

# 深入理解 AWRL6432 EVM 的功耗测量



软硬件平台及版本

本文内容基于以下软硬件平台：

硬件平台：AWRL6432BOOST EVM <https://www.ti.com/tool/AWRL6432BOOST>

软件平台：[MMWAVE\\_L\\_SDK\\_05\\_05\\_00\\_02](https://www.ti.com/tool/MMWAVE_L_SDK_05_05_00_02) [MMWAVE\\_L\\_SDK\\_05\\_05\\_00\\_02](https://www.ti.com/tool/MMWAVE_L_SDK_05_05_00_02) SDK

## 引言

AWRL6432 是一款集成的单芯片毫米波雷达传感器，专为汽车应用设计，能够在 57GHz 至 64GHz 频段内运行。该芯片适用于车内感应、儿童检测、占位检测等应用场景。AWRL6432 在功耗管理方面具有显著的优势。它支持多种低功耗模式，包括空闲模式和深度睡眠模式，以降低能源消耗。此外，AWRL6432 的电源设计可以使用成本优化模式或者低功耗模式，成本优化模式可以有效降低系统成本，低功耗模式在不同的应用场景下可根据需求使用一条或多条电源轨来优化系统性能和功耗。INA228 是一款高精度的 20 位数字功耗监控器，专为电流和电压测量而设计，AWRL6432BOOST EVM 上正是使用了这一款数字功耗监控器对 AWRL6432 的功耗进行监测。

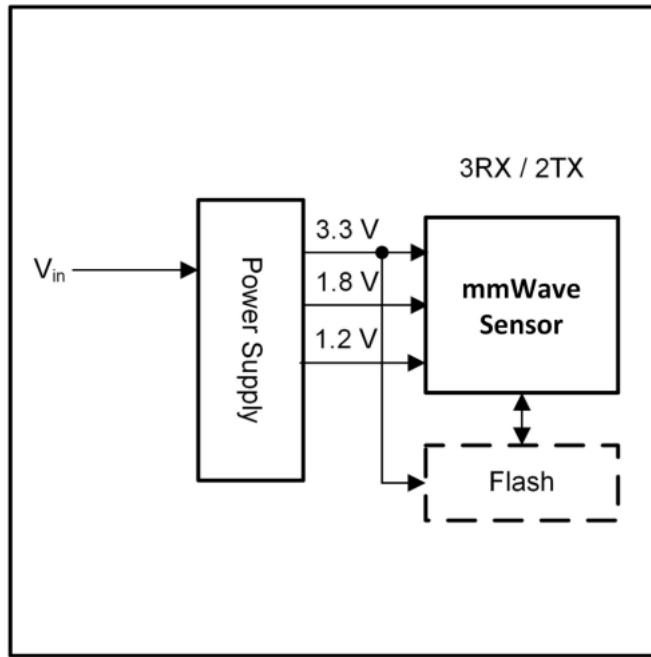
## AWRL6432BOOST EVM 的供电模式

AWRL6432 支持两种供电模式，包括成本优化和功耗优化模式。这两个模式决定了如何从外部电源模块向设备提供电源，更多信息请参考 AWRL6432 数据手册。成本优化模式更加节省成本，而功耗优化模式有更好的功耗表现，AWRL6432BOOST EVM 的电源设计采用功耗优化模式。

在功耗优化模式下，设备可以采用以下两种方式之一供电：

双电源轨（1.8V 和 1.2V）：使用一个 1.8V 和一个 1.2V 的稳压器为设备供电，或者三电源轨（3.3V、1.8V 和 1.2V）：使用一个 3.3V、一个 1.8V 和一个 1.2V 的稳压器为设备供电。在功耗优化模式下，1.2V 的电源轨不是由设备内部生成的，而是需要从外部提供。这种模式旨在优化功耗消耗，适用于对低功耗有严格要求的应用场景。

AWRL6432BOOST EVM 采用三电源轨模式，如下图所示：



由于 3.3V 电源轨同时给 Flash 供电，且供给雷达传感器的 3.3V 数字电源轨对整体功耗表现影响较小，在 AWRL6432 运动与存在检测中，仅对 1.8V 和 1.2V 两条电源轨做了功耗测量。三条电源轨供应的设备模块和相应引脚如下图所示：

