

# Aufbau eines Versuchsfahrzeugs zur AUTOSAR- konformen Softwareentwicklung

## **Hauptseminar**

18. Februar 2011

vorgelegt von

Andreas Loth, 43358

Maximilian Hamm, 43815

## **Betreuer**

Dipl.-Inf. Christoph Bodenstein

## **TU-Ilmenau**

Fakultät für Informatik und Automatisierung

Fachgebiet System- und Softwareengineering

## **Abstract**

In der realen Welt sind Automobile komplexe verteilte Systeme und erfüllen wichtige Aufgaben.

In dieser Arbeit wurden die ersten Schritte zu einem voll funktionsfähigen Modellfahrzeug unternommen, welches als verteiltes System realisiert wird.

Im ersten Teil werden der Aufbau des Modellfahrzeugs und die dabei entstandenen Probleme und Lösungen beschrieben. Im zweiten Teil wird näher auf die Nutzung der Evaluation Boards eingegangen und wie diese für die Programmierung vorbereitet werden. Im dritten Teil wird die Elektrik sowie die Verkabelung der LEDs für die Beleuchtung erklärt.

Abschließend wird ein Ausblick gegeben, was alles schon vorbereitet wurde und woran noch gearbeitet werden muss.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
<b>2</b>	<b>Aufbau</b>	<b>5</b>
2.1	Grundmodell . . . . .	5
2.2	Farbgebung . . . . .	5
2.3	Komponenten . . . . .	6
2.4	Probleme beim Aufbau . . . . .	7
<b>3</b>	<b>Installation</b>	<b>9</b>
3.1	Windows XP . . . . .	9
3.2	Windows 7 . . . . .	9
3.3	Programmiertools . . . . .	9
<b>4</b>	<b>Konfiguration</b>	<b>10</b>
4.1	Boards . . . . .	10
4.1.1	Energieversorgung . . . . .	10
4.1.2	Starteinstellung . . . . .	10
4.2	Keil $\mu$ Vision . . . . .	10
<b>5</b>	<b>Programmierung</b>	<b>14</b>
5.1	Parallele Ports . . . . .	14
5.2	Wahl der Funktion einer Pins: Pn_IOCRRxx . . . . .	14
5.3	Einlesen der Eingabewertes eines Ports: Port Input Register Pn_IN . . . . .	14
5.4	Steuerung des Ausgabebetreibers: Pn_POCON . . . . .	14
5.5	Ausgabe auf einen Port: Port Output Register Pn_OUT . . . . .	15
5.6	Ändern des Port Output Registers: Port Output Modification Register . . . . .	15
5.7	Probleme beim Flashen der Boards . . . . .	15
<b>6</b>	<b>Vernetzung der Boards</b>	<b>19</b>
<b>7</b>	<b>Elektrik</b>	<b>20</b>
7.1	Energieversorgung . . . . .	20
7.2	Ein-/Ausschalter für die Boards . . . . .	20
7.3	Widerstandsleiste . . . . .	20
7.4	Stromverteiler . . . . .	21
7.5	Spannungsteiler . . . . .	22
7.6	Verkabelung der LEDs . . . . .	23
<b>8</b>	<b>Ausblick</b>	<b>25</b>
<b>9</b>	<b>Glossar</b>	<b>26</b>

**10 Quellen**

**27**

**11 Referenzen**

**28**

# 1 Einleitung

In der realen Welt sind Automobile komplexe verteilte Systeme und erfüllen wichtige, teilweise lebensrettende Aufgaben, wie zum Beispiel ABS und ESP.

Auch für diese Systeme muss Software entwickelt werden. An diesem Punkt setzt diese Hauptseminararbeit an. Das Modellfahrzeug ist als verteiltes System konzipiert, um die Softwareentwicklung für automobiler eingebettete Systeme lehren und nachvollziehen zu können. Dies könnte zwar auch virtuell durch Computersimulation geschehen, aber das Arbeiten mit einem physischen Modell und das direkte Erleben der Wirkungen der Software bringt doch einen ganz anderen Ansporn und besseres Feedback als eine reine Computersimulation.

## 2 Aufbau

### 2.1 Grundmodell

Das Grundmodell des Fahrzeugs vom Typ Mercedes Benz Actros, bestehend aus Polymer und Metall, wurde als Bausatz geliefert. Die Einzelteile wurde mithilfe von Cuttermesser und Seitenschneider aus ihren Halterungen gelöst und verschraubt, in einzelnen Fällen auch verklebt.

Das Modellfahrzeug besteht im Wesentlichen aus drei Teilkomponenten: dem Fahrgestell, dem Führerhaus und dem Ladungsaufbau.

Ein großer Teil des Fahrzeugs ist durch die Verschraubung und das Verbauen von Scharnieren abnehmbar, beziehungsweise kann weggeklappt werden. Somit ist es möglich auf einfache Weise an die verbauten Komponenten zu kommen.

### 2.2 Farbgebung

Das Fahrgestell des Fahrzeugs bestand bereits aus schwarzem Polymer und musste somit keiner weiteren Farbgebung unterzogen werden.

Der Führerhausaufbau sowie der Ladungsaufbau wurden in weißen Polymer- und Metalleinzelteilen geliefert. Dies machte es erforderlich selbige Teile einer neuen Farbgebung zu unterziehen.

Hierfür wurden die Farben Seegrün und Mattschwarz verwendet (Bild 2.1). Zur Durchführung wurde ein Airbrush Toolkit verwendet.

Es wurden jeweils zwei Schichten Farbe aufgetragen, wobei die Einzelteile mindestens 24 Stunden zwischendurch trocknen konnten.



Bild 2.1: Fertiges Modellfahrzeug mit Lackierung



Bild 2.2: Erste Lackierung der Einzelteile in Seegrün

## 2.3 Komponenten

In dem Modellfahrzeug finden viele Komponenten, welche unterschiedlichen Zwecken dienen, Platz.

Im Fahrzeuggestell sind zwei Servos, jeweils für die Lenkung und das Schaltgetriebe sowie ein bürstenloser Motor mit Steuergerät verbaut. Des weiteren befindet sich am Fahrzeuggestell auf der Unterseite die Haltevorrichtung für den benötigten Akkumulator.

Weiterhin befinden sich im Ladeaufbau fünf Infineon Evaluation Boards XC2287 sowie eine Widerstandsleiste und ein Stromverteiler.

Hinzu kommen noch 16 LEDs für die Beleuchtung des Modellfahrzeugs.

