storno.h

```
/* Diverse Funktioner til Storno PRM6662   Version 2.0 */


Puls()
{
 unsigned char aa;
  AUTO=1;
  XBYTE[1]=ICC;
  for(aa=0;aa<0;); /* Delay */
  AUTO=0;
  XBYTE[1]=ICC;
}

RX()                                  /* RX seriel data fra strono 11 bits mode */
{
  SCON  = 0xD8;
  PCON  = 0x80 | PCON;
  TMOD  = 32;
  TH1   = 0xfa;
  TR1   = 1;
  ES    = 0;
  TI    = 1;
  TB8   = 1;
  PT1=1;
  EA=1;
}


Opdater_syn()                                 /* Sender data til 4093 og MC145159 */
{
unsigned int i,x;
unsigned char a;
SCL=0;

Puls();

for(i=2;i<16;)                                                  /* 14 bit R tæller */
    {
     x = R_teller<<i++>>15;
     SDA = x;
     SCL =1; SCL=0;
     }
for(i=6;i<16;)                                                  /* 10 bit N tæller */
    {
     x = N_teller<<i++>>15;
     SDA=x;
     SCL=1; SCL=0;
     }

for(i=1;i<8;)                                                     /* 7 bit A tæller */
    {
     x = A_teller<<i++>>7;
     SDA=x;
     SCL=1; SCL=0;
    }
                                    /* bit 32, bestemmer om R tæller skal endres */
SDA=1; for(x=1;x<20;x++); SCL=1; SCL=0;


SDA=0;
```

```
SCL=1;SCL=0;
SCL=1;SCL=0;
SCL=1;SCL=0;
SCL=1;SCL=0;
SCL=1;SCL=0;
SCL=1;SCL=0;
SCL=1;SCL=0;
SDA=1;
SCL=1;
STROBE=1;
STROBE=0;
}

Opdater_RF()
{
unsigned int i,x;
unsigned char f;
if(PTT==0) f=16+effekt;
   else f=effekt;

for(i=0;i<8;)                  /* 8 bit, ¢vrig radio funktioner på HF printet */
    {
     x = f<<i++>>7;
     SDA=x;
     SCL=1; SCL=0;
    }
STROBE=1;
STROBE=0;

if(PTT==0) TX_ADMIT=0;                                  /* Sikkerheds funktion */
   else TX_ADMIT=1;
XBYTE[1]=ICC;
}

S_beep()                                              /* laver 2 beep i h¢jtaleren */
{
unsigned int i;
if(XBYTE[BEEP]==3)                                              /* Ved beep on */
 {
 XBYTE[0]=16+64+128;
 for(i=0;i<10000;i++);
 XBYTE[0]=ICB;
 XBYTE[0]=16+0+128;
 for(i=0;i<10000;i++);
 XBYTE[0]=ICB+volume;
 }
}

Key_beep()                                           /* Keybord 1 beep i h¢jtaleren */
{
unsigned int i;
 XBYTE[0]=16+64+128;
 for(i=0;i<10000;i++);
 XBYTE[0]=ICB+volume;
}

Set_parity(unsigned int t)
{
 unsigned char x;
 unsigned int res=0;
```

```
 for (x=0; x<8; x++, t>>=1) res += (t & 0x01);
 res &= 0x01;
 if(res==0) TB8=0;                                    /* sætter party bit */
   else TB8=1;
}



Poll()                        /* Sender Poll kommando til slave og henter svar */
{
 unsigned int i;
 DIR=0;
 for(i=0;i<80;i++); /* delay */
 Set_parity(224);
 putchar(224);
 for(i=0;i<80;i++); /* delay */
 Set_parity(32);
 putchar(32);
 for(i=0;i<80;i++); /* delay */

 ES=1;                                              /* Tænder Seriel interrupt */
 DIR=1;
 for(i=0;i<30;i++);           /* Venter på at Slanve sender data til buffer */
 ES=0;                                             /* Slukker Seriel interrupt */
}


Init_disp()                                         /* default power on */
{
unsigned int i;
DIR=0;
for(i=0;i<140;i++);
Set_parity(96);
putchar(96);
for(i=0;i<140;i++);
Set_parity(16);
putchar(16);
for(i=0;i<140;i++);
Set_parity(1);
putchar(1);
for(i=0;i<140;i++);
Set_parity(142);
putchar(142);
for(i=0;i<140;i++);
DIR=1;
}

Tjek_svar()                    /* tjekker om slave har modtager data korekt */
{
