

基于数字电源控制器 UCD3138 的一种新的 输入电流检测方法

Neil Li, Sundy Xu

China Telecom Application Team

摘要

数字电源控制器 UCD3138 含有 AFE (analog front end, 模拟前端) 模块和 Filter 模块, 用于输出电压的环路反馈。而对上述模块灵活配置并组合后, 也可以完成对输入到 EADC(差分 ADC) 模块信号的平均值计算, 关键配置涉及到 EADC 的过采样功能(oversample), 平均功能(averaging) 功能和 Filter 模块的积分(integration)支路配置。考虑到电流互感器的输出端信号的平均值与实际输入电流存在比例关系, 因此通过读取该输出端信号的平均值便可得到实际的输入电流信息。

目录

1	引言	3
1.1	数字电源 UCD3138 的应用	3
1.2	隔离电源拓扑中的电流互感器	3
1.3	输入电流检测的方法	4
2	UCD3138 的 AFE 模块和 Filter 模块	4
2.1	模块功能概述	4
2.2	EADC 的 Oversample	5
2.3	EADC 的 averaging	5
2.4	UCD3138 的 Filter	6
2.5	全流程数据处理	6
3	读取三角波的平均值	7
3.1	测试原理简述	7
3.2	直流电平信号平均值的读取	7
3.3	三角波平均值的读取	7
3.4	软件流程与代码	8
4	实测单板输入电流	11
4.1	测试单板概述	11
4.2	实测数据	11
5	总结	12
6	参考文献	12

图

Figure 1.	硬开关全桥系统框图	3
Figure 2.	电流互感器应用电路	3
Figure 3.	电流互感器输出信号	4
Figure 4.	UCD3138 的 AFE 和 Filter	4

Figure 5.	EADC 的 Oversample.....	5
Figure 6.	EADC 的 Oversample.....	5
Figure 7.	UCD3138 的 Filter 结构.....	6
Figure 8.	全流程数据处理	6
Figure 9.	直流信号平均值读取	7
Figure 10.	三角波信号平均值读取.....	8
Figure 11.	软件处理流程.....	8
Figure 12.	软件处理流程.....	Error! Bookmark not defined.
Figure 13.	软件处理流程.....	Error! Bookmark not defined.

1 引言

1.1 数字电源控制器 UCD3138 的应用

数字电源控制器 UCD3138 因其自身所具备的良好的前馈功能，通信功能和可编程性等特点，在 DC/DC 电源中通常置于副边侧。常见的拓扑方案包括全桥，半桥和 LLC 等。图 1 所示的是应用数字电源（控制器）UCD3138 的硬开关全桥系统框图。UCD3138 位于副边侧，通过数字隔离器 ISO7420CF 完成驱动信号向原边侧的传递。

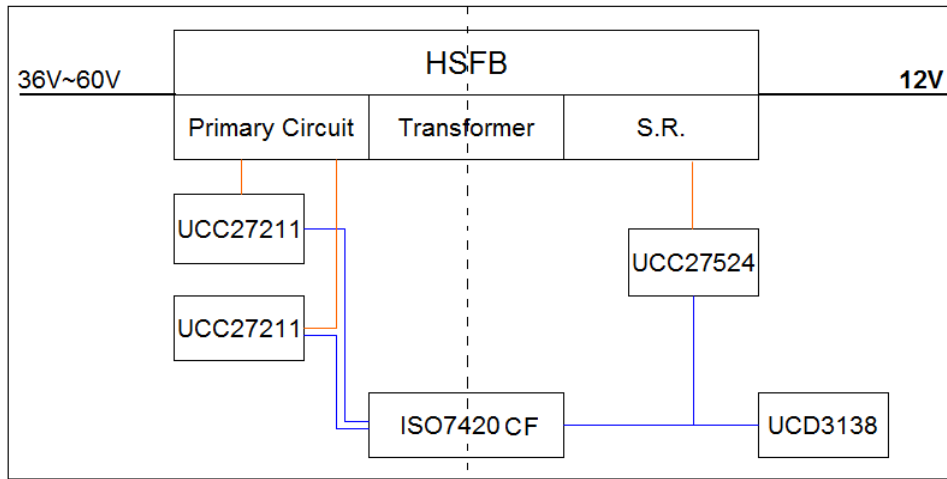


Figure 1. 硬开关全桥系统框图

1.2 隔离电源拓扑中的电流互感器

图 2 所示的是应用于全桥等拓扑中的电流互感器。其原边侧串入主功率回路，副边侧将按比例（比例系数为互感器的匝比 T ）衰减后的电流信息通过与采样电阻相乘得电压信息。位于副边侧的控制器 UCD3138 通过读取该电压信息，可以完成逐周期保护等功能。

