应用笔记

Application Note

AN1122

G32A14xx 系列 ADC 多通道连续/周期采样方案

版本：**V1.0**

1 引言

本应用笔记提供如何在 G32A14xx 系列应用程序实现 ADC 多通道连续/周期采样的指南。它提供了 PCB 原理图、模块配置、实现方式实践。

目录

2 应用目标

使用 G32A14xx MCU 实现多通道连续/周期采样的功能应用。

使用 G32A1465 评估板进行实例应用。

3 模块介绍

3.1 LPITMR 模块

LPITMR（低功耗周期中断定时器），G32A14xx 系列配置 1 个，4 通道的 LPITMR；在定时器达到设定的计数值后，相应的定时器通道将产生预触发和触发输出信号，这些预触发和触发输出可用于触发其他模块。

3.2 TMC 模块

TMC（多路触发控制），G32A14xx 系列配置 1 个，最多支持 4 个触发通道的 TMC；TMC 提供了一种可以灵活的将各个触发源通过软件配置连接到多个外设或引脚的机制。

3.3 PDU 模块

PDU（可编程延迟单元），G32A14xx 系列配置 2 个 PDU；为两个 ADC 产生触发和预触发（ADC 和 PDU 成对工作，如 PDU0 触发 ADC0，PDU1 触发 ADC1）；

3.4 ADC 模块

ADC（模数转换器），G32A14xx 系列配置 2 个 ADC，精度 12-bit；每个 ADC 多达 16 个采样通道；各通道转换模式有单次、连续转换。还可以选择硬件转换触发器和硬件通道。

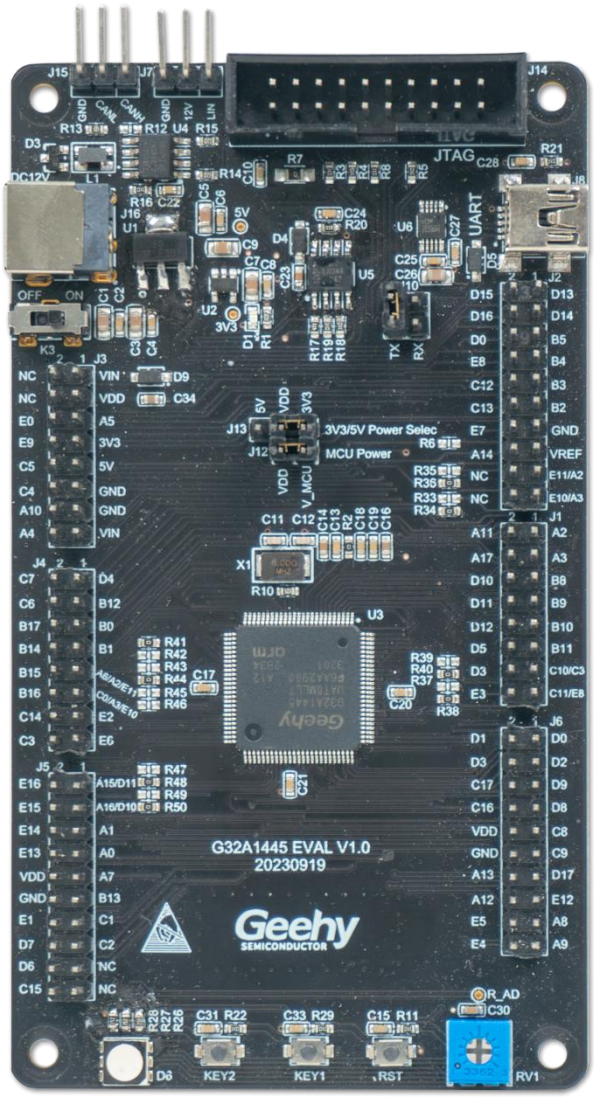
※ 注：模块详细说明请参考极海官方手册；

4 硬件介绍

4.1 硬件电路板

G32A1465 评估板是 G32A1465 MCU 的完整演示和开发平台,搭载了一颗 G32A1465 MCU 芯片，该芯片基于 Arm® Cortex® -M4 内核、工作主频 112MHz、Flash 1024KB。本评估板拥有丰富的外设功能，同时配套 EVAL SDK，可帮助开发者高效评估芯片性能或相关开发应用程序。

