CODE FOR PELTIER BASED LIQUID COOLING SYSTEM

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Pin definitions for temperature sensors
#define ONE_WIRE_BUS_HOT 2 // Hot side sensor (Arduino UNO pin D2)
#define ONE_WIRE_BUS_COLD 3 // Cold side sensor (Arduino UNO pin D3)

// Relay control pins (active low)
#define RELAY_PELTIER 7
#define RELAY_PUMP 8
#define RELAY_FAN 9

// MOSFET control pins for PWM
#define PWM_PELTIER 5
#define PWM_PUMP 6
#define PWM_FAN 10

// Warm-up duration (2 minutes)
#define WARMUP_DURATION 10000 // 120000 ms = 2 minutes

// Temperature thresholds for conditions
#define TEMP_MINIMUM 30.0
#define TEMP_LOW 40.0
#define TEMP_MEDIUM 60.0
#define TEMP_HIGH 90.0

// Create temperature sensor instances
OneWire oneWireHot(ONE_WIRE_BUS_HOT);
```

```
OneWire oneWireCold(ONE_WIRE_BUS_COLD);
DallasTemperature sensorsHot(&oneWireHot);
DallasTemperature sensorsCold(&oneWireCold);

// Variables to store temperature readings
float hotTemp, coldTemp;

// System states
bool warmUpCompleted = false;
unsigned long warmUpStartTime;

// PWM levels
const int PWM_OFF = 0;
const int PWM_MIN = 85;  // ~33% duty cycle for 8-bit PWM
const int PWM_MEDIUM = 170; // ~66% duty cycle
const int PWM_MAX = 255; // 100% duty cycle

void setup() {
  // Initialize serial monitor
  Serial.begin(9600);

  // Initialize temperature sensors
  sensorsHot.begin();
  sensorsCold.begin();

  // Configure relay pins as outputs
  pinMode(RELAY_PELTIER, OUTPUT);
  pinMode(RELAY_PUMP, OUTPUT);
  pinMode(RELAY_FAN, OUTPUT);

  // Configure PWM pins as outputs
```

```
  pinMode(PWM_PELTIER, OUTPUT);
  pinMode(PWM_PUMP, OUTPUT);
  pinMode(PWM_FAN, OUTPUT);

  // Initialize relays (all components OFF)
  digitalWrite(RELAY_PELTIER, HIGH); // Relay active low
  digitalWrite(RELAY_PUMP, HIGH);
  digitalWrite(RELAY_FAN, HIGH);

  // Initialize PWM outputs to OFF
  analogWrite(PWM_PELTIER, PWM_OFF);
  analogWrite(PWM_PUMP, PWM_OFF);
  analogWrite(PWM_FAN, PWM_OFF);

  // Start warm-up sequence
  Serial.println("Starting warm-up...");
  warmUpStartTime = millis();
}

void loop() {
  // Read temperature sensors
  sensorsHot.requestTemperatures();
  sensorsCold.requestTemperatures();
  hotTemp = sensorsHot.getTempCByIndex(0);
  coldTemp = sensorsCold.getTempCByIndex(0);

  // Check if sensors are disconnected
  if (hotTemp == DEVICE_DISCONNECTED_C || coldTemp == DEVICE_DISCONNECTED_C) {
    Serial.println("Error: Temperature sensor disconnected!");
    delay(1000);
    return;
```

```
  }

  // Print temperatures
  Serial.print("Hot Temp: ");
  Serial.print(hotTemp);
  Serial.print(" °C, Cold Temp: ");
  Serial.println(coldTemp);

  // Warm-up phase (2 minutes)
  if (!warmUpCompleted) {
    if (millis() - warmUpStartTime < WARMUP_DURATION) {
      warmUp();
      return;
    } else {
      warmUpCompleted = true;
      Serial.println("Warm-up complete. Starting priority conditioning...");
    }
  }

