

UNIVERSITÄT KASSEL  
FACHGEBIET VERTEILTE SYSTEME

Master-Projekt

**Entwicklung einer Ground Truth Kamera zur  
Verifizierung von Prozessen im Bereich  
autonomer mobiler Roboter**

Sven Steenbock

Lesekampstrasse 19

25436 Uetersen

Tel: +49 (0) 4122 2941

Matrikelnummer: 29102442



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Thema . . . . .	1
1.2	Aufgabe . . . . .	1
<b>2</b>	<b>Konzept</b>	<b>3</b>
<b>3</b>	<b>Realisierung</b>	<b>5</b>
3.1	Hardware . . . . .	5
3.1.1	Modularisierung . . . . .	5
3.1.2	Anzeige . . . . .	6
3.1.3	Ansteuerung . . . . .	9
3.1.4	Schnittstellen . . . . .	10
3.1.5	Mikrocontroller . . . . .	11
3.1.6	Spannungsversorgung . . . . .	11
3.2	Software . . . . .	12
3.2.1	Mikrocontroller Firmware . . . . .	13
3.2.2	Bildauswertung . . . . .	15
<b>4</b>	<b>Abschlussbetrachtung</b>	<b>17</b>
4.1	Konzept . . . . .	17
4.2	Hardware . . . . .	17
4.3	Software . . . . .	18
<b>5</b>	<b>Anhang</b>	<b>19</b>



# Abbildungsverzeichnis

3.1	Produktdarstellung einer Flea2 Kamera . . . . .	6
3.2	Das Anzeigemodul der Hardware . . . . .	7
3.3	Produktdarstellung einer Dot-Matrix LED Anzeige . . . . .	9
3.4	Ansicht der LED-Matrix aus Sicht der Kamera . . . . .	16
3.5	Overlay, das wichtige Zustandsdaten in das aktuelle Bild einblendet. . . . .	16
5.1	Schaltplan: Anzeigemodul . . . . .	19
5.2	Schaltplan: Basismodul . . . . .	20



# 1 Einleitung

Bei der Entwicklung von autonomen mobilen Robotern ist die zuverlässige optische Abtastung und Auswertung der Umgebung des Roboters ein wichtiger Prozessschritt. Die daraus gewonnenen Daten müssen eine hohe Genauigkeit und Zuverlässigkeit aufweisen, da sie als Grundlage für viele weitere Aufgaben dienen. Da einige Eigenschaften, wie beispielsweise die zeitliche Auflösung, die sich im Bereich von Millisekunden befindet, nicht ohne Hilfsmittel verifiziert werden können, soll eine Möglichkeit gefunden werden, die gewonnenen Daten mithilfe einer zusätzlichen Komponente überprüfen zu können.

## 1.1 Thema

Das Thema dieser Projektarbeit ist die Entwicklung einer Ground Truth Kamera zur Verifizierung von Prozessen im Bereich autonomer mobiler Roboter. Dabei liegt der Schwerpunkt dieser Arbeit in der Ermittlung und Auswertung der exakten Aufnahmezeitpunkte von Kamerabildern.

## 1.2 Aufgabe

Es soll ein Konzept ausgearbeitet und umgesetzt werden. Dieses Konzept beinhaltet die kontinuierliche Aufnahme von Bildern einer beliebigen Szene mit einer Kamera. Die Bilder sollen für die weitere Verarbeitung abgespeichert werden. Die Aufnahmezeitpunkte jedes einzelnen Bildes müssen mit einer geeigneten Technik mit einer Genauigkeit im einstelligen Millisekundenbereich erfasst werden können. Diese Erfassung muss aufgrund der großen Bildanzahl die sich durch die kontinuierliche Bildaufnahme ergibt automatisiert ablaufen. Die ermittelten Aufnahmezeitpunkte müssen mit dem Bild verknüpft abgespeichert werden damit sie für weitere Verarbeitungsschritte zur Verfügung stehen.





## 2 Konzept

Der kritische Punkt der Aufgabenstellung ist die exakte Ermittlung der Aufnahmezeitpunkte. Dies ist auf mehreren Wegen zu erreichen:

- Der naheliegende Ansatz ist die Verwendung einer Kamera die entweder eine vernachlässigbar geringe oder eine vollkommen deterministische Verzögerung zwischen Bildaufnahme und Bildausgabe an das verarbeitende Gerät, in diesem Fall einem Notebook, besitzt. Damit ist es möglich, von dem Zeitpunkt zu dem ein Bild an das Notebook übermittelt wird direkt auf den Aufnahmezeitpunkt zu schließen. Ein signifikanter Nachteil dieses Konzepts ist jedoch, dass dadurch keine der etablierten Kameraschnittstellen genutzt werden kann. Das sind unter anderem Firewire, Ethernet oder vereinzelt USB. All diese Schnittstellen haben gemein, dass sowohl auf der Sende- als auch auf der Empfangsseite komplexe Operationen, die die Daten unter anderem puffern, ausgeführt werden müssen. Dies erzeugt nicht exakt bestimmbare Verzögerungen. Ein Verzicht auf diese Schnittstellen erfordert wesentlich teurere, speziell angepasste Hardware.
- Eine weitere Möglichkeit ist die Verwendung von intelligenten Kameras. Dabei handelt es sich um Kameramodule auf denen bereits ein Prozessor integriert ist. Damit entfällt eine Schnittstelle mit den genannten Nachteilen, da die Kamera über eine direkte parallele Verbindung an den Prozessor angeschlossen ist. Bei diesem Konzept kann der Prozessor auf dem Kameramodul zu jedem Bild den exakten Aufnahmezeitpunkt bestimmen bevor es über eine Schnittstelle mit nicht deterministischer Verzögerung an ein Notebook zur Weiterverarbeitung gesendet wird. Der Nachteil dieses Konzepts liegt ebenfalls in den hohen Kosten, die durch die intelligenten Kameras entstehen. Darüber hinaus ist die Produktpalette dieser Kameras wesentlich kleiner als die der Ethernet- oder Firewirekameras ohne Prozessor, sodass unter Umständen nicht alle gewünschten optischen Eigenschaften zur Verfügung stehen.
- Neben den Konzepten, die den Aufnahmezeitpunkt nach der Bildaufnahme bestimmen, gibt es die Möglichkeit den Aufnahmezeitpunkt bereits während der Aufnahme im Bild zu kodieren. Man kann sich das Konzept so vorstellen, dass man eine Uhr in der Szene anbringt. Jedes Bild hält automatisch seinen eigenen Aufnahmezeitpunkt fest, unabhängig von Verzögerungen. Für die geforderte Aufgabe ist eine klassische Uhr jedoch ungeeignet, da zum einen Zeitauflösungen im Bereich von Millisekunden erreicht

werden sollen und zum anderen eine automatisierte Auswertung stattfinden soll. Eine geeignete Anzeigemöglichkeit ist die Kodierung der Zeit millisekundengenau mithilfe von LEDs, die im Randbereich des Bildes angebracht sind. Vorteil dieses Konzepts ist die vollständige Eliminierung von Fehlern durch Verzögerungen im Aufnahmeprozess. Desweiteren können weiterhin beliebige Kamera- und Schnittstellentypen verwendet werden. Der Nachteil dieses Konzepts liegt in der Notwendigkeit einer zusätzlichen Hardware zur Anzeige der Zeit. Zusätzlich ist bei diesem Konzept eine Auswertung der Bilddaten erforderlich um die Zeitinformation aus dem Bild zu extrahieren.

