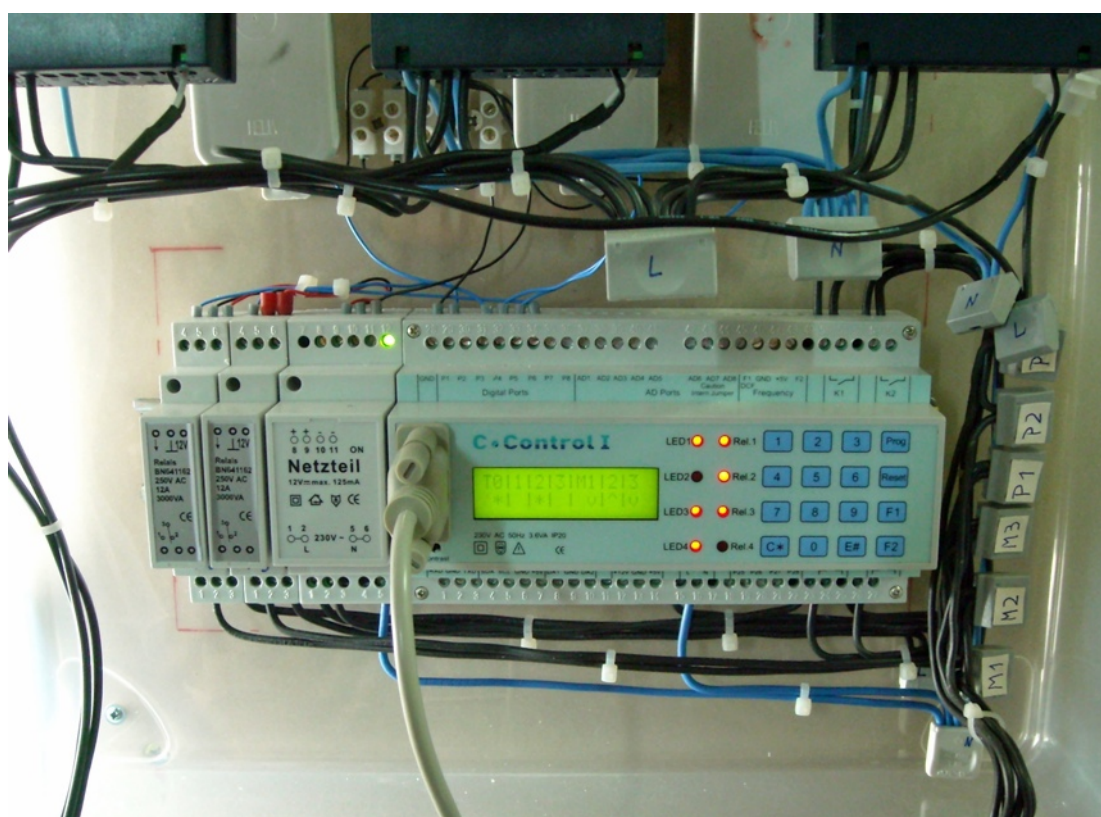


Kurzanleitung

Intelligente Heizungssteuerung V1.0

Stefan Schwope / März 2009



Das System

Die intelligente Heizungssteuerung entstand aus der Idee heraus die Pumpenlaufzeiten und den Wärmetransport in einem Haus zu optimieren. Betriebskosten werden so eingespart. Die Regelung ist speziell auf den Haushalt des Kunden ausgerichtet, und wurde von ihm - nach manuellen Praxistests im Haushalt - vorgeschlagen. Drei Heizkreisläufe müssen optimal mit Warmwasser versorgt werden. Abbildung 1 zeigt die Struktur des Heizsystems.