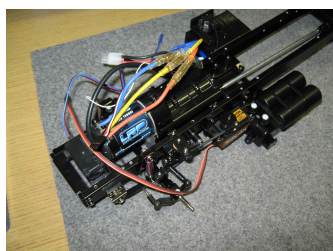


Bild 2.3: Fahrzeuggestell mit Motor und Servos

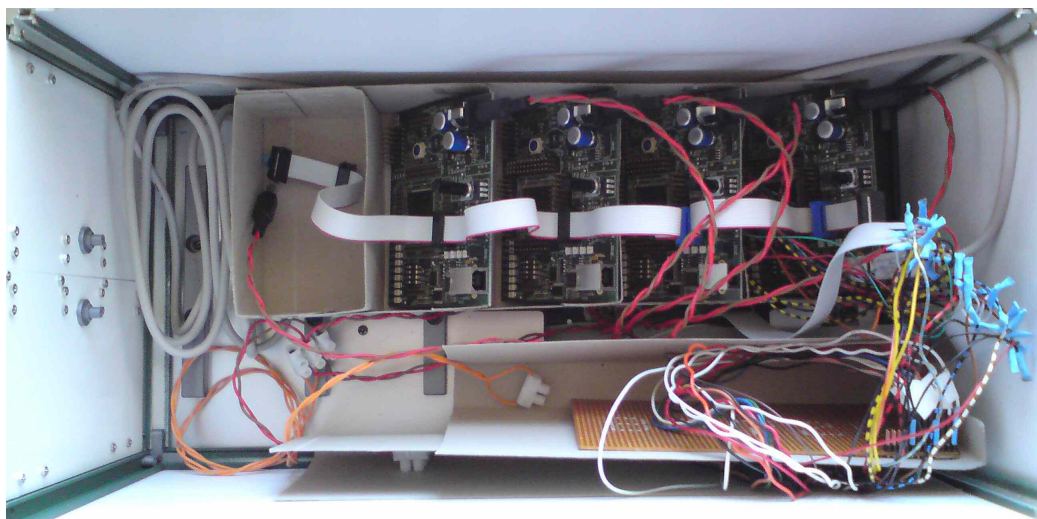


Bild 2.4: Fahrzeugladeaufbau mit Evaluation Boards, Stromverteiler und Widerstandsleiste

## 2.4 Probleme beim Aufbau

Bei dem Aufbauen des Modellfahrzeugs sind mehrer Probleme aufgetreten, welche den Aufbau teilweise verzögert haben.

Eines der ersten Hindernisse waren die noch nicht gelieferten Lenk- und Schaltservos, wodurch mit dem Aufbau des Fahrzeuggestells erst mit zwei Wochen Verzögerung begonnen werden konnte.

Kurz darauf folgte die Erkenntnis, dass einigen Einzelteilen noch eine Farbgebung fehlt. Es konnten zwar rechtzeitig Farben und Airbrush Toolkit beschafft werden, jedoch fiel erst nach Beginn der Lackierungsarbeiten auf, dass die benötigte Farbe nicht ausreichte. Daraufhin verzögerte sich das Lackieren und somit der weitere Aufbau um weitere

zwei Wochen.

Nach der weiteren Feststellung, dass nun auch die Druckluft, welche für das Airbrush Toolkit benötigt wird, nicht reichen sollte, einigte man sich darauf, vorerst einige Teile vorerst nicht zu lackieren. Hiervon betroffen sind der gesamte Inneneinbau des Führerhauses sowie Teile der vorderen Stoßstange (Bild 2.4).



Bild 2.4: Einige nicht lackierte Teile sind sichtbar



## 3 Installation

Die Infineon Evaluation Boards können mit einem Computer über eine USB-Verbindung programmiert werden. Dazu werden passende Treiber und Programmierertools benötigt. Diese sind auf der den Boards beiliegenden CD enthalten.

Im Rahmen dieser Arbeit wurden bei der Programmierung der Boards die Betriebssysteme Windows XP SP3 32 Bit und Windows 7 SP0 64 Bit getestet.

Wenn kein Treiber für den COM-Port gefunden wird, können Trotzdem *µVision* und der *DAS Server* bzw. der *DAS Basic Client* ohne Probleme verwendet werden.

### 3.1 Windows XP

Um mit Windows XP die Boards benutzen zu können, muss zuerst der Device Access Server (DAS) installiert werden. Dieser befindet sich auf der CD im folgenden Verzeichnis: *CD:\Tools\DAS\DAS\_v262* .

Danach kann das Board per USB mit dem Computer verbunden werden. Windows installiert nun die benötigten Treiber.

### 3.2 Windows 7

Wenn Windows 7 so konfiguriert ist, dass es Treiber bei Windows Update sucht, lädt es sich bei bestehender Internetverbindung die benötigten Treiber inklusive DAS herunter.

Das Board kann also gleich mit dem Computer verbunden werden, das Laden und Installieren der nötigen Treiber übernimmt Windows.

### 3.3 Programmierertools

Zur Programmierung wurde in dieser Arbeit die IDE Keil *µVision* 3 benutzt.

Sie befindet sich auf der dem Board beiliegenden CD im Verzeichnis *CD:\Tools\Compiler\Keil* .

*µVision* 3 konnte unter Windows XP und Windows 7 ohne Probleme eingesetzt werden. Allerdings muss darauf geachtet werden, dass der Benutzer, unter dem *µVision* ausgeführt wird, Schreibrechte im Verzeichnis des aktuellen Projekts hat.

Weiterhin wird der *DAS Basic Client* verwendet, um das Board zurückzusetzen und zu stoppen, wenn es sich nicht programmieren lässt.

## 4 Konfiguration

### 4.1 Boards

#### 4.1.1 Energieversorgung

Die Boards können entweder über die USB-Schnittstelle mit Energie versorgt werden oder über eine externe Energieversorgung (Buchse für Rundstecker auf dem Board).

Die Wahl der Energieversorgung erfolgt auf den Boards über Jumper X502, der sich in der Nähe der USB-Buchse befindet.

Verbindet der Jumper die Pins 1 und 2, so wird das Board über USB mit Energie versorgt. Sind Pin 2 und 3 verbunden, so erfolgt die Energieversorgung über die externe Stromversorgung.

Siehe dazu auch Tabelle 3 auf Seite 11 in [2].

Im Rahmen dieser Arbeit wurden die Boards auf externe Energieversorgung eingestellt, um die Versorgung über den Akkumulator des Modellautos zu ermöglichen.

Die externe Energieversorgung muss eine Gleichspannung von 6 V bis 12 V liefern und eine Stromstärke von maximal 400 mA. Die Polarität der Rundstecker ist wie folgt festgelegt:

- außen: Minus
- innen: Plus

Vgl. Bild 5 auf Seite 12 in [2].

Bei Verpolung arbeiten die Boards nicht, scheinen aber auch nicht beschädigt zu werden.

#### 4.1.2 Starteinstellung

Den Startcode können die Evaluation Boards von verschiedenen Quellen laden. Die Quelle wird über die DIP-Schalter S102 konfiguriert.

Alle vier Schalter auf *OFF* bedeuten einen standard Start vom internen Flashspeicher. In diesem Startmodus können die Boards auch mit  $\mu$ Vision programmiert werden.

Im Rahmen dieser Arbeit wird nur dieser Modus verwendet.

Alle Verfügbaren Modi und die entsprechenden Schalterstellungen sind in Tabelle 4 auf Seite 13 in [2] zu finden.

## 4.2 Keil $\mu$ Vision

Damit die Evaluation Boards mit  $\mu$ Vision programmiert werden können, muss in  $\mu$ Vision eingestellt werden, welches Gerät programmiert werden soll und wie darauf zugegriffen werden kann.

