Pilotage d'une installation solaire avec microcontrôleur ARDUINO

par BIOULEZ Ph.

Après avoir piloté l'installation pendant 3 ans avec une électronique câblée, je me suis lancé dans la programmation d'un microcontrôleur ARDUINO.

Plan de l'installation

Plan de l'installation avec l'emplacement des capteurs (voir description de l'installation sur le site http://www.apper-solaire.org/Pages/Experiences/Bioulez Philippe 31/Installation de 9 m2 pour ECS et chauffage/index.html).

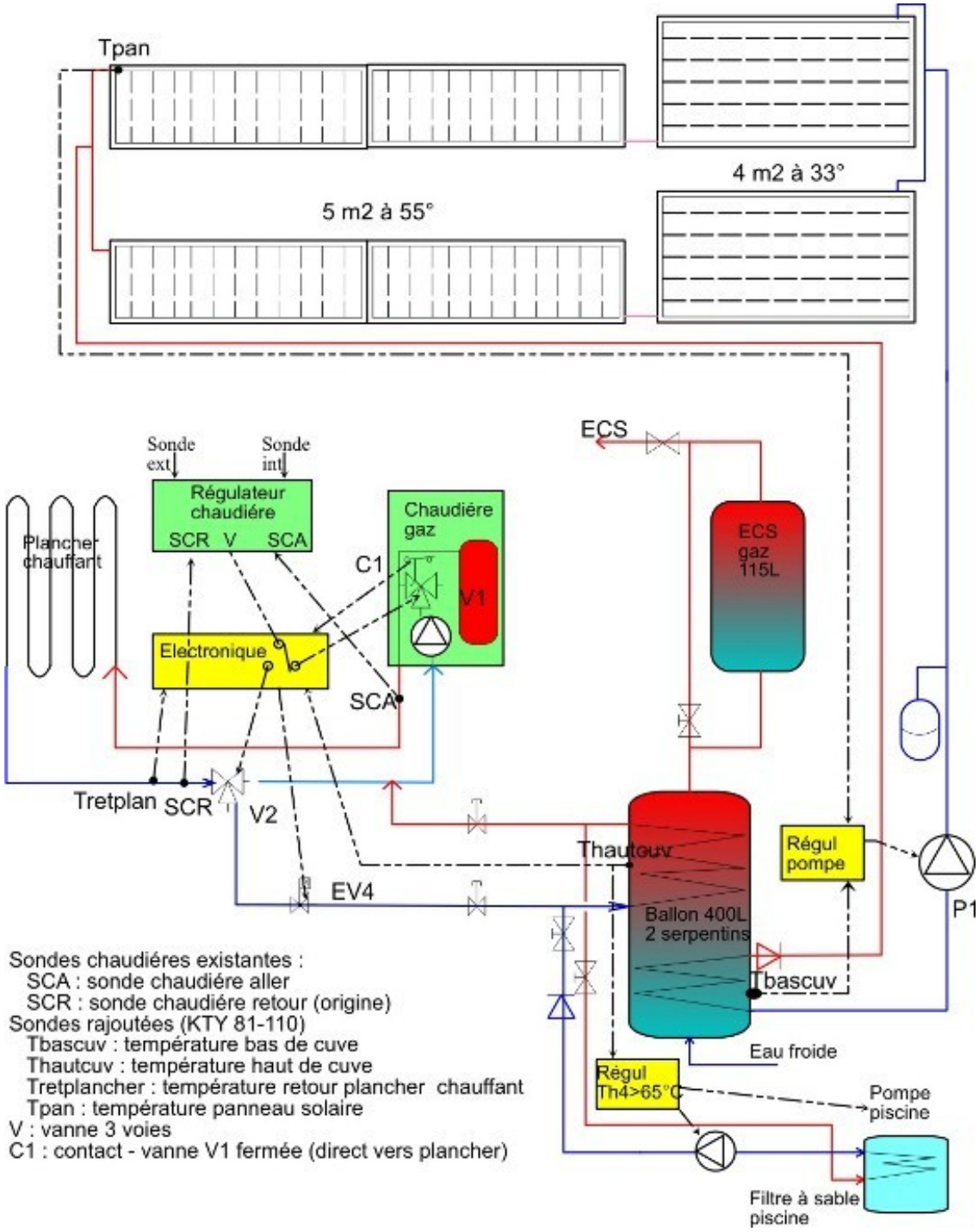
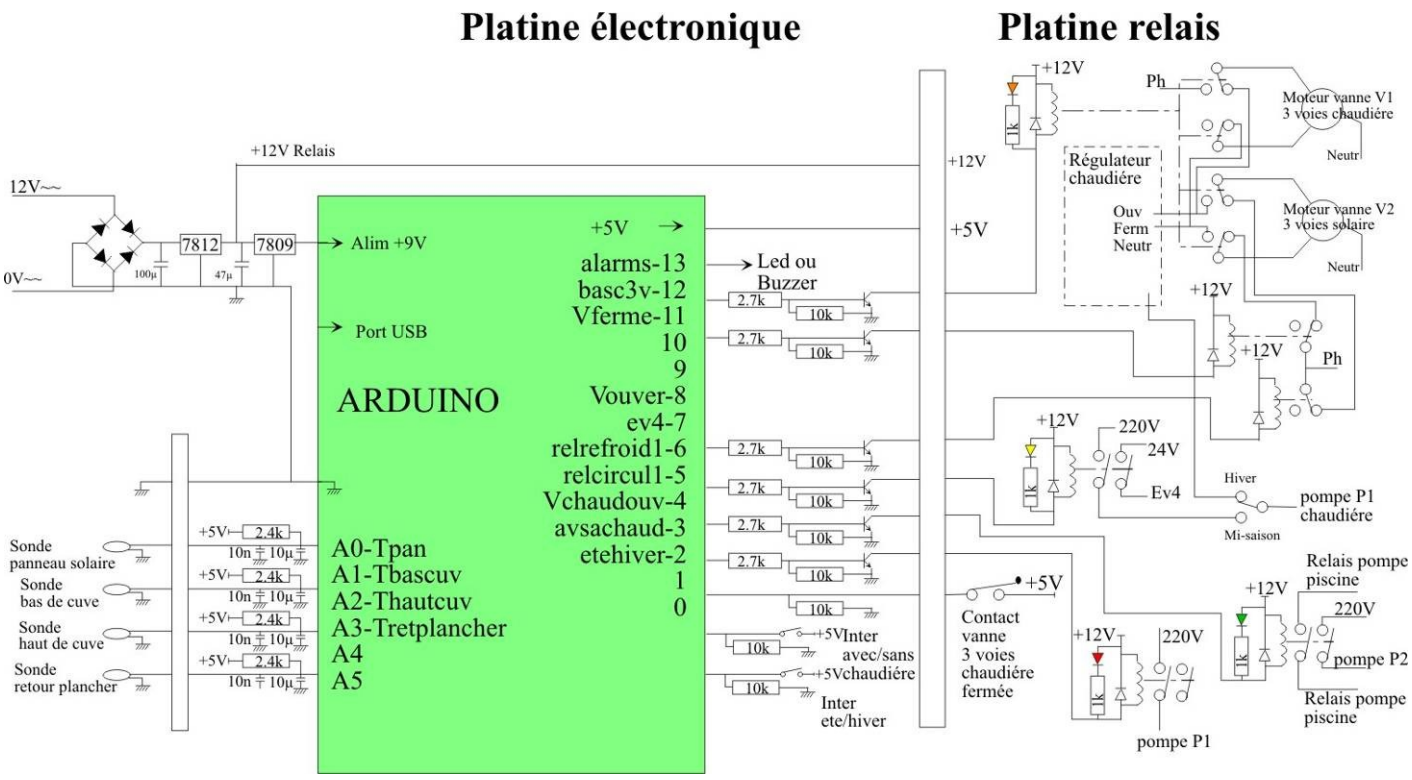