G32A1465 评估板可以由外部 12V 电源适配器供电，也可以通过 USB 供电；JTAG/SWD 接口可以提供给 MCU 3.3V 供电；评估板预留了 MCU 供电选择（J13 5V/3.3V）。



4.2 硬件原理图

参考极海官网 https://geehy.com

5 软件介绍

5.1 用户层

main.c：主函数入口，负责模块初始化，中断的触发，任务循环等；

g32a1xxx_int.c：中断处理函数；

user_xxxx_config.c:用户自定义配置信息；

5.2 外设驱动层

G32a1xxx_xxxx.c：负责 G32A14xx 芯片的外设驱动函数及配置，主要包括 CLOCK、GPIO、ADC、PDU、LPITMR 等；

5.3 板层驱动层

board.c：负责板端的驱动设置，端口的使用；如 LED，button 等；

基于 G32A14XX SDK 进行开发应用，SDK 包由极海官网（https://geehy.com）获取；

6 应用实现

6.1 ADC 多通道周期采样

使用 LPITMR 定时 3s 中断周期进行触发 PDU，由 PDU 进行预触发及触发 ADC 进行多通道采样；

(1) 配置 LPITMR 定时中断；

初始化 LPITMR：

```
LPITMR_USER_CHANNEL_CONFIG_T g_lpitmrChannelConfig =
{
    .timerMode = LPITMR_PERIODIC_COUNTER,
    周期计数模式；
    .periodUnits = LPIT_PERIOD_UNITS_MICROSECONDS,
    计数单位 ms；
    .period = 3000000U,
    计数周期 3s;
    .triggerSrc = LPITMR_TRIG_SRC_INTERNAL,
    触发源选择；
    .triggerSelect = 1U,
    选择内部触发；
    .reloadOnTriggerEn = true,
    设置是否重加载；
    .stopOnInterruptEn = true,
    设置定时器超时停止；
    .startOnTriggerEn = true,
    设置定时器检测计数递减；
    .channelChainEn = false,
    .channelChainEn = false,
    设置通道链接（使用多通道时有效）；
```

```
    .interruptEn = true，
    使能中断；
};
```

(2) 配置 TMC 多路触发方式；

```
TMC_USER_CONFIG_T tmc0_InitConfig = {
  .inOutMapCfgsCount = 1,
  配置输入输出触发映射数量；
  .inOutMapCfg = tmc0_InOutMappingConfig,
};
TMC_INOUT_MAPPING_CONFIG_T tmc0_InOutMappingConfig[2] =
{
    {TMC_TRIGGER_LPITMR_CH1, TMC_TARGET_PDU0_TRG_IN, false},
    配置触发源 LPITMR，触发目标 PDU0；
};
```