Die Einstellungen werden im Projekt gespeichert. Sie müssen also für jedes Projekt neu eingestellt werden.

Folgende Einstellungen müssen getätigt werden:

- Als erstes wird in der Symbolleiste *Build* im Drop Down Menü *On-chip Flash: DAS* ausgewählt (Bild 4.1)
- In der Menüleiste wird dann *Flash - Configure Flash Tools...* ausgewählt. (Bild 4.2)
- Im sich daraufhin öffnenden Dialogfeld *Options for Target 'On-chip Flash: DAS'* wird in der Registerkarte *Utilities Use Target Driver for Flash Programming* und im Drop Down-Feld *Infineon DAS Client for XC16x* ausgewählt. (Bild 4.3)
- Danach wird auf *Settings* geklickt und im *Infineon DAS Driver Setup* als Host Computer *localhost* und als DAS Server *JTAG oder USB Chip* gewählt. Im Bereich *Programming Algorithm* wird als einziger Algorithmus *XC2xxx-96F On-chip Flash* hinzugefügt. Die Einstellungen in diesem Dialogfeld werden nun mit *OK* bestätigt. (Bild 4.4)
- Jetzt wird im Dialogfeld *Options for Target 'On-chip Flash: DAS'* zur Registerkarte *Device* gewechselt und in der Liste der Geräte *Infineon - XC2287-96F* ausgewählt. (Bild 4.5)
- Als letztes wird in der Registerkarte *Output Create HEX File* aktiviert. Das Dialogfeld *Options for Target 'On-chip Flash: DAS'* kann nun mit einem Klick auf *OK* geschlossen werden. (Bild 4.6)



