

# Anhang

Anhang A: Gebrauchsanleitung .....	2
Netzteil .....	2
Aufstellort .....	2
Software Installaltion .....	2
Software Konfiguration .....	4
Anhang B: Herstellungsanleitung.....	5
Layout: Platine / Gehäuse .....	5
Stückliste & Kosten .....	7
Platinen Schema .....	10
Anhang C: Flussdiagramm Mikrocontroller .....	12
Anhang D: Quellcode Mikrocontroller .....	17
Anhang E: Flussdiagramm SyncTime (Computer-Software).....	34
Anhang F: Quellcode SyncTime .....	35
Anhang G: Arbeitsjournal.....	59
Anhang H: Selbsteinschätzung.....	60
Anhang I: Beurteilung durch Lehrer .....	61

## Anhang A: Gebrauchsanleitung

### Netzteil

Als Netzteil eignen sich alle Netzteile die über eine gleichgerichtete Ausgangsspannung von 5-9V bei 200mA verfügen. Der durchschnittlich fließende Strom liegt bei 170mA.

### Aufstellort

Die Uhr sollte so aufgestellt werden, dass das Display einen Winkel von 90° zur Richtung Mainflingen im Deutschen Bundesstaat Hessen (Standort des Zeitzeichensenders) aufweist.

Unter folgenden Bedingungen kann der Empfang gestört oder ganz verhindert werden:

- Ø Aufstellort in einem Haus mit Blechfassade oder Armierungseisen oder in einem Metallkasten (Prinzip des Faraday'schen Käfig)
- Ø Aufstellort in der Nähe von Störquellen wie Drucker, Fernseher, Stromsparlampen, o.Ä.
- Ø Aufstellort in der Nähe einer Strasse (Störfelder der Zündspulen der Autos)

Zur Hilfe kann ein tragbares Radiogerät genommen werden: Mit dem Empfang auf Langwelle eingestellt kann nach einer Stelle mit gutem Radioempfang gesucht werden. An dieser Stelle sollte auch die Funkuhr funktionieren, sofern keine Störquelle in der Nähe ist.

### Software Installation

#### Systemvoraussetzungen:

- Windows XP / 2000
- mind. .Net Framework 1.0 mit SP2

#### Benötigte Software

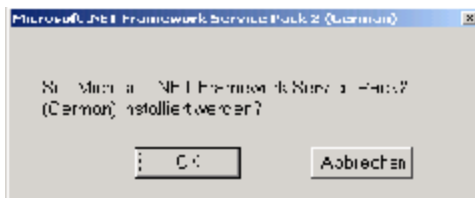
Für die Installation werden folgende Applikationen benötigt:

- .Net Framework Redistributable 1.0
  - Auf der CD:  
`Software\Framework\dotnetfx.exe`
  - Download:  
<http://download.microsoft.com/download/NETFrameworkRedistributable/Install/1.0/W98NT42KMeXP/DE/dotnetredist.exe>
- .Net Framework Service Pack 2
  - Auf der CD:  
`Software\Framework SP2\SP2_Q321898_De.exe`

- Download:
  - [http://download.microsoft.com/download/netframeworkredistributable/patch/2/w98nt42kmexp/de/NDP10\\_SP\\_Q321898\\_De.exe](http://download.microsoft.com/download/netframeworkredistributable/patch/2/w98nt42kmexp/de/NDP10_SP_Q321898_De.exe)
- SyncTime
  - Auf der CD:
    - \\Software\\SyncTime\\Setup.exe
  - Download:
    - <http://zeit.havalan.ch/synctime/SyncTime.rar>

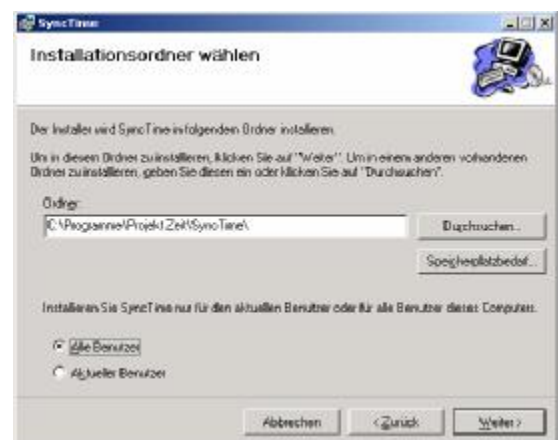
Als erstes werden das .Net Framework und das neuste Service Pack installiert.

Nach dem ausführen der Datei *dotnetfx.exe* startet der Assistent, der in einfachen Schritten den Benutzer durch die Installation führt.



Als nächstes kann das Service Pack 2 (*NDP10\_SP\_Q321898\_De.exe*) installiert werden. Nachdem man die Installation bestätigt und die EULA akzeptiert hat erfolgt die Installation.

Als letztes kann nun das eigentliche Programm installiert werden. Dazu startet man die Installations-Routine mit der *Setup.exe*. Nachdem die Benutzerdefinierten Einstellungen vorgenommen wurden startet die Installation. Nach einer erfolgreichen Installation erscheint im Startmenü eine neue Programmgruppe mit dem Namen SyncTime. Darin befindet sich eine Verknüpfung zur Anwendung.



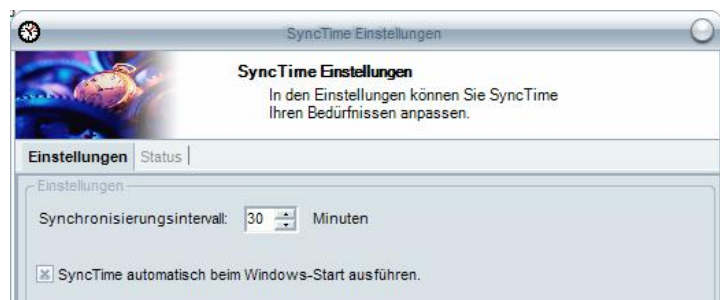
## Software Konfiguration

Nach dem Start von SyncTime erscheint in der Notifiliste ein neues Icon mit einer Uhr.



Durch einen Rechtsklick mit der Maus auf das Icon erscheint ein Context-Menü welches einem ermöglicht die Synchronisation zu Stoppen, das Programme zu beenden oder Einstellungen vorzunehmen.

In den Einstellungen kann der Benutzer selber das Synchronisations-Intervall einstellen. Zudem hat er die Möglichkeit SyncTime automatische beim Windows-Start zu starten.



Im zweiten Register, „Status“, hat der Benutzer die Möglichkeit, den Zustand von SyncTime zu betrachten. In der Liste werden alle Synchronisations-

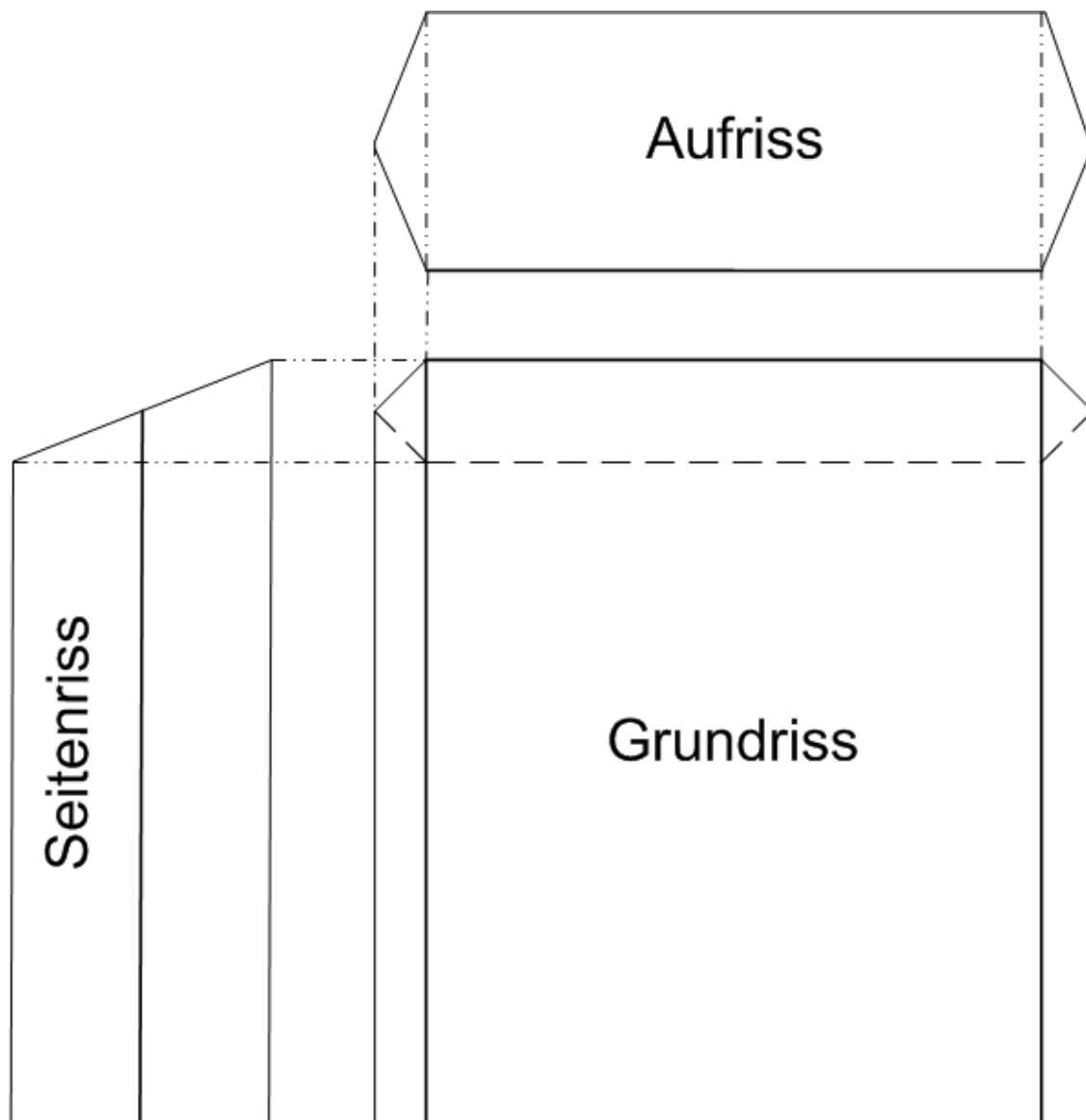
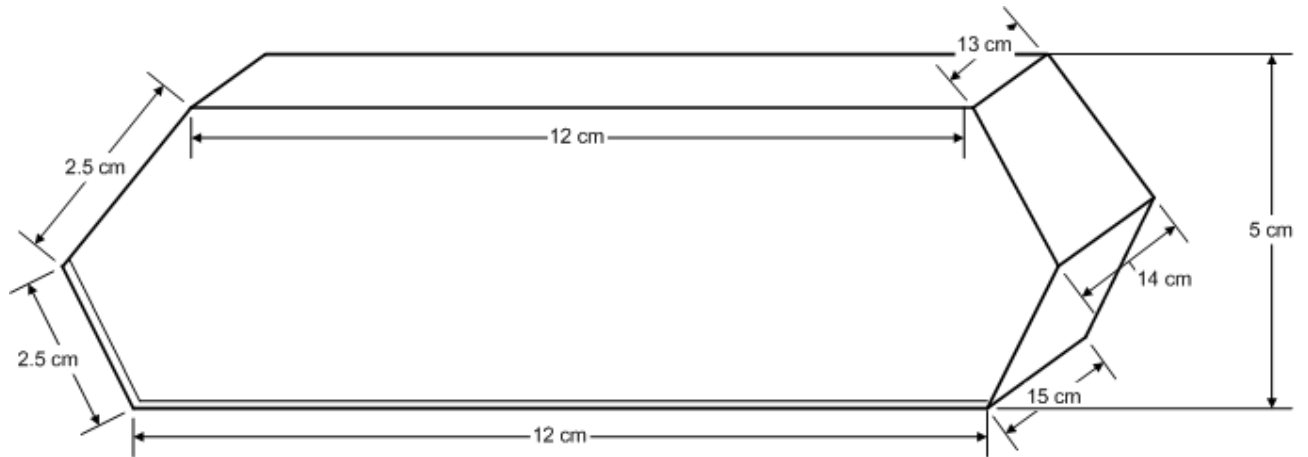
Vorgänge und fehlgeschlagene Aktualisierungen protokolliert.

