Precision measurement circuit with 16 singled-ended channels and I²C interface

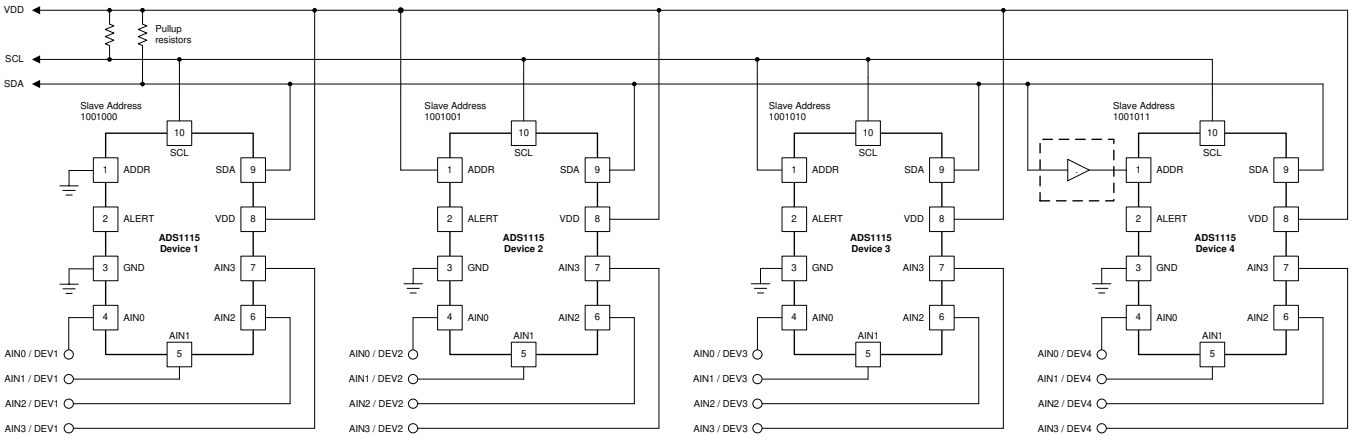
Joseph Wu

Input Measurement	ADC Input	Digital Output ADS1115
PGA[2:0] = 000, FSR = ±6.144V	0V to 5V	0 to 26667 0000h to 682Bh

Power Supplies	
VDD	**GND**
5V	0V

Design Description

Many applications require a large array of measurements using a minimal amount of space. This circuit describes a 16-channel measurement system using four ultra-small ADS1115 devices with an I²C interface. These devices have an accurate internal voltage reference, and can be programmed for multiple input ranges as 15-bit single-ended analog-to-digital converters (ADCs). The ADCs in the system are highly configurable for different data rates and voltage ranges and can even be used for alert functions. Additionally, inputs can be configured in pairs to make an 8-channel measurement system with differential input measurements. The ADS1115 device is available in an ultra-small 1.5mm × 2.0mm X2QFN package that takes up very little space on a board. This circuit can be used in applications such as *analog input modules* for PLCs, power management on server boards, and a variety of generic measurements.



