

MultiMeasureSys

Funktion

Das MultiMeasureSys soll einfachen Zugang zu Sensordaten und deren Aufzeichnung auf SD-Karte ermöglichen. Zur Kommunikation mit den Sensoren stehen fünf Schnittstellen bereit: I2C, SPI, oneWire, Analoginput (verstärkt), Interrupteingang (vorwiegend zum Zählen elektrischer Pulse). Vielfältige Anwendungen sind denkbar – umgesetzt wurden:

- Powermeter,
- Hell-Dunkel-Wechsel Zähler,
- Temperaturlogger,
- Zähler für Binärdetektoren.

Performance

Der Zähler für Binärdetektoren kann bis etwa 140kHz betrieben werden, dabei ist das beschränkende Element die Darstellung auf dem Display. Bei Ausgabe über UART könnte mehr drin sein.

Als Temperatursensor verwende ich einen DS18B20 in einem wasserdichten Housing. Die Unsicherheit ist mit $\pm 0,5^{\circ}\text{C}$ im Bereich von -10°C bis 85°C angegeben.

Datum

Beginn des Projekts: August 2018

Status

Eine Kleinserie wurde umgesetzt. Stencil ist vorhanden. Mit etwas Übung können in drei Stunden vier Geräte gefertigt werden.

Entwickler

Kim Weber, weber@iqo.uni-hannover.de

Anwender

- FoexLab
- PhysikPraktikum Versuche B04 spez. Wärme, D02 Interferometer

Schaltungsprinzip

Arduino als Herzstück: Im Zentrum des Projekts steht der Arduino Nano. Die Platine ermöglicht die Interaktion mit dem Nutzer durch vier Pushbuttons und ein Display. Weiter werden die Schnittstellen komfortabel bereitgestellt, die zur Kommunikation mit zahlreichen Sensoren genutzt werden können.

Analogeingang: Der Analogeingang ist dafür ausgelegt einen Photodiodenstrom in eine Spannung zu Wandeln. Hierzu wird ein Transimpedanzverstärker genutzt. Ein Rail-to-Rail Verstärker ermöglicht eine Dynamik bis (fast) an die Grenzen der Versorgungsspannung: Hinter dem ADC des Arduinos sehe ich minimal den Wert Null und maximal den Wert 1022 (von 1023).

Taster: Die Taster sind mittels Schmitt-Trigger-Inverter (74HC14) und RC-Gliedern hardwareseitig entprellt.

Display: Das Display wird über einen i2C Portexpander (PCF8574) angeschlossen.

SD-Karte: SD-Karten können direkt über SPI gelesen und beschrieben werden. Wichtig ist hierbei ein Levelshift auf 3V3. Dieser kann für den Input zur SD-Karte passiv (Spannungsteiler 2k2-3k3) vorgenommen werden. Für den Output der SD-Karte benötigt man keinen Shift, da der Arduino 3V3-sensitiv ist. Die Versorgung der SD-Karte mit 3V3 wird über einen Pin am Arduino bereitgestellt.

Schaltplan

- Der [Schaltplan](#) im PDF-Format
- Die Source des Schaltplans ist auf der [Download-Seite des Wiki](#) abgelegt.

Layout

- Abmessungen der Leiterplatte:  **Fix Me!**
- Versorgung:  **Fix Me!**
- Eingang:  **Fix Me!**
- Ausgang:  **Fix Me!**
- Anzeigen:  **Fix Me!**
- Der Bestückungsdruck: [multimeasuresys_layout.pdf](#)
- Die Bestückungsliste: [multimeasuresys_bom.pdf](#), [multimeasuresys_bom.xls](#)
- Die [gezippten Gerberdaten](#) für die Bestellung der Platine
- Die Source des Layouts im pcb-Format liegt auf der [Download-Seite des Wiki](#).