(3) 配置 PDU 触发及通道；

```
PDU_TIMER_CONFIG_T pduTimerCfg0 = {
    .clkPscMultFactor = PDU_CLK_PSCMULT_FACTOR_AS_10,
    预分频乘法系数选择；
    .loadValMode = PDU_LOAD_VAL_AT_NEXT_TRIGGER,
    加载模式选择；
    .triggerInput = PDU_TRIGGER_IN0,
    触发输入源；
```

```
    .clkPscDiv = PDU_CLK_PSCDIV2,
    预分频除法系数选择；
    .instanceBackToBackEn    = true,
    使能背靠背模式；
    .continuousModeEn = false,
    持续模式；
    .seqErrIntEn = false,
    错误中断；
    .dmaEn = false,
    DMA 使能；
    .intEn = true,
    中断使能；
};
PDU_Init(PDU0_INSTANCE,&pduTimerCfg0);
PDU_Init(PDU1_INSTANCE,&pduTimerCfg1);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg0);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg1);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg2);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg3);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg4);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg5);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg6);
PDU_ConfigAdcPreTrigger(PDU0_INSTANCE,0,&pdu0AdcTriggerCfg7);
配置 PDU0 通道 0 预触发配置 0~7;
Example：PDU_ADC_PRETRIGGER_CONFIG_T pdu0AdcTriggerCfg0 = {
 .preTriggerEn                = true,
 .preTriggerOutputEn          = true,
 .preTriggerBackToBackEn = false,
```

```
 .adcPreTriggerIndex        = 0U
};第一个通道配置（即 PDU0CH0 预触发 0）不使能背靠背模式，剩余通道使能背靠背；
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg0);
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg1);
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg2);
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg3);
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg4);
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg5);
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg6);
PDU_ConfigAdcPreTrigger(PDU1_INSTANCE,0,&pdu1AdcTriggerCfg7);
配置 PDU1 通道 0 预触发配置 0~7;
```

(4) 配置 ADC 触发通道；

```
ADC_CONV_CFG_T adcConvCfg0 = {
   .resolution = ADC_RESOLUTION_RATIO_12BIT,
      设置分辨率；
   .trigger = ADC_HARDWARE_TRIGGER,
      设置触发方式；
   .clockDivision = ADC_CLK_DIVISION_2,
      设置输入时钟分频系数；调制采样速率；
};
ADC_ConfigConverter(ADC0_INSTANCE,&adcConvCfg0);
ADC_ConfigConverter(ADC1_INSTANCE,&adcConvCfg0);
ADC_ConfigUserCalibration(ADC0_INSTANCE,&adcCalibration0);
ADC_ConfigUserCalibration(ADC1_INSTANCE,&adcCalibration1);
```

```
ADC_ConfigChan(ADC0_INSTANCE,0,&adcChanCfg0);
ADC_ConfigChan(ADC0_INSTANCE,1,&adcChanCfg1);
ADC_ConfigChan(ADC0_INSTANCE,2,&adcChanCfg2);
ADC_ConfigChan(ADC0_INSTANCE,3,&adcChanCfg3);
ADC_ConfigChan(ADC0_INSTANCE,4,&adcChanCfg4);
ADC_ConfigChan(ADC0_INSTANCE,5,&adcChanCfg5);
ADC_ConfigChan(ADC0_INSTANCE,6,&adcChanCfg6);
ADC_ConfigChan(ADC0_INSTANCE,7,&adcChanCfg7);

ADC_ConfigChan(ADC1_INSTANCE,0,&adcChanCfg16);
ADC_ConfigChan(ADC1_INSTANCE,1,&adcChanCfg17);
ADC_ConfigChan(ADC1_INSTANCE,2,&adcChanCfg18);
ADC_ConfigChan(ADC1_INSTANCE,3,&adcChanCfg19);
ADC_ConfigChan(ADC1_INSTANCE,4,&adcChanCfg20);
ADC_ConfigChan(ADC1_INSTANCE,5,&adcChanCfg21);
ADC_ConfigChan(ADC1_INSTANCE,6,&adcChanCfg22);
ADC_ConfigChan(ADC1_INSTANCE,7,&adcChanCfg23);

Example：ADC_CHAN_CONFIG_T adcChanCfg0 = {
    .interruptEnable = true,
    .channel    = ADC_INPUT_CHANNEL_TEMP
};因为使能了 PDU 中断，第一个通道配置使能中断，后面通道不使能中断由背靠背模式进行运行；
```

