

# Analog Engineer's Circuit

## 具有 16 个单端通道和 I<sup>2</sup>C 接口的精密测量电路



Joseph Wu

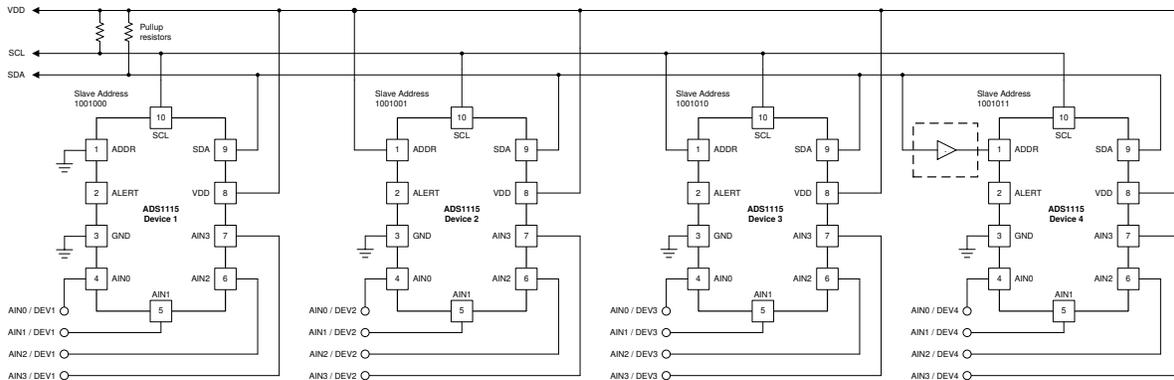
输入测量	ADC 输入	数字输出 ADS1115
PGA[2:0] = 000, FSR = ±6.144V	0V 至 5V	0 至 26667 0000h 至 682Bh

### 电源

VDD	GND
5V	0V

### 设计说明

许多应用要求在尽可能小的空间内进行大量测量。该电路介绍了一个 16 通道测量系统，该系统使用四个具有 I<sup>2</sup>C 接口的超小型 ADS1115 器件。这些器件具有精确的内部电压基准，并可针对多个输入范围进行编程，用作 15 位单端模数转换器 (ADC)。系统中的 ADC 高度可配置，可针对不同的数据速率和电压范围进行配置，甚至可用于警报功能。此外，输入可成对配置，从而构成一个具有差分输入测量的 8 通道测量系统。ADS1115 器件采用超小型 1.5mm × 2.0mm X2QFN 封装，在电路板上占用的空间非常小。该电路可用于 PLC 模拟输入模块、服务器板上的电源管理和各种通用测量等应用。



### 设计说明

1. ADS1115 器件具有差分 ADC，但该系统使用具有单端输入的器件。在此系统中，ADC 正 AIN<sub>P</sub> 输入连接至模拟输入端，而负 AIN<sub>N</sub> 输入在内部接地。
2. ADC 模拟输入的工作范围介于地和 VDD 之间，即使将满标量程 (FSR) 设置为大于 VDD 的电压也是如此。例如，当 FSR 设置为 ±6.144V 且 VDD 设置为 5V 时，所有模拟输入均限制为 0V 至 5V。大于电源电压的 FSR 可防止 ADC 在工作范围内进行任何测量时超出量程。
3. 有了内部基准，ADS1115 ADC 无需分压器即可利用内部基准方便地测量自身电源。
4. 对电源使用电源去耦电容器。必须使用连接到 GND 的至少为 0.1 μF 的电容器对 VDD 进行去耦。有关电源建议的详细信息，请参阅 [ADS111x 具有内部基准、振荡器和可编程比较器且兼容 I<sup>2</sup>C 的超小型、低功耗、860SPS、16 位 ADC 数据表](#)。
5. 如果可能，使用 C0G (NPO) 陶瓷电容器进行输入滤波。这些电容器中使用的电介质在电压、频率和温度变化时提供最稳定的电气性能。由于尺寸的原因，这并不总是实用，X7R 电容器是下一个最佳选择。