Schéma électrique

Voici le schéma électrique à partir d'une carte USB Duemilanove ARDUINO, assez facile à programmer grâce aux nombreux exemples fournis et aux tutoriaux sur internet (tous gratuits).

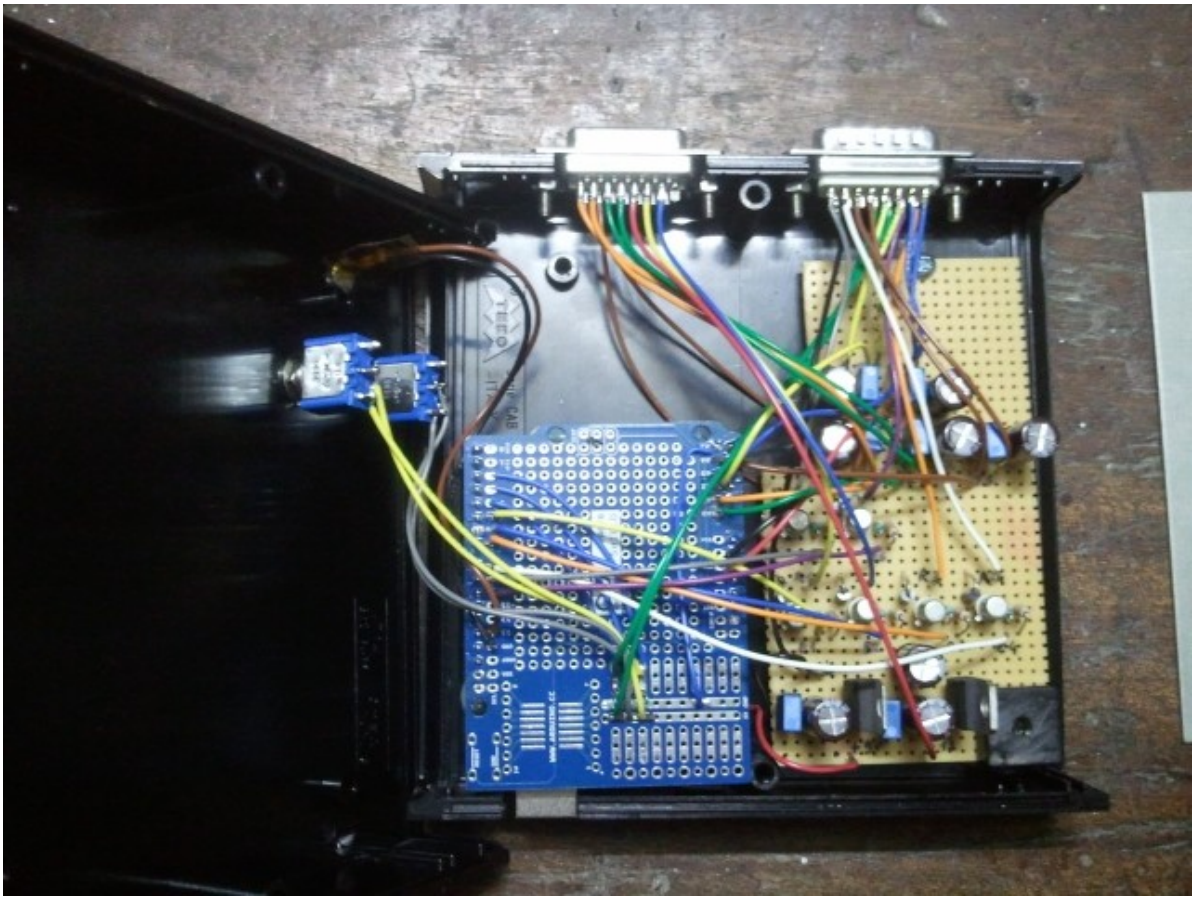
Les sondes de températures sont des KTY81-110 mise en pont avec une résistance de 2,4 kOhms, ce qui donne une précision de 0,6°C/bit après la conversion analogique/numérique.



Avantages :

- simplicité de la partie électronique,
- faibles coûts (environ 50€),
- facilité de reprogrammation (évolution du programme, modification des seuils, visualisation des capteurs avec un PC, ...)

