Automatisch opbellen

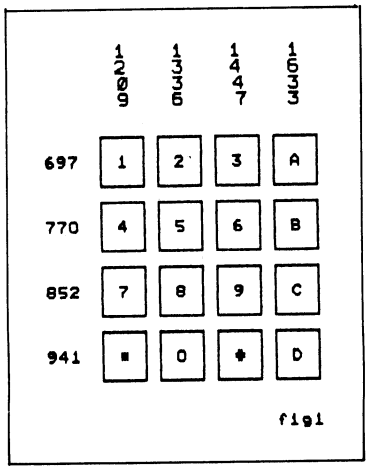
Raymond vd Geest, CUC 34/35 1990

Gescand en via OCR naar PDF omgezet door HansO, 2001

De laatste tijd worden steeds meer oude "analoge" PTT centrales vervangen door nieuwe "digitale" centrales. Deze nieuwe centrales kunnen behalve via puls dial (oude kiesschijf) nu ook via 'tone dial' nummers ontvangen. Dit was voor ons reden een project te ontwerpen dat en een bestand van uw telefoon klapper kan opbouwen, maar daarnaast automatisch voor u het juiste nummer kan draaien. Hoe? Dat volgt hieronder.

Werking 'tone dial'

Eerst even de vraag: hoe werk het 'tone dial' opbel systeem ? De naam 'tone dial' zegt het eigenlijk al, bij het intoetsen van een telefoonnummer (met een nieuw toestel), wordt er voor iedere toets een bepaalde toon op de PTT telefoonlijn gezet. idere toets - dus ieder cijfer - heeft zijn eigen toon(hoogte). ledere toon is op
gebouwd uit twee sinusvormige frequenties (zie figuur l). De meesten van u hebben er intussen al wel kennis mee gemaakt, of op TV of al thuis. Je toetst op het toestel een nummer in en wat je hoort is blieb-blab-bloeb-blieb-blieb-blub. Even wachten en ja hoor, je krijgt die meneer die je zo vriendelijk wil helpen je belasting formulier in te vullen nog aan de lijn ook. Als we ons niet vergissen, beschikt nog niet geheel Nederland over deze moderne centrales. Informeer eerst even voor u aan dit project begint (of u moet het gewoon machtig interessant vinden, zoals ik).



De opzet

Willen we nu via het toetsenbord van de computer kunnen bellen, dan moet de computer voor iedere toon (toetsaanslag) die bepaalde bijbehorende toon kunnen opwekken. Nog mooier is het degene die je wilt bellen middels het intikken van zijn naam uit het bestand op het scherm te halen en via ja of nee de computer automatisch het nummer te laten kiezen (of niet). We gaan kijken of dat lukt.De oplossing voor het 'componeren' van de juiste tonen werd gevonden in een D/A convertor. Door deze op de printerpoort aan te sluiten en via een LF versterker aan een speaker toe te voeren, kunnen met de juiste software de juiste toonhoogten en dus het juiste nummer geproduceerd worden. Door nu simpelweg het speakertje bij de hoorn van de telefoon te houden - zoals tegen je oor -, kan nu door de computer een nummer "gedraaid" worden.De hierbij afgedrukte software is een high-tech super moderne C.U.C, sophisticated telefoonklapper. Het aantal telefoonnummers dat opgeslagen kan worden hangt af van de computer. Maar zo'n 40 kbyte aan gegevens kunt u wel kwijt.De werking is heel simpel. U toetst uw namenbestand en de telefoonnummers in. Dit bestand wordt op diskette bewaard en zonodig weer in de computer geladen. Met de cursor kunt u nu door het gehele bestand dwalen en bij de juiste naam te kennen geven dat de computer dat nummer automatisch kiest.