互感器副边侧输出电压： $V_T = (I_{in} \div T) \times R_s$

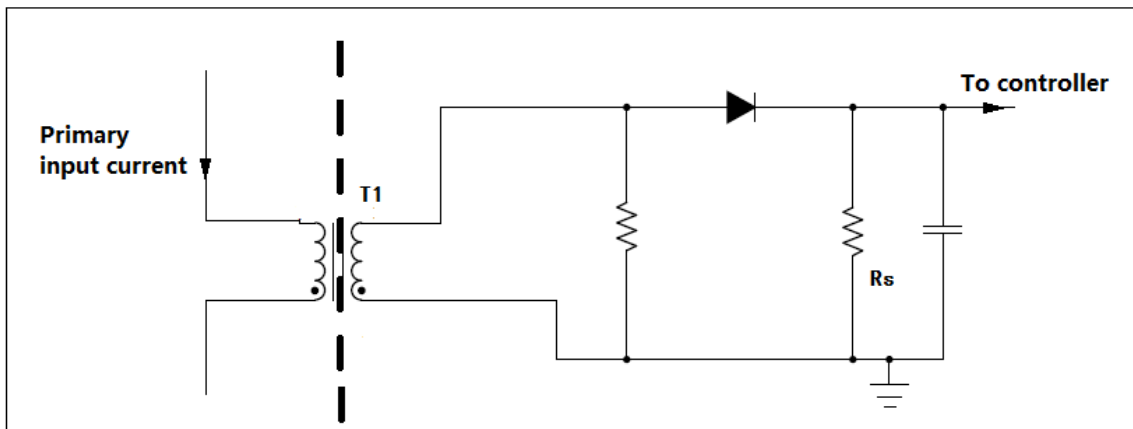


Figure 2. 电流互感器应用电路

1.3 输入电流检测的原理

图 3 所示意的是电流互感器副边侧输出端的电压信号。上下两组波形是在输入电压不同时所对应的输出。在输出功率确定后，随着输入电压变高，梯形波的上升沿将变陡，其平均值将变低。

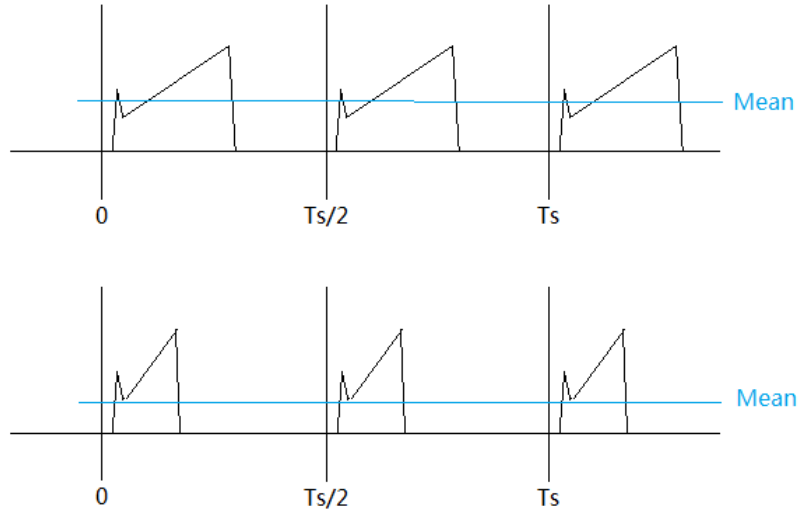


Figure 3. 电流互感器输出信号

电流互感器输出端的信号平均值与系统输入电流的平均值成近似比例关系，因此可以通过读取电流互感器输出端的平均值来推算输入电流。

2 UCD3138 的 AFE 模块和 Filter 模块

2.1 模块功能概述

UCD3138 的 AFE 和 Filter 用来完成对输出电压误差的采集，转换和环路计算，输出的数据进入到 DPWM 模块，最终生成合适的占空比。如图 4 所示。

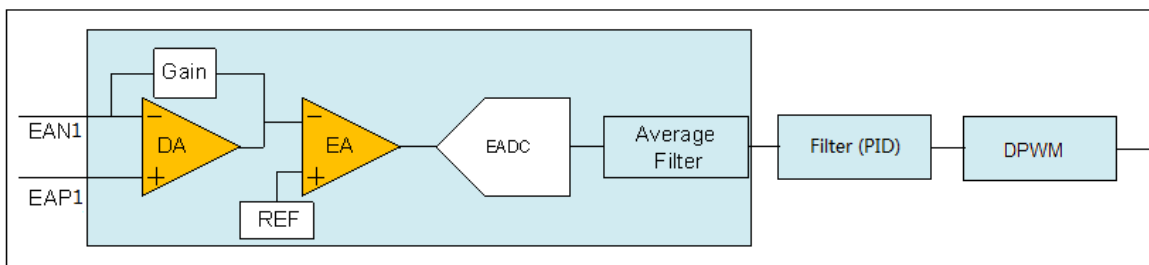


Figure 4. UCD3138 的 AFE 和 Filter

在实际应用中，可以用 AFE 和 Filter 来采集电流互感器输出端的信号，并最终计算出其平均值。该功能的实现依赖于 AFE 和 Filter 所具备的如下特点：

