

Embedded Systems WS 20/21

Team-Projekt

Puls-Oxymeter v1.0

Teammitglieder: Kevin Schultz
Sebastian Hofmaier
Andreas Roth

22.02.2021

Abstract

Deutsch

Kommunikationssysteme wie Smartphones und Smartwatches sind in den letzten Jahren sehr beliebt geworden. Viele von ihnen verfügen auch über Gesundheitsüberwachungsfunktionen wie die optische Puls- und Sauerstoffmessung.

In diesem Projekt werden der menschliche Puls sowie die Sauerstoffsättigung des Blutes mit einem optischen Sensor gemessen und anschließend auf einem LCD visualisiert.

In den folgenden Kapiteln wird zunächst auf den medizinischen Hintergrund und das optische Messsystem eingegangen. Danach wird der Entwicklungsprozess der Hard- und Software genauer erläutert.

Am Ende werden die gewonnenen Ergebnisse gezeigt und ein Fazit gezogen.

English

Communication systems, such as smartphones and smartwatches have become quite popular over the past few years. Many of them also feature health-monitoring functionality such as optical pulse and oxygen measurement.

In this project, the human pulse as well as the oxygen saturation of the blood is measured using an optical sensor and then finally visualized on an LCD-Display.

In the following chapters, the medical background and the optical measurement system will be discussed first. Afterwards the development process of hardware and software are explained.

In the end the results are shown and a conclusion is given.

Inhalt

Abstract	2
Abbildungsverzeichnis	4
Tabellenverzeichnis	5
Abkürzungsverzeichnis	6
1. Einleitung	7
2. Funktionsweise Pulsoxymetrie	10
3. Requirements Engineering	13
4. Hardwareentwurf	16
5. Softwareentwurf	26
6. Inbetriebnahme und Ergebnisse	35
7. Fazit und Ausblick	41
Literatur	43
Anhang	44

Abbildungsverzeichnis

Abb. 1: Eigenaufnahmen TM4C129EXL Crypto Connected Launchpad (1), Educational Boosterpack MK II von Texas Instruments(2)	8
Abb. 2: Messprinzip (1), Auslöschungskoeffizient auf Wellenlänge (2)[1]	11
Abb. 3: Zusammenhang zwischen der Ratio "r" und der Sauerstoffsättigung (SpO2) [1]	12
Abb. 4: Blockschaltbild des Systemaufbaues	16
Abb. 5: Blockschaltbild der Hardwarestruktur und Signalverläufe	17
Abb. 6: Steckverbinder zum Microcontrollerboard	18
Abb. 7: Steckverbinder zum Nellcor-DS100A Fingerclipsensor	19
Abb. 8: Analog-Schalter zur zeitlichen Ansteuerung beider LED's des Fingersensors	19
Abb. 9: Digital-Analog-Converter zur Regelung des LED-Stromes und zum DC-Offset-Abgleich der Verstärkerschaltung	20
Abb. 10: Verstärkerschaltung des Ausgangssignales der Photodiode im Fingerclipsensor	20
Abb. 11: Platinenlayout der Erweiterungsplatine des Pulsoxymeter	21
Abb. 12: Nellcor-DS100A Fingerclipsensor	22
Abb. 13: Selbstbausensoren mit zwei LEDs und einer Photodiode	22
Abb. 14: LTspice-Simulation der Ausgangswerte der Signalaufbereitungsschaltung	24
Abb. 15: fertige Erweiterungsplatine - montiert auf Microcontrollerboard	25
Abb. 16: Übersicht der Software-Architektur	27
Abb. 17: Flow-Chart des Messverfahrens	29
Abb. 18: State-Maschine	31
Abb. 19: Oszillogramm der PWM Signale für rote (rot) und infrarote (blau) LED	36
Abb. 20: Oszillogramm I2C Bus; SDA(blau), SCL(rot)	37
Abb. 21: Oszillogramm ADC0-Eingang (blau; rot ohne Bedeutung)	38
Abb. 22: Oszillogramm Spannungssignal an LED rot zu Masse	39
Abb. 23: Oszillogramm RDY/BS (rot), DAC_A (blau), DAC_B (grün)	40

Tabellenverzeichnis

Tab. 1: Projektspezifische Anforderungen	14
Tab. 2: Projektspezifikationen	15
Tab. 3: Materialliste der elektronischen Bauelemente für die Erweiterungsplatine	23
Tab. 4: Software-Events.....	30

Abkürzungsverzeichnis

SpO2	Sauerstoffsättigung
ADC	Analog Digital Converter
LCD	Liquid Crystal Display
Hb	Hämoglobin
HbO ₂	Hämoglobin, sauerstoffgesättigt
LED	Light-Emitting Diode
DC	Gleichspannung
AC	Wechselspannung
I2C	Inter-Integrated Circuit
DAC	Digital Analog Converter
VCC	Voltage at the common collector
IR	Infrarot

1. Einleitung

Pulsoxymetrie

Heutzutage ist in der ambulanten medizinischen Versorgung von Patienten das Verfahren der Pulsoxymetrie kaum noch wegzudenken. Mit Hilfe dieses optischen Verfahrens können mehrere Vitalwerte des Menschen nicht-invasiv überwacht werden: Sauerstoffsättigung des Bluts, Herzschlag- sowie Atemfrequenz. Die Pulsoxymetrie ist hierfür der gängige Standard in der Medizin.

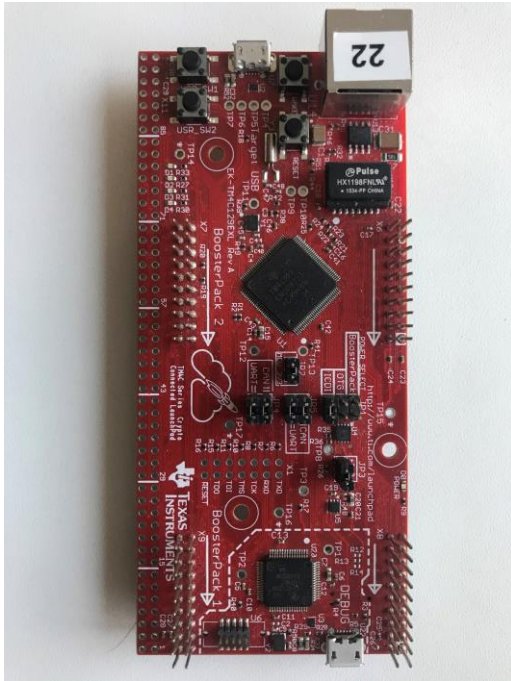
Die hierbei erzeugten Daten zum Patientenzustand sind vor allem im momentanen Kontext der Corona-Pandemie unverzichtbar. Bei schweren Verläufen dieser Krankheit reichert sich das SARS-COV-2 Virus in den Atemwegen sehr stark an. Die dadurch verminderte Sauerstoffaufnahme durch die Lunge, kann schnell lebensgefährlich werden. Über den SpO2-Wert ist man in der Lage – effizient – Rückschlüsse über die Funktionsfähigkeit der Lunge zu ziehen.

Auch im alltäglichen Leben wie im Freizeitsport findet das Verfahren Anwendung, beispielsweise in Fitnessuhren oder integriert in SmartWatches.

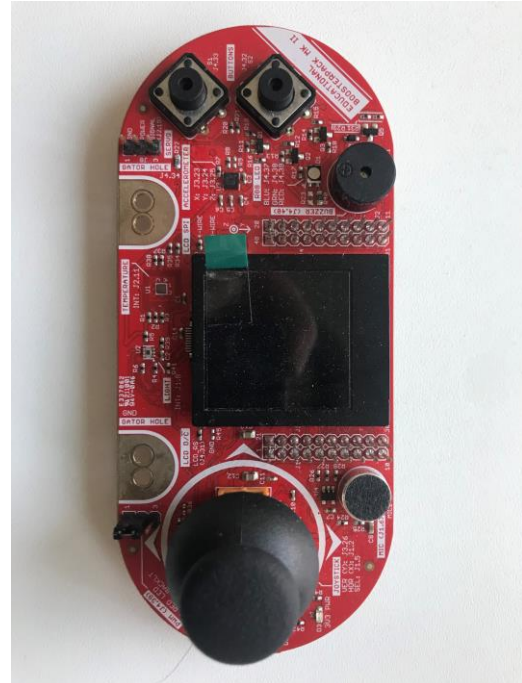
Aufgabenstellung

Diese Arbeit erfolgt im Rahmen des Kurses „Embedded Systems“ im Wintersemester 2020/2021 des Masterstudiengangs „Elektrische Systeme“. Aufgabe ist es, ein Pulsoxymeter als eingebettetes System zu entwickeln.

Zur Verfügung gestellt werden die Entwicklungsboards „TM4C129EXL Crypto Connected Launchpad“ sowie „Educational Boosterpack MK II“ von Texas Instruments (siehe Abb. 1).



(1)



(2)

Abb. 1: Eigenaufnahmen TM4C129EXL Crypto Connected Launchpad (1), Educational Boosterpack MK II von Texas Instruments (2)

Die oben zu sehenden Boards haben schon diverse Funktionalität enthalten, wie unter anderem...

- ...mehrere integrierte ADCs
- ...Buttons oder einen Joystick als mechanische Eingabeelemente
- ...einen LCD oder eine Beeper zur Ausgabe

Für die gesamte Entwicklung soll zunächst eigenständig eine Anforderungsanalyse abgeleitet werden. Im Anschluss gilt es eine zusätzliche Hardware zu entwerfen, zu dimensionieren und zu bestücken – passend für die gegebenen Boards von Texas Instruments. Des Weiteren ist eine Firmware zur Auswertung und Steuerung der Messungen für den Mikrocontroller „Tiva™ TM4C129ENCPDT“ in der Hochsprache C zu

programmieren. Als Anleitung zur Inbetriebnahme soll ein Quick Start Guide dienen. Abschließend dient diese Arbeit als Dokumentation.

Zielsetzung

Ziel dieser Ausarbeitung ist mit einem optischen Sensor ein analoges Signal zu generieren, über eine geeignete Signalaufbereitung zu verarbeiten und anschließend digital zu verwerten. Durch die Erarbeitung einer passenden Softwarearchitektur sollen Maßnahmen zur Programmrobustheit sowie Performance ermöglicht werden. Die errechneten Werte zu Sauerstoffsättigung und Puls werden über den LCD auf dem Erweiterungsboard ausgegeben.

Strukturierung

Zu Beginn gibt das Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** einen kurzen Einblick in die theoretischen Hintergründe sowie zum Stand der Technik hinsichtlich des Pulsoxymetrie-Verfahrens.

In Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** wird die abgeleitete Anforderungsanalyse vorgestellt. Es wird geklärt welche Projektanforderung sich für Hardware wie Software ableiten sowie welche Use Cases, Funktionsweisen und Testing-Methoden es gibt. Zum Schluss wird ein Lösungskonzept aufgezeigt.

Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** geht auf die Entwicklung der elektronischen Schaltung ein und beschreibt den erarbeiteten Hardwareentwurf.

Die Beschreibung der Softwarearchitektur sowie der einzelnen Softwaremodule ist in Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** finden.

Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** widmet sich der Inbetriebnahme der Hardware sowie der Softwareintegration. Es werden die durchgeführten Messungen und Tests als Ergebnisse dargelegt.

Im Anschluss werden in Kapitel 0 ein Fazit gezogen und ein Ausblick für eine mögliche Projektfortsetzung gegeben.

2. Funktionsweise Pulsoxymetrie

Theoretische Grundlagen

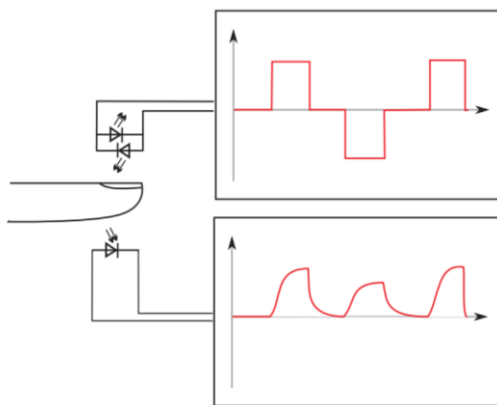
Das optische Pulsmessverfahren ist für die Anwendung des Pulsoxymeters gut erforscht. Das grundlegende Prinzip, welches ausgenutzt wird, ist die unterschiedlich Absorption von Licht durch das menschliche Gewebe.

Für eine zeitliche Änderung der Lichtabsorption sorgt der Unterschied zwischen dem venösen und dem arteriellen Gefäßsystem im Blutkreislauf. Das arterielle System ist im Gegensatz zum Venösen großvolumig und von verhältnismäßig viel Muskulatur umgeben, was hohe Elastizität mit sich bringt. Dies ist die Bedingung für Pulswellen. Bei einem Herzschlag wird nun diese Pulswelle an Blut durch die Arterien gedrückt, welche auf diesem Wege den gesamten Körper mit Blut versorgt und dadurch mit Sauerstoff durch Hämoglobin (Hb). Dieses Protein, welches dem Blut die rote Farbe verleiht, absorbiert je nach Sauerstoffsättigung mehr oder weniger rotes bzw. infrarotes Licht. Aus diesem Verhältnis kann der Sauerstoffgehalt des Bluts berechnet werden, wie im Folgenden beschrieben wird.

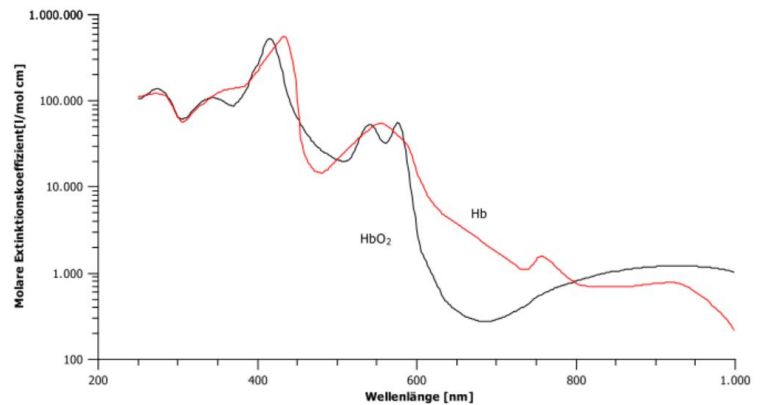
Technische Umsetzung

Durch das Anbringen zweier LEDs mit rotem bzw. infrarotem Licht und einer gegenüberliegenden Photodiode in Sperrrichtung, die sensitiv für die entsprechenden Spektralbereiche ist, können zwei verschiedene Spannungsverläufe generiert werden. Die Photodiode wird hierbei als Stromquelle genutzt. Die LEDs werden vertauscht gepolt mit einer Rechteckspannung betrieben und erzeugen so unterschiedliche Ströme durch die Photodiode. Durch eine Verstärkungsstufe wird das Stromsignal in ein Spannungssignal umgewandelt und aufbereitet (siehe Abb. 2.1).

Diese Spannungen bestehen aus einem DC-Offset, der durch festes Gewebe und dem venösen System erzeugt wird, sowie aus einem AC-Anteil durch das arterielle System. Durch die sich unterscheidenden Absorptionseigenschaften (siehe Abb. 2.2) von rotem und infrarotem Licht bei Sauerstoff-gesättigtem (HbO_2) und -ungesättigtem (Hb) Blut werden unterschiedlich große Spannungen erzeugt.



(1)



(2)

Abb. 2: Messprinzip (1), Auslöschungskoeffizient auf Wellenlänge (2)[1]

Durch das Lambert-Beer'sche Gesetz kann die folgende Formel zur Berechnung der Sauerstoffsättigung hergeleitet werden 43:

$$S_{O_2} = \frac{\epsilon_{R,Hb} - \epsilon_{IR,Hb} \cdot r}{(\epsilon_{R,Hb} - \epsilon_{R,HbO_2}) - (\epsilon_{IR,Hb} - \epsilon_{IR,HbO_2}) \cdot r} \quad (1)$$

Der Molare Extinktionskoeffizient „ ϵ “ ist für die jeweilige Kombination von Wellenlänge und Sauerstoffsättigung konstant und kann vom oben gezeigten Graphen abgelesen werden (siehe Abb. 2: Messprinzip (1), Auslöschungskoeffizient auf Wellenlänge (2)[1].2).

Das Verhältnis „ r “ der DC- und AC-Anteile des roten respektive des infraroten Signals in (1) kann wie folgt gebildet werden 43:

$$r = \frac{U_{AC,R} \cdot U_{DC,IR}}{U_{DC,R} \cdot U_{AC,IR}} \quad (2)$$

Für den Einsatz im Pulsoxymeter kann bei errechnetem „r“ die Sauerstoffsättigung direkt von dem Graphen in Abb. 3 entnommen werden. Der dargestellte Zusammenhang wurde laut Quelle 43 experimentell ermittelt.

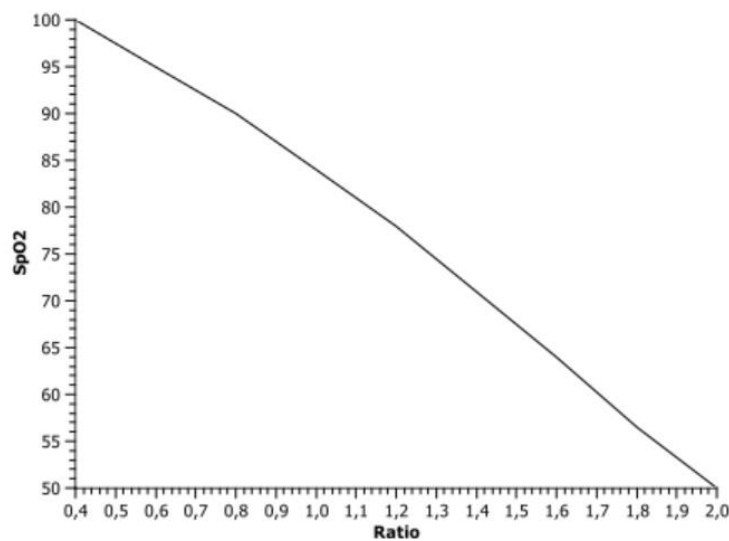


Abb. 3: Zusammenhang zwischen der Ratio "r" und der Sauerstoffsättigung (SpO2) [1]

Neben der bereits vorgestellten Transmissionsmethode gibt es auch Lösungen durch eine reflexive Signalerfassung, bei welcher die LEDs und die Photodiode auf der gleichen Seite angebracht sind.

3. Requirements Engineering

Anforderungen und Lastenheft

Die bereits genannten Anforderungen aus Einleitung werden hier nochmals kurz wiederholt. Diese stellen die Basis dar für dieses Projekt und dienen zu einer eigenständigen Ableitung für ein entsprechendes Pflichtenheft.

Allgemein:

Ziel dieser Arbeit ist es in einem vollumfänglichem Projekt ein Pulsoximeter zu entwerfen. Zur Verfügung gestellt werden die Entwicklungsboards „TM4C129EXL Crypto Connected Launchpad“ sowie „Educational Boosterpack MK II“ von Texas Instruments. Hinsichtlich Sensorik steht uns ein Nellocoor-DS100A zu Verfügung. Optional kann ein Selbstbausensor ebenfalls verwendet werden.

Zielbestimmung

Anforderungen selbstständig ableiten, Hardware selbst entwickeln und eine geeignete Software-Architektur zu konzipieren. Die Ergebnisse der Arbeit sollen in Form einer Dokumentation und einer Abschlusspräsentation zusammengefasst werden.

Produkteinsatz

Der Anwendungsbereich soll stets nur innerhalb des Projekts und nicht für medizinische Zwecke verwendet werden. Als Zielgruppen kommen daher nur die Kursteilnehmer sowie der Professor selbst in Frage. Hierfür soll ein Quick Start Guide geschrieben werden.

Rahmenprojektanforderungen

Softwareseitig soll in der Programmiersprache C programmiert werden in Anlehnung an den Coding-Standard "Clean-Code". Zur Umsetzung des Projekts wird kein Betriebssystem verwendet. Der Speicher für die Messwerte und deren Verarbeitung wird hierbei nach dem Prinzip eines FIFOs organisiert. Die Dokumentation des Projekts sowie der Quick Start Guide sollen in Deutsch erstellt werden. Der Betrieb soll in Schutzkleinspannung ermöglicht werden.

Anforderungsanalyse

Auf Basis des Lastenhefts wurden projektspezifische Anforderungen entwickelt. Einen Auszug hierüber ist in Tab. 1 **Fehler! Verweisquelle konnte nicht gefunden werden.**dargestellt. Diese geben aus technischer Sicht einen detaillierten Ausblick auf die geplante Realisierung dieser Arbeit und bilden die Grundfunktionalitäten an. Ziel dieses Projektes ist es jeder dieser Anforderungen mit jeweils mind. einer Spezifikation zu beschreiben, diese zu implementieren und zu testen. Hierbei wurden aus organisatorischen Gründen jeder Spezifikation eine ID zugewiesen die als Referenz für die Abdeckung der einzelnen Punkte dient.

Req. Nr.	Beschreibung	Abdeckung
1	Ansteuerung, Messdatenverarbeitung und Aktorik soll über eine zentrale Steuereinheit erfolgen	SWF1,HW1
2	Messung soll über eine zusätzlich ansteckbare Platine erfolgen	SWF1,SWF2,SWF22,HW6,HW8
3	Schnittstelle für Sensorik soll auf der Messplatine bereit gestellt werden	SWNF5,HW7
4	Digitale Signalverwertung des gesampelten analogen Messsignals	SWF4,SWNF2,HW6,SWF4,SWNF3,HW9,SWF16,SWF17,SWF18
5	Zuordnung der Messsignale zu den jeweiligen LEDs	SWNF7,SWNF6,SWNF8
6	Der Herzschlag soll detektiert werden	SWF4,SWNF3
7	Detektierung ob ein Finger erkannt wird	SWF2,SWF3,SWF10,SWF13
8	Dynamische Ansteuermöglichkeit des Sensors zur Reduzierung von Störeinflüssen	SWF13,SWF21,HW11,HW13
9	Eine geeignete Initialisierungsroutine soll die Einheit nach dem einschalten betriebsbereit machen	SWNF12
10	Aktualisierung der Displayanzeige in Abhängigkeit des Herzschlags	SWF7,SWF8,HW2
11	Herzschlag akustisch und optisch auf dem Display darstellen	SWF7,SWF8,HW2,HW4
12	Ausgabe der Messwerte in Abhängigkeit der aktiven Betriebsart	SWF2,SWF10,SWF11,SWF4,SWF7,SWF12,HW4
13	Zwischenspeicherung der Werte während der Messung	SWF5,SWNF4,SWF12,SWNF6,SWNF7,SWNF8
14	In Betriebsart "aktiv" befindet sich der Controller sofern keine Fehlerzustände eintreten und ein Finger am Sensor detektiert wird	SWF3,SWF4,SWNF2,HW6,SWF4,SWNF3
15	In Betriebsart "inaktiv" befindet sich der Controller wenn über 5s kein Finger detektiert wird	SWF9
16	Betriebsart Energiesparmodus ("Eco") soll bereit gestellt werden falls kein eindeutiges Signal über 10s detektiert wird	SWF12,SWF11,SWF9
17	In der Betriebsart "Error" soll der entsprechende Fehlerzustand (Timeout, Overflow, HW, Variablenzuweisung, Watchdog,...) über das Display ausgegeben werden	SWF2,SWNF8,SWF14,SWF15
18	Nach 10s soll im Modus "Error" ein Reset durchgeführt werden	SWF20
19	Es soll sicher gestellt werden, dass die Messfrequenz stets kleiner als die benötigte Verarbeitungszeit ist	SWF16,SWF17,SWF18
20	Implementierung eines Watchdogtimer, der zyklisch zurückgesetzt werden soll, ansonst in Betriebsart "error" und nach 10s HW Reset	SWF14,SWF20
21	Es sollen Funktionen implementiert werden zur Sicherstellung gegen einen Speicherüberlauf	SWNF8
22	Auslesung der zwischengespeicherten Daten und Ermittlung der Sauerstoffsättigung	SWNF3,SWNF6,SWNF7
23	Berechnung des Pulses	SWNF3

Tab. 1: Projektspezifische Anforderungen

Spezifikationen und Testing

Aus den Grundfunktionalitäten wurden nun Spezifikationen abgeleitet, die diese abdecken sollen und implementiert bzw. umgesetzt werden können. Hierbei wird grundsätzlich zwischen folgenden Kategorien unterschieden:

- Software - Funktionale Anforderungen
- Software - Nicht funktionale Anforderungen

- Hardwarespezifische Anforderungen

Ein Auszug in Tab. 2 zeigt wie die einzelnen Spezifikationen aufgelistet wurden. Hierbei spielte nicht nur die Erfassung aller notwendigen Funktionalitäten der Anforderungsanalyse, sondern auch das Testing eine große Rolle. Hierbei wurde zu jeder Anforderung auch die entsprechende Testkategorie zugewiesen und der Testfall beschrieben. Darüber hinaus wurde dieser Test des entsprechenden Entwicklers ausgeführt und das Testergebnis in das File eingetragen. Die Datei, in der alle Anforderungen enthalten sind, ist dieser Arbeit beigelegt.

Kategorie	Anforderung	Beschreibung	Abdeckung	Verifikation	Test Design	Test Status
DAC	SWF13	Wird ein Finger erkannt sich aber das Signal noch nicht im richtigen Bereich befindet wird über den DAC die Helligkeit der LED nachgeregelt	Test	Finger in der Messklemme positionieren und mit Bewegungen prüfen ob Signalwert abweicht	1. Einheit starten und Messklemme am Finger anbringen 2. Am Display verifizieren das Herschlag und Sauerstoff angezeigt wird 3. Mit dem Finger in der Messklemme wackeln und schauen ob sich die Werte signifikant ändern 4. Ausgangsposition einnehmen und Messklemme mit Tuch bedecken und per Sichtprüfung schauen das sich Werte nicht signifikant ändern	n.A.
Safety	SWF14	Zyklische Zurücksetzung eines Watchdog-Timers um einen Timeout der Software zu verhindern	Debugging	Auskommentierung der Funktion zur Zurücksetzung des Watchdog-Timers und Ausführung eines Resets	1. Auskommentierung der Funktion ROM_WatchdogIntClear() in der State Machine 2. Einheit starten und verifizieren dass nach 5s auf dem Display "SW Fehler: Watchdog! Reset wird ausgeführt...." steht 3. Prüfen ob nach 10 weiteren Sekunden ein Reset durchgeführt wird	PASS

Tab. 2: Projektspezifikationen

4. Hardwareentwurf

In der folgenden Abb. 4 ist ein Überblick des gesamten Systemaufbaues dargestellt. Dieses besteht aus den drei Hauptkomponenten

1. Microcontroller für Spannungsversorgung und Auswertung der Messergebnisse
2. Erweiterungsboard für Ansteuerung und Verarbeitung der Messsignale
3. Sensor: Fingerclipsensor zur Messaufnahme

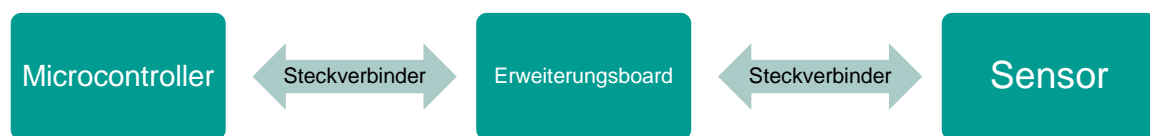


Abb. 4: Blockschaltbild des Systemaufbaues

Der Microcontroller, als erste Hauptkomponente, wurde im Rahmen dieser Veranstaltung vorgegeben, sodass im weiteren Verlauf vor allem die zwei anderen Hauptkomponenten genauer betrachtet werden.

Überblick über Hardwarestruktur und Signalverläufe

Wie im Blockschaltbild in Abb. 5 ersichtlich besteht das Herzstück unseres Systemes der Microcontroller von TM4C129EXL von Texas Instruments. Dieser interagiert über verschiedene Schnittstellen mit den weiteren Hardware-Komponenten.

Durch den Microcontroller werden über integrierte Timerbausteine die Zeitsignale und über einen externen DAC die Intensität der LED's softwareseitig vorgegeben. Die beiden betreiben antiparallel die rote- sowie die IR-LED über eine H-Messbrücke.

Auf der Empfängerseite sitzt eine Photodiode, welche dann die abwechselnden Lichtsignale beider LEDs empfängt und daraus ein entsprechendes Stromsignal erzeugt.

In einer nachfolgenden Signalkonditionierungskette wird das Stromsignal durch einen ersten Operationsverstärker zuerst in ein Spannungssignal umgewandelt. Die anschließende zweite Verstärkerstufe verstärkt dann noch dieses Signal für die Digitalisierung in dem integrierten ADC-Baustein des Microcontrollers.

Der Microcontroller kann dann die Messwerte auslesen und damit die Ergebnisse für den Sauerstoffgehalt im Blut sowie den Puls berechnen.

Die Ausgabe erfolgt dann visuell über eine Anzeige sowie akustisch über einen Beeper auf einer extra Erweiterungsplatine für das TM4C129EXL (Boosterpack MK 2).

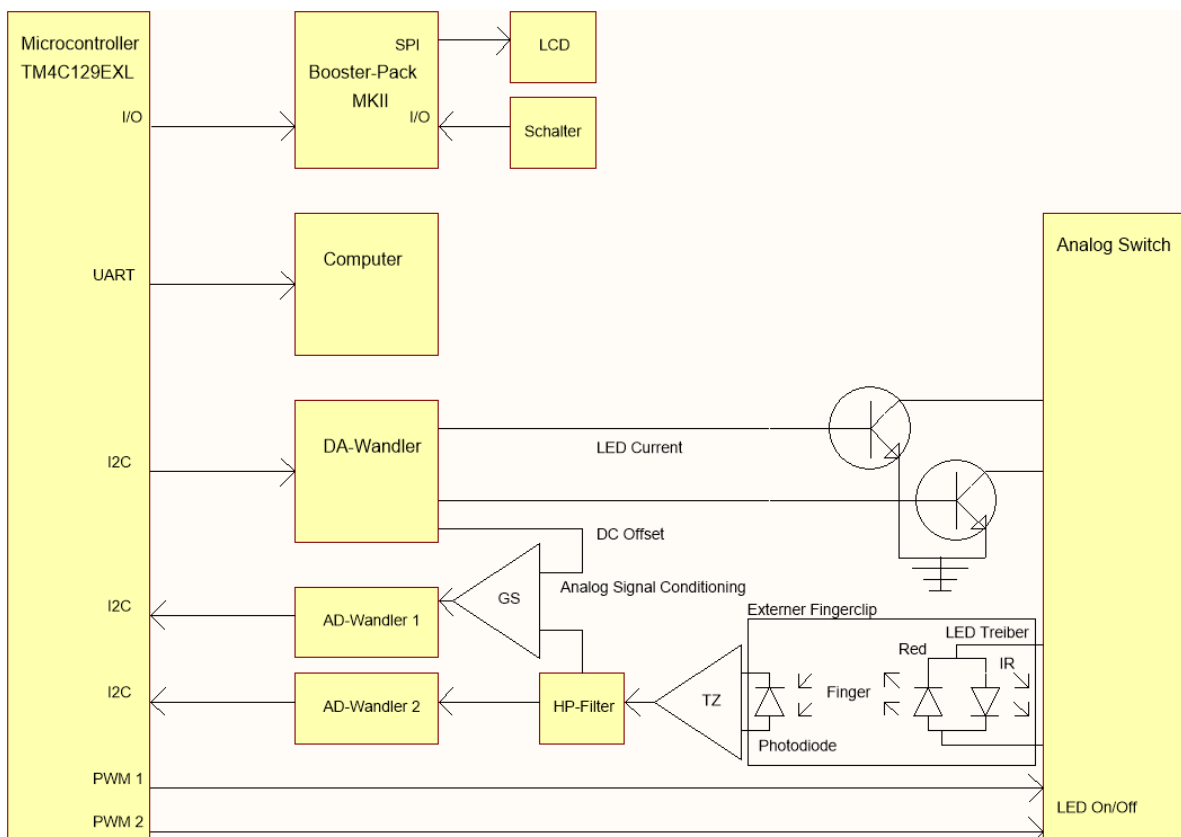


Abb. 5: Blockschaltbild der Hardwarestruktur und Signalverläufe

Detaillierte Beschreibung der einzelnen Schaltungskomponenten

Im nun folgenden Abschnitt wird detaillierter auf die einzelnen Schaltungsteile eingegangen. Dabei wird der Aufbau und die Funktionsweise der Komponenten genauer beleuchtet. Bei der Auslegung der Bauteile orientierten wir uns anhand der gemachten Angaben aus einem Fachartikel von Digikey 43 zum Thema eines einfachen embedded Pulsoxymeters.

Die Spannungsversorgung der Erweiterungsplatine mit dem 3,3 V Pegel wird über Platinensteckverbinder zur Microcontrollerplatine realisiert. Dazu sind weitere notwendige Ports für die PWM-Signale, die Ansteuerung des DACs über den I2C-Bus sowie die beiden Analog-Eingänge für den im Microcontroller integrierten Analog-Digitalconverter vorgesehen.

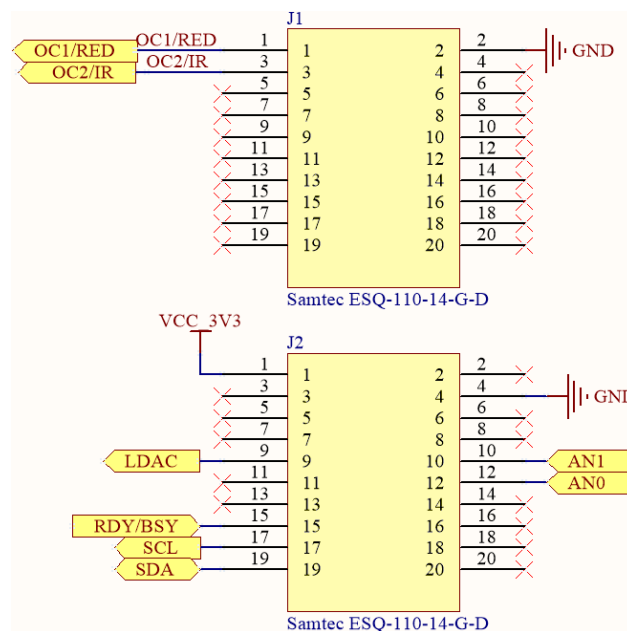


Abb. 6: Steckverbinder zum Microcontrollerboard

Als Schnittstelle zum Fingerclipsensor wird ein Sub-D9 Stecker (Female). Dieser überträgt die Steuersignale für die LEDs an den Sensor sowie das resultierende Signal der Photodiode zur Signalkonditionierungskette.

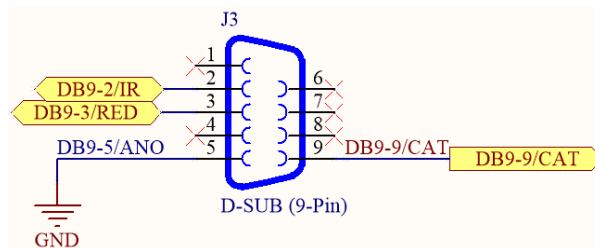


Abb. 7: Steckverbinder zum Nellcor-DS100A Fingerclipsensor

Ein analoger Schalter wird über zwei PWM-Signale des Microcontroller zur genauen Zeitsteuerung der High- und Low-Pegel beider LED's genutzt. Die Ausgänge des Bausteines sind als H-Brücke geschaltet, um die rote- und IR-LED abwechseln ein- und auszuschalten.

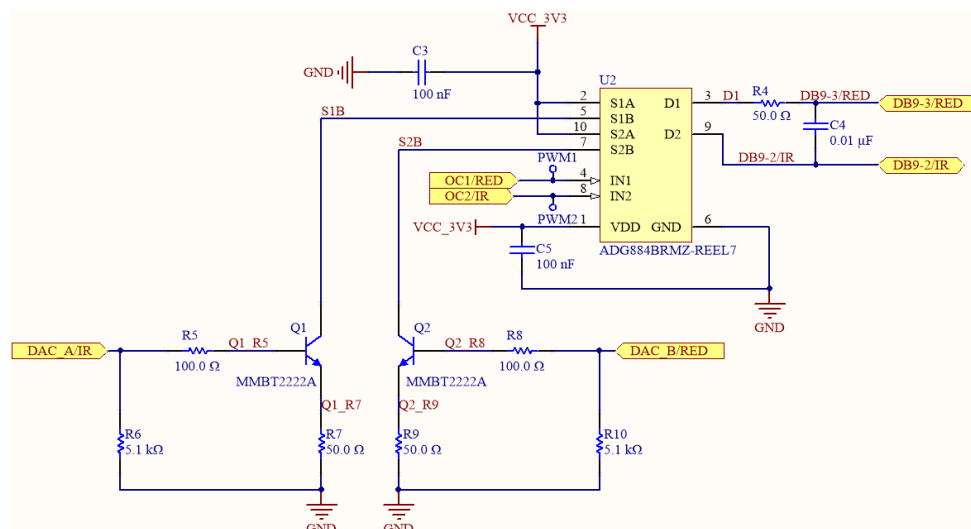


Abb. 8: Analog-Schalter zur zeitlichen Ansteuerung beider LED's des Fingersensores

Dabei wird ein separater Digital-Analog-Wandler (DAC) genutzt, um über zwei Transistoren die gewünschte Intensität (Strom) beider LED anzusteuern. Zusätzlich ermöglicht der DAC einen DC-Offset-Abgleichswert zu setzen, der im zweiten Teil der Signalkonditionierungskette in der Verstärkungsschaltung zur Optimierung des konditionierten Signales einfließt.

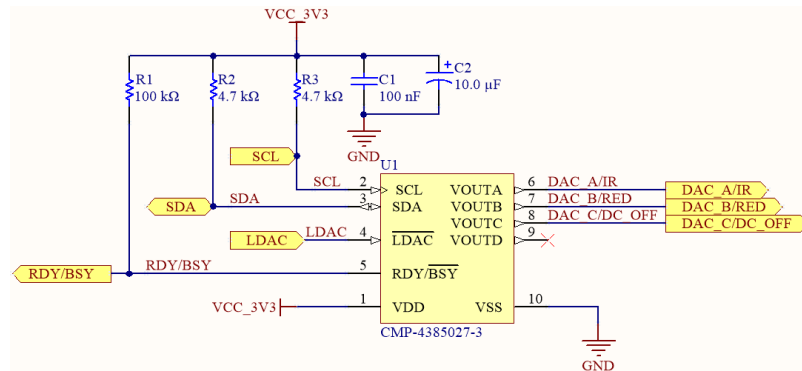


Abb. 9: Digital-Analog-Converter zur Regelung des LED-Stromes und zum DC-Offset-Abgleich der Verstärkerschaltung

Zur Signalverarbeitung am Photodiodenausgang wird ein Baustein mit einem Paar an Operationsverstärkern verwendet. Der erste Operationsverstärker dient dabei als Transimpedanzverstärker zur Umwandlung des Stromes in das benötigte Spannungssignal. Vor dem zweiten OP befindet sich noch eine einfache Hochpasssschaltung, welche eventuell auftretendes Rauschen im Signal unterdrückt. Die zweite Stufe des Bausteines verstärkt, durch den DC-Offsetwert, den gefilterten AC-Anteil des Signales noch final, bevor diese in dem integrierten ADC des Microcontroller ausgewertet wird.

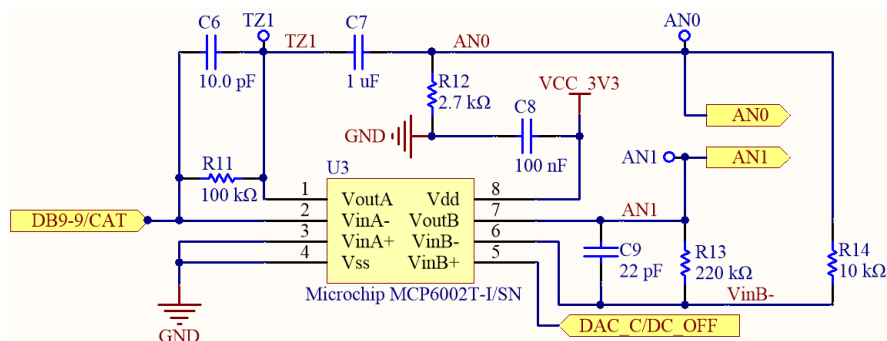


Abb. 10: Verstärkerschaltung des Ausgangssignales der Photodiode im Fingerclipsensor

Layout und PCB-Design

Der Schaltplan und das dazugehörige Layout der Platine wurden mit der Software CircuitMaker (v1.3) des Unternehmens Altium Limited 43 erstellt.

Die Umfangsmaße der Platine betragen dabei 48 mm in der Länge und 30 mm in der Breite. Als Ausführung wurde hierbei eine 2-Layer-Platine ausgewählt, da die Anzahl ausgewählter Komponenten mit ihren Leiterbahnen nicht allzu groß war. Für die Breite der Leiterbahnen von VCC und GND wurden 0,5mm gewählt und für Signalleitungen eine Breite von 0,254 mm.

Als generelle Vorgabe wurde versucht möglichst große Abstände zwischen den einzelnen Signalleitungen zu halten (insbesondere bei Messleitungen und hochfrequenten Signalen wie PWM). An einigen Stellen, wie z.B. den Stiftleisten zum Microcontrollerboard war dies leider nicht immer gegeben, sodass dort im Rahmen der Design-Regeln der Abstand verringert wurde. Für die einfache Signalerfassung zu späteren Testzwecken wurden an einigen Stellen (PWM, AN0/1) Messpunkte in Form von Kupferpads integriert.

Die einzelnen Schaltungsbestandteile wurden nach Kriterien wie Baugruppenzugehörigkeit, Leitungslängen und Größe der Bauelemente auf der Platine platziert und so geordnet das je eine VCC- und GND-Fläche zum gemeinsamen Anbinden eingefügt werden kann (siehe Abb. 11)

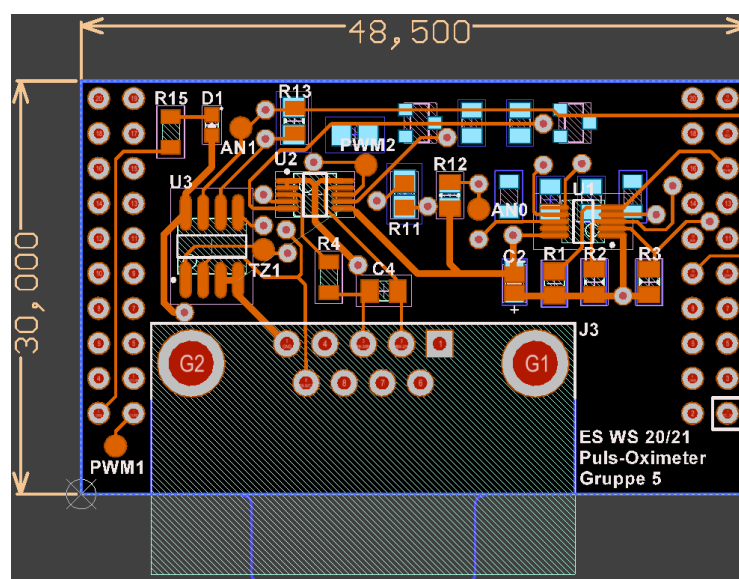


Abb. 11: Platinenlayout der Erweiterungsplatine des Pulsoxymeter

Sensorauswahl

Als Sensor wird ein Fingerclip von Nellcor in der Modellausführung DS100A 43 verwendet (Abb. 12). Die Auswahl fiel auf diesen Sensor aufgrund des günstigen Anschaffungspreises, der einfachen Verwendung und der standardisierten Schnittstelle über eine siebenpolige Version des Sub-D9-Steckers. Die LEDs in dem Sensor sind parallel invertiert geschaltet, sodass diese über unsere H-Messbrücke einfach betrieben werden können.

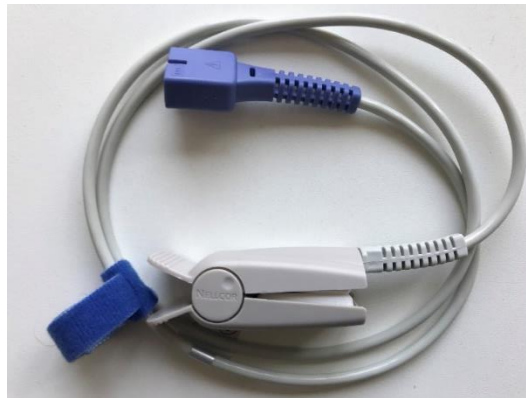


Abb. 12: Nellcor-DS100A Fingerclipsensor

Da mit einer langen Lieferzeit der Sensoren zu rechnen ist, sind als alternative Möglichkeit noch einfache „Selbstbau-Sensoren“ (siehe Abb. 13) angedacht. Diese bestehen in unserer Version aus zwei LEDs in einem Gehäuse und einer Photodiode. Diese Bauteile sind jeweils auf einfache Platinen gelötet und werden über Kabelsteckverbinder mit der Erweiterungsplatine verbunden werden.

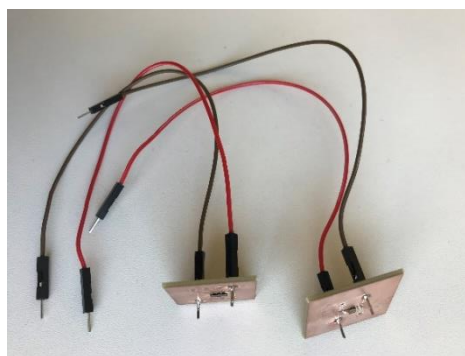


Abb. 13: Selbstbausensoren mit zwei LEDs und einer Photodiode

Materialliste

Alle benötigten elektronischen Bauelemente für die Platine sind in der folgenden Materialliste Tab. 3 aufgezählt.

Nr.	Bezeichner	Kommentar	Bauteilbezeichnung
1	U1	DAC	MCP4728T-E/UN
2	U2	Analog Switch	ADG884BRMZ-REEL7
3	U3	OP	Microchip MCP6002T-I/SN
4	D1	LED	LTST-C190GKT Green 2.1 V 20 mA
5	Q1, Q2	Transistor	MMBT2222A-TP
6	J1, J2	TI Board Connector	SSW-110-23-S-D
7	J3	D-SUB (9-Pin)	WR-DSUB Female PCB 9Pin (618009231121)
8	R4, R7, R9, R15	50.0 O	Dickfilmwiderstand - SMD 1/8 watt 50ohms 1%
9	R1, R5, R8	100.0 O	Dickfilmwiderstand - SMD 100 OHM 1% 1/8W
10	R2, R3	4.7k O	Dickfilmwiderstand - SMD 1/8Watt 4.7 Kohms 5%
11	R6, R10	5.1 kO	Dickfilmwiderstand - SMD 1/8watt 5.1 Kohms 5%
12	R11	100 kO	Dickfilmwiderstand - SMD 1/8Watt 100 Kohms 1%
13	R12	2.7k O	Dickfilmwiderstand - SMD 1/8watt 2.7 Kohms 5%
14	R13	220 kO	Dickfilmwiderstand - SMD 1/8watt 220 Kohms 1%
15	R14	10 kO	Dickfilmwiderstand - SMD 1/8 Watt 10 Kohms 1%
16	C1, C3, C5, C8	0.1 uF	MLCC - SMD/SMT WCAP-CSGP 100V 0.1uF 0805 10%
17	C2	10 uF	Tantalkond. fest für SMD 10uF 6.3V 20% 0805 P Case
18	C4	0.01 uF	MLCC - SMD/SMT 100V .01uF X7R 0805 10%TOL
19	C6	10 pF	MLCC - SMD/SMT WCAP-CSGP 100V 10pF 0805 5%
20	C7	1.00 µF	MLCC - MLCC - SMD/SMT 0805 1uF 10volts X7R 10%
21	C9	22 pF	MLCC - SMD/SMT WCAP-CSGP 22pF 0805 5% 25V

Tab. 3: Materialliste der elektronischen Bauelemente für die Erweiterungsplatine

Simulation

Zur Überprüfung der erwarteten Funktionsweise der Platine wird vor der Fertigung eine Simulation durchgeführt. Dazu wird das PC-Programm LTspice (v17) 43 benutzt. In diesen wird die Ansteuerung der LEDs über die H-Messbrücke sowie die Signalverarbeitung über den Operationsverstärkerbaustein aufgenommen und ausgewertet. In der Abb. 14 sind die Spannungsverläufe an dem Eingang des ADC0 vor und nach dem Hochpass abgebildet. Zur Simulation des Stromsignals der Photodiode bei der Bestrahlung durch die rote LED wurden zwei Konstantstromquellen verwendet. Die eine für den DC- und die andere wiederum für den AC-Anteil. Gleichzeitig wurde ein hochfrequentes Rauschen mitbeaufschlagt.

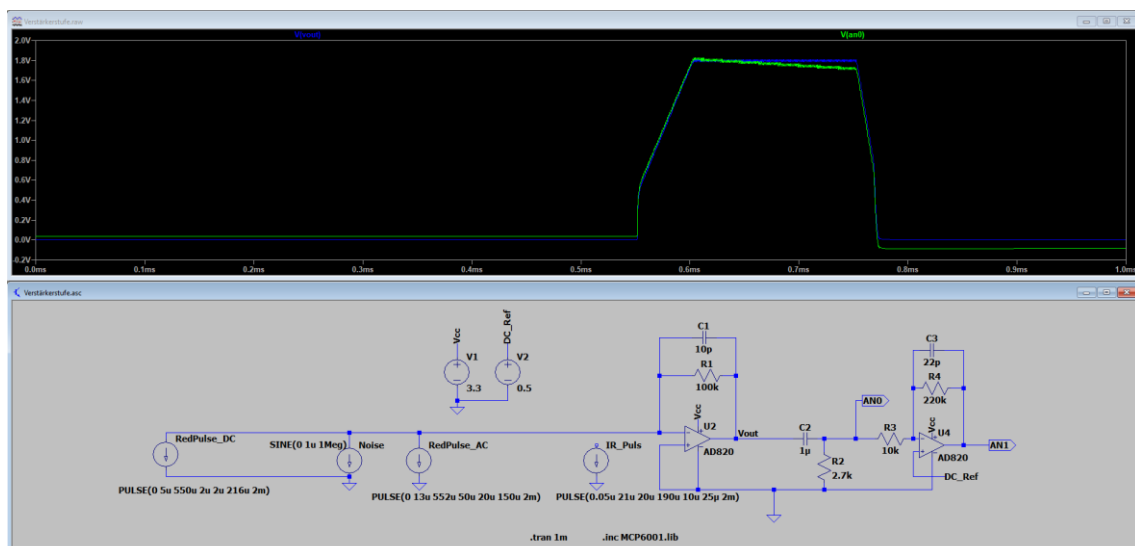


Abb. 14: LTspice-Simulation der Ausgangswerte der Signalaufbereitungsschaltung

Fertigung

Für die Fertigung der Platine werden aus dem Layout die entsprechend benötigten Gerberdaten generiert. Mit diesen Dateien kann dann die Platine bei den Laboringenieuren der Fakultät Elektrotechnik der HTWG Konstanz in Auftrag gegeben werden. Diese wird dabei aus einem großen Kupfernutzen mittels einer Platinenfräse herausgearbeitet.

Die benötigten Bauteile werden im Elektronikshop des Herstellers Mouser herausgesucht und nach schreiben einer entsprechenden Materialliste ebenfalls bei den Laboringenieuren zur Bestellung in Auftrag gegeben.

Nach Eintreffen aller notwendigen Komponenten wurde die Platine mittels händischen Lötens an einem ESD-Arbeitsplatz mit den elektronischen Bauteilen bestückt.

Da die Vias von Grund auf noch keine Kontaktierung von Oberseite und Unterseite bieten, werden diese mithilfe von Fädeldraht durchkontaktiert. Ebenso wie einige fehlende Leitbahnverbindungen, welche bei der Sicht- und Durchkontaktierungsprüfung noch aufgefallen sind. In der Abb. 15 ist die fertige Platine schon fertig montiert auf dem Microcontrollerboard abgebildet.

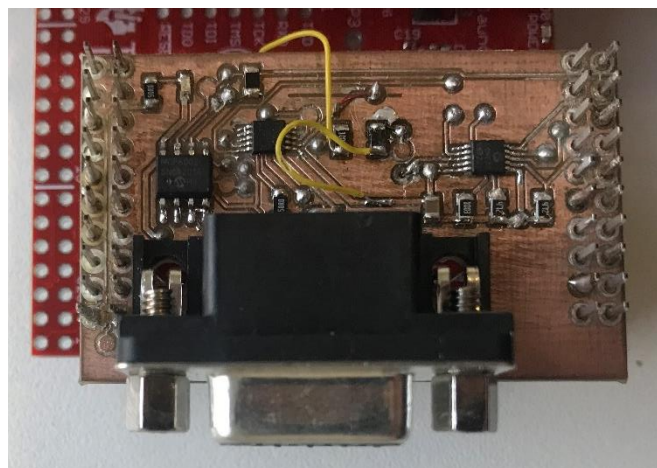


Abb. 15: fertige Erweiterungsplatine - montiert auf Microcontrollerboard

5. Softwareentwurf

Die Software für den Pulsoxymeter wurde in der Programmiersprache C mit dem Tool „Code Composer Studio 10.1.0“ entworfen.

Software-Konzept/Architektur

Eine detaillierte Softwarearchitektur im graphischen Sinne ist in Abb. 16 dargestellt. Das Konzept konzentriert sich im Kern um das Messverfahren, welches von wichtigen umliegenden Komponenten initialisiert, gesteuert und überwacht wird.

Nach der Initialisierung wird zwischen verschiedenen Betriebszustände bzw. Modi in welchen das Pulsoxymeter jeweils unterschiedlich operiert. Im normalen Betriebszustand („aktiv“) wird das Messverfahren gestartet. Um sämtliche nachkommende Kettenstrukturen dennoch synchron zu halten, steuern zwei Timer den gesamten Ablauf und triggern die Prozesse zur Ansteuerung des Sensors und der Verarbeitung der Messsignale.

Bei Erfassen von validen Messsignalen werden diese zwischengespeichert und zyklisch auf dem Display ausgegeben. Sollte kein valides Signal vorliegen, kann über den DAC die abgegebene Leistung an den Sensor nachjustiert werden. Wird allerdings kein Finger detektiert wird dies dem Benutzer mitgeteilt bzw. fährt das System in einen Ruhemodus.

Überwacht und unterstützt wird diese Prozesskette von einem Safety-Modul, welches einen direkten Einfluss auf die Betriebszustände haben kann. Sollte ein Fehler vorliegen wird dieser von der Zustandsmaschine detektiert und entsprechende Folgeaktionen ausführt.

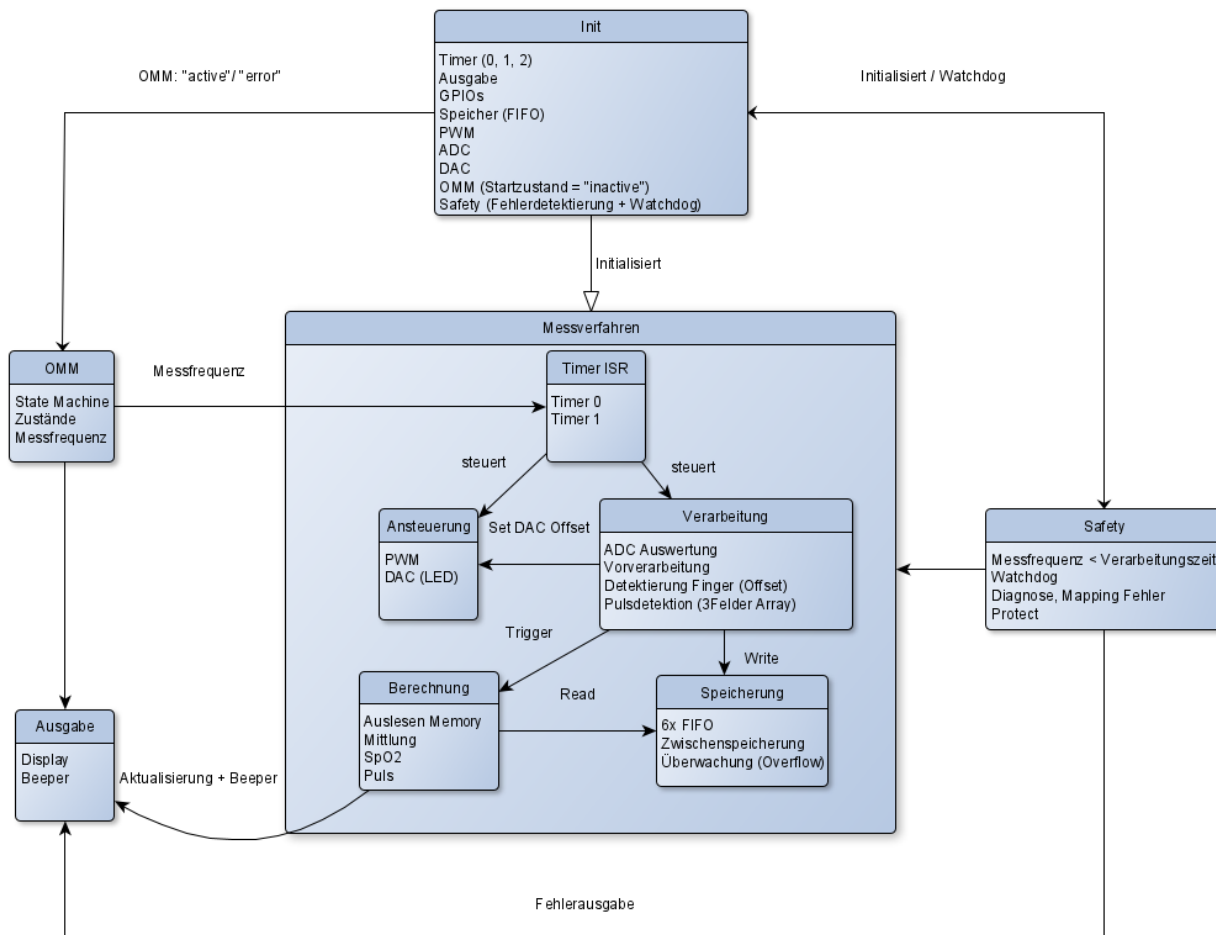


Abb. 16: Übersicht der Software-Architektur

Software-Modulbeschreibung / Ablaufplan

Main/Initialisierung

Die Main der Software beinhaltet den Aufruf der Initialisierungsroutine (Init.c) die nach starten des Pulsoximeters direkt aufgerufen wird. Im Anschluss wird die Zustandsmaschine kontinuierlich wiederholt.

Die Initialisierungsroutine beinhaltet alle essenziellen Initialisierungen für die verwendete Softwaremodule. Hierbei werden zuerst alle wichtigen Peripherien aktiviert:

- GPIOs – zur Ansteuerung der LEDs auf dem Microcontrollerboard
- Timer – für alle zeitgesteuerten Prozesse und Betriebszustände
- PWM – zum Setzen des High- und Low-Pegels der LEDs
- ADC – zur Digitalisierung des analogen Messsignales
- DAC – zum Einstellen der Intensität der LEDs sowie dem DC-Offset
- Interrupt Routinen – um bei erfüllten Auslösebedingungen entsprechende Aktionen durchführen zu können

Darüber hinaus werden die einzelnen Timer entsprechend angelegt und spezifiziert. Timer 0 und Timer 1 sind hierbei verantwortlich für die synchrone Ausführung des Messverfahrens. Da jeder Timer bzw. die entsprechenden Interrupts für den Prozess einer jeweiligen LED stehen, müssen diese zeitversetzt auslösen. Realisiert wurde dies durch eine Delay Funktion, die zwischen den Startfunktionen für die Timer eingepflegt wurde.

Darüber hinaus wird für das Safety-Modul der Watchdog angelegt, konfiguriert und gestartet und der Betriebszustand auf „inaktiv“ gesetzt.

Ablaufplan Messverfahren

Sofern kein Fehlerzustand anliegt wird das Messverfahren über die Interrupts der Timer abwechselnd (je LED) ausgelöst und der in Abb. 17: Flow-Chart dargestellte Prozess durchlaufen. Hierbei aktiviert der Timer das PWM Signal zum Sensor und gleichzeitig die

Verarbeitung der analogen Signale durch den ADC im Verarbeitungsmodul. Wird ein Pulsschlag detektiert, startet die Verarbeitung der im FIFO gespeicherten Werte.



Abb. 17: Flow-Chart des Messverfahrens

Operation Mode Management

Dieses Modul („OMM.c“) besteht im Kern aus der Zustandsmaschine und stellt die in Tab. 4: Software-Events beschriebene Events allen Software-Modulen bereit. Da die Zustandsmaschine zyklisch während des gesamten Betriebs aufgerufen wird, prüft die Funktion *GetEvent()* das aktuell anliegende Event an welches von den Software-Modulen mit *SetEvent()* gesetzt wurde.

Event	Beschreibung/Bedingung
<i>event_no_finger_detected</i>	Ein Finger wurde detektiert
<i>event_finger_detected</i>	Kein Finger im Sensor
<i>event_error_watchdog</i>	Watchdog hat ausgelöst
<i>event_error_diagnostic</i>	Spezifischer Software-Fehler liegt vor
<i>event_error_proceccing_time_extended</i>	Messfrequenz zu hoch
<i>event_error_memoryoverflow</i>	FIFO voll

Tab. 4: Software-Events

Darüber hinaus stellt dieses Modul die einzelnen Modi dar, die eingenommen werden können.

- Betriebszustand „aktiv“ stellt den normalen Betriebszustand der Einheit dar. Hier wird die System Clock mit 120MHz betrieben welche die Messfrequenz des Pulsoxymeters bestimmt. Die ermittelte Sauerstoffsättigung sowie der Pulsschlag werden zyklisch auf dem Display ausgegeben.
- Sobald der Benutzer den Finger aus dem Sensor entfernt, wird nach fünf Sekunden in den Zustand „inaktiv“ gewechselt. Hier wird der Systemtakt aus Energiespargründen heruntergefahren und dem Benutzer über eine Meldung auf dem Display mitgeteilt, dass kein Finger erkannt wurde.
- Nach fünf weiteren Sekunden fährt das System in den „Eco“-Modus. Hierbei wird der Systemtakt bis auf ein Minimum heruntergefahren sowie alle Ausgabeelemente (Display, LEDs, Beeper...) ausgeschalten. Erst wenn wieder ein Finger detektiert wird, wacht das System wieder in den Zustand „aktiv“ auf.
- Wird, während dem Betrieb ein Fehler detektiert, so springt das System in den „Error“-Zustand und der Benutzer wird über den Fehler auf dem Display informiert. Das System kommt aus solch einem Fehlerzustand nur über einen automatisch nach zehn Sekunden ausgeführten Reset heraus.

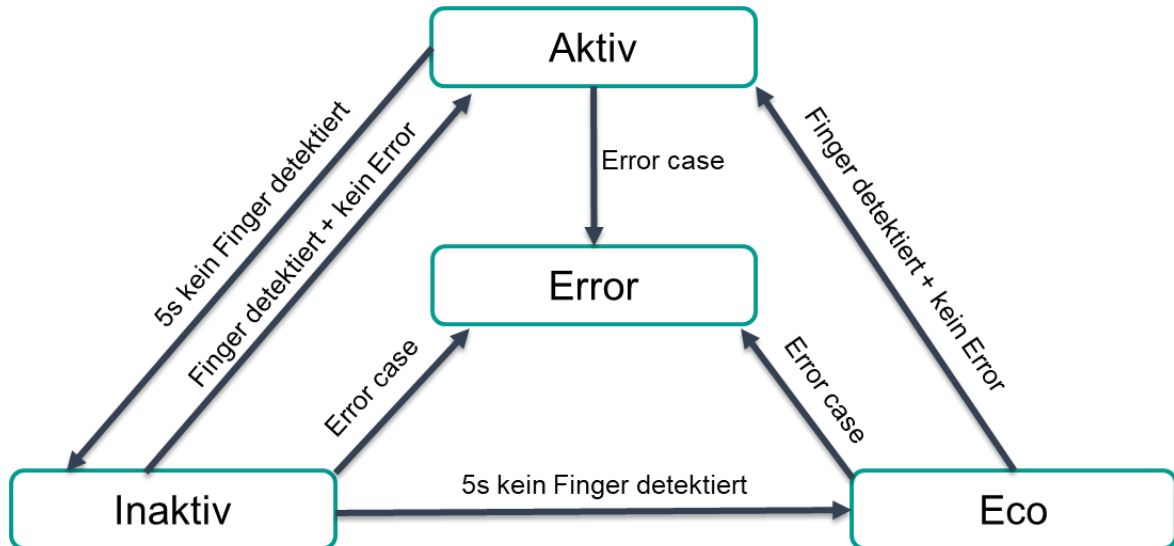


Abb. 18: State-Maschine

Timer

Die Timer0 und Timer1 stellen die zentrale Steuereinheit des Messverfahrens dar. Durch deren zeitversetzte Interrupt Routinen ist sichergestellt, dass es im Messverfahren keine Überlappungen zwischen den Messzyklen der Signalauswertung für die infrarote bzw. rote LED. Des Weiteren wird über den Timer2 die im OMM – Modul benötigte Ablaufzeit von fünf Sekunden erzielt.

Safety

Dieses Modul ist für die Sicherstellung verantwortlich, dass signifikante Fehler während des Betriebs, sowohl hardware- als auch softwareseitig, detektiert werden. Als Beispiel werden sämtliche Signalwerte plausibilisiert. Sollten hierbei die ausgelesene Werte einen definierten Bereich unter- bzw. überschreiten wird dies über das Event `event_error_diagnostic` der Zustandsmaschine mitgeteilt. Darüber hinaus stellt ein Watchdog sicher, dass der Programmablauf an einer Stelle hängen bleibt.

PWM

Zu Beginn wird in diesem Modul der Peripherie-Block für den PWM0 und die dazugehörigen Ausgänge am GPIO-Port F aktiviert. Als nächstes wird die PWM-Clock auf ein Achtel der der System-Clock eingestellt und jeweils der PWM-Generator 0/1 konfiguriert. Im folgenden Schritt wird die Periodendauer und der Zeitraum für ein Spannungssignal am Ausgang (Duty-Cycle) eingestellt und noch die entsprechenden Interrupts konfiguriert.

Über eine Aktivierungsfunktion kann bei Aufruf, der als Parameter übergebene PWM-Generator, gestartet werden. Dabei wird das entsprechende Ausgangsbit gesetzt und der Generator gestartet.

Über eine Deaktivierungsfunktion kann bei Aufruf, der als Parameter übergebene PWM-Generator, gestoppt werden. Dabei wird das entsprechende Ausgangsbit zurückgesetzt und der Generator gestoppt.

In den Interrupt-Funktionen der PWM-Generatoren wird jeweils nach einem Periodendurchlauf der Generator angehalten.

ADC

Für den Analog-Digital-Converter wird zuerst die Peripherie mit den ausgewählten Hardwareeingängen an Port E gestartet. Nach dem Setzen der Frequenz des Modules auf 20 MHz wird als Sequenz eine einmalige Messung (3 mit Step 0) bei Auslösen des Prozessortriggers eingestellt. Durch Hardwareoversampling werden dabei statt nur einem Messwert der Mittelwert von 8 Einzelmessungen hintereinander berechnet.

Nach der Aktivierung der dieser Sequenz kann anschließend mit der Prozessortriggerfunktion eine Analog-Digital-Wandlung des anliegenden Signales erfolgen.

DAC

Zur Kommunikation mit dem externen DAC wird der I2C-Bus verwendet. Dieser wird im ersten Schritt über die Aktivierung des Peripherie-Elementes eingeschaltet. Als zweites werden die entsprechenden GPIO-Pins (Port B 2/3) konfiguriert und der Microcontroller als Master am Bus eingestellt. Diesem wird zum Schluss noch die Slave-Adresse des DAC mitgeteilt.

Mittels einer entsprechenden Funktion können byteweise die Daten über den I2C-Bus in die Register des DAC geschrieben werden. Dabei wird nach jeder Übertragung überprüft, ob diese auch erfolgreich verlief.

Ausgabe

Die Ausgabe der verarbeiteten Messwerte erfolgt über das Display des Boosterpack's. Dafür sind in einem LCD Modul Funktionen für die Ausgabe, der oben beschriebenen Operationszustände definiert. Zusätzlich wird ein PWM-Signal für die Signalisierung des Herzschlages mit einem Buzzerton erzeugt.

Verarbeitung

Im Verarbeitungsmodul (engl. „Processing“) werden zu Beginn die beiden ADC-Werte ausgelesen und anschließend auf das Einhalten der gesetzten Plausibilitätsgrenzen getestet, um zu erkennen, ob ein Finger eingelegt wurde. Anschließend wird geprüft, ob ein Puls vorliegt. Falls ja, werden zunächst die ADC- Werte in die entsprechenden FIFOs zwischengespeichert und anschließend die Berechnung getriggert. Über eine Laufvariable werden die ausgelösten Timer Interrupts mitgezählt, um diese Zeitangabe mit in das Berechnungsmodul übergeben zu können.

Berechnung

In diesem Modul finden die Berechnung von Sauerstoffsättigung und Puls statt. Es werden die FIFOs ausgelesen und aus den Werten die SpO2 berechnet, danach wird eine Mittelung über alle Spo2 der vergangenen Zeit seit dem letzten Puls durchgeführt. Mit der übergebenen Variablen mit der Anzahl der verstrichenen Timer Interrupts wird der Puls ermittelt. Abschließend werden die beiden Werte zur Ausgabe an das LCD Modul übergeben.

FIFO

Im FIFO-Modul sind alle Methoden zum Initialisieren, zum Schreiben und zum Lesen des entsprechenden FIFOs implementiert. Gegen ein paralleles Auslesen und Schreiben sind Vorbereitungen getroffen.

LUT (Look Up Table)

In diesem Modul ist eine Look Up Table für das errechnete Verhältnis der Messsignale von roter und infraroter LED definiert. Das Modul wird lediglich im Berechnungsmodul verwendet.

6. Inbetriebnahme und Ergebnisse

Nach der Bestückung der Platine und den Korrekturen, der in Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** beschriebenen Layoutfehlern wurde eine Durchgangsprüfung per Multimeter an sämtlichen Komponenten der Messplatine durchgeführt, um die Verbindung zu gewährleisten. Nach der Fehlerkorrektur (siehe Fertigung Kapitel 4) konnte, bei einer erneuten Durchgangsprüfung, das Messergebnis die korrekte Kontaktierung der einzelnen Bauteile bestätigen.

Bei der Erstinbetriebnahme konnte auch per Multimeter die entsprechende Versorgungsspannung von 3,3V nachgewiesen werden. Der Funktionalitätsbeweis des selbstgebaute Sensors sowie des Fingerclip-Sensors von Nellcor konnte per Diodentest erbracht werden.

Im Folgenden wird nun die Softwareintegration beschrieben und welche Tests hierfür durchgeführt wurden.

Eines der grundlegenden Komponenten für die Funktion ist das PWM-Modul, welches für die Erzeugung der Rechteckspannungen für die beiden LED zuständig ist. Aus der Anforderungsanalyse (siehe Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**) sind ein Tastgrad der beiden Signale von 0,11 sowie ein zeitlicher Versatz von 540µs zu entnehmen.

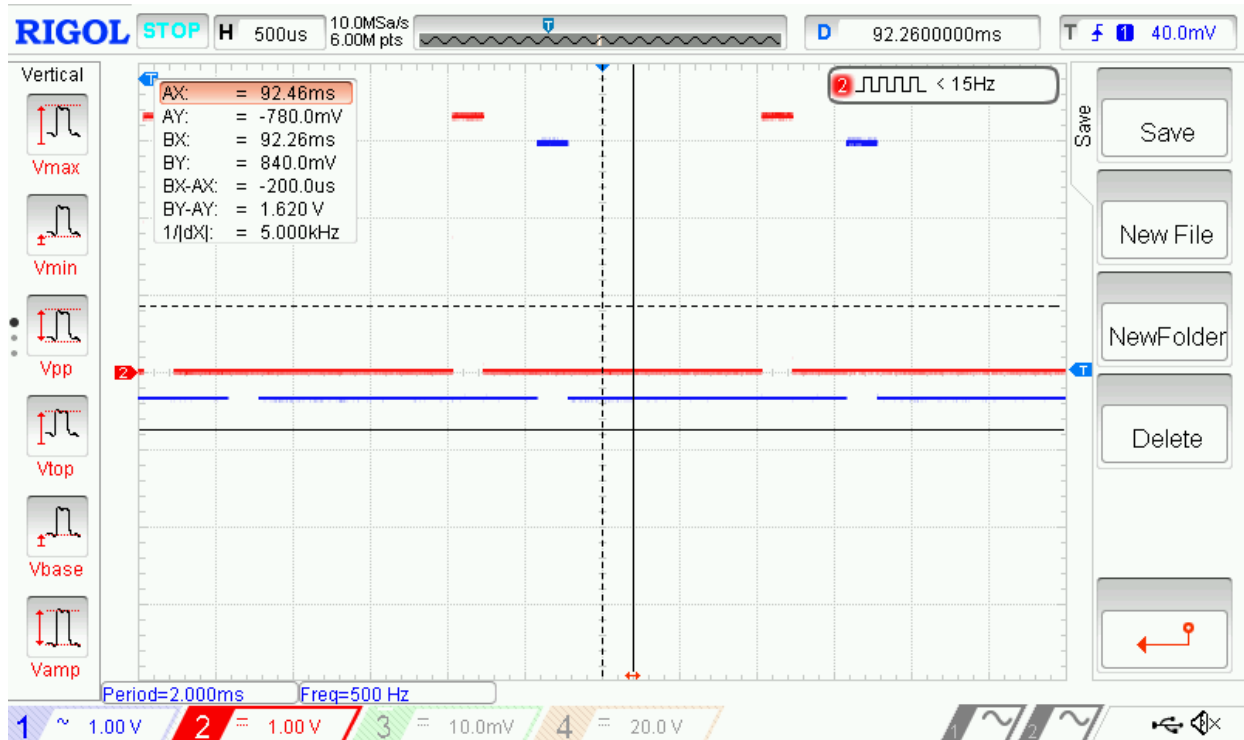


Abb. 19: Oszillogramm der PWM Signale für rote (rot) und infrarote (blau) LED

Wie in dem Oszillogramm von Abb. 19: Oszillogramm der PWM Signale für rote (rot) und infrarote (blau) LED zu sehen ist, konnten die PWM Signale mit einer Periodendauer von 2ms und einer Pulsbreite von 220µs nachgemessen werden. Dies entspricht dem zuvor gewünschten Tastgrad. Auch der Versatz von 540µs ist korrekt.

Der DAC „MCP4728“ von Microchip, welcher die Helligkeitssteuerung der LEDs durch die Stromregelung der Transistoren vornimmt, wird per I2C Bus parametrierbar. Das in Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** erklärte DAC-Modul ist hierfür implementiert. Per Oszilloskop wurde, wie in Abb. 20: Oszillogramm I2C Bus; SDA(blau), SCL(rot) zu sehen, die Buskommunikation überprüft. Der untenstehende Ausschnitt zeigt die umgesetzte I2C Kommunikation zwischen dem Mikrocontroller und dem DAC.

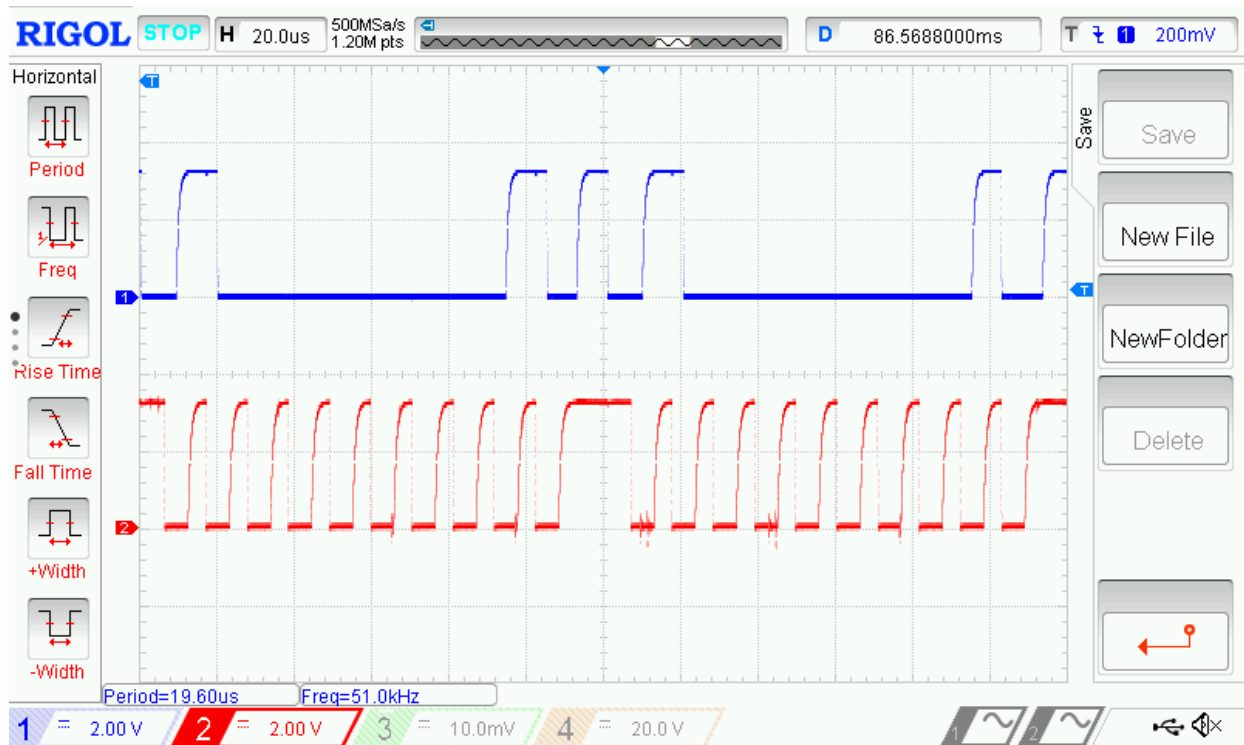


Abb. 20: Oszillogramm I2C Bus; SDA(blau), SCL(rot)

Zu erkennen ist auch ein Verschleifen der Pegel, dies ist ein Indiz dafür, dass die Dimensionierung der Kondensatoren C1 und C2 für die Buskapazität zu hoch angesetzt war. Da das verwendete Oszilloskop keine Triggerfunktion für I2C besitzt können die Digitalwerte nicht direkt im Oszillogramm angezeigt werden. Die gewünschte Initialspannung stellte sich nach der Initialisierung der Kommunikation an den Ausgängen des DAC ein.

Am Eingang des ADC0 stellte sich nach der Sicherstellung der Funktionalität der vorigen Komponenten der folgende Spannungsverlauf ein:

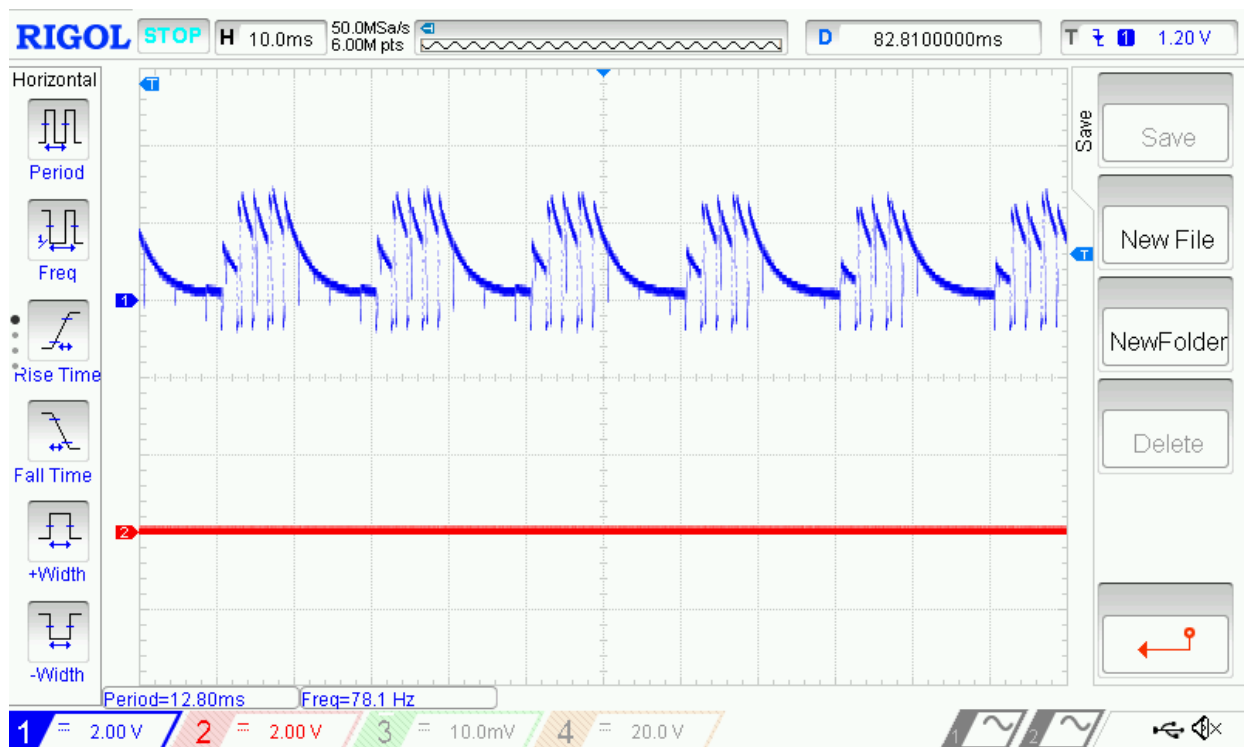


Abb. 21: Oszillogramm ADC0-Eingang (blau; rot ohne Bedeutung)

Wie Abb. 21: Oszillogramm ADC0-Eingang (blau; rot ohne Bedeutung) zu sehen, liegt nach der Signalkonditionierung eine pulsende Spannung an. Die Funktionalität der Regelung konnte so zunächst bestätigt werden.