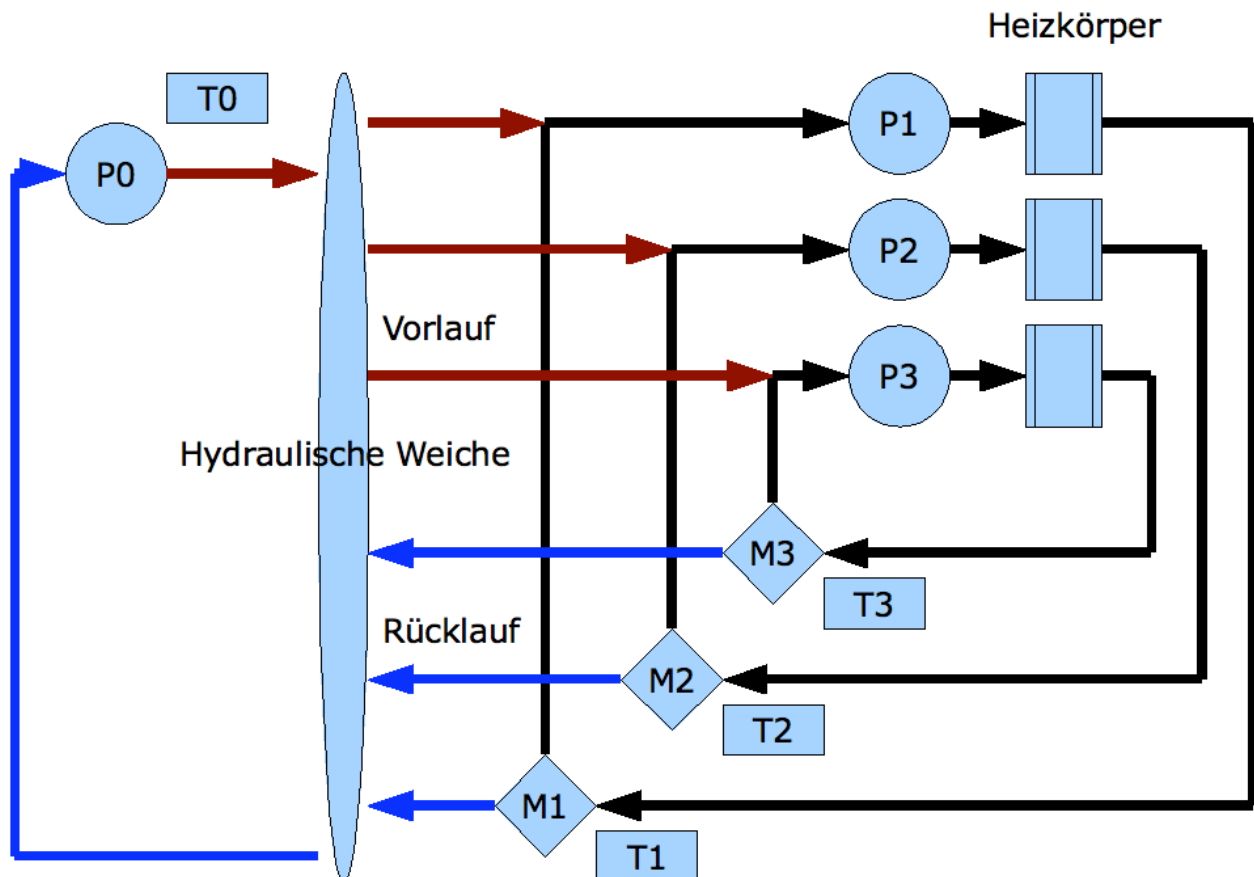


Abbildung 1

Vier Temperatursensoren messen die Temperatur des Heizwassers im Warmwasserspeicher (T0) und an den Rückläufen des jeweiligen Etagen-Rücklaufs (T1, T2, T3). Die Mischer M1, M2 und M3 öffnen die einzelnen Heizkreisläufe zum Warmwasserspeicher, um Warmwasser aufzunehmen oder öffnen die Heizkreisläufe zur entsprechenden Etage um das Wasser im Etagenkreislauf zu halten.

Die Pumpe P0 befördert das Wasser aus dem Wasserspeicher in die einzelnen Kreisläufe und die Pumpen P1, P2 und P3 pumpen das Wasser innerhalb der jeweiligen Etagenkreisläufe. Um das System redundant gegenüber der Regelungs-Elektronik zu machen, sind alle Verbraucher steckbar an das

System angeschlossen, und können bei Bedarf auf eine Zeitsteuerung umkonfiguriert werden.

Die Komponenten

Zur Umsetzung des oben beschriebenen Regelungssystem in die Realität wurden die C-Control Produkte der Firma Conrad verwendet, da sie - im Gegensatz zur Konkurrenz - günstig, erweiterungsfähig und einfach zu programmieren sind. Folgende Grundkomponenten wurden gewählt:

- Eine C-Control I zur Steuerung von P0 und zum Auslesen des Temperaturreglers T0 im Heizungsraum. Der Heizungsraum befindet sich in einem eigenen Gebäude. Da aus Platzgründen die P0/T0 Anschlüsse nicht direkt in das Haus gelegt werden konnten, wurde mit nur einem Kabel eine RS232-Verbindung gelegt, über die die C-Control I mit einer C-Control Station angesteuert (P0) bzw. ausgelesen (T0) werden kann.
- Eine C-Control Station als Hauptrechner für die Regelungs-Logik und die Ansteuerung der Haus-Etagenkreisläufe.
- Vier Fox-Temperaturregler zur Messung der Kreislauftemperaturen.
- Vier Optokoppler zur Pegelumsetzung (230V Fox-Ausgang zu digitalem 5V Eingang).
- Sieben Relais (zusätzlich zu den Relais in den beiden C-Control Rechnern) zur Ansteuerung von Mischern und Pumpen.

Die Fox-Temperaturregler für die einzelnen Etagen sind über der C-Control Station angebracht, wie in Abbildung 2 zu sehen.



Abbildung 2

Die C-Control Station Hauptlogik im Haus und der C-Control I Steuerungs-Rechner im Heizungsraum sind in den Abbildungen 3 und 4 zu sehen.



Abbildung 3 / C-Control Station

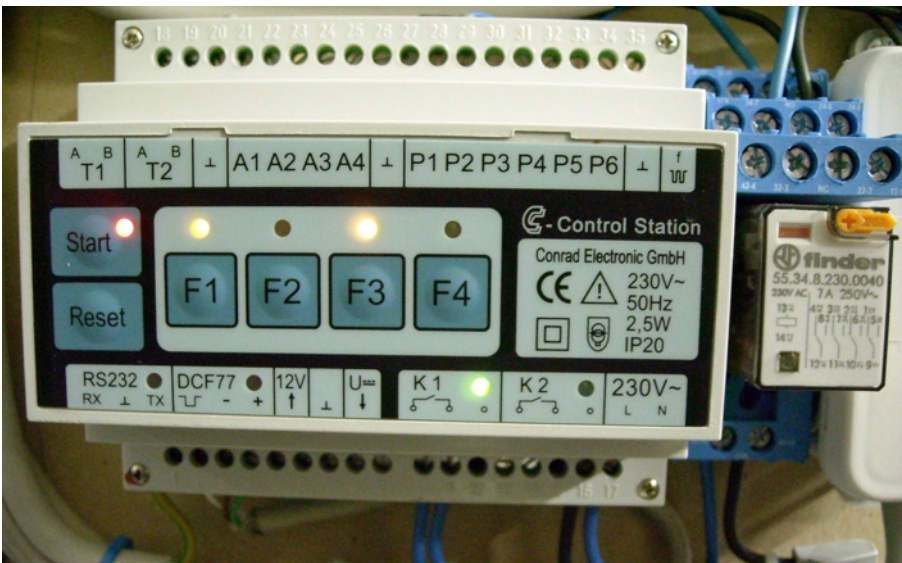


Abbildung 4 / C-Control I

Die Regelung

Folgende Regelung wurde zusammen mit dem Kunden besprochen:

- (1) wenn T0 eingeschaltet und M1 offen und M2 offen und M3 offen, dann P0 einschalten und zwar insgesamt maximal x Minuten (abhängig vom eingestellten Programm 60, 120 oder 180 Minuten) pro Tageszeitintervall
- (2) sofort ausschalten wenn alle Mischer offen für die einzelnen Etagen
- (3) dieser Mechanismus soll nur bei bestimmten Tageszeiten laufen, ansonsten P0 aus - P1, P2, P3 aus - M1, M2, M3 offen für Wärmespeicher
- (4) P0 muss 1 Minute laufen bevor die anderen Pumpen anlaufen
- (5) Px einschalten wenn T0 ein und Mx offen für Wärmespeicher
- (6) wenn Tx ein, dann Mx offen für Etagen und Px im Intervall schalten (Intervall 1-10 Minuten einstellbar)
- (7) wenn Mx offen für Wärmespeicher, dann Px ausschalten

Die Programme zur Umsetzung der Logik für die C-Control I und die C-Control Station befinden sich im Anhang.

Die Einstellung

Die Fox-Temperaturregler T0, T1, T2, T3 müssen voreingestellt werden. Die Einstellmöglichkeiten sind im beiliegenden Datenblatt von Conrad Elektronik (198545-an-01-ml-TEMPERATURREGLER_FOX-1004_de-en-fr-nl.pdf) nachzulesen.

Beim Einschalten der C-Control Station muss die augenblickliche Uhrzeit eingestellt werden, damit die Tageszeitintervalle korrekt erkannt werden (siehe nächstes Kapitel).

Bedienung C-Control Station (große C-Control)

Die C-Control Station kann eingestellt werden, indem die Tasten aus Tabelle 1 solange betätigt werden, bis die bezeichnete Wirkung eintritt. Den Status der einzelnen Komponenten kann man an den LEDs der C-Control Station - wie in Tabelle 2 zu sehen - ablesen.

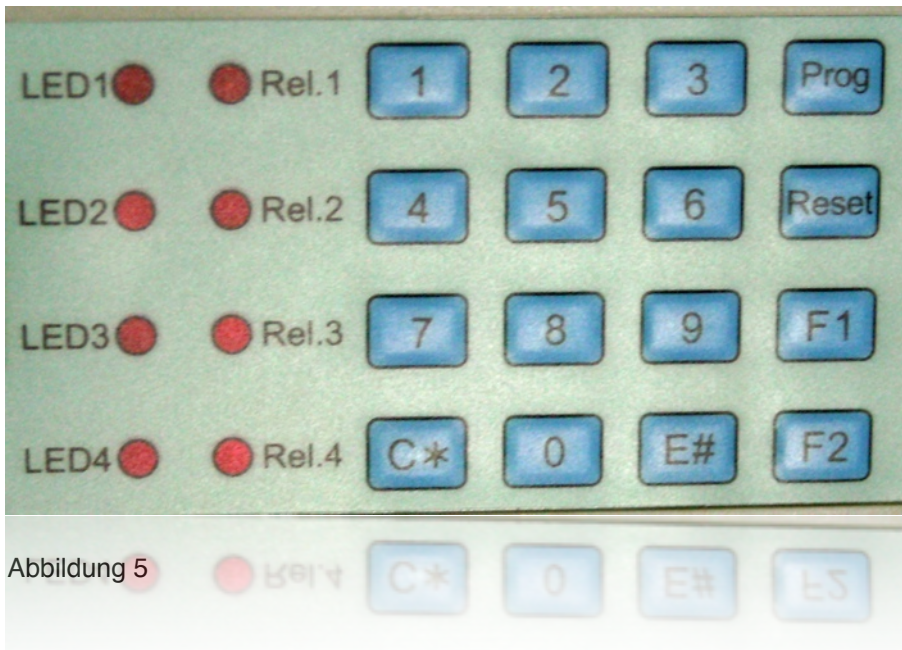
Taste an der C-Control Station	Bedeutung
Taste 1	Programm 1 (Nachlaufzeit von 60M)
Taste 2	Programm 2 (Nachlaufzeit von 120M)
Taste 3	Programm 3 (Nachlaufzeit von 180M)
Taste 9	Displaybeleuchtung ein- bzw. ausschalten
Taste c*	zum Menü wechseln
Taste E#	nur im Menü gültig: Wert hoch zählen
Taste F1	nur im Menü gültig: letzten Menüpunkt wählen
Taste F2	nur im Menü gültig: nächsten Menüpunkt wählen

Tabelle 1

LED/REL-LED Leuchten der C-Control Station	Bedeutung
LED1 an	Mischer 1 offen für Etage
LED2 an	Mischer 2 offen für Etage
LED3 an	Mischer 3 offen für Etage
LED4 blinkt	Lebenssignal der Steuersoftware
REL1-LED an	P3 eingeschaltet
REL2-LED an	P2 eingeschaltet
REL3-LED an	P1 eingeschaltet
REL4-LED an/aus	Status M3 (siehe LED3)

Tabelle 2

Die Lage der Tasten und LEDs an der C-Control Station ist in Abbildung 5 zu sehen.