- 1) AFE 中的 EADC 具备 oversample 功能，可以在一个周期内采集 1,2,4,8 个样本；

- 2) AFE 对 EADC 输出的数据可以进行平均 (averaging)，即可以对连续采集到的 2,4,或 8 个数据做累加，然后除于个数以得到平均值。
- 3) Filter 是 PID 结构，因此可以只使用累加环节 (Integration 支路)，从而可以计算出一段时间内的累加和。

2.2 EADC 的 Oversample

DPWM 模块可以在 EADC 模块中产生采样触发信号，使 EADC 完成一次采样。同时，EADC 还具有多次 (2 次，4 次和 8 次) 采样的功能。以 8 次采样为例，当 EADC 收到 DPWM 的采样触发信号后，EADC 分别在采样基准的 1/8 处，2/8 处，3/8 处，4/8 处，5/8 处，6/8 处，7/8 处和采样基准处完成 8 次采样，如图 5 所示。

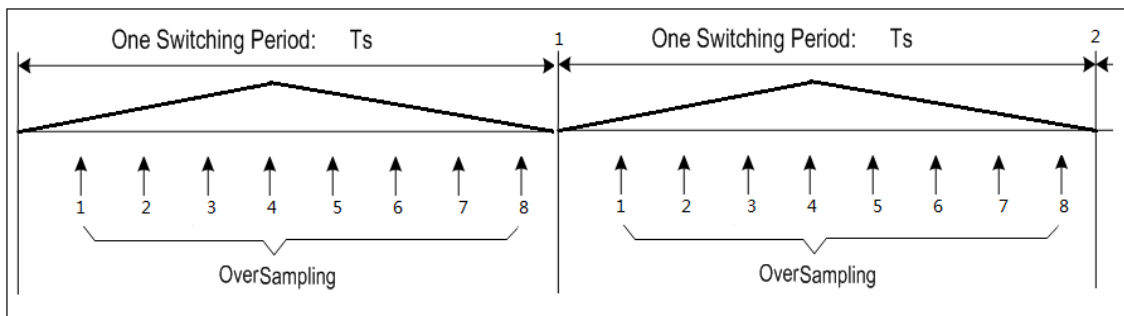


Figure 5. EADC 的 Oversample

2.3 EADC 的 averaging

EADC 提供了两种数据平均方式，分别是连续方式 (consecutive mode) 平均和空间方式 (spatial mode) 平均。图 6 示意的如何对数据进行连续方式平均。其原理就是对连续采样的 2,4 或 8 个数据进行累加，然后除于累加的个数，便得到了平均值。

计算后的平均值将送入到 Filter 环节。

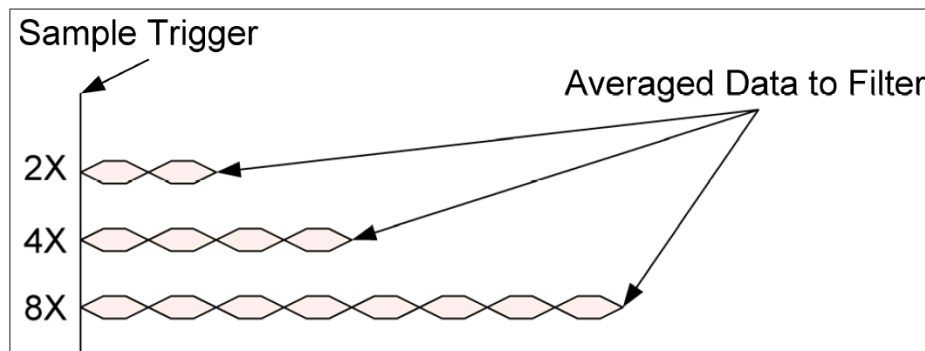


Figure 6. EADC 的 Oversample

2.4 UCD3138 的 Filter

图 7 是 UCD3138 的 filter，基于 PID 结构。当只使用 I 支路（即累加环节），可以对 X_n 数据进行不断的累加，累加的结果存储在 KI_YN 寄存器中。而 X_n 数据就是来自 EADC 的输出。

