

Ein IOT-Projekt: Preiswerte Digitalwaage wird zum WiFi-Kraftmesser

Robert Kastl, René Dohrmann, Volkhard Nordmeier

Freie Universität Berlin, Didaktik der Physik, Arnimallee 14, 14195 Berlin
roka@physik.fu-berlin.de, rdohrmann@zedat.fu-berlin.de, volkhard.nordmeier@fu-berlin.de

Abstract

Elektronische Waagen, ob Personen-, Koffer- oder Briefwaagen basieren meist auf der Messung der Verformung eines Biegebalkens über aufgeklebte Dehnungsmessstreifen (DMS) [1]. Geschickt verschaltet entsteht aus ihnen eine Wheatstone'sche Messbrücke, an deren Ausgang ein preiswertes Modul zur Digitalisierung und graphischen Aufbereitung durch eines der mittlerweile weit verbreiteten Arduino-Boards angeschlossen werden kann. Die Gesamtkosten für ein derartiges, WLAN-fähiges System betragen etwa 50 €. Angesichts des günstigen Preises und der einfachen Realisierbarkeit kann der Umbau einer elektronischen Waage gut als Projekt realisiert werden.

1. Motivation

Die Preise der von Lehrmittelherstellern angebotenen Kraftmessplatten und elektronischen Kraftsensoren bewegen sich im Bereich mehrerer hundert Euro. Ziel des hier vorgestellten Projektes ist es, preisgünstig z. B. eine elektronische Badezimmerwaage oder andere elektronische Waagen so zu modifizieren, dass sie die Messwerte an einen Computer oder, wenn gewünscht, auch per WiFi an ein Smartphone oder Tablet übertragen (Abb. 1). Die Gesamtkosten betragen etwa 50 €. (Eine Materialliste mit Preisen befindet sich im Anhang.)



Abb. 1: Rutschfest umgebaute Personenwaage an WiFi-fähigem ‚WIO-Link‘-Controller

2. Grundlegende Softwareinstallation

Die Programmierung der verwendeten Microcontroller erfolgt über die auch in Schulen gern genutzte Arduino-Entwicklungsumgebung (IDE), die für die

unterschiedlichsten Betriebssysteme und Controllerbausteine kostenlos angeboten wird.

Dazu muss in einem ersten Schritt die Arduino-Software (www.arduino.cc) auf einen Computer heruntergeladen und installiert werden. Nach dem Start der Arduino-IDE kann über den Menüpunkt Werkzeuge → Bibliotheken verwalten → HX711 (von Bogdan Neclau) die essentielle Bibliothek für den Wägezellenverstärker HX711 installiert werden. Der Computer ist damit bereit, und es kann mit den Hardwareänderungen an der Waage begonnen werden.

3. Umbau einer Personenwaage

Zunächst müssen die Abdeckung am Boden der Waage geöffnet und die Anschlussleitungen der Wägezellen zur vorhandenen Mess- und Anzeigeelektronik entfernt werden.

Die Waage sollte ungefähr so aussehen wie in Abbildung 2 dargestellt. An Stelle des LCD und der zugehörigen Elektronik sieht man hier schon den verbauten ‚Load Sensor Combinator‘, den ‚Load Cell Amplifier‘ und die Leitung zum Anschluss an einen Microcontroller [2].

An den Ecken der Personenwaage sind gut erkennbar die Wägezellen angebracht, von denen je drei Leitungen wegführen (Abb. 3). Diese Anschlussleitungen müssen mit dem ‚Load Sensor Combinator‘ (Abb. 4 und Abb. 5) verbunden werden. Um sie korrekt zu identifizieren, wird mit einem Multimeter der jeweilige Widerstand zu den beiden anderen Drähten bestimmt: Der Gesamtwiderstand über beide in einer Zelle eingebauten Widerstände (Dehnungsmessstreifen DMS) beträgt meist ca. 2000 Ohm. In der Leitung zum Mittelabgriff misst man entsprechend etwa den halben Widerstand.