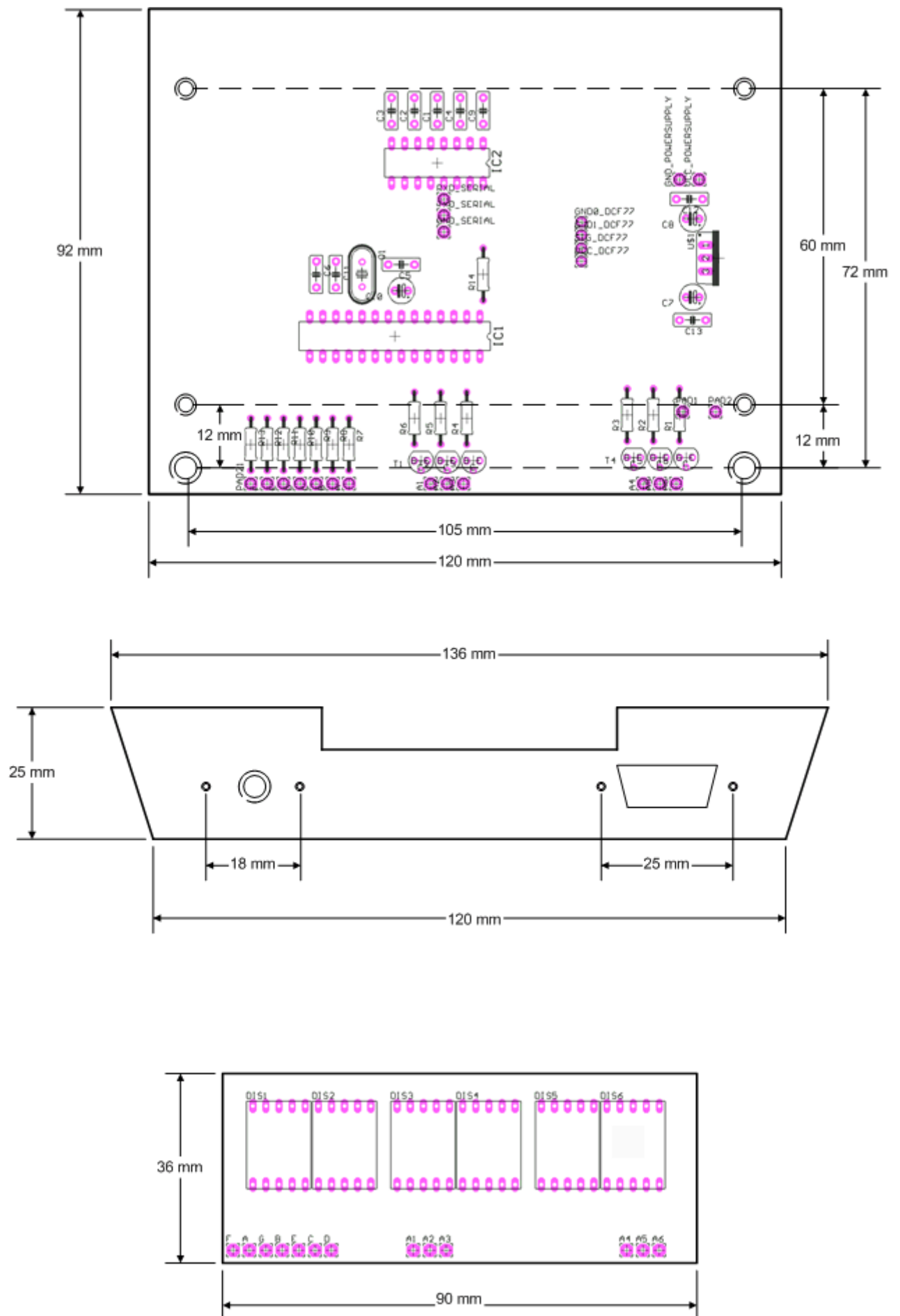


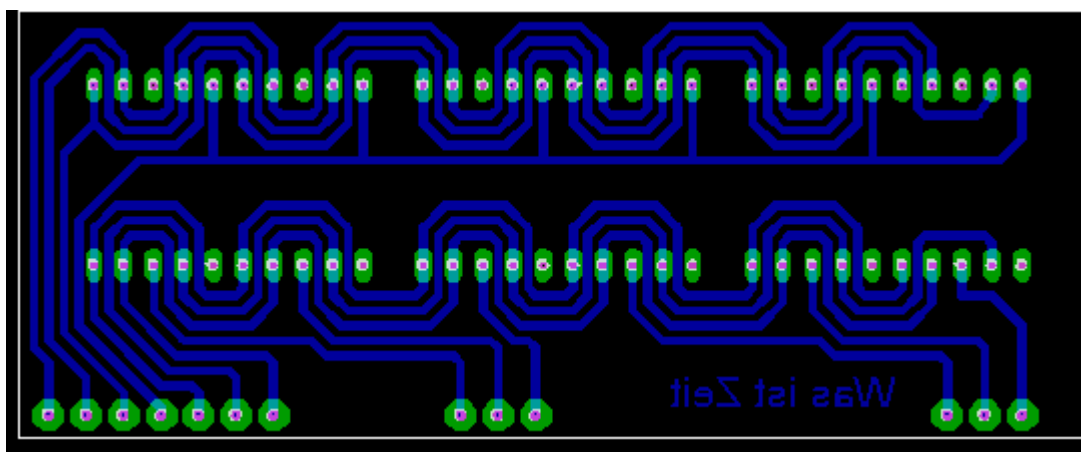
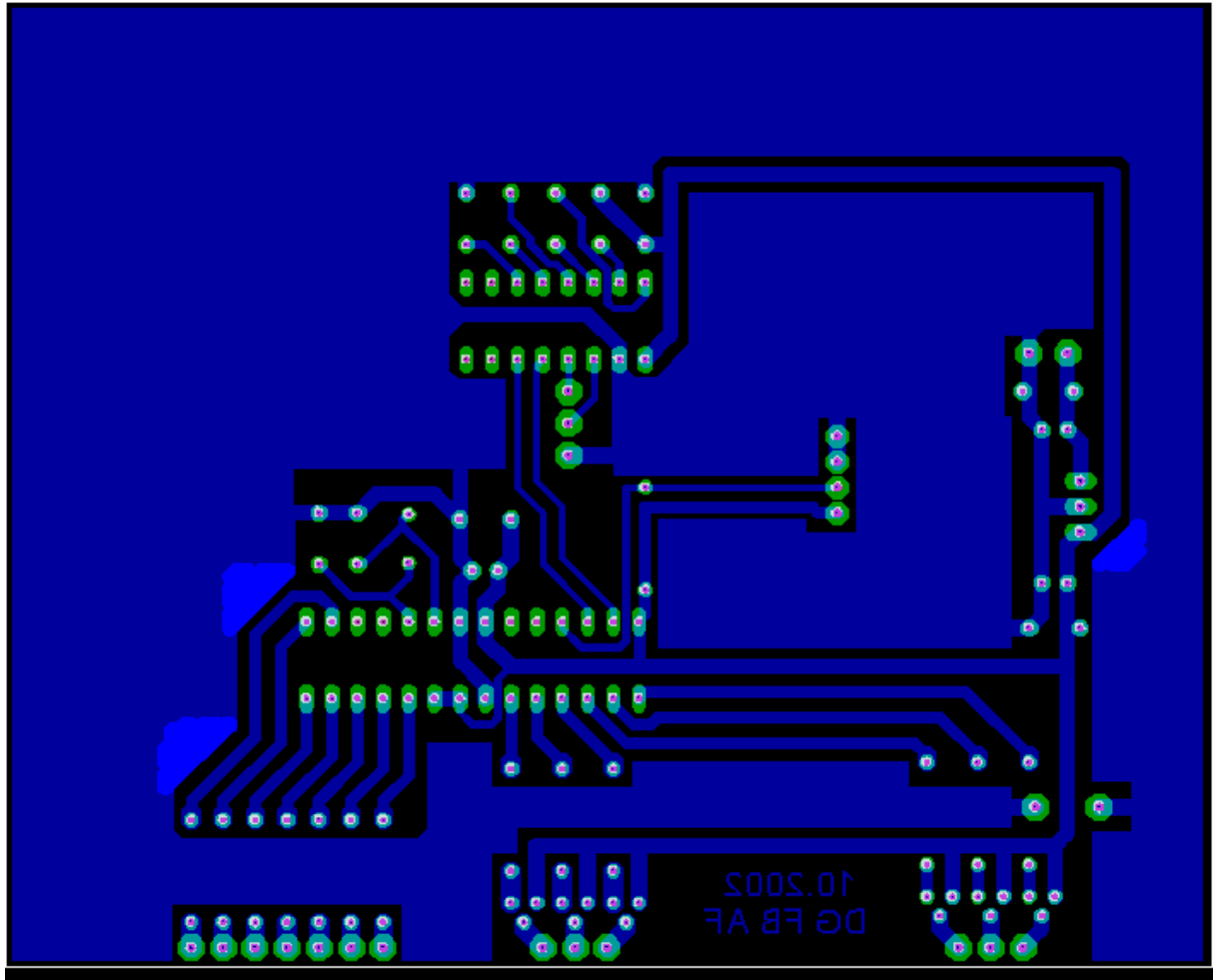
In Windows 2000 und XP werden fehlgeschlagene Synchronisationen zusätzlich in der Ereignisanzeige angezeigt.

