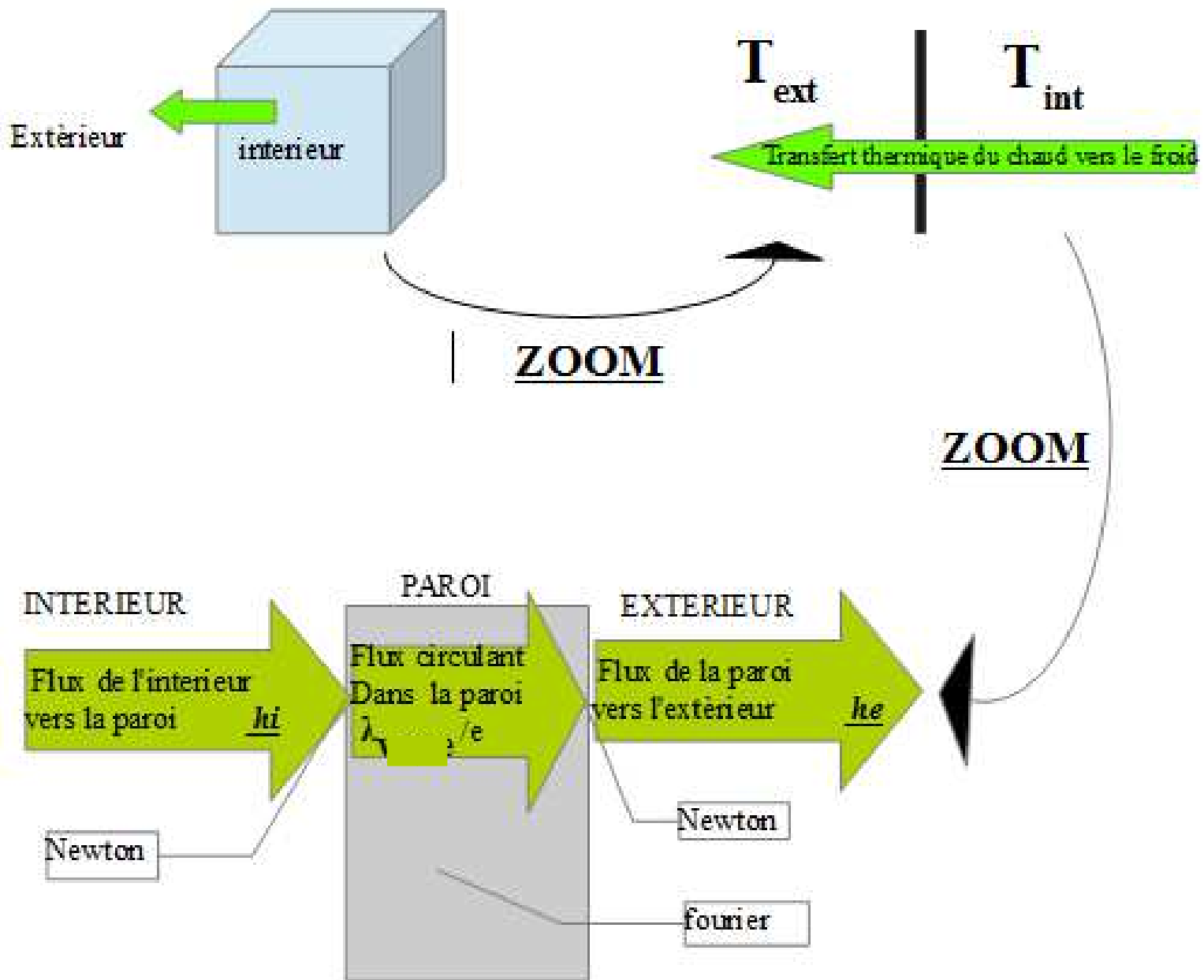
1 tube de mastic de fixation Cotéfix 3 10ml



1. On **coupe** soigneusement 6 planches de polystyrène de 60x60x60cm.
2. On **colle** avec le mastic 4 planches ensemble (1tranche sur le côté d'une face) pour former les côtés du cube.
3. On les **maintient** en place grâce à des vis auto-persçantes.
4. On **colle et visse** la planche correspondant au plafond du cube.
5. On rajoute du mastic dans les coins et sur les bords pour qu'il n'y ait pas de trou.