unsigned char w;
teller=0;
 ES=1;                                              /* Tænder Seriel interrupt */
 DIR=1;
 for(w=0;w<30;w++);           /* Venter på at Slanve sender data til buffer */
 ES=0;                                             /* Slukker Seriel interrupt */
teller=0;
if(buffer[0]==32) return(0);
return(1);
}
```

```
/* Sender 2 data byte til display */
Data_to_disp2(unsigned int byte1,byte2)
{
 unsigned char p_byte;
 unsigned int i;

FEJL:
Puls();
DIR=0;
TB8=0;
putchar(96);

TB8=1;
putchar(35);
p_byte=67;
Set_parity(byte1);
putchar(byte1);
p_byte=p_byte^byte1;

Set_parity(byte2);
putchar(byte2);
p_byte=p_byte^byte2;

Set_parity(~p_byte);
putchar(~p_byte);
for(i=0;i<80;i++);
if(Tjek_svar()==1) goto FEJL;
}



/* Sender flere data byte til disp */
Data_to_disp(unsigned int type,pos,unsigned char mess[16])
{
unsigned int i,a,antal_data;
unsigned char h,p_byte;
FEJL:
Puls();
DIR=0;
antal_data=strlen(mess);

TB8=0;
putchar(96);
p_byte=96;

h=(antal_data+2<<4)+3;        /* antal data byte + mess type, 3 til display */
Set_parity(h);
putchar(h);
p_byte=p_byte^h;

Set_parity(type);                                        /* Funktions type */
putchar(type);
p_byte=p_byte^type;

Set_parity(pos);                                              /* Possion */
putchar(pos);
p_byte=p_byte^pos;

for(a=0;a<antal_data;)
{
```

```
 Set_parity(mess[a]);
 putchar(mess[a]);
 p_byte=p_byte^mess[a++];
}

Set_parity(~p_byte);
putchar(~p_byte);
for(i=0;i<80;i++);
if(Tjek_svar()==1) goto FEJL;          /* Tjekker om pakken er modtager ok */
}


Data_to_disp1(unsigned int type)
{
unsigned int i;
unsigned char p_byte;
FEJL:
Puls();
DIR=0;

TB8=0;
putchar(96);
TB8=1;
putchar(19);
p_byte=115;
Set_parity(type);
putchar(type);
p_byte=p_byte^type;

Set_parity(~p_byte);
putchar(~p_byte);
for(i=0;i<80;i++);
if(Tjek_svar()==1) goto FEJL;          /* Tjekker om pakken er modtager ok */
}


Data_to_audio(unsigned char type) /* Sender 1 byte til Audio controler i R¥R*/
{
unsigned int i;
unsigned char p_byte;
FEJL:
Puls();
DIR=0;

TB8=0;
putchar(96);
p_byte=96;

TB8=1;
putchar(22);
p_byte=p_byte^22;

Set_parity(type);
putchar(type);
p_byte=p_byte^type;

Set_parity(~p_byte);
putchar(~p_byte);
for(i=0;i<80;i++);
if(Tjek_svar()==1) goto FEJL;          /* Tjekker om pakken er modtager ok */
```

```
}

                              /* endret fra 16 til 17 i V2h */
Skriv_string1(unsigned char str[16])              /* Skriver tekst fra H¢jre */
{
unsigned char i;          /* endret for char til int i V2h */
for(i=0;i<16;i++) Data_to_disp2(0xc,str[i]);
}

Nummer_tilt_A(unsigned char nr)
{
 if(nr==3) return(1);
 if(nr==6) return(2);
 if(nr==0) return(3);
 if(nr==2) return(4);
 if(nr==1) return(5);
 if(nr==8) return(6);
 if(nr==4) return(7);
 if(nr==5) return(8);
 if(nr==7) return(9);
 if(nr==9) return(0);
}

Nummer_tilt_B(unsigned char nr)
{
 if(nr==5) return(1);
 if(nr==4) return(2);
 if(nr==0) return(3);
 if(nr==1) return(4);
 if(nr==8) return(5);
 if(nr==9) return(6);
 if(nr==7) return(7);
 if(nr==2) return(8);
 if(nr==6) return(9);
 if(nr==3) return(0);
}

Nummer_tilt_C(unsigned char nr)
{
 if(nr==4) return(1);
 if(nr==5) return(2);
 if(nr==0) return(3);
 if(nr==8) return(4);
 if(nr==3) return(5);
 if(nr==6) return(6);
 if(nr==9) return(7);
 if(nr==7) return(8);
 if(nr==2) return(9);
 if(nr==1) return(0);
