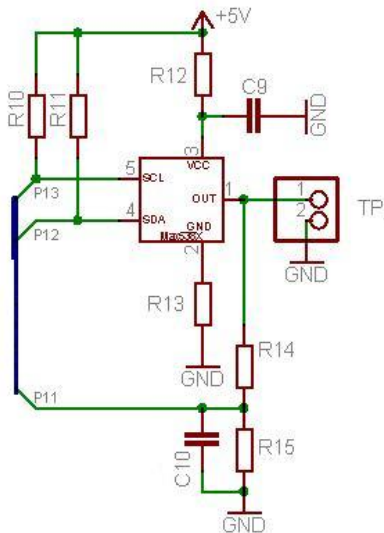


## 8.0 Erweiterung mit dem 8 Bit D/A Maxim MAX5382

Der MAX5382 ist ein 8 Bit DA Wandler im SOT23 Gehäuse. Der MAX5380 besitzt eine interne Referenzspannung von 2 V, der MAX5381 von 4 Volt und der verwendete MAX5382 von  $0.9 \cdot$  Betriebsspannung. Die Betriebsspannungen sind entsprechend unterschiedlich. Mit 5 Pins (SCL, SDA, Vcc, Gnd und Analog out) wird das Layout sehr einfach. Mit mehreren dieser Bausteine (und paralleler I2C Ansteuerung mit unterschiedlichen Adressen) könnten Threshold, Verstärkung und Offset erzeugt werden.

### 8.1 Der MAX5382 DA Wandler in der Schaltung



Der Maxim DA Wandler wird über die I2C Schnittstelle (eine Zweidrahtverbindung mit P1.2 und P1.3) gesteuert. Nur eine 8 Bit Kommunikation wird unterstützt.

Die Widerstände R12 und R13 sind Drahtbrücken. R12 kann dazu dienen um zusammen mit C9 ggf. Störeinflüsse zu filtern. R13 um ggf. zu Testzwecken das GND Potential zu verändern, aber Vorsicht, in diesem Fall ändert sich auch das Potential für die I2C Eingänge.

Über den Teiler R14 und R15, die ebenfalls als konventionelle Bauelemente ausgelegt sind, wird der Ausgang des DA Wandlers mit dem Eingang des Komparators des AT89LPx052 verbunden.

Zusammen mit dem Komparator kann eine Analog / Digital Wandlung nach dem Zählverfahren bzw. Kompensationsverfahren erfolgen. Bei der Zählmethode wird das analoge Ausgangssignal des D/A Wandlers solange erhöht und an den Komparator geliefert, bis der Wert gleich oder größer der angelegten analogen Eingangsspannung (P1.0 oder Testpunkt TP\_AI) ist. Die Umsetzungszeit ist abhängig von der

Eingangsgröße, im ungünstigsten Fall muss der Zähler alle Stufen durchlaufen.

*Hinweis: Der Widerstand R\_37\_RV, nur im kompletten Schaltbild dargestellt, wird nicht benötigt.*

## **8.2 MAX5382 DA Wandler und I<sup>2</sup>C**

Der I<sup>2</sup>C-Bus basiert auf zwei Leitungen:

SCL ist das Taktsignal. Dieses wird ausschließlich vom Busmaster erzeugt und von den angeschlossenen Chips gelesen, die Übertragungsrichtung ist daher eindeutig festgelegt.

SDA überträgt die Datenbits. Über diesen Anschluss senden sowohl Master als auch Slave ihre Daten.

Es gibt eine Reihe genau definierter Situationen die den gesamten Ablauf vom Ruhezustand über Start, Datenübermittlung Datenbestätigung und Stop zwischen Master und Slave regeln. Der Zustand des auf SDA angelegten Datenbits darf z.B. nur dann wechseln, wenn SCL=0 ist. Bei SCL=1 sind die angelegten Daten grundsätzlich gültig.

Der Max D/A Wandler kann nur als I<sup>2</sup>C Slave betrieben werden. Ein Lesen des D/A Wandlers über I<sup>2</sup>C ist nicht notwendig und daher vereinfacht sich die Programmierung auf wenige Zeilen. Nach der Startsequenz und Übermittlung der 7bit Adresse wird der 8 bit DAC Code und zuletzt die I<sup>2</sup>C Stop Bedingung übermittelt. Natürlich muss das Acknowledge Signal des MAX5382 nach der Übermittlung des Adressbytes bzw. Datenbytes überprüft werden.

Das letzte Adressbit, das eigentliche Bit um bei I<sup>2</sup>C zwischen Schreiben und Lesen umzuschalten, wird im Übrigen für den Powerdown Mode verwendet.

Der eingesetzte Maxim Baustein 5382MEUK besitzt die feste Adresse 62H. Die SDA und SCL Signale sind durch das Ghost Hardware Design mit 92H bzw. 93H vorgegeben.

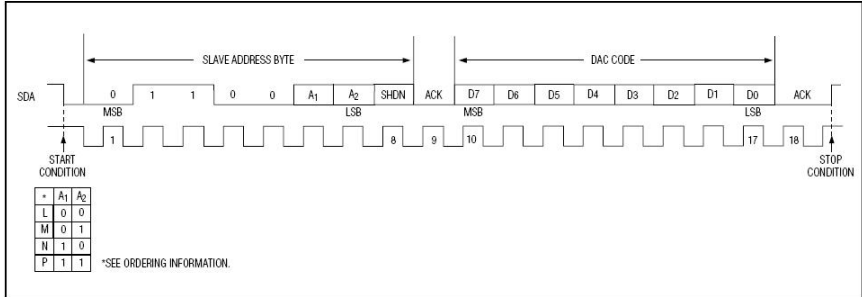


Figure 6. A Complete Serial Transmission

Ein Auszug aus den Maxim Spezifikationen

### 8.3 I<sup>2</sup>C Assembler Programm für den MAX5382 DA Wandler

Im Folgenden ist ein Assembler Programm aufgeführt, das im Ergebnis zu einem dreiecksförmigen Ausgangssignal am Testpunkt TP führt.

Die I<sup>2</sup>C Unterprogramme I2Out , I2Stop, I2Stop wurden ohne Änderungen aus dem Buch Basiskurs Mikrocontroller von B.Kainka übernommen.

```

; TASM
; Serial port access if you want
; to modify I2C DA values, but not used
; in Main
; 11,059MHz, 19200 Baud (or change SMOD1)
#include LPx052.H

SDA    .equ  92h          ; Port 1.2
SCL    .equ  93h          ; Port 1.3
.org 0000H
INIT   mov  P1M0,#00H    ; set ports
       mov  P1M1,#00H
       mov  P3M0,#00H
       mov  P3M1,#00H
       clr  TR1          ; stop T1
       mov  TH1,#0DCH    ; 9600 baud
       mov  TL1,#0DCH    ;
       anl  TMOD,#00H    ; auto-reload
       orl  TMOD,#20H    ; Mode 0
       setb TR1          ; start timer
       anl  PCON,#3FH    ;
       mov  SCON,#50H    ; init UART
       setb TI
       orl  PCON,#80H    ; double Baudrate 19,2K

```

```

        acall I2Init          ; init

; MAX 5382 MEUK : adr=62,
; SCL max 400 khz = 2,5 us
; settling time 20 us

MAIN    acall I2Start
        mov   a,#62H        ; I2C adr
        acall I2Out         ; A= 0 = error
        mov   a,R1          ; I2C data
        acall I2Out
        acall I2Stop
        mov   r2,#50        ; DA settling time
loops   djnz  R2,loops      ; R2 * 90ns * 3cyc
        dec  R1             ; next value
        sjmp MAIN

Delay   nop                 ; command delay
        nop
        nop
        nop
        nop
        nop
        ret

I2Init  setb  SDA
        setb  SCL
        ret

I2Start clr   SDA
        acall Delay
        clr   SCL
        ret

I2Stop  clr   SCL
        clr   SDA
        acall Delay
        setb  SCL
        acall Delay
        setb  SDA
        ret

I2Out   mov   r3,#8         ; 8 bits
S0      jb   ACC.7,S1       ; bit 7 = 1?
        clr   SDA          ; bit = 0
        sjmp S2            ; bit = 1
S1      setb          ; bit = 1
S2      acall Delay
        setb  SCL          ; clock
        acall Delay
        clr   SCL
        rl   a              ; next bit in A
        djnz r3,S0         ; 8 bits

```

```

        setb  SDA            ; SDA high Z
        acall Delay
        setb  SCL            ; clock 9
        acall Delay
        jb    SDA,Err        ; Ack?
        clr   SCL
        clr   SDA
        acall Delay
        ret
Err     mov   A,#0          ; error: return 0
        clr   SCL
        clr   SDA
        acall Delay
        ret

        ; ser com
RX      jnb   RI,RX        ; get from user
        mov   A,SBUF
        clr   RI
        ret

TX      jnb   TI,TX        ; commit value
        clr   TI
        mov   SBUF,A
        ret
        .end

```

Die Daten in R1 werden in der Hauptroutine in einer Endlosschleife an die Adresse 62H ausgegeben. Mit jeder Schleife wird R1 einfach um 1 reduziert und damit auch das DA Wandler Ausgangssignal um ein Bit verändert. Würde man R1 in der Schleife um eins erhöhen, dann würde eine fallende Flanke des Signals zur steigenden Flanke werden.