Hardware

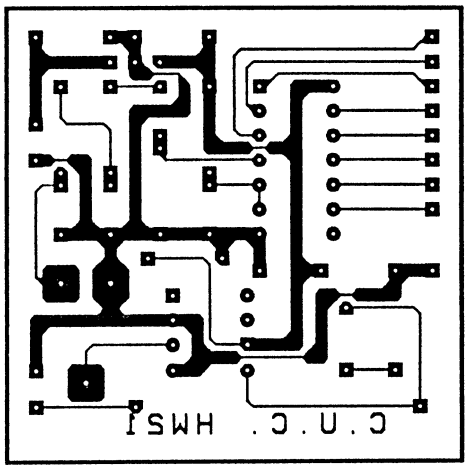
Voor het genereren van de benodigde tonen is gekozen voor een D/A convertor van Ferrantie (ZN 426). Deze chip bevat een nauwkeurig R2R ladder netwerk en een 2,5 Volt referentie bron. De chip dient aangesloten te worden op de printer poort. Het analoge signaal dat uit de ZN426 komt, wordt middels een laagdoorlaatfilter ontdaan van alle frequenties boven de 8000 Hz. Achter dit filter is een single chip LF versterker opgenomen voor de nodige versterking van het signaal.De oplossing voor de 5 V voeding is een 5 volts spanningsregelaar, die de ingangsspanning (8 .. 30V) naar 5 Volt omzet. Op deze manier kunnen we in plaats van een 9 Volt batterij ook een netadapter gebruiken.

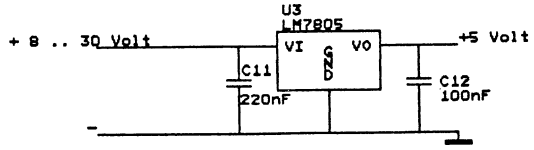
onderdelen lijst:

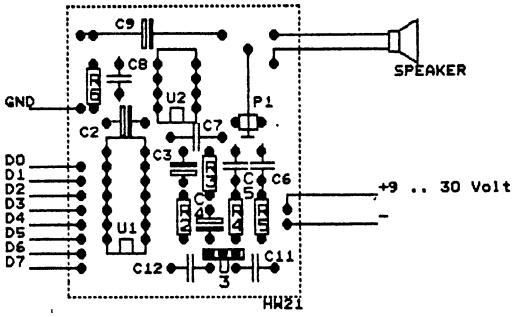
R1	--		R4	1	K
R2	390	Ohm	R5	10	K
R3	82	K	R6	10	Ohm
P1	47	K	LS	8	Ohm
C1	--		C7	22	nF
C2	10	µF	C8	47	nF
C3	1	µF	C9	100	µF
C4	10	µF	C10	--	
C5	18	nF	C11	220	nF
C6	1,8	nF	C12	100	nF
UC1	ZN 426		Print HW.21		
UC2	LM 386		D-con. 25 pins male		
UC3	LM 7805		IC-voet 14 pins DIL		

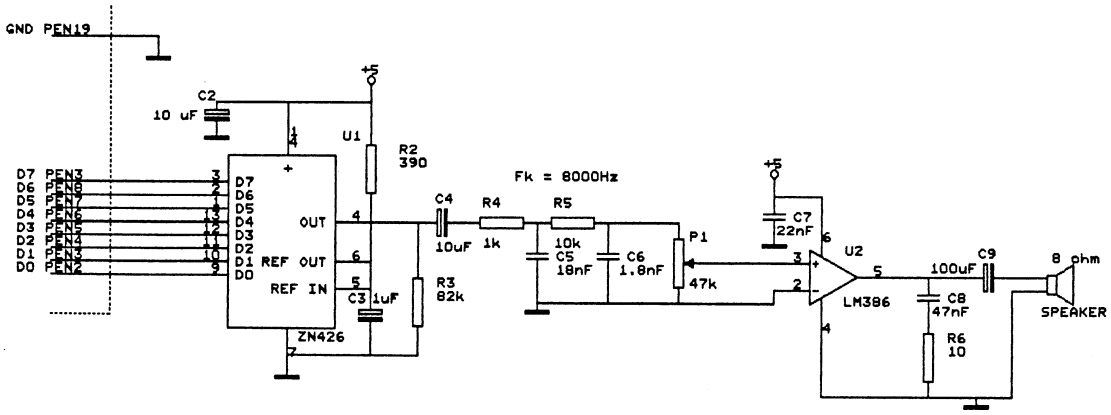
Opbouw

Voor de audio interface is een printje ontworpen. De opbouw van het printje zal verder weinig problemen opleveren, maar er zijn een aantal puntjes waar op gelet moet worden:
PI is een instel potmeter voor het volume;
LJ3 is de spanningsregelaar LM7805 en dient met de koelbevestiging naar onder te worden gemonteerd (dikke streep op componenten opstelling);
- Voor de elco's moeten tantaals worden genomen (anders passen ze niet op de print).