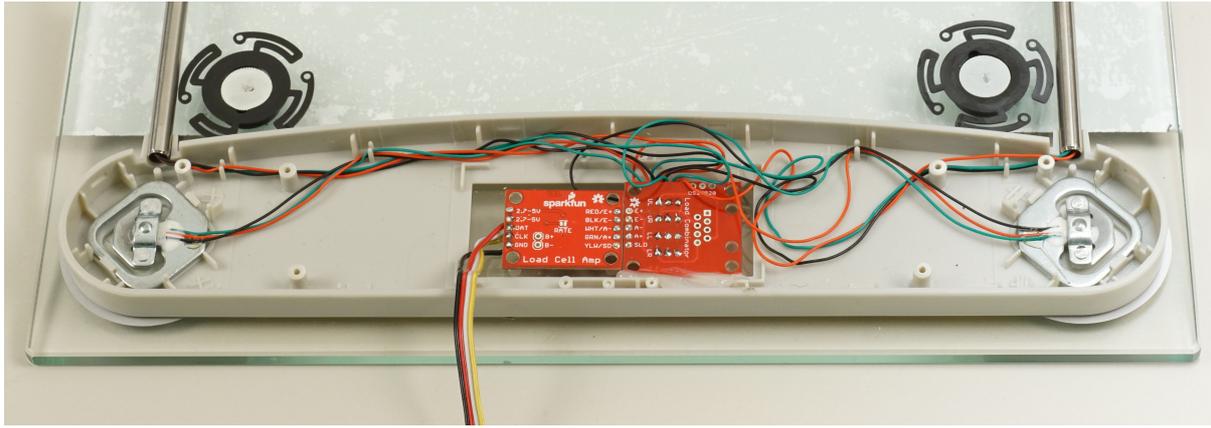


Abb. 2: Die vier Wägezellen sind auf diesem Bild schon mit dem ‚Load Sensor Combinator‘ in Wheatstone‘scher Brückenschaltung verbunden.

Die Leitungen des Mittelabgriffs werden jeweils in die mit ‚C‘ (Center) bezeichneten Lötunkte des ‚Load Combinators‘ gelötet. Die beiden anderen Leitungen der Wägezellen werden jeweils mit dem ‚+‘-Eingang bzw. mit dem ‚-‘-Eingang der Platine verbunden. LR bedeutet lower right, LL lower left, UR upper right und UL upper left.

Die Lötverbindungen zum Verstärker-/Wandlermodul erfolgen recht einfach gemäß den Beschriftungen auf den Boards (Abb. 6). Nicht vergessen: Die Anschlüsse VCC und VDD auf der Platine von SparkFun müssen miteinander verbunden werden.

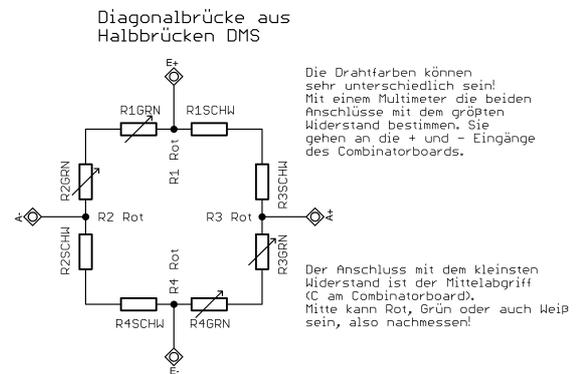


Abb. 5: Schaltungstechnisch erzeugt das ‚Combinator Board‘ aus den vier Wägezellen der Personenwaage eine Wheatstone‘sche Diagonalbrücke.