Alle genannten Konzepte besitzen die Gemeinsamkeit, dass das Element das den Aufnahmezeitpunkt ermittelt bzw. die Zeitinformation erzeugt selbst über eine ausreichend genaue Zeitbasis verfügen muss. Zusätzlich muss diese Zeitbasis für den späteren Abgleich der Daten mit dem zu überprüfenden System mit diesem synchronisiert sein. Bei den Konzepten bei denen ein Notebook die Zeit ermittelt stellt dies keinen zusätzlichen Aufwand dar, da auf bekannte Techniken, wie beispielsweise das Network-Time-Protokoll zurückgegriffen werden kann. Bei dem letztgenannten Konzept muss diese Anforderung gesondert betrachtet werden.

Als geeignetes Konzept wurde die optische Kodierung der Zeitinformation im Bild ausgewählt, da die erforderlichen Hardware aufgrund einer überschaubaren Komplexität selbst entwickelt werden kann und alle weiteren Komponenten bereits vorhanden sind. Die Aufgabe der Zeitsynchronisation der Hardware wird durch die Implementierung des Network-Time-Protokolls auf der Hardware gelöst.

## 3 Realisierung

Die folgenden Kapitel dokumentieren die konkrete Umsetzung des in der Einleitung ausgewählten Konzepts. Dabei wird mit der Beschreibung der Hardware begonnen. In dem darauf folgenden Teil, der die Software beschreibt, wird sowohl die Firmwareprogrammierung des auf der Hardware eingesetzten Prozessors, als auch das Programm zur Extraktion der Zeitinformation aus den aufgenommenen Bildern erläutert.

### 3.1 Hardware

Die Hardware besteht zum einen aus Komponenten zur Bildaufnahme und zum anderen aus Komponenten zur Erzeugung und Anzeige der Zeitinformation. Zur Bildaufnahme wird eine Kamera der Firma Point Grey mit der Bezeichnung Flea2 genutzt (Abb. 3.1). Die technischen Daten dieser Kamera sind in Tab. 3.1 aufgeführt. Je nach eingesetztem Computer bzw. Notebook wird darüber hinaus ein Gerät zur Verbindung der Schnittstelle mit dem Computer benötigt. In dem Testaufbau wird ein Notebook genutzt, das keine direkte Anschlussmöglichkeit für die Kamera besitzt, sodass eine Expresscard mit einer Firewire-B Schnittstelle verwendet wird. Zusätzlich wird ein Firewire Hub eingesetzt, das jedoch lediglich dazu dient die Kamera mit Spannung zu versorgen, da die Schnittstellenkarte nicht in der Lage ist, ausreichend Leistung zur Verfügung zu stellen.

Den zweiten Teil der Hardware bilden die Komponenten, die zur Darstellung der Zeitinformationen genutzt werden. Da es sich dabei um speziell für dieses Projekt entwickelte Komponenten handelt, werden diese im folgenden näher erläutert.

#### 3.1.1 Modularisierung

Es gibt mehrere Aspekte, die nahelegen, den zweiten Teil der Hardware auf mehrere Komponenten aufzuteilen. Die Aufgabe dieses Teils ist die Anzeige der Zeitinformationen. Dadurch ist es per Definition notwendig, dass sich ein Teil dieser Hardware im von der Kamera aufgenommenen Bildbereich befindet. Gleichzeitig soll der von der Hardware eingenommene Bildbereich so klein wie möglich gehalten werden um die primäre Aufgabe der Kamera, die Aufnahme einer Szene zur Überprüfung von Roboterdaten, nicht einzuschränken. Durch eine Unterteilung der Hardware in ein Modul das ausschließlich die Anzeige übernimmt und ein weiteres das alle übrigen Aufgaben übernimmt kann flexibler auf diese Anforderung ein-

Typ	FL2G-13S2M/C
Auflösung	1288 x 964
Bildwiederholrate	30FPS (bei voller Auflösung)
Schnittstelle	Firewire-B (IEEE 1394b)
Datentransferrate	100, 200, 400, 800 Mbit/s
Shuttergeschwindigkeit	0,02ms - 10s
Datenformate	Y8, Y16, RGB, YUV411, YUV422, YUV444 and raw

**Tabelle 3.1:** Die technischen Daten des Kamera[Poi09]

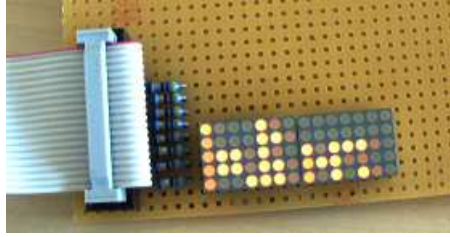


**Abbildung 3.1:** Produktdarstellung einer Flea2 Kamera. Quelle: PointGrey[Poi11]

gegangen werden. Das Anzeigemodul kann mit minimalen Abmessungen gefertigt werden, während die Abmessungen des Steuermoduls nicht relevant sind. Einen weiteren Vorteil bietet die Möglichkeit das Anzeigemodul unabhängig vom Steuermodul auswechseln zu können. Somit können mehrere Anzeigekonzepte evaluiert werden ohne den Steuerteil mehrfach fertigen zu müssen. Abb. 3.2 zeigt das Anzeigemodul der Hardware.

### 3.1.2 Anzeige

Die Aufgabe der Hardware ist die millisekundengenaue Anzeige der aktuellen Zeit in einem einfach auswertbaren Format. Als logisches Format wird eine veränderte Form des NTP-Zeitstempels verwendet. Ein NTP-Zeitstempel ist 64Bit lang und kodiert in den ersten 32Bit die Sekunden seit dem 01.01.1900 und in den zweiten 32Bit die Sekundenbruchteile. Damit hat ein NTP-Zeitstempel eine theoretische Genauigkeit von 0,23Nanosekunden. Da die Aufgabenstellung lediglich eine Auflösung im Bereich von Millisekunden fordert und eine höhere Auflösung durch andere Faktoren, wie zum Beispiel der Zeit die für die Bildaufnahme selbst



**Abbildung 3.2:** Das Anzeigemodul der Hardware

benötigt wird, beschränkt wird, reicht ein Zeitstempel mit 10Bit für die Sekundenbruchteile. Damit ergibt sich eine Gesamtlänge von mindestens 42Bit.

