



# « Thermostat intelligent » Groupe 11

Membres : AMZUR Soufiane DAUBRY Wilson SERCU Stéphane

**VERSTRAETEN Denis** 

Superviseur: EPPE Stefan

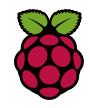
Lecteur : VERMEIR Aurélien

### Table des matières

- 1. Introduction
- 2. Conception
- 3. Thermostats auxiliaires
- 4. Thermostat central
- 5. Intelligence
- 6. Validation
- 7. Conclusion

### Introduction

- Demande en objets intelligents croissante
- Création du thermostat connecté
  - Utilisation simple et intuitive pour l'utilisateur
- Périphériques
  - Central: Raspberry Pi Model 2b
  - Auxiliaires : Arduino Nano V2

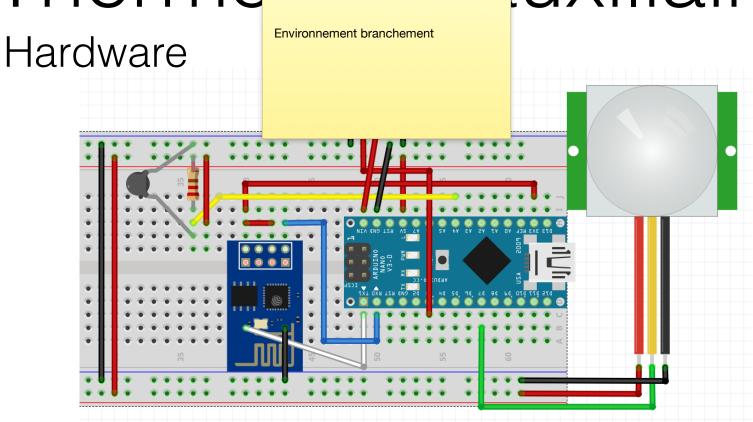




## Conception

- Diagramme complet
- Vue globale et structurée
- Partage efficace des différentes parties
- Développé dans la suite de la présentation