6. On **pause** délicatement le tout sur la planche du bas et on **attend** 48heures.

## II/ Les travaux réalisés :4) Calcul des déperditions thermiques objectives.



Loi phénoménologique de Fourier :  $j_{th} = -\lambda \cdot \text{grad}(T)$

Loi de Newton:  $j = h \cdot (T_{\text{paroi}} - T_{\text{fluide}})$ , où  $h$  est un coefficient empirique appelé coefficient de transfert convectif.

$$P_{\text{perdu}} = A(T_{\text{ext}} - T_c)$$

$$A = \underline{2,2653 \text{ W. K}^{-1}}$$

—————► Le calcul se trouve dans le fichier\_val()



III/ Les résultats :

Evolution temporelle de la température

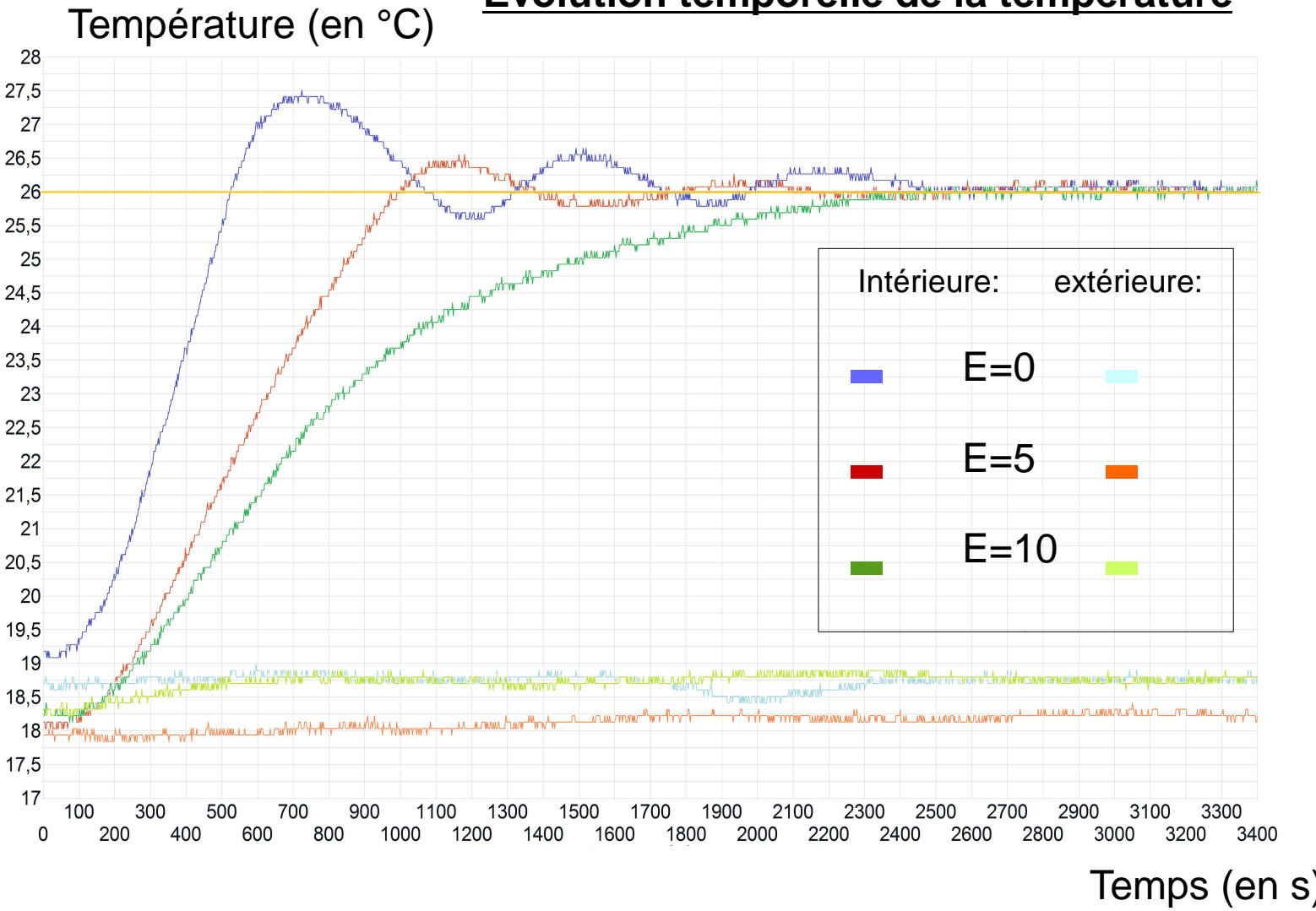


Tableau récapitulatif:

Caractère économe (note /10)	Retard (+ ou - 2s)	Température de départ  (+ ou - 0,1°C)	Température extérieure moyenne (+ ou - 0,1°C au maximum)	Variation de température extérieure (+ ou - 0,1°C)	Temps de montée	Dépassement maximal / temps associé (+ ou - 0,1°C)	
0	74s	19,2°C	18,7°C	0,6°	526s	1,5°C	725s
5	110s	18,1°C	18,1°C	0,6°	1012s	0,4°C	1158s
10	180s	18,3 °C	18,7°C	0,7°	2472s	0,1°C	2650s

Conclusion

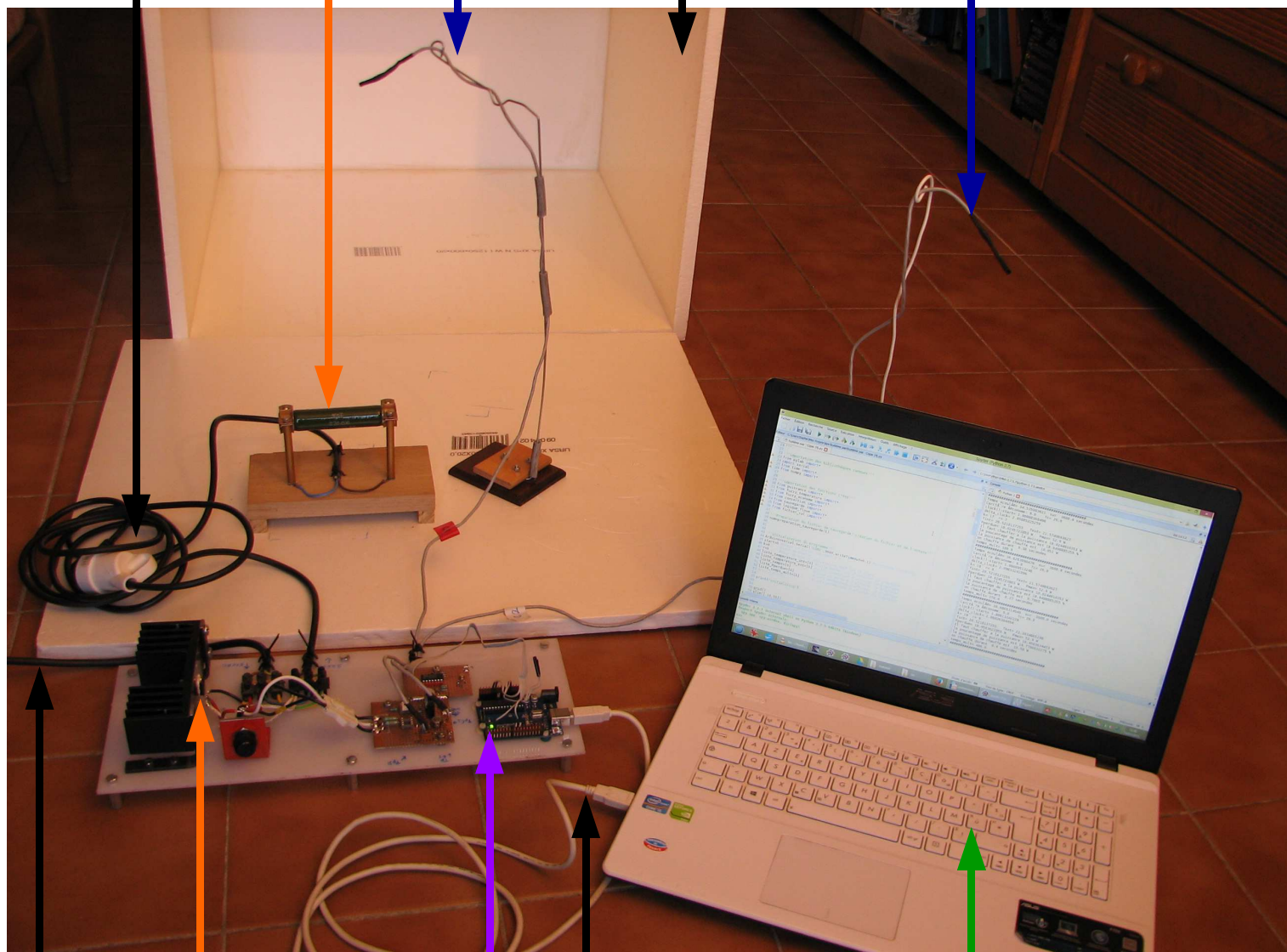
Câble d'alimentation  
(alimente le résistor)

Enceinte expérimentale

Capteur intérieur

Résistor  
(chauffage)

Capteur extérieur



Triac

Arduino

Ordinateur

Câble secteur

Câble USB

( Alimentation de l'Arduino + écriture et lecture de donnée série )