## Anhang B: Herstellungsanleitung

### Layout: Platine / Gehäuse







## Stückliste & Kosten

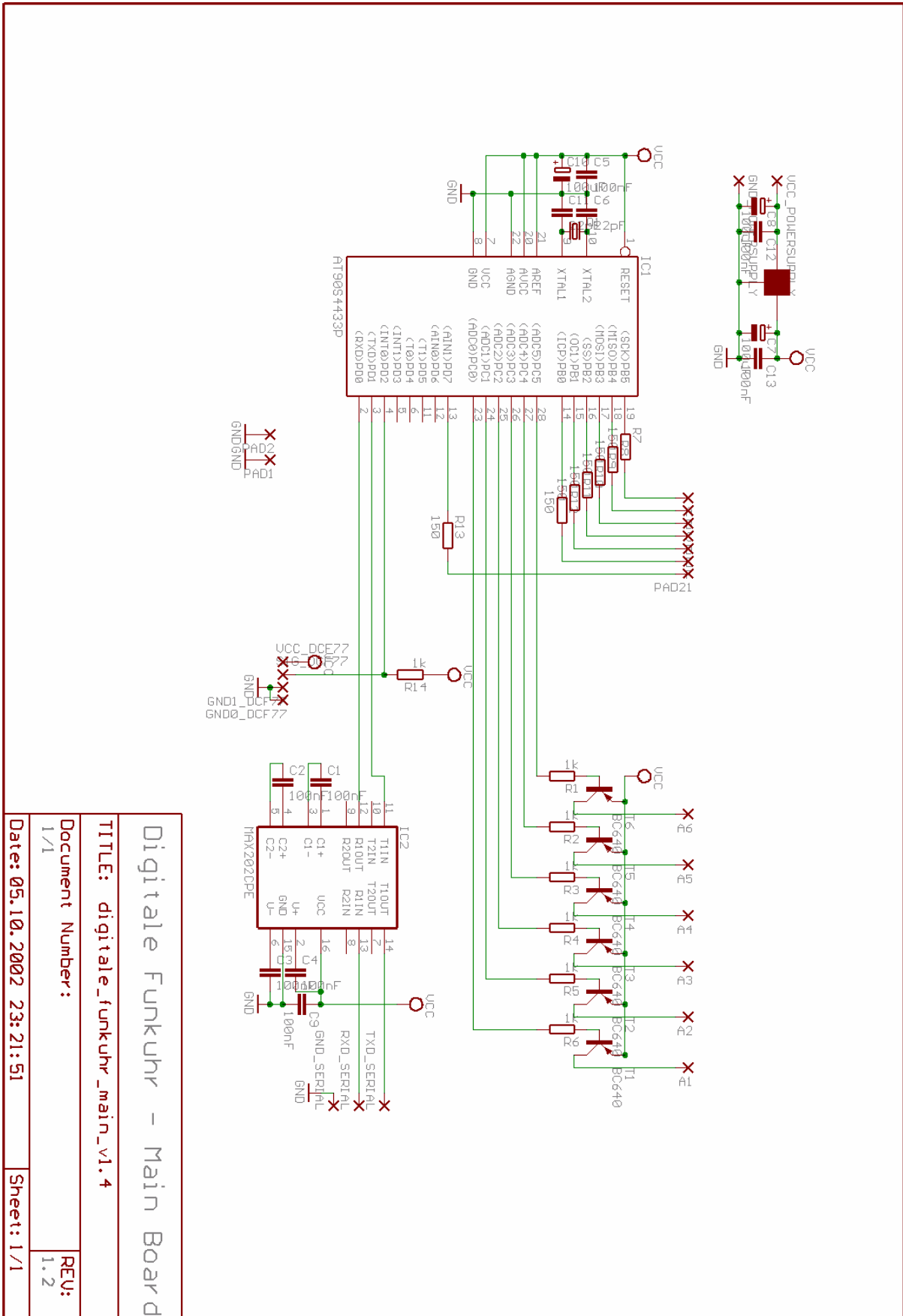
Das benötigte Material wurde bei der Firma Distrelec eingekauft  
(<http://www.distrelec.ch>)

Anzahl	Beschreibung	Kosten einzel / total	Layout – Nr.	Bestell-Nr
1	Kunststoffgehäuse	8.50 / 8.50		300223
1	Spannungsregler	2.30 / 2.30		640721
6	NF-Transistoren / BC 640	0.50 / 3.00		610381
1	AVR 8-Bit RISC Mikrocon..	24.00 / 24.00		645083
1	Diverse ICs / MAX 202CPE	5.80 / 5.80		647220
6	LED-Anzeigen, Ziffernanzeige	2.60 / 15.60		660410
1	Funkuhr- Komponenten für.. eitzeichensender / EM2-S-DCF	25.80 / 25.80		680260
1	Funkuhr- Komponenten für.. eitzeichensender / ANT	10.50 / 10.50		680261
1	Quarze, Low Profile HC4	2.60 / 2.60		680349
1	Netzgerät unstabilisier	12.10 / 12.10		920266
4	Kondensator 100nF	weitere Kosten	c1-c4	
4	100nF		c5,c9,c12,c13	
2	22pF		c11,c6	
3	100 mikroF		c7,c8,c10	
	Draht und Lötzinn	weitere Kosten		
1	Widerstand	weitere Kosten	r14	



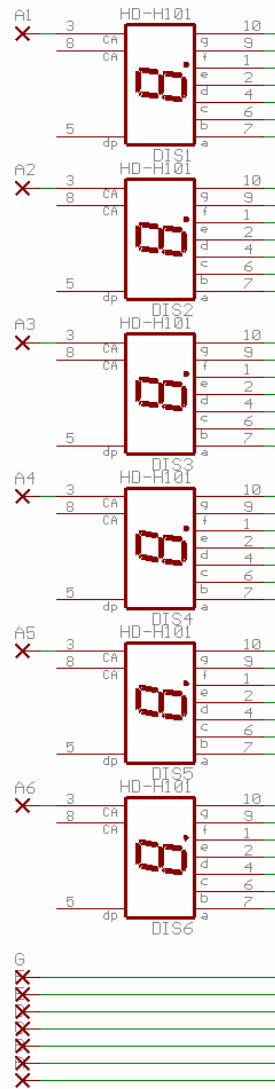
	10 KOhm			
6	Widerstände 150 Ohm	weitere Kosten	r7-13	
6	Widerstände 1 kOhm	weitere Kosten	r1-r6	
<b>Total</b>	<b>(inkl. weitere Kosten)</b>	<b>~140 SFr.</b>		

# Platinen Schema

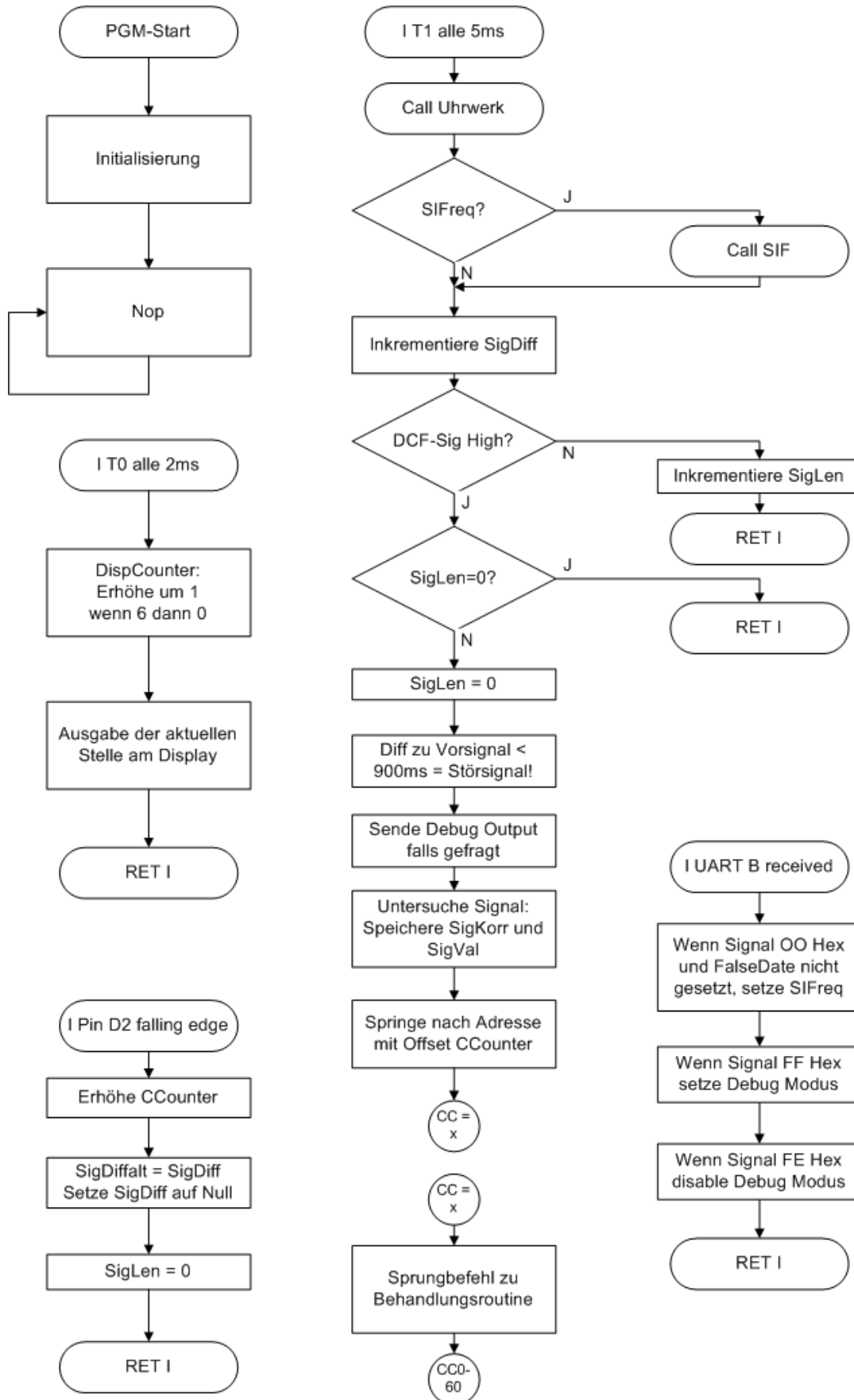


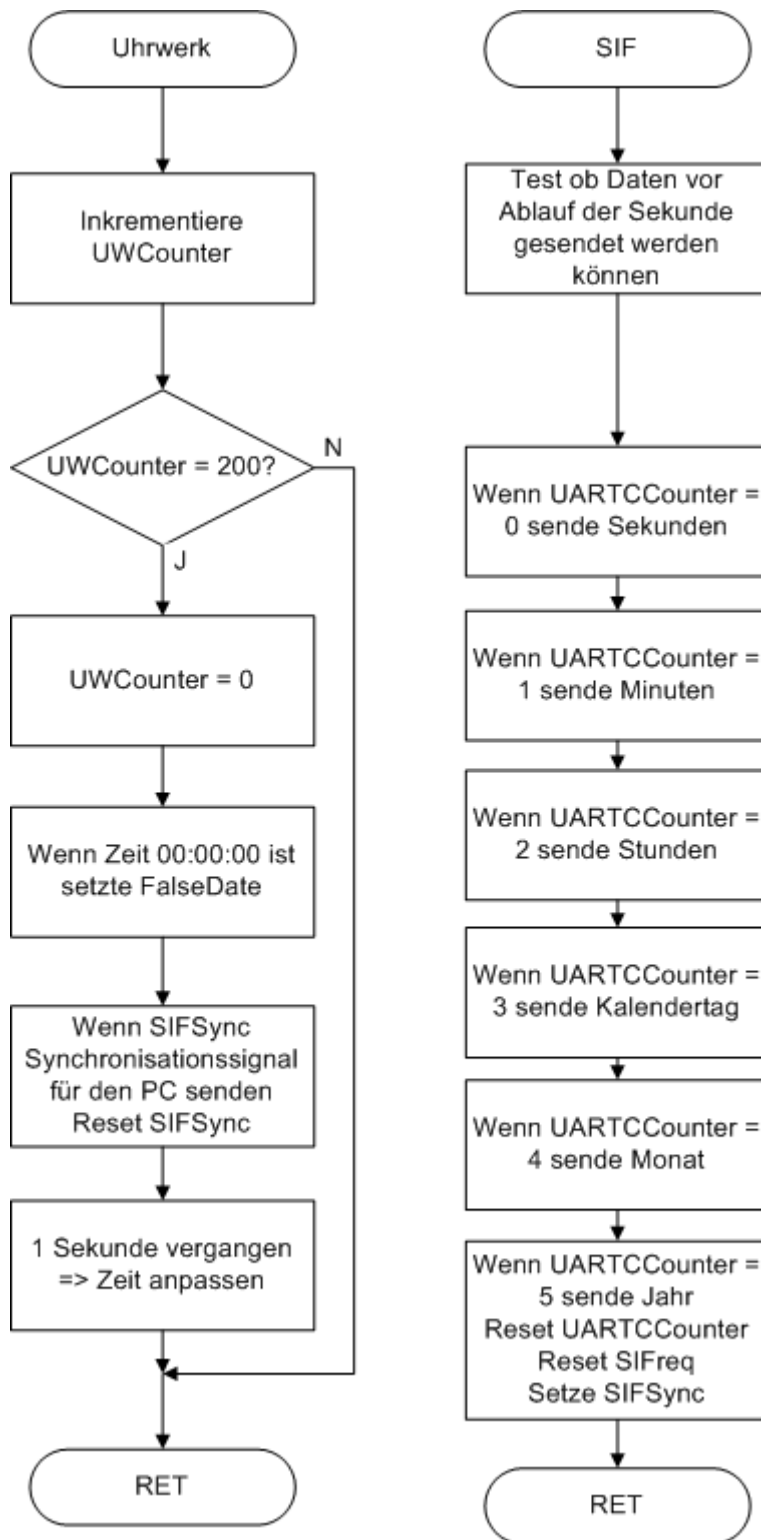
Digitale Funkuhr - Main Board	
TITLE: digitale_funkuhr_main_v1.4	
Document Number:	REVU:
1/1	1.2
Date: 05.10.2002 23:21:51	Sheet: 1/1