}



CRC_call()
{
unsigned char crca=0,crcb=0,crcc=0,ch,string[6],a,w;

string[0]=XBYTE[TEKST_15];
string[1]=XBYTE[TEKST_14];
string[3]=XBYTE[TEKST_10];
```

```
string[6]=XBYTE[TEKST_16];
string[4]=XBYTE[TEKST_5];
string[5]=XBYTE[TEKST_4];
string[2]=XBYTE[TEKST_9];

for(w=0;w<7;)
 {
  a=string[w];
  crca=crca+a<<1>>1;
  a=string[w];
  crcb=crcb+~a;
  a=string[w];
  crcc=crcc+a<<2;
  w++;
 }

if(crcc !=XBYTE[TEKST_1 ]*100 + XBYTE[TEKST_2 ]*10 + XBYTE[TEKST_3 ]) return(0)
if(crca !=XBYTE[TEKST_6 ]*100 + XBYTE[TEKST_7 ]*10 + XBYTE[TEKST_8 ]) return(0)
if(crcb !=XBYTE[TEKST_11]*100 + XBYTE[TEKST_12]*10 + XBYTE[TEKST_13]) return(0)

return(1);
}


Key_hent(bit q)   /* Venter på key tryk,  normal 0 ved 1 konveteres til nr */
{
D:
while(T1!=0) Puls();                                  /* Venter på data fra r¢r */
 {
 Poll();                                /* Extra poll, for at t¢mme salve buffer */
 if(buffer[0]==96 && buffer[1]==17)                     /* keyb input fra slave */
   {
    Puls();
    if(buffer[2]>>6==1)
      {
       Poll();
       if(beep==0) Key_beep();                                  /* key beep ON */
       if(q==0) return(buffer[2]<<2>>2);
         else switch(buffer[2]<<2>>2)                    /* Konveter til nr */
                    {
                     case  0: return(55);
                     case  1: return(56);
                     case  2: return(57);
                     case  4: return(52);
                     case  5: return(53);
                     case  6: return(54);
                     case  7: return(1);                        /* key 'C' */
                     case  8: return(49);
                     case  9: return(50);
                     case 10: return(51);
                     case 18: return(2);                       /* key 'CL' */
                     case 20: return(48);
                    } goto D;        /* Hvis der trykkes andet en nr taster */


      }
   }
 }
goto D;
}
```

```
Fejl(unsigned int f)
{
 unsigned int x;
 Puls();
 if(f==1)
    {
     Data_to_disp(14,3,"LOW PO");
     Data_to_disp(14,9,"WER !");

     while(LOWPOWER==0) Puls();
     Data_to_disp(14,3,"      ");
     Data_to_disp(14,9,"      ");
     opdate=10;
     return(0);
    }
 if(f==2) /* Fejl ved eeprom */
   {
    Data_to_disp(14,2 ,"FEJL ");
    Data_to_disp(14,7 ,"VED ");
    Data_to_disp(14,11,"EEPROM");
    while(1) Puls();
   }
}

EEgem(unsigned int adresse,dta)                    /* Gemmer data i EEprom */
{
unsigned int a;

XBYTE[adresse]=dta;
for(a=0;a<6000;a++);
if(dta!=XBYTE[adresse]) Fejl(2);          /* Tjekker om data er blevet gemt */
}



/*                                                                    L
                                                                      Y
   DATA TIL MENU                                                      S */
code char  on[20][6]       ={" ","ON" ,"ON" ,"ON" ,"ON" ,"ON" ,"ON" ,"MER","ON"
code char dat_on[20][1]   ={  0 ,0    ,1    ,0    ,3    ,1    ,1    ,1    ,3
code char off[20][6]       ={" ","OFF","OFF","OFF","OFF","OFF","OFF","MER","OFF
code char dat_of[20][1]   ={  0 ,1    ,0    ,1    ,4    ,2    ,0    ,1    ,4

Menu()
{
 unsigned int ia;
 unsigned char pdata tekst[1],ch,fnu=0,fnu1;
/* Data_to_disp(14,6,"10");  v 2e*/
 if(CRC_call()==1) fnu1=12;              /* Ved Super bruger 12 menu punkter */