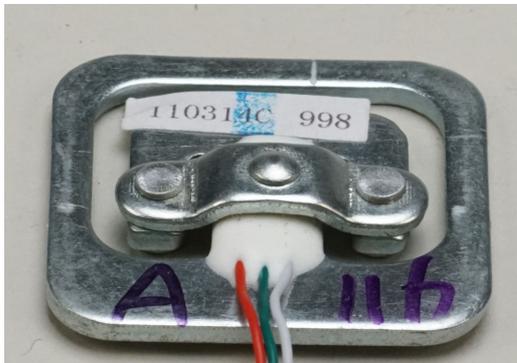


Abb. 3: Wägezelle mit anderen Leitungsfarben



Abb. 6: Blick auf die beiden Platinen in der umgebauten Waage (Bestückungsseite)

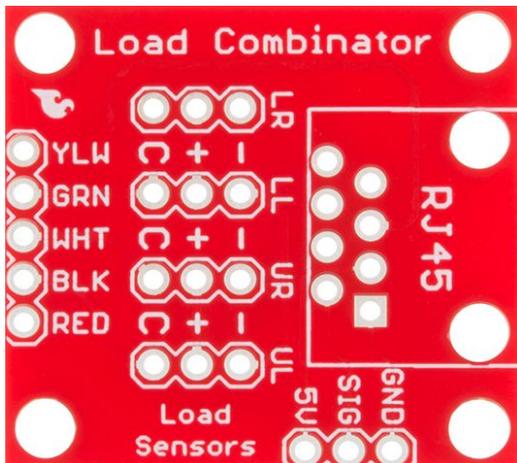


Abb. 4: SparkFun ‚Load Cell Combinator‘

Weitere Hinweise und Hilfestellungen finden sich auf der Webseite von SparkFun.

Unter <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide> wird recht detailliert erläutert, wie die Beschaltung der Waage zu erfolgen hat (Abb. 7).

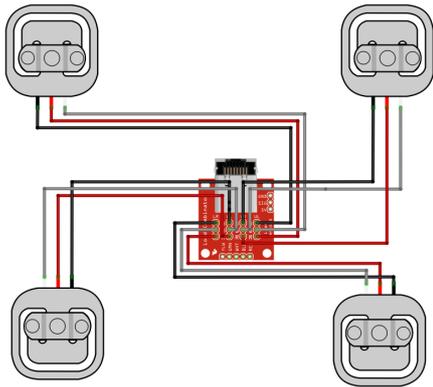


Abb. 7: Nach SparkFun mit dem Programm ‚Fritzing‘ erzeugtes Verdrahtungsschema für Personenwaagen

4. Umbau von verschiedenen Waagen

Elektronische Briefwaagen (Abb. 10) oder Kofferwaagen (Abb. 11), deren Funktionsweise auch auf Wägezellen auf DMS-Basis beruhen, können ohne das ‚Combinator Board‘ erheblich einfacher an die HX711-Wandlerplatine angeschlossen werden (Abb. 9). Sie haben nur vier Leitungen, die zudem entsprechend farblich codiert sind, was den Umbau erheblich vereinfacht. Es gilt: Rot für E+, schwarz für E-, weiß für A- und grün für A+. Die Leitungen können entsprechend ihrer Farbe an das Modul gelötet werden. An den Anschluss YLW könnte eine vorhandene Schirmung angeschlossen werden.

Um die einzelnen Module der elektronischen Waagen darstellen und einen Blick auf Wägebalken mit Dehnungsmesstreifen werfen zu können, haben wir auch diese Teile einer Briefwaage in ein transparentes Gehäuse platziert (Abb. 8).

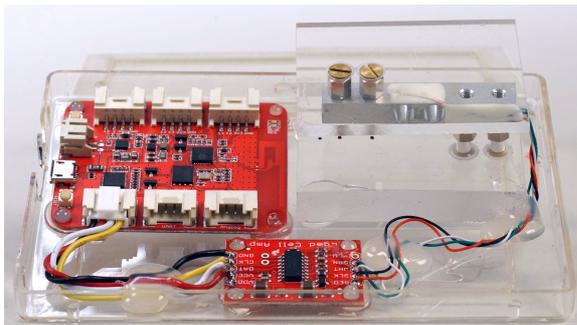


Abb. 8: Transparente, WiFi-fähige Briefwaage mit WIO-Link und HX711-Board

5. Wandlermodul (A/D) vorbereiten und anschließen

Auf dem Wandlermodul befindet sich ein 24bit-Analog/Digitalwandler vom Typ HX711 [3], dessen Wandlungsrate mit einer kleinen Brücke am Boden des Moduls auf 10 Samples pro Sekunde (SPS)