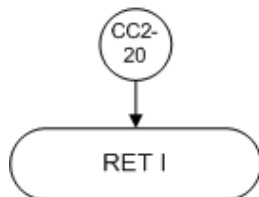
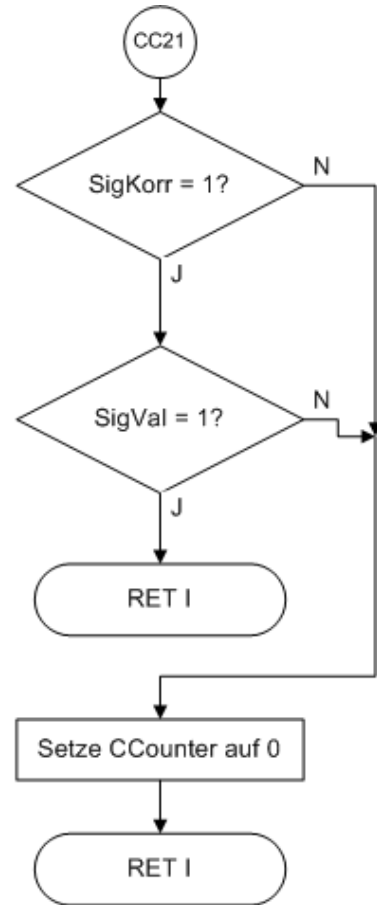
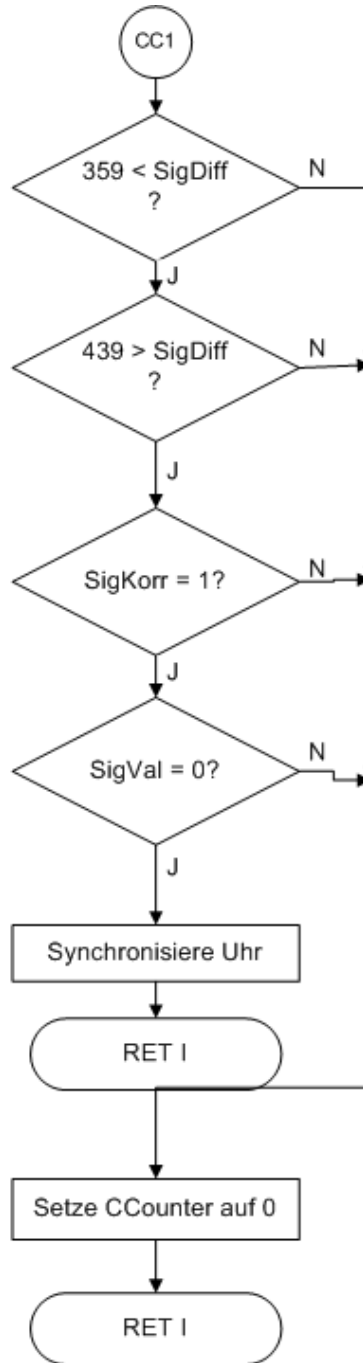
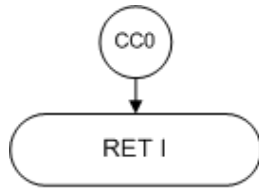
Digitale Funkuhr - Display	
TITLE: digitale_funkuhr_disp_v1.4	
Document Number:	REU: 1.2
Date: 12.10.2002 00:14:18	Sheet: 1/1

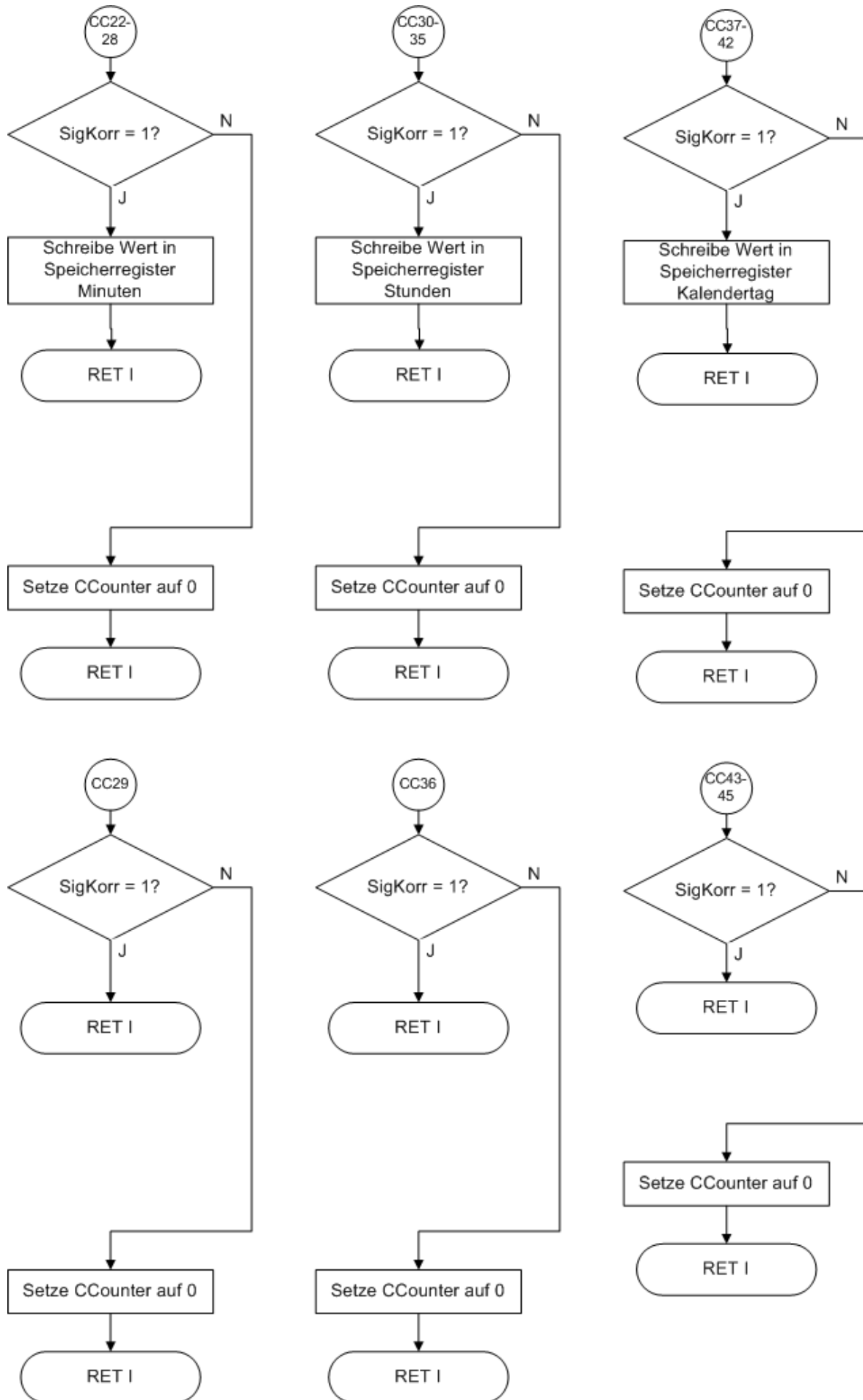


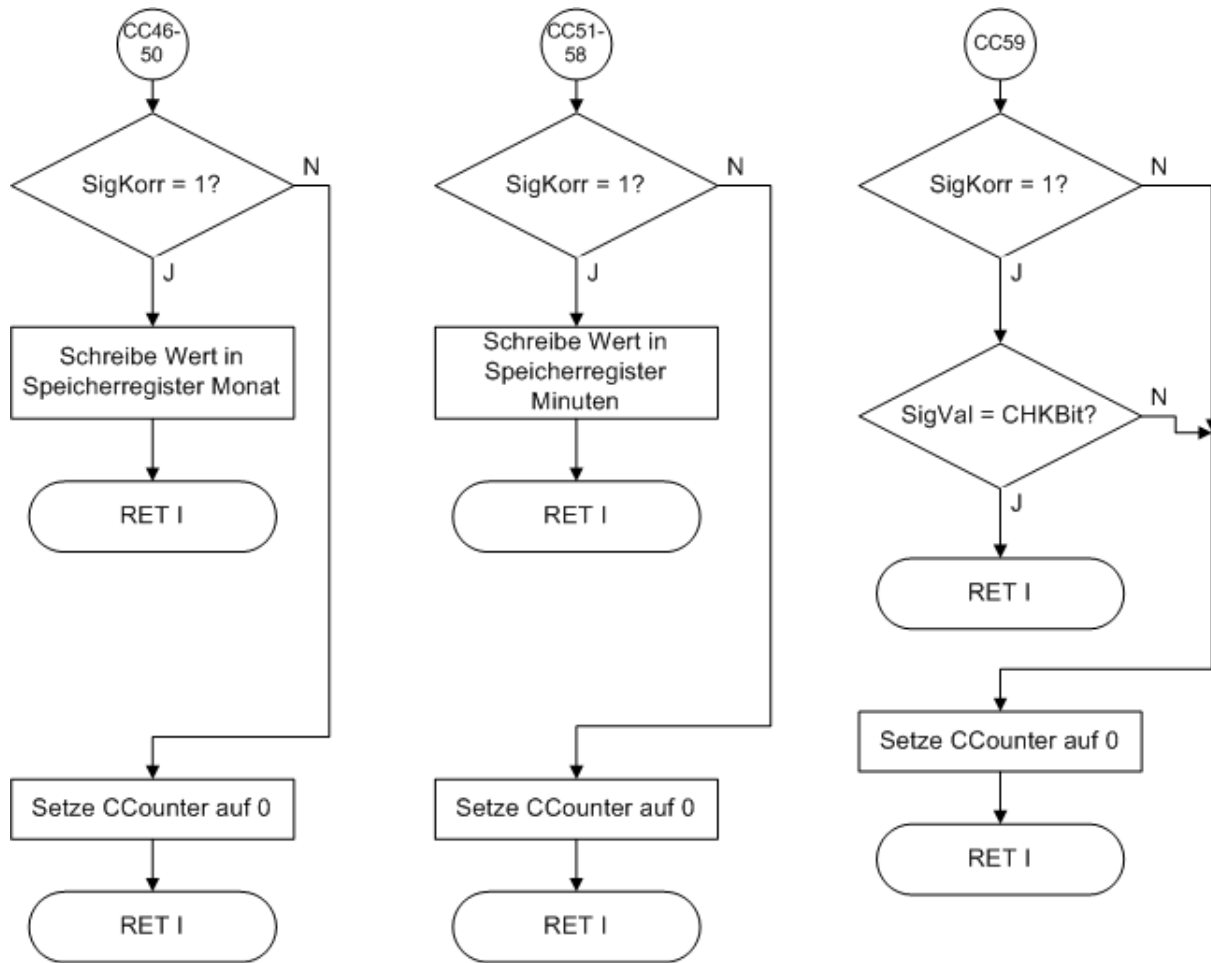
## Anhang C: Flussdiagramm Mikrokontroller