Bei der Darstellung des Zeitstempels muss zum einen sichergestellt sein, dass dieser zuverlässig erkannt werden kann und zum anderen ein Verfahren benutzt werden, welches diese Aufgabe mit möglichst geringem Rechenaufwand erfüllt. Ein Konzept, das diese Anforderungen erfüllt, ist die direkte Darstellung aller Bits mittels LEDs in einer LED-Matrix. Nachdem die Position der LED-Matrix im Bild ermittelt wurde, kann der Zustand jedes Bits über eine Messung der Helligkeit erfolgen. Zur weiteren Optimierung der Darstellung wurden folgende Konzepte evaluiert:

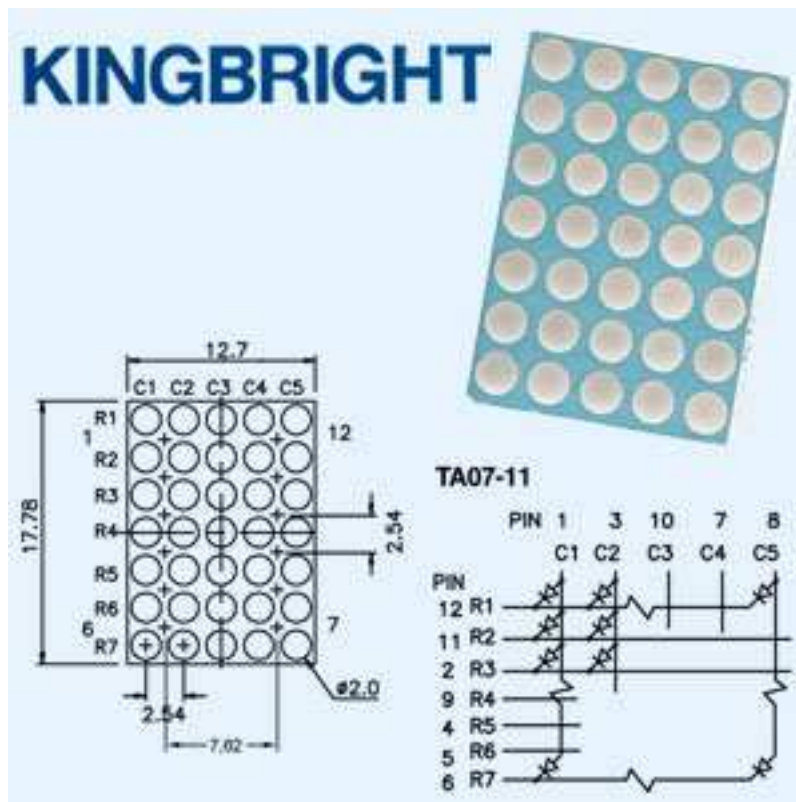
- Da es sich bei der verwendeten Kamera um eine Farbkamera handelt, bestand ein erster Test in der Kodierung mehrerer Zustände durch verschiedenfarbige LEDs. Dadurch kann die Anzahl der benötigten LEDs reduziert werden, was die insgesamt von der LED-Matrix belegte Fläche verringert. Im konkreten Beispiel wurden zweifarbige LEDs zur Darstellung von drei Zuständen genutzt (Dunkel = Zustand1, Rot = Zustand2, Grün = Zustand3). Es wurde ein Anzeigemodul aufgebaut das 30 zweifarbige LEDs in zwei Reihen a 15 LEDs enthält. Zusätzlich wurden am Anfang der ersten und am Ende der zweiten Reihe eine blaue LED, die zur Erkennung der Position der LED-Matrix dient, eingefügt. Ohne die blauen LEDs lassen sich damit theoretisch  $3^{30} = 2^{47}$  Zustände, also 47Bit, darstellen. Der Test hat jedoch ergeben, dass der Bildsensor der Kamera von der Helligkeit der LEDs übersteuert wird, sodass jede LED als überwiegend weiß erkannt wird. Im Randbereich der LED läßt sich die Farbe erkennen, bei einem geringen Abstand zu der nächsten LED in der Matrix entstehen jedoch Überlappungseffekte, sodass das Verfahren insgesamt ungeeignet ist.
- Eine weitere Option zur Kodierung mehrere Zustände mit einer LED ergibt sich durch die Nutzung mehrerer Helligkeitsstufen. Einige kurze Tests haben jedoch ergeben, dass bereits eine Aufteilung des maximalen Helligkeitsbereichs auf drei Zustände zu großen Fehlern bei der Detektion führt. Dies hat zwei Gründe:
  - Zum einen sorgen Schwankungen im Umgebungslicht dazu, dass sich die absoluten Helligkeitswerte stark verändern. Dies wird unter anderem dadurch verstärkt,

dass die Kamera durch automatische Gain-Anpassung versucht, diese Effekte zu kompensieren, was sich negativ auf die sich nicht ändernden Helligkeitswerte der LEDs auswirkt.

- Ein weiterer Grund ist die Tatsache das in dem Steuermodul nicht vorgesehen war die Helligkeit zu regeln. Die LEDs werden über ein Multiplexverfahren angesteuert das alle Zeilen der Matrix nacheinander aktiviert. Der scheinbare Effekt dass mehrere Zeilen gleichzeitig aktiv sind ergibt sich durch die Trägheit des menschlichen Auges oder, im Fall der Kamera, durch die Aktivierung aller Zeilen innerhalb einer Belichtungsphase. Um zusätzlich die Helligkeit einzelner LEDs steuern zu können muss das Multiplexverfahren so erweitert werden dass innerhalb einer Belichtungsphase jede LED potentiell mehrere Einschaltphasen durchläuft. Wird eine LED nur in jeder zweiten Einschaltphase genutzt, ergibt dies scheinbar die halbe Leuchtkraft. Der Nachteil dieses Verfahrens ist die insgesamt geringere Multiplexfrequenz sodass dadurch die minimale Belichtungszeit erhöht wird.
- Neben den Konzepten zur Verringerung der LED-Anzahl wurden auch verschiedene Optionen bei der Wahl der LEDs evaluiert. Bei den bisherigen Versuchen wurden einzelne bedrahtete LEDs verwendet die auf einer Leiterplatte zu einer Matrix angeordnet wurden. Bei der Beschränkung auf einfarbige LEDs können bereits vollintegrierte LED-Matrix Bausteine mit 5x7 LEDs genutzt werden (Abb. 3.3). Der Vorteil dieser Bausteine liegt in der hohen geometrischen Präzision, mit der die LEDs angeordnet sind. Dadurch werden Fehler bei der Erkennung einzelner LEDs vermieden. Darüber hinaus verhindert der Aufbau ein optisches Übersprechen zwischen den LEDs. Durch den hohen Öffnungswinkel und Streulicht tritt dieser Effekt bei bedrahteten LEDs auf. Der Nachteil dieses Konzepts ergibt sich durch die festgelegten Eigenschaften der LED-Matrix Bausteine.