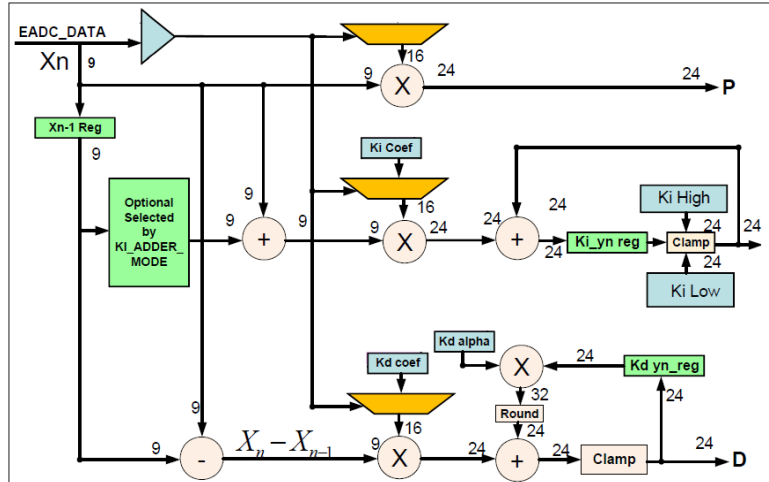


Figure 7. UCD3138 的 Filter 结构

2.5 全流程数据处理

当 oversample 配置为 8 次，EADC 的平均次数配置为 2 次，在 2 个周期的时间内 KI_YN 寄存器得到的数据如下图 8 所示：

- 1) 每个周期采样 8 个样本，故 2 个周期共采集 16 个；
- 2) 每 2 个样本进行平均，平均后的数据进入到累加环节；
- 3) 在 2 个周期内， KI_YN 寄存器中总共存储有 $8 \times 2 = 16$ 个样本的累加和；

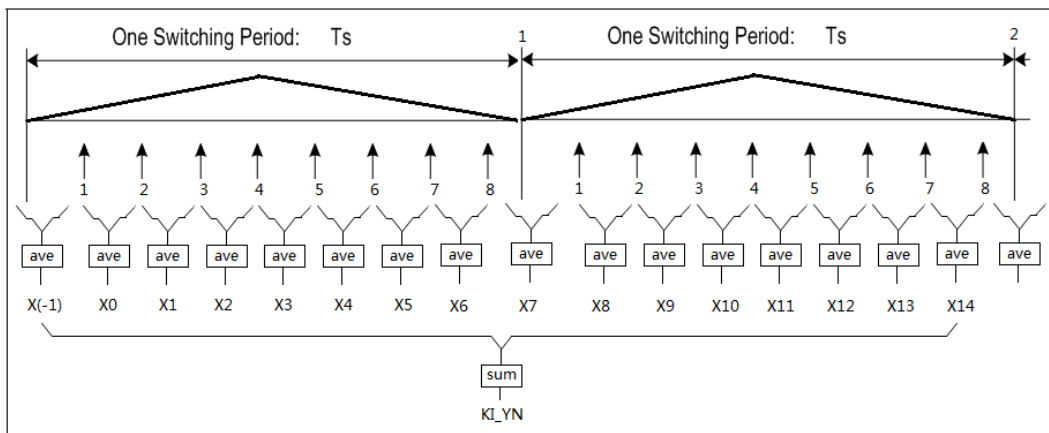


Figure 8. 全流程数据处理

3 读取三角波的平均值

3.1 测试原理简述

在 UCD3138 的 EAP1/EAN1 引脚（如图 4）接入直流电平信号或三角波信号，然后配置合理的 oversample, averaging 和采集的周期个数，最终在 KI_YN 寄存器可以得到样本总和。

然后将样本总和与样本个数相除，便得到了样本的平均值。

3.2 直流电平信号平均值的读取

在 EAP1 引脚接入直流电平信号（如图 9 左中的 CH1），实测平均值为 220mV。

通过配置 oversample 为 8 次，averaging 为 8 次，在两个周期内得到的平均值为 $-156 \div 8 = -19.5$ 。此时，DAC_Value 寄存器中写入的值为 2048，因此，根据软件采样确定的平均值为：

