SCIENCES DE L'INGENIEUR

Séquence 5 - Activité 2d

Confort thermique dans l'habitat

Durée : 03H00

Objectif :

→ **Prototyper** le dispositif de régulation d'un réseau de chaleur dans une habitation.

Préalable : *l'activité 2c doit être faite pour comprendre le contexte de cette activité.*

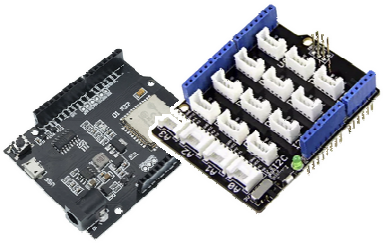
Prototyper le dispositif de régulation de température implique de :

→ **Câbler** correctement du matériel,

→ **Programmer** un fonctionnement dans le respect du cahier des charges.

Composants à mettre en œuvre pour le prototypage :

Capteur de température Grove 101020015



Microcontrôleur ESP32 + Shield

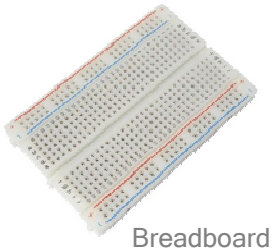
Régulateur de tension LM7805

Servomoteur SG90

Pile 9V CC

Connecteur pile 9V

Connectique 4 fils



Breadboard

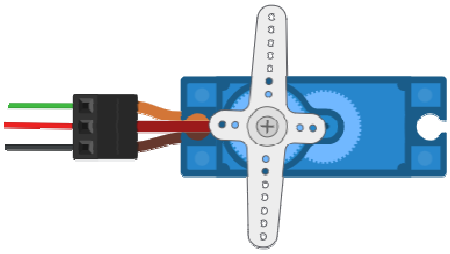
Dans toute la suite, seul le mode automatique est considéré ; l'algorigramme vu en partie C de l'activité 2a s'en trouve allégé (voir celui de l'activité 2b !).

Pour des raisons évidentes de sécurité, nous ne manipulerons ni gaz, ni flamme.

Par soucis de simplicité, la température de consigne sera définie directement dans le programme.

Cahier des charges :

→ Le pilotage de l'électrovanne qui distribue le gaz est limité à la gestion de la position du servomoteur.



Servomoteur SG90
Position « Vanne fermée »
Le gaz n'est pas distribué

Servomoteur SG90
Position « Vanne ouverte »
Le gaz est distribué

→ Si la température réelle de l'habitat T_{reelle} est <u>inférieure</u> à la température de consigne $T_{consigne}$, alors le gaz est distribué.

→ Si la température réelle de l'habitat T_{reelle} est <u>supérieure ou égale</u> à la température de consigne $T_{consigne}$, alors le gaz n'est pas distribué.

→ Petit plus en option : La température réelle de l'habitat (T_{reelle}) sera affichée dans la console de l'EDI avec son unité (°C).

Spécifications techniques :

→ La température de l'habitat évolue dans la plage : + 5°C < $T_{habitat}$ < + 40°C

→ On admet une précision sur la mesure de la température du logement : $\Delta T_{habitat}$ = 4°C

→ La température de consigne est fixée à $T_{consigne} = 19\ °C$ (pour faire plaisir à l'exécutif).

→ La température de consigne $T_{consigne}$ sera écrite en dur dans le programme.