Bild 4.1: Symbolleiste *Build*

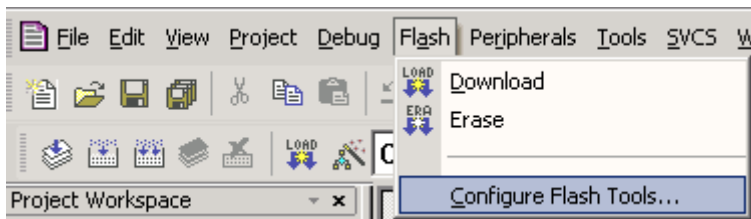


Bild 4.2: Menüleiste - *Flash*

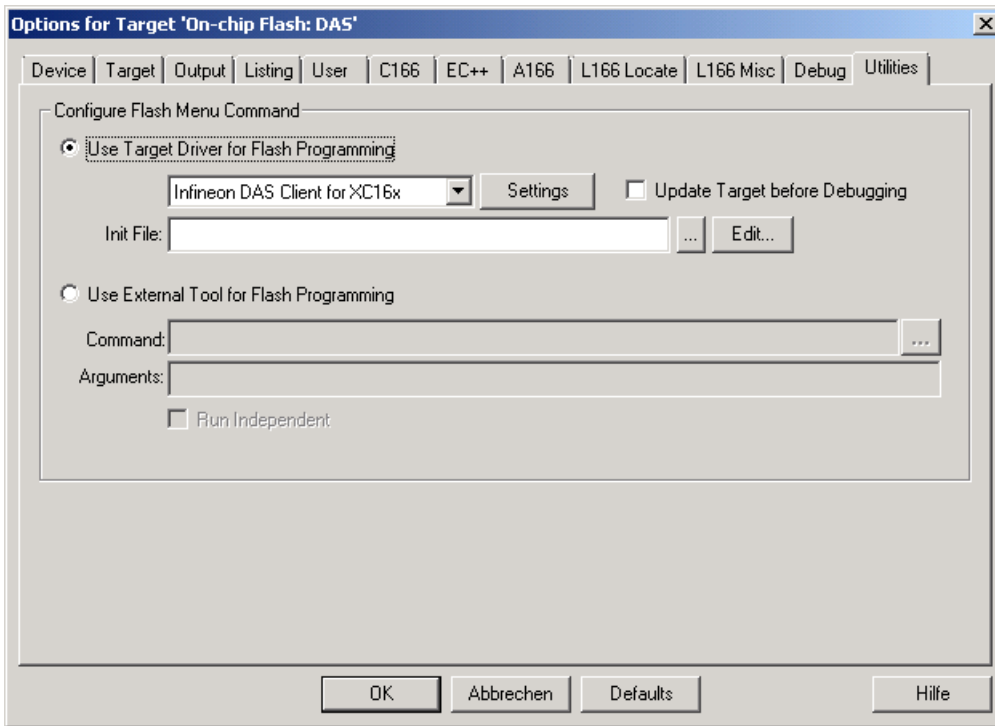


Bild 4.3: Options for Target 'On-chip Flash: DAS' - Utilities

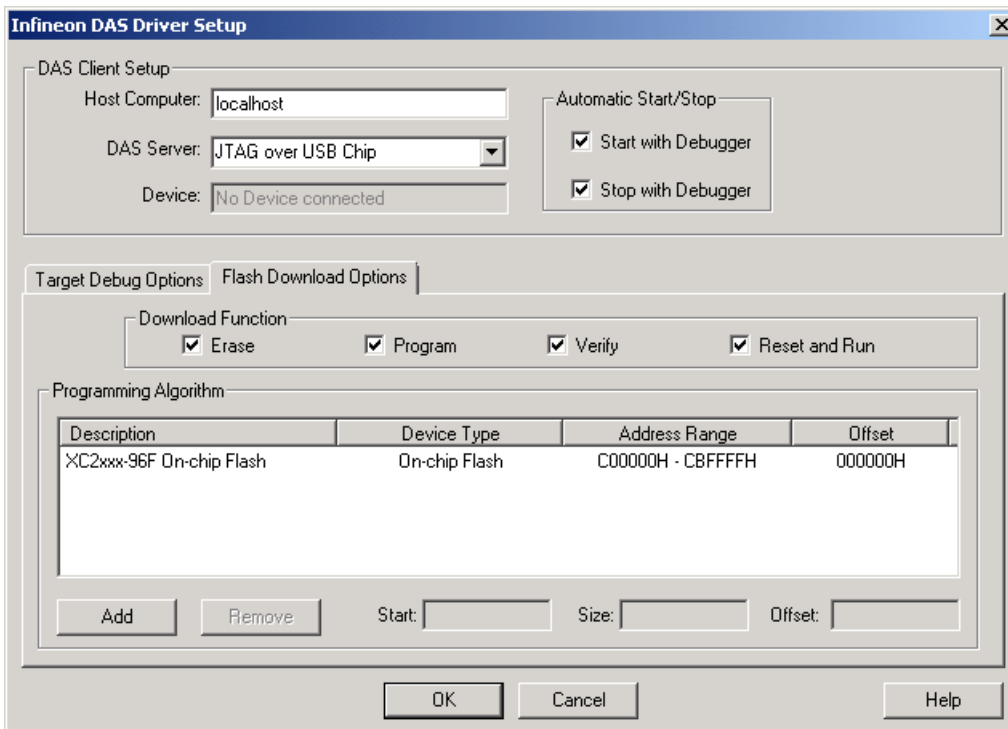


Bild 4.4: Infineon DAS Driver Setup

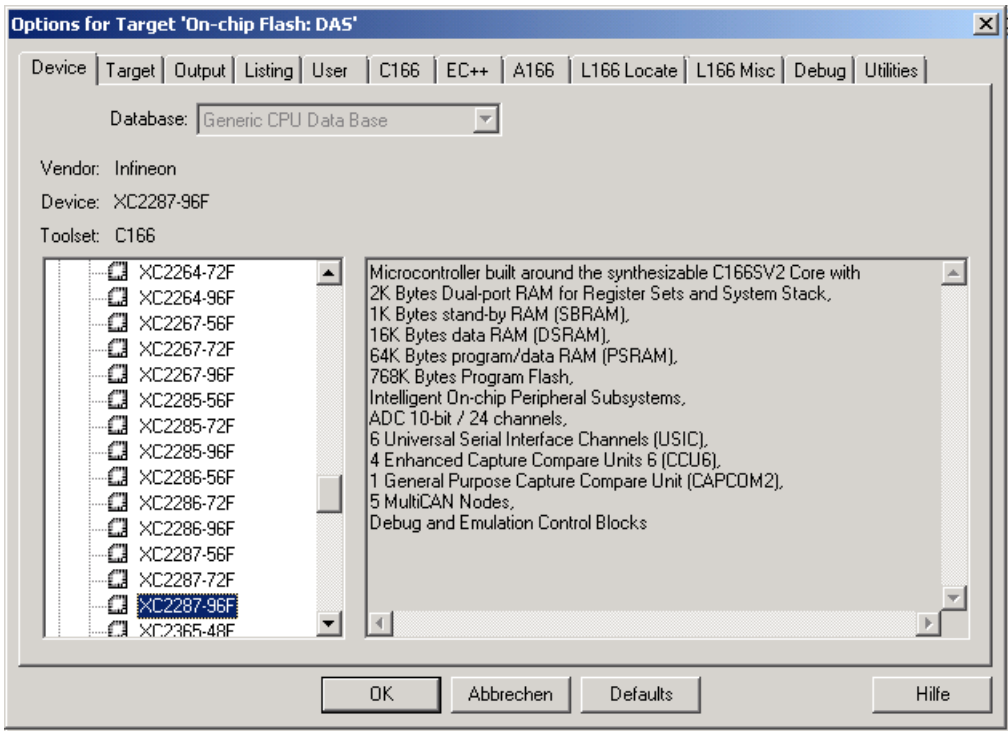


Bild 4.5: Options for Target 'On-chip Flash: DAS' - Device

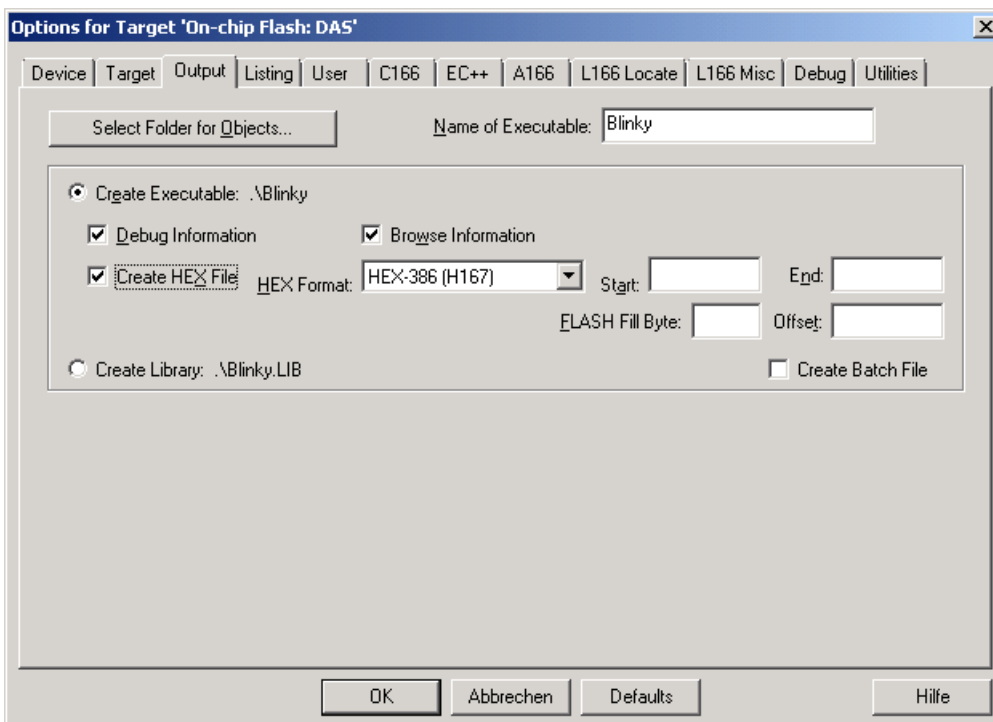


Bild 4.6: Options for Target 'On-chip Flash: DAS' - Output

## 5 Programmierung

Mit Hilfe von Keil  $\mu$ Vision können C-Programme für die Infineon Evaluation Boards geschrieben, kompiliert und auf die Boards geflasht werden. Die in dieser Arbeit erstellten Demonstrationsprogramme benutzen das mit  $\mu$ Vision mitgelieferte Beispielprogramm Blinky. Dieses befindet sich, wenn  $\mu$ Vision bei der Installation in das Standardverzeichnis `C:\Keil` installiert wurde, im Verzeichnis `C:\Keil\CI66\Boards\Infineon XC2287`.

Die include-Dateien und die Initialisierung des Boards wurden beibehalten. Die delay-Prozedur wurde angepasst. Die Initialisierung der Ports und die eigentliche Programmlogik wurden komplett neu geschrieben.

### 5.1 Parallele Ports

Die Benutzung der parallelen Ports ist in Kapitel 7 Parallel Ports in [1] beschrieben.

Bevor die Ports benutzt werden können, müssen sie initialisiert werden. Dabei wird bestimmt, welche Funktion sie übernehmen und welche Stärke die Ausgabetreiber haben sollen.

### 5.2 Wahl der Funktion einer Pins: `Pn_IOCRxx`

Das Register `Pn_IOCRxx` bestimmen die Funktion des Pins `xx` von Port `n`.

Der Wert `0x80` bedeutet dabei Mehrzweckausgabe im Push-Pull Modus. Mehrzweckausgabe bedeutet dabei, dass im Programm der Pin manuell auf 0 oder 1 gesetzt werden kann. Der Push-Pull-Modus wird verwendet, um den Pin als Stromquelle und -senke verwenden zu können.

Für die digitale Eingabe wurden die Konfigurationen `0x00`, `0x01` und `0x02` getestet. Wenn der Pin offen ist, also weder mit `VDDPB`, noch mit `Vss` verbunden ist, wird der Wert 0 gelesen. Mit `0x00` fließt kein messbarer Strom, wenn der Pin mit `VDDPB` oder mit `Vss` verbunden wird. Bei `0x01` und `0x02` fließt ein Strom von ungefähr  $150\ \mu\text{A}$ , wenn der Pin mit `Vss` bzw. `VDDPB` verbunden wird.

