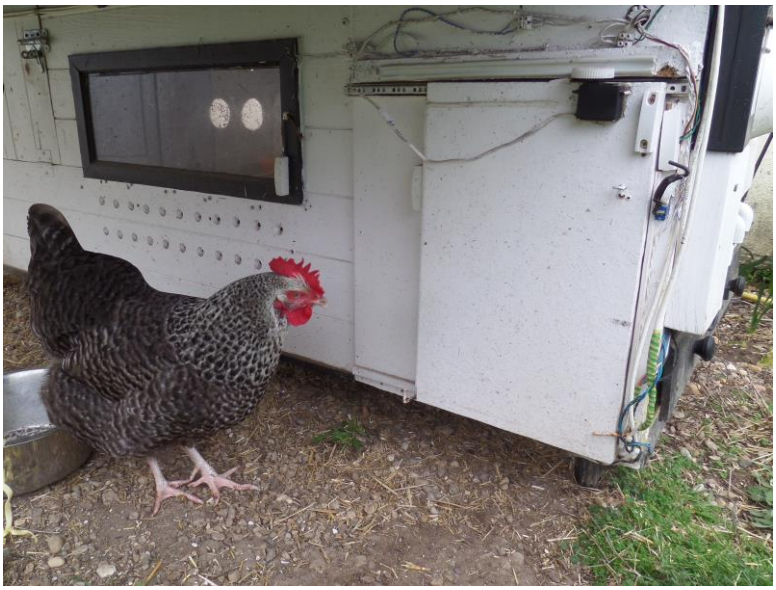
Poulailler Connecté

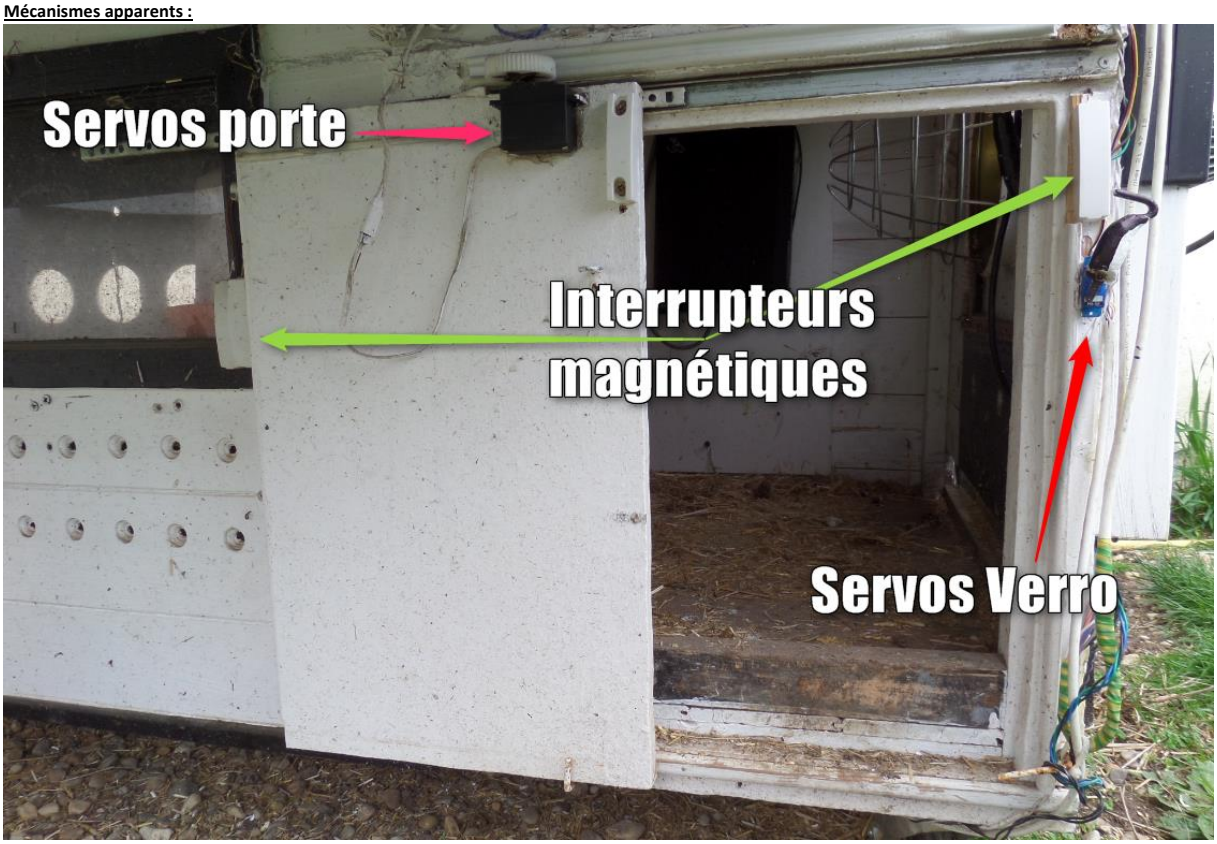
Ce projet consiste en la réalisation d'un poulailler complétement automatisé et connecté au web.
Le poulailler est capable de gérer intelligemment la fermeture et l'ouverture de la porte principale à des heures différentes suivant les saisons, mais aussi en cas de coupure électrique.
A savoir que la porte principale est verrouillée par un loquet automatisé, ce qui permet d'éviter l'intrusion de fouines qui pourraient venir gratter la porte et s'introduire à l'intérieur.
Il est capable de gérer ses portes donnant accès au pondoir, ce qui évite aux poules d'aller dormir et salir celui-ci la nuit.
Il est en mesure de nous fournir la température et l'humidité interne et externe au poulailler.
De plus, étant équipé d'une lampe chauffante Infra rouge, il est capable de gérer la température en interne lors de gros froids.
Il est aussi équipé d'un système d'éclairage afin que les poules le repère de loin le soir.
Enfin il est connecté à la mangeoire ce qui nous permet de savoir s'il reste de la nourriture ou non dans la trémie.
A savoir que les informations recueillies par l'arduino sont affichées sur un écran LCD et que le boitier contient différentes LED correspondant à un message spécifique. Ce qui est bien pratique lorsque votre poulailler est au fond du jardin et que vous voulez vérifier que tout se passe bien sans aller dehors sous la pluie.
Et le meilleur, c'est que toutes ces informations sont relayées à travers une deuxième carte arduino qui va tout retransmettre à un serveur web.

Ainsi nous obtenons une page web nous permettant de consulter l'état de notre poulailler depuis n'importe où, y compris depuis smartphone.

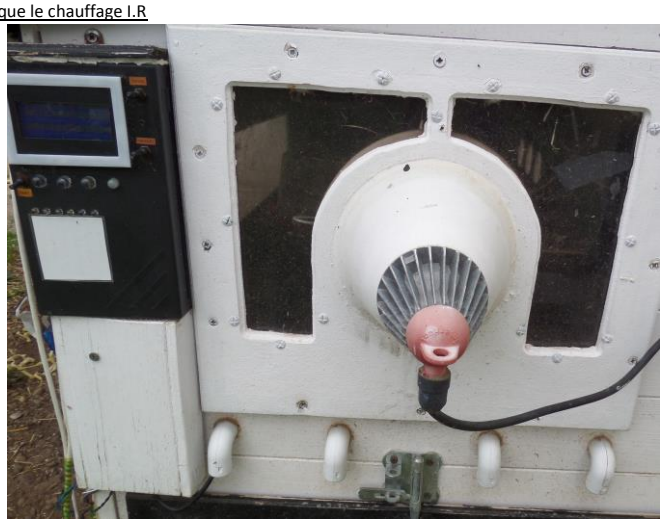
Photo de la porte :



Mécanismes apparents :



Le boitier contenant l arduino ainsi que le chauffage I.R

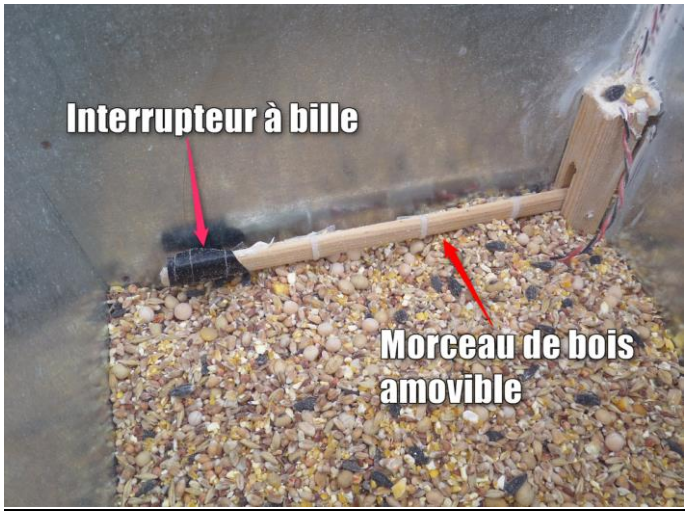


Chauffage vu de l'interieur :

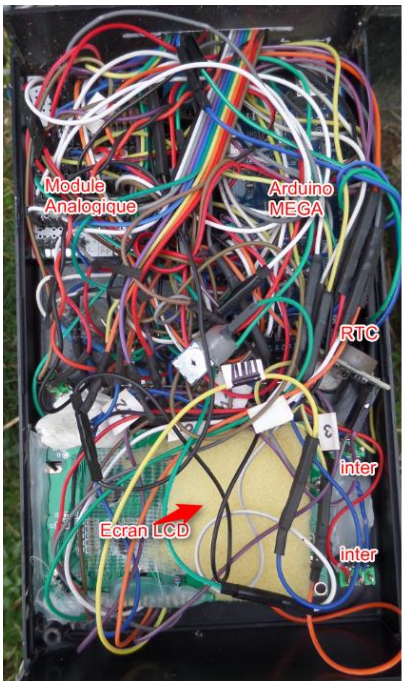
Portes donnant accès au pondoir :

La mangeoire

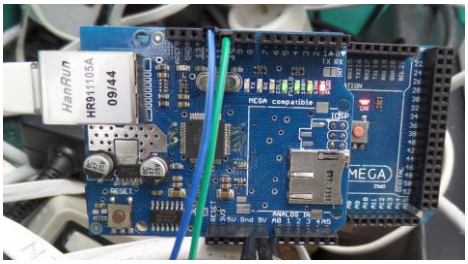
Vue de l'interieur :



Vue grossiere de l'interrieur du boitier :



Vue de la deuxieme carte :



Vue du boitier gérant l'alimentation le reseau etc…

Modules de la première carte :

La première carte est une ARDUINO MEGA, celle-ci est connectée à l'électronique suivante :

- Un écran LCD 4 Lignes.

- Un Relais pour le chauffage.

- Une horloge (module RTC).

- Des servos :
  - Servo PortePondoir (servo a rotation continue)
  - Servo Verrou
  - Servo Moteur Porte

- D'un relais pour gérer le chauffage et l'éclairage.

- De capteur de température et d'humidité (2 modules DHT11)

- D'interrupteurs :
  - Interrupteur magnétique pour la fermeture.
  - Interrupteur magnétique pour l'ouverture.
  - Interrupteur à bille pour la détection de la nourriture dans la mangeoire.
  - Interrupteur pour les Infos
  - Interrupteur pour le Mode manuel porte
  - Interrupteur pour le Mode Test Matos

De LED :
- Ouvert - Blanche
- Fermee - Bleue
- Chauffage - Rouge
- LED Vert
- LED Orange 1
- LED Orange 2
- LED Rouge 1
- LED Rouge_2
- LED Rouge 3

De résistances diverses.

Connectivité – Branchement des modules de la premiere carte :

Ecran 2004 lcd :

PIN 1 --> GND
PIN 2 --> 5V
PIN 3 --> milieu du 10 K les deux autres au + et - inverser si pb constaté
PIN 4 --> 12 PIN MEGA 2560
PIN 5 --> GND
PIN 6 --> 11 PIN MEGA 2560
PIN 7 -->
PIN 8 -->
PIN 9 -->
PIN 10 -->
PIN 11 --> 5 PIN MEGA 2560
PIN 12 --> 4 PIN MEGA 2560
PIN 13 --> 3 PIN MEGA 2560
PIN 14 --> 2 PIN MEGA 2560
PIN 15 --> 5V via 220
PIN 16 --> GND

//

PIN ARDUINO MEGA 2560

//0 - X -
//1 - X -
//2 lcd
//3 lcd
//4 lcd
//5 lcd
//6 LED Ouvert - Blanche - LED
//7 LED Fermee - Bleue - LED
//8 LED Chauffage - Rouge - LED
//9 LED Vert - LED
//10 LED Orange 1 - LED
//11 lcd
//12 lcd
//13 LED Rouge 3 - LED
//14
//15
//16
//17
//18 Relais Chauffage - RELAIS
//19
//20 RTC_SDA - HORLOGE
//21 RTC_SCL - HORLOGE
//22 laison serie // RX
//23 laison serie // TX
//24
//25
//26
//27
//28
//29
//30
//31
//32
//33
//34
//35
//36
//37 LED Rouge 1 - LED
//38 LED Orange 2 - LED
//39 Servo PortePondoir - SERVO
//40 Servo Verrou - SERVO
//41 Servo Moteur Porte - SERVO
//42
//43 Relais Eclairage - RELAIS
//44
//45
//46
//47 Led Rouge_2 - LED
//48 DHT11 Ext - TEMPERATURE
//49 DHT11 Int - TEMPERATURE
//50

//51
//52
//53

//A0 InterF - INTERRUPTEUR
//A1 InterO - INTERRUPTEUR
//A2 Inter Infos - INTERRUPTEUR
//A3 Inter Mode manuel porte - INTERRUPTEUR
//A4
//A5 Inter Food - INTERRUPTEUR
//A6 Inter Test Matos - INTERRUPTEUR
//A7
//A8
//A9
//A10
//A11
//A12
//A13
//A14
//A15

Architecture du programme de la première carte :

Definition des modules lcd
Definition des modules Servo
Définition des broches utilisées pour capteur DHT 11 et bibliothèques associées
Définition des broches utilisées pour les Inter
Definition des broches utilisees par les leds de la porte
Définition des broches utilisées pour module Relais éclairage
Definition Relais chauffage
Définition des broches utilisées pour module RTC & Bibliothèques associées
Définition des broches utilisées pour les autres capteurs

setup()

```
// Initialisation du port serie
// Affichage d'un message de start
// Définition des outputs pour led ouvert-ferme
// Définition des output pour LED  Chauff
// Définition des variables panneau led infos
// Définition des inputs pour les micro rupteurs
// Définition des inputs pour le micro rupteur InterFood
// Définition des inputs pour InterTestMatos
// Définition de l'inter info
// Définition de l'inter mode manuel
// Définition des outputs pour Relais Lumiere
// Définition des outputs pour Relais_Servos
// Définition des outputs pour Relais Chauff
// Définition des outputs pour lcd
```

loop()

```
// Boucle Initiale
// Lecture des capteurs
// Affichage des informations
// Gestion des actions
// Gestion des actions manuelles via interrupteurs
// Detection des erreurs
```

// fonctions appelées dans le loop

First_Init()
LectureRTC()
LectureTemperature()
LectureHumidite()
LectureInterF()
LectureInterO()
LectureInterModeManuel()
LectureInterInfos()
LectureInterTestMatos()
LectureInterFood()
GestionAffichageLED()
AffichagePrincipal()
GestionPondoir()
GestionPortePondoirMatin()
GestionPortePondoirSoir()
GestionEclairageExt()
ChauffagePostFermeture()
Chauffage()
GestionPortePrincipale()
OuverturePorte()
FermeturePorte()
DetectionBloquage()
AffichageVersion()
GestionTest_Matos()
Test_Matos()
GestionAffichageInfos()
AffichageInfos()
Conversion()
Envoie_Infos()

Programme de la premiere carte :

