



Optische Messung der Herzfrequenz

Projektdurchführung: Sinan Aziret, Bernd Herrmann, Michael Scheible
Betreuung und redaktionelle Bearbeitung: Albrecht Kettler

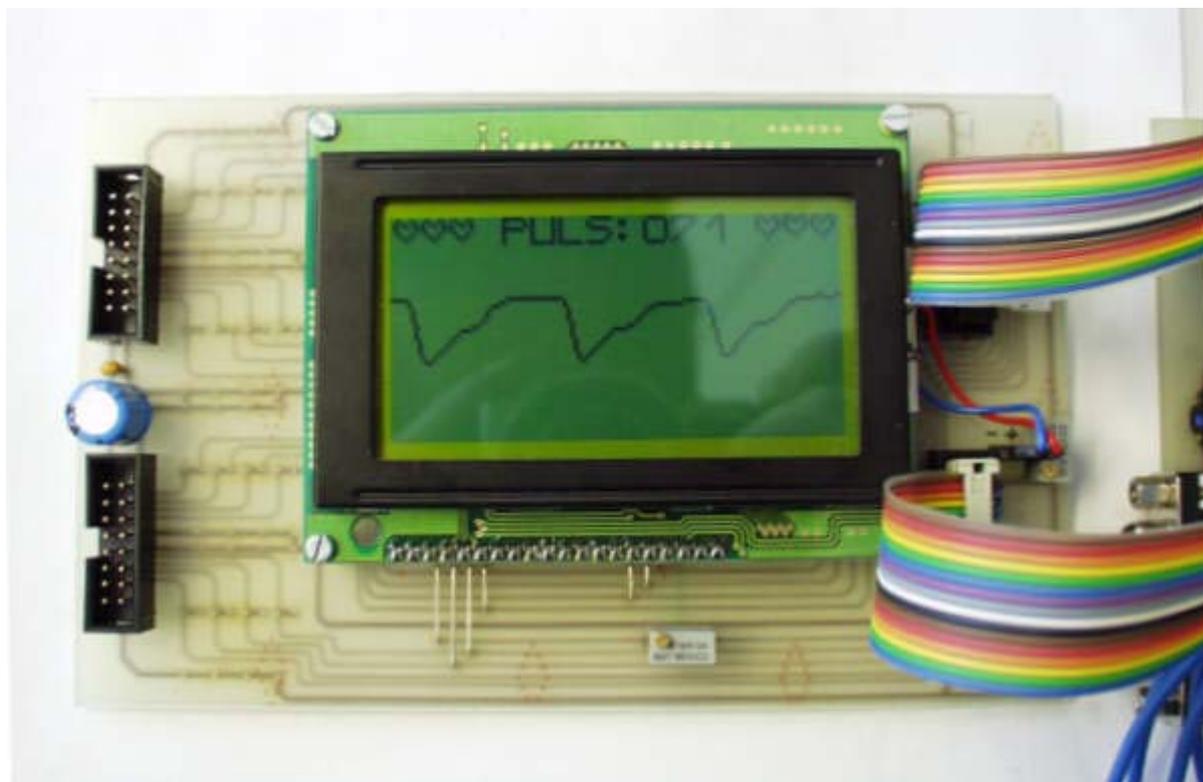


Abb. 1: Darstellung von Pulssignal und Pulsfrequenz auf dem LC-Display

Einleitung

Der menschliche Puls lässt sich optisch messen. Der mit dem Herzschlag variierende Blutdruck führt zu einer optisch meßbaren Veränderung des Blutgehalts der peripheren Gefäße. Dies lässt sich z.B. gut am Finger detektieren.

Uns standen zwei Projektarbeiten zur Verfügung, die so weit fortgeschritten waren, dass das optisch erfaßte Pulssignal auf einem LC-Display grafisch dargestellt werden konnte. Beide Arbeiten wurden mit einem Mikrocontroller PIC16C74 zur Erfassung und Visualisierung des Meßsignals realisiert. Die im Sommersemester 99 durchgeführten Arbeiten verfolgten zwei unterschiedliche Ansätze:

[Pulsmessung im Durchlicht](#)

Als Lichtquelle wurde eine rote, superhelle LED eingesetzt. Das den Finger durchdringende Licht wurde mit einem Licht-Frequenzwandler TSL230 detektiert. Die Periodendauer des Signal des TSL230 wurde unter Verwendung der Capture Einheit des Mikrocontrollers mit einer Genauigkeit von ca 15 Bit erfaßt. Durch digitale Filterung wurden 50Hz Störsignale unterdrückt und das Pulssignal extrahiert.

[Pulsmessung in Reflexion](#)

Zur Beleuchtung der Hautoberfläche wurde eine Infrarot LED eingesetzt. Das zurückgestreute Licht wurde mit einer PIN-Photodiode detektiert. Durch eine analoge Filterschaltung wurden Gleichlicht und Störsignale hoher Frequenz eliminiert. Das Pulssignal wurde durch den A/D Wandler des Mikrocontrollers mit 8 Bit digitalisiert.

Ziel der hier beschriebenen Projektarbeit war die Ermittlung der Pulsfrequenz und deren numerische Darstellung auf einem LC-Display zusammen mit der grafischen Darstellung des Pulssignals.