Da die Erzeugung des Zeitstempels und die Bildaufnahme nicht synchron ablaufen, muss der Fall berücksichtigt werden, dass ein Bild während des Übergangs zwischen zwei Zeitstempeln aufgenommen wird. Dadurch entsteht ein Bild, dass die Überlagerung der beiden Zustände anzeigt. Der Nachteil dieser Überlagerung liegt darin, dass einige Zustände nicht eindeutig bestimmten Zeitstempelübergängen zugeordnet werden können. Daher wurde entschieden, zusätzlich zu dem binären Zeitstempel einen Indikator in Form eines laufenden LED-Punkts über mehrere LEDs zu nutzen (im Testaufbau wurden hierfür 8 LEDs verwendet). Damit ist es zum einen möglich, jeden Zustandsübergang eindeutig zuzuordnen zu können und zum anderen kann bei mehreren Zustandsübergängen pro Bild die Anzahl der Zustandsänderungen ablesen zu können. Es ist somit möglich die Belichtungszeit der Kamera messen bzw. überprüfen zu können.

Der Schaltplan des Anzeigemoduls ist im Anhang zu finden (Abb. 5.1).



**Abbildung 3.3:** Produktdarstellung einer Dot-Matrix LED Anzeige. Quelle: Kingbright[Kin11]

### 3.1.3 Ansteuerung

In dem vorausgegangenen Kapitel wurde ein Anzeigekonzept ausgewählt, das eine große Anzahl von LEDs nutzt. Um den Schaltungs- und Steueraufwand gering zu halten, muss eine geeignete Ansteuerung ausgewählt werden. Ein sogenanntes Multiplexverfahren bietet die Möglichkeit, mit einer geringen Anzahl von Steuerleitungen eine große Anzahl von LEDs anzusteuern. Dazu werden die LEDs in einem zweidimensionalen Feld in Zeilen und Spalten angeordnet. Der minimale Schaltungsaufwand bei maximaler LED-Anzahl ergibt sich wenn die Zeilenanzahl gleich der Spaltenanzahl gewählt wird. Nun werden in jeder Zeile die Anoden aller LEDs miteinander verbunden sowie in jeder Spalte die Kathoden miteinander verbunden. Somit erhält man für jede Zeile eine Anodenleitung und für jede Spalte eine Kathodenleitung. Eine beliebige LED kann nun angesteuert werden, indem eine Spannung an die Anodenleitung der Zeile in der sich die LED befindet und die Kathodenleitung der Spalte in der sich die LED befindet angelegt wird. Beispielsweise können mit 16 Steuerleitungen im Idealfall  $8 \times 8 = 64$  LEDs angesteuert werden.

Der Nachteil dieses Konzepts besteht darin, dass es nicht möglich ist, beliebige LEDs gleichzeitig anzusteuern. Lediglich beliebige LEDs einer Zeile oder einer Spalte können gleichzeitig angesteuert werden. Um trotzdem ein vollständiges Bild mit einer beliebigen Kombination an aktivierten LEDs zu erhalten, wird ein Zeitmultiplexverfahren angewandt. Dabei werden die benötigten LED Zeilen nacheinander aktiviert, wobei die Schaltfrequenz so hoch gewählt wird, dass das Auge den Umschaltvorgang nicht wahrnimmt und ein statisches Gesamtbild erkennt.

Für dieses Projekt ebenfalls relevant ist die Tatsache, dass das Multiplexverfahren auch mit einer Kamera genutzt werden kann. Dabei müssen alle Zeilen der Matrix während einer Belichtungsphase aktiviert werden. Daraus kann die folgende Mindestfrequenz für das Multiplexverfahren errechnet werden: Bei einer Bildwiederholrate von 50Hz beträgt die maximale Belichtungszeit 20ms. Bei einer LED-Matrix mit 8 Zeilen müssen diese in dieser Zeit mindestens einmal vollständig durchlaufen werden. Die Multiplexfrequenz beträgt demnach mindestens 400Hz.

Ist die Belichtungszeit der Kamera nicht konstant, muss eine wesentlich höhere Multiplexfrequenz gewählt werden, da andernfalls Zeilen der Matrix während der Belichtung bereits ein zweites mal aktiviert werden können, was zu signifikanten Helligkeitsunterschieden führt. Bei einer Multiplexfrequenz die beispielsweise um den Faktor 100 über der Mindestfrequenz liegt tritt der Effekt ebenso auf, dadurch dass in diesem Fall 100 Aktivierungsphasen gegenüber 99 Aktivierungsphasen auftreten, ist der Helligkeitsunterschied jedoch minimal.

### 3.1.4 Schnittstellen

Wie im Konzept bereits erläutert, soll die Zeitsynchronisierung mithilfe des Network-Time-Protocols erfolgen. Dazu muss die Anzeigehardware über eine Netzwerkschnittstelle verfügen. Der überwiegende Teil aller Netzwerke arbeitet mit der Ethernet-Technologie, weswegen diese Technologie genutzt werden soll.

Eine Möglichkeit zur Anbindung einer 10Mbit Ethernetschnittstelle an einen Mikrocontroller bietet der Schnittstellenbaustein ENC28J60 der Firma Microchip. Dieser Baustein ist insbesondere bei der Nutzung kleinerer Mikrocontroller sinnvoll, da er über eine serielle Schnittstelle angesteuert wird, eigenen Speicher zur Pufferung von Netzwerkdaten enthält sowie einen Großteil der benötigten Netzwerkfunktionen, inklusive einiger Checksummenberechnungen, bereitstellt. Das Network-Time-Protocol erzeugt nur geringe Datenmengen sodass die Übertragungsrate von 10Mbit ausreichend ist. Desweiteren sind die aktuell genutzten Technologien mit 100Mbit und 1Gbit abwärtskompatibel zu der 10Mbit Technik, sodass diese gemischt betrieben werden können. Weitere technische Daten des Bausteins sind unter Tab. 3.2 aufgeführt.

Die Beschaltung des Schnittstellenbausteins entspricht der im Datenblatt angegebenen Beispielschaltung. Neben der Minimalkonfiguration bei der der Baustein lediglich per SPI



Hersteller	Microchip
Produktbezeichnung	ENC28J60
Externe Schnittstelle	Ethernet
Ethernet Standard	IEEE 802.3 (10BASE-T)
Interne Schnittstelle	SPI
Interne Geschwindigkeit	10Mbit/s
Speicher	8KByte
Versorgungsspannung	3,3V

**Tabelle 3.2:** Die technischen Daten des Ethernet Bausteins[Mic04]

mit dem Mikrocontroller verbunden ist, kann zusätzlich ein Interruptsignal genutzt werden. Aufgrund der später erläuterten Struktur der Firmware wird dies jedoch nicht genutzt. Eingehende Pakete werden per polling<sup>1</sup> registriert.

