应用笔记

Application Note

文档编号：**AN1091**

APM32F4xx 系列 TSensor demo 例程应用笔记

版本：**V1.0**

1 引言

本应用笔记提供如何在 APM32F4xx 系列上使用 ADC1_CH16 进行实时检测内部温度传感器的温度的应用指南。

本应用笔记内容主要包含：内部温度传感器的介绍、应用例程设计和应用例程实际演示。

目录

2 温度传感器介绍

2.1 简介

温度传感器是一种测量物体冷热程度的设备，以可读的形式通过电信号提供温度测量。

2.2 温度传感器分类

接触式类型温度传感器和非接触式类型温度传感器。

2.3 常见的温度传感器

接触式类型温度传感器：热电阻、热电偶和恒温器等。

非接触式类型温度传感器：红外温度传感器。

3 芯片内部温度传感器介绍

3.1 概述

本文主要介绍使用 APM32F4xx_SDK_V1.2 中 ADC_TSensor 例程的用法，该例程监控芯片内部温度传感器的电压值，再通过公式转化成温度值。

3.2 内部温度传感器简介

1. APM32F40x 内部集成一个温度传感器 ，用来测量 CPU 及周围的温度。

2. 该内部传感器在芯片内部与 ADC1_CH16 通道相连，通过 ADC1_CH16 通道可以把传感器的模拟电压值转化成数字值，再通过公式计算得出温度值。

3. 温度传感器模拟输入推荐采样时间是 17.1μs。

4. APM32F40x 的内部温度传感器支持的温度范围为：-40~125°C。

3.3 原理

原理及转化过程如下图 1：

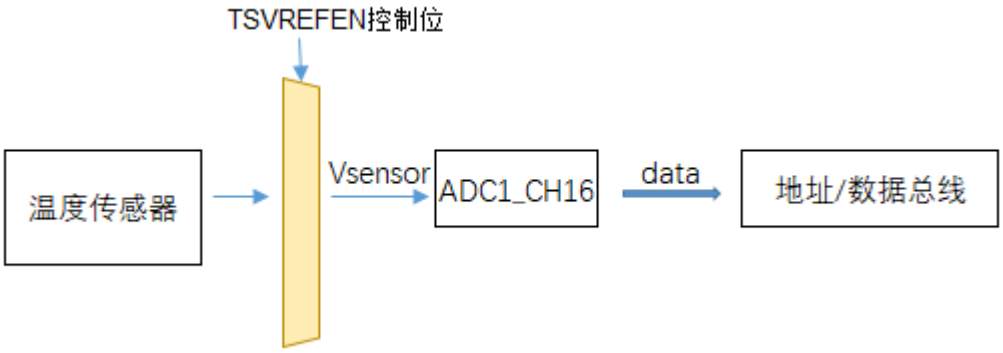

图 1

3.3.1 TSVREFEN 控制位

ADC_CCTRL 寄存器中的 TSVREFEN 位是控制内部温度传感器的使能状态。

3.3.2 计算公式

通过 ADC1_CH16 通道对温度传感器的模拟电压 V_{sensor} 进行采集，并转化成数字量 data，之后根据公式计算出温度传感器的温度值。

公式如下：

$$T = (V_{sensor} - V_{25}) / Slope + 25$$

公式说明：

T, V_{sensor}，是温度传感器的温度和输出电压；

V_{25}, 25 ，是基准值，25℃时，温度传感器的输出电压值；

Slope，是温度与 V_{sensor} 曲线的平均斜率（单位为 mv/° C 或 uv/° C）。

因为每个芯片的工艺特性都不会完全一样，所以根据公式计算出来的温度与实际温度会存在误差的。

根据多批次的测试数据，得出基准值和平均斜率的值，最后公式为：

$T = (V_{sensor} - 0.7782) / 0.0024 + 28$

误差大概在 4℃之内。

4 应用例程设计

4.1 编程思路

1. 配置 DMA 外设，启用 DMA 传输；
2. 配置 ADC1，并选择 channel16；
3. 配置采样时间为 480 周期，采样时间大于 17.1μs；
4. 设置 ADC1_CCTRL 的 TSVREFEN 位，使能内部温度传感器；
5. 设置 ADC1_CTRL2 的 DMAEN 位，使能 DMA 模式；
6. 设置 ADC1_CTRL2 的 ADCEN 位启动转换；
7. 读取 DMA 传输的 ADC1 转换结果；
8. 根据公式计算对应的温度值。
9. 配置串口，再利用串口把相关信息打印出来。

