

µProzessor – Programmierung

Abschlussaufgabe

Programmthema: Elektronische Glückwunschkarte

Programmiersprache: Assembler (Motorola)

Programmname: birthday.asm

Rüdiger Klante (B_TInf 2381, 3. Fachsemester)
Alexander Bertram (B_TInf 2616, 3. Fachsemester)

Inhaltsverzeichnis

1	.Allgemeine Problemstellung.....	3
2	.Benutzerhandbuch.....	4
2.1	.Ablaufbedingungen.....	4
2.2	.Aufbau.....	4
2.3	.Programminstallation und -start.....	4
2.4	.Bedienungsanleitung.....	5
2.5	.Fehlermeldungen.....	6
2.6	.Wiederanlaufbedingungen.....	6
3	.Programmierhandbuch.....	6
3.1	.Entwicklungskonfiguration.....	6
3.2	.Problemanalyse und Realisation.....	6
3.4	.Sourcecode mit Inline-Doku.....	7

1 . Allgemeine Problemstellung

Es soll eine Mini-Orgel mit Hilfe des Mikroprozessors Motorola 68000 programmiert werden.

Gespielt wird auf einer 'Telefontastatur'. Mit der Taste '#' kann die vorprogrammierte Melodie 'Happy Birthday' abgespielt werden und mit der '*'-Taste wird zwischen verschiedenen Klangfarben ausgewählt.

2 . Benutzerhandbuch

2.1 . Ablaufbedingungen

Für den Ablauf des Programms sind folgende Hard- und Softwarekomponenten erforderlich:

Hardware:

- IBM kompatibler PC (mit MS-DOS 5.0)
- Grafikkarte
- Monitor
- Tastatur
- Parallele Schnittstelle
- Hardwareplatine mit Motorola 68331 Mikroprozessor (FH Wedel)
- Parallelportkabel
- Netzgerät mit 5V Gleichspannung zur Versorgung der Hardwareplatine
- Stromkabel zur Verbindung von Netzgerät und Hardwareplatine
- Lautsprecher mit 3 Anschlüssen (Plus, Minus, Signal) (?)

Software:

- Assembler Editor Version 1.20 (Editor)
- Motorola Freeware CPU32 Assembler Version 1.2 (Assemblerersoftware)
- BD32 Version 1.22 (1990-94 Scott Howard) (Zum Übertragen des Programms vom PC auf die Hardwareplatine)

2.2 . Aufbau

1. Zunächst muss der PC ordnungsgemäß aufgebaut werden.
2. Anschließend verbinden Sie die parallele Schnittstelle des PC und die Hardwareplatine mit Hilfe des Parallelportkabels.
3. Verbinden Sie den 5V Ausgang des Netzgerätes mit dem 5V Eingang der Hardwareplatine (+/-5V und Ground).
4. Schließen Sie den Plus- und Minuspol des Lautsprechers an den entsprechenden 15V Anschlüssen der Hardwareplatine an.
5. Den Signalanschluss des Lautsprechers verbinden Sie mit dem analogen Ausgang der Hardwareplatine.
6. Erst jetzt dürfen PC und Netzgerät eingeschaltet werden.

2.3 . Programminstallation und -start

Die unter dem Punkt Software (s.o.) aufgeführten Programme sollten sich im Verzeichnis C:\cross68k befinden. Das Programm wird durch den Aufruf der Datei cross68k.bat, die sich im Hauptverzeichnis befindet, gestartet. Die Quellcodedatei birthday.asm sollte ebenfalls in diesem Verzeichnis vorhanden sein. Ist dies nicht der Fall, muss in der Programmierumgebung unter

File / Change directory der Pfad eingestellt werden, in dem sich die Datei befindet.

Nachdem die Programmierumgebung gestartet wurde, öffnen Sie die Quellcodedatei unter *File / Open*. In dem nun erscheinenden Fenster wählen Sie die Datei birthday.asm mit Hilfe der Pfeiltasten aus und drücken dann die Enter-Taste zum Öffnen der Datei. Wurde die Datei erfolgreich geladen, lässt sich der Quelltext im Editor-Fenster erkennen.