De software

Het programma is geschreven in de CP/ M versie 3.0 van Turbo Pascal. Dit programma houdt een telefoonlijst bij en zal na opvragen van de gewenste naam indien gewenst ook het er bij behorende telefoon nummer bellen. Voor het genereren van de tonen maakt het programma een aparte file aan (DIAL.DAT).Het schrijven naar de printerpoort wordt verzorgd door een kleine machinetaal routine. Het enige verschil tussen de SVI.328 en MSX versie van het programma is het nummer van de poort. Bij de SVI.328 is het 10h en bij de MSX machines poort 81h.

```
PROGRAM Bellen_via_de_printerpoort;

{ original version by CUC

  updated and debugged by MSXHans, 2001

  version 2.0

}

CONST
     regel =9;           { Aantal regels van het venster  }

     Tijd_tussen_sampel =  115E-6; { tijd tussen sampels }
     aantal_samples     = 1000    ;
                          { aantal sampels -  tijdsduur
                            van l toets = aantal sampels *
                            Tijd tussen sampel }
     Freq_laag1  =   697 ;{ Frequentie voor de lage band }
     Freq_laag2   =  770 ;
     Freq_laag3   =  852 ;
     Freq_laag4   =  941 ;
     Freq_hoog1  =  1209 ; { Frequentie voor de hoge band }
     Freq_hoog2  =  1336 ;
     Freq_hoog3  =  1447 ;
     Freq_hoog4  =  1633 ;

TYPE
     string20 = string[20];

     filerec = RECORD
               naam     :string[20];
               tel      :string[20];
     END;

     telefoon =  ^telefoonrec;
        { Pointertype naar een record van de Linked List }

     telefoonrec  = RECORD
                    telefoonnummer :string[20];
                    naam           :string[20];
                    next           :telefoon;
                                    { wijst naar volgend record }
                    prev           :telefoon;
                                    { wijst naar vorig record }
     END;
                               { record definitie waarin de
                                 gegevens worden opgeslagen }

     Tonen =RECORD
            Laag   : array[1..4,0..aantal_samples] of byte;
            Hoog   : array[1..4,0..aantal_samples] of byte;
     END;

     Tonenptr =  ^Tonen;
                   { sampels voor de verschillende tonen }
```

```
{
    Globale variabelen declaratie

}

VAR  keuze        : char;         { aanslag van hoofdmenu }
     int_keuze    : integer;
     tellijst     : telefoon;
                         { Wijst naar het begin van lijst }
     belptr       : telefoon;  { Wijst naar cursor record }
     topptr       : telefoon;
                  { Wijst naar bovenste record op scherm }
     naam, tel    : string[20];

     dirty        : boolean;
     sampels      : Tonenptr;

PROCEDURE init_lijst(var lijst:telefoon);

BEGIN
  new(lijst);              { Maakt een dummy record aan }
  WITH lijst^ DO BEGIN
    next:=NIL;
    prev:=lijst;
    naam:= ' ' ;
    telefoonnummer:= '';
   END;
 END;

 PROCEDURE maak_record(var p :telefoon;var naam,
                                 telefoonnummer :string20);


 BEGIN
   NEW(p);                    { Maakt ruimte in heap vrij }
   P^.naam:=naam;
   P^.telefoonnummer:=telefoonnummer;
   P^.next:=nil;
   P^.prev:=nil;
 END;

 PROCEDURE search(var lijst, p :telefoon;
                   searchnaam : string20;
                  var exist : boolean);
 BEGIN
  p:=lijst;
  WHILE  (p^.next <> nil) and (p^.naam < searchnaam) DO
