

```

/*****
* SENSEO v24 - MACHINE À CAFÉ PROFESSIONNELLE
* Auteur : JEAN SHEMA
* Fonctions : 1/2 tasses, Mode ÉCO, Chauffe réaliste, Pompe,
*             Réservoir vide, Compteur, LCD, Buzzer, LEDs
* TMP36 : Température ambiante au démarrage
* Chauffe : Simulée (+5°C toutes les 0.25s)
*****/

```

```
#include <LiquidCrystal.h>
```

```
// === BROCHES ===
```

```

const int tempPin    = A1; // TMP36 (temp ambiante au démarrage)
const int potPin     = A5; // Potentiomètre (niveau d'eau)
const int btnStart   = A4; // Bouton START
const int btn1       = A2; // Bouton 1 tasse
const int btn2       = A3; // Bouton 2 tasses
const int led1       = 6;  // LED 1 tasse (jaune)
const int led2       = 7;  // LED 2 tasses (orange)
const int ledHeat    = 8;  // LED chauffe (rouge)
const int ledPump    = 9;  // LED pompe (bleue)
const int ledRun     = 10; // LED en marche (verte)
const int ledEmpty   = 13; // LED réservoir vide (rouge clignotante)
const int buzzer     = A0; // Buzzer

```

```
// === LCD 16x2 (6 fils) ===
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
// === VARIABLES ===
```

```

int mode = 0; // 0 = attente, 1 = 1 tasse, 2 = 2 tasses
int totalCoffees = 0; // Compteur total de cafés
int coffees = 0; // Compteur pour réservoir
const int MAX_COFFEES = 8; // 8 cafés = réservoir vide
bool isEmpty = false; // Réservoir vide ?
bool ecoMode = false; // Mode ÉCO activé ?
unsigned long lastBlink = 0; // Clignotement LED vide
unsigned long btnPressTime = 0; // Appui long
int waterLevel = 50; // Niveau d'eau (%)
float temperature = 20.0; // Température (°C)
const float TARGET_TEMP = 85.0; // Cible chauffe
unsigned long heatStart = 0; // Début chauffe

```

```
// === MÉLODIES ===
```

```

int victory[] = {660, 880, 660, 880, 660, 880, 1320, 1320};
int durVictory[] = {200, 200, 200, 200, 200, 200, 400, 600};

```

```
int startup[] = {523, 659, 784, 1047};
```

```

int durStartup[] = {150, 150, 150, 300};

void setup() {
  // Boutons en entrée avec pull-up
  pinMode(btnStart, INPUT_PULLUP);
  pinMode(btn1, INPUT_PULLUP);
  pinMode(btn2, INPUT_PULLUP);

  // LEDs + Buzzer en sortie
  for (int i = 6; i <= 10; i++) pinMode(i, OUTPUT);
  pinMode(ledEmpty, OUTPUT);
  pinMode(buzzer, OUTPUT);

  // === TEMP AMBIANTE AU DÉMARRAGE (TMP36) ===
  int raw = analogRead(tempPin);
  float voltage = raw * 5.0 / 1024.0;
  temperature = (voltage - 0.5) * 100.0;
  if (temperature < 15 || temperature > 40) temperature = 20.0; // Valeur sûre

  // LCD
  lcd.begin(16, 2);
  lcd.print(" Bonjour JEAN! ");
  lcd.setCursor(0, 1);
  lcd.print("Temp: "); lcd.print(temperature, 1); lcd.print(" C");

  // Mélodie de démarrage
  playMelody(startup, durStartup, 4);
  delay(1000);

  lcd.clear();
  lcd.print("SENSEO READY");
  lcd.setCursor(0, 1);
  lcd.print("1 ou 2 tasses?");

  // Moniteur série
  Serial.begin(9600);
  Serial.println("=== SENSEO v24 - JEAN ===");
  Serial.print("Temp ambiante: "); Serial.println(temperature, 1);
}

void loop() {
  waterLevel = map(analogRead(potPin), 0, 1023, 30, 100);

  // === MODE ÉCO (appui long sur 1 tasse) ===
  if (digitalRead(btn1) == LOW && mode == 0 && !isEmpty) {
    btnPressTime = millis();
    while (digitalRead(btn1) == LOW && millis() - btnPressTime < 2000) {

```

```

    if (millis() - btnPressTime > 1000) {
        ecoMode = true;
        lcd.setCursor(0, 1); lcd.print("MODE ECO ON  ");
        digitalWrite(led1, HIGH);
        tone(buzzer, 1500, 100);
        Serial.println("MODE ECO active");
        delay(500);
        break;
    }
}
if (!ecoMode && millis() - btnPressTime < 1000) {
    mode = 1;
    digitalWrite(led1, HIGH); digitalWrite(led2, LOW);
    lcd.setCursor(0, 1); lcd.print("1 TASSE OK  ");
    Serial.println("1 TASSE selectionnee");
}
while (digitalRead(btn1) == LOW);
}

// === 2 TASSES ===
if (digitalRead(btn2) == LOW && mode == 0 && !isEmpty) {
    delay(50);
    if (digitalRead(btn2) == LOW) {
        mode = 2;
        digitalWrite(led2, HIGH); digitalWrite(led1, LOW);
        lcd.setCursor(0, 1); lcd.print("2 TASSES OK  ");
        Serial.println("2 TASSES selectionnees");
        while (digitalRead(btn2) == LOW);
    }
}

// === START ===
if (digitalRead(btnStart) == LOW && mode > 0 && !isEmpty) {
    delay(50);
    if (digitalRead(btnStart) == LOW) {
        digitalWrite(ledRun, HIGH);
        heatStart = millis();
        temperature = 20.0; // Réinitialisation chauffe
        lcd.clear(); lcd.print("CHAUFFAGE...");
        Serial.println("ETAPE: Chauffage lent...");

        digitalWrite(ledHeat, HIGH);
        while (temperature < TARGET_TEMP) {
            unsigned long elapsed = (millis() - heatStart);
            temperature = 20.0 + (elapsed / 250.0) * 5.0; // +5°C / 0.25s
            if (temperature > TARGET_TEMP) temperature = TARGET_TEMP;
        }
    }
}

```

```

    lcd.setCursor(0, 1);
    lcd.print("Temp: "); lcd.print(temperature, 1); lcd.print(" C ");
    Serial.print("Chauffage... "); Serial.println(temperature, 1);
    delay(250);
}
digitalWrite(ledHeat, LOW);

// === POMPE ===
lcd.clear(); lcd.print("POMPE ACTIVE");
Serial.println("ETAPE: Pompe...");
digitalWrite(ledPump, HIGH);
int baseTime = (mode == 1 && ecoMode) ? 2000 : (mode == 1 ? 3000 : 5000);
unsigned long start = millis();
while (millis() - start < baseTime) {
    int remaining = (baseTime - (millis() - start)) / 1000;
    lcd.setCursor(0, 1);
    lcd.print(mode); lcd.print("x"); lcd.print(waterLevel); lcd.print("% ");
    lcd.print(remaining); lcd.print("s ");
    lcd.print((millis() / 300) % 3 == 0 ? "." : (millis() / 300) % 3 == 1 ? ".." : "...");
    delay(100);
}
digitalWrite(ledPump, LOW);

// === CAFÉ PRÊT ===
playMelody(victory, durVictory, 8);
Serial.println("CAFE PRET !");

totalCoffees += mode;
coffees += mode;

if (coffees >= MAX_COFFEES) {
    isEmpty = true;
    lcd.clear(); lcd.print("RESERVOIR VIDE!");
    lcd.setCursor(0, 1); lcd.print("START=remplir");
    Serial.println("RESERVOIR VIDE ! Appuyez sur START");
} else {
    lcd.clear(); lcd.print("CAFE PRET !");
    lcd.setCursor(0, 1); lcd.print("ENJOY YOUR COFFEE!");
    Serial.println("ENJOY YOUR COFFEE! Total: " + String(totalCoffees));
    delay(2500);
    resetToReady();
}

resetCycle();
while (digitalRead(btnStart) == LOW);
}
}

```

```

// === RÉSERVOIR VIDE ===
if (isEmpty) {
  if (millis() - lastBlink > 500) {
    digitalWrite(ledEmpty, !digitalRead(ledEmpty));
    lastBlink = millis();
  }
  if (digitalRead(btnStart) == LOW) {
    delay(300);
    coffees = 0; isEmpty = false; digitalWrite(ledEmpty, LOW);
    lcd.clear(); lcd.print("  REEMPLI ! ");
    tone(buzzer, 2000, 500);
    Serial.println("Reservoir rempli !");
    delay(1500); resetToReady();
  }
}
else if (mode > 0) {
  lcd.setCursor(0, 1);
  lcd.print("Qte: "); lcd.print(waterLevel); lcd.print("% T:");
  lcd.print(totalCoffees); lcd.print("  ");
}

delay(100);
}

// === FONCTIONS UTILITAIRES ===
void resetToReady() {
  lcd.clear(); lcd.print("SENSEO READY");
  lcd.setCursor(0, 1); lcd.print("1 ou 2 tasses?");
}

void resetCycle() {
  digitalWrite(ledRun, LOW); digitalWrite(led1, LOW); digitalWrite(led2, LOW);
  mode = 0; ecoMode = false; heatStart = 0;
}

void playMelody(int notes[], int durations[], int length) {
  for (int i = 0; i < length; i++) {
    tone(buzzer, notes[i], durations[i]);
    delay(durations[i] + 50);
  }
  noTone(buzzer);
}

```