## Anhang D: Quellcode Mikrocontroller

```
; *****  
; DIGITAL RADIO CLOCK, software for AT90S4433, oktober 02 by Adrian R. Fischer  
; *****  
  
; include definition file  
.include "4433def.inc"  
  
; -----  
; definitions of the register  
; -----  
  
; binary coded register  
.def SigLen = r2  
.def SigDiffalt_L = r3  
.def SigDiffalt_H = r4  
.def SigDiff_L = r5  
.def SigDiff_H = r6  
.def SigDiffaltalt_L = r7  
.def SigDiffaltalt_H = r8  
; bcd coded register  
.def AusgSek = r9  
.def AusgMin = r10  
.def AusgStd = r11  
.def AusgDay = r12  
.def AusgMon = r13  
.def AusgYea = r14  
  
; binary coded register  
.def CCounter = r16  
.def UARTCounter = r17  
.def DispCounter = r18  
.def temp = r19  
.def UWCounter = r21  
  
.def Minuten = r22  
.def Stunden = r23  
.def KalTag = r24  
.def Monat = r25  
.def Jahr = r26  
  
.def flagreg = r27  
; bit 0 = SIFreq  
.equ SIFreq = 0  
.equ SIFreqM = 0b00000001  
; bit 1 = SIFsync  
.equ SIFsync = 1  
.equ SIFsyncM = 0b00000010  
; bit 2 = FalseDate  
.equ FalseDate = 2  
.equ FalseDateM = 0b00000100  
; bit 3 = CHKBit  
.equ CHKBit = 3  
.equ CHKBitM = 0b00001000  
; bit 4 = SigKorr  
.equ SigKorr = 4  
.equ SigKorrM = 0b00010000  
; bit 5 = SigVal  
.equ SigVal = 5  
.equ SigValM = 0b00100000  
; bit 6 = TimeVal  
.equ TimeVal = 6  
.equ TimeValM = 0b01000000  
; bit 7 = DebugMode  
.equ DebugMode = 7  
.equ DebugModeM = 0b10000000  
  
.equ digit0 = $0061
```

```

.equ dcfsig = 2

; timervalue (255 - countervalue)
.equ timer0val = 0b11000110 ;198

; -----
; interrupt start addresses
; -----
; vector 1, RESET
.org $000
rjmp init
; vector 2, INT0
.org $001
rjmp INT0R
; vector 3, INT1
.org $002
reti
; vector 4, TIMER1 CAPT
.org $003
reti
; vector 5, TIMER1 COMP
.org $004
rjmp TIMER1COMP
; vector 6, TIMER1 OVF
.org $005
reti
; vector 7, TIMER0 OVF
.org $006
rjmp TIMER0OVF
; vector 8, SPI, STC
.org $007
reti
; vector 9, UART, RX
.org $008
rjmp UARTRX
; vector 10, UART, UDRE
.org $009
reti
; vector 11, UART, TX
.org $00A
reti
; vector 12, ADC
.org $00B
reti
; vector 13, EE_RDY
.org $00C
reti
; vector 14, ANA_COMP
.org $00D
reti
; -----
; initialize the hardware
; -----
.org $00E
init:
; initialize the stackpointer
ldi temp, $A0
out SPL, temp
; initialize port B and C as output
; 1 = out, 0 = in
; both are 6-Bit-Port, Bit 0 and 1 not used
ldi temp, 0b00111111
out DDRB, temp
out DDRC, temp
; initialize port D, Pin 1 as output
; the rest is not used or has special function
; 1 = out, 0 = in
ldi temp, 0b10000000
out DDRD, temp

```



```

; Load digits
ldi YL, low(digit0)
ldi YH, high(digit0)
;sts digit0, temp
ldi temp, 0b00000010
st Y+, temp
;sts digit1, temp
ldi temp, 0b01101011
st Y+, temp
;sts digit2, temp
ldi temp, 0b01010000
st Y+, temp
;sts digit3, temp
ldi temp, 0b01001000
st Y+, temp
;sts digit4, temp
ldi temp, 0b00101001
st Y+, temp
;sts digit5, temp
ldi temp, 0b00001100
st Y+, temp
;sts digit6, temp
ldi temp, 0b00000100
st Y+, temp
;sts digit7, temp
ldi temp, 0b01101010
st Y+, temp
;sts digit8, temp
ldi temp, 0b00000000
st Y+, temp
;sts digit9, temp
ldi temp, 0b00001000
st Y+, temp

; -----
; no operation until interrupt request
; -----
noop:
rjmp noop

; -----
; interrupt routine for timer0 overflow - output the actual time
; -----
TIMER0OVF:
; set new value of the counter
ldi temp, timer0val
out TCNT0, temp
; Jump to displ-6 depending value of DispCounter
ldi ZH, 0b00000000
mov ZL, DispCounter
ldi temp, adr
add ZL, temp
ijmp
adr: rjmp disp0
rjmp disp1
rjmp disp2
rjmp disp3
rjmp disp4
rjmp disp5
disp0:
; Increment DispCounter
inc DispCounter
; Load 4bit value to the temp register
mov temp, AusgSek
andi temp, 0b00001111
; Prepare loading the digit from SRAM
ldi YH, high(digit0)
ldi YL, low(digit0)
add YL, temp

```

```
; Turn off all digits
ldi temp, 0b00111111
out PORTC, temp
; Load the digit from SRAM
ld temp, Y
; Set Pin 7 of Port D to the right value
cbi PORTD, 7
sbrc temp, 6
sbi PORTD, 7
; Set Port B
andi temp, 0b00111111
out PORTB, temp
; Turn on the required digit
ldi temp, 0b00011111
out PORTC, temp
; Return from Interrupt
reti

displ:
; Increment DispCounter
inc DispCounter
; Load 4bit value to the temp register
mov temp, AusgSek
swap temp
andi temp, 0b00001111
; Prepare loading the digit from SRAM
ldi YH, high(digit0)
ldi YL, low(digit0)
add YL, temp
; Turn off all digits
ldi temp, 0b00111111
out PORTC, temp
; Load the digit from SRAM
ld temp, Y
; Set Pin 7 of Port D to the right value
cbi PORTD, 7
sbrc temp, 6
sbi PORTD, 7
; Set Port B
andi temp, 0b00111111
out PORTB, temp
; Turn on the required digit
ldi temp, 0b00101111
out PORTC, temp
; Return from Interrupt
reti

disp2:
; Increment DispCounter
inc DispCounter
; Load 4bit value to the temp register
mov temp, AusgMin
andi temp, 0b00001111
; Prepare loading the digit from SRAM
ldi YH, high(digit0)
ldi YL, low(digit0)
add YL, temp
; Turn off all digits
ldi temp, 0b00111111
out PORTC, temp
; Load the digit from SRAM
ld temp, Y
; Set Pin 7 of Port D to the right value
cbi PORTD, 7
sbrc temp, 6
sbi PORTD, 7
; Set Port B
andi temp, 0b00111111
out PORTB, temp
; Turn on the required digit
```

```
ldi temp, 0b00110111
out PORTC, temp
; Return from Interrupt
reti

disp3:
; Increment DispCounter
inc DispCounter
; Load 4bit value to the temp register
mov temp, AusgMin
swap temp
andi temp, 0b00001111
; Prepare loading the digit from SRAM
ldi YH, high(digit0)
ldi YL, low(digit0)
add YL, temp
; Turn off all digits
ldi temp, 0b00111111
out PORTC, temp
; Load the digit from SRAM
ld temp, Y
; Set Pin 7 of Port D to the right value
cbi PORTD, 7
sbrc temp, 6
sbi PORTD, 7
; Set Port B
andi temp, 0b00111111
out PORTB, temp
; Turn on the required digit
ldi temp, 0b00111011
out PORTC, temp
; Return from Interrupt
reti

disp4:
; Increment DispCounter
inc DispCounter
; Load 4bit value to the temp register
mov temp, AusgStd
andi temp, 0b00001111
; Prepare loading the digit from SRAM
ldi YH, high(digit0)
ldi YL, low(digit0)
add YL, temp
; Turn off all digits
ldi temp, 0b00111111
out PORTC, temp
; Load the digit from SRAM
ld temp, Y
; Set Pin 7 of Port D to the right value
cbi PORTD, 7
sbrc temp, 6
sbi PORTD, 7
; Set Port B
andi temp, 0b00111111
out PORTB, temp
; Turn on the required digit
ldi temp, 0b00111101
out PORTC, temp
; Return from Interrupt
reti

disp5:
; Clear DispCounter
clr DispCounter
; Load 4bit value to the temp register
mov temp, AusgStd
swap temp
andi temp, 0b00001111
; Prepare loading the digit from SRAM
```

```

ldi YH, high(digit0)
ldi YL, low(digit0)
add YL, temp
; Turn off all digits
ldi temp, 0b00111111
out PORTC, temp
; Load the digit from SRAM
ld temp, Y
; Set Pin 7 of Port D to the right value
cbi PORTD, 7
sbrc temp, 6
sbi PORTD, 7
; Set Port B
andi temp, 0b00111111
out PORTB, temp
; Turn on the required digit
ldi temp, 0b00111110
out PORTC, temp
; Return from Interrupt
reti
; -----
; interrup routine for external interrupt on int0
; -----
INTOR:
inc CCounter
mov SigDiffalt_H, SigDiffalt_H
mov SigDiffalt_L, SigDiffalt_L
mov SigDiffalt_H, SigDiff_H
mov SigDiffalt_L, SigDiff_L
clr SigDiff_H
clr SigDiff_L
clr SigLen
reti
; -----
; interrup routine for timer1 compare match
; -----
TIMER1COMP:
; Call Uhrwerk
rcall Uhrwerk
; Call SIF if requested
sbrc flagreg, SIFreq
rcall SIF
; Inc SigDiff
inc SigDiff_L
brne goon001
inc SigDiff_H
goon001:
; if DCF-Sig low inc SigLen and ret i
sbic PIND, dcfsig
rjmp goon002
inc SigLen
reti
goon002:
; if SigLen = 0 then ret i
tst SigLen
brne goon003
reti
goon003:
; cheq the difference to the signal before
; test SigDiffalt_H for zero, if not equal goto goonchkdiff
tst SigDiffalt_H
brne goonchkdiff
; test SigDiffalt_L for a value smaller than 170 (equals 850ms)
mov temp, SigDiffalt_L
cpi temp, 0b10101010
; if higher goto goonchkdiff
brsh goonchkdiff
mov SigDiff_L, SigDiffalt_L
clr SigDiff_H ; SigDiffalt_H is anyway zero
add SigDiff_L, SigLen