### 5.3 Einlesen der Eingabewertes eines Ports: Port Input Register `Pn_IN`

Über das Register `Pn_IN` wird der Wert eingelesen, der aktuell am Port `n` anliegt. Nicht vorhandene Pins werden dabei immer als 0 gelesen.

Das Register kann nach Anleitung auch benutzt werden, wenn der Port als Ausgabeport konfiguriert wurde. Dies wurde aber im Rahmen dieser Arbeit nicht getestet.

Um auf einzelne Pins zuzugreifen, kann das Register `Pn_IN_Px` benutzt werden. Dabei steht `n` für die Portnummer und `x` für die Nummer des Pins.

### 5.4 Steuerung des Ausgabebetreibers: `Pn_POCON`

Mit dem Register `Pn_POCON` können die Stärke des Ausgabebetreibers, die Steilheit der Flanken und das Verhalten der Pins im Energiesparmodus eingestellt werden.

Die Konfiguration erfolgt immer in Gruppen von 4 Pins.

Durch die Wahl der Stärke des Ausgabetreibers können Probleme durch elektromagnetische Störstrahlung minimiert werden.

Die Steilheit der Signalfanken beeinflusst die maximale Stromstärke, die durch den Treiber fließt, wenn der Ausgabewert geändert wird.

## 5.5 Ausgabe auf einen Port: Port Output Register Pn\_OUT

Mit dem Register *Pn\_OUT* wird der Ausgabewert des Ports *n* gesteuert, wenn die entsprechenden Pins als Mehrzweckausgabe konfiguriert wurden.

Auf einzelne Pins kann mit Hilfe des Registers *Pn\_OUT\_Px* zugegriffen werden. Dabei steht *n* für den Port und *x* für den Pin.

## 5.6 Ändern des Port Output Registers: Port Output Modification Register

Über die Register *Pn\_OMRH* und *Pn\_OMRL* können einzelne Bits des *Pn\_OUT*-Registers gesetzt, gelöscht und umgekehrt werden. *n* steht dabei für die Nummer des Ports.

Die selbe Wirkung kann jedoch auch durch direkte Zuweisung des gewünschten Wertes an das Register *Pn\_OUT* bzw. *Pn\_OUT\_Px* erzielt werden.

*Pn\_OMRH* und *Pn\_OMRL* modifizieren *Pn\_OUT*. Beim Lesen der Output Modification Register wird immer 0 zurückgegeben.

## 5.7 Probleme beim Flashen der Boards

Bei der Arbeit mit den Boards ist öfters der Fehler aufgetreten, dass beim Flashen kein Board gefunden wurde, obwohl es per USB mit dem Computer verbunden war und auch Spannung hatte. Die in Bild 5.1 dargestellte Fehlermeldung wurde dabei von  $\mu$ Vision angezeigt.

Der Fehler lässt sich beheben, indem mit dem *DAS Basic Client* das Board zurückgesetzt und angehalten wird.

Dazu wird zuerst der *DAS Basic Client* gestartet. Im *DAS Device Select*-Fenster ist *localhost* als *Host Computer* schon richtig eingetragen (Bild 5.2). Nun wird per Klick auf *Start DAS Server* und danach auf *Installed Servers* die Übersicht der installierten Server aufgerufen (Bild 5.3). Der Server *JTAG over USB Chip* wird gestartet. Die Fenster *Installed DAS Server* und *DAS Server Control Panel* können geschlossen werden. Im Fenster *DAS Device Select* wird nun das Gerät *XE166/XC2000-Family* ausgewählt (Bild 5.4). Als letztes wird jetzt das Board mittels Klick auf die Schaltfläche *Reset and Halt* im *DAS Basic Client* zurückgesetzt und angehalten (Bild 5.5).

Nach dem Zurücksetzen und Anhalten kann das Board mit  $\mu$ Vision geflasht werden.

Diese Vorgehensweise funktionierte auch als der *DAS Basic Client* das Board nicht richtig erkannt hatte.



