**H ö h e r e T e c h n i s c h e B u n d e s l e h r a n s t a l t**

**S a l z b u r g**

**Abteilung für Elektronik und Technische Informatik**

**Übungen im**

**Laboratorium für Elektronik**

**Protokoll**

**für die Übung SreS 05**

**Gegenstand der Übung**

|  |
| --- |
| **DAC SPI** |

|  |  |
| --- | --- |
| **Name:** | **Eschbacher Jan-Lorenz** |
| **Jahrgang:** | **4AHEL** |
| **Gruppe Nr.:** | **A01** |
| **Übung am:** | **08.01.2020** |

|  |  |
| --- | --- |
| **Anwesende:** | **Evelyn Aichinger, Eschbacher Jan, Baumgartner Lukas** |

***Inhaltsverzeichnis***

[1. Einleitung 3](#_Toc29960605)

[1.1 DAC Chip LTC1661 3](#_Toc29960606)

[2. Inventarliste 3](#_Toc29960607)

[3. Übungsdurchführung 4](#_Toc29960608)

[3.1 Vorbereitungsfragen 4](#_Toc29960609)

[3.2 Aufbau der Signalübertragung 5](#_Toc29960610)

[3.3 Berechnung 5](#_Toc29960611)

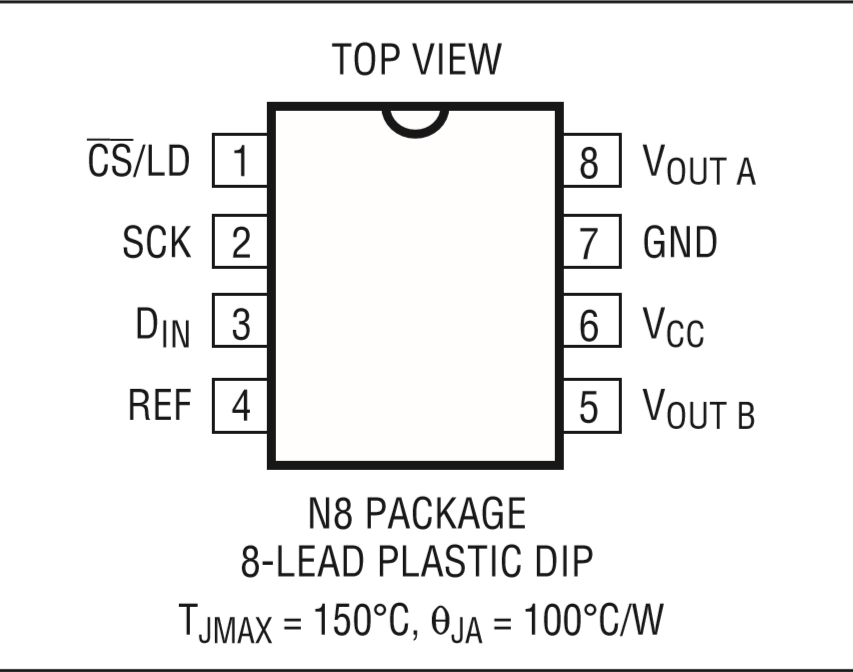
[3.4 Programmcode 6](#_Toc29960612)

[3.5 Messergebnis 7](#_Toc29960613)

[4. Zusammenfassung 8](#_Toc29960614)

# Einleitung

## 1.1 DAC Chip LTC1661

In dieser Unterrichtseinheit verwenden wir einen DAC(Digital-to-analog converter), welcher über das Bus System SPI angesprochen wird und aus dem Digitalen Signal eine Analoge Spannung macht. Der zu verwendete Chip ist der LTC1661. In der Abbildung 1. ist das Layout des Bausteines zu sehen. Vout A und Vout B können unabhängig voneinander angesteuert werden.

*Abbildung 1. (Layout des Chips LTC1661)*

# Inventarliste

|  |  |  |
| --- | --- | --- |
| **Stück** | **Gerätebezeichnung** | **Inventarnummer/Identifikation** |
| 1 | UNI-T UT803 true RMS multimeter | - |
| 1 | Logic Analyzer | - |
| 1 | TBS 1102B Digital Oscilloscope 100Mz 2GS/s | - |

# Übungsdurchführung

## 3.1 Vorbereitungsfragen

**a) Welche Datenwortbreite ist am SPI Interface zu realisieren?**

Eine Vollständige Übertragung beinhaltet 16 Bit.