DEFINITION DES BIBLIOTHEQUES ET MODULES

```
////////////////////////////////////////////////////////////////////////////////
//                  Definition des modules lcd
////////////////////////////////////////////////////////////////////////////////

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);



////////////////////////////////////////////////////////////////////////////////
//                  Definition des modules Servo
////////////////////////////////////////////////////////////////////////////////

#include <Servo.h>

// create a servo object pour la porte
Servo Servo_Porte_Principale;
// permet de savoir si la Porte Principale est ouverte ou fermée
int Valeur_Servo_Porte_Principale = 0;


// create a servo object pour le verrou
Servo Servo_Verrou;
// permet de savoir si le verrou et ouvert ou fermée
int Valeur_Servo_Verrou = 0;


// create a servo object pour la PortePondoir
Servo Servo_Porte_Pondoir;
// permet de savoir si le val Porte Pondoir est ouvert ou fermée
int Valeur_Servo_Porte_Pondoir = 0;



//////////////////////////////////////////////////////////////////////////////////
//    Définition des broches utilisées pour capteur DHT 11 et bibliothèques associées
//////////////////////////////////////////////////////////////////////////////////

#include <dht11.h>

// Définition Capteur T Ext & int
dht11 DHT11_Temp_Ext;
dht11 DHT11_Temp_Int;

// Capteur DHT11 - pin 48
#define DHT11_Temp_ExtPin 48
// Capteur DHT11 - pin 49
#define DHT11_Temp_IntPin 49


// Déclaration et init des Valeurs des capteurs Ext et Int
float DHT11_Valeur_Ext_Temperature = 0;
float DHT11_Valeur_Ext_Humidite = 0;

float DHT11_Valeur_Int_Temperature = 0;
float DHT11_Valeur_Int_Humidite = 0;



////////////////////////////////////////////////////////////////////////////////
//          Définition des broches utilisées pour les Inter de la porte
////////////////////////////////////////////////////////////////////////////////

// Inter        microrupteur magnétique fermeture
//int InterF = A0;
// Inter        microrupteur magnétique ouverture
//int InterO = A1;
```

```
// Valeur Inter F
int ValeurInterF = 0;
// Valeur Inter O
int ValeurInterO = 0;


// Inter Infos
//int InterInfos = A2;
// Valeur InterInfos
int ValeurInterInfos = 0;


// Inter mode manuel
//int InterModeManuel = A3;
// Valeur Inter Mode Manuel
int ValeurInterModeManuel = 0;


// Inter Food
//int InterFood = A5;
// Valeur Inter Mode Manuel
int ValeurInterFood = 1;


// Inter TestMatos
//int InterTestMatos = A6;
// Valeur Inter Mode Manuel
int ValeurInterTestMatos = 0;


// Valeur Bloquage des portes si pb (automatique)
int PorteBloquageO = 0;
int PorteBloquageF = 0;


// Timer bloquage porte ouverte / fermee
unsigned long TimerBPO = 0;
unsigned long TimerBPF = 0;



////////////////////////////////////////////////////////////////////////////////////////////
//      Definition des broches utilisees par les leds de la porte
////////////////////////////////////////////////////////////////////////////////////////////

// Ecl_Led_Ouvert
int Led_Ouvert = 6;
int Valeur_LED_Blanche = 0;  // correspond  LED ouvert


// Ecl_Led_Fermee
int Led_Fermee = 7;
int Valeur_LED_Bleu = 0;  // correspond LED ferme


// Led d'informations --> action
int Led_Verte = 9;
int Valeur_Led_Verte = 0;  // correspond LED action


// Led d'informations --> Plus de nourriture
int Led_Orange_1 = 10;
int Valeur_Led_Orange_1 = 0;  // correspond LED Plus de nourriture


// Led d'informations --> Servos sous tension
int Led_Orange_2 = 38;
int Valeur_Led_Orange_2 = 0;


// Led d'informations --> erreur porte avec les inter
int Led_Rouge_1 = 37;
int Valeur_Led_Rouge_1 = 0;  // correspond    LED erreur porte avec les inter
```

```
// Led d'informations --> erreur porte timer out
int Led_Rouge_2 = 47;
int Valeur_Led_Rouge_2 = 0;  // correspond    LED erreur porte timer out


// Led d'informations la 13 correspond aussi au pin du serial donc s'allume pour ça
int Led_Rouge_3 = 13;
int Valeur_Led_Rouge_3 = 0;  // correspond    LED data


// LED Chauffage
int LEDChauffage = 8;
int ValeurLEDChauffage = 0;



////////////////////////////////////////////////////////////////////////////////////////
//       Définition des broches utilisées pour module Relais éclairage
////////////////////////////////////////////////////////////////////////////////////////

// Relais éclairage
int Relais_Eclairage = 43;

// Valeur Relais
int Valeur_Relais_Eclairage = 0;




////////////////////////////////////////////////////////////////////////////////////////
//                    Definition Relais chauffage
////////////////////////////////////////////////////////////////////////////////////////

// Relais Chauffage
int RelaisChauffage = 18;

// Valeur Relais Chauffage
int  Valeur_Relais_Chauffage = 0;


// Timer de fonctionnement mini pour le chauffageRelaisChauffage
unsigned long  Timer_Chauffage = 0;



////////////////////////////////////////////////////////////////////////////////////////
//    Définition des broches utilisées pour module RTC & Bibliothèques associées
////////////////////////////////////////////////////////////////////////////////////////

#include "Wire.h"
#include <HCRTC.h>
#define I2CDS1307Add 0x68
#define I2C24C32Add  0x50
//char ExampleString[] = "Horloge Aurelien";
//byte index;
HCRTC HCRTC;


// Déclaration des variables du module RTC
int  RTCHeure = 0;
int  RTCMinute = 0;
int  RTCAnnee = 0;
int  RTCMois = 0;
int  RTCJour = 0;
int  RTCSeconde = 0;
int  RTCJourSemaine = 0;

// permet un seul passage dans init
int Var_Init = 0;
```

```
////////////////////////////////////////////////////////
//     Bblio et variables pour la liaison série
////////////////////////////////////////////////////////

#include <SoftwareSerial.h>

SoftwareSerial mySerial(22, 23);

// chaine d caractères pour l'envoie des données via liaison série

String  STR_RTCAnnee;
String  STR_RTCMois;
String  STR_RTCJour ;
String  STR_RTCHeure;
String  STR_RTCMinute;
String  STR_DHT11_Valeur_Int_Temperature_Entier;
String  STR_DHT11_Valeur_Int_Humidite_Entier;
String  STR_DHT11_Valeur_Ext_Temperature_Entier;
String  STR_DHT11_Valeur_Ext_Humidite_Entier;
String  STR_Valeur_Servo_Porte_Principale;
String  STR_ValeurInterF;
String  STR_ValeurInterO;
String  STR_PorteBloquageO;
String  STR_PorteBloquageF;
String  STR_Valeur_Servo_Verrou;
String  STR_Valeur_Servo_Porte_Pondoir;
String  STR_Valeur_Relais_Chauffage;
String  STR_Valeur_Relais_Eclairage;
String  STR_ValeurInterInfos;
String  STR_ValeurInterModeManuel;
String  STR_ValeurInterTestMatos;
String  STR_ValeurInterFood;
String  STR_Valeur_LED_Blanche;
String  STR_Valeur_LED_Bleu;
String  STR_Valeur_Led_Verte;
String  STR_Valeur_Led_Orange_1;
String  STR_Valeur_Led_Orange_2;
String  STR_Valeur_Led_Rouge_1;
String  STR_Valeur_Led_Rouge_2;
String  STR_Valeur_Led_Rouge_3;
String  STR_ValeurLEDChauffage;



////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                              SETUP
//                              NE RIEN METTRE D'AUTRE QUE LES INPUT ET OUTPUT - AUCUN TEST ET AFFICHAGE ECRAN TFT
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void setup()
{

  Serial.begin(9600);

  // initialisation du port serie
  mySerial.begin(4800);

  // Affichage d'un message de start
  Serial.println("Initialisation Programme --> SETUP");


  // Définition des outputs pour led ouvert-ferme
  Serial.println("Definition Output pour led ouvert-ferme");
  pinMode(Led_Ouvert, OUTPUT);
  pinMode(Led_Fermee, OUTPUT);

  // Définition des output pour LED  Chauff
  Serial.println("Definition Output pour LEDChauffage");
  pinMode(LEDChauffage, OUTPUT);

  // Définition des variables panneau led infos
  Serial.println("Definition Output panneau led infos");
  pinMode(Led_Verte, OUTPUT);
  pinMode(Led_Orange_1, OUTPUT);
  pinMode(Led_Orange_2, OUTPUT);
```

```
  pinMode(Led_Rouge_1, OUTPUT);
  pinMode(Led_Rouge_2, OUTPUT);
  pinMode(Led_Rouge_3, OUTPUT);

  // Définition des inputs pour les micro rupteurs
  Serial.println("Definition Input pour les microrupteurs");
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);

  // Définition des inputs pour le micro rupteur InterFood
  Serial.println("Definition Input pour InterFood");
  pinMode(A5, INPUT);

  // Définition des inputs pour InterTestMatos
  Serial.println("Definition Input pour InterTestMatos");
  pinMode(A6, INPUT);

  // Définition de l'inter info
  Serial.println("Definition Input de l inter info");
  pinMode(A2, INPUT);

  // Définition de l'inter mode manuel
  Serial.println("Definition Input de l inter mode manuel");
  pinMode(A3, INPUT);

  // Définition des outputs pour Relais Lumiere
  Serial.println("Definition Output pour Relais_Eclairage");
  pinMode(Relais_Eclairage, OUTPUT);

  // Définition des outputs pour Relais_Servos
  Serial.println("Definition Output pour les 3 Relais Servos");


  // Définition des outputs pour Relais Chauff
  Serial.println("Definition Output pour RelaisChauffage");
  pinMode(RelaisChauffage, OUTPUT);


  // Définition des outputs pour lcd
  Serial.println("Definition Output pour lcd");
  // set up the lcd's number of columns and rows:
  lcd.begin(20, 4);

  Serial.println("-----------------------");

}


//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                                          LOOP
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void loop()
{


  //////////////////////////////////////////////
  //          BOUCLE INITIALE
  //////////////////////////////////////////////

  Serial.println("------------DEBUT------------");

  // mise en place initiale des elements pour le demarrage à froid.
  if (Var_Init == 0)
  {
   First_Init();
  }
  Var_Init = 1;




  //////////////////////////////////////////////
  // Lecture des capteurs
  //////////////////////////////////////////////
```

```
LectureRTC();
delay(100);

LectureTemperature();
delay(100);

LectureHumidite();
delay(100);

LectureInterF();
delay(100);

LectureInterO();
delay(100);

LectureInterModeManuel();
delay(100);

LectureInterTestMatos();
delay(100);

LectureInterInfos();
delay(100);

LectureInterFood();
delay(100);


////////////////////////////////////////////////
// Affichage des informations
////////////////////////////////////////////////

AffichagePrincipal();
delay(1000);

GestionAffichageLED();
delay(100);


////////////////////////////////////////////////
// Gestion des actions
////////////////////////////////////////////////

GestionPortePrincipale();
delay(1000);

GestionPondoir();
delay(1000);

GestionEclairageExt();
delay(1000);

GestionChauffage();
delay(1000);


/////////////////////////////////////////////////
// Gestion des actions manuelles via interrupteurs
/////////////////////////////////////////////////

GestionAffichageInfos();
delay(1000);

GestionTest_Matos();
delay(1000);


////////////////////////////////////////////////
// Detection des erreurs
////////////////////////////////////////////////

LectureInterF();
delay(100);

LectureInterO();
```

```
  delay(100);

  DetectionBloquage();

}


/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                    Fonctions
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////




//////////////////////////////////////////////////////////////////////////////////
//                     Fonction - Première boucle.
//    Ici on ouvre le verrou de la porte principale et on ferme la porte du pondoir.
//    Ceci s'effectue à chaque mise sous tension, ce qui évité tout problème avec les poules
//////////////////////////////////////////////////////////////////////////////////

void First_Init()
{

  delay(250);

  // affichage de la version
  AffichageVersion();


  delay(250);
  lcd.setCursor(0, 0);
  lcd.print("   Start Arduino !  ");

  lcd.setCursor(0, 1);
  lcd.print("      MADE BY       ");

  lcd.setCursor(0, 2);
  lcd.print("     Aurelien       ");

  lcd.setCursor(0, 3);
  lcd.print("     XXXXXXXX       ");

  delay(2000);
  lcd.clear();



  /////////////////////////////////////////////phase de test
  Serial.println("--> Test_Matos");

  lcd.setCursor(0, 0);
  lcd.print("--------------------");

  lcd.setCursor(0, 1);
  lcd.print("       TEST        ");

  lcd.setCursor(0, 2);
  lcd.print("       MATOS       ");

  lcd.setCursor(0, 3);
  lcd.print("--------------------");
  delay(2000);

  lcd.clear();


  //***// Allume et eteint tous les relais
  Serial.println("Allume tous les relais qq instant");

  lcd.setCursor(0, 1);
  lcd.print("ON-OFF ts les Relais");

  digitalWrite(RelaisChauffage, LOW );
  digitalWrite(Relais_Eclairage, HIGH);
```

```
delay(2000);
lcd.clear();


//***// Mise à zéro de tous les paramètres
Serial.println("Mise a zero de tous les parametres");

lcd.setCursor(0, 1);
lcd.print("Tous les param a 0  ");


digitalWrite(Led_Fermee, LOW);
digitalWrite(Led_Ouvert, LOW);
digitalWrite(LEDChauffage, LOW);
digitalWrite(Led_Verte, LOW);
digitalWrite(Led_Orange_1, LOW);
digitalWrite(Led_Orange_2, LOW);
digitalWrite(Led_Rouge_1, LOW);
digitalWrite(Led_Rouge_2, LOW);
digitalWrite(Led_Rouge_3, LOW);
digitalWrite(RelaisChauffage, HIGH);
digitalWrite(Relais_Eclairage, LOW);

delay(500);

lcd.clear();

//***//Lecture des capteurs
Serial.println("Lecture capteurs");

lcd.setCursor(0, 1);
lcd.print("Lecture des capteurs");

// Lecture de ValeurInterF
Serial.print("ValeurInterF = ");
ValeurInterF = digitalRead(A0);
Serial.println(ValeurInterF);

// Lecture du ValeurInterO
Serial.print("ValeurInterO = ");
ValeurInterO = digitalRead(A1);
Serial.println(ValeurInterO);

// Lecture du ValeurInterFood
Serial.print("ValeurInterFood = ");
ValeurInterFood = digitalRead(A5);
Serial.println(ValeurInterFood);

// Lecture du ValeurInterInfos
Serial.print("ValeurInterInfos = ");
ValeurInterInfos = digitalRead(A2);
Serial.println(ValeurInterInfos);

// Lecture du ValeurModeManuel
Serial.print("ValeurInterModeManuel = ");
ValeurInterModeManuel = digitalRead(A3);
Serial.println(ValeurInterModeManuel);

// Lecture du ValeurInterTestMatos
Serial.print("ValeurInterTestMatos = ");
ValeurInterTestMatos = digitalRead(A6);
Serial.println(ValeurInterTestMatos);


delay(2000);
lcd.clear();

//***// test les relais
Serial.println("Test tous les relais");

lcd.setCursor(0, 1);
lcd.print("Test ts les relais  ");

Serial.println("Test Relais Chauffage LOW");
digitalWrite(RelaisChauffage, LOW); // High car c'est la logique inverse
delay(2000);
```

```
Serial.println("Test Relais Chauffage HIGH");
digitalWrite(RelaisChauffage, HIGH);

delay(1000);
Serial.println("Test Relais Chauffage LOW");
digitalWrite(RelaisChauffage, LOW); // High car c'est la logique inverse

delay(2000);
Serial.println("Test Relais Chauffage HIGH");
digitalWrite(RelaisChauffage, HIGH);
delay(1000);

Serial.println("Test Relais_Eclairage HIGH ");
digitalWrite(Relais_Eclairage, HIGH);
delay(1000);

Serial.println("Test Relais_Eclairage LOW");
digitalWrite(Relais_Eclairage, LOW);
delay(1000);

lcd.clear();


//***// Test du verrou

/*  Serial.println("Test verrou");

  lcd.setCursor(0, 1);
  lcd.print("Test  verrou       ");

  digitalWrite(Led_Orange_2, HIGH);
  Valeur_Led_Orange_2 = 1;
  digitalWrite(Led_Verte, HIGH);
  Valeur_Led_Verte = 1;

  // test du verrou
  Serial.println("Test Servo Verrou");
  Servo_Verrou.attach(40);
  delay(250);
  Servo_Verrou.write(120);