6. 连接到 SCK 的 ADDR 引脚会将器件的 I<sup>2</sup>C 地址设置为 1001011。使用此器件时，SDA 必须在 SCL 线路变为低电平后保持低电平至少 100ns，以确认器件正确解码地址。因此，原理图显示了一个将 SCL 线路连接到 ADDR 的延迟缓冲器。
7. 如果可以接受较低的分辨率，请在类似系统中设置 [ADS1015](#) ADC 或 [TLA2024](#) ADC。这些 12 位 ADC 可实现更快的数据速率和更低的成本。
8. [ADS1115](#) 器件使用一个 I<sup>2</sup>C 接口。如果需要 SPI 接口，[ADS1118](#) ADC 具有类似的功能。此外，如果可以使用分辨率较低的 ADC，则替换为 [ADS1018](#) ADC。有关更多详细信息，请参阅 [具有 8 个差分通道和 SPI 的精密测量电路](#) 电路指南。
9. 使用 8 个差分通道（而不是 16 个单端通道）或使用不同的差分 and 单端通道组合来构建此测量系统。该系统还具有可扩展性，并且器件和通道更少。
10. [ADS1115](#) ADC 用于单次转换模式。在这种运行模式下，每次转换都会启用器件，然后断电。单独使用所有四个器件的总功耗与在连续转换中使用单个器件的总功耗相同。

## 配置器件

1. 设置每个 [ADS1115](#) 器件的 I<sup>2</sup>C 地址。

此电路使用四个多路复用的 [ADS1115](#) ADC 来测量十六个不同的通道。为此，这四个器件必须使用不同的 I<sup>2</sup>C 地址，这些地址通过连接到每个器件的 ADDR 引脚来设置。ADDR 引脚连接到四个节点之一，以响应不同的从器件地址，如下表所示。

ADDR 引脚连接	从器件地址
GND	1001000
VDD	1001001
SDA	1001010
SCL	1001011

2. 使用多路复用器循环使用每个模拟输入。

[ADS1115](#) 器件具有一个多路复用器，用于测量多个通道。在此设计中，系统会循环使用每个器件的每个模拟输入。虽然可以进行差分测量，但图中显示，该系统通过循环使用所有模拟输入来对 GND 节点进行单端测量。

可通过下表所示的配置寄存器来配置器件。在配置寄存器中，MUX[2:0] 为 ADC 选择 AIN<sub>P</sub> 和 AIN<sub>N</sub> 节点，并设置每个器件的输入通道。表中的设置展示了单端测量的设置。如 [ADS111x 具有内部基准、振荡器和可编程比较器且兼容 I<sup>2</sup>C 的超小型、低功耗、860SPS、16 位 ADC](#) 数据表所示，可以使用 MUX[2:0] = 000 至 011 设置来进行差分测量。

输入测量	ADC 输入	器件输入选择
通道 1	MUX[2:0] = 100	AIN <sub>P</sub> = AIN <sub>0</sub> , AIN <sub>N</sub> = GND
通道 2	MUX[2:0] = 101	AIN <sub>P</sub> = AIN <sub>1</sub> , AIN <sub>N</sub> = GND
通道 3	MUX[2:0] = 110	AIN <sub>P</sub> = AIN <sub>2</sub> , AIN <sub>N</sub> = GND
通道 4	MUX[2:0] = 111	AIN <sub>P</sub> = AIN <sub>3</sub> , AIN <sub>N</sub> = GND

循环使用器件的全部四个通道后，系统会选择下一个器件并重复该循环。

3. 确定每个模拟输入测量的工作范围。

ADS1115 ADC 具有用于精确测量的内部基准和可扩展的测量增益。在本电路文档中，所示的测量值均为单端测量，负模拟输入接地。由于 ADS1115 器件是用于差分测量的 16 位 ADC，因此单端测量只有 15 位分辨率。

该 ADC 可设置为 FSR ( 满标量程 ) 的 6 种设置之一。这相当于在 ADC 前端安装了一个可编程增益放大器。配置寄存器的 FSR[2:0] 位的设置范围为 ±0.256V 至 ±6.144V。用于单端测量时，输入范围为 0V 至 0.256V 到 0V 至正 FSR 值。如前所述，即使 FSR 设置为大于 VDD 的电压，ADC 模拟输入的工作范围也介于地和 VDD 之间。例如，当 FSR 设置为 ±6.144V 且 VDD 设置为 5V 时，模拟输入限制为 0V 至 5V。

