- MA728 -

**Master Thesis Electrical Engineering**

# Chameleon

# Compact Multi Interface Probe Station for High Security µControllers

Institute for Electronics
Graz University of Technology
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Wolfgang Bösch
in
Cooperation with
Infineon Technologies Austria
Design Center: Graz

submitted by
Ing. Georg Skacel, BSc.; 435 / 0631067

Assessor: Ass.Prof. Dipl.-Ing. Dr.techn. Peter Söser
Advisor: Dipl.-Ing. Mihai Tudosi

February 4, 2013

# Acronyms

| | |
|---|---|
| **AES** | Advanced Encryption Standard |
| **BJT** | Bipolar Junction Transistor |
| **CB** | Contactbased |
| **CL** | Contactless |
| **CS** | Chip Select |
| **DAC** | Digital-to-Analog-Converter |
| **DCLB** | Direct Contactless Bridge |
| **DES** | Data Encryption Standard |
| **3DES** | Triple DES |
| **DTR** | Data Terminal Ready |
| **DUT** | Device Under Test |
| **ECC** | Elliptic Curve Cryptography |
| **EEPROM** | Electrically Erasable Programmable Read-Only Memory |
| **FET** | Field-Effect Transistor |
| **GPIB** | General Purpose Interface Bus |
| **GPIO** | General Purpose Input Output |
| **GUI** | Graphical User Interface |
| **I2C** | Inter-Integrated Circuit |
| **IC** | Integrated Circuit |
| **IDE** | Integrated Development Environment |
| **IO** | Input-Output |
| **ISA** | Industry Standard Architecture |
| **LDO** | Low Drop Out |
| **LED** | Light Emitting Diode |
| **LPC** | Low Pin Count |
| **MISO** | Master In Slave Out |
| **MOSFET** | Metal–Oxide–Semiconductor Field-Effect Transistor |
| **MOSI** | Master Out Slave In |
| **NFC** | Near Field Communication |
| **OPAMP** | Operational Amplifier |
| **OTA** | Operational Transconductance Amplifier |
| **PCB** | Printed Circuit Board |
| **PSRR** | Power Supply Rejection Ratio |
| **RAM** | Random Access Memory |
| **RFID** | Radio-Frequency Identification |
| **RSA** | Rivest, Shamir and Adleman; encryption algorithm |

| | |
|---|---|
| **ROM** | Read-Only Memory |
| **SCL** | Serial Clock |
| **SCLK** | Serial Clock |
| **SCPI** | Standard Commandos for Programmable Instruments |
| **SDA** | Serial Data |
| **SPI** | Serial Peripheral Interface |
| **TPM** | Trusted Platform Module |
| **UART** | Universal Asynchronous Receiver Transmitter |
| **uC** | Microcontroller |
| **USB** | Universal Serial Bus |
| **VCP** | Virtual Com Port |
| **ZIF** | Zero Insertion Force |

# Zusammenfassung

High Security Microcontroller (uC) werden in einem immer vielfältigeren Bereich eingesetzt (z.B. U-Bahn Tickets, Bankomat-Karten, Reisepässe, PayTV und vielen mehr). Dies erfordert immer mehr Fähigkeiten des Controllers. Vor allem werden verschiedenste Interfaces benötigt um den Markt zu bedienen. Jedes einzelne dieser Interfaces muss getestet und verifiziert werden. Aber es müssen nicht nur alle Interfaces für sich, sondern auch deren Interaktion miteinander getestet werden. Weiters geht der Trend zu immer stromsparenderen Geräten. Dies macht es erforderlich, dass Controller über verschiedene stromsparende Betriebs-Modi verfügt. Dadurch wird es unerlässlich, dass die Tests nicht nur in den jeweiligen Modi sondern auch im Übergang zwischen diesen Zuständen durchgeführt werden, um qualitativ hochwertige Aussagen treffen zu können.

Diese Diplomarbeit beschäftigt sich mit der Entwicklung einer Testumgebung, die es ermöglicht, solche Tests durchzuführen. Es wurde eine spezielle Versorgungsspannung entworfen, um zu erkennen, in welchem Betriebszustand sich der Controller befindet und um dieses zu signalisieren. Dadurch ist es möglich, vollautomatisch, in jedem beliebigen Betriebszustand und auch in den Übergangsphasen, Tests durchzuführen und unter anderem diese mittels Oszilloskop exakt grafisch darzustellen. Des Weiteren vereint dieses Projekt alle bisherigen Interfaces auf einem Board, was eine erhebliche Platzersparnis bringt. Da es nur eine Hardware gibt, wird auch nur eine Software mit nur einer Oberfläche benötigt. Das wiederum erleichtert die Handhabung und erspart dem Benutzer das Kennenlernen und Handhaben vieler Oberflächen. Um zukünftige Änderungen unterstützen zu können wurde die Anbindung der Interfaces sehr flexibel gestaltet. Aufgrund dieser Wandlungsfähigkeit erhielt das Projekt den Namen „Chameleon“.

# Abstract

High security uC are used in a very versatile market range. This requires more and more skills of the uC. Especially, different interfaces are needed to handle the requirements of the market. All of these interfaces have to be tested and verified but not only by themselves. Also the interaction with different interfaces has to be tested. Furthermore, there is a trend to low power devices and a necessity to support different power states. This requires testing not only in a steady state but also during the transition from one power state to another power state to make a high quality statement/verification.

This diploma thesis is about the development of a test environment which is able to fulfill the requirements. A special power supply was designed to recognize and signalize in which state the uC is. Thereby it is possible to make full automatic tests in every operating point and also in the transition-phase between different operating points. It is also possible to visualize this test via an oscilloscope. Furthermore, this project can handle all currently available interfaces on one single Printed Circuit Board (PCB). This reduces the needed area for the setup considerably. Because of a single hardware, there is only one software which handles all tests. This makes it easier for everyone to use hardware and the software and there is no need to know different softwares. The connections to the different interfaces where built very flexibly to support changes and new requirements for future products. Because of this adaptability, the project got the name ”Chameleon”, based on the animal.

# Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ______________________________ ____________________________________

Date Signature

# Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ______________________________ ____________________________________

Datum Unterschrift

[1] Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

# Contents

**Zusammenfassung** III

**Abstract** IV

**1 Introduction** 3

**2 Theory** 4

2.1 Device Under Test (DUT) 4

2.2 I2C 5

2.2.1 Dataframe 6

2.3 SPI 6

2.4 USB 7

2.5 LPC 7

2.6 ISO 7816: Contactbased Interface 8

2.7 ISO 14443: Contactless Interface 8

2.8 DCLB 8

2.9 Summery 9

**3 Hardware Implementation** 10

3.1 uC and Interface to PC 14

3.2 Power Supply for Core of the DUT 15

3.2.1 VDDP 16

Resistor after emitter/drain 17

Resistor in drain path (realized in Version 1.5) 19

OTA + voltage follower + current mirror 20

3.2.2 Triggering of State 26

Current sensing with current resistor in drain path 27

Current sensing with current mirror (for redesign) 28

Dimensioning of the resistors 29

3.3 Power Supplies for Interfaces 30

3.3.1 VDDPISO and VDDGPIO 30

3.3.2 VDDLPC 31

3.4 LPC 32

3.5 USB 33

3.6 ISOPAD source switching 34

3.7 DCLB 35

3.8 ISO 7816 Reader . . . . . 36
3.9 Headerboard . . . . . 39
3.9.1 I2C . . . . . 40
3.9.2 SPI . . . . . 42
3.10 Logicanalyzer and Pattern generator . . . . . 45
3.10.1 Clock Input . . . . . 47
3.10.2 Trigger . . . . . 47

**4 Firmware Development** **49**
4.1 SCPI-Parser Implementation . . . . . 50
4.2 Commands . . . . . 52
4.2.1 Voltage Source . . . . . 53
4.2.2 Switch Matrix . . . . . 54
4.2.3 Clock selecting . . . . . 55
4.2.4 ISO-Pad source selecting . . . . . 55
4.3 Control Software . . . . . 55

**5 Conclusion and Outlook** **59**
5.1 Pictures . . . . . 59
5.2 Improvements . . . . . 66

**6 Acknowledgements** **68**

**Bibliography** **69**

**7 Appendix** **75**

# 1 Introduction

High Security uC capture a wide range of usage which demands high flexibility. To handle all requirements of the market, a large amount of interfaces is necessary. All these interfaces have to be tested and verified. This takes a long time and a lot of effort. Currently, there is no setup that can manage all needed interfaces. There are setups available for a single interface but not for all together. This leads to several problems and disadvantages.

- Different setups for different interfaces are needed
- Every setup can have a different control program
- It takes time and space to rebuild a setup
- It is difficult to test the interaction of different interfaces
- Different PCBs for different packages and setups are needed
- . . . . . .

Furthermore there is one thing, which is impossible with all our setups: It is not possible to measure the supply current of the DUT without influencing the supply voltage. In our current setups, the supply voltage can change up to 20%. So, the operating point of the DUT is not very stable and unfortunately not very accurate.

An extreme is the low-power mode: In this mode, the current consumption is so low, that it is not possible to measure or detect the current with our current setups.

Another drawback of our current setups is, that they are optimized for only one package of the DUT. But time changes things and so the number of packages were also changed/increased. Therefor the number of adapters and expansion- boards increases.

This diploma thesis aims to address all of the above mentioned disadvantages and offer a single solution which solves all these problems to allow quick and simple get through verification of the uCs.

On the following pages, you will see how I managed all these problems and what can be improved.

*Note: All used links to datasheets and specifications were valid on the 19.12.2012*

# 2 Theory

*Note: The following sources were used: [1] [2] [3] [4] [5] [6] [7] [8] [9]*

The next sections will give an overview of the DUT and all implemented interfaces. These basics are important to understand why the problems have to be solved in the way as they are solved.

The theory is reduced to the absolute minimum. More information on these topics are easily accessed on the Internet or several books.This additional information is helpful but not necessary to understand the solutions.

## 2.1 DUT

The DUT is not a single chip type, it is a family of high security controllers. An overview can be found in [1].

The major features of this family, at current state, are:

- CPU:16-bit
- Memory
  - Flash: up to 400 kB
  - RAM: up to 8 kB
  - EEPROM: up to 160 kB
  - ROM: up to 280 kB
- Interfaces
  - I2C
  - SPI
  - USB
  - LPC
  - ISO 7816
  - ISO 14443

    - DCLB
- symmetrical encryption
    - DES
    - 3DES
    - AES up to 256-bit
- asymmetrical encryption
    - RSA up to 4096-bit
    - ECC up to 521-bit
- Temperature range: -25 to +85 °C

These controllers offer a contact less and different contact based interfaces. Therefor they can be used in many applications like ePassport, eSignature, Payment, Ticketing, NFC and many more.

## 2.2 I2C

The Inter-Integrated Circuit (I2C)-Bus is a synchronous serial bus with a master-slave topology which was developed by Phillips[1] in the 1980's. It is a two-wire bus (one data line "SDA" and one clock line "SCL"). This bus is used to connect several Integrated Circuit (IC)s over a short range. The Master controls the whole bus traffic. A slave can only send data, when the master requests data.
Because of the single data line, this line has to be bidirectional. To avoid a damage of a bus member the output driver of the interface is an open-drain configuration. A pull-up resistor is used to generate the high-state. Therefor it is impossible to generate a short circuit if one member wants to send a high and an other member wants to send a low. The low state will dominate. The value of the pull-up resistor is important for the transfer rate. Higher data rates require smaller resistors. The reason for this is the capacitive load of the lines and the resulting low-pass filter.

The standard supported data rates are

| | |
|---|---|
| Standard Mode | 100 kHz |
| Fast Mode | 400 kHz |
| Fast Mode Plus | 1 MHz |
| High Speed Mode | 3.4 MHz |

[1] Now NXP

### 2.2.1 Dataframe

An I2C command sequence begins with a start-condition followed by 7 address-bits and a read/write-bit. After this sequence, the slave sends an acknowledge signal (ACK or NACK) to the master. If the master gets an ACK, he knows, that the/a slave understood the command and is ready. If the read/write-bit is cleared, the master continues to send data to the slave. The slave acknowledges the reception of the data by sending an ACK (a ”low” on the dataline). If the data is not valid, the slave answers with a NACK (a ”high” on the dataline). To close the communication, the master sends a stop-condition. Please refer to the Specification of the I2C bus[2] for detailed information.

The start-condition is defined as a high-to-low transition on the Serial Data (SDA) line while the Serial Clock (SCL) is high. The stop-condition is defined as a low-to-high transition on the SDA line while SCL is high. A data bit is valid while SCL is high and its only allowed to change the SDA line while the SCL line is low.

The following picture shows a write sequence to the used analog switch matrix (ADG2128). Please, have a look at the datasheet of the ADG2128[3]

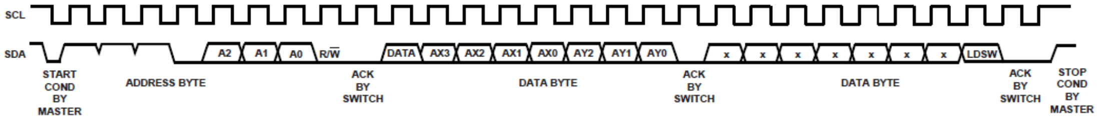


Figure 2.1: Write sequence to the ADG2128. Picture taken from the Datasheet of the ADG2128

It should be possible to test this bus over a voltage-range from 1.65 V up to 5.5 V and up to a frequency of 5 MHz.

## 2.3 SPI

Serial Peripheral Interface (SPI) is also a synchronous serial bus with master-slave topology. It was developed by Motorola. One difference to the I2C-Bus is, that the dataline is unidirectional and so there are two datalines needed. One line to send data from the master to the slave and vice versa. A slave is selected by a low on his Chip Select (CS)-Pin and not by an address on the dataline.

[2] The full specification can be found on the homepage of the I2C bus

[3] Analog Devices

| | |
|---|---|
| SCLK | Serial Clock |
| MOSI | Master out slave in |
| MISO | Master in slave out |
| #CS, #SS | Chip select, slave select |

SPI has four different modes. This mode indicates when the data are valid and will be sampled from the slave.

| Mode | CPOL | CPHA |
|---|---|---|
| 0 | Clock idle low | high to low edge |
| 1 | Clock idle low | low to high edge |
| 2 | Clock idle high | high to low edge |
| 3 | Clock idle high | low to high edge |

It should be possible to test this bus over a voltage-range from 1.65 V up to 5.5 V and up to a frequency of 33 MHz.

## 2.4 USB

Universal Serial Bus (USB) is a symmetrical serial bus with two bidirectional data lines. It is a master-slave-bus and is used for connecting different devices to a computer. There are different data-rates available:

| Name | Datarate |
|---|---|
| Low Speed | 1.5 Mbit/s |
| Full Speed | 12 Mbit/s |
| Hi-Speed | 480 Mbit/s |

In contrary to the voltage range of SPI and I2C, USB is only at 5 V ($V_{Bus} = 5$ V and $Datalines = 3.3$ V) specified.

The full specification of the USB-Bus can be found on the Homepage of USB.org.

## 2.5 LPC

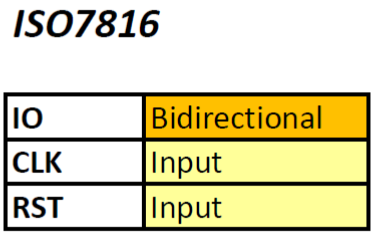
Low Pin Count (LPC) is a bus system which is used in personal computers. It is like an Industry Standard Architecture (ISA)-Bus but without the address lines. LPC is used to replace the ISA-Bus because it needs less space and less pins and is therefor cheaper. Furthermore it is more compatible with the common used frequencies. LPC is important for our DUT because this bus is used for Trusted Platform Module (TPM)-devices and our DUT can be used as such a device.

- LPC standard

- TPM Specification

## 2.6 ISO 7816: Contactbased Interface

The Contactbased (CB)-Interface consists of five lines. Two for the power ( GND and VDDPISO) and three for communication (RST, CLK, IO). This interface is used for contactbased applications like cash cards or pay-tv cards.



ISO7816

| | |
|---|---|
| IO | Bidirectional |
| CLK | Input |
| RST | Input |

Figure 2.2: Direction of the ISO 7816 Interface at the DUT

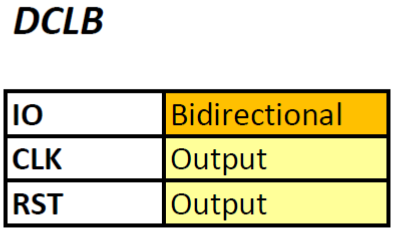
The protocol looks like Universal Asynchronous Receiver Transmitter (UART) communication with special settings but additionally there is a clock line. This clock signal is divided by 372 per default on the beginning of a communication at the DUT. This means, that in a bit-cycle on the Input-Output (IO)-line the clock line toggles 372 times.

## 2.7 ISO 14443: Contactless Interface

For contactless communication, we use the standard ISO 14443 at a carrier frequency of 13.56 MHz. We have to connect an antenna to the Contactless (CL)-Interface pins (LA and LB). The input capacitance of the chip and the inductance of the antenna form a resonant circuit. The power and the data come from the magnetic field. Data from the chip to the reader will be send also over the magnetic field by modulating the field strength. There are different common types of communication. Most common are Type A (100% modulation) and Type B (10% modulation).

## 2.8 DCLB

The Direct Contactless Bridge (DCLB)-Interface uses the same pins as the ISO 7816 Interface. But the RST-Line is not essential and can be left open. DCLB is used to expand a device without contactless interface with a contactless interface. For example, you have a normal uC (like used PIC18Fxxxx) and you want to build a device with a CL-Interface, you can add a Radio-Frequency Identification (RFID)-Chip with a DCLB-Interface in between. The RFID-Chip will modulate the data from the DCLB interface on the carrier of the CL-Interface.



DCLB

| | |
|---|---|
| IO | Bidirectional |
| CLK | Output |
| RST | Output |

Figure 2.3: Direction of the DCLB Interface at the DUT

## 2.9 Summery

As described throughout the chapter, there are a lot of different interfaces, all of which need to be handled as well as their various operating modes. To accomplish all these requirements, the board has to be very flexible and interchangeable. This transformation ability leads to the name ”Chameleon”, based on the highly adaptable animal.

# 3 Hardware Implementation

*Note: For finding and realizing the hardware, I used the following sources: [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32].*

The following page gives you an overview of the Chameleon board.

The dark blue blocks are ready-to-use devices from external companies. There are the USB-Hub, the pattern-generator and the logic-analyzer. The supply of the USB-Hub is used as power supply for the whole board.

The Major Blocks are the DUT-Socket, the headerboard, the different onboard-supplies, the uC with the computer connectivity and a large block for the needed supplies. This block will be explained in detail in a separate block diagram.

The connection to the DUT is build as headerboard with a Zero Insertion Force (ZIF)-socket. There are different headerboards available for all different packages of the DUT.

The CL-Interface and the USB-Interface require no bus-driver or leveltranslator, so they are connected directly to an appropriate connector.

There are three blocks with switches. One block is used to switch between a (RJ45) connector for an external ISO7816 reader and a multiplexer. This multiplexer can select the onboard ISO7816 reader, the DCLB source or the I2C source. The second block can switch between connectors external supply voltages or the internally generated supply voltages. A multiplexer is connected to the internal supply voltages and its output is connected to 4mm-banana-sockets to connect an external voltmeter. This can and should be used to verify the internal supplies automatically by a calibrated external voltmeter (i.e. a Keithley 2000 connected via General Purpose Interface Bus (GPIB) to the computer). All supplies are standard linear regulators except the supply for VDDP. This regulator provides an additional feature: A current sensing path is included and can be used to detect the different power-states of the DUT. Furthermore a current limitation is included. The last block of switches can change between an internal clock source (100 MHz) or an externally applied clock signal for the pattern-generator and the logic-analyzer.

The LPC block represents the bus converter which is needed for LPC to convert the unidirectional lines from the pattern generator to the needed bidirectional lines for the DUT.

The I2C- and the SPI-blocks are used to convert the output of the pattern generator into the appropriate output configuration (i. e. push-pull to open-drain) and translates the levels to meet the voltage-range-requirements. The switch-matrix routes the I2C and the SPI signals to the General Purpose Input Output (GPIO)s of the DUT. It can route the input signal from the pattern generator to every GPIO from the DUT.

The FTDI-Chip block provides two UART ports. Port 1 is used for the ISO7816 reader and Port 2 is connected to the uC for transferring the commands.

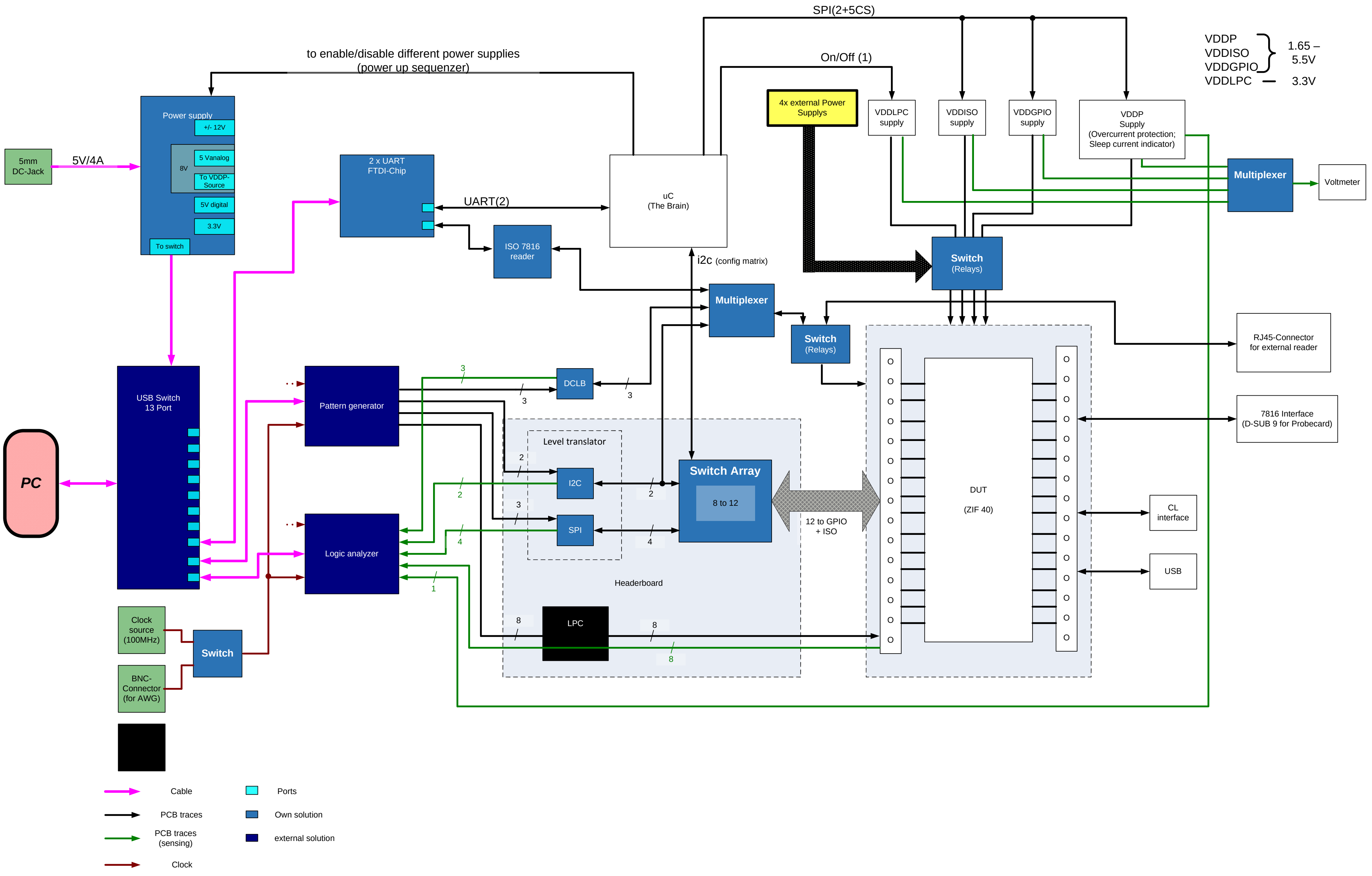
The uC is responsible for the entire control of the board. It controls the on-board power supplies, all switches and enables the supplies which are internally needed.

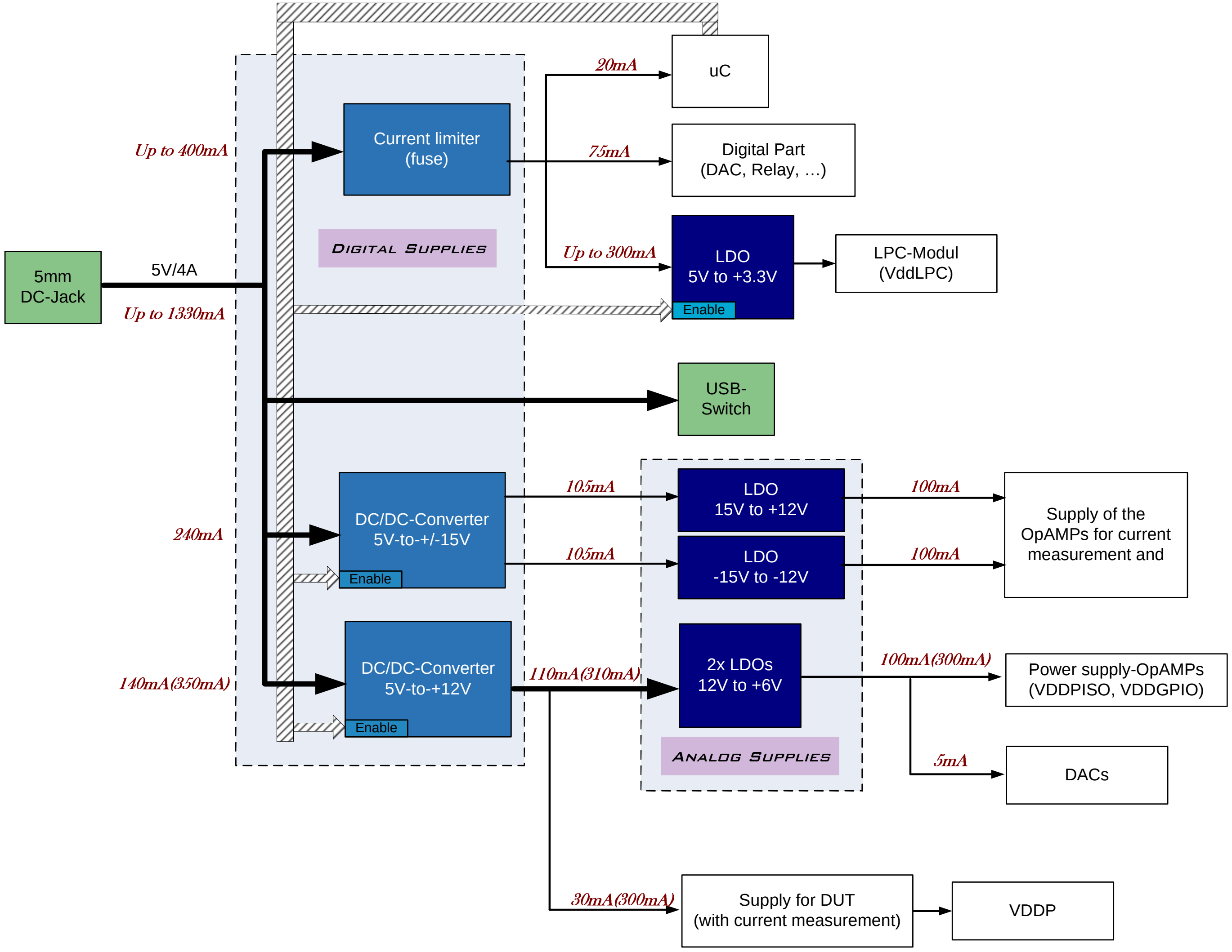
**Power supply block diagram**

Due to the high amount of different devices on board, there have to be different voltage levels and also different lines. The power comes from an extern DC-supply with 5 V/4 A (it is the supply of the USB-Hub). For the less important parts (”DIGITAL SUPPLIES”, the 5 V of the supply can be used directly but a current limitation device is included to prevent any damage of the modules. From this supply, the 3.3 V for the LPC Interface are generated.