Anforderungen und Auswahl der Meßmethode

Anhand der Ergebnisse der hier beschriebenen Projektarbeit soll ein Ausstellungsobjekt entstehen, das die optische Messung der Herzfrequenz als Beispiel für optoelektronische Schaltungen in einem allgemein zugänglichen Schaukasten präsentiert. Folgende Anforderungen müssen daher erfüllt sein:

- Unempfindlich gegen Erschütterung und Unachtsamkeit durch Benutzer
- Kein Kontakt des Benutzers mit spannungsführenden Teilen
- Batteriebetrieb
- Gleiche Versorgungsspannung aller Komponenten
- Grafische Darstellung des Pulssignals geeignet für unterschiedliche Pulsamplituden
- Numerische Ausgabe der Pulsfrequenz auf Grafik Display
- Möglichkeit zum Anschluss eines PC's über die RS 232 Schnittstelle

Wir entschieden uns für die Methode der Pulsmessung in Reflexion, da der Aufbau mechanisch robuster gestaltet werden kann und zwischen Sensor und Finger eine isolierende Plexiglasscheibe angebracht werden kann.

Aufgabenverteilung

Folgende Aufgaben mussten gelöst werden:

Signalverarbeitung und Frequenzmessung

- Einlesen der Pulssignale
- Übertragung der unbearbeiteten Pulssignale an die LCD-Anzeige
- Übertragung der unbearbeiteten Pulssignale zum PC
- Detektion der Pulsflanken
- Berechnung der Pulsfrequenz

Diese Aufgabe wurde von Sinan Aziret und Bernd Herrmann durchgeführt.

Programmierung LCD-Ausgabe

- Wandlung der Binärzahl in eine Dezimalzahl
- Anschalten und Initialisierung des LCD
- Text- und Zahlenausgabe auf dem LCD

Diese Aufgabe wurde von Michael Scheible durchgeführt.

Hardware Blockdiagramm und Schnittstellenbeschreibung

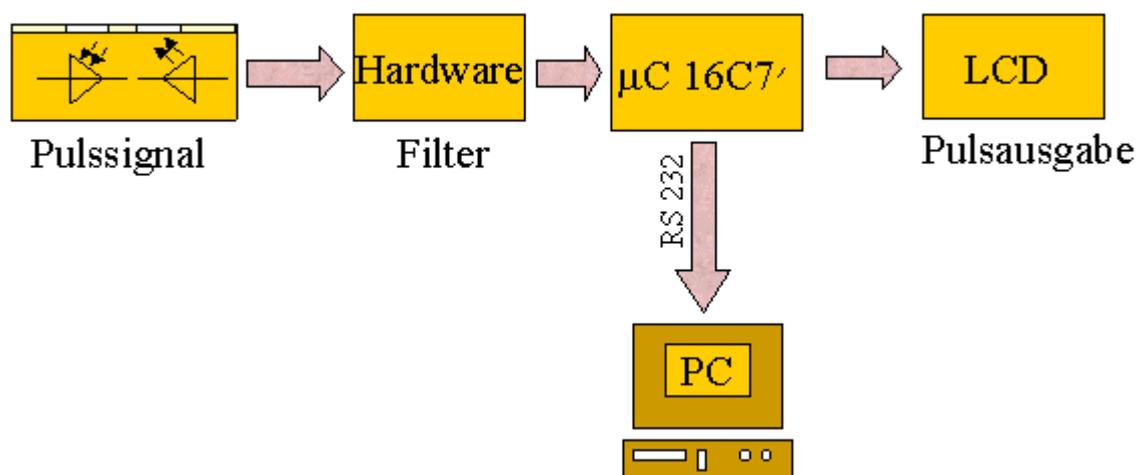


Abb. 2: Blockschaltbild der Hardware

An den Hardware Schnittstellen ergeben sich folgende Aufgaben

- Detektion des Pulssignals
 - die [Detektion des Pulssignals](#) erfolgt durch eine Anordnung aus IR-LED und PIN Photodiode
 - zur Verstärkung und Filterung des Detektorsignals wird eine [vorhandene Schaltung](#) verwendet
- Wandlung des analogen Pulssignals
 - Pulssignalpegel von 0V bis 5V
 - A/D-Werte zwischen 0 bis 255
 - A/D-Wandlerbereich von 0V bis 5V
 - A/D-Wandlung alle 10ms 100Hz Abtastfrequenz
- LCD Anzeige
 - kontinuierliche Übergabe der unbearbeiteten A/D-Werte an LCD über die Ports B und D
 - Übergabe des errechneten Pulswertes binär über einen 8-Bit Register [1/min]
 - Übergabe eines Pulswertes nur bei korrektem Pulssignal
- Datenübertragung PIC16C74 - PC
 - Übertragung über die serielle Schnittstelle (RS 232)
 - Übertragungsrate von 9600 bps (Baudrate)
 - Nur kontinuierliche Übertragung der unbearbeiteten A/D-Werte

Überlegungen zur Ermittlung der Pulsfrequenz

Das optisch erfasste Pulssignal ist zwar periodisch, unterliegt aber gewissen Schwankungen. Einerseits ist die Pulsfrequenz nicht völlig konstant, andererseits kommt es vor allem durch Bewegungen des Fingers zu Signalschwankungen. Die Puls-Amplitude kann von Versuchsperson zu Versuchsperson sehr unterschiedlich sein. Auch bei ein und derselben Person kommt es durch unterschiedliche Durchblutung bzw. unterschiedlich starkes Aufdrücken des Fingers zu deutlichen Änderungen der Signalamplitude.