Photo du boitier



Programmation du microcontrôleur

Voici un exemple de programmation qui fonctionne.

```
/*
  logiciel regulation solaire
  cree le 25/10/2012

 */

// initialisation des capteurs analogiques:
const int Tpan = A0; // temperature panneau solaire en pin 0
const int Tbascuv = A1; // temperature bas de cuve en pin 1
const int Thautcuv = A2; // temperature haut de cuve en pin 2
const int Tretplancher = A3; // temperature retour plancher chauffant en pin 3
const int Tallplancher = A4; // temperature aller plancher chauffant en pin 4 pour info
// these constants won't change, init capteurs entrees logiques:
// pin 0 et 1 sont en reserve
const int etehiver = 2;      // inter ete (ouvert) hiver pin 2, ete LOW hiver HIGH
const int avsachaud = 3;      // inter avec ou sans chaudiere pin 3, sans LOW avec HIGH
const int Vchaudouv = 4;      // inter actionné par vanne chaudiere pin 4, fluide ne passant pas ds chaudiere HIGH

// initialisation des sorties relais:
const int relcircul1 = 5;      // circulateur P1 chauffage cuve relais R1 pin 5
const int relrefroid1 = 6;      // circulateur P2 refroidissement vers piscine relais R2 pin 6
```

```
const int ev4 = 7;      // commande EV4 relais R3 pin 7
const int Vouver = 8;      // commande ouverture vanne 3 voies V2 pin 8
const int Vferme = 11;      // commande fermeture vanne 3 voies V2  pin 9
const int basc3v = 12;      // commande basculement vanne 3 voies V1 et sonde chaudiere SCR relais R4 R5 pin 10
const int alarms = 13;      // commande du buzzer pour les alarmes
// pin 10 est en reserve

// initialisation des variables
int cuvechaude = LOW;  // variable si cuve chaude pour chauffage
int inhibcde3v = HIGH;  // inhibe la commande de la vanne 3 voies
int etehiverState = LOW;         // variable pour lire l etat de l inter ete-hiver ete=low
int avsachaudState = LOW;         // variable pour lire l etat de l inter avec ou sans chaudiere
int VchaudouvState = LOW;         // variable pour lire l etat de l inter de la vanne V1 chaudiere
// courbe de tendance KTY81-110 avec resistance de 2400 Ohms = deg=bit*cal-157.21
float cal = 0.604;                              // valeur de calibration - degre par bit
// initialisation des variables pour commande vanne 3 voies
long previousMillis = 0;        // will store last time Vouver was updated
int interval = 5000;           // duree de fonctionnement vanne V2 (milliseconds), periode 20 sec
int i = 0;
int j = 0;

// initialisation des entrees-sorties
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // declaration entrees sorties:
  pinMode(etehiver, INPUT);
  pinMode(avsachaud, INPUT);
  pinMode(Vchaudouv, INPUT);
  pinMode(relcircul1, OUTPUT);
  pinMode(relrefroid1, OUTPUT);
  pinMode(ev4, OUTPUT);
  pinMode(Vouver, OUTPUT);
  pinMode(Vferme, OUTPUT);
  pinMode(basc3v, OUTPUT);
  pinMode(alarms, OUTPUT);
  digitalWrite(alarms, LOW);
}

// boucle principale faisant appel aux differents sous-programmes :
void loop() {
  // regulation chauffage cuve par panneaux solaires
  regulpan();
  // regulation refroidissement cuve - limitation a 63deg
  limitcuv();
  // alarmes de securite
  alarm();
  // verification position ete hiver:
  etehiverState = digitalRead(etehiver);
  Serial.print("\t etehiverState: ");
  Serial.println(etehiverState);
  // ete
  if (etehiverState == LOW) {
```

```
        digitalWrite(ev4, LOW);
        digitalWrite(Vouver, LOW);
        digitalWrite(Vferme, LOW);
        digitalWrite(basc3v, LOW);