For proper operation of the Operational Amplifier (OPAMP)s, the need for a positive and a negative supply with a recommended voltage arises. This is done by a DC-DC-Converter, which generates ±15 V out of the 5 V. To increase the performance and reduce ripples, the output voltages are reduced by linear regulators with a high Power Supply Rejection Ratio (PSRR).

The output voltage of the onboard regulators should be up to 5.5 V. Therefor it is necessary to generate a higher voltage level. This is done by a step-up-converter. Again, a linear regulator is used to increase performance and reduce ripples.


SPI(2+5CS)
On/Off (1)
to enable/disable different power supplies
(power up sequenzer)
VDDP
VDDISO
VDDGPIO
VDDLPC
1.65 –
5.5V
3.3V
5mm
DC-Jack
5V/4A
Power supply
+/- 12V
8V
5 Vanalog
To VDDP-Source
5V digital
3.3V
To switch
2 x UART
FTDI-Chip
UART(2)
uC
(The Brain)
ISO 7816
reader
i2c (config matrix)
4x external Power Supplys
VDDLPC
supply
VDDISO
supply
VDDGPIO
supply
VDDP
Supply
(Overcurrent protection;
Sleep current indicator)
Multiplexer
Voltmeter
Switch
(Relays)
Multiplexer
Switch
(Relays)
RJ45-Connector
for external reader
7816 Interface
(D-SUB 9 for Probecard)
CL
interface
USB
USB Switch
13 Port
PC
Pattern generator
Logic analyzer
DCLB
3
3
3
Level translator
2
I2C
2
2
3
SPI
4
4
Switch Array
8 to 12
12 to GPIO
+ ISO
Headerboard
DUT
(ZIF 40)
LPC
8
8
8
1
Clock
source
(100MHz)
BNC-
Connector
(for AWG)
Switch
Cable
Ports
PCB traces
Own solution
PCB traces
(sensing)
external solution
Clock


5mm
DC-Jack
5V/4A
Up to 1330mA
Up to 400mA
Current limiter
(fuse)
Digital Supplies
20mA
uC
75mA
Digital Part
(DAC, Relay, …)
Up to 300mA
LDO
5V to +3.3V
Enable
LPC-Modul
(VddLPC)
USB-
Switch
240mA
DC/DC-Converter
5V-to-+/-15V
Enable
105mA
105mA
LDO
15V to +12V
LDO
-15V to -12V
100mA
100mA
Supply of the
OpAMPs for current
measurement and
140mA(350mA)
DC/DC-Converter
5V-to-+12V
Enable
110mA(310mA)
2x LDOs
12V to +6V
Analog Supplies
100mA(300mA)
Power supply-OpAMPs
(VDDPISO, VDDGPIO)
5mA
DACs
30mA(300mA)
Supply for DUT
(with current measurement)
VDDP

## 3.1 uC and Interface to PC

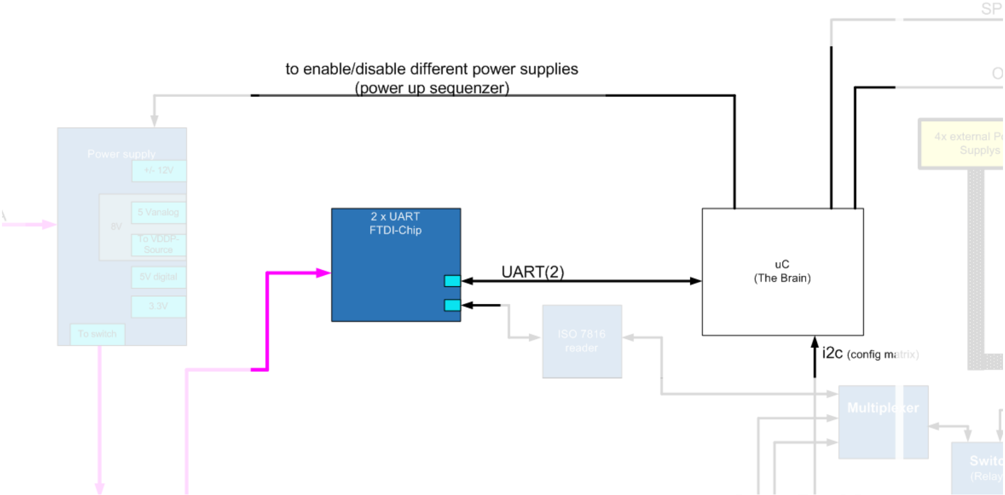


Figure 3.1: uC and Interface to PC

The Interface to the uC is realized by an FTDI-Chip (FT2232C). This chip provides an USB Slave Controller with two independent UART outputs. One output is used to communicate with the uC and the other output is used for the on-board ISO-7816 reader (this reader will be explained later).
The second channel of the device is used for communicating with the uC and is configured as a normal serial port.
The settings are:

| Baudrate | Databits | Stoppbits | Parity |
|---|---|---|---|
| 28800 | 8 | None | None |

Before you can use this Com-Port, you have to install the correct driver. Download the newest Virtual Com Port (VCP) driver from the download site of the FTDI-Chip homepage[1]. After installing the driver, you can communicate via a simple terminal program with the chameleon board. The best way to verify the communication is to send the command ”*IDN?” to the board. If the settings are correct, you will get the following identification string from the board,

```
Infineon Technologies Austria,Chameleon,Chameleon,1.1.1;
```

as answer.

[1] http://www.ftdichip.com/Drivers/VCP.htm

## 3.2 Power Supply for Core of the DUT

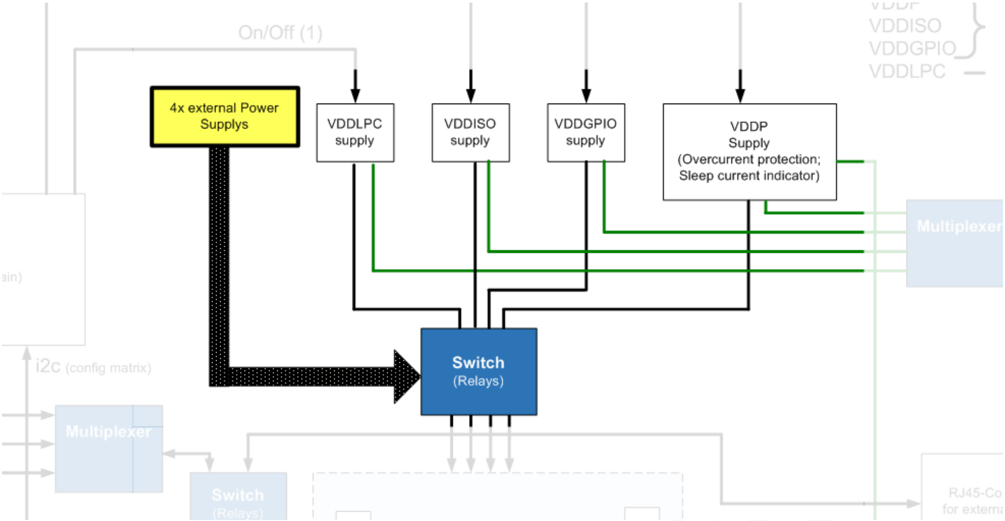


Figure 3.2: Power Supply for Core of the DUT

The requirements for the supplies of the core of the DUT are:

- Voltage range: 1.65 V to 5.5 V
- Current: up to 20 mA continuous
- Current spikes: up to 300 mA
- measuring the current-profile without influencing the supply voltage of the DUT

The user can choose between internally generated or externally connected supply voltages. I use a relay to switch between the sources for lowest influences and best performance. The supply pins of the DUT are connected to the external supply connectors per default. To connect it to the internal voltages, you have to send a command. You will find the command in the section Commands.

### 3.2.1 VDDP

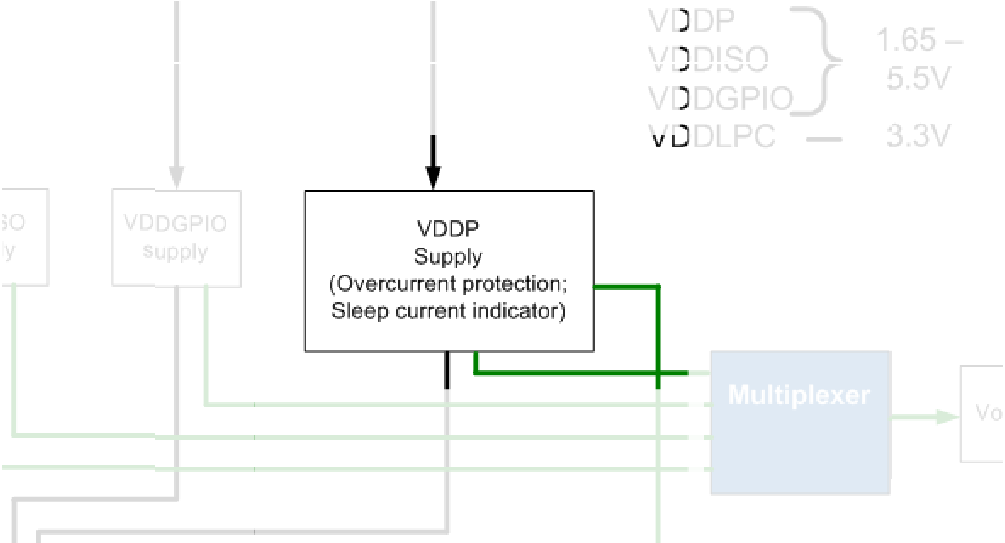


Figure 3.3: VDDP

All the following variants have the same input structure. The control voltage is generated by a Digital-to-Analog-Converter (DAC) and is amplified by a standard OPAMP. The gain of this amplification (set by R27 and R31) is 2.2. It is chosen so, that at full-scale output of the DAC, an output voltage of 5.5 V will be generated. This voltage has the same value as the output voltage of the regulator and is used to supply the different levelshifters for the interfaces and therefor does not influence the current sensing of the supply current of the DUT. R24 and R93 have the same value and form a voltage divider and divide V_leveltranslator by 2. R91 and R92 have the same value and form another voltage divider with a factor of 2. This configuration has two major advantages. The first advantage is, that the feedback-loop of the regulation has a higher gain. This increases the stability of the loop. The other advantage is related to the current limiter, which is included in this regulator. As you will read in the next sections, there is a current sensing included and this is also used to indicate an over-current. If the over-current is reached, the CURRENTLIMITER_FLAG is set to ”High” and activates the Metal–Oxide–Semiconductor Field-Effect Transistor (MOSFET)(it is conductive). Now, R93 is bypassed by a short circuit and the input voltage of the OPAMP IC8 is zero and also VDDP is zero. In this case, R24 is used as current limiter. CURRENTLIMITER_FLAG is connected to the uC and can be used to react on an over current (i.e. to reduce the input voltage).

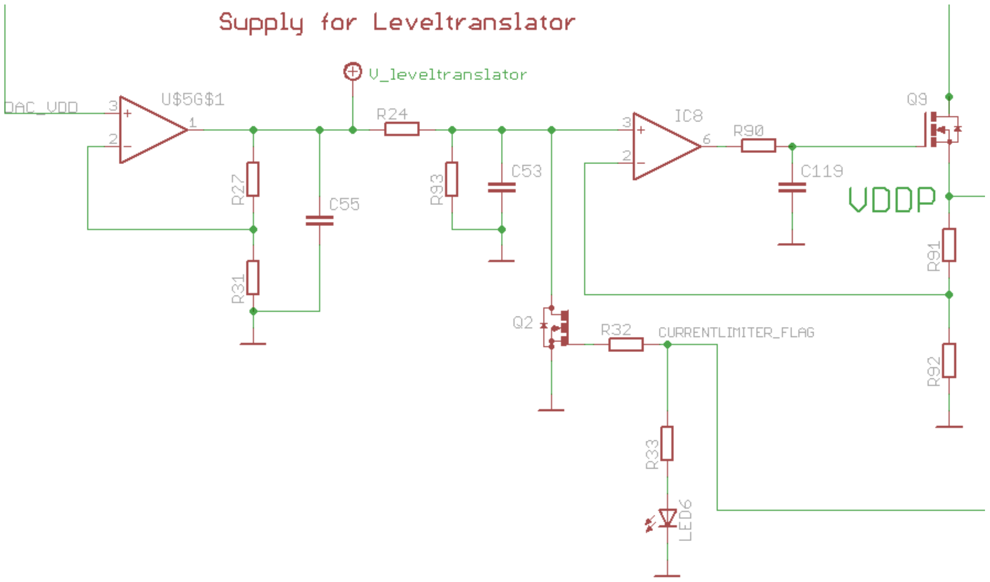


Figure 3.4: Amplification and current limiting of VDDP

### Resistor after emitter/drain

The first idea was to use a normal emitter/drain follower structure where the base/gate is controlled by an OPAMP. The output voltage to the DUT (VDDP) is fed back to the negative input of the OPAMP and the positive input is connected to the DAC, which sets the voltage. To measure the current through the DUT, a resistor R6 was added after the emitter/drain. One problem is that the OPAMP has to be very fast because all current spikes create a high voltage spike at the sense resistor R6 and lead to reducing the output voltage . Now, there is a voltage difference between the OPAMP inputs and this lead to an increase of the output voltage as long as the voltage difference is still applied. For fast load responses, this regulation loop has to be very fast and tends to be instable.

| Advantage | Disadvantage |
|---|---|
| low common mode voltage<br>BJT & FET can be used | instable<br>response time<br>wrong results |

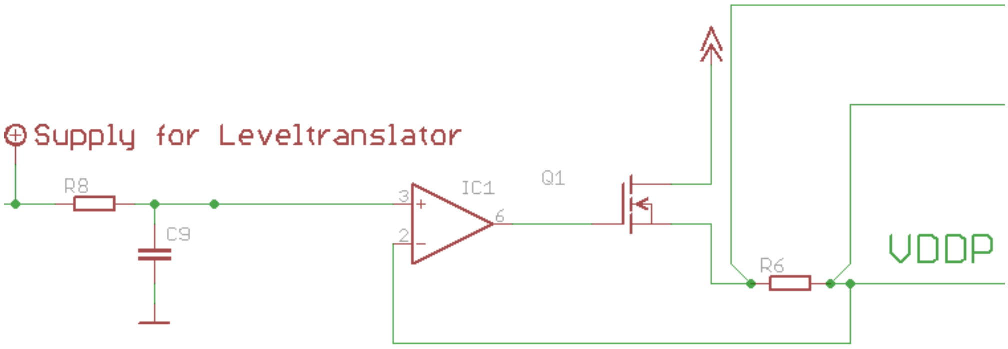


Figure 3.5: Resistor after emitter/drain

### Resistor in drain path (realized in Version 1.5)

The difference to the previous structure is, that the resistor (previously R6, now R25) has been moved from the emitter-path to the collector/drain-path. This results in a slower regulation loop and therefor the OPAMP's requirement can be decreased and the load response can be increased.

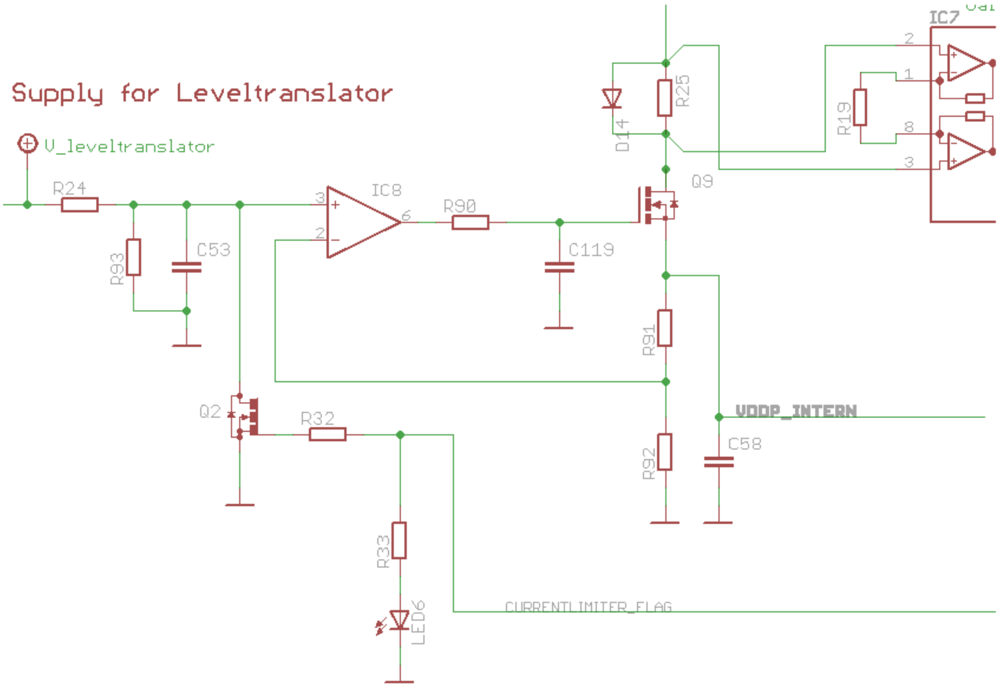
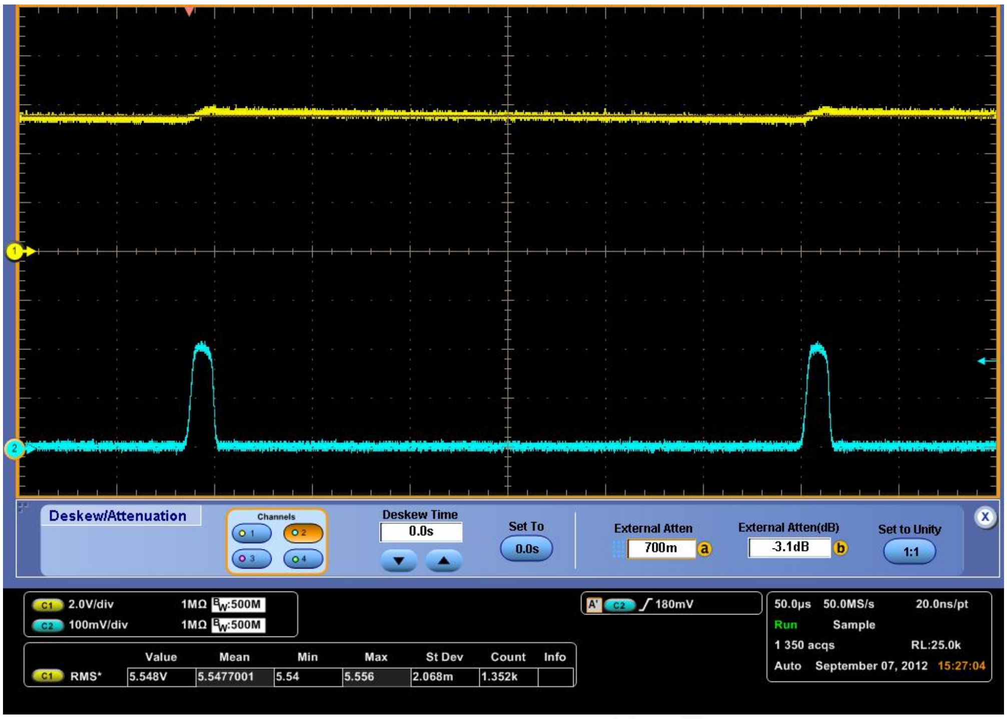


Figure 3.6: Voltage regulator with sense resistor in the drain path

It is important to use a MOSFET instead of a Bipolar Junction Transistor (BJT). If aBJT is used, it is impossible to detect the sleep state due to the low current. The DUT will be supplied mostly by the base current and not by the collector current (sense current). If you use a MOSFET, there is no gate current and all other current flows throw the measurement resistor.

Figure 3.7 shows the results of the regulator without any load attached. The yellow line represents the output voltage VDDP. The cyan line represents the output of the instrumentation amplifier and shows the converted current signal. The expected results is two straight lines. Yellow at 5.5 V and cyan at 0 V.



Deskew Time 0.0s

External Atten 700m

External Atten(dB) -3.1dB

C1 2.0V/div 1MΩ BW:500M

C2 100mV/div 1MΩ BW:500M

| | Value | Mean | Min | Max | St Dev | Count | Info |
|---|---|---|---|---|---|---|---|
| C1 RMS* | 5.548V | 5.5477001 | 5.54 | 5.556 | 2.068m | 1.352k | |

A' C2 ∫ 180mV

50.0μs 50.0MS/s 20.0ns/pt
Run Sample
1 350 acqs RL:25.0k
Auto September 07, 2012 15:27:04

Figure 3.7: Output voltage (yellow) of the regulator with converted sense current (cyan)

A problem which has been recognized is, that the drain potential changes to much at load response. This lead to swinging of the output voltage at a load step. A capacitor (C58) has been added between ground and drain to improve the performance of the voltage regulation. This however decreases the performance of the current measurement part. Fast current spikes will be buffered by this capacitor and will not flow through the resistor. So it is not possible to see a current profile at the resistor. You just see a filter profile.

| Advantage | Disadvantage |
|---|---|
| stable<br>fast response<br>more flexibility at OPAMP choice | higher common mode voltage<br>only FETs can be used<br>filtered current profile |

### OTA + voltage follower + current mirror

This construction consists of an Operational Transconductance Amplifier (OTA) followed from a voltage follower. A current mirror is used to mirror the current, flowing through the DUT, to a sensing path.

An OTA is an amplifier with a differential input stage (voltage input) and generates an output current depending on the voltage difference at the input.

Figure 3.9 shows the discretely built OTA. T1A and T1B form the differential input stage. T5 is the current source which generates the bias current for the input stage. The bias current is set by the resistor R3.

$$I_{Bias} = \frac{8\ \mathrm{V} - U_{BE}}{R_3} = \frac{8\ \mathrm{V} - 0.6\ \mathrm{V}}{1\ \mathrm{k}\Omega} = 7.4\ \mathrm{mA} \tag{3.1}$$

T2, T3 and T4 are normal current mirrors. T3 is responsible for sourcing a current. T4 is responsible for sinking current. T2 converts the high side current through T1A to a low side current. If both input voltages are equal, the current through the transistors are also equal. Therefor the mirrored currents are the same. This means, that the sourced current from T3 will be sinked by T4 and the output current is zero. If the positive input voltage (at T1B) is higher than the voltage at the negative input, the current through T1B and the source current rises but the current through T1A and the sink current decreases. This leads to a positive output current (current will be sourced). If the positive input voltage (at T1B) is less than the negative input voltage (T1A) it works the other way around and the sink current rises and the source current decreases. So we get a negative output current (current will be sinked).

| Input | Current stage | Output current |
|---|---|---|
| $V_{in+} = V_{in-}$ | $I_{Source} = I_{Sink}$ | $I_{out} = 0$ |
| $V_{in+} > V_{in-}$ | $I_{Source} > I_{Sink}$ | $I_{out} > 0$ |
| $V_{in+} < V_{in-}$ | $I_{Source} < I_{Sink}$ | $I_{out} < 0$ |

The output current is stored in a capacitance and generates a voltage. The transistor Q1 buffers this output voltage but reduces it by the threshold voltage of the transistor.

**3a.) Simple current mirror:**
The first idea was to use a simple current mirror. $I_{DUT}$ is mirrored to $I_{Sense}$ .

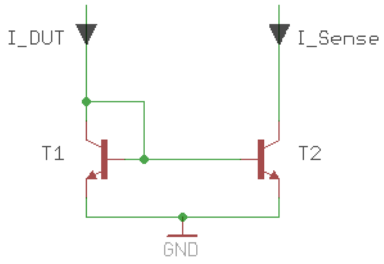


Figure 3.8: Current mirror

The expected gain was 10 but the measured gain was 23.8. The problem is the big difference between the collector potentials of the current mirror.

A current mirror consists of two matched (identical) transistors. One transistor (T1) is connected as a diode and acts like a voltage stabilizer. The base is connected to the collector and only the Base-Emitter diode exists. The second transistor (T2) is used as normal amplifier. The trick of the current mirror is, that both transistors have the same operating point. A higher current in the diode leads to a little bit of a high voltage drop of the diode. This voltage drop is the input voltage of the amplifier and will increase the conductivity (reduce the resistance) of the collector-emitter path. This leads to a higher current.

This prediction is only valid, if both transistors are in the same operating point and that means, the collector potential of both transistors has to be the same.

And this is the problem of the simple current mirror. The collector potential of the left transistor is nearly constant. But the collector potential of the right transistor can swing up to the rails of the supply. So the two transistors are in quite different states and this changes the gain of the current mirror.

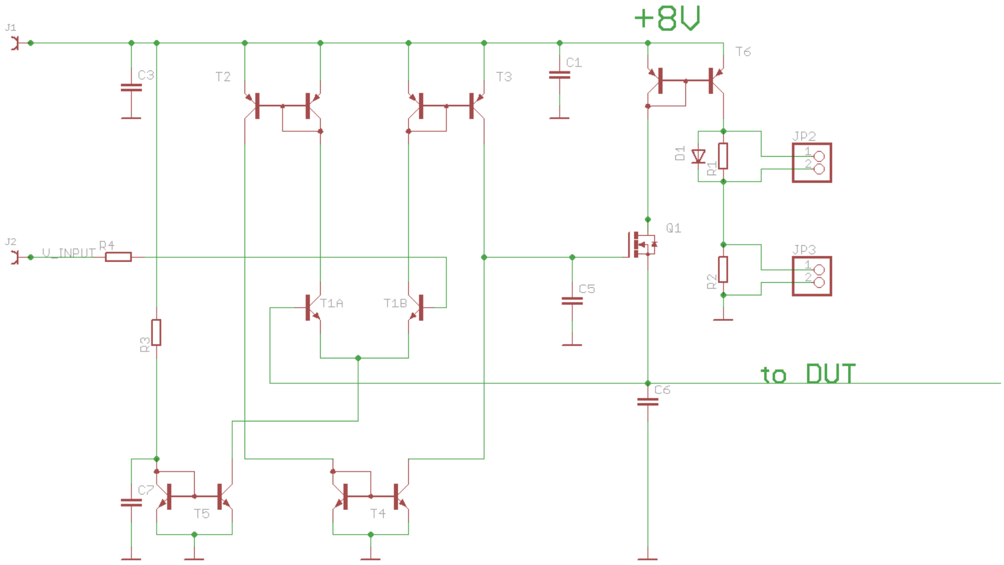
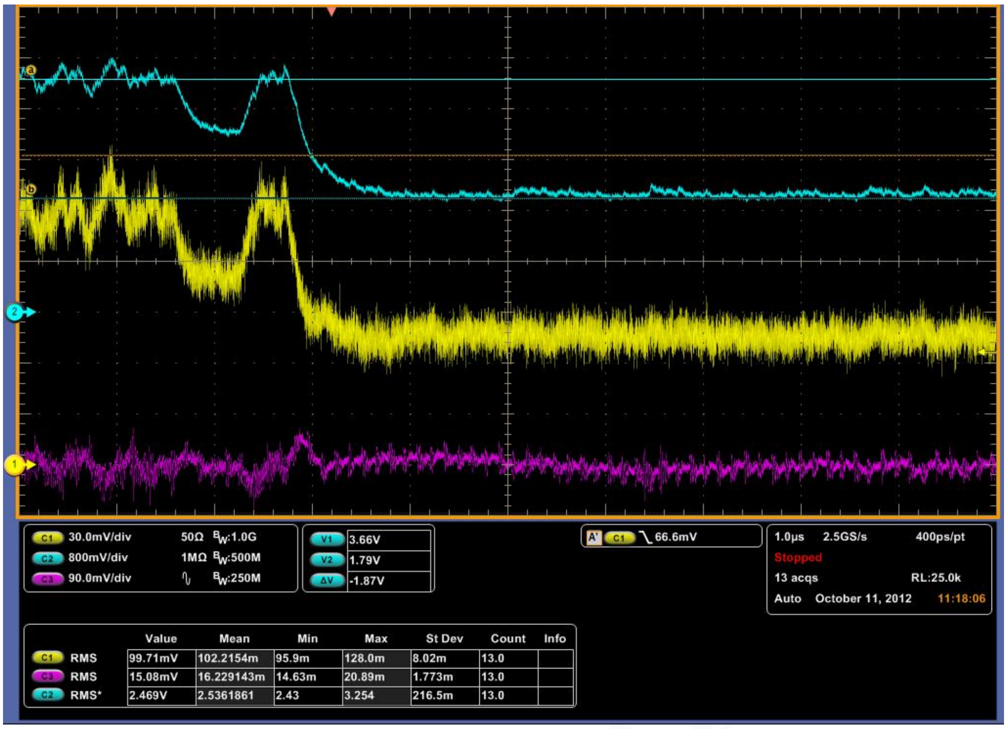


Figure 3.9: OTA with current mirror



C1 30.0mV/div 50Ω BW:1.0G
C2 800mV/div 1MΩ BW:500M
C3 90.0mV/div BW:250M

V1 3.66V
V2 1.79V
ΔV -1.87V

A' C1 ╲ 66.6mV

1.0µs 2.5GS/s 400ps/pt
Stopped
13 acqs RL:25.0k
Auto October 11, 2012 11:18:06

| | | Value | Mean | Min | Max | St Dev | Count | Info |
|---|---|---|---|---|---|---|---|---|
| C1 | RMS | 99.71mV | 102.2154m | 95.9m | 128.0m | 8.02m | 13.0 | |
| C3 | RMS | 15.08mV | 16.229143m | 14.63m | 20.89m | 1.773m | 13.0 | |
| C2 | RMS* | 2.469V | 2.5361861 | 2.43 | 3.254 | 216.5m | 13.0 | |

Figure 3.10: Current profile measured with current mirror

| | Voltagedrop on resistor |
|---|---|
| $I_{sense}$@100Ω | 3.66 V |
| $I_{DUT}$@10Ω | 0.154 V |
| Gain | 23.8 |

| Advantage | Disadvantage |
|---|---|
| stable<br>fast response<br>single ended signal<br>high voltage swing/ high resolution<br>low common mode voltage | current gain not constant<br>less parts available |

Table 3.1: Advantages and disadvantages of the current mirror

The schematic in figure 3.9 has very useful advantages. The voltage drop, which is proportional to $I_{DUT}$ is now a single ended signal (this means, one reference point is ground). This results in the possibility to use normal (cheap) oscilloscope probes instead of differential probes (expensive).

