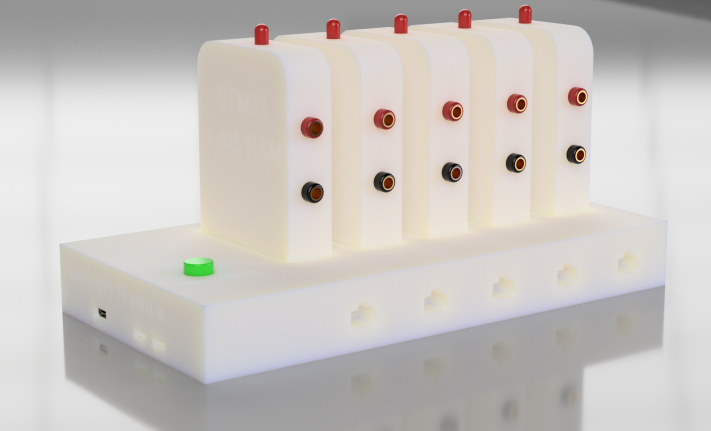
Messbox 2000

Projekt 2020 von Timo Erdmannsdörfer und Johannes Wittmann



Inhalt

[1. Zweck/Kernfunktionalität: 2](#_Toc31810883)

[1. Mögliche Umsetzung: 2](#_Toc31810884)

[1.1. Software: 2](#_Toc31810885)

[1.2. Basisstation: 3](#_Toc31810886)

[1.3. Messkarten: 3](#_Toc31810887)

[2. Bussystem 4](#_Toc31810888)

[3. Spannungsversorgung 5](#_Toc31810889)

[4. Zeichnung 6](#_Toc31810890)

# Zweck/Kernfunktionalität:

Die Messbox soll Messdaten(z.B Spannung, Temperatur) erfassen und diese graphisch anzeigen lassen können und abspeichern. Dabei soll diese ortsungebunden Messdaten aufnehmen können.

# Mögliche Umsetzung:

# Software:

Die graphische Anzeige wird über einen Pi(zero)(ca.20€) realisiert. Dieser hat usb für Maus und mini HDMI für einen Bildschirm.

Die zu erstellende Software auf dem pi soll Auswahl und Konfiguration der Messstationen über ein GUI und die graphische Anzeige der Messwerte ermöglichen.

Auch sollen die Messwerte und graphische Anzeige dieser abspeicherbar sein(png, csv).

Die Software hat mehrere Mess-Modi: kontinuierliche Messung, start-stop-Messung:

Kontinuierliche Messung: Anzeige der letzten aktuellen Daten, die vom pi empfangen wurden(überschreiben), Steckkarten übertragen Daten in Paketen. Bspw. werden die Messwerte der letzten 10s angezeigt. Maximale Buffergröße (z.B 10kS) als Ringbuffer. (Nach 10kS wird der erste Messwert gelöscht (Sonst würden nicht ausgelesene Werte immer mehr Speicher verbrauchen))

Start-Stop Messung: Start/Stop der Messung über Taster auf Platine, oder Zeitraum. Es werden alle Messwerte kontinuierlich gespeichert; So kann auch im Nachhinein entschieden werden, welcher Kanal ausgewertet werden soll. Ist der Buffer voll, werden keine weiteren Messwerte aufgenommen.

Zusätzlich könnte man über einen zweiten Button nachdenken, der alle Parameter der Karten ausliest und abspeichert, um absolut headless arbeiten zu können.

Die Software soll ermöglichen, die Daten miteinander zu verrechnen und anzuzeigen(z.B Differenzbildung/Multiplikation), wobei die Achsen automatisch skalieren. Ob das ohne Tastatur möglich ist, muss sich zeigen.

Die Samplefrequenz der Steckkarten soll schrittweise wählbar sein(z.B 1Hz, 100Hz, 1kHz, 10kHz).  
Somit ist der Benutzer verantwortlich, den Messzeitraum geeignet zu wählen.

# Basisstation:

Die Basisstation soll Platz für 5 Steckkartenplätze und 5 RJ12 Buchsen bieten, wobei über die RJ12 Buchsen analog zu den Steckkartenplätzen Steckkarten betreiben können sollen.

Die Software des Raspberry Pi soll in C++ geschrieben werden. Für die graphische Oberfläche soll QT verwendet werden.  
Für QT gibt es ein Educational Programm. Dabei kann der Professor für die Studenten kostenfreie Vollversionen beziehen. (<https://www.qt.io/qt-for-educational-program> )

Die Kommunikation der Basistation(pi) mit Streckkarten soll über SPI erfolgen. Dabei hat jeder Steckplatz und jede RJ12 Buchse eine eigene Slave-Select Leitung, über die der Raspberry erkennt, ob eine Karte angeschlossen ist. (kein Gerät: leichter Pulldown, Gerät angesteckt: Pullup, Kommunikation: GPIO: LOW)

# Messkarten:

Um die Messungen unabhängig voneinander durchführen zu können, sollen die Steckkarte galvanisch von der Basisstation getrennt sein.

Die galvanische Trennung der Kommunikation soll über Optokopler erfolgen, die sich auf der Steckkarte befinden.

Um eine flexible Einsetzbarkeit der Messbox zu gewährleisten, soll die Basisstation mit Steckkarten über Lithium-Ionen-Akkumulator betrieben werden(ACT2801QL-T1028). Das Laden dieser muss ebenfalls galvanisch getrennt erfolgen.

Der Akku der Basisstation soll über eine DC-Buchse geladen werden können. Angedacht ist eine oder mehrere Standard Li-Ionen Zelle.

Der Akku der Messkarte soll ebenfalls über die Basisstation geladen werden können. Es wird sich dabei vermutlich um 2023-Knopfzellen handeln.

Um die galvanische Trennung aufrecht zu erhalten, benötigt es hier einen galvanisch getrennten DC-DC-Konverter. Dieser befindet sich in der Basisstation

Nur eine Messkarte kann gleichzeitig geladen werden, um die galvanische Trennung unter den Karten aufrecht zu erhalten. So soll die Verbindung zur Basisplatine über ein Relais getrennt werden können.

Die Software für die Steckkarten wird in C++ geschrieben

Die Gehäuse sollen in 3D-Druck erstellt werden

Für die Steckkarten soll als uC der STM32L431CB verwendet werden. Dieser bietet genug Schnittstellen für RAM, Kommunikation und Debugging, sowie eine FPU und zwei DACs, die zukünftig von Vorteil sein könnten.

Eine Status-LED ist in den Steckkarten zu verbauen, die anzeigt, wenn der Buffer der Karten voll ist. (im kontinuierlichen Modus, wenn in der letzten Übertragung der gesamte Buffer voll war; im Start/Stop Modus, wenn der gesamte RAM voll ist.)

Im Zuge des Projekts soll 1xSteckkarte für Spannungsmessung und 1xSteckkarte für Widerstandsmessung/Temperaturmessung aufgebaut werden.

# Bussystem



# Spannungsversorgung



# Zeichnung