Thermostats auxiliaires

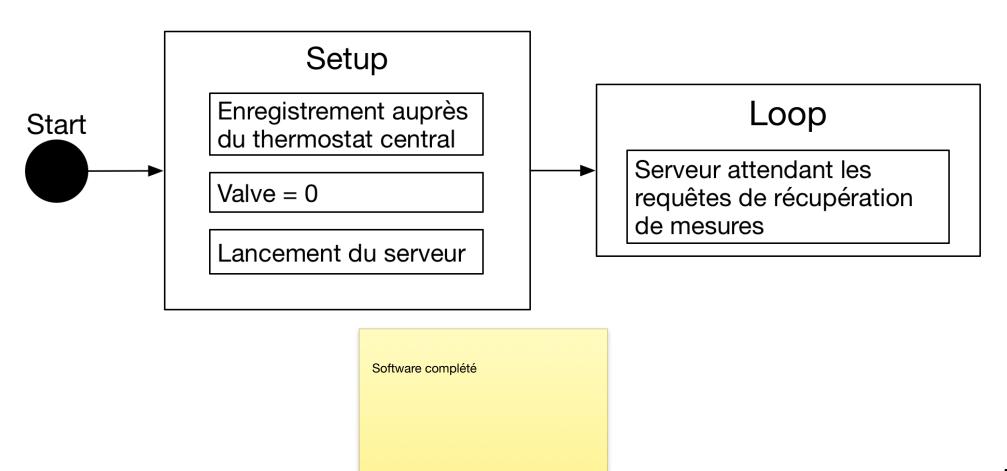


- Arduino Nano V2
- Thermistance

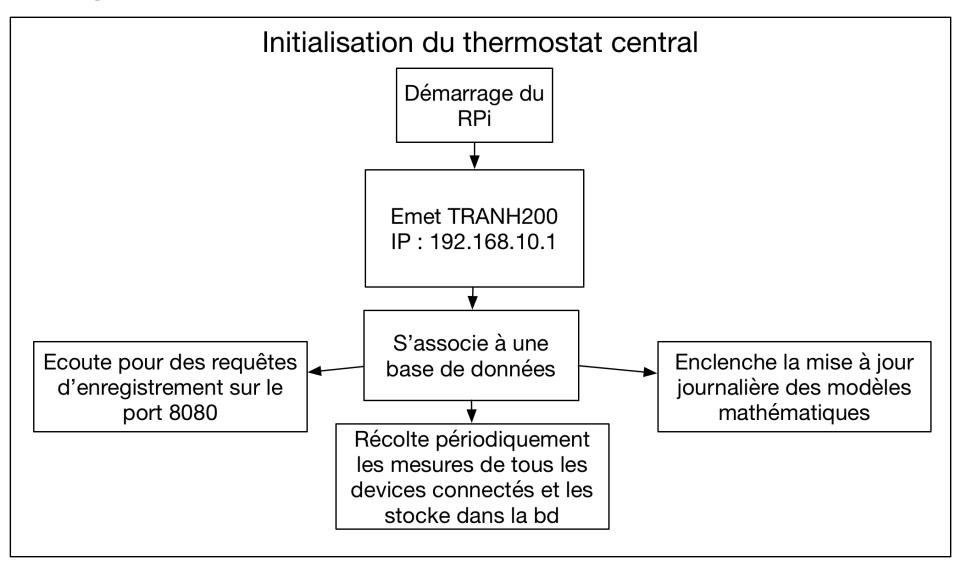
- Capteur de présence
- Module Wi-Fi
- Vanne thermostatique

### Thermostats auxiliaires

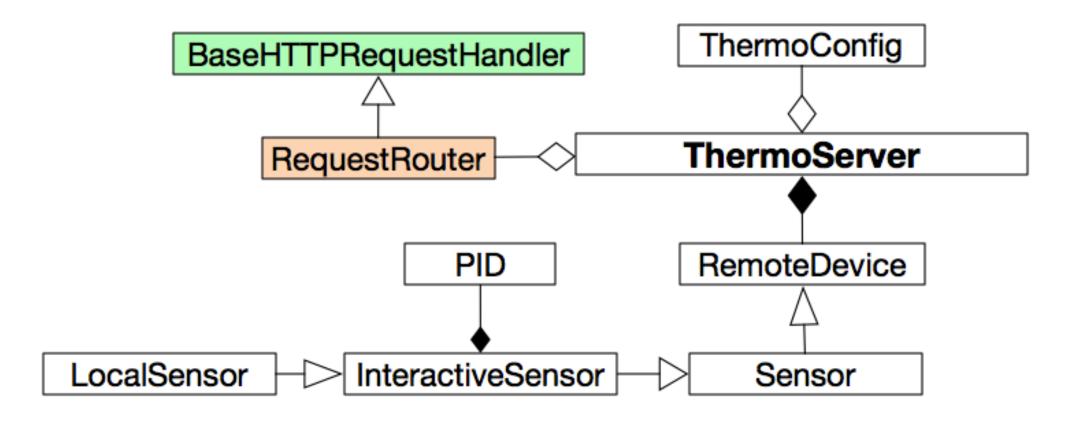
Software



#### Serveur



Serveur



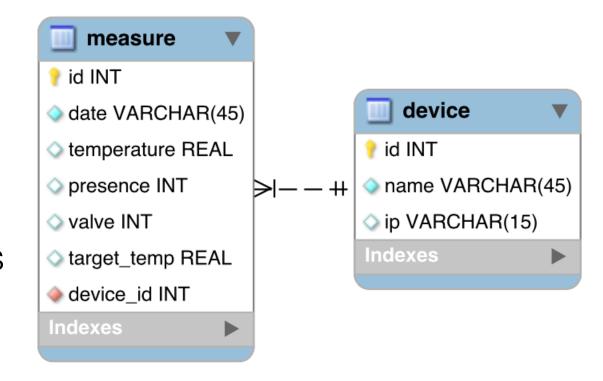
### Stockage des données

 Choix d'une base de données



 Local, par défaut dans Python





### Stockage des données Implémentation

#### **ThermoDB**

- + db\_path
- + measure\_table\_name
- + device\_table\_name
- + query\_handler
- + create\_tables(conn, device\_types)
- + test\_db\_creation()
- + test\_device\_getting\_data(TDB)
- + test\_measure\_getting\_data(TDB)
- + test devices measures saving(TDB)
- \_get\_cols\_from\_device\_types(device\_types)
- \_get\_table\_names(conn)
- \_get\_cols(conn, table\_name)

#### QueryHandler

- + db\_path
- + conn
- + get\_cursor()
- + insert(values, table\_name, commit=True)
- + select(columns, table\_name, cond, limit=-1, group\_by, order\_by)