Der nächste Schritt besteht im Assemblieren des Programms. Dazu drücken Sie die Taste F9 (Alternativ: Menü *Run / Assemble*).

Achten Sie darauf, dass nach dem Assemblieren auf dem schwarzen Bildschirm die Statusmeldungen 0 Errors und 0 Warnings erscheinen. Danach werden Sie aufgefordert eine beliebige Taste zu drücken. Jetzt befinden Sie sich wieder im Editor.

Um das assemblierte Programm auf die Hardwareplatine zu übertragen, drücken Sie erst einmal die Tasten Alt + R (Menü *Run*) und wählen dann den Menüeintrag *Debug*. Nun erscheint ebenfalls ein anderer Bildschirmmodus, in dem Sie folgende Befehle eingeben müssen (jeder Befehl wird mit der Enter-Taste bestätigt):

Reset: resetet die CPU auf der Hardwareplatine

lo birthday.s19: lädt den Programmcode auf die Hardwareplatine

go: startet die CPU mit dem Programm

Jetzt kann das Programm auf der Hardwareplatine genutzt werden.

2.4 . Bedienungsanleitung

Mit den Tasten '1'-'8' auf der Hardwareplatine kann die C-Dur Tonleiter abgespielt werden. Nach dem Druck einer der Tasten wird der Ton so lange abgespielt, bis eine andere Taste gedrückt wird.

Die Taste '#' dient dazu, die vorprogrammierte Melodie 'Happy Birthday' ein Mal abzuspielen.

Mit der Taste '0' wird das Abspielen eines Tones bzw. der Melodie gestoppt. Die Taste '9' verhält sich wie '0'.

Um zwischen verschiedenen Klangfarben zu wechseln, verwenden Sie die '*'-Taste. Das Abspielen des aktuellen Tones wird dadurch gestoppt.

Die verschiedenen Klangfarben werden durch Ausgabe von sinus-, sägezahn- und rechteckförmigen Schwingungen erzeugt. Es wird in folgender Reihenfolge gewechselt: Sinus => Sägezahn => Rechteck => Sinus => ...

2.5 . Fehlermeldungen

Das Programm erzeugt keine Fehlermeldungen.

2.6 . Wiederanlaufbedingungen

Sollte während des Programmablaufs der Strom ausfallen, muss das Programm neu übertragen werden, indem Sie wie unter dem Programmpunkt *Programminstallation und -start* vorgehen, jedoch ist ein erneutes Assemblieren nicht notwendig.

Sollte es vorkommen, dass die CPU der Hardwareplatine abstürzt, kann das Programm durch die Eingabe der Befehle *Reset* und *Go* neu gestartet werden. Ist dies nicht möglich, so übertragen Sie bitte das Programm erneut auf die Hardwareplatine.

3 . Programmierhandbuch

3.1 . Entwicklungskonfiguration

Zur Entwicklung des Programms stand uns folgende Hard – und Software zur Verfügung:

Hardware:

PC: PC mit MS-DOS 5.0, Grafikkarte, Monitor, Tastatur, Parallelport-Schnittstelle, 3,5" Diskettenlaufwerk

Platine: Hardwareplatine mit Motorola 68331 Mikroprozessor, Tastatur mit 12 Tasten, 4 LED-7-Segmentanzeigen (bereitgestellt durch die FH Wedel)

Zubehör: Parallelportkabel, Netzgerät mit 5V Gleichspannung zur Versorgung der Hardwareplatine, Stromkabel, Lautsprecher

Software:

OS: MS-DOS 5.0

Editor: Assembler Editor Version 1.20

Assembler: Motorola Freeware CPU32 Assembler Version 1.2

Übertragungssoftware: BD32 Version 1.22 (1990-94 Scott Howard)

3.2 . Problemanalyse und Realisation

Das Programm lässt sich in 2 wesentliche Probleme unterteilen. Es muss die Abfrage der Tastatur realisiert werden und die Ausgabe der richtigen Töne auf dem analogen Ausgang.

