

Module d'assistance et surveillance agricole (MASA)



Image non contractuelle

DOLOIR Julian
PRADEL Kellian
LARONDE-SERIN Dimitri

Pourquoi MASA ?

Mise en situation

Une personne souhaite pouvoir cultiver des fruits et des légumes sur son terrain. Il doit choisir les espèces les plus adaptées aux conditions climatiques du terrain.

Enjeu

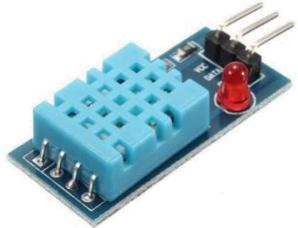
- Améliorer la productivité d'une parcelle cultivée
- Aider le cultivateur à surveiller les paramètres environnementaux
- Suggérer la meilleure sélection de plantes adaptées à l'environnement

Problématique

Comment recueillir et analyser les données mesurées sur une surface étendue par secteurs ?

Schéma de principe [1/2]

Thermomètre +
Humidité de l'air



Luxmètre



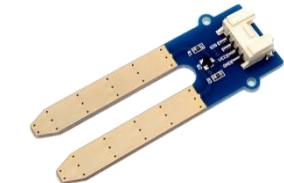
Anémomètre



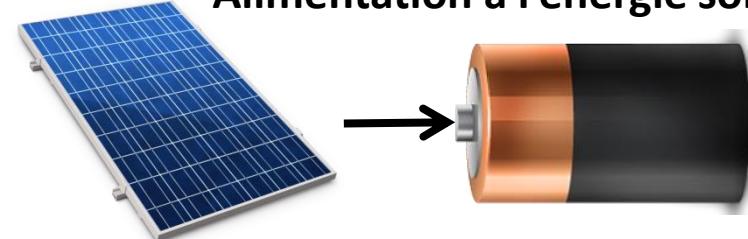
Pluviomètre



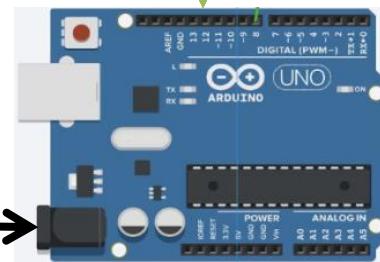
Humidité du sol



Alimentation à l'énergie solaire



Données entrants



LoRa

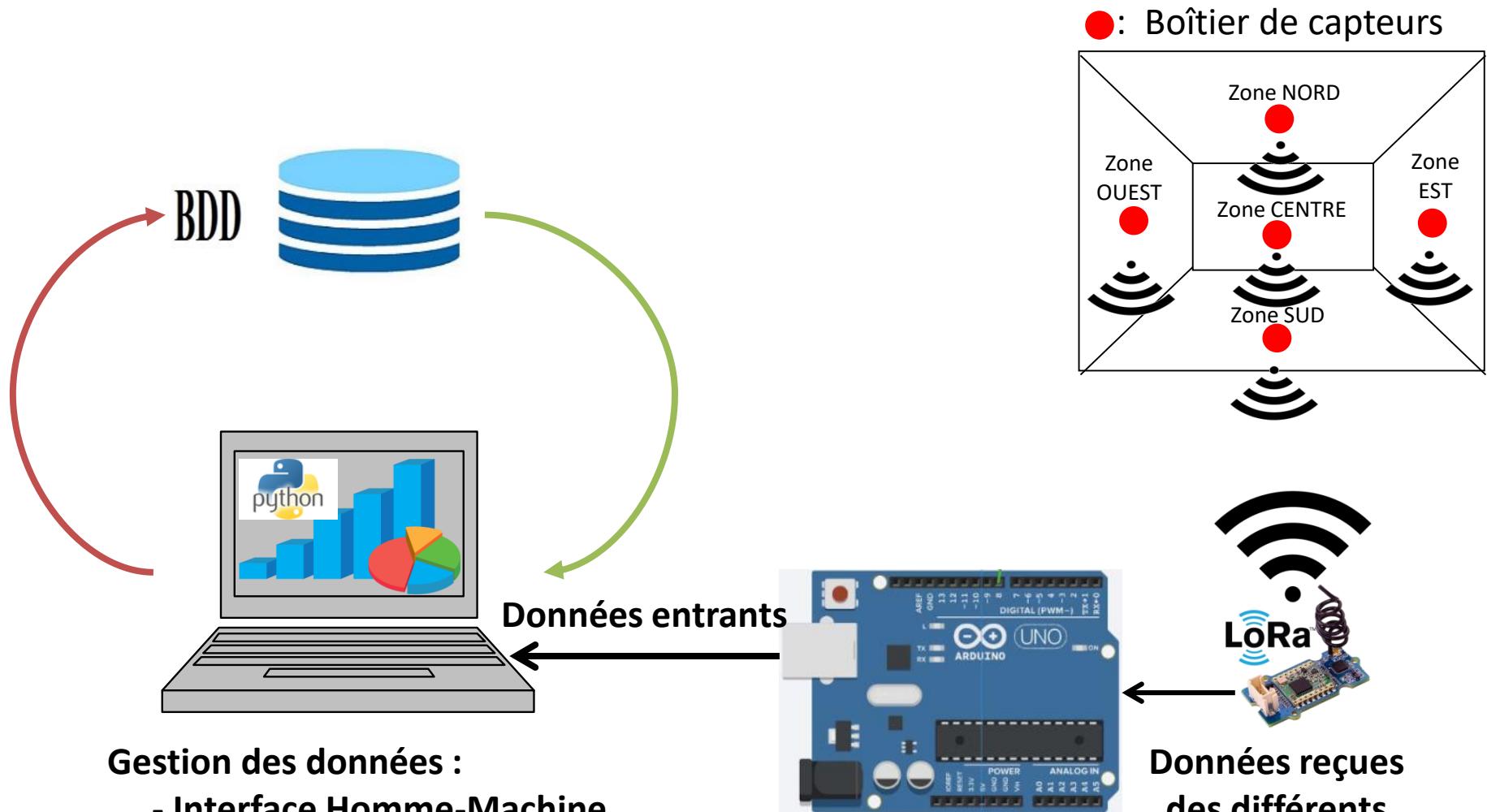


Envoie des
données
recueillis



Base
réceptrice

Schéma de principe [2/2]



MASA MOODBOARD



Répartition des tâches

Kellian

- Design et Gestion de l'énergie

Julian

- Programmation, Gestion des données et Communication

Dimitri

- Capteurs

Diagramme UseCase

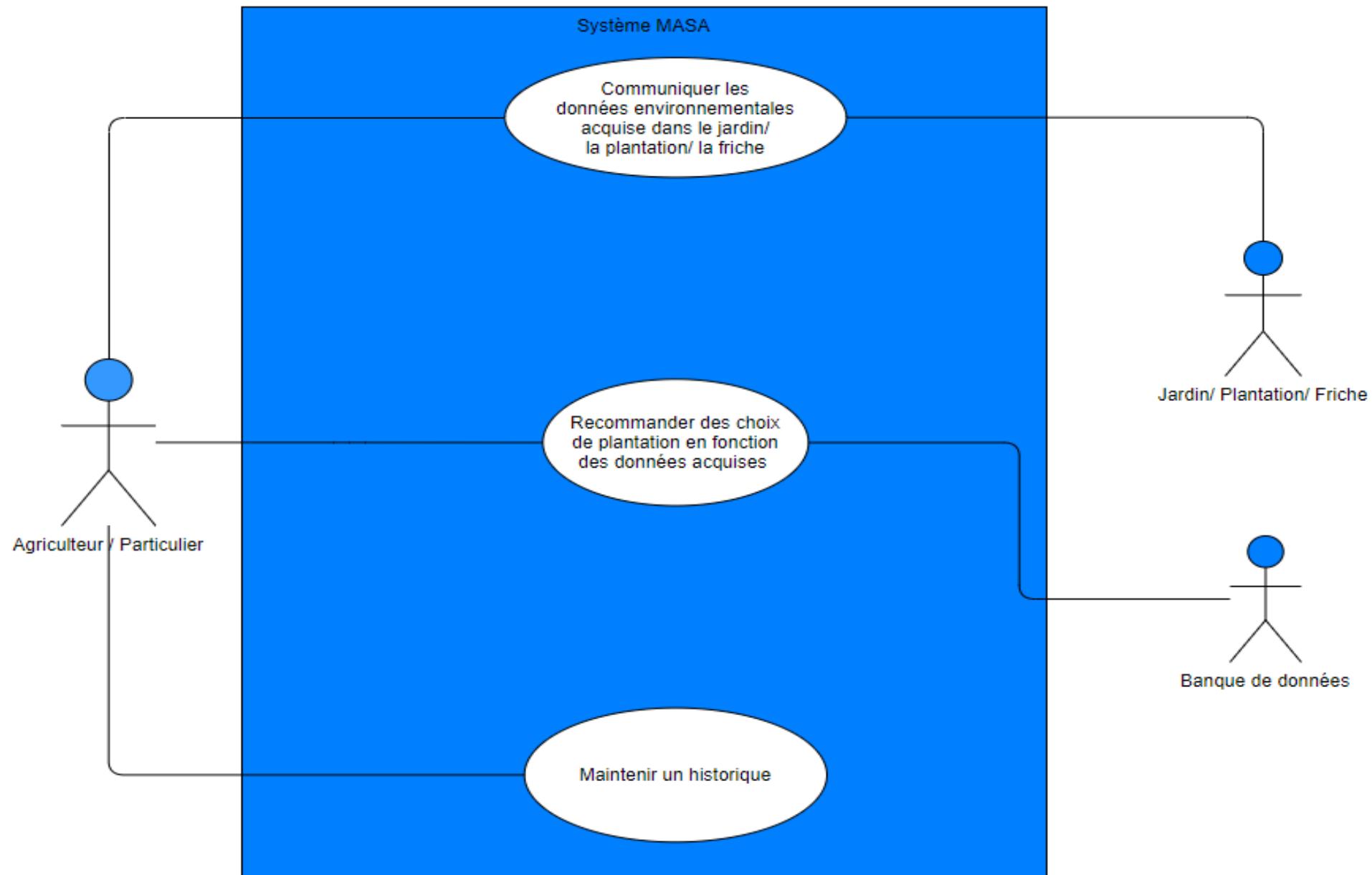


Diagramme SysML d'exigence [1/2]

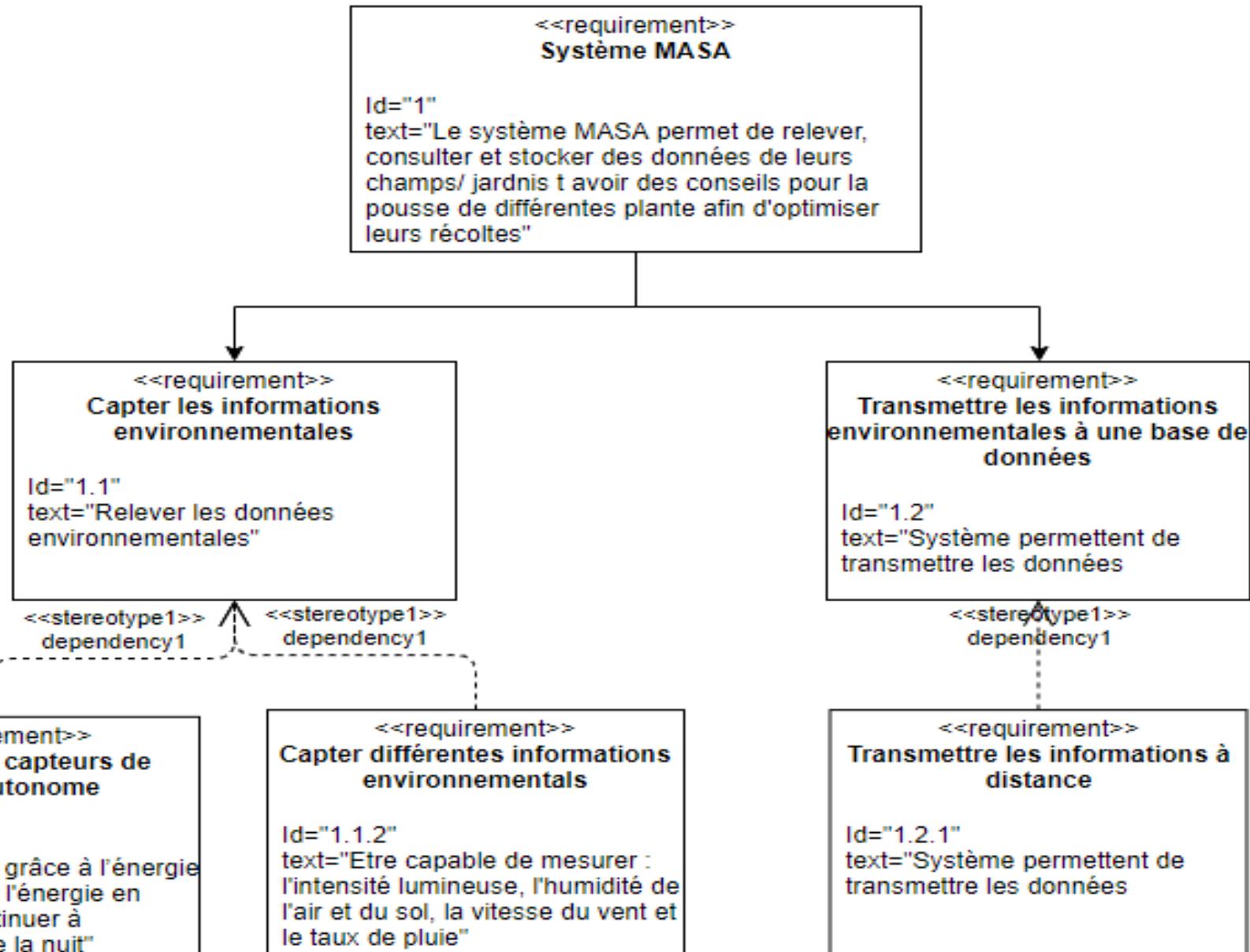


Diagramme SysML d'exigence [2/2]

