COOL-WEB

Analog-Digital Wander ADS1115 als Voltmeter am ESP32 (Lolin32)

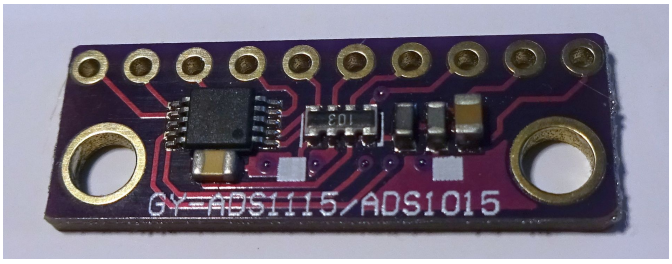
Der ADS1115 ist ein Analog Digital-Wandler von Texas Instruments mit einer Auflösung von 16 bit. Jetzt werdet ihr euch vielleicht fragen: Warum einen externen ADC an einen ESP32 anschließen, wo dieser doch genügend davon eingebaut hat?

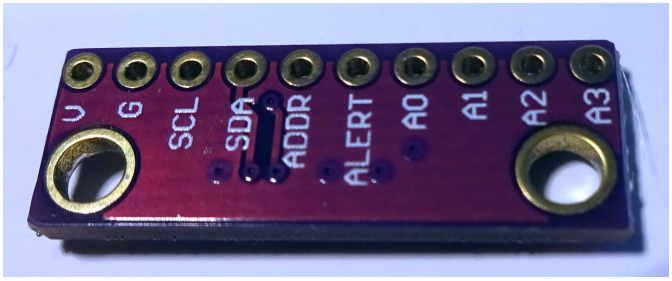
Nun, die Antwort lautet: Projektziel soll ein Voltmeter oder auch einfaches Oszilloskop am Odroid-Go sein. Und der führt keine Analog-Ausgänge an seinem GPIO-Port heraus. Also benutzen wir den ADS1115 am I2C-Port für die Messung. Außerdem ist die Auflösung mit 16 Bit (0-65535, ADS1115) zu 12 Bit (0-4095, ESP32) nochmal genauer. Ein genauere Auflösung ist für ein Voltmeter von Vorteil und für ein Oszilloskop unabdingbar.

Der ADS1115 ADC

Der ADS1115 bietet vier Analog-Kanäle (A0 bis A4), allesamt mit 16 bit Auflösung. Es kann mit 3.3V oder auch mit 5V betrieben werden. Und jetzt kommt ein ganz **wichtiger Punkt**: er verträgt nicht mehr als 0.3 Volt mehr an seinen Eingängen als die Betriebsspannung ist. Da wir Spannungen bis 5V messen und darstellen wollen, wählen wir als Versorgungsspannung ebenfalls 5V. Wenn wir dann mal einen USB-Port mit 5.2V messen, dann liegt das noch in der Toleranz, mehr sollte es dann aber nicht sein.

Würden wir den ADS1115 mit nur 3.3V betreiben, wäre die Maximalspannung 3.6V und er würde bei anliegenden 5V mit der Zeit Schaden nehmen. Wollen wir ihn gleich auf den Friedhof der Elektronikkomponenten befördern, messen wir eine 9V Batterie. Also bei diesem Projekt dran denken: den ADS1115 mit 5V betreiben und max. 5.3V anlegen.





Das ADS1115 Modul habe ich aus Fernost für kleines Geld (etwa 2 Euro) gekauft. Der eBay-Händler hat das Modul wie folgt beschrieben:

```
16 Bit I2C 4 Channel ADS1115 Module ADC with Pro Gain Amplifier For Arduino

Description

For microcontrollers without an analog-to-digital converter or when you want a higher-precision
ADC,
the ADS1115 provides 16-bit precision at 860 samples second over I2C.
The chip can be configured as 4 single-ended input channels, or two differential channels.
As a nice bonus, it even includes a programmable gain amplifier, up to x16,
to help boost up smaller single/differential signals to the full range.
We like this ADC because it can run from 2V to 5V power/logic,
can measure a large range of signals and its super easy to use.
It is a great general purpose 16 bit converter.
The chip's fairly small so it comes on a breakout board with ferrites to keep the AVDD and AGND
quiet.
```

```
Interfacing is done via I2C.
The address can be changed to one of four options (see the datasheet table 5)
so you can have up to 4 ADS1115's connected on a single 2-wire I2C bus for 16 single ended inputs.
```