  delay(1000);
  Servo_Verrou.write(0);
  delay(1000);
  Servo_Verrou.write(120);
  delay(1000);
  Servo_Verrou.detach();
*/



//***// Test de la porte

/*Serial.println("Test Servo Moteur Porte");

lcd.setCursor(0, 1);
lcd.print("Test  porte        ");

Servo_Porte_Principale.attach(41);
delay(250);

ValeurInterF = digitalRead(A0);
// fermeture
while (ValeurInterF == 0)
{
  Servo_Porte_Principale.write(70);

  // Lecture de l etat des capteur inter
  Serial.print("Lecture de l etat du capteur inter = ");
  ValeurInterF = digitalRead(A0);
  delay(10);
  Serial.println(ValeurInterF);

}
Servo_Porte_Principale.detach();
```

```
delay(3000);

Servo_Porte_Principale.attach(41);
delay(250);


ValeurInterO = digitalRead(A1);
//ouverture
while (ValeurInterO == 0)
{
  Serial.println("Ouverture de la porte tant que l on est pas en butee");

  // active le servo
  Servo_Porte_Principale.write(108);
  delay(200); // permet d'avoir un peu d'inertie

  // Lecture de l etat des capteur inter
  Serial.print("Lecture de l etat des capteur inter = ");
  ValeurInterO = digitalRead(A1);
  delay(250);
}


// desactive le servo
Servo_Porte_Principale.detach();
*/


//***// test du PortePondoir

/*
  Serial.println("Test Servos PortePondoir");

  lcd.setCursor(0, 1);
  lcd.print("Test  pondoir       ");


  Servo_Porte_Pondoir.attach(39);
  delay(250);
  Servo_Porte_Pondoir.write(153);
  delay(2000);
  Servo_Porte_Pondoir.write(75);
  delay(2000);
  Servo_Porte_Pondoir.detach();


  digitalWrite(Led_Verte, LOW);
  Valeur_Led_Verte = 0;
  digitalWrite(Led_Orange_2, LOW);
  Valeur_Led_Orange_2 = 0;
  delay(250);
*/



//***// test leds

/*  Serial.println("Test LED");

  lcd.setCursor(0, 1);
  lcd.print("Test  LED          ");


  Serial.println("Test LED Ouvert");
  // test des leds
  digitalWrite(Led_Ouvert, HIGH);
  delay(500);
  digitalWrite(Led_Ouvert, LOW);

  Serial.println("Test LED Fermee");
  digitalWrite(Led_Fermee, HIGH);
  delay(500);
  digitalWrite(Led_Fermee, LOW);

  Serial.println("Test LED Chauffage");
  digitalWrite(LEDChauffage, HIGH);
```

```
  delay(500);
  digitalWrite(LEDChauffage, LOW);

  Serial.println("Test LED Verte");
  digitalWrite(Led_Verte, HIGH);
  Valeur_Led_Verte = 1;
  delay(500);
  digitalWrite(Led_Verte, LOW);
  Valeur_Led_Verte = 0;

  Serial.println("Test LED Orange 1");
  digitalWrite(Led_Orange_1, HIGH);
  Valeur_Led_Orange_1 = 1;
  delay(500);
  digitalWrite(Led_Orange_1, LOW);
  Valeur_Led_Orange_1 = 0;

  Serial.println("Test LED Orange 2");
  digitalWrite(Led_Orange_2, HIGH);
  Valeur_Led_Orange_2 = 1;
  delay(500);
  digitalWrite(Led_Orange_2, LOW);
  Valeur_Led_Orange_2 = 0;

  Serial.println("Test LED Rouge 1");
  digitalWrite(Led_Rouge_1, HIGH);
  Valeur_Led_Rouge_1 = 1;
  delay(500);
  digitalWrite(Led_Rouge_1, LOW);
  Valeur_Led_Rouge_1 = 0;

  Serial.println("Test LED Rouge 2");
  digitalWrite(Led_Rouge_2, HIGH);
  Valeur_Led_Rouge_2 = 1;
  delay(500);
  digitalWrite(Led_Rouge_2, LOW);
  Valeur_Led_Rouge_2 = 0;

  Serial.println("Test LED Rouge 3");
  digitalWrite(Led_Rouge_3, HIGH);
  Valeur_Led_Rouge_3 = 1;
  delay(500);
  digitalWrite(Led_Rouge_3, LOW);
  Valeur_Led_Rouge_3 = 0;


  delay(1000);
*/

//***//Mise a zero de tous les parametres

Serial.println("Mise a zero de tous les parametres");

lcd.setCursor(0, 1);
lcd.print("Ts les param a 0    ");


digitalWrite(Led_Fermee, LOW);
digitalWrite(Led_Ouvert, LOW);
digitalWrite(LEDChauffage, LOW);
digitalWrite(Led_Verte, LOW);
digitalWrite(Led_Orange_1, LOW);
digitalWrite(Led_Orange_2, LOW);
digitalWrite(Led_Rouge_1, LOW);
digitalWrite(Led_Rouge_2, LOW);
digitalWrite(Led_Rouge_3, LOW);
digitalWrite(RelaisChauffage, HIGH);
digitalWrite(Relais_Eclairage, LOW);

delay(1000);

lcd.clear();

//////////////////////////////////////////////// Fin phase de test
```

```
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
lcd.setCursor(0, 0);

lcd.print("  Ouverture verrou  ");

lcd.setCursor(0, 1);

lcd.print("        et        ");

lcd.setCursor(0, 2);

lcd.print("  porte du pondoir  ");

lcd.setCursor(0, 3);

lcd.print("         *        ");
Serial.println("");

Serial.println("Passe toutes les sorties a low");

digitalWrite(Led_Fermee, LOW);
digitalWrite(Led_Ouvert, LOW);
digitalWrite(LEDChauffage, LOW);
digitalWrite(Led_Verte, LOW);
digitalWrite(Led_Orange_1, LOW);
digitalWrite(Led_Orange_2, LOW);
digitalWrite(Led_Rouge_1, LOW);
digitalWrite(Led_Rouge_2, LOW);
digitalWrite(Led_Rouge_3, LOW);
digitalWrite(RelaisChauffage, HIGH); // High car c'est la logique inverse
digitalWrite(Relais_Eclairage, LOW);


// Ouverture des SERVOS en cas de reboot

digitalWrite(Led_Orange_2, HIGH);
Valeur_Led_Orange_2 = 1;
digitalWrite(Led_Verte, HIGH);
Valeur_Led_Verte = 1;

// verrou
Serial.println("Servo Verrou");
Servo_Verrou.attach(40);
delay(250);
Servo_Verrou.write(120);
delay(1000);
Servo_Verrou.detach();



// PortePondoir
Serial.println("Servos PortePondoir");
Servo_Porte_Pondoir.attach(39);
delay(250);
Servo_Porte_Pondoir.write(75);
delay(2000);
Servo_Porte_Pondoir.detach();


digitalWrite(Led_Verte, LOW);
Valeur_Led_Verte = 0;
digitalWrite(Led_Orange_2, LOW);
Valeur_Led_Orange_2 = 0;
delay(250);


Serial.println("Passe toutes les sorties a low");

digitalWrite(Led_Fermee, LOW);
digitalWrite(Led_Ouvert, LOW);
```

```
  digitalWrite(LEDChauffage, LOW);
  digitalWrite(Led_Verte, LOW);
  digitalWrite(Led_Orange_1, LOW);
  digitalWrite(Led_Orange_2, LOW);
  digitalWrite(Led_Rouge_1, LOW);
  digitalWrite(Led_Rouge_2, LOW);
  digitalWrite(Led_Rouge_3, LOW);
  digitalWrite(RelaisChauffage, HIGH); // High car c'est la logique inverse
  digitalWrite(Relais_Eclairage, LOW);

  AffichagePrincipal();
  delay(100);

}

//***********************************/   /***********************************/   /***********************************/
//                                                    Fonctions Lecture Capteurs
//***********************************/   /***********************************/   /***********************************/


////////////////////////////////////////////////////////////////////////////////////////////
//                        Fonction -  Lecture RTC
////////////////////////////////////////////////////////////////////////////////////////////

void LectureRTC()
{
  Serial.println("--> Lecture Module RTC");

  // Lecture de l'heure
  HCRTC.EEStartRead(I2C24C32Add, 32000);
  delay(80);
  HCRTC.RTCRead(I2CDS1307Add);
  delay(80);

  // Recuperation des datas concernant le module RTC et stockage dans une variable de type entier
  RTCAnnee = HCRTC.GetYear();
  delay(200);
  RTCMois = HCRTC.GetMonth();
  RTCJour = HCRTC.GetDay();
  delay(200);
  RTCJourSemaine = HCRTC.GetWeekday();
  delay(200);
  RTCHeure = HCRTC.GetHour();
  delay(200);
  RTCMinute = HCRTC.GetMinute();
  delay(200);
  RTCSeconde = HCRTC.GetSecond();
  delay(200);

  // Affichage serial de la date et de l'heure
  Serial.print("Jour de la semaine : ");
  Serial.println(RTCJourSemaine);
  Serial.print("Date : ");
  Serial.print(RTCJour);
  Serial.print("/");
  Serial.print(RTCMois);
  Serial.print("/");
  Serial.println(RTCAnnee);
  Serial.print("Heure : ");
  Serial.print(RTCHeure);
  Serial.print(":");
  Serial.print(RTCMinute);
  Serial.print(":");
  Serial.println(RTCSeconde);
}



////////////////////////////////////////////////////////////////////////////////////////////
//                        Fonction -  Lecture Temperature
////////////////////////////////////////////////////////////////////////////////////////////

void LectureTemperature()
{
  Serial.println("--> Lecture Temperature");
```

```
  if (DHT11_Temp_Ext.read(DHT11_Temp_ExtPin) == DHTLIB_OK)
  {
    // Recuperation de la T du capteur DHT11 et stockage des infos dans un flottant
    DHT11_Valeur_Ext_Temperature = (int)DHT11_Temp_Ext.temperature, 2;
    Serial.print("Temperature DHT ext : ");
    Serial.println(DHT11_Valeur_Ext_Temperature);
    delay(500);
  }


  if (DHT11_Temp_Int.read(DHT11_Temp_IntPin) == DHTLIB_OK)
  {
    // Recuperation de la T du capteur DHT11 et stockage des infos dans un flottant
    DHT11_Valeur_Int_Temperature = (int)DHT11_Temp_Int.temperature, 2;
    Serial.print("Temperature DHT int : ");
    Serial.println(DHT11_Valeur_Int_Temperature);
    delay(500);
  }

}



////////////////////////////////////////////////////////////////////////////////////////////////////
//                          Fonction - Lecture Humidite
////////////////////////////////////////////////////////////////////////////////////////////////////
void LectureHumidite()
{
  Serial.println("--> Lecture Humidite");

  if (DHT11_Temp_Ext.read(DHT11_Temp_ExtPin) == DHTLIB_OK)
  {
    // Recuperation de l'humidite du capteur DHT11 et stockage des infos dans un flottant
    DHT11_Valeur_Ext_Humidite = (int)DHT11_Temp_Ext.humidity, 2;
    Serial.print("Humidite DHT Ext: ");
    Serial.println(DHT11_Valeur_Ext_Humidite);
    delay(500);
  }


  // Releve de la temperature et de l humidite des sondes int
  if (DHT11_Temp_Int.read(DHT11_Temp_IntPin) == DHTLIB_OK)
  {
    // Recuperation de l'humidite du capteur DHT11 et stockage des infos dans un flottant
    DHT11_Valeur_Int_Humidite = (int)DHT11_Temp_Int.humidity, 2;
    Serial.print("Humidite DHT Int: ");
    Serial.println(DHT11_Valeur_Int_Humidite);
    delay(500);
  }
}




////////////////////////////////////////////////////////////////////////////////////////////////////
//                  Fonction - Lecture du ValeurInterF
////////////////////////////////////////////////////////////////////////////////////////////////////

void LectureInterF()
{
  Serial.println("--> Lecture ValeurInterF ");

  ValeurInterF = digitalRead(A0);
  delay(50);
  Serial.print("ValeurInterF = ");
  Serial.println(ValeurInterF);
}



////////////////////////////////////////////////////////////////////////////////////////////////////
//                Fonction - Lecture du ValeurInterO
////////////////////////////////////////////////////////////////////////////////////////////////////

void LectureInterO()
```

```
{
  Serial.println("--> Lecture ValeurInterO");

  ValeurInterO = digitalRead(A1);
  delay(50);
  Serial.print("ValeurInterO = ");
  Serial.println(ValeurInterO);
}



////////////////////////////////////////////////////////////////////////////////////////////////
//                    Fonction - Lecture Inter Mode MANUEL
////////////////////////////////////////////////////////////////////////////////////////////////

void LectureInterModeManuel()
{
  Serial.println("--> Lecture Valeur Inter Mode Manuel");

  ValeurInterModeManuel = digitalRead(A3);
  delay(50);
  Serial.print("ValeurInterModeManuel = ");
  Serial.println(ValeurInterModeManuel);
}



////////////////////////////////////////////////////////////////////////////////////////////////
//                    Fonction - Lecture Inter Infos
////////////////////////////////////////////////////////////////////////////////////////////////

void LectureInterInfos()
{
  // Lecture du ValeurInterInfos
  Serial.println("--> Lecture de Valeur Inter Infos");

  ValeurInterInfos = digitalRead(A2);
  delay(50);
  Serial.print("ValeurInterInfos = ");
  Serial.println(ValeurInterInfos);
}



////////////////////////////////////////////////////////////////////////////////////////////////
//                    Fonction - Lecture Inter Test Matos
////////////////////////////////////////////////////////////////////////////////////////////////

void LectureInterTestMatos()
{
  // Lecture du Valeur Inter Infos
  Serial.println("--> Lecture de Valeur Inter Test Matos");

  ValeurInterTestMatos = digitalRead(A6);
  delay(50);
  Serial.print("ValeurInterTestMatos = ");
  Serial.println(ValeurInterTestMatos);
}



////////////////////////////////////////////////////////////////////////////////////////////////
//                    Fonction - Lecture Inter nourriture
////////////////////////////////////////////////////////////////////////////////////////////////

void LectureInterFood()
{
  // Lecture du ValeurInterFood
  Serial.println("--> Lecture de Valeur Inter Food");

  ValeurInterFood = digitalRead(A5);
  delay(50);
  Serial.print("ValeurInterFood = ");
  Serial.println(ValeurInterFood);
}
```

```
//***********************************/   /***********************************/   /***********************************/
//                                          Fonction -  Gestion des led suivant inter
//***********************************/   /***********************************/   /***********************************/

////// LED Porte

void GestionAffichageLED()
{
 Serial.println("--> Gestion Panneau LED infos");

 // Si la porte est fermee allumage led  fermee
 if (ValeurInterF == 1)
 {
  digitalWrite(Led_Fermee, HIGH);
  Valeur_LED_Bleu = 1;
  Serial.println("LED ON Bleu");
  delay(500);

  digitalWrite(Led_Fermee, LOW);
  Valeur_LED_Bleu = 0;
  Serial.println("LED OFF Bleu");
  delay(2000);

  digitalWrite(Led_Fermee, HIGH);
  Valeur_LED_Bleu = 1;
  Serial.println("LED ON Bleu");
 }
 else
 {
  digitalWrite(Led_Fermee, LOW);
  Valeur_LED_Bleu = 0;
  Serial.println("LED OFF Bleu");
 }



 // Si la porte est ouverte allumage led (cligno) ouvert
 if (ValeurInterO == 1)
 {
  digitalWrite(Led_Ouvert, HIGH);
  Valeur_LED_Blanche = 1;
  Serial.println("LED ON Blanche");
  delay(500);

  digitalWrite(Led_Ouvert, LOW);
  Valeur_LED_Blanche = 0;
  Serial.println("LED OFF Blanche");
  delay(2000);

  digitalWrite(Led_Ouvert, HIGH);
  Valeur_LED_Blanche = 1;
  Serial.println("LED ON Blanche");

 }
 else
 {
  digitalWrite(Led_Ouvert, LOW);
  Valeur_LED_Blanche = 0;
  Serial.println("LED OFF Blanche");
 }


 ////// Led Chauffage

 if (Valeur_Relais_Chauffage == 1)
 {
  digitalWrite(LEDChauffage, HIGH);
  Serial.println("LED ON Chauffage");
 }
 else
```

```
  {
    digitalWrite(LEDChauffage, LOW);
    Serial.println("LED OFF Chauffage");
  }



  // Porte bloquee 0 pour les 2 inter
  if ((ValeurInterO == 0) && (ValeurInterF == 0))
  {
    Serial.println("LED ON 0 pour les 2 inter");
    digitalWrite(Led_Rouge_1, HIGH);
    Valeur_Led_Rouge_1 = 1;
  }


  // Porte bloquee    1 pour les 2 inter
  if ((ValeurInterO == 1) && (ValeurInterF == 1))
  {
    Serial.println("LED ON 1 pour les 2 inter");
    digitalWrite(Led_Rouge_1, HIGH);
    Valeur_Led_Rouge_1 = 1;
  }


  // Porte bloquee Timer out ouverture
  if (PorteBloquageO == 1)
  {
    Serial.println("LED ON Timer out ouverture");
    digitalWrite(Led_Rouge_2, HIGH);
    Valeur_Led_Rouge_2 = 1;
  }

  // Porte bloquee Timer out fermeture
  if (PorteBloquageF == 1)
  {
    Serial.println("LED OFF Timer out Fermeture");
    digitalWrite(Led_Rouge_2, HIGH);
    Valeur_Led_Rouge_2 = 1;
  }


  // Plus de nourriture
  if (ValeurInterFood == 0)
  {
    Serial.println("LED ON Plus de nourriture");
    digitalWrite(Led_Orange_1, HIGH);
    Valeur_Led_Orange_1 = 1;
  }
  else
  {
    Serial.println("LED OFF Plus de nourriture");
    digitalWrite(Led_Orange_1, LOW);
    Valeur_Led_Orange_1 = 0;
  }


}



//***********************************/   /***********************************/   /***********************************/
//                                                  Fonction -  Affichage LCD
//***********************************/   /***********************************/   /***********************************/