**3b.) Cascode current mirror:**
To solve the problem with the different collector potentials, we just have to add two additional transistors. This transistor pair looks like a current mirror, but it has a different functionality. This pair acts like an emitter follower with a reference voltage. The left transistor acts as diode and generates a reference voltage. The right transistor acts like a voltage regulator. Both transistors have the same base-emitter-voltage, because they are matched and as well have the same current. So, the transistors of the current mirror have the same collector potential. The transistor T4 therefor limits the effect of the output voltage swing on the current mirror.

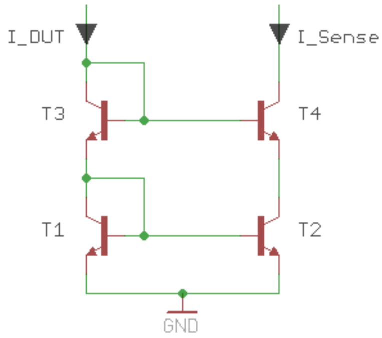
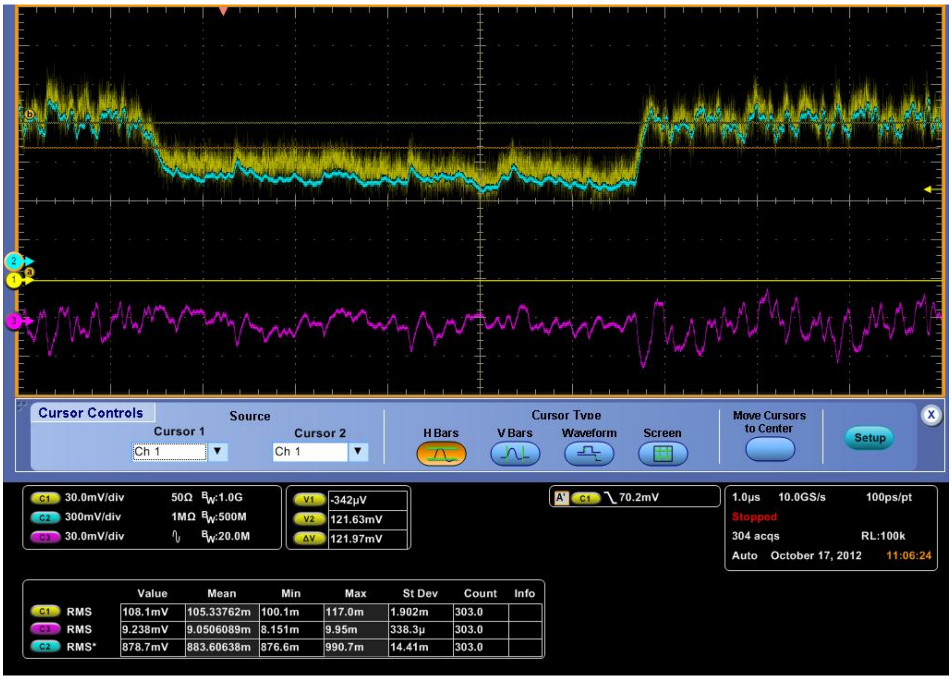


Figure 3.11: Cascode current mirror



C1 30.0mV/div 50Ω BW:1.0G
C2 300mV/div 1MΩ BW:500M
C3 30.0mV/div BW:20.0M

V1 -342μV
V2 121.63mV
ΔV 121.97mV

A' C1 ╲ 70.2mV

1.0μs 10.0GS/s 100ps/pt
Stopped
304 acqs RL:100k
Auto October 17, 2012 11:06:24

| | Value | Mean | Min | Max | St Dev | Count | Info |
|---|---|---|---|---|---|---|---|
| C1 RMS | 108.1mV | 105.33762m | 100.1m | 117.0m | 1.902m | 303.0 | |
| C3 RMS | 9.238mV | 9.0506089m | 8.151m | 9.95m | 338.3μ | 303.0 | |
| C2 RMS* | 878.7mV | 883.60638m | 876.6m | 990.7m | 14.41m | 303.0 | |

Figure 3.12: Current profile measured with cascode current mirror

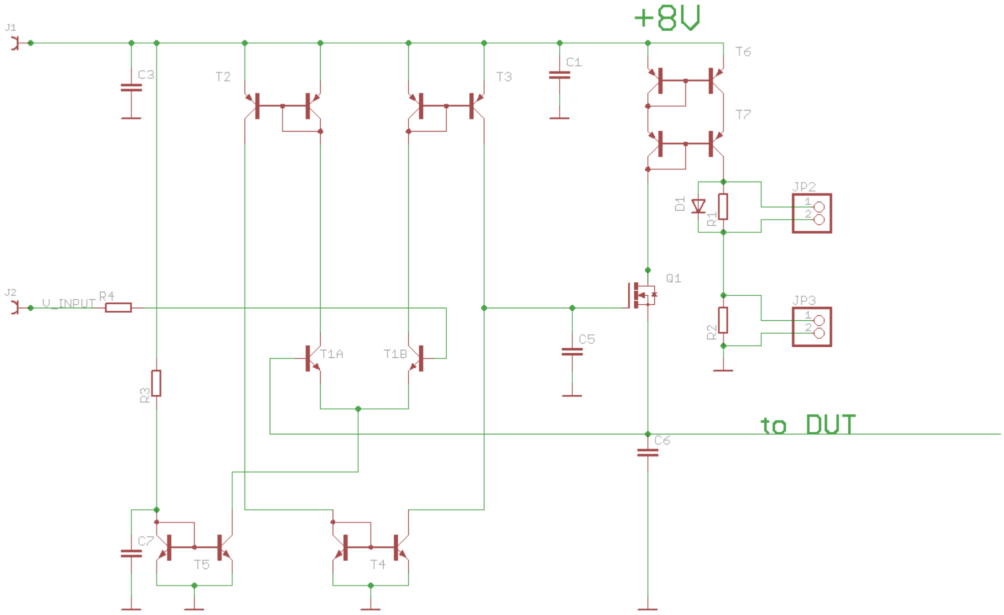
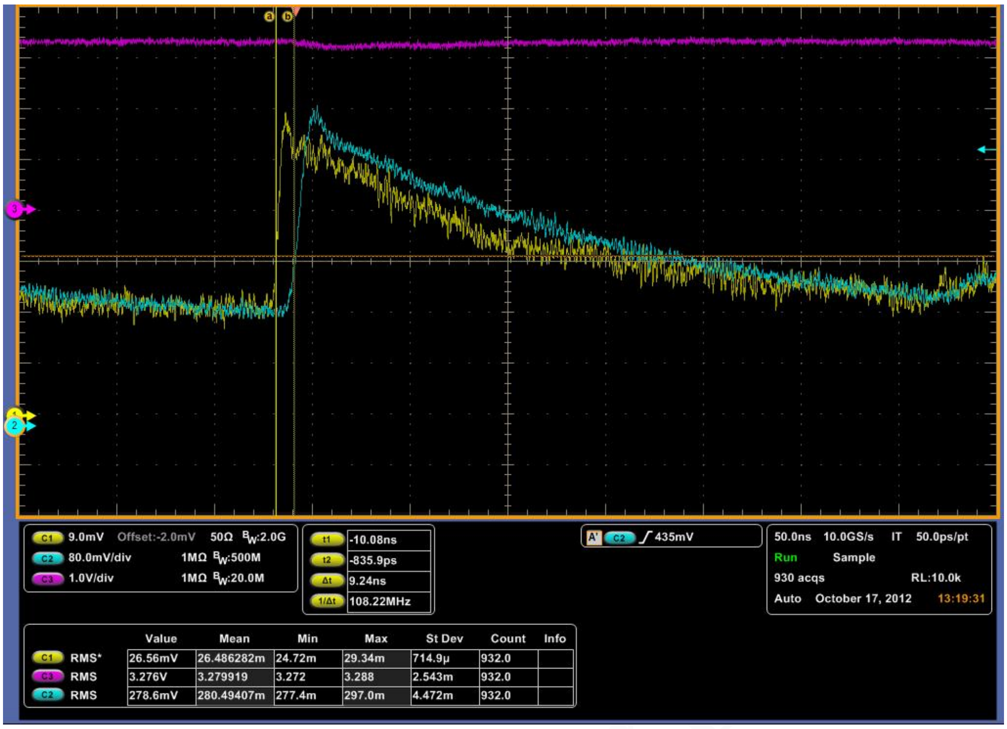


Figure 3.13: OTA with cascode current mirror

As can be seen in figure 3.12, the performance is very good. The load regulation is about 60 mV at a current step of 6 mA. The current gain is 1 (Note: the yellow line is a 10 Ω shunt in series to the DUT and the cyan line is the voltage drop at the 100 Ω resistor R2 of figure 3.9) .

As seen in figure 3.14, the delay between $I_{DUT}$ and $I_{Sense}$ is about 9.24 ns.



C1 9.0mV Offset:-2.0mV 50Ω BW:2.0G
C2 80.0mV/div 1MΩ BW:500M
C3 1.0V/div 1MΩ BW:20.0M

t1 -10.08ns
t2 -835.9ps
Δt 9.24ns
1/Δt 108.22MHz

A' C2 ∫ 435mV

50.0ns 10.0GS/s IT 50.0ps/pt
Run Sample
930 acqs RL:10.0k
Auto October 17, 2012 13:19:31

| | | Value | Mean | Min | Max | St Dev | Count | Info |
|---|---|---|---|---|---|---|---|---|
| C1 | RMS* | 26.56mV | 26.486282m | 24.72m | 29.34m | 714.9µ | 932.0 | |
| C3 | RMS | 3.276V | 3.279919 | 3.272 | 3.288 | 2.543m | 932.0 | |
| C2 | RMS | 278.6mV | 280.49407m | 277.4m | 297.0m | 4.472m | 932.0 | |

Figure 3.14: Delay of $I_{DUT}$ (yellow) and $I_{Sense}$ (cyan)

With this configuration, we got an additional advantage. The current gain is constant!

### 3.2.2 Triggering of State

One important requirement is to detect and indicate the different power modes of the DUT. This modes can be recognized by the value of the supply current. The trigger output is important because one test-case is to send data during the transition from an active state to the sleep state. To verify if the DUT is in the sleep state and the data is sent at the specific time, this output is used to generate a trigger pulse at the logic analyzer which can be used to trigger an external oscilloscope.

Requirements:

- sleep current: less then 100 µA
- active current: more then 1 mA
- current spikes: up to 100 mA (useful: 50 mA)

Note: The following description and device labels refer to figure 3.13.

Because of the wide range of current, an automatic range selection to fit all requirements had to be implemented.

### Current sensing with current resistor in drain path

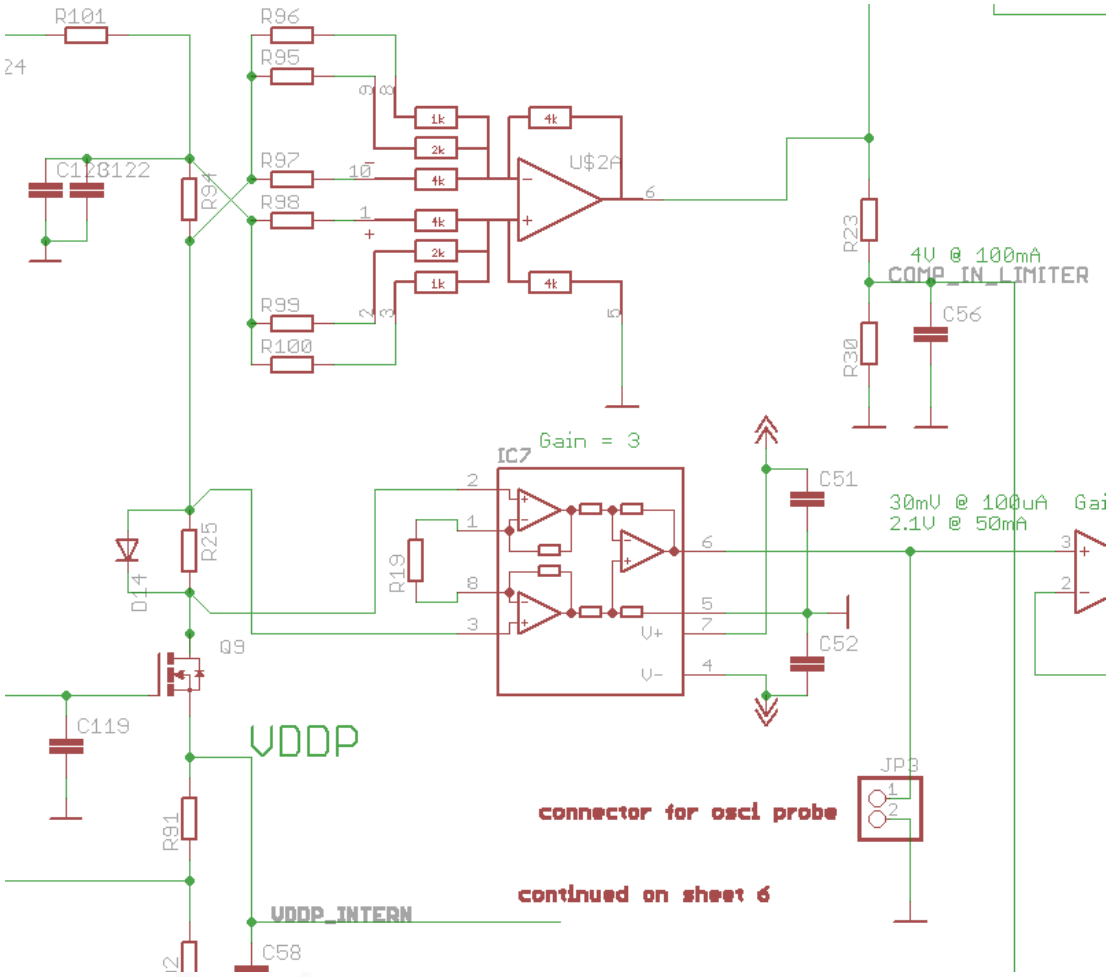


Figure 3.15: Schematic of the current sensing path of the realized version (Chameleon V.15)

The current through the DUT flows through R25 and R94 and generates a voltage drop. This voltage drop is amplified and compared with a reference voltage by a comparator. A high at the output of the comparator indicates, that the trigger level is reached.

For the high currents, R94 in connection with U$2A is used. This amplifier is a differential amplifier with a high common mode voltage range and it is very fast (to detect all current spikes) but has a high input current (low input resistance).

R25, D14 and IC7 are used for the low currents. The diode acts as a voltage limiter and limits the voltage drop at R25 to approximately 0.6 V. For better results, the usable voltage range for R25 should be limited with 300 mV (depending on the diode). IC7 is an instrumentation amplifier with a high common mode voltage range and a very small input bias current (very high input resistance) but the bandwidth is not as high as the differential amplifier's.

So, the arrangement of the amplifiers is very important. The input current of the differential amplifier is in the range of the sleep current of the DUT. If you exchange the high current sensing part with the low current sensing part, the latter would detect the input current of the differential amplifier. A static current could be calibrated. But the input current of the differential amplifier depends on the input voltage and this voltage changes often and also very fast. So, this current cannot be compensated for.

### Current sensing with current mirror (for redesign)

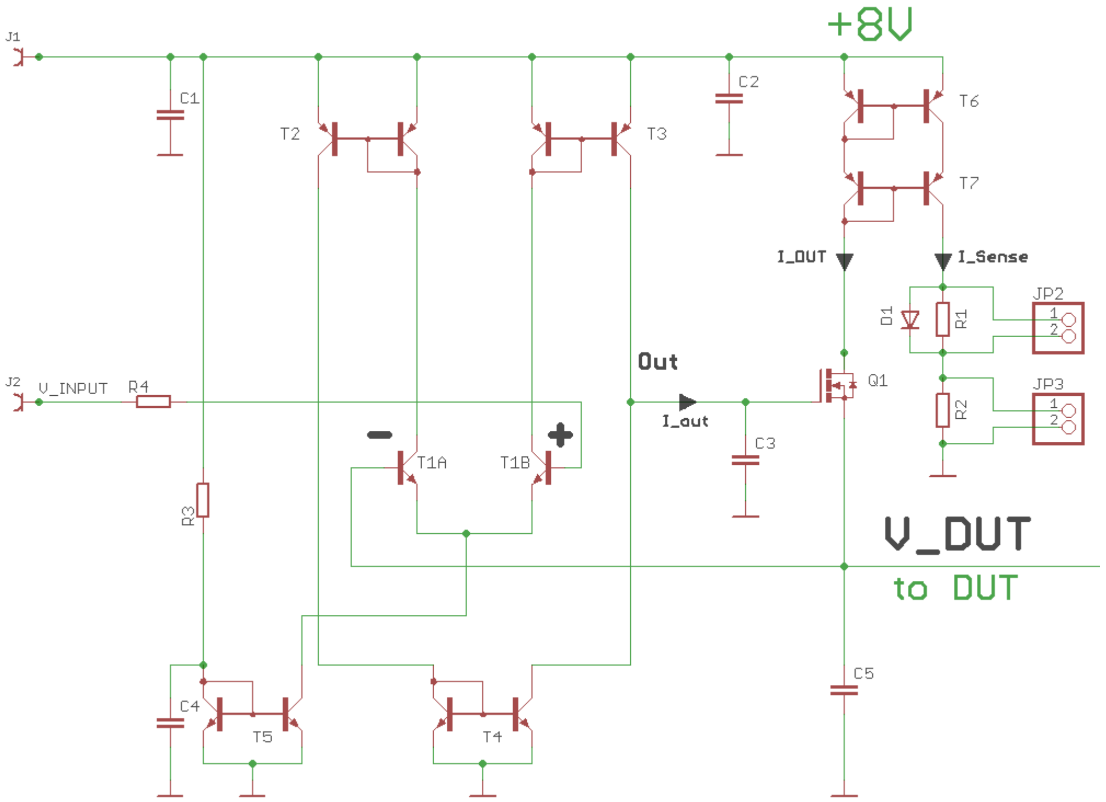


Figure 3.16: Schematic of the current sensing path with the OTA

The current through the DUT is mirrored by using the cascoded current mirror to the sense current $I_{sense}$. The sense current flows throw the resistors and generates a voltage drop. $R_1$ is 100 times larger then $R_2$ and therefor the voltage drop is 100 times larger at $R_1$ than $R_2$. The voltage at the resistor $R_1$ will be limited by the diode which acts as the range selection. A small current generates a very small voltage drop at $R_2$ but a bigger voltage drop at $R_1$. Resistor $R_1$ has to be chosen carefully. For best results, choose the maximum usable voltage at $R_2$ to be less then 300 mV. This value depends on the used diode and should be much less than the forward break down voltage. In this region, the diode is almost linear. This non linearity is not very important because we have to calibrate this region anyway. For such small currents, the bias current of the OTA and the base current of the transistors of the cascode are no longer negligible. At higher currents, the voltage drop at $R_1$ is limited by the diode and is approximately 0.7 V and almost constant. This change does not influence the results, because now only the resistor $R_2$ is used for measuring and $R_1$ is out of its usable range. In this case, the bias currents of the OTA and the cascode are negligible.

This voltages will be amplified with an OPAMP and an instrumentation amplifier. Because of the common mode voltage at $R_1$, we have to use the instrumentation amplifier to convert this voltage drop to a single ended signal. The OPAMP buffers and amplifies the voltage drop at $R_2$. These output signals are compared by a comparator with a reference voltage, which are generated by DACs. The reference voltages are controlled by the uC and can be set by the user by using the Standard Commandos for Programmable Instruments (SCPI) commands.

### Dimensioning of the resistors

The maximum allowed voltage drop at R1 is 300 mV. This gives us

$$R_1 = \frac{300\ \text{mV}}{I_{sleep\_max}} = \frac{300\ \text{mV}}{100\ \text{uA}} = 3\ \text{k}\Omega. \tag{3.2}$$

For the maximum voltage drop at R2 we have to consider the collector-emitter voltage of T6 and T7 and the forward breakdown voltage of D1. The collector-emitter voltage of T6 is the same as the as the base-emitter voltage of T6, because of the cascode. The collector-emitter voltage of T7 is given by the saturation voltage of this transistor and should be choosen higher then the saturation voltage.

$$U_{R2\_max} = 8\ \text{V} - U_{CE\_T6} - U_{CE\_T7} - U_{D1} = 8\ \text{V} - 0.7\ \text{V} - 0.4\ \text{V} - 0.7\ \text{V} = 6.2\ \text{V} \tag{3.3}$$

$$R_2 = \frac{U_{R2\_max}}{50\ \text{mA}} = \frac{6.2\ \text{V}}{50\ \text{mA}} = 124\ \Omega \tag{3.4}$$

For easier calculation, we use a 100 Ω resistor for $R_2$.

Our standard shunt resistor for measuring current profiles was 10 Ω but now we can use 100 Ω. This increases the resolution and also the accuracy, because the accuracy of a scope is worse at very low input voltages and gets better at higher voltages. And now we increased the voltage by a factor of 10.

## 3.3 Power Supplies for Interfaces

### 3.3.1 VDDPISO and VDDGPIO

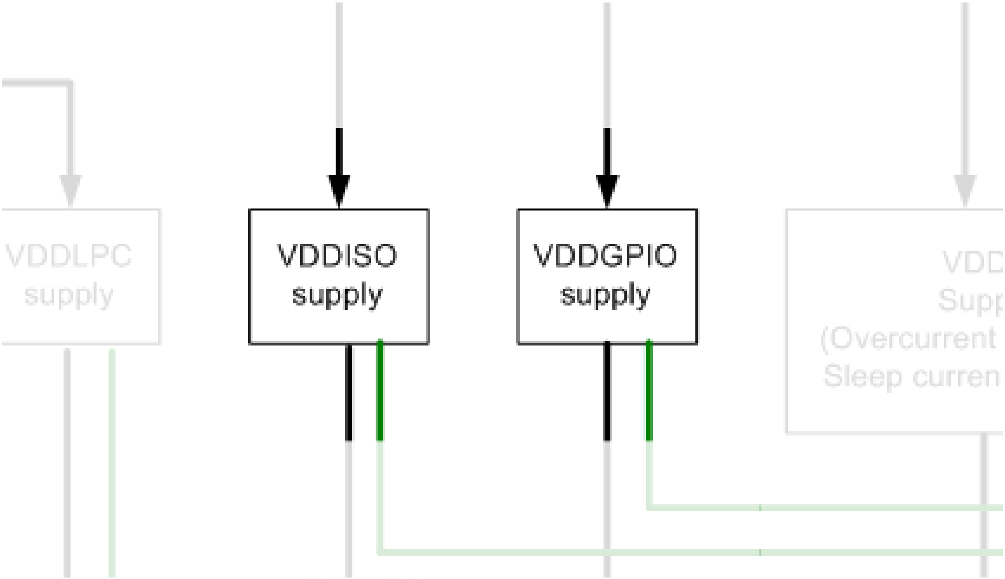


Figure 3.17: VDDGPIO

At the beginning of this project there were several approaches to implement the power supply setup. For Example:

- 10V DAC with a Z-Diode for voltage limiting
- 10 V DAC with a voltage divider and OPAMP
- 5 V DAC with an OPAMP and a Gain of 1.1
- 2.5 V DAC with an OPAMP and an output transistor

The last idea was put into practice because it allows the use of commonly used devices, which were easy to get and easy to replace. Another advantage regarding the 5 V DAC idea is, that the gain of the feedback loop is higher. A higher feedback loop results in more stability because of a higher phase margin at the unity gain frequency.

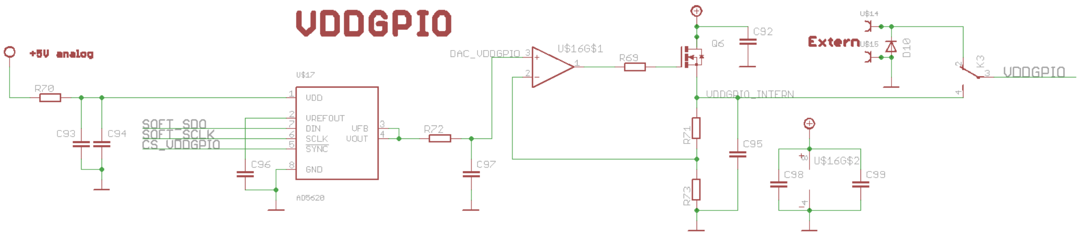


Figure 3.18: Schematic of the regulator for VDDGPIO and VDDPISO

R70 of the schematic in figure 3.18 is not a normal resistor. It is an 1000Z device. This device behaves like an inductance but has a non linear resistor characteristic over the frequency. This means, that the resistance of the device rises with higher frequencies. The advantage of such a device is, you can build a low pass filter with a second order which will not be able to oscillate (because the quality factor decreases with higher frequencies).

In the next revision, the same regulator topology will be used as has been for VDDP. The performance of this regulator is better than that of the currently used one. The load regulation and the stability is also better.

## 3.3.2 VDDLPC

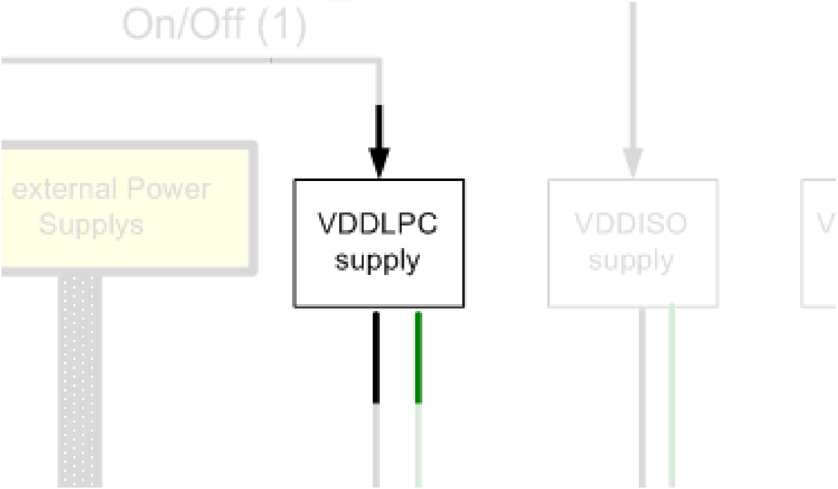


Figure 3.19: VDDLPC

This interface is specified for a supply voltage of 3.3 V. So it is enough to include a 3.3 V Low Drop Out (LDO) voltage regulator with an enable input.