eingestellt ist. Wird diese Brücke z. B. mit einem Cuttermesser aufgetrennt, sind maximal 80 SPS möglich. Damit erhöht sich auch das Rauschen etwas, was jedoch für unsere Anwendung nicht so relevant ist. Da unser Ziel eine möglichst hohe Samplerate ist, trennen wir diese Brücke auf. Nicht vergessen: Es müssen auf der Platine von SparkFun die Anschlüsse VDD und VCC verbunden werden (Abb. 9)!

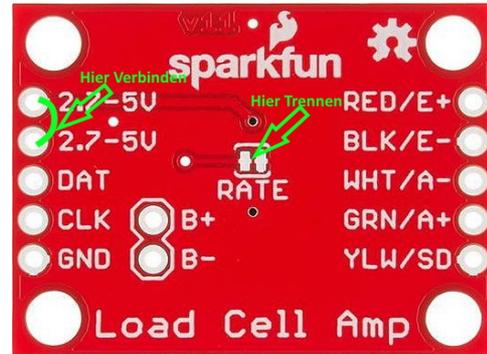


Abb. 9: Bodenseite – VCC und VDD verbinden, und für 80 SPS Brücke ‚RATE‘ trennen.

Um jetzt noch eine möglichst einfache, leicht lösbare und universelle Verbindung zu den möglichen Microcontrollern herzustellen, wird eine vieradrige ‚jumper to grove‘-Flachbandleitung an die Wandlerplatine gelötet. Belegung: Schwarz an GND, Rot an VCC und VDD, Weiß an DAT und Gelb an CLK. Die Wahl fiel deshalb auf die Grove-Belegung, weil der Hersteller ‚Seeed Studio‘ u. a. eine große Anzahl an preiswerten Sensoren mit eben diesem Steckersystem im Angebot hat und sich dieses System zum Quasistandard zu entwickeln scheint. Auch das für Kinder vorgesehene System von Calliope verfügt über zwei Buchsen dieses Typs.

Auf den Seeduino [4] wird noch in die mittlere Buchse (I2C-Buchse; HX711 benutzt allerdings ein abweichendes Protokoll) der kleine Pfostenstecker dieser Leitung gesteckt. Danach können die ersten Programme geschrieben und gestartet werden.

6. Erster Funktionstest

Nachdem die Hardware nun entsprechend vorbereitet ist, kann ein erstes Programm zum Funktionstest der Waage in den Microcontroller programmiert werden.

Statt der benutzten Pins im Beispiel der HX711-Bibliothek ‚HX711_basic_example‘ verwenden wir abweichend diese Definitionen:

```
const int LOADCELL_DOUT_PIN = SDA;
const int LOADCELL_SCK_PIN = SCL;
```