**b) Wie ist der SPI Befehl für den DAC aufgebaut?**

Sieh 3.2 Aufbau der Signalübertragung.

**c) Welche DAC Referenzspannung kann gewählt werden, und wie?**

Die Referenzspannung kann zwischen 0[V] und Vcc[V] gewählt werden. Und diese Referenzspannung ist am Pin 4(VREF) am Chip LTC1661, siehe Abbildung 1., anzulegen.

**d) Wie wird die auszugebende Spannung errechnet?**

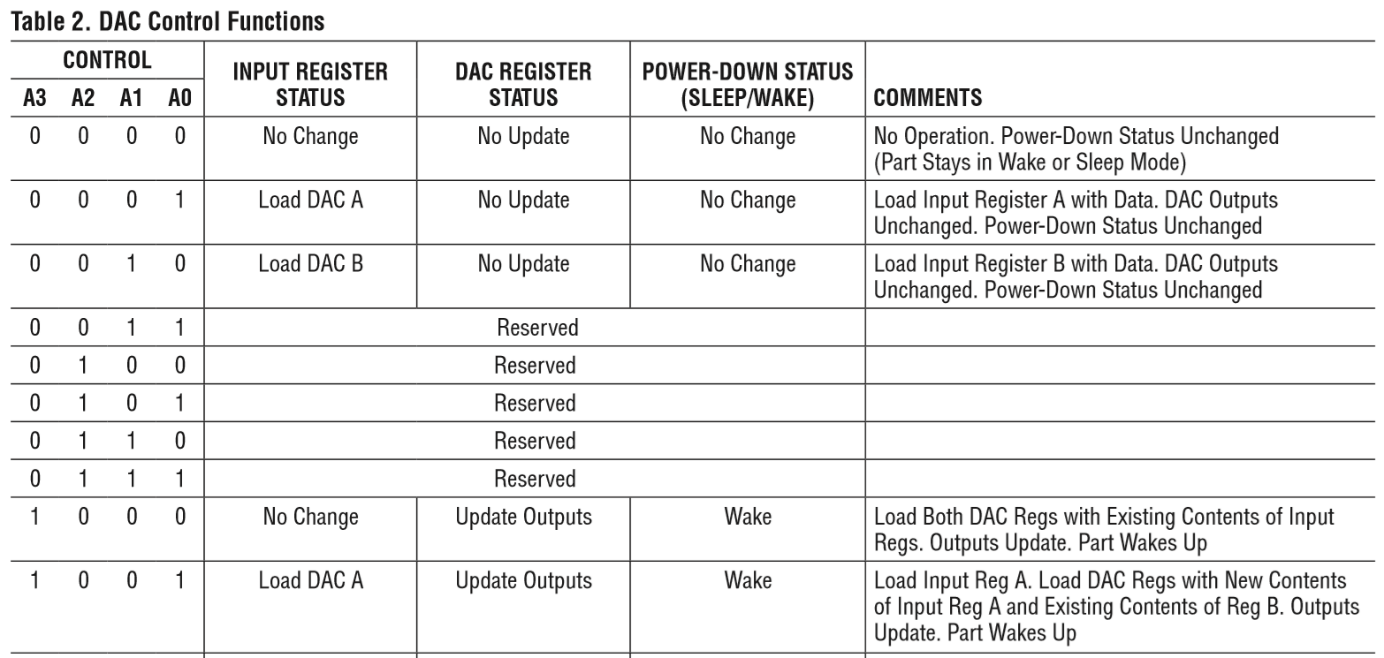
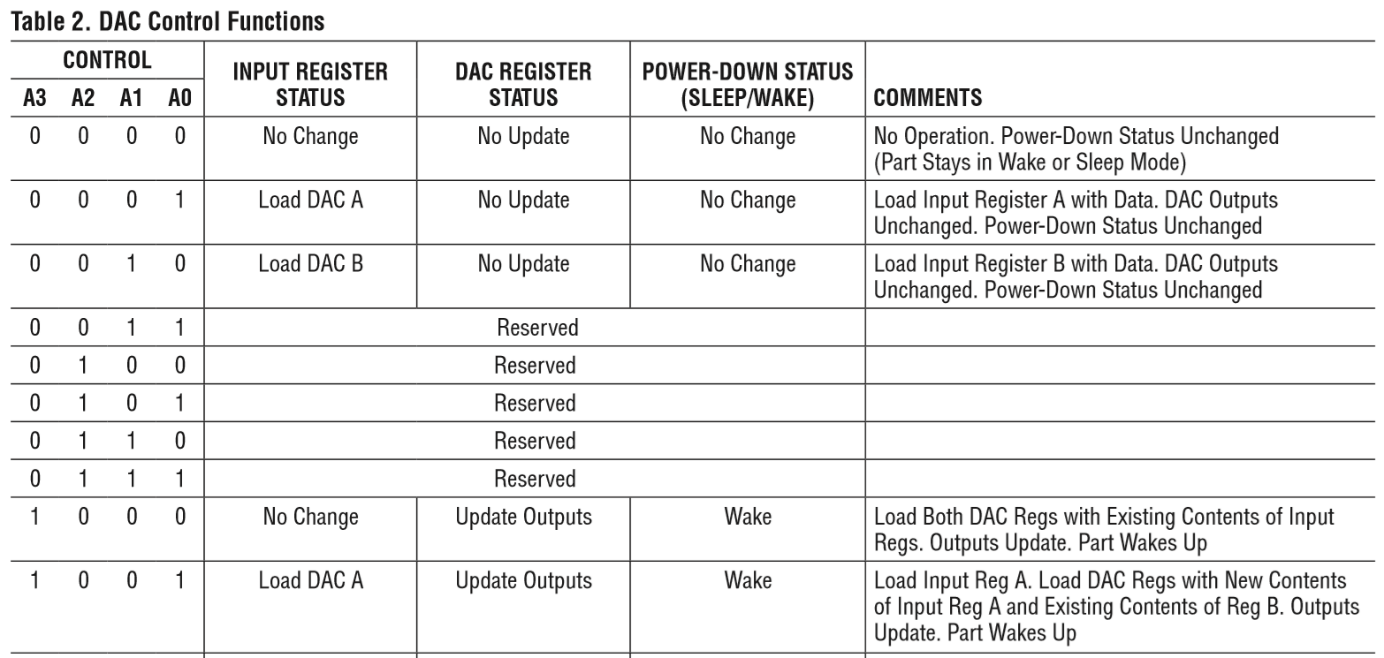
Sieh *3.2 Aufbau der Signalübertragung* und *3.3 Berechnung*