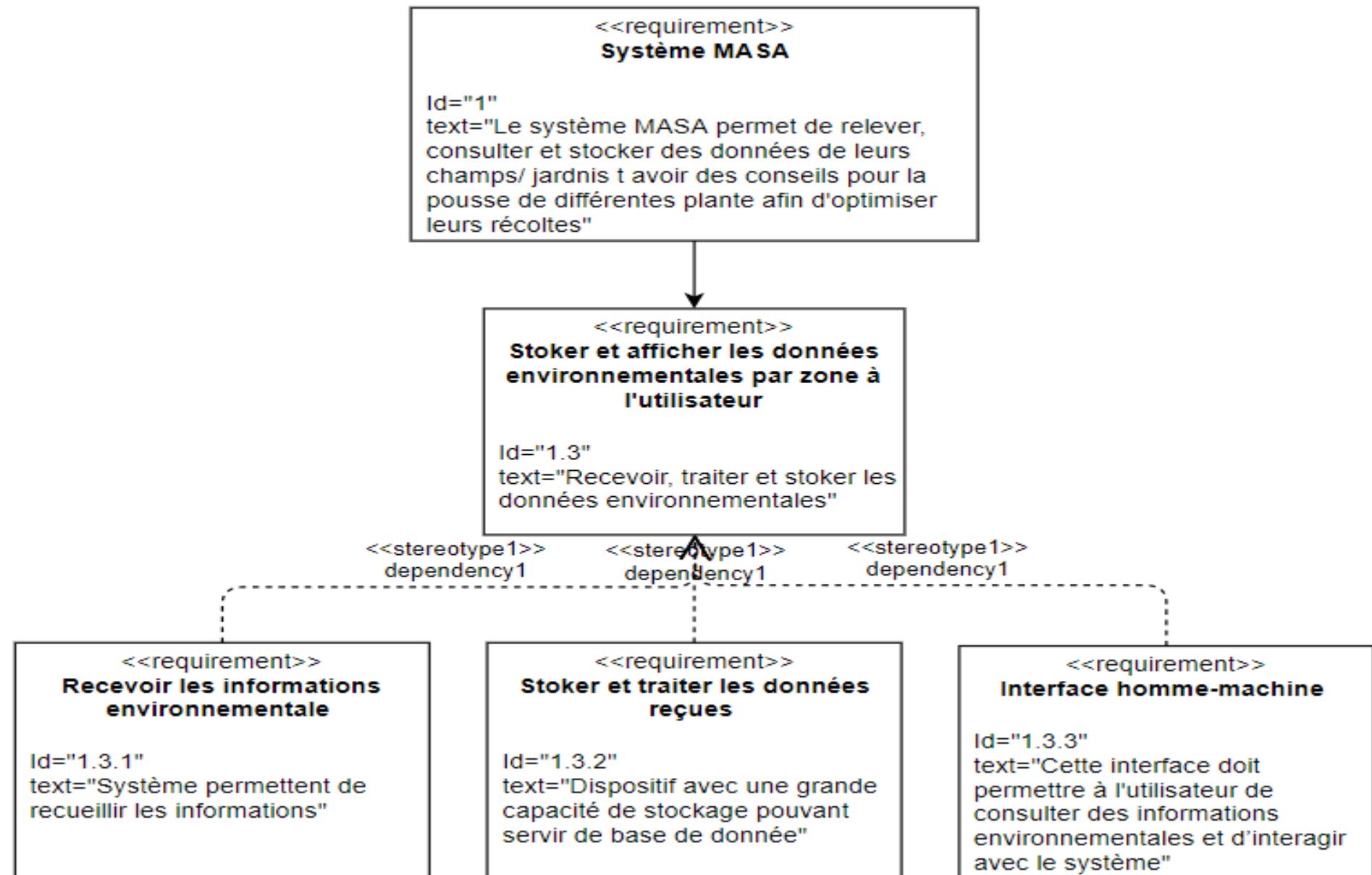
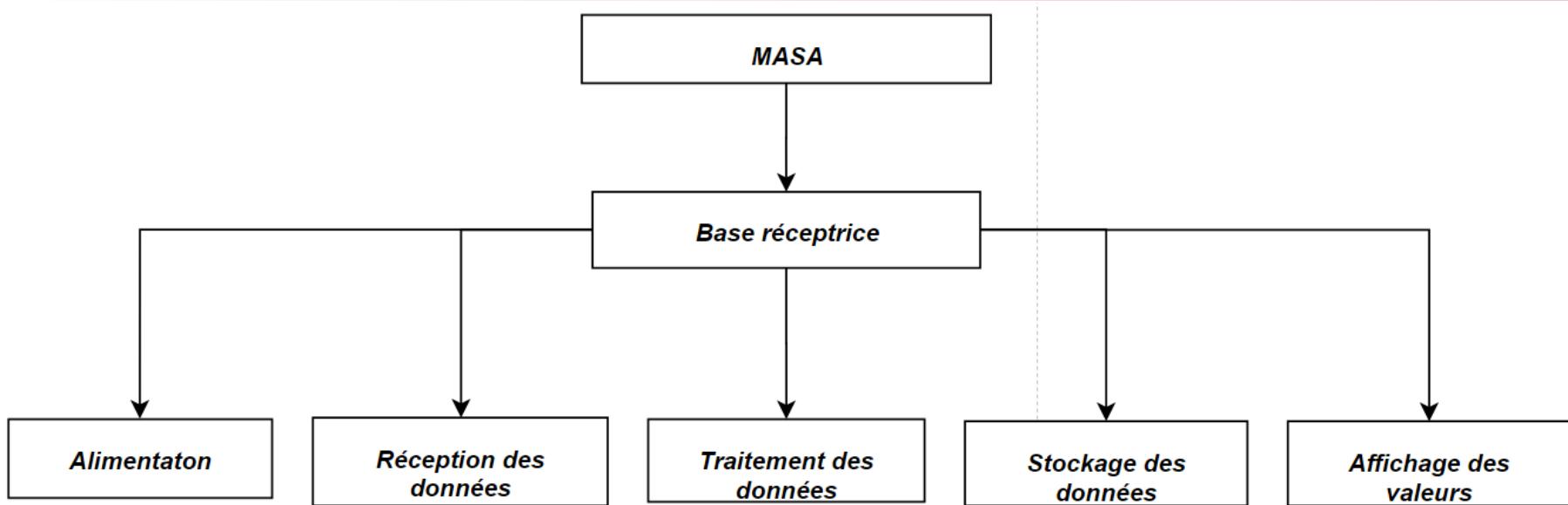
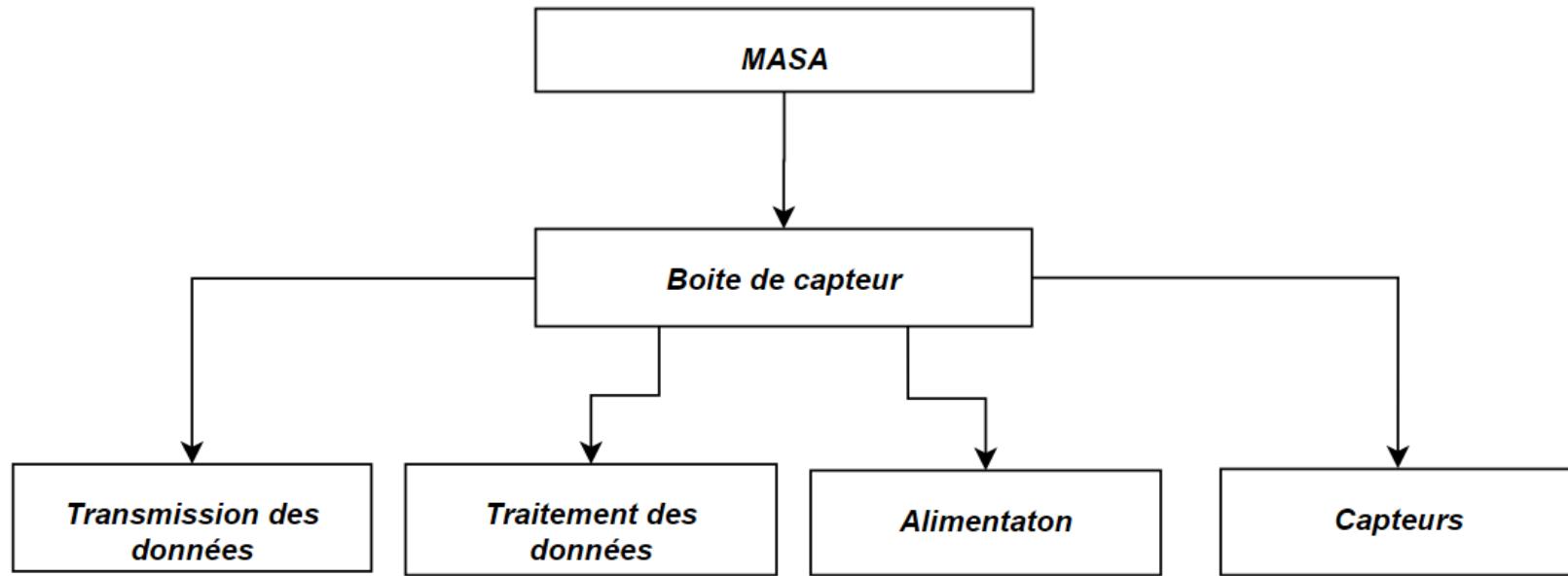