```

```
brcc goonchkdiff1
inc SigDiff_H
goonchkdiff1:
mov SigDiffalt_L, SigDiffaltalt_L
mov SigDiffalt_H, SigDiffaltalt_H
clr SigLen
dec CCounter
reti
goonchkdiff:
; send debug output if required
sbrc flagreg, DebugMode
out UDR, CCounter
; verify the detected signals
mov temp, SigLen
cbr flagreg, SigKorrM
cpi temp, 0b00001111
brcs goon008
sbr flagreg, SigKorrM
cbr flagreg, SigValM
cpi temp, 0b00011001
brcs goon008
cbr flagreg, SigKorrM
cpi temp, 0b00100010
brcs goon008
sbr flagreg, SigKorrM
sbr flagreg, SigValM
cpi temp, 0b00101101
brcs goon008
cbr flagreg, SigKorrM
goon008:
; set SigLen to zero
clr SigLen
; test CCounter for a value higher than 59 to prevent hang ups
; caused by two falling edges of dcfsig within 5ms
cpi CCounter, 0b00111100
brcs goon008a
rjmp falsesignal0
goon008a:
; Jump to cc00-59 depending value of CCounter
ldi ZH, high(ccstart00)
ldi ZL, low(ccstart00)
add ZL, CCounter
ijmp
ccstart00:
; case 0
reti
; case 1
rjmp cc01
; case 2
reti
; case 3
reti
; case 4
reti
; case 5
reti
; case 6
reti
; case 7
reti
; case 8
reti
; case 9
reti
; case 10
reti
; case 11
reti
; case 12
reti
```



```
; case 13
reti
; case 14
reti
; case 15
reti
; case 16
reti
; case 17
reti
; case 18
reti
; case 19
reti
; case 20
reti
; case 21
rjmp cc21
; case 22
rjmp cc22
; case 23
rjmp cc23
; case 24
rjmp cc24
; case 25
rjmp cc25
; case 26
rjmp cc26
; case 27
rjmp cc27
; case 28
rjmp cc28
; case 29
rjmp cc29
; case 30
rjmp cc30
; case 31
rjmp cc31
; case 32
rjmp cc32
; case 33
rjmp cc33
; case 34
rjmp cc34
; case 35
rjmp cc35
; case 36
rjmp cc36
; case 37
rjmp cc37
; case 38
rjmp cc38
; case 39
rjmp cc39
; case 40
rjmp cc40
; case 41
rjmp cc41
; case 42
rjmp cc42
; case 43
rjmp cc43
; case 44
rjmp cc44
; case 45
rjmp cc45
; case 46
rjmp cc46
; case 47
rjmp cc47
```

```
; case 48
rjmp cc48
; case 49
rjmp cc49
; case 50
rjmp cc50
; case 51
rjmp cc51
; case 52
rjmp cc52
; case 53
rjmp cc53
; case 54
rjmp cc54
; case 55
rjmp cc55
; case 56
rjmp cc56
; case 57
rjmp cc57
; case 58
rjmp cc58
; case 59
rjmp cc59

cc01:
sbrs flagreg, SigKorr
rjmp falsesignal0
sbrs flagreg, SigVal
rjmp falsesignal0
; test SigDiffalt_H for 1, if not equal goto falsesignal
mov temp, SigDiffalt_H
cpi temp, 0b00000001
brne falsesignal0
; test SigDiffalt_L for a value between 104 and 184
; if false value goto falsesignal
mov temp, SigDiffalt_L
cpi temp, 0b01101000
brcs falsesignal0
cpi temp, 0b10111001
brsh falsesignal0
sbrs flagreg, TimeVal
reti
ldi temp, $00
mov AusgSek, temp
mov AusgMin, Minuten
mov AusgStd, Stunden
mov AusgDay, KalTag
mov AusgMon, Monat
mov AusgYea, Jahr
; 0,1 sec have gone since the beginning of the second
; load UWCounter with 100(10):5(10)=20(10)=00010100(2)
ldi UWCounter, 0b00010100
reti

falsesignal0:
; jump-jump because of relativ jump out of reach
rjmp falsesignal

cc21:
sbrs flagreg, SigKorr
rjmp falsesignal
sbrs flagreg, SigVal
rjmp falsesignal
reti

cc22:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Minuten, 0b00000001
```

```
sbrs flagreg, SigVal
cbr Minuten, 0b00000001
reti
```

```
cc23:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Minuten, 0b00000010
sbrs flagreg, SigVal
cbr Minuten, 0b00000010
reti
```

```
cc24:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Minuten, 0b00000100
sbrs flagreg, SigVal
cbr Minuten, 0b00000100
reti
```

```
cc25:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Minuten, 0b00001000
sbrs flagreg, SigVal
cbr Minuten, 0b00001000
reti
```

```
cc26:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Minuten, 0b00010000
sbrs flagreg, SigVal
cbr Minuten, 0b00010000
reti
```

```
cc27:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Minuten, 0b00100000
sbrs flagreg, SigVal
cbr Minuten, 0b00100000
reti
```

```
cc28:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Minuten, 0b01000000
sbrs flagreg, SigVal
cbr Minuten, 0b01000000
reti
```

```
cc29:
sbrs flagreg, SigKorr
rjmp falsesignal
reti
```

```
cc30:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Stunden, 0b00000001
sbrs flagreg, SigVal
cbr Stunden, 0b00000001
reti
```

```
cc31:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Stunden, 0b00000010
sbrs flagreg, SigVal
```

```
cbr Stunden, 0b00000010  
reti
```

```
cc32:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Stunden, 0b00000100  
sbrs flagreg, SigVal  
cbr Stunden, 0b00000100  
reti
```

```
cc33:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Stunden, 0b00001000  
sbrs flagreg, SigVal  
cbr Stunden, 0b00001000  
reti
```

```
cc34:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Stunden, 0b00010000  
sbrs flagreg, SigVal  
cbr Stunden, 0b00010000  
reti
```

```
cc35:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Stunden, 0b00100000  
sbrs flagreg, SigVal  
cbr Stunden, 0b00100000  
reti
```

```
cc36:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
reti
```

```
cc37:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr KalTag, 0b00000001  
sbrs flagreg, SigVal  
cbr KalTag, 0b00000001  
reti
```

```
cc38:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr KalTag, 0b00000010  
sbrs flagreg, SigVal  
cbr KalTag, 0b00000010  
reti
```

```
cc39:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr KalTag, 0b00000100  
sbrs flagreg, SigVal  
cbr KalTag, 0b00000100  
reti
```

```
cc40:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr KalTag, 0b00001000  
sbrs flagreg, SigVal  
cbr KalTag, 0b00001000
```

```
reti
```

```
cc41:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr KalTag, 0b00010000  
sbrs flagreg, SigVal  
cbr KalTag, 0b00010000  
reti
```

```
cc42:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr KalTag, 0b00100000  
sbrs flagreg, SigVal  
cbr KalTag, 0b00100000  
reti
```

```
cc43:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
reti
```

```
cc44:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
reti
```

```
cc45:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
reti
```

```
cc46:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Monat, 0b00000001  
sbrs flagreg, SigVal  
cbr Monat, 0b00000001  
reti
```

```
cc47:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Monat, 0b00000010  
sbrs flagreg, SigVal  
cbr Monat, 0b00000010  
reti
```

```
cc48:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Monat, 0b00000100  
sbrs flagreg, SigVal  
cbr Monat, 0b00000100  
reti
```

```
cc49:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Monat, 0b00001000  
sbrs flagreg, SigVal  
cbr Monat, 0b00001000  
reti
```

```
cc50:  
sbrs flagreg, SigKorr  
rjmp falsesignal  
sbr Monat, 0b00010000  
sbrs flagreg, SigVal
```

```
cbr Monat, 0b00010000
reti

cc51:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b00000001
sbrs flagreg, SigVal
cbr Jahr, 0b00000001
reti

cc52:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b00000010
sbrs flagreg, SigVal
cbr Jahr, 0b00000010
reti

cc53:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b00000100
sbrs flagreg, SigVal
cbr Jahr, 0b00000100
reti

cc54:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b00001000
sbrs flagreg, SigVal
cbr Jahr, 0b00001000
reti

cc55:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b00010000
sbrs flagreg, SigVal
cbr Jahr, 0b00010000
reti

cc56:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b00100000
sbrs flagreg, SigVal
cbr Jahr, 0b00100000
reti

cc57:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b01000000
sbrs flagreg, SigVal
cbr Jahr, 0b01000000
reti

cc58:
sbrs flagreg, SigKorr
rjmp falsesignal
sbr Jahr, 0b10000000
sbrs flagreg, SigVal
cbr Jahr, 0b10000000
reti

cc59:
sbrs flagreg, SigKorr
rjmp falsesignal
```

```
sbr flagreg, TimeValM
clr CCounter
; date has been received, date is not false
cbr flagreg, FalseDateM
reti
```

```
falsesignal:
cbr flagreg, TimeValM
clr CCounter
reti
```

```
; -----
; Uhrwerk - calculate the time
; -----
Uhrwerk:
; increment UWCounter
inc UWCounter
; if it is 200, a second has gone
cpi UWCounter, 0b11001000
breq gone_second
ret
; a second has gone
gone_second:
; clear UWCounter
clr UWCounter
sbrs flagreg, SIFSync
rjmp goon00
ldi temp, 0b11111111
out UDR, temp
cbr flagreg, SIFSyncM
goon00:
mov temp, AusgSek
andi temp, 0b00001111
cpi temp, 0b00001001
breq goon10
inc AusgSek
ret
goon10:
mov temp, AusgSek
andi temp, 0b11110000
cpi temp, 0b01010000
breq goon11
mov AusgSek, temp
ldi temp, 0b00010000
add AusgSek, temp
ret
goon11:
clr AusgSek
mov temp, AusgMin
andi temp, 0b00001111
cpi temp, 0b00001001
breq goon12
inc AusgMin
ret
goon12:
mov temp, AusgMin
andi temp, 0b11110000
cpi temp, 0b01010000
breq goon13
mov AusgMin, temp
ldi temp, 0b00010000
add AusgMin, temp
ret
goon13:
clr AusgMin
mov temp, AusgStd
cpi temp, 0b00100011
breq goon15
andi temp, 0b00001111
cpi temp, 0b00001001
```

```
breg goon14
inc AusgStd
ret
goon14:
mov temp, AusgStd
andi temp, 0b11110000
mov AusgStd, temp
ldi temp, 0b00010000
add AusgStd, temp
ret
goon15:
; date has changed, date is flase
sbr flagreg, FalseDateM
clr AusgStd
ret
; -----
; SIF - Send the time over the serial interface if requested
; -----
SIF:
ldi ZH, high(SIFstart00)
ldi ZL, low(SIFstart00)
add ZL, UARTCounter
ijmp
SIFstart00:
rjmp SendSec
rjmp SendMin
rjmp SendStd
rjmp SendDay
rjmp SendMon
rjmp SendYea
SendSec:
; if UWCounter smaller than 190(10) data can be sent before the end of the second
cpi UWCounter, 0b10111110
brcs goonsif00
ret
goonsif00:
inc UARTCounter
out UDR, AusgSek
ret
SendMin:
inc UARTCounter
out UDR, AusgMin
ret
SendStd:
inc UARTCounter
out UDR, AusgStd
ret
SendDay:
inc UARTCounter
out UDR, AusgDay
ret
SendMon:
inc UARTCounter
out UDR, AusgMon
ret
SendYea:
clr UARTCounter
out UDR, AusgYea
cbr flagreg, SIFreqM
sbr flagreg, SIFsyncM
ret
; -----
; UART RX complete
; -----
UARTRX:
; detect frame error
sbic UCSRA, 4
rjmp erx
; detect overflow
```