Bild 5.1: Fehlermeldung von  $\mu$ Vision

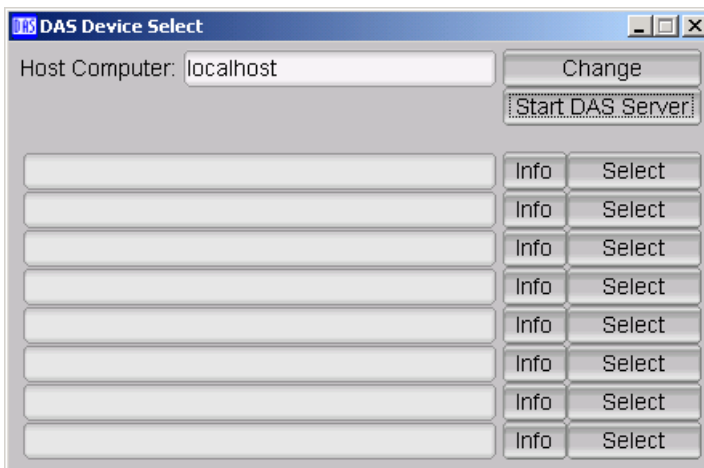


Bild 5.2: *DAS Device Select*



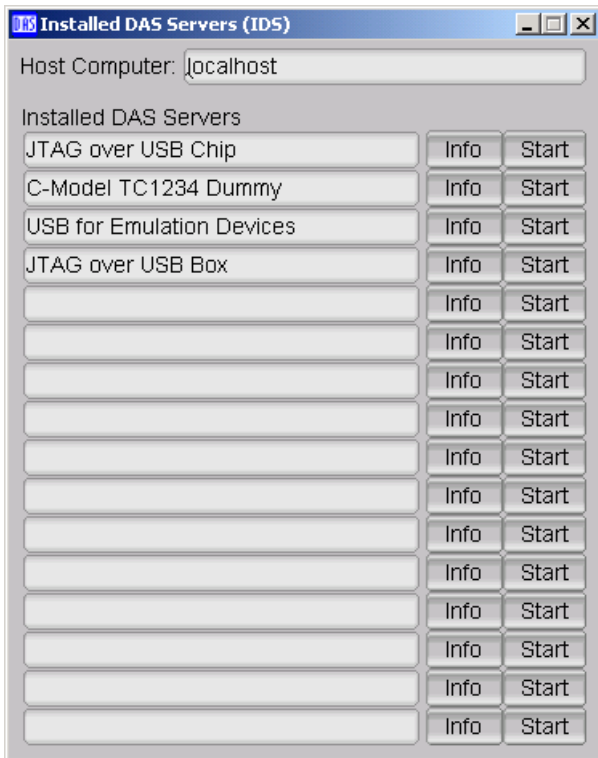


Bild 5.3: Installierte DAS Server

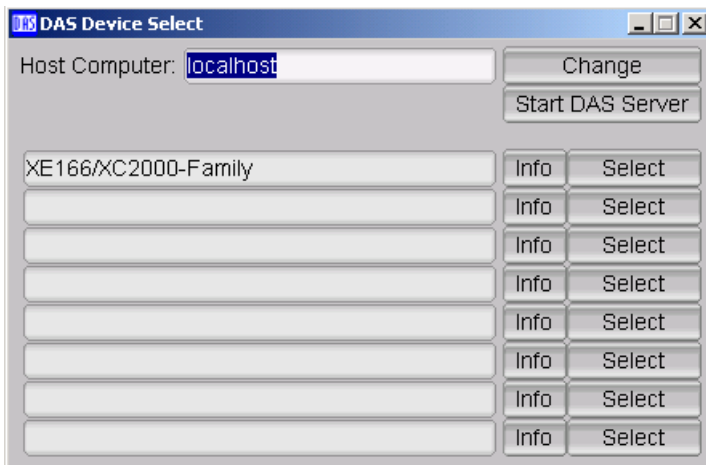


Bild 5.4: DAS Device Select, ein Gerät kann ausgewählt werden

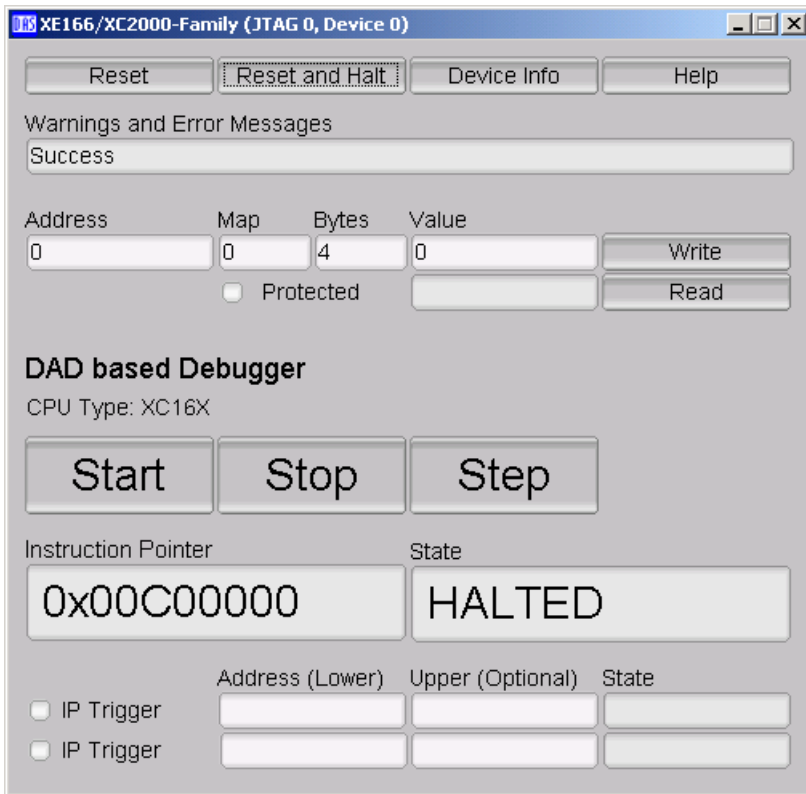


Bild 5.5: DAS Basic Client

## **6 Vernetzung der Boards**

Die Boards verfügen über je 2 CAN-Bus-Schnittstellen. Ein Kabel zur Verbindung der Boards wurde angefertigt und aufgesteckt. Terminiert wurde es einseitig mit 110 Ohm Widerständen.

## 7 Elektrik

### 7.1 Energieversorgung

Die Energie für die Boards wird direkt vom Akkumulator genommen. Dazu wurden an die Stromleitung vom Motorsteuergerät, die direkt mit dem Akkumulator verbunden wird, zwei Kabel angelötet. Das rote Kabel ist Plus, das braune Minus.

### 7.2 Ein-/Ausschalter für die Boards

Ursprünglich war ein separater Schalter zum Ein- und Ausschalten der Boards geplant. Da aber die Motorsteuerung immer etwas Strom verbraucht und daher der Stecker vom Akkumulator trotzdem immer herausgezogen werden muss, wurde der Schalter nicht realisiert.

Die Vorbereitung dafür ist allerdings, wie in Bild 7.1 zu sehen, immernoch vorhanden.

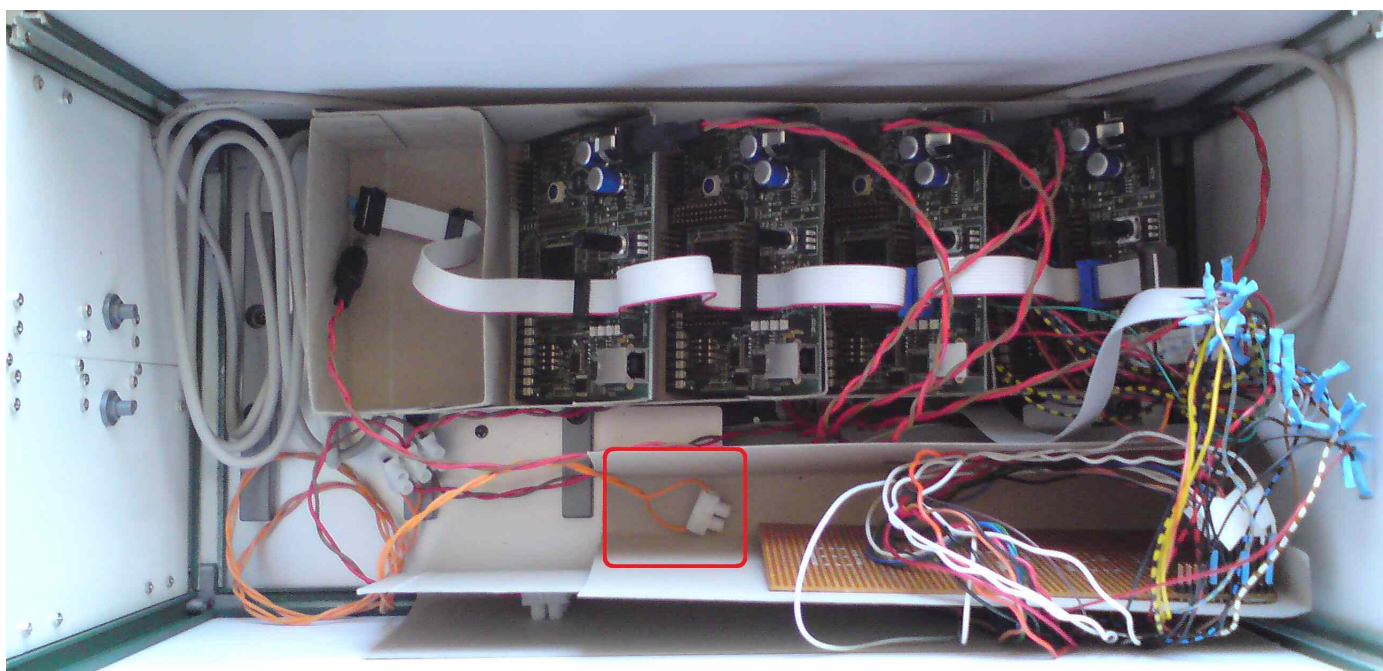


Bild 7.1: Laderaum, Vorbereitung für Schalter markiert

### 7.3 Widerstandsleiste

Zur Begrenzung der Spannung für die LEDs werden Vorwiderstände benötigt. Diese werden auf die im Bild 7.2 dargestellte Widerstandsleiste gesteckt. Dadurch ist es leicht möglich, die Widerstände auszutauschen.