$$(2048 \div 16) \times 1.5625\text{mV} - (-19.5) = 219.5\text{mV}$$

该值与实际值的偏差小于-0.5%。上述公式的含义可以参考“参考文献 5”。

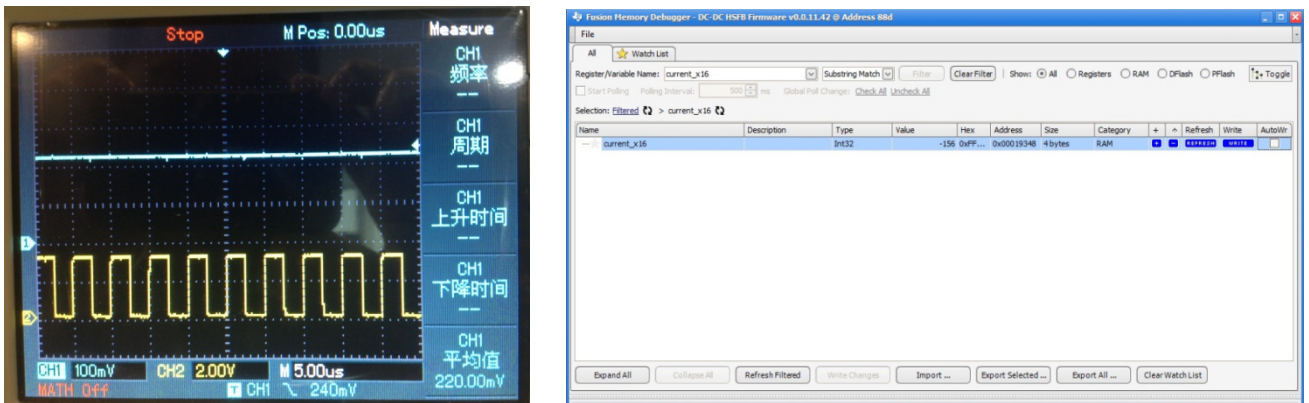


Figure 9. 直流信号平均值读取

3.3 三角流平均值的读取

在 EAP1 引脚接入三角波信号（如图 10 左中的 CH1），实测平均值为 136mV。

通过配置 oversample 为 8 次，averaging 为 8 次，在两个周期内得到的平均值为 $466 \div 8 = 58.25$ 。此时，DAC_Value 寄存器中写入的值为 2048，因此，根据软件采样确定的平均值为：

$$(2048 \div 16) \times 1.5625\text{mV} - 58.25 = 141.75\text{mV}$$

该值与实际值的偏差小于 5%。

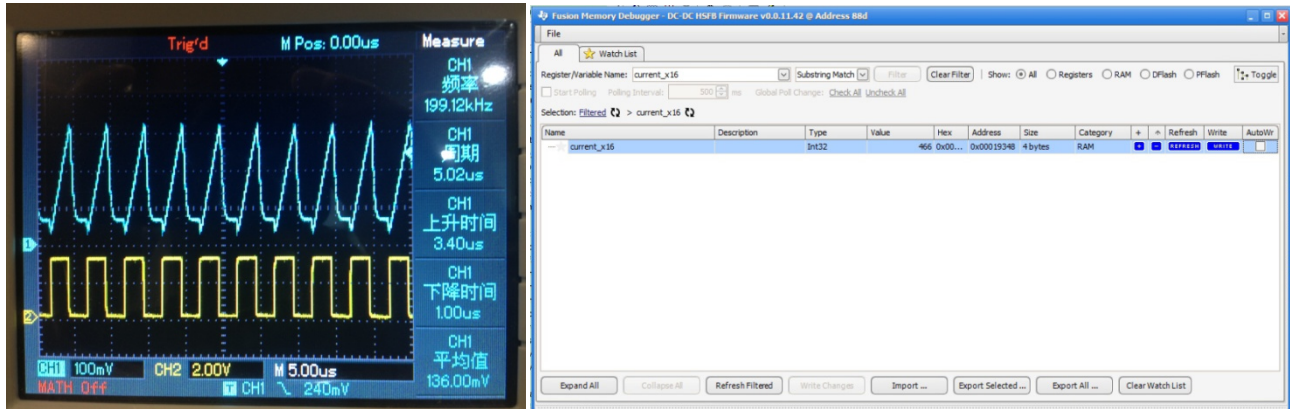


Figure 10. 三角波信号平均值读取

3.4 软件流程与代码

图 11 是整个数据处理的软件流程图，主要包含主程序中的初始化与配置，快中断程序中的数据处理等两个部分。

对快中断部分，使用周期快中断，中断间隔为 256 个周期。每次处理都是连续三次进入快中断，在第一次进入快中断后，配置 EADC 和 Filter；在第二次进入后进行数据读取，此时在 KI_YN 中共有 8 × 256=2048 个样本的累加和。（oversample 设置为 8）