Eine naheliegende Lösung zur Frequenzmessung wäre die Ermittlung der Grundfrequenz des Signals durch Fourieranalyse. Hierfür reicht jedoch die Rechenkapazität des Mikrocontrollers nicht aus. Es wurden daher Methoden gesucht, die mit weniger Rechenleistung auskommen. Die grundlegende Idee war, ausgezeichnete Punkte im Kurvenverlauf des Pulssignals zu erkennen. Hier bieten sich Maxima bzw. Minima oder Wendepunkte an.

Der Verlauf des Pulssignals ist schematisch in Abb. 3 dargestellt. (Auf dem LCD Display ist der Signalverlauf invertiert dargestellt.) Typisch ist vor allem der steile Anstieg mit einem klar ausgeprägten ersten Maximum, dem bei der fallenden Pulsflanke eventuell ein Nebenmaximum (bedingt durch die Kontraktion der 2.

Herzkammer) folgt.

Pulserkennung durch Detektion der Signalmaxima

Bei einer formelmäßig gegebenen Kurve $y=f(t)$ erhält man Maxima durch Suche nach Nullstellen der ersten Ableitung y' bei denen die zweite Ableitung y'' kleiner null ist. Ist der Verlauf einer Kurve durch diskrete Abtastwerte gegeben, tritt an die Stelle der Ableitung als Näherung der Differenzenquotient. Dieser wird, wenn die Meßkurve ein Maximum durchläuft, nur sehr selten exakt null werden. Ein besseres Kriterium ist daher der Nulldurchgang des Differenzenquotienten von positiven zu negativen Werten.

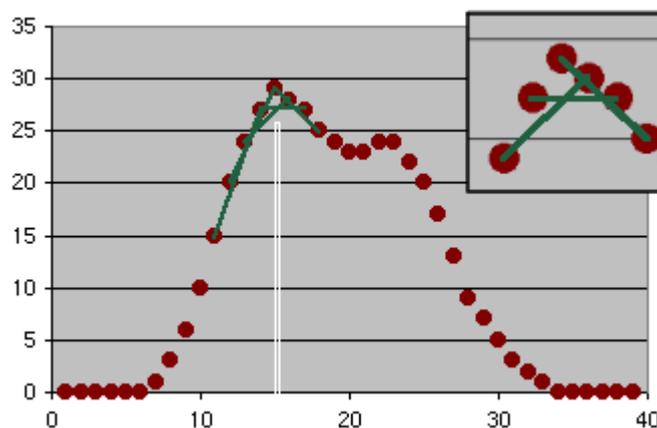


Abb.3: "Bestimmung der "1. Ableitung" über 3 Abtastintervalle

Zur Erhöhung der numerischen Genauigkeit kann es sinnvoll sein, die Differenz nicht aus unmittelbar aufeinanderfolgenden Werten sondern über mehrere Abtastintervalle hinweg zu ermitteln. Der Einfluß von Meßwertschwankungen durch Rauschen und Störsignale wird dadurch vermindert.

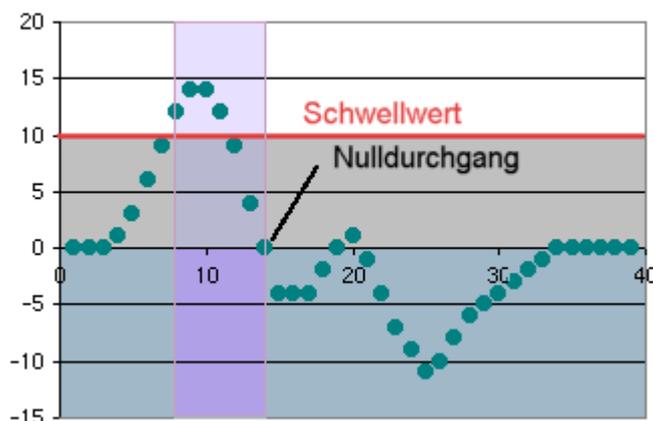
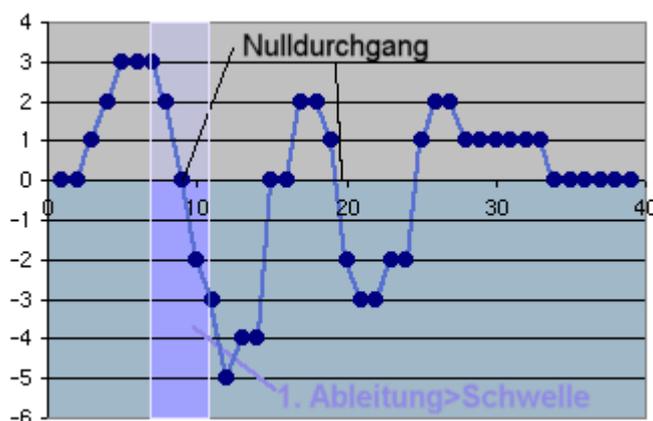


Abb. 4: Detektion des Maximums als Nulldurchgang der "1.Ableitung "

Um sicherzustellen, dass tatsächlich nur das Hauptmaximum ermittelt wird, benötigt man ein weiteres Kriterium. Zunächst wird gewartet bis der Differenzenquotient einen (empirisch zu ermittelnden) Schwellwert überschreitet. Anschließend wird der auf dieses Ereignis folgende Nulldurchgang des Differenzenquotienten abgewartet, der dann das Hauptmaximum darstellt.