Jedoch brach das Messsignal im weiteren Verlauf der Inbetriebnahme weg und konnte nicht mehr hergestellt werden. Auf Grund der anschließenden Fehlersuche, liegt die Vermutung nahe, dass der DAC nicht mehr funktionsfähig ist. Die folgende Messung für der Spannung an der roten LED ergab den Signalverlauf in Abb. 22: Oszillogramm Spannungssignal an LED rot zu Masse.

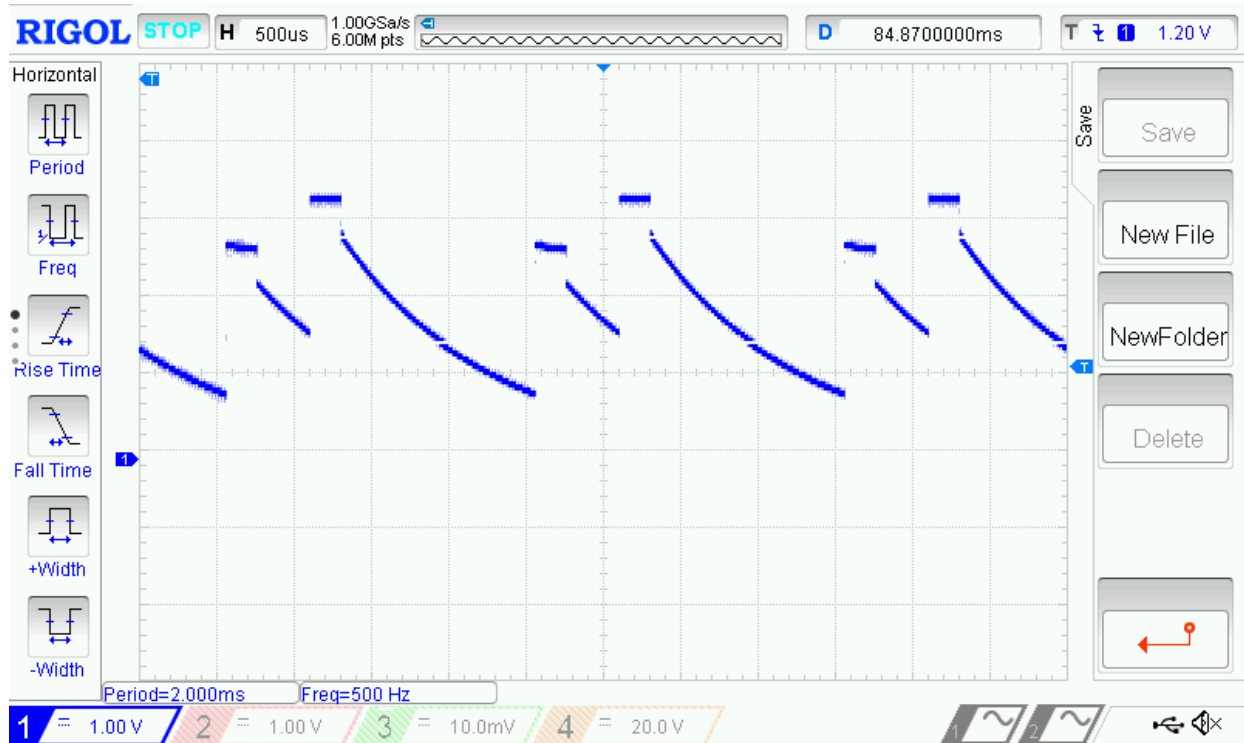


Abb. 22: Oszillogramm Spannungssignal an LED rot zu Masse

Da ohne das Aussteuern der beiden Transistoren Q1 und Q2 den LEDs der Massebezug fehlt, kann sich der Kondensator C4 nicht mehr korrekt entladen, dieser wurde ursprünglich zum Glätten von Einschaltstromspitzen integriert. Dies ist eine Erklärung für das Verschleifen des obigen Signals.

Die anschließende Messung aus Abb. 23: Oszillogramm RDY/BS (rot), DAC_A (blau), DAC_B (grün) der Ausgänge A und B sowie des Busy-Bits des DACs zeigt, dass die Ausgänge bei 0V und das invertierte Busy-Bit auf HIGH verbleiben. Die Kommunikation per I2C fand statt wie zu Beginn (siehe Abb. 9: Digital-Analog-Converter zur Regelung des LED-Stromes und zum DC-Offset-Abgleich der Verstärkerschaltung).

Da auch durch Zurücksetzen der Software, auf den vorigen Stand, kein Signal mehr generiert werden konnte, ist von einer Zerstörung des DACs auszugehen.

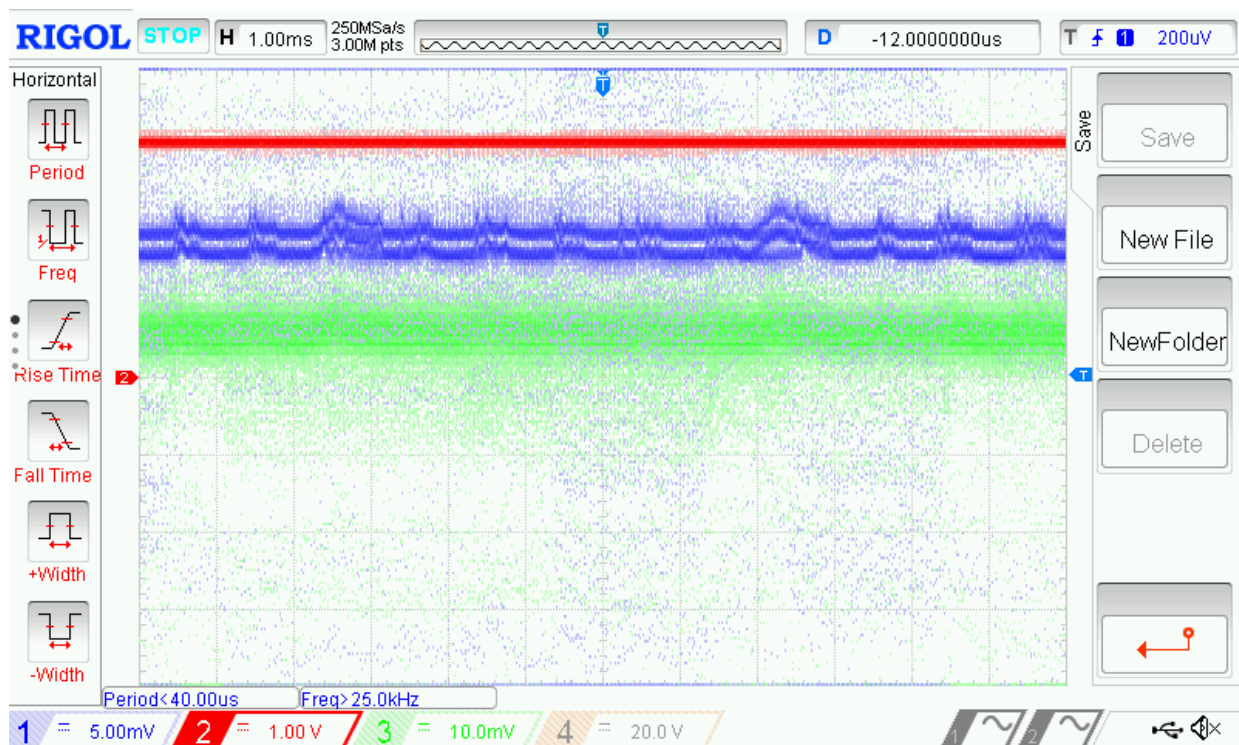


Abb. 23: Oszillogramm RDY/BS (rot), DAC_A (blau), DAC_B (grün)

Die Funktion der Schaltung konnte somit nachgewiesen werden, da sämtliche Hardwareanforderungen (siehe Anforderungsanalyse) getestet wurden, jedoch konnte die Ursache für den später auftretenden Funktionsverlust des DAC nicht ermittelt werden.

Das weitere Testen der Softwaremodule konnte daraufhin nicht mit funktionierender Messplatine erfolgen. Deswegen musste auf Debugging und Überprüfung der syntaktischen Korrektheit zurückgegriffen werden. Die einzelnen Tests sind in der Anforderungsanalyse aufgelistet, solche die nicht durchführbar sind, wurden mit n.A. („not applicable“). Jedoch konnte für den mehrere Module so die Testreihen noch durchgeführt werden. Beispielhaft konnte so die Initialisierung sowie das Auslösen des Watchdog-Timers getestet werden.

Für die testbaren Requirements konnte eine Testabdeckung von 100% erfolgen. Berücksichtigt man die n.A. bezeichneten Anforderungen wird eine Abdeckung von 76% erzielt.

7. Fazit und Ausblick

Auf Grund des vermutlich defekten DAC konnte das Projekt nicht fertig gestellt werden. Auch die Ursache für den Funktionsverlust konnte nicht abschließend geklärt werden. Man hätte durch ein redundant geschaltetes Potentiometer zunächst analog die Helligkeit regeln können, um dann im Nachgang die digitale Regelung zu implementieren. So hätte der Ausfall des DAC möglicherweise kompensiert werden können. Auf diesem Wege wären die Schwellen auch experimentell einfacher zu ermitteln gewesen.

Als Ausblick könnten noch mehrere softwareseitige Ergänzungen programmiert werden, welche auch in der Anforderungsanalyse gelistet sind. Beispielhaft ist eine Plausibilitätsüberprüfung der Übergabeparameter einer jeden Funktion zu nennen (Safety-Modul) oder ein Verarbeitungsflag, welches gesetzt wird, wenn das Processing-Modul arbeitet.

Zu den größten Herausforderungen zählte das Entwerfen und Bestücken der Hardware sowie die Programmierung der Softwarearchitektur und die Pflege der Anforderungsanalyse.

Abschließend ist zu sagen, dass durch das Projekt der Erfahrungsschatz im Umgang mit dem Layoutprogramm „Circuit Maker“, der Entwicklungsumgebung „Code Composer

Studio“ und der Programmiersprache C erweitert wurde. Die Bandbreite des Projekts erstreckte sich über den gesamten Entwicklungsprozess eines Embedded Systems.

Literatur

- [1] Physikalisches Anfängerpraktikum der Universität Konstanz, B.-U. Runge, www.t1p.de/Spektroskopie-VIS-NIR, Sep. 2019
- [2] Entwerfen Sie ein kostengünstiges Pulsoximeter unter Verwendung von Standardkomponenten, Digi-Key, Stephen Evanczuk, www.t1p.de/design-a-low-cost-pulse-oximeter, 13.08.2020
- [3] CircuitMaker, v1.3, Altium Limited 2021, www.circuitmaker.com, 21.02.2021
- [4] Nellcor-DS100A, www.t1p.de/Nellcor-DS-100A, 21.02.2021
- [5] LTspice, v17, Analog Devices 2021, www.t1p.de/ltspice-simulator, 21.02.2021

Anhang

Die im Anhang aufgeführten Punkte sind als einzelne Dateien in einem .zip-Archiv enthalten.

- Pulsoxymeter Anforderungsanalyse in Requirements ES - Oxymeter V6.xlsx
- Pulsoxymeter Programmcode in CCS-Projekt Puls-Oxymeter_v1_Group_5
- Pulsoxymeter Quick Start Guide
- Pulsoxymeter Präsentation
- Pulsoxymeter Schaltplan und Layout als PDF