Folgende Parameter können im Menü mit den Tasten E#, F1 und F2 verändert werden:

- Zeit Stunde (Tageszeit)
- Zeit Minute (Tageszeit)
- P0 Nachlauf (Maximale Gesamtlaufzeit der Pumpe P0 innerhalb jedes Tageszeitintervalls)
- Px Intervall (Intervallzeit der Pumpen P1, P2 und P3 bei geschlossenen Etagenkreislauf)
- Menü verlassen

Durch drücken der Taste „Reset“ wird ein Neustart des Systems veranlasst (siehe Handbuch C-Control Station).

Um Programmplatz in der C-Control Station zu sparen sind die Tageszeitintervalle, in denen die Regelungslogik eingeschaltet sein soll, fest im Programm codiert. In Tabelle 3 sind die Tageszeitintervalle aufgeführt.

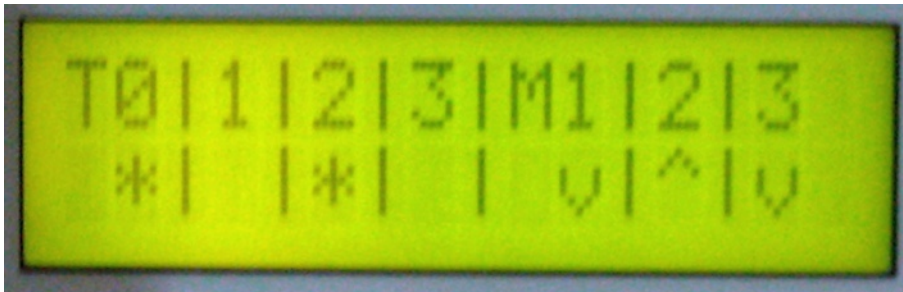
Startzeit	Endzeit
02:00	03:30
07:00	10:30
13:00	14:30
16:00	18:00
20:00	22:30

Tabelle 3

LCD-Anzeige an der C-Control Station

Im Normalbetrieb zeigt die C-Control Station abwechselnd folgende beiden Informations-Seiten.

Seite 1:



Seite 1 zeigt den Schaltstatus der Temperaturregler (T0, T1, T2, T3) und der Mischer (M1, M2, M3). Sterne (*) zeigen dabei „Ein“ an. „v“ bzw. „^“ zeigen die Öffnungsrichtung der Mischer an („v“: geöffnet zum Wärmespeicher, „^“: geöffnet zum Etagenkreislauf).

Seite 2:



Seite 2 zeigt den Status der Pumpen (P0, P1, P2, P3) und das Erreichen der eingestellten Tageszeitintervalle (IT) an. Sterne (*) zeigen dabei „Ein“ an.

Bedienung C-Control I (kleine C-Control)

Die Bedeutung der vier Tasten an der C-Control I sind in Tabelle 4 aufgeführt.

Taste an der C-Control I	Bedeutung
Taste F1	augenblicklichen Schaltstatus von P0 wechseln (ein->aus, aus->ein)
Taste F2	augenblicklichen Schaltstatus von Relais 2 wechseln (ein->aus, aus->ein)
Taste F3	Alarm-Test
Taste F4	<p>Programmnummer hoch zählen (die LEDs zeigen in den nächsten Sekunden die Programmnummer an)</p> <p>LED 1: Programm 1 (Nachlaufzeit von 60M)</p> <p>LED 2: Programm 2 (Nachlaufzeit von 120M)</p> <p>LED 3: Programm 3 (Nachlaufzeit von 180M)</p>

Tabelle 4

Die Bedeutung der vier LEDs an der C-Control I ist in Tabelle 5 aufgeführt.

LEDs an der C-Control I	Bedeutung
LED über F1	Status P0 (leuchtend: P0 eingeschaltet)
LED über F2	Relais 2 (leuchtend: eingeschaltet)
LED über F3	Schaltstatus T0 (leuchtend: Temperatur T0 erreicht)
LED über F4	Lebenssignal der Steuersoftware

Tabelle 5

Mit der Taste „Reset“ wird das C-Control I System angehalten und neu initialisiert. Erst die Betätigung der Taste „Start“ startet die Heizungssteuerung auf der C-Control I (siehe Handbuch der C-Control I).

Mögliche Erweiterungen

Folgende Erweiterungen sind möglich:

- Temperaturüberwachung des Heizungsraums auf Überhitzung für z.B. Feuersalarm
- Alarmmeldungs-Transport von Rauchmeldern in andere Räume
- automatische Zeiteinstellung der C-Control Station über Funkuhr

Anhang Programmcode C-Control Station (große C-Control)

```

' -----
' -- Main Control C-Control Station I Firmware
' -- Copyright (c) 2008 by Intelligent Programming
' -- Author: Stefan Schwope, 09\08
' -----

' -- variables -----

define b1          BYTE[11] ' -- read byte
define b2          BYTE[12] ' -- inverted 'b1'
define b3          BYTE[13] ' -- 1=error, 0=ok or return value
define loopCntr   BYTE[14] ' -- main loop counter
define iMenu      BYTE[15] ' -- selected menu item
define P0OnTime   BYTE[16] ' -- current time P0 is running
define P0OnTimeUser BYTE[17] ' -- time to switch on P0 (set by user)
define lastMinute BYTE[18] ' -- last minute in loop to detect minute changes
define interTime  BYTE[19] ' -- interval time
define interTimeUser BYTE[20] ' -- interval time (set by user)

define bT0        BIT[181]
define bT1        BIT[180]
define bT2        BIT[179]
define bT3        BIT[178]
define bP0        BIT[177]
define bP1        BIT[176]
define bP2        BIT[175]
define bP3        BIT[174]
define bM1        BIT[173]
define bM2        BIT[172]
define bM3        BIT[171]
define bInTime    BIT[170]

' -----
' ----- I/O PORTS -----
' -----
define ADC8        ad[8]
define SDA         port[9]
define SCL         port[10]
define LIGHT       port[16]
define EXTPORT     byteport[4]

' -----
' ----- SYSTEM MEMORY -----
' -----
define STATION_IO  byte[3]
define LED1       bit [17]
define LED2       bit [18]
define LED3       bit [19]
define LED4       bit [20]
define REL1       bit [21]
define REL2       bit [22]
define REL3       bit [23]
define REL4       bit [24]
define REL5       port[1]
define REL6       port[6]
define TEMP1IN    port[3]
define TEMP2IN    port[4]
define TEMP3IN    port[5]
define KEY        ad [8]

' -- initialize hardware -----
BAUD R9600