  // Priority conditioning
  applyConditions();
  delay(1000); // Wait 1 second before next loop
}

// Warm-up function
void warmUp() {
  digitalWrite(RELAY_PELTIER, LOW); // Turn on Peltier relay
  digitalWrite(RELAY_FAN, LOW);    // Turn on Fan relay
  digitalWrite(RELAY_PUMP, HIGH); // Keep pump relay OFF

  analogWrite(PWM_PELTIER, PWM_MEDIUM); // Peltier at medium level
```

```
  analogWrite(PWM_FAN, PWM_MAX);       // Fan at maximum speed
  analogWrite(PWM_PUMP, PWM_OFF);      // Pump OFF

  Serial.println("Warm-up in progress: Peltier (MEDIUM), Fan (MAX), Pump (OFF)");
}

// Apply conditions based on temperature
void applyConditions() {
  // Safety: Turn OFF relays before switching to PWM
  digitalWrite(RELAY_PELTIER, HIGH);
  digitalWrite(RELAY_PUMP, HIGH);
  digitalWrite(RELAY_FAN, HIGH);

  if (hotTemp <= TEMP_MINIMUM) {
    // Condition 1: 0°C to 30°C
    digitalWrite(RELAY_PELTIER, LOW);
    digitalWrite(RELAY_FAN, LOW);
    analogWrite(PWM_PELTIER, PWM_MIN);
    analogWrite(PWM_FAN, PWM_MIN);
    analogWrite(PWM_PUMP, PWM_OFF);
    Serial.println("Condition 1: Peltier (MIN), Fan (MIN), Pump (OFF)");
  } else if (hotTemp <= TEMP_LOW) {
    // Condition 2: 30°C to 40°C
    digitalWrite(RELAY_PELTIER, LOW);
    digitalWrite(RELAY_FAN, LOW);
    digitalWrite(RELAY_PUMP, LOW);
    analogWrite(PWM_PELTIER, PWM_MIN);
    analogWrite(PWM_FAN, PWM_MIN);
    analogWrite(PWM_PUMP, PWM_MIN);
    Serial.println("Condition 2: Peltier (MIN), Fan (MIN), Pump (MIN)");
  } else if (hotTemp <= TEMP_MEDIUM) {
```

```
    // Condition 3: 40°C to 60°C
    digitalWrite(RELAY_PELTIER, LOW);
    digitalWrite(RELAY_FAN, LOW);
    digitalWrite(RELAY_PUMP, LOW);
    analogWrite(PWM_PELTIER, PWM_MEDIUM);
    analogWrite(PWM_FAN, PWM_MAX);
    analogWrite(PWM_PUMP, PWM_MEDIUM);
    Serial.println("Condition 3: Peltier (MEDIUM), Fan (MAX), Pump (MEDIUM)");
  } else if (hotTemp <= TEMP_HIGH) {
    // Condition 4: 60°C to 90°C
    digitalWrite(RELAY_PELTIER, LOW);
    digitalWrite(RELAY_FAN, LOW);
    digitalWrite(RELAY_PUMP, LOW);
    analogWrite(PWM_PELTIER, PWM_MAX);
    analogWrite(PWM_FAN, PWM_MAX);
    analogWrite(PWM_PUMP, PWM_MEDIUM);
    Serial.println("Condition 4: Peltier (MAX), Fan (MAX), Pump (MEDIUM)");
  } else {
    // Safety: Turn everything OFF if temperature exceeds 90°C
    digitalWrite(RELAY_PELTIER, HIGH);
    digitalWrite(RELAY_FAN, HIGH);
    digitalWrite(RELAY_PUMP, HIGH);
    analogWrite(PWM_PELTIER, PWM_OFF);
    analogWrite(PWM_FAN, PWM_OFF);
    analogWrite(PWM_PUMP, PWM_OFF);
    Serial.println("Safety: All components turned OFF.");
  }
}
```