void AffichagePrincipal()
{


  delay(250);

  Serial.println("--> Affichage principal");


  lcd.setCursor(0, 0);
  lcd.print("XXXXXXXXXXXXXXXXXXXX");
```

```
lcd.setCursor(0, 1);
lcd.print("XXXXXXXXXXXXXXXXXXXX");
lcd.setCursor(0, 2);
lcd.print("XXXXXXXXXXXXXXXXXXXX");
lcd.setCursor(0, 3);
lcd.print("XXXXXXXXXXXXXXXXXXXX");
delay(250);


//purge de l'écran
lcd.clear();

// positionnement Ligne 0
lcd.setCursor(0, 0);

// Affichage des infos concernant la date et l'heure
lcd.print(RTCJour);
lcd.print("/");
lcd.print(RTCMois);
lcd.print("/");
lcd.print(RTCAnnee);
lcd.print("      ");
lcd.print(RTCHeure);
lcd.print(":");
lcd.print(RTCMinute);


// passage de flottant a entier pour Pb affichage
int IntDHT11_Valeur_Int_Temperature = DHT11_Valeur_Int_Temperature;
int IntDHT11_Valeur_Ext_Temperature = DHT11_Valeur_Ext_Temperature;
int IntDHT11_Valeur_Int_Humidite = DHT11_Valeur_Int_Humidite;
int IntDHT11_Valeur_Ext_Humidite = DHT11_Valeur_Ext_Humidite;


// positionnement Ligne 2
lcd.setCursor(0, 1);

lcd.print("  T.I: ");
lcd.print(IntDHT11_Valeur_Int_Temperature);
lcd.print("  ");
lcd.print(" H.I: ");
lcd.print(IntDHT11_Valeur_Int_Humidite);
lcd.print(" ");


// positionnement Ligne 3
lcd.setCursor(0, 2);

lcd.print("  T.E: ");
lcd.print(IntDHT11_Valeur_Ext_Temperature);
lcd.print("  ");
lcd.print(" H.E: ");
lcd.print(IntDHT11_Valeur_Ext_Humidite);
lcd.print(" ");


// positionnement Ligne 4
lcd.setCursor(0, 3);

lcd.print("P:");
lcd.print(ValeurInterF);

lcd.print(" V:");
lcd.print(Valeur_Servo_Verrou);

lcd.print(" PP:");
lcd.print(Valeur_Servo_Porte_Pondoir);

lcd.print(" C:");
lcd.print(Valeur_Relais_Chauffage);

lcd.print(" L:");
lcd.print(Valeur_Relais_Eclairage);
```

```
  /*
    while ((RTCSeconde != 59) || (RTCSeconde != 0))
    {
      LectureRTC();
      delay(500);
    }
  */
}



//***********************************/   /***********************************/   /***********************************/
//                                                    Gestion PortePondoir
//***********************************/   /***********************************/   /***********************************/

void GestionPondoir()
{

  ////////////////////////////////////////////////////////////////////////////////
  // Gestion PortePondoir Matin- Ouverture entre 8h00 et 18h30 par tranche de 30 min
  ////////////////////////////////////////////////////////////////////////////////

  // La fonction ci-dessous sert aussi pour la sécurité concernant le fait qu'une poule pourrait fermer une des deux portes

  if ((RTCHeure == 8) || (RTCHeure == 9) || (RTCHeure == 10) || (RTCHeure == 11) || (RTCHeure == 12) || (RTCHeure == 13) || (RTCHeure == 14) || (RTCHeure
== 15) || (RTCHeure == 16) || (RTCHeure == 17) || (RTCHeure == 18))
  {
    if ((RTCMinute == 0) || (RTCMinute == 30))
    {
      Serial.println("Gestion PortePondoir du Matin");
      GestionPortePondoirMatin();
      delay(65000);
    }
  }

  ////////////////////////////////////////////////////////
  // ouvertures exceptionnelles
  ////////////////////////////////////////////////////////
  /*
    // Janvier 7h00

    if (RTCMois == 1)
    {
      if (RTCHeure == 7)
      {
        if ((RTCMinute == 0) || (RTCMinute == 30))
        {
          Serial.println("Gestion PortePondoir du Matin");
          GestionPortePondoirMatin();
          delay(65000);
        }
      }
    }


    // Fevrier 7h00

    if (RTCMois == 2)
    {
      if (RTCHeure == 7)
      {
        if ((RTCMinute == 0) || (RTCMinute == 30))
        {
          Serial.println("Gestion PortePondoir du Matin");
          GestionPortePondoirMatin();
          delay(65000);
        }
      }
    }
  */
```

```
  ////////////////////////////////////////////////////////////////////////////////
  // Gestion de la fermeture du pondoir pour fermeture à 17h00-17h15-17h30-17h45
  ////////////////////////////////////////////////////////////////////////////////
  /*
    if (RTCHeure == 17)
    {
      if ((RTCMinute == 0) || (RTCMinute == 15) || (RTCMinute == 30) || (RTCMinute == 45))
      {
        Serial.println("Gestion PortePondoir du Soir desactivée");
        GestionPortePondoirSoir();
        delay(65000);
      }
    }
    */
}



////////////////////////////////////////////////////////////////////////////////////////
//                Fonction -  ouverture du PortePondoir Matin
////////////////////////////////////////////////////////////////////////////////////////

void GestionPortePondoirMatin()
{
  Serial.println("--> Gestion Porte Pondoir Matin");

  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 0);

  lcd.print("--------------------");


  lcd.setCursor(0, 1);

  lcd.print("  Ouverture portes  ");


  lcd.setCursor(0, 2);

  lcd.print("   du  pondoir !    ");


  lcd.setCursor(0, 3);

  lcd.print("--------------------");


  delay(2000);


  digitalWrite(Led_Orange_2, HIGH);
  Valeur_Led_Orange_2 = 1;
  digitalWrite(Led_Verte, HIGH);
  Valeur_Led_Verte = 1;


  /*
    // verrou
    Serial.println("Servo Verrou");
    Servo_Verrou.attach(40);
    delay(250);
    Servo_Verrou.write(120);
    delay(1000);
    Servo_Verrou.detach();
  */


  // PortePondoir
  Serial.println("Servos PortePondoir");
  Servo_Porte_Pondoir.attach(39);
```

```
  delay(250);
  Servo_Porte_Pondoir.write(75);
  delay(2000);
  Servo_Porte_Pondoir.detach();


  digitalWrite(Led_Verte, LOW);
  Valeur_Led_Verte = 0;
  digitalWrite(Led_Orange_2, LOW);
  Valeur_Led_Orange_2 = 0;
  delay(250);

  Valeur_Servo_Porte_Pondoir = 0;

}



////////////////////////////////////////////////////////////////////////////////////////////////
//                          Fonction -  Fermeture du PortePondoir Soir
////////////////////////////////////////////////////////////////////////////////////////////////

void GestionPortePondoirSoir()
{
  Serial.println("--> Gestion PortePondoir soir");


  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 0);

  lcd.print("--------------------");


  lcd.setCursor(0, 1);

  lcd.print("  Fermeture portes  ");


  lcd.setCursor(0, 2);

  lcd.print("    du pondoir !    ");


  lcd.setCursor(0, 3);

  lcd.print("--------------------");


  delay(2000);

  digitalWrite(Led_Orange_2, HIGH);
  Valeur_Led_Orange_2 = 1;
  digitalWrite(Led_Verte, HIGH);
  Valeur_Led_Verte = 1;

  Serial.println("Activation Relais Servos PortePondoir");
  Servo_Porte_Pondoir.attach(39);
  delay(250);
  Servo_Porte_Pondoir.write(153);
  delay(2000);
  Servo_Porte_Pondoir.detach();


  digitalWrite(Led_Verte, LOW);
  Valeur_Led_Verte = 0;
  digitalWrite(Led_Orange_2, LOW);
  Valeur_Led_Orange_2 = 0;
  delay(250);

  Valeur_Servo_Porte_Pondoir = 1;


}
```

```
//***********************************/    /***********************************/    /***********************************/
//                                        Fonction - Gestion de la lumiere de la porte suivant les saisons
//***********************************/    /***********************************/    /***********************************/

void GestionEclairageExt()
{
  // allumage de la lumière extérieure avant fermeture de la porte
  // permet aux poules de retrouver leur chemin si elles sont dehors la nuit tombée

  Serial.println("--> Eclairage exterieur");

  // Janvier
  if ((RTCMois == 1) && (RTCHeure == 18))
  {
    Serial.println("Eclairage Janvier ");
    digitalWrite(Relais_Eclairage, HIGH);
    Valeur_Relais_Eclairage = 1;
  }
  if ((RTCMois == 1) && (RTCHeure != 18))
  {
    Serial.println("Non Eclairage Janvier ");
    digitalWrite(Relais_Eclairage, LOW);
    Valeur_Relais_Eclairage = 0;
  }


  // Fevrier
  if ((RTCMois == 2) && (RTCHeure == 18))
  {
    Serial.println("Eclairage Fevrier");
    digitalWrite(Relais_Eclairage, HIGH);
    Valeur_Relais_Eclairage = 1;
  }
  if ((RTCMois == 2) && (RTCHeure != 18))
  {
    Serial.println("Non Eclairage Fevrier ");
    digitalWrite(Relais_Eclairage, LOW);
    Valeur_Relais_Eclairage = 0;
  }


  // Mars
  if ((RTCMois == 3) && (RTCHeure == 19))
  {
    Serial.println("Eclairage Mars");
    digitalWrite(Relais_Eclairage, HIGH);
    Valeur_Relais_Eclairage = 1;
  }
  if ((RTCMois == 3) && (RTCHeure != 19))
  {
    Serial.println("Non Eclairage Mars ");
    digitalWrite(Relais_Eclairage, LOW);
    Valeur_Relais_Eclairage = 0;
  }


  // Avril
  if ((RTCMois == 4) && (RTCHeure == 20))
  {
    Serial.println("Eclairage Avril");
    digitalWrite(Relais_Eclairage, HIGH);
    Valeur_Relais_Eclairage = 1;
  }
  if ((RTCMois == 4) && (RTCHeure != 20))
```

```
{
  Serial.println("Non Eclairage Avril ");
  digitalWrite(Relais_Eclairage, LOW);
  Valeur_Relais_Eclairage = 0;
}


// Mai
if ((RTCMois == 5) && (RTCHeure == 21))
{
  Serial.println("Eclairage Mai");
  digitalWrite(Relais_Eclairage, HIGH);
  Valeur_Relais_Eclairage = 1;
}
if ((RTCMois == 5) && (RTCHeure != 21))
{
  Serial.println("Non Eclairage Mai ");
  digitalWrite(Relais_Eclairage, LOW);
  Valeur_Relais_Eclairage = 0;
}


// Juin
if ((RTCMois == 6) && (RTCHeure == 22))
{
  Serial.println("Eclairage Juin");
  digitalWrite(Relais_Eclairage, HIGH);
  Valeur_Relais_Eclairage = 1;
}
if ((RTCMois == 6) && (RTCHeure != 22))
{
  Serial.println("Non Eclairage Juin ");
  digitalWrite(Relais_Eclairage, LOW);
  Valeur_Relais_Eclairage = 0;
}


// Juillet

if ((RTCMois == 7) && (RTCHeure == 22))
{
  Serial.println("Eclairage Juillet");
  digitalWrite(Relais_Eclairage, HIGH);
  Valeur_Relais_Eclairage = 1;
}
if ((RTCMois == 7) && (RTCHeure != 22))
{
  Serial.println("Non Eclairage Juillet ");
  digitalWrite(Relais_Eclairage, LOW);
  Valeur_Relais_Eclairage = 0;
}


// Aout
if ((RTCMois == 8) && (RTCHeure == 21))
{
  Serial.println("Eclairage Aout");
  digitalWrite(Relais_Eclairage, HIGH);
  Valeur_Relais_Eclairage = 1;
}
if ((RTCMois == 8) && (RTCHeure != 21))
{
  Serial.println("Non Eclairage Aout ");
  digitalWrite(Relais_Eclairage, LOW);
  Valeur_Relais_Eclairage = 0;
}


// Septembre
if ((RTCMois == 9) && ((RTCHeure == 20) || (RTCHeure == 7)))
{
  Serial.println("Eclairage Septembre");
  digitalWrite(Relais_Eclairage, HIGH);
  Valeur_Relais_Eclairage = 1;
}
if ((RTCMois == 9) && ((RTCHeure != 20) && (RTCHeure != 7)))
```

```
  {
   Serial.println("Non Eclairage Septembre ");
   digitalWrite(Relais_Eclairage, LOW);
   Valeur_Relais_Eclairage = 0;
  }


  // Octobre
  if ((RTCMois == 10) && (RTCHeure == 18))
  {
   Serial.println("Eclairage Octobre");
   digitalWrite(Relais_Eclairage, HIGH);
   Valeur_Relais_Eclairage = 1;
  }
  if ((RTCMois == 10) && (RTCHeure != 18))
  {
   Serial.println("Non Eclairage Octobre ");
   digitalWrite(Relais_Eclairage, LOW);
   Valeur_Relais_Eclairage = 0;
  }


  // Novembre
  if ((RTCMois == 11) && (RTCHeure == 18))
  {
   Serial.println("Eclairage Novembre");
   digitalWrite(Relais_Eclairage, HIGH);
   Valeur_Relais_Eclairage = 1;
  }
  if ((RTCMois == 11) && (RTCHeure != 18))
  {
   Serial.println("Non Eclairage Novembre ");
   digitalWrite(Relais_Eclairage, LOW);
   Valeur_Relais_Eclairage = 0;
  }


  // Decembre
  if ((RTCMois == 12) && (RTCHeure == 17))
  {
   Serial.println("Eclairage Decembre");
   digitalWrite(Relais_Eclairage, HIGH);
   Valeur_Relais_Eclairage = 1;
  }
  if ((RTCMois == 12) && (RTCHeure != 17))
  {
   Serial.println("Non Eclairage Decembre ");
   digitalWrite(Relais_Eclairage, LOW);
   Valeur_Relais_Eclairage = 0;
  }

}





//************************************/   /************************************/   /************************************/   //
                                          Fonctions concernant le  Chauffage
//************************************/   /************************************/   /************************************/


////////////////////////////////////////////////////////////////////////////////////
//                Fonction de Gestion du Chauffage fermeture
////////////////////////////////////////////////////////////////////////////////////

void ChauffagePostFermeture()
{
 Serial.println("--> Gestion chauffage lors de la fermeture");

 // Petite chauffe de 10 min lors de la fermeture de la porte

 Serial.println("Allumage du Chauffage pdt 10 min");
```

```
  digitalWrite(RelaisChauffage, LOW); //logique inverse
  Valeur_Relais_Chauffage = 1;

  // Allumage LED de chauffage
  Serial.println("Allumage LED de chauffage");
  digitalWrite(LEDChauffage, HIGH);
  ValeurLEDChauffage = 1;

  lcd.setCursor(0, 0);
  lcd.print("--------------------");
  lcd.setCursor(0, 1);
  lcd.print("  !  Petite  !     ");
  lcd.setCursor(0, 2);
  lcd.print("  !  Chauffe !     ");
  lcd.setCursor(0, 3);
  lcd.print("--------------------");
  delay(250);

  // Delais de 10min de chauff
  Serial.println("Delais de 10min de chauff");
  delay(600000);

  Serial.println("Exctinction du Chauffage");
  digitalWrite(RelaisChauffage, HIGH);
  Valeur_Relais_Chauffage = 0;


  // Exctinction LED de chauffage
  Serial.println("Exctinction LED de chauffage");
  digitalWrite(LEDChauffage, LOW);
  ValeurLEDChauffage = 0;


}









////////////////////////////////////////////////////////////////////////////////////
//                    Fonction de Gestion du Chauffage Classique
////////////////////////////////////////////////////////////////////////////////////
void GestionChauffage()
{
  if ((RTCMois == 1) || (RTCMois == 2) || (RTCMois == 3) || (RTCMois == 4) || (RTCMois == 10) || (RTCMois == 11) || (RTCMois == 12))
  {
    Chauffage();
  }
}




void Chauffage()
{
  Serial.print("--> Gestion du chauffage Classique ");

  if (Timer_Chauffage < millis())
  {

    if (ValeurInterF == 1)
    {

      if ((DHT11_Valeur_Int_Temperature <= 8))
      {
        Serial.println(" - Allumage du Chauffage");
        digitalWrite(RelaisChauffage, LOW);
        Valeur_Relais_Chauffage = 1;

        // Allumage LED de chauffage
        digitalWrite(LEDChauffage, HIGH);
        ValeurLEDChauffage = 1;
```

```
    }
    else
    {
      Serial.println("- Exctinction du Chauffage");
      digitalWrite(RelaisChauffage, HIGH);
      Valeur_Relais_Chauffage = 0;

      // Extinction LED de chauffage
      digitalWrite(LEDChauffage, LOW);
      ValeurLEDChauffage = 0;
    }

  }
  else
  {
    Serial.println("- Exctinction du Chauffage");
    digitalWrite(RelaisChauffage, HIGH);
    Valeur_Relais_Chauffage = 0;

    // Extinction LED de chauffage
    digitalWrite(LEDChauffage, LOW);
    ValeurLEDChauffage = 0;
  }



  // Timer mini de chauff et de pause - regle a 5 min mini
  Timer_Chauffage = millis() + 300000;

 }

 if (RTCHeure == 7)
 {
  Serial.println("- Exctinction du Chauffage");
  digitalWrite(RelaisChauffage, HIGH);
  Valeur_Relais_Chauffage = 0;

  // Extinction LED de chauffage
  digitalWrite(LEDChauffage, LOW);
  ValeurLEDChauffage = 0;
 }
}









//**********************************/   /**********************************/   /**********************************/
//                                                    GestionPortePrincipale
//**********************************/   /**********************************/   /**********************************/


void GestionPortePrincipale()
{


 // Gestion Mode Manuel ou Auto
 if (ValeurInterModeManuel == 0)
 {


  ////////////////////////////////////////////////////////////
  //     Gestion Ouverture porte auto suivant les saisons
  ////////////////////////////////////////////////////////////