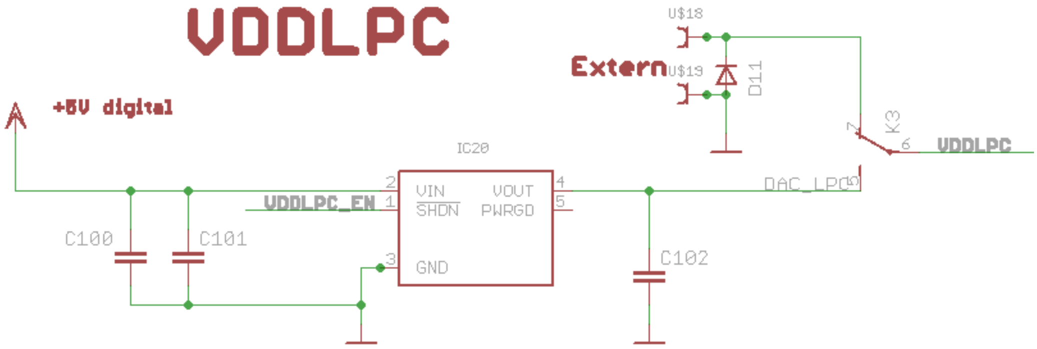


Figure 3.20: Schematic of the VDDLPC supply

## 3.4 LPC

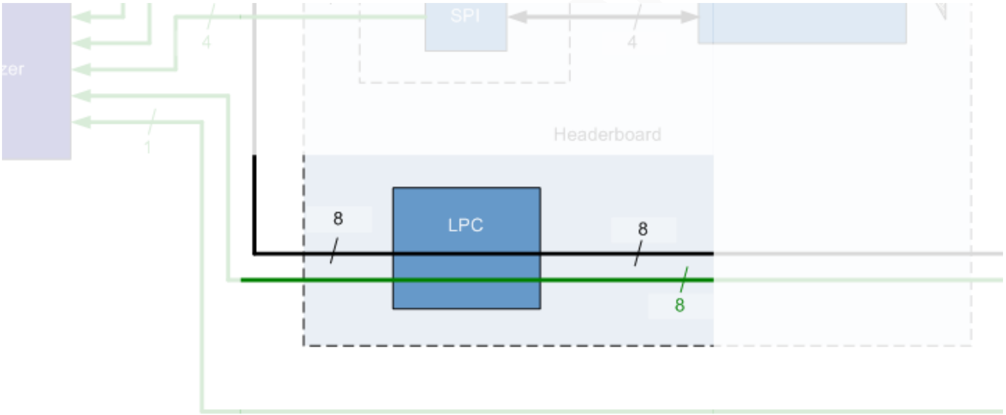


Figure 3.21: LPC

As previously described, this module is only specified for 3.3 V. This means, that there is no need for a leveltranslator. But we have to consider that the patterngenerator has push-pull outputs and the data lines of the LPC-Bus are bidirectional. This is a problem! If the Master applies a ”high” on a data line and the slave pulls down the line, the output current of the patterngenerator will be exceeded and could damage it and the data will not be recognized correctly. To solve this problem, a buffer with tristate output was added. By applying a ”low” on the pins $\overline{OE_1}$ and $\overline{OE_2}$, the buffer is in the push-pull-mode and acts as an output. If either $\overline{OE_1}$ or $\overline{OE_2}$ is ”high”, the buffer is in high-impedance-mode and does not influence the signal on the line. This is the state, where the slave can send data. Both output enable lines are connected together and are controlled from the patterngenerator. After sending data from the master (patterngenerator) to the slave (DUT), the OE-Pins have to be set to ”high”, in order for the slave to send its answer.

Due to the high speed of this bus, the delays on the lines are critical. To ensure, that all lines have the same delay, a buffer to the control lines has been added. These buffers add the same delay to the control lines as the data lines have. So the signals are all delayed but synchronized.

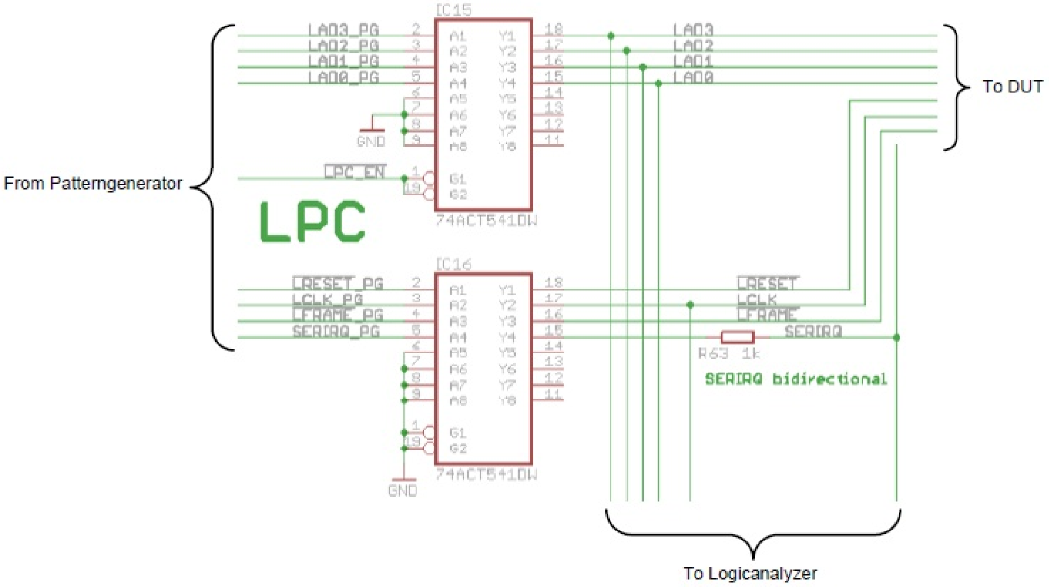


Figure 3.22: Buffer for the LPC Interface

## 3.5 USB

There is no need for a leveltranslator for the USB Interface, because it is only supported at the specified voltage. Jumpers to swap the data lines have been added. This is necessary for different pinnings of the packages.

## 3.6 ISOPAD source switching

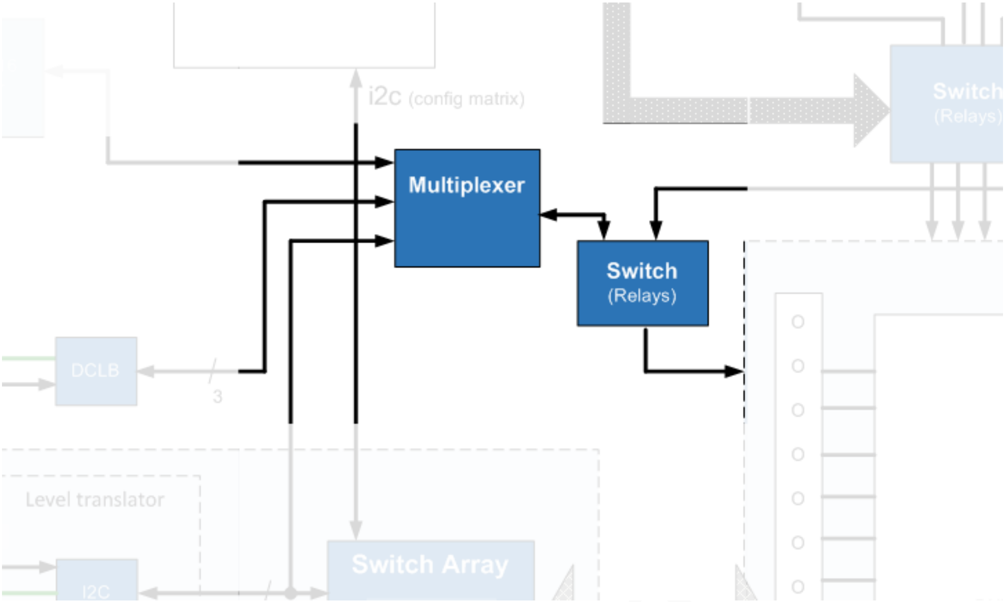
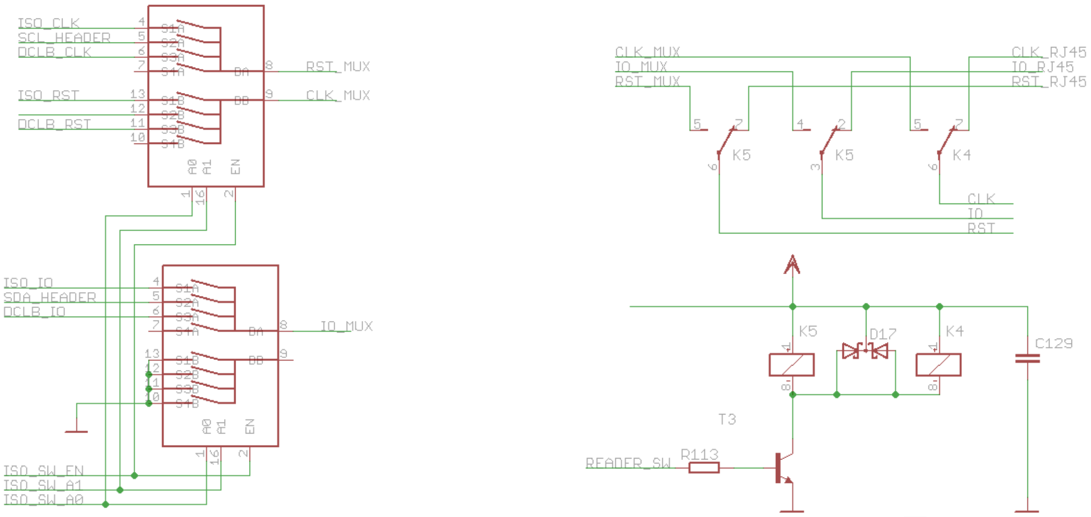


Figure 3.23: ISOPAD source switching

The supported DUTs features the possibility to change the mapping of the ISOPAD-pins. This pins can be mapped with the ISO 7816 interface, the DCLB interface or the I2C interface. Analog multiplexers (IC17 and IC18; ADG3109) and two relays (K4 and K5) to do this job have been added. The relays switch between the internal ISO 7816-Reader and an external ISO 7816-Reader. The multiplexers switch between the different buses. Analog switches were used in order not to influence the shape of the signals.



(a) Analog switches for the interfaces (Schematic: Chameleon_V15, Sheet 5, D5-E5)

(b) Switch between internal and external source (Schematic: Chameleon_V15, Sheet 5, D3-E6)

Figure 3.24: Schematic of the ISOPAD source switching

## 3.7 DCLB

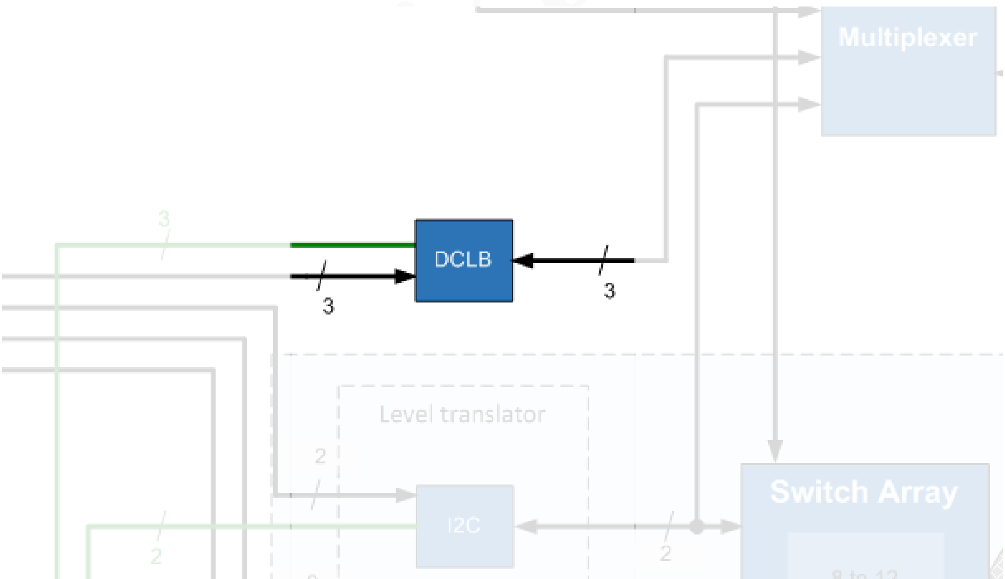


Figure 3.25: DCLB

To save output pins at the logic-analyzer, open-drain buffers are used. In this case, the control line for controlling the data direction can be skipped. This leads to two disadvantages.

1. Data-rates depending on the size of the pull-up resistor. The shapes of the signal will be influenced by the low pass filter which is built with the resistor and the parasitic cap of the line.
2. The outputs of the DUT have to sink more current while generating a low.

Currently, there are no problems detected in contract to these disadvantages.

The pull-ups yield also an advantage: A level translation is done very easily, just by connecting one terminal of the pull up resistor to VDDPISO.

## 3.8 ISO 7816 Reader

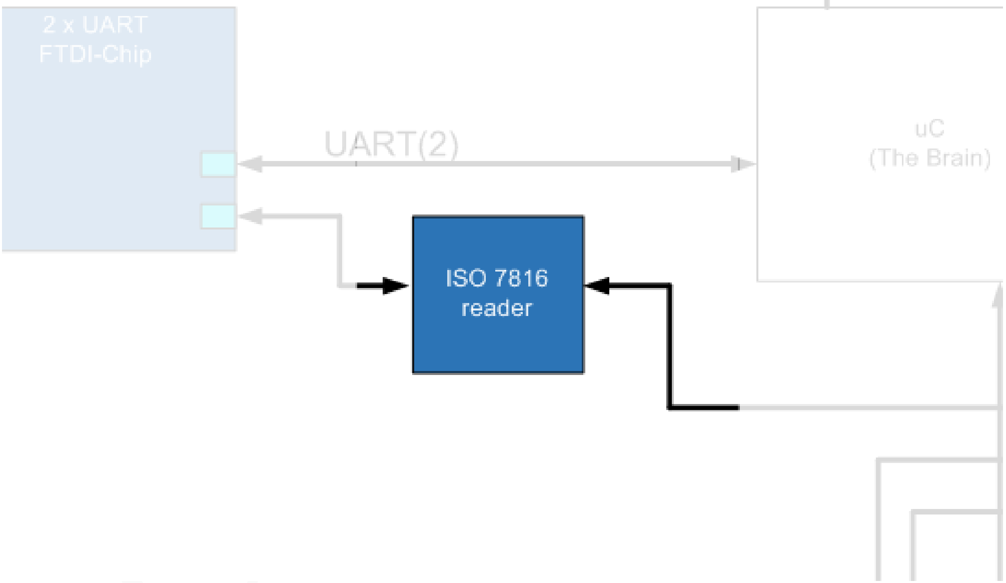


Figure 3.26: ISO 7816 Reader

Figure 3.27 shows the schematic of the on-board ISO 7816 reader.

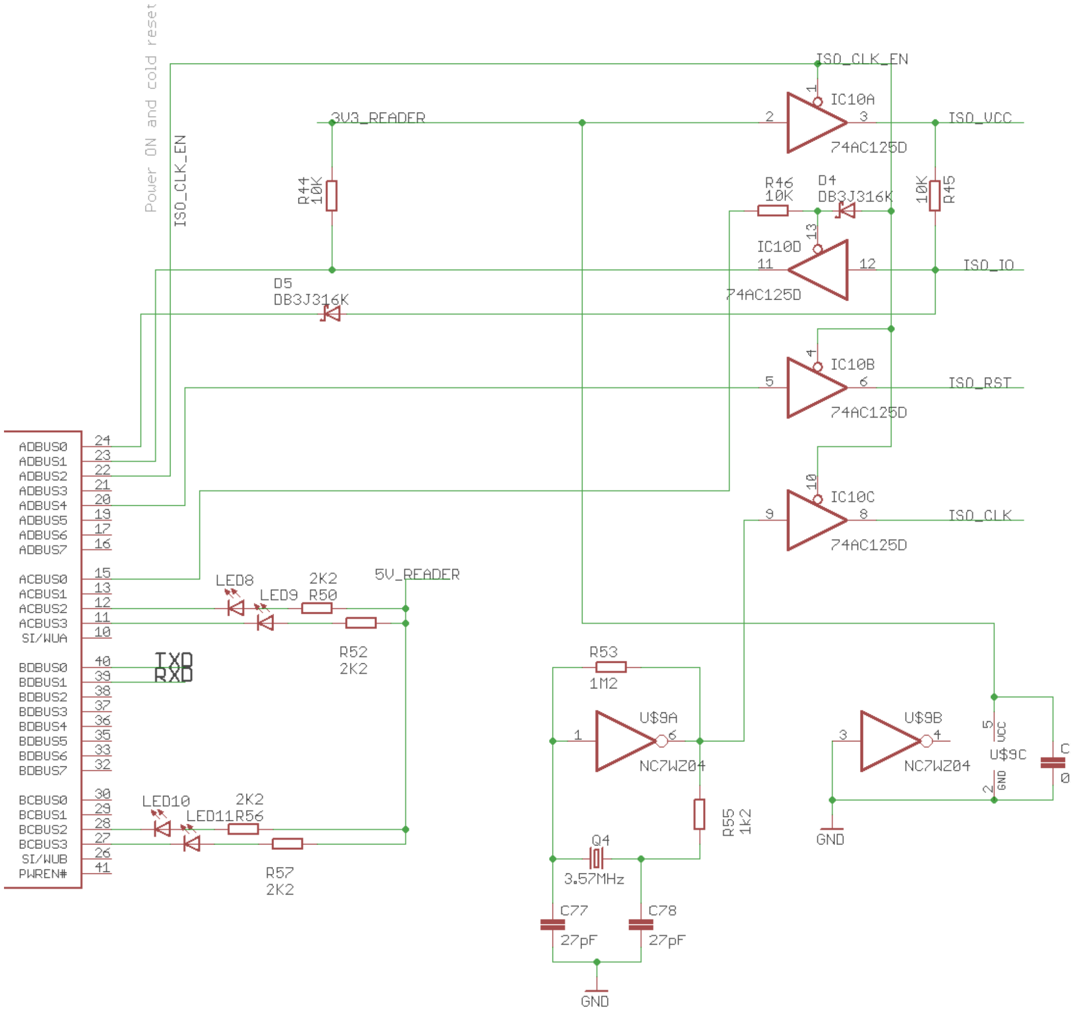


Figure 3.27: Schematic of the ISO 7816 reader

U$9A and the crystal build an oscillator which generates the clock signal. As described in the theory part of the ISO 7816 Interface, the clock frequency has to be 372 times higher than the data-rate on the IO-Line. The standard baudrate is 9600 Baud. So the best clock frequency would be:

$$f_{clock\_best} = Baudrate \cdot 372 = 9600 \text{ Baud} \cdot 372 = 3571200 \text{ Hz} = 3.5712 \text{ MHz} \tag{3.5}$$

This crystal frequency is not available, so the closest value has to be chosen. The closest

| | | |
|---|---|---|
| ADBUS0 | TXD | Transmit Data |
| ADBUS1 | RXD | Receive Data |
| ADBUS2 | CTS# | Clear To Send |
| ADBUS4 | DTR# | Data Terminal Ready |
| ACBUS0 | TXDEN | Transmit enable |
| ACBUS2 | RXLED# | receive status LED |
| ACBUS3 | TXLED# | transmit status LED |

Table 3.2: Pin assignment of the FT2232D

available value is 3.579 MHz. This leads to a negligible mismatch of 0.2%.

$$Mismatch = \left(1 - \frac{3.5712\text{ MHz}}{3.5712\text{ MHz}}\right) \cdot 100\ \% \approx 0.2\% \tag{3.6}$$

D5, R45 and IC10D split the bidirectional ISO_IO line in an input (RXD) and an output (TXD) line. If TXD is ”high”, the diode is always reverse biased and the ISO_IO state can be changed by the DUT. If the TXD line is ”low”, the diode is conductive and pulls the ISO_IO line to ”low”. R46, D4 and IC10D are used to pass the data from the DUT. When the DUT is allowed to send data, TXDEN and CTS are ”low” and the enable pin of IC10D is held at ”low” and enables this buffer. While the reader sends data to the DUT, TXDEN is high and pulls the enable pin of IC10D to high. This disables the buffer and no data can pass IC10D. In this state, R44 held RXD at ”high”.

Standard settings for the communication interface:

| Baudrate | Databits | Stoppbits | Parity | Flow Control |
|---|---|---|---|---|
| 9600 | 8 | 2 | even | none |

Note: It is not possible to tell exactly the right COM-Port, because it depends on the installation of the driver but it is the lower number of the two newly installed COM-Ports.

## 3.9 Headerboard

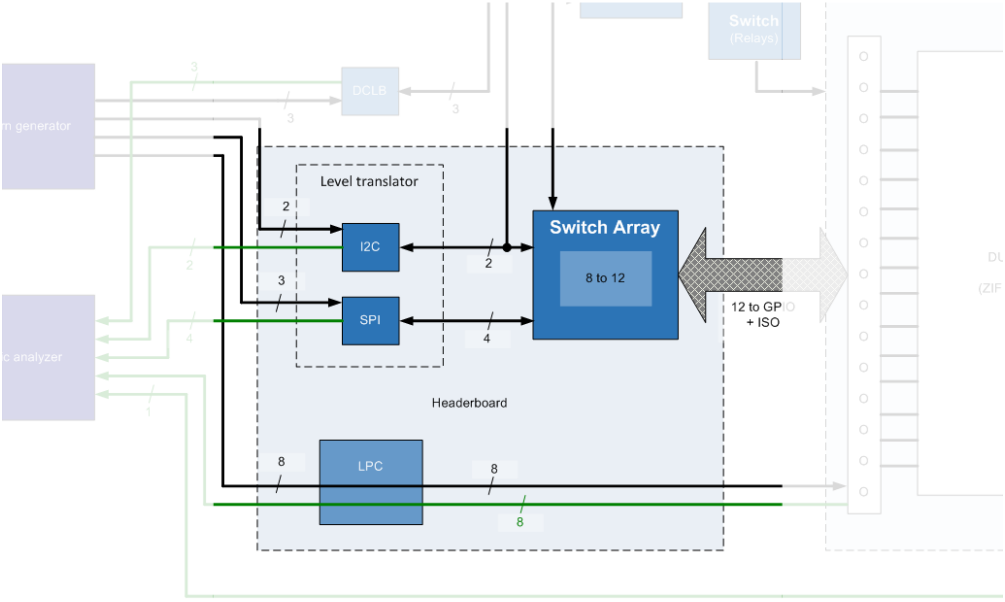


Figure 3.28: Headerboard

Sometimes it is necessary to change something in the signal path to generate other or new test-conditions. To fit this requirement, headerboards were used. This gives two major advantages. There is no need to redesign the entire board, if there are some changes in the signal lines and a board without level-translator can be used. The latter point adds a different advantage: The entire signal path can be tested very easily and influences generated by the level-translators are impossible.

Normally, the propagation delay is very bad for high speed circuits and much effort has to be spent to handle the timings. But this is not a big problem in this case. Since the modules of the DUT need to be tested at different timings a shmoo-test is used in order to change the delay time of the pattern. However, it has to be ensured that the data lines and the clock are synchronized to each other.

In some test cases, it is necessary to change the pin mapping of the DUT. To fit this requirement without having hundreds of headerboards with every possibility, a switch matrix has been added. This switch matrix (ADG2128[26]) can connect every input line (Y0 to Y7) to every output line (X0 to x11). The switch matrix has to be configured by the user using the SCPI commands otherwise the connections are left open. The configuration is not stored! So you have to reconfigure the matrix after you power down the board and power it up again.

| Input | Bus line |
|---|---|
| Y0 | SCL |
| Y1 | SDA |
| Y2 | MOSI |
| Y3 | SCLK |
| Y4 | #CS |
| Y5 | MISO |

| GPIO | Output |
|---|---|
| 0.0 | X6 |
| 0.1 | X7 |
| 0.2 | X8 |
| 0.3 | X9 |
| 0.4 | X10 |
| 0.5 | X11 |
| 0.6 | X0 |
| 0.7 | X1 |
| 1.0 | X2 |
| 1.1 | X3 |
| 1.2 | X4 |
| 1.3 | X5 |

Table 3.3: Pin assignment of the switch matrix

Both schematics, the schematic of the headerboard without the leveltranslator (title: NoLS_Headerboard_V1) and the schematic of the headerboard with the level-translator and the switch-matrix (title: Levelshifter_Headerboard_V5) can be found in the appendix.

### 3.9.1 I2C

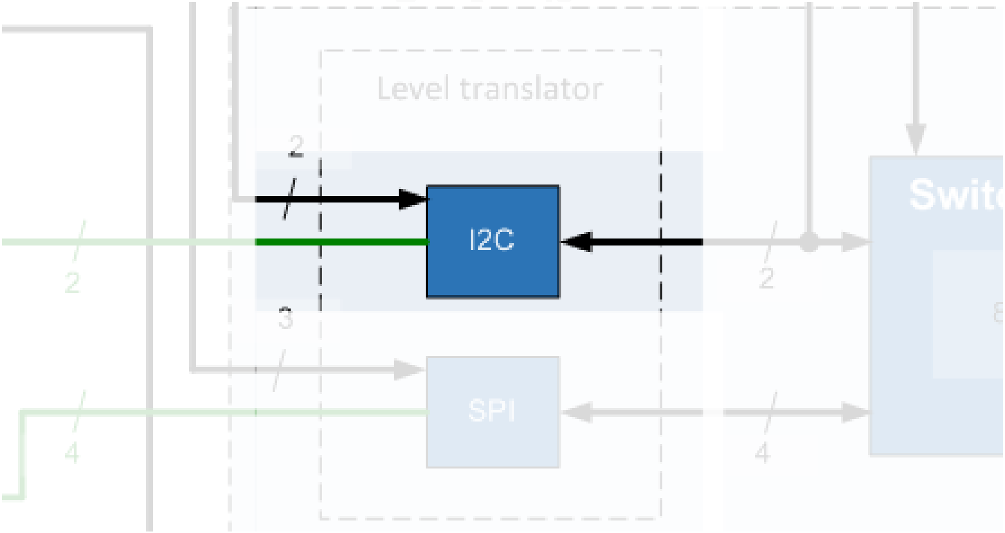


Figure 3.29: I2C Block

The internal I2C interface of the Byteparadigm provides a data rate up to 1 MHz. But this is too slow. A data rate of up to 5 MHz is required. To fit this requirement, we have to use normal outputs of the pattern generator. This leads to two additional necessary extensions. The push-pull stage has to be converted to an open-drain stage, because the output of the pattern generator is a push-pull configuration and we have to program the protocol on the bit-layer and cannot use an abstracted higher layer. The conversation from push-pull to open drain is done by the NC7WZ07 devices ( IC13A and IC13B; Schematic: Chameleon_V15, Sheet 5, A2-A3) on the mainboard (Chameleon).

You can see the schematic of the I2C-level-translator in 3.30.