     BEGIN
     p:=p^.next;
     END;
  exist:=  (p^.naam = searchnaam);
 END;

PROCEDURE  insert(var lijst, voor, p:telefoon);


BEGIN
```

```
  p^.next:=voor;
  p^.prev:=voor^.prev;
  IF lijst <>voor  THEN
      voor^.prev^.next:=p
  ELSE
      lijst:=p;
  voor^.prev:=p;
END;

PROCEDURE delete(var  lijst, p:telefoon);

BEGIN
  IF p^.next <>  NIL THEN BEGIN
    IF p <> lijst  THEN
        p^.prev^.next:=p^.next
    ELSE
        lijst:=p^.next;
    p^.next^.prev:=p^.prev;
    dispose(p);
  END;
END;

PROCEDURE next_record(var lijst, p:telefoon;
                                      var nonext:boolean);
BEGIN
  IF  (p^.next=nil) or (p^.next^.next=nil) THEN
      nonext:=true
  ELSE BEGIN
         nonext:=false;
         p:=p^.next
  END;
END;

PROCEDURE prev_record(var lijst, p:telefoon;
                                     var noprev:boolean);
BEGIN
  IF p=lijst THEN noprev:=true
   ELSE BEGIN
          noprev:=false;
          p:=p^.prev;
   END;
END;

PROCEDURE print_lijst(lijst  :telefoon);

BEGIN
  WHILE lijst^.next <>  nil DO BEGIN
    Write (lijst^.naam,'   ');
    WriteLn( lijst^.telefoonnummer);
    lijst:=lijst^.next;
  END;
END;

 PROCEDURE Vul_array (nummer, hoog, laag:integer);
    { Bereken de sinusen voor de hoge en lage frequenties.
      Sla de berekende monsters op in het record }
 VAR
```

```
   tijd      :real;
   teller    :integer;

BEGIN
  write('Even wachten...');
  tijd:=0;
  FOR teller:= 0 to aantal_samples DO BEGIN
                                  { bereken de hoge band }
    sampels^.hoog[nummer,teller]:=
            BYTE(TRUNC(((SIN(2*PI*hoog*tijd)+1)/4)*255)) ;
                     { bereken de lage band met een extra
                       dempingsfactor 0.8 }
    sampels ^.laag[nummer,teller]:=
        BYTE(TRUNC(((SIN(2*PI*laag*tijd)+1)/4)*0.8*255)) ;
    tijd:=tijd+tijd_tussen_sampel;     { volgende sampel }
  END;
  WRITELN(nummer);
END;

PROCEDURE Maak_tabel_voor_DAC;
  { maak de tabel aan met de sampels. Dit berekenen zal
    maar een keer worden uitgevoerd. Na berekening zullen
    de sampels op schijf worden opgeslagen }

BEGIN
  WRITELN(' Berekenen  sinusen voor de verschillende frequenties. ');
  WRITELN(' Na de berekeningen worden  de gegevens op schijf
weggeschreven');
  WRITELN(' onder de  filenaam DIAL.DAT.');
  Vul_array( 1, Freq_hoog1,  Freq_laag1) ;
  Vul_array( 2, Freq_hoog2,  Freq_laag2) ;
  Vul_array( 3, Freq_hoog3,  Freq_laag3) ;
  Vul_array( 4, Freq_hoog4,  Freq_laag4) ;
END;

PROCEDURE DAC(offsetH, offsetL, aantal: integer);
     { Machinecode procedure waarbij de volgende gegevens
       worden meegegeven :
  offsetH : offset van tabel hoge monsters
  offsetL : offset van tabel lage monsters
  aantal : aantal monsters }

BEGIN
INLINE ($F3/            {          DI                   }
        $ED/$4B/aantal/ {          ld    bc,aantal      }
        $ED/$5B/offsetH/{          ld    de,offsetH     }
        $2A    /offsetL/{          ld    hl,offsetL     }
        $1A    /        {  lusl:   ld    a,(de)         }
        $86    /        {          add   a,(hl)         }
{       $D3/$10/                  out   (10h),a        }

{voorgaande regel is bij MSX vervangen door               }
        $D3/$91/        {          out   (91h),a        }

        $E5    /        {          push  hl             }
        $21/ 10/$00/    {          ld    hl,10          }