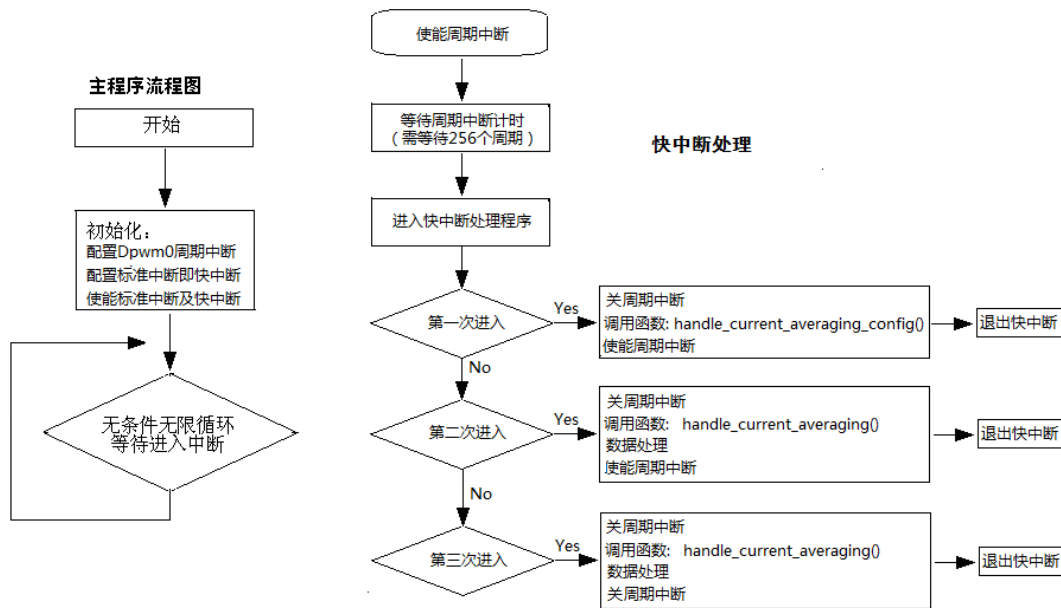


Figure 11. 软件处理流程

关键代码如下：

1. 配置 Dpwm0 周期中断及打开中断功能


```
// generate an interrupt on each 256 switching cycles
Dpwm0Regs.DPWMINT.bit.PRD_INT_SCALE = 15;

//Disable interrupt
Dpwm0Regs.DPWMINT.bit.PRD_INT_EN = 0;

disable_interrupt();
disable_fast_interrupt(); //Make sure fast interrupt is disabled
write_reqmask(CIMINT_ALL_DPWM0 | CIMINT_ALL_PWM2_COMP);
write_firqpr (CIMINT_ALL_DPWM0); //DPWM0 interrupt is mapped to FIQ
enable_fast_interrupt(); //Enable fast interrupt
enable_interrupt(); //Enable interrupt
```

2. 快中断处理程序

仅在第二次和第三次进入快中断后进行数据的读取。

```
#pragma INTERRUPT(fast_interrupt,FIQ)
void fast_interrupt(void)
{
    int temp;

    switch (cs_flag)
    {
        case 1:
            Dpwm0Regs.DPWMINT.bit.PRD_INT_EN = 0; // disable interrupt
            temp = FaultMuxRegs.FAULTMUXINTSTAT.all; // clear status

            handle_current_averaging_config();
            Dpwm0Regs.DPWMINT.bit.PRD_INT_EN = 1;
            cs_flag=2;
            break;

        case 2:
            Dpwm0Regs.DPWMINT.bit.PRD_INT_EN = 0; // disable interrupt
            temp = FaultMuxRegs.FAULTMUXINTSTAT.all; // clear status

            handle_current_averaging();
            current_x16 = current_x16 /256; //256 cycles sum, so need to be divided by 256
            current_x16 = current_x16 >> 3; //8 times oversampl every cycle, so need to be divided by 8

            cs_flag=3;
            Dpwm0Regs.DPWMINT.bit.PRD_INT_EN = 1;
            break;

        case 3:
            Dpwm0Regs.DPWMINT.bit.PRD_INT_EN = 0; // disable interrupt
            temp = FaultMuxRegs.FAULTMUXINTSTAT.all; // clear status
            cs_flag=1;
            cc_flag=5;

            handle_current_averaging();
            current_x16 = current_x16 /256; //256 cycles sum, so need to be divided by 256
            current_x16 = current_x16 >> 3; //8 times oversampl every cycle, so need to be divided by 8
            break;
    }
}
```

3. 配置函数 handle_current_averaging_config()