   else fnu1=7;                            /* Almindig bruger 9 menu punkter */
 Data_to_disp2(7,8);                                  /* Blinker med lampe */
/* Data_to_disp(14,6,"11"); v 2e*/
/* fnu=1;   v 2f */
/*Skriv_string1("TEST    :   "); V2g */

LOOP:
/*goto LOOP;  V2g */
```

```
tekst[0]=0; Data_to_disp(9,16,tekst);   /* undladt i s2c*/  /* Sletter disp */

switch(fnu)
 {
  case 1:  Skriv_string1("MIKROFON:       "); break;
  case 2:  Skriv_string1("HANDSFREE:      "); break;
  case 3:  Skriv_string1("H\\JTTALER:      ");break;
  case 4:  Skriv_string1("R\\R MIC:        ");break;
  case 5:  Skriv_string1("R\\R LF:}        ");break;
  case 6:  Skriv_string1("SPACING:        "); break;
  case 7:  Skriv_string1("SET REG. NUMMER"); break;
/* Version F Super bruger funktioner */
  case 8:  Skriv_string1("KEYB LYS:       "); break;
  case 9: Skriv_string1("A. SPACING:     "); break;
  case 10: Skriv_string1("SPACING +/-:    "); break;
  case 11: Skriv_string1("START BEEP:     "); break;
  case 12: Skriv_string1("KEY BEEP:       "); break;
  case 13: Skriv_string1("SPACING afstand"); break; /* ikke ferdig */
 }

LOOP1:
Puls();

                    /* For at undgå at de forskellig spacing mode konflikter */
if(XBYTE[SPACING]==1 && XBYTE[AUTO_SPACING]==1) EEgem(AUTO_SPACING,0);
if(XBYTE[AUTO_SPACING]==1)
  {
   if(XBYTE[SPACING_PN]!=0)  EEgem(SPACING_PN,0);              /* - spacing */
   if(XBYTE[SPACING_KA]!=64) EEgem(SPACING_KA,64);
  }


if(XBYTE[BASE+fnu]==dat_on[fnu][0])  Data_to_disp(14,13, on[fnu]);
   else Data_to_disp(14,13,off[fnu]);

ch=Key_hent(0);


if(ch==19)  fnu++;                                          /* Næste funktion */
if(ch==18 || fnu>fnu1)          /* Går tilbage ved CL eller gennem bladning */
   {
    teller=0;
     Data_to_disp2(6,8);                                /* Blinker med lampe off */
     tekst[1]=0; tekst[0]=1; Data_to_disp(9,16,tekst);      /* Sletter disp */
     Data_to_audio(XBYTE[ROR_LF]);
     Data_to_audio(XBYTE[ROR_MIC]);
     return(0);                                          /* Return to Varmboot */
   }


if(ch==3 && fnu==7)                          /* Standart opsætning            */
  {                                 /* Ved 'SET REG. NUMMER' og tryk på 'D' */
   unsigned int vent;
   EEgem(TJECK,255);
   Puls();Skriv_string1(" EEPROM RESET 6 ");
   for(vent=0;vent<10;vent++) EEgem(16470+vent,0);
   for(vent=0;vent<10;vent++) EEgem(16480+vent,0);
   Puls();Skriv_string1(" EEPROM RESET 4 ");
   for(vent=0;vent<100;vent++) Puls();
   for(vent=0;vent<10;vent++) EEgem(16500+vent,0);
```

```
   Puls();Skriv_string1(" EEPROM RESET 3 ");
   for(vent=0;vent<10;vent++) EEgem(16510+vent,0);
   Puls();Skriv_string1(" EEPROM RESET 2 ");
   for(vent=0;vent<100;vent++) Puls();
   for(vent=0;vent<10;vent++) EEgem(16520+vent,0);
   Puls();Skriv_string1(" EEPROM RESET 1 ");
   for(vent=0;vent<10;vent++) EEgem(16530+vent,0);
   while(1);
  }

if(ch==17 && fnu==7)                /* Ved 'SET REG.NUMMER' og tryk på 'TLFR¥R' *
  {
   unsigned int vent;
   tekst[1]=0;tekst[0]=0; Data_to_disp(9,16,tekst);          /* Sletter disp */
   tekst[0]=1; Data_to_disp(0xf,4,tekst);                  /* Cursor on pos 8 */

