

Supervision et régulation de la température d'une habitation

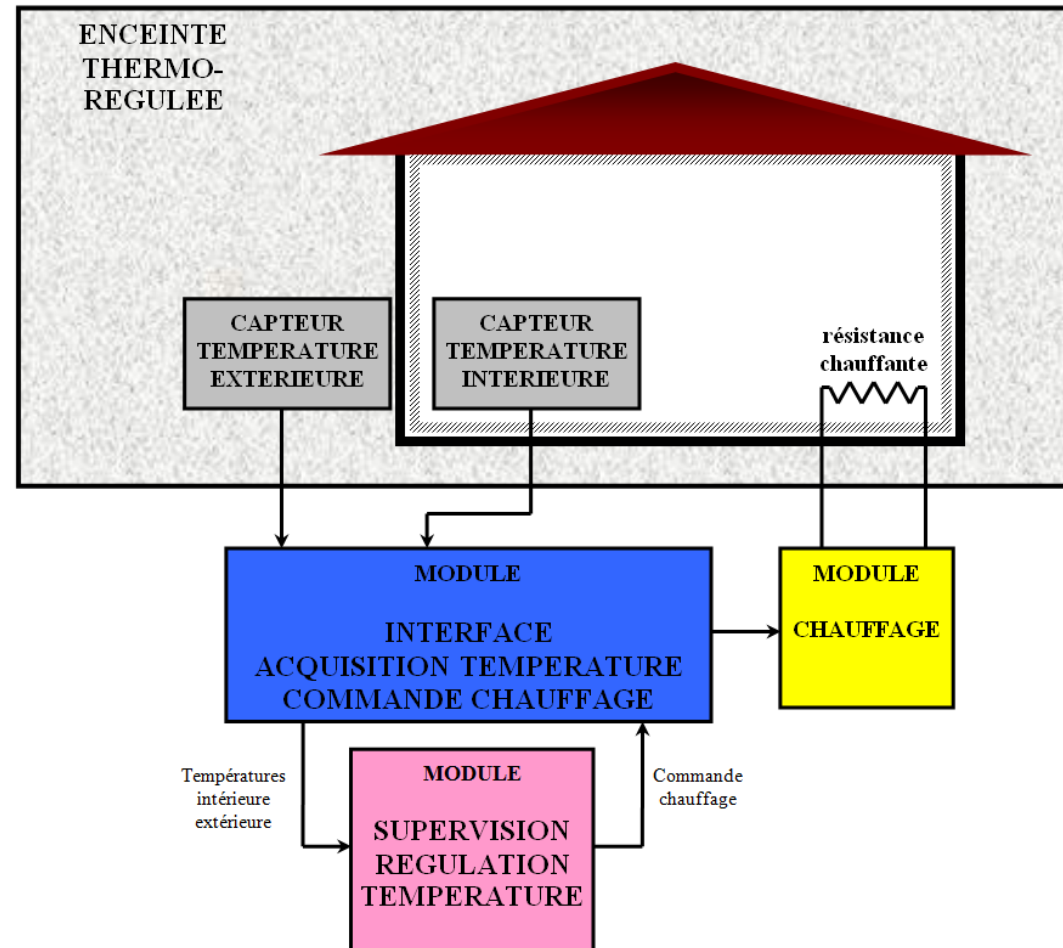
Partie informatique

Pierre-Jean BOUVET



Présentation

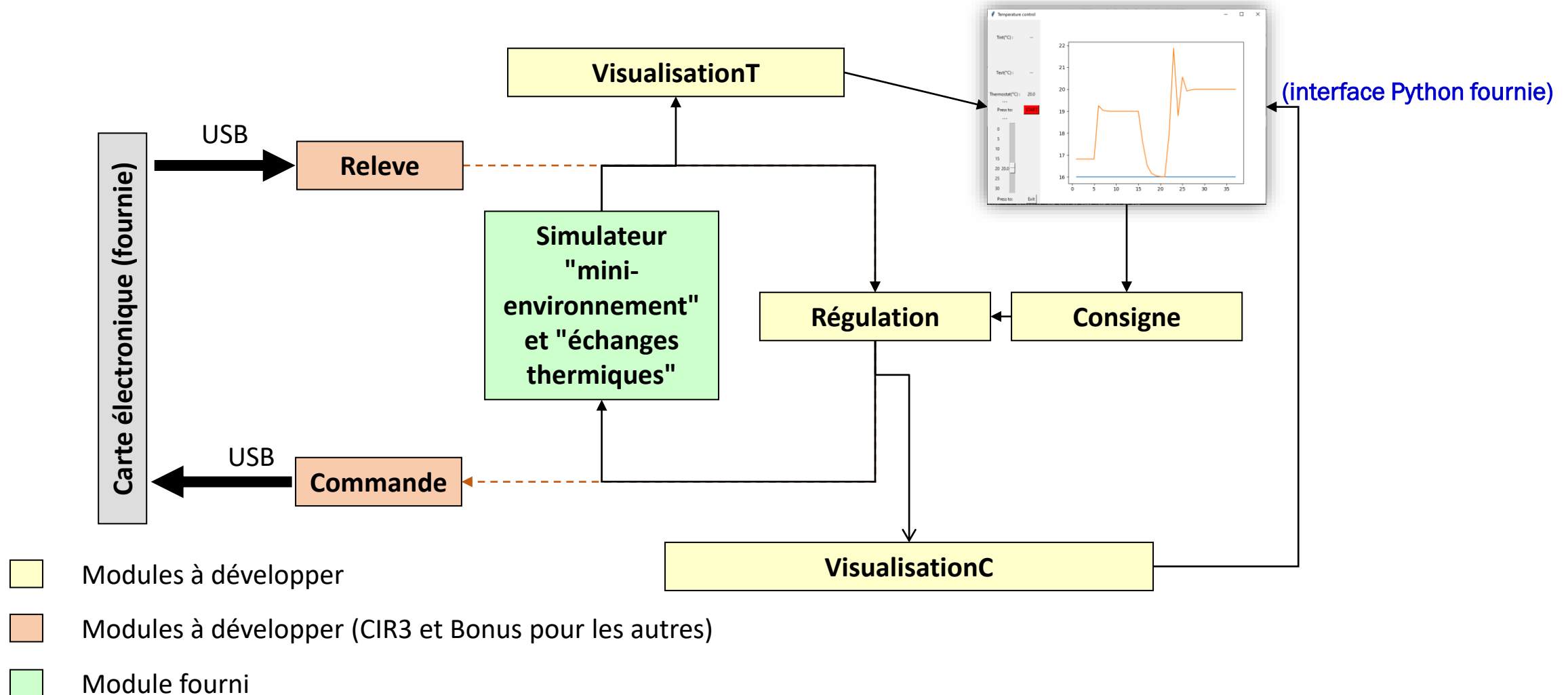
Description générale



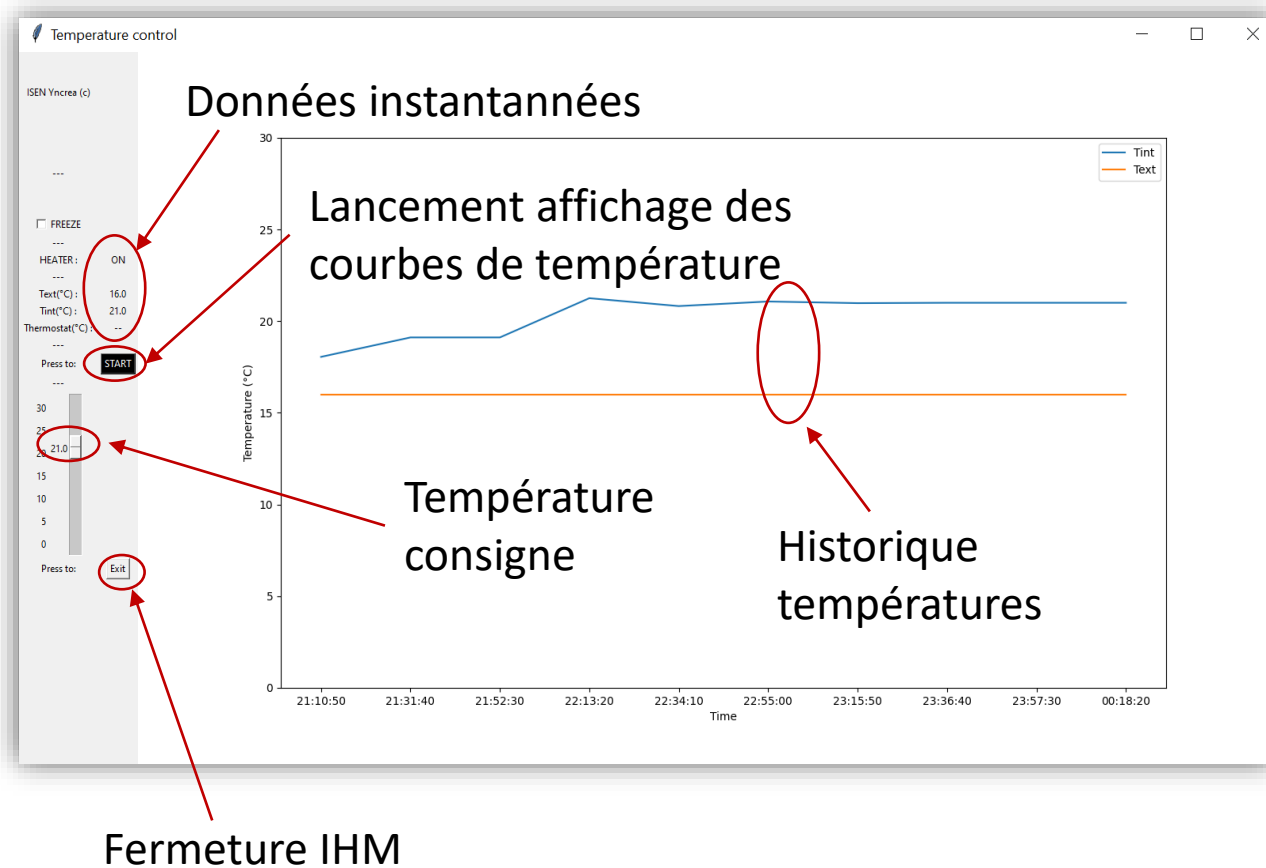
Description de la partie informatique

- Régulation de la température intérieure à partir des informations reçues : mise en œuvre de la régulation tout ou rien et de la régulation PID
- Visualisation des températures et du témoin de chauffage via une interface utilisateur
- Relevé des températures intérieures et extérieures via liaison USB
- Commande de chauffage via liaison USB

Synoptique de la partie informatique



Interface de commande et de visualisation



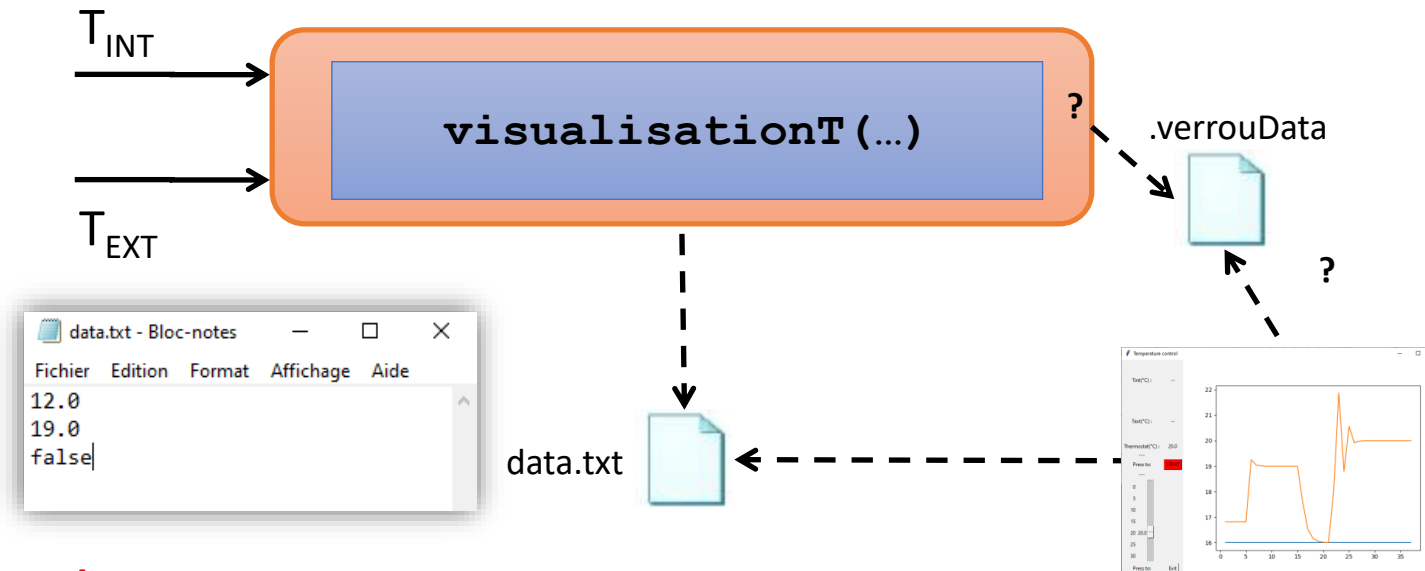
- Module fourni (Python)
- Interface de commande
 - Réglage de la consigne de température (thermostat)
- Visualisation
