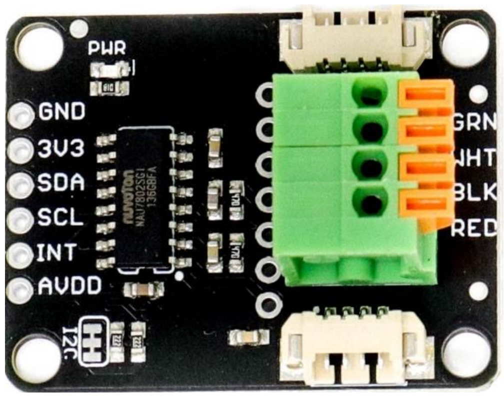
SmartElex Scale Breakout - NAU7802



We’ve designed the Scale so that little or no soldering is required.

Hardware Overview

The Scale is designed with the NAU7802 IC. This 24-bit analog-to-digital-converter is extremely precise and is designed to read the very small changes in voltage that a load cell or scale produce. The NAU7802 shines in that it has all the great features of the HX711 but uses a true I^{2}C interface allowing the Scale to be daisy chained with other sensors and devices on the bus.

I^{2}C Connections

As we mentioned, two connectors allow for easy daisy-chaining but if you prefer to wire in your own I^{2}C connection the pins are available broken out at the edge of the board.

Spring Terminal

A spring terminal is provided to make connecting to a scale possible without soldering. This terminal is operated by depressing the arm and inserting a wire. We found a ball-point pen or a small flat head screwdriver were handy to press one arm down at a time. The terminal can handle 20AWG to 26AWG size wire which is found on nearly all load cells.

There are four silkscreen indicators next to the terminal showing which color wires should be inserted where. These are the 'standard' load cell wire colors but may vary slightly between manufacturers. If you're in doubt, use your best guess and don't worry - you can't harm your load cell by wiring it backward!

If the arm returns to level with terminal after inserting the wire then the spring is correctly pinching the wire. If the arm still looks pressed, slightly remove the wire until the terminal pinches on the wire, not the insulation.

PTH Pins

Additionally, we've broken out the ADC and shielding to PTH holes for advanced users. These are helpful if you have a more complex load cell that has shielding around the wires (not required but helps reduce EMI).

Pin Definitions:

- **Red/E+** - Positive voltage of load cell energizing the wheatstone bridge. Most often connected to the red wire of a load cell.
- **Black/E-** - Negative voltage of the load cell, most often connected to the black wire of a load cell.
- **White/A-** - Negative branch of the wheatstone bridge, most often connected to the white wire of a load cell.
- **Green/A+** - Positive branch of the wheatstone bridge, most often connected to the green wire of a load cell.
- **Yellow/Shield** - Shield pin for more complex load cells that has shielding around the wire. Sometimes instead of a yellow wire there is a larger black wire, foil, or loose wires to shield the signal wires to lessen EMI. This is not required but helps reduce EMI.
- **B-** - Called VIN2N in the datasheet. Optional 2nd ADC channel, negative input. This would connect to a 2nd load cell, sharing the E+/E- connections above.
- **B+** - Called VIN2P in the datasheet. Optional 2nd ADC channel, positive input. This would connect to a 2nd load cell, sharing the E+/E- connections above.

The NAU7802 is optimized to work with differential pairs meaning load cells and wheatstone bridge type devices. However, it is possible to hook up a regular analog signal between **VIN1P** with **VIN1N** connected to ground.

From the datasheet:

This device is optimized to accept differential input signals, but can also measure single-ended signals. When measuring single-ended signals with respect to ground, connect the negative input (VIN1N or VIN2N) to ground and connect the input signal to the positive input (VIN1P or VIN2P). Note that when this device is configured this way, only half of the converter full-scale range is used, since only positive digital output codes are produced.

Channel 2

The NAU7802 has a 2nd, optional ADC channel that has a much lower input capacitance (5pF vs 14pF) but is otherwise identical to the 1st channel. By default, this channel has a 330pF capacitor across its inputs. This helps reduce noise on channel one. If you'd like to use the 2nd channel, we recommend removing the capacitor by cutting the **CAP** jumper.