  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_1]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_2]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_3]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_4]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_5]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_6]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_7]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */

  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_8]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_9]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_10]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_11]=ch-48;
   for(vent=0;vent<6000;vent++);
```

```
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_12]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_13]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_14]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_15]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
  ch=Key_hent(1); teller=0;
   XBYTE[TEKST_16]=ch-48;
   for(vent=0;vent<6000;vent++);
   Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */

   tekst[1]=0; tekst[0]=0; Data_to_disp(9,16,tekst);        /* Sletter disp */
   tekst[0]=0; Data_to_disp(0xf,0,tekst);                 /* Cursor on pos 8 */
}



if(ch==16)                                                /* Gemmer ON/OFF verdi */
   {
    if(XBYTE[BASE+fnu]==dat_on[fnu][0]) EEgem(BASE+fnu,dat_of[fnu][0]);
    else EEgem(BASE+fnu,dat_on[fnu][0]);
    goto LOOP1;
   }

goto LOOP;
}

Frekvens_indtasting()                              /* Virker lige som på IC32ET */
{
unsigned long frekvens;
unsigned char tekst[1],ch=0;
tekst[1]=0;
tekst[0]=0; Data_to_disp(9,16,tekst);                    /* Sletter disp */
tekst[0]=8; Data_to_disp(0xf,4,tekst);                 /* Cursor on pos 8 */
Data_to_disp(14,6,"43");

teller=0;
ch=Key_hent(1); teller=0;

if(ch<50) ch=50;                                        /* ved under 432 MHz */

frekvens= (ch-48)*1000;
Data_to_disp2(0x11,ch);
Data_to_disp2(0x11,46);                                  /* laver komma */

ch=Key_hent(1); teller=0;
frekvens+= (ch-48)*100;
Data_to_disp2(0x11,ch);                            /* Skriver tal på cursor pos */
```

```
fejl1:

ch=Key_hent(1); teller=0;
if(ch==52 || ch==57) goto fejl1;                    /* ved fejl indtasting */
frekvens+= (ch-48)*10;
Data_to_disp2(0x11,ch);                          /* Skriver tal på cursor pos */

                              /* 25KHz */
if(ch==53) ch=48;
else
if(ch==50 || ch==55) ch=53;
else                          /* 12.5KHz */
if(ch==49 || ch==54) ch=50;
else
if(ch==51 || ch==53) ch=55;

frekvens+=ch-48;

Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */

if(ch==50 || ch==55) /* ved 12.5 */
   {
    ch=53;
    step=2;
    frekvens*=10;
    frekvens+=ch-48;
    Data_to_disp2(0x11,ch); /* Skriver tal på cursor pos */
   }
  else step=1;

if(step==1) kanal= (frekvens-2000)/25;
   else kanal= (frekvens-20000)/125;

if(kanal>MAX_FREKVENS*step) kanal=MAX_FREKVENS*step;          /* Bånd stop */
if(XBYTE[STEP]!=step)                    /* Gemmer Step i EEprom ved endring */
   {
    XBYTE[STEP]=step;
    for(frekvens=0;frekvens<2000;frekvens++);
   }
tekst[1]=0;tekst[0]=0; Data_to_disp(0xf,0,tekst);       /* Slukker Cursor */
opdate=10;
tekst[1]=0; tekst[0]=0; Data_to_disp(9,16,tekst);         /* Sletter disp */
}

code char start[]=

"P[6BGK!!.!!W/3/1!"