        $2B    /        {  lus2:   dec   hl             }
```

```
          $7C    /        {          ld    a,h            }
          $B5    /        {          or    l              }
          $20/$FB/        {          jr    nz,lus2        }
          $E1    /        {          pop   hl             }
          $23    /        {          inc   hl             }
          $13    /        {          inc   de             }
          $0B    /        {          dec   bc             }
          $78    /        {          ld    a,b            }
          $B1    /        {          or    c              }
          $20/$EB/        {          jr    nz,lusl        }
          $FB)            {          EI                   }

 END;

 PROCEDURE WAIT(ticks:integer);{ wacht een aantal (tlicks) }
 VAR  t :integer;

 BEGIN
   FOR t:= 1 TO ticks*30 do;

 END;

PROCEDURE Dial_nummer(nummer:char);
                                  { bel het gekozen nummer }
VAR offsetH,  offsetL, segment    :integer;
    laag,hoog                     :integer;

BEGIN
  CASE nummer OF
       '1' : begin  laag :=1 ; hoog :=1 end ;
       '2' : begin  laag :=1 ; hoog :=2 end ;
       '3' : begin  laag :=1 ; hoog :=3 end ;
       '4' : begin  laag :=2 ; hoog :=1 end ;
       '5' : begin  laag :=2 ; hoog :=2 end ;
       '6' : begin  laag :=2 ; hoog :=3 end ;
       '7' : begin  laag :=3 ; hoog :=1 end ;
       '8' : begin  laag :=3 ; hoog :=2 end ;
       '9' : begin  laag :=3 ; hoog :=3 end ;
       '0' : begin  laag :=4 ; hoog :=2 end ;
       '*' : begin  laag :=4 ; hoog :=1 end ;
       '#' : begin  laag :=4 ; hoog :=3 end ;
       'A' : begin  laag :=1 ; hoog :=4 end ;
       'B' : begin  laag :=2 ; hoog :=4 end ;
       'C' : begin  laag :=3 ; hoog :=4 end ;
       'D' : begin  laag :=4 ; hoog :=4 end ;
 END;
 offsetH:=addr(sampels^.hoog[hoog,0]);
 offsetL:=addr(sampels^.laag[laag,0]);
 DAC(offsetH,offsetL,aantal_samples);
END;

PROCEDURE Dial_string(telefoonnummer : string20);
                                  { bel een telefoonnummer }
VAR i: integer;

BEGIN
  GotoXY(4,18) ;
```

```
  write('Er wordt gebeld naar ');
  FOR i :=  1 TO LENGTH(telefoonnummer) DO
    BEGIN
      IF telefoonnummer[i] <> '-' THEN
        BEGIN
          write(telefoonnummer[i]);
          Dial_nummer(telefoonnummer[i]);
          wait(100) ;
        END;
    END;
  writeln;
  GotoXY(4,18) ;
  writeln('                                                              ') ;
END;

PROCEDURE InitTabel;
  { zoek naar de file  'dial.dat' en leest hem in. Wanneer
    de file niet bestaat wordt de  tabel opnieuw berekend}

VAR
    fl               :file of tonen;

BEGIN
  Assign(fl,'dial.dat');
  {$I-} reset(fl); {$I+}
  IF ioresult = 0 THEN BEGIN
       Read(fl, sampels^);
       close(fl);
    END
  ELSE BEGIN
       Maak_tabel_voor_DAC;
       Assign(fl,'dial.dat');
       ReWrite(fl) ;
       Write(fl,sampels^);
       close(fl);
    END;
END;

PROCEDURE kader(x1,y1,x2,y2  :integer);
      { Tekent een kader op  het beeldscherm met x1,y1 als
        linker hoekpunt en x2,y2 als rechter hoekpunt }

VAR i, j   :integer;

BEGIN
  GotoXY(x1,y1);WRITE('+');
  FOR j:= x1+1 TO x2-1 DO BEGIN
     gotoXY(j,y1);
     write('-');
  END;
  GotoXY(x2,y1);WRITE('+');
  FOR i:= y1+1 TO y2-1 DO BEGIN
     GotoXY(x1,i);WRITE('|');
     GotoXY(x2,i);WRITE('|');
  END;