```

```

#INIT
SDA=on
SCL=on
EXTPORT=off
LIGHT=off
print"#ON_LCD#";
print"#INIT#";
print"#CLR#";
print"#L101#";"HeizungCtrl V1.0";
print"#L201#";"Stefan Schwope ";

GOSUB alloff

PAUSE 100 ' -- wait some time

' -- debug
'GOSUB M1close
'PAUSE 100
'GOSUB M2close
'PAUSE 100
'GOSUB M3close
'PAUSE 100
'GOSUB M1open
'PAUSE 100
'GOSUB M2open
'PAUSE 100
'GOSUB M3open
'PAUSE 100

P0OnTimeUser = 30
interTimeUser = 1

' -- config
iMenu = 0
GOSUB menu3

' -- main menu (main loop) -----
loopCntr = 0
bInTime = 0 ' -- not in time
P0OnTime = 0
lastMinute = MINUTE
#mainMenu

if lastMinute <> MINUTE then GOSUB nextMinute

' -- get temperature
GOSUB ist0
GOSUB ist1
GOSUB ist2
GOSUB ist3

GOSUB isPrg ' -- program set by C-Control inside heating room?

' -- RS232
if RXD then GOSUB readChar

' -- config system
GOSUB KEYDECODE
if b3 = 1 then GOSUB prg1
if b3 = 2 then GOSUB prg2
if b3 = 3 then GOSUB prg3

' -- debug -----
if b3 = 8 then bT0 = NOT bT0 ' -- switch bT0 off and on
' -----

```

```

if b3 = 9 then LIGHT = NOT LIGHT ' -- switch light off and on
if b3 > 9 AND b3 < 255 then GOSUB MENU3

' -- draw main status
print"#CLR#";
if loopCntr <= 5 then GOSUB menu1 else GOSUB menu2

' -- main logic
GOSUB logic

' -- alive LED
if loopCntr = 10 then GOSUB alive
loopCntr=loopCntr+1

pause 25 ' -- wait a second

GOTO mainMenu
END

' -- program 1 -----
#prg1
P0OnTimeUser = 90
print"#CLR#";
print"#L101#";
print"Prg1: 90M";
PAUSE 50

b1 = 75 ' -- reset prg on C-Control inside heating room
GOSUB writeChar
GOSUB testAlarm

RETURN

' -- program 2 -----
#prg2
P0OnTimeUser = 150
print"#CLR#";
print"#L101#";
print"Prg2: 150M";
PAUSE 50

b1 = 75 ' -- reset prg on C-Control inside heating room
GOSUB writeChar
GOSUB testAlarm

RETURN

' -- program 1 -----
#prg3
P0OnTimeUser = 210
print"#CLR#";
print"#L101#";
print"Prg3: 210M";
PAUSE 50

b1 = 75 ' -- reset prg on C-Control inside heating room
GOSUB writeChar
GOSUB testAlarm

RETURN

' -- next minute is over -----
#nextMinute
if (P0OnTime > 0) then P0OnTime = P0OnTime-1 else GOSUB P0off
if (interTime > 0) then interTime = interTime-1 else interTime = interTimeUser

```



```

if (bM1=OFF) AND (interTime = 0) then GOSUB P1Intervall
if (bM2=OFF) AND (interTime = 0) then GOSUB P2Intervall
if (bM3=OFF) AND (interTime = 0) then GOSUB P3Intervall
lastMinute = MINUTE
RETURN

' -- logic -----
#logic
GOSUB isInTime
if bInTime = OFF then RETURN

' -- do the regulation -----
if bT0=ON AND bP0=OFF AND (bM1=ON OR bM2=ON OR bM3=ON) then GOSUB P0on
if (bM1=OFF AND bM2=OFF AND bM3=OFF) then GOSUB P0off
if (bT0=ON AND bM1=ON) then GOSUB P1on
if (bT0=ON AND bM2=ON) then GOSUB P2on
if (bT0=ON AND bM3=ON) then GOSUB P3on
if (bT1=ON) then GOSUB M1close else GOSUB M1open
if (bT2=ON) then GOSUB M2close else GOSUB M2open
if (bT3=ON) then GOSUB M3close else GOSUB M3open

if bM1=ON AND bP0=OFF then GOSUB P1off
if bM2=ON AND bP0=OFF then GOSUB P2off
if bM3=ON AND bP0=OFF then GOSUB P3off
RETURN

' --
#P1Intervall
print"#CLR#";
print"#L101#";
print"Interval P1";
if bP1=ON then GOSUB P1off else GOSUB P1on
PAUSE 1
RETURN

' --
#P2Intervall
print"#CLR#";
print"#L101#";
print"Interval P2";
if bP2=ON then GOSUB P2off else GOSUB P2on
PAUSE 1
RETURN

' --
#P3Intervall
print"#CLR#";
print"#L101#";
print"Interval P3";
if bP3=ON then GOSUB P3off else GOSUB P3on
PAUSE 1
RETURN

' --
#isInTime
bInTime = ON ' -- default: in working time

' -- in time?
if (HOUR >= 2) AND ((HOUR < 3) OR (HOUR = 3 AND
MINUTE <= 30)) then RETURN
if (HOUR >= 7) AND ((HOUR < 10) OR (HOUR = 10 AND
MINUTE <= 30)) then RETURN
if ((HOUR > 13) OR (HOUR = 13 AND MINUTE >= 0)) AND ((HOUR < 14) OR (HOUR = 14 AND
MINUTE <= 30)) then RETURN
if ((HOUR > 16) OR (HOUR = 16 AND MINUTE >= 30)) AND (HOUR < 18)
then RETURN

