\*\*GD32F103C8T6控制LED阵列及获取芯片内部温度传感器开发文档\*\*

\*\*一、简介\*\*

本开发文档旨在为开发者提供关于如何使用GD32F103C8T6微控制器控制LED阵列，并从中获取芯片内部温度传感器数据的详细指南。

\*\*二、硬件连接\*\*

1. \*\*LED阵列连接\*\*

将LED阵列的阳极（长脚）通过限流电阻连接到微控制器的GPIO口上，作为输出端口。将LED阵列的阴极（短脚）连接到微控制器的GND上，作为公共地线。

2. \*\*温度传感器连接\*\*

GD32F103C8T6内部集成了温度传感器，可以直接从芯片的ADC通道读取数据。将温度传感器的输出连接到ADC的输入，通过配置ADC的相关寄存器，即可读取温度数据。

\*\*三、软件编程\*\*

1. \*\*LED阵列控制\*\*

使用C语言编写程序，通过配置GPIO的相关寄存器来控制LED灯的亮灭。可以使用循环和条件语句来控制LED阵列显示不同的效果，如滚动文字、动态图像等。

2. \*\*温度传感器数据获取\*\*

首先，需要配置ADC的相关寄存器，设置合适的采样时间和分辨率。然后，通过读取ADC通道的数据，根据内部提供的转换公式，将ADC值转换为实际的温度值。转换公式通常可以在微控制器的数据手册中找到。

示例代码（伪代码）：

```c

#include "gd32f10x.h" // 包含微控制器头文件

#define LED\_PIN GPIO\_PIN\_0 // LED控制引脚定义

#define LED\_PORT GPIOA // LED端口定义

#define TEMP\_ADC\_CHANNEL ADC\_CHANNEL\_4 // 温度传感器ADC通道定义

// 初始化GPIO和ADC

void init\_hardware() {

rcu\_periph\_clock\_enable(RCU\_GPIOA); // 使能GPIOA时钟

gpio\_init(LED\_PORT, GPIO\_MODE\_OUT\_PP, GPIO\_OSPEED\_50MHZ, LED\_PIN); // 初始化LED引脚为输出模式

adc\_init(TEMP\_ADC\_CHANNEL, 1000000); // 初始化ADC，设置采样率为1MHz

}

// 读取温度传感器数据

int read\_temperature() {

int adc\_value = adc\_get\_value(TEMP\_ADC\_CHANNEL); // 读取ADC值

int temperature = convert\_adc\_to\_temperature(adc\_value); // 将ADC值转换为温度值

return temperature;

}

```

\*\*四、注意事项\*\*

1. 确保LED灯的正负极连接正确，避免短路或开路。

2. 根据实际应用需求，选择合适的限流电阻值，保护LED灯珠和微控制器。

3. 在读取温度传感器数据时，注意配置合适的采样时间和分辨率，以保证数据的准确性和实时性。