该函数主要完成 EADC1 与 Filter1 的连接配置、EADC 的基本配置（包括 DAC_VALUE 的写入，AFE_GAIN 的配置，Averaging 的配置等）、Oversample 的配置及 Filter 的配置。

```

void handle_current_averaging_config(void)
{
LoopMuxRegs.FILTERMUX.bit.FILTER1_FE_SEL=1;
LoopMuxRegs.SAMPTRIGCTRL.bit.FE1_TRIG_DPWMO_EN=1;
LoopMuxRegs.GLBEN.bit.FE_CTRL1_EN=1; // must enable FE1

/* The below code is used to config EADC */
FeCtrl1Regs.EADC_DAC.bit.DAC_DITHER_EN=0;
FeCtrl1Regs.EADC_DAC.bit.DAC_VALUE=1024; //reference value
FeCtrl1Regs.EADC_CTRL.bit.AFE_GAIN=1; //EADC AFE Gain is 2;
FeCtrl1Regs.EADC_CTRL.bit.AVG_MODE_SEL=2; //averaging is 8X
FeCtrl1Regs.EADC_CTRL.bit.EADC_MODE=1;
FeCtrl1Regs.EADC_CTRL.bit.AVG_SPATIAL_EN=0;
FeCtrl1Regs.EADC_CTRL.bit.SCFE_GAIN_FILTER_SEL=1;
FeCtrl1Regs.EADC_CTRL.bit.SCFE_CLK_DIV_2=0;
FeCtrl1Regs.EADC_CTRL.bit.SCFE_ENA=1;
FeCtrl1Regs.EADC_CTRL.bit.EADC_ENA=1;

Dpwm0Regs.DPWMCTRL2.bit.SAMPLE_TRIG_1_EN=1;
Dpwm0Regs.DPWMCTRL2.bit.SAMPLE_TRIG1_OVERSAMPLE=3; //oversample is 8
Dpwm0Regs.DPWMCTRL2.bit.SAMPLE_TRIG1_MODE=0;
Dpwm0Regs.DPWMSAMPTRIG1.all=20000-2000;

/* The below code is used to config Filter */
Filter1Regs.FILTERKPCOEF0.bit.KP_COEF_0=0;
Filter1Regs.FILTERKICOEF0.bit.KI_COEF_0=1; // only enable I
Filter1Regs.FILTERKDCOEF0.bit.KD_COEF_0=0;
Filter1Regs.FILTERKICLPHI.bit.KI_CLAMP_HIGH=0x7FFFFFFF;
Filter1Regs.FILTERKICLPLO.bit.KI_CLAMP_LOW=0x800002;
Filter1Regs.FILTEROCLPHI.bit.OUTPUT_CLAMP_HIGH=0x7FFFFFFF;
Filter1Regs.FILTEROCLPLO.bit.OUTPUT_CLAMP_LOW=0x800002;
Filter1Regs.FILTERPRESET.all = (1 << 27) + (1 << 24)+0;
Filter1Regs.FILTERCTRL.bit.FILTER_EN=1;
}

```

4. 配置函数 handle_current_averaging()

该函数主要完成 KI_YN 寄存器中数据的读取，Filter 的复位（需要对 KI_YN 寄存器清零，不然该寄存器会溢出）及 Filter 的使能等。

```

void handle_current_averaging(void)
{
//Read the output value of integrator(Filter1)
current_x16 = Filter1Regs.FILTERKIYNREAD.bit.KI_YN;
// Disable Filter1
Filter1Regs.FILTERCTRL.bit.FILTER_EN = 0;
//Enable Filter1 Jamming and write zero to the output of integrator (Reset integration for averaging)
Filter1Regs.FILTERPRESET.all = (1 << 27) + (1 << 24);
//Re-enable the filter
Filter1Regs.FILTERCTRL.bit.FILTER_EN = 1;
}

```

4 实测单板输入电流

4.1 测试单板概述

在一款基于 UCD3138 的硬开关全桥 EVM 板（UCD3138HSFBEVM-029）上进行输入电流的实际测试。该单板的关键技术规格如下：

- 输入电压：36V~72V
- 输出功率：12V×30A
- 功率拓扑：单级硬开关全桥
- 电流互感器：如图 2 所示，T1 的匝比为 100:1，Rs 为 10 ohm。

为实现 EADC1 和 Filter1 读取和计算输入电流，需要将电流互感器副边侧的输出连接到 EAP1 和 EAN1。单板其余部分保留原有设计。

4.2 实测数据

1. 输出电流设定为 3A，输入电压设定为 50V

实测电流互感器的输出信号如图 12（左），其平均值为 89.26mV。此时输入电流为 850mA，二者存在近似比例关系。比例系数主要由互感器的匝比与采样电阻决定： $0.01 \times 10 = 0.1$ 。