4.2 硬件设计

本例程只需要利用串口线，连接 USART1 的 TX 和 RX（TX:PA9, RX:PA10）。

4.3 软件设计

4.3.1 USART1 初始化

本例程运用了 USART1 打印信息，所以需要对 USART1 进行初始化配置。

USART1 配置如下：

```
USART_Config_T usartConfigStruct;


    /* USART configuration */
    USART_ConfigStructInit(&usartConfigStruct);
    usartConfigStruct.baudRate          = 115200;
    usartConfigStruct.mode              = USART_MODE_TX_RX;
    usartConfigStruct.parity            = USART_PARITY_NONE;
    usartConfigStruct.stopBits          = USART_STOP_BIT_1;
    usartConfigStruct.wordLength        = USART_WORD_LEN_8B;
    usartConfigStruct.hardwareFlow      = USART_HARDWARE_FLOW_NONE;

    /* COM1 init*/
    APM_MINI_COMInit(COM1, &usartConfigStruct);
```

4.3.2 DMA 初始化

本例程运用到了 DMA 外设，所以需要对 DMA 外设进行初始化配置。

DMA 配置如下：

```
/*!
 * @brief       DMA Init
*
 * @param       None
 *
 * @retval      None
 */
void DMA_Init(uint32_t* Buf)
{
    /* DMA Configure */
    DMA_Config_T dmaConfig;
    /* Enable DMA clock */
    RCM_EnableAHB1PeriphClock(RCM_AHB1_PERIPH_DMA2);
```

```
    /* size of buffer*/
    dmaConfig.bufferSize = 1;
    /* set memory Data Size*/
    dmaConfig.memoryDataSize = DMA_MEMORY_DATA_SIZE_HALFWORD;
    /* Set peripheral Data Size*/
    dmaConfig.peripheralDataSize = DMA_PERIPHERAL_DATA_SIZE_HALFWORD;
    /* Enable Memory Address increase*/
    dmaConfig.memoryInc = DMA_MEMORY_INC_DISABLE;
    /* Disable Peripheral Address increase*/
    dmaConfig.peripheralInc = DMA_PERIPHERAL_INC_DISABLE;
    /* Reset Circular Mode*/
    dmaConfig.loopMode = DMA_MODE_CIRCULAR;
    /* set priority*/
    dmaConfig.priority = DMA_PRIORITY_HIGH;
    /* read from peripheral*/
    dmaConfig.dir = DMA_DIR_PERIPHERALTOMEMORY;
    /* Set memory Address*/
    dmaConfig.memoryBaseAddr = (uint32_t)Buf;
    /* Set Peripheral Address*/
    dmaConfig.peripheralBaseAddr = (uint32_t)&ADC1->REGDATA;


    dmaConfig.channel = DMA_CHANNEL_0;
    dmaConfig.fifoMode = DMA_FIFOMODE_DISABLE;
    dmaConfig.fifoThreshold = DMA_FIFOTHRESHOLD_FULL;
    dmaConfig.peripheralBurst = DMA_PERIPHERALBURST_SINGLE;
    dmaConfig.memoryBurst = DMA_MEMORYBURST_SINGLE;

    DMA_Config(DMA2_Stream0, &dmaConfig);
    /* Clear DMA TF flag*/
    DMA_ClearIntFlag(DMA2_Stream0, DMA_INT_TCIFLG0);
    /* Enable DMA Interrupt*/
    DMA_EnableInterrupt(DMA2_Stream0, DMA_INT_TCIFLG);
    DMA_Enable(DMA2_Stream0);

}
```

4.3.3 ADC 初始化

本例程运用到了 ADC 外设，所以需要对 ADC 外设进行初始化配置。

ADC 配置如下：

```
/*!
 * @brief        ADC Init
 *
 * @param        None
 *
 * @retval       None
 */
void ADC_Init(void)
{
    ADC_Config_T    adcConfig;
    ADC_CommonConfig_T    adcCommonConfig;

    RCM_EnableAPB2PeriphClock(RCM_APB2_PERIPH_ADC1);

    /* ADC Configuration */
    ADC_Reset();
    ADC_ConfigStructInit(&adcConfig);
    /* Set resolution*/
    adcConfig.resolution = ADC_RESOLUTION_12BIT;
    /* Set dataAlign*/
    adcConfig.dataAlign = ADC_DATA_ALIGN_RIGHT;
    /* Set scanDir*/
    adcConfig.scanConvMode = DISABLE;
    /* Set convMode continous*/
    adcConfig.continuousConvMode = ENABLE;
    /* Set extTrigEdge*/
    adcConfig.extTrigEdge = ADC_EXT_TRIG_EDGE_NONE;
```

```
    /*Set ADC Clock Prescaler. ADC_Clock = APB2_Clock/4, 84/4=21MHz*/
    adcCommonConfig.prescaler = ADC_PRESCALER_DIV4;
    ADC_CommonConfig(&adcCommonConfig);

    ADC_Config(ADC1, &adcConfig);
    ADC_ConfigRegularChannel(ADC1, ADC_CHANNEL_16, 1, ADC_SAMPLETIME_480CYCLES);

    ADC_EnableTempSensorVrefint();

    ADC_EnableDMA(ADC1);
    ADC_EnableDMARequest(ADC1);

    /* Enable ADC*/
    ADC_Enable(ADC1);
    ADC_SoftwareStartConv(ADC1);
}
```