Pulserkennung durch Detektion des Wendepunkts der ansteigenden Pulsflanke



Der Wendepunkt der ansteigenden Pulsflanke ist gekennzeichnet durch ein Maximum der ersten Ableitung, erkennbar an einem Nulldurchgang der zweiten Ableitung von positiven zu negativen Werten. Bei der praktischen Umsetzung tritt wieder der Differenzenquotient an die Stelle der Ableitung. Auch hier ist es sinnvoll den Differenzenquotienten erster Ordnung über mehrere Abtastintervalle hinweg zu bestimmen. Als Zusatzkriterium, um tatsächlich den Wendepunkt der steilen ansteigenden Pulsflanke zu erkennen, wird gefordert, daß die erste Ableitung (der Differenzenquotient) an der Stelle

Abb. 5: "2.Ableitung", Detektion des Wendepunkts der steilen Anstiegsflanke

des Wendepunkts größer ist als ein empirisch zu ermittelnder Schwellwert.

Da der Wendepunkt der ansteigenden Pulsflanke ein deutlich ausgeprägtes und konstantes Merkmal der Pulssignale unterschiedlichster Versuchspersonen darstellt, entschieden wir uns für die zweite Methode.

Ermittlung der Frequenz

Die Anzeige der Herzfrequenz soll in Schlägen pro Minute erfolgen. Eine Zählung über eine Minute hinweg würde zu lange dauern, daher ist es sinnvoller die Frequenz aus der Periodendauer zu errechnen. Um die numerische Genauigkeit zu erhöhen, werden zwei Signalperioden zur Frequenzberechnung verwendet. Dies ermöglicht zudem durch Vergleich zweier aufeinanderfolgend ermittelter Periodendauern eine einfache Plausibilitätskontrolle, ob überhaupt ein korrektes Pulssignal vorliegt.

Die Digitalisierung des Pulssignals erfolgt durch den internen A/D-Wandler des PIC16C74. Die A/D-Wandlung wird alle 10 ms gesteuert durch den 16-Bit Timer und die Compare Einheit des Mikrocontrollers ausgelöst. Die Grenzfrequenz der [Filter/Verstärkerschaltung](#) liegt bei etwa 40 Hz, sodass mit einer 100 Hz Abtastfrequenz das Shannonsche Abtasttheorem erfüllt ist.

Beschreibung der Software zur Ermittlung der Pulsfrequenz

Hauptprogramm

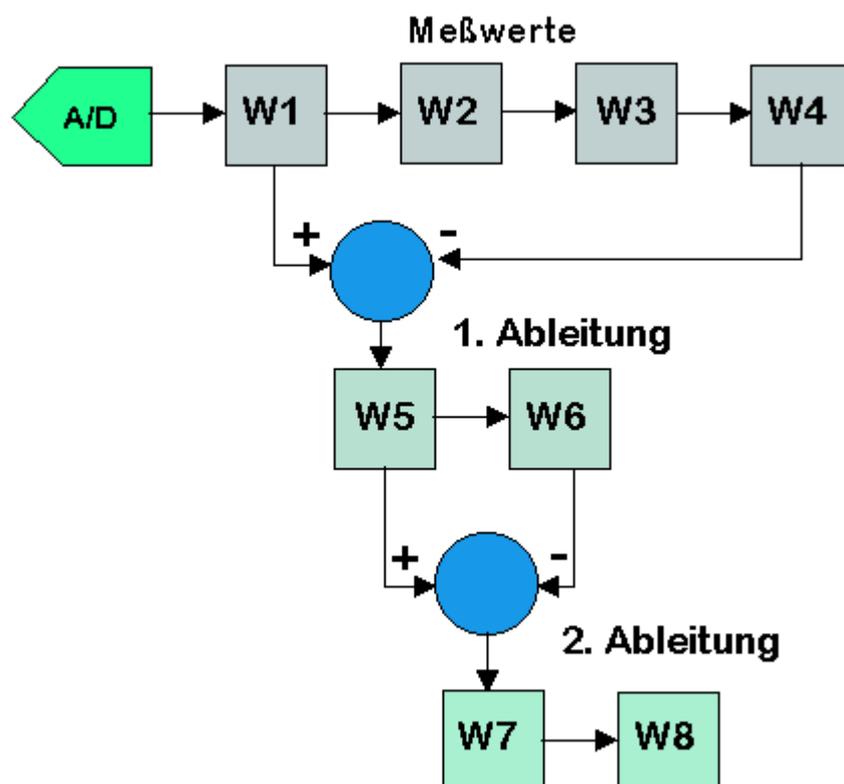
Initialisierung	
Btfss	Flag,AD_FLAG ;Warte bis A/D-Wandlung ausgeführt
call	LCD_Aus ;LCD-Ausgabe
call	Serial Init ;RS 232
call	Scann ;A/D-Werte einlesen und
call	Flad ;Flankendetektion
bcf	FLAG,AD_FLAG
Endlosschleife	

Das Einlesen der Analogwerte erfolgt in einer zeitgesteuerten Interruptroutine, die dem Hauptprogramm durch ein Flag-Bit anzeigt, wenn ein neuer Meßwert vorliegt. Der Meßwert wird zunächst an die LCD-Anzeigeroutine weitergegeben und grafisch dargestellt. Anschließend wird der Meßwert über die serielle Schnittstelle z.B. an einen PC weitergegeben. Im nächsten Unterprogramm werden Meßwerte in einem "Software-Schieberegister" zwischengespeichert und die erste und zweite "Ableitung" ermittelt. Aus den Ableitungen ermittelt

Abb. 6: Struktogramm des Hauptprogramms

die folgende
Unteroutine ob der
Wendepunkt der
ansteigenden
Pulsflanke erreicht
wurde.