  Serial.println("-->Detection Ouverture de porte");

  // gestion de l'ouverture du lundi au samedi
```

```
  // ouverture obligatoire de 9h00 a 16h59 (concerne tous les mois mais ouverture a 9h00 pour Decembre / Janvier / Fevrier / Mars
  if ((RTCHeure == 9) || (RTCHeure == 10) || (RTCHeure == 11) || (RTCHeure == 12) || (RTCHeure == 13) || (RTCHeure == 14) || (RTCHeure == 15) ||
(RTCHeure == 16))
  {
    OuverturePorte();
  }


  //// SAUF :


  // pour Avril / Mai / Aout / Septembre ouverture des 7h00
  if (((RTCMois == 4) || (RTCMois == 5) || (RTCMois == 8) || (RTCMois == 9)) && (RTCHeure == 7))
  {
    OuverturePorte();
  }

  // pour Juin ouverture des 6h30
  if ((RTCMois == 6) && (RTCHeure == 6) && (RTCMinute > 29))
  {
    OuverturePorte();
  }

  // pour Juillet / Aout ouverture des 6h15
  if (((RTCMois == 7)) && ((RTCHeure == 6) && (RTCMinute > 14)))
  {
    OuverturePorte();
  }

  // pour Mars / Octobre / Novembre ouverture des 7h30
  if (((RTCMois == 3) || (RTCMois == 10) || (RTCMois == 11) ) && ((RTCHeure == 7) && (RTCMinute > 29)))
  {
    OuverturePorte();
  }

  // gestion de l'ouverture pour samedi et dimanche à partir de 9h30 pour tous les mois - a modifier avec le temps









  ////////////////////////////////////////////////////////////////////////////////////////////
  //        Gestion Fermeture porte auto variable suivant les mois de l année
  ////////////////////////////////////////////////////////////////////////////////////////////

  Serial.println("--> Detection Fermeture de porte");


  if ((RTCMois == 1) && (RTCHeure == 19) && (ValeurInterF != 1))
  {
    Serial.println("--> Janvier");
    FermeturePorte();
    ChauffagePostFermeture();
  }


  if ((RTCMois == 2) && (RTCHeure == 19) && (ValeurInterF != 1))
  {
    Serial.println("--> Fevrier");
    FermeturePorte();
    ChauffagePostFermeture();
  }


  if ((RTCMois == 3) && (RTCHeure == 20) && (ValeurInterF != 1))
  {
    Serial.println("--> Mars");
    FermeturePorte();
    ChauffagePostFermeture();
  }
```

```
if ((RTCMois == 4) && (RTCHeure == 21) && (RTCMinute > 30) && (ValeurInterF != 1))
{
  Serial.println("--> Avril");
  FermeturePorte();
}


if ((RTCMois == 5) && (RTCHeure == 22)  && (ValeurInterF != 1))
{
  Serial.println("-->  Mai");
  FermeturePorte();
}


if ((RTCMois == 6) && (RTCHeure == 22) && (RTCMinute > 40) && (ValeurInterF != 1))
{
  Serial.println("--> Juin");
  FermeturePorte();
}

if ((RTCMois == 7) && (RTCHeure == 22) && (RTCMinute > 30) && (ValeurInterF != 1))
{
  Serial.println("--> Juillet");
  FermeturePorte();
}

// 0 a 15
if ((RTCMois == 8) && (RTCJour < 16) && (RTCHeure == 22) && (ValeurInterF != 1))
{
  Serial.println("--> Aout");
  FermeturePorte();
}

// 16 a 31
if ((RTCMois == 8) && (RTCJour > 15) && (RTCHeure == 21) && (RTCMinute > 29) && (ValeurInterF != 1))
{
  Serial.println("--> Aout");
  FermeturePorte();
}


// 0 a 15
if ((RTCMois == 9) && (RTCJour < 16) && (RTCHeure == 21) && (ValeurInterF != 1))
{
  Serial.println("--> Septembre");
  FermeturePorte();
}

// 16 a 31
if ((RTCMois == 9) && (RTCJour > 15) && (RTCHeure == 20) && (RTCMinute > 29) && (ValeurInterF != 1))
{
  Serial.println("--> Septembre");
  FermeturePorte();
}


if ((RTCMois == 10) && (RTCJour < 16) && (RTCHeure == 20) && ((RTCMinute == 00) || (RTCMinute == 05)) && (ValeurInterF != 1))
{
  Serial.println("--> Octobre");
  FermeturePorte();
}


if ((RTCMois == 10) && (RTCJour > 15) && (RTCHeure == 19) && ((RTCMinute == 45) || (RTCMinute == 50)) && (ValeurInterF != 1))
{
  Serial.println("--> Octobre");
  FermeturePorte();
  ChauffagePostFermeture();
}




if ((RTCMois == 11) && (RTCHeure == 18) && ((RTCMinute == 29) || (RTCMinute == 30)) && (ValeurInterF != 1))
{
```

```
    Serial.println("--> Novembre < 15");
    FermeturePorte();
    ChauffagePostFermeture();
  }


  if ((RTCMois == 11) && (RTCJour > 15) && (RTCHeure == 18) && (ValeurInterF != 1))
  {
    Serial.println("--> Novembre > 15");
    FermeturePorte();
    ChauffagePostFermeture();
  }


  if ((RTCMois == 12) && (RTCHeure == 18) && (ValeurInterF != 1))
  {
    Serial.println("--> Decembre");
    FermeturePorte();
    ChauffagePostFermeture();
  }

 }
 else
 {
  Serial.println("Activation du mode manuel");

  lcd.clear();

  lcd.setCursor(0, 1);
  lcd.print("     MODE MANUEL    ");

  // Affichage des led ouvert et fermee afin d'avertir le mode manuel via led
  digitalWrite(Led_Fermee, HIGH);
  digitalWrite(Led_Ouvert, HIGH);

  delay(15000);
 }

 // conversion des variable pour l enoiie serie
 Serial.println("Conversion des data");
 Conversion();

 // envoie des variables sur le port serie
 Serial.println("Envoi des infos sur port serie");
 Envoie_Infos();

 // delais  qui sert pour la synchro sinon c trop rapide
 delay(15000);



}




////////////////////////////////////////////////////////////////////////////////////////////
//                    Fonction pour l'ouverture de la porte
////////////////////////////////////////////////////////////////////////////////////////////

void OuverturePorte()
{

 if (ValeurInterO != 1)
 {
  Serial.println("--> Ouverture de la porte");

  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 0);

  lcd.print("--------------------");


  lcd.setCursor(0, 1);
```

```
lcd.print("     Gestion      ");


lcd.setCursor(0, 2);

lcd.print("  PORTE PRINCIPALE  ");


lcd.setCursor(0, 3);

lcd.print("     du matin     ");


delay(2000);

// allumage de la led verte et orange
digitalWrite(Led_Orange_2, HIGH);
Valeur_Led_Orange_2 = 1;
digitalWrite(Led_Verte, HIGH);
Valeur_Led_Verte = 1;


// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
lcd.setCursor(0, 0);

lcd.print("--------------------");


lcd.setCursor(0, 1);

lcd.print("    Ouverture     ");


lcd.setCursor(0, 2);

lcd.print("     verrou       ");


lcd.setCursor(0, 3);

lcd.print("--------------------");



delay(2000);


// ouverture du verrou
Serial.println("Servo Verrou");


Servo_Verrou.attach(40);
delay(250);
Servo_Verrou.write(120);
delay(1000);
Servo_Verrou.detach();


Valeur_Servo_Verrou = 0;

// On pose un timer de 30 sec pour la porte en cas de pb
Serial.println("On pose un timer de 30 sec pour la porte en cas de pb");
TimerBPO = millis() + 30000;





// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
lcd.setCursor(0, 0);

lcd.print("--------------------");
```

```
  lcd.setCursor(0, 1);

  lcd.print("   Ouverture de la  ");


  lcd.setCursor(0, 2);

  lcd.print("       porte       ");


  lcd.setCursor(0, 3);

  lcd.print("--------------------");


  delay(2000);


  Servo_Porte_Principale.attach(41);
  delay(250);

  // Tant que l'on est pas en butee
  while (ValeurInterO == 0)
  {
    Serial.println("Ouverture de la porte tant que l on est pas en butee");

    digitalWrite(Led_Fermee, HIGH);
    digitalWrite(Led_Ouvert, HIGH);

    // active le servo


    // ancienne valeur au 14/08/2016 Servo_Porte_Principale.write(108);
    Servo_Porte_Principale.write(125);



    delay(150); // permet d 'avoir un peu d'inertie

    // Lecture de l etat des capteur inter
    Serial.print("Lecture de l etat des capteur inter = ");
    ValeurInterO = digitalRead(A1);
    delay(10);

    // Si la porte met plus de 20 sec a se ouvrir -> pb on arrete
    // Pour ca on active la variable PorteBloquageO et on casse la boucle
    Serial.println("si la porte met plus de 20 sec a se ouvrir -> pb on arrete");

    if (TimerBPO < millis())
    {
      PorteBloquageO = 1;
      // desactive la led verte
      digitalWrite(Led_Verte, LOW);
      Valeur_Led_Verte = 0;
      break;
    }

    digitalWrite(Led_Fermee, LOW);
    digitalWrite(Led_Ouvert, LOW);

    delay(150);
  }

  // desactive le servo
  Servo_Porte_Principale.detach();



  //Desactive les leds
  digitalWrite(Led_Orange_2, LOW);
  Valeur_Led_Orange_2 = 0;
  digitalWrite(Led_Verte, LOW);
  Valeur_Led_Verte = 0;
}
```

```
}




////////////////////////////////////////////////////////////////////////////////////////////
//                    Fonction pour la fermeture de la porte
////////////////////////////////////////////////////////////////////////////////////////////

void FermeturePorte()
{
  Serial.println("--> Fermeture de la porte");


  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 0);

  lcd.print("--------------------");


  lcd.setCursor(0, 1);

  lcd.print("      Gestion       ");


  lcd.setCursor(0, 2);

  lcd.print("  PORTE PRINCIPALE  ");


  lcd.setCursor(0, 3);

  lcd.print("      du soir       ");


  delay(2000);

  // activation led servo
  digitalWrite(Led_Orange_2, HIGH);
  Valeur_Led_Orange_2 = 1;
  digitalWrite(Led_Verte, HIGH);
  Valeur_Led_Verte = 0;


  // verrou
  Serial.println("Servo Verrou");


  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 0);

  lcd.print("--------------------");


  lcd.setCursor(0, 1);

  lcd.print("     Ouverture      ");


  lcd.setCursor(0, 2);

  lcd.print("      verrou        ");


  lcd.setCursor(0, 3);

  lcd.print("--------------------");


  delay(2000);
```

```
// mouvement servo
Servo_Verrou.attach(40);
delay(250);



Servo_Verrou.write(120);
delay(1000);



Servo_Verrou.detach();


// desactivation led servo
digitalWrite(Led_Orange_2, LOW);
Valeur_Led_Orange_2 = 0;
digitalWrite(Led_Verte, LOW);
Valeur_Led_Verte = 0;


Valeur_Servo_Verrou = 0;


// Si la porte met plus de 30 sec a se fermer -> pb on arrete
// pour ca on active la variable PorteBloquageO et on casse la boucle
Serial.println("Si la porte met plus de 30 sec a se fermer -> pb on arrete");
TimerBPF = millis() + 45000;



// activation led servo
digitalWrite(Led_Orange_2, HIGH);
Valeur_Led_Orange_2 = 1;
digitalWrite(Led_Verte, HIGH);
Valeur_Led_Verte = 1;


//  porte
Serial.println("Servo Moteur Porte");


// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
lcd.setCursor(0, 0);

lcd.print("--------------------");


lcd.setCursor(0, 1);

lcd.print("   Fermeture de la  ");


lcd.setCursor(0, 2);

lcd.print("       porte       ");


lcd.setCursor(0, 3);

lcd.print("--------------------");


delay(2000);


Servo_Porte_Principale.attach(41);
delay(250);



// On ferme la porte
while (ValeurInterF == 0)
```

```
{
  Servo_Porte_Principale.write(70);
  //108

  // Lecture de l etat des capteur inter
  Serial.print("Lecture de l etat du capteur inter = ");
  ValeurInterF = digitalRead(A0);
  delay(10);
  Serial.println(ValeurInterF);

  // Gestion du bloquage & interruption en cas de pb
  Serial.println("Gestion du bloquage & interruption en cas de pb");
  if (TimerBPF < millis())
  {
    Serial.println("Bloquage de la porte delais trop long");
    PorteBloquageF = 1;

    // desactive la led verte
    digitalWrite(Led_Verte, LOW);
    Valeur_Led_Verte = 0;

    break;
  }

}

// desactive le servo
Servo_Porte_Principale.detach();



// desactivation led servo
digitalWrite(Led_Orange_2, LOW);
Valeur_Led_Orange_2 = 0;
digitalWrite(Led_Verte, LOW);


Serial.println("Servo Verrou");



// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
lcd.setCursor(0, 0);

lcd.print("--------------------");


lcd.setCursor(0, 1);

lcd.print("     Fermeture      ");


lcd.setCursor(0, 2);

lcd.print("      verrou        ");


lcd.setCursor(0, 3);

lcd.print("--------------------");


delay(2000);


// desactivation led servo
digitalWrite(Led_Orange_2, HIGH);
Valeur_Led_Orange_2 = 1;
digitalWrite(Led_Verte, HIGH);
Valeur_Led_Verte = 1;


Servo_Verrou.attach(40);
delay(250);
Servo_Verrou.write(0);
```

```
  delay(1000);
  Servo_Verrou.detach();


  // desactivation led servo
  digitalWrite(Led_Orange_2, LOW);
  Valeur_Led_Orange_2 = 0;
  digitalWrite(Led_Verte, LOW);
  Valeur_Led_Verte = 0;

  // Servo est en mode Ferme
  Serial.println("Verrou est en mode Ferme");
  Valeur_Servo_Verrou = 1;

}









//************************************/   /************************************/   /************************************/
//                                          Fonction du cas du double bloquage inter O et F
//************************************/   /************************************/   /************************************/

void DetectionBloquage()
{
  Serial.println("--> Detection Bloquage");

  if (((ValeurInterO == 0) && (ValeurInterF == 0)) || ((ValeurInterO == 1) && (ValeurInterF == 1)))
  {
    Serial.println("ERREUR - Avec les interrupteurs");


    //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!//

    // Par securite desactive le chauffage
    Serial.println("Par securite desactive le chauffage");
    digitalWrite(RelaisChauffage, HIGH);


    // Par securite desactive l'eclairage
    Serial.println("Par securite desactive l eclairage");
    digitalWrite(Relais_Eclairage, LOW);
    Valeur_Relais_Eclairage = 0;


    // Par securite desactive le verrou
    Serial.println("Par securite desactive le verrou");
    Servo_Verrou.detach();


    // Par securite desactive le moteur
    Serial.println("Par securite desactive le moteur ");
    Servo_Porte_Principale.detach();

//  while (1);
  }
}
```

```
//************************************/   /************************************/   /************************************/
//                                                Fonction Affichage des versions
//************************************/   /************************************/   /************************************/

void AffichageVersion()
{
  Serial.println("--> Affichage de la version");
  Serial.print("VERSION PPV5-VF6 DU 11/11/2016 - 12h00");


  delay(250);

  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 0);

  lcd.print("   Version PPFV5-6  ");


  lcd.setCursor(0, 1);

  lcd.print("--------------------");


  lcd.setCursor(0, 2);

  lcd.print("        du        ");


  lcd.setCursor(0, 3);

  lcd.print("  11/11/2016-12h00  ");


  delay(2000);


}




//************************************/   /************************************/   /************************************/
//                                          Fonction - Test Matos.
//************************************/   /************************************/   /************************************/

void GestionTest_Matos()
{
  if (ValeurInterTestMatos == 1)
  {
    Test_Matos();
  }
}



void Test_Matos()
{
  Serial.println("--> Test_Matos");

  lcd.setCursor(0, 0);
  lcd.print("--------------------");

  lcd.setCursor(0, 1);

  lcd.print("       TEST       ");

  lcd.setCursor(0, 2);

  lcd.print("      MATOS       ");

  lcd.setCursor(0, 3);

  lcd.print("--------------------");
```

```
delay(2000);



lcd.clear();
delay(250);



//***// Allume tous les relais
Serial.println("Allume tous les relais");

lcd.setCursor(0, 1);
lcd.print("ON-OFF ts les Relais");

digitalWrite(RelaisChauffage, LOW );
digitalWrite(Relais_Eclairage, HIGH);

delay(2000);



//***// Mise a zero de tous les parametres
Serial.println("Mise a zero de tous les parametres");

lcd.setCursor(0, 1);
lcd.print("Tous les param a 0  ");


digitalWrite(Led_Fermee, LOW);
digitalWrite(Led_Ouvert, LOW);
digitalWrite(LEDChauffage, LOW);
digitalWrite(Led_Verte, LOW);
digitalWrite(Led_Orange_1, LOW);
digitalWrite(Led_Orange_2, LOW);
digitalWrite(Led_Rouge_1, LOW);
digitalWrite(Led_Rouge_2, LOW);
digitalWrite(Led_Rouge_3, LOW);
digitalWrite(RelaisChauffage, HIGH);
digitalWrite(Relais_Eclairage, LOW);

delay(500);