电源轨	电源供应的设备模块	设备中相关的电源引脚
3.3V	数字I/O	输出: VIOIN
1.8V	合成器和数字锁相环晶振, 中频放大器, ADC	输入: VDDA_18VCO, VIOIN_18CLK, VDDA_18BB, VIOIN_18, VIN_18PMLDO 输出: VOUT_14SYNTH, VOUT_14APLL
1.2V	数字内核, 静态随机存储器, 射频等	输入: VDD, VDD_SRAM, VNWA, VDDA_12RF, LDO 输出: VDDA_10RF

## INA228 的电流测量

INA228 是一款超精密数字功耗监控器，配备专为电流检测应用而设计的 20 位  $\Delta$ - $\Sigma$  ADC。AWRL6432BOOST EVM 采用三颗 INA228 分别对 AWRL6432 的 3.3V, 1.8V, 1.2V 三条电源轨进行功耗测量。

ADC\_CONFIG 寄存器是 INA228 设备中用于配置模数转换器 (ADC) 行为的关键寄存器，AWRL6432 的功耗测量主要与这个寄存器相关。它允许用户设置测量模式、转换时间和采样平均值等参数。以下是 ADC\_CONFIG 寄存器的详细介绍。

1. 地址 ADC\_CONFIG 寄存器的地址是 1h。
2. 寄存器字段说明

位范围	字段	类型	默认值	说明
15-12	MODE	R/W	Fh	控制 ADC 的测量模式。支持连续模式和触发模式的不同组合。
11-9	VBUSCT	R/W	5h	设置总线电压测量的转换时间。
8-6	VSHCT	R/W	5h	设置分流电压测量的转换时间。
5-3	VTCT	R/W	5h	设置温度测量的转换时间。
2-0	AVG	R/W	0h	选择 ADC 样片的平均计数，影响有效输入信号的平均值计算。

- MODE (位 15-12)**：定义了 ADC 的测量模式，不同的设置允许用户选择测量总线电压、分流电压、温度，或者它们的组合，以及是否为连续模式或触发模式。
- VBUSCT (位 11-9)**：用于设置总线电压测量的转换时间，转换时间可以从 50 微秒到 4.12 毫秒，转换时间越长测量越稳定。
- VSHCT (位 8-6)**：用于设置分流电压测量的转换时间，可以从 50 微秒到 4.12 毫秒不等。
- VTCT (位 5-3)**：用于设置温度测量的转换时间，范围也是从 50 微秒到 4.12 毫秒。
- AVG (位 2-0)**：用于选择 ADC 样片的采样数，采样数可以是 1、4、16、64、128、256、512 或 1024。

### 运动与存在检测例程的 INA228 寄存器设置

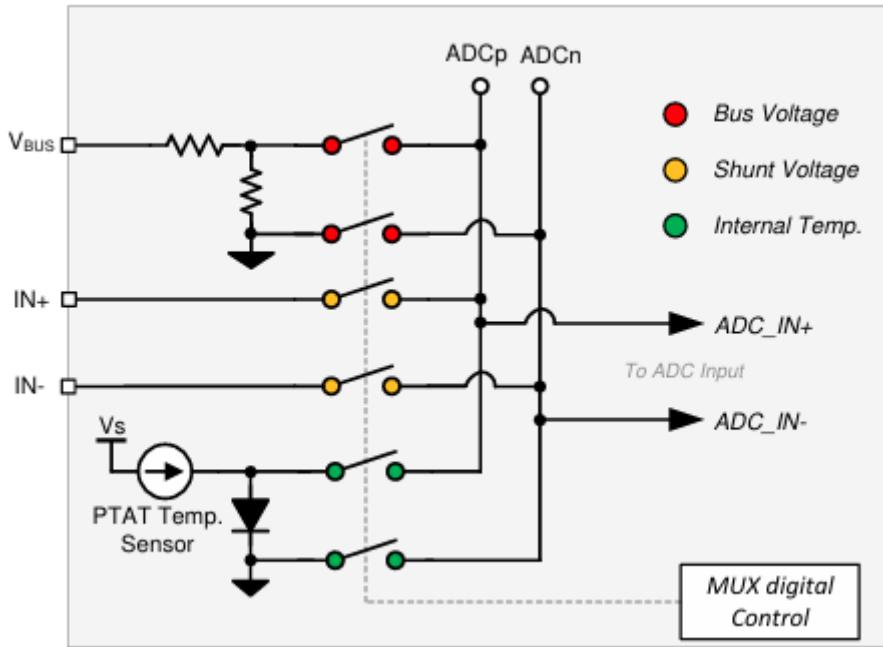
运行在 AWRL6432BOOST EVM 上的运动与存在检测例程 ( motion and presence detection ) 利用了 EVM 上的四个 INA228 进行电流测量。该例程对 INA228 的 ADC\_CONFIG 的设置如下：

```
void SensorConfig(I2C_HandleTypeDef i2cHandle)
{
    // Configure the INA228
    SensorConfig228(i2cHandle, INA_CONFIG0_TARGET_ADDRESS, INA_CONFIG0_REG0_VALUE, INA_CONFIG0_REG1_VALUE, INA_CONFIG0_SHUNT_TEMP_REG_VALUE, INA_CONFIG0_C
#define INA_CONFIG0_REG1_VALUE 65535
Expands to:
65535
```