 - Température intérieure et extérieure
 - Témoin de chauffe

Equipe encadrante

BREST :	P.-J. BOUVET	Pierre-jean.bouvet@isen-ouest.yncrea.fr
	J.-B. PIERROT	jean-benoit.pierrot@isen-ouest.yncrea.fr
	F. LEGRAS	legrasf@me.com
	H. BENALLAL	hafsa.benallal@isen-ouest.yncrea.fr
NANTES :	N. BEAUSSE	nils.beausse@isen-ouest.yncrea.fr
CAEN :	N. ABDALLAH SAAB	nadine.abdallah-saab@isen-ouest.yncrea.fr
	S. LE GLOANNEC	simon.le-gloannec@isen-ouest.yncrea.fr

Les différents blocs

Visualisation des températures



■ Points essentiels

- Ecriture des informations dans le fichier `data.txt`

■ Points critiques

- Exclusion mutuelle à l'aide d'un verrou

Visualisation des températures

■ Test existence d'un fichier

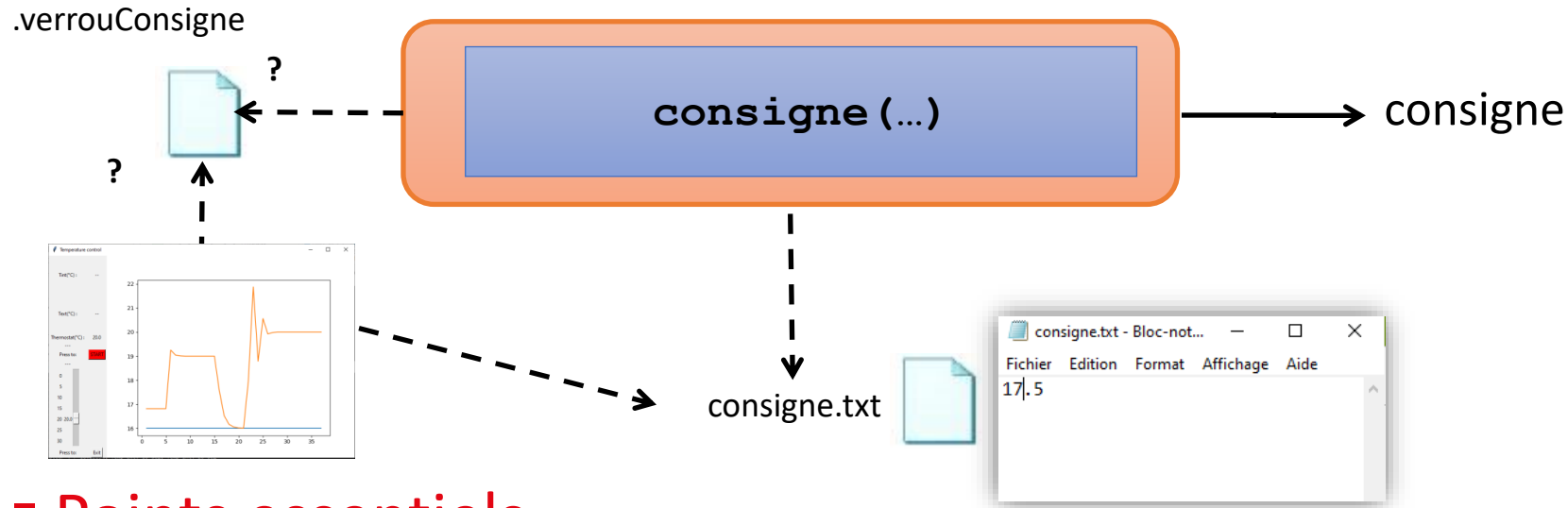
```
#include <unistd.h>

if( access( ".verrouData", F_OK ) != -1 )
{
    // Fichier existe
}else{
    // Fichier n'existe pas
}
```

