

在 S32K 上使用 DMA 进行脉冲计数

1. 介绍

本应用笔记介绍了在恩智浦 S32K 系列 32 位汽车 MCU 上使用增强型直接存储器存取 (eDMA) 和 PORT 模块进行脉冲计数。信号脉冲的数量可以使用低功耗定时器 (LPTMR) 或 FlexTimer (FTM) 来计数。但是, 它们支持的脉冲输入的数量是有限的, 因为 FTM 模块通常用于生成 PWM, 而 LPTMR 内部只有一个计数器。

在本应用笔记中, PORT 模块用于捕获多个脉冲输入, eDMA 通道的当前主迭代计数器 (CITER) 寄存器用作简单的脉冲计数器。

这种 S32K 芯片配置提供多达 5 个可同时评估的脉冲输入通道。这样, CPU 负载也可迅速降低。

本应用笔记主要介绍 S32K 系列 32 位汽车 MCU。

目录

1. 介绍.....	1
2. 功能介绍.....	2
3. 初始化和应用.....	3
4. 功能限制.....	5
4.1. 最大 eDMA 传输.....	5
4.2. eDMA 主循环计数器.....	5
5. 结论.....	5
6. 修订历史.....	6



2. 功能介绍

eDMA 主要用于主存储器和另一个外设寄存器空间之间的数据传输，不需 CPU 干预。CITER 是 eDMA 结构的一部分，也可用于脉冲计数。

在接收到外设请求后，eDMA 模块的次循环开始执行基本数据传输。次循环完成后，CITER 递减。要从端口引脚捕获活动边沿，需要将 eDMA 模块的主循环计数器设置为最大值。然后，CITER 寄存器值反映实际脉冲计数。

S32K 器件包括 DMA 多路复用器 (DMAMUX)，它可将多达 63 个 eDMA 请求信号映射到 16 个 eDMA 通道中的任何一个。为了捕获多个脉冲输入，可以将 PORTA-PORTE 设置为 eDMA 模块 0-5 通道的源。

表 1. 通道请求源

eDMA 通道号	eDMA 请求源号	源描述
0	49	PORTA 请求
1	50	PORTB 请求
2	51	PORTC 请求
3	52	PORTD 请求
4	53	PORTE 请求

下面的代码显示了如何将 eDMA 通道请求源配置为相应的 PORT 请求：

```
DMAMUX->CHCFG[0] = DMAMUX_CHCFG_SOURCE(49) | DMAMUX_CHCFG_ENBL_MASK; // PORTA request
DMAMUX->CHCFG[1] = DMAMUX_CHCFG_SOURCE(50) | DMAMUX_CHCFG_ENBL_MASK; // PORTB request
DMAMUX->CHCFG[2] = DMAMUX_CHCFG_SOURCE(51) | DMAMUX_CHCFG_ENBL_MASK; // PORTC request
DMAMUX->CHCFG[3] = DMAMUX_CHCFG_SOURCE(52) | DMAMUX_CHCFG_ENBL_MASK; // PORTD request
DMAMUX->CHCFG[4] = DMAMUX_CHCFG_SOURCE(53) | DMAMUX_CHCFG_ENBL_MASK; // PORTE request
```

注意

因为单个 PORT 实例可以生成单个 eDMA 请求信号，所以只需将 PORT 实例中的单个引脚配置为脉冲输入。因此，脉冲输入的最大数量等于芯片的 PORT 模块实例的数量。每个 PORT 实例必须仅有一个引脚配置为脉冲输入。

3. 初始化和应用

要捕获多个脉冲输入，请按照以下步骤初始化 eDMA 和 PORT：

PORT 模块：

1. 启用 PORT 时钟。
2. 将 PORT 引脚多路复用器设置为 GPIO 模式。
3. 在上升沿/下降沿启用 eDMA 请求。

eDMA 模块：

1. 启用 DMAMUX 和 eDMA 时钟。
2. 在 DMAMUX 中配置 DMA 通道请求源。
3. 将 eDMA 通道源地址和目标地址设置为虚拟变量地址。
4. 将 eDMA 通道源和目标次要和主要地址调整和偏移设置为 0。
5. 将每个 eDMA 通道 CITER 和 BITER 寄存器设置为最大值。
6. 启用 eDMA 通道请求。