"!!!!!!!!!!!!!!!!!!";

Start_tekst()
{
unsigned int i,t;
unsigned char s_meter;
for(i=0;i<36;)
  {
  for(t=0;t<6000;t++);
  Puls();
  s_meter+=start[i]-1;
  Puls();
```

```
  Data_to_disp2(0xc,start[i++]-1);
  Puls();
  }
ERR=s_meter;
i=s_meter;                          /* Tjekker om start tekst er blevet endret */
Puls();

if(i!=ERR_DATA)
   {
   Puls();Skriv_string1(" ERROR 1203     ");
   while(1) Puls();
  }
  else s_tekst=1;                   /* Giver besked om atstart teksten er k¢rt */
/*
  sprintf(buffer,"%u",i);
  Data_to_disp(14,6,buffer);
  while(1);
*/
}


code char start1[]=
"DENNE STATION TILH\\RER: OZ"
"                       ";

code char start2[]=
" DETTE SOFTWARE ER PROGRAMMERET AF OZ5AFJ - BO CAR\\E, VIBORGVEJ 133, 8600"
" SILKEBORG, DK.                  SOFTWAREN M] KUN BENYTTES AF RADIO AMAT\\RER MED
"           DER M] P] INGEN M]DE [NDRES I SOFTWAREN.                "
"  (C) COPYRIGHT OZ5AFJ 1999."
"                               ";

Bruger_tekst()
{
unsigned int i,t;
unsigned char s_meter,te[4];

if(CRC_call()==1)
 {
  te[0]=Nummer_tilt_A(XBYTE[TEKST_15])+48; /* 5  */

  t=10* Nummer_tilt_B(XBYTE[TEKST_14]) + Nummer_tilt_A(XBYTE[TEKST_9 ]);
  te[1]=t;   /* A */
  t=10*Nummer_tilt_C(XBYTE[TEKST_4 ]) + Nummer_tilt_A(XBYTE[TEKST_16]);
  te[3]=t;  /* J */
  t=10*Nummer_tilt_C(XBYTE[TEKST_10]) + Nummer_tilt_B(XBYTE[TEKST_5 ]);
  te[2]=t; /* F */

 if(te[3]<65 || te[3]>90) te[3]=32;            /* Hvis call kun er på 5 sifre *
 for(i=0;i<47;)
  {
  for(t=0;t<5000;t++);
  Puls();
  if(i<27) s_meter+=start1[i];
  Puls();
  if(i<26 || i>30) Data_to_disp2(0xc,start1[i]);
  if(i==27) Data_to_disp2(0xc,te[0]);
  if(i==28) Data_to_disp2(0xc,te[1]);
  if(i==29) Data_to_disp2(0xc,te[2]);
  if(i==30) Data_to_disp2(0xc,te[3]);
```

```
  i++;
  Puls();
  }
  i=s_meter;                          /* Tjekker om Bruger tekst er blevet endret *
  Puls();
 if(i!=101)
   {
    Puls();Skriv_string1(" ERROR 1202     ");
    while(1) Puls();  /* SIKRING */
   }
/*
  sprintf(te,"%u",i);
  Data_to_disp(14,6,te);
while(1);
*/
 }
else  /* Ikke betalende bruger */
 {
/*
unsigned tx[6];
 i=strlen(start2);
sprintf(tx," %u ",i);
  Data_to_disp(14,6,tx);