Tastaturabfrage:

Zur Abfrage der Tastatur wird der Port F des Controllers verwendet. Dabei werden die Leitungen 1–7 verwendet. Die Leitungen 1–3 dienen zur Auswahl der Spalten, die restlichen 4 für die Zeilen. Leitung Null wird nicht verwendet. Alle diese 7 Leitungen arbeiten mit negativer Logik, d.h. eine logische 0 bedeutet, dass die Leitung aktiv ist. Durch diesen Aufbau gehört zu jeder Taste eine bestimmte Bitkombination. Wird z.B. die 1 gedrückt, sind die Leitungen drei (erste Spalte) und sieben (erste Zeile) aktiv. Daraus ergibt sich folgende Bitkombination: 01110110, dezimal also die Zahl 118.

Tonausgabe:

Bei der Tonausgabe spielen 2 Dinge eine wichtige Rolle: die Tabellen mit den Werten der 3 Schwingungen: Sinus, Sägezahn, Rechteck und ein 16 Bit Zähler. Der Zähler wird bei jedem Interrupt um die Frequenz des aktuellen Tones inkrementiert. Dabei werden die Bits 8-12 als Index in die aktuelle Tontabelle verwendet. Der indizierte Wert aus der Tabelle wird dann einfach auf dem analogen Ausgang der Platine ausgegeben.

Abspielen von 'Happy Birthday':

Die Ausgabe der vorprogrammierten Melodie wurde mit Hilfe von 2 Tabellen und einem Zähler realisiert. In der ersten Tabelle stehen die Noten der Melodie und in der zweiten der Takt. In den Zähler wird der aktuelle Wert aus der Takttabelle geladen und bei jedem Interrupt dekrementiert. Solange der Zähler nicht bei Null angekommen ist, wird der entsprechende Ton erzeugt. Sonst wird der nächste Ton und der nächste Takt geladen.

Umschalten der Klangfarben:

Es wird ein Flag, verwendet, das anzeigt, welche Tontabelle zur Zeit geladen ist. Wird die '*'-Taste gedrückt wird in einem Case-Verteiler das Flag umgeschaltet und somit die nächste gewünschte Tontabelle geladen.

3.3. Struktogramme

3.4. Sourcecode mit Inline-Doku

```
*****
*                               Mikroprozessor-Programmierung                               *
*                               Abschlusssaufgabe WS 05/06                               *
*                               "Glueckwunschkarte"                                       *
*                               *                                                         *
*                               Ruediger Klante                                           *
*                               Alexander Bertram                                         *
*****
* happy bday: g',g',a',g',c'',h',g',g',a',g',d'',c'',g',g',g'',e''
* (g-dur)      c'',c'',h',a',f'',f'',e'',c'',d'',c''

include simregs.inc
include gptregs.inc

anaout      equ    $21001      ; Analoger Ausgang (Multi I/O)
c           equ    264        ; c' -> 264 Hz
d           equ    297        ; d' -> 297 Hz
e           equ    334        ; e' -> 334 Hz
f           equ    356        ; f' -> 356 Hz
g           equ    401        ; g' -> 401 Hz
a           equ    440        ; a' -> 440 Hz ("Kammerton" A)
h           equ    495        ; h' -> 495 Hz
c1          equ    528        ; c'' -> 528 Hz
d1          equ    594        ; d'' -> 594 Hz
e1          equ    668        ; e'' -> 668 Hz
f1          equ    713        ; f'' -> 713 Hz
g1          equ    802        ; g'' -> 802 Hz
achtel      equ    32         ; Laenge fuer Achtelnote
viertel     equ    64         ; Laenge fuer Viertelnote
halbe       equ    128        ; Laenge fuer Halbe Note
ganze       equ    256        ; Laenge fuer Ganze Note

org         $10000
SSP         dc.l    $20000
Reset       dc.l    $10400

org         $10100
dc.l        Interrupt

org         $11000
*           C-Dur Tonleiter zum freien Spiel
toene       dc.w    c,d,e,f,g,a,h,c1

*           Melodie "Happy Birthday" (in G-Dur)
happy       dc.w    g,g,a,g,c1,h,g,g,a,g,d1,c1,g
            dc.w    g,g1,e1,c1,c1,h,a,f1,f1,e1,c1,d1,c1

*           Tabelle mit Werten fuer Rechtecksignal
rechteck    dc.b    255,255,255,255,255,255,255,255
            dc.b    255,255,255,255,255,255,255,255
            dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```