Design Notes

1. The ADS1115 device has a differential ADC, but this system uses the device with single-ended inputs. In this system, the ADC positive AIN_P input is connected to the analog input, while the negative AIN_N input is internally connected to ground.
2. The operating range of the ADC analog inputs is between ground and VDD, even with the full-scale range (FSR) set to voltages larger than the VDD. For example, with the FSR set to ±6.144V and VDD set to 5V, all analog inputs are limited from 0V to 5V. An FSR larger than the supply prevents over-ranging the ADC for any measurement within the operating range.
3. With the internal reference, the ADS1115 ADC can conveniently measure its own supply using the internal reference without the need of a voltage divider.
4. Use supply decoupling capacitors for the power supplies. VDD must be decoupled with at least a 0.1-µF capacitor to GND. See the *ADS111x Ultra-Small, Low-Power, I²C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator* data sheet for details on power-supply recommendations.
5. When possible, use C0G (NPO) ceramic capacitors for input filtering. The dielectric used in these capacitors provides the most stable electrical properties over voltage, frequency, and temperature changes. Because of size, this may not always be practical and X7R capacitors are the next best alternative.
6. The ADDR pin connected to SCK sets the I²C address of the device to 1001011. When this device is used, the SDA must be held low for at least 100ns after the SCL line goes low to ensure the device decodes the address correctly. The schematic shows a delay buffer connecting the SCL line to ADDR for this reason.
7. If lower resolution is acceptable, the ADS1015 ADC or the TLA2024 ADC may be set up in a similar system. These 12-bit ADCs allow for a faster data rate at lower cost.
8. The ADS1115 device uses an I²C interface. If an SPI interface is required, the ADS1118 ADC has similar functions and may be used. Additionally, the ADS1018 ADC may be substituted if an ADC with less resolution can be used. See the *Precision measurement circuit with 8 differential channels and SPI* circuit guide for more details.
9. This measurement system may be constructed with 8 differential channels instead of 16 single-ended channels, or with different combinations of differential and single-ended channels. This system is also scalable with fewer devices and channels.
10. The ADS1115 ADC is used in single-shot conversion mode. In this mode of operation, devices are enabled for each conversion and then powered down. The total power of using all four devices individually is the same as using a single device in continuous conversion.

Configuring the Device

1. Set the I^2C address for each ADS1115 device.

   This circuit uses four ADS1115 ADCs multiplexed to measure sixteen different channels. To do this, the four devices must use different I^2C addresses, set by connections to the ADDR pin of each device. The ADDR pin connects to the one of four nodes to respond to a different slave address as shown in the following table.

ADDR Pin Connection	Slave Address
GND	1001000
VDD	1001001
SDA	1001010
SCL	1001011

2. Use the multiplexer to cycle through each analog input.

   The ADS1115 device has a multiplexer for measuring multiple channels. In this design, the system cycles through each of the analog inputs of each device. While differential measurements are available, this system is shown by cycling through all of the analog inputs to make single-ended measurements with respect to the GND node.

   Devices are configured through the configuration register shown in the following table. Within the configuration register, the MUX[2:0] selects the AIN_P and AIN_N nodes for the ADC and sets the input channel for each device. The settings in the table show settings for single-ended measurements. As the *ADS111x Ultra-Small, Low-Power, I^2C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator* data sheet shows, differential measurements can be made with settings MUX[2:0] = 000 to 011.

Input Measurement	ADC Input	Device Input Selection
Channel 1	MUX[2:0] = 100	AIN_P = AIN0 and AIN_N = GND
Channel 2	MUX[2:0] = 101	AIN_P = AIN1 and AIN_N = GND
Channel 3	MUX[2:0] = 110	AIN_P = AIN2 and AIN_N = GND
Channel 4	MUX[2:0] = 111	AIN_P = AIN3 and AIN_N = GND

   After cycling through all four channels of the device, the system selects the next device and repeats the cycle.

3. Identify the range of operation for each analog input measurement.

   The ADS1115 ADC has an internal reference for accurate measurements and a scalable gain for the measurement. In this circuit document, measurements shown are all single-ended with the negative analog input attached to ground. Because the ADS1115 device is a 16-bit ADC as a differential measurement, a single-ended measurement has only 15-bits of resolution.

   The ADC can be set to one of 6 settings for the FSR (full-scale range). This is the equivalent to having a programmable gain amplifier on the front end of the ADC. The FSR[2:0] bits of the configuration register have settings from ±0.256V up to ±6.144V. When used in a single-ended measurement, the input ranges from 0V to 0.256V up to 0V to the positive FSR value. As mentioned previously, the operating range of the ADC analog inputs is between ground and VDD, even with the FSR set to voltages larger than the VDD. For example, with the FSR set to ±6.144V and VDD set to 5V, the analog inputs are limited to 0 to 5V.

   The FSR settings allow for a variety of different measurements from small voltage measurements such as temperature sensors or current shunt resistors, to larger voltage measurements ADC measuring its own supply without the need for a voltage divider.

The settings for the ADS1115 FSR are shown in the following table.