FSR 设置允许进行各种不同的测量，从温度传感器或电流采样电阻等小电压测量，到无需分压器即可测量自身电源的较大电压测量 ADC。

下表显示了 ADS1115 FSR 的设置。

4. 设置数据速率。

可根据器件内部振荡器将 ADS1115 器件配置为八种数据速率之一。此数据速率具有 ±10% 的容差。数据速率由 DR[2:0] 位设置，如下表所示。

数据速率设置	数据速率
DR[2:0] = 000	8SPS
DR[2:0] = 001	16SPS
DR[2:0] = 010	32SPS
DR[2:0] = 011	64SPS
DR[2:0] = 100	128SPS
DR[2:0] = 101	250SPS
DR[2:0] = 110	475SPS
DR[2:0] = 111	860SPS

所选数据速率为 128SPS，这是 ADS1115 器件的默认设置。该数据速率是可用的最快速率，对于所有 FSR 设置，可提供优于一个 LSB 的噪声性能。

### 配置寄存器设置

配置寄存器会设置 ADC 的操作模式和配置。配置包括前面各节中描述的所有设置。在 16 个位之间使用九位来配置器件。下表显示了配置寄存器字段描述以及位名称和位置、读写用法和复位值。

15	14	13	12	11	10	9	8
OS	MUX[2:0]			PGA[2:0]			模式
R/W-1h		R/W-0h			R/W-2h		R/W-1h
7	6	5	4	3	2	1	0
DR[2:0]			COMP_MODE	COMP_POL	COMP_LAT	COMP_QUE	
R/W-4h			R/W-0h	R/W-0h	R/W-0h	R/W-3h	

OS 会设置操作状态并启动单次转换。MUX[2:0] 位将输入多路复用器设置为循环执行不同的测量。MODE 位会将器件设置为单次转换模式。DR[2:0] 位会设置器件的数据速率。其余字段用于本设计中未使用的 ADC 比较器设置。有关配置寄存器的更多详细信息，请参阅 [ADS111x 具有内部基准、振荡器和可编程比较器且兼容 I<sup>2</sup>C 的超小型、低功耗、860SPS、16 位 ADC 数据表](#)。

例如，一个 ADC 设置为使用连接到 VDD 的 AIN0 测量其自身的接地电源。FSR 设为  $\pm 6.144V$ ，数据速率为 128SPS。配置寄存器字段的设置如下表所示。

位	字段	设置	说明
15	OS	1	开始转换
14:12	MUX[2:0]	100	单端输入测量，AINP-AINN = AIN0-GND，选择第一通道
11:9	PGA[2:0]	000	FSR = $\pm 6.144V$ ，将 ADC 设置为能够测量 0V 至 VDD 的整个电源电压范围
9	模式	1	单次转换模式下的运行
7:5	DR[2:0]	100	数据速率 = 128SPS
4	COMP_MODE	0	传统比较器
3	COMP_POL	0	低电平有效
2	COMP_LAT	0	非锁存比较器
1:0	COMP_QUE[1:0]	11	比较器已禁用

结合字段描述中的这些位，配置寄存器值为 1100 0001 1000 0011 或 C183h。

## 通道循环

若要循环使用该系统的每个通道，需逐一启动每个通道的转换过程，等待转换完成，然后读回数据。然后，开始下一个通道的转换过程。在移动到下一个 ADC 之前，对四个单端输入通道重复每次测量。对系统中的所有四个 ADC 重复此序列，即可循环使用所有通道。

写入配置寄存器会启动转换并将 ADC 配置为正确的运行模式。通信从写入器件的 I<sup>2</sup>C 从器件地址开始。I<sup>2</sup>C 写入后跟三个字节。第一个字节是 01h，用以指示配置寄存器。接下来的两个字节是写入配置寄存器的数据。

对器件配置寄存器的写入从对所选 I<sup>2</sup>C 地址 (1001000) 的写入开始。下一个字节是地址指针，指示配置寄存器 01h。写入操作通过将两个字节的数据写入配置寄存器来完成。四个字节的完整通信如下表所示。

I <sup>2</sup> C 地址： 1001000 写入	地址指针： 配置寄存器	配置 MSB：开始转换、设置输入、FSR、单次转换模式	配置 LSB：128SPS， 比较器已禁用
1001 0000	0000 0001	1100 0001	1000 0011