while(1);
*/
  for(i=0;i<283;)
  {
  for(t=0;t<6000;t++);
  Puls();
  s_meter+=start2[i];
  Puls();
  Data_to_disp2(0xc,start2[i++]);
  Puls();
 }
 i=s_meter;                          /* Tjekker om Bruger tekst er blevet endret */
Puls();

if(i!=80)
  {
   Puls();Skriv_string1(" ERROR 1201     ");
   while(1) Puls();
  }

/*
  sprintf(te,"%u",i);
  Data_to_disp(14,6,te);
while(1);
*/
 }
}

Gem_paa(unsigned char mem_pos)
{
unsigned char pdata ch,pos,c[1];
c[1]='\0';
Skriv_string1("GEM I NR.  ? 1  ");

while(1)
 {
  ch=c[0];
```

```
  c[0]=Key_hent(1);
  if(c[0]==1) return(ch-48);
  if(c[0]==2) return(255);                              /* Ved anuler med 'CL' */
  Data_to_disp(14,14,c);                                   /* EXIT ved key 'C' */
 }

}
/*"              DET ER MULIGT AT BLIVE REGISTRERET BRUGER !                "*/
/*" FORDELEN ER AT STATIONEN F]R NOGLE FLERE FUNKTIONER S]DAN SOM:          "*/
/*" AUTOMATISK REPEATER SPACING,  KEYB LYS,  START BEEP,  KEYB BEEP, "*/
/*" HUKOMMELSESKANALER,  1800HZ TONE,  +/-SPACING,  OSV."*/
/*"ALT DET KAN MAN F] FOR 100 KR, SOM MAN SENDER TIL BO CAR\\E, VIBORGVEJ 133,"
/*" 8600 SILKEBORG, DK.                  ";*/

code char info1[]=
"!!!!!!!!!!!!!EFU!FS!NVMJHU!BU!CMJWF!SFHJTUSFSFU!CSVHFS!\"!!!!!!!!!!!!!"
"!GPSEFMFO!FS!BU!TUBUJPOFO!G^S!OPHMF!GMFSF!GVOLUJPMFS!T^EBO!TPN;!!!!!!!!"
"!BVUPNBUJTL!SFQFBUFS!TQBDJOH-!!LFZC!MZT-!!TUBSU!CFFQ-!!LFZC!CFFQ-!"
"!IVLPNNFMTFTLBOBMFS-!!2911I[!UPOF-!!,0.TQBDJOH-!!PTW/"
"!!!!!!!!!!!!!"
"BMU!EFU!LBO!NBO!G^!GPS!211!LS/!TPN!NBO!TFOEFS!UJM!CP!DBS]F-WJCPSHWFK!244-"
"!9711!TJMLFCPSH-!EL/!!!!!!!!!!!!!!!!!";

Info()
{
  unsigned int pdata i,t;
  unsigned char pdata s_meter=0;
/*
unsigned te[4];
unsigned tx[6];
 i=strlen(info1);
sprintf(tx," %u ",i);
  Data_to_disp(14,6,tx);
while(1);*/

  for(i=0;i<381;)
  {
  for(t=0;t<5000;t++);
  Puls();
  s_meter+=info1[i]-1;
  Puls();
  Data_to_disp2(0xc,info1[i++]-1);
  Puls();
  }
 i=s_meter;                        /* Tjekker om Bruger tekst er blevet endret */

if(i!=14)
    {
     Puls();Skriv_string1(" ERROR 1200     ");
     while(1) Puls();
    }
Puls;
/*

sprintf(te,"%u",i);
Data_to_disp(14,6,te);
while(1);
*/

}
```