### 3.1.5 Mikrocontroller

Als Mikrocontroller wird ein ATmega32 der Firma Atmel genutzt. Die technischen Daten dieses Bausteins sind unter Tab. 3.3 aufgelistet. Aus diesen Daten geht hervor, dass der Mikrocontroller über eine SPI Schnittstelle verfügt, die für den Anschluss des Ethernet-Schnittstellenbausteins benötigt wird. Desweiteren verfügt der Mikrocontroller über 32 nutzbare Eingangs- bzw. Ausgangspins, sodass die LED-Matrix mit 16 Steuerleitungen realisiert werden kann. Die Programmierung des Mikrocontrollers erfolgt über eine serielle Schnittstelle, die es erlaubt, die Firmware direkt in der Zielhardware anzupassen. Die maximale Taktfrequenz des Mikroprozessors beträgt 16MHz, was für das angestrebte Multiplexverfahren ausreichend ist.

### 3.1.6 Spannungsversorgung

Zum Betrieb der Hardware werden drei verschiedene Spannungspegel benötigt:

- 3,3V werden zum Betrieb des Netzwerkbausteins benötigt
- Der Mikrocontroller kann mit einer Spannung zwischen 3,3V und 5V betrieben werden, sodass es möglich ist, diesen ebenfalls mit den ohnehin benötigten 3,3V zu versorgen.

<sup>1</sup>Polling bezeichnet das zyklische Überprüfen einer bestimmten Bedingung. Konzeptbedingt ergibt sich aus der Zykluszeit die maximale Latenz zwischen Änderung des Zustands und der Erkennung des Zustandswechsels.

Hersteller	Atmel
Produktbezeichnung	ATmega32
Flash-Speicher	32KByte
RAM-Speicher	2KByte
EEPROM-Speicher	1KByte
Taktgeschwindigkeit	bis 16MHz
Versorgungsspannung	4,5V bis 5,5V
I/O-Pins	32
Schnittstellen	SPI, UART, I2C

**Tabelle 3.3:** Die technischen Daten des Mikrocontrollers[Atm11]

Allerdings ist die maximale Taktfrequenz von der Versorgungsspannung abhängig, so dass bei 3,3V keine 16MHz Systemtakt erreicht werden. Daher wird für den Mikrocontroller eine weitere Spannung mit 5V genutzt.

- Eine dritte Spannung wird zur Ansteuerung der LED-Matrix genutzt. Aufgrund von Spannungsabfällen über den Schalttransistoren und den LEDs bliebe bei einer Versorgungsspannung von 5V nur wenig Spielraum zur Stromregulierung mittels Vorwiderständen. Daher wird für diesen Teil der Schaltung eine Versorgung von 9V bis 12V benötigt.

Die Erzeugung der Versorgung der LEDs erfolgt extern mithilfe eines Steckernetzteils. Dieses erzeugt eine Ausgangsspannung von ungefähr 9V<sup>2</sup>. Es erfolgt keine weitere Aufbereitung dieser Spannung.

Die Erzeugung der beiden weiteren Spannungen geschieht mithilfe von Linearreglern. Linearregler eignen sich aufgrund der einfachen Beschaltung für diese Aufgabe. Der Nachteil von Linearreglern, der geringe Wirkungsgrad, kann aufgrund der insgesamt niedrigen Leistungsaufnahme vernachlässigt werden.

Der komplette Schaltplan des Basismoduls befindet sich im Anhang (Abb. 5.2).

## 3.2 Software

In den folgenden Kapiteln wird die entwickelte Software beschrieben. Dabei gliedert sich die Software in zwei Bereiche:

---

<sup>2</sup>Es handelt sich um ein unregelmäßiges Netzteil, daher steigt die Spannung bei fehlender Belastung um einige Volt an. Diese Schwankungen liegen jedoch im Toleranzbereich der Bauteile.

- Zum einen existiert eine Firmware auf dem Mikrocontroller, die die Ansteuerung der LED-Matrix, die Ansteuerung des Netzwerkbausteins sowie die notwendigen Operationen zur Auswertung und Anzeige des Zeitstempels übernimmt.
- Zum anderen gibt es eine Software die auf dem mit der Kamera verbundenen PC ausgeführt wird. Sie realisiert die Bildaufnahme und besitzt Bildauswertungsalgorithmen zur Erkennung des Zeitstempels.

### 3.2.1 Mikrocontroller Firmware

Die Firmware, die die Funktion des Mikrocontrollers realisiert, besteht aus mehreren Teilen. Da es sich bei dem Mikrocontroller um einen einfachen 8Bit Controller ohne Speicherschutzmechanismen oder ähnlichem handelt, sind die verschiedenen Module in einem einzigen Programm realisiert.

#### 3.2.1.1 LED-Matrix Ansteuerung

Die Ansteuerung der LED-Matrix ist weitestgehend in Software realisiert. Ein Timer ist so programmiert, dass in festen Intervallen ein Interrupt ausgelöst wird. Um eine minimale Ausführungszeit dieses Interrupt und damit die kleinstmögliche Unterbrechung des Hauptprogramms zu erreichen, ist der Interrupt in Assembler programmiert. Innerhalb des Interrupts werden zwei Aktionen ausgeführt. Zum einen ist zu jedem Zeitpunkt genau eine Zeile der Matrix aktiv. Bei Ausführung des Interrupts wird jeweils in die nächste Zeile, bzw. von der letzten Zeile in die erste Zeile gewechselt. Da es sich genau um 8 Zeilen handelt<sup>3</sup>, kann eine einfache Shift-Operation genutzt werden, die ein einzelnes Bit zyklisch durch eine 8Bit Variable rotiert und anschließend auf einem Port ausgibt. Die zweite Aktion ist die Ausgabe der passenden Spalte aus einem Array auf einem weiteren Port. Dazu wird ein Zähler genutzt, dessen Wert zur Adressierung des Arrays genutzt wird.

#### 3.2.1.2 Netzwerkbaustein Ansteuerung

Die Ansteuerung des Netzwerkbausteins erfolgt, wie bereits erwähnt, über die SPI-Schnittstelle. Da diese im Mikrocontroller bereits als Hardwaremodul vorhanden ist, kann die Kommunikation auf Byte- statt Bitebene gesteuert werden. Desweiteren sorgt die im Netzwerkbaustein vorhandene Intelligenz dafür, dass viele Operationen, wie beispielsweise das Auslesen von Paketdaten, stark vereinfacht werden.