Über die Wahl der Widerstände kann auch die Helligkeit der LEDs beeinflusst werden.

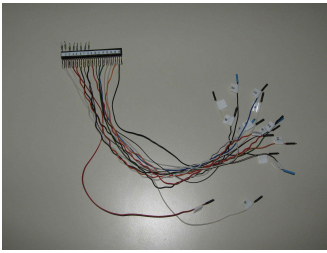


Bild 7.2: Widerstandsleiste

## 7.4 Stromverteiler

Die Boards verfügen nur über 8 Pins für VDDPB (Plus) und 4 für V<sub>ss</sub> (Masse). Damit die 16 LEDs alle mit Masse versorgt werden können, dient die in den Bildern 7.3 und 7.4 dargestellte Platine zur Stromverteilung.

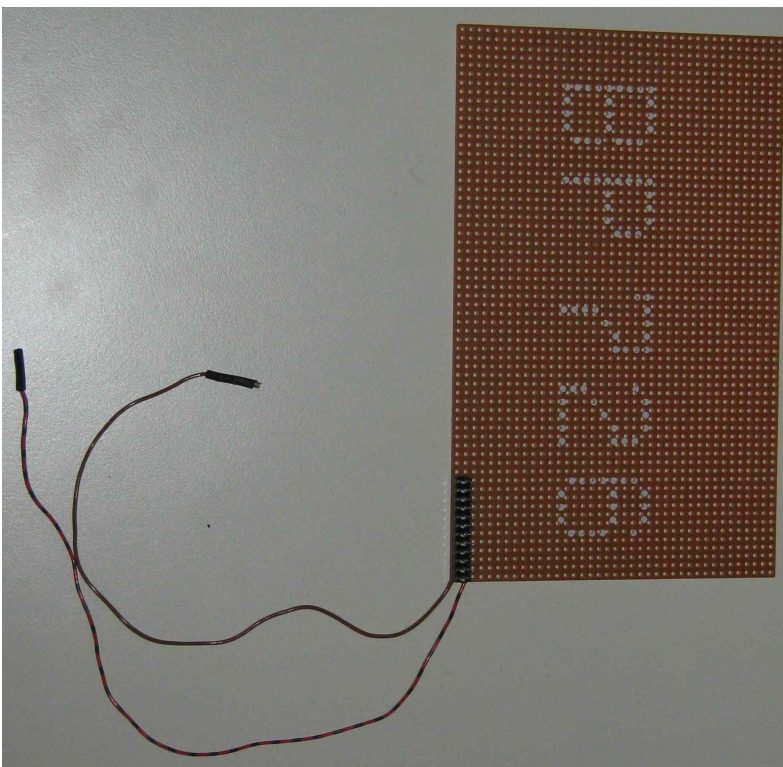


Bild 7.3: Stromverteiler, Vorderseite

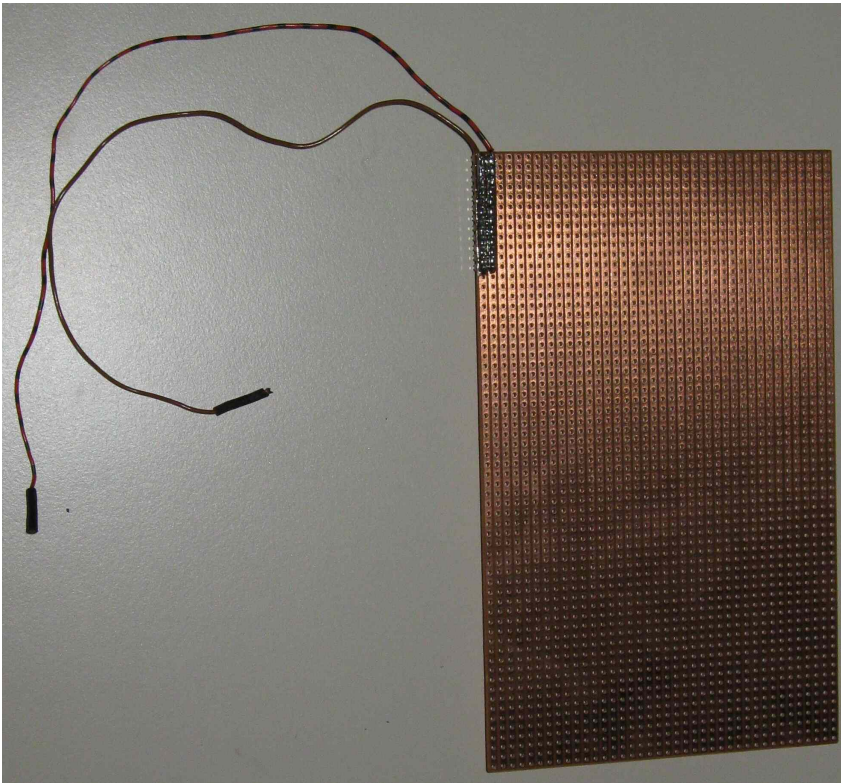


Bild 7.4: Stromverteiler, Rückseite

## 7.5 Spannungsteiler

Der Analog-Digital-Umsetzer der Boards erlaubt eine maximale Spannung von 5 V. Der Akkumulator hat aber eine Nennspannung von 7,2 V. Um trotzdem die Spannung messen zu können, wird sie über einen Spannungsteiler, in Bild 7.5 dargestellt, halbiert. Der Spannungsteiler besteht aus zwei in Reihe geschalteten 6,2 kOhm Widerständen. Dadurch ist der Spannungsabfall über einen Widerstand genau die Hälfte des Spannungsabfalls über beide Widerstände, also auch die Hälfte der Akkumulatorspannung. Dadurch können Spannungen bis zu 10 V gemessen werden.

Bei der Weiterverarbeitung muss dann allerdings darauf geachtet werden, dass die gemessene Spannung nur die Hälfte der Akkumulatorspannung ist.