应用：

1. 初始化 eDMA 和 PORT 模块。
2. 读取 eDMA 通道 CITER 寄存器的值，以确定实际脉冲数。

下面是示例代码：

```

/* Variables declaration */
uint32_t dummy1;
uint32_t frequency_ch1;

void PORT_Init()
{
    /* Enable clock for PORTA-PORTE */
    PCC->PCCn[PCC_PORTA_INDEX] = PCC_PCCn_CGC_MASK;
    PCC->PCCn[PCC_PORTB_INDEX] = PCC_PCCn_CGC_MASK;
    PCC->PCCn[PCC_PORTC_INDEX] = PCC_PCCn_CGC_MASK;
    PCC->PCCn[PCC_PORTD_INDEX] = PCC_PCCn_CGC_MASK;
    PCC->PCCn[PCC_PORTE_INDEX] = PCC_PCCn_CGC_MASK;

    /* Configure PORT pin to GPIO mode and enable eDMA request on rising/falling edge */
    PORTA->PCR[8] = PORT_PCR_MUX(1) | PORT_PCR_IRQC(1);
    PORTB->PCR[8] = PORT_PCR_MUX(1) | PORT_PCR_IRQC(1);
    PORTC->PCR[8] = PORT_PCR_MUX(1) | PORT_PCR_IRQC(1);
    PORTD->PCR[8] = PORT_PCR_MUX(1) | PORT_PCR_IRQC(1);
    PORTE->PCR[8] = PORT_PCR_MUX(1) | PORT_PCR_IRQC(1);
}

void DMAMUX_Init()
{
    /* Enable clock for DMAMUX */

```

```
PCC->PCCn[PCC_DMAMUX0_INDEX] = PCC_PCCn_CGC_MASK;

/* Select an enable eDMA channel trigger source */
DMAMUX->CHCFG[0] = DMAMUX_CHCFG_SOURCE(49) | DMAMUX_CHCFG_ENBL_MASK;
DMAMUX->CHCFG[1] = DMAMUX_CHCFG_SOURCE(50) | DMAMUX_CHCFG_ENBL_MASK;
DMAMUX->CHCFG[2] = DMAMUX_CHCFG_SOURCE(51) | DMAMUX_CHCFG_ENBL_MASK;
DMAMUX->CHCFG[3] = DMAMUX_CHCFG_SOURCE(52) | DMAMUX_CHCFG_ENBL_MASK;
DMAMUX->CHCFG[4] = DMAMUX_CHCFG_SOURCE(53) | DMAMUX_CHCFG_ENBL_MASK;
}
```

在初始化 eDMA 寄存器之前，需要将 eDMA 源地址和目标地址填入到虚拟变量的地址值中。eDMA 模块的一个通道的初始化按照以下步骤进行：

```
void DMA_Init()
{
/* Enable clock for eDMA */
PCC->PCCn[PCC_DMA0_INDEX] = PCC_PCCn_CGC_MASK;

/* Source Configuration */
DMA->TCD0_SADDR = &dummys1;
DMA->TCD0_ATTR = DMA_TCD0_ATTR_SSIZE(0);
DMA->TCD0_SOFF = 0; // no address shift after each transfer
DMA->TCD0_SLAST = 0;

/* Destination Configuration */
DMA->TCD0_DADDR = &dummys1;
DMA->TCD0_ATTR = DMA_TCD0_ATTR_DSIZE(0);
DMA->TCD0_DOFF = 0; // no address shift after each transfer
DMA->TCD0_DLASTSGA = 0;

/* Set Citer and Biter to Maximum Value */
DMA->TCD0_CITER.TCD0_CITER_ELINKNO = DMA_TCD0_CITER_ELINKNO_CITER_MASK;
DMA->TCD0_BITER.TCD0_BITER_ELINKNO = DMA_TCD0_BITER_ELINKNO_BITER_MASK;
DMA->TCD0_NBYTES.TCD0_NBYTES_MLNO = 1; // transfer one byte on each trigger arrived

/* Start Transfer for Channel0 */
DMA->SERQ = DMA_SERQ_SERQ(0);
}
```

读取应用中的 eDMA 通道计数器。脉冲计数值等于 BITTER 减去 CITTER 寄存器值：

```
frequency_ch1 = (DMA->TCD0_BITER.TCD0_BITER_ELINKNO & DMA_TCD0_BITER_ELINKNO_BITER_MASK) -
(DMA->TCD0_CITER.TCD0_CITER_ELINKNO & DMA_TCD0_CITER_ELINKNO_CITER_MASK);
```

注意

在主迭代计数耗尽之前（即 CITER 减到 0），可以随时在应用中读取 eDMA 通道计数器。当 CITER 减到 0 时，用户必须将 CITER 寄存器重置为 BITTER，并重新启动计数操作。通过设置 TCDn_CSR 寄存器中的 INTMAJOR 位来配置 eDMA 通道，以便在 CITER 减到 0 时生成中断。

4. 功能限制

请注意本应用笔记中展示的实现的局限性。

4.1. 最大 eDMA 传输

被测量的信号的最大频率受限于 eDMA 引擎的最大传输速率，还受限于使用本应用笔记中描述的方法来同时测量的信号的数量。要将最终应用中启用的其他 eDMA 传输视为限制因素。

4.2. eDMA 主循环计数器

CITER 限制为 15 位。检查 CITER 寄存器并确保它不会减到零。当 CITER 寄存器减到零时，CITER 会从 BITER 寄存器重新加载值并继续递减。