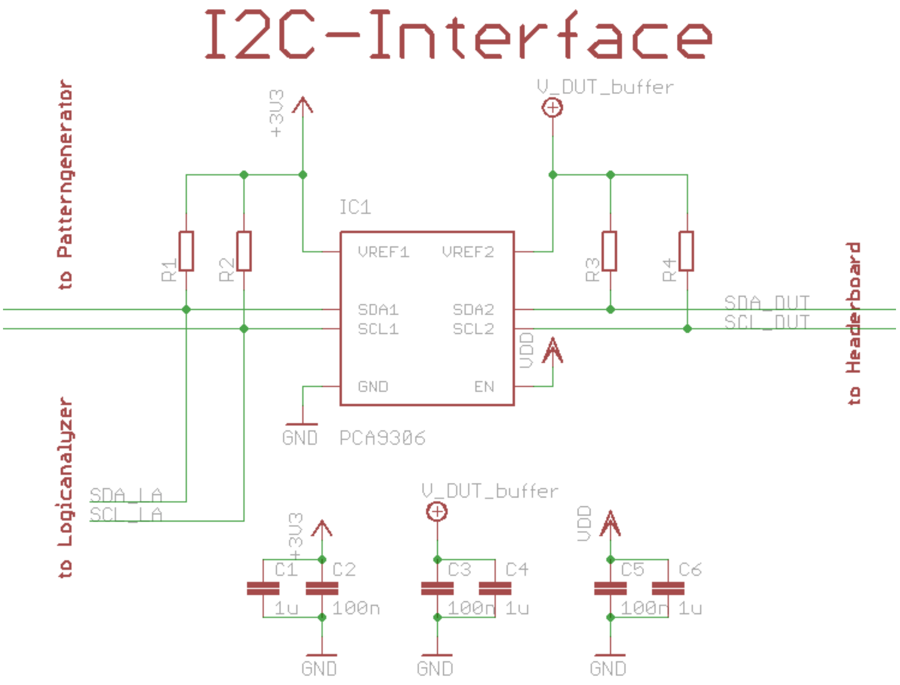


Figure 3.30: Level translator of the I2C interface

The PCA9306 is a dedicated device for leveltranslating of an I2C-Bus.

| Input DIR | Operation |
|---|---|
| Low | B data to A bus |
| High | A data to B bus |

Table 3.4: Data direction of the SN74LVC2T45

### 3.9.2 SPI

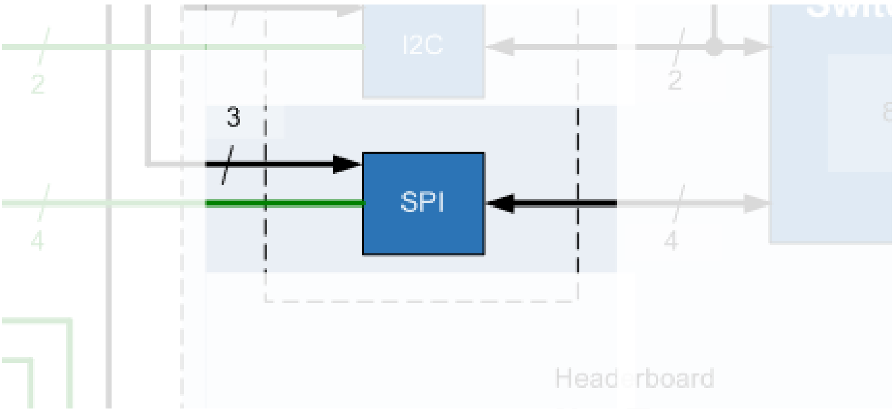


Figure 3.31: SPI Block

The maximum frequency of this interface is 25 MHz. So the buffer has to be capable to handle this high frequency. For the leveltranslation of the SPI-Interface, a SN74LVC2T45 device has been used. This device is a dual bit bus transceiver with two supplies. The data direction can be controlled by the DIR pin. If the DIR-Pin is high, the direction is from A to B. If DIR is low, the data goes from B to A.

You can see the schematic of the SPI-leveltranslator in figure 3.32.

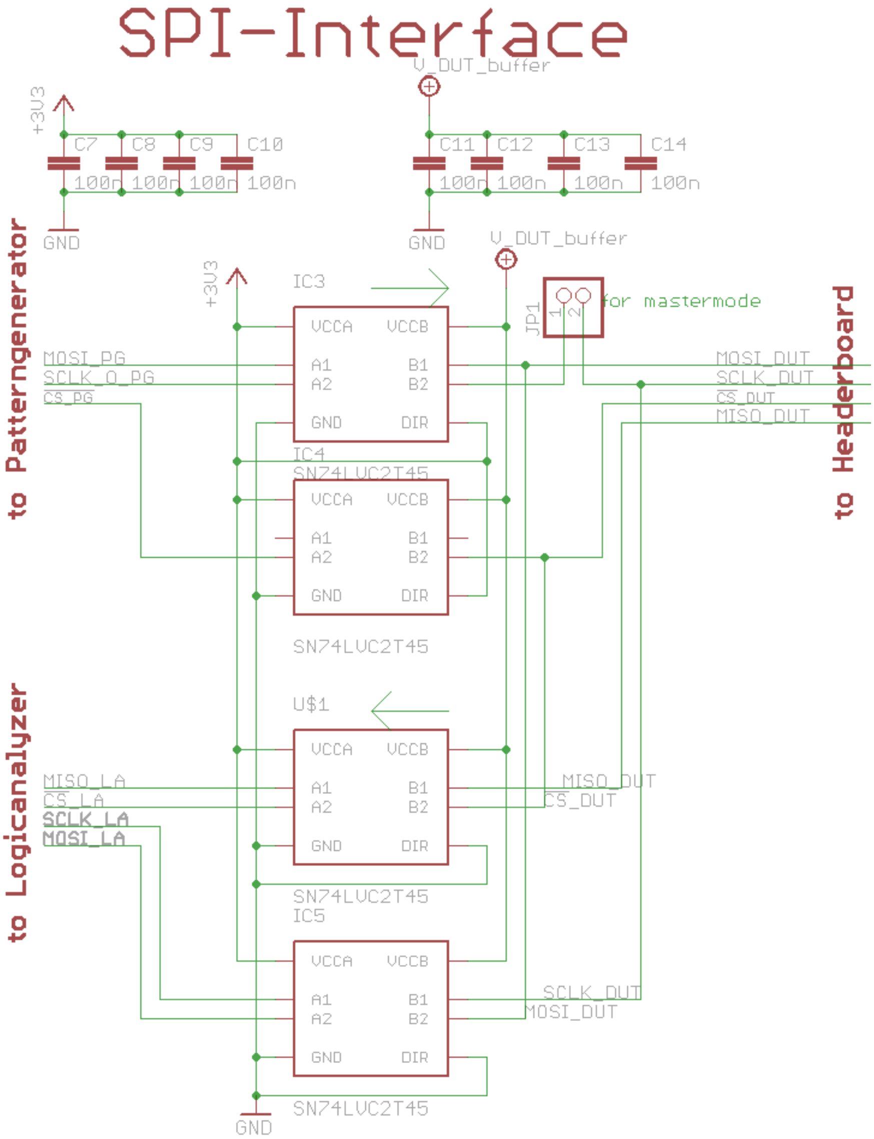


Figure 3.32: Leveltranslator of the SPI interface

| | | maximum propagation delay [ns] | | | |
|---|---|---|---|---|---|
| From | To | @1.8V | @2.5V | @3.3V | @5.5V |
| Pattern generator | DUT | 17.7 | 10.3 | 8.3 | 7.2 |
| DUT | Logic analyzer | 17.7 | 16.0 | 15.6 | 15.1 |

Table 3.5: The maximum propagation delay of the SN74LVC2T45

## 3.10 Logicanalyzer and Pattern generator

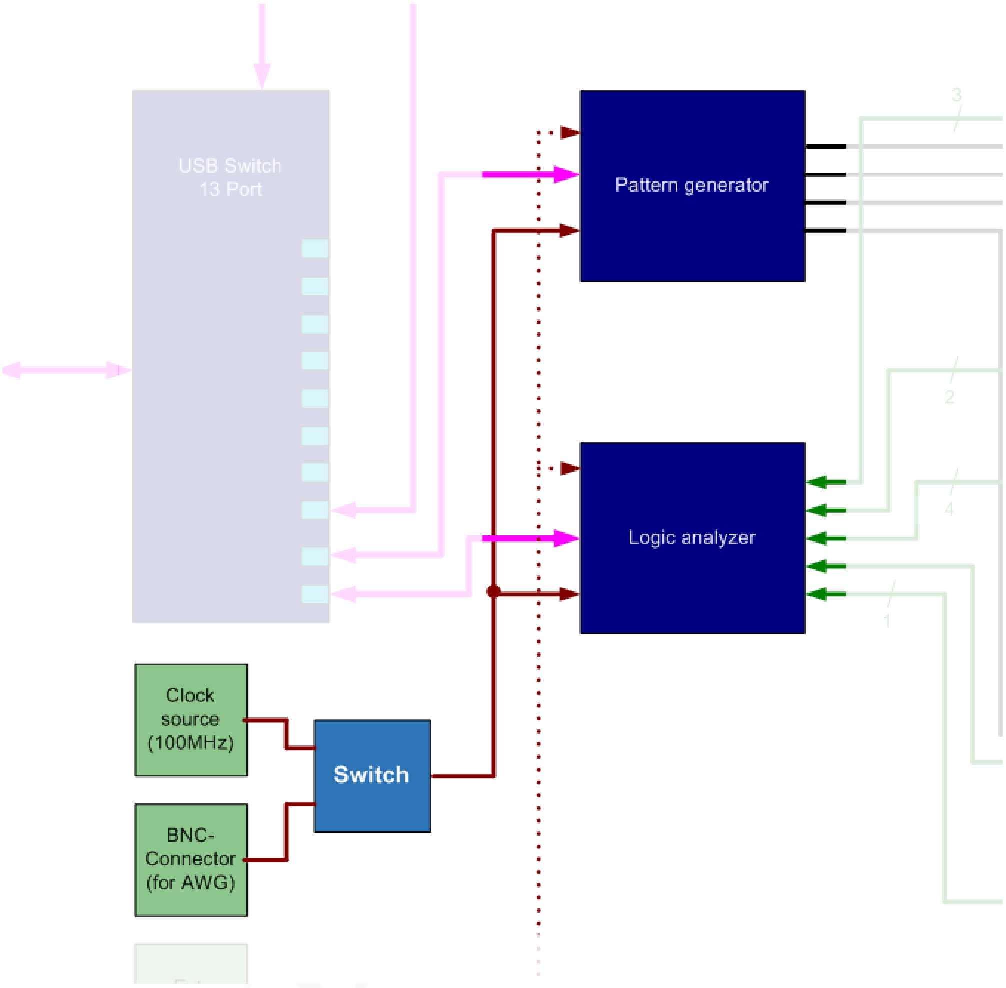
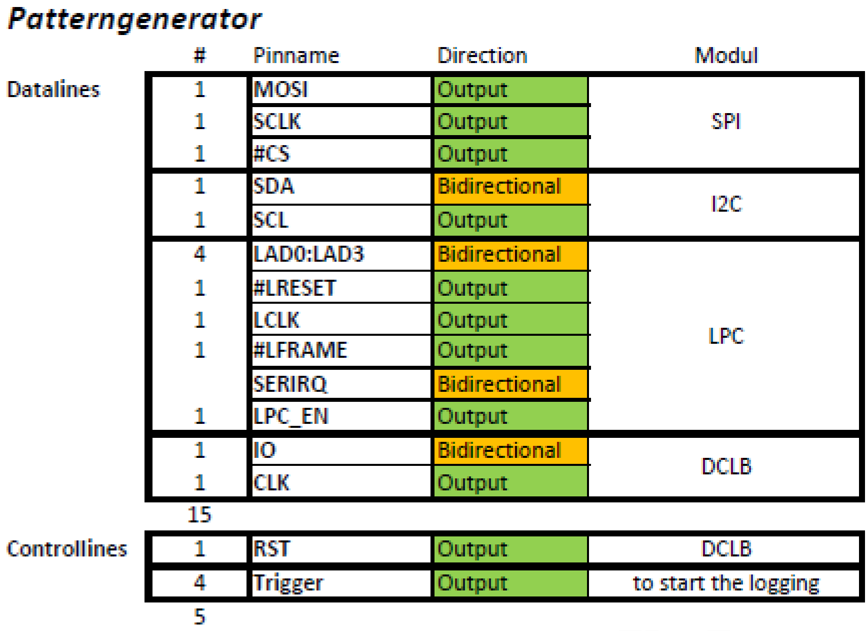


Figure 3.33: Logicanalyzer and Pattern generator

The test pattern for the different buses will be generated and logged by devices (GP-24132) from ByteParadigm. This device can be configured either as pattern generator or as logic analyzer. These devices have an internal memory of 32 MB and provide a sample rate of up to 100 MHz with the possibility to connect an external clock. This can be and will be used to synchronize these devices.

Figure 3.34 shows the lines which are connected to the pattern generator and their directions. The direction in this case means, the direction of the bus but not of the pattern generator because all pins of the pattern generator are outputs. The external devices convert the outputs to their required direction. These schematics can be found in the previous sections.

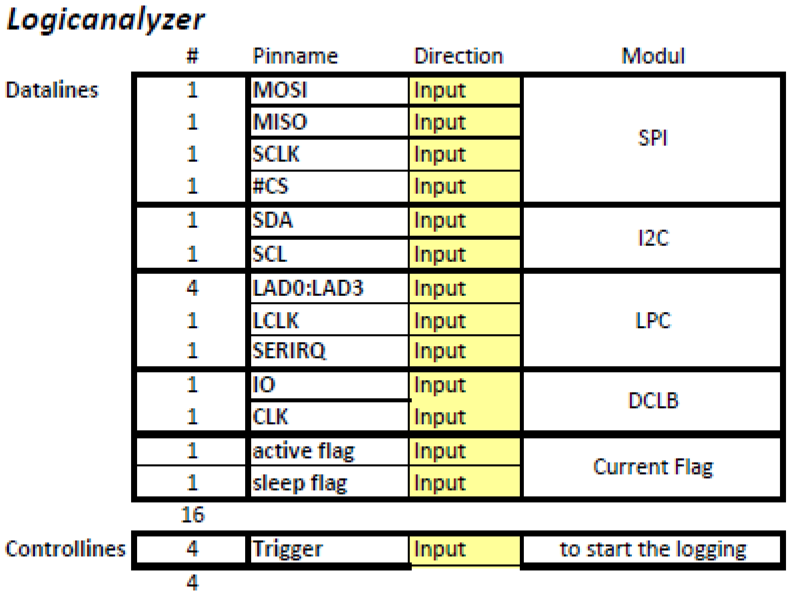


**Patterngenerator**

| | # | Pinname | Direction | Modul |
|---|---|---|---|---|
| Datalines | 1 | MOSI | Output | SPI |
| | 1 | SCLK | Output | |
| | 1 | #CS | Output | |
| | 1 | SDA | Bidirectional | I2C |
| | 1 | SCL | Output | |
| | 4 | LAD0:LAD3 | Bidirectional | LPC |
| | 1 | #LRESET | Output | |
| | 1 | LCLK | Output | |
| | 1 | #LFRAME | Output | |
| | | SERIRQ | Bidirectional | |
| | 1 | LPC_EN | Output | |
| | 1 | IO | Bidirectional | DCLB |
| | 1 | CLK | Output | |
| | 15 | | | |
| Controllines | 1 | RST | Output | DCLB |
| | 4 | Trigger | Output | to start the logging |
| | 5 | | | |

Figure 3.34: Directions of the lines at the pattern generator

Figure 3.35 shows the lines which are connected to the logic analyzer.



**Logicanalyzer**

| | # | Pinname | Direction | Modul |
|---|---|---|---|---|
| Datalines | 1 | MOSI | Input | SPI |
| | 1 | MISO | Input | |
| | 1 | SCLK | Input | |
| | 1 | #CS | Input | |
| | 1 | SDA | Input | I2C |
| | 1 | SCL | Input | |
| | 4 | LAD0:LAD3 | Input | LPC |
| | 1 | LCLK | Input | |
| | 1 | SERIRQ | Input | |
| | 1 | IO | Input | DCLB |
| | 1 | CLK | Input | |
| | 1 | active flag | Input | Current Flag |
| | 1 | sleep flag | Input | |
| | 16 | | | |
| Controllines | 4 | Trigger | Input | to start the logging |
| | 4 | | | |

Figure 3.35: Directions of the lines at the logic analyzer

### 3.10.1 Clock Input

The clock inputs of the logic analyzer and the pattern generator are connected together and are connected to a relay. One can choose between an 100 MHz on-board-clock or an external clock source. It has to be taken care, that the external clock does not exceed the limits of the input. The high state of the clock should be 3.3 V and the low state should be 0 V. The maximum input frequency is 100 MHz. Higher frequencies will most likely not work properly.

### 3.10.2 Trigger

The logging will be started by a trigger. This trigger event can be done internally (generated by the logic analyzer itself) or by a trigger pulse on a control line of the logic analyzer.

The first idea was to use a control line to generate this trigger event. But after some tests we found out, that it is not possible to generate a reliable trigger event on this line. The state of this line can be set to a defined constant state, but the problem is, that the state is unknown before configuring the device. So it is not possible to create a repeatable trigger edge.

However two solutions were implemented. It is possible to generate a continuous pattern on the control lines. But this pattern will fit in one cycle of samples of the normal output. This means, that the usable sample rate will be divided by two. If you do a simple calculation, you will see, that it is not possible to generate a test pattern for example for SPI.

$$\frac{100\text{ MSPS}}{2} = 50\text{ MSPS} \tag{3.7}$$

Equation 3.7 shows the usable sample rate for the data lines. If we want to have an oversampling of 4, then we get the maximum clock frequency by:

$$\frac{50\text{ MSPS}}{4} = 12.5\text{ MSPS} \rightarrow 6.75\text{ MHz} \tag{3.8}$$

To fix this problem we found another solution. We just use one data line of the LPC-module. The data lines of the LPC-module can be set transparent, so the DUT does not see a change in the state if the output state of the pattern generator is changed. So the LPC buffer driver has to be disabled, which generates the trigger pulse. Then the test pattern can be started. Once a trigger occurred and the logic analyzer was started, it will continue to sample data until the buffer is full. If an additional trigger pulse happens, while the logic analyzer is sampling, this pulse has no influence.

One disadvantage of this workaround is, that a trigger event happens at every time you start a LPC communication. This is only important, if you do an LPC communication,

wait for a long time and then start the test pattern you want to log and verify. But you can fix this problem, by using an offset of the trigger. The logic analyzer gives you the possibility to set a sample offset. It waits the amount of samples until the trigger occurred and then starts to log the data. The second disadvantage is, that at least three additional samples to generate the trigger pulse are needed.
The three samples are:

- setting LAD0 and LPC-Enable to low
- setting LAD0 to high
- setting LAD0 to low

# 4 Firmware Development

*Note: For developing the hardware, i used the following sources: [33] [34].*

The firmware on the uC is used to control and configure the whole board. It is written in C. The used Integrated Development Environment (IDE) is MPLAB® in combination with the C18 Compiler. Both programs are provided by Microchip.

The UART interface of the uC is used for the connection to the FTDI chip.

At start up the controller is in state ”99”, all ports and all internal modules of the uC are configured. Then the uC enables the different supplies one by one. The interrupts for the UART interface and the timer will be enabled and the timers will be configured. This timer is used as watchdog for the input buffer to empty the buffer after a certain time to ensure a clean state after a wrong or a broken command was received. After this configuration phase, the calibration values for the on-board voltage source will be loaded from the Electrically Erasable Programmable Read-Only Memory (EEPROM) and allocated.

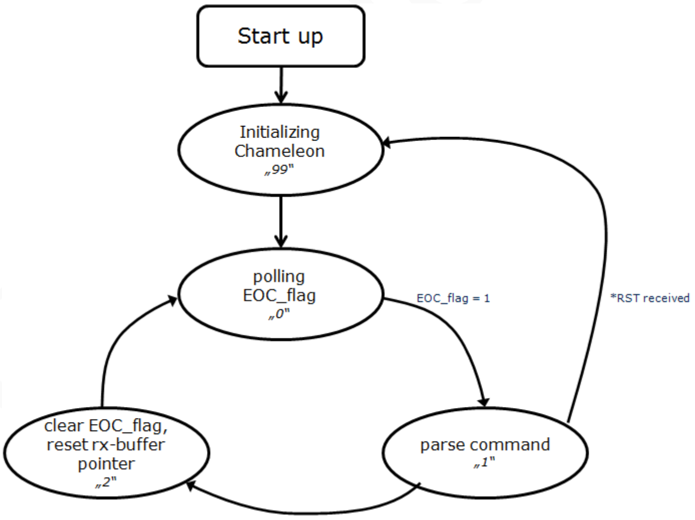

Start up
Initializing Chameleon „99"
polling EOC_flag „0"
EOC_flag = 1
*RST received
clear EOC_flag, reset rx-buffer pointer „2"
parse command „1"

Figure 4.1: The state-machine of chameleon

At this point, the uC waits to receive a command.

To indicate that the controller is still running, it toggles a Light Emitting Diode (LED) (LED16 on schematic).

If a command was received, the state of LED17 toggles and the command will be interpreted by the SCPI-Parser. The controller accepts standard SCPI commands[1] and some additional SCPI commands. Not every command, which is required for the standard, is implemented. Particularly, the query capability of all commands is not implemented. A command ends with a "\n". If this sign is received, a flag (EOC_flag; End-Of-Command flag) will be set. The main state-machine polles this flag. If it is set, the uC changes into state "1" and starts to parse the command and execute it. Once this is done, state "2" resets the pointer/counter (uart_rxbuffer_counter) and clears the received flag (EOC_flag).

## 4.1 SCPI-Parser Implementation

The parser consists of two main functions: One function compares a single character, the other function compares a string. The second function calls the first function in a loop. So, comparing a string is comparing character by character. A string-compare-function is provided by most compilers, but it cannot be used in this case because the SCPI-standard allows capitalized and uncapitalized letters. The routine tests both writing variants in a single call.

```
char parse(char ch)
{
        char t = *cmd;  // get token

        if (t == ch ||
                (t >= 65 && t <= 90 && (t+32) == ch) ||  // test capitalized letter with uncapitalized letter
                (t >= 97 && t <= 122 && (t-32) == ch))   // test uncapitalized letter with capitalized letter
        {
                cmd++;  // move to next letter
                return 1;
        }
        else
                return 0;
}
```

Listing 4.1: Parsing a character

"*cmd" is a pointer which points to the actual position of a character in the received command. "ch" is the character which is compared with the actual character of the command. The parameter can be a capitalized or a low case letter and so we have to test for both variants.

```
(t >= 65 && t <= 90 && (t+32) == ch)
```

[1] The commands which can be found on the homepage of IVI FOUNDATION.

in this case "ch" is a capitalized character and the character of the command is lower case.

```
t >= 97 \&\& t <= 122 \&\& (t-32) == ch
```

in this case "ch" is a lower case character and the character of the command is capitalized. Note: Please refer to the ASCII-Table for easier understanding.

If a character matches, the pointer is incremented and the next position will be parsed.

```
char parseWord(far rom char *word_orig)
{
        char* cmd_backup = cmd;
        char* word = (char*)parseWordString;

        strcpypgm2ram(parseWordString,word_orig);
        while (*word != '\0')
        {
                if (!parse(*word))
                {
                        cmd = cmd_backup;
                        return 0;
                }
                word++;
        }
        return 1;
}
```

Listing 4.2: Parsing a word

The pointer of the receive buffer points to the first character of the received command. This character will be compared to the first allowed symbol i.e. ":" for user defined commands or "*" for common commands of the deposited command list. If there is no matching, the command will be compared to "IDN?\n". If there is a match, the identification will be sent back to the computer otherwise nothing happens.

```
void parseCmd(char* tmp)
{
        cmd = tmp;
        // detect first letter and deside if it's a UserCommand or an CommmonCommand
        if (parse(':'))
                parseUserCmd();
        if (parse('*'))
                parseCommonCmd();
        if (parseWord((far rom char*)"IDN?\n"))
                send_IDN();
}
```

Listing 4.3: Parsing the first character

If the first symbol matches, the next possible part of a command will be compared as long as the whole command is compared. After a successful decoding of a command, the appropriate function will be executed.

For example:

> *We want to enable the on-board ISO7816 reader and send:*
> `:ISOP:ISOR ON\n`

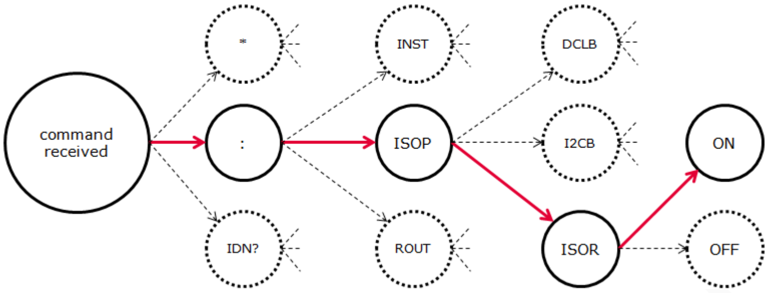


Figure 4.2: Decoding the example

```
        ...
        if (parse(':'))
                parseUserCmd();
        ...
                if (parseWord((far rom char*)"ISOP"))
                        parseISOPad();
                ...

                        if (parseWord((far rom char*)":ISOR"))
                        {
                                parseWord((far rom char*)"eader");

                                if (parseWord((far rom char*)"␣ON\n"))
                                        select_ISO_PAD_function(1); // Set
                                            Mux to ISO-Reader
                                ...
                        }
                        ...
```

Listing 4.4: Decoding the example

It is important to parse also the unimportant parts of a command ("eader" in our example), otherwise the parser will not be able to successfully decode a command, because it will interpret the useless part as important part and will not find a match.

## 4.2 Commands

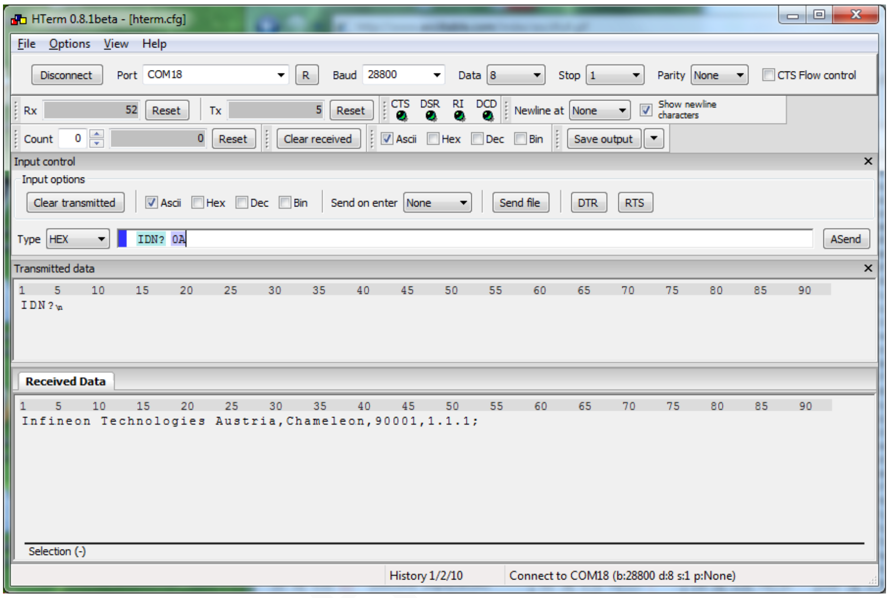
For proper work of the commands just send them as plain text (ASCII String) to the chameleon. The commands have to be terminated with a "\n" , otherwise the end of a command will not be recognized.

It is not allowed to send more commands in series and then terminate them. Every command has to be sent separately.

All capitalized letters have to be used. The uncapitalized letters are optional and can be used in order to make the code more readable.

The commands are not case sensitive.

It is important to use a point (.) as decimal point and not a comma (,) because a comma separates the elements of a list.



HTerm 0.8.1beta - [hterm.cfg]

Port COM18 Baud 28800 Data 8 Stop 1 Parity None □ CTS Flow control

Rx 52 Tx 5 Newline at None ☑ Show newline characters

Count 0 0 ☑ Ascii □ Hex □ Dec □ Bin

Input control

Input options

☑ Ascii □ Hex □ Dec □ Bin Send on enter None

Type HEX IDN? 0A

Transmitted data

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90
IDN?\n
```

Received Data

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90
Infineon Technologies Austria,Chameleon,90001,1.1.1;
```

Selection (-)

History 1/2/10 Connect to COM18 (b:28800 d:8 s:1 p:None)

Figure 4.3: Example for sending an IND?-Query

## 4.2.1 Voltage Source

Selecting the Output :

:INSTrument:NSELect $<$ *channel_number* $>$

Channel_number options:

| | | Voltage Source | Current Limit |
|---|---|---|---|
| < *channel_number* > | 1 | VDDP | current limiter |
| | 2 | VDDPISO | active current |
| | 3 | VDDGPIO | sleep current |
| | 4 | VDDLPC | |

*Example:*

```
:INST:NSEL 3\n
```

Setting the output voltage (the right channel has to be selected before setting the voltage):

:SOURce:VOLTage:LEVEl:IMMediate AMPLitude < *numeric_value* >

*Example:*

```
:SOUR:VOLT:LEVEl:IMM AMPL 3.3\n
```

Setting the current limits:

:SOURce:CURRent:LEVEl:IMMediate AMPLitude < *numeric_value* >

In order to set the current limit to 50 mA, just send the command:

*Example:*

```
:SOUR:CURR:LEVEl:IMM AMPL 50.0\n
```

### 4.2.2 Switch Matrix

To route the signales use the following command :

:ROUTe:CLOSe < *channel_list* >

Channel_list-Example:

< *channel_list* > (@X1Y1)
(@X1Y1,X2Y3)

| Input | Bus line |
|---|---|
| Y0 | SCL |
| Y1 | SDA |
| Y2 | Master Out Slave In (MOSI) |
| Y3 | Serial Clock (SCLK) |
| Y4 | #CS |
| Y5 | Master In Slave Out (MISO) |

| GPIO | Output |
|---|---|
| 0.0 | X6 |
| 0.1 | X7 |
| 0.2 | X8 |
| 0.3 | X9 |
| 0.4 | X10 |
| 0.5 | X11 |
| 0.6 | X0 |
| 0.7 | X1 |
| 1.0 | X2 |
| 1.1 | X3 |
| 1.2 | X4 |
| 1.3 | X5 |

*Example:*

```
:ROUT:CLOS (@X1Y1)\n
```

To open all switches (this command resets the switch matrix and opens all connections):

:ROUTe:OPEN:ALL

### 4.2.3 Clock selecting

To select between the internal and the external clock for the Byteparadigm-devices, this command is used. The internal clock is 100 MHz. The maximum frequency for the external clock is 100 MHz. Higher frequencies will not work.