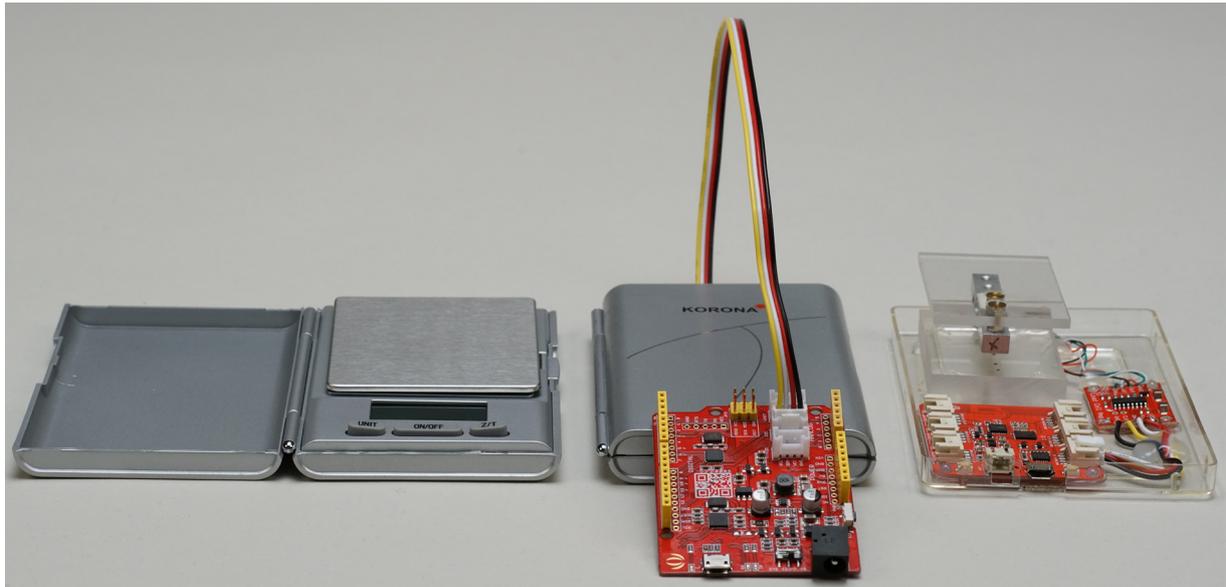


Abb. 10: Verschiedene Stufen des Umbaus einer Briefwaage: neue Briefwaage (li.), an „Seesduino“ angeschlossen (Mi.) sowie WiFi-fähigers (transparentes) Demonstrationsmodell (re.; vgl. Abb. 8.)

Die Datenausgabe erfolgt leitungsgebunden über die USB-Schnittstelle an den Computer, über den Menüpunkt der Programmierumgebung Arduino: Werkzeuge → Serieller Monitor. Die Schnittstelle wird in diesem Beispiel mit 57 600 Baud betrieben.

```
//Programm: HX711_basic_example:

#include "HX711.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 2;
//← hier statt der 2 SDA eintragen
const int LOADCELL_SCK_PIN = 3;
//← hier statt der 3 SCL eintragen

HX711 scale;
void setup() {
  Serial.begin(57600);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
}
void loop() {
  if (scale.is_ready()) {
    long reading = scale.read();
    Serial.print("HX711 reading: ");
    Serial.println(reading);
  } else {
    Serial.println("HX711 not found.");
  }
  delay(100);
}
```

7. Kalibrierung und Betrieb am USB-Port

Ändern sich die ausgegebenen Werte bei Belastung der Waage adäquat, kann das System kalibriert werden. Mit einem Wägestück von 1 kg und dem Programm auf unserer Webseite didaktik.physik.fu-berlin.de/projekte/kraftmessung/Arduino.zip gelingt dies recht einfach. Die Kalibrierdaten werden dabei auf das EEPROM des Microcontrollers geschrieben und bleiben dort bis zum nächsten Kalibriervorgang gespeichert.

Das Programm ist etwas universeller gehalten, da auch Microcontroller vom Typ ESP8266 und ESP32 verwendet werden können, die nach entsprechender Programmierung ihre Messwerte drahtlos per WiFi an Computer, Tablets oder Handys übertragen. Die aktuelle Arduino-Entwicklungsumgebung bietet neben der reinen Zahlenausgabe auch die Möglichkeit, die erfassten Daten über Werkzeuge → serieller Plotter grafisch darzustellen.

8. Internet Of Things (IOT)

IOT bezeichnet die Fähigkeit der Kommunikation zwischen Gegenständen untereinander bzw. nach außen hin über Netzwerkprotokolle. Das bekannteste Internetprotokoll dürfte das HTTP-Protokoll sein. Es hat allerdings den Nachteil, dass jede Anfrage eine Verbindung zum Server aufbaut und danach wieder schließt. Das braucht Zeit und ist mit einem großen Overhead verbunden. In der Welt des IOT ist das Protokoll MQTT recht weit verbreitet, wie eine Internetrecherche mit den Suchbegriffen HX711 und MQTT zeigt. Dabei senden die Geräte Daten an einen Server, der sie anschließend weiter verteilen kann. Dieses Protokoll ist zuverlässig, allerdings nicht echtzeitfähig, und es benötigt einen externen Server. In einem früheren Projekt konnten wir bereits positive Erfahrungen mit einer serverbasierten Lösung und dem WebSocket-Protokoll sammeln (<https://tetfolio.fu-berlin.de/tet/850629>). Die aktuelle Lösung benutzt zum schnellen Datentransfer ebenfalls das WebSocket-Protokoll, das geschaffen wurde, um eine dauerhafte, bidirektionale Verbindung zwischen Server und Client mit einer einzigen Anfrage zu halten, und für Streaming-Anwendungen geeignet ist.