Interrupts / Data Ready

By default, the NAU7802 has an interrupt output pin that goes low when a conversion is complete and new data is available to be read. This interrupt can also be configured to go high when new data is available. For most applications the interrupt pin is not needed. New data is available at 10, 20, 40, 80, and 320Hz so we can poll the device at regular intervals rather than checking the interrupt pin but it's available for more advanced applications (such as extremely low power projects where the main controller is put to sleep between readings).

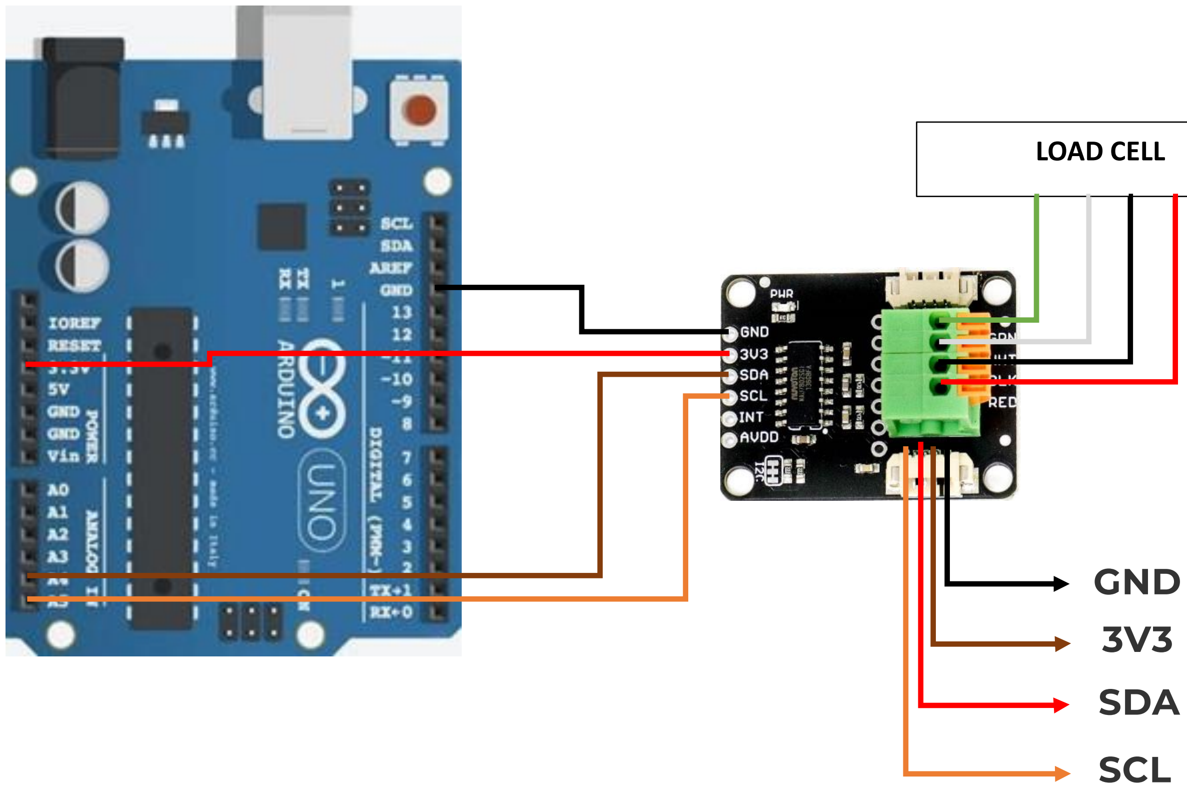
AVDD Pin

The NAU7802 has a built-in voltage regulator and will output 4.5V to 2.4V used to bias the load cell. By default, we set AVDD to **3.3V** but lowering the bias voltage can be useful for power saving. The **AVDD** pin is available in case the user wishes to access this voltage as a reference; the **AVDD** *should not* be used as a current source. Note that the regulator can only regulate down; you cannot power the board with 3.3V and ask the regulator to output 4.5V.

I^{2}C Jumper

Cutting the **I^{2}C** jumper will remove the 2.2kΩ resistors from the I^{2}C bus. In general, you shouldn't need to mess with this jumper. But if you have many devices on your I^{2}C bus, you may want to remove these resistors by cutting the two small traces between the solder pads.

Wiring:



Arduino	NAU7802
A5(SCL)	SCL
A4(SDA)	SDA
3.3V	3V3
GND	GND

Arduino Library

The SparkFun Qwiic Scale NAU7802 Arduino Library is a fully featured library for reading weights as well as configuring the IC. We recommend you install the library via the Arduino IDE by using the library manager and search for **SparkFun Scale**.