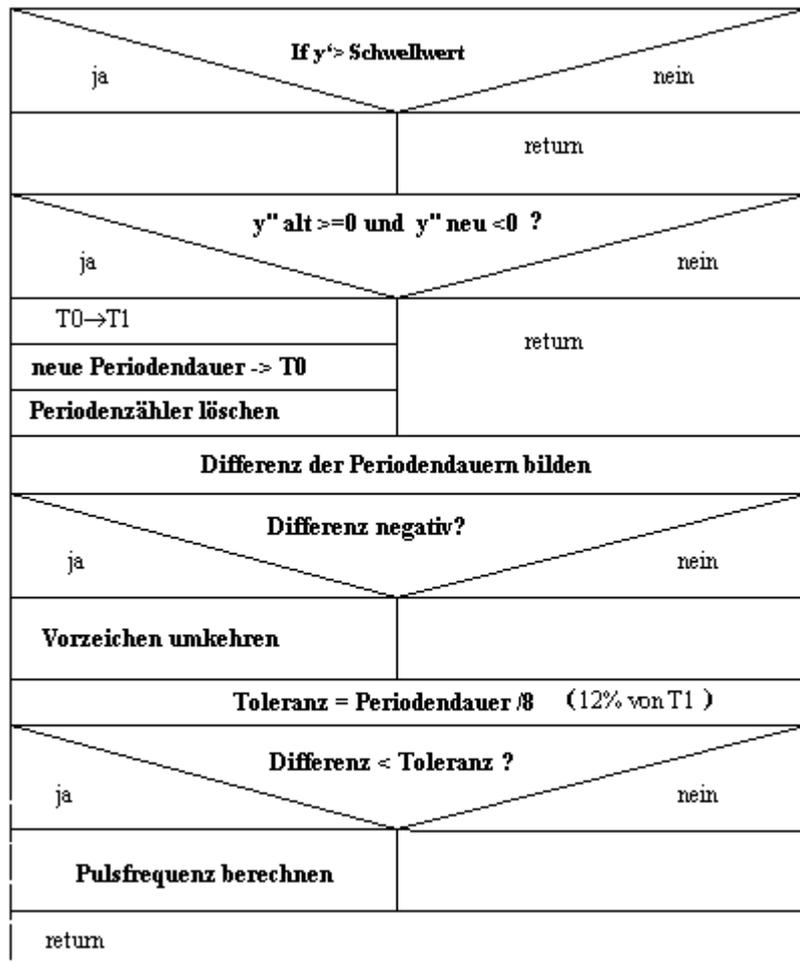
Unterprogramm Scann (A/D-Werte einlesen und y' , y'' bilden)



Die Differenz der A/D-Werte repräsentiert eine Näherung für die 1. Ableitung y' . Zur Erhöhung der numerischen Genauigkeit und zur Unterdrückung von Störungen bildet man die Differenz, wie oben gezeigt, über drei Abtastintervalle hinweg. Es müssen daher jeweils die letzten vier Meßwerte (W1 bis W4) gespeichert werden. Die zweite "Ableitung" wird aus zwei aufeinanderfolgenden Werten der ersten gebildet, daher müssen nur zwei Ableitungswerte (W5 und W6) gespeichert werden. Um einen Nulldurchgang der zweiten "Ableitung" feststellen zu können, benötigt man zwei aufeinanderfolgende Werte (W7 und W8) die folglich gespeichert werden.

Abb. 7: Signalfluß des Unterprogramms "scann"

Unterprogramm Flankendetektion



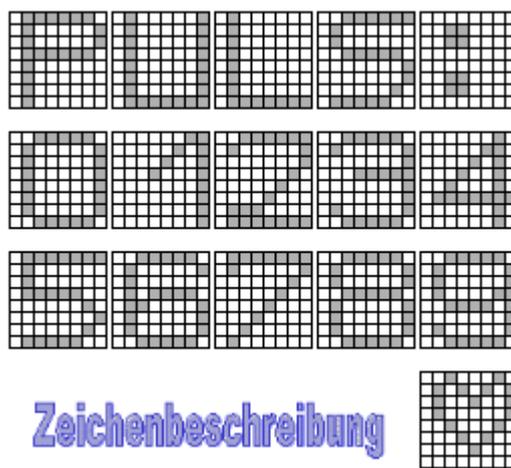
Bei der Flankendetektion wird zuerst geprüft, ob die "erste Ableitung" oberhalb eines fest vorgegebenen Schwellwerts liegt. Ist dies der Fall wird geprüft, ob ein Nulldurchgang der "zweiten Ableitung" von positiven zu negativen Werten vorliegt. Ist dies erfüllt wurde der Wendepunkt der ansteigende Pulsflanke detektiert. Die zuletzt ermittelte Periodendauer wird gespeichert. Die neue Periodendauer in Vielfachen des Abtastintervalls von 10 ms ergibt sich aus der Zahl der A/D Wandlungen seit der letzten detektierten Flanke. (Der Zähler wird bei Detektion einer neuen Flanke auf null gesetzt.)