■ Effacer un fichier

```
remove( ".verrouData" )
```

Lecture de la consigne



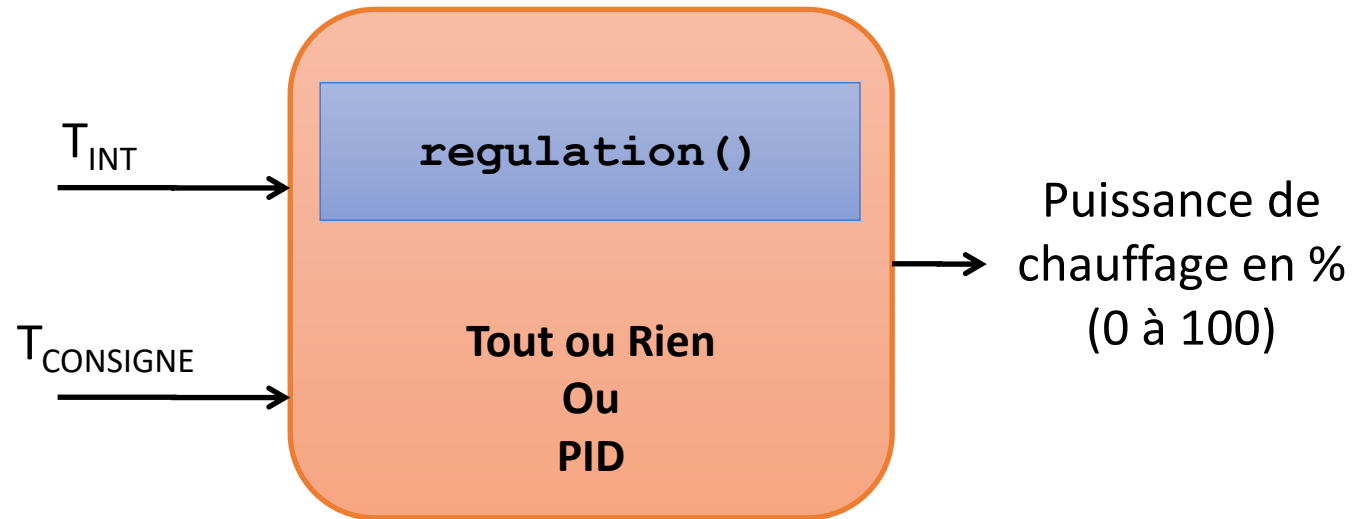
■ Points essentiels

- Lecture des informations dans le fichier consigne.txt

■ Points critiques

- Exclusion mutuelle à l'aide d'un verrou

Régulation



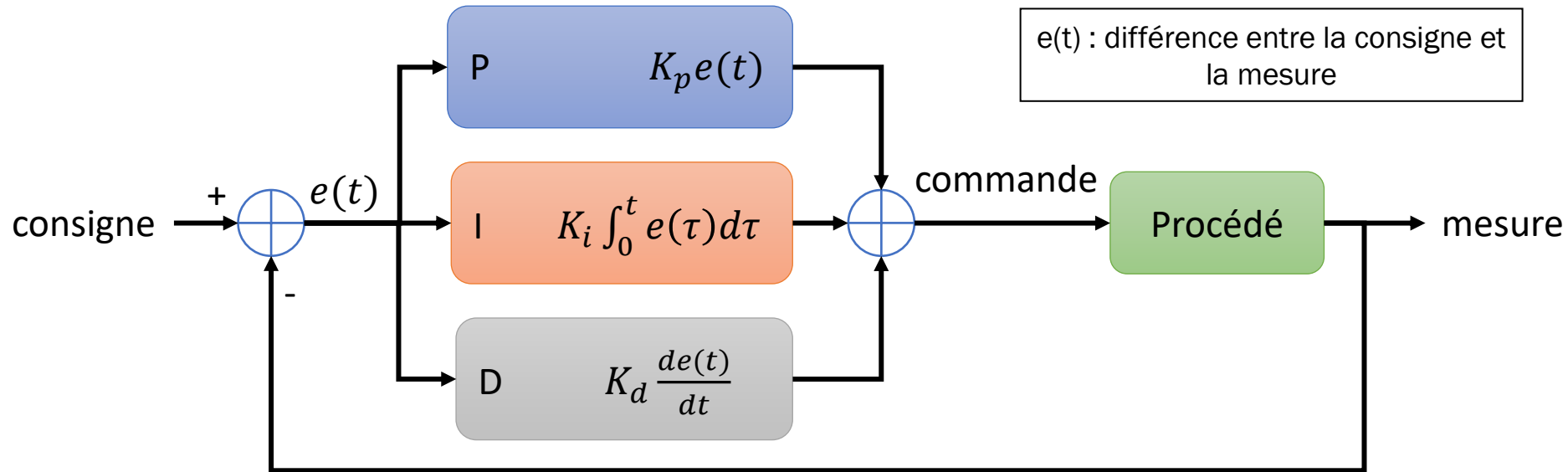
■ Points essentiels

- Déterminer la puissance de chauffage

■ Valeurs possibles

- Tout ou rien : 0 (éteint) ou 40% de la puissance maximale (allumé)
- PID : 0 à 100 %

Régulation : calcul du PID



- Terme proportionnel (P) : terme principal
- Terme Intégral (I) : accélère la convergence
- Terme Dérivé (D) : ralenti les changements du régulateur (important proche de la convergence)

Régulation : calcul du PID

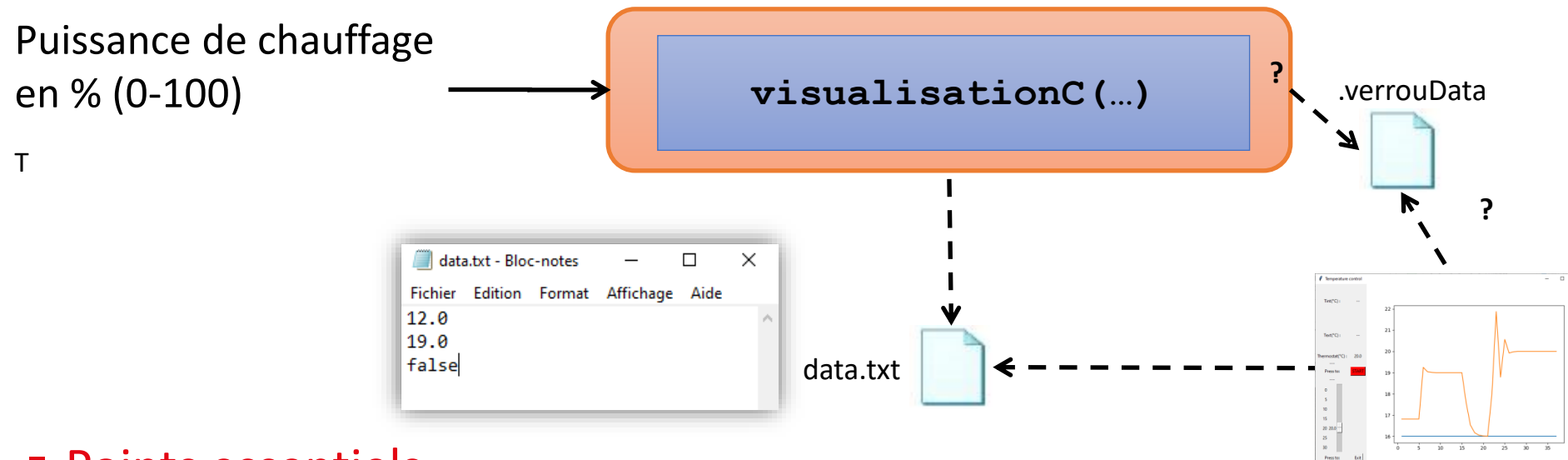
■ Points essentiels

- Calcul du PID
- Déterminer la puissance de chauffage

■ Points critiques

- Calcul du terme intégral et dérivé
- Lien entre le PID et la puissance de chauffage : loi + saturation
- Gains optimisant la régulation (K_P , K_I , K_D)

Visualisation du témoin de chauffe



■ Points essentiels

- Ecriture des informations dans le fichier data.txt
- Si chauffage==0 : false
- Si chauffage>0 : true

■ Points critiques

- Exclusion mutuelle à l'aide d'un verrou

Relevé de températures



```
typedef struct {  
    float interieure;  
    float exterieure;  
} temp_t;
```

■ Points essentiels

- Lire les températures en différenciant intérieur et extérieur

■ Points critiques

- Etablir la synchronisation afin de récupérer les valeurs
- Ne pas perdre la synchronisation
- Représentation des données (décimal/binaire)

Relevé de températures

$$\text{Température absolue en } ^\circ\text{C} = -39,64 + 0,04 \times \text{SOT}$$

Température extérieure	Octet	Bit							
		7	6	5	4	3	2	1	0
	1 ^{er}	0	0	0	0	SOT bit 11	SOT bit 10	SOT bit 9	SOT bit 8
	2 nd	0	0	0	1	SOT bit 7	SOT bit 6	SOT bit 5	SOT bit 4
	3 ^{ème}	0	0	1	0	SOT bit 3	SOT bit 2	SOT bit 1	SOT bit 0

Température intérieure	Octet	Bit							
		7	6	5	4	3	2	1	0
	1 ^{er}	1	0	0	0	SOT bit 11	SOT bit 10	SOT bit 9	SOT bit 8
	2 nd	1	0	0	1	SOT bit 7	SOT bit 6	SOT bit 5	SOT bit 4
	3 ^{ème}	1	0	1	0	SOT bit 3	SOT bit 2	SOT bit 1	SOT bit 0

1 échantillon ($T_{\text{INT}} + T_{\text{EXT}}$)
toutes les 4 ms (représente
10 s en réalité)

