Kapazitätsmessung mit dem PC – einfach und schnell

DIETER STOTZ

Als Anwendung der Druckerport-Programmierung zeigen wir hier eine sehr einfache Methode, Kapazitäten zu messen. Man benötigt bei herkömmlichen unidirektionalen Schnittstellen mit CMOS-Technologie außer den zu messenden Kondensator prinzipiell nur noch einen Widerstand und natürlich einen 25poligen Sub-D-Stecker für die Parallelschnittstelle. Bei den neueren bidirektionalen Ausführungen steigt der Aufwand um einen CMOS-Baustein, einen Transistor und einen Widerstand an.

Sofern man voraussetzen kann, daß die meisten Leute, die heute im weitesten Sinne mit Technik etwas zu tun haben, einen PC besitzen, ist eine sehr einfache Kapazitätsmessung möglich, die außer dem PC nur noch einen Widerstand und einen 25poligen D-Stecker benötigt. Außerdem braucht man natürlich noch Software, die unter DOS läuft.

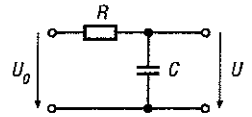


Bild 1: Anordnung und Parameter beim Ladevorgang

Das im folgenden beschriebene Verfahren der Kapazitätsmessung kommt ohne zusätzliche Meßkarten oder dergleichen aus; dennoch zeigt es alle Vorzüge einer computergestützten Messung: Das Resultat kann für beliebige Zwecke ausgewertet werden (z.B. für Sortier- oder Selektionsverfahren).

Mit einer entsprechenden Vorrichtung können natürlich auch ganze Serien von Kondensatoren ausgemessen werden. Neben der Kapazitätsmessung ist selbstverständlich auch eine Widerstandsmessung möglich, wenn man den Kondensator als feste Referenz kennt.

■ Funktionsweise des Verfahrens

Zur Messung einer Kapazität gibt es viele verschiedene Verfahren mit ebenso situationsbedingten Vor- und Nachteilen. So kann eine Kapazität mittels Wechselspannung über eine Meßbrücke oder auch einfach über den Blindstrom bestimmt werden.

Verfahren, die Gleichspannung einsetzen, beruhen meist auf dem Messen von Ladezeiten. Auch die hier vorgestellte Methode legt an den zu bestimmenden Kondensator eine Gleichspannung an und mißt dann die Zeit bis zum Erreichen einer bestimmten Spannung. Genauer gesagt soll die Zeit bis zum Erreichen der halben Meßspannung ermittelt werden.

Bei einer Anordnung nach Bild 1 ergibt sich für die laufende Spannung:

$$U = U_0 \cdot (1 - e^{-t/RC}). \quad (1)$$

Für $U = {}^1/_2 \cdot U_0$, also Aufladung auf halbe Endspannung, gilt nach Auflösung von Gleichung (1):

$$t = RC \cdot \ln 2. \quad (2)$$

Nach dieser Gleichung ist die Ladezeit völlig proportional zu R und C. Man müßte also lediglich, wie in Bild 1, über einen Widerstand eine (vorher entladene) Kapazität aufladen und das Erreichen der halben Endspannung, z.B. mittels Komparator, überprüfen. Vom Anlegen der Spannung bis zum Kippen des Komparators inkrementiert ein Zähler, der mithin über seinen Endstand eine der Ladezeit proportionale Größe vermitteln kann.

Nun ist ja allgemein bekannt, daß CMOS-Gatter etwa bei Erreichen der halben Betriebsspannung ihren Ausgangszustand kippen lassen. Abweichungen hiervon sind ebenfalls nicht allzu tragisch; lediglich Gleichung (2) wird dann nicht mehr hinreichend genau erfüllt, aber die Proportionalität ist dennoch gewährleistet – und darauf kommt es letztlich an. Eine Forderung jedoch besteht für die Schwellenspannung: sie sollte eine recht geringe Temperaturdrift haben, was i.a. erfüllt wird.

Es wird also bei Beginn der Messung an den Ladewiderstand und das andere Ende des Kondensators Spannung gelegt und gleichzeitig ein Zähler gestartet. Bei Erreichen der halben Spannung wird der Zähler gestoppt, und der Zählerstand gelangt zur Weiterverarbeitung.

Ein Assembler-Programm (oder besser gesagt: ein Maschinenprogramm) vermag diese Aufgabe mit der nötigen Geschwindigkeit und ohne Störungen und Unterbrechungen von außen zu lösen. Denn es leuchtet ein, daß kleinere Kapazitäten extrem kurze Ladezeiten verursachen, welche nur durch einen schnellen Zähler aufgelöst werden können.

Die Messung wird über die Parallelschnittstelle durchgeführt – sie läßt sich auf geeignete Weise programmieren und kann mit der nötigen Geschwindigkeit gesteuert werden.

Diese Funktionsweise macht auch klar, daß die Meßzeit unmittelbar von der Größe der Kapazität abhängig ist; große Kondensatoren benötigen deshalb auch eine längere Zeit.

■ Aufbau der Meßanordnung

Bild 2 zeigt uns die einfache Schaltung für die Messung von Kapazitäten. Der Ladewiderstand sollte auf 100 kΩ festgelegt werden; damit können Werte ab ca. 1000 pF gemessen werden.

Die Meßanschlüsse sollten möglichst kurz gehalten werden. Läßt sich dies nicht bewerkstelligen, muß eine abgeschirmte Leitung zum Prüfling führen, wobei allerdings zu bedenken ist, daß die Abschirmung selbst bereits eine Grundkapazität aufweist, die in Abzug zu bringen ist.

Man muß hier allerdings ergänzen, daß die vorliegende Schaltung nur für ältere unidirektionale mit CMOS-Eingängen geeignet ist. Neue Standards sehen einen bidirektionalen Betrieb vor, bei dem erstens die Ausgänge nur mit nicht vernachlässigbarer Restspannung auf Low gehen und – was noch wichtiger ist – die Eingänge sind über einen mehr oder weniger niederohmigen Pullup-Widerstand auf High gezogen. Hier hilft nur eine Puffer- bzw. Treiberschaltung nach Bild 3.

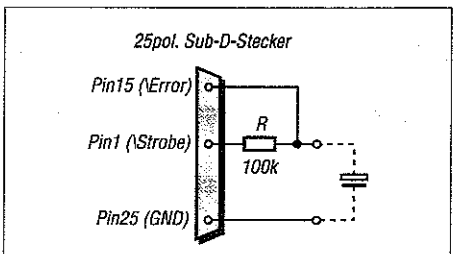


Bild 2: Meßanordnung zum Anschluß an die Parallelschnittstelle

Die Steuerspannung von Pin 1 der Schnittstelle wird zunächst gepuffert, um vor allem die Low-Schwelle annähernd auf Null zu legen. Am Ausgang des oberen Gatters liegt dann wieder das für die Messung relevante RC-Glied. Zur Abtastung der Ladespannung kommt ein zweites Gatter zum Einsatz, welches einen PNP-Transistor ansteuert, der auch niederohmige Pullups der Schnittstelle (z.B. 1 kΩ) bedienen kann. Damit auch diese erweiterte Schaltung kompatibel zu herkömmlichen unidirektionalen Schnittstellen ist, wird jedoch ein gesonderter Arbeitswiderstand von der Größe 10 kΩ nötig.

■ Genauigkeit und Meßbereiche

Die gesamte Schaltung erhält die Versorgungsspannung von Pin 14 der Schnittstelle, also ebenfalls einem Ausgang.

Die Genauigkeit hängt von verschiedenen Faktoren ab, nicht zuletzt auch von der Auflösung, d.h. von derjenigen Kapazität, bei der der Zähler nur noch bis eins zählt. Dieses kleinste Auflösungsquantum wiederum ist abhängig von der Geschwindigkeit des Rechners und natürlich vom Ladewiderstand.
Macht man letzteren größer, so wächst damit auch der Zählerstand und somit die Meßzeit bei der gleichen zu messenden Kapazität. Es ist aber aus Gründen der Störeinstrahlung usw. nicht empfehlenswert, den Ladewiderstand über 1 MΩ zu wählen.

Die Temperaturdrift der Schaltschwelle beeinflußt ebenfalls die Genauigkeit. Hier verhalten sich jedoch moderne CMOS-Gatter äußerst günstig, d.h., die Temperaturdrift ist (im Vergleich zu anderen Genauigkeitsfaktoren) zu vernachlässigen.
Möchte man dennoch die Temperaturdrift der Schwellenspannung kompensieren, so ließe sich untenstehendes Programm auch erweitern, indem man nicht nur eine Ladezeit von Null bis zur Schwelle, sondern auch eine Entladezeit von ca. 5 V bis zur Schwelle mißt und beide Werte mittelt.
Bewegt sich die Drift nicht allzuweit vom Potentialmittelpunkt weg, so ergibt sich eine recht gute Kompensation. Vorausgesetzt, die Meßanordnung befindet sich in kalibriertem Zustand (siehe unten), so läßt sich eine Genauigkeit von ca. ±5 % erwarten, was selbst noch für kleine Kapazitäten bis herab auf 1000 pF zutreffend ist (bei einem 486er DX 33).
Wie oben bereits erwähnt, verlängert sich die Meßzeit mit der zu bestimmenden Kapazität. Diesem Effekt kann man mit einer Verkleinerung des Ladewiderstandes begegnen. Das läßt sich jedoch nur in begrenztem Maße durchführen, da der Ausgang der Parallelschnittstelle keine zu vernachlässigende Impedanz aufweist, die man

```
uses crt,CMess;

var
  fehler              : integer;
  messwert            : real;
  cap_unit,r_unit,t_unit    : string;

begin
  messwert:=GetCap(fehler);
  cap_unit:=' pF';
  r_unit:=' Ohm';
  t_unit:=' µs';
  if messwert>=1000 then begin
    messwert:=messwert/1000;
    cap_unit:=' nF';
    r_unit:=' kOhm';
    t_unit:=' ms';
  end;
  if messwert>=1000 then begin
    messwert:=messwert/1000;
    cap_unit:=' µF';
    r_unit:=' MOhm';
    t_unit:=' s';
  end;

  if fehler=1 then begin
    writeln('Zu messende Kapazität ist zu groß oder hat Kurzschluß!');
    sound(1000);
    delay(100);
    nosound;
  end
  else begin
    writeln('C = ',messwert:1:2,cap_unit,' (bei R=100 kOhm)');
    writeln('R = ',messwert:1:2,r_unit,' (bei C=0,1 µF)');
    writeln('RC = ',0.1*messwert:1:3,t_unit);
  end;
end.


(*
Unit zum Messen von Kapazitäten über die Parallelschnittstelle.

Autor: Dieter Stotz
Programmsprache: TurboPascal 6.0
*)


UNIT CMess;

INTERFACE

uses crt,dos;

var
  perf                : real;
  err                 : integer;
  env1                : string;

function GetCap (var errflag:integer): real;

IMPLEMENTATION

function GetCap;

var
  LSWord,MSWord             : word;
  cap                       : longint;
  capreal                   : real;
  errf                      : integer;

begin
  MSWord:=0;
  errf:=0;

  asm                                   (* Initialisierung *)
    CLI
    MOV BX,0
    MOV CX,0

  @Discharge:                           (* Kapazität zunächst entladen *)
    DEC CX
    MOV DX,$37A  {Kontrollport-Adresse initialisieren}

    MOV AL,1  {L-Pegel auf \STROBE (Pin 1)}
    OUT DX,AL
    JCXZ @Discharge

    AND AL,$FE  {H-Pegel auf \STROBE (Pin 1)}
    OUT DX,AL

    MOV DX,$379  {Inputport-Adresse initialisieren}
    MOV CX,0  {Zeitzähler auf Null setzen}

  @Measure:                             (* Messung *)
    IN  AL,DX  {Abfrage des Inputport}
    INC CX  {Zeitzähler inkrementieren}
    JZ  @Carry  {Sprung bei Überlauf nach @Carry}
    AND AL,8  {Ist Pin 15 auf H-Potential}
    JZ  @Measure  {Falls nein, Messung weiterlaufen lassen}

    DEC CX  {Zeitzähler dekrementieren (Korrektur)}
    MOV [LSWord],CX  {CX ist niederwertiges Wort}

    MOV DX,$37A  {Kontrollport-Adresse initialisieren}
    MOV AL,1  {L-Pegel auf \STROBE (Pin 1)}
    OUT DX,AL

    JMP  @End  {Sprung nach End}

  @Carry:          (* Höherwertiges Wort ermitteln *)
    INC BX  {BX inkrementieren}
    CMP BH,$7F
    JG  @Error  {Fehlerbehandlung bei Negativwert}
    JMP  @Measure  {Messung weiterlaufen lassen}

  @Error:
    MOV  [errf],1  {Errorflag setzen}
         (* Kapazität entladen *)
    MOV  DX,$37A  {Kontrollport-Adresse initialisieren}
    MOV  AL,1  {L-Pegel auf \STROBE (Pin 1)}
    OUT  DX,AL

  @End:
    MOV  [MSWord],BX  {BX ist höherwertiges Wort}
  end;

  errflag:=errf;
  cap:=MSWord;
  cap:=cap shl 16;
  cap:=cap + LSWord;
  capreal:=cap*perf;
  GetCap:=capreal;
end;

begin
  env1:=getenv('perf');
  val(env1,perf,err);
  if (err<>0) or (env1='') then perf:=20;
end.
```

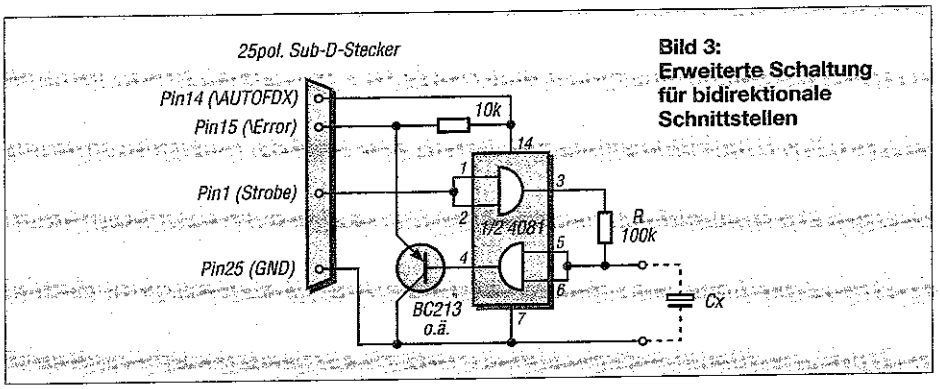


Bild 3: Erweiterte Schaltung für bidirektionale Schnittstellen

auf wenige Kiloohm schätzen kann. Effektiv kommt man mit dem Ladewiderstand deshalb i.a. nicht unter ca. 5 kΩ, selbst wenn man den physischen Ladewiderstand auf Null setzt.

Um also eine Verzehnfachung des Bereichs zu erzielen (mit einem Zehntel der Anzeige bei konstanter Kapazität), ist der externe Ladewiderstand nicht einfach von 100 kΩ auf 10 kΩ zu ändern, sondern auf einen Wert darunter, der die Gatterimpedanz berücksichtigt.

Am besten wird der Widerstandswert empirisch ermittelt, indem nach der Messung mit dem 100-kΩ-Widerstand zunächst ein Potentiometer mit 10 kΩ eingesetzt und dieses so lange verkleinert wird, bis der angezeigte Wert auf ein Zehntel abgefallen ist. Das Poti kann nun durch einen entsprechenden Festwiderstand ersetzt werden.

Eine derartige „Bereichsumschaltung" sollte am besten durch die Software abgefragt und ausgewertet werden, damit der Anzeigewert entsprechend korrigiert bzw. wieder verzehnfacht werden kann.

Bei der Messung von Elektrolytkondensatoren muß auf die Polarität nach Bild 2 bzw. Bild 3 geachtet werden. Die Meßspannung beträgt systembedingt maximal 5 V; allerdings bricht ja im Normalfall bereits nach dem Erreichen einer Spannung von 2,5 V am Kondensator die Meßspannung wieder zusammen.

Mit dem Programm lassen sich natürlich auch Widerstände bestimmen, wenn man dafür die Kapazität fest auf den Wert 0,1 µF setzt. Sowohl für die R- als auch die C-Messung muß die jeweils andere Größe definiert sein. Es können aber auch RC-Zeitkonstanten direkt angezeigt werden; hierzu dürfen R und C beliebig variieren (innerhalb gewisser Meßgrenzen).

Kalibrierung

Die Kalibrierung geschieht prinzipiell dadurch, daß der Zählerwert, der vom Assembler-Programm übergeben wird, mit einem Faktor multipliziert wird, welcher an den Rechner bzw. seine Geschwindigkeit angepaßt werden muß.

Man benötigt eine bekannte Kapazität, die möglichst groß (0,1 µF bis 1 µF) und auch genau sein sollte – wenigstens sollte man den tatsächlichen Wert durch eine Meßbrücke vorher hinreichend genau bestimmt haben.

Bei der ersten Messung zeigt der Rechner zunächst einen falschen Wert, weil der Korrekturfaktor noch nicht stimmt. Dieser ist defaultmäßig auf 20 eingestellt, solange keine entsprechende DOS-Environment-Variable gesetzt ist. Ist der angezeigte Wert zu klein, muß die Variable genau um das Verhältnis Istwert/Anzeigewert vergrößert werden, ansonsten muß sie ebenfalls mit oben genanntem Verhältnis verkleinert werden.

Die Kalibrierung wird von einer DOS-Environment-Variable mit dem Namen *perf* übernommen. Unter DOS wird ihr über den SET-Befehl ein Wert zugeordnet; dies geschieht z.B. mit der Zeile:

set perf=20.5

Dies kann in jedem Verzeichnis nach dem DOS-Prompt geschehen; nach einem Neustart geht solch ein Eintrag verloren, es sei denn, man legt den Befehl in eine Stapeldatei, die vor dem Programmstart bereits durchlaufen wurde, z.B. die AUTOEXEC.BAT oder eine für das Programm spezifische Stapeldatei.

Der eingetragene Wert wird als Real-Wert interpretiert – als Dezimaltrennstelle ist ein Punkt zu setzen.

Software

Das Programm besteht aus zwei Teilen, nämlich einer Turbo-Pascal-Unit sowie einem kurzen Hauptprogramm, das die Anwendung der Unit demonstrieren soll. Die Unit ermöglicht die Einbindung in andere Programme. Sie beinhaltet die Funktion *GetCap*, die – bei korrekter Kalibrierung – den Kapazitätswert in pF zurückgibt. Ebenfalls wird ein Fehlerwert geliefert, der bei 0 Fehlerfreiheit und bei 1 Überlauf signalisiert.

Der Aufruf des Programms geschieht durch Eingabe von *RC* beim DOS-Prompt. Zu empfehlen ist ein einmaliger „Trockenstart", weil dann die Steuerspannung erst mal auf Low geht und die Kapazität somit genug Zeit hat für eine richtige Entladung.

Innerhalb des Assembler-Programms wird die Messung vorgenommen und das Zwischenergebnis auf eine DWord-Variable (Longint) gelegt. Dieser Wert wird mit dem Kalibrierungsfaktor *perf* multipliziert und als Funktionswert übergeben.

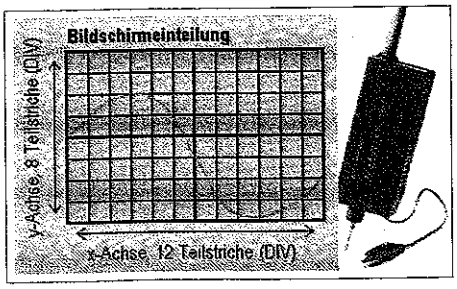
Am Ende der Messung wird Pin 15 der Parallelschnittstelle wieder auf L-Pegel gesetzt, damit sich die Kapazität wieder entlädt. Das gleiche geschieht vor jeder Messung für eine kurze Zeit. Diese Entladung ist für größere Kondensatoren natürlich nicht ausreichend – hier muß eine gesonderte schnelle Entladung für die notwendige Initialisierung sorgen.

Im Hauptteil der Unit wird die Environment-Variable abgefragt (falls vorhanden). Das Hauptprogramm ruft die Funktion auf und weist den erhaltenen Wert einer Variablen zu. Danach wird zunächst das Fehler-Flag behandelt. Bei Fehlerfreiheit kommt der ermittelte Wert zur Anzeige.

Wer keinen Compiler hat, kann das fertige ausführbare Programm per Download direkt vom Internet unter *http://www.funkamateur.de/download/* oder auch vom Autor (eMail: *stotzd@aol.com*) erhalten.

Handyprobe am LPT-Port

Ein Handyprobe ist ein kompaktes Meßgerät für periodische Spannungssignale, welches es auch zum Anschluß an einen PC-Parallelport gibt. Dabei wird das Meßmodul mit Hilfe einer Kabelverbindung an einen der am Rechner verfügbaren freien Parallelports angeschlossen. Die gesamte Steuerung der Meß- und Analysevorgänge erfolgt dann über die PC-Software.



Mittels einer derartigen Software kann ein solcher „Meßsignalabtaster" dann z.B. als Speicheroszilloskop (bis zu etwa 100 kHz Abtastfrequenz), als Spektralanalyzer, komfortables Digitalvoltmeter oder Transientenrecorder für Langzeitaufnahmen benutzt werden. Das interessierende Signal ist bequem über die Kontaktspitze am Modul zu messen.

Informationen, Preise und sonstige Angebote zu derartigen und anderen Meßgeräten sind z.B. im WWW auf *http://bitzer-digitaltechnik.de/* aufgelistet. **AE**