#### ThermoMeasureHandler

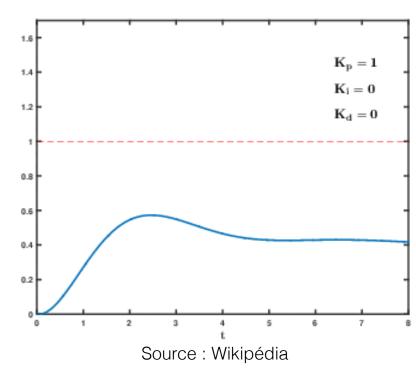
- + measure table name
- + device\_table\_name
- + device col names
- + measure\_col\_names
- + add\_measure(measures, device\_name)
- + save\_measure\_of\_all\_devices(measures)
- + get\_measure(where=1)
- + register\_device(device\_name, ip)
- + get\_device\_where(where)
- + get\_device\_by\_id(d\_id)

Méthodes spécifiques ajoutées au fur et à mesure

#### Contrôleur PID

- But : Atteindre et maintenir un température de manière optimale
- Influence des constantes

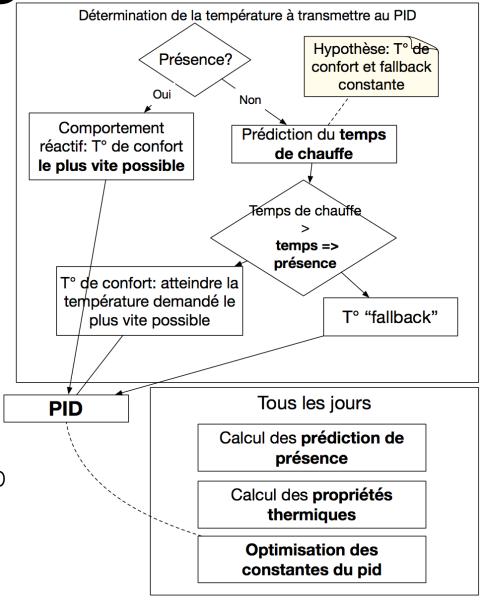
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$



- Critères d'intelligence :
  - Confort : atteindre et maintenir une température désirée
  - Optimisation énergétique : raccourcir les périodes de chauffe
- Modèles :
  - Prédiction de présence
  - Temps de chauffe
    - Caractéristiques thermiques

Pacharche des constantes du PID

Préciser que les prochains slides détailleront les modèles



### Prédiction de présence

- Fonction de probabilité
  - Interpolation polynomiale

$$p: \mathbb{R}^+ \to [0; 1]: t \mapsto At^5 + Bt^4 + Ct^3 + Dt^2 + Et + F$$

$$E(t_i) = \frac{n(t_i)}{N(t_i)}$$

Préciser un modèle par jour

$$\sum_{i=0}^{n} (p(t_i) - E(t_i))^2$$

### Caractéristiques thermiques

- Evolution de la température
  - Bilan énergétique

$$T(t) = T(t_0) + P_{max} \int_{t_0}^{t} PID(\tau)d\tau - \sigma \int_{t_0}^{t} (T(\tau) - T_{ext}(\tau)d\tau)d\tau$$

$$\begin{cases} \Delta T_{0,1} = P_{max} PID(t_0) - \sigma (T_0 - T_{ext,0}) \\ \Delta T_{1,2} = P_{max} PID(t_1) - \sigma (T_1 - T_{ext,1}) \end{cases}$$

#### Recherche des constantes du PID

 Minimisation de la différence entre la température actuelle et la température désirée