  GotoXY(x1,y2); WRITE('+');
```

```
  FOR j:= x1+1 TO x2-1 DO begin
      GotoXY( j , y2 ) ; WRITE('-');
   END;
   gotoXY(x2 , y2 );write('+');
END;

PROCEDURE Menu(var keuze: char);           { Menu Scherm }

BEGIN
  GotoXY(4, 7); WriteLn('[I]nvoer ');
  GotoXY(4, 9); WriteLn('[V]erwijder ');
  GotoXY(4, 11);WriteLn('[B]el ');
  GotoXY(4, 13);WriteLn('[E]inde ' );
  read(kbd, keuze);
END;

PROCEDURE GetName(var naam:string20);  { invoer van naam }

BEGIN
  GotoXY(4,18); Write(' Geef naam : ');
  ReadLn(naam);
  GotoXY(4,18);
  Write('                                          ');
END;

PROCEDURE GetTel(var tel:string20);
                             { Invoer van telefoonnummer }
BEGIN
 GotoXY(4,18);  Write(' Geef telefoonnummer : ');
 ReadLn(tel) ;
 GotoXY(4,18);
 Write('                                           ' ) ;
END;

PROCEDURE  printlijst(lijst,top, cursor:telefoon );
  { Drukt telefoonlijst op scherm af. Top wijst naar eerste
   record  en cursor wijst naar het record waar de cursor
   op  staat }

VAR  aantal    :integer;
     nonext    :boolean;

BEGIN
  aantal:=0;
  REPEAT
    GotoXY(21,7+aantal);           { wis eerst regel uit }
    Write

     ('                                                ' ) ;
    GotoXY(21,7+aantal) ;
    IF top=cursor THEN  BEGIN
        write('>') ;
        lowVideo;          { laat cursor record oplichten }
        Write(top^.naam);
        GotoXY(50,7+aantal); write(top^.telefoonnummer);
        NormVideo;
    END ELSE  BEGIN
```

```
        Write(' ',top^.naam);
        GotoXY(50,7+aantal); write(top^.telefoonnummer);
    END;
    next_record(lijst,top,nonext);      { volgend record }
    aantal:=aantal+1;
  UNTIL nonext  or (aantal>=regel);
                             { herhaal totdat kader vol is }
  WHILE aantal < regel  DO  BEGIN
    GotoXY(21,7+aantal);       { Veeg rest van kader uit }
    Write('                                              ') ;
    aantal:=aantal+1;
  END;
END;



PROCEDURE offset_tussen_pointer(top, bel:telefoon;
                                          var ver:integer);
  { bepaal aantal records tussen 2 pointers  (top en bel) }

VAR nonext   :boolean;

BEGIN
  ver:=0;
  WHILE top <> bel  DO BEGIN
     next_record(tellijst,top,nonext);
     ver:=ver+1;
  END;
END;

PROCEDURE  invoer (naam, tel:string20);
             { Voeg naam en telefoonnummer in de lijst in }
VAR    p, voor    :telefoon;
       exist       :boolean;
BEGIN
  Maak_record(p,naam,tel);          { maak een record aan }
  Search(tellijst,voor,naam,exist);
                              { zoek de plaats in de lijst }
  Insert(tellijst,voor,p);      { voeg record in de lijst }
  belptr:=tellijst;                       { init pointers }
  topptr:=tellijst;
  printlijst(tellijst,topptr,belptr);     { druk lijst af }
END;

PROCEDURE omhoog;
                   { Pijltje toets omhoog ingedrukt.
                     Pas de pointer aan en druk lijst af }
VAR noprev         :boolean;

BEGIN
  IF belptr=topptr THEN  { Als cursor boven in kader dan }
     prev_record(tellijst,topptr,noprev);{ nieuwe topptr }
  prev_record(tellijst,belptr,noprev);    { nieuwe belptr }
  printlijst(tellijst,topptr,belptr);     { druk lijst af }
END;

PROCEDURE omlaag;
```

```
                      { Pijltje toets omlaag ingedrukt.
                        Pas pointers aan en druk lijst af }