5. 结论

eDMA 多脉冲计数器的实现可以使用内部硬件逻辑完成，无需 CPU 干预。CPU 只需读取 eDMA 通道 CITER 寄存器，即可获取实际脉冲计数值。

下图显示了 3 个不同频率（20 kHz、10 kHz 和 1 kHz）的测试方波信号。特定信号的频率是使用 eDMA 模块来确定的。

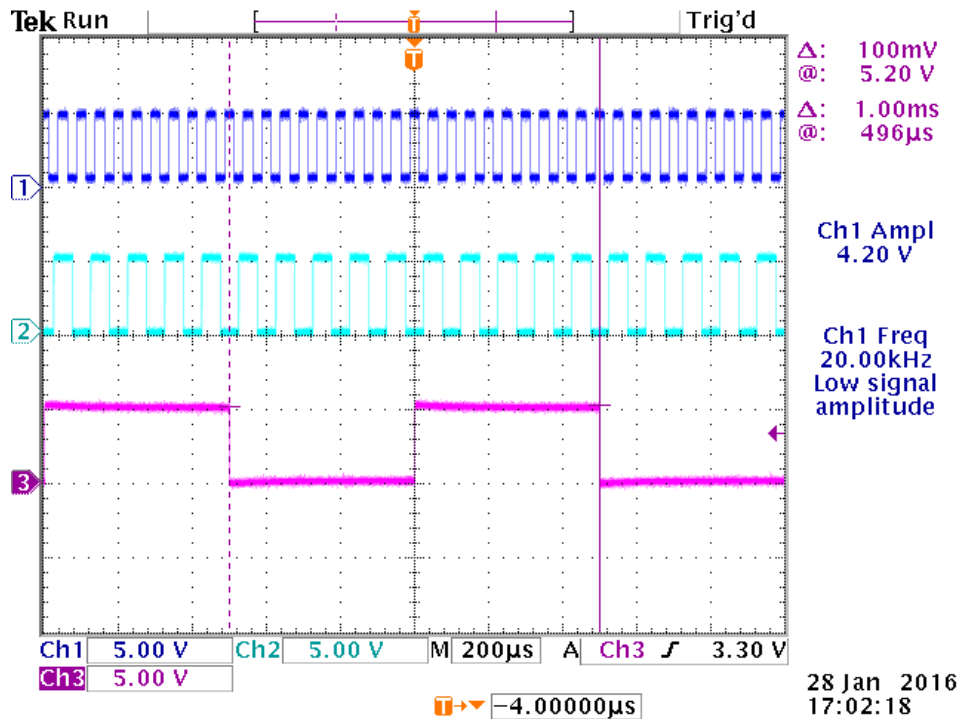


图 1. 输入方波

要测量输入信号的频率，请按以下步骤操作：

1. 启用 PIT 定时器以产生 1 秒的中断。
2. 在中断例程中，读取 BITER 和 CITER 寄存器之间的差值，并复位 eDMA 计数器。
3. 检查 S32 Design Studio 中“表达式 (Expression)”选项卡中的值。

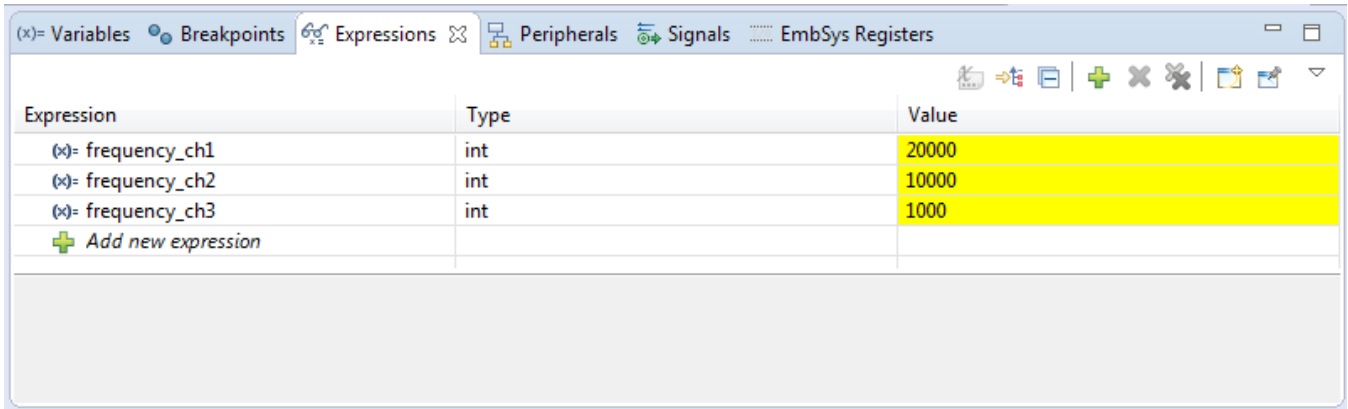


图 2. 在 S32 Design Studio 中测试方波的频率

6. 修订历史

下表总结了初次发布以来对本文档所做的变更。

表 2. 修订历史

版本号	日期	实质性变更
0	2016 年 2 月	初版发布

How to Reach Us:

Home Page:
nxp.com.cn

Web Support:
nxp.com.cn/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com.cn/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, OOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE LTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, TARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2016 NXP B.V.

Document Number: AN5258
Rev. 0
02/2016