```

*
saenge      Tabelle mit Werten fuer Saegezahnsignal
dc.b        128,136,144,152,160,168,176,184
dc.b        192,200,208,216,224,232,240,248
dc.b        0,8,16,24,32,40,48,56,64,72,80,88
dc.b        96,104,112,120

*
sinus       Tabelle mit Werten fuer Sinussignal
dc.b        128,153,177,199,219,234,246,255
dc.b        254,246,234,219,199,177,153,128
dc.b        103,79,57,37,22,10,2,0
dc.b        2,10,22,37,57,79,103

*
laenge      Tabelle fuer den Takt der Melodie
dc.b        achtel,achtel,viertel,viertel,viertel,halbe
dc.b        achtel,achtel,viertel,viertel,viertel,halbe
dc.b        achtel,achtel,viertel,viertel,achtel,achtel
dc.b        viertel,viertel,achtel,achtel,viertel,viertel
dc.b        viertel,halbe

*
tcode       Tabelle mit Tastencodes 0-9 und #,*
dc.b        234,118,122,124,182,186,188,214,218,220,236,230

*
stelle      Stellencodes zum Multiplexen der Anzeige
dc.b        $70,$b0,$d0,$e0

timer       org      $14000
ds.w        1          ; Timer fuer Takt der Melodie
tontimer    ds.w        1          ; Timer fuer Tonerzeugung
frequenz    ds.w        1          ; Frequenz des aktuellen Tons
index       ds.b        1          ; index in die Melodietabelle
bday        ds.b        1          ; Melodie-Flag
stell_index ds.b        1          ; Stellenindex fuer Anzeige
ton         ds.b        1          ; Tonindex
stop        ds.b        1          ; Stop-Flag (Stoppen alles sounds)
tonpointer  ds.b        1          ; Pointer fuer aktuellen Tontyp
notoggle    ds.b        1          ; Flag, ob Tontyp gewechselt werden kan
n

*****
*          Start des Hauptprogramms          *
*          "Happy Birthday"                  *
*****

start      org      $10400
move.w     #1,pitr      ; Interruptrate = 8 kHz
move.w     #$0140,picr   ; IRQ-Level 1 , Vektor 64
move.b     #$FF,ddrgp    ; Initialisierung
move.b     #$00,portgp   ; der Sieben-Segment
move.b     #$F0,ddrf     ; Anzeige
move.b     #$FF,portf

clr.l      d0          ; Loeschen aller
clr.l      d1          ; Register und
clr.l      d2          ; Variablen
clr.l      d3
clr.l      d4
clr.l      d5
clr.l      d6
clr.l      d7

```

```

        clr.w    frequenz
        clr.b    index
        clr.b    timer
        clr.b    bday
        clr.b    stell_index
        clr.b    ton
        clr.b    stop
        clr.w    tontimer
        clr.b    tonpointer

        lea      happy,a0                ; Initialisierung
        move.b   0(a0),frequenz          ; der Variablen
        lea      laenge,a1
        move.b   0(a1),d2
        asl.l    #5,d2
        move.w   d2,timer

        move.b   #0,bday
        move.b   #1,stop
        move.b   #4,stell_index
        move.b   #1,tonpointer
        move     $2000,SR

endlos      bra      endlos

*****
* Subroutine zur Erzeugung der Melodie *
* "Happy Birthday" *
*****
happybday   movem.l d0-d2/a0/a1,-(a7)      ;Register auf Stack
            clr.l    d0                    ;Leeren
            clr.l    d1                    ;der
            clr.l    d2                    ;Datenregister

            lea      happy,a0              ; Laden der Tabellen Melodie
            lea      laenge,a1            ; und Laenge
            subi.w   #1,timer              ; Decrement Timer Tonlaenge
            bne      go

            addi.b   #1,index              ; Increment Index in Melodietabelle
            move.b   index,d1              ; Index -> d1

            cmp.b    #26,d1                ; Melodie Zuende?
            bne      gogo
            move.b   #0,bday              ; Stoppen der Melodie
            move.b   #1,stop              ; Alle Toene aus
            clr.w    timer                 ; Ruecksetzen timer
            clr.b    index                 ; Ruecksetzen index
            clr.l    d1

gogo
n           lsl.b    #1,d1                  ; Index an Word-Tabelle anpasse

            move.w   0(a0,d1.w),frequenz   ; Tonfrequenz laden

            move.b   index,d1
            move.b   0(a1,d1.w),d2         ; Tonlaenge
            asl.w    #6,d2                 ; laden
            move.w   d2,timer              ; und anpassen
go          bsr      makethesound          ; -> Tonerzeugung

            clr.l    d1

```

```

        move.b    index,d1

        lsl.b     #1,d1                ; Index an Word-Tabelle anpasse
n
        move.w    0(a0,d1.w),frequenz ; Laden der Frequenz

weiter   movem.l  (a7)+,d0-d2/a0/a1    ;Register vom Stack

        rts

*****
* Subroutine zur Tonerzeugung *
* Hier wird die jeweilige Signalkurve durchlaufen *
*****
makethesound    movem.l  d0/d1/a0/a1,-(a7)
                clr.l     d0            ; Register Loeschen
                clr.l     d1
                move.w    frequenz,d1  ; Tonfrequenz laden
                add.w     d1,tontimer   ; Zaehler + Tonfrequenz
                move.w    tontimer,d0
                and.w     #%1111111100000000,d0 ; Auslesen
                lsr.w     #8,d0         ; Signifikanter Stellen
                and.b     #%00011111,d0 ; des Tontimers
                clr.l     d1
                move.b     tonpointer,d1 ; Auswahl des
                cmp.b     #1,d1         ; Tontyps
                beq       loadsinus     ; fuer Wiedergabe
                cmp.b     #2,d1
                beq       loadsaeege
                lea        rechteck,a1   ; Rechteck laden
                bra        endmts

loadsinus       lea        sinus,a1     ; Sinus laden
                bra        endmts

loadsaeege      lea        saeege,a1    ; Saeegezahn laden

endmts          move.b     0(a1,d0.w),anaout ; Tonausgabe
                movem.l  (a7)+,d0/d1/a0/a1
                rts

*****
* Subroutine zum Umschalten der Tonform *
* Ein Tonpointer wird gesetzt, wobei *
* 1 = sinus ; 2 = saeegezahn ; 3 = rechteck *
*****
toggle_sounds   movem.l  d0,-(a7)
                move.b     tonpointer,d0 ; Setzen des
                cmp.b     #1,d0         ; Tonpointers
                beq       setsaeege     ; in Abhaengigkeit
                cmp.b     #2,d0         ; von aktueller
                beq       setrechteck   ; Tonform
                move.b     #1,tonpointer ; 1 = Sinus
                bra        endtoggle
setsaeege       move.b     #2,tonpointer ; 2 = Saeegezahn
                bra        endtoggle
setrechteck     move.b     #3,tonpointer ; 3 = Rechteck

endtoggle       movem.l  (a7)+,d0

```

rts

```
*****
* Interrupt-Routine                                     *
* Multiplexing der Anzeige                             *
* Tastenabfrage                                         *
*****
Interrupt      movem.l  d0-d7/a0-a2,-(a7)              ;Register auf Stack
               clr.l    d0                            ;Leeren
               clr.l    d1                            ;der
               clr.l    d2                            ;Daten-
               clr.l    d3                            ;Register
               clr.l    d4
               clr.l    d5
               clr.l    d6
               clr.l    d7

               move.b   bday,d3                      ; Pruefen des Birthday-Flags
               beq      anzeige                       ; wenn nicht gesetzt, ueberspringen

               bsr      happybday                     ; Wiedergabe "Happy Birthday"