Commande de chauffage



Commande chauffage	Bit							
	7	6	5	4	3	2	1	0
	0 = chauffage 1 = froid	PUIS bit 6	PUIS bit 5	PUIS bit 4	PUIS bit 3	PUIS bit 2	PUIS bit 1	PUIS bit 0

$$\text{puissance en \%} = (\text{PUIS} / 127) \times 100$$

■ Points essentiels

- Envoyer la puissance de chauffage

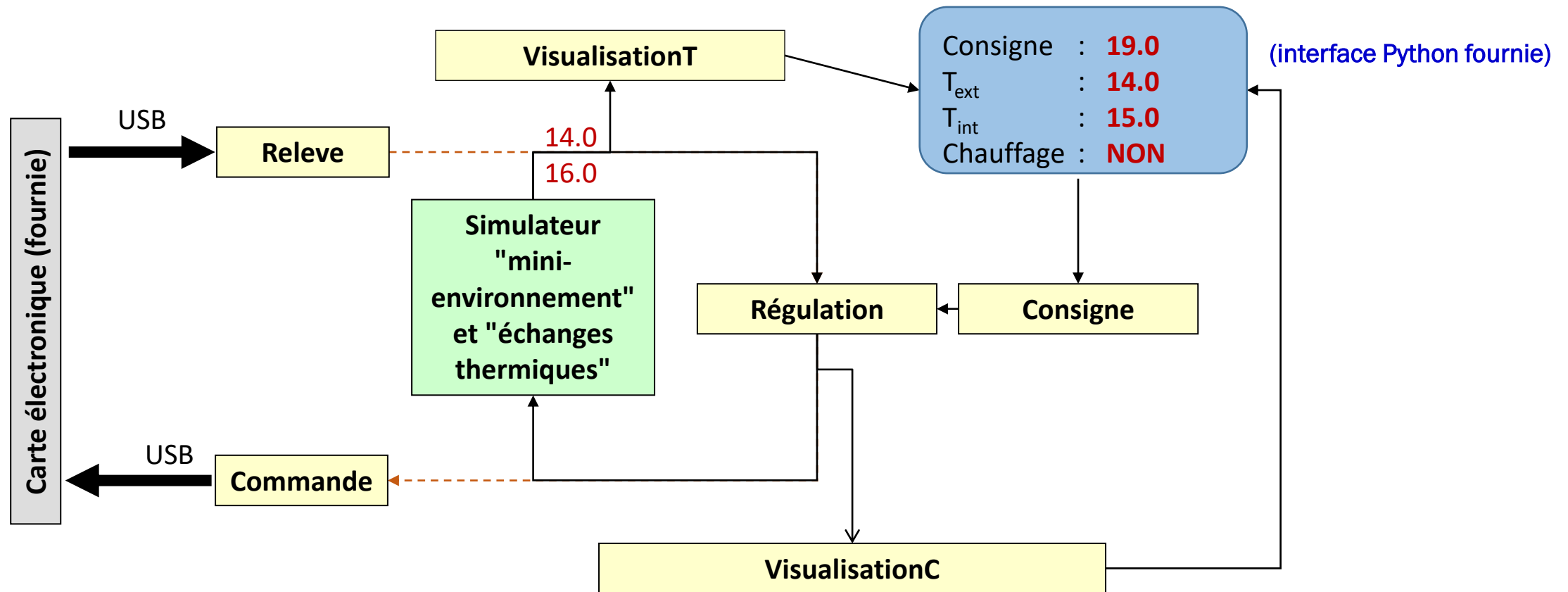
■ Points critiques

- Représentation des données (décimal/binaire)