Zu kaufen gibt es das Modul zum Beispiel hier (Preisstand März 2020):

- 2 Stück für ca. 9 Euro aus Deutschland über amazon.de
- ab ca. 2 Euro aus Fernost über amazon.de
- ab ca. 2 Euro aus Fernost über ebay.de

Genaueres weiß natürlich das Datasheet zum ADS1115 von TI zu berichten:

Features:

- Ultra-Small X2QFN Package: 2 mm × 1.5 mm × 0.4 mm
- Wide Supply Range: 2.0 V to 5.5 V
- Low Current Consumption: 150 µA
- Programmable DataRate: 8 SPS to 860 SPS
- Internal Low-Drift Voltage Reference
- I2C Interface: Four Pin-Selectable Addresses
- Four Single-Ended or Two Differential Inputs
- Programmable Comparator
- Operating TemperatureRange: –40°C to +125°C

SPS heißt so viel wie Samples per Second, also Messungen pro Sekunde. In der schnellsten Betriebsart sind das max. 860 SPS.

Und schon macht sich Ernüchterung breit. für ein Oszilloskop braucht man ungefähr 10 Samples pro Hertz. Es wären also nur Frequenzen bis 8 Hertz anständig möglich. Und selbst wenn man jedes Messung für bare Münze nimmt und nicht den Mittelwert aus 10 Messungen, sind wir immer noch bei unter 1 kHz. Da fällt mir momentan keine sinnvolle Verwendung zu ein.

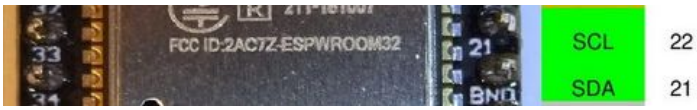
Realistische und realisierbares Projektziel wird also ein Voltmeter bis 5 Volt, vielleicht mit kontinuierlichen Messung und Darstellung dieser in einer Verlaufskurve, etwa, um zu schauen, wie schnell sich ein Akku entleert.

Da das ständige Umwandeln und Hochladen bzw. Kopieren der Firmware auf die SD-Karte des Odroid-Go etwas nervt, habe ich mich entschlossen, die Entwicklung auf einem normalen ESP32-Board, dem Lolin32 zu beginnen, den ADS 1115 zu testen und die Ausgaben erst einmal über die serielle Schnittstelle zu machen.

Und wenn dann soweit alles läuft, füge ich die Befehle zur Ausgabe auf das Odroid-Go-Display hinzu und teste dann den Rest auf dem Odroid-Go. So muss ich nur am Ende den Entwicklungszyklus mit Firmware-Umwandlung durchführen.

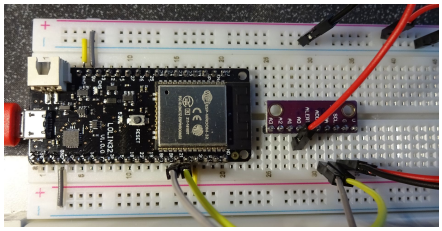
Anschluss und Schaltung auf Breadboard

Werfen wir einen Blick auf das Lolin32-Pinout (Download hier als PDF), um die I2C-Ports herauszufinden, als die wären: SDA (serial data) und SCL (serial clock)