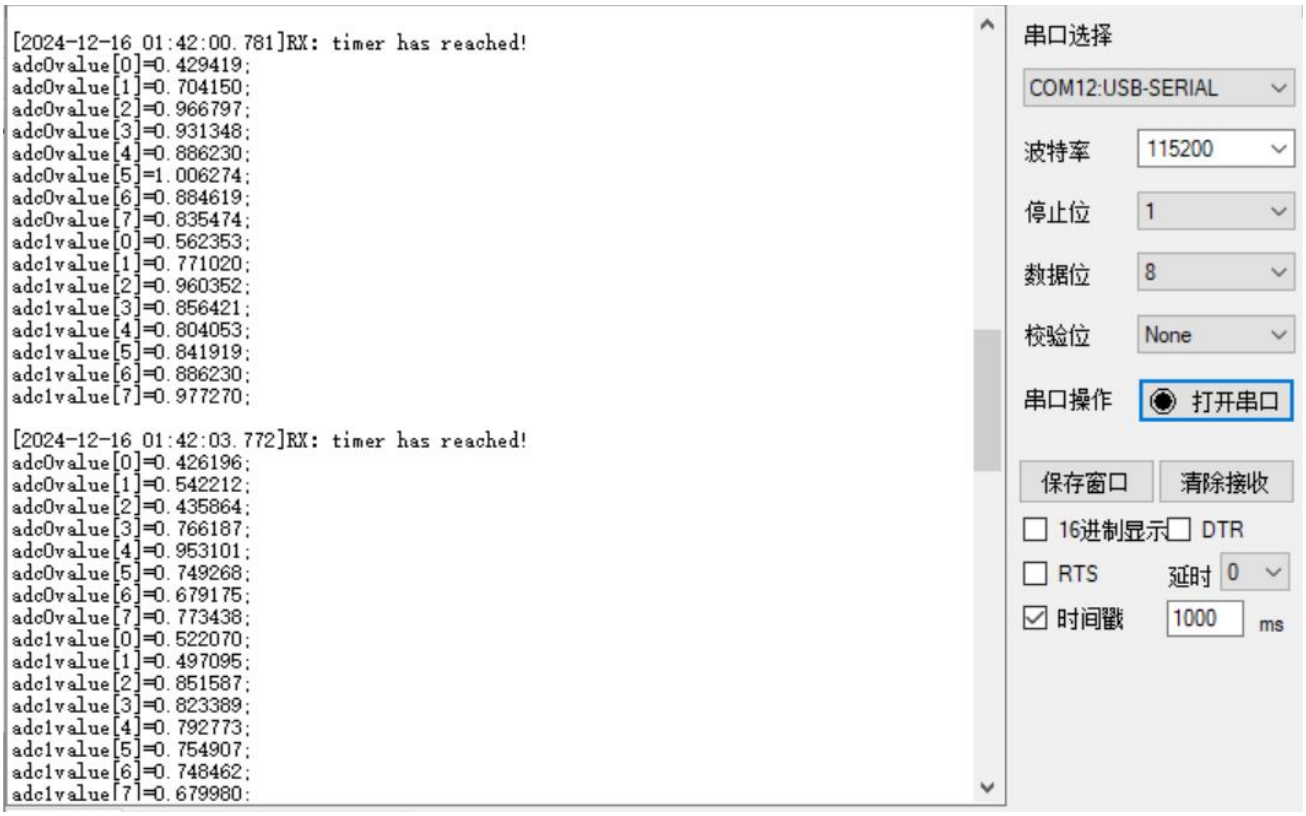
(5) 配置 LPUART 打印 ADC 转换数据；

```
adc0Value = (((float)adc0value[i]) / g_adcMaxValue)*g_adcDifference;
计算 ADC 值；
```

```
printf("adc0value[%d]=%f;\r\n",i,adc0Value);
```

使用 PMD14 作为 TX，PMD13 作为 RX；

打印通道 ADC 值；



6.2 示例例程

示例例程见附件《G32A1465 ADC multi-channels》。

7 版本历史

表格 1 文件版本历史

日期	版本	变更历史
2024.12	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“™”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，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在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。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。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

及产品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及产品而产生的直接、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），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

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。