  }
  // hiver
  else {
    // programme du chauffage du plancher chauffant:
        // commande electrovanne EV4 - ouverture circuit:
        cdeEV4();
        // programme de commande vanne 3 voies V2 solaire:
        cde3v();
        // commande de basculement vannes 3 voies (V1 et V2):
        bascul3v();
  }
  delay(5000);   // delai pour la thermique, doit etre inferieur a la variable interval
}

// programme de régulation du chauffage de la cuve par les panneaux solaires
void regulpan() {
  // lecture temperature panneau pin A0:
  int Tpan1 = analogRead(Tpan);
  float Tpand = Tpan1 * cal - 157.21;
  // lecture temperature bas de cuve pin A1:
  int Tbascuv1 = analogRead(Tbascuv);
  float Tbascuvd = Tbascuv1 * cal - 157.21;

  // calcul difference en degre
  float diffpancuv = (Tpand - Tbascuvd);

 // si la difference est superieure a 7 deg (12 bit):
  if (diffpancuv >= 7) {
    // commande relais R1 du circulateur panneau P1:
    digitalWrite(relcircul1, HIGH);
  }
 // si la difference est inferieure a 3 deg (5 bit):
  if (diffpancuv <= 3) {
    // arret circulateur panneau P1:
    digitalWrite(relcircul1, LOW);
  }
  // envoi des valeurs Tpan1 Tbascuv1 et diffpancuv sur port serie:
  Serial.print("Tpand: ");
  Serial.print(Tpand);
  Serial.print("\t Tbascuvd: ");
  Serial.print(Tbascuvd);
  Serial.print("\t diffpancuv: ");
  Serial.println(diffpancuv);
}

// programme de refroidissement de la cuve - limitation a 63deg
void limitcuv() {
  // lecture temperature haut de cuve pin A2:
  int Thautcuv1 = analogRead(Thautcuv);
```

```
  float Thautcuvd = Thautcuv1 * cal - 157.21;
  // verification position ete hiver pour modifier la limitation haute:
  etehiverState = digitalRead(etehiver);

  // limitation a 63deg (365 bit) ete:
  float limith = 64;
  float limitb = 62; //evite les basculements intempestifs

  if (etehiverState == HIGH) {
    // limitation a 75deg (365 bit) hiver:
    float limith = limith + 12;
    float limitb = limitb + 12;
  }

  Serial.print("\t Thautcuvd: ");
  Serial.print(Thautcuvd);

 // limitation de la cuve a 63deg:
  if (Thautcuvd >= limith) {
    // commande relais R2 du circulateur de refroidissement P2:
    digitalWrite(relrefroid1, HIGH);
  }
 // retour situation nominale:
  if (Thautcuvd < limitb) {
    // arret du circulateur de refroidissement P2:
    digitalWrite(relrefroid1, LOW);
  }
}

// programmes de chauffage du plancher chauffant:
// commande electrovanne EV4:
void cdeEV4() {
  // lecture temperature haut de la cuve pin A2:
  int Thautcuv1 = analogRead(Thautcuv);
  float Thautcuvd = Thautcuv1 * cal - 157.21;
  // lecture temperature retour plancher chauffant pin A3:
  int Tretplancher1 = analogRead(Tretplancher);
  float Tretplancherd = Tretplancher1 * cal - 157.21;
  // lecture temperature aller plancher chauffant pin A3 pour info:
  int Tallplancher1 = analogRead(Tallplancher);
  float Tallplancherd = Tallplancher1 * cal - 157.21;

  // calcul difference
  float diffcuvplan = (Thautcuvd - Tretplancherd);
  // envoi sur port serie:
  Serial.print("\t Tretplancher1: ");
  Serial.print(Tretplancherd);
  Serial.print("\t Tallplancher1: ");
  Serial.print(Tallplancherd);
  Serial.print("\t diffcuvplan: ");
  Serial.println(diffcuvplan);

 // si la difference est superieure a 4 deg (7 bit):
```

```
  if (diffcuvplan >= 4) {
    // commande ev4 relais R3:
    digitalWrite(ev4, HIGH);
  }
  else {
          digitalWrite(ev4, LOW);
  }
}
```