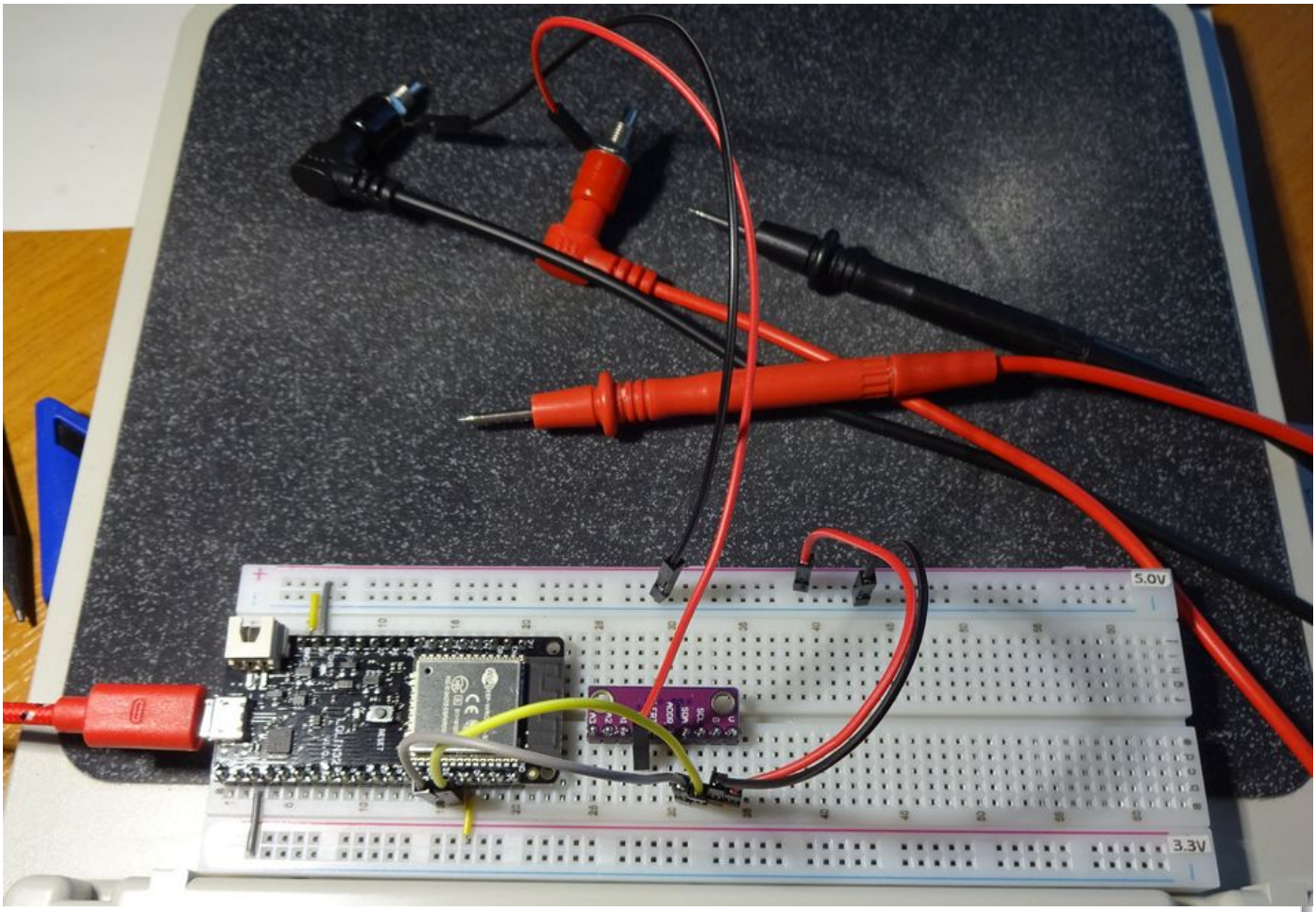


Den 5V-Ausgang finden wir neben der Akku-Buchse. Die Verkabelung ist - wie von I2C gewohnt - mit ein paar Jumperkabeln schnell erledigt:

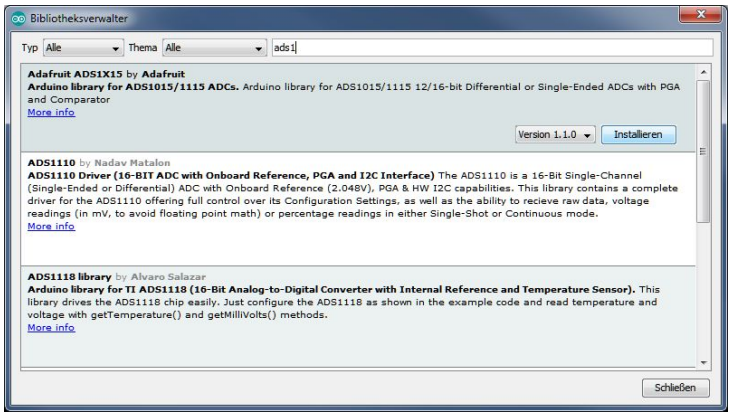
Lolin32	ADS1115 Modul	Messkabel
SDA, Pin 22	SDA	
SCL, Pin 21	SCL	
5V	V	
GND	GND	schwarz
	A0	rot