4.3.4 应用例程的功能逻辑主体

```
if (DMA_ReadStatusFlag(DMA2_Stream0, DMA_FLAG_TCIFLG0))
        {
            DataBuf = DMA_ConvertedValue;
            VSensorValue = (float)DataBuf/4095*3.3;

            /*!
             * According to actual test data of multiple bathes of chips,
             * V28 is adopted instead of V25 for this example. And 0.7782v is the voltage for 28℃
             */
            TSensorValue = (VSensorValue - 0.7782f)/0.0024f + 28;

            printf("\r\n");
            printf("ADC REGDATA = 0x%04X \r\n", DataBuf);
            printf("Volatage      = %f V \r\n", VSensorValue);
            printf("Temperature      = %f ℃ \r\n", TSensorValue);

            Delay(0xFFFFFF);
            DMA_ClearStatusFlag(DMA2_Stream0, DMA_FLAG_TCIFLG0);
        }
```

5 应用例程实际演示

本例程的实际演示环境是：

1. 在 APM32F407IG MINIBOARD 实物上演示；
2. 在 Keil MDK-ARM V5.29.0.0 的 IDE 上建立的工程，芯片 Device 选择 APM32F40IG

5.1 应用例程 Keil MDK 工程编译

完成工程代码后，编译无错误无警告，如下图 2 所示：

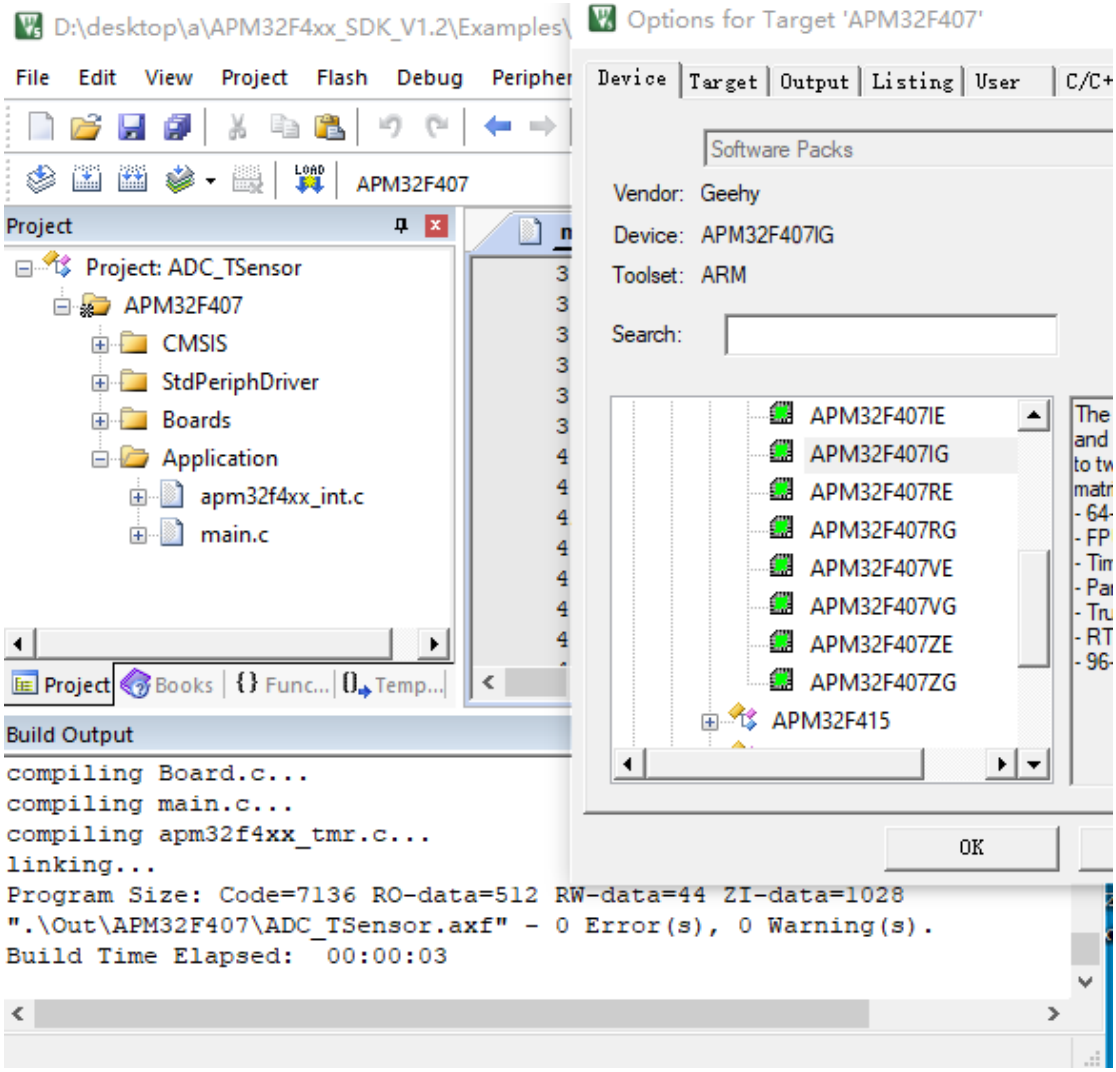

图 2

5.2 烧写程序，并观察数据结果

烧写程序成功，并在串口助手工具上显示相关的信息。如下图 3 所示：

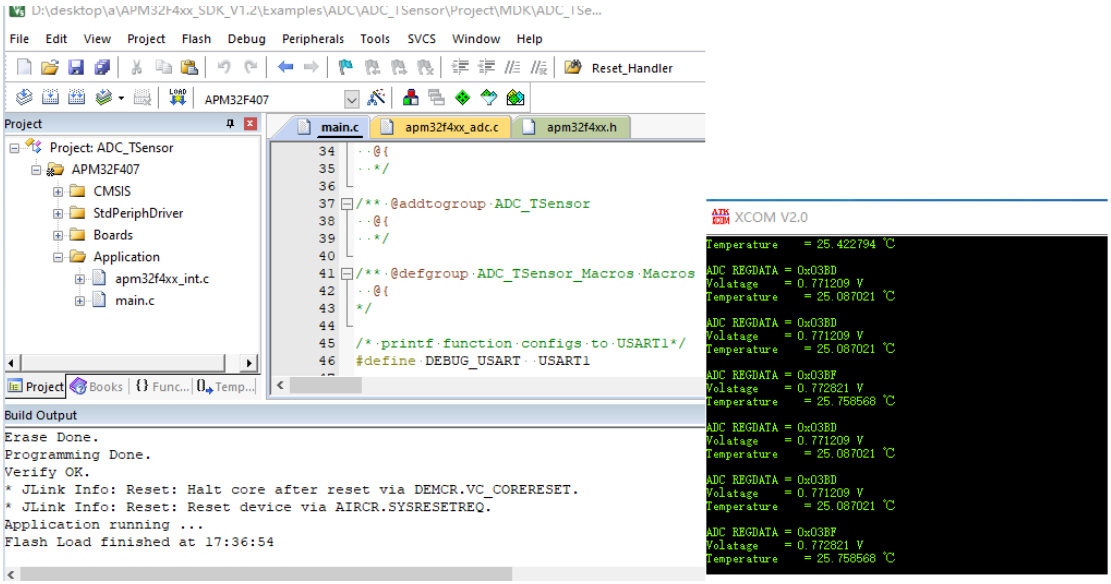
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Reset_Handler
APM32F407
Project
Project: ADC_TSensor
APM32F407
CMSIS
StdPeriphDriver
Boards
Application
apm32f4xx_int.c
main.c
main.c
apm32f4xx_adc.c
apm32f4xx.h