然后，主器件等待转换完成。在本示例中，ADS1115 器件设置为默认数据速率 128SPS。该器件使用内部振荡器，因此数据速率存在一些变化。为确认在 ADC 完成转换后读取器件，微控制器会等待转换完成所需的最长时间。该等待时间是标称数据周期加上 10%（用以补偿器件的内部振荡器变化）。对于每个单次转换，ADC 唤醒时间都额外增加 20 $\mu$ s。使用以下公式计算总等待时间。

$$\text{Wait time} = \text{nominal data period} + 10\% + 20\mu\text{s}$$

例如，如果器件以 128SPS 运行，则标称数据周期为 7.82ms。所需的等待时间为：

$$\text{Wait time} = (7.82\text{ms} \times 1.1) + 20\mu\text{s} = 8.62\text{ms}$$

读取器件从写入转换寄存器 (00h) 的寄存器指针开始，然后从同一 I<sup>2</sup>C 地址再次读取两个字节。五个字节的完整通信如下表所示。

I <sup>2</sup> C 地址： 1001000 写入	地址指针： 配置寄存器	I <sup>2</sup> C 地址： 1001000 读取	读取转换数据 MSB	读取转换数据 LSB
1001 0000	0000 0000	1001 0001	XXXX XXXX	XXXX XXXX

## 测量转换

输入电压的转换基于 ADC 的满标量程 (FSR) 设置。FSR 由配置寄存器中的 PGA[2:0] 位进行设置。

$$\text{Output Code} = 2^{15} \times [V_{\text{AINx}} \div (\text{positive FSR})]$$

$$\text{Input Voltage} = V_{\text{AINx}} = (\text{Output Code}) \times (\text{positive FSR}) \div (2^{15})$$

即使 ADC 用于单端测量，ADS1115 器件也是差分 ADC。如果 ADC 具有负偏移，并且 ADC 在输入通道上测得 0V，则 ADC 会报告一个负数。用二进制补码表示法报告负读数。例如，十进制 -1 在转换寄存器中报告为 FFFFh。

如前所述，16 位 ADS1115 器件替换为 12 位 ADS1015 ADC 或 TLA2024 ADC。对于这些器件，数据格式为 12 位，右侧填充了四位零。满标量程读数为 7FFh，而从转换寄存器中读取时为 7FF0h。