Auf die Funktionen, die Hardwareansteuerung des Netzwerkbausteins und damit gewissermaßen die Ebene 1 und 2 der Netzwerkübertragung realisieren, folgt ein rudimentärer IP-Stack. Der Stack realisiert alle zwingend erforderlichen Funktionen auf IP Ebene wie

---

<sup>3</sup>In der LED-Matrix werden nur 7 Zeilen genutzt, in der Software und der Ansteuerung sind jedoch 8 Zeilen realisiert.

beispielsweise ARP oder IP Pakete selbst. Darüber hinaus beherrscht der Stack Ping Pakete, also ICMP, und DHCP. Auf der IP Ebene aufbauend ist UDP implementiert. Eine Unterstützung für TCP existiert nicht. Die Struktur dieses Stacks ist maximal modular gestaltet, sodass alle Module per Define-Direktiven im Quellcode aktiviert und konfiguriert werden können.

### 3.2.1.3 Zeitberechnung

Die fortlaufende Zeit wird mittels eines Zählers realisiert. Der Zähler ist in einem Interrupt, der von einem Timer angesteuert wird, implementiert. Aufgrund der Tatsache, dass die Frequenz des Timers von dem Systemtakt und damit von dem 16MHz Quarz, das keine hohe Genauigkeit besitzt, abhängt, reicht die Genauigkeit eines fest programmierten Zählers nicht aus. Daher ist der Zähler so aufgebaut, dass das Inkrement des Zählers angepasst werden kann. Die Methode zum Setzen der aktuellen Zeit überprüft dazu bei jeder Aktualisierung ob der Zähler bis zu der Aktualisierung zu schnell oder zu langsam lief und passt das Inkrement entsprechend an. Praktische Versuche haben gezeigt, dass die Abweichungen mit diesem Verfahren innerhalb weniger Minuten so gering sind, dass sie nicht mehr messbar sind<sup>4</sup>.

### 3.2.1.4 Zeitstempel Auswertung

Als letztes Glied in der Kette wird ein Modul benötigt, welches die aktuelle Zeit mithilfe des Network-Time-Protocols über das Netzwerk ermittelt. Es wurde eine Basisimplementierung nach RFC5905 vorgenommen. Diese Implementierung unterstützt keinerlei Sonderfunktionen wie dem "Kiss of Death" oder einer Fehlerbehandlung oder Plausibilitätsprüfung. Es sollten daher nur NTP-Server genutzt werden die zuverlässig und im Idealfall im eigenen Netzwerk arbeiten. Die Implementierung arbeitet vollständig mit 64Bit Zeitstempeln.

### 3.2.1.5 Ablaufsteuerung

Mithilfe der vorgestellten Module kann die Funktion der main-Methode als Reihe einfacher Aktionen beschrieben werden. Nachdem alle Module und Schnittstellen initialisiert wurden, sendet die main-Methode in regelmäßigen Zeitabständen Anfragen an den konfigurierten NTP-Server. Eingehende Antworten werden ausgewertet und aktualisieren entsprechend den lokalen Zeitstempel. Der lokale Zeitstempel wird kontinuierlich an die LED-Matrix zur Anzeige gesendet.

---

<sup>4</sup>Aufgrund fehlender Spezialhardware für genaue Zeitanalysen wurden die Messungen mit Wireshark auf einem Laptop vorgenommen. Dazu wurde die Hardware so konfiguriert, dass sie zu bestimmten Zeitpunkten Pakete verschickt. Die Abweichungen waren konstant und kleiner als 1ms.

### 3.2.2 Bildauswertung

Die Bildauswertung bildet den Teil der Software der auf einem an der Kamera angeschlossenen PC ausgeführt wird. Aufgrund des gewählten Konzepts, den Zeitstempel als Teil der Bildinformationen zu kodieren, ist keine Verbindung zwischen PC und der entwickelten Hardware erforderlich.

Die Software übernimmt folgende Aufgaben, die in getrennten Modulen implementiert sind:

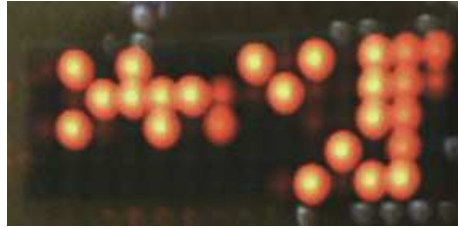
- Ansteuerung der Kamera zur Bildaufnahme und Konfiguration der Kameraparameter
- Automatische Erkennung der Position der LED-Matrix
- Erkennung der einzelnen LEDs der Matrix und Übersetzung der LED-Zustände in Bitinformationen
- Konvertierung der Bitinformationen zu einem Zeitstempel
- Oberfläche zur Konfiguration diverser Modulparameter sowie zur Anzeige des aktuellen Bilds und Statusinformationen inklusive des Zeitstempels

Einen Großteil der Aufgaben des Moduls zur Ansteuerung der Kamera übernehmen Methoden aus der OpenCV-Bibliothek[Ope11], die in diesem Projekt genutzt wird. Die Bibliothek stellt eine einheitliche Schnittstelle zur Ansteuerung von Kameras inklusive Konfigurationsschnittstelle zur Verfügung und enthält Treiber für diverse Kamertypen. Leider enthält die Bibliothek für die genutzte Kamera nur einen rudimentär implementierten Treiber der ausschließlich das Auslesen von Bildern ermöglicht. Die überwiegende Arbeit bei der Entwicklung dieses Moduls lag daher in der Erweiterung des OpenCV Treibers. Im Detail wurde die Möglichkeit geschaffen, Parameter wie die Belichtungszeit und die Verstärkung einzustellen sowie weitere Bildformate auszuwählen.

Die Konfiguration der Position der LED-Matrix kann auf zwei Arten erfolgen: Zum einen kann die Position mithilfe der grafischen Oberfläche manuell festgelegt werden, zum anderen kann ein Bilderkennungsalgorithmus die Position erkennen. Wird der Algorithmus genutzt, muss der PC über eine Verbindung zu dem Hardwaremodul verfügen, da zur Erkennung ein bestimmtes Bildmuster auf dem Modul aktiviert werden muss. Das Modul fordert zuerst ein Bild mit ausgeschalteter LED-Matrix und darauf mit eingeschaltetem Muster an. Die Differenz der beiden Bilder zeigt die Teile der Szene, die sich verändert haben, im Idealfall ausschließlich die LED-Matrix. Durch Auswertung des Musters kann die genaue Lage und Ausrichtung der LED-Matrix ermittelt werden. Im einfachsten Fall sind bei dem erwähnten Muster die LEDs in den Ecken der Matrix aktiviert sodass das deren erkannte Positionen ein Rechteck bilden, das der LED-Matrix entspricht.