**e) Welche maximale Taktfrequenz ist am SPI IF des LTC möglich?**

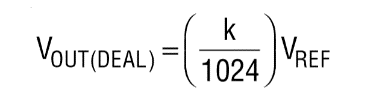
In unserem Fall ist eine maximale Taktfrequenz von 16,7MHz einzustellen da wir den Chip zwischen 4.5 bis 5.5V betreiben, sonst ist eine maximale Taktfrequenz von 10MHz der Fall.

## 3.2 Aufbau der Signalübertragung

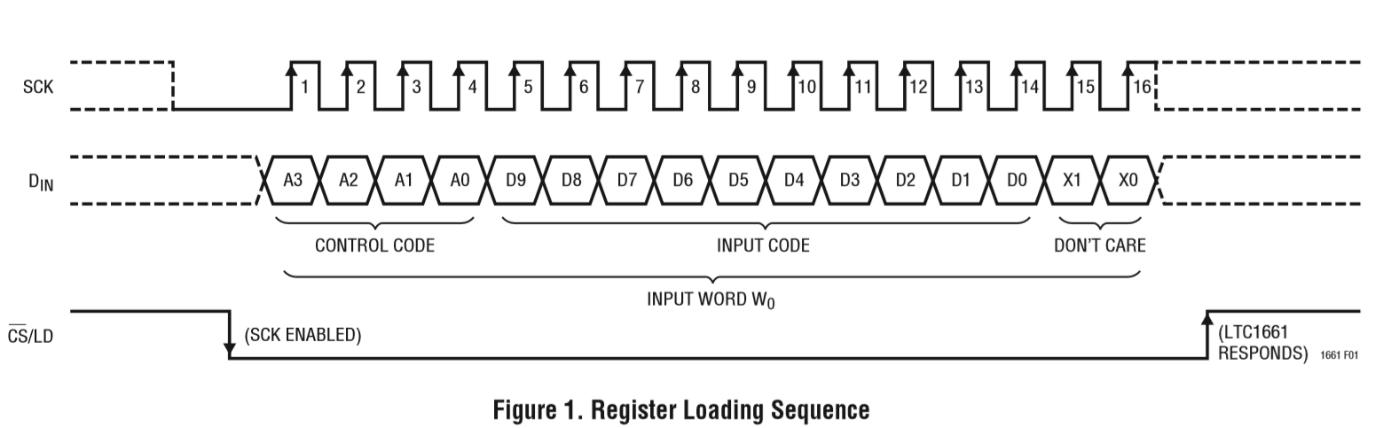
Um den Chip richtig anzusprechen muss das SPI Protokoll eigehalten werden. Die ersten 4Bit bestehen aus den Control Bits. Diese bestimmen in welches Register geschrieben wird. In unseren Fall möchten wir den Ausgang Vout A einstellen. Dafür wird wie in der Abbildung 2. dargestellt, die richtigen Control Bits A3-A0 herausgesucht.



*Abbildung 2. (DAC Control Bits table)*

Die nächsten 10 Bits besteht nun aus dem zu berechneten Inputwert ***k***, welcher das Ausgangssignal Vout A ausmacht. Vout A wird folgendermaßen berechnet:

Dabei ist zu beachten, dass k dezimal einzugeben ist. Um nun k auszurechen, welches binär umgerechnet unsere 10 Bits bestimmen, muss die Formel nur umgeformt werden.

Die letzten 2 Bits sind Don´t Care Felder und werden daher einfach 0 gesetzt. In der Abbildung 3. sieht man nun wie eine korrekte SPI Übertragung aussieht. Dort ist die komplette Übertragung zu sehen mit CLK, Din und CC/LD. Um die Übertragung zu starten wird aus erster die CC Leitung auf LOW gezogen. Danach können die 16 Bits mit dem dazugehöriger CLK Signal übertragen werden. Wenn die Übertragung vollendet ist, wird die CC Leitung wieder auf HIGH gezogen und der Chip beachtet die SCK und Din Leitung nicht mehr.