*Note: The recommended conditons for the high state of the clock signal is 3.3 V and should not be exceeded. For details, please refer to the datasheet of the GP24132.*

:SOURce:CLOCk:INTErnal $< ON, OFF >$

*Example:*

```
:SOUR:CLOC:INTE ON\n
```

### 4.2.4 ISO-Pad source selecting

The following commands allow switching the connection of the ISO-Pads of the DUT to different sources:

:ISOPad :DCLB $< ON, OFF >$
:I2CBus $< ON, OFF >$
:ISOReader $< ON, OFF >$
:EXTErn $< ON, OFF >$
:NONE $< ON, OFF >$

To connect the internal ISO7816 reader to the ISOPADs of the DUT, use:

*Example:*

```
:ISOP:ISOR ON\n
```

## 4.3 Control Software

As previously described, the chameleon can be controlled by SCPI commands. Therefore every tool which can send such commands can be used to control the chameleon.

But we use a self made tool for our tests. It is structured in different layers.

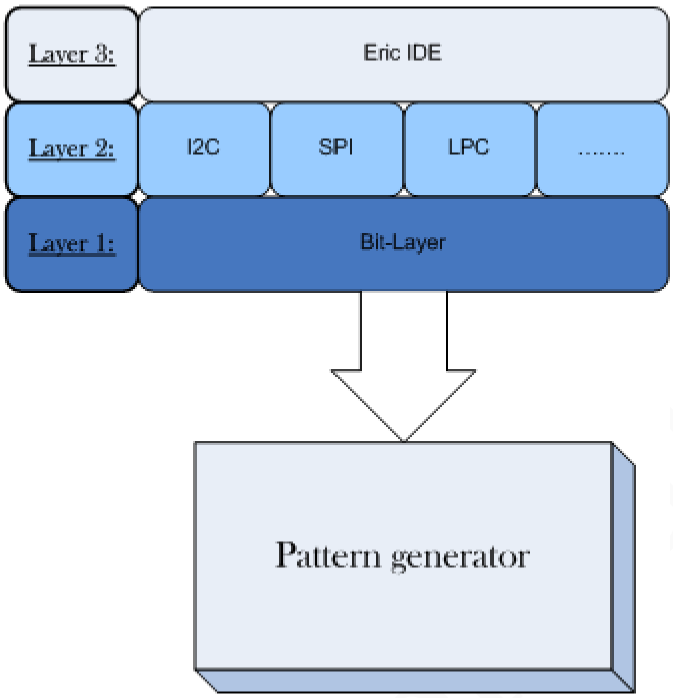
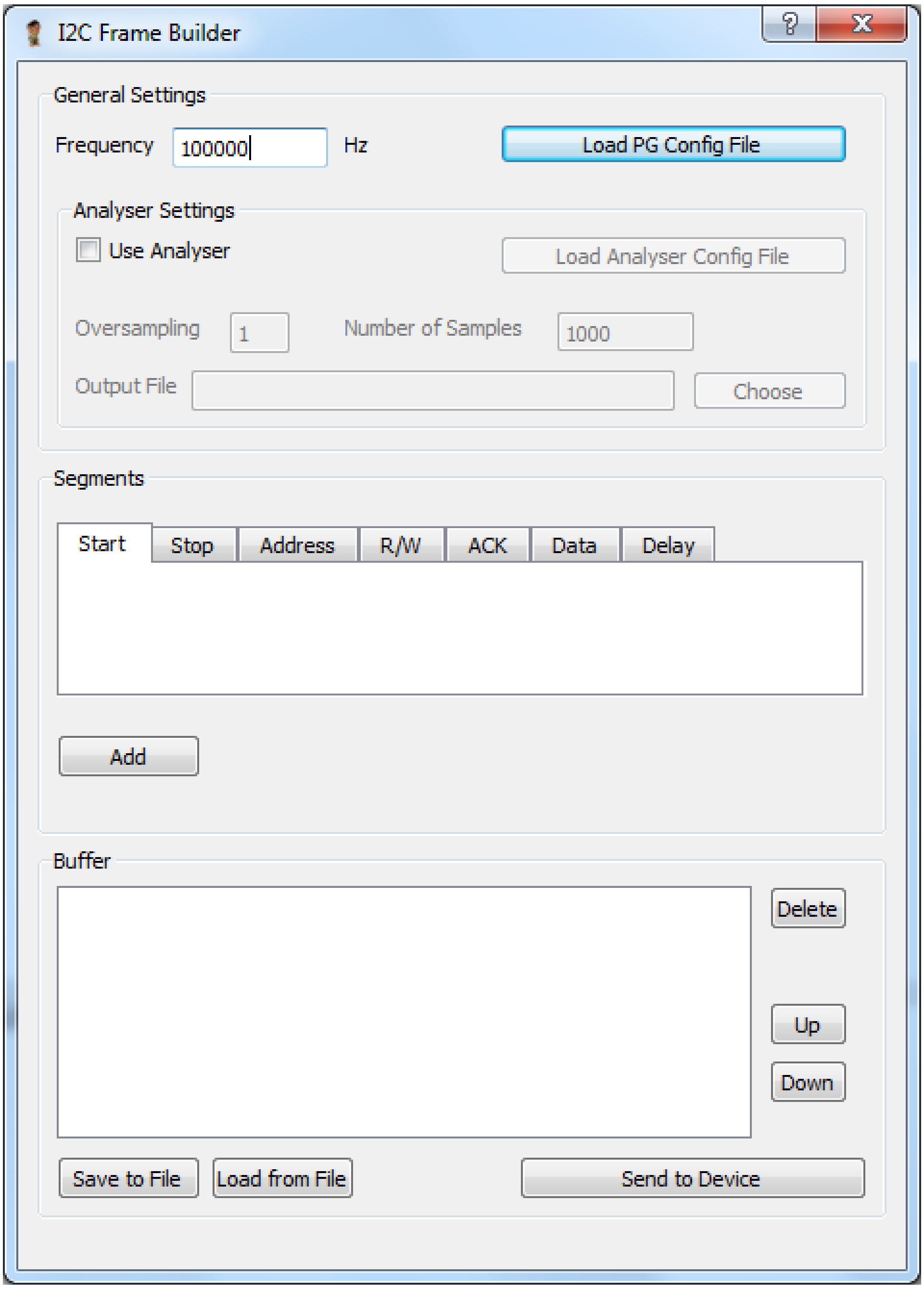


Figure 4.4: Layers of the software

Layer 1: Bit-Layer: In this layer, the bits are set directly. Every single sample of the pattern can be set.

Layer 2: This layer generates a test pattern for the selected interface. For example: Start-condition, Stop-condition, Address-Byte, Read, Write and Data-Bytes can be generated very easily by using the appropriate command.

Layer 3: GUI-Interface: A simple and intuitive Graphical User Interface (GUI) can be used to set up an entire test. Test pattern can be generated, voltages can be set or a test-shmoo can be started. In this layer, also other devices like external power supplies (i.e. as supply voltages) or function generators (i.e. as clock source) can be controlled.



I2C Frame Builder

General Settings

Frequency 100000 Hz

Analyser Settings

☐ Use Analyser

Oversampling 1

Number of Samples 1000

Output File

Figure 4.5: Screenshoot of the GUI

These layers are realized in a Python environment[2] and all layers can be used. This gives the possibility on one hand to generate very easily test pattern on an abstract

[2] Eric is an IDE and editor for Python

layer and on the other hand it is possible to generate individual test pattern by setting every single sample.

The analysis of the data is more difficult. A script downloads the data from the logic analyzer and starts to analyze the sample stream sample by sample. It decodes the stream depending on the test case, stores the data in a file and displays the results in GTKWave[3].

[3] http://gtkwave.sourceforge.net/

# 5 Conclusion and Outlook

As can be seen in the next pictures, the Chameleon board and all headerboards were built successfully. There were some problems recognized but some of them could be fixed by small workarounds.

For the new power supply (OTA + voltage follower), a small test board was built and the supply was tested successfully. It is stable over the whole operation region and also at load jumps. The performance of the current sensing path meets all requirements and gives us better results than expected.

The firmware on the uC works also properly and no bugs could be recognized at the current firmware version (1.0.2). There were some problems with the internal I2C interface of the PIC18F-Controller which are, due to hardware limitations at the current design revision and lack of workaround possibilities, not resolvable at the moment. Therefor a suitable software workaround was made and a software I2C-protocol was implemented.

Currently, tests with the entire system are in preparation. Very simple I2C communication tests are possible (sending and receiving data). The control software on the computer is currently not able to do larger tests and it needs more time to build the control software to handle all remaining features of Chameleon.

## 5.1 Pictures

The following pictures will show some aspects of the finished hardware.

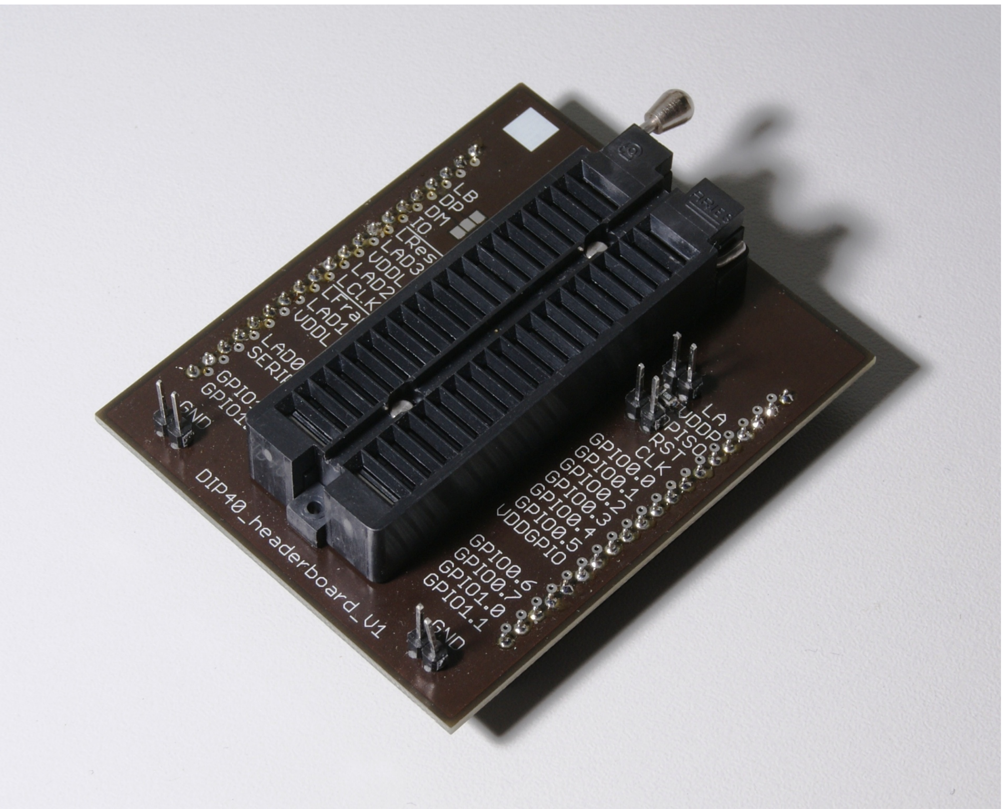


Figure 5.1: Picture of a headerboard for a DUT

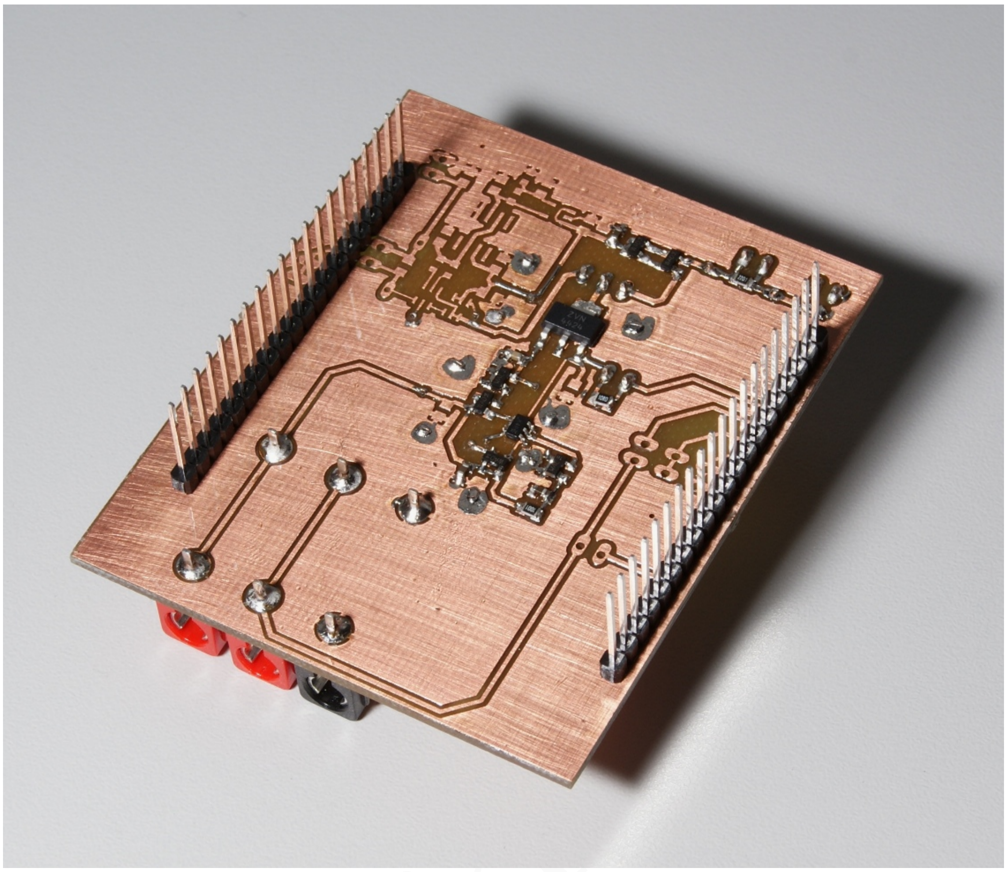


Figure 5.2: Picture of the OTA + current mirror testboard

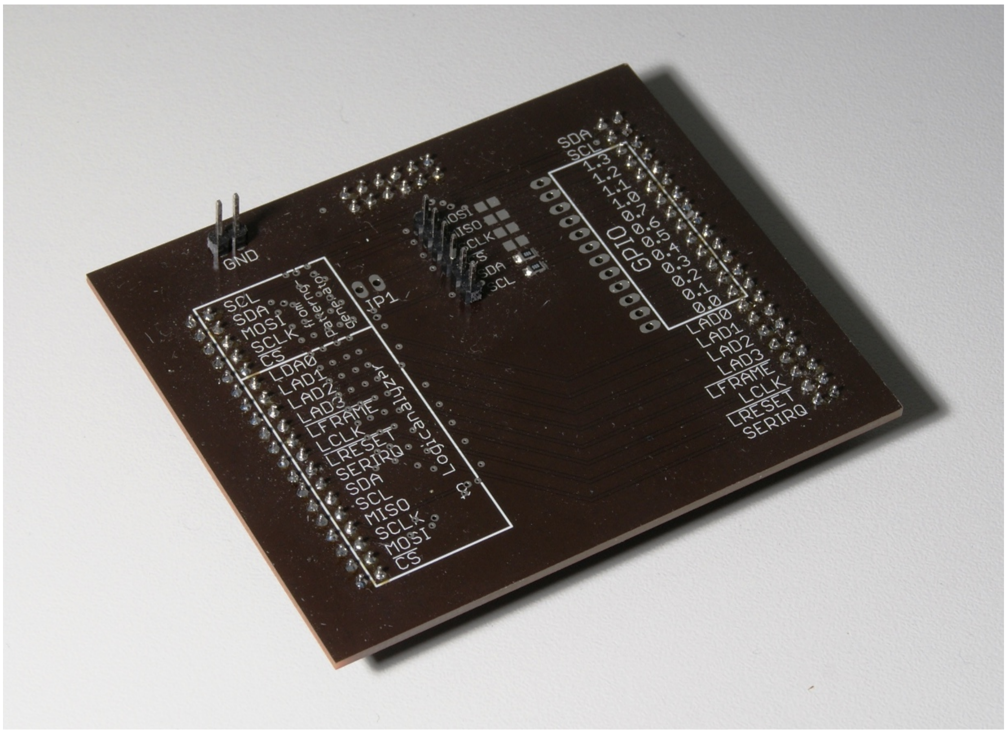


Figure 5.3: Picture of a headerboard without levelshifter

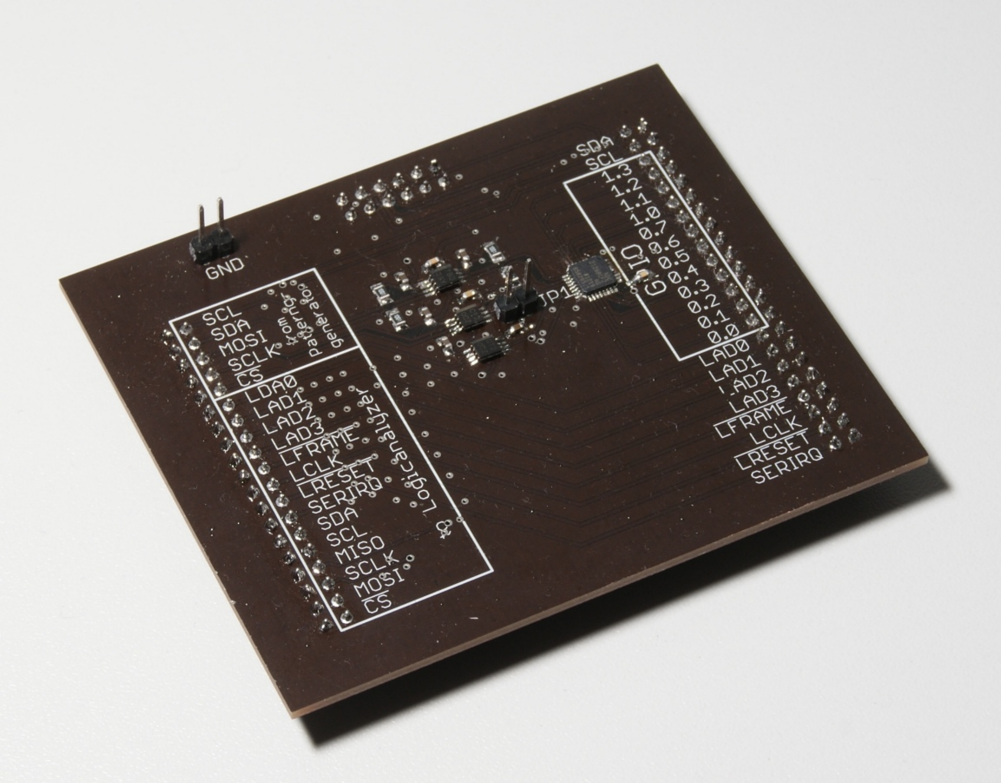


Figure 5.4: Picture of a headerboard with levelshifter

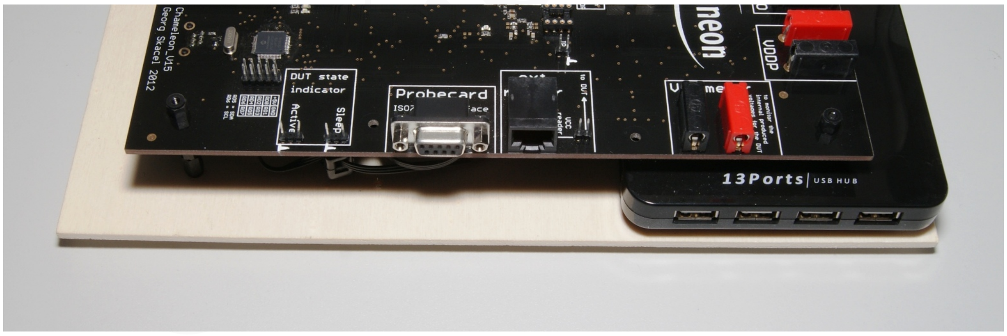


Figure 5.5: Picture of connectors for probecard and external reader

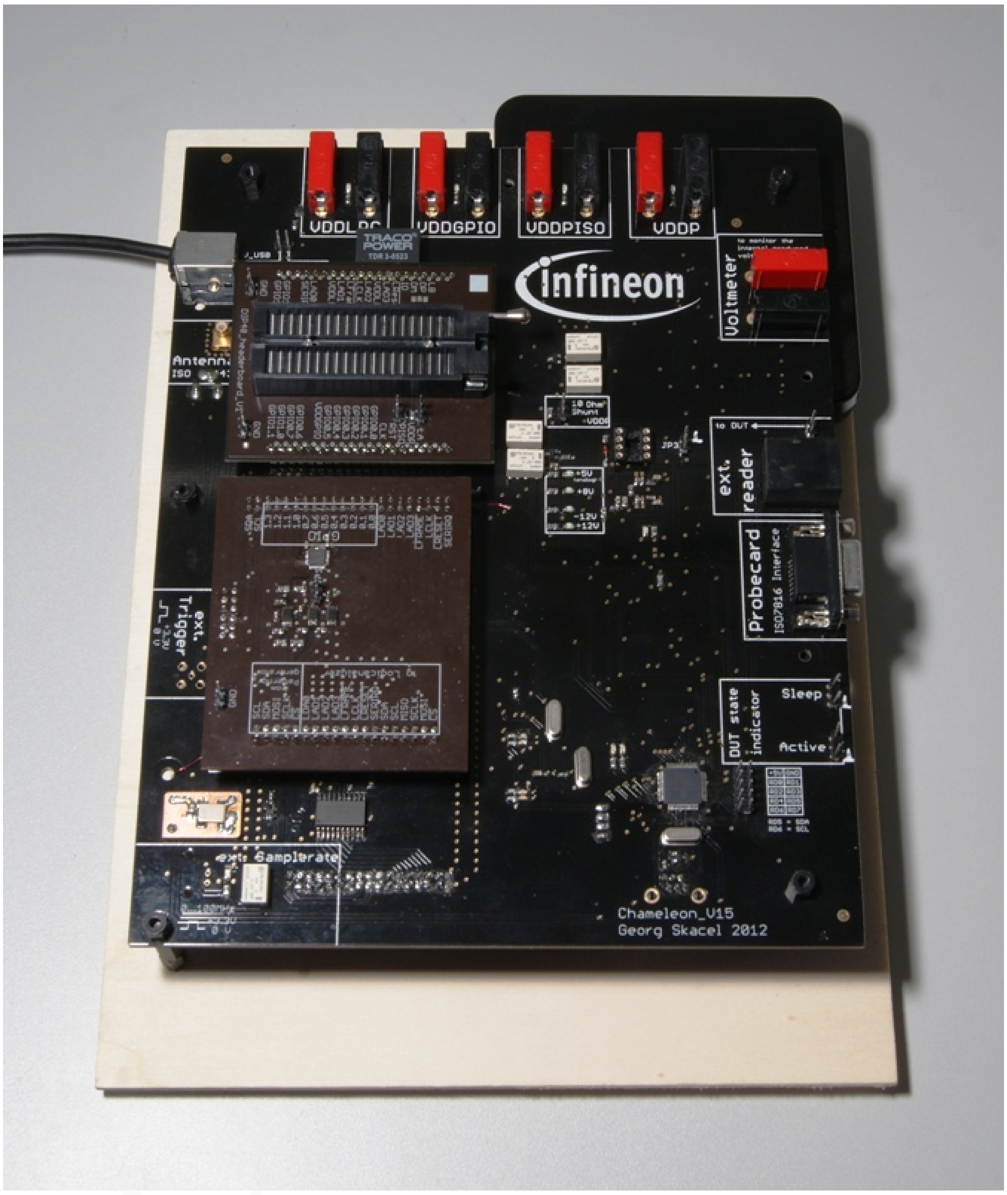


Figure 5.6: Picture of the Chameleon with headerboards

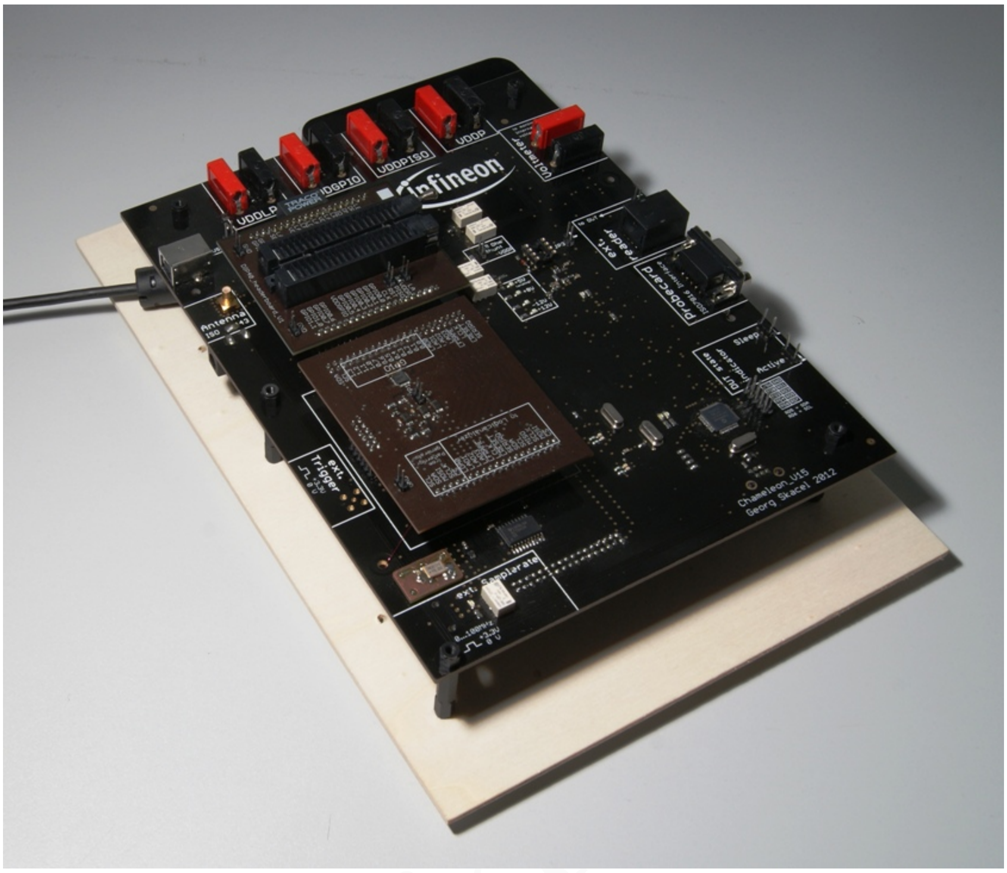


Figure 5.7: Picture of the Chameleon with headerboards

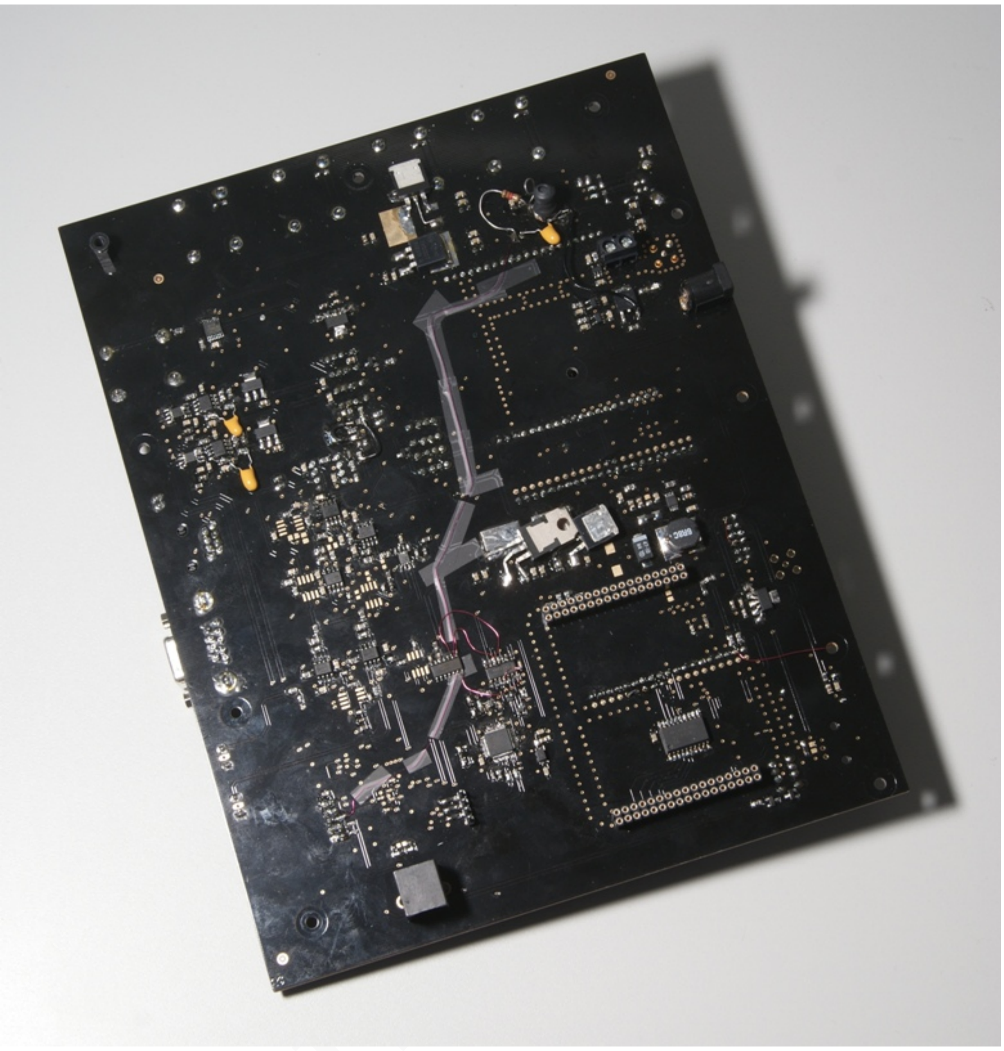


Figure 5.8: Picture of the Chameleon (Bottom Side)

## 5.2 Improvements

The performance of the board can be increased. To do this, the following things should be done:

- Replacing the power supplies for the interface of the OTA and voltage follower regulator
- Redesign the traces of the CL-Interface to reduce the parasitic capacitance for higher resonance frequencies

- Redesign the traces of the clock line for the Byte Paradigm devices to match the impedance
- Adding a resistor in series to the output of the crystal oscillator
- Changing the package of the crystal oscillator
- Improve the filter of the supply of the crystal oscillator
- Adding an inverter in the Data Terminal Ready (DTR) line of the ISO 7816 Reader for better compatibility

There are two major problems in the circuit.