Input Measurement	ADC Input	Digital Output ADS1115
PGA[2:0] = 000, FSR = ±6.144V	0V to 5V	0000h to 682Bh
PGA[2:0] = 001, FSR = ±4.096V	0V to 4.096V	0000h to 7FFFh
PGA[2:0] = 010, FSR = ±2.048V	0V to 2.048V	0000h to 7FFFh
PGA[2:0] = 011, FSR = ±1.024V	0V to 1.024V	0000h to 7FFFh
PGA[2:0] = 100, FSR = ±0.512V	0V to 0.512V	0000h to 7FFFh
PGA[2:0] = 101-111, FSR = ±0.256V	0V to 0.256V	0000h to 7FFFh

4. Set the data rate.

   The ADS1115 device can be configured to one of eight data rates, based on the internal oscillator of the device. This data rate has a tolerance of ±10%. The data rate is set by the DR[2:0] bits shown in the following table.

Data Rate Setting	Data Rate
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

   The data rate is selected to be 128SPS, which is the default setting of the ADS1115 device. This data rate is the fastest available that gives a noise performance better than one LSB for all FSR settings.

Configuration Register Settings

The configuration register sets the mode of operation and configuration of the ADC. Configurations include all of the settings described in the previous sections. Nine fields across 16 bits are used to configure the device. Configuration register field descriptions are shown with bit names and positions, read and write usage, and reset values in the following table.

15	14	13	12	11	10	9	8
OS	MUX[2:0]			PGA[2:0]			MODE
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 sets the operational status and starts a single conversion. The MUX[2:0] bits set the input multiplexer to cycle through different measurements. The MODE bit sets the device to single-shot conversion mode. The DR[2:0] bits set the data rate of the device. The remaining fields are used for the ADC comparator settings which are not used in this design. See the *ADS111x Ultra-Small, Low-Power, I²C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator* data sheet for details on the configuration register.

As an example, one ADC is set to measure its own supply to ground, using AIN0 connected to VDD. The FSR is set to ±6.144V, with a data rate of 128SPS. The settings for the configuration register fields are shown in the following table.

Bit	Field	Setting	Description
15	OS	1	Start conversion
14:12	MUX[2:0]	100	Single-ended input measurement, AINP-AINN = AIN0-GND, selection of the first channel
11:9	PGA[2:0]	000	FSR = ±6.144V, sets the ADC to be able to measure the full supply range of 0V to VDD
9	MODE	1	Operation in single-shot conversion mode
7:5	DR[2:0]	100	Data rate = 128SPS
4	COMP_MODE	0	Traditional comparator
3	COMP_POL	0	Active low
2	COMP_LAT	0	Non-latching comparator
1:0	COMP_QUE[1:0]	11	Comparator is disabled

Combining these bits from the field descriptions, the configuration register values are 1100 0001 1000 0011 or C183h.

Channel Cycling

To cycle through each channel of the system, start each conversion, wait for the conversion to complete, and then read back the data. Then start the conversion for the next channel. Repeat each measurement for the four single-ended input channels, before moving on to the next ADC. Repeating this sequence for all four ADCs in the system cycles through all channels.

A write to the configuration register starts the conversion and configures the ADC for the proper mode of operation. The communication starts with a write to the I²C slave address of the device. The I2C write is followed by three bytes. The first byte is 01h to indicate the configuration register. The next two bytes are the data written to the configuration register.

A write to device configuration register starts with a write to the selected I²C address (1001000). The next byte is the address pointer which indicates the configuration register of 01h. The write is completed with the two bytes of data to be written to the configuration register. The complete communication of four bytes is shown in the following table.

I²C Address: 1001000 Write	Address Pointer: Configuration Register	Configuration MSB: Start Conversion, Set Input, FSR, Single-Shot Mode	Configuration LSB: 128SPS, Comparator Disabled
1001 0000	0000 0001	1100 0001	1000 0011

The master then waits for the conversion to complete. For this example, the ADS1115 device is set to the default data rate of 128SPS. Because the device uses an internal oscillator, there is some variation in the data rate. To ensure that the device is read after the ADC completes a conversion, the microcontroller waits for the maximum time required for the conversion to complete. This wait time is the nominal data period plus 10% (to compensate for the internal oscillator variation of the device). An additional 20µs is added for the wake up time of the ADC for each single-shot conversion. The total wait time is calculated in the following equation.