VAR nonext         :boolean;
    verschil       :integer;

BEGIN
  next_record(tellijst,belptr,nonext);
  offset_tussen_pointer(topptr,belptr,verschil);
  IF  (verschil=regel) and not nonext THEN
      next_record(tellijst,topptr,nonext);
         { Als cursor onderaan kader verplaats dan topptr }
  printlijst(tellijst,topptr,belptr);
END;

PROCEDURE verwijder;
                     { Verwijder het record waar de
                      cursor pointer  (belptr) naar wijst. }
VAR   del        :telefoon;
      nonext     :boolean;
BEGIN
  del:=belptr;  { Bewaar pointer dat gedelete moet worden }
  IF del=tellijst THEN BEGIN       { Het eerste record ? }
     omlaag;                    { Verplaats belptr omlaag }
     IF belptr=tellijst THEN BEGIN
                                  { enige record in lijst }
        delete(tellijst,del);            { delete pointer }
        belptr:=tellijst;   { zet belprt naar begin lijst }
        END
      ELSE delete(tellijst,del) ;
                     { er zijn meerdere records dus delete }
     topptr:=tellijst;      { zet topptr naar begin lijst }
    END
   ELSE BEGIN
     omhoog;                      { cursor pointer omhoog }
     delete(tellijst,del);                { delete record }
  END;
  printlijst(tellijst,topptr,belptr);    { update screen }
END;

PROCEDURE save(lijst  :telefoon);
                       { Schrijf telefoonlijst naar disk }
VAR  flrec           :filerec;
     fl              :file of filerec;
     p               :telefoon;
     nonext,noprev   :boolean;

BEGIN
  p:=lijst;
  next_record(lijst,p,nonext);
  prev_record(lijst,p,noprev);
  Assign(fl,'telklap.dat');
  Rewrite(fl);
  WHILE not nonext DO BEGIN
    flrec.naam:=p^.naam;
    flrec.tel:=p^.telefoonnummer;
    write(fl,flrec);
    next_record(lijst,p,nonext);
```

```
  END;
  close(fl);
END;

PROCEDURE  load(var lijst: telefoon);
                        { Lees eventueel telefoonbestand }
VAR   fl         :file of filerec;
      flrec       :filerec;

BEGIN
  Assign(fl,'telklap.dat');
  {$I-} reset(fl);  {$I+}
  IF  (ioresult = 0) and not eof(fl) THEN BEGIN
     REPEAT
       Read(fl,flrec);
       invoer(flrec.naam,flrec.tel);
     UNTIL eof(fl);
     close(fl);
  END;
END;

BEGIN                             { VAN HET HOOFDPROGRAMMA }
  dirty:=false;
  new(sampels);
              { maak ruimte vrij in heap voor de sampels }
  InitTabel;                           { vul sampel tabel }
  clrscr;                                          { CLS }
  GotoXY(22,2); WriteLn(' C.U.C. Telefoon Klapper V2.0 ');
  kader(20,6,71,7+regel);
  Init_lijst(tellijst);                { Maak lijst aan }
  load(tellijst);
  belptr:=tellijst;                    { Init. pointers }
  topptr:=tellijst;
  REPEAT                                    { HOOFD LUS }
    Menu(keuze);
    int_keuze:=ord(keuze);
    CASE int_keuze OF
       73,105  : BEGIN
                    GetName(naam);
                    GetTel(tel) ;
                    invoer(naam,tel);
                    dirty:=true;
                  END;
       86,118  : BEGIN
                    verwijder;
                    dirty:=true;
                 END;
       66,98   : Dial_string(belptr^.telefoonnummer);
       5,49,30       :
                       omhoog;
       24,50,31      : omlaag;
     END;
  UNTIL (keuze='E') or (keuze='e');
  IF dirty THEN save(tellijst) ;
END.
```