由程序可得，ADC\_CONFIG 被设置为 0b{1111 111 111 111 111}即：

Mode [15:12]	VBUSCT [11:9]	VSHCT [8:6]	VTCT [5:3]	AVG [2:0]
0xF	0x7	0x7	0x7	0x7
连续总线电压、 分流电压和温度	总线电压 转换时间 4.12ms	分流电压 转换时间 4.12ms	温度测量转 换时间 4.12ms	ADC采样 平均数 x1024

注意，分流电压、总线电压和温度测量值在内部多路复用到单个 ADC，所以分流电压，总线电压和温度测量只能依次进行采集。如下图所示：



因此，基于 AWRL6432 运动与存在检测例程的设定，一次功耗测量的时间为：

$$4.12ms * 1024 * 3 = 12.6s \quad (1)$$

### AWRL6432 运动与存在检测例程的功耗计算

INA228 计算出的电流值可以通过 CURRENT 寄存器读取。CURRENT 寄存器中的值以 CURRENT\_LSB 为单位，可以通过以下公式转换为实际电流值：

$$Current(A) = CURRENT_LSB * CURRENT \quad (2)$$

其中，CURRENT 是 CURRENT 寄存器中读取的值，CURRENT\_LSB 是 CURRENT 寄存器的最小步长值：

$$CURRENT_LSB = \frac{\text{最大预期电流} A}{2^{19}} \quad (3)$$

AWRL6432 运动与存在检测例程的功耗计算部分代码如图所示：

```
void mmwDemo_PowerMeasurement(I2C_HandleTypeDef i2cHandle, uint16_t *ptrPwrMeasured)
{
    float current;
    current = currentRead228(i2cHandle, 64, INA_CONFIG0_CURRENT_LSB);
    ptrPwrMeasured[0] = (current == (float)0xFFFFFFFF) ? 0xFFFF : (uint16_t) round(current * 18000.0); //Rail 1.8V, current reading in mA, power measured in mW

    current = currentRead228(i2cHandle, 65, INA_CONFIG1_CURRENT_LSB);
    ptrPwrMeasured[1] = (current == (float)0xFFFFFFFF) ? 0xFFFF : (uint16_t) round(current * 12000.0); //Rail 1.2V, current reading in mA, power measured in mW

    current = currentRead228(i2cHandle, 68, INA_CONFIG2_CURRENT_LSB);
    ptrPwrMeasured[2] = (current == (float)0xFFFFFFFF) ? 0xFFFF : (uint16_t) round(current * 12000.0); //Rail RF 1.2V, current reading in mA, power measured in mW
}
```

读取到的数值以毫安为单位，通过和电压相乘，获得功耗值，其中 ptrPwrMeasured 以 100uW 为单位。最后的平均功耗为：

$$\text{平均功率} = (\text{ptrMeasured}[0] + \text{ptrMeasured}[1] + \text{ptrMeasured}[2]) * 0.1 \quad (4)$$