Once you have the library installed, we recommend you start with *Example1* and advance through the examples as you go. You do not need to read or absorb all the following functions, they are demonstrated and described in the examples. But if you're really advanced and just want the available functions, here they are:

- `bool begin(TwoWire &wirePort = Wire)` - Check communication and initialize sensor. Returns true if sensor is successfully started. Optionally, you can pass in a different Wire port.
- `bool isConnected()` - Returns true if device acknowledges a call to its I2C address.
- `bool available()` - Returns true if Cycle Ready bit is set (conversion is complete).
- `int32_t getReading()` - Returns 24-bit reading. Assumes CR Cycle Ready bit (ADC conversion complete) has been checked by `.available()`.
- `int32_t getAverage(uint8_t samplesToTake)` - Return the average of a given number of readings.
- `void calculateZeroOffset(uint8_t averageAmount = 8)` - Also called taring. Call this with nothing on the scale.
- `void setZeroOffset(int32_t newZeroOffset)` - Sets the internal variable. Useful for users who are loading values from NVM.
- `int32_t getZeroOffset()` - Ask library for the zero offset value. Useful for storing value into NVM.
- `void calculateCalibrationFactor(float weightOnScale, uint8_t averageAmount = 8)` - Call this with the value of the thing on the scale. Sets the calibration factor based on the weight on scale and zero offset.
- `void setCalibrationFactor(float calFactor)` - Pass a known calibration factor into library. Helpful when loading settings from NVM.
- `float getCalibrationFactor()` - Ask library for the calibration factor. Useful for storing value into NVM.
- `float getWeight(bool allowNegativeWeights = false)` - Once you've set zero offset and cal factor, you can ask the library to do the calculations for you. By default, negative weights will be returned as 0.
- `bool setGain(uint8_t gainValue)` - Set the gain by calling `myScale.setGain(NAU7802_GAIN_16)`. x1, 2, 4, 8, 16, 32, 64, 128 are available.
- `bool setLDO(uint8_t ldoValue)` - Set the onboard Low-Drop-Out voltage regulator to a given value by calling `myScale.setLDO(NAU7802_LDO_3V6)`. 2.4, 2.7, 3.0, 3.3, 3.6, 3.9, 4.2, 4.5V are available.
- `bool setSampleRate(uint8_t rate)` - Set the readings per second by calling `myScale.setSampleRate(NAU7802_SPS_80)`. 10, 20, 40, 80, and 320 samples per second is available.
- `bool setChannel(uint8_t channelNumber)` - Select between 1 and 2
- `bool calibrateAFE()` - Calibrate the analog front end of the IC. This function is unrelated to setting the zero offset and calibration factor and should rarely be used. It is recommended that the AFE be re-calibrated any time the gain, SPS, or channel number is changed. Returns true if CAL_ERR bit is 0 (no error).
- `bool reset()` - Resets all registers to Power On Defaults
- `bool powerUp()` - Power up digital and analog sections of scale, ~2mA
- `bool powerDown()` - Puts scale into low-power 200nA mode
- `bool setIntPolarityHigh()` - Set Int pin to be high when data is ready (default)
- `bool setIntPolarityLow()` - Set Int pin to be low when data is ready
- `uint8_t getRevisionCode()` - Get the revision code of this IC. Always 0x0F.

Here are the lower-level functions to manipulate registers:

- `bool setBit(uint8_t bitNumber, uint8_t registerAddress)` - Mask & set a given bit within a register
- `bool clearBit(uint8_t bitNumber, uint8_t registerAddress)` - Mask & clear a given bit within a register
- `bool getBit(uint8_t bitNumber, uint8_t registerAddress)` - Return a given bit within a register

- `uint8_t getRegister(uint8_t registerAddress)` - Get contents of a register
- `bool setRegister(uint8_t registerAddress, uint8_t value)` - Send a given value to be written to given address. Return true if successful.