Der von uns entwickelte Kraftmesser bietet durch die Verwendung entsprechender Controller-Bausteine die Möglichkeit, dieses Protokoll einfach zu nutzen. Wir haben uns für die Verwendung von

ESP8266- und ESP32-Modulen entschieden, da diese neben schnellen Prozessoren und großem Arbeitsspeicher auch nativ WIFI 802.11 b/g/n unterstützen [5].

Ausschlaggebend war allerdings, dass diese Microcontroller auch sehr gut von der Programmierumgebung Arduino (ab Version 1.6.4) unterstützt werden.

In den folgenden Links wird die Vorgehensweise zur Installation der nötigen Board-Bibliotheken für beide Controller detailliert beschrieben [6]:

- Die Vorbereitung von Boards mit dem ESP-8266 erfolgt nach dieser Beschreibung: <https://github.com/esp8266/Arduino#installing-with-boards-manager>
- Zur Installation des Upload-tools um auszuliefernden statischen Inhalte in das Dateiensystem des ESP8266 abzulegen, wird nach folgender Anleitung verfahren: <https://github.com/esp8266/arduino-esp8266fs-plugin>
- Vorbereitung von Boards mit dem ESP-32: https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards_manager.md
- Uploads in das Filesystem des ESP32 sind nach Ausführung folgender Anleitung möglich: <https://github.com/me-no-dev/arduino-esp32fs-plugin>

Über Werkzeuge → Board → Boardverwalter kommt man in den ‚Boardverwalter‘. Klickt man mit dem Mauszeiger auf die entsprechenden Einträge, erscheint ein Button mit der Aufschrift ‚Installieren‘, womit dieses dann auch erledigt wird.

Für beide Boards muss noch nach dem vorherigen Muster die Bibliothek ‚Websockets‘ (von Markus Sattler) installiert werden.

Nach erfolgter Installation bietet die Arduino-Entwicklungsumgebung unter dem Menüpunkt ‚Werkzeuge‘ die Möglichkeit, das verwendete Board und die passende Schnittstelle auszuwählen (Seed Wio Link).

Bei Verwendung eines ESP-8266 wird unter ‚Flash-Size‘ 1M SPIFFS gewählt, bei Verwendung eines ESP32 ist dies nicht notwendig.

9. Programmierung der ESP-Controller

Die Leistungsfähigkeit der ESP-Controller macht es möglich, sowohl die Spannung der Wheatstone-Brücke aus den Wägezellen zu erfassen, als auch einen Accesspoint mit Webserver und Websocket-Server zu betreiben.

Der Webserver liefert die statischen Inhalte unseres Webinterfaces, der Websocket-Server dagegen die kontinuierlich erfassten Daten. Diese werden über einen in Javascript geschriebenen Client in empfangen, decodiert und grafisch dargestellt [7].

Das bedeutet, dass zunächst ein „C“-Programm kompiliert und auf die Controller geladen werden muss. Dies funktioniert, wie bei jedem Arduino-

Board, über Sketch upload. Für die auszuliefernde Webseite wird im Verzeichnis des Sketches ein zusätzlicher Ordner mit der Bezeichnung ‚data‘ angelegt. In diesem Ordner liegen die notwendigen Daten für die Webseite. Beide Programme/Daten sind ebenfalls in der obigen Datei Arduino.zip enthalten und auf der Webseite der Physikdidaktik der Freien Universität Berlin abrufbar.