## 伪代码示例

下面显示了伪代码序列以及从四个器件的每个通道设置转换并在每次转换后收集数据所需的步骤。它还包括连接到 ADC 的微控制器的设置。

对于每个通道，微控制器同时设置 ADC 配置并启动转换。微控制器会等待转换完成，等待足够的时间以完成转换以及内部振荡器频率的任何变化。然后从转换寄存器中读取数据。继续循环使用每个器件的每个通道。该设置假设采用之前的配置，转换等待时间的数据速率设置为 128SPS。

```

Configure microcontroller for I2C mode
Loop
{
  Conversions from four channels of device 1:
  {
    write I2C addr 1001000, send 0x01 0xC1 0x83; // start conversion for device 1, AIN0-GND
    wait 8.62ms
    write I2C addr 1001000, send 0x00, read I2C addr 1001000, read two bytes; // Read conversion
    write I2C addr 1001000, send 0x01 0xD1 0x83; // start conversion for device 1, AIN1-GND
    wait 8.62ms
    write I2C addr 1001000, send 0x00, read I2C addr 1001000, read two bytes; // Read conversion
    write I2C addr 1001000, send 0x01 0xE1 0x83; // start conversion for device 1, AIN2-GND
    wait 8.62ms
    write I2C addr 1001000, send 0x00, read I2C addr 1001000, read two bytes; // Read conversion
    write I2C addr 1001000, send 0x01 0xF1 0x83; // start conversion for device 1, AIN3-GND
    wait 8.62ms
    write I2C addr 1001000, send 0x00, read I2C addr 1001000, read two bytes; // Read conversion
  }
  Conversions from four channels of device 2:
  {
    write I2C addr 1001001, send 0x01 0xC1 0x83; // start conversion for device 2, AIN0-GND
    wait 8.62ms
    write I2C addr 1001001, send 0x00, read I2C addr 1001001, read two bytes; // Read conversion
    write I2C addr 1001001, send 0x01 0xD1 0x83; // start conversion for device 2, AIN1-GND
    wait 8.62ms
    write I2C addr 1001001, send 0x00, read I2C addr 1001001, read two bytes; // Read conversion
    write I2C addr 1001001, send 0x01 0xE1 0x83; // start conversion for device 2, AIN2-GND
    wait 8.62ms
    write I2C addr 1001001, send 0x00, read I2C addr 1001001, read two bytes; // Read conversion
    write I2C addr 1001001, send 0x01 0xF1 0x83; // start conversion for device 2, AIN3-GND
    wait 8.62ms
    write I2C addr 1001001, send 0x00, read I2C addr 1001001, read two bytes; // Read conversion
  }
  Conversions from four channels of device 3:
  {
    write I2C addr 1001010, send 0x01 0xC1 0x83; // start conversion for device 3, AIN0-GND
    wait 8.62ms
    write I2C addr 1001010, send 0x00, read I2C addr 1001010, read two bytes; // Read conversion
    write I2C addr 1001010, send 0x01 0xD1 0x83; // start conversion for device 3, AIN1-GND
    wait 8.62ms
    write I2C addr 1001010, send 0x00, read I2C addr 1001010, read two bytes; // Read conversion
    write I2C addr 1001010, send 0x01 0xE1 0x83; // start conversion for device 3, AIN2-GND
    wait 8.62ms
    write I2C addr 1001010, send 0x00, read I2C addr 1001010, read two bytes; // Read conversion
    write I2C addr 1001010, send 0x01 0xF1 0x83; // start conversion for device 3, AIN3-GND
    wait 8.62ms
    write I2C addr 1001010, send 0x00, read I2C addr 1001010, read two bytes; // Read conversion
  }
  Conversions from four channels of device 4:
  {
    write I2C addr 1001011, send 0x01 0xC1 0x83; // start conversion for device 4, AIN0-GND
    wait 8.62ms
    write I2C addr 1001011, send 0x00, read I2C addr 1001011, read two bytes; // Read conversion
    write I2C addr 1001011, send 0x01 0xD1 0x83; // start conversion for device 4, AIN1-GND
    wait 8.62ms
    write I2C addr 1001011, send 0x00, read I2C addr 1001011, read two bytes; // Read conversion
    write I2C addr 1001011, send 0x01 0xE1 0x83; // start conversion for device 4, AIN2-GND
    wait 8.62ms
    write I2C addr 1001011, send 0x00, read I2C addr 1001011, read two bytes; // Read conversion
    write I2C addr 1001011, send 0x01 0xF1 0x83; // start conversion for device 4, AIN3-GND
    wait 8.62ms
    write I2C addr 1001011, send 0x00, read I2C addr 1001011, read two bytes; // Read conversion
  }
}
}

