

---

# **Nanoshield\_ADC Documentation**

***Release 1.0***

**Nanoshield\_ADC**

December 21, 2015



<b>1</b>	<b>Class Documentation</b>	<b>3</b>
----------	----------------------------	----------



This is the Arduino software library to access the ADC Nanoshield (ADS1x15 IC) from Texas Instruments.

- Source code: [https://github.com/circuitar/Nanoshield\\_ADC](https://github.com/circuitar/Nanoshield_ADC)
- Documentation: <http://nanoshield-adc.readthedocs.org/>
- Reference boards: [ADC Nanoshield](#) and [4-20 ADC Nanoshield](#) from [Circuitar](#)

Library features include:

- Analog to digital conversion with 16 bits resolution.
- Continuous mode to capture measures with a given sample rate.
- Gain adjust to pair the signal magnitude with the resolution.
- Differential voltage measurements.
- High threshold setting to trigger interruptions.

To install, just click **Download ZIP** and install it using **Sketch > Include Library... > Add .ZIP Library** in the Arduino IDE.

This library is based on the [Adafruit\\_ADS1X15](#) library from Adafruit.

The following **examples** are provided:

- **Voltage**: read a voltage in single-ended mode.
- **VoltageWithHighGain**: read a voltage in single-ended mode with a high gain. Increasing the gain allows reading smaller voltages with increased resolution.
- **Voltage4Ch**: read four voltages in a single ADC Nanoshield.
- **Voltage16Ch**: using multiple boards. Read sixteen voltages across four units of the ADC Nanoshield. Make sure that the jumpers for the I2C address are properly configured to support this example.
- **DifferentialVoltage**: read a differential voltage between inputs 2 (+) and 3 (-). The channels can be easily changed to 0 (+) and 1 (-).
- **Read4to20mA**: how to read data from industrial 4-20mA sensors (or 4-20mA “transmistters”). Make sure to use the 4-20 ADC Nanoshield board for this example.
- **ContinuousMode**: read data in continuous mode to achieve high sample rates.
- **ContinuousMode4Ch**: read multiple channels in continuous mode.
- **Comparator**: trigger interruptions on arduino using the ADC Nanoshield in comparator mode.
- **RawAdcValue**: read the raw ADC value in single-ended mode.



---

## Class Documentation

---

class **Nanoshield\_ADC**

### Public Functions

**Nanoshield\_ADC** (uint8\_t *i2cAddress*)

Constructor.

Instantiates an object to control the ADC Nanoshield converter.

### See

*begin()*

### Parameters

- *i2cAddress* - The nanoshield I2C address on the bus.

void **begin** ()

Initializes the module.

uint16\_t **readRegister** (uint8\_t *i2cAddress*, uint8\_t *reg*)

Read an ADC Nanoshield register.

### Return

The value of the addressed register.

### Parameters

- *i2cAddress* - I2C address of the ADC Nanoshield.
- *reg* - Register address to read.

void **writeRegister** (uint8\_t *i2cAddress*, uint8\_t *reg*, uint16\_t *value*)

Writes to Nanoshield ADC config register.

### Parameters

- *i2cAddress* - I2C address of the Nanoshield ADC.
- *reg* - Config register address to write.
- *value* - Value to write.

bool **conversionDone** ()

Checks if there is data available to read.

**Return**

True if no conversion is being done and the value is ready to be read. Otherwise, false.

int16\_t **getLastConversionResults** ()

Reads the binary voltage requested when it is ready.

To use when in continuous mode. Same as *readNext()*.

**Return**

The binary voltage representation.

**See**

*readNext()*

int16\_t **readNext** ()

Reads the binary voltage requested when it is ready.

To use when in continuous mode.

**Return**

The binary voltage representation.

int16\_t **readADC\_SingleEnded** (uint8\_t *channel*)

Reads channel binary voltage.

**Return**

Binary representation of the voltage in the channel.

**Parameters**

- *channel* - Channel to read. Must be in 0, 1, 2 or 3.

int16\_t **readADC\_Differential\_0\_1** ()

Reads differential voltage in binary mode between channels 0(+) and 1(-).

Reads 16 or 12 bits representation of the difference between the voltage in channels 0(+) and 1(-).

**Return**

Binary representation of the differential voltage.

int16\_t **readADC\_Differential\_2\_3** ()

Reads differential voltage in binary mode between channels 2(+) and 3(-).

Reads 16 or 12 bits representation of the difference between the voltage in channels 2(+) and 3(-).

**Return**

Binary representation of the differential voltage.



void **setComparator** (uint8\_t *channel*, int16\_t *highThreshold*, int16\_t *lowThreshold*)

Sets the Nanoshield ADC in comparator mode.

In comparator mode, the Nanoshield sets the ALERT/RDY pin to LOW when the voltage read exceed a high threshold. This pin is set to HIGH again when the voltage read falls below a low threshold. The ALERT/RDY pin is connected to arduino D3 pin.

#### Parameters

- *channel* - The channel to compare.
- *highThreshold* - The high threshold.
- *lowThreshold* - The low threshold.

void **setNotComparator** ()

Turns comparator mode off.

Note that comparator mode needs the continuous mode. So when comparator mode is unset, continuous mode still set. Unset continuous mode too if necessary.

#### See

*setContinuous()*

bool **isComparator** ()

Checks if in comparator mode.

#### Return

True if comparator mode is active. False if not.

void **startComparator\_SingleEnded** (uint8\_t *channel*, int16\_t *highThreshold*, int16\_t *lowThreshold*)

Sets the high threshold value to comparator mode.