Architecture



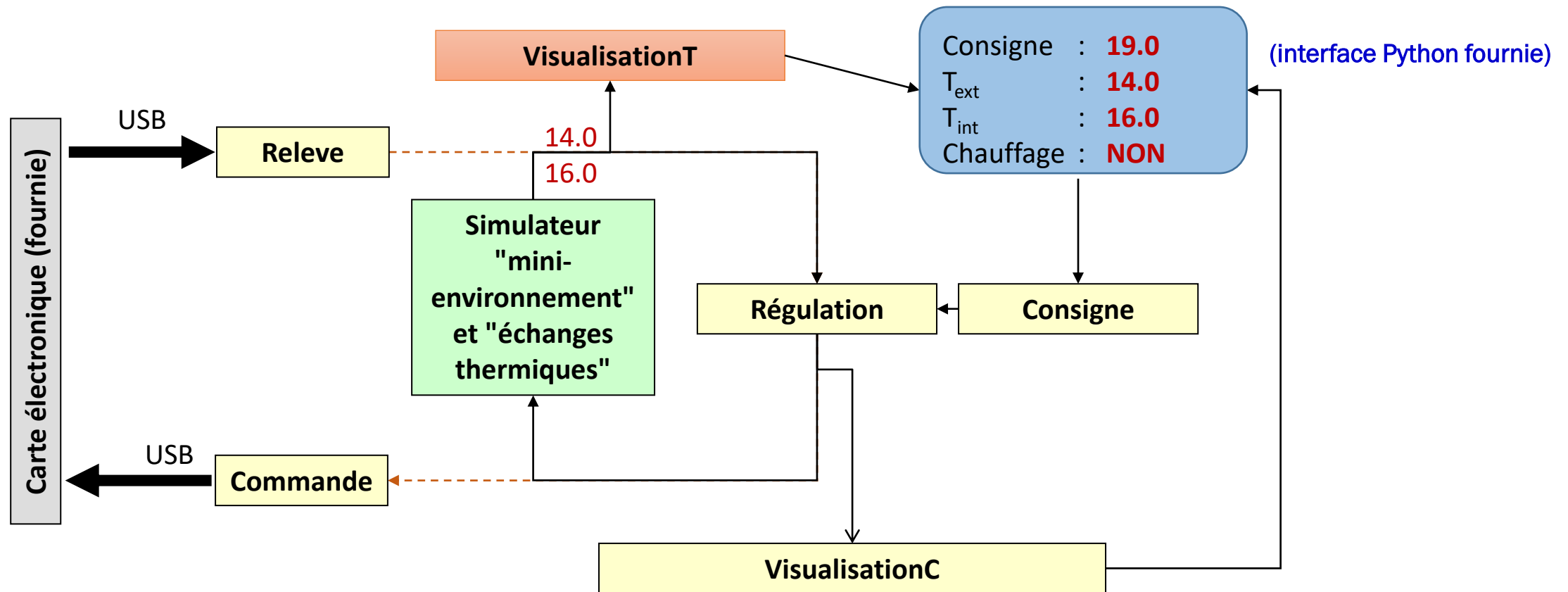
Synoptique de la partie informatique



Modules à développer

Module fourni

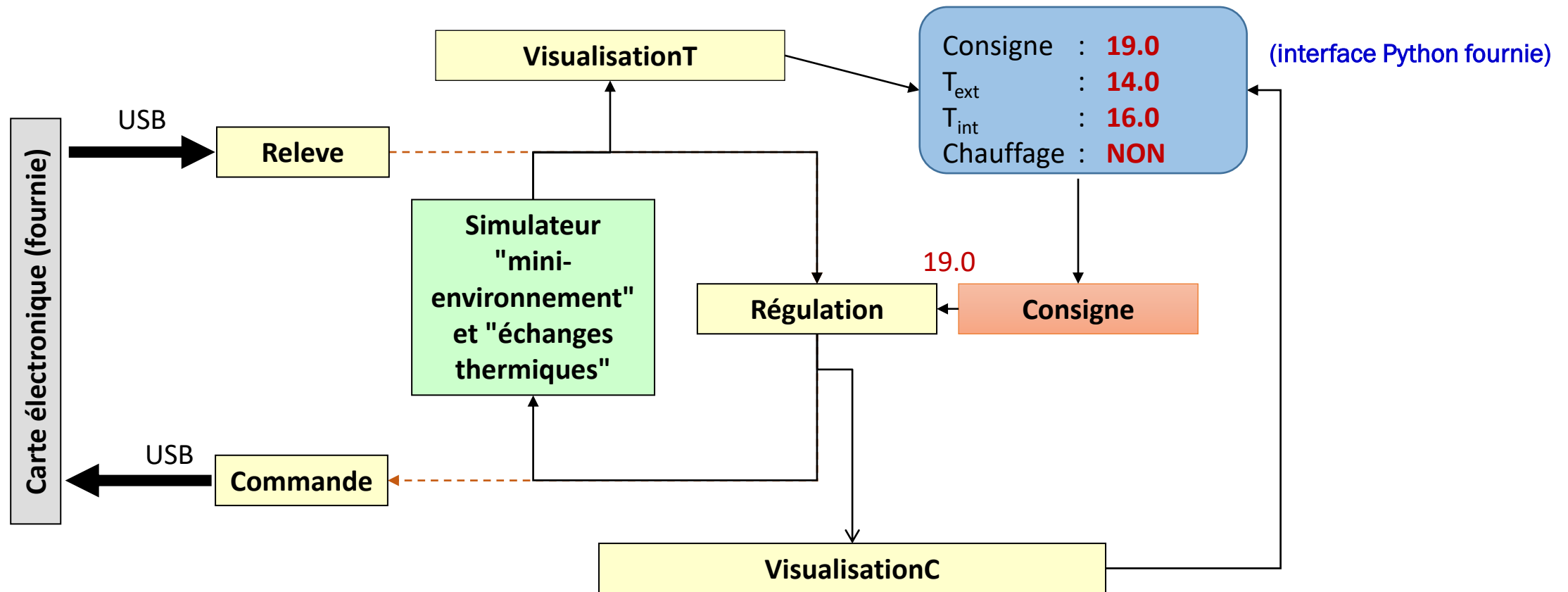
Synoptique de la partie informatique



Modules à développer

Module fourni

Synoptique de la partie informatique

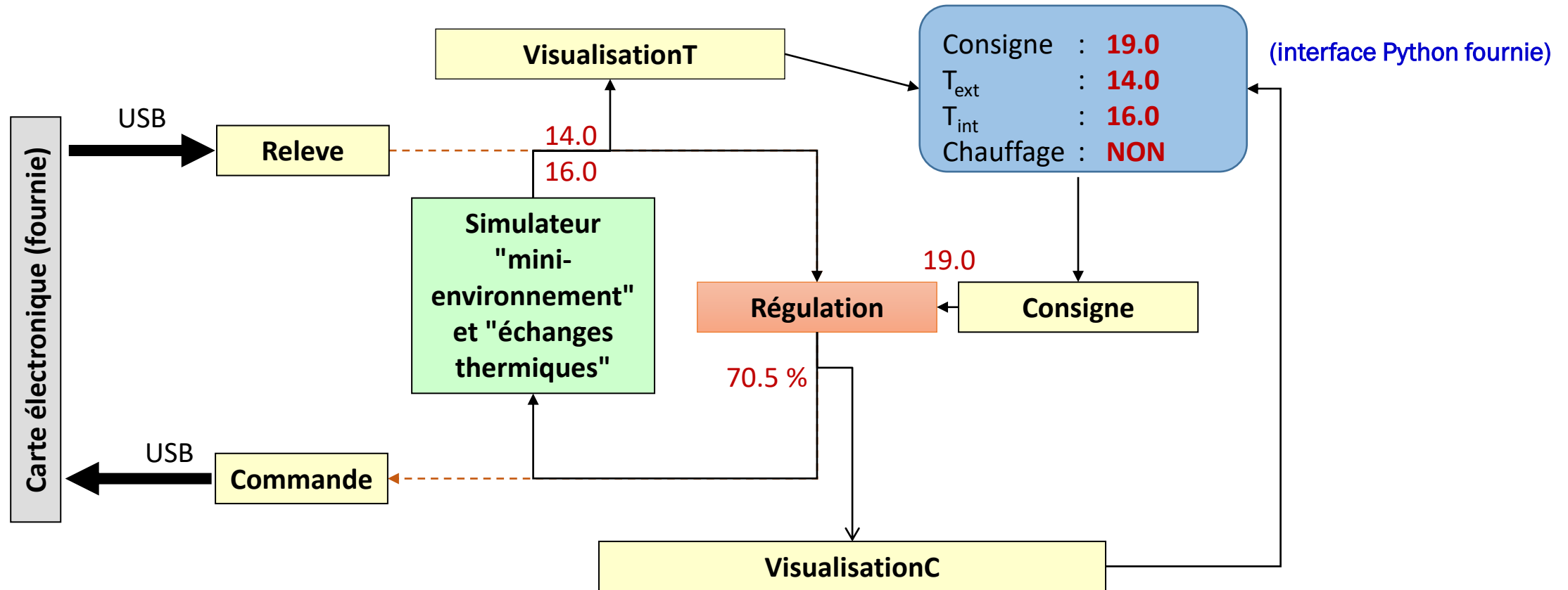


Modules à développer



Module fourni

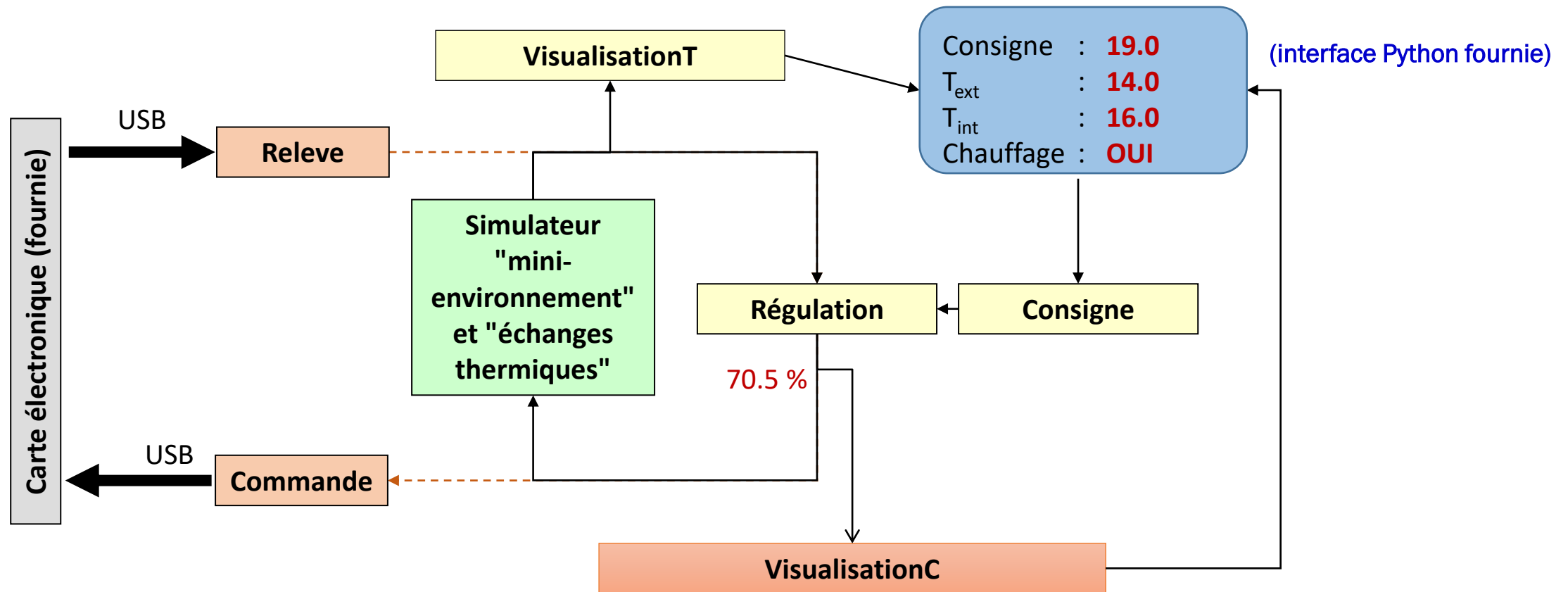
Synoptique de la partie informatique



Modules à développer

Module fourni

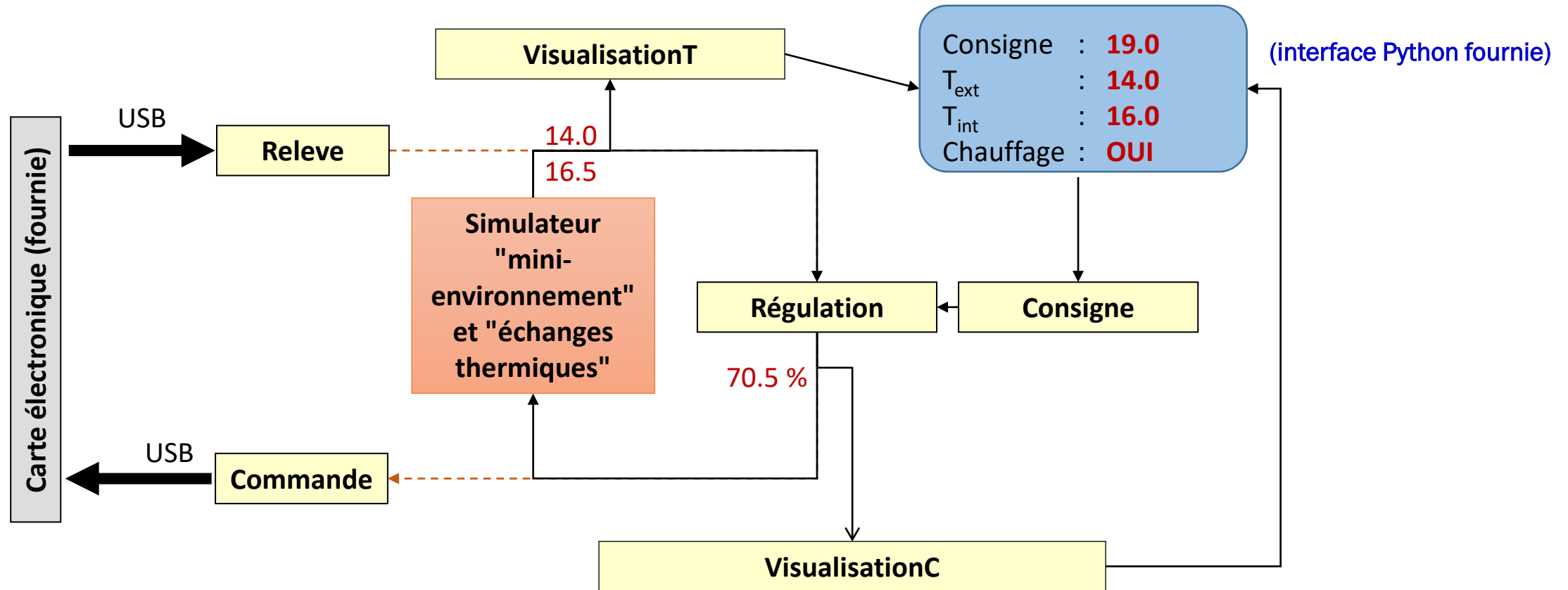
Synoptique de la partie informatique



Modules à développer

Module fourni

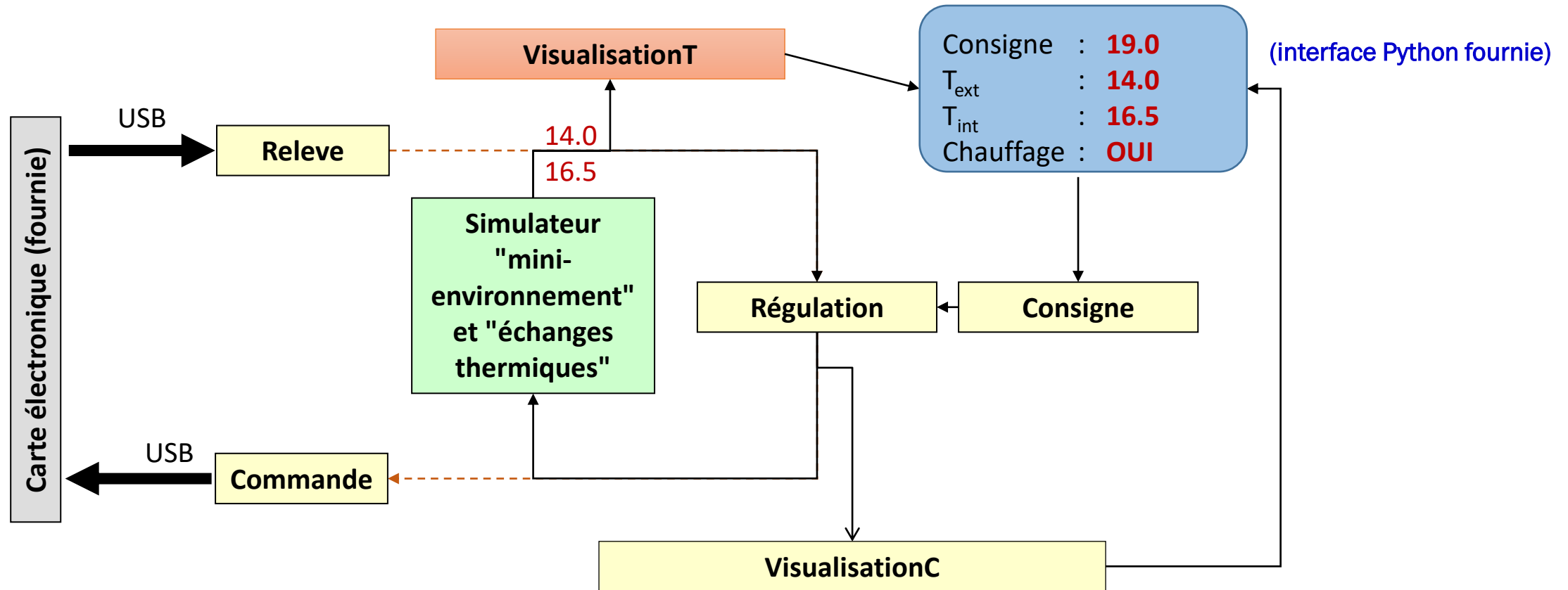
Synoptique de la partie informatique



Modules à développer

Module fourni

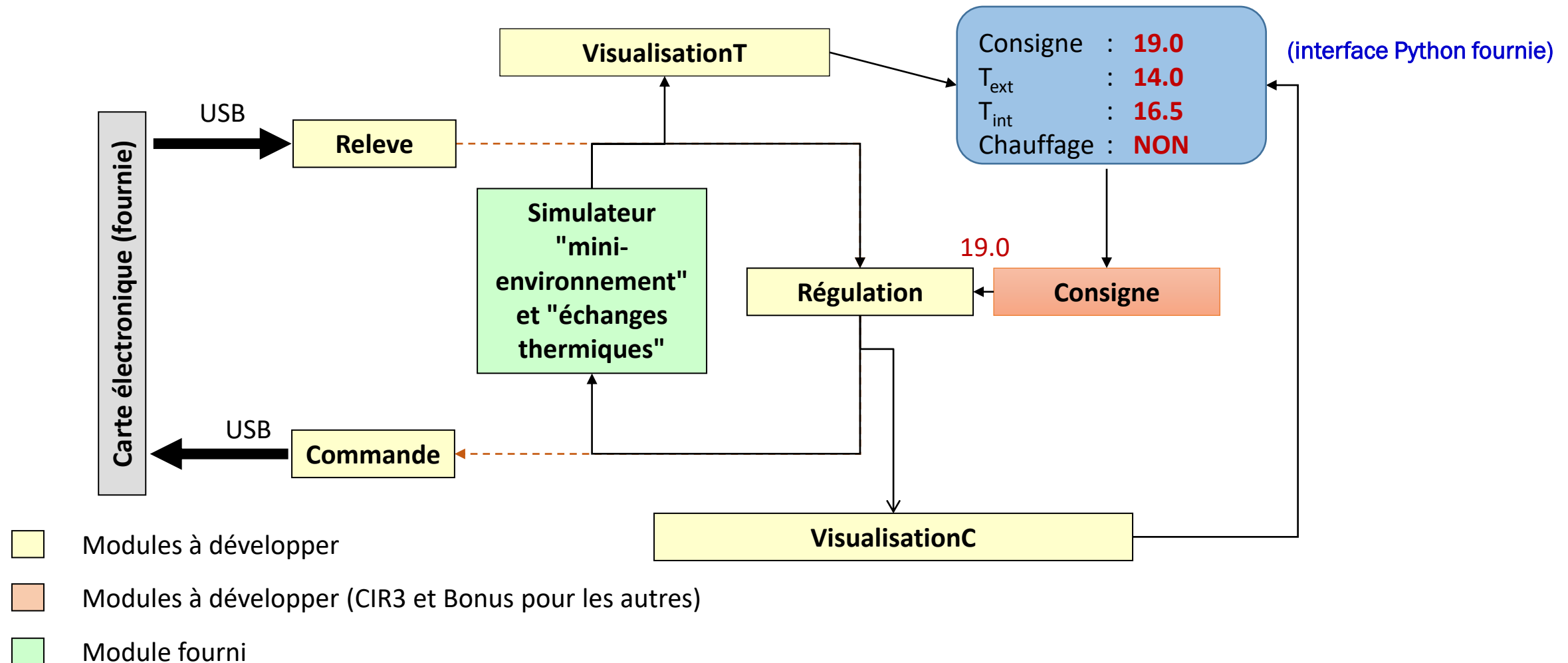
Synoptique de la partie informatique



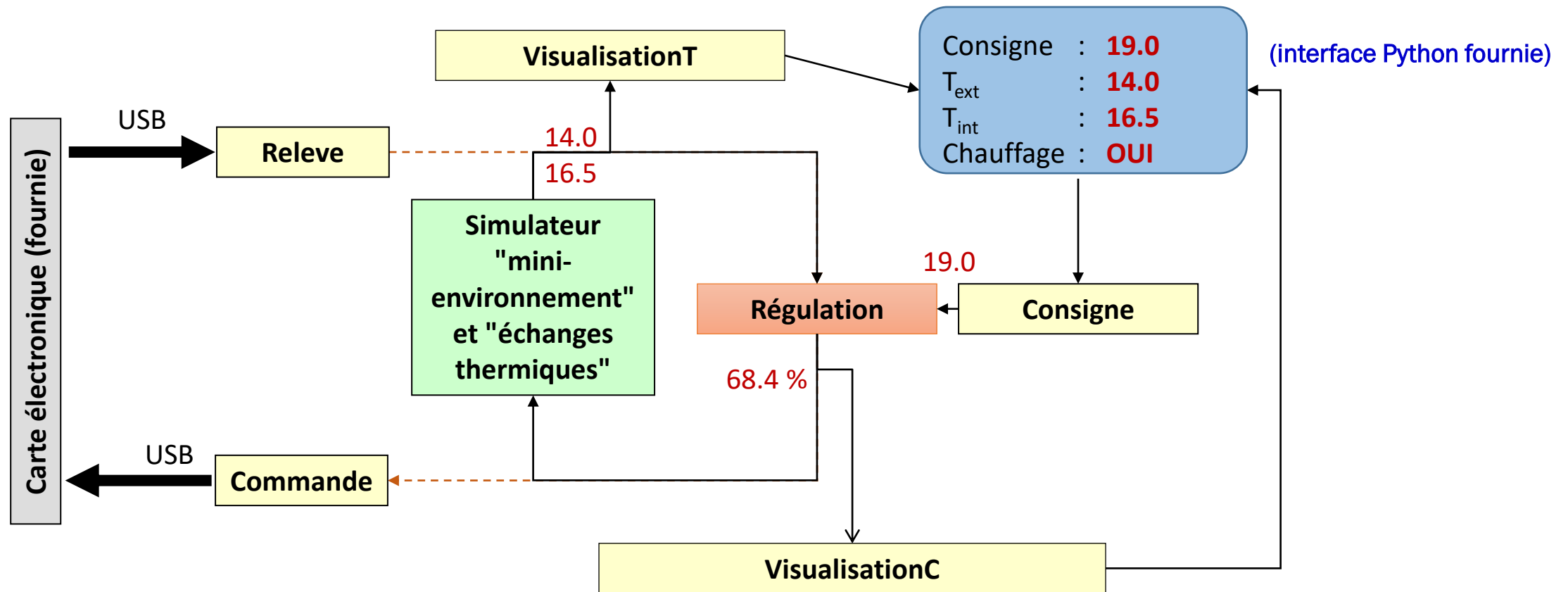
Modules à développer

Module fourni

Synoptique de la partie informatique



Synoptique de la partie informatique



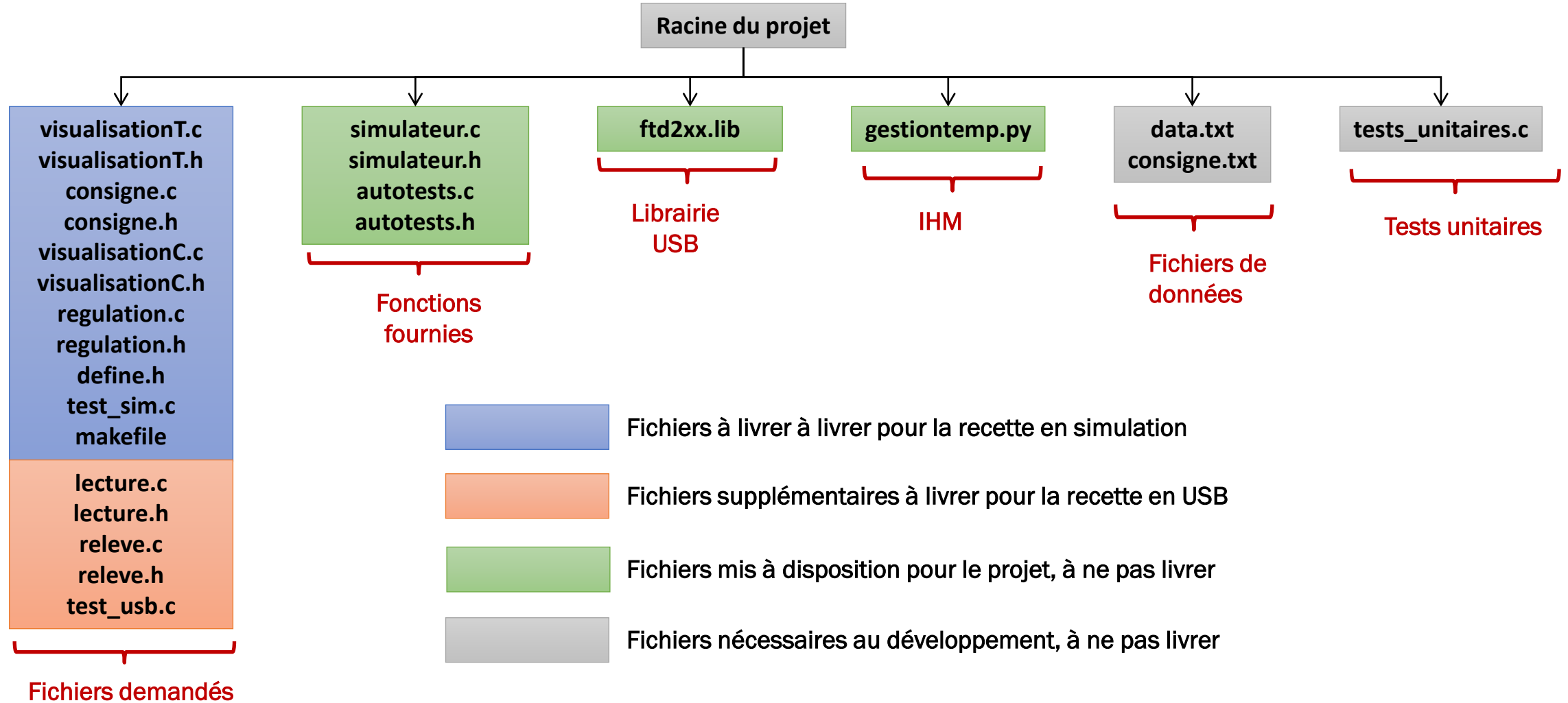
Modules à développer

Module fourni

