The connected greenhouse

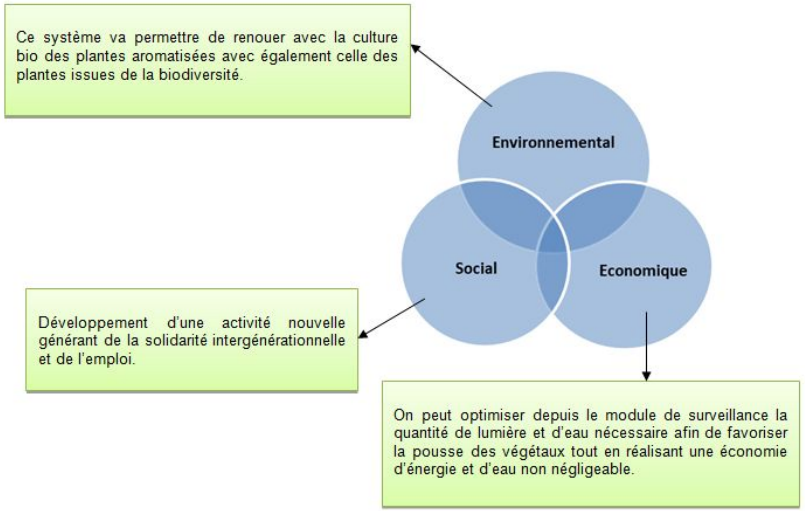
AHODOMON Jaurix / AMOUR Fiora / LE FLANCHEC Maxime (ITEC)
NGALULA MUTOKE Youssouf / POCHEBONNE Baptiste (SIN)

Summary

I.Overview:

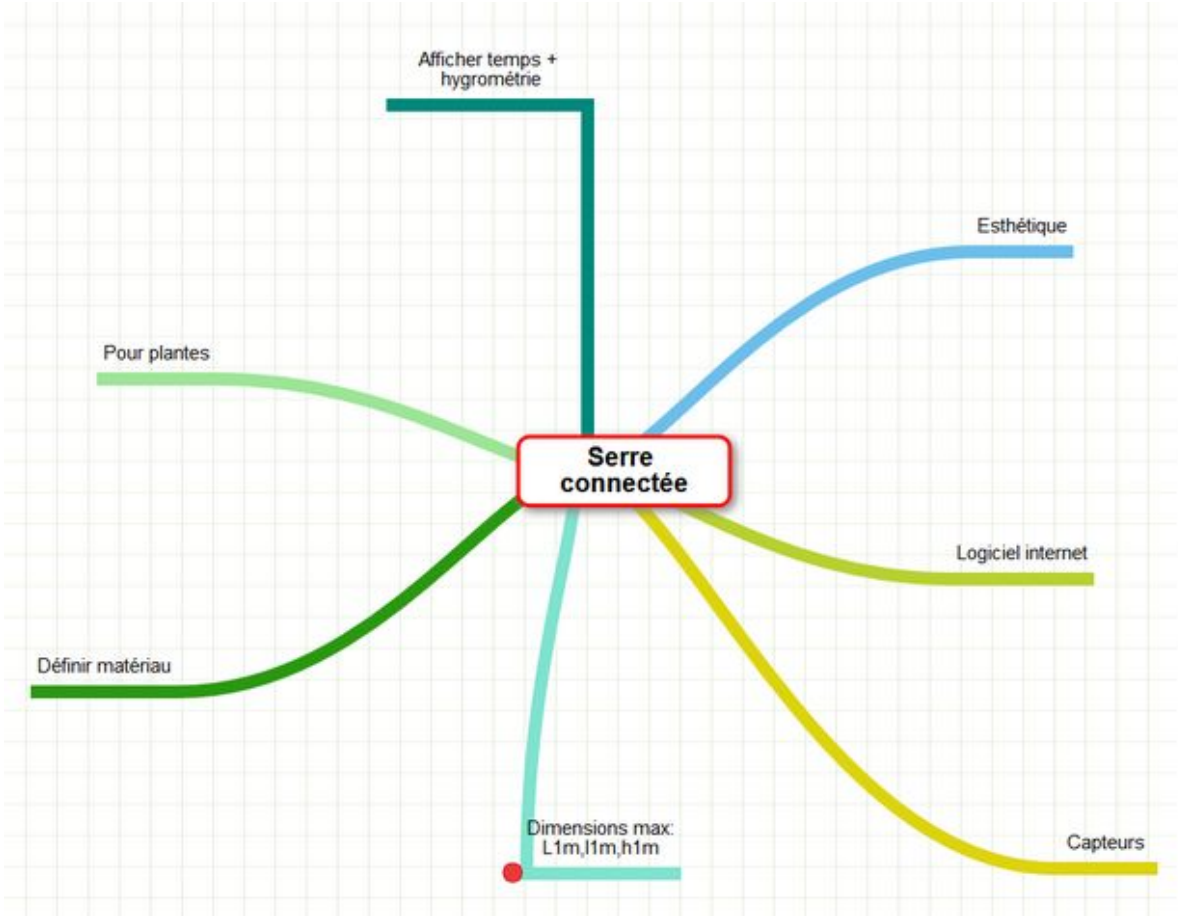
Background: large metropolises - very polluted - little space to cultivate.

Goal: -Design an aeration component,
-Integrating a water tank,
-Create the outside of the greenhouse,
- Acquire and restore humidity and temperature levels.