```

## I<sup>2</sup>C 事务

该器件通过写入配置寄存器来开始转换。这需要一个四字节的**事务**。字节 1 是地址写入，后跟一个字节的配置寄存器指针和两个字节的配置数据。

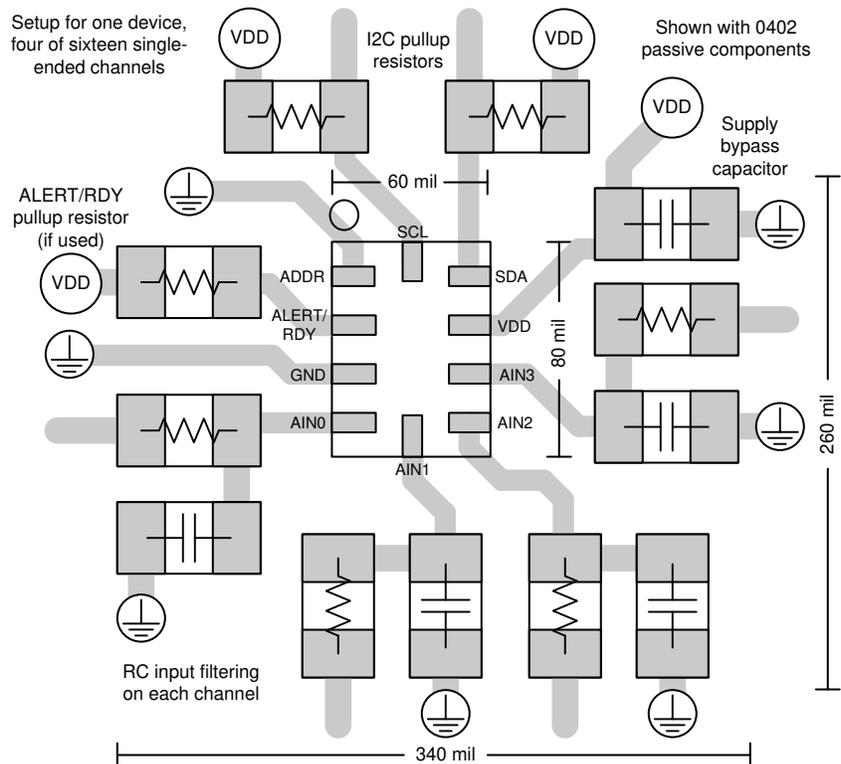
当器件读取转换时，需要一个五字节的事务。字节 1 是写入器件的地址，后跟字节 2 中的转换寄存器指针。字节 3 是从转换寄存器读取的地址，后跟两个字节的 **ADC** 数据。

器件	输入通道	命令	字节 1	字节 2	字节 3	字节 4	字节 5
器件 1	通道 1 AIN0 至 GND	开始转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1100 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0001 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 2 AIN1 至 GND	开始转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1101 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0001 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 3 AIN2 至 GND	开始转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1110 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0001 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 4 AIN3 至 GND	开始转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1111 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0000 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0001 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
器件 2	通道 5 AIN0 至 GND	开始转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1100 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0011 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 6 AIN1 至 GND	开始转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1101 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0011 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 7 AIN2 至 GND	开始转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1110 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0011 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 8 AIN3 至 GND	开始转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1111 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0010 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0011 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
器件 3	通道 9 AIN0 至 GND	开始转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1100 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0101 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 10 AIN1 至 GND	开始转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1101 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0101 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 11 AIN2 至 GND	开始转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1110 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0101 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 12 AIN3 至 GND	开始转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1111 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0100 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0101 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>

器件	输入通道	命令	字节 1	字节 2	字节 3	字节 4	字节 5
器件 4	通道 13 AIN0 至 GND	开始转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1100 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0111 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 14 AIN1 至 GND	开始转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1101 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0111 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 15 AIN2 至 GND	开始转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1110 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0111 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>
	通道 16 AIN3 至 GND	开始转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0001 (指向配置寄存器的指针)	1111 0001 (配置寄存器 MSB)	1000 0011 (配置寄存器 LSB)	-
		读取转换	1001 0110 (I <sup>2</sup> C 地址, 写入)	0000 0000 (指向转换寄存器的指针)	1001 0111 (I <sup>2</sup> C 地址, 读取)	<数据 MSB>	<数据 LSB>

### 布局示例

下面显示了四个 ADS1115 器件之一的示例布局。使用 0402 电阻器和电容器添加 RC 输入滤波。生成的布局约为 260mil × 340mil。此测量未考虑 I<sup>2</sup>C 上拉电阻器或 ALERT/RDY 上拉电阻器。每个系统都需要一套此类电阻器。



示例布局

**设计中采用的器件**

器件	主要特性	链接	其他可能的器件
<a href="#">ADS1115</a>	ADS111x 具有内部基准、振荡器和可编程比较器且兼容 I <sup>2</sup> C 的超小型、低功耗、860SPS、16 位 ADC	具有 <a href="#">PGA</a> 、 <a href="#">振荡器</a> 、 <a href="#">VREF</a> 、 <a href="#">比较器</a> 和 <a href="#">I2C</a> 的 <a href="#">16 位</a> 、 <a href="#">860SPS</a> 、 <a href="#">4 通道</a> $\Delta$ - $\Sigma$ <a href="#">ADC</a>	<a href="#">精密 ADC</a>

**商标**

所有商标均为其各自所有者的财产。

## 重要声明和免责声明

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 产品的应用。严禁对这些资源进行其他复制或展示。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  
Copyright © 2024，德州仪器 (TI) 公司