Convention de nommage

Bloc	Nom fonction	Nom fichier source	Nom fichier header
Relevé des températures	releve	releve.c	releve.h
VisualisationT	visualisationT	visualisationT.c	visualisationT.h
Consigne	consigne	consigne.c	consigne.h
Régulation	regulation	regulation.c	regulation.h
VisualisationC	visualisationC	visualisationC.c	visualisationC.h
Commande de chauffage	commande	commande.c	commande.h
Simulateur	simulateur	simulateur.c	simulateur.h
Définitions globales			define.h

Arborescence



Déroulé du projet

4 phases



Lundi matin	Lundi après-midi	Mardi matin	Mardi après-midi	Mercredi matin	Mercredi après-midi	Vendredi matin	Vendredi après-midi
Analyse	Réalisation				Intégration		Recette

- ✓ Analyse de chaque bloc
- ✓ Architecture du programme
- ✓ Définition des Types de données

- ✓ Code source de chaque bloc
- ✓ Tests unitaires
- ✓ Makefile

- ✓ Code source programme principal
- ✓ Test intégration avec le simulateur ou avec la carte
- ✓ Validation avec l'IHM

- ✓ **Code source à rendre avant 15:00**
- ✓ QCM de 15:00 à 15:30
- ✓ Validation automatique et manuelle par l'enseignant sur PC enseignant à partir de 15:30
- ✓ Qualimétrie

Au jour le jour

■ Travail en binôme

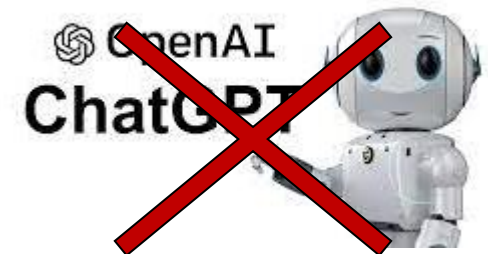
- Les binômes sont constitués le premier jour
 - Les binômes à constituer au sein de chaque sous-groupe
 - Les redoublants doivent se mettre ensemble ou en monôme le cas échéant
- Chaque étudiant connaît l'ensemble du projet
- Attention à bien se répartir le travail

■ Ressources externes

- Tous les documents sont autorisés...
 - ... mais les informations nécessaires au projet sont fournies !!!
- La transmission de code entre étudiants de binômes différentes est interdits
 - (un outil anti-plagiat sera utilisé en fin projet)
- L'utilisation d'outils de génération de codes par intelligence artificielle de type ChatGPT est totalement proscrit et sera sévèrement sanctionné

■ Livraison de code ou de document

- Ne pas attendre la dernière minute pour poster le livrable
 - Préparer des livrables intermédiaires
 - Sauvegarder régulièrement vos données