// **programme de pilotage de la vanne 3 voies solaire avec Arduino:**

```
void cde3v() {
  // lire temperature haut de la cuve pin A2:
  int Thautcuv1 = analogRead(Thautcuv);
  float Thautcuvd = Thautcuv1 * cal - 157.21;
  // lire temperature retour plancher chauffant pin A3:
  int Tretplancher1 = analogRead(Tretplancher);
  float Tretplancherd = Tretplancher1 * cal - 157.21;
  // calcul difference
  float diffcuvplan = (Thautcuvd - Tretplancherd);

  // ouverture si la difference est superieure a 7 deg (12 bit):
  if (diffcuvplan >= 7) {
    // commande relais ouverture vanne 3 voies:
          digitalWrite(Vferme, LOW);
          j = 0;
          // evite le basculement vanne 3 voies si temperature de la cuve basse
          if (Thautcuvd > 32) {
                  cuvechaude = HIGH;
          }
          else {
                  cuvechaude = LOW;
          }
    // inhibe commande si vanne 3 voies pilotee par regul chaudiere
          if (inhibcde3v == HIGH) {
                  // commande vanne 3 voies V2 periode 20sec fonctionnement 5sec
                  unsigned long currentMillis = millis();
                  if(currentMillis - previousMillis < 0) {
                          previousMillis = currentMillis;
                  }
                  if(currentMillis - previousMillis > interval) {
                          previousMillis = currentMillis;
                          if (digitalRead(Vouver) == LOW) {
                            digitalWrite(Vouver, HIGH);
                            interval = 5000;
                          }
                          else {
                            digitalWrite(Vouver, LOW);
                            interval = 15000;
                          }
                  }
          }
  }
  // rien faire entre 6 et 7 deg (8 et 12 bit):
```

```
  else if (diffcuvplan < 7 && diffcuvplan >= 6) {
          digitalWrite(Vferme, LOW);
          digitalWrite(Vouver, LOW);
          cuvechaude = LOW;
  }
  // fermeture si la difference est inferieure a 6 deg (8 bit):
  else if (diffcuvplan < 6 && diffcuvplan > 3) {
    // commande relais fermeture vanne 3 voies:
          digitalWrite(Vouver, LOW);
          cuvechaude = LOW;
          i = 0;
          if (j == 1) {
        digitalWrite(Vferme, LOW);
      }
          if (inhibcde3v == HIGH && j == 0) {
                  // commande vanne 3 voies V2 periode 20sec fonctionnement 5sec
                  unsigned long currentMillis = millis();
                  if(currentMillis - previousMillis < 0) {
                          previousMillis = currentMillis;
                  }
                  if(currentMillis - previousMillis > interval) {
                  previousMillis = currentMillis;
                          if (digitalRead(Vferme) == LOW) {
                            digitalWrite(Vferme, HIGH);
                            interval = 5000;
                          }
                          else {
                            digitalWrite(Vferme, LOW);
                            interval = 10000;
                          }
                  }
          }
  }
  // arret alimentation relais
  else {
          cuvechaude = LOW;
          j = 1;
          // permet de fermer completement la vanne en 36*5 secondes = 3min
          if (i < 36) {
                  digitalWrite(Vferme, HIGH);
                  i = i + 1;
          }
          else {
                  digitalWrite(Vferme, LOW);
          }
  }
}
```