```

```

if (HOUR >= 20)
MINUTE <= 30)) then RETURN
bInTime = OFF ' -- out of working time

' -- all close
GOSUB P0off
GOSUB P1off
GOSUB P2off
GOSUB P3off
GOSUB M1open
GOSUB M2open
GOSUB M3open
RETURN

' --
#allOpen
GOSUB P0on
GOSUB P1on
GOSUB P2on
GOSUB P3on
GOSUB M1open
GOSUB M2open
GOSUB M3open
RETURN

' -- show menu 1 -----
#menu1
print"#L101#";
print"T0|1|2|3|M1|2|3";
print"#L201#";
print" ";
IF bT0=ON THEN print"*"; ELSE print" ";
print"|";
IF bT1=ON THEN print"*"; ELSE print" ";
print"|";
IF bT2=ON THEN print"*"; ELSE print" ";
print"|";
IF bT3=ON THEN print"*"; ELSE print" ";
print"| ";
IF bM1=ON THEN print"v"; ELSE print"^";
print"|";
IF bM2=ON THEN print"v"; ELSE print"^";
print"|";
IF bM3=ON THEN print"v"; ELSE print"^";
RETURN

' -- show menu 2 -----
#menu2
print"#L101#";
print"P0|1|2|3|IT";
print"#L201#";
print" ";
IF bP0=ON THEN print"*"; ELSE print" ";
print"|";
IF bP1=ON THEN print"*"; ELSE print" ";
print"|";
IF bP2=ON THEN print"*"; ELSE print" ";
print"|";
IF bP3=ON THEN print"*"; ELSE print" ";
print"|";
IF bInTime=ON THEN print "***"; ELSE print" ";

' -- show mischer
IF bM1=OFF THEN LED1 = OFF ELSE LED1 = ON
IF bM2=OFF THEN LED2 = OFF ELSE LED2 = ON
IF bM3=OFF THEN LED3 = OFF ELSE LED3 = ON

```

```

RETURN

' -- show menu 3 -----
#menu3
print"#CLR#";
print"#L101#";"Einstellungen";

' -- ENTER
if b3 = 11 AND iMenu = 4 then RETURN
if b3 = 11 then ON iMenu GOSUB timeHour, timeMinute, P0Time, IntervalTime
if iMenu > 4 then iMenu = 4

' -- F1
if b3 = 12 then iMenu = iMenu-1
if iMenu > 4 then iMenu = 4

' -- F2
if b3 = 13 then iMenu = iMenu+1
if iMenu > 4 then iMenu = 0

' -- ENTER
' -- draw menu
print"#L201#";
if iMenu = 0 then print"Zeit Stunde:";HOUR;
if iMenu = 1 then print"Zeit Minute:";MINUTE;
if iMenu = 2 then print"P0 Nachlauf:";P0OnTimeUser;"M";
if iMenu = 3 then print"Px Interval:";interTimeUser;"M";
if iMenu = 4 then print"Menu verlassen";

' -- wait for key
#keyAgain
PAUSE 10
GOSUB KEYDECODE
if b3 = 255 then GOTO keyAgain
GOTO menu3

' --
#IntervalTime
interTimeUser = interTimeUser+1
if interTimeUser > 10 then interTimeUser = 0
RETURN

' --
#P0Time
P0OnTimeUser = P0OnTimeUser+10
if P0OnTimeUser > 180 then P0OnTimeUser = 0
RETURN

' --
#timeHour
HOUR = HOUR+1
if HOUR > 23 then HOUR = 0
RETURN

' --
#timeMinute
MINUTE = MINUTE+1
if MINUTE>59 then MINUTE = 0
RETURN

' -- alive LED -----
#alive
loopCntr=0
IF LED4 THEN LED4 = OFF ELSE LED4 = ON
RETURN

```

```

' -- M1 open -----
#M1open
REL5=OFF
bM1=ON
RETURN

' -- M1 close -----
#M1close
REL5=ON
bM1=OFF
RETURN

' -- M2 open -----
#M2open
REL6=OFF
bM2=ON
RETURN

' -- M2 close -----
#M2close
REL6=ON
bM2=OFF
RETURN

' -- M3 open -----
#M3open
REL4=OFF:EXTPORT=STATION_IO
bM3=ON
RETURN

' -- M3 close -----
#M3close
REL4=ON:EXTPORT=STATION_IO
bM3=OFF
RETURN

' -- P0 on -----
#P0on
P0OnTime = P0OnTimeUser
' -- debug
'bP0=ON
'RETURN
b1 = 65 ' -- P0 on
GOSUB writeChar
GOSUB testAlarm
bP0=ON
'LED1=ON

print"#CLR#";
print"#L101#";"P0 1min. Vorlauf";

' -- wait one minute and do nothing
PAUSE 1500
RETURN

' -- P0 off -----
#P0off
' -- debug
'bP0=OFF
'RETURN
b1 = 66 ' -- P0 off
GOSUB writeChar
GOSUB testAlarm
bP0=OFF
'LED1=OFF

