**Embedded C Code for Temperature Monitoring**

Here's a simple Embedded C code that reads the temperature from a sensor and prints an alert message if the temperature exceeds 50 degrees Celsius:

#include <stdio.h>

// Mock function to simulate reading from a temperature sensor

float readTemperatureSensor() {

// Replace this with actual sensor reading code

return 51.0; // Example temperature for testing

}

int main() {

// Initialize the sensor (if needed)

// This part of the code will depend on the specific sensor and microcontroller used

while (1) {

// Read temperature from the sensor

float temperature = readTemperatureSensor();

// Check if the temperature exceeds 50 degrees Celsius

if (temperature > 50.0) {

// Print alert message

printf("Alert: Temperature exceeds 50 degrees Celsius! Current temperature: %.2f°C\n", temperature);

}

// Add a delay to avoid continuous polling

// This is just a simple example, adjust the delay as needed

\_delay\_ms(1000);

}

return 0;

}

### Explanation

1. **Initialize Sensor:**
   * Depending on your specific temperature sensor and microcontroller, you might need to initialize the sensor before reading data from it. This could involve setting up communication protocols like I2C, SPI, or UART.
2. **Read Temperature:**
   * The readTemperatureSensor function simulates reading from a temperature sensor. In a real implementation, this function would contain the actual code to read data from the sensor.
3. **Check Temperature:**
   * The main loop continuously reads the temperature from the sensor. It then checks if the temperature exceeds 50 degrees Celsius.
4. **Print Alert:**
   * If the temperature exceeds 50 degrees, the code prints an alert message with the current temperature.
5. **Delay:**
   * A delay is added to prevent continuous polling of the sensor. Adjust the delay duration based on the requirements of your application.