Über den Menüpunkt Werkzeuge → (ESP32 oder EAP8266) Sketch Data Upload werden sämtliche zur Darstellung der Messwerte (s. Abb. 11) auf dem Endgerät notwendigen Dateien in einen speziellen Datenbereich der Controller geladen.

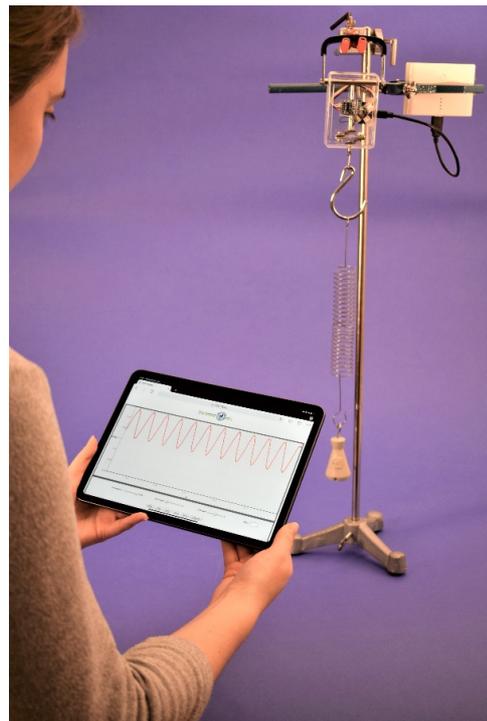


Abb. 11: Federpendel an einer nun transparenten, umgebauten Kofferwaage mit Akku.

10. Bedienung des Programms

Nach einem Neustart der ESPs erscheint auf dem Computer/Tablet/Smartphone ein neuer WiFi-Zugriffspunkt, mit der MAC-Adresse des verwendeten ESPs als SSID. Auf ein Passwort wurde vorerst verzichtet, es ist allerdings bereits im Sketch vorgesehen. Ähnlich wie bei Bluetooth-Verbindungen ist aktuell der Zugriff bewusst nur einem einzelnen Gerät gestattet. Um dies zu ändern oder den Zugang über einen Router aus der Ferne zu ermöglichen, sind nur wenige Zeilen im Sketch zu ergänzen, ähnlich der oben genannten tet.folio-Anwendung.

Eines der Ziele des Projektes war es, einen systemübergreifenden Zugang zu den ermittelten Messdaten der umgebauten Waagen zu ermöglichen. So können sowohl Computer mit verschiedenen Betriebssystemen als auch Tablets und Smartphones benutzt werden, sofern sie mit einem halbwegs aktuellen Browser ausgestattet sind.

Nach Eingabe der IP-Adresse 192.168.4.1 (Abb. 12) im Browser wird eine Verbindung zum HTTP-

Server auf Port 80 des ESP-Controllers aufgebaut. Dieser liefert zunächst eine statische Seite aus. Über Port 81 hält der WebSocket-Server eine ständige Verbindung zum Browser, liefert die gewandelten Analogwerte und nimmt Änderungen zur Abtastrate, Start/Stop und Kalibrierung entgegen. Werte wie Puffergröße und Schwelle (Trigger) werden vom ebenfalls vorab ausgelieferten Javascript-Programm bedient.



Abb. 12: Adressszeile

Die grafische Aufbereitung (Abb. 15) erfolgt über die Java-script-Bibliothek ‚Flotr2‘ [8], die einem 7-Segment-Display nachgebildete Mittelwertanzeige (Abb. 13) im unteren Bereich des Bedienpanels benutzt einen freien Zeichensatz [9]. Beide Maßnahmen sorgen dafür, dass auch mit etwas älteren Endgeräten eine zügige Darstellung möglich ist.