Die neue Periodendauer wird mit der zuletzt gemessenen verglichen. Ist die Differenz kleiner als 1/8 (12,5%) der zuletzt gemessenen Periodendauer, wird ein Unterprogramm zur Berechnung der Pulsfrequenz aufgerufen. Für die Ermittlung der Pulsfrequenz wird der Mittelwert von T0 und T1 herangezogen.

Abb. 8: Struktogramm des Unterprogramms "flad"

[Quellcode des Assemblerprogramms zur Pulsmessung](#)

Anzeigen der ermittelten Pulsfrequenz auf einem LC-Display



Grundroutinen zur Ansteuerung des LC-Displays lagen bereits vor. Bereits in den vorausgegangenen Projekten im Sommersemester 1999 wurde das Pulssignal grafisch dargestellt. Zusätzlich soll nun die errechnete Pulsfrequenz in [1/min] auf dem LCD-Display zur Anzeige gebracht werden. Hierzu muß die übergebene Binärzahl zunächst in eine dezimale (BCD) Darstellung gewandelt werden. Da das LC-Display keinen Zeichengenerator besitzt, müssen Ziffern und Buchstaben in Software generiert werden. Nebenstehende Abbildung zeigt die Pixelmatrix der darzustellenden Zeichen, die im EEPROM des Mikrocontrollers als Tabelle abgelegt wird. (Näheres siehe unten)

Abb. 9: Verwendeter Zeichensatz

Technische Daten des LC-Displays (NLC-128x064CSC)

Bei dem Display handelt es sich um ein LCD-Display mit STN-Technologie. Die Auflösung beträgt 64 x 128 Dots. Die Dotgröße beträgt 0,48 mm x 0,48 mm, die Spacer haben die Größe 0,04 mm x 0,04 mm. Somit kommt man auf eine Gesamtgröße von 0,52 mm x 0,52 mm. Die Hinterleuchtung geschieht mittels gelber

LED's.

Weitere Technische Daten des LCD-Displays sind in der folgenden Tabelle dargestellt:

• Betriebsspannung	5 V
• Strom	15 mA
• Einblickwinkel	50°
• Kontrast	4
• Arbeitstemperatur	-20° bis 70°
• Lagertemperatur	-30° bis 80°
• Ansprechzeit Pixel an	200 ms
• Ansprechzeit Pixel aus	109 ms

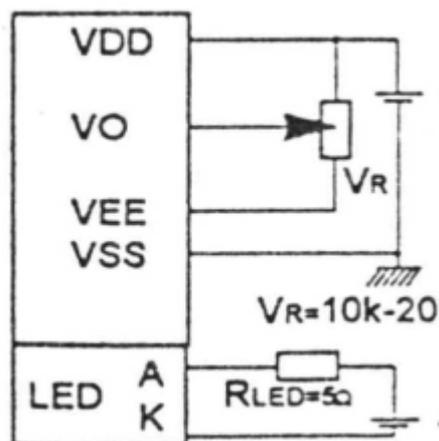


Abb. 10: Spannungsversorgung des LC-Displays

Tab. 1: Technische Daten des LC-Displays

Aufbau des LC-Displays

Im Gehäuse des Displays sind die beiden für die Matrixansteuerung der LCD-Pixel benötigten 64-Kanal-Segmenttreiber-IC integriert. Diese IC's beinhalten jeweils ein RAM mit 512 Byte und die Digital-Logik, die einerseits nötig ist, um mit einem 8-Bit Mikrocontroller Daten auszutauschen und andererseits, um die für LCD-Displays zwingend notwendige Wechselspannungsansteuerung zu realisieren. Die Ansteuerung des gesamten Display erfolgt über zwei Ports des PIC. Port B dient der Übertragung von 8-Bit breiten Datenworten, während Port D einige Steuersignale bereitstellt. Diese Pinbelegung ist in der folgenden Tabelle dargestellt:

Port B	Datenrichtung	NLC	Port D	Datenrichtung	NLC
B0	↔	DB0	D0	⇒	Daten/Instr.
B1	↔	DB1	D1	⇒	Read/Write
B2	↔	DB2	D2	⇒	Enable
B3	↔	DB3	D3	⇒	CS1
B4	↔	DB4	D4	⇒	CS2
B5	↔	DB5	D5	⇒	Reset
B6	↔	DB6			
B7	↔	DB7			

Tab. 2: Anschluß des LC-Displays an den Mikrocontroller PIC16C74

Auf Port B können sowohl die eigentlichen Grafikdaten byteweise zum Display übertragen werden als auch Statusinformationen (Busy, On/Off, Reset) des Displays vom PIC eingelesen werden. Port D ist immer als Ausgang geschaltet. Aufgrund der Aufteilung des Anzeigebereichs in zwei 64 x 64 Pixel große Hälften, die jeweils von einem LCD-Treiber (KS0108B) verwaltet werden, muß über Port D immer einer der beiden Treiber mittels der "Chip Select" Signale CS1 bzw. CS2 ausgewählt werden. In der folgenden Abbildung sind die Pixelbereiche der beiden LCD-Teile näher dargestellt. Diese Information braucht man zur Adressierung des jeweiligen Anzeigebereichs. Bei der grafischen Darstellung des Pulssignals stellt Y die Zeitachse dar.