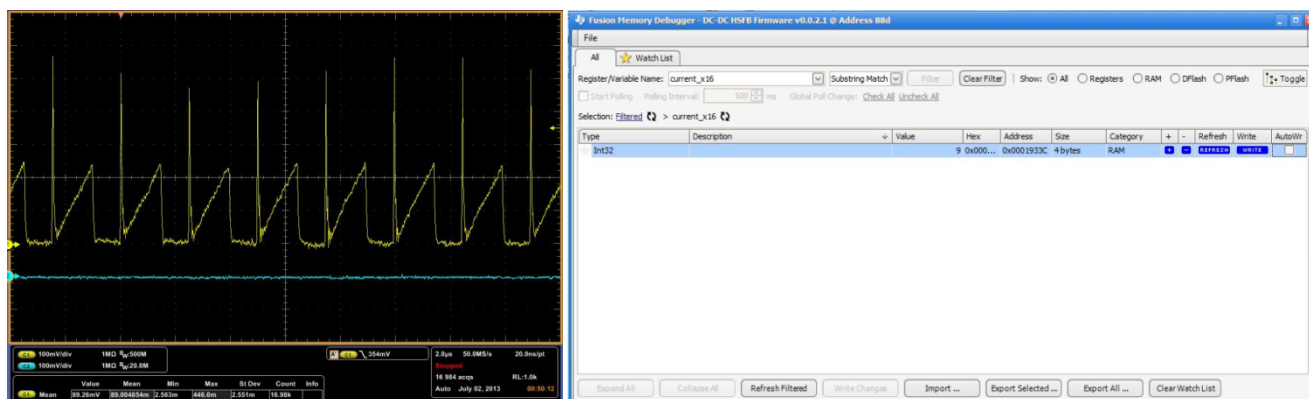


Figure 12. 实测波形及实际读取数据 1

将 DAC_VALUE 设定为 1024，通过软件计算后，读取到的 current_x16 变量（该变量含义参考 3.4 小节）值为 9，如图 12（右）所示。因此，计算出的平均值为：

$$(1024 \div 16) \times 1.5625\text{mV} - 9 = 91\text{mV}$$

该值与实测值的误差小于 5%，与实际输入电流的误差小于 7%。

2. 输出电流设定为 3A，输入电压设定为 55V

实测电流互感器的输出信号如图 13（左），其平均值为 82.48mV。此时输入电流为 780mA。

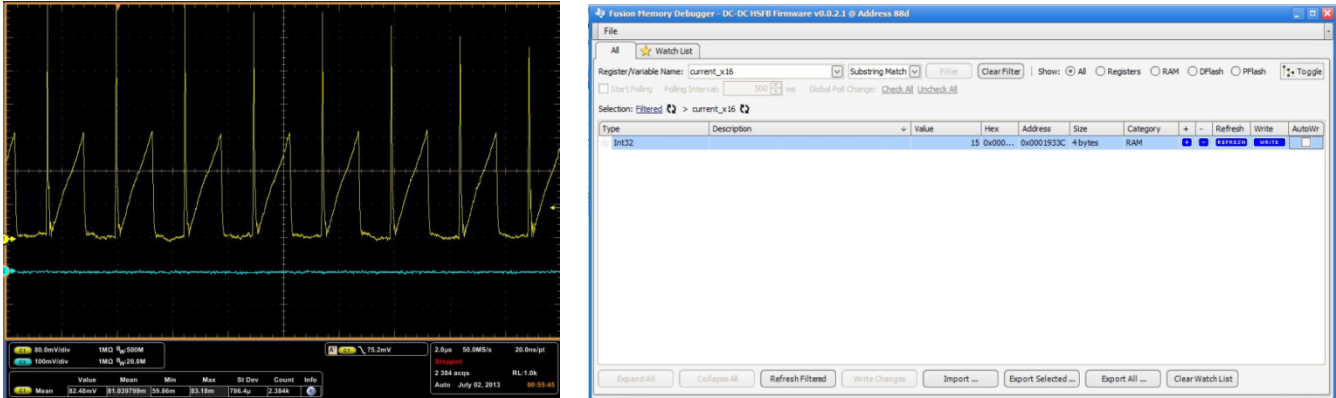


Figure 13. 实测波形及实际数据读取 2

将 DAC_VALUE 设定为 1024，通过软件计算后，读取到的 current_x16 变量值为 15，如图 13（右）所示。因此，计算出的平均值为：

$$(1024 \div 16) \times 1.5625\text{mV} - 13 = 87\text{mV}$$

该值与实测值的误差小于 5%，与实际输入电流的误差小于 12%。

5 总结

通过上文描述可知，在对 UCD3138 的 EADC 和 Filter 进行相应配置后，可以完成对 EAP/EAN 引脚输入信号平均值的读取，而且读取值与实际值的误差较小。

同样，该功能可以应用于单板输入电流的读取，实测结果亦证实了这一点。受限于轻载条件下实际输入电流与电流互感器的输出存在较大误差，因此，软件读取值与实际输入电流存有一定误差。

6 参考文献

1. UCD3138 datasheet, Texas Instruments Inc., 2011
2. UCD31xx Fusion Digital Power Peripherals Programmer's Manual, Texas Instruments Inc., 2011
3. UCD31xx Miscellaneous Analog Control _MAC_, Texas Instruments Inc., 2011
4. Using the UCD3138HSFBEVM-029 User's Guide, Texas Instruments Inc., 2013
5. Application note: 数字电源控制器 UCD3138 的数字比较器与模数转换器的应用说明, Texas Instruments Inc., 2013

重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予的直接或隐含权作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独力负责满足与其产品及其应用中使用的 TI 产品相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要求, TI 不承担任何责任。

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com.cn/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com.cn/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP应用处理器	www.ti.com.cn/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity	德州仪器在线技术支持社区	www.deyisupport.com

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122
Copyright © 2013 德州仪器 半导体技术 (上海) 有限公司