Endbericht zum Berufspraktikum

Werner Dichler

BACHELORARBEIT

Nr. 238-003-045-B

eingereicht am
Fachhochschul-Bachelorstudiengang

Hardware Software System Engineering

in Hagenberg

im Juli 2008

Praktikumsstelle:

CDE - Communications Data Engineering GmbH
Softwarepark Hagenberg
4232 Hagenberg, Softwarepark 37/1

+43 7236 3351-4300
www.cde.at

Kontaktperson:

DI(fh) Klaus Koppenberger

Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Hagenberg, am 3. Juni 2008

Werner Dichler

Inhaltsverzeichnis

Erklärung **iii**

1 Das Unternehmen **1**
1.1 Kurzbeschreibung 1
1.2 Standort 1
1.3 Betreuer 2

2 Projekte und Tätigkeiten während des Praktikums **3**

3 Auszüge Hardware-Entwurf **6**
3.1 Prozessorwahl 6
3.2 Abstandsensor 6
3.3 Temperatursensor 6
3.4 Lasertemperatur 7
3.5 Laserstrom 7
3.6 Beleuchtung 7
3.6.1 Schrittmotor Endschalter 8
3.7 Lüfter Überwachung 8
3.8 Spannungsversorgung 9
3.9 Schutzbeschaltung 10

4 Auszüge Software-Entwurf **12**
4.1 Entzerren 12
4.2 Bildübergang 12
4.3 Positionssuche 13

5 Erfahrungen und Zusammenfassung **18**

Kapitel 1

Das Unternehmen

1.1 Kurzbeschreibung

Die Firma CDE-GmbH wurde im Jahr 2001 durch Dipl.-Ing. Mag. Dr. Josef Langer gegründet. Seit 2005 ist sie EN ISO 9001:2000 zertifiziert. Somit wurde ein Qualitätsmanagement-System eingeführt, welches die internen Abläufe optimiert. Die Schwerpunkte der Tätigkeiten liegen bei Payment-Systemen sowie bei der Entwicklung von medizinischen Geräten.
Bei den Payment-Systemen werden unterschiedliche Lösungen angeboten, die für den jeweiligen Zweck optimiert und angepasst werden. Mögliche Einsatzbereiche sind Kantinen, Firmen-, Betriebslösungen, Freizeiteinrichtungen, Kliniken und Krankenhäuser.
Für Embedded Systems Entwicklungen werden verschiedene Prozessoren eingesetzt, die für das Aufgabengebiet maßgeschneidert sind. Neben dem Entwurf der Hardware wird auch die Software für diese System erarbeitet. Diese werden ebenfalls individuell gestaltet und reichen von einfachen Steuerungsaufgaben bis hin zu komplexen Real-Time-Systemen.
Im Bereich der Medizintechnik wurden bereits einige Produkte entwickelt. Im Umfang der Erarbeitung wurden diese Produkte auch in Europa bzw. in Amerika zertifiziert.

1.2 Standort

CDE - Communications Data Engineering GmbH
Softwarepark Hagenberg
Softwarepark 37/1
4232 Hagenberg

1.3 Betreuer

DI(fh) Klaus Koppenberger
Tel.: +43 7236 3351-4310
E-Mail: Klaus.Koppenberger@cde.at

Kapitel 2

Projekte und Tätigkeiten während des Praktikums

Meine Aufgabe im Unternehmen war die Hardware- und Softwareentwicklung eines externen Projektes in der Medizintechnik. Bei diesem Projekt handelt es sich um ein Therapiegerät, welches für die Keimabtötung bei größeren Wunden eingesetzt werden soll.

Die erste Tätigkeit lag darin, für den Hardwareaufbau geeignete Komponenten zu ermitteln. Die Hardware ist für die Ansteuerung des Lasers und des Schrittmotors, sowie für die Auswertung verschiedener Sensorwerte zuständig. Zu den Sensoren zählen Abstands-, Drehzahl-, Temperatur- sowie Stromsensoren. Diese müssen geeignet Beschalten werden, um die Informationen dem Controller zugänglich zu machen.

Die Bauteile der Schaltungen wurden aus dem Sortiment zweier Elektronik-Bauteil-Lieferanten (RS-Components und Farnell) ausgewählt und in einer Stückliste zusammengefasst. Der Grobentwurf der Hardware-Schaltung erfolgte auf einem Notizblock und wurde, nach Sicherstellung der Richtigen Beschaltung, mit Eagle (einem Schaltplan-Entwurfs Programm) gezeichnet. Während diesem Vorgang wurden immer wieder Dokumente für den Hardwareaufbau verfasst und aktualisiert. Sobald der erste Entwurf der Hardware vorlag, wurden einzelne Komponenten (wenn möglich bedrahtet) bestellt und Teile der Hardwareschaltung aufgebaut und ausführlich getestet. Die Testergebnisse wurden ebenfalls dokumentiert und mit den Vorgaben verglichen. Bei Abweichen der Resultate wurde die Schaltung nochmals überprüft und falls ein Fehler vorlag abgeändert. Während der Tests kamen verschiedene Geräte zum Einsatz, wobei das Multimeter das am häufigsten eingesetzte Messinstrument war.

Nachdem alle Schaltungsgruppen ausführlich getestet wurden, wurde das Boardlayout geroutet. Während dem Routen werden die Gruppen immer zusammengefasst und geeignet platziert, wobei die Bauteile des öfteren umplatziert werden, um das Layout möglichst kompakt zu halten. Nachdem die

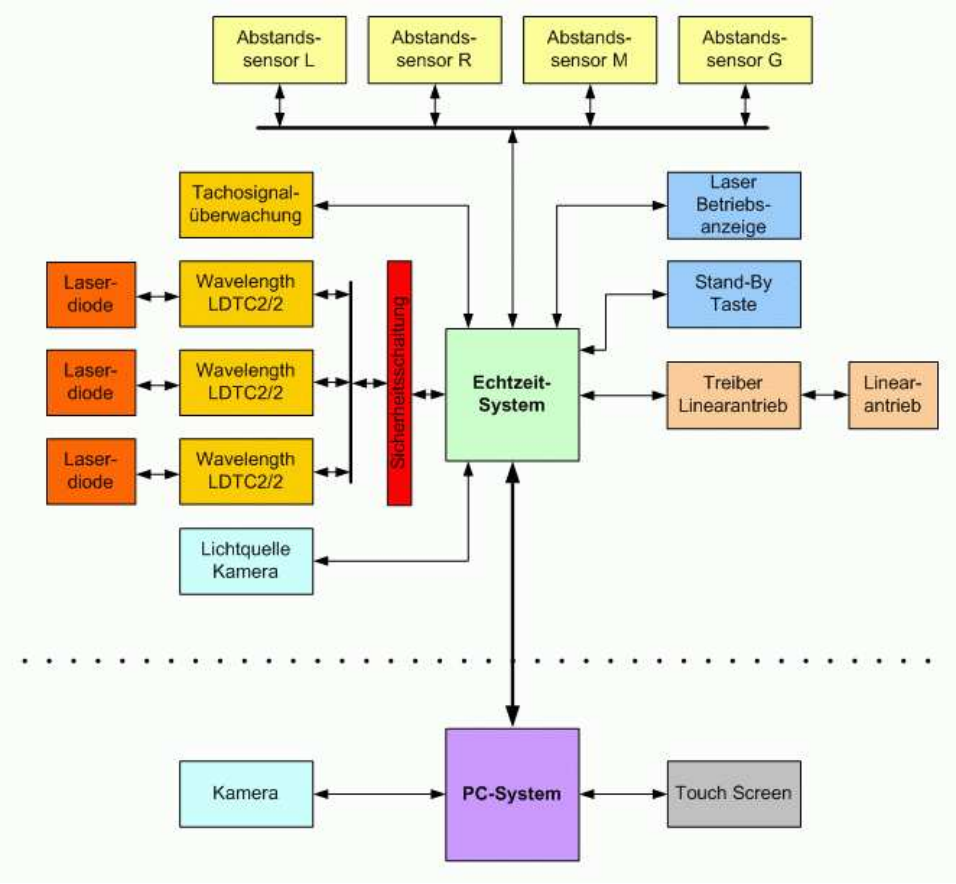


Abbildung 2.1: Gesamtüberblick

Bauteile grob positioniert sind, können die Verbindungen aufgebaut werden. Das fertige Board wurde anschließend für den Prototypen angefertigt und aufgebaut bzw. gelötet. Da es sich bei den meisten Bauteilen um Bauteile mit SMD-Ausführung handelt, wurden sie mit einem Mikroskop und einem feinen Lötkolben aufgelötet. Das fertig zusammengebaute Board wurde im Anschluss genau überprüft. Bei der Überprüfung wurden sämtliche verschiedenen Spannungen überprüft sowie das Verhalten beim Anschluss der externen Komponenten (Lüfter, Schrittmotor, Laser).
Zwischendurch bei Wartezeiten auf Lieferanten bzw. Hersteller, wurde bereits an der Software gearbeitet. Da der eingesetzte Controller gleiches Verhalten aufweist, wie der Controller auf der Sandbox, wurden die Softwareteile bereits auf der Sandbox getestet. Bevor die Entwicklung einzelner Funktionen und Prozesse startete musste der Hardware Abstraction Layer (kurz HAL) geschrieben werden. Mit diesem wird das abstrahierte ansprechen der Hardware erreicht, somit kann die Firmware bei Pin-Änderungen beibehalten

werden. Es muss nur der HAL angepasst werden. Für eine weitere Unterstützung während der Software-Entwicklung wurden einige Standard-Funktionen bzw. Definitionen verfasst.
Nachdem die unterste Ebene der Software fertig war, musste das eingesetzte Echtzeit-Betriebssystem angepasst und erweitert werden. Für die Hardwarekomponenten wurden verschiedene Treiberroutinen (Ansteuerung des Schrittmotors, Kommunikation, Standby, Debug, Selbsttest) geschrieben, welche in den Abarbeitungs-Prozessen verwendet werden können. Die allgemeinen Aufgaben werden von mehreren Prozessen übernommen. Es existiert jeweils ein Prozess für die Kommunikation mit dem Computer, für den Selbsttest (der nach jedem Start durchgeführt wird), für den Scanvorgang der Abstände, sowie für den Therapievorgang.
Ein weiterer Bereich im Softwareentwurf ist das C#-Programm am Computer. Der Computer ist für den Gesamtablauf der Therapie zuständig und visualisiert die erfassten Daten (Sensorwerte, Kamerabild der Wunde, Scanvorgang, Therapievorgang). Die Interaktion mit dem Benutzer erfolgt über einen Touch-Screen, wobei auf einfache Handhabung wert gelegt wurde. Somit werden im C#-Programm GUI-Elemente sowie Funktionen für den Kontrollfluss programmiert. Bereits während dem Entwurf des Programms, wurden die Teilsysteme miteinander verbunden (Computer mit Controller) um Software-Bereiche im Gesamtsystem testen zu können.

Im Zuge der Qualitäts-Management Aufgaben wurden laufend Dokumente bezüglich der verschiedenen Entwicklungs-Bereiche verfasst. Des weiteren wurden laufend Absprachen mit dem Projektleiter des Auftraggebers geführt. Somit wurde sichergestellt, dass die Entwicklung in die richtige Richtung verläuft.

Kapitel 3

Auszüge Hardware-Entwurf

3.1 Prozessorwahl

Für die Realisierung des Systems werden 11 Analog-Eingänge, 10 Digital-Eingänge, 27 Digital-Ausgänge und 4 Interrupt-Eingänge benötigt. Die Kommunikation mit dem Industrie-Computer setzt eine serielle Schnittstelle voraus. Für regelmäßige Routinen-Ausführungen werden mindestens 3 unabhängige Timer benötigt. Der AVR AtMega1280 Prozessor besitzt 86 IO-Ports, 16 ADC-Eingänge mit 10 Bit Auflösung, 8 Interrupt-Eingänge und insgesamt 6 Timer (2 x 8 Bit, 4 x 16 Bit). Der Prozessor kann mit maximal 16 MHz getaktet werden und erfüllt somit alle vorausgesetzten Bedingungen.

3.2 Abstandsensor

Der Abstandssensor wird mit +5V versorgt und liefert je nach Abstand zum Objekt eine bestimmte Referenz-Spannung. Der Abstand kann zwischen 3 und 40cm betragen, wobei die Spannung in einem Bereich von 0 und 3,2V liegt. Das Abstandssignal wird direkt auf einem ADC-Eingang geführt, welcher eine analoge Spannung im Bereich von 0 und 5V (Vcc des Prozessors) digitalisieren kann.

3.3 Temperatursensor

Der Temperatursensor wird mit +5V versorgt und liefert proportional zur Temperatur eine Gleichspannung im Bereich 0,1 bis 1,75V. Die Ausgangsspannung errechnet sich aus der aktuellen Temperatur in °C mal 10mV plus einen Offset von 500mV. Damit der ADC-Eingang besser ausgesteuert wird, wird die Ausgangsspannung des Sensors mittels nicht invertierenden Verstärkers verdoppelt.

$$V_{out} = (10\frac{mV}{°C} \cdot T) + 500mV$$

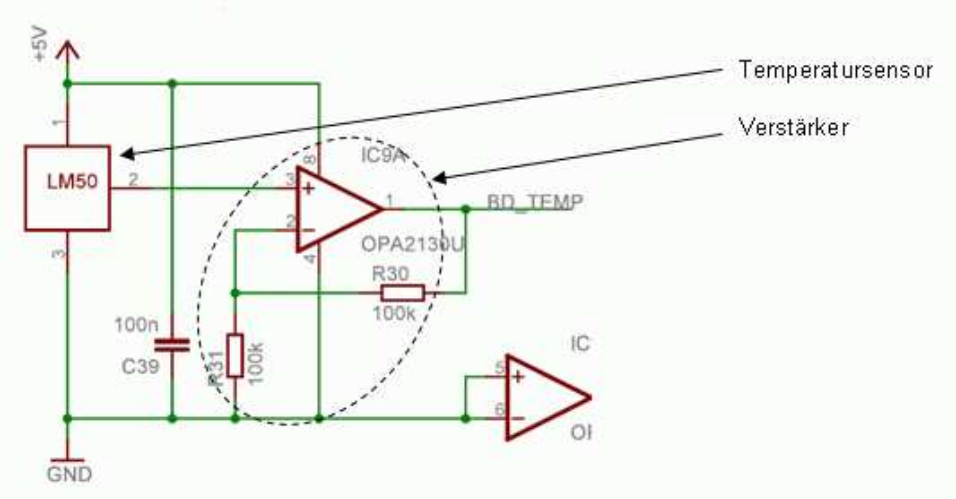


Abbildung 3.1: Temperatur Sensor

3.4 Lasertemperatur

Die Spannung des Lasertemperatur-Signals soll bei 1,22V (1,0 - 1,5V) liegen. Um eine Spannung im idealen ADC-Eingangsbereich zu erhalten wird ebenfalls eine Spannungsverdopplung durchgeführt. Damit das Signal von Störeinflüssen bereinigt wird, wird es von zwei Tiefpässen gefiltert. Die Tiefpässe werden so dimensioniert dass die Grenzfrequenzen bei ca. 30Hz liegen.

$$f_g = 30Hz, C = 100nF$$

$$R = \frac{1}{2 \cdot \pi \cdot f \cdot C} = 53,05kOhm$$

3.5 Laserstrom

Die Spannung des Laserstrom-Signals soll bei 1,17V (+-5%) liegen. Es wird wieder ein Verstärker nach geschalten um den ADC besser auszusteuern. Ebenfalls wird das Signal mit 2 Tiefpassfiltern bereinigt.

3.6 Beleuchtung

Für eine extern angebrachte Beleuchtung der zu bestrahlenden Oberfläche, wird ein Strom von maximal 1A bereitgestellt. Der analoge Schalter wird

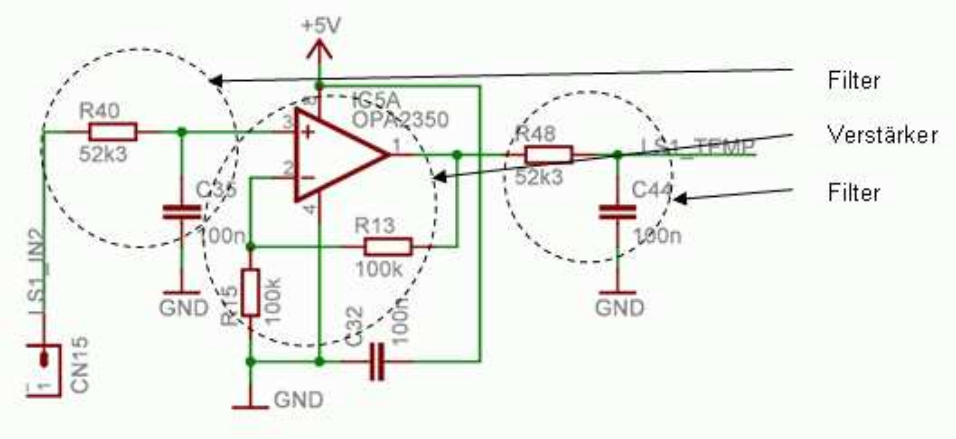


Abbildung 3.2: Laser Temperatur

mittels n-Kanal MOSFET realisiert, der durch einen Vorwiderstand vor Zerstörung geschützt wird.

3.6.1 Schrittmotor Endschalter

Damit die Endpositionen der Schrittmotoren dem Controller bekannt sind, werden Miniaturschalter geeignet positioniert. Für einen definierten Pegel bei offenen Schaltern werden PullUp-Widerstände verwendet. Ist der Schalter geschlossen, so wird der Controller-Eingang auf Masse gezogen und über die Widerstände soll ein geringer Strom von ca. 0,1mA fließen.

3.7 Lüfter Überwachung

Die Überwachung wird von einem monostabilen Multivibrator übernommen. Er wird so beschalten, damit das invertierte Ausgangssignal bei einer regelmäßigen Flanke des Lüfters auf Low-Pegel bleibt. Setzt das Signal des Lüfters aus, so geht das Signal auf High-Pegel. Die Zeitspanne tw gibt vor, wie lange der Lüfter Zeit hat um das Monoflop wieder neu zu triggern. Kommt nach dem Ablauf dieser Zeit keine steigende Flanke, so wird ein Lüfterstopp signalisiert. Die Periodendauer des Laserlüfter-Drehzahlsignals beträgt 4,9ms (12000U/min). Damit bei geringer Verschmutzung des Rotors kein Lüfter-Error signalisiert wird und damit der Gehäuselüfter mit einer geringeren Drehzahl (1000 - 5000U/min) ebenfalls betrieben werden kann, wird die Zeitspanne großzügig bemessen.

$$t_w\,[ns] = K \cdot R\,[kOhm] \cdot C\,[pF] = 0,55 \cdot 1MOhm \cdot 100nF = 55ms$$

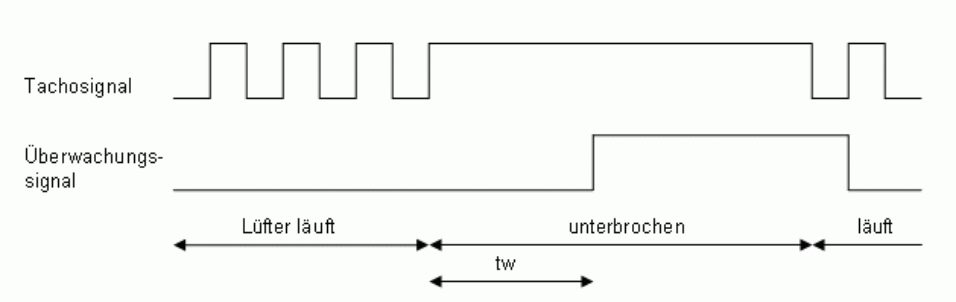


Abbildung 3.3: Lüfter Zeitverhalten

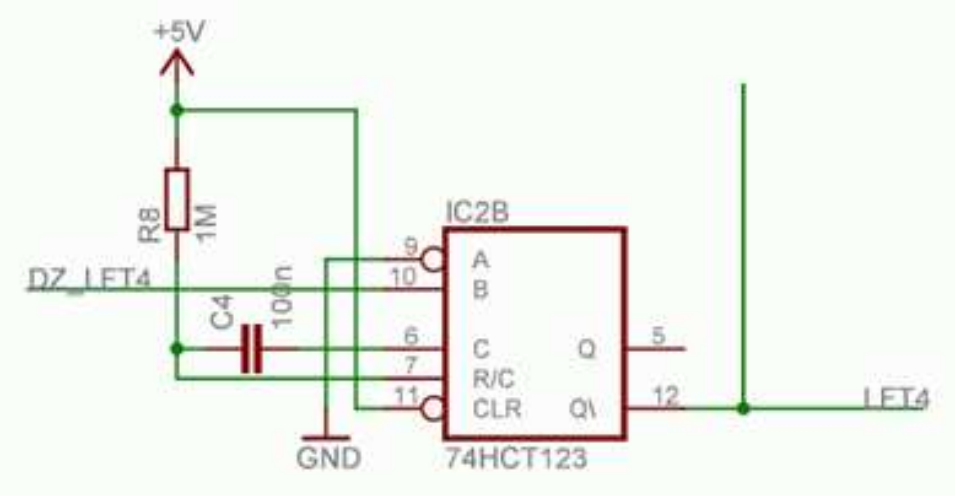


Abbildung 3.4: Lüfter Überwachung

3.8 Spannungsversorgung

Damit das bereits vorhandene Netzteil für den Industrie-Computer auch als Energieversorgung der Controller-Platine genutzt werden kann, wird ein DCDC Wandler von 12 auf 5V benötigt. Dieser wird so ausgelegt dass ein Strom von 2,2A je Laser verbraucht werden kann (40W ’ 8A).
Damit die Leitungen zum Controller durch einen fehlerhaft hohen Stromverbrauch nicht beschädigt werden, wird eine rückstellende Sicherung in Serie verbaut. Als Verpolungsschutz am 12V Eingang wird eine Diode verwendet, dabei wird ein kurzschlussfestes Computer-Netzteil voraus gesetzt.

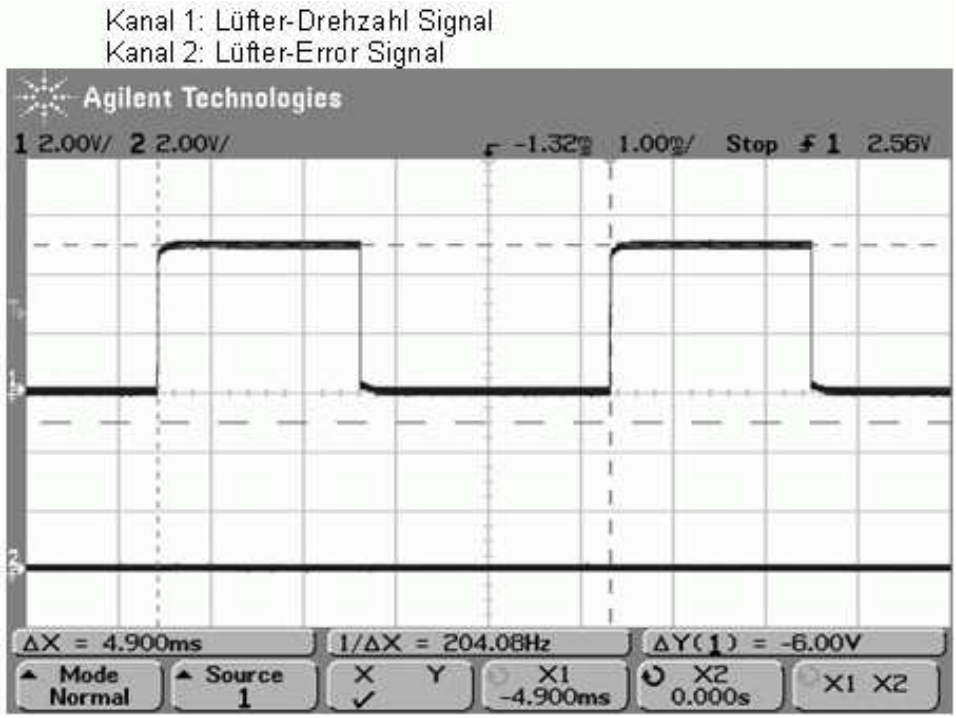


Abbildung 3.5: Lüfter läuft

3.9 Schutzbeschaltung

Damit die Platine vor elektrostatischen Entladungen geschützt ist, werden bei allen Steckeranschlüssen, die zu externen Komponenten führen, ESD-Dioden verwendet. Der Pegelwandler des UARTs ist ebenfalls ESD-sicher ausgeführt.

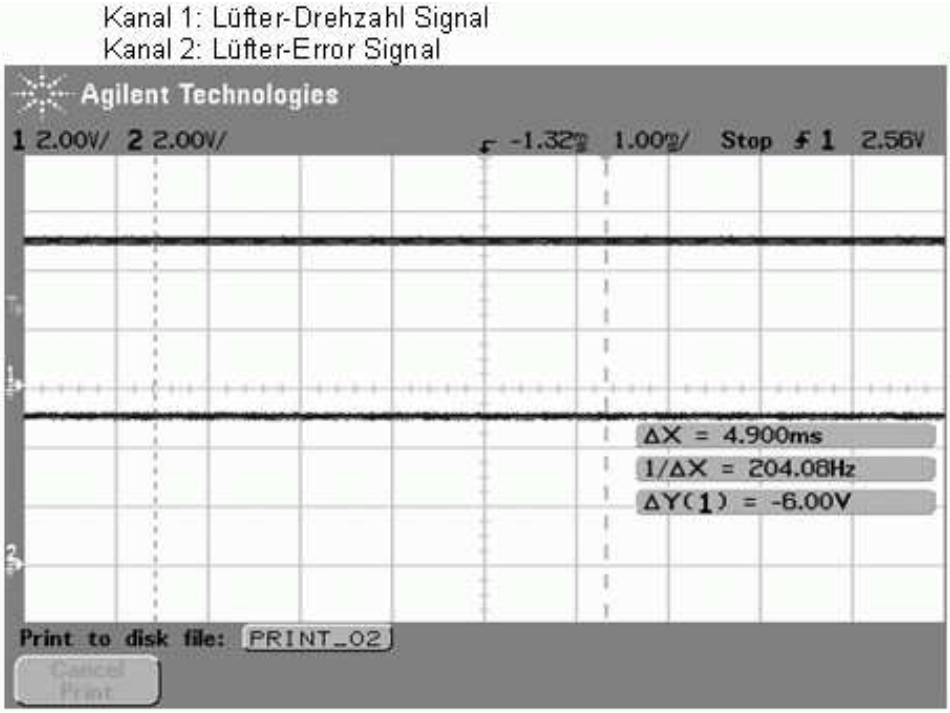


Abbildung 3.6: Lüfter blockiert

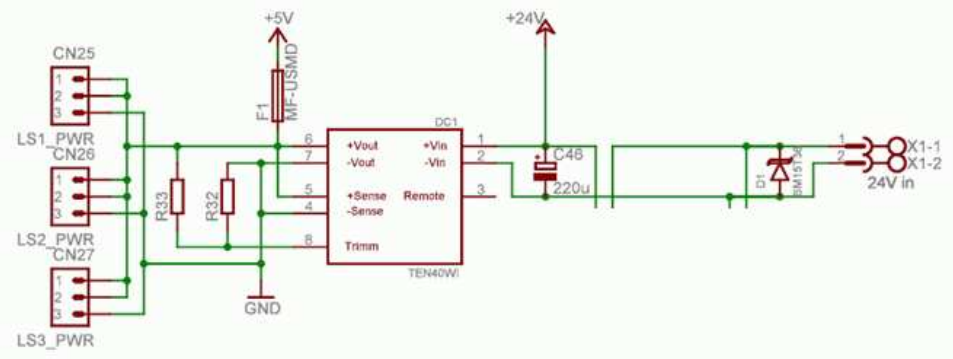


Abbildung 3.7: Spannungsversorgung

Kapitel 4

Auszüge Software-Entwurf

4.1 Entzerren

Die Kamera für die Aufnahme des Bestrahlungsbereiches liefert ein verzerrtes Bild (Fischaugen-Optik), Ursache ist bereits das Objektiv. Damit das Bild möglichst verzerrungsfrei dargestellt werden kann, wird es mittels Software entzerrt. Da das Bild annähernd kugelförmig verzerrt ist, basiert der einfachste Entzerr-Algorithmus auf Kugelberechnungen bzw. Kreisberechnungen. Zu Beginn wird ein leeres Bild mit der identischen Größe wie das Original angelegt. Mit einer Schleife wird für jeden Bildpunkt, des leeren Bildes, der zugehörige Punkt im originalen Bild berechnet und übernommen. Am einfachsten Vorzustellen ist es als würde man das Bild auf der Kugel-Oberfläche auf eine Ebene aufklappen. Da für die obere Berechnung der Abstand zur Kugelmitte benötigt wird, muss aus den Koordinaten ein Radius ermittelt werden. Dieser Radius wird danach entzerrt und muss wieder in XY-Koordinaten umgerechnet werden. Für Testzwecke wurde das Rasterfoto (kariertes Blatt Papier) mit dem Algorithmus entzerrt. Die Ergebnisse sind sehr gut, denn man muss die Berechnungsdauer von 1.828 Sekunden positiv hervorheben.

4.2 Bildübergang

Würden die aneinander gereihten Bilder ohne fliesenden Übergang zusammen gefügt werden, so würde man immer harte Kanten feststellen können. Einfachster Überlauf zwischen den Bildern besteht darin, die Bilder zu überlappen und die Farben zu addieren, wobei die Farben je nach Lage des Pixels unterschiedlich gewichtet werden.

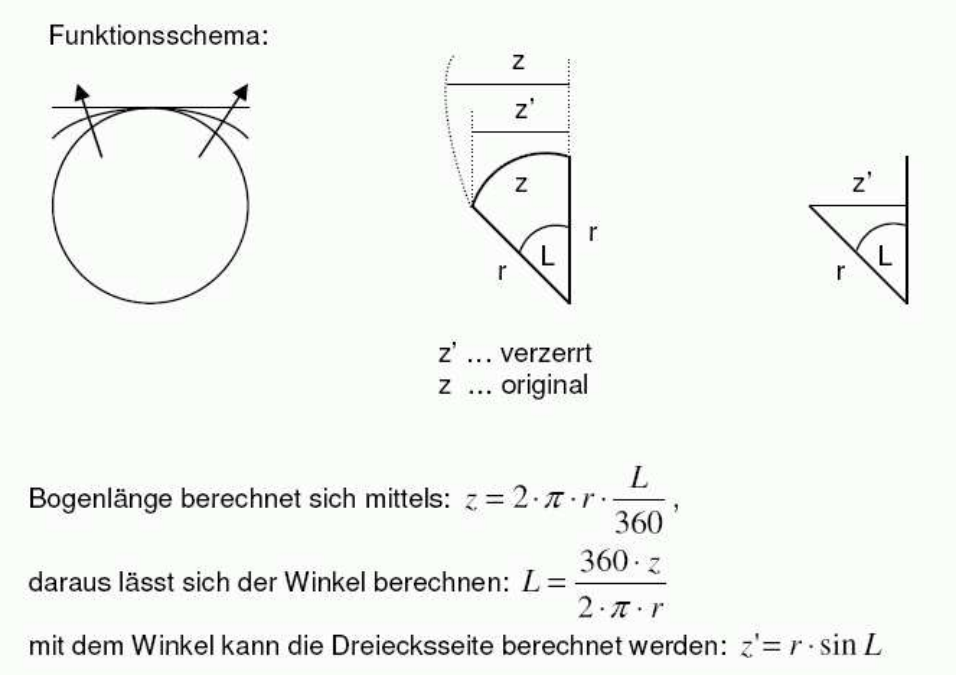
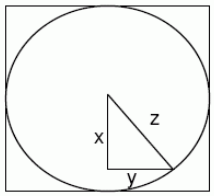


Abbildung 4.1: Schema Entzerren

4.3 Positionssuche

Damit das Zusammenfügen von Einzelbildern nicht bei einer fixen Stelle passieren muss, wird die Position mittels Bildvergleich gesucht. Damit diese Verarbeitung funktioniert, müssen die vorhandenen Bilder überlappend fotografiert werden.
Die Suche verschiebt das zweite Bild von einer maximalen Überlappungsbreite bis zum Ende des ersten Bildes und berechnet einen Wert der verglichen werden kann. Damit auch Versatzfehler ausgeglichen werden können, wird das zweite Bild auch nach oben und unten verschoben. Diese Parameter, die Berechnungsbreite und die Berechnungsgenauigkeit können durch Funktionsparameter angepasst werden.
Der Vergleichswert wird durch die Farbdifferenz aller Pixel gebildet (Aufsummieren und Mittelwert bilden). Ist der neu ermittelte Wert kleiner als der gespeicherte, so ist die aktuelle Position passender als die vorherig ermittelte Position. Der Vergleich funktioniert unterschiedlich gut, sind viele unterschiede in der horizontalen Ebene, so sind die Vergleichswerte unterschiedlicher und können somit besser verglichen werden. Ist der Überlappungsbereich aber ziemlich ähnlich so ist der Vergleich schwierig bis zu überhaupt nicht möglich.



Berechnung des Abstandes zur Mitte (leeres Bild):

$$z = \sqrt{x^2 + y^2}$$

$$k = \frac{y}{x}$$

→ z' Berechnen (obere Anleitung)

$$y' = k \cdot x'$$

$$z' = x'^2 + y'^2 = x'^2 + k \cdot x'^2$$

z' in XY-Koordinaten umrechnen (original Bild):