Abb. 3.4 zeigt den Ausschnitt eines erfassten Bildes, der die LED-Matrix enthält. Die Erkennung der Zustände der einzelnen LEDs erfolgt ebenfalls mittels Differenzbild-Bildung.

Dazu wird das gleiche Bild verwendet, welches bereits zur Erkennung der Matrixposition genutzt wurde. Dieses dient als Schwarzbild und wird von jedem empfangenen Bild abgezogen. Soweit sich die Matrixposition oder die gesamte Bildhelligkeit nicht verändert hat, ergibt die Differenz ein Bild das im Bereich der Matrix die deaktivierten LEDs komplett schwarz und die aktivierten LEDs leicht dunkler als im Originalbild zeigt. Somit kann der Zustand der einzelnen LEDs als einfacher Vergleich mit einem Schwellwert ermittelt werden. Auch in diesem Modul werden die notwendigen Algorithmen zur Bildbearbeitung aus der OpenCV Bibliothek genutzt.



**Abbildung 3.4:** Ansicht der LED-Matrix aus Sicht der Kamera

Ein weiteres Modul übernimmt die Übersetzung der Bitinformationen in das Zeitstempelformat. Da ein Großteil der LEDs direkt das Bitmuster des Zeitstempels anzeigen, genügt es, einige Bitschiebeoperationen auszuführen. Darüber hinaus wird ein Teil des Zeitstempels mit dem bereits erwähnten Verfahren des sich bewegenden Punkts übertragen. Durch dessen Erkennung läßt sich die Auflösung des Zeitstempels weiter erhöhen, sodass an dieser Stelle ein Algorithmus eine Rücktransformation ausführt und die gewonnenen Informationen zu dem Zeitstempel hinzufügt.

Das letzte Modul bildet eine Oberfläche über die sich die gesamte Anwendung steuern und überwachen läßt. Da OpenCV auch Methoden zur Anzeige von Bilddaten und grundlegende Funktionen zur Erstellung einer Oberfläche enthält, wird darauf zurückgegriffen. Die Oberfläche besteht aus einem einfachen Fenster, das permanent das aktuelle Kamerabild anzeigt. In das Bild eingezeichnet wird die momentan konfigurierte LED-Matrix-Position, der erkannte Zustand aller LEDs sowie der daraus abgeleitete Zeitstempel (Abb. 3.5). Desweiteren wird ein Menü in das Fenster eingeblendet, über das diverse Parameter eingestellt werden können.



**Abbildung 3.5:** Overlay, das wichtige Zustandsdaten in das aktuelle Bild einblendet.

## 4 Abschlussbetrachtung

Die Abschlussbetrachtung dieses Projekts erfolgt ähnlich dem Aufbau dieser Dokumentation getrennt in den Bereichen Konzept, Hardware und Software. Grundsätzlich zeigt der funktionsfähige Prototyp, dass die in diesem Projekt gestellte Aufgabe lösbar ist. Dabei ist insbesondere der finanzielle Aufwand durch geschickte Nutzung vorhandener Mittel gering geblieben. Die zeitliche Auflösung des realisierten Konzepts genügt dabei den gestellten Anforderungen, wie erste Versuche gezeigt haben.

### 4.1 Konzept

In der Aufgabenstellung des Projekts war kein konkretes Konzept vorgegeben, dieses sollte im Rahmen dieses Projekts erarbeitet werden. Das ausgewählte Konzept hat im Praxistest gezeigt, dass es möglich ist einen Zeitstempel mit der geforderten Auflösung optisch zu erzeugen und diesen mit einer Kamera sicher zu erkennen. Der in der Konzeptionierungsphase genannte Vorteil der Unabhängigkeit des Anzeigemoduls von der Kamera ist gegeben, wird jedoch durch den nicht trivialen Abgleich beim Wechsel der Komponenten leicht abgeschwächt. Ein weiterer Vorteil der das Konzept auszeichnet, ist die Fähigkeit des Anzeigemoduls, beliebige Zeitserver als Synchronisierungsquellen zu nutzen. Dies wurde in mehreren Tests bereits getestet und fehlerfrei genutzt.

### 4.2 Hardware

Die Entwicklung und der Aufbau der Hardware verlief weitestgehend problemlos. Durch den Verzicht auf SMD Bauteile sind die Lötarbeiten auch ohne Lötterfahrung durchzuführen. Aufgrund der geringen Bauteilanzahl ergibt sich durch die Größe der Bauteile kein Nachteil gegenüber SMD Bauteilen. Gleichzeitig ermöglicht die Nutzung von bedrahteten Bauteilen den Aufbau der Module auf Lochrasterplatinen, sodass keine Platine gefertigt werden muss. Die Nutzung eines Steckernetzteils sorgt dafür, dass keine hohen und damit gefährlichen Spannungen auf dem Modul vorhanden sind. Die Trennung von Anzeige und Basismodul bringt, wie erwartet, Vorteile bei der mechanischen Anordnung des Anzeigemoduls. Da als Verbindungselement ein Flachbandkabel vorgesehen ist, bestehen dennoch Einschränkungen bei dem Aufbau. Aufgrund einer ungünstigen Verschaltung der LEDs in den LED-Matrix Modulen ist es nicht möglich, die maximale Anzahl von LEDs anzusteuern. Das Basismodul

verfügt über 8 Zeilen und 8 Spaltenleitungen, sodass im Idealfall 64 LEDs angesteuert werden können. Obwohl in den Bausteinen insgesamt 70 LEDs vorhanden sind, können davon nur 56 angesteuert werden. Da dies jedoch über der errechneten Minimalanzahl von 42 LEDs liegt, ergibt sich daraus kein wesentlicher Nachteil für das Projekt.

### 4.3 Software

Im momentanen Zustand ist die Software grundsätzlich in der Lage die Aufgabe der Zeiterfassung und Auswertung zu erfüllen. Nach der Modifikation einiger Teile der OpenCV Bibliotheksfunktionen können Bilder von der Kamera in verschiedenen Formaten ausgelesen werden und alle erforderlichen Parameter über eine generische Schnittstelle angepasst werden. Die Kamerabilder bei bekannter Position der LED-Matrix und konstanten Lichtverhältnissen korrekt ausgewertet. Anschließend werden die Bilder sowohl auf dem Bildschirm angezeigt als auch abgespeichert. Mithilfe eines über dem angezeigten Kamerabild eingeblendeten Menü können Parameter der Kamera und der Software angezeigt und verändert werden. Darüber hinaus erlaubt das Menü verschiedene Betriebsarten, beispielsweise bei der Erkennung der LED-Matrix Position auszuwählen. Dennoch gibt es, überwiegend im Bereich der Software, Optimierungspotential. Da die Darstellung einer Bedienoberfläche nicht zu den primären Aufgaben der OpenCV Bibliothek gehört, ist für die weitere Entwicklung eine externe Lösung wie beispielsweise Qt vorzuziehen. Dadurch wird der Aufwand verringert, während gleichzeitig die Möglichkeiten der Anzeige steigen. Desweiteren sorgt eine ungeschickte Implementierung einiger Algorithmen die für jede Bildauswertung benötigt werden dafür, dass die optimale Bildwiederholrate nicht erreicht wird. Darüber hinaus birgt das Projekt noch diverse Möglichkeiten für Erweiterungen. Unter anderem kann eine direkte Exportfunktion für die Kamera- und Zeitdaten geschaffen werden, sowie eine dynamischere Anpassung auf sich verändernde Lichtverhältnisse implementiert werden.



