
Nanoshield_LoadCell Documentation

Release 1.0

Nanoshield_LoadCell

Apr 01, 2017

Contents

1	Class Documentation	3
----------	----------------------------	----------

This is an Arduino library to measure weight using load cells reawith the ADS1230 IC from Texas Instruments.

- Source code: https://github.com/circuitar/Nanoshield_LoadCell
- Documentation: <http://nanoshield-loadcell.readthedocs.org/>
- Reference board: [LoadCell Nanoshield](#) from [Circuitar](#)

Library features include:

- Conversion to weight units, based on load cell capacity and sensitivity
- Moving average filter, with configurable number of samples
- Set zero weight point (tare)
- Get measurement in weight units or raw 20-bit value
- Continuous sampling using hardware timer (Timer 2)
- ADS1230 offset calibration

Note: this library uses Timer 2 on the ATmega microcontroller, or Timer 5 on ATmega2560 and ATmega1280. On ATmega168 and ATmega328 (i.e. Arduino UNO or similar), usage of this library will interfere with the *tone()* function.

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

The following [examples](#) are provided:

- [ReadWeight](#) shows how to do a simple weight measurement using a load cell.
- [LcdDigitalWeightScale](#) a digital weight scale using a load cell and an LCD display.
- [MultiWeight](#) shows how to read more than one weight using multiple load cells.
- [ReadRawValue](#) reads the raw 20-bit integer value from the ADS1230 IC.
- [ReadWeightNoAveraging](#) reads weight without averaging (default is to average the last 10 samples).
- [ReadWeightLowGain](#) uses low gain to measure a wider weight range.
- [OffsetCalibration](#) shows how to use offset calibration.

Class Documentation

class **Nanoshield_LoadCell**

A LoadCell Nanoshield or similar module using the ADS1230 IC.

Public Functions

Nanoshield_LoadCell (float *capacity*, float *sensitivity*, int *cs* = 8, bool *hiGain* = true, int *numSamples* = LOADCELL_MAX_SAMPLES)

Constructor.

Creates an object to access one LoadCell Nanoshield. The constructor parameters must match the hardware configuration used.

See *begin()*

Parameters

- *capacity*: load cell capacity. Any measurement unit can be used.
- *sensitivity*: load cell sensitivity, in mV/V.
- *cs*: chip select pin matching the jumper selected on the board (D4, D7, etc.).
- *hiGain*: true if operating in high gain mode (GAIN jumper closed), false if in low gain mode (GAIN jumper open).
- *numSamples*: number of samples used in moving average filter. Minimum is one (no filtering) and maximum is LOADCELL_MAX_SAMPLES.

void **begin** (bool *calibrate* = true)

Initializes the module.

Initializes SPI, CS pin, Timer 2 interrupts and moving average circular buffer. Performs initial self-calibration if requested via the *calibrate* parameter.

Parameters

- `calibrate`: if true, performs ADS1230 offset calibration during initialization.

bool **updated** ()

Checks if a new load cell reading is available.

Return true if a new load cell reading is available.

int32_t **getValue** ()

Gets the current 20-bit value, filtered and offset-corrected.

Return current load cell value.

int32_t **getRawValue** ()

Gets the current 20-bit value, filtered but not offset-corrected.

Return current load cell value without offset correction.

int32_t **getLatestValue** ()

Gets the latest reading (unfiltered 20-bit value), offset-corrected.

Return latest load cell value.

int32_t **getLatestRawValue** ()

Gets the latest reading (unfiltered 20-bit value), offset-corrected.

Return latest load cell value, without offset correction.

float **getWeight** ()

Gets the weight, filtered and offset-corrected.

Return current weight reading.

void **setZero** ()

Sets the current value to be the zero-weight value (offset or tare).

void **calibrate** ()

Performs ADS1230 offset calibration.

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

N

Nanoshield_LoadCell (C++ class), 3
Nanoshield_LoadCell::begin (C++ function), 3
Nanoshield_LoadCell::calibrate (C++ function), 4
Nanoshield_LoadCell::getLatestRawValue (C++ function), 4
Nanoshield_LoadCell::getLatestValue (C++ function), 4
Nanoshield_LoadCell::getRawValue (C++ function), 4
Nanoshield_LoadCell::getValue (C++ function), 4
Nanoshield_LoadCell::getWeight (C++ function), 4
Nanoshield_LoadCell::Nanoshield_LoadCell (C++ function), 3
Nanoshield_LoadCell::setZero (C++ function), 4
Nanoshield_LoadCell::updated (C++ function), 4