//***//Lecture des capteurs
Serial.println("Lecture capteurs");

lcd.setCursor(0, 1);
lcd.print("Lecture des capteurs");

// Lecture de ValeurInterF
Serial.print("ValeurInterF = ");
ValeurInterF = digitalRead(A0);
Serial.println(ValeurInterF);

// Lecture du ValeurInterO
Serial.print("ValeurInterO = ");
ValeurInterO = digitalRead(A1);
Serial.println(ValeurInterO);

// Lecture du ValeurInterFood
Serial.print("ValeurInterFood = ");
ValeurInterFood = digitalRead(A5);
Serial.println(ValeurInterFood);

// Lecture du ValeurInterInfos
Serial.print("ValeurInterInfos = ");
ValeurInterInfos = digitalRead(A2);
Serial.println(ValeurInterInfos);

// Lecture du ValeurModeManuel
Serial.print("ValeurInterModeManuel = ");
ValeurInterModeManuel = digitalRead(A3);
Serial.println(ValeurInterModeManuel);
```

```
// Lecture du ValeurInterTestMatos
Serial.print("ValeurInterTestMatos = ");
ValeurInterTestMatos = digitalRead(A6);
Serial.println(ValeurInterTestMatos);



//***// test les relais
Serial.println("Test tous les relais");

lcd.setCursor(0, 1);
lcd.print("Test ts les relais  ");

Serial.println("Test Relais Chauffage LOW");
digitalWrite(RelaisChauffage, LOW); // High car c'est la logique inverse
delay(2000);

Serial.println("Test Relais Chauffage HIGH");
digitalWrite(RelaisChauffage, HIGH);

delay(1000);
Serial.println("Test Relais Chauffage LOW");
digitalWrite(RelaisChauffage, LOW); // High car c'est la logique inverse

delay(2000);
Serial.println("Test Relais Chauffage HIGH");
digitalWrite(RelaisChauffage, HIGH);
delay(1000);

Serial.println("Test Relais_Eclairage HIGH ");
digitalWrite(Relais_Eclairage, HIGH);
delay(1000);

Serial.println("Test Relais_Eclairage LOW");
digitalWrite(Relais_Eclairage, LOW);
delay(1000);



//***// Test du verrou
Serial.println("Test verrou");

lcd.setCursor(0, 1);
lcd.print("Test  verrou       ");

digitalWrite(Led_Orange_2, HIGH);
Valeur_Led_Orange_2 = 1;
digitalWrite(Led_Verte, HIGH);
Valeur_Led_Verte = 1;

// test du verrou
Serial.println("Test Servo Verrou");
Servo_Verrou.attach(40);
delay(250);
Servo_Verrou.write(120);
delay(1000);
Servo_Verrou.write(0);
delay(1000);
Servo_Verrou.write(120);
delay(1000);
Servo_Verrou.detach();



//***// Test de la porte
Serial.println("Test Servo Moteur Porte");

lcd.setCursor(0, 1);
lcd.print("Test  porte        ");

Servo_Porte_Principale.attach(41);
delay(250);
```

```
ValeurInterF = digitalRead(A0);
// fermeture
while (ValeurInterF == 0)
{
  Servo_Porte_Principale.write(70);

  // Lecture de l etat des capteur inter
  Serial.print("Lecture de l etat du capteur inter = ");
  ValeurInterF = digitalRead(A0);
  delay(10);
  Serial.println(ValeurInterF);

}
Servo_Porte_Principale.detach();

delay(3000);

Servo_Porte_Principale.attach(41);
delay(250);


ValeurInterO = digitalRead(A1);
//ouverture
while (ValeurInterO == 0)
{
  Serial.println("Ouverture de la porte tant que l on est pas en butee");

  // active le servo
  Servo_Porte_Principale.write(108);
  delay(200); // permet d'avoir un peu d'inertie

  // Lecture de l etat des capteur inter
  Serial.print("Lecture de l etat des capteur inter = ");
  ValeurInterO = digitalRead(A1);
  delay(250);
}


// desactive le servo
Servo_Porte_Principale.detach();


//***// test du PortePondoir

Serial.println("Test Servos PortePondoir");

lcd.setCursor(0, 1);
lcd.print("Test  pondoir       ");


Servo_Porte_Pondoir.attach(39);
delay(250);
Servo_Porte_Pondoir.write(153);
delay(2000);
Servo_Porte_Pondoir.write(75);
delay(2000);
Servo_Porte_Pondoir.detach();


digitalWrite(Led_Verte, LOW);
Valeur_Led_Verte = 0;
digitalWrite(Led_Orange_2, LOW);
Valeur_Led_Orange_2 = 0;
delay(250);



//***// test leds
Serial.println("Test LED");

lcd.setCursor(0, 1);
lcd.print("Test  LED          ");


Serial.println("Test LED Ouvert");
// test des leds
```

```
digitalWrite(Led_Ouvert, HIGH);
delay(500);
digitalWrite(Led_Ouvert, LOW);

Serial.println("Test LED Fermee");
digitalWrite(Led_Fermee, HIGH);
delay(500);
digitalWrite(Led_Fermee, LOW);

Serial.println("Test LED Chauffage");
digitalWrite(LEDChauffage, HIGH);
delay(500);
digitalWrite(LEDChauffage, LOW);

Serial.println("Test LED Verte");
digitalWrite(Led_Verte, HIGH);
Valeur_Led_Verte = 1;
delay(500);
digitalWrite(Led_Verte, LOW);
Valeur_Led_Verte = 0;

Serial.println("Test LED Orange 1");
digitalWrite(Led_Orange_1, HIGH);
Valeur_Led_Orange_1 = 1;
delay(500);
digitalWrite(Led_Orange_1, LOW);
Valeur_Led_Orange_1 = 0;

Serial.println("Test LED Orange 2");
digitalWrite(Led_Orange_2, HIGH);
Valeur_Led_Orange_2 = 1;
delay(500);
digitalWrite(Led_Orange_2, LOW);
Valeur_Led_Orange_2 = 0;

Serial.println("Test LED Rouge 1");
digitalWrite(Led_Rouge_1, HIGH);
Valeur_Led_Rouge_1 = 1;
delay(500);
digitalWrite(Led_Rouge_1, LOW);
Valeur_Led_Rouge_1 = 0;

Serial.println("Test LED Rouge 2");
digitalWrite(Led_Rouge_2, HIGH);
Valeur_Led_Rouge_2 = 1;
delay(500);
digitalWrite(Led_Rouge_2, LOW);
Valeur_Led_Rouge_2 = 0;

Serial.println("Test LED Rouge 3");
digitalWrite(Led_Rouge_3, HIGH);
Valeur_Led_Rouge_3 = 1;
delay(500);
digitalWrite(Led_Rouge_3, LOW);
Valeur_Led_Rouge_3 = 0;


delay(1000);


//***//Mise a zero de tous les parametres

Serial.println("Mise a zero de tous les parametres");

lcd.setCursor(0, 1);
lcd.print("Ts les param a 0    ");


digitalWrite(Led_Fermee, LOW);
digitalWrite(Led_Ouvert, LOW);
digitalWrite(LEDChauffage, LOW);
digitalWrite(Led_Verte, LOW);
digitalWrite(Led_Orange_1, LOW);
digitalWrite(Led_Orange_2, LOW);
digitalWrite(Led_Rouge_1, LOW);
digitalWrite(Led_Rouge_2, LOW);
```

```
  digitalWrite(Led_Rouge_3, LOW);
  digitalWrite(RelaisChauffage, HIGH);
  digitalWrite(Relais_Eclairage, LOW);

  delay(1000);

  LectureInterF();
  delay(100);
  LectureInterO();
  delay(100);

  lcd.clear();

}




//************************************/   /************************************/   /************************************/
//                                          Fonction Affichage des Infos via interr
//************************************/   /************************************/   /************************************/

void GestionAffichageInfos()
{
  if (ValeurInterInfos == 1)
  {
    AffichageInfos();
  }
}



void AffichageInfos()
{

  lcd.setCursor(0, 0);
  lcd.print("InterF :          ");
  lcd.print(ValeurInterF);

  lcd.setCursor(0, 1);
  lcd.print("InterO :          ");
  lcd.print(ValeurInterO);

  lcd.setCursor(0, 2);
  lcd.print("PorteBloquageO:    ");
  lcd.print(PorteBloquageO);

  lcd.setCursor(0, 3);
  lcd.print("PorteBloquageF:    ");
  lcd.print(PorteBloquageF);

  delay(10000);


}



//************************************/   /************************************/   /************************************/
//                                          Fonction conversion des Infos
//************************************/   /************************************/   /************************************/

void Conversion()
{
  // conversion des float en int
```

```
int DHT11_Valeur_Int_Temperature_Entier = DHT11_Valeur_Int_Temperature;
int DHT11_Valeur_Int_Humidite_Entier = DHT11_Valeur_Int_Humidite;
int DHT11_Valeur_Ext_Temperature_Entier = DHT11_Valeur_Ext_Temperature;
int DHT11_Valeur_Ext_Humidite_Entier = DHT11_Valeur_Ext_Humidite;

delay(100);

// conversion des int en string
STR_RTCAnnee = String(RTCAnnee);
STR_RTCMois = String(RTCMois);
STR_RTCJour = String(RTCJour);
STR_RTCHeure = String(RTCHeure);
STR_RTCMinute = String(RTCMinute);
STR_DHT11_Valeur_Int_Temperature_Entier = String(DHT11_Valeur_Int_Temperature_Entier);
STR_DHT11_Valeur_Int_Humidite_Entier = String(DHT11_Valeur_Int_Humidite_Entier);
STR_DHT11_Valeur_Ext_Temperature_Entier = String(DHT11_Valeur_Ext_Temperature_Entier);
STR_DHT11_Valeur_Ext_Humidite_Entier = String(DHT11_Valeur_Ext_Humidite_Entier);
STR_Valeur_Servo_Porte_Principale = String(Valeur_Servo_Porte_Principale);
STR_ValeurInterF = String(ValeurInterF);
STR_ValeurInterO = String(ValeurInterO);
STR_PorteBloquageO = String(PorteBloquageO);
STR_PorteBloquageF = String(PorteBloquageF);
STR_Valeur_Servo_Verrou = String(Valeur_Servo_Verrou);
STR_Valeur_Servo_Porte_Pondoir = String(Valeur_Servo_Porte_Pondoir);
STR_Valeur_Relais_Chauffage = String(Valeur_Relais_Chauffage);
STR_Valeur_Relais_Eclairage = String(Valeur_Relais_Eclairage);
STR_ValeurInterInfos = String(ValeurInterInfos);
STR_ValeurInterModeManuel = String(ValeurInterModeManuel);
STR_ValeurInterTestMatos = String(ValeurInterTestMatos);
STR_ValeurInterFood = String(ValeurInterFood);
STR_Valeur_LED_Blanche = String(Valeur_LED_Blanche);
STR_Valeur_LED_Bleu = String(Valeur_LED_Bleu);
STR_Valeur_Led_Verte = String(Valeur_Led_Verte);
STR_Valeur_Led_Orange_1 = String(Valeur_Led_Orange_1);
STR_Valeur_Led_Orange_2 = String(Valeur_Led_Orange_2);
STR_Valeur_Led_Rouge_1 = String(Valeur_Led_Rouge_1);
STR_Valeur_Led_Rouge_2 = String(Valeur_Led_Rouge_2);
STR_Valeur_Led_Rouge_3 = String(Valeur_Led_Rouge_3);
STR_ValeurLEDChauffage = String(ValeurLEDChauffage);

delay(100);


// gestion des Zeros pour variable Mois

if (STR_RTCMois == "1")
{
  STR_RTCMois = "O1";
}
if (STR_RTCMois == "2")
{
  STR_RTCMois = "O2";
}
if (STR_RTCMois == "3")
{
  STR_RTCMois = "O3";
}
if (STR_RTCMois == "4")
{
  STR_RTCMois = "O4";
}
if (STR_RTCMois == "5")
{
  STR_RTCMois = "O5";
}
if (STR_RTCMois == "6")
{
  STR_RTCMois = "O6";
}
if (STR_RTCMois == "7")
{
  STR_RTCMois = "O7";
}
if (STR_RTCMois == "8")
{
  STR_RTCMois = "O8";
```

```
}
if (STR_RTCMois == "9")
{
  STR_RTCMois = "O9";
}


// gestion des Zeros pour variable Jour

if (STR_RTCJour == "1")
{
  STR_RTCJour = "O1";
}
if (STR_RTCJour == "2")
{
  STR_RTCJour = "O2";
}
if (STR_RTCJour == "3")
{
  STR_RTCJour = "O3";
}
if (STR_RTCJour == "4")
{
  STR_RTCJour = "O4";
}
if (STR_RTCJour == "5")
{
  STR_RTCJour = "O5";
}
if (STR_RTCJour == "6")
{
  STR_RTCJour = "O6";
}
if (STR_RTCJour == "7")
{
  STR_RTCJour = "O7";
}
if (STR_RTCJour == "8")
{
  STR_RTCJour = "O8";
}
if (STR_RTCJour == "9")
{
  STR_RTCJour = "O9";
}


// gestion des Zeros pour variable Heure

if (STR_RTCHeure == "0")
{
  STR_RTCHeure = "OO";
}
if (STR_RTCHeure == "1")
{
  STR_RTCHeure = "O1";
}
if (STR_RTCHeure == "2")
{
  STR_RTCHeure = "O2";
}
if (STR_RTCHeure == "3")
{
  STR_RTCHeure = "O3";
}
if (STR_RTCHeure == "4")
{
  STR_RTCHeure = "O4";
}
if (STR_RTCHeure == "5")
{
  STR_RTCHeure = "O5";
}
if (STR_RTCHeure == "6")
```

```
{
  STR_RTCHeure = "O6";
}
if (STR_RTCHeure == "7")
{
  STR_RTCHeure = "O7";
}
if (STR_RTCHeure == "8")
{
  STR_RTCHeure = "O8";
}
if (STR_RTCHeure == "9")
{
  STR_RTCHeure = "O9";
}



// gestion des Zeros pour variable Minute

if (STR_RTCMinute == "0")
{
  STR_RTCMinute = "OO";
}
if (STR_RTCMinute == "1")
{
  STR_RTCMinute = "O1";
}
if (STR_RTCMinute == "2")
{
  STR_RTCMinute = "O2";
}
if (STR_RTCMinute == "3")
{
  STR_RTCMinute = "O3";
}
if (STR_RTCMinute == "4")
{
  STR_RTCMinute = "O4";
}
if (STR_RTCMinute == "5")
{
  STR_RTCMinute = "O5";
}
if (STR_RTCMinute == "6")
{
  STR_RTCMinute = "O6";
}
if (STR_RTCMinute == "7")
{
  STR_RTCMinute = "O7";
}
if (STR_RTCMinute == "8")
{
  STR_RTCMinute = "O8";
}
if (STR_RTCMinute == "9")
{
  STR_RTCMinute = "O9";
}



// gestion des Zeros pour variable Temp Int

if (STR_DHT11_Valeur_Int_Temperature_Entier == "0")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "OO";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "1")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O1";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "2")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O2";
```

```
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "3")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O3";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "4")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O4";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "5")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O5";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "6")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O6";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "7")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O7";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "8")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O8";
}
if (STR_DHT11_Valeur_Int_Temperature_Entier == "9")
{
  STR_DHT11_Valeur_Int_Temperature_Entier = "O9";
}


// gestion des Zeros pour variable Temp Ext

if (STR_DHT11_Valeur_Ext_Temperature_Entier == "0")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "OO";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "1")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O1";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "2")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O2";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "3")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O3";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "4")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O4";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "5")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O5";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "6")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O6";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "7")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O7";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "8")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O8";
}
if (STR_DHT11_Valeur_Ext_Temperature_Entier == "9")
{
  STR_DHT11_Valeur_Ext_Temperature_Entier = "O9";
}
```

```
}



//***********************************/   /***********************************/   /***********************************/
//                                                Fonction envoie des Infos
//***********************************/   /***********************************/   /***********************************/

void Envoie_Infos()
{

  // envoie des infos

  mySerial.print(STR_RTCAnnee);
  mySerial.print(STR_RTCMois);
  mySerial.print(STR_RTCJour);
  mySerial.print(STR_RTCHeure);
  mySerial.print(STR_RTCMinute);
  mySerial.print(STR_DHT11_Valeur_Int_Temperature_Entier);
  mySerial.print(STR_DHT11_Valeur_Int_Humidite_Entier);
  mySerial.print(STR_DHT11_Valeur_Ext_Temperature_Entier);
  mySerial.print(STR_DHT11_Valeur_Ext_Humidite_Entier);
  mySerial.print(STR_Valeur_Servo_Porte_Principale);
  mySerial.print(STR_ValeurInterF);
  mySerial.print(STR_ValeurInterO);
  mySerial.print(STR_PorteBloquageO);
  mySerial.print(STR_PorteBloquageF);
  mySerial.print(STR_Valeur_Servo_Verrou);
  mySerial.print(STR_Valeur_Servo_Porte_Pondoir);
  mySerial.print(STR_Valeur_Relais_Chauffage);
  mySerial.print(STR_Valeur_Relais_Eclairage);
  mySerial.print(STR_ValeurInterInfos);
  mySerial.print(STR_ValeurInterModeManuel);
  mySerial.print(STR_ValeurInterTestMatos);
  mySerial.print(STR_ValeurInterFood);

  mySerial.print("X");//fin de premier envoie // obligé de mettre ceci sinon une variable manque, surement un pb de memoire de trame

  mySerial.print(STR_Valeur_LED_Blanche);
  mySerial.print(STR_Valeur_LED_Bleu);
  mySerial.print(STR_Valeur_Led_Verte);
  mySerial.print(STR_Valeur_Led_Orange_1);
  mySerial.print(STR_Valeur_Led_Orange_2);
  mySerial.print(STR_Valeur_Led_Rouge_1);
  mySerial.print(STR_Valeur_Led_Rouge_2);
  mySerial.print(STR_Valeur_Led_Rouge_3);
  mySerial.print(STR_ValeurLEDChauffage);

}
```