$$x' = \sqrt{\frac{z'^2}{1 + k^2}}$$

$$y' = k \cdot x'$$

Abbildung 4.2: Berechnung Entzerren

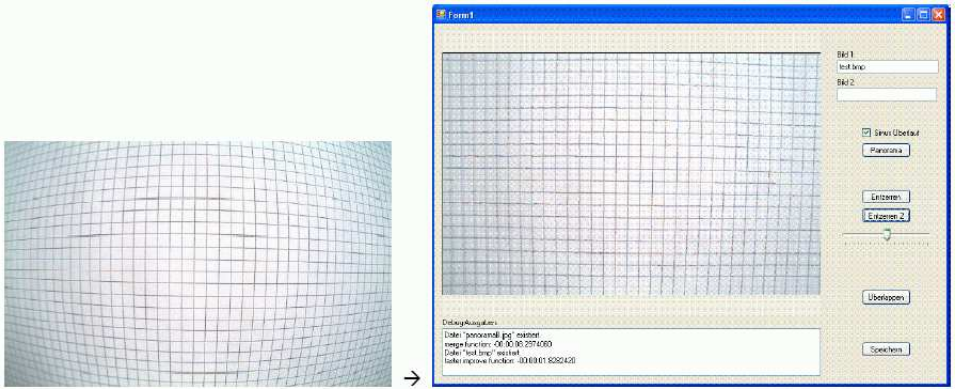


Abbildung 4.3: Entzerren

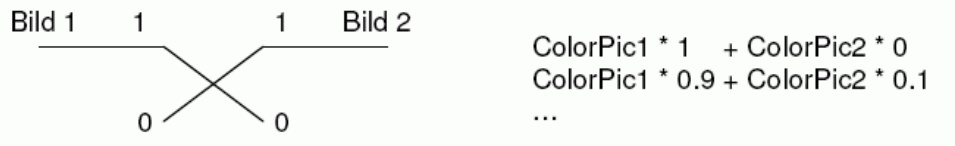


Abbildung 4.4: Schema Übergang

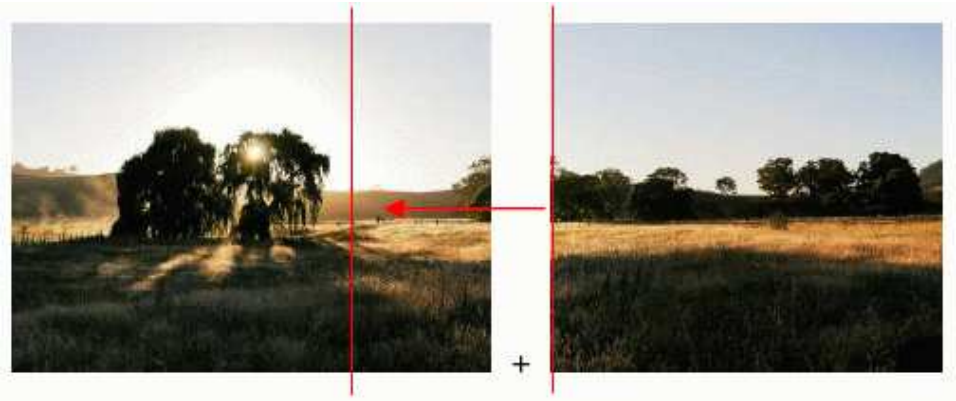


Abbildung 4.5: Original Übergang

Abbildung 4.6: Übergang

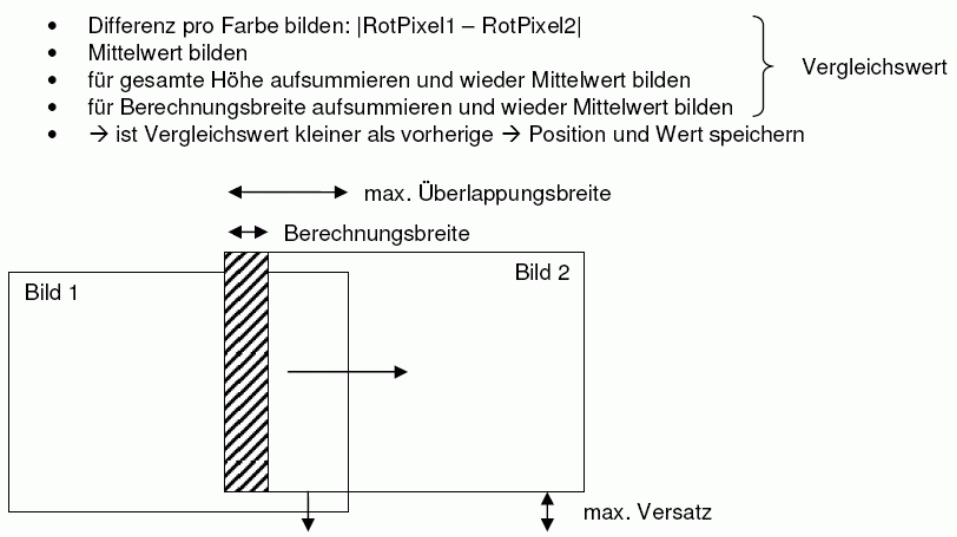


Abbildung 4.7: Schema Positionssuche

Abbildung 4.8: Teilbilder Positionssuche

Abbildung 4.9: Gesamtbild Positionssuche

Kapitel 5

Erfahrungen und Zusammenfassung

Im Laufe des Praktikums in der Firma CDE konnte ich mein, während des Studiums, gesammeltes Wissen gut einsetzen. Wichtigste Wissens-Gebiete waren die Mikroprozessor Technik, Hardwarenahe Programmierung und Softwareentwurf.
Im Bereich Hardware-Entwicklung konnte ich erste praxisnahe Erfahrungen sammeln. Diese beinhalten das Entwerfen eigener Schaltungen, das Layouten des Boards, sowie den Ablauf für die Fertigung der gesamten Platine. Ebenso konnte ich mein Wissen über Realtime Operating Systems festigen indem ich das firmeneigen Betriebssystem analysierte und für das Projekt erweiterte.
Die Firmeninternen Abläufe wurden für das Qualitäts-Management optimiert. So werden zum Beispiel Jour Fixes abgehalten um alle auf den neuesten Stand zu bringen. Des weiteren stehen für sämtliche Dokumentationen, für laufende Projekte, Vorlagen bereit. Diese sind in unterschiedliche Bereiche gegliedert. Ebenso ist die Ordnerstruktur der Repositories vorgegeben. Diese sollte weitestgehend eingehalten werden um das rasche Auffinden von Informationen und Dateien zu garantieren.
Das Betriebsklima ist hervorragend, es wird viel miteinander besprochen und es konnten auch private Dinge ausgetauscht werden. Durch das offene Verhältnis zueinander entstand eine hervorragendes Arbeitsumfeld.