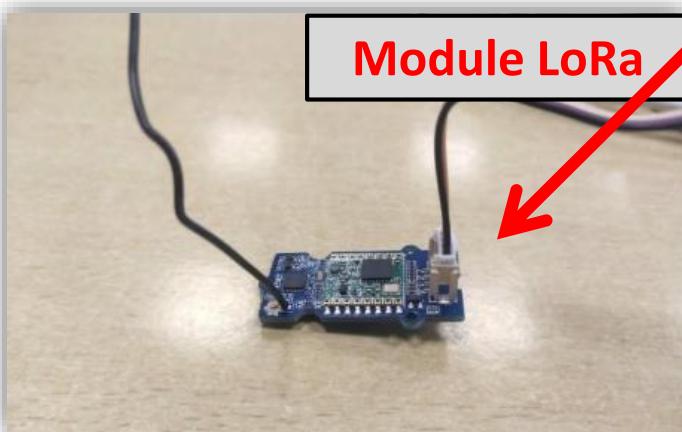
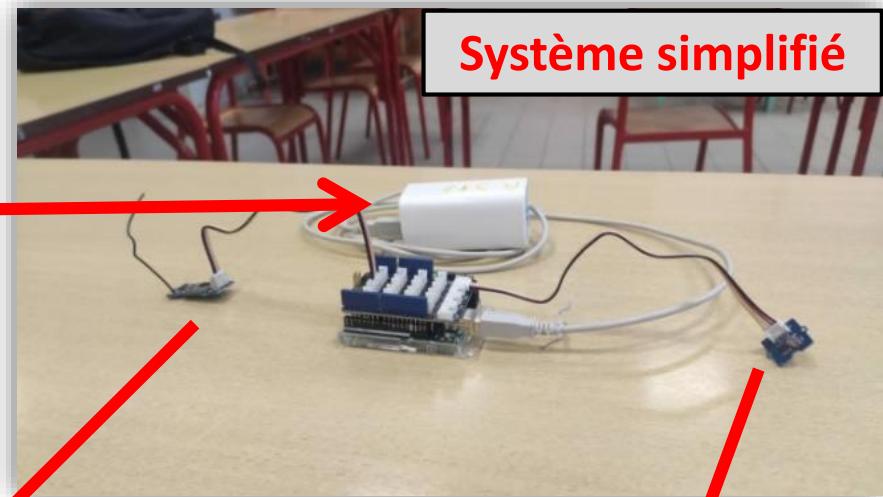