Modules de la deuxieme carte :

- Module ethershield

Connectivité – Branchement des modules de la deuxieme carte :

Port serie (11, 10); // RX, TX donc 10 au 22 et 11 au 23

Architecture du programme de la deuxieme carte

```
// Definition des broches utilisees pour module serie et bibliotheques associees
// Variable tampon de reception.
// Reception des infos caractere par caractere
// Declaration des variables qui seront envoyées sur le serveur SQL
// Definition des broches utilisees pour module ethernet et bibliotheques associees

setup()

// Ouverture du port info
// Ouverture du port serie
// initialisation de la carte RX
// Start Ethernet connection:


LOOP

// Recuperation des donnes du port serie
// Envoie de la requete au SRV SQL

Fonctions

// Fonction -  Connexion HTTP
// Fonction -  Requete HTTP
// Fonction -  Recuperation des donnes via la liaison serie
```

Programme de la deuxieme carte

```
////////////////////////////////////////////////////////////////////////////////////////////
//    Definition des broches utilisees pour module serie et bibliotheques associees
////////////////////////////////////////////////////////////////////////////////////////////

#include <SoftwareSerial.h>

SoftwareSerial mySerial(11, 10); // RX, TX donc  10 au 22 et 11 au 23


// variable tampon de reception.
// on recoit les infos charactere par charactere

char var1;
char var2;
char var3;
char var4;
char var5;
char var6;
char var7;
char var8;
char var9;
char var10;
char var11;
char var12;
char var13;
char var14;
char var15;
char var16;
char var17;
char var18;
char var19;
char var20;
char var21;
char var22;
char var23;
```

```
char var24;
char var25;
char var26;
char var27;
char var28;
char var29;
char var30;
char var31;
char var32;
char var33;
char var34;
char var35;
char var36;
char var37;
char var38;
char var39;
char var40;
char var41;
char var42;


// Declaration des variables qui seront envoyées sur le serveur SQL
String RTCAnnee;
String RTCMois;
String RTCJour;
String RTCHeure;
String RTCMinute;
String Valeur_Int_Temperature;
String Valeur_Int_Humidite;
String Valeur_Ext_Temperature;
String Valeur_Ext_Humidite;
String Valeur_Servo_Porte_Principale;
String ValeurInterF;
String ValeurInterO;
String PorteBloquageO;
String PorteBloquageF;
String Valeur_Servo_Verrou;
String Valeur_Servo_Porte_Pondoir;
String Valeur_Relais_Chauffage;
String Valeur_Relais_Eclairage;
String ValeurInterInfos;
String ValeurInterModeManuel;
String ValeurInterTestMatos;
String ValeurInterFood;
String Valeur_LED_Blanche;
String Valeur_LED_Bleu;
String Valeur_Led_Verte;
String Valeur_Led_Orange_1;
String Valeur_Led_Orange_2;
String Valeur_Led_Rouge_1;
String Valeur_Led_Rouge_2;
String Valeur_Led_Rouge_3;
String ValeurLEDChauffage;


////////////////////////////////////////////////////////////////////////////////////////////
//    Definition des broches utilisees pour module ethernet et bibliotheques associees
////////////////////////////////////////////////////////////////////////////////////////////

#include <SPI.h>
#include <Ethernet.h>

// Configuration carte reseau ethershield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 1, 151 };
byte gateway[] = { 192, 168, 1, 254 };
byte subnet[] = { 255, 255, 255, 0 };

// Configuration serveur cible
correspond au ping du site web
IPAddress server(000, 000, 000, 000); // 000.000.000.000 correspond a l adresse ip du serveur sql

EthernetClient client;
```

```
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                          SETUP
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void setup() {

  // Ouverture du port info
  Serial.begin(9600);

  // Ouverture du port serie
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  mySerial.begin(4800);


  // initialisation de la carte RX
  Serial.println("Configuration du LAN");

  Serial.println("Connection reseau...");

  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("Configuration Ethernet NOK");
    Ethernet.begin(mac, ip);
  }
  else
  {
    Serial.println("Configuration Ethernet OK");
    Serial.println("");
  }

}



/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                          LOOP
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void loop()
{

  // recuperation des donnes du port serie
  RecupData();

  // Envoie de la requete au SRV SQL
  RequeteHTTP();

  delay(5000); // tres inportant, le delais doit etre toujours plus petit que celui de l emission

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                          Fonctions
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////



//////////////////////////////////////////////////////////////////////////////////////////
//                          Fonction -  Connexion HTTP
//////////////////////////////////////////////////////////////////////////////////////////

void connect()
{

  if (client.connect(server, 80))
  {
    Serial.println("Server --> Connecte");
    if (client.connected())
    {
```

```
Serial.println("Client --> Connecte");

Serial.println("Client --> Envoi de requete");

// Make a HTTP request:
//client.print( "GET /add.php?");
client.print( "GET /Poulailler/update.php?"); // c'est en fait le dossier qui contient le fichier de requete SQL

client.print("RTCAnnee=");
client.print(RTCAnnee);
client.print("&&");
client.print("RTCMois=");
client.print(RTCMois);
client.print("&&");
client.print("RTCJour=");
client.print(RTCJour);
client.print("&&");
client.print("RTCHeure=");
client.print(RTCHeure);
client.print("&&");
client.print("RTCMinute=");
client.print(RTCMinute);
client.print("&&");
client.print("Valeur_Int_Temperature=");
client.print(Valeur_Int_Temperature);
client.print("&&");
client.print("Valeur_Int_Humidite=");
client.print(Valeur_Int_Humidite);
client.print("&&");
client.print("Valeur_Ext_Temperature=");
client.print(Valeur_Ext_Temperature);
client.print("&&");
client.print("Valeur_Ext_Humidite=");
client.print(Valeur_Ext_Humidite);
client.print("&&");
client.print("Valeur_Servo_Porte_Principale=");
client.print(Valeur_Servo_Porte_Principale);
client.print("&&");
client.print("ValeurInterF=");
client.print(ValeurInterF);
client.print("&&");
client.print("ValeurInterO=");
client.print(ValeurInterO);
client.print("&&");
client.print("PorteBloquageO=");
client.print(PorteBloquageO);
client.print("&&");
client.print("PorteBloquageF=");
client.print(PorteBloquageF);
client.print("&&");
client.print("Valeur_Servo_Verrou=");
client.print(Valeur_Servo_Verrou);
client.print("&&");
client.print("Valeur_Servo_Porte_Pondoir=");
client.print(Valeur_Servo_Porte_Pondoir);
client.print("&&");
client.print("Valeur_Relais_Chauffage=");
client.print(Valeur_Relais_Chauffage);
client.print("&&");
client.print("Valeur_Relais_Eclairage=");
client.print(Valeur_Relais_Eclairage);
client.print("&&");
client.print("ValeurInterInfos=");
client.print(ValeurInterInfos);
client.print("&&");
client.print("ValeurInterModeManuel=");
client.print(ValeurInterModeManuel);
client.print("&&");
client.print("ValeurInterTestMatos=");
client.print(ValeurInterTestMatos);
client.print("&&");
client.print("ValeurInterFood=");
client.print(ValeurInterFood);
client.print("&&");
client.print("Valeur_LED_Blanche=");
client.print(Valeur_LED_Blanche);
```

```
    client.print("&&");
    client.print("Valeur_LED_Bleu=");
    client.print(Valeur_LED_Bleu);
    client.print("&&");
    client.print("Valeur_Led_Verte=");
    client.print(Valeur_Led_Verte);
    client.print("&&");
    client.print("Valeur_Led_Orange_1=");
    client.print(Valeur_Led_Orange_1);
    client.print("&&");
    client.print("Valeur_Led_Orange_2=");
    client.print(Valeur_Led_Orange_2);
    client.print("&&");
    client.print("Valeur_Led_Rouge_1=");
    client.print(Valeur_Led_Rouge_1);
    client.print("&&");
    client.print("Valeur_Led_Rouge_2=");
    client.print(Valeur_Led_Rouge_2);
    client.print("&&");
    client.print("Valeur_Led_Rouge_3=");
    client.print(Valeur_Led_Rouge_3);
    client.print("&&");
    client.print("ValeurLEDChauffage=");
    client.print(ValeurLEDChauffage);

    client.println( " HTTP/1.1");

    //ici on indique l'hote soit ladresse ip fixe soit un nom de domaine
    client.println( "Host: NomDeDomaine.free.fr" ); //free

    client.println( "Connection: close" );
    client.println();
    client.println();
    client.stop();
    Serial.println("Client --> Requete envoyee");
  }
  else
  {
    Serial.println("Client --> ERREUR NON CONNECTE");
  }
 }

 else
 {
  Serial.println("Server --> ERREUR NON CONNECTE");
 }
}



////////////////////////////////////////////////////////////////////////////////////////////////
//                          Fonction -  Requete HTTP
////////////////////////////////////////////////////////////////////////////////////////////////

void RequeteHTTP()
{
 // gestion de l'envoie de la requete HTTP
 if (client.available())
 {
  Serial.println("Client Disponible -->  OK");
  char c = client.read();
  Serial.print(c);
 }
 else
 {
  Serial.println("Client Disponible --> ERREUR NOK");
 }

 // si client est deconecté on reconnecte et envoie la requete
 if (!client.connected())
 {
  Serial.println("Client Connecte --> NOK");

  Serial.println("Client --> Stop");
  client.stop();
```

```
  Serial.println("Client --> Reconnection");
  connect();
  // on reste 1 sec avant de couper la requete.
  delay(1000);

 }
 else
 {
  Serial.println("Client Connecte --> OK");
 }
}



////////////////////////////////////////////////////////////////////////////////////////////////////
//                    Fonction -  Recuperation des donnes via la liaison serie
////////////////////////////////////////////////////////////////////////////////////////////////////


void RecupData()
{
 if (mySerial.available()) {


  var1 = mySerial.read();
  delay(50);
  var2 = mySerial.read();
  delay(50);
  var3 = mySerial.read();
  delay(50);
  var4 = mySerial.read();
  delay(50);
  var5 = mySerial.read();
  delay(50);
  var6 = mySerial.read();
  delay(50);
  var7 = mySerial.read();
  delay(50);
  var8 = mySerial.read();
  delay(50);
  var9 = mySerial.read();
  delay(50);
  var10 = mySerial.read();
  delay(50);
  var11 = mySerial.read();
  delay(50);
  var12 = mySerial.read();
  delay(50);
  var13 = mySerial.read();
  delay(50);
  var14 = mySerial.read();
  delay(50);
  var15 = mySerial.read();
  delay(50);
  var16 = mySerial.read();
  delay(50);
  var17 = mySerial.read();
  delay(50);
  var18 = mySerial.read();
  delay(50);
  var19 = mySerial.read();
  delay(50);
  var20 = mySerial.read();
  delay(50);
  var21 = mySerial.read();
  delay(50);
  var22 = mySerial.read();
  delay(50);
  var23 = mySerial.read();
  delay(50);
  var24 = mySerial.read();
  delay(50);
  var25 = mySerial.read();
  delay(50);
  var26 = mySerial.read();
```

```
delay(50);
var27 = mySerial.read();
delay(50);
var28 = mySerial.read();
delay(50);
var29 = mySerial.read();
delay(50);
var30 = mySerial.read();
delay(50);
var31 = mySerial.read();
delay(50);
var32 = mySerial.read();
delay(50);
var32 = mySerial.read();
delay(50);
var33 = mySerial.read();
delay(50);
var34 = mySerial.read();
delay(50);
var35 = mySerial.read();
delay(50);
var36 = mySerial.read();
delay(50);
var37 = mySerial.read();
delay(50);
var38 = mySerial.read();
delay(50);
var39 = mySerial.read();
delay(50);
var40 = mySerial.read();
delay(50);


/*

    Serial.print("var1 = ");
    Serial.println(var1);
    Serial.print("var2 = ");
    Serial.println(var2);
    Serial.print("var3 = ");
    Serial.println(var3);
    Serial.print("var4 = ");
    Serial.println(var4);
    Serial.print("var5 = ");
    Serial.println(var5);
    Serial.print("var6 = ");
    Serial.println(var6);
    Serial.print("var7 = ");
    Serial.println(var7);
    Serial.print("var8 = ");
    Serial.println(var8);
    Serial.print("var9 = ");
    Serial.println(var9);
    Serial.print("var10 = ");
    Serial.println(var10);
    Serial.print("var11 = ");
    Serial.println(var11);
    Serial.print("var12 = ");
    Serial.println(var12);
    Serial.print("var13 = ");
    Serial.println(var13);
    Serial.print("var14 = ");
    Serial.println(var14);
    Serial.print("var15 = ");
    Serial.println(var15);
    Serial.print("var16 = ");
    Serial.println(var16);
    Serial.print("var17 = ");
    Serial.println(var17);
    Serial.print("var18 = ");
    Serial.println(var18);
    Serial.print("var19 = ");
    Serial.println(var19);
    Serial.print("var20 = ");
    Serial.println(var20);
    Serial.print("var21 = ");
```

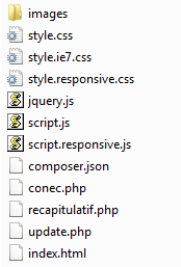
```
    Serial.println(var21);
    Serial.print("var22 = ");
    Serial.println(var22);
    Serial.print("var23 = ");
    Serial.println(var23);
    Serial.print("var24 = ");
    Serial.println(var24);
    Serial.print("var25 = ");
    Serial.println(var25);
    Serial.print("var26 = ");
    Serial.println(var26);
    Serial.print("var27 = ");
    Serial.println(var27);
    Serial.print("var28 = ");
    Serial.println(var28);
    Serial.print("var29 = ");
    Serial.println(var29);
    Serial.print("var30 = ");
    Serial.println(var30);
    Serial.print("var31 = ");
    Serial.println(var31);
    Serial.print("var32 = ");
    Serial.println(var32);
    Serial.print("var33 = ");
    Serial.println(var33);
    Serial.print("var34 = ");
    Serial.println(var34);
    Serial.print("var35 = ");
    Serial.println(var35);
    Serial.print("var36 = ");
    Serial.println(var36);
    Serial.print("var37 = ");
    Serial.println(var37);
    Serial.print("var38 = ");
    Serial.println(var38);
    Serial.print("var39 = ");
    Serial.println(var39);
    Serial.print("var40 = ");
    Serial.println(var40);
    Serial.print("var41 = ");
    Serial.println(var41);
    Serial.print("var42 = ");
    Serial.println(var42);

*/

// concatenation des caracteres pour recuperer les valeurs d origine

RTCAnnee = String(var1) + String(var2);
RTCMois = String(var3) + String(var4);
RTCJour = String(var5) + String(var6);
RTCHeure = String(var7) + String(var8);
RTCMinute = String(var9) + String(var10);
Valeur_Int_Temperature = String(var11) + String(var12);
Valeur_Int_Humidite =  String(var13) + String(var14);
Valeur_Ext_Temperature = String(var15) + String(var16);
Valeur_Ext_Humidite =  String(var17) + String(var18);
Valeur_Servo_Porte_Principale =  String(var19);
ValeurInterF =  String(var20);
ValeurInterO =  String(var21);
PorteBloquageO =  String(var22);
PorteBloquageF =  String(var23);
Valeur_Servo_Verrou =  String(var24);
Valeur_Servo_Porte_Pondoir =  String(var25);
Valeur_Relais_Chauffage =  String(var26);
Valeur_Relais_Eclairage =  String(var27);
ValeurInterInfos =  String(var28);
ValeurInterModeManuel =  String(var29);
ValeurInterTestMatos =  String(var30);
ValeurInterFood =  String(var31);
Valeur_LED_Blanche =  String(var32);
Valeur_LED_Bleu =  String(var33);
Valeur_Led_Verte =  String(var34);
Valeur_Led_Orange_1 =  String(var35);
Valeur_Led_Orange_2 =  String(var36);
Valeur_Led_Rouge_1 =  String(var37);
```

```
  Valeur_Led_Rouge_2 =  String(var38);
  Valeur_Led_Rouge_3 =  String(var39);
  ValeurLEDChauffage =  String(var40);


  // affichage des valeurs recupéréés

  Serial.println("RTCAnnee = " + RTCAnnee);
  Serial.println("RTCMois = " + RTCMois);
  Serial.println("RTCJour = " + RTCJour);
  Serial.println("RTCHeure = " + RTCHeure);
  Serial.println("RTCMinute = " + RTCMinute);
  Serial.println("Valeur_Int_Temperature = " + Valeur_Int_Temperature);
  Serial.println("Valeur_Int_Humidite = " + Valeur_Int_Humidite);
  Serial.println("Valeur_Ext_Temperature = " + Valeur_Ext_Temperature);
  Serial.println("Valeur_Ext_Humidite = " + Valeur_Ext_Humidite);
  Serial.println("Valeur_Servo_Porte_Principale = " + Valeur_Servo_Porte_Principale);
  Serial.println("ValeurInterF = " + ValeurInterF);
  Serial.println("ValeurInterO = " + ValeurInterO);
  Serial.println("PorteBloquageO = " + PorteBloquageO);
  Serial.println("PorteBloquageF = " + PorteBloquageF);
  Serial.println("Valeur_Servo_Verrou = " + Valeur_Servo_Verrou);
  Serial.println("Valeur_Servo_Porte_Pondoir = " + Valeur_Servo_Porte_Pondoir);
  Serial.println("Valeur_Relais_Chauffage = " + Valeur_Relais_Chauffage);
  Serial.println("Valeur_Relais_Eclairage = " + Valeur_Relais_Eclairage);
  Serial.println("ValeurInterInfos = " + ValeurInterInfos);
  Serial.println("ValeurInterModeManuel = " + ValeurInterModeManuel);
  Serial.println("ValeurInterTestMatos = " + ValeurInterTestMatos);
  Serial.println("ValeurInterFood = " + ValeurInterFood);
  Serial.println("Valeur_LED_Blanche = " + Valeur_LED_Blanche);
  Serial.println("Valeur_LED_Bleu = " + Valeur_LED_Bleu);
  Serial.println("Valeur_Led_Verte = " + Valeur_Led_Verte);
  Serial.println("Valeur_Led_Orange_1 = " + Valeur_Led_Orange_1);
  Serial.println("Valeur_Led_Orange_2 = " + Valeur_Led_Orange_2);
  Serial.println("Valeur_Led_Rouge_1 = " + Valeur_Led_Rouge_1);
  Serial.println("Valeur_Led_Rouge_2 = " + Valeur_Led_Rouge_2);
  Serial.println("Valeur_Led_Rouge_3 = " + Valeur_Led_Rouge_3);
  Serial.println("ValeurLEDChauffage = " + ValeurLEDChauffage);
  Serial.println("");
 }
}
```

Construction du serveur WEB.

Clui ci doit bien evidemment detenir APACHE pour le HTML, PHP mais aussi MySQL, personellement j'ai aussi ajouté phpmyadmin pour une utilisation aisée de la base de donnée.