One problem is the DC-DC-Converter which is used to produce the $\pm 15V$ for the supply of the OPAMPs. This converter has a very high start-up current (up to 600 mA) and requires a low ohmic supply voltage. If this is not available, the converter starts to draw the start-up current continuously. An FPF2123[1] device is used to protect and switch on and off this part of the circuit. While using the device, some problems with this device by using it to protect this converter have been recognized. Either the resistance of the FPF2123 is too high or the current limitation is too fast but the FPF2123 switch is not able to enable the power supply properly. A workaround for the current version is to bypass the FPF2123 by using a glass tube fuse.

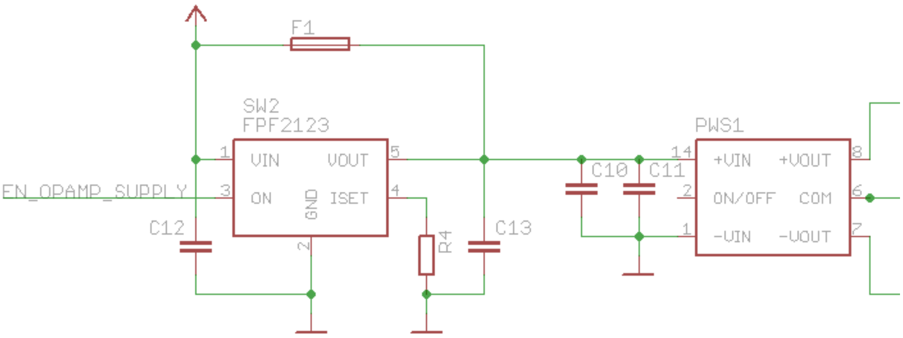


Figure 5.9: Workaround for the DC-DC-Converter

The other problem is the leveltranslator for the I2C-Module. Apparently the supply voltage $V_{Ref1}$ of the PCA9306[2] has to be lower than $V_{Ref2}$. Tests showed, that the lower limit of $V_{Ref2}$ is $V_{Ref1}$. It seems, that $V_{Ref2}$ will be supplied over a internal protection circuit of the PCA9306. So, the interfaces can only be tested in the range of 3.3 V to 5.5 V.

[1] FPF2123: It is a high side switch with over current and short circuit protection

[2] Please refer to the datasheet of the PCA9306

# 6 Acknowledgements

I want to thank Infineon Technologies Austria AG for giving me the chance to write my diploma thesis there. I especially want to thank my supervisor of the company, Mihai Tudosi, for finding such an interesting topic and helping me if I had problems. I learned a lot while working with you.

A great thanks goes to Günter Weber for all organizational things and letting me join his team and also for teaching and pushing me in all planning and presentation topics.

A special thanks goes to my University assessor, Peter Söser, for the smooth collaboration.

I also want to thank my colleagues for their inputs and ideas and for the motivation that they gave. A thanks goes also to Richard Sbuell for explanations of voltage regulators.

And a great thanks goes to Michael Stadler, Jessica Stadler and Matthias Skacel for the constructive discussions and the help with linguistic problems.

Last but not least, I want to thank Thomas Fischer for implementing the SCPI-Parser.

Ing. Georg Skacel, BSc.
Graz, Austria, February 4, 2013

# Bibliography

[1] Chip Card & Security ICs Product Overview
Link: http://www.infineon.com/cms/en/product/applications/chip-card-and-security/index.html

[2] Specification I2C, NXP, Homepage from NXP
Link: http://www.nxp.com/documents/user_manual/UM10204.pdf

[3] Specification LPC, Intel, Homepage from Intel
Link: http://www.intel.com/design/chipsets/industry/lpc.htm

[4] Specification of USB, Homepage from USB
Link: http://www.usb.org/developers/docs/

[5] Specification of DCLB by Infineon Technologies

[6] ISO/IEC 7816-1:2011 Identification cards – Integrated circuit cards – Part 1: Cards with contacts – Physical characteristics
Link: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=54089

[7] ISO/IEC 7816-3:2006 Identification cards – Integrated circuit cards – Part 3: Cards with contacts – Electrical interface and transmission protocols
Link: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38770

[8] ISO/IEC 14443-1:2008 Identification cards – Contactless integrated circuit cards – Proximity cards – Part 1: Physical characteristics
Link: http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=39693

[9] ISO/IEC 14443-2:2010 Identification cards – Contactless integrated circuit cards – Proximity cards – Part 2: Radio frequency power and signal interface
Link: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50941

[10] U. Tietze, Ch. Schenk: *Halbleiter-Schaltungstechnik*, Verlag: Springer, 1999, ISBN: 978-3540641926

[11] Linear Technology, Apllication Note 105, December 2005, Current Sense Circuit Collection
Link: http://www.linear.com/docs/12479

[12] Howard W. Johnson and Martin Graham, *High Speed Digital Design: A Handbook of Black Magic (Prentice Hall Modern Semiconductor Design)*.Verlag: Prentice Hall, 1993, ISBN: 978-0133957242

[13] Howard W. Johnson and Martin Graham, *High Speed Signal Propagation: Advanced Black Magic (Prentice Hall Modern Semiconductor Design)* .Verlag: Prentice Hall, 2003, ISBN: 978-0130844088

[14] Microchip Technology Inc., Datasheet of the PIC18F6722
Link: http://ww1.microchip.com/downloads/en/devicedoc/39646c.pdf

[15] Microchip Technology Inc., Datasheet of the MCP1826
Link: http://ww1.microchip.com/downloads/en/devicedoc/22057a.pdf

[16] Texas Instruments, Datasheet of the PCA9306
Link: http://www.ti.com/lit/ds/symlink/pca9306.pdf

[17] Texas Instruments, Datasheet of the SN74LVC2T45
Link: http://www.ti.com/product/sn74lvc2t45

[18] Texas Instruments, Datasheet of the THS4631
Link: http://www.ti.com/lit/ds/symlink/ths4631.pdf

[19] Texas Instruments, Datasheet of the INA111
Link: www.ti.com/lit/ds/symlink/ina111.pdf

[20] Linear Technology, Datasheet of the LT1995
Link: http://cds.linear.com/docs/Datasheet/1995fb.pdf

[21] Fairchild Semiconductor, Datasheet of the FPF2123
Link: http://www.fairchildsemi.com/ds/FP/FPF2124.pdf

[22] Fairchild Semiconductor, Datasheet of the NC7WZ07
Link: http://www.fairchildsemi.com/ds/NC/NC7WZ07.pdf

[23] Fairchild Semiconductor, Datasheet of the NC7WZ04
Link: http://www.fairchildsemi.com/ds/NC/NC7WZ04.pdf

[24] Fairchild Semiconductor, Datasheet of the 74ACT125
Link: http://www.fairchildsemi.com/ds/74/74AC125.pdf

[25] Fairchild Semiconductor, Datasheet of the 74ACT541
Link: http://www.fairchildsemi.com/ds/74/74AC541.pdf

[26] Analog Devices, Datasheet of the ADG2128
Link: http://www.analog.com/static/imported-files/data_sheets/ADG2128.pdf

[27] Analog Devices, Datasheet of the AD5620
Link: http://www.analog.com/static/imported-files/data_sheets/AD5620_5640_5660.pdf

[28] Analog Devices, Datasheet of the OP262
Link: http://www.analog.com/static/imported-files/data_sheets/OP162_262_462.pdf

[29] Analog Devices, Datasheet of the AD790
Link: http://www.analog.com/static/imported-files/data_sheets/AD790.pdf

[30] Analog Devices, Datasheet of the ADG1309
Link: http://www.analog.com/static/imported-files/data_sheets/ADG1308_1309.pdf

[31] Byteparadigm,Datasheet of the GP24132
Link: http://www.byteparadigm.com/download.php?file=ds_GP241xx.pdf

[32] FTDI Chip, Datasheet of the FT2232C
Link: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232C.pdf

[33] Henry Bruhns: *USB in der Messtechnik*. Franzis–Verlag, Haar bei München, 1996, ISBN: 978-3772355097, http://www.usbtmc.de/

[34] Specifiaction of SCPI; Homepage from IVI-Foundation
Link: http://www.ivifoundation.org/docs/scpi-99.pdf

# List of Figures

2.1 Write sequence to the ADG2128. Picture taken from the Datasheet of the ADG2128 . . . 6
2.2 Direction of the ISO 7816 Interface at the DUT . . . 8
2.3 Direction of the DCLB Interface at the DUT . . . 9

3.1 uC and Interface to PC . . . 14
3.2 Power Supply for Core of the DUT . . . 15
3.3 VDDP . . . 16
3.4 Amplification and current limiting of VDDP . . . 17
3.5 Resistor after emitter/drain . . . 18
3.6 Voltage regulator with sense resistor in the drain path . . . 19
3.7 Output voltage (yellow) of the regulator with converted sense current (cyan) . . . 20
3.8 Current mirror . . . 21
3.9 OTA with current mirror . . . 22
3.10 Current profile measured with current mirror . . . 23
3.11 Cascode current mirror . . . 24
3.12 Current profile measured with cascode current mirror . . . 24
3.13 OTA with cascode current mirror . . . 25
3.14 Delay of $I_{DUT}$ (yellow) and $I_{Sense}$ (cyan) . . . 26
3.15 Schematic of the current sensing path of the realized version (Chameleon V.15) . . . 27
3.16 Schematic of the current sensing path with the OTA . . . 28
3.17 VDDGPIO . . . 30
3.18 Schematic of the regulator for VDDGPIO and VDDPISO . . . 31
3.19 VDDLPC . . . 31
3.20 Schematic of the VDDLPC supply . . . 32
3.21 LPC . . . 32
3.22 Buffer for the LPC Interface . . . 33
3.23 ISOPAD source switching . . . 34
3.24 Schematic of the ISOPAD source switching . . . 35
3.25 DCLB . . . 35
3.26 ISO 7816 Reader . . . 36
3.27 Schematic of the ISO 7816 reader . . . 37
3.28 Headerboard . . . 39
3.29 I2C Block . . . 40

3.30 Level translator of the I2C interface . . . . . . . . . . . . . . . . . . . . . 41
3.31 SPI Block . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
3.32 Leveltranslator of the SPI interface . . . . . . . . . . . . . . . . . . . . . 43
3.33 Logicanalyzer and Pattern generator . . . . . . . . . . . . . . . . . . . . . 45
3.34 Directions of the lines at the pattern generator . . . . . . . . . . . . . . 46
3.35 Directions of the lines at the logic analyzer . . . . . . . . . . . . . . . . . 46

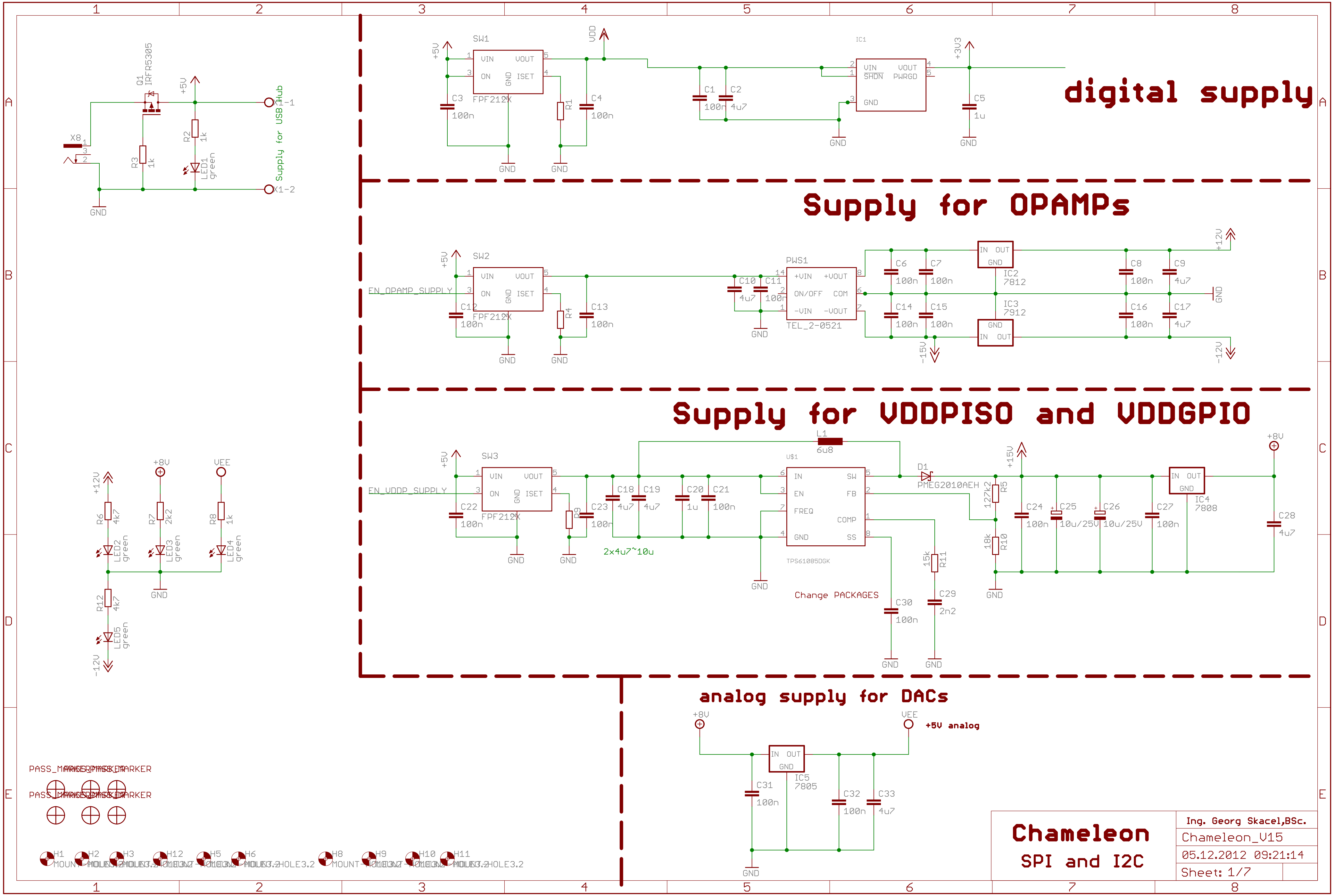
4.1 The state-machine of chameleon . . . . . . . . . . . . . . . . . . . . . . . 49
4.2 Decoding the example . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
4.3 Example for sending an IND?-Query . . . . . . . . . . . . . . . . . . . . . 53
4.4 Layers of the software . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
4.5 Screenshoot of the GUI . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

5.1 Picture of a headerboard for a DUT . . . . . . . . . . . . . . . . . . . . . 60
5.2 Picture of the OTA + current mirror testboard . . . . . . . . . . . . . . . . 61
5.3 Picture of a headerboard without levelshifter . . . . . . . . . . . . . . . . 62
5.4 Picture of a headerboard with levelshifter . . . . . . . . . . . . . . . . . . 63
5.5 Picture of connectors for probecard and external reader . . . . . . . . . . 63
5.6 Picture of the Chameleon with headerboards . . . . . . . . . . . . . . . . 64
5.7 Picture of the Chameleon with headerboards . . . . . . . . . . . . . . . . 65
5.8 Picture of the Chameleon (Bottom Side) . . . . . . . . . . . . . . . . . . . 66
5.9 Workaround for the DC-DC-Converter . . . . . . . . . . . . . . . . . . . . 67

# List of Tables

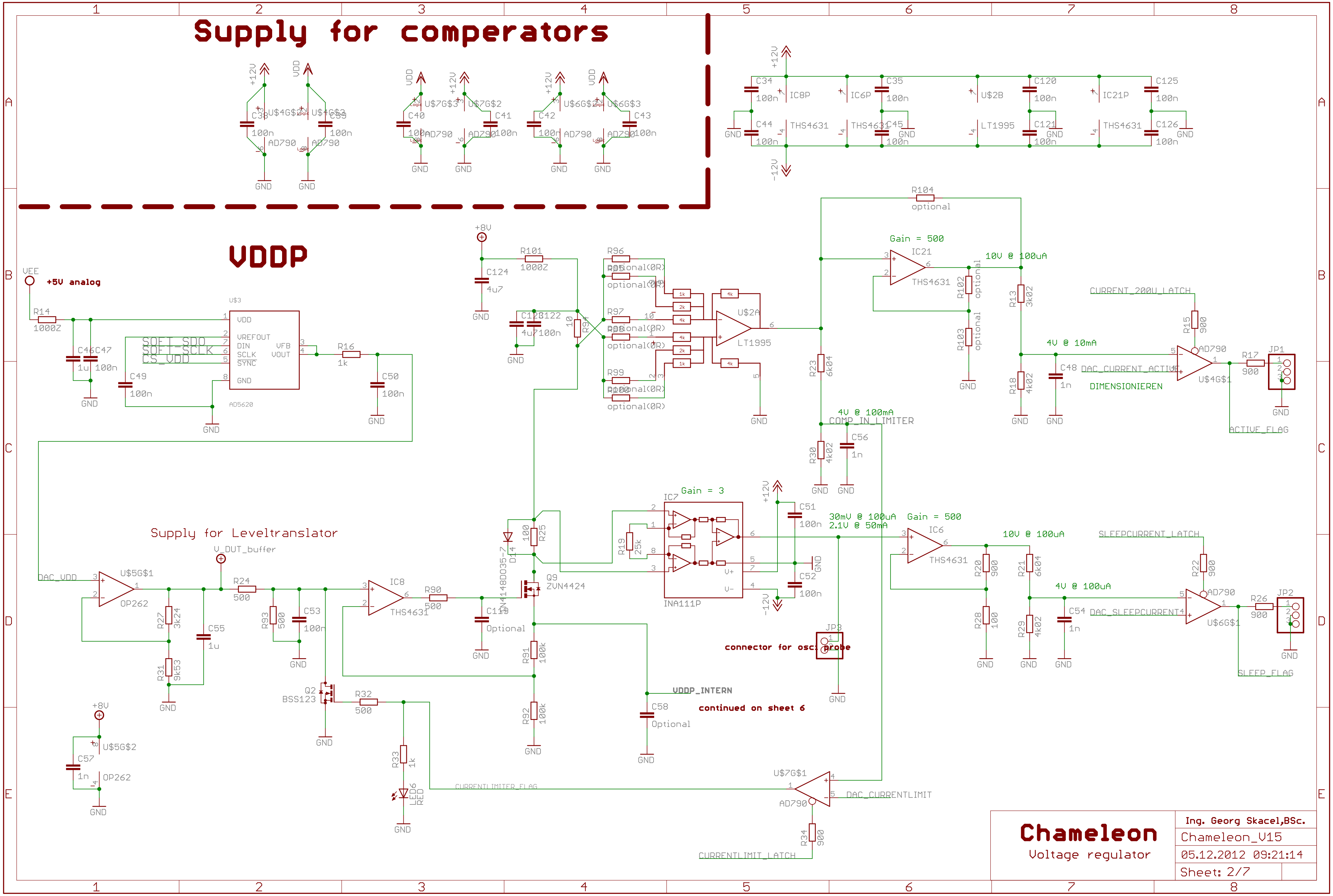
3.1 Advantages and disadvantages of the current mirror . . . . . . . . . . . . 23
3.2 Pin assignment of the FT2232D . . . . . . . . . . . . . . . . . . . . . . . . . 38
3.3 Pin assignment of the switch matrix . . . . . . . . . . . . . . . . . . . . . . 40
3.4 Data direction of the SN74LVC2T45 . . . . . . . . . . . . . . . . . . . . . . 42
3.5 The maximum propagation delay of the SN74LVC2T45 . . . . . . . . . . . 44

# 7 Appendix



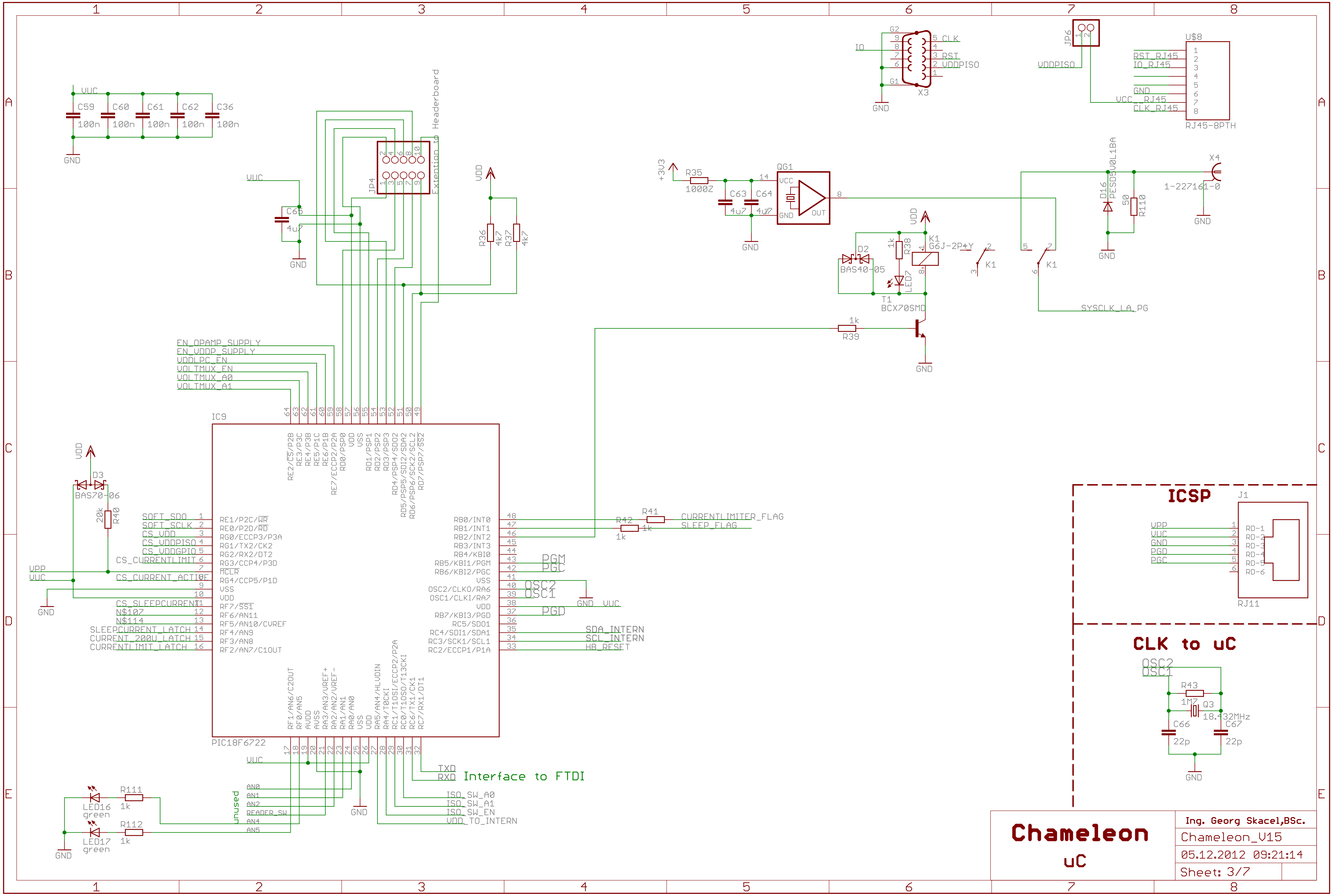
Chameleon
SPI and I2C

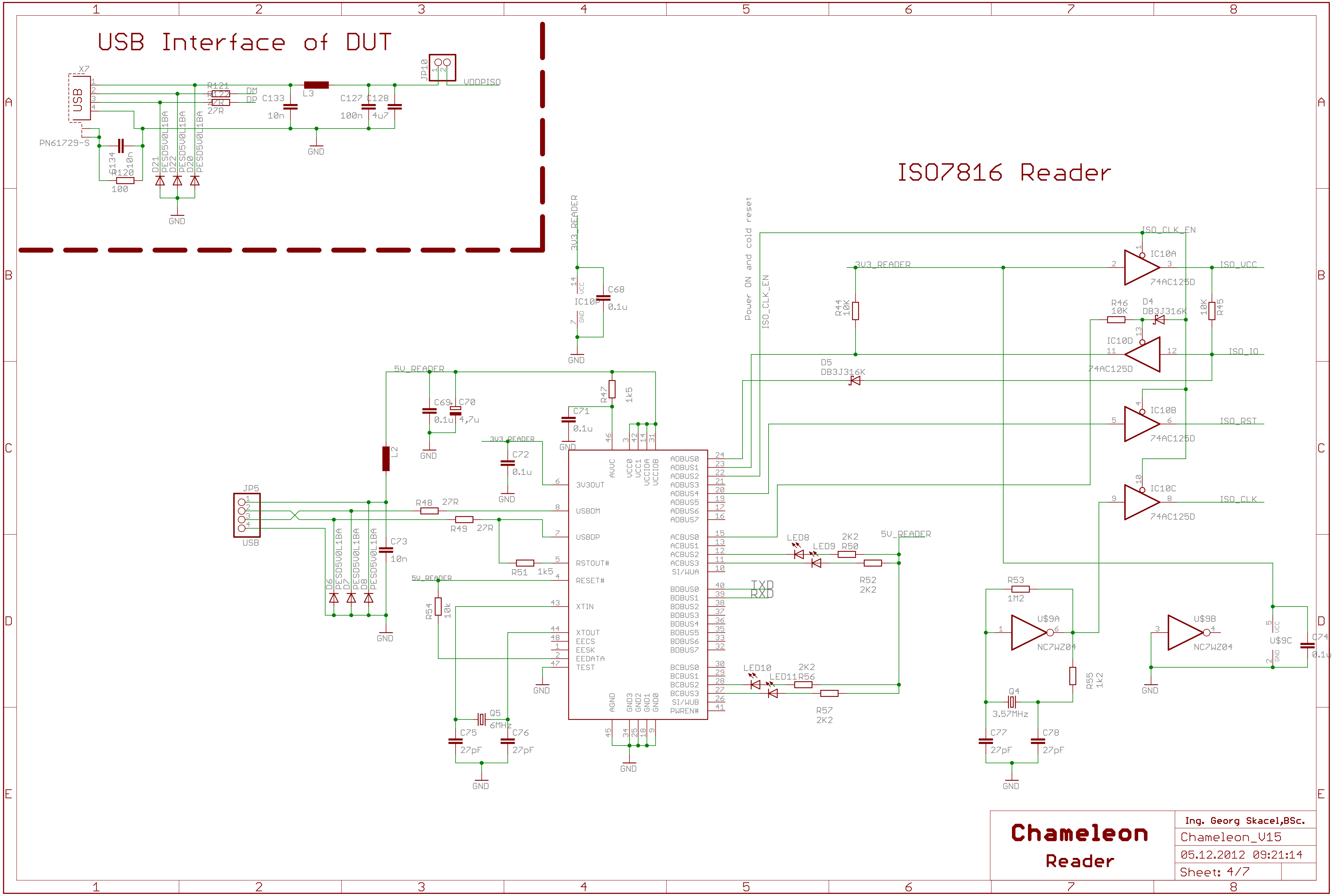
Ing. Georg Skacel,BSc.
Chameleon_V15
05.12.2012 09:21:14
Sheet: 1/7