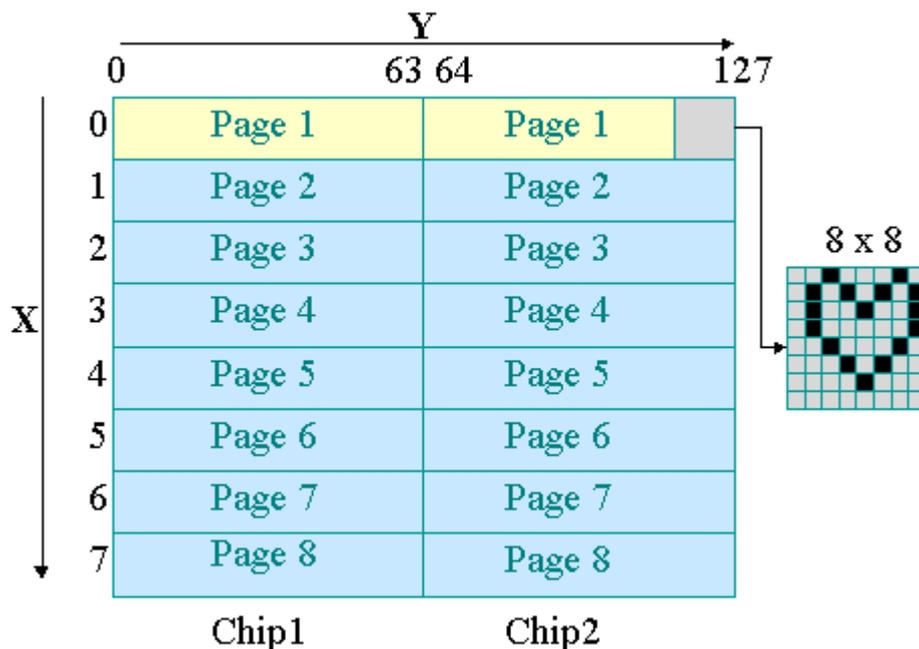


Abb. 11: Pixeladressierung

Vorgehensweise beim Programmieren der LCD-Ausgabe

Eine Übersicht über die Kommandos zur Ansteuerung der LCD-Treiber-ICs gibt folgende Tabelle:

Befehl	D/I	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Beschreibung
Display ON/OFF	0	0	0	0	1	1	1	1	1	0/1	Schaltet das Display aus (0) oder ein (1)
Set y-Adress	0	0	0	1	y-Adresse (0...63)					y-Adress-Zähler einstellen	
Set Page (x-Adress)	0	0	1	0	1	1	1	Page (0...7)			x-Adresse einstellen
Display Start Line	0	0	1	1	Display-Start-Adresse (0...63)					Beginn des Anzeigebereichs	
Status Read	0	1	B u s y	0	O N / O F F	R e s e t	0	0	0	0	Busy/Ready (1/0) ON/OFF (1/0) Res./Norm. (1/0)
Write Display Data	1	0	Write Data								Schreibt Daten in Display-RAM *
Read Display Data	1	1	Read Data								Liest Daten aus Display-RAM *

Tab. 3: Ansteuerung des LCD Controllers

Anhand der Chip Select Signale CS1 und CS2 muß zunächst der gewünschte LCD-Controller Baustein ausgewählt werden. Wie man obiger Tabelle entnehmen kann, gibt die Leitung D/I an, ob Instruktionen bzw. Statusinformationen oder Display Daten (Pixel) übergeben werden. Die Richtung des Datentransfers wird durch die R/W Leitung bestimmt. Die Übergabe von Daten oder Instruktionen wird durch die Enable-Leitung getriggert. Bei einem Wechsel von high nach low übernimmt der LCD-Controller die Daten. Es hat sich gezeigt, dass an verschiedenen Stellen Wartezeiten (realisiert durch nop-Befehle) notwendig sind, die im Datenblatt des LCD-Controllers nicht erwähnt sind.

Nach dem Lesen oder Schreiben von Display-Daten wird automatisch der Y-Adress-Zähler um eins erhöht.

Um Grafikdaten auf dem Display auszugeben, ist folgendermaßen vorzugehen:

Initialisierung

Als erstes müssen beide LCD-Treiberbausteine zurückgesetzt werden (Reset). Dies geschieht im Unterprogramm "lcdinit". Danach werden beide Treiber in den ON-Modus geschaltet, was im Unterprogramm

"lcond" geschieht. Nun wird das Display (der RAM-Speicher der Treiber) gelöscht. Dies geschieht dadurch, dass der gesamte Grafikspeicher der beiden Treiber mit 0 (ausgeschaltete Pixel) überschrieben wird. Das Unterprogramm "lcondlx" erledigt dies.

Ausgabe von Grafikdaten

Damit an einer bestimmten Stelle des Displays ein Byte (8 Bit entsprechen 8 Grafikpixeln) ausgegeben werden kann, muss vorher die gewünschte Position in Form einer x- und einer y-Adresse definiert werden. Das Unterprogramm "setaddrx" ist dafür zuständig. Entsprechend Tabelle 3 muss hierfür erst sichergestellt werden, dass sich der Treiber im Instruktionsmodus (D/I=0) und im Schreibmodus (R/W=0) befindet und entsprechend der gewünschten y-Adresse der richtige Treiber ausgewählt ist

(Für y=0...63 muss CS1=1 sein und für y=64...127 muss CS2=1 sein) Bei der y-Adressierung sind die Bits DB7=0 und DB6=1. Die Bits DB5 bis DB0 enthalten die eigentliche y-Adresse (0...63). Bei der x-Adressierung sind die Bits DB7=1, DB6=0, DB5,4,3=1 und DB2 bis DB0 enthalten die eigentliche x-Adresse (0...7).