Contraintes matérielles :

→ Carte de développement imposée : ESP32 avec shield

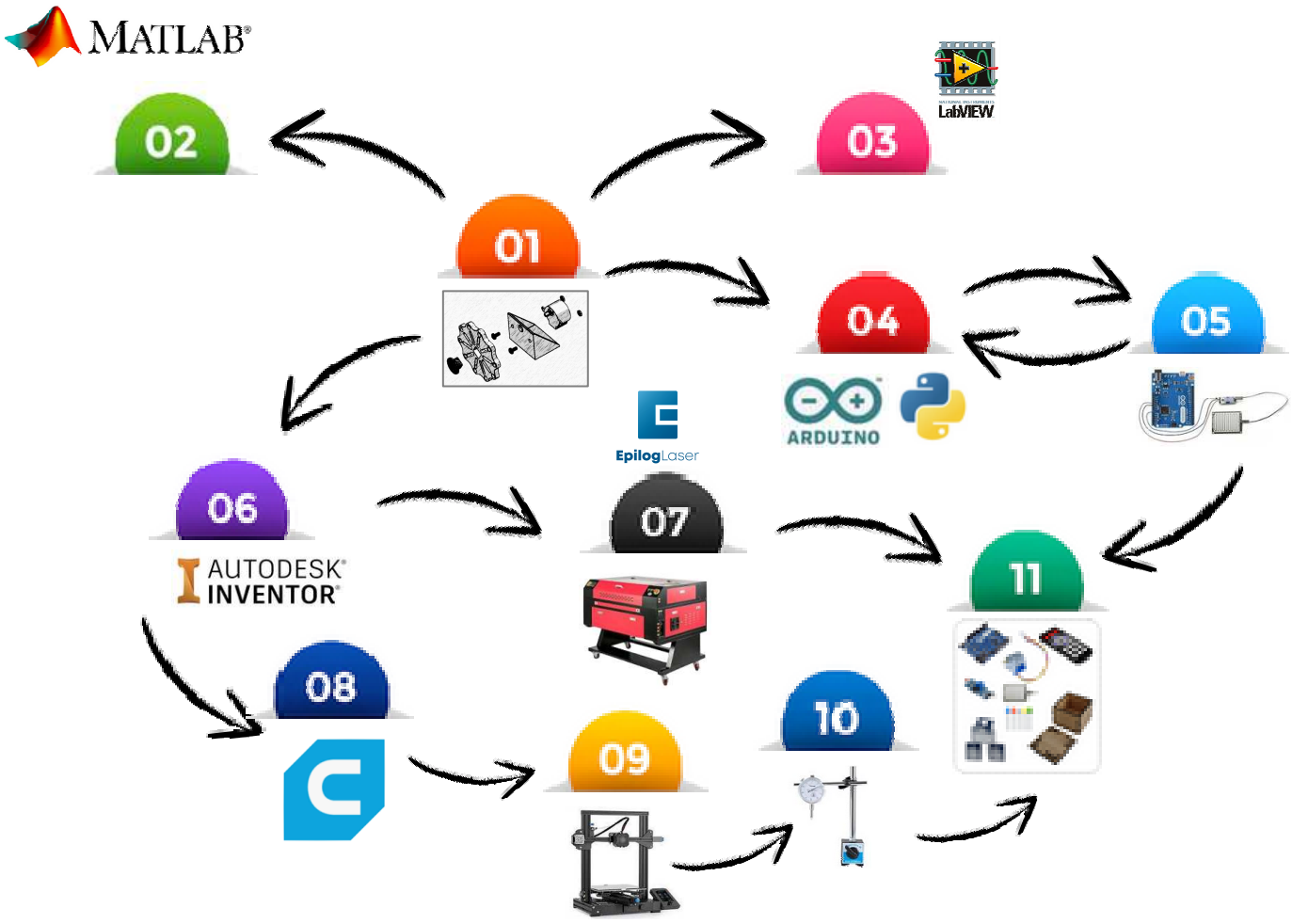
→ Langage de programmation : Python

PARTIE A

Positionnement du travail dans une démarche de projet

➘ **Consulter** la fiche de cours n°4 du chapitre 1, en particulier le synoptique donné à la page 2.

Q1 – Entourer dans le synoptique simplifié ci-dessous les compétences (un numéro et sa vignette) qui vont être travaillées au regard du travail à faire, c'est-à-dire le prototypage du système de régulation.