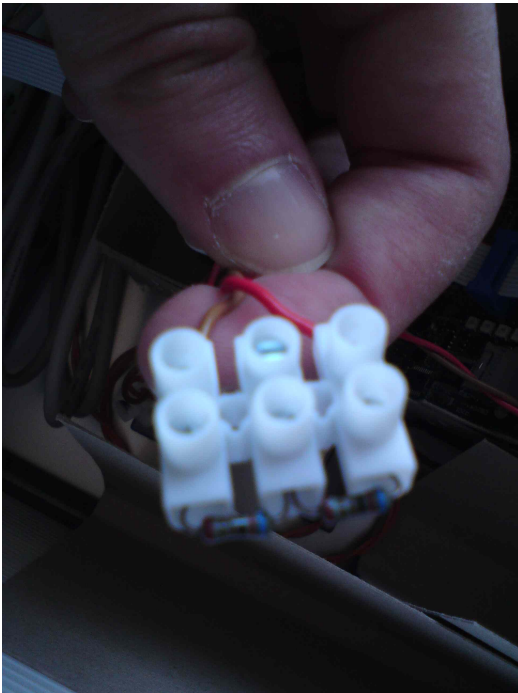


Bild 7.5: Spannungsteiler

## 7.6 Verkabelung der LEDs

In dem Modellfahrzeug wurden insgesamt 16 LEDs verbaut. Diese verteilen sich zu jeweils 8 LEDs an der hinteren Stoßstange und an der vorderen Stoßstange. Die Anoden der LEDs wurden über insgesamt 2 Flachbandkabel (jeweils vorne und hinten eins) direkt an die Pins der Evaluation Boards gesteckt.

Die Kathoden jedoch werden zuerst an die Widerstandsleiste gesteckt. Von dort ausgehend werden die Kathoden auf den Stromverteiler geleitet. Dieser bezieht vom Evaluation Board die Masse und verteilt diese an die einzelnen Kathoden über die Widerstandsleiste (Bild 7.6).

Die LEDs werden über die Pins der parallelen Ports der Evaluation Boards geschaltet (Anode), wodurch ermöglicht wird, dass jede LED einzeln angesteuert werden kann. Die LEDs teilen sich in folgende Farbgruppen auf:

- 5 Rote
- 7 Weiße
- 4 Orangene

wobei jeweils 2 weiße LEDs Abblendlicht, Standlicht und Fernlicht, jeweils 2 rote LEDs Schlussleuchte und Bremslicht, alle 4 orangenen LEDs Blinker und die beiden verbleibenden rote und weiße LED Nebelschlussleuchte und Rückfahrlicht bilden.

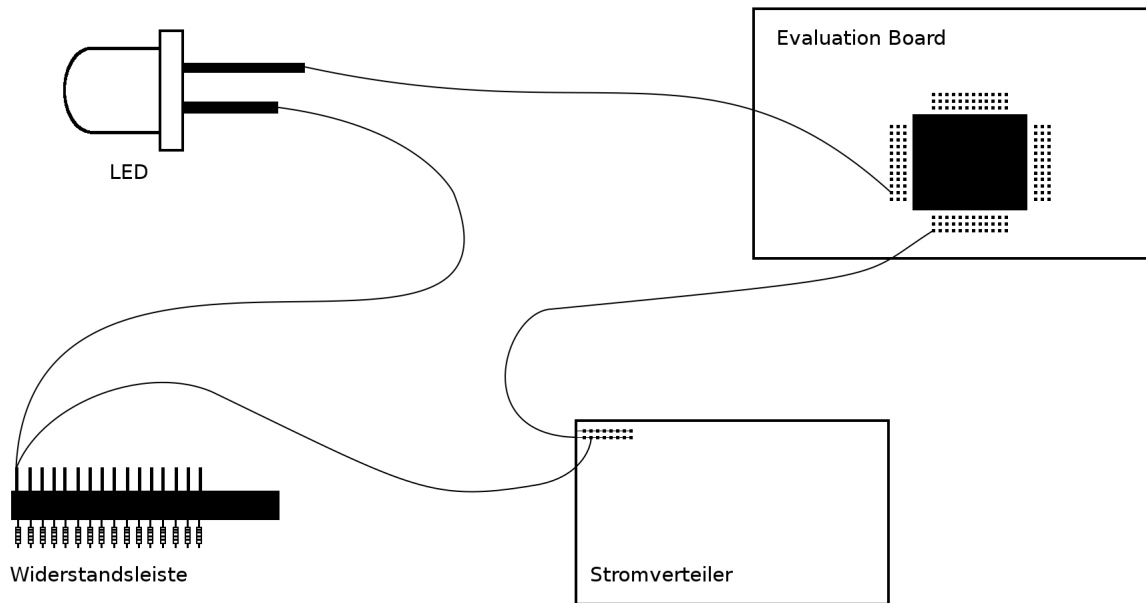


Bild 7.6: LED-Verkabelung

Die LEDs können über folgende Pins gesteuert werden:

Pin	LED-Funktion	Abkürzung
P3.0	Fernlicht Fahrer	FL.F
P3.1	Fernlicht Beifahre	FL.B
P3.2	Abblendlicht Fahre	ABBL.F
P3.3	Abblendlicht Beifahrer	ABBL.B
P3.4	Standlicht Fahrer	STL.F
P3.5	Standlicht Beifahrer	STL.B
P3.6	Blinker Fahrer vorne	BL.F.V
P3.7	Blinker Fahrer hinten	BL.F.H
P4.0	Blinker Beifahrer vorne	BL.B.V
P4.1	Blinker Beifahrer hinten	BL.B.H
P4.2	Schlussleuchte Fahrer	SCHL.F
P4.3	Schlussleuchte Beifahrer	SCHL.B
P4.4	Bremslicht Fahrer	BRL.F
P4.5	Bremslicht Beifahrer	BRL.B
P4.6	Nebelschlussleuchte	NSLN
P4.7	Rückwärtsfahrtlicht	RL



## 8 Ausblick

Das Modellfahrzeug ist nahezu komplett zusammengebaut. Die Boards werden vom Fahrzeug mit Energie versorgt und auf einem Board ist ein Demonstrationsprogramm installiert, welches die eingebauten Lichter blinken lässt.

Dies ist der grundlegende Schritt. Dennoch müssen noch einige Arbeiten erledigt werden:

Die Lackierung ist noch unvollständig, die Servos und der Motor noch nicht angeschlossen. Weiterhin ist noch keine Sensorik vorhanden und die Boards sind zwar untereinander über CAN-Bus verkabelt, „sprechen“ aber noch nicht miteinander. Last but not least ist die Software noch nicht AUTOSAR-konform.

Wir sind zuversichtlich, dass dieses Modellfahrzeug gute Dienste in der Lehre und als Demonstrationsobjekt leisten wird, sobald diese Arbeiten abgeschlossen sind.

## 9 Glossar

Treiber	Signalverstärker
Push-Pull	Im Push-Pull-Betrieb wird der Ausgang mittels zweier Transistoren wahlweise mit Puls oder Masse verbunden. Dadurch kann der Ausgang als Stromquelle und -senke verwendet werden und erreicht nahezu das Spannungsniveau der Versorgungsspannung.
$V_{ss}$	In der Anleitung der Evaluation Boards verwendet für Masse.

## 10 Quellen

Peripheral User Manual XC2000

<http://de.wikipedia.org/wiki/Spannungsbezeichnung> (Stand: 09.02.2011)

<http://www.info-magazin.com/index.php?suchbegriff=Treiber> (Stand: 08.02.2011)

[http://www.mikrocontroller.net/articles/Ausgangsstufen\\_Logik-ICs](http://www.mikrocontroller.net/articles/Ausgangsstufen_Logik-ICs) (Stand: 04.02.2011)

# 11 Referenzen

## Literatur

- [1] System User Manual XC2000
- [2] XC2000 Evaluation Board Manual