$$T(t_k) = T(t_0) + \int_{t_0}^{t_k} PID(t)dt$$

$$T(t_i) \approx T(t_{i-1}) + PID(t_{i-1}) * (t_i - t_{i-1})$$

$$\sum_{i=1}^{n} (T(t_{i-1}) + PID(t_{i-1}) * (t_i - t_{i-1}) - T_T(t_i))^2$$

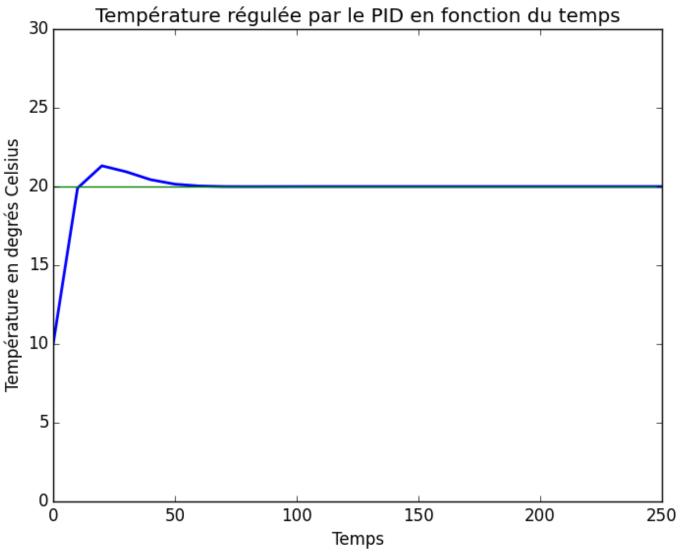
### Validation

Tests d'implémentation

- Tests du « thin thermostat »
- Tests unitaires des modules du serveur
- Tests généraux du serveur

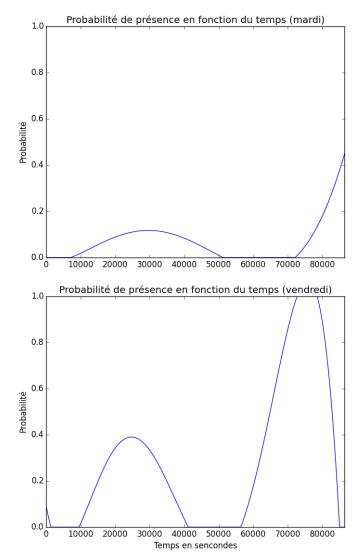
### Validation

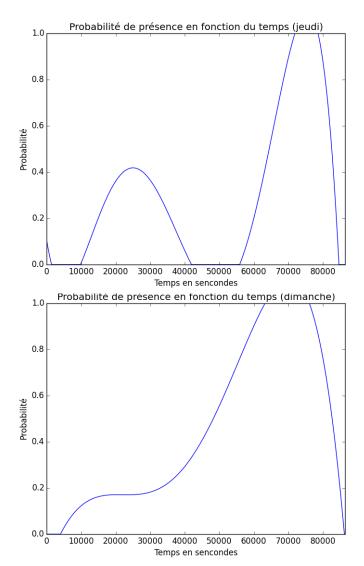
Régulation PID



### Validation

#### Prédiction de présence





### Conclusion

- Perspectives d'évolution
  - Corriger certains et affiner les modèles théoriques
  - Identification des Remote Devices
  - Ajouter d'autres types de senseurs
  - Affiner la notion de confort
  - Interface graphique, accès aux informations
  - Optimiser
- Réflexion
  - Pas de perte de temps sur le débogage

Merci pour votre attention

Bibliographie : cf.rapport intermédiaire

### Annexe

- Fonctionnement de groupe
  - Une réunion par semaine
  - OverLeaf
  - Dropbox
  - Skype
  - Facebook

#### Annexe ThermoConfig + cfg\_path BaseHTTPRequestHandler + port + db path + period ThermoServer + device\_types + cfa + DEFAULT\_VALUES + connected devices + refresh() RequestRouter reg handler thread \_parse\_port() \_data\_getter\_thread parse db name() + routes \_data\_analyse\_thread \_parse\_period() + do PUT() http server parse device types() + do\_GET() database + do\_POST() start\_reg\_handling() \_route(method, path) + register\_new\_device({'headers', 'content'}) \_send\_code(code) RemoteDevice + get\_devices\_measures() + ip + save devices measures() + name + magic\_function() + port + status() PID + sensors + run(req\_handling=True, data\_getting=True) + Kp + stop() + get\_measures() + get\_sensors\_by\_name() - \_build\_device\_of\_type(d\_type, ip, port, name) + Ki + Kd + Integral InteractiveSensor Sensor + target\_temp + valve + measure\_name + pid + temperature + ip + set\_option(value) + date + port + error(target\_temp, temperature) + get\_measure() + derivature(target\_temp, temperature, date) LocalSensor + integrate(target\_temp, temperature, date) + update(dic\_measure) + get\_measure() Module Utils + set\_constant(cst) + set\_option(value) Module http.server Module ThermoServer