# 5 Anhang

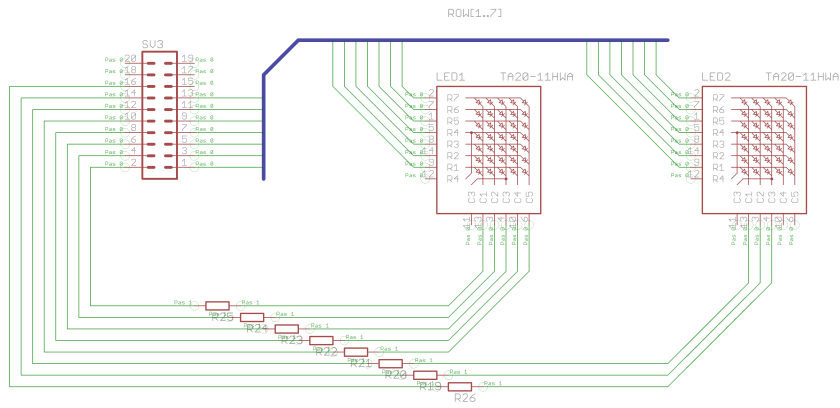


Abbildung 5.1: Schaltplan: Anzeigemodul

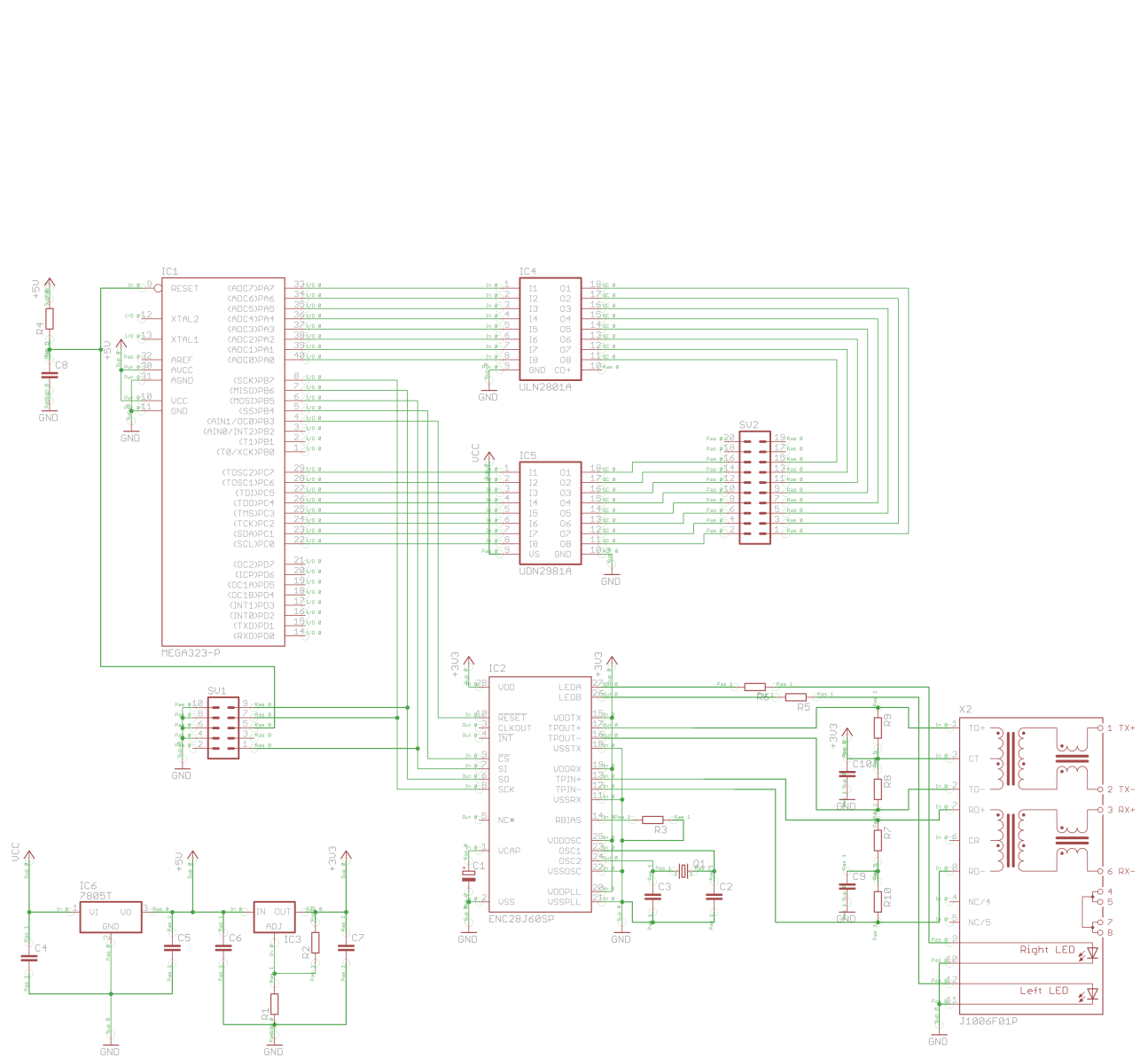


Abbildung 5.2: Schaltplan: Basismodul

# Literaturverzeichnis

- [Atm11] ATMEL, San Jose, USA: *ATmega32/L Datasheet*, Februar 2011. <http://www.atmel.com/atmel/acrobat/doc2503.pdf>.
- [Kin11] KINGBRIGHT: *Corporation Website*, 2011. <http://www.kingbright.com/>.
- [Mic04] MICROCHIP, Chandler, USA: *ENC28J60 Datasheet*, 2004. <http://ww1.microchip.com/downloads/en/devicedoc/39662a.pdf>.
- [Ope11] OPENCV COMMUNITY (BSD LICENSE): *Open Source Computer Vision library*, August 2011. <http://sourceforge.net/projects/opencvlibrary/>.
- [Poi09] POINTGREY, Richmond, Canada: *Flea2 Datasheet*, Februar 2009. <http://www.ptgrey.com/products/flea2/flea2.pdf>.
- [Poi11] POINTGREY: *Corporation Website*, 2011. <http://www.ptgrey.com>.