```

```

P0OnTime = 0
RETURN

' -- P1 on -----
#P1on
REL3=ON:EXTPORT=STATION_IO
bP1=ON
'LED2=ON
RETURN

' -- P1 off -----
#P1off
REL3=OFF:EXTPORT=STATION_IO
bP1=OFF
'LED2=OFF
RETURN

' -- P2 on -----
#P2on
REL2=ON:EXTPORT=STATION_IO
bP2=ON
'LED3=ON
RETURN

' -- P2 off -----
#P2off
REL2=OFF:EXTPORT=STATION_IO
bP2=OFF
'LED3=OFF
RETURN

' -- P3 on -----
#P3on
REL1=ON:EXTPORT=STATION_IO
bP3=ON
'LED4=ON
RETURN

' -- P3 off -----
#P3off
REL1=OFF:EXTPORT=STATION_IO
bP3=OFF
'LED4=OFF
RETURN

' -- is prg -----
#isPrg
b1 = 74 ' -- get prg
GOSUB writeChar
GOSUB testAlarm
IF NOT RXD THEN RETURN
GOSUB readChar
GOSUB testAlarm
if b1 = 0 then RETURN
if b1 = 1 then P0OnTimeUser = 90
if b1 = 2 then P0OnTimeUser = 150
if b1 = 3 then P0OnTimeUser = 210
RETURN

' -- is temp0 -----
#isT0
' -- debug
'RETURN
bT0 = ON ' -- default
b1 = 73 ' -- get temp0
GOSUB writeChar

```



```

GOSUB testAlarm
IF NOT RXD THEN RETURN
GOSUB readChar
GOSUB testAlarm
if b1 = 0 then bT0 = OFF else bT0 = ON
RETURN

' -- is temp1 -----
#isT1
DEACT TEMP1IN
if TEMP1IN=ON then bT1 = OFF else bT1 = ON
RETURN

' -- is temp2 -----
#isT2
DEACT TEMP2IN
if TEMP2IN=ON then bT2 = OFF else bT2 = ON

' -- only for now because circle 2 not connected yet!
bT2 = ON
RETURN

' -- is temp3 -----
#isT3
DEACT TEMP3IN
if TEMP3IN=ON then bT3 = OFF else bT3 = ON
RETURN

' -- alarm -----
#alarm
print"#CLR#";
print"#L101#";
print"Alarm!!!";
GOSUB M1open
GOSUB M2open
GOSUB M3open
GOSUB P1off
GOSUB P2off
GOSUB P3off

b1 = 71          ' -- alarm on station in heating room
GOSUB writeChar
b1 = 1          ' -- default: temp reached (no overheat possible)
BEEP 500, 350, 10 ' -- A for 10*20ms, pause for 3*20ms
print"#ON_LCD#";
RETURN

' -- version -----
#version
' -- send version 1.0
b1 = 1
GOSUB writeChar
GOSUB testAlarm
b1 = 0
GOSUB writeChar
GOSUB testAlarm
RETURN

' -- write character -----
#writeChar
b3 = 1
print"#OFF#";
b2 = NOT b1
PUT b1
PAUSE 1
PUT b2

```

```

PAUSE 50          ' -- wait one second
IF NOT RXD THEN RETURN ' -- no answer -> communication error
GET b2
if b1 = b2 then b3 = 0
print"#ON_LCD#";
RETURN

' -- read character from RS232 -----
#readChar
print"#OFF#";
b3 = 1
GET b1            ' -- get first commando byte
PAUSE 50          ' -- wait 0,5 sec. for next byte
if NOT RXD then RETURN ' -- no second byte -> communication error
GET b2            ' -- get second command byte (inverted by sender)
if NOT b2 <> b1 then RETURN ' -- wrong character
b3 = 0            ' -- character read
print"#ON_LCD#";
RETURN

' -- test alarm -----
#testAlarm
if b3 = 1 then GOSUB alarm
print"#ON_LCD#";
RETURN

' -- switch all off -----
#alloff
GOSUB M1open
GOSUB M2open
GOSUB M3open
GOSUB P0off
GOSUB P1off
GOSUB P2off
GOSUB P3off
PAUSE 1
RETURN

' -- get keycode -----
#KEYDECODE
b3=255:if ADC8<10 then return
b3=0:if abs(17-ADC8)<5 then return
b3=1:if abs(33-ADC8)<5 then return
b3=2:if abs(50-ADC8)<5 then return
b3=3:if abs(66-ADC8)<5 then return
b3=4:if abs(82-ADC8)<5 then return
b3=5:if abs(98-ADC8)<5 then return
b3=6:if abs(114-ADC8)<5 then return
b3=7:if abs(130-ADC8)<5 then return
b3=8:if abs(147-ADC8)<5 then return
b3=9:if abs(164-ADC8)<5 then return
b3=10:if abs(181-ADC8)<5 then return'CLR
b3=11:if abs(199-ADC8)<5 then return'ENTER
b3=12:if abs(218-ADC8)<5 then return'F1
b3=13:if abs(236-ADC8)<5 then return'F2
b3=255:return