// **programme de basculement du pilotage de la vanne 3 voies (Arduino ou régulateur chaudiére)**:

```
void bascul3v() {
  // lecture de l etat des inters:
  avsachaudState = digitalRead(avsachaud);
  VchaudouvState = digitalRead(Vchaudouv);
```

```
  // envoi sur port serie
  Serial.print("\t avsachaudState: ");
  Serial.print(avsachaudState);
  Serial.print("\t VchaudouvState: ");
  Serial.println(VchaudouvState);

  if (avsachaudState == LOW) {
    // commande relais R4 R5:
          digitalWrite(basc3v, HIGH);
          digitalWrite(Vouver, LOW);
          digitalWrite(Vferme, LOW);
          inhibcde3v = LOW;
  }
  else {
    // cuve froide
    if (cuvechaude == LOW) {
      digitalWrite(basc3v, LOW);
      inhibcde3v = HIGH;
      }
      // cuve chaude
      else if (cuvechaude == HIGH && VchaudouvState == HIGH) {
          digitalWrite(basc3v, HIGH);
          digitalWrite(Vouver, LOW);
          digitalWrite(Vferme, LOW);
          inhibcde3v = LOW;
      }
      // cuve chaude mais contact vanne chaudiere non active
      else {
          digitalWrite(basc3v, LOW);
          inhibcde3v = HIGH;
          }
  }
}

// alarme de depassement de temperature sur panneau et sur cuve
void alarm() {
  // lire temperature haut de la cuve pin A2:
  int Thautcuv1 = analogRead(Thautcuv);
  float Thautcuvd = Thautcuv1 * cal - 157.21;
  // lire temperature panneau pin A0:
  int Tpan1 = analogRead(Tpan);
  float Tpand = Tpan1 * cal - 157.21;

  // limitation panneau a 100 deg (425 bit):
  int limithpan = 100;
  // limitation cuve a 80 deg (392  bit):
  int limithcuv = 80;

 // alarme sur panneau:
  if (Tpand >= limithpan) {
    // commande buzzer:
    digitalWrite(alarms, HIGH);
```

```
  }
 // alarme sur cuve:
  if (Thautcuvd >= limithcuv) {
    // commande buzzer:
    digitalWrite(alarms, HIGH);
  }
  // pour arreter l alarme il faut eteindre le microcontroleur
}
```