*
* Multiplexen der Anzeige (zur Tastenerkennung)
anzeige        clr.l    d0
               lea      stelle,a2                    ; Ansteuern
               move.b   stell_index,d7               ; des Stellentreibers
               move.b   -1(a2,d7.w),portf            ; zum Lesen der Tastatur

               sub.b    #1,stell_index                ; letzte Stelle erreicht?
               bne      weiter2
               move.b   #4,stell_index               ; Index zuruecksetzen

*
* keine taste gedrueckt?
weiter2         move.b   portf,d5                    ; Pruefen ob keine Taste
               and.b    #%00001110,d5                ; gedrueckt
               cmp.b    #14,d5                       ; wenn ja
               beq      weiter1                       ; direkt weiter

               move.b   portf,d5
               and.b    #%11111110,d5                ; loeschen bit 0
*
               ; -> nicht signifikant

               lea      tcode,a1
               clr.l    d6
               cmp.b    11(a1),d5                    ; wurde * gedrueckt?
               beq      starpressed                   ; -> starpressed
               cmp.b    10(a1),d5                    ; wurde # gedrueckt?
               beq      rautepressed                  ; -> rautepressed
               cmp.b    (a1)+,d5                     ; wurde 0 gedrueckt?
               beq      nullpressed                   ; -> nullpressed

tloop          add.b    #1,d6                        ; Erkennung gedrueckter Taste
               cmp.b    (a1)+,d5                    ; mit Erzeugung von Index
               beq      continue                     ; in Tastentabelle
               cmp.b    #8,d6
               bne      tloop
               bra      weiter1                      ; Taste 9 verhaelt sich wie 0

nullpressed    move.b   #1,stop                      ; bei gedrueckter 0
               move.b   #0,bday                     ; -> Stoppen des Tons bzw
```

```

        clr.b    notoggle          ; notoggle flag zuruecksetzen
        bra     weiter1           ; der Melodie

rautepressed    move.b    #1,bday          ; bei gedruckter #
                move.b    #0,index        ; starten der Melodie "Happy Birthday"
                move.b    #1,notoggle      ; notoggle flag setzen
                bra     weiter1

starpressed     clr.l    d0
                move.b    notoggle,d0      ; wenn notoggle gesetzt
                bne     weiter1           ; weiter, NICHT togglen
                bsr      toggle_sounds    ; -> Ton umschalten
                move.b    #1,notoggle      ; notoggle flag setzen
                move.b    #1,stop         ; Ton wird gestoppt
                bra     weiter1

continue        lea     toene,a1
                move.b    d6,ton           ; Sichern des Tonindexes
                sub.b    #1,d6            ; Anpassung des Tastenindexes
                lsl.b    #1,d6            ; an Word-Tabelle
                move.w    0(a1,d6.w),frequenz ; Laden entsprechender Frequenz
                move.b    #0,bday         ; Stoppen der Melodie
                move.b    #0,stop         ; loeschen Stop-Flag
                clr.b    notoggle

weiter1         move.b    bday,d0          ; soll Melodie gespielt werden?
                bne     fertig
                move.b    stop,d0         ; soll nichts gespielt werden?
                bne     fertig
                bsr      makethesound      ; -> Ton erzeugen
                lea     toene,a1
                move.b    ton,d6           ; Laden des aktuellen Tons
                sub.b    #1,d6            ; und Anpassen des Indexes
                lsl.b    #1,d6            ; an Word-Tabelle
                move.w    0(a1,d6.w),frequenz ; Laden Frequenz aktuellen Tons

fertig         movem.l    (a7)+,d0-d7/a0-a2 ;Register vom Stack

                rte

                end     start

```