The ADS1115 has a interrupt signal connected to arduino ALERT/RDY pin (D3) to notify voltages over a high threshold. To use this feature, the alert jumper on arduino must be closed and a interrupt handler must be set with `attachInterrupt(6, <interruptHandler>, LOW)`, where `<interruptHandler>` is the function to be called on interrupt signal.

#### Parameters

- *channel* - The channel to keep track of value. Must be in 0, 1, 2 or 3.
- *highThreshold* - The high threshold set ALERT/RDY pin LOW.
- *lowThreshold* - The low threshold to set ALERT/RDY pin HIGH.

void **setContinuous** (bool *c*)

Sets the library to work on continuous mode.

In continuous mode, the library will reads the voltage value based on a sample rate. For example, if reading at 860Hz, then 860 measures will be done in a second.

#### See

*setSampleRate()*

*getSampleRate()*

*readNext()*

*conversionDone()*

#### Parameters

- `c` - True to activate continuous mode, false to deactivate continuous mode.

bool **isContinuous** ()

Checks if in continuous mode.

#### Return

True if in continuous mode, false if not.

void **setGain** (Gain\_t *gain*)

Sets the input gain.

The input gain is used to adjust the converter resolution to the signal magnitude. Greater gains adjust lower voltages over the resolution.

#### See

Gain\_t

#### Parameters

- `gain` - A available gain from the Gain\_t enumerator.

Gain\_t **getGain** ()

Gets the gain set on the Nanoshield ADC.

#### Return

A value from Gain\_t enumerator.

#### See

Gain\_t

*setGain()*

float **getRange** ()

[brief description]

[long description]

#### Return

[description]

virtual void **setSampleRate** (uint16\_t *sps*)

Sets the conversor sample rate on continuous mode.

#### See

setContinuous()

#### Parameters

- `sps` - Sample rate. Possible values are:
  - If using Nanoshield\_ADC12 (12 bits): 128, 250, 490, 920, 1600, 2400, 3300
  - If using *Nanoshield\_ADC* (16 bits): 8, 16, 32, 64, 128, 250, 475, 860 If no one of the above, the closest lower value will be selected.

virtual uint16\_t **getSampleRate** ()  
Gets actual *Nanoshield\_ADC* sample rate.

**Return**

Actual sample rate set.

float **read4to20mA** (uint8\_t *channel*)  
Reads channel current from 4mA to 20mA.

**Return**

The current on channel from 4mA to 20mA.

**Parameters**

- *channel* - Channel to read. Must be one of 0, 1, 2 or 3.

float **readVoltage** (uint8\_t *channel*)  
Reads channel voltage.

**Return**

The voltage read up to 5V.

**Parameters**

- *channel* - Channel to read. Must be one of 0, 1, 2 or 3.

float **readDifferentialVoltage01** ()  
Reads voltage in differential mode between channels 0(+) and 1(-).

Reads the difference between voltages in channel 0(+) and 1(-). For example, if the voltage in both channels is the same, then the differential voltage is zero. Notice that the voltage margin in channels still 0V to 5V in relation to GND.

**Return**

The difference between the voltages in channel 0(+) and 1(-).

float **readDifferentialVoltage23** ()  
Reads voltage in differential mode between channels 2(+) and 3(-).

Reads the difference between voltages in channel 2(+) and 3(-). For example, if the voltage in both channels is the same, then the differential voltage is zero. Notice that the voltage margin in channels still 0V to 5V in relation to GND.

**Return**

The difference between the voltages in channel 2(+) and 3(-).

---

This documentation was built using [ArduinoDocs](#).



## N

Nanoshield\_ADC (C++ class), 3  
Nanoshield\_ADC::begin (C++ function), 3  
Nanoshield\_ADC::conversionDone (C++ function), 3  
Nanoshield\_ADC::getGain (C++ function), 6  
Nanoshield\_ADC::getLastConversionResults (C++ function), 4  
Nanoshield\_ADC::getRange (C++ function), 6  
Nanoshield\_ADC::getSampleRate (C++ function), 7  
Nanoshield\_ADC::isComparator (C++ function), 5  
Nanoshield\_ADC::isContinuous (C++ function), 6  
Nanoshield\_ADC::Nanoshield\_ADC (C++ function), 3  
Nanoshield\_ADC::read4to20mA (C++ function), 7  
Nanoshield\_ADC::readADC\_Differential\_0\_1 (C++ function), 4  
Nanoshield\_ADC::readADC\_Differential\_2\_3 (C++ function), 4  
Nanoshield\_ADC::readADC\_SingleEnded (C++ function), 4  
Nanoshield\_ADC::readDifferentialVoltage01 (C++ function), 7  
Nanoshield\_ADC::readDifferentialVoltage23 (C++ function), 7  
Nanoshield\_ADC::readNext (C++ function), 4  
Nanoshield\_ADC::readRegister (C++ function), 3  
Nanoshield\_ADC::readVoltage (C++ function), 7  
Nanoshield\_ADC::setComparator (C++ function), 4  
Nanoshield\_ADC::setContinuous (C++ function), 5  
Nanoshield\_ADC::setGain (C++ function), 6  
Nanoshield\_ADC::setNotComparator (C++ function), 5  
Nanoshield\_ADC::setSampleRate (C++ function), 6  
Nanoshield\_ADC::startComparator\_SingleEnded (C++ function), 5  
Nanoshield\_ADC::writeRegister (C++ function), 3