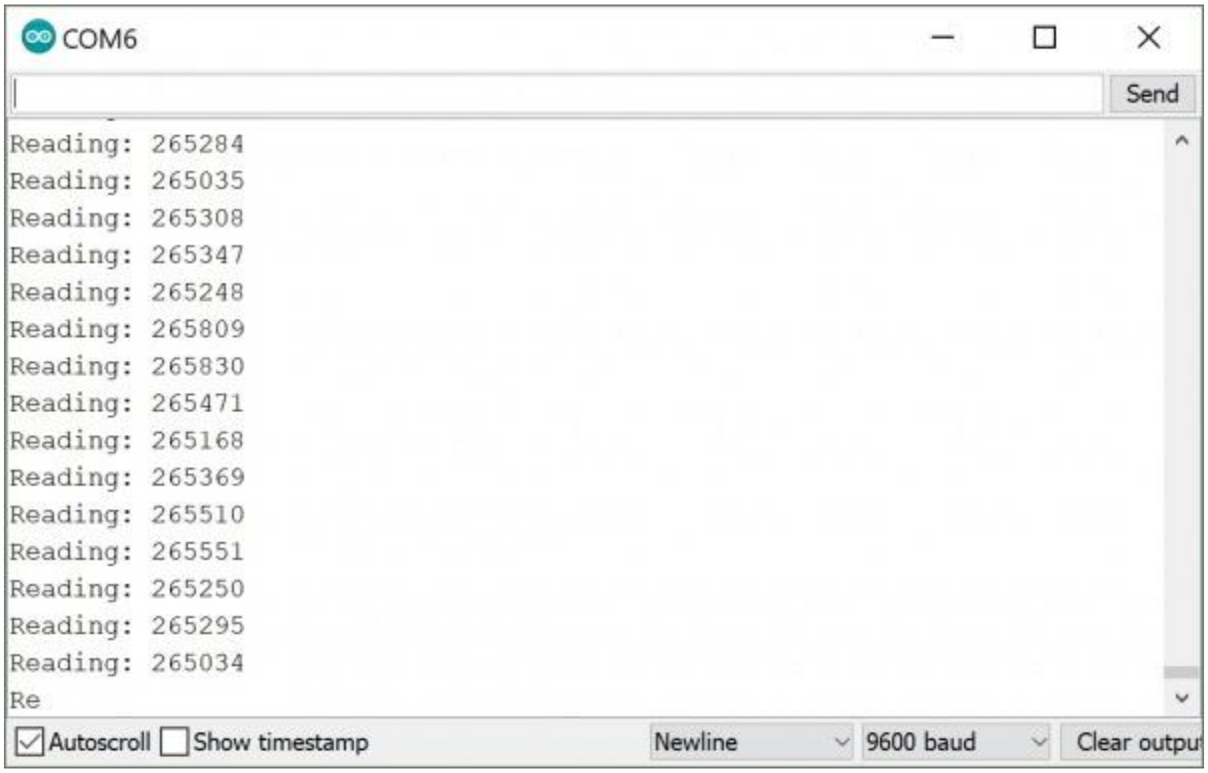
Example Code:

The following examples highlighted from the library will show the readings on a serial monitor.

```
#include <Wire.h>

#include "SparkFun_Qwiic_Scale_NAU7802_Arduino_Library.h" // Click here to get the library: http://librarymanager/All#SparkFun_NAU7802

NAU7802 myScale; //Create instance of the NAU7802 class

void setup()
{
  Serial.begin(115200);
  Serial.println("Qwiic Scale Example");

  Wire.begin();

  if (myScale.begin() == false)
  {
    Serial.println("Scale not detected. Please check wiring. Freezing...");
    while (1);
  }
  Serial.println("Scale detected!");
}

void loop()
{
  if(myScale.available() == true)
  {
    int32_t currentReading = myScale.getReading();
    Serial.print("Reading: ");
    Serial.println(currentReading);
  }
}
///////////////////////////////////////END/////////////////////////////////////////////
```

Example 1: Basic Readings

Example 1 will show you basic output from the NAU7802. This is helpful for getting the basic hardware setup. If you've got everything wired up correctly than the value should change by thousands or hundreds of thousands as you press on your scale.



```
Reading: 265284
Reading: 265035
Reading: 265308
Reading: 265347
Reading: 265248
Reading: 265809
Reading: 265830
Reading: 265471
Reading: 265168
Reading: 265369
Reading: 265510
Reading: 265551
Reading: 265250
Reading: 265295
Reading: 265034
Re
```

☑ Autoscroll ☐ Show timestamp   Newline   9600 baud   Clear outpu