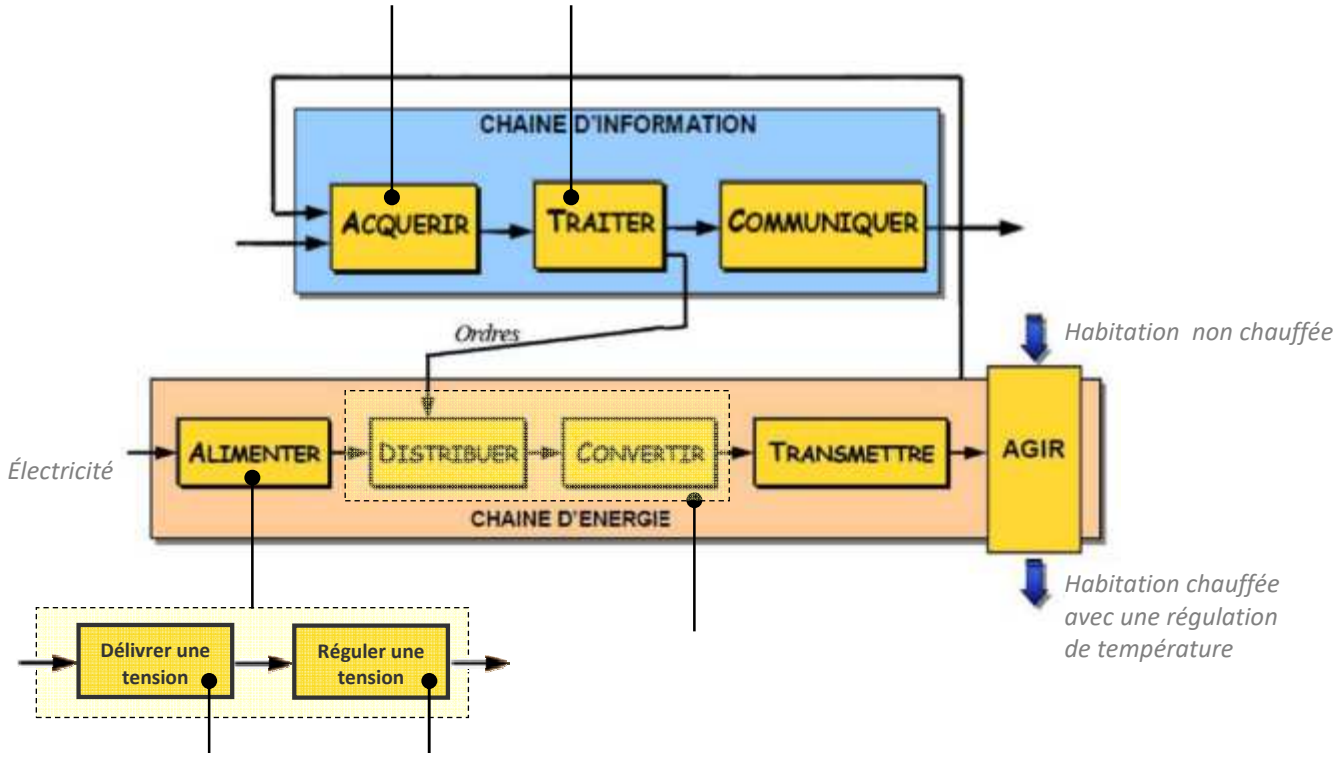


PARTIE B

Analyse fonctionnelle des composants

Q2 – Positionner les cinq composants demandés dans le modèle fonctionnel.