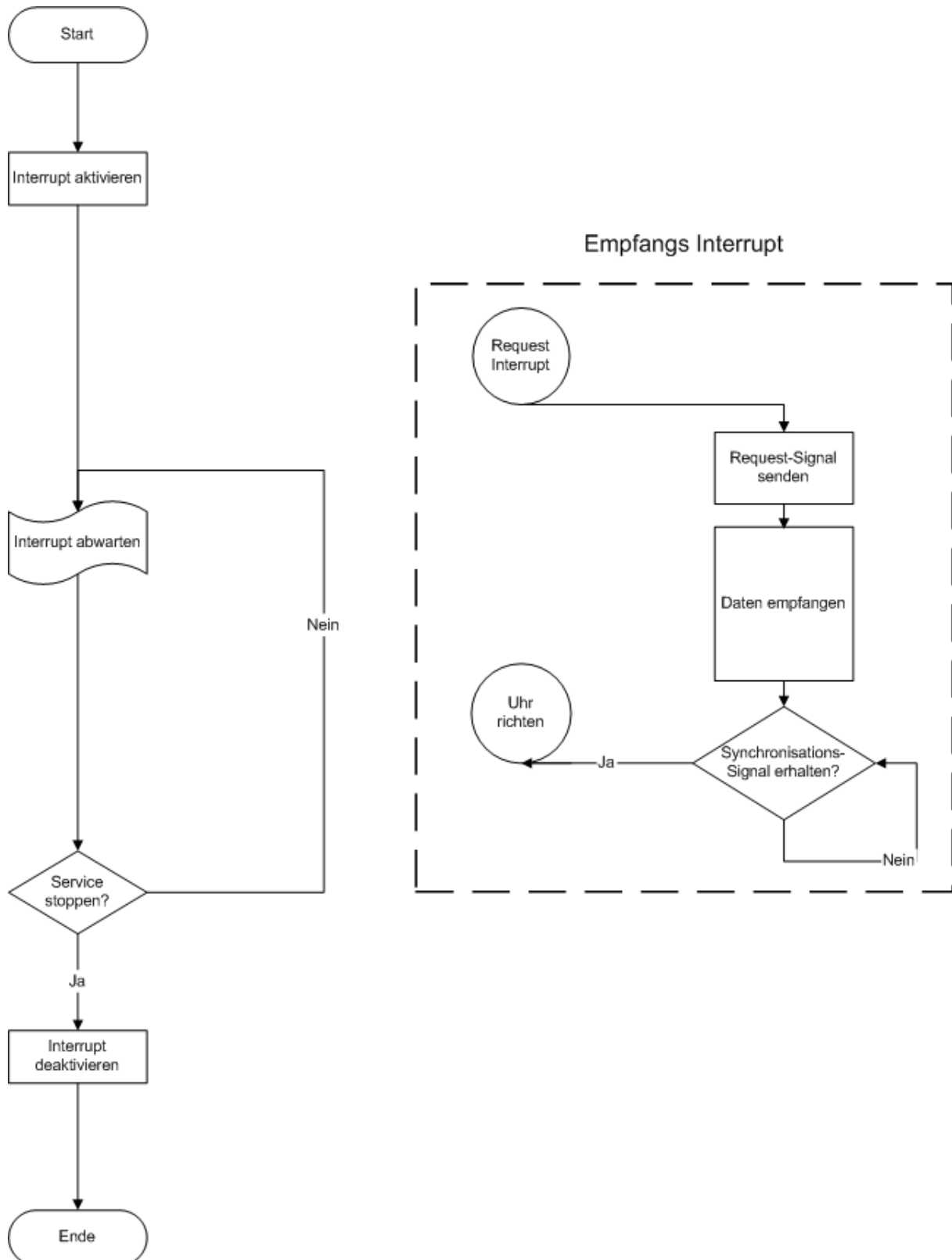
```
sbic UCSRA, 3
rjmp erx
; test if signal is 00(16)
in temp, UDR
tst temp
brne goonrx0
; test if FalseDate not set
sbr flagreg, FalseDate
rjmp erx
; ok, SIFreq can be set
sbr flagreg, SIFreqM
; DebugMode can be disabled
cbr flagreg, DebugModeM
reti
```

```
goonrx0:
; test if signal is FF(16)
cpi temp, 0b11111111
brne goonrx1
; DebugMode can be set
sbr flagreg, DebugModeM
reti
```

```
goonrx1:
; test if signal is FE(16)
cpi temp, 0b11111110
brne erx
; DebugMode can be disabled
cbr flagreg, DebugModeM
reti
```

```
erx:
cbr flagreg, SIFreqM
reti
```

## Anhang E: Flussdiagramm SyncTime (Computer-Software)



## Anhang F: Quellcode SyncTime

```
////////////////////////////////////  
// Copyright 2002 by Daniel Gut, SyncTime, 09.11.2002, C# .Net Framework 1.0 SP2//  
////////////////////////////////////
```

```
using System;  
using System.Drawing;  
using System.Collections;  
using System.ComponentModel;  
using System.Windows.Forms;  
using System.Data;  
using System.Diagnostics;  
using System.Runtime.InteropServices;  
using System.Text;  
using System.Xml;  
using System.IO;  
using Microsoft.Win32;  
  
namespace syncTime  
{  
    public class frmMain : System.Windows.Forms.Form  
    {  
        #region Declare Objects  
        public int intTag;  
        public int intMonat;  
        public int intJahr;  
        public int intStunde;  
        public int intMinute;  
        public int intSekunde;  
        public int intStatusCounter;  
        public bool visibility;  
  
        private AxMSCommLib.AxMSComm com;  
        private System.Windows.Forms.NotifyIcon notIcon;  
        private System.Windows.Forms.ContextMenu cmnNotify;  
        private System.Windows.Forms.Timer tmrRequest;  
        private System.Windows.Forms.MenuItem menuItem1;  
        private System.Windows.Forms.Button cmdCancle;  
        private System.Windows.Forms.Button cmdApply;  
        private System.Windows.Forms.MenuItem mnuOptions;  
        private System.Windows.Forms.GroupBox grpStatus;  
        private System.Windows.Forms.GroupBox grpOptions;  
        private System.Windows.Forms.NumericUpDown txtInterval;  
        private System.Windows.Forms.Label lblMinutes;  
        private System.Windows.Forms.ListBox lstStatus;  
        private System.Windows.Forms.Label lblAktIntervall;  
        private System.Windows.Forms.MenuItem mnuPause;  
        private System.Windows.Forms.MenuItem mnuQuit;  
        private Crownwood.Magic.Controls.TabControl tabOptions;  
        private Crownwood.Magic.Controls.TabPage tabpgOptions;  
        private Crownwood.Magic.Controls.TabPage tabpgState;  
        private System.Windows.Forms.Panel pnlHeader;  
        private System.Windows.Forms.PictureBox picLogo;  
        private System.Windows.Forms.Label lblTitel;  
        private System.Windows.Forms.Button cmdAktual;  
        private System.Windows.Forms.Label lblDescription;  
        private System.Windows.Forms.GroupBox grpCopyright;  
        private System.Windows.Forms.Label lblCopyright;  
        private System.Windows.Forms.CheckBox chkAutorun;  
        private System.ComponentModel.IContainer components;  
        #endregion  
  
        public frmMain()  
        {  
  
            this.visibility = false;  
        }  
    }  
}
```

```
InitializeComponent();

try
{
    XmlTextReader objXmlReader = new
XmlTextReader(Application.StartupPath + "\\settings.xml");
    objXmlReader.ReadStartElement("SyncTime");
    objXmlReader.ReadStartElement("Settings");

    objXmlReader.ReadStartElement("SyncIntervall");
    txtInterval.Value =
Convert.ToInt16(objXmlReader.ReadString());
    objXmlReader.ReadEndElement();

    objXmlReader.ReadStartElement("AutoRun");

    if (objXmlReader.ReadString() == "True")
    {
        RegistryKey key =
Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
", true);
        key.SetValue("SyncTime",
Application.ExecutablePath.ToString());
        chkAutorun.Checked = true;
    }
    else
    {
        try
        {
            RegistryKey key =
Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
", true);
            key.DeleteValue("SyncTime");
            chkAutorun.Checked = false;
        }
        catch{}
    }
    objXmlReader.ReadEndElement();

    objXmlReader.ReadEndElement();
    objXmlReader.ReadEndElement();
    objXmlReader.Close();

}
catch{}
finally{}

this.intStatusCounter = 99;
tmrRequest.Interval = ((int)txtInterval.Value * 60000);
InitComPort();
tmrRequest.Enabled = true;
}

protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}

#region Windows Form
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
```

```

        System.Resources.ResourceManager resources = new
System.Resources.ResourceManager(typeof(frmMain));
        this.tmrRequest = new
System.Windows.Forms.Timer(this.components);
        this.com = new AxMSCommLib.AxMSComm();
        this.notIcon = new
System.Windows.Forms.NotifyIcon(this.components);
        this.cnmNotify = new System.Windows.Forms.ContextMenu();
        this.mnuOptions = new System.Windows.Forms.MenuItem();
        this.mnuPause = new System.Windows.Forms.MenuItem();
        this.menuItem1 = new System.Windows.Forms.MenuItem();
        this.mnuQuit = new System.Windows.Forms.MenuItem();
        this.cmdCancle = new System.Windows.Forms.Button();
        this.cmdApply = new System.Windows.Forms.Button();
        this.grpOptions = new System.Windows.Forms.GroupBox();
        this.chkAutorun = new System.Windows.Forms.CheckBox();
        this.lblAktIntervall = new System.Windows.Forms.Label();
        this.lblMinutes = new System.Windows.Forms.Label();
        this.txtInterval = new System.Windows.Forms.NumericUpDown();
        this.grpStatus = new System.Windows.Forms.GroupBox();
        this.cmdAktual = new System.Windows.Forms.Button();
        this.lstStatus = new System.Windows.Forms.ListBox();
        this.pnlHeader = new System.Windows.Forms.Panel();
        this.lblDescription = new System.Windows.Forms.Label();
        this.lblTitel = new System.Windows.Forms.Label();
        this.picLogo = new System.Windows.Forms.PictureBox();
        this.tabOptions = new Crownwood.Magic.Controls.TabControl();
        this.tabpgOptions = new Crownwood.Magic.Controls.TabPage();
        this.grpCopyright = new System.Windows.Forms.GroupBox();
        this.lblCopyright = new System.Windows.Forms.Label();
        this.tabpgState = new Crownwood.Magic.Controls.TabPage();

        ((System.ComponentModel.ISupportInitialize)(this.com)).BeginInit();
        this.grpOptions.SuspendLayout();

        ((System.ComponentModel.ISupportInitialize)(this.txtInterval)).BeginInit();
        this.grpStatus.SuspendLayout();
        this.pnlHeader.SuspendLayout();
        this.tabpgOptions.SuspendLayout();
        this.grpCopyright.SuspendLayout();
        this.tabpgState.SuspendLayout();
        this.SuspendLayout();
        //
        // tmrRequest
        //
        this.tmrRequest.Interval = 1000;
        this.tmrRequest.Tick += new
System.EventHandler(this.newRequest);
        //
        // com
        //
        this.com.AccessibleDescription =
((string)(resources.GetObject("com.AccessibleDescription")));
        this.com.AccessibleName =
((string)(resources.GetObject("com.AccessibleName")));
        this.com.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("com.Anchor")));
        this.com.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("com.BackgroundImage")));
        this.com.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("com.Dock")));
        this.com.Enabled = ((bool)(resources.GetObject("com.Enabled")));
        this.com.Font =
((System.Drawing.Font)(resources.GetObject("com.Font")));
        this.com.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("com.ImeMode")));
        this.com.Location =
((System.Drawing.Point)(resources.GetObject("com.Location")));
        this.com.Name = "com";