如果用户需要测量 3.3V 电源轨的功耗，可以增加语句：

```
current = currentRead228(i2cHandle, 69, INA_CONFIG2_CURRENT_LSB);
ptrPwrMeasured[2] = (current == (float)0xFFFFFFFF) ? 0xFFFF : (uint16_t) round(current * 33000.0);
下面是使用 SDK 中 MotionDetect.cfg 作为配置情况下获得的真实功耗数据:
119.0mw= (448+657+85)*0.1(非低功耗模式)
32.1mw= (97+140+84)*0.1(低功耗模式)
```

### 如何缩短功耗测量时间？

缩短功耗测量时间主要有两种方法，第一种方法是关闭其余的数据采集，第二种是缩短采样时间和采样点数。

对于第一种方法 INA228 使用以下公式计算电流：

$$\text{Current}(A) = \frac{VSENSE(V)}{RSHUNT(\Omega)} \quad (5)$$

其中，VSENSE 是分流电阻器上的电压降，RSHUNT 是分流电阻的阻值。因此，总线电压和温度可以不进行采集。根据 INA228 的寄存器说明，通过将 ADC\_CONFIG 寄存器中的 MODE(15:12)位配置为 0xCh{1100}可以仅对分流电压数据进行采集。一次功耗测量的时间为：

$$4.12ms * 1024 * 1 = 4.2s \quad (6)$$

对于第二种方法：可以通过减少 ADC 的采样时间和采用平均数来减少功耗测量时间。例如将 ADC\_CONFIG 设置为的值设置为 63780 {1111 1001 0010 0100}

注意：如果要修改 ADC\_CONFIG 的数值应该直接以硬编码形式写入，如下图所示：

```
void SensorConfig228(I2C_HandleTypeDef i2cHandle, uint8_t i2cTargetAddress, uint32_t config0Val, uint32_t config1Val,
{
    uint8_t          i2cTargetRegAddress;
    int32_t          retVal = 0;
    uint16_t         dieID = 0;

    i2cTargetRegAddress = INA228_DIE_ID_REG;
    retVal = i2cReadAddress(i2cHandle, i2cTargetAddress, i2cTargetRegAddress);
    dieID = rxData[0]<<8 | rxData[1];
    dieID = dieID >> 4;

    if(dieID == INA228_DIE_ID_REG_VALUE)
    {

        i2cTargetRegAddress = INA228_CONFIG_REG0;
        retVal = i2cWriteAddress(i2cHandle, i2cTargetAddress, i2cTargetRegAddress, config0Val);
        DebugP_assert(retVal == 0);
        retVal = i2cReadAddress(i2cHandle, i2cTargetAddress, i2cTargetRegAddress);
        DebugP_assert(retVal == 0);

        i2cTargetRegAddress = INA228_CONFIG_REG1;
        retVal = i2cWriteAddress(i2cHandle, i2cTargetAddress, i2cTargetRegAddress, 63780); 63780
        DebugP_assert(retVal == 0);
        retVal = i2cReadAddress(i2cHandle, i2cTargetAddress, i2cTargetRegAddress);
        DebugP_assert(retVal == 0);
    }
}
```

图中直接以常量 63780 替换了此处原本的参数值 config1Val，这是因为该参数值 config1Val 在 SensorConfig228 的传入值是一个宏定义，该宏定义由 TI 的 sysconfig 工具生成，如果通过修改相关宏的值进行修改会在再次编译后被 sysconfig 再次覆盖，目前 sysconfig 没有提供相关值的修改接口，为了避免被 sysconfig 覆写，可以硬编码形式写入该值。

此时，总线电压、分流电压、温度测量转换时间为 540us，平均采样数为 128. 此时一次功耗测量的时间为：

$$540\mu s * 128 * 3 = 4.2s = 207ms$$

(7)

其中需要注意的是，降低转换时间和采用平均数会降低结果的准确性和稳定性。由于在 AWRL6432 运动与存在检测例程中在一个帧周期结束后 **UART** 才会将相关数据发送，因此功耗测量时间低于帧周期将没有太大意义。

### 参考资料

- INA228 数据手册：<https://www.ti.com/lit/ds/symlink/ina228.pdf>
- AWRL6432 数据手册：<https://www.ti.com/lit/ds/symlink/awrl6432.pdf>

## 重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

版权所有 © 2025, 德州仪器 (TI) 公司