Abb. 13: Bedienelemente und Mittelwertanzeige

Bevor die eigentlichen Messungen (Abb. 14) beginnen können, muss die Kraftmessplatte über den dafür vorgesehenen Menüpunkt kalibriert werden. Die vorgesehene Prozedur erfragt die Auflage eines 1-kg-Massestücks und bestimmt daraus Offset und Steigung der linearen Funktion. Diese grundlegenden Kalibrierdaten werden im EEPROM gespeichert und stehen bis zu einer möglichen erneuten Kalibrierung zur Verfügung.



Abb. 14: Schüler bei Tests der Kraftmessplatte



Abb. 15: Screenshot – Landung des Schülers auf und Abprung von der Kraftmessplatte

11. Datenexport

Um die Messdaten auch noch in anderer Form darstellen zu können, wurde eine Exportmöglichkeit (Abb. 16) implementiert. Die im Puffer vorhandenen Daten stehen im Endgerät dann als Komma separierte Werte bereit und können damit leicht wieder importiert werden.

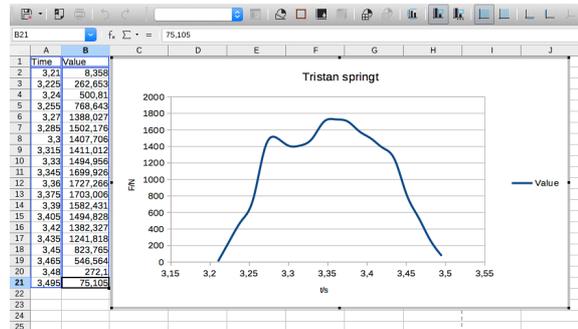


Abb. 16: Import in LibreOffice Calc

Abschließend der Hinweis, dass für das hier vorgestellte Projekt ausschließlich Open-Source-Programme verwendet wurden, und auch die Platinen sind Open Hardware. Sämtliche Schaltpläne sind somit frei zugänglich.

12. Quellen (Stand 03.2020)

- [1] Einführende Darstellung in die Thematik von J. Plate:
www.netzmafia.de/skripten/hardware/Arduino/Waage/index.html
- [2] Detailliertere Beschreibung bei SparkFun:
<https://learn.sparkfun.com/tutorials/getting-started-with-load-cells/all>
- [3] Datenblatt des HX711 von AVIA:
<http://en.aviaic.com/images/ueditor/1566291913105272.pdf>
- [4] Datenblatt Seeedduino (Arduinokompatibel):
www.seeedstudio.com/Seeedduino-V4-2-p-2517.html
- [5] Datenblätter der ESP-Controller von Espressiv:
www.espressif.com/en
- [6] Programmierung der ESP-Boards mit der Arduino-IDE:
www.heise.de/make/artikel/ESP-Boards-mit-

[der-Arduino-IDE-programmieren-4130861.html](#)

- [7] Websocket-Anwendung ESPScope:
<https://github.com/krzychb/EspScopeA0>
- [8] Javascript Grafikbibliothek flot2:
<https://github.com/HumbleSoftware/Flotr2>
- [9] Font: www.keshikan.net/fonts-e.html

Anhang-Materielliste

Personenwaage z. B. WAAGE PS101N	reichelt.de	14,95 €
Powerbank z. B. Varta VAR PP	reichelt.de	6,90 €
Seeeduino V4.2	exp-tech.de	8,90 €
Seedstudio WIO Link ESP8266	exp-tech.de	13,33 €
Micro-USB Kabel 0.6m	exp-tech.de	1,40 €
SparkFun Load Cell Amplifier (HX711)	exp-tech.de	10,53 €
SparkFun Load Sensor Combinator	exp-tech.de	2,09 €
Seed Studio 4pin male Jumper to grove 4pin (5 Stück)	exp-tech.de	2,49 €

Modernere Alternative zu WIO-Link:

Huzzah32 ESP32 Feather Board	distrelec.de	18,96 €
Grove Shield Feather-Wing Adapter	distrelec.de	6,19 €

Falls kein LötKolben mit feiner Lötspitze, Lötzinn und Draht/Litze (0,25mm² ist ausreichend) vorhanden sind, müssen auch diese besorgt werden. Alternativ kann man auch eine der 4pin-,Male to Grove'-Leitungen passend abschneiden