```

Anhang Programmcode C-Control I (kleine C-Control)

```

' -----
' -- Heating Room C-Control Station I Firmware
' -- Copyright (c) 2008 by Intelligent Programming
' -- Author: Stefan Schwope, 09\08
' -----

' -- variables -----
DEFINE b1    BYTE
DEFINE b2    BYTE
DEFINE b3    BYTE
DEFINE cntr  BYTE
DEFINE prog  BYTE ' -- 1, 2 ,3: change to prog1, 2, 3

DEFINE P1    PORT[ 1]
DEFINE P2    PORT[ 2]
DEFINE P3    PORT[ 3]
DEFINE P4    PORT[ 4]
DEFINE P5    PORT[ 5]
DEFINE P6    PORT[ 6]
DEFINE K1    PORT[ 7]
DEFINE K2    PORT[ 8]
DEFINE F1    PORT[ 9] ' -- tog K1
DEFINE F2    PORT[10] ' -- tog K2
DEFINE F3    PORT[11] ' -- alarm test
DEFINE F4    PORT[12] ' -- toggle program
DEFINE LED1  PORT[13] ' -- K1
DEFINE LED2  PORT[14] ' -- K2
DEFINE LED3  PORT[15] ' -- switch P1
DEFINE LED4  PORT[16] ' -- alive
DEFINE A1    AD  [ 1]
DEFINE A2    AD  [ 2]
DEFINE A3    AD  [ 3]
DEFINE A4    AD  [ 4]
DEFINE T1    AD  [ 5]
DEFINE T2    AD  [ 6]
DEFINE U1    AD  [ 7]
DEFINE U2    AD  [ 8]

' -- initialize hardware -----
BAUD R9600

' -- main menu (main loop) -----
prog = 0 ' -- no program change
b3 = 0
#mainMenu

' -- RS232
if RXD then GOSUB readChar

' -- keyboard
if NOT F1 then GOSUB k1tog
if NOT F2 then GOSUB k2tog
if NOT F3 then GOSUB alarm
if NOT F4 then GOSUB progInc

' -- alive LED
if b3 = 200 then GOSUB alive
b3 = b3+1

GOTO mainMenu
END

```

```

' -- program increase -----
#progInc
prog = prog+1
if prog > 3 then prog = 1 ' -- limit prog
LED1 = OFF
LED2 = OFF
LED3 = OFF
LED4 = OFF
if prog = 1 then LED1 = ON
if prog = 2 then LED2 = ON
if prog = 3 then LED3 = ON
PAUSE 100 ' -- 2sec

' -- restore relay state (rest is done by 'alive')
IF K1 THEN LED1 = ON ELSE LED1 = OFF
IF K2 THEN LED2 = ON ELSE LED2 = OFF
RETURN

' -- alive LED -----
#alive

' -- filter 50Hz
cntr = 0
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if cntr > 3 then LED3 = OFF else LED3 = ON

IF LED4 THEN LED4 = OFF ELSE LED4 = ON
b3 = 0 ' -- timer reset
RETURN

' -- read character from RS232 -----
#readChar
GET b1 ' -- get first commando byte
PAUSE 25 ' -- wait 0,5 sec. for next byte
if NOT RXD then RETURN ' -- no second byte -> communication error
GET b2 ' -- get second command byte (inverted by sender)
if NOT b2 <> b1 then RETURN ' -- wrong character
PUT b1 ' -- send back command
b1=b1-65 ' -- 65=A

' -- commands
' k1on = A
' k1off = B
' k2on = C
' k2off = D
' sendtemperature = E
' sendtime = F
' alarm = G
' version = H
' getP1 = I
' getProg = J
' resetProg = K
ON b1 GOSUB k1on, k1off, k2on, k2off, sendtemperature, sendtime, alarm, version, getP1,
getProg, resetProg
RETURN

' -- get program -----

```

```
#getProg
b1 = prog
GOSUB writeChar
RETURN

' -- reset program -----
#resetProg
prog = 0
RETURN

' -- k1 tog -----
#k1tog
IF K1 THEN K1 = OFF ELSE K1 = ON
IF K1 THEN LED1 = ON ELSE LED1 = OFF
PAUSE 5
WAIT F1
RETURN

' -- k1 on -----
#k1on
K1 = ON
LED1 = ON
RETURN

' -- k1 off -----
#k1off
K1 = OFF
LED1 = OFF
RETURN

' -- k2 tog -----
#k2tog
IF K2 THEN K2 = OFF ELSE K2 = ON
IF K2 THEN LED2 = ON ELSE LED2 = OFF
PAUSE 5
WAIT F2
RETURN

' -- k2 on -----
#k2on
K2 = ON
LED2 = ON
RETURN

' -- k2 off -----
#k2off
K2 = OFF
LED2 = OFF
RETURN

' -- send temperature -----
#sendtemperature
b1 = T1
GOSUB writeChar
b1 = T2
GOSUB writeChar
RETURN

' -- send time -----
#sendtime
WAIT SECOND <= 55 ' -- no minute change while reading time
' -- send time
b1 = SECOND
GOSUB writeChar
b1 = MINUTE
GOSUB writeChar
```



```
b1 = HOUR
GOSUB writeChar
b1 = DOW
GOSUB writeChar
b1 = DAY
GOSUB writeChar
b1 = MONTH
GOSUB writeChar
b1 = YEAR
GOSUB writeChar
RETURN

' -- alarm -----
#alarm
BEEP 500, 250, 30 ' -- A for 10*20ms, pause for 3*20ms
RETURN

' -- version -----
#version
' -- send version 1.0
b1 = 1
GOSUB writeChar
b1 = 0
GOSUB writeChar
RETURN

' -- P1 -----
#getP1
' -- filter 50Hz
cntr = 0
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if P2 then cntr = cntr+1
PAUSE 1
if cntr > 3 then b1 = 0 else b1 = 1
GOSUB writeChar
RETURN

' -- write character -----
#writeChar
b2 = NOT b1
PUT b1
PUT b2
RETURN
```