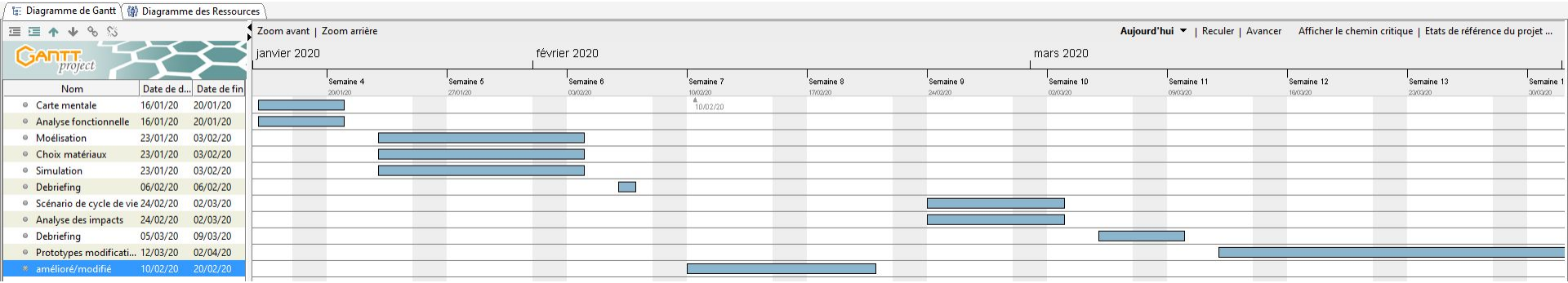


Purpose: create a connected greenhouse, display humidity - temperature

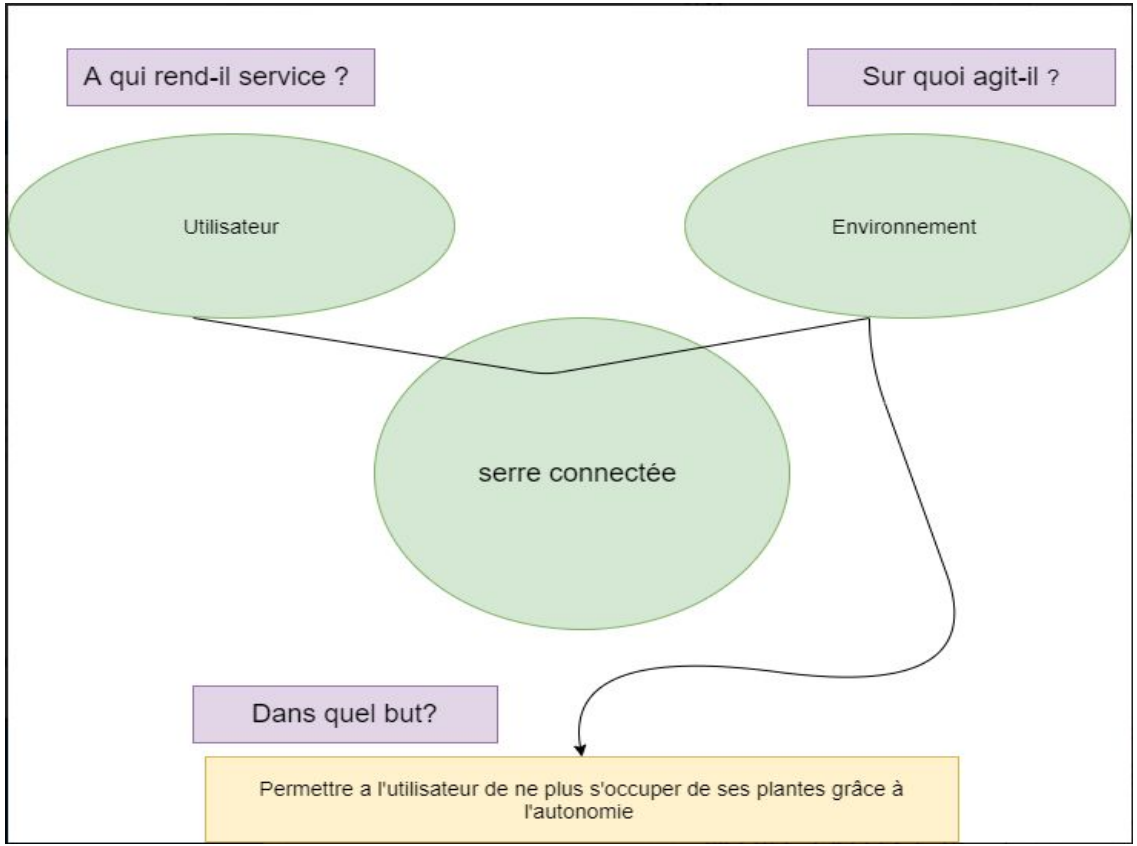
II.a.mentale card:



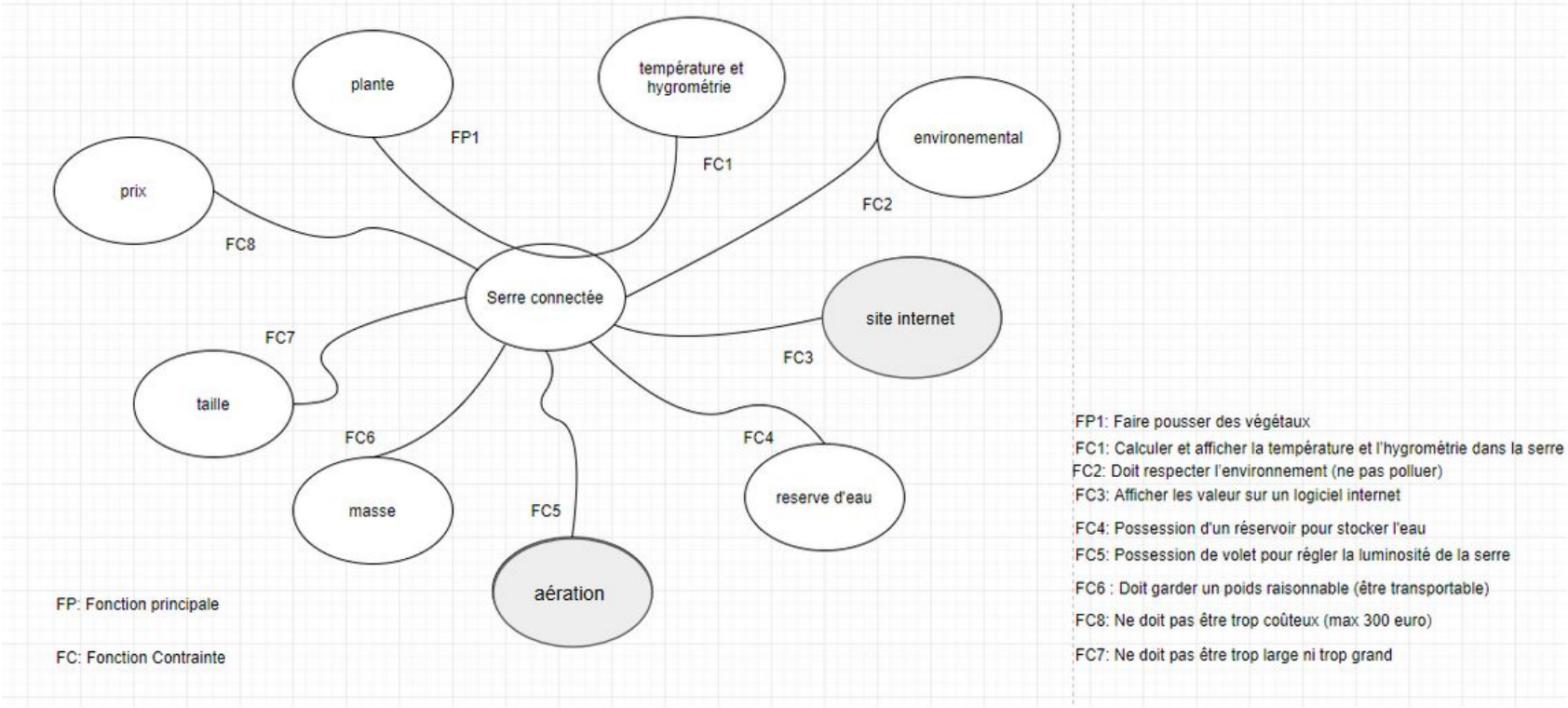
II.b.Gantt diagram:



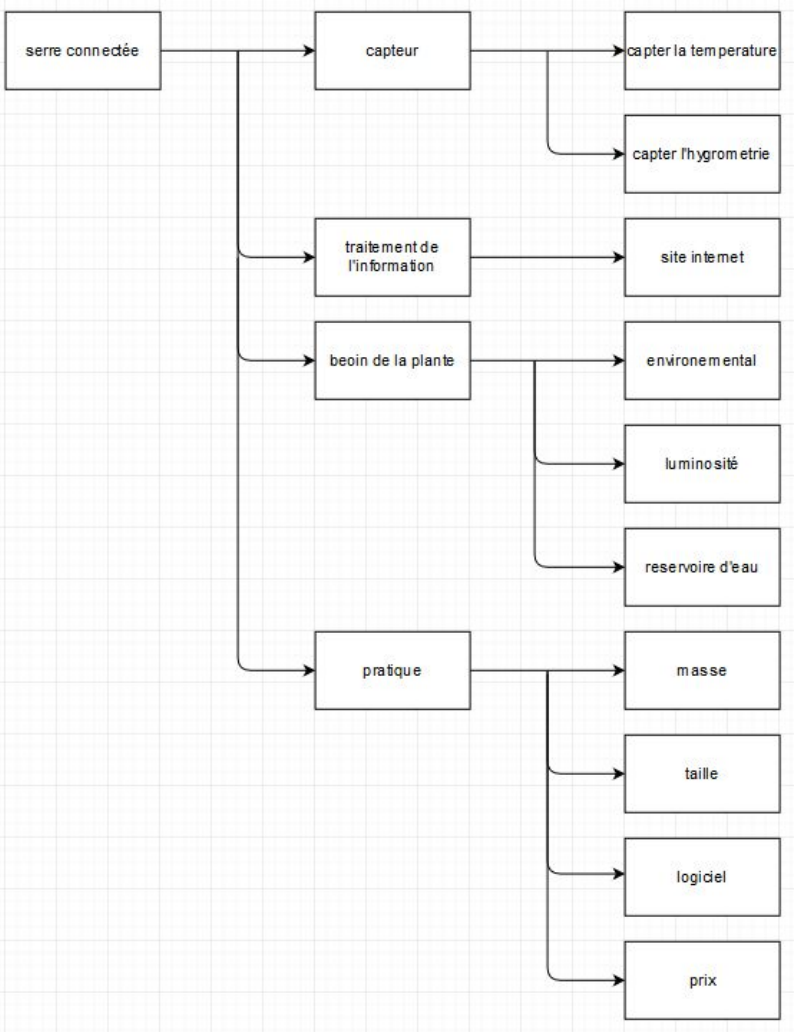
II.c.Horned Beast:



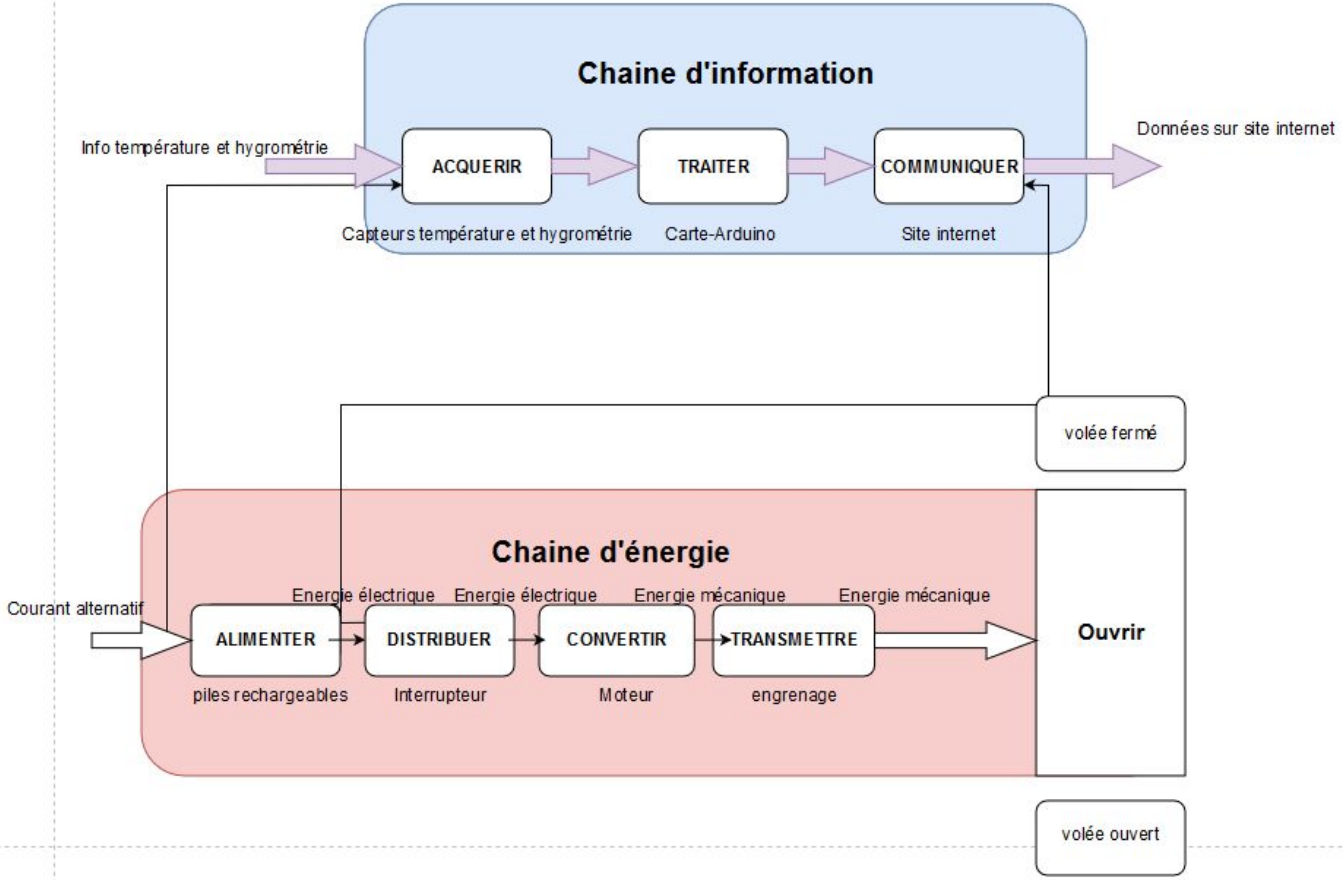
II.d.octopus diagram:



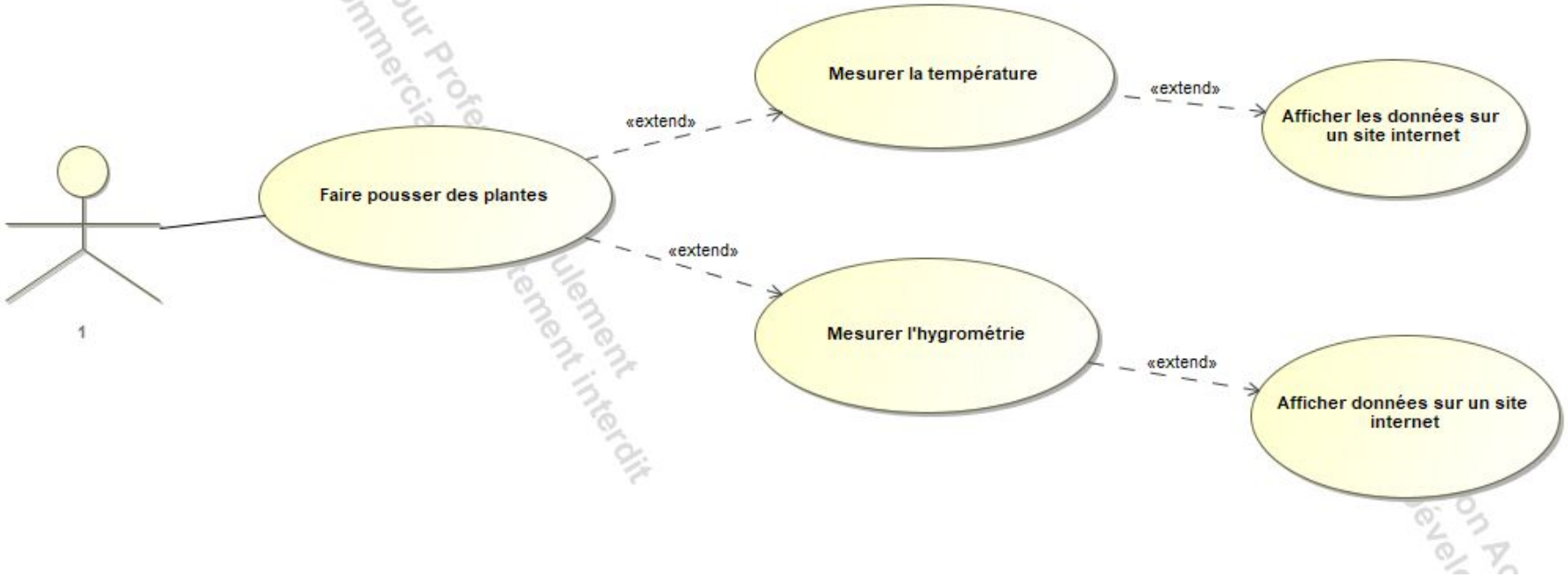
II.e.FAST diagram:



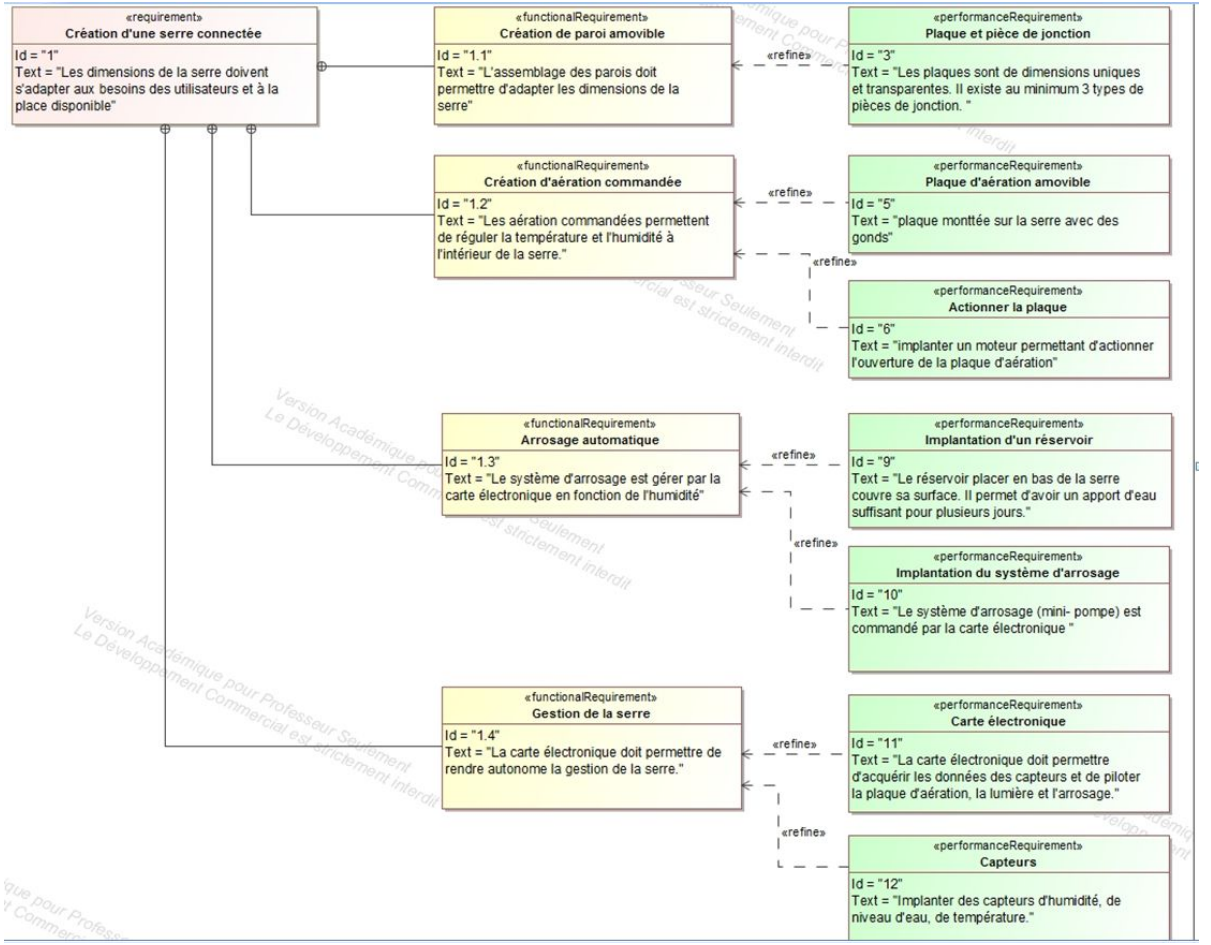
II.f.information and energy channel:



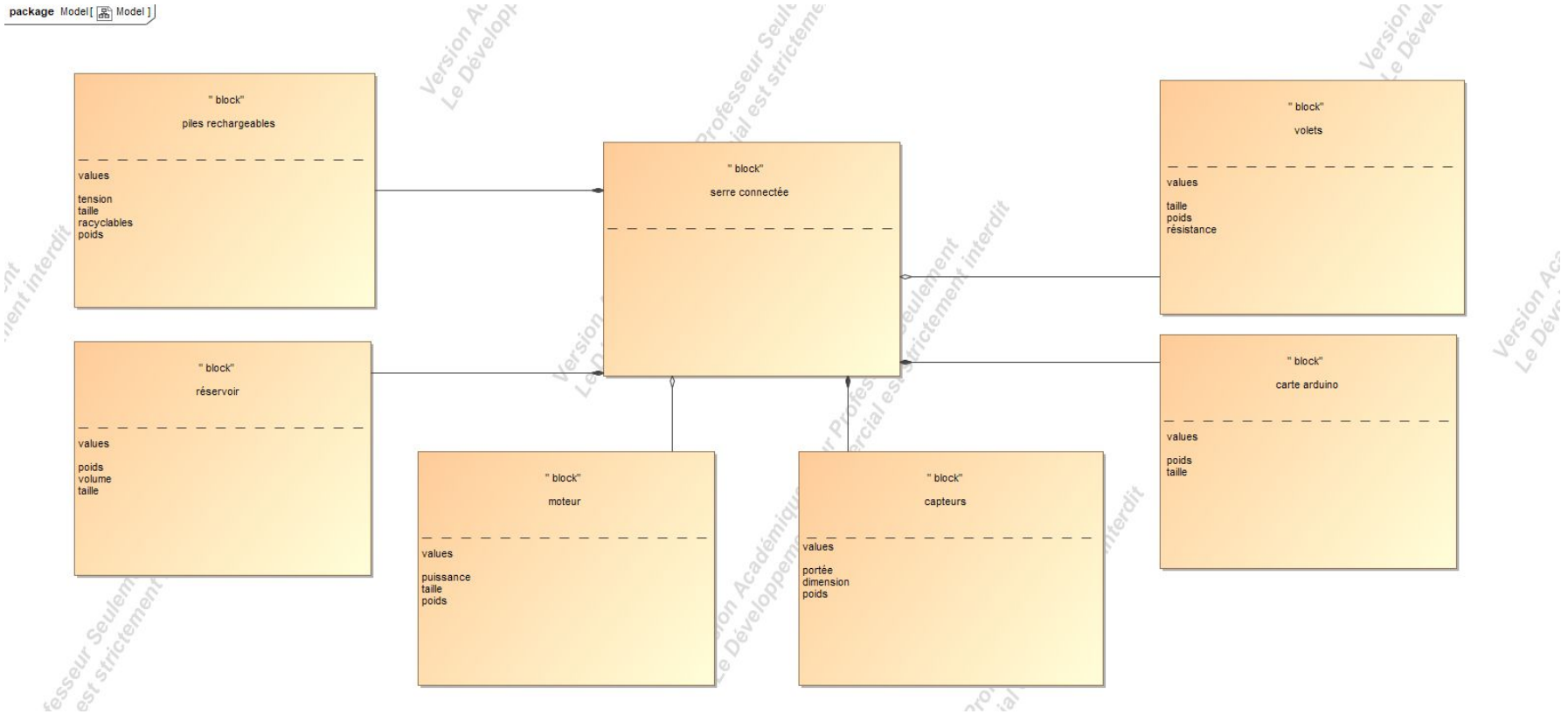
III.a.use case diagram:



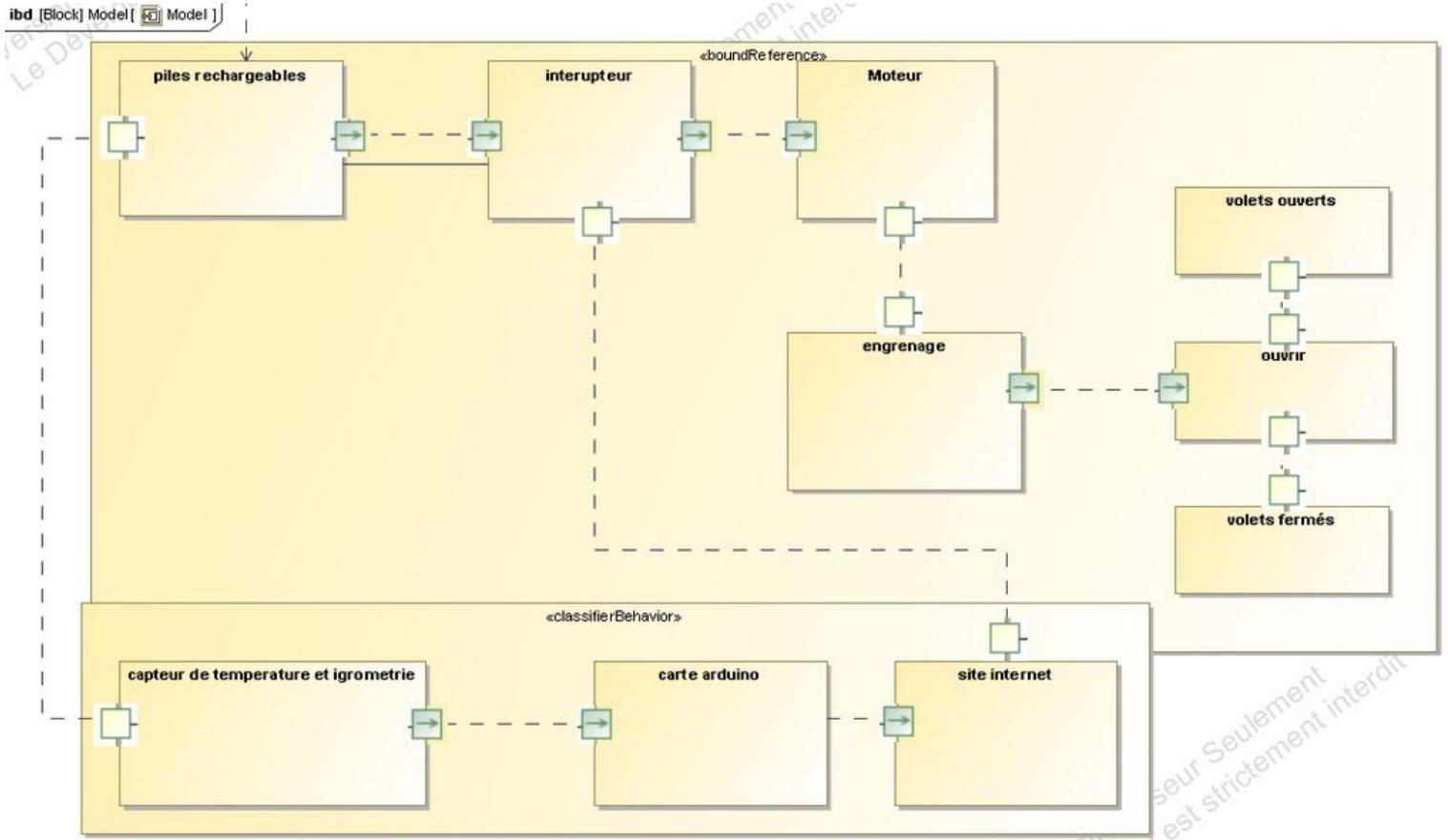
III.b.requirements diagram:



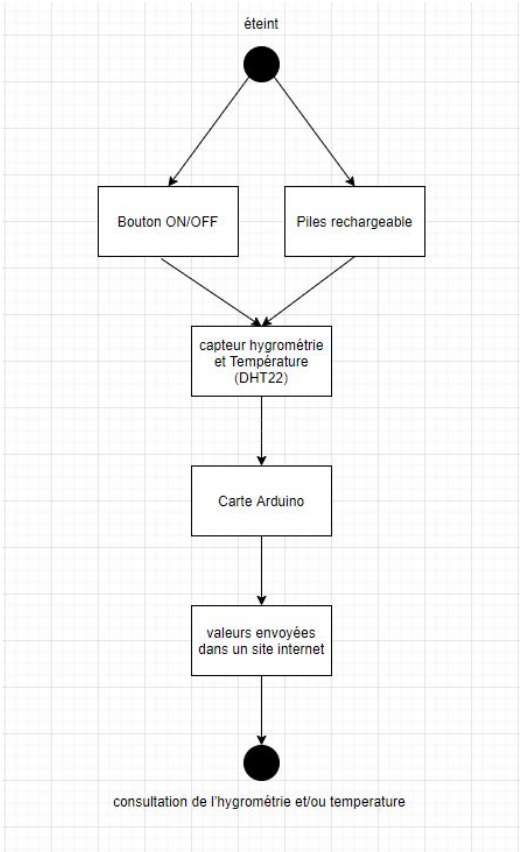
III.c.Block definition:

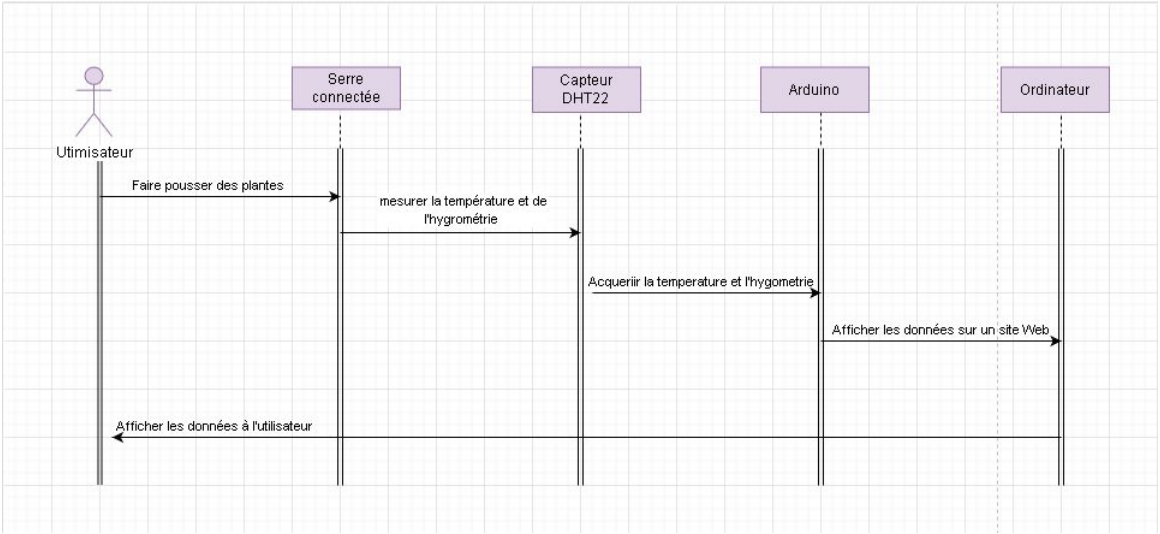


III.d.internal block diagram:



III.e.transition state diagram and sequence diagram:





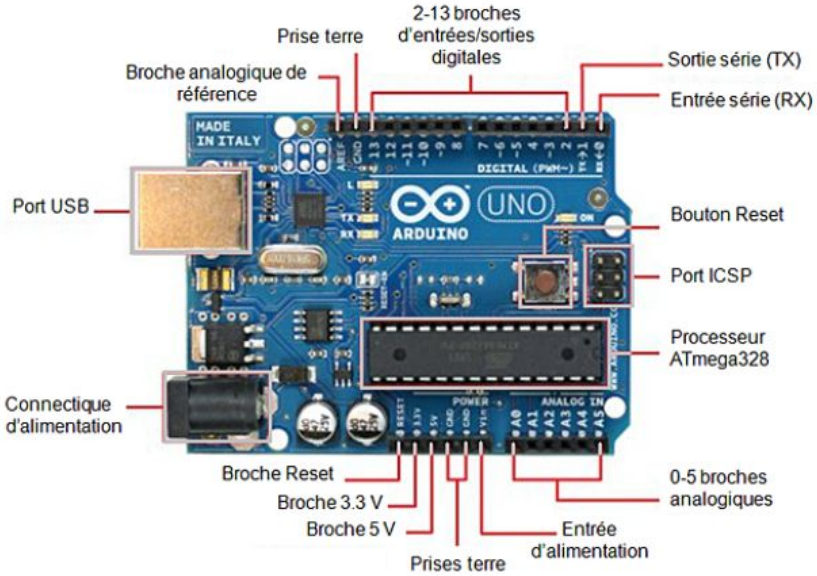
IV.a.sensor features:

Nom des capteurs	alimentation	consommation Max	plage de mesure	précision	Dimension	Prix (EUR)	conclusion
LM35 (Température)	4 à 20V	60µA	-40/+110°C	0,75 °C	X	1,58 € HT **1,90 € TTC**	calcule seulement la température (n'ai pas choisi car le GT110 ne répond pas à nos critère)
GT110 (humidité)	5Vcc	X	X	X	58*25*8mm	2,00 € HT **2,40 € TTC**	ne calcule que l'humidité dans le sol + ma
DHT11 (Température + humidité)	3 à 5 Vcc	2,5mA	-température: 0 à +50 °C - humidité: 20 à 100 % HR	-température: ± 2 °C - humidité: ± 5 % HR	16*12*7 mm	4,13 € HT **4,95 € TTC**	permet de calculer a la fois la température et l'hygrométrie
DHT22 (Température + humidité)	3,3 à 6 Vcc	1.5mA	-température: -40 à +80 °C - humidité:-0 à 100 % RH	-température: ± 0,5 °C - humidité: ± 2 % RH	25*15*9 mm	8,50 € HT **10,20 € TTC**	permet de calculer a la fois la température et l'hygrométrie

IV.b.Choice matrix:

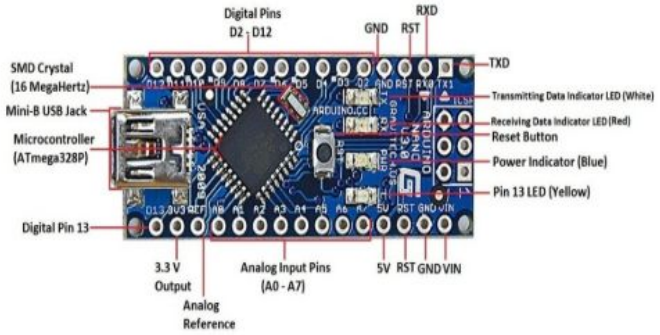
critère	LM35 (Température)	DHT11 (Température + humidité)	DHT22 (Température + humidité)
précision du capteur	0,75 °C	-température: ± 2 °C - humidité: ± 5 % HR	-température: ± 0,5 °C - humidité: ± 2 % RH
plage de mesure	-40/+110°C	-température: 0 à +50 °C - humidité: 20 à 100 % HR	-température: -40 à +80 °C - humidité:-0 à 100 % RH
faisabilité	Moyen	Fort	Fort
prix (EUR)	1,58 € HT **1,90 € TTC**	4,13 € HT **4,95 € TTC**	8,50 € HT **10,20 € TTC**
total	2/4	2/4	3/4

IV.c. Arduino uno



La carte Arduino Uno est basée sur un ATMega328 cadencé à 16 MHz. C'est la plus récente et la plus économique carte à microcontrôleur d'Arduino. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

IV.d. Arduino nano

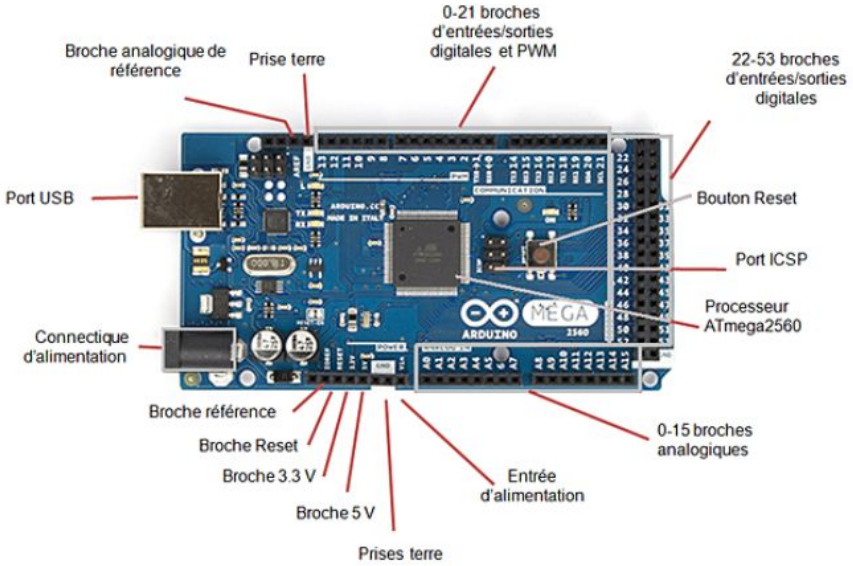


Arduino Nano V3.0 Pinout

www.CircuitsToday.com

La carte **Arduino Nano** est basée sur un ATMega328. Sa mémoire de 32 kB et ses E/S font de ce circuit un élément idéal pour les systèmes embarqués ou pour des applications nécessitant du multitâches. La carte Arduino Nano est basée sur un ATMega328 cadencé à 16 MHz.

IV.e. Arduino MEGA



La carte **Arduino Mega 2560** est basée sur un ATMega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la Uno. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'afficher une série de modules complémentaires.

IV.f.Characteristic of the Arduino:

Critère	Arduino Uno	Arduino Nano	Arduino Mega
Nombre de broches numériques	14	14	54
Nombre de broches analogiques	6	8	16
Mémoires SRAM	2kb	2kb	4kb
Mémoires EEPROM	1kb	1kb	8kb
Microprocesseur	ATMega328	ATMega328	ATMega2560
Prix (EUR)	19.50	36.50	39.50
Total	2/5	2/5	4/5

IV.g. Module ESP8266

Ce module émetteur-récepteur WiFi est basé sur un ESP8266 qui intègre le protocole TCP/IP ainsi que 1 MB de mémoire flash. Il convient particulièrement aux objets connectés.
Ce module est réservé à des utilisateurs avertis qui souhaitent effectuer des expérimentations avec le chipset WiFi ESP8266.

IV.h. Shield Ethernet EF 02029

Le module Ethernet Shield d'Elecfreaks permet à une carte Arduino de se connecter à internet. Il est basé sur le composant Wiznet 5100 et communique via le bus SPI.
Compatibilité: Arduino Uno, Mega, etc.
Le module est équipé d'un support pour une carte mémoire micro-SD, d'un connecteur RJ45, d'un bouton reset et de leds d'indication de statut.