Wait time = nominal data period + 10% + 20µs

As an example, if the device is run at 128SPS, the nominal data period is 7.82ms. The necessary wait time would be:

Wait time = (7.82ms × 1.1) + 20µs = 8.62ms

A read from the device starts with a write to the register pointer for the conversion register (00h) and then another read of two bytes from the same I²C address. The complete communication of five bytes is shown in the following table.

I²C Address: 1001000 Write	Address Pointer: Configuration Register	I²C Address: 1001000 Read	Read Conversion Data MSB	Read Conversion Data LSB
1001 0000	0000 0000	1001 0001	xxxx xxxx	xxxx xxxx

Measurement Conversion

Conversions for the input voltage are based on the full-scale range (FSR) setting of the ADC. The FSR is set by the PGA[2:0] bits in the configuration register.

Output Code = 2^{15} × [V_{AINx} / (positive FSR)]

Input Voltage = V_{AINx} = (Output Code) × (positive FSR) / (2^{15})

Even though the ADC is being used for a single-ended measurement, the ADS1115 device is a differential ADC. If the ADC has a negative offset, and the ADC measures a 0V on the input channel, then the ADC may report a negative number. Negative readings are reported in two's complement notation. For example, –1 in decimal is reported as FFFFh in the conversion register.

As previously mentioned, the 16-bit ADS1115 device may be replaced with the 12-bit ADS1015 ADC or the TLA2024 ADC. For these devices, the data format is 12 bits with four bits zero padded on the right. A full scale reading is 7FFh and would be read from the conversion register as 7FF0h.

Pseudo Code Example

The following shows a pseudo-code sequence with the required steps to set up conversions from each channel of the four devices and collect the data after each conversion. It also includes the setup for the microcontroller that interfaces to the ADC.

For each channel, the microcontroller simultaneously sets up the ADC configuration and starts the conversion. The microcontroller waits for conversion to complete, waiting enough time for the conversion and any variation in the internal oscillator frequency. Then the data is read from the conversion register. The cycle continues through each channel of each device. The setup assumes the previous configuration with the data rate set to 128SPS for the wait time for the conversion.

```
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
  }
}
```

I²C Transactions

The device starts conversions with a write to the configuration register. This requires a four-byte transaction. Byte 1 is the address write, followed by a byte for the configuration register pointer, and two bytes of configuration data.

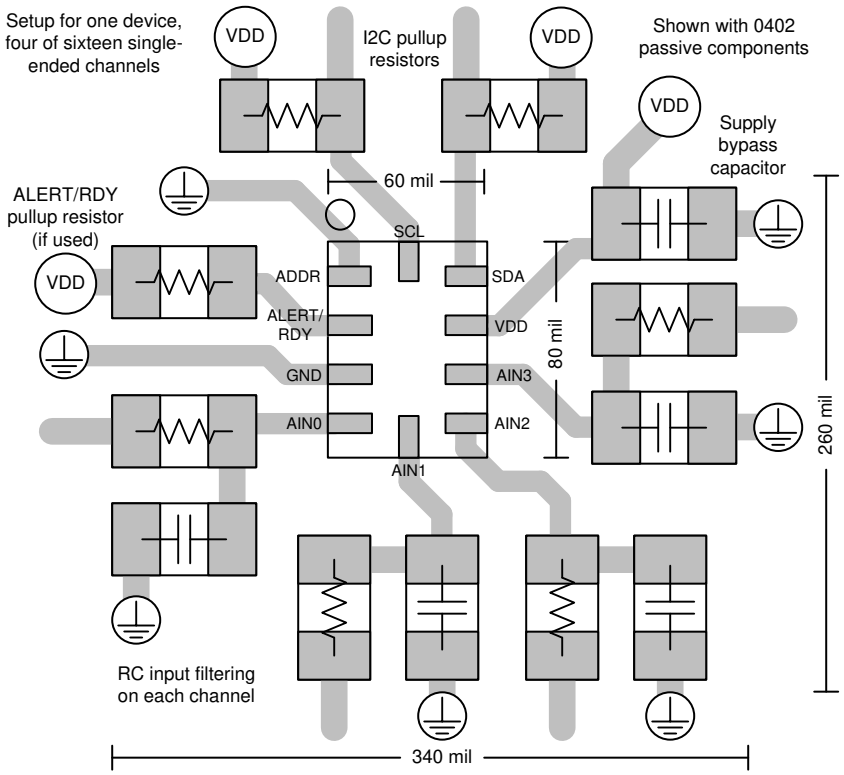
When the device reads the conversion, a five-byte transaction is required. Byte 1 is the address write to the device followed by the conversion register pointer in byte 2. Byte 3 is the address read from the conversion register followed by two bytes of ADC data.