/** @addtogroup ADC_TSensor
/** @defgroup ADC_TSensor_Macros Macros
/* printf function configs to USART1*/
#define DEBUG_USART USART1
Build Output
Erase Done.
Programming Done.
Verify OK.
* JLink Info: Reset: Halt core after reset via DEMCR.VC_CORERESET.
* JLink Info: Reset: Reset device via AIRCR.SYSRESETREQ.
Application running ...
Flash Load finished at 17:36:54
XCOM V2.0
Temperature = 25.422794 ℃
ADC REGDATA = 0x03BD
Volatage = 0.771209 V
Temperature = 25.087021 ℃
ADC REGDATA = 0x03BD
Volatage = 0.771209 V
Temperature = 25.087021 ℃
ADC REGDATA = 0x03BF
Volatage = 0.772821 V
Temperature = 25.758568 ℃
ADC REGDATA = 0x03BD
Volatage = 0.771209 V
Temperature = 25.087021 ℃
ADC REGDATA = 0x03BD
Volatage = 0.771209 V
Temperature = 25.087021 ℃
ADC REGDATA = 0x03BF
Volatage = 0.772821 V
Temperature = 25.758568 ℃

图 3

6 版本历史

表 1 文档版本历史记录

日期	版本	变更历史
2023.03.01	1.0	新建

声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“TM”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，除非在极海销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其

本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及 / 或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册的任何第三方均不承担损害赔偿责任，包括任何一般、特殊因使用或无法使用本手册相关信息而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失）。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。