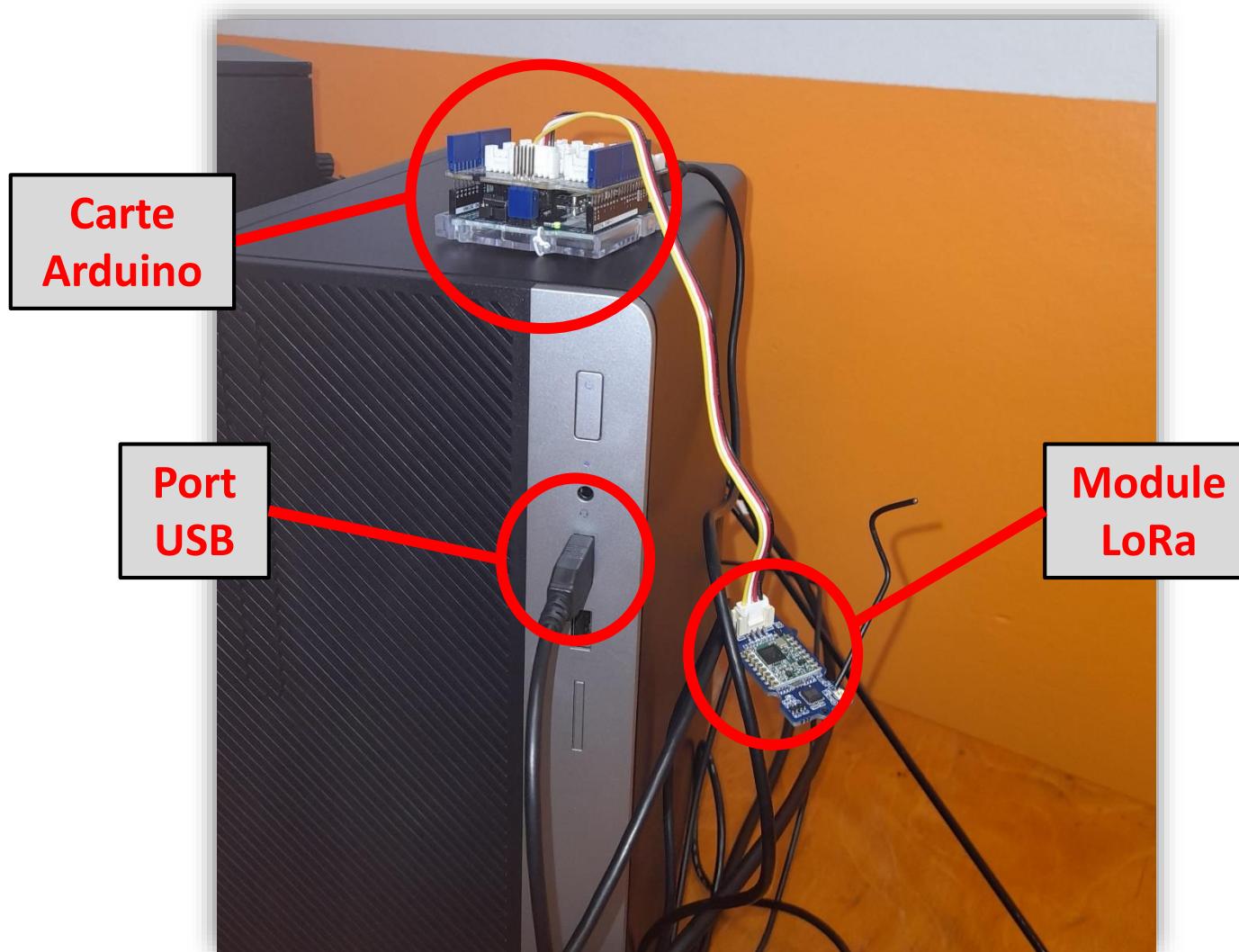
Diagramme BDD



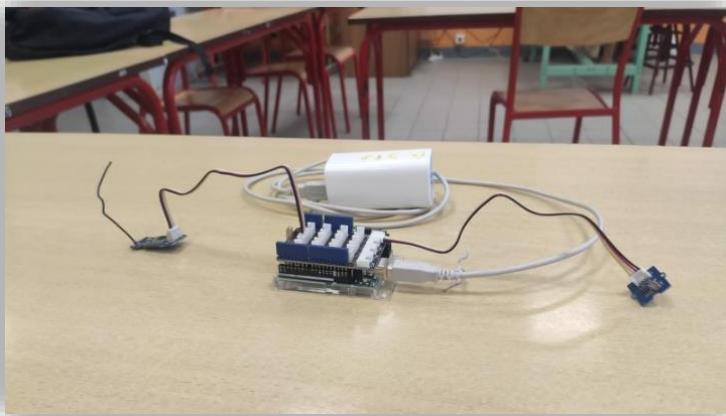
Exemple du système simplifié (capteur)