Auf einem Breadboard aufgebaut sieht das Ganze dann so aus:



Library installieren und testen



Auch bei diesem Modul müssen wir nicht händisch das dafür gütlige über I2C realisierte Protokoll programmieren, denn auch für das ADS1115-Modul gibt es bereits eine fertige Library.

Wir können also erstmal das 54 Seiten starke Data Sheet zur Seite packen, zumindestens solange, bis wir etwas brauchen, dass die Library nicht abdeckt.

Ich habe mich für die *Adafruit ADS1x15 Library* entschieden, die einfach über den Bibliotheksverwalter in der Arduino IDE zu installieren ist. Danach findet man unter *Datei / Beispiele / Adafruit ADS1x15* ein paar Beispiele, z. B. den *comparator*, das für unser Projekt brauchbar erscheint.

Ich habe es nur minimal angepasst:

```
#include <Wire.h>
#include <Adafruit_ADS1015.h>

 Adafruit_ADS1115 ads;  /* Use this for the 16-bit version */
//Adafruit_ADS1015 ads;     /* Use thi for the 12-bit version */

void setup(void)
{
  Serial.begin(115200);
  Serial.println("Hello!");

  Serial.println("Single-ended readings from AIN0 with >3.0V comparator");
  Serial.println("ADC Range: +/- 6.144V (1 bit = 3mV/ADS1015, 0.1875mV/ADS1115)");
  Serial.println("Comparator Threshold: 1000 (3.000V)");

  // The ADC input range (or gain) can be changed via the following
  // functions, but be careful never to exceed VDD +0.3V max, or to
  // exceed the upper and lower limits if you adjust the input range!
```

```
  // Setting these values incorrectly may destroy your ADC!
  //                                                                ADS1015  ADS1115
  //                                                                -------  -------
  ads.setGain(GAIN_TWOTHIRDS);  // 2/3x gain +/- 6.144V  1 bit = 3mV      0.1875mV (default)
  // ads.setGain(GAIN_ONE);        // 1x gain   +/- 4.096V  1 bit = 2mV      0.125mV
  // ads.setGain(GAIN_TWO);        // 2x gain   +/- 2.048V  1 bit = 1mV      0.0625mV
  // ads.setGain(GAIN_FOUR);       // 4x gain   +/- 1.024V  1 bit = 0.5mV    0.03125mV
  // ads.setGain(GAIN_EIGHT);      // 8x gain   +/- 0.512V  1 bit = 0.25mV   0.015625mV
  // ads.setGain(GAIN_SIXTEEN);    // 16x gain  +/- 0.256V  1 bit = 0.125mV  0.0078125mV

  ads.begin();

  // Setup 3V comparator on channel 0
  ads.startComparator_SingleEnded(0, 1000);
}

void loop(void)
{
  int16_t adc0;
  int16_t mv=0;

  // Comparator will only de-assert after a read
  adc0 = ads.getLastConversionResults();
  //Serial.print("AIN0: "); Serial.println(adc0);
  mv = adc0 * 0.1875;
  Serial.print(mv); Serial.println(" mV");

  delay(500);
}
```

Das einzustellende Board unter Werkzeuge ist in diesem Fall "WEMOS LOLIN32". Der Upload funktioniert ohne weitere Besonderheiten.

Wie man dem Beispiel-Source entnehmen kann, kann man noch genauere Messwerte erreichen, wenn man die maximale Voltzahl heruntersetzt, weil sich die 65536 Stufen dann auf einen kleineren Bereich aufteilen.

Als Testobjekt habe ich eine 1.5 AA Batterie hergenommen. Der Serial Monitor zeigt dabei einen Messwert von 1520 mV (also 1.52 V) an, was ziemlich genau dem Wert entspricht, den auch mein Multimeter liefert: