

13.0 Ein Low Cost R-C A/D Wandler

In diesem Kapitel möchte ich eine alternative Möglichkeit zur Messung der Stromaufnahme des Elliptecmotors zeigen. Die Idee dazu kam mir aufgrund eines Applikationshinweises von Atmel, die ein Voltmeter mit einem At89Cx051 beschrieb.

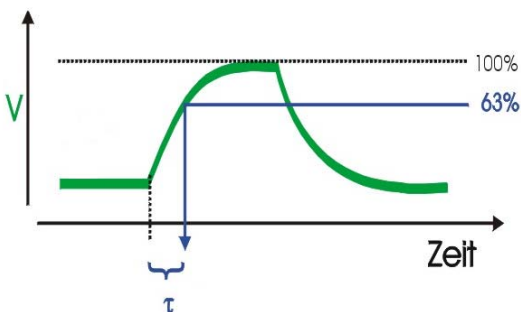
Beim A/D Wandler nach dem Zählverfahren mit dem Max5382 D/A Wandler wird die zu messende Eingangsspannung auf den einen Eingang eines Komparators gelegt. Die Spannung am anderen Eingang wird durch den D/A Wandler bitweise erhöht bis der Komparator am Ausgang kippt. Ein einfaches Verfahren, aber ein A/D Wandler oder D/A Wandler ist immer mit Mehrkosten verbunden.

Nun ja, für den Elliptecmotor kann man natürlich mit einem hochwertigen AD Wandler die Ströme am genauesten messen. Bei den Resonanzfrequenzen des Schwingkreises (Elliptecmotor/ Spule) ist der Strombedarf fast am geringsten. Es sollte für einfache Applikationen, in Zusammenhang mit dem Sweepen, doch eine einfache Messung des relativen Stromverlaufs ausreichend sein ohne absolute Werte mit höchster Genauigkeit messen zu müssen.

13.1 Ein R-C Glied und ein μP als A/D Wandler

Ist die Ladezeit eines Kondensators eine Alternative zum A/D bzw. D/A Wandler in Zusammenarbeit mit einem Mikrocontroller und dem Elliptecmotor?

Verbindet man eine Kapazität über einen Widerstand mit einer Versorgungsspannung, dann steigt die Spannung am Kondensator nach einer e-Funktion vom Anfangswert bis zum Endwert an.



Die Ladezeit ist nur von den Größen des Kondensators C und des Widerstandes R abhängig. Die Dauer der Ladezeit ist um so länger, je größer die Kapazität C des

Kondensators und je größer der Widerstand R ist. Das Produkt aus Kondensator C und Widerstand R ist als Zeitkonstante τ (tau) festgelegt.

Während der Dauer der Zeitkonstanten von einem τ nimmt die Spannung jeweils um das 0,63 fache des bis zum Endwert fehlenden Restes zu.

Nach einem τ beträgt die Spannung am Kondensator 63 % der Eingangsspannung, nach 2τ 86 %..... nach 5τ liegt nahezu die volle Spannung am Kondensator an.

Der Spannungsverlauf V_C am Kondensator entspricht der Gleichung:
 $V_C = V * (1 - e^{-\tau/RC})$.

V ist dabei die Eingangsspannung. Die Einheiten sind V und V_C in Volt, τ in Sekunden, R in Ohm und C in Farad.

Die Eulersche Zahl e, mit dem Wert 2,71828..., wurde nach dem Mathematiker Leonhard Euler benannt. Die Eulersche Zahl ergibt sich aus einer Grenzwertbetrachtung und ist die Basis des natürlichen Logarithmus. e ist eine irrationale und transzendente Zahl. Sie lässt sich weder als Bruch zweier natürlicher Zahlen noch als Lösung einer algebraischen Gleichung endlichen Grades darstellen.

Würde man aus dem Ladezustand einer Kapazität und einem Mikrocontroller einen genauen A/D Wandler aufbauen müsste man im Mikroprozessor eine Fließkomma Arithmetik für die e-Funktion einbauen. Um dieses zu umgehen könnte man in der Programmierung eine so genannte Lookup Table verwenden, die bei einer 8 Bit Auflösung die möglichen 256 Ergebnisse vom Mikrocontroller entsprechend dem Ergebnis der Exponentialfunktion zuordnet. Zudem müssten der Widerstand und der Kondensator von hoher Qualität sein.

Der Stromverlauf für die unterschiedlichen Betriebsfrequenzen des Elliptecmotors ist geprägt durch die minimalen Stromaufnahmen bei den Resonanzfrequenzen und durch die umgebende Flankensteilheit. Den Stromverlauf beim Ladevorgang eines Kondensators bis zur Hälfte der Versorgungsspannung nehmen wir einfach als linear an. Es sollen für die Steuerung des Elliptecmotors keine absoluten Messungen mit hoher Genauigkeit gemacht werden.

Verbindet man den Kondensator mit dem Eingang des Komparators (anstelle des D/A Wandlers) und die Versorgungsspannung wird über einen Widerstand einem Port Pin zugeführt, dann kann man die

Spannung am Kondensator damit ein- und ausschalten und den Ladevorgang- bzw. Entladevorgang starten.

Zum Ladestartzeitpunkt kann man einen internen 16 Bit Zähler des Mikrocontrollers starten, der gestoppt wird, wenn der Komparator kippt. Der Zählerzustand ist dann ein (nicht ganz korrektes) Maß für die Spannung am anderen Eingang des Komparators.

Ein 16 Bit Zähler des Atmel AT89LP2052 zählt von 0 bis zu 65635, multipliziert mal 90 ns Taktzeit (11Mhz Taktfrequenz), innerhalb von 0,00589 Sekunden; also in ungefähr 6 ms. Wir wollen nur im ersten Bereich der Ladezeit messen. Dann entspricht 6 ms ca. einem τ . $C = \tau/R$: daraus folgt bei einer Kapazität von 10 nF ein Widerstand von 600 kOhm, oder bei 100nF einem Widerstandswert von 60KOhm.

In diesem Fall würde die Spannung am Kondensator bei 63% Ladekapazität (einem τ), schon ca. 3 Volt betragen. Über den Shuntwiderstand am Elliptecmotor R19 (0,5Ohm) fallen bei maximal 1 A Stromverbrauch aber nur 0,5 Volt an, die damit dann bereits nach ca. einer ms erreicht wären. Man würde nur in einem kleinen Bereich arbeiten. Das wäre dann eine sehr schnelle Messung.

Würde man die Zeitkonstante von etwa $6 * 6ms$ wählen, dann werden 0,5 Volt nach ca. 6ms erreicht. Bei einer Kapazität von 100 nF kann der Widerstand auf entsprechend 360kOhm vergrößert werden. Diese Messung wäre entsprechend einiges langsamer.

Nun ja, für 1024 Zählvorgänge (10 bit) würden $1024 * 90ns$ anfallen. Dazu im Assembler Code über eine Lookup Table etwas umrechnen und der Wandler würde nach 100 μs ein Ergebnis liefern und hätte eine Entladezeit von $5 * 100\mu s$ für die nächste Messung. Das ist nicht unbedingt langsam, wenn Sie dieses mal mit anderen A/D Wandlern vergleichen.

Würde man einen hochwertigen Widerstand und Kondensator entsprechender Güte verwenden, und die Zählergebnisse über die serielle Schnittstelle übermitteln, dann könnte auch ein PC die Berechnung der e-Funktion übernehmen.

Die Ergebnisse wurden nur überschlagen. Die Schwellwertigenschaften des Komparators und die Berechnung der Exponentialfunktion im Assembler Programm blieben ebenfalls unberücksichtigt.

Nach dem Laden und Ermittlung des Zählerstandes muss der Kondensator wieder entladen werden. Das dauert entsprechend der Ladezeit und kann ggf. verkürzt werden, indem man den Komparator Port Pin als Ausgang schaltet, auf Low legt und wieder umschaltet. Laut Atmel Datenblatt soll der Komparator nach einer Umschaltung innerhalb von $10\mu\text{s}$ stabil sein.

13.2 Hardware und Assembler zum R-C μP A/D Wandler

Und nun zur Hardware und zum Programm zur Überprüfung des Wandlers.

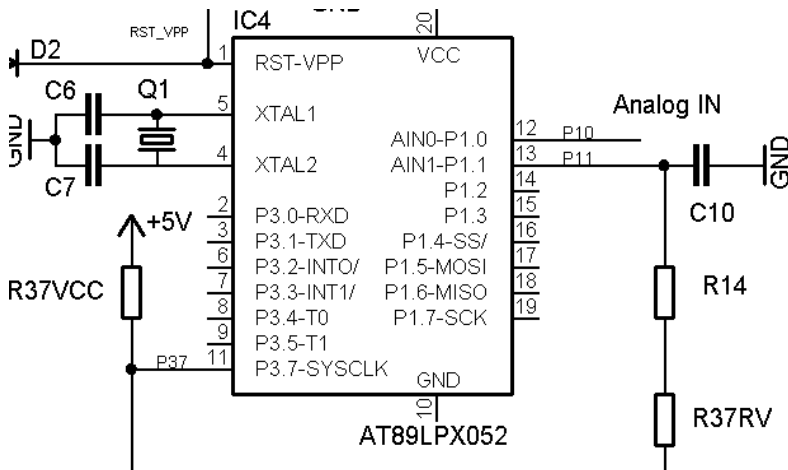


Abbildung Hardware R-C Wandler

Das Schaltbild wurde zur Verdeutlichung ein wenig abgeändert. R37VCC ist ein Pull up mit $5.1\text{k}\Omega$, R37RV wurde mit $220\text{k}\Omega$ eingesetzt, R 14 ($2\text{k}\Omega$) blieb im Design. C10 wurde mit 100nF eingelötet.

R9, im obigen Bild nicht sichtbar, muss ggf. entfernt werden, sofern Sie diesen nachträglich bestückt haben. R 15, einer der Widerstandsteiler des Max D/A Wandlers, muss ebenfalls entfernt werden.

Das zu messende Eingangssignal liegt an Port 1.0 (P10).

Die System Clock Fuse des AT89LP2052 muss deaktiviert sein, da sonst an P3.7 ein Ausgangssignal liefert.

Mit $\tau=(R37RV+R14)*C10 = 222*10^3 * 100*10^{-9}$ ergeben sich maximal 22ms für eine Messung bis 63% der Betriebsspannung an C10. Der maximale 16 bit Zählerstand würde bis dahin zwar mehrmals überschritten werden, aber es sollen ja nur die ersten 0,5 Volt ermittelt werden.

Mit dem folgenden Assemblerprogramm wurden die Möglichkeiten der D/A Wandlung überprüft:

```
#include LPx052.H
.org 0000H

start:  mov  P1M0,#03H      ; set ports to quasi bi-direc
        mov  P1M1,#00H      ; and P1.0 /P1.1 to input only
        mov  P3M0,#00H
        mov  P3M1,#10H      ; push pull output for P3.4
        mov  SP,#20H        ; Stack pointer
        clr  TR1            ; stop timer 1
        clr  TR0            ; stop timer 0
        mov  TH1,#0DCH      ; 256-6: 9600 baud
        mov  TL1,#0DCH      ; 11.059MHz for SMOD1 =0
        anl  TMOD,#00H      ; Timer1: 8 bit auto-reload
        orl  TMOD,#20H      ; Timer 1 Mode
        setb TR1            ; TCON start timer 1
        anl  PCON,#3FH      ; PCON: clear SMOD0 and SMOD0
        mov  SCON,#50H      ; InitsRS232 8 bit UART Model
        setb TI
        orl  PCON,#80H      ; SMOD=1 double Baudrate Timer 0
        mov  RL0,#00H      ; set first RL and RH
        mov  RH0,#00H      ; default settings
        orl  TMOD,#1        ; 16 bit autoreload with 0
        orl  ACSR,#08H      ; activate comparator inputs
        clr  P3.4           ; to reduce FET power consumption
        clr  P3.7           ; clear 100n CAP
NEXT    acall RX            ; get start user
        nop
        acall TX            ; confirm same value to user
        mov  TH0,#00H      ; clear Hb counter 0
        mov  TL0,#00H      ; clear Lb counter 0
        setb P3.7           ; load CAP
        setb TR0            ; start timer 0 by Software
CWAIT  jnb  P3.6,sendval   ; poll comparator P3.6
        sjmp CWAIT         ; reaches input voltage
sendval clr  TR0            ; stop timer 0
        clr  P3.7           ; clear 100n CAP
        mov  A,TH0          ; Hb
        acall TX            ; send Hb to user
        mov  A,TL0          ; Lb
        acall TX            ; send Lb to user
        sjmp NEXT          ; wait for user
```

```

RX      jnb    RI,RX      ; get value from user
        mov    A,SBUF
        clr    RI
        ret
TX      jnb    TI,TX      ; committ value
        clr    TI
        mov    SBUF,A
        ret
        .end

```

Das Programm erwartet ein Startsignal über die serielle Schnittstelle, setzt dann P3.7 des AT89LP2052 und startet den Timer 0. Der Zustand an P3.6, dem Ausgang des Komparators, wird einfach durch Pollen abgefragt. Hier hätte man ggf. auch mit einem Interrupt arbeiten können.

Überschreitet die Kondensatorspannung die Eingangsspannung wird der Timer gestoppt, die Entladung des Kondensators eingeleitet und der Zählerzustand über die serielle Schnittstelle übermittelt.

Das VB Programm, bzw. das VBA Programm dazu ist einfach:

```

Dim dummy As Integer
Dim result As Long

dummy = -1
Me.Ergebnis.Value = 0

SENDBYTE (1)
Text1.Value = READBYTE

While dummy = -1
    dummy = READBYTE
Wend

Me.Text1.Value = dummy
result = dummy * 256

Me.Text2.Value = READBYTE
result = result + Val(Me.Text2.Value)

Me.Ergebnis.Value = result

```

Hinweis: In dieser VB Programmdarstellung fehlt das Modul für die serielle Kommunikation.

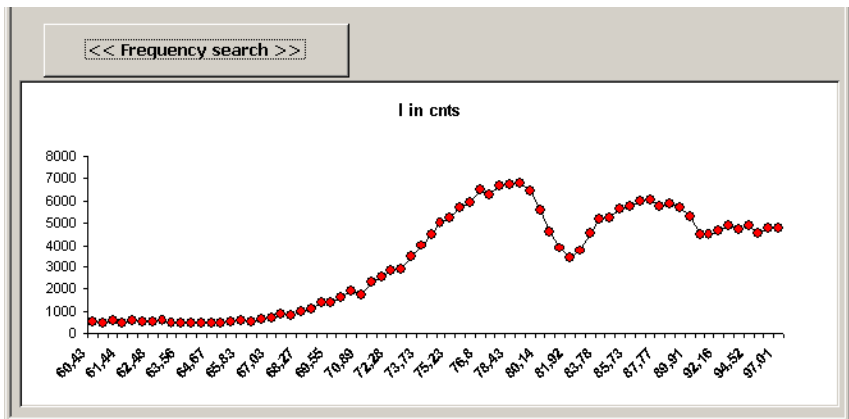
In einem ersten Versuch wurden mit einem Atmel AT89LP2052 Evaluierungsmuster, mit 11.052 MHz Taktfrequenz, ab 25 mV Eingangssignal die ersten Ergebnisse des Zählers ermittelt. Bei 0,5 Volt erreichte der Zählerstand ca. 21.500. d.h. 0,5 Volt wurden in etwa in 2 ms erreicht.

13.3 Der R-C μ P A/D Wandler im Einsatz mit dem Elliptecmotor

Im Gegensatz zum Schaltbild in Kapitel 13.2 wurde beim Einsatz dieser R-C A/D Wandlung mit dem Elliptecmotor ein Widerstand von 100KOhm, statt 220KOhm, verwendet um die Wandlergeschwindigkeit zu erhöhen.

Die Access Benutzeroberfläche wurde ein wenig modifiziert, um die gewandelten 16bit Werte (vorher 8bit des Maxim D/A Wandlers) der A/D Wandlung vom Mikrocontroller zu erhalten.

Das folgende Bild zeigt Ihnen die Ergebnisse der D/A Wandlung für die Vorwärtsbewegung:



Wie auch bei der D/A Wandlung mit dem Max Wandler wurden die Werte für eine Frequenz dreimal hintereinander ermittelt und dabei jedes Mal der maximale Zählerwert für den dargestellten Verlauf festgehalten. Auch in diesem Fall ist, im Vergleich zur Messung nach dem Zählverfahren mit dem Max D/A Wandler, bei der Vorwärtsbewegung die minimale Stromaufnahme bei ca. 81kHz deutlich ausgeprägt.

Im Mikrocontrollerprogramm wurde Timer 1 sowohl für die serielle Schnittstelle als auch für die A/D Wandlung benutzt, da Timer 0 für eine interrupt gesteuerte Erzeugung der Betriebsfrequenzen des Elliptecmotors verwendet wurde.

Dazu ein kurzer Auszug aus dem Assemblerprogramm mit einigen Erläuterungen:

Timer 1 umprogrammieren von serieller Kommunikation auf 16 bit counter einstellen:

```

    clr    TR1
    mov    RL1,#00H    ; set first RL and RH
    mov    RH1,#00H    ;
    anl    TMOD,#0FH   ; Timer1, clr
    orl    TMOD,#10H   ; 16 bit autoreload Timer 1
    mov    TH1,#00H   ; clear Hbyte counter 1
    mov    TL1,#00H   ; clear Lbyte counter 1

```

Laden der Kapazität und dann Timer starten:

```

    setb   P3.7        ; load CAP
    setb   TR1        ; start timer 1 for C_DA

```

Warten bis der Komparator kippt:

```

WAIT    jnb    P3.6,sendval ; poll comparator P3.6
                                ; wait until CAP Voltage
                                ; reaches input voltage
    sjmp   WAIT
    clr    TR1        ; stop timer 1
    clr    P3.7        ; clear 100n CAP

```

Schnelleres Entladen des Kondensators indem der Port 1.1 temporär auf 0 gesetzt wird:

```

    anl    ACSR,#00H   ; de-activate comparator
    mov    P1M0,#00H   ; set ports to quasi bi-direc
    mov    P1M1,#00H   ;
    clr    P1.1        ; clear Port pin
    mov    R7,#100     ; wait
looppr  djnz   R7,looppr ; R7 * 90ns * 3cyc
    mov    P1M0,#03H   ; set P1.0 / back to input
    mov    P1M1,#00H   ; and P1.0 /P1.1 to input only
    orl    ACSR,#08H   ; activate comparator inputs

```

Settling Time des Komparators abwarten:

```

looppc  mov    R7,#100
    djnz   R7,looppc  ; R7 * 90ns * 3cyc
    mov    R4,TH1     ; save TH1

```



```

        mov     R5,TL1           ; save TL1

Timer 1 wieder für serielle Kommunikation umstellen und Daten
an User übermitteln:

setser  mov     TH1,#0DCH       ; 256-6: 9600 baud
        mov     TL1,#0DCH       ; 11.059MHz for SMOD1 =0
        anl    TMOD,#0FH       ; Timer1: 8 bit auto-reload
        orl    TMOD,#20H
        setb   TR1             ; TCON start timer 1 for ser com
        setb   TI
        mov    A,R4             ; Highbyte
        acall  TX               ; send Hb to user
        mov    A,R5             ; Lowbyte
        acall  TX               ; send Lb to user
        sjmp  NEXT             ; .....

```

13.4 Den R-C μ P A/D Wandler linearisieren

In Kapitel 13.1 wurde das Ladeverhalten eines Kondensators mit einem Widerstand bei konstanter Ladespannung beschrieben.

Die Ladezeit ist von den Größen des Kondensators C und des Widerstandes R abhängig. Das Produkt aus Kondensator C und Widerstand R ist als Zeitkonstante τ (tau) festgelegt.

Während der Dauer der Zeitkonstanten von einem τ nimmt die Spannung jeweils um das 0,63 fache des bis zum Endwert fehlenden Restes zu.

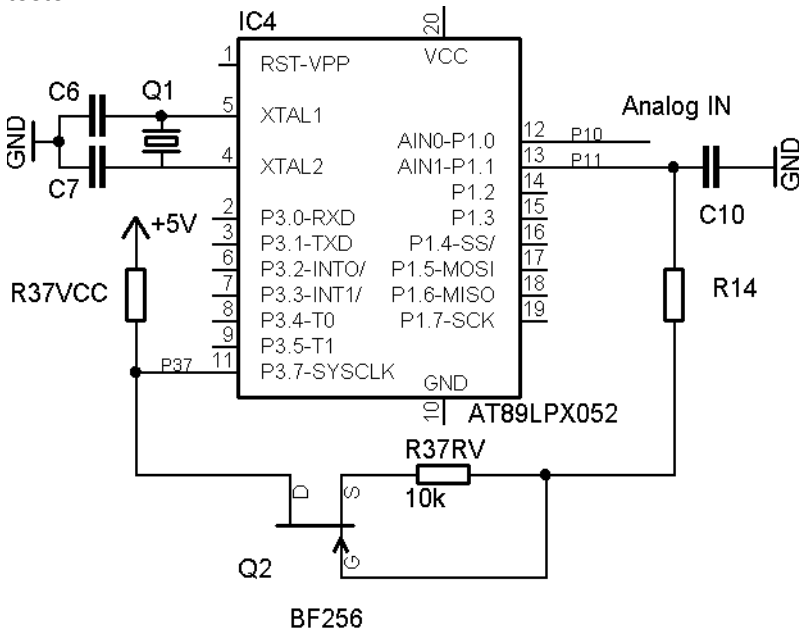
Nach einem τ beträgt die Spannung am Kondensator 63 % der Eingangsspannung. Der Spannungsverlauf V_C am Kondensator entspricht der Gleichung: $V_C = V * (1 - e^{-(\tau/RC)})$.

V ist dabei die Eingangsspannung. Die Einheiten sind V und V_C in Volt, τ in Sekunden, R in Ohm und C in Farad.

Wird ein Kondensator mit einem konstanten Ladestrom geladen steigt die Ladung q des Kondensators linear in Abhängigkeit zur Zeit t an: $q=I*t$. Dabei sind I der konstante Strom in A und t die Zeit in Sekunden. Die Spannung am Kondensator U_c ist $U_c =q/C$ und beide Formeln zusammengefasst ergeben: $I*t = C* U_c$. Daraus ergibt sich für U_c ein linearer Spannungsverlauf am Kondensator $U_c=I/c*t$.

Mit einer Konstantstromquelle wird sich der R-C A/D Wandler mit höherer Genauigkeit auch für andere Zwecke einsetzen lassen. Eine einfache Stromquelle lässt sich mit einem FET umsetzen. Dieser wird einfach in Reihe zur Strom führenden Leitung eingefügt. Ein FET

ist auf dem Ghost Experimentier Board nicht vorgesehen. Die Schaltung lässt sich aber über den Expansion Bus aufbauen und testen.



Es kann das gleiche Testprogramm wie in Kapitel 13.2 verwendet werden. Der konstante Strom ermittelt sich aus

$$I = U_{R37RV} / R37RV$$

Mit einem zusätzlichen Trimmer in Reihe zu R37RV könnte ein Voltmeter abgeglichen werden. Mit dem Einschwingverhalten des Mikrocontroller Ports P37 wird der untere Bereich ungenauer sein. Durch Verändern des konstanten Ladestroms oder durch verändern des Ladekondensators kann der Messbereich verändert werden. Die tatsächliche Eingangsspannung an Analog IN kann durch multiplizieren oder dividieren mit dem aktuellen Zählerstand umgerechnet werden. Dabei ist die Taktfrequenz des Mikrocontrollers zu berücksichtigen.

Für 0-2 Volt Eingangsspannung, einem konstanten Strom von 1 mA und 100 nF Kondensator (C10) werden 2Volt am Kondensator nach ($t = c * U_0 / I = 100 * 10^{-9} * 2 / (1 * 10^{-3})$) 200 μ s erreicht. Bei 10Mhz Taktfrequenz des Atmel Mikrocontrollers, bzw. 100ns Zykluszeit, also nach 2000 Zählertakten. In diesem Fall wäre jeder Zähltakt mit 1 mV gleichzusetzen, also keine Umrechnung erforderlich.