 **Fix Me!**

Gehäuse



Test



Bedienung



Bilder

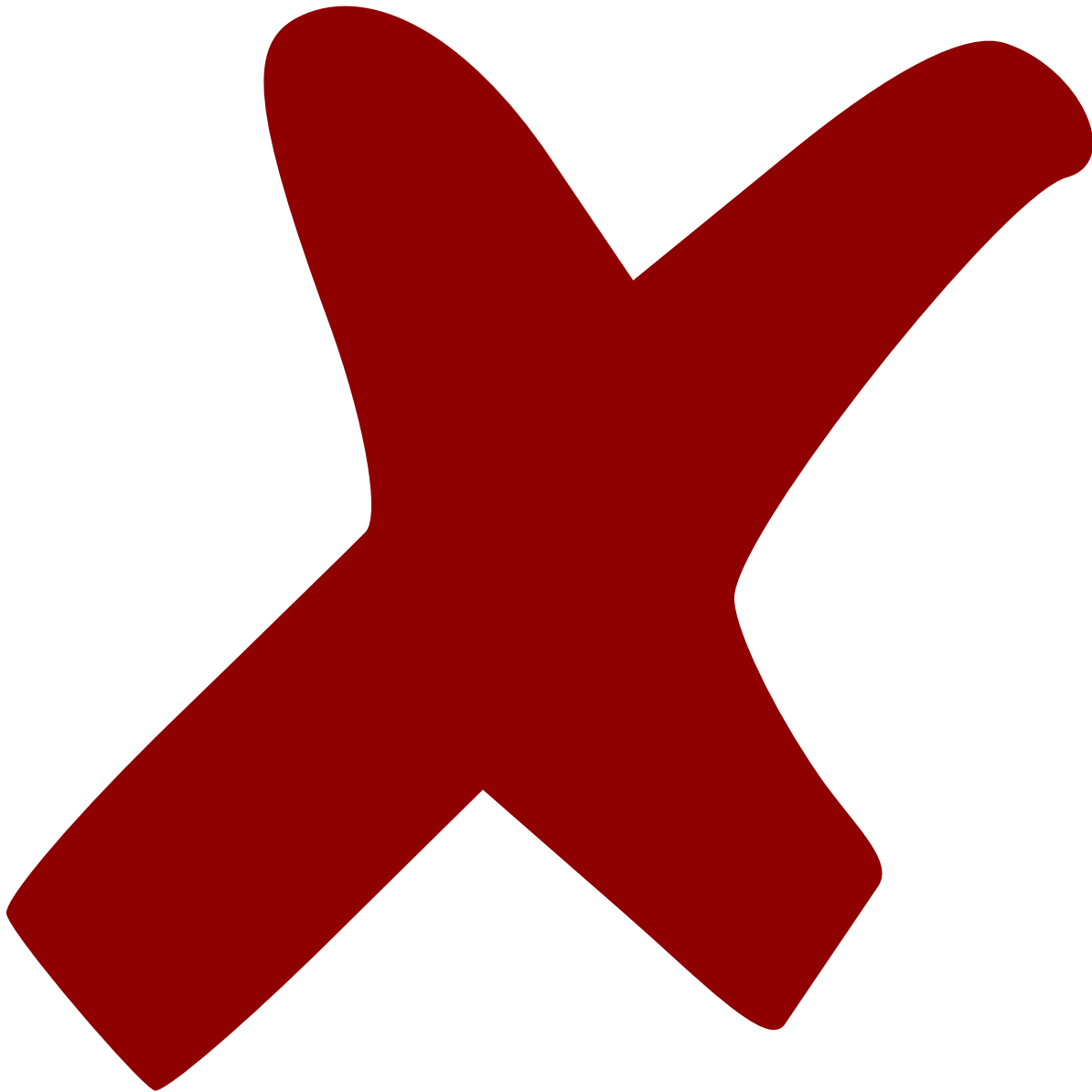
Kalkulation



was	wieviel	E-Preis	Preis	Anmerkung
Leiterplatte	1x	??.?? €	€	1/n von XXX EUR
Gehäuse	1x	??.?? €	€	
*	?x	??.?? €	€	...
R,C	??x	0.02 €	€	Bauform 0805
Bestückung		??.00 €		bei SRM
Verschnitt		?.?? €		
Summe			€	

Meckerliste

Was für die nächste Version zu tun ist: (



: verworfen,



: in Arbeit,



Schaltplan, aber noch nicht im Layout,

: im



: erledigt)

From:

<https://elektroniq.iqo.uni-hannover.de/> - **ElektronIQ**

Permanent link:

<https://elektroniq.iqo.uni-hannover.de/doku.php?id=eigenbau:lehrmittel:multimeasuresys&rev=1536609680>

Last update: **2018/09/10 20:01**

