**Höhere Technische Bundeslehr und Versuchsanstalt Salzburg**

**Abteilung für Elektronik**

**Übungen im**

**Laboratorium für Elektronik**

**Protokoll**

**für die Übung SRES 04**

**Gegenstand der Übung**

|  |
| --- |
| **ADC 2** |
|  |

|  |  |
| --- | --- |
| **Name:** | **Ingomar Mayer** |
| **Jahrgang:** | **4AHEL** |
| **Gruppe Nr.:** | **B02** |
| **Übung am:** | **27.11.2019** |

|  |  |
| --- | --- |
| **Anwesende:** | **Ingomar Mayer, Jakob Mayer** |

**Inhaltsverzeichnis**

[1. Einleitung 3](#_Toc26309115)

[2. Inventarliste 3](#_Toc26309116)

[3. Übungsdurchführung: 4](#_Toc26309117)

[3.1. Beantwortung der Kontrollfragen 4](#_Toc26309118)

[3.2. Schaltungsentwurf ADC 5](#_Toc26309119)

[3.3. Messung am ADC 6](#_Toc26309120)

# Einleitung

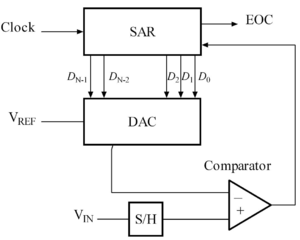
In der heutigen Übung verwenden wir einen SAR Analog-Digital-Converter. SAR steht für successive approximation register auf Deutsch Wägeverfahren. Bei diesem Verfahren wird die Eingangsspannung immer mit dem halben Referenz Wert von einem Schritt zuvor verglichen. Der erste Referenzwert ist Vref/2. Ist die Eingangsspannung niedriger so wird eine 0 in das MSB geschrieben und es geht einen Schritt weiter. Nun wird die Eingangspannung mit Vref/2-Vref/4 verglichen. Ist die Eingangsspannung größer als die Referenz wird im nächsten Schritt subtrahiert. Ist die Eingangsspannung kleiner so wird im nächsten Schritt addiert. Dieser Vorgang wiederholt sich bis alle Bits, in unserem Fall 8 Stück befüllt sind.  
Ein Vorteil dieser Methode ist die konstante Konvertierungszeit:

Abbildung 1 Schaltung eines SAR

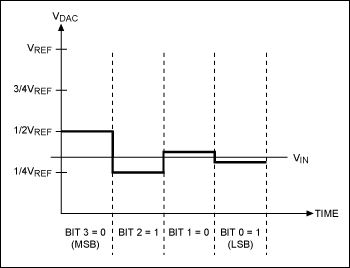


Abbildung 2 Wägeverfahren

# Inventarliste

|  |  |  |
| --- | --- | --- |
| **Stück** | **Gerätebezeichnung** | **Inventarnummer** |
| 1 | Spannungsversorgung Laborplatz |  |
| 1 | Logikanalysator |  |
| 1 | HTL Uno |  |

# Übungsdurchführung:

## Beantwortung der Kontrollfragen

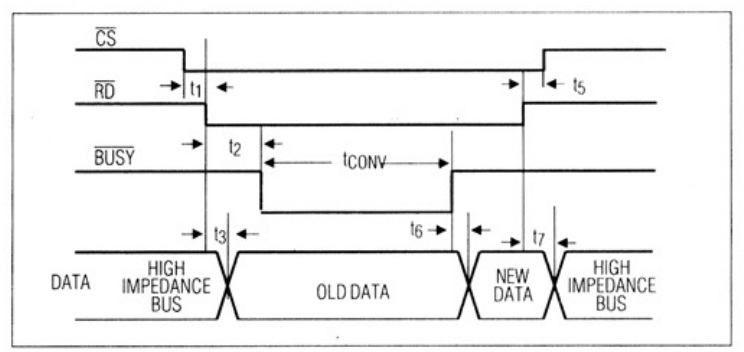
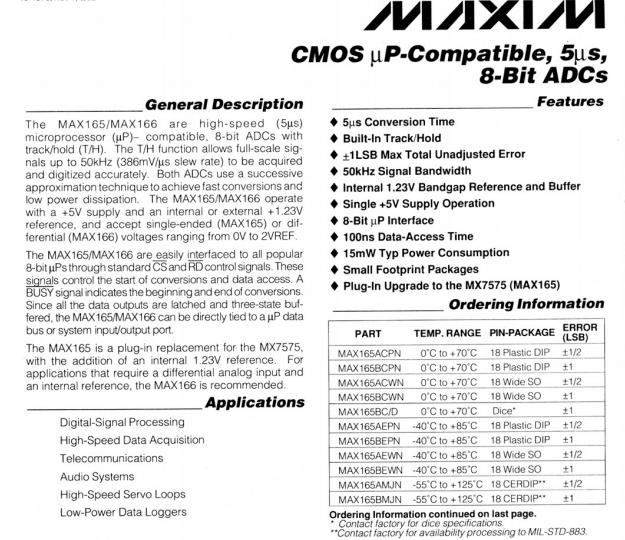
1. Welcher Messbereich ist für die Analogspannung anzuwenden?   
   A: Der Messbereich ist von 0V- 2\*Vref.

Abbildung 3 Timing Diagramm aus dem Datenblatt

1. Wie ist die Taktfrequenz einzustellen?  
   A: Mit einem RC Glied.
2. Mit welchen Signalen wird die Wandlung gesteuert?   
   A: Chip-Select (CS), Read  
   Mit der Chip-Select Leitung wählt man einen IC aus den man ansprechen will. Mit der Read Leitung startet man einen Lese Befehl. Diese beiden Signale müssen laut Abbildung 2 auf LOW sein um einen Lese Befehl zu schicken und den Umwandlungsvorgang zu starten. Solange das Busy Signal auf LOW ist befindet sich der ADC im Wandlungsverfahren.
3. Welche Auflösung hat der ADC?  
   A: 8-Bit.
4. Wie groß ist die conversion time?  
   A: 5µs.
5. Wie groß ist ULSB?  
   A: ULSB beträgt = .

**e)**

**d)  
  
 a)**

**c)**

Abbildung 4 Datenblattauszug

## Schaltungsentwurf ADC

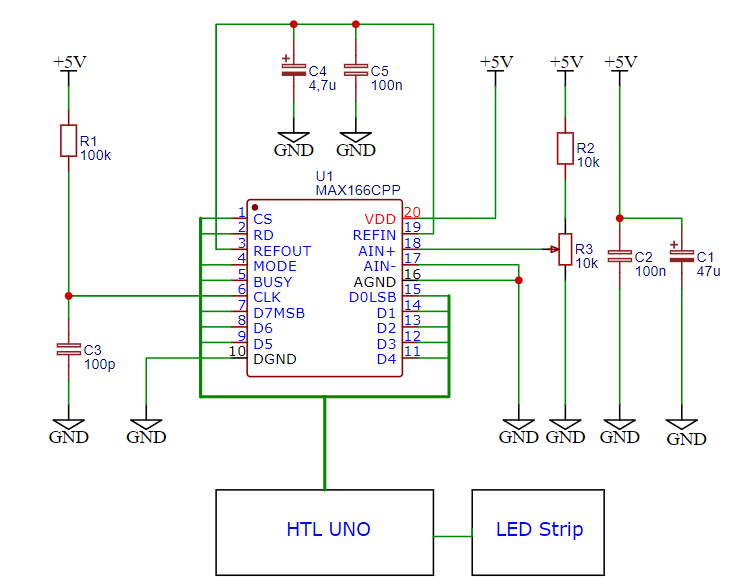


Abbildung 5 Schaltplan

## Messung am ADC

Aufgrund einen unerklärlichen Fehlers der im MODE 0 auftrat, konnten wir zu diesem Thema keine korrekte Messung aufzeichnen. Deswegen wird nur der MODE 1 beschrieben.

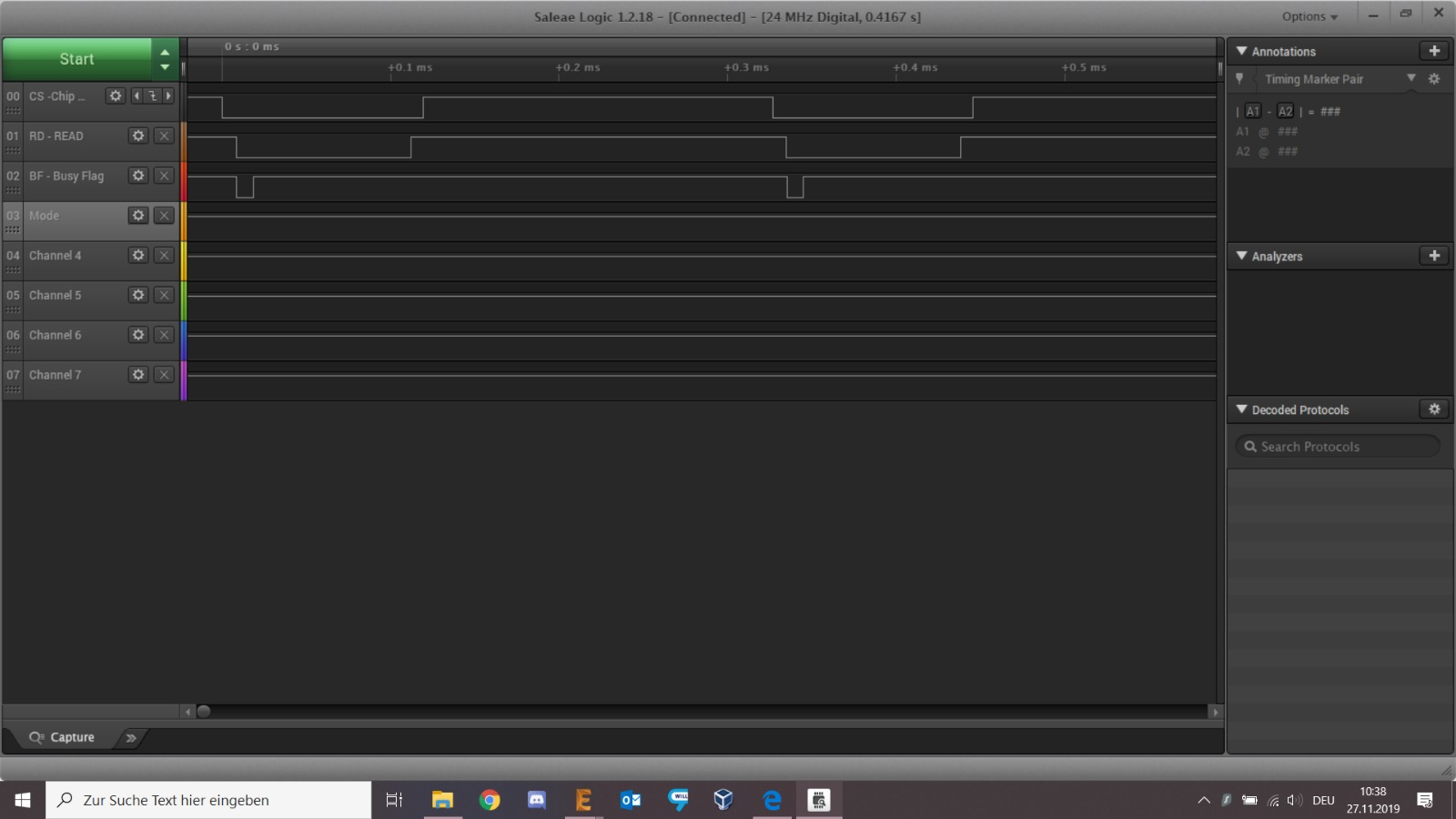
Die zwei verschiedenen Modis kann man durch einen Pin umschalten. Ist dieser auf LOW (Ground) wird der ADC in Mode 0 betrieben. Ist der Pin auf HIGH (VDD) wird der ADC im Mode 1 betrieben.  
Im Mode 0 sollte der ADC kontionierlich den Analogwert auswerten und wird mit dem Read Befehl nur ausgelesen. Im Mode 1 wartet der ADC auf den Read Befehl des µC und beginnt erst danach seine Auswertung.  
Man erkennt das die gemessenen Signale mit dem Timing Diagramm des Datenblattes übereinstimmen. Nach dem das Chip-Select und das Read Signal auf LOW gezogen werden beginnt der ADC mit seiner Wandlung und setzt das Busy Flag Signal auf LOW. Nach dem der ADC fertig gewandelt hat zieht er das Busy Signal wieder auf HIGH und die Daten können ausgelesen werden. Nach dem der µC die gemessenen Daten verarbeitet hat setzt er die Chip-Select und die Read Leitung wieder auf HIGH.

Abbildung 6 Logik-Analyzer Messung für den Mode 1



Abbildung 7 Z in Abhängigkeit von Ue

Aus Abbildung 7 erkennt man, dass der Ausgangswert linear zur Eingangsspannung ist. Die sichtbaren Schwankungen enstanden da das genaue Ablesen eines Wertes nicht möglich war da der digitale Ausgangswert nie konstant auf einem Wert blieb.

Die gemessene Spannung an Pin Refout beträgt 1.22V und bleibt konstant bei veränderlichen Ain.

Programcode des Arduinos:

//Pin definition:  
#define CS 2

#define RD 3

#define BU 4

//Inkludieren der LED Strip Libary

#include <Adafruit\_NeoPixel.h>

//Erstellen eines LED Strip Objektes

Adafruit\_NeoPixel pixel(11, 5, NEO\_GRB + NEO\_KHZ800);

void setup() {

//Chip-Select und Read auf Output setzen. Busy auf Input setzen.

pinMode(CS, OUTPUT);

pinMode(RD, OUTPUT);

pinMode(BU, INPUT);

//LED Strip starten und die Helligkeit einstellen.

pixel.begin();

pixel.setBrightness(30);

}

void loop() {

//CS und Read laut Timing Diagram auf LOW Setzten.

digitalWrite(CS, LOW);

delayMicroseconds(5);

digitalWrite(RD, LOW);

delayMicroseconds(15);

//Die 8 LEDs auf dem LED Strip auf die entsprechenden Werte der Eingangssignale setzten.

pixel.clear();

for(int i = 0; i<8; i++){

pixel.setPixelColor(i, pixel.Color(0, 150\*(int)digitalRead(6+i), 0));

}

pixel.show();

//CS und Read wieder auf HIGH ziehen um das Ende zu signalisieren.

digitalWrite(RD, HIGH);

delayMicroseconds(5);

digitalWrite(CS, HIGH);

delay(10);

}

Unterschrift:

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