☞ *Les composants à considérer font partie de la liste définie dans la page 1.*



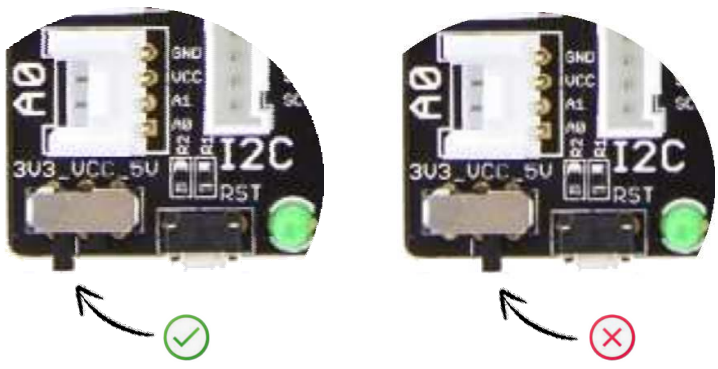
PARTIE C

Câblage des composants

➘ **Câbler** les composants.

☞ *Consulter les datasheets en ligne (capteur Grove, Shield, carte ESP32, etc.).*

Régler le shield sur 3,3V pour l'adapter à la carte ESP32.



➘ **Appeler** le professeur avant d'aller plus loin.

PARTIE D

Validation du choix de capteur

On s'attache ici à vérifier si les caractéristiques générales du capteur de température Grove 101020015 sont compatibles avec les spécifications du contexte qui est le nôtre.

➘ **Consulter** la datasheet du capteur de température Grove 101020015.

Q2 – Donner en °C la plage de température pour laquelle le capteur est prévu : _____

Q3 – Donner en °C la tolérance du capteur : _____

Q4 – Calculer en °C l'intervalle de tolérance du capteur : _____

Q5 – Conclure sur la capabilité du capteur au regard des spécifications imposées.

☞ *Un textuel court mais précis et bien construit est attendu.*

_____

_____

_____

_____

_____

_____

➘ **Appeler** le professeur avant d'aller plus loin.

PARTIE E

Étalonnage du capteur

Afin de réaliser le programme, il s'avère nécessaire de connaître pour le capteur la relation donnant la tension V_{out} (en V) qu'il délivre en fonction de la température T_{reelle} (en °C) à laquelle il est soumis.

*La relation mathématique $T_{reelle} = f(V_{out})$ s'appelle « **courbe d'étalonnage** » ; il faut la trouver…*

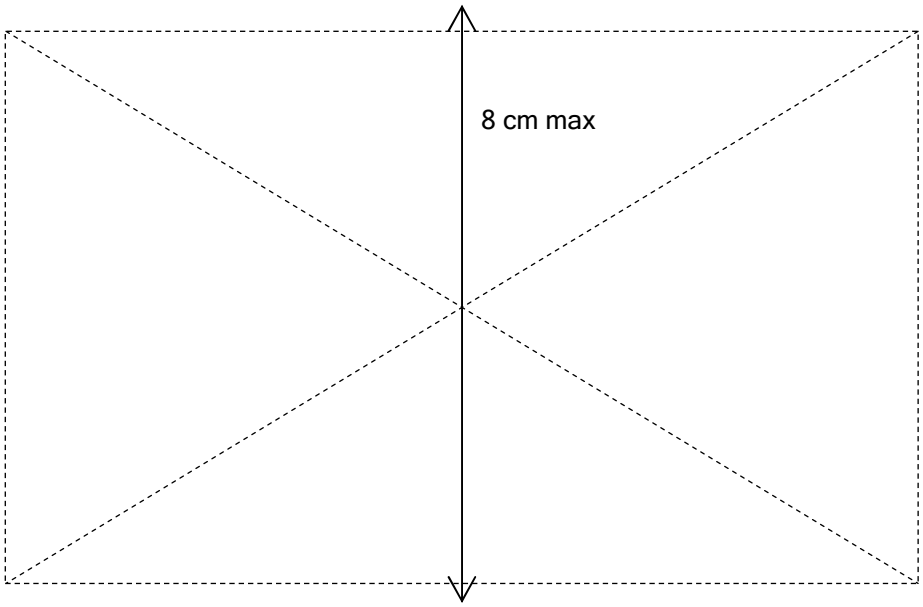
➘ **Consulter** la datasheet du capteur de température Grove 101020015 et **constater** que la relation $T_{reelle} = f(V_{out})$ *n'est pas donnée (pas de chance ; il va falloir construire la courbe d'étalonnage…).*

Hypothèse forte : on admet que le capteur a un comportement linéaire.