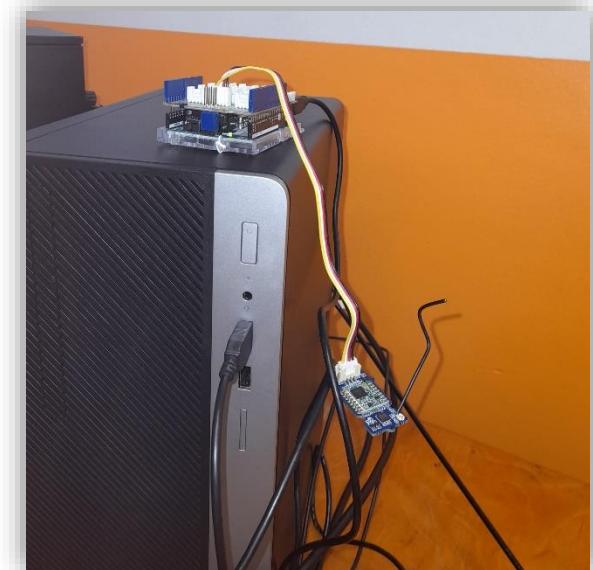
Exemple du système simplifié (base de donnée)



Représentation de la communication entre les modules



Communication
entre les capteur et
la base de données
par onde Radio LoRa
868MHz



Code Serveur

Acquisition des données

```
void loop()
{
    if(rf95.available())
    {
        // Should be a message for us now
        uint8_t buf[RH_RF95_MAX_MESSAGE_LEN]
        uint8_t len = sizeof(buf);
        if(rf95.recv(buf, &len))
        {
            digitalWrite(led, HIGH);

            ShowSerial.print("got request: ");
            ShowSerial.println((char*)buf);

            // Send a reply
            uint8_t data[] = "And hello back to you";
            rf95.send(data, sizeof(data));
            rf95.waitPacketSent();
            ShowSerial.println("Sent a reply");

            digitalWrite(led, LOW);
        }
        else
        {
            ShowSerial.println("recv failed");
        }
    }
}
```

Envie du paquet de vérification

Code Client

Envoie des données environnementales

Attente de la réception des données

```
}

rf95.setFrequency(868.0);
}

void loop()
{
    ShowSerial.println("Sending to rf95_server");
    // Send a message to rf95_server
    uint8_t data[5] = "";

    sensorValue = analogRead(sensorPin);           / read the value from the sensor:

    dtostrf(sensorValue, 4, 0, data);

    rf95.send(data, sizeof(data));

    rf95.waitPacketSent();

    // Now wait for a reply
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);

    if(rf95.waitAvailableTimeout(3000))
    {
        // Should be a reply message for us now
        if(rf95.recv(buf, &len))
        {
            ShowSerial.print("got reply: ");
            ShowSerial.println((char*)buf);
        }
        else
        {
            ShowSerial.println("recv failed");
        }
    }
    else
    {
        ShowSerial.println("No reply, is rf95_server running?");
    }

    delay(1000);
}
```

Exemple de programme : Capteur de température

```
// Demo code for Grove - Temperature Sensor V1.1/1.2
// Loovee @ 2015-8-26

#include <math.h>

const int B = 4275;           // B value of the thermistor
const int R0 = 100000;         // R0 = 100k
const int pinTempSensor = A0; // Grove - Temperature Sensor connect to A0

#if defined(ARDUINO_ARCH_AVR)
#define debug Serial
#elif defined(ARDUINO_ARCH_SAMD) || defined(ARDUINO_ARCH_SAM)
#define debug SerialUSB
#else
#define debug Serial
#endif

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = analogRead(pinTempSensor);

    float R = 1023.0/a-1.0;
    R = R0*R;

    float temperature = 1.0/(log(R/R0)/B+1/298.15)-273.15; // convert to temperature via datasheet

    Serial.print("temperature = ");
    Serial.println(temperature);

    delay(60000);
}
```