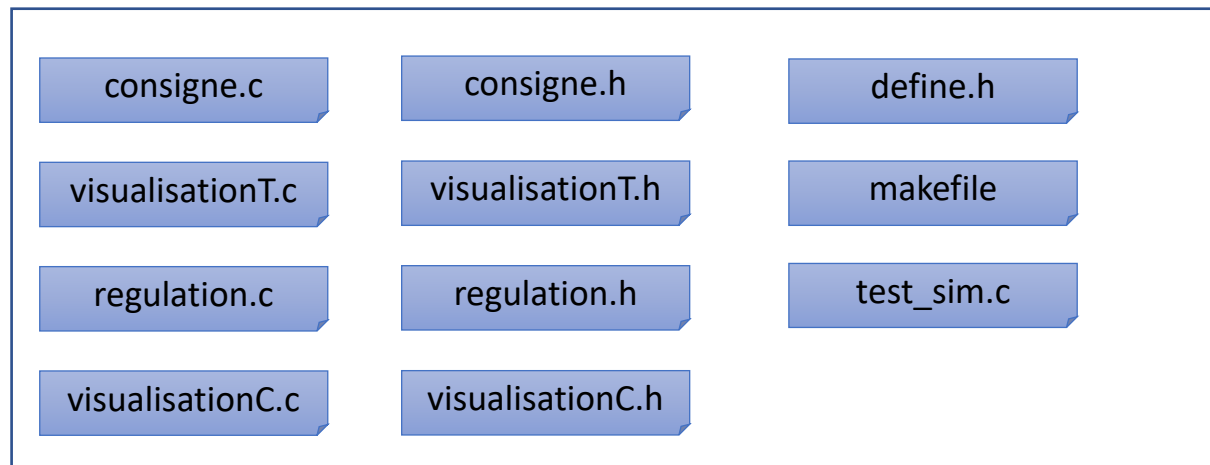


Notation



Ce qu'il faut rendre

■ Fichiers sources à rendre (au minimum)



sources_xx_yy.zip

Pour vous aider, une coquille vide comprenant l'architecture imposée est disponible sur l'ENT

■ Fichiers sources supplémentaires pour tests USB

- Fichiers précédents
- releve.c
- releve.h
- commande.c
- comande.h
- test_usb.c

Barème final (indicatif)

	Coefficient
Recette programme	3
Audit code	1
QCM	1
Total	5

Barème recette (indicatif)

#	Nom du test	Type de test	A3 hors CIR3	CIR3
1	Visualisation températures	Automatique	2,5	2
2	Lecture consigne	Automatique	2,5	2
3	Visualisation Commande	Automatique	2,5	2
4	Régulation ToR	Automatique	2,5	2
5	Régulation PID	Automatique	4	4
6	Programme global en simulation	Manuel	4	4
7	Releve	Manuel	1	1
8	Commande	Manuel	1	1
9	Programme global USB	Manuel	bonus	2



- **Aucun point donné au regard du code source (l'évaluateur ne rentre pas dans le code)**
- **Seulement sont testées les fonctionnalités d'exécution de votre code**

Tests automatiques

- Nous vous mettons à disposition un fichier de fonctions (autotests.c/autotests.h) permettant de tester unitairement les différents blocs :
 - testVisualisationT()
 - testConsigne()
 - testVisualisationC()
 - testRegulationTOR()
 - testRegulationPID()

```
----- Auto tests results: -----  
testVisualisationT      : score=100 %  
testConsigne            : score=100 %  
testVisualisationC      : score=100 %  
testRegulationTOR       : score=100 %  
testRegulationPID       : score=100 %
```

Les ressources



Le matériel

■ Votre PC portable

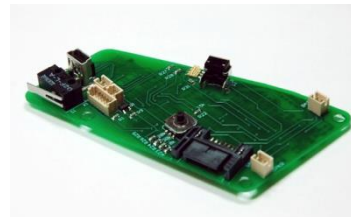
- Analyse
- Réalisation
- Intégration



Casque / écouteur
audio déconseillés (consignes non
appliqués = pénalité) !

■ Carte électronique

- Validation de la partie USB

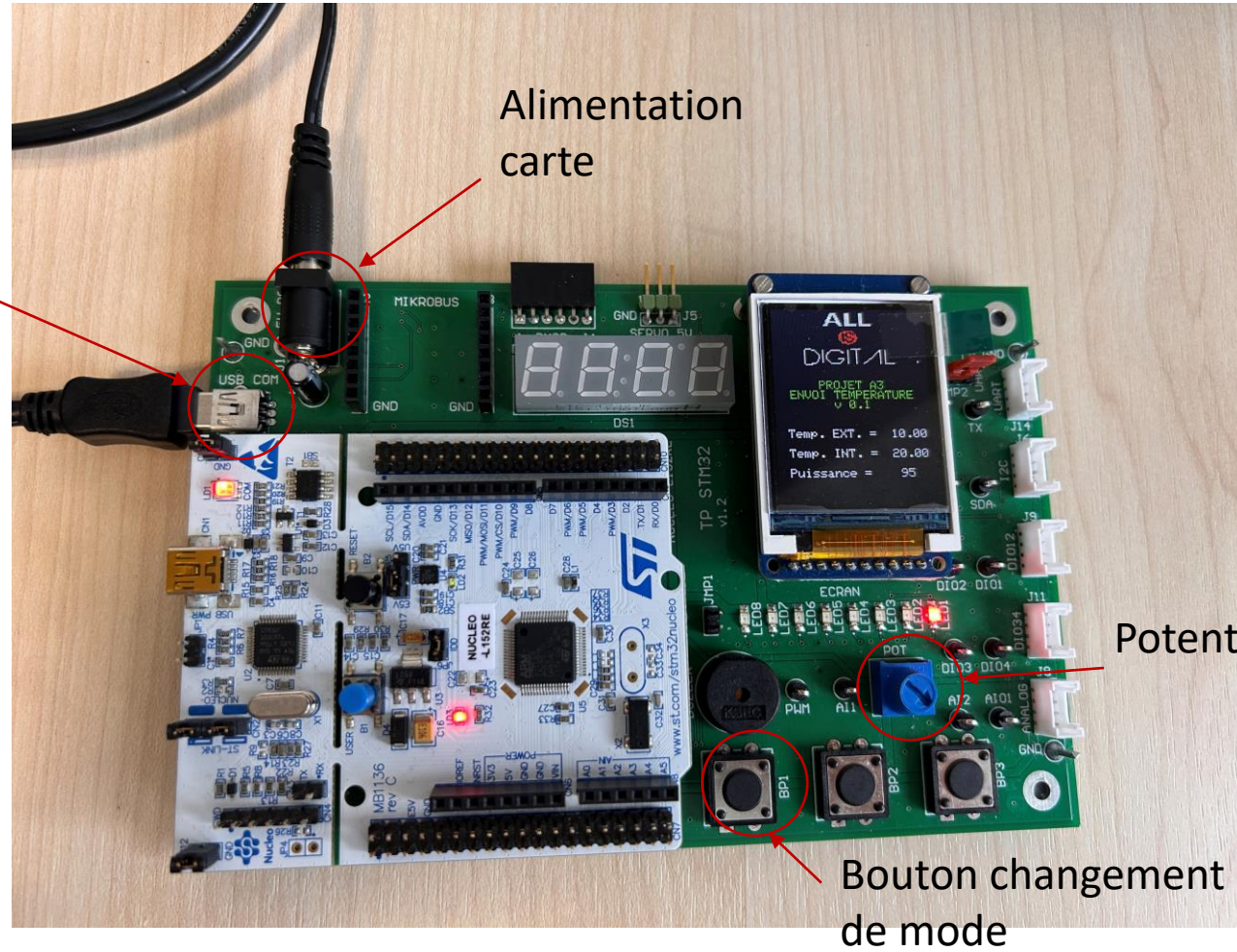


Boissons et nourriture
interdites !

Carte électronique

Communication
USB

Alimentation
carte



Potentiomètre

Bouton changement
de mode

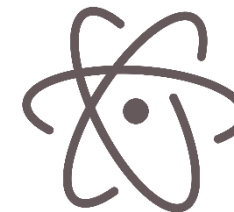
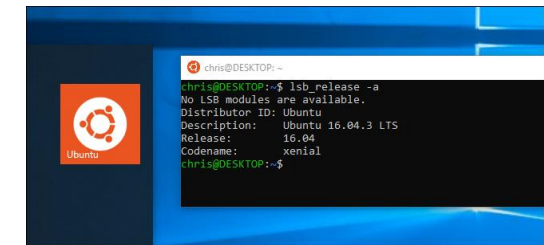
Outils de développement

■ Noyau de compilation

- Bash Windows
- MinGW (indispensable pour USB)
- Jupyterhub
- Linux natif

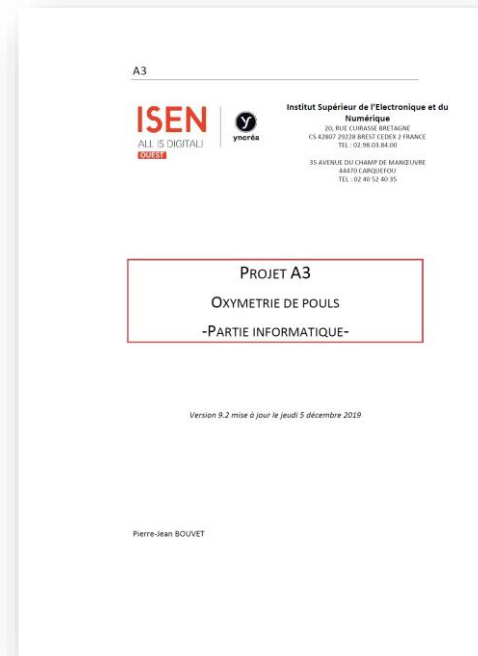
■ Editeur

- Notepad++
- Sublime Text
- Atom
- Visual studio code
- ...



Documentation

- Support de présentation
- Description détaillée
- Guide de programmation FTDI



MERCI
Des questions ?