➘ **Construire** la courbe d'étalonnage $T_{reelle} = f(V_{out})$ du capteur.

- ☞ *Suivre la procédure donnée dans les documents en ligne « Mise en œuvre >> Etalonnage des capteurs ».*
- ☞ *Voir avec le professeur pour le nombre de points de mesure à prendre en compte.*
- ☞ *Les températures mini et maxi sont à caler autour des limites de la plage de température de l'habitat.*
- ☞ ***Télécharger et travailler dans le classeur Excel « Modèle pour étalonnage.xls ».***

➘ **Imprimer** et **coller** ici le graphique fait sous Excel (appeler le professeur pour vérification préalable).



Équation d'étalonnage $T_{reelle} = f(V_{out})$ du capteur de température Grove 101020015
(préciser les unités)

$T_{reelle}(V_{out}) =$ _____

➲ Type de carte de prototypage : ☐ Arduino ☐ ESP32 ☐ Autre : _____

➲ Nombre de bits pour la numérisation : ____ ➲ Amplitude de la tension sur la broche : _____

➲ Calculs : _____

_____

Équation $T_{reelle}(N)$ à utiliser dans le programme
(préciser les unités)

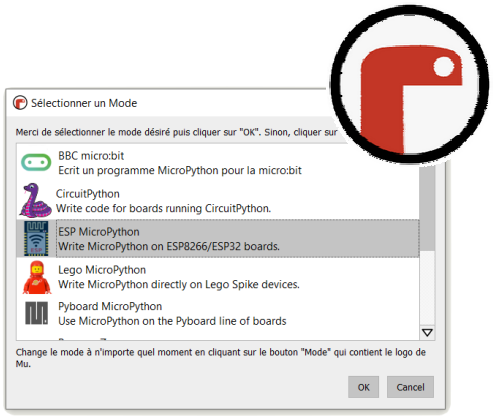
$T_{reelle}(N) =$ _____

PARTIE F

Programmation du fonctionnement

➘ **Exécuter** l'EDI « Mu Python ».

☞ *Voir avec le professeur pour cette exécution.*



➘ **Sélectionner** le mode « ESP MicroPython ».

➘ **Ouvrir** le fichier « Régulateur de température.py ».

☞ *Ce programme sera à compléter (on vous dira plus loin avec quoi).*

➘ **Analyser le code** de façon à en comprendre l'essentiel et peut être deviner ce qui manque.

☞ *Recouper les lignes de code avec l'algorigramme de l'activité 2b pour comprendre ce qu'elles signifient.*

➘ **Intégrer dans le code** la valeur de la température de consigne « $T_{consigne}$ ».

☞ *voir les spécifications techniques en page 2.*

➘ **Intégrer dans le code** la courbe d'étalonnage de la température mesurée, $T_{reelle}(N)$.

Note : les variables utilisées dans le code sont les suivantes :

Nom de la variable	Type de variable	Ce que représente la variable
Tconsigne	Entier (int)	Température de consigne déclenchant l'allumage de la chaudière : $T_{consigne}$ = 19 °C
N	Entier (int)	Valeur numérique renvoyée par le CAN lors de l'acquisition de la tension délivrée par le capteur
Valim	Nombre à virgule flottante (float)	Valeur (en volt) de la tension d'alimentation du capteur : V_{alim} = 3,3 V
Vout	Nombre à virgule flottante (float)	Valeur (en volt) de la tension acquise par le microcontrôleur (entrée du CAN), fournie par le capteur
MonServoMoteur	Broche (pin)	Broche à laquelle est connectée la commande du servomoteur. N° de la broche du GPIO : 25 - Sens : sortie (OUT)