Dans le dossier Carte 3 Raspberry Pi se trouve ceci :



Images = contient les images du site.
Style.css = ça parle de lui-même et idem pour ie7 et responsive.
****.js = Tous les fichiers.js sont à laisser tel quel.
Index.html = fichier principal de l'interface web c'est ici que l'on va configurer l'adresse de votre serveur afin que l'arduino pointe bien dessus.

<u>Son contenu :</u>

```
<!DOCTYPE html>
<html dir="ltr" lang="en-US">

<head><!-- Created by Artisteer v4.1.0.59782 -->
		<meta charset="utf-8">
		<title></title>
		<meta name="viewport" content="initial-scale = 1.0, maximum-scale = 1.0, user-scalable = no, width = device-width">

		<!--[if lt IE 9]><script src="https://html5shiv.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
		<link rel="stylesheet" href="style.css" media="screen">
		<!--[if lte IE 7]><link rel="stylesheet" href="style.ie7.css" media="screen" /><![endif]-->
		<link rel="stylesheet" href="style.responsive.css" media="all">
		<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Molengo|Lekton&amp;subset=latin">

		<script src="jquery.js"></script>
		<script src="script.js"></script>
		<script src="script.responsive.js"></script>


		<style>.art-content .art-postcontent-0 .layout-item-0 { padding-right: 10px;padding-left: 50px;  }
		.art-content .art-postcontent-0 .layout-item-1 { padding-right: 10px;padding-left: 10px;  }
		.ie7 .art-post .art-layout-cell {border:none !important; padding:0 !important; }
		.ie6 .art-post .art-layout-cell {border:none !important; padding:0 !important; }

		</style>
</head>

<body>

		<div id="art-main">

		<header class="art-header">

				<div class="art-shapes">
					<div class="art-object1008736887" data-left="0%"></div>
					<div class="art-object2115474892" data-left="100%"></div>
				</div>

					<h2 class="art-slogan" data-left="41.08%">Toutes les infos à distance et en temps réel.</h2>

					<div id="art-flash-area">
						<div id="art-flash-container">
					</div>
		</div>

		</header>
				<nav class="art-nav">
				</nav>
		<div class="art-sheet clearfix">
				<div class="art-layout-wrapper">
					<div class="art-content-layout">
						<div class="art-content-layout-row">
							<div class="art-layout-cell art-content"><article class="art-post art-article">
							<h2 class="art-postheader"><span class="art-postheadericon"></span></h2>

								<div class="art-postcontent art-postcontent-0 clearfix"><div class="art-content-layout">
	<div class="art-content-layout-row">
		<div class="art-layout-cell layout-item-0" style="width: 60%" >
			<h3 style="text-align: center;">Quoi de neuf docteur ?<br><span
style="color: rgb(10, 12, 9); font-size: 9px; font-style: italic; font-weight: normal; text-transform: none;">(Informations relatives à l'arduino.)</span></h3><p><span style="font-style: italic;">


		<iframe src="http://VOTRE ADRESSE IP DE
SERVEUR:PORT/recapitulatif.php" height="360" width="100%" frameborder="0" allowtransparency="true"></iframe></span></p>
		</div><div class="art-layout-cell layout-item-1" style="width: 40%" >
		<p> <img width="264" height="392" alt="" class="art-lightbox"
src="images/Poulette.png" style="float: left; margin-left: 0px;"><br></p>
		</div>
	</div>
</div>
</div>


</article></div>
</div>
</div>
```

```
                    </div>
          </div>
                    <footer class="art-footer">
                              <div class="art-footer-inner">
                              <p>Poulailler d'Aurélien.</p>
                                        <p class="art-page-footer">

                                        </p>
                              </div>
                    </footer>

          </div>


</body></html>
```

Le fichier **recapitulatif.php** est le fichier qui va aller interroger la base de donnée et les afficher dans la page html.
C'est lui qui aura l'intelligence d'interpreter les infos de la base de donnée et de les retranscrire en texte.

Dans cet exemple, la base créée sera **Poulailler** et la table **TablePoulailler ,** et pour ma part **login** et **m.d.p** égal à **poulailler**

```
          <html>



          <body>

          <?php

          // connection au serveur

          $link = mysql_connect('localhost', 'Poulailler', 'Poulailler'); //serveur, login, mdp
          if (!$link) {
            die('Impossible de se connecter : ' . mysql_error());
          }
            //echo 'Connexion Serveur reussie'; //sert de test pour afficher un message sur la page afin de voir si on se connecte bien


          // connection a la base de donnee

            $db_selected = mysql_select_db('Poulailler', $link);
                    if (!$db_selected) {
                    die ('Impossible de selectionner la base de donnees : ' . mysql_error());
          }
          //echo 'Connexion Base reussie';   //sert de test pour afficher un message sur la page afin de voir si on se connecte bien


          // connection a la table
            $result=mysql_query("select * from TablePoulailler WHERE `id` = 1",$link);


          $data = mysql_fetch_array($result);


          ?>


          Nous sommes le
          <b>
          <?php echo $data['RTCJour']; ?>
          /
          <?php echo $data['RTCMois']; ?>
          /
          <?php echo $data['RTCAnnee']; ?>
          </b>

           et il est
          <b>
           <?php echo $data['RTCHeure'];?>
          h
          <?php echo $data['RTCMinute']; ?>
          </b>
          .
          <br>



          La temperature interieure est de
          <font color="#ff9900">
          <b>
          <?php echo $data['Valeur_Int_Temperature']; ?>
          &deg;
          </b>
          </font>avec une humidite de
          <font color="#0033cc">
          <?php echo  $data['Valeur_Int_Humidite']; ?>
          %
          </font>
          .
          <br>
          Par contre, dehors il fait
          <font color="#ff9900">
          <?php echo $data['Valeur_Ext_Temperature']; ?>
          &deg;
          </font>
          avec une humidite de
          <font color="#0033cc">
          <?php echo $data['Valeur_Ext_Humidite']; ?>
          %
          </font>
          .



          <br>

          La porte est bien
          <font color="#33cc33">
          <?php
```

```
                $ValeurInterO = $data['ValeurInterO'];

                if($ValeurInterO == 1)
                {
                                echo 'ouverte';
                }
                else
                {
                                echo '';
                }
?>
</font>

<font color="#0000ff">
<?php
                $ValeurInterF = $data['ValeurInterF'];

                if($ValeurInterF == 1)
                {
                                echo 'fermee';
                }
                else
                {
                                echo '';
                }
?>
</font>




avec le verrou
<font color="#0000ff">
<?php
                $Valeur_Servo_Verrou = $data['Valeur_Servo_Verrou'];

                if($Valeur_Servo_Verrou == 1)
                {
                                echo 'ferme.';
                }
                else
                {
                                echo '';
                }
?>
</font>

<font color="#33cc33">

<?php
                $Valeur_Servo_Verrou = $data['Valeur_Servo_Verrou'];

                if($Valeur_Servo_Verrou == 1)
                {
                                echo '';
                }
                else
                {
                                echo 'ouvert.';
                }
?>
</font>

<br>


Quant aux portes du pondoir, elles sont

<font color="#0000ff">
<?php
                $Valeur_Servo_Porte_Pondoir = $data['Valeur_Servo_Porte_Pondoir'];

                if($Valeur_Servo_Porte_Pondoir == 1)
                {
                                echo 'fermees.';
                }
                else
                {
                                echo '';
                }
?>
</font>

<font color="#33cc33">
<?php
                $Valeur_Servo_Porte_Pondoir = $data['Valeur_Servo_Porte_Pondoir'];

                if($Valeur_Servo_Porte_Pondoir == 1)
                {
                                echo '';
                }
                else
                {
                                echo 'ouvertes.';
                }
?>
</font>

<center>
<?php
                $Valeur_LED_Blanche = $data['Valeur_LED_Blanche'];

                if($Valeur_LED_Blanche == 1)
                {
                                echo '<img src="./images/Jour.jpg" width="40" height="40"/>';
                }
                else
                {
                                echo '';

                }
?>
```

```
<?php
		$Valeur_LED_Bleu = $data['Valeur_LED_Bleu'];

		if($Valeur_LED_Bleu == 1)
		{
				echo '<img src="./images/Nuit.jpg" width="40" height="40"/>';
		}
		else
		{
				echo '';
		}
?>


<?php
		$Valeur_Relais_Eclairage = $data['Valeur_Relais_Eclairage'];

		if($Valeur_Relais_Eclairage == 1)
		{
				echo '<img src="./images/Lumiere.jpg" width="40" height="40"/>';
		}
		else
		{
				echo '';
		}
?>

<?php
		$Valeur_Relais_Chauffage = $data['Valeur_Relais_Chauffage'];

		if($Valeur_Relais_Chauffage == 1)
		{
				echo '<img src="./images/Chauffage.jpg" width="40" height="40"/>';
		}
		else
		{
				echo '';
		}
?>
</center>


Le chauffage est
<font color="#ff4000">
<b>
<?php
		$Valeur_Relais_Chauffage = $data['Valeur_Relais_Chauffage'];

		if($Valeur_Relais_Chauffage == 1)
		{
				echo 'allume,';
		}
		else
		{
				echo '';
		}
?>
</font>
</b>

<?php
		$Valeur_Relais_Chauffage = $data['Valeur_Relais_Chauffage'];

		if($Valeur_Relais_Chauffage == 1)
		{
				echo '';
		}
		else
		{
				echo 'eteint,';
		}
?>


<?php
		$ValeurLEDChauffage = $data['ValeurLEDChauffage'];

		if($ValeurLEDChauffage == 1)
		{
				echo 'et sa led on.';
		}
		else
		{
				echo 'et sa led off.';
		}
?>

<br>

La lumiere exterieure est
<font color="#ff9900">
<b>
<?php
		$Valeur_Relais_Eclairage = $data['Valeur_Relais_Eclairage'];

		if($Valeur_Relais_Eclairage == 1)
		{
				echo 'allumee.';
		}
		else
		{
				echo '';
		}
?>
</font>
</b>
<?php
		$Valeur_Relais_Eclairage = $data['Valeur_Relais_Eclairage'];

		if($Valeur_Relais_Eclairage == 1)
		{
				echo '';
		}
		else
```

```
		{
				echo 'eteinte.';
		}
?>


<br>



<?php
		$ValeurInterFood = $data['ValeurInterFood'];

		if($ValeurInterFood == 1)
		{
				 echo 'Au fait, il y a toujours de la nourriture dans la mangeoire !';
		}
		else
		{
				echo '';
		}
?>


<center>
<font color="#ff0000">
<b>
<?php
		$ValeurInterFood = $data['ValeurInterFood'];

		if($ValeurInterFood == 1)
		{
				 echo '';
		}
		else
		{
				echo 'IL N Y A PLUS DE NOURRITURE !!!';
		}
?>




<?php
		$Valeur_Led_Rouge_1 = $data['Valeur_Led_Rouge_1'];

		if($Valeur_Led_Rouge_1 == 1)
		{
				 echo '!!! IL Y A UN PROBLEME DE BUTEE !!!';
		}
		else
		{
				echo '';
		}
?>

<?php
		$PorteBloquageO = $data['PorteBloquageO'];

		if($PorteBloquageO == 1)
		{
				 echo 'LA PORTE EST BLOQUEE EN OUVERTURE';
		}
		else
		{
				echo '';
		}
?>

<?php
		$PorteBloquageF = $data['PorteBloquageF'];

		if($PorteBloquageF == 1)
		{
				 echo 'LA PORTE EST BLOQUEE EN FERMETURE';
		}
		else
		{
				echo '';
		}
?>

<?php
		$Valeur_Led_Rouge_2 = $data['v'];

		if($Valeur_Led_Rouge_2 == 1)
		{
				 echo '!!! LA PORTE A UN DYSFONCTIONNEMENT : TIMER OUT !!!';
		}
		else
		{
				echo '';
		}
?>



<?php
		$ValeurInterInfos = $data['ValeurInterInfos'];

		if($ValeurInterInfos == 0)
		{
				 echo '';
		}
		else
		{
				echo 'L interrupteur Informations est actif !';
		}
?>

<?php
		$ValeurInterModeManuel = $data['ValeurInterModeManuel'];
```

```
                    if($ValeurInterModeManuel == 0)
                    {
                                echo '';
                    }
                    else
                    {
                                echo 'L interrupteur Mode Manuel est actif !';
                    }
?>

<?php
                    $ValeurInterTestMatos = $data['ValeurInterTestMatos'];

                    if($ValeurInterTestMatos == 0)
                    {
                                echo '';
                    }
                    else
                    {
                                echo 'L interrupteur Test Matos est actif !';
                    }
?>

<?php
                    $Valeur_Servo_Porte_Principale = $data['Valeur_Servo_Porte_Principale'];

                    if($Valeur_Servo_Porte_Principale == 0)
                    {
                                echo '';
                    }
                    else
                    {
                                echo 'La porte est en mouvement.';
                    }
?>
</center>
</b>
</font>
<br>

</body>
</html>
```

Le fichier **conec.php** sera le fichier servant a la connexion au serveur

```
<?php
function Conection(){

$link = mysql_connect('localhost', 'root', 'toor'); //serveur, login, mdp
if (!$link) {
  die('Impossible de se connecter : ' . mysql_error());
}
  echo 'Connexion Serveur reussie'; //sert de test pour afficher un message sur la page afin de voir si on se connecte bien

  $db_selected = mysql_select_db('Poulailler', $link);
                    if (!$db_selected) {
                    die ('Impossible de selectionner la base de donnees : ' . mysql_error());
}
  echo 'Connexion Base reussie';   //sert de test pour afficher un message sur la page afin de voir si on se connecte bien
}

?>
```

Et enfin **update.php** sera la page ou l'arduino (carte2) va envoyer ses infos.
C'est via cette page que les données de l'arduino vous être pushée sur le servuerSQL (c'est en fait juste une requete sql update):
Lors de la création de la base, faire bien attention a ce que vous saisissez ainsi qu'a la casse.

```
<?php

  // connection au serveur

$link = mysql_connect('localhost', 'Poulailler', 'Poulailler'); //base, login, mdp
if (!$link) {
  die('Impossible de se connecter : ' . mysql_error());
}
  //echo 'Connexion Serveur reussie'; //sert de test pour afficher un message sur la page afin de voir si on se connecte bien


// connection a la base de donnee

  $db_selected = mysql_select_db('Poulailler', $link);
                    if (!$db_selected) {
                    die ('Impossible de selectionner la base de donnees : ' . mysql_error());
}
  //echo 'Connexion Base reussie';   //sert de test pour afficher un message sur la page afin de voir si on se connecte bien

  $Sql="UPDATE `Poulailler`.`TablePoulailler` SET `RTCAnnee` = '".$_GET["RTCAnnee"]."',`RTCMois` = '".$_GET["RTCMois"]."',`RTCJour` = '".$_GET["RTCJour"]."',`RTCHeure` = '".$_GET["RTCHeure"]."',`RTCMinute` =
'".$_GET["RTCMinute"]."',`Valeur_Int_Temperature` = '".$_GET["Valeur_Int_Temperature"]."',`Valeur_Int_Humidite` = '".$_GET["Valeur_Int_Humidite"]."',`Valeur_Ext_Temperature` =
'".$_GET["Valeur_Ext_Temperature"]."',`Valeur_Ext_Humidite` = '".$_GET["Valeur_Ext_Humidite"]."',`Valeur_Servo_Porte_Principale` = '".$_GET["Valeur_Servo_Porte_Principale"]."',`ValeurInterF` =
'".$_GET["ValeurInterF"]."',`ValeurInterO` = '".$_GET["ValeurInterO"]."',`PorteBloquageO` = '".$_GET["PorteBloquageO"]."',`PorteBloquageF` = '".$_GET["PorteBloquageF"]."',`Valeur_Servo_Verrou` =
'".$_GET["Valeur_Servo_Verrou"]."',`Valeur_Servo_Porte_Pondoir` = '".$_GET["Valeur_Servo_Porte_Pondoir"]."',`Valeur_Relais_Chauffage` = '".$_GET["Valeur_Relais_Chauffage"]."',`Valeur_Relais_Eclairage` =
'".$_GET["Valeur_Relais_Eclairage"]."',`ValeurInterInfos` = '".$_GET["ValeurInterInfos"]."',`ValeurInterModeManuel` = '".$_GET["ValeurInterModeManuel"]."',`ValeurInterTestMatos` =
'".$_GET["ValeurInterTestMatos"]."',`ValeurInterFood` = '".$_GET["ValeurInterFood"]."',`Valeur_LED_Blanche` = '".$_GET["Valeur_LED_Blanche"]."',`Valeur_LED_Bleu` = '".$_GET["Valeur_LED_Bleu"]."',`Valeur_Led_Verte` =
'".$_GET["Valeur_Led_Verte"]."',`Valeur_Led_Orange_1` = '".$_GET["Valeur_Led_Orange_1"]."',`Valeur_Led_Orange_2` = '".$_GET["Valeur_Led_Orange_2"]."',`Valeur_Led_Rouge_1` =
'".$_GET["Valeur_Led_Rouge_1"]."',`Valeur_Led_Rouge_2` = '".$_GET["Valeur_Led_Rouge_2"]."',`Valeur_Led_Rouge_3` = '".$_GET["Valeur_Led_Rouge_3"]."',`ValeurLEDChauffage` = '".$_GET["ValeurLEDChauffage"]."'
WHERE `TablePoulailler`.`id` = 1";

 mysql_query($Sql,$link);
  header("Location: insertareg.php");
?>
```