Device	Input Channel	Command	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Device 1	Channel 1 AIN0 to GND	Start conversion	1001 0000 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1100 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0000 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0001 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 2 AIN1 to GND	Start conversion	1001 0000 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1101 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0000 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0001 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 3 AIN2 to GND	Start conversion	1001 0000 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1110 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0000 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0001 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 4 AIN3 to GND	Start conversion	1001 0000 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1111 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0000 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0001 (I^2C address, read)	<Data MSB>	<Data LSB>
Device 2	Channel 5 AIN0 to GND	Start conversion	1001 0010 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1100 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0010 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0011 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 6 AIN1 to GND	Start conversion	1001 0010 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1101 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0010 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0011 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 7 AIN2 to GND	Start conversion	1001 0010 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1110 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0010 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0011 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 8 AIN3 to GND	Start conversion	1001 0010 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1111 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0010 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0011 (I^2C address, read)	<Data MSB>	<Data LSB>

Device	Input Channel	Command	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Device 3	Channel 9 AIN0 to GND	Start conversion	1001 0100 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1100 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0100 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0101 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 10 AIN1 to GND	Start conversion	1001 0100 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1101 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0100 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0101 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 11 AIN2 to GND	Start conversion	1001 0100 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1110 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0100 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0101 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 12 AIN3 to GND	Start conversion	1001 0100 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1111 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0100 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0101 (I^2C address, read)	<Data MSB>	<Data LSB>
Device 4	Channel 13 AIN0 to GND	Start conversion	1001 0110 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1100 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0110 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0111 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 14 AIN1 to GND	Start conversion	1001 0110 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1101 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0110 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0111 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 15 AIN2 to GND	Start conversion	1001 0110 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1110 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0110 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0111 (I^2C address, read)	<Data MSB>	<Data LSB>
	Channel 16 AIN3 to GND	Start conversion	1001 0110 (I^2C address, write)	0000 0001 (Pointer to configuration reg.)	1111 0001 (Configuration reg. MSB)	1000 0011 (Configuration reg. LSB)	–
		Read conversion	1001 0110 (I^2C address, write)	0000 0000 (Pointer to conversion reg.)	1001 0111 (I^2C address, read)	<Data MSB>	<Data LSB>

Layout Example

The following shows an example layout of one of the four ADS1115 devices. RC input filtering is added using 0402 resistors and capacitors. The resulting layout is about 260 mils by 340 mils. This measurement does not include the I^2C pullup resistors or the ALERT/RDY pullup resistor. A single set of these resistors are required for each system.



Example Layout

Design Featured Devices

Device	Key Features	Link	Other Possible Devices
ADS1115	ADS111x ultra-small, low-power, I^2C-compatible, 860-SPS, 16-bit ADCs with internal reference, oscillator, and programmable comparator	http://www.ti.com/product/ADS1115	Link to similar devices Link to similar SPI devices

Design References

See the *Analog Engineer's Circuit Cookbooks* for TI's comprehensive circuit library.

For direct support from TI Engineers use the E2E community:

e2e.ti.com

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265