Base de Données : Serveur



Gestionnaire de
base de données

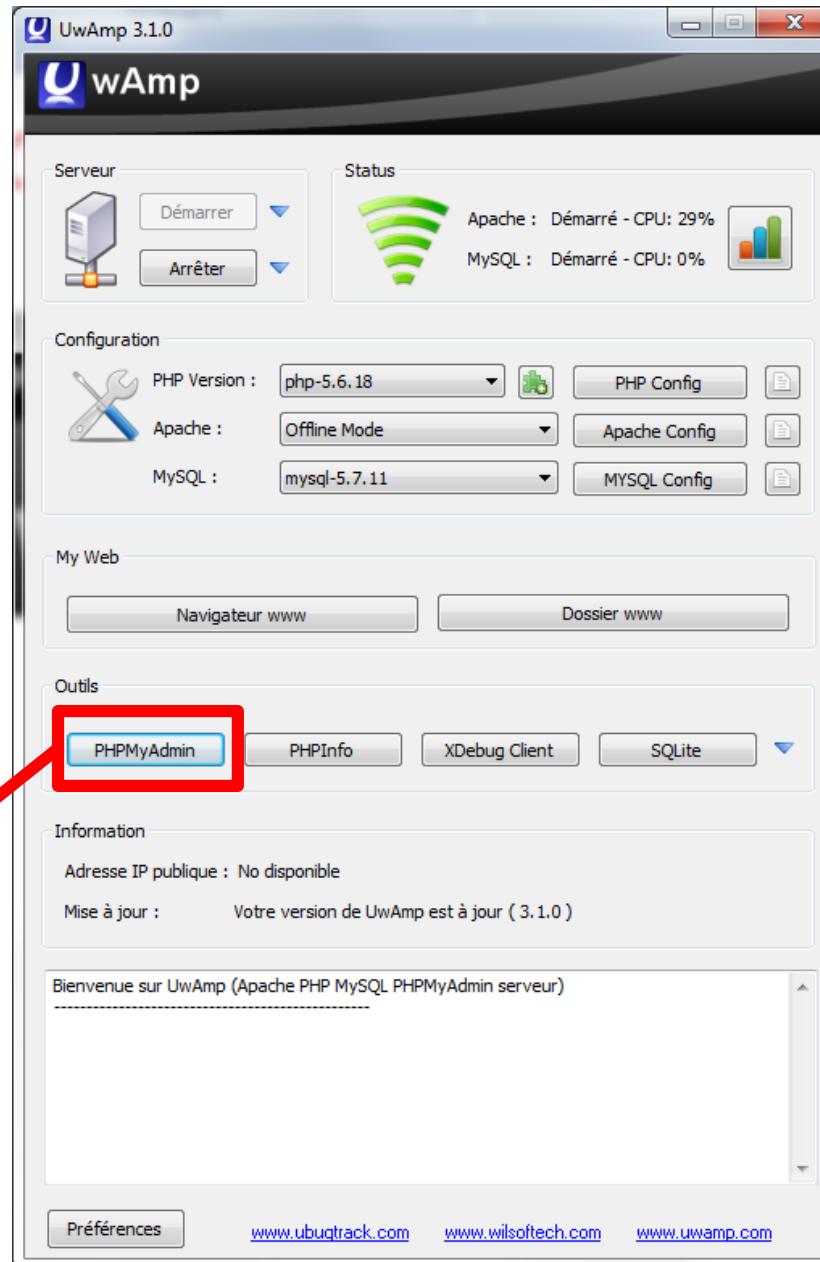


Table de mesure BDD

phpMyAdmin

Serveur: localhost » Base de données: mabase » Table: test

Afficher Structure SQL Rechercher Ins

Nouvelle base de données information_schema mabase Nouvelle table personne test Colonnes Index mysql

Structure de table Vue relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
1	idTest	int(11)			Non	Aucune
2	jour	date			Oui	NULL
3	instant	time			Oui	NULL
4	mesure	int(11)			Oui	NULL

Code Python stockage des données

```
• import mysql.connector
• from datetime import datetime
• import serial # On importe le module de communication série version pyserial-3.4

• ser1 = serial.Serial("COM5",timeout=10)
• print (ser1.portstr)      # check which port was really used

• date = str(datetime.now())
# print(date)

• jour = date[:10]
• instantMesure = date[11:19]
#print(jour)
#print(instantMesure)

#1. Connexion à la BD
• con = mysql.connector.connect(
    host='localhost',
    user='root',
    password='root'
    )

• cursor = con.cursor()
• cursor.execute("USE mabase") # select the database

• mesure=0

#while ser1.inWaiting() == 0 :
• mesure = ser1.readline().decode('utf-8') #lecture
• f.write(str(mesure))

req= "INSERT INTO test (jour, instant, mesure) VALUES ('"+jour+"', '"+instantMesure+"','"+str(mesure)+"')"
print(req)

cursor.execute(req)
con.commit()

#6. Fermeture
• con.close()
• ser1.close()
```

The diagram illustrates the flow of the Python script. It begins with a red box highlighting the MySQL connection code, which then points to a box labeled "Connexion à la base de données". This leads to another red box highlighting the code that reads data from the Arduino, which points to a box labeled "Acquisition des données de la carte Arduino". Finally, a red box highlights the MySQL insertion code, which points to a box labeled "Enregistrement des reçues de l'Arduino vers la base de données".

Connexion à la base de données

Acquisition des données de la carte Arduino

Enregistrement des reçues de l'Arduino vers la base de données

Exemple de mesures

				idTest	jour	instant	mesure
				▼			
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	1	2021-03-14	18:13:41	NULL
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2	2021-03-14	19:10:54	NULL
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	13	2021-05-11	15:07:57	765
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	12	2021-05-11	15:07:39	766
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	11	2021-05-11	15:07:25	196
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	10	2021-05-11	15:07:10	410
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	9	2021-05-11	15:01:26	762