Nun können die Grafik Daten übergeben werden. Die R/W-Leitung bleibt auf 0 für Schreiben. Lediglich D/I muss 1 sein, damit der Treiber die Daten an Port B als Grafikdaten versteht.

Zusammenwirken von Text und Grafik:

Text soll lediglich in Page 1 ausgegeben werden. Page 2 bis Page 8 stehen für die grafische Darstellung des Pulssignals zur Verfügung. Das in den Vorgängerprojekten entwickelte Programm zur Grafikausgabe löscht bei jedem dargestellten Meßwert eine gesamte Spalte (Pages 1 bis 8) des Displays. Die LösCHFunktion wurde für das neue Programm auf die Pages 2 bis 8 beschränkt. Die erfaßten Meßwerte werden so skaliert, dass sichergestellt ist, dass die Grafik nicht den Textbereich überschreibt.

Textausgabe auf dem LCD

Da das Display keinen eingebauten Zeichengenerator besitzt, muss der darzustellende Text in Form von Grafikdaten übergeben werden. Dazu müssen, wie bereits erwähnt, die Textbereiche spaltenweise in eine Tabelle überführt werden. Die Tabelle wird im Programm-Speicher des PIC mit Hilfe der Befehle retlw B'XXXXXXXX' angelegt, auf die mit einem definierten Sprung zugegriffen werden kann.

Die Tabelle soll die Zeichen: "P" "U" "L" "S" ":", " ", sowie die Zahlen von 0 bis 9 und ein Herz-Symbol enthalten.

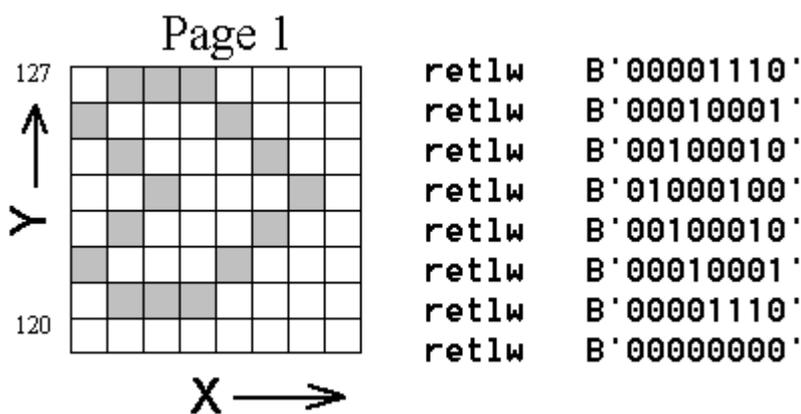


Abb. 12: Pixelmatrix in PIC Assembler als Tabelle dargestellt

In der obigen Abbildung ist am Beispiel des Herz-Symbols der Zusammenhang zwischen Pixelmatrix und der Darstellung der zugehörigen Tabelle in PIC Assembler gezeigt. Zu beachten ist, dass das Display die Bits eines Bytes aufsteigend von links nach rechts in einer Spalte anordnet, wie oben zu sehen ist. Im PIC-Befehl retlw B'D7,D6,D5,D4,D3,D2,D1,D0' jedoch sind die Bits ihrer Wertigkeit entsprechend gerade andersherum angeordnet. Deshalb müssen die Bits gespiegelt in die Tabelle abgelegt werden.

Der Programmteil zur Textausgabe besteht aus drei Teilen: Der Tabelle, dem Unterprogramm "txt" und dem Aufruf des Unterprogramms, wobei bei jedem Unterprogrammaufruf das entsprechende Zeichen aus der Tabelle an einer beliebigen y-Koordinate ausgegeben wird.

Die Ausgabe eines Zeichens sieht folgendermaßen aus:

Zunächst wird mit der Variablen "offset" angegeben an welcher Stelle in der Tabelle sich die Daten für das gewünschte Zeichen befinden (zum Beispiel 8 für die Zahl 1). Mit der Variablen "Y" wird festgelegt an welcher y-Koordinate der Text ausgegeben werden soll. (Da nur in Page 1 Text ausgegeben wird erübrigt sich die Angabe einer x-Koordinate.). Danach erfolgt der Aufruf des Unterprogramms "txt". Das Unterprogramm "txt"

liest nacheinander die 8 Spalten, die zur Darstellung eines Zeichens benötigt werden aus der Tabelle und gibt sie auf das Display aus.

Die als Binärzahl übergebene Pulsfrequenz wird zuerst in eine BCD Darstellung umgewandelt. Da die Zeichentabelle mit der Null beginnt und jedes Zeichen 8x8 Pixel groß ist, werden die Werte für die Einer-, die Zehner- und die Hunderterstelle einfach mit 8 multipliziert, um den Adress-Offset zu ermitteln, ab dem die Pixelmatrix für die jeweilige Ziffer in der Tabelle zu finden ist. Zum Beispiel wäre dies bei einer Einerstelle von 4: $4 \times 8 = 32$ (Position 32 in der Tabelle).

zurück zu: [Studentische Projektarbeiten](#)
[Labor Digitaltechnik/Technische Informatik](#)