**Chameleon**
Voltage regulator

Ing. Georg Skacel,BSc.
Chameleon_V15
05.12.2012 09:21:14
Sheet: 2/7


Chameleon
uC
Ing. Georg Skacel,BSc.
Chameleon_V15
05.12.2012 09:21:14
Sheet: 3/7
ICSP
J1
RJ11
RD-1
RD-2
RD-3
RD-4
RD-5
RD-6
VPP
VUC
GND
PGD
PGC
CLK to uC
OSC2
OSC1
R43
1M7
Q3
18.432MHz
C66
22p
C67
22p
GND
IC9
PIC18F6722
RE1/P2C/WR
RE0/P2D/RD
RG0/ECCP3/P3A
RG1/TX2/CK2
RG2/RX2/DT2
RG3/CCP4/P3D
MCLR
RG4/CCP5/P1D
VSS
VDD
RF7/SS1
RF6/AN11
RF5/AN10/CVREF
RF4/AN9
RF3/AN8
RF2/AN7/C1OUT
RB0/INT0
RB1/INT1
RB2/INT2
RB3/INT3
RB4/KBI0
RB5/KBI1/PGM
RB6/KBI2/PGC
VSS
OSC2/CLKO/RA6
OSC1/CLKI/RA7
VDD
RB7/KBI3/PGD
RC5/SDO1
RC4/SDI1/SDA1
RC3/SCK1/SCL1
RC2/ECCP1/P1A
RE2/CS/P2B
RE3/P3C
RE4/P3B
RE5/P1C
RE6/P1B
RE7/ECCP2/P2A
RD0/PSP0
VDD
VSS
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4/SDO2
RD5/PSP5/SDI2/SDA2
RD6/PSP6/SCK2/SCL2
RD7/PSP7/SS2
RF1/AN6/C2OUT
RF0/AN5
AVDD
AVSS
RA3/AN3/VREF+
RA2/AN2/VREF-
RA1/AN1
RA0/AN0
VSS
VDD
RA5/AN4/HLVDIN
RA4/T0CKI
RC1/T1OSI/ECCP2/P2A
RC0/T1OSO/T13CKI
RC6/TX1/CK1
RC7/RX1/DT1
SOFT_SDO
SOFT_SCLK
CS_VDD
CS_VDDPISO
CS_VDDGPIO
CS_CURRENTLIMIT
CS_CURRENT_ACTIVE
CS_SLEEPCURRENT
N$107
N$114
SLEEPCURRENT_LATCH
CURRENT_200U_LATCH
CURRENTLIMIT_LATCH
VPP
VUC
GND
VDD
D3
BAS70-06
R40
20k
CURRENTLIMITER_FLAG
SLEEP_FLAG
R41
R42
1k
PGM
PGC
OSC2
OSC1
GND
VUC
PGD
SDA_INTERN
SCL_INTERN
HB_RESET
EN_OPAMP_SUPPLY
EN_VDDP_SUPPLY
VDDLPC_EN
VOLTMUX_EN
VOLTMUX_A0
VOLTMUX_A1
TXD
RXD
Interface to FTDI
ISO_SW_A0
ISO_SW_A1
ISO_SW_EN
VDD_TO_INTERN
AN0
AN1
AN2
READER_SW
AN4
AN5
unused
R111
R112
1k
LED16
LED17
green
C59
C60
C61
C62
C36
100n
C65
4u7
JP4
Extention to Headerboard
R36
R37
4k7
+3V3
R35
1000Z
C63
C64
QG1
VCC
OUT
K1
G6J-2P+Y
R38
LED7
D2
BAS40-05
T1
BCX70SMD
R39
SYSCLK_LA_PG
D16
PESD5V0L1BA
R110
50
X4
1-227161-0
X3
G1
G2
IO
CLK
RST
VDDPISO
JP6
U$8
RJ45-8PTH
RST_RJ45
IO_RJ45
VCC__RJ45
CLK_RJ45



Chameleon

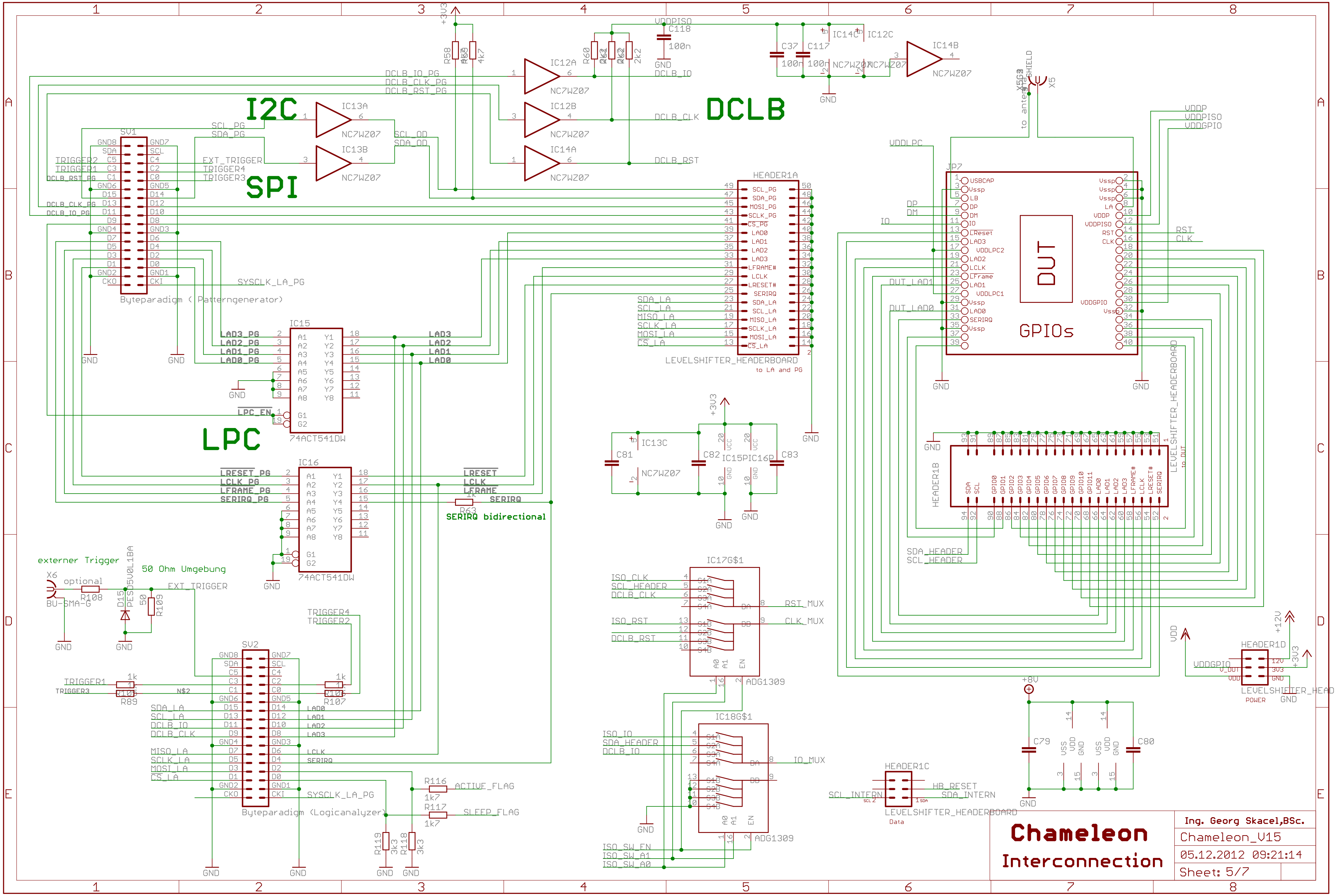
Reader

Ing. Georg Skacel,BSc.

Chameleon_V15

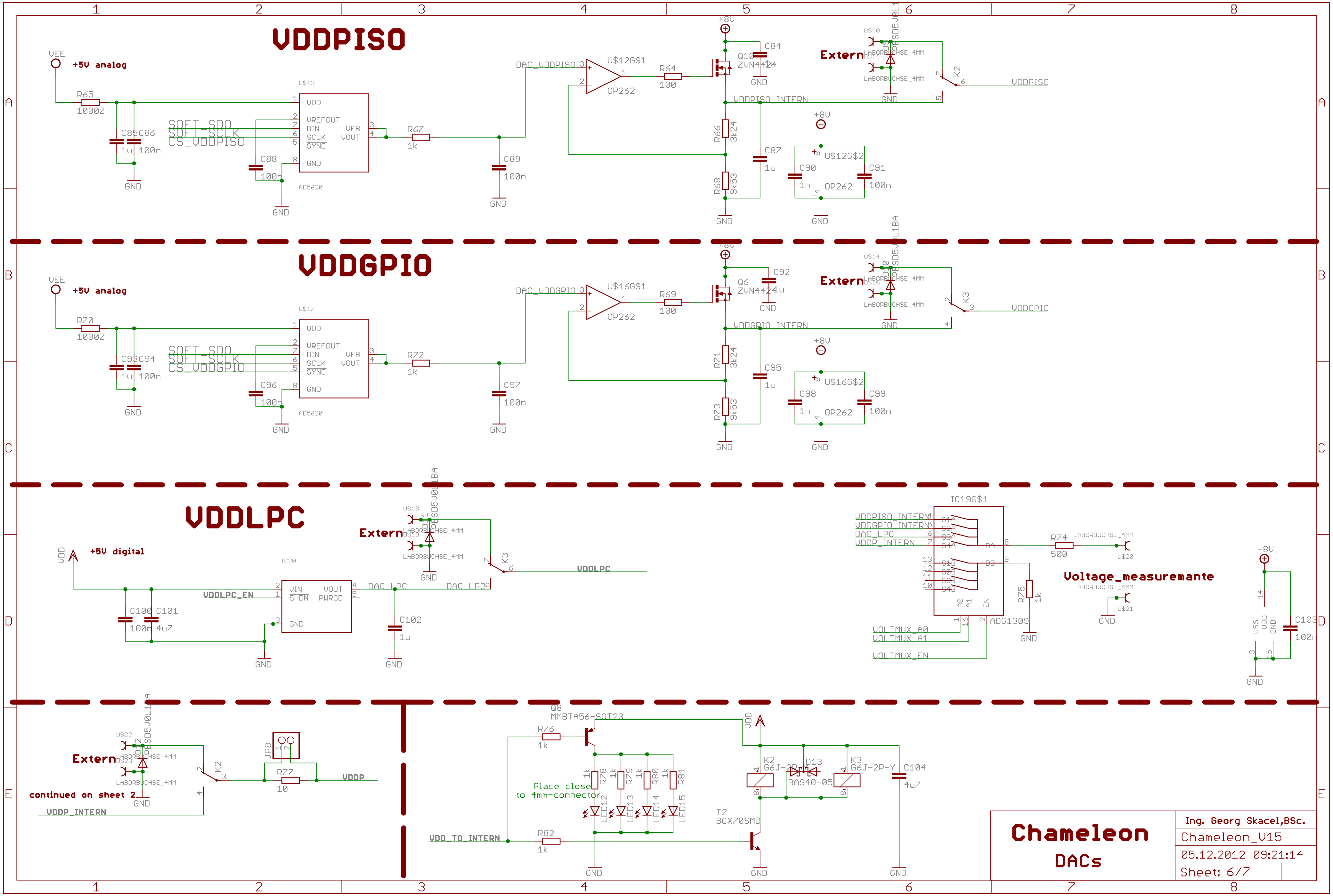
05.12.2012 09:21:14

Sheet: 4/7



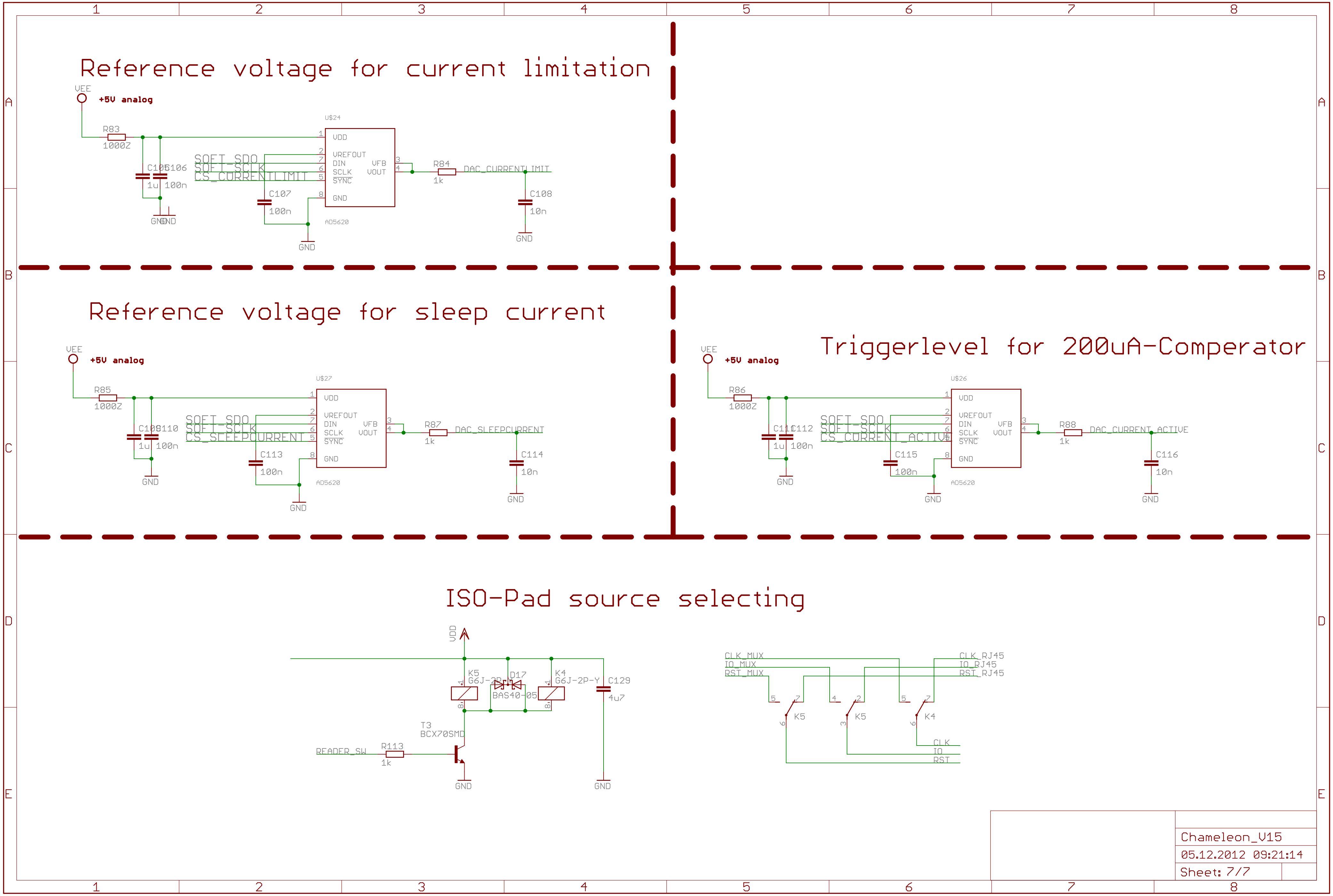
Ing. Georg Skacel,BSc.
Chameleon_V15
05.12.2012 09:21:14
Sheet: 5/7

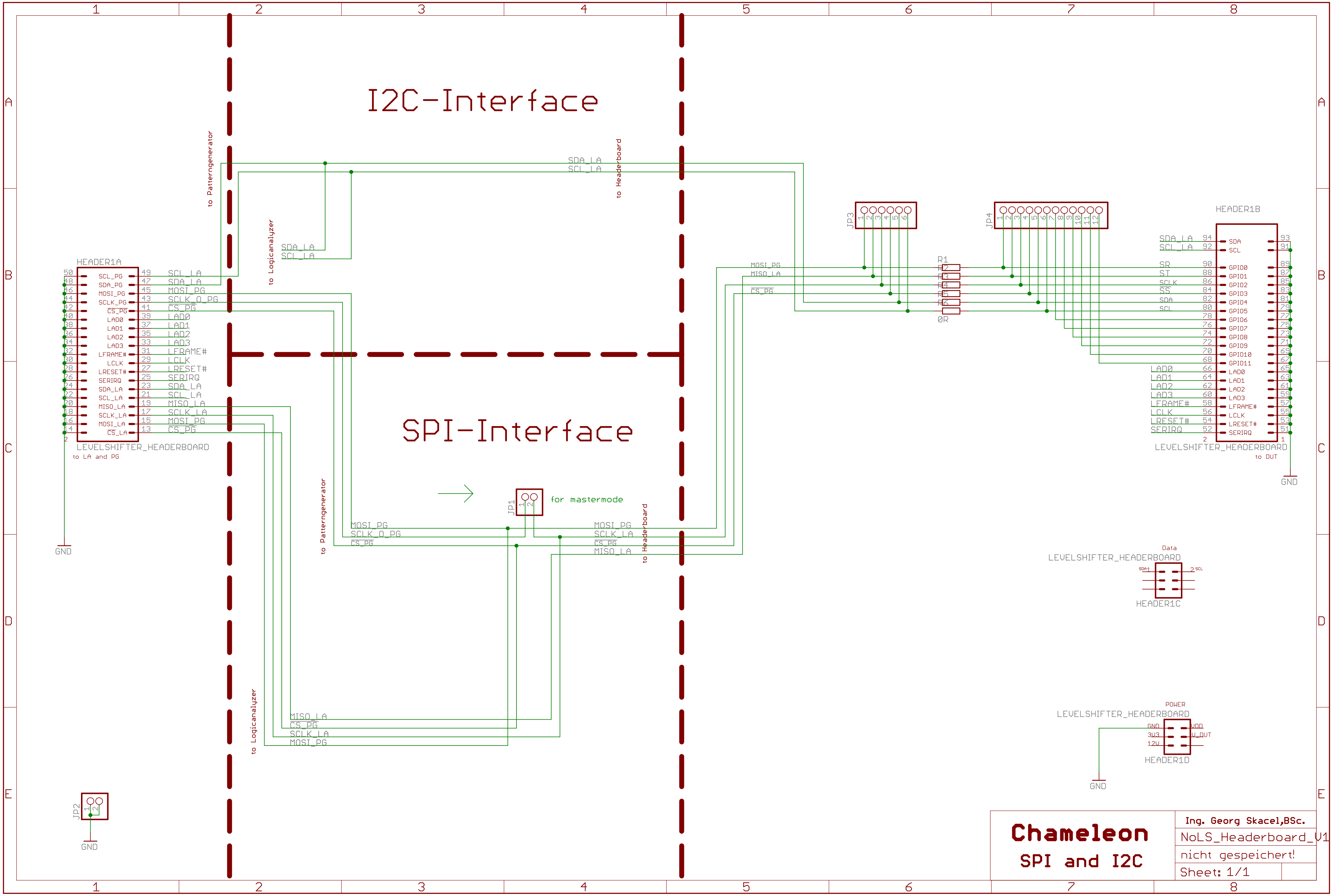
Chameleon
Interconnection



Chameleon
DACs

Ing. Georg Skacel,BSc.
Chameleon_V15
05.12.2012 09:21:14
Sheet: 6/7


Reference voltage for current limitation
VEE
+5V analog
R83
1000Z
C105
C106
1u
100n
GND
GND
U$24
VDD
VREFOUT
DIN
SCLK
SYNC
GND
VFB
VOUT
AD5620
SOFT_SDO
SOFT_SCLK
CS_CURRENTLIMIT
C107
100n
GND
R84
1k
DAC_CURRENTLIMIT
C108
10n
GND
Reference voltage for sleep current
VEE
+5V analog
R85
1000Z
C109
C110
1u
100n
GND
U$27
AD5620
SOFT_SDO
SOFT_SCLK
CS_SLEEPCURRENT
C113
100n
GND
R87
1k
DAC_SLEEPCURRENT
C114
10n
GND
Triggerlevel for 200uA-Comperator
VEE
+5V analog
R86
1000Z
C111
C112
1u
100n
GND
U$26
AD5620
SOFT_SDO
SOFT_SCLK
CS_CURRENT_ACTIVE
C115
100n
GND
R88
1k
DAC_CURRENT_ACTIVE
C116
10n
GND
ISO-Pad source selecting
VDD
K5
G6J-2P-Y
D17
BAS40-05
K4
G6J-2P-Y
C129
4u7
T3
BCX70SMD
READER_SW
R113
1k
GND
GND
CLK_MUX
IO_MUX
RST_MUX
CLK_RJ45
IO_RJ45
RST_RJ45
K5
K5
K4
CLK
IO
RST
Chameleon_V15
05.12.2012 09:21:14
Sheet: 7/7



Chameleon

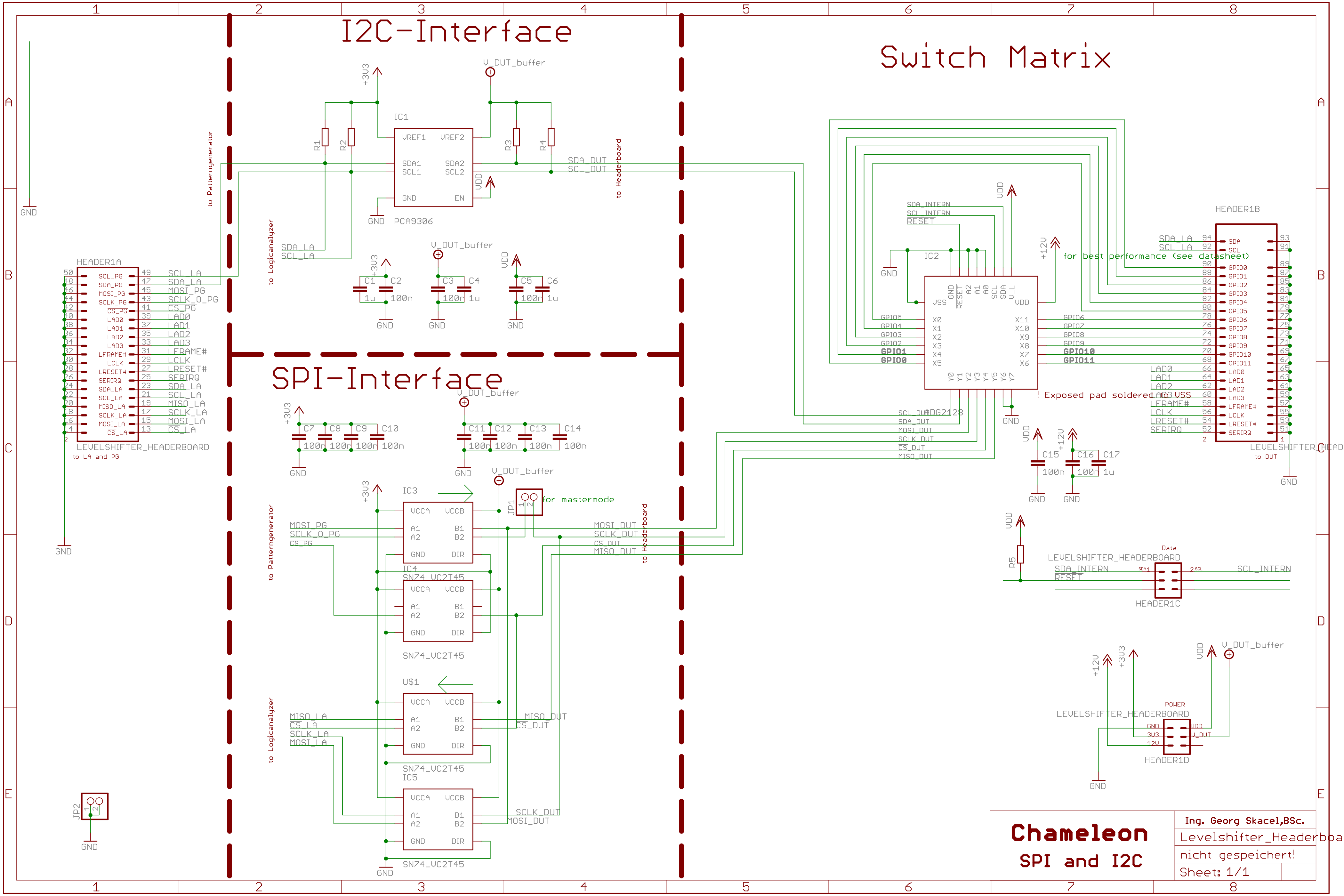
SPI and I2C

Ing. Georg Skacel,BSc.

NoLS_Headerboard_V1

nicht gespeichert!

Sheet: 1/1


I2C-Interface
Switch Matrix
SPI-Interface
HEADER1A
LEVELSHIFTER_HEADERBOARD
to LA and PG
SCL_PG
SDA_PG
MOSI_PG
SCLK_PG
CS_PG
LAD0
LAD1
LAD2
LAD3
LFRAME#
LCLK
LRESET#
SERIRQ
SDA_LA
SCL_LA
MISO_LA
SCLK_LA
MOSI_LA
CS_LA
SCLK_O_PG
to Patterngenerator
to Logicanalyzer
to Headerboard
IC1
PCA9306
VREF1
VREF2
SDA1
SCL1
SDA2
SCL2
GND
EN
R1
R2
R3
R4
+3V3
V_DUT_buffer
VDD
SDA_DUT
SCL_DUT
C1
C2
C3
C4
C5
C6
1u
100n
C7
C8
C9
C10
C11
C12
C13
C14
IC3
IC4
IC5
U$1
SN74LVC2T45
VCCA
VCCB
A1
A2
B1
B2
DIR
JP1
for mastermode
MOSI_DUT
SCLK_DUT
CS_DUT
MISO_DUT
JP2
IC2
ADG2128
VSS
RESET
A0
A1
A2
SCL
SDA
V_L
VDD
X0
X1
X2
X3
X4
X5
X6
X7
X8
X9
X10
X11
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
SDA_INTERN
SCL_INTERN
GPIO0
GPIO1
GPIO2
GPIO3
GPIO4
GPIO5
GPIO6
GPIO7
GPIO8
GPIO9
GPIO10
GPIO11
! Exposed pad soldered to USS
+12V
for best performance (see datasheet)
HEADER1B
to DUT
C15
C16
C17
R5
Data
HEADER1C
POWER
HEADER1D
3V3
12V
V_DUT
GND
Chameleon
SPI and I2C
Ing. Georg Skacel,BSc.
Levelshifter_Headerboard_V5
nicht gespeichert!
Sheet: 1/1