*Abbildung 3. (SPI Übertragung für LTC1661)*

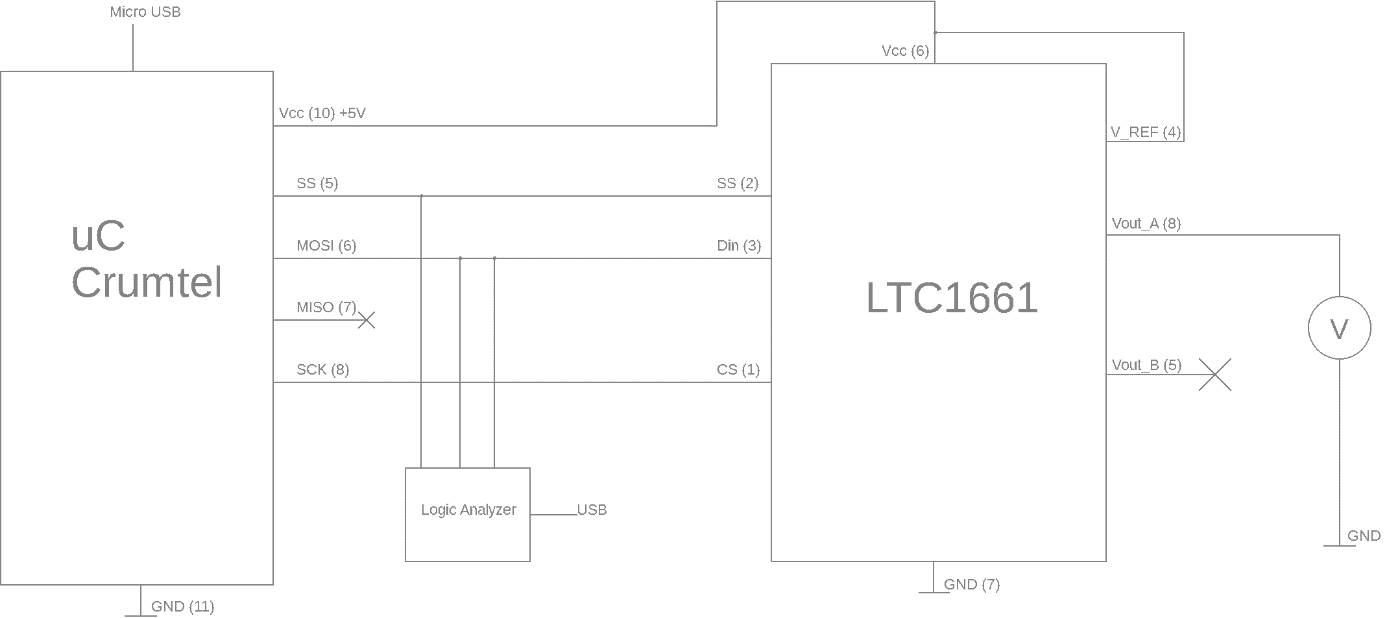
## 3.3 Berechnung

Zu berechnen ist Vout A = 2,7V Vref =5V Vcc=5V

Es muss immer auf die nähere Zahl aufgerundet werden, da keine Kommastellen miteinbezogen werden können.

553 ist binär 10 0010 1001

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Übertragungscode HEX [98A4] | | | | | | | | | | | | | | | | | | |
| Control Bits | | | |  | Code Word 553 binär | | | | | | | | | | | | X X | |
| 1 | 0 | 0 | 1 |  | 1 | 0 | 0 | 0 |  | 1 | 0 | 1 | 0 |  | 0 | 1 | 0 | 0 |

In der Abbildung 4. ist der Aufbau der Schaltung zu sehen, welche verwendet worden ist. Für den uC ist in unserem Fall ein Crumtel verwendet worden. Der MOSI Pin 6 wird in unserem Fall nicht benötigt, da das Crumtel der Master ist und nur ein Slave (LTC1661) vorhanden ist.

*Abbildung 4. (Aufbau der zu Messenden Schaltung)*

## 3.4 Programmcode

/\*

\* SPI.c

\* Created: 08.01.2020 08:37:39

\* Author : Jan

\*/

#include <avr/io.h>

#define DDR\_SPI DDRB

#define DD\_MOSI DDB5

#define DD\_SCK DDB7

#define DD\_SS PB4

void SPI\_MasterInit(void) {

/\* Set MOSI and SCK output, all others input \*/

DDR\_SPI = (1<<DD\_MOSI)|(1<<DD\_SCK)|(1<<DDB4);

/\* Enable SPI, Master, set clock rate fck/16 \*/

SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);

}

void SPI\_MasterTransmit(char cData) {

/\* Start transmission \*/

SPDR = cData;

/\* Wait for transmission complete \*/

while(!(SPSR & (1<<SPIF)));

}

int main(void)

{

SPI\_MasterInit();

PORTB &= ~(1<< DD\_SS); //Pull SS Pin LOW

SPI\_MasterTransmit("98"); //first 8 Bits 1001 1000

SPI\_MasterTransmit("A4"); //second 8 Bits 1010 0100

PORTB |= (1<< DD\_SS); //High SS Pin LOW

}

## 3.5 Messergebnis

In der Abbildung 5 sieht man das übertragene Signal, welches mit dem Logik Analyzer gemessen wurde. Es ist zu sehen, dass das Signal korrekt übermittelt würde. Außerdem ist auch kurz zu sehen das eine gewisse Zeit gebraucht wird bis die zweiten 8 Bits übertragen werden. Die gemessene Ausgangsspannung des Chips lag bei 2,72V. Also eine sehr geringe Abweichung des gewollten Wert 2,7V.



*Abbildung 5. (SPI Übertragung, gemessen mit Logik Analyzer)*

# Zusammenfassung

Die letzte Übung konnte leider nicht durchgeführt werden, da nach der ersten Übung der Chip auf keine der SPI befehle mehr reagierte und uns leider keine Zeit mehr dafür blieb die Schaltung erneut aufzubauen.

Unterschrift:

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum:** | **Note:** | **Punkte:** | **Unterschrift:** |