# Temperature sensor

A simple Embedded C code snippet that demonstrates how you can read temperature data from a sensor and print an alert message if the temperature exceeds 50 degrees Celsius

#include <stdio.h>

// Function to simulate reading temperature from sensor (replace with actual sensor reading code)

float readTemperature() {

// Simulating temperature reading (replace with actual sensor reading logic)

float temperature = 48.5; // Example temperature (in degrees Celsius)

return temperature;

}

int main() {

float currentTemperature;

// Simulate reading temperature from sensor

currentTemperature = readTemperature();

// Check if temperature exceeds 50 degrees Celsius

if (currentTemperature > 50.0) {

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

} else {

printf("Current temperature is %.2f°C. Within normal range.\n", currentTemperature);

}

return 0;

}

Explanation:

1. **readTemperature() Function:**
   * This function is a placeholder to simulate reading temperature from a sensor. In a real embedded system, you would replace this function with actual code to read data from your temperature sensor.
2. **Main Function:**
   * **main()** initializes a variable currentTemperature to store the temperature read from the sensor.
   * It calls readTemperature() to get the current temperature value.
3. **Temperature Check:**
   * The code then checks if currentTemperature exceeds 50 degrees Celsius.
   * If the temperature is above 50°C, it prints an alert message indicating the current temperature and that it exceeds the threshold.
   * If the temperature is 50°C or below, it prints a message indicating the temperature is within normal range.

Notes:

* **Sensor Integration:** Replace readTemperature() with actual code that communicates with your temperature sensor. This typically involves using hardware-specific libraries or protocols (like I2C, SPI, etc.) depending on your sensor.
* **Alert Mechanism:** In a real system, you might use additional mechanisms like setting a flag, triggering an interrupt, or communicating the alert to a monitoring system rather than just printing to the console.

Ensure to adapt this example to the specific hardware and sensor you're using, as embedded systems development can vary significantly based on the hardware and environment constraints.