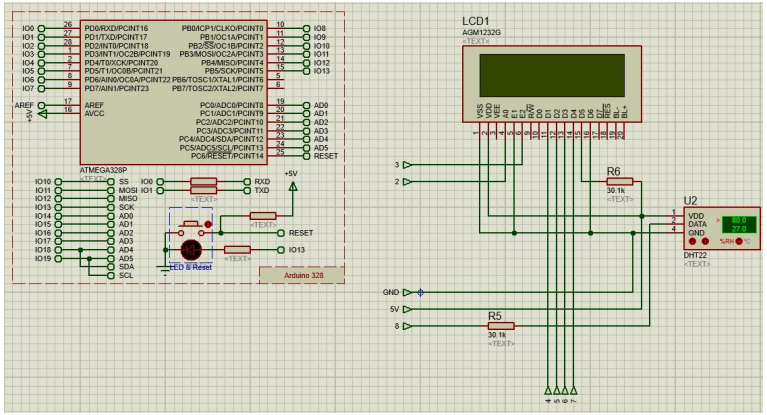
IV.i. Choice matrix:

Critères	Shield ethernet EF 02029	Module Wifi ESP8266
Alimentation	+5 Vcc (via la carte arduino)	3.3 Vcc
Dimension	73 x 53 x 30	25 x 15 x 12 mm
facilité de programmation	X	X
facilité d'installation	X	X
prix (EUR)	19,90 € TTC	8,60 € TTC
Total	3/5	2/5

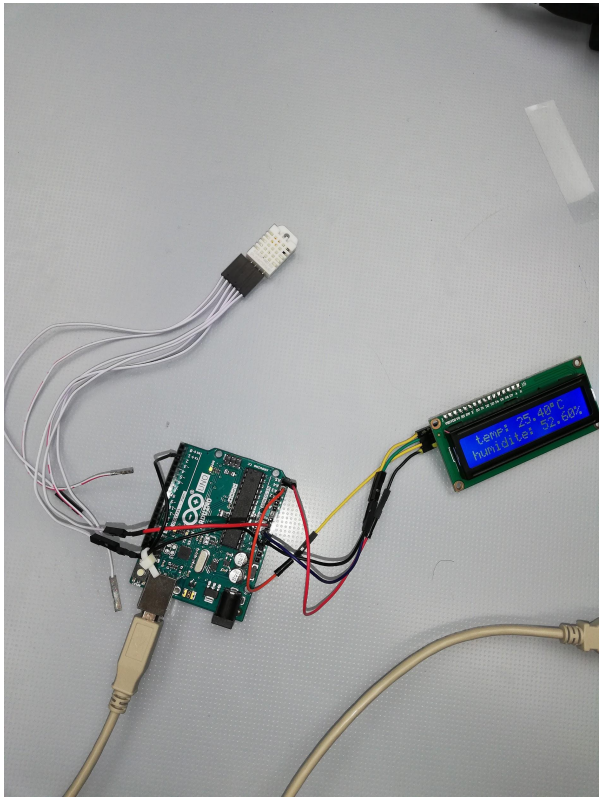
V.a. Simulation Proteus

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#define DHTPIN 8
#include <LiquidCrystal.h>
#define DHTTYPE DHT22

uint32_t delayMS;
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

void setup()
{
 Serial.begin(9600);
 lcd.begin(40, 2);


 Serial.println("DHTxx Unified Sensor Example");
 sensor_t sensor;

 Serial.println("------------------------------------");
 Serial.println("Temperature");
 Serial.print  ("Sensor: "); Serial.println(sensor.name);
 Serial.print  ("Driver Ver: "); Serial.println(sensor.version);
 Serial.print  ("Unique ID: "); Serial.println(sensor.sensor_id);
 Serial.print  ("Max Value: "); Serial.print(sensor.max_value); Serial.println(" *C");
 Serial.print  ("Min Value: "); Serial.print(sensor.min_value); Serial.println(" *C");
 Serial.print  ("Resolution: "); Serial.print(sensor.resolution); Serial.println(" *C");
 Serial.println("------------------------------------");

 Serial.println("------------------------------------");
 Serial.println("Humidity");
 Serial.print  ("Sensor: "); Serial.println(sensor.name);
 Serial.print  ("Driver Ver: "); Serial.println(sensor.version);
 Serial.print  ("Unique ID: "); Serial.println(sensor.sensor_id);
 Serial.print  ("Max Value: "); Serial.print(sensor.max_value); Serial.println("%");
 Serial.print  ("Min Value: "); Serial.print(sensor.min_value); Serial.println("%");
 Serial.print  ("Resolution: "); Serial.print(sensor.resolution); Serial.println("%");
 Serial.println("------------------------------------");
 delayMS = sensor.min_delay / 1000;

}
```

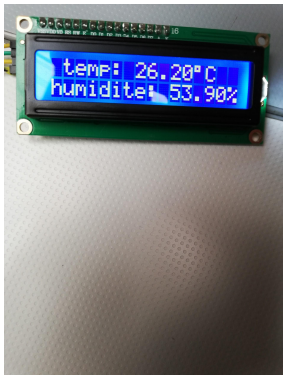
```
void loop()
{
 lcd.setCursor(0, 1);
 delay(delayMS);
 sensors_event_t event;

 if (isnan(event.temperature))
 {
 Serial.println("Error reading temperature!");
 }
 else
 {
 Serial.print("Temperature: ");
 Serial.print(event.temperature);
 Serial.println(" *C");
 lcd.print("temperature : ");
 lcd.print(event.temperature);
 }

 if (isnan(event.relative_humidity))
 {
 Serial.println("Error reading humidity!");
 }
 else
 {
 Serial.print("Humidity: ");
 Serial.print(event.relative_humidity);
 Serial.println("%");
 lcd.print(" humidity : ");
 lcd.print(event.relative_humidity);
 }


}
```



IV.k.Arduino Program




```
//libraries

#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);

//constant

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN,DHTTYPE);

//variable

float hum;
float temp;
```




```
void setup()
{
  Serial.begin(9600);
  dht.begin();
  lcd.init(); //initialisation de l'ecran LCD
  lcd.cursor_on();
  lcd.blink_on();
  lcd.backlight();
  lcd.setCursor(5,0); //ecriture de texte sur ligne 1, colonne 5
  lcd.print(temp);
  lcd.setCursor(7,1); //ecriture du texte sur ligne 2, colonne 7
  lcd.print(hum);
  delay(5000);
  lcd.cursor_off();
  lcd.blink_off();

}

void loop()
{
  hum = dht.readHumidity();
  temp = dht.readTemperature();
  lcd.setCursor(1,0);//ecriture du texte ligne 2, colonne 7
  lcd.print("temp: ");
  lcd.print(temp);//affichage des valeur de la temperature
  lcd.print(char(223));
  lcd.print("C");
  delay(5000);
  lcd.setCursor(0,1); //ecriture du texte sur ligne 2, colonne 7
  lcd.print("humidite: ");
  lcd.print(hum);//affichage des valeurs de l'humidité
  lcd.print("%");
  delay(1000);
}
```