```

```

        this.com.OcxState =
((System.Windows.Forms.AxHost.State)(resources.GetObject("com.OcxState")));
        this.com.RightToLeft =
((bool)(resources.GetObject("com.RightToLeft")));
        this.com.Size =
((System.Drawing.Size)(resources.GetObject("com.Size")));
        this.com.TabIndex =
((int)(resources.GetObject("com.TabIndex")));
        this.com.Text = resources.GetString("com.Text");
        this.com.Visible = ((bool)(resources.GetObject("com.Visible")));
        //
        // notIcon
        //
        this.notIcon.ContextMenu = this.cnmNotify;
        this.notIcon.Icon =
((System.Drawing.Icon)(resources.GetObject("notIcon.Icon")));
        this.notIcon.Text = resources.GetString("notIcon.Text");
        this.notIcon.Visible =
((bool)(resources.GetObject("notIcon.Visible")));
        this.notIcon.DoubleClick += new
System.EventHandler(this.notIcon_DoubleClick);
        //
        // cnmNotify
        //
        this.cnmNotify.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {

                                                                 this.mnuOptions,

                                                                 this.mnuPause,

                                                                 this.menuItem1,

                                                                 this.mnuQuit});

        this.cnmNotify.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("cnmNotify.RightToLeft")));
        //
        // mnuOptions
        //
        this.mnuOptions.Enabled =
((bool)(resources.GetObject("mnuOptions.Enabled")));
        this.mnuOptions.Index = 0;
        this.mnuOptions.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("mnuOptions.Shortcut")));
        this.mnuOptions.ShowShortcut =
((bool)(resources.GetObject("mnuOptions.ShowShortcut")));
        this.mnuOptions.Text = resources.GetString("mnuOptions.Text");
        this.mnuOptions.Visible =
((bool)(resources.GetObject("mnuOptions.Visible")));
        this.mnuOptions.Click += new
System.EventHandler(this.mnuOptions_Click);
        //
        // mnuPause
        //
        this.mnuPause.Enabled =
((bool)(resources.GetObject("mnuPause.Enabled")));
        this.mnuPause.Index = 1;
        this.mnuPause.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("mnuPause.Shortcut")));
        this.mnuPause.ShowShortcut =
((bool)(resources.GetObject("mnuPause.ShowShortcut")));
        this.mnuPause.Text = resources.GetString("mnuPause.Text");
        this.mnuPause.Visible =
((bool)(resources.GetObject("mnuPause.Visible")));
        this.mnuPause.Click += new
System.EventHandler(this.menuItem2_Click);
        //
        // menuItem1
        //

```

```
        this.menuItem1.Enabled =
((bool)(resources.GetObject("menuItem1.Enabled")));
        this.menuItem1.Index = 2;
        this.menuItem1.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem1.Shortcut")));
        this.menuItem1.ShowShortcut =
((bool)(resources.GetObject("menuItem1.ShowShortcut")));
        this.menuItem1.Text = resources.GetString("menuItem1.Text");
        this.menuItem1.Visible =
((bool)(resources.GetObject("menuItem1.Visible")));
        //
        // mnuQuit
        //
        this.mnuQuit.Enabled =
((bool)(resources.GetObject("mnuQuit.Enabled")));
        this.mnuQuit.Index = 3;
        this.mnuQuit.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("mnuQuit.Shortcut")));
        this.mnuQuit.ShowShortcut =
((bool)(resources.GetObject("mnuQuit.ShowShortcut")));
        this.mnuQuit.Text = resources.GetString("mnuQuit.Text");
        this.mnuQuit.Visible =
((bool)(resources.GetObject("mnuQuit.Visible")));
        this.mnuQuit.Click += new
System.EventHandler(this.cmdNotify_Quit_Click);
        //
        // cmdCancle
        //
        this.cmdCancle.AccessibleDescription =
((string)(resources.GetObject("cmdCancle.AccessibleDescription")));
        this.cmdCancle.AccessibleName =
((string)(resources.GetObject("cmdCancle.AccessibleName")));
        this.cmdCancle.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("cmdCancle.Anchor")));
        this.cmdCancle.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("cmdCancle.BackgroundImage")));
        this.cmdCancle.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("cmdCancle.Dock")));
        this.cmdCancle.Enabled =
((bool)(resources.GetObject("cmdCancle.Enabled")));
        this.cmdCancle.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("cmdCancle.FlatStyle")));
        this.cmdCancle.Font =
((System.Drawing.Font)(resources.GetObject("cmdCancle.Font")));
        this.cmdCancle.Image =
((System.Drawing.Image)(resources.GetObject("cmdCancle.Image")));
        this.cmdCancle.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("cmdCancle.ImageAlign")));
        this.cmdCancle.ImageIndex =
((int)(resources.GetObject("cmdCancle.ImageIndex")));
        this.cmdCancle.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("cmdCancle.ImeMode")));
        this.cmdCancle.Location =
((System.Drawing.Point)(resources.GetObject("cmdCancle.Location")));
        this.cmdCancle.Name = "cmdCancle";
        this.cmdCancle.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("cmdCancle.RightToLeft")));
        this.cmdCancle.Size =
((System.Drawing.Size)(resources.GetObject("cmdCancle.Size")));
        this.cmdCancle.TabIndex =
((int)(resources.GetObject("cmdCancle.TabIndex")));
        this.cmdCancle.Text = resources.GetString("cmdCancle.Text");
        this.cmdCancle.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("cmdCancle.TextAlign")));
        this.cmdCancle.Visible =
((bool)(resources.GetObject("cmdCancle.Visible")));
        this.cmdCancle.Click += new
System.EventHandler(this.cmdCancle_Click);
        //
        // cmdApply
```

```
//
    this.cmdApply.AccessibleDescription =
((string)(resources.GetObject("cmdApply.AccessibleDescription")));
    this.cmdApply.AccessibleName =
((string)(resources.GetObject("cmdApply.AccessibleName")));
    this.cmdApply.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("cmdApply.Anchor")));
    this.cmdApply.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("cmdApply.BackgroundImage")));
    this.cmdApply.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("cmdApply.Dock")));
    this.cmdApply.Enabled =
((bool)(resources.GetObject("cmdApply.Enabled")));
    this.cmdApply.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("cmdApply.FlatStyle")));
    this.cmdApply.Font =
((System.Drawing.Font)(resources.GetObject("cmdApply.Font")));
    this.cmdApply.Image =
((System.Drawing.Image)(resources.GetObject("cmdApply.Image")));
    this.cmdApply.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("cmdApply.ImageAlign")));
    this.cmdApply.ImageIndex =
((int)(resources.GetObject("cmdApply.ImageIndex")));
    this.cmdApply.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("cmdApply.ImeMode")));
    this.cmdApply.Location =
((System.Drawing.Point)(resources.GetObject("cmdApply.Location")));
    this.cmdApply.Name = "cmdApply";
    this.cmdApply.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("cmdApply.RightToLeft")));
    this.cmdApply.Size =
((System.Drawing.Size)(resources.GetObject("cmdApply.Size")));
    this.cmdApply.TabIndex =
((int)(resources.GetObject("cmdApply.TabIndex")));
    this.cmdApply.Text = resources.GetString("cmdApply.Text");
    this.cmdApply.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("cmdApply.TextAlign")));
    this.cmdApply.Visible =
((bool)(resources.GetObject("cmdApply.Visible")));
    this.cmdApply.Click += new
System.EventHandler(this.cmdApply_Click);
//
// grpOptions
//
    this.grpOptions.AccessibleDescription =
((string)(resources.GetObject("grpOptions.AccessibleDescription")));
    this.grpOptions.AccessibleName =
((string)(resources.GetObject("grpOptions.AccessibleName")));
    this.grpOptions.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("grpOptions.Anchor")));
    this.grpOptions.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("grpOptions.BackgroundImage")));
    this.grpOptions.Controls.AddRange(new
System.Windows.Forms.Control[] {
//
// grpOptions
//
        this.chkAutorun,
        this.lblAktIntervall,
        this.lblMinutes,
        this.txtInterval});
    this.grpOptions.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("grpOptions.Dock")));
    this.grpOptions.Enabled =
((bool)(resources.GetObject("grpOptions.Enabled")));
    this.grpOptions.FlatStyle =
System.Windows.Forms.FlatStyle.System;
    this.grpOptions.Font =
((System.Drawing.Font)(resources.GetObject("grpOptions.Font")));
```



```
        this.grpOptions.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("grpOptions.ImeMode")));
        this.grpOptions.Location =
((System.Drawing.Point)(resources.GetObject("grpOptions.Location")));
        this.grpOptions.Name = "grpOptions";
        this.grpOptions.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("grpOptions.RightToLeft"))
);
        this.grpOptions.Size =
((System.Drawing.Size)(resources.GetObject("grpOptions.Size")));
        this.grpOptions.TabIndex =
((int)(resources.GetObject("grpOptions.TabIndex")));
        this.grpOptions.TabStop = false;
        this.grpOptions.Text = resources.GetString("grpOptions.Text");
        this.grpOptions.Visible =
((bool)(resources.GetObject("grpOptions.Visible")));
        //
        // chkAutorun
        //
        this.chkAutorun.AccessibleDescription =
((string)(resources.GetObject("chkAutorun.AccessibleDescription")));
        this.chkAutorun.AccessibleName =
((string)(resources.GetObject("chkAutorun.AccessibleName")));
        this.chkAutorun.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("chkAutorun.Anchor")));
        this.chkAutorun.Appearance =
((System.Windows.Forms.Appearance)(resources.GetObject("chkAutorun.Appearance")));
        this.chkAutorun.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("chkAutorun.BackgroundImage")));
        this.chkAutorun.CheckAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("chkAutorun.CheckAlign")));
        this.chkAutorun.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("chkAutorun.Dock")));
        this.chkAutorun.Enabled =
((bool)(resources.GetObject("chkAutorun.Enabled")));
        this.chkAutorun.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("chkAutorun.FlatStyle")));
        this.chkAutorun.Font =
((System.Drawing.Font)(resources.GetObject("chkAutorun.Font")));
        this.chkAutorun.ForeColor =
System.Drawing.SystemColors.ControlText;
        this.chkAutorun.Image =
((System.Drawing.Image)(resources.GetObject("chkAutorun.Image")));
        this.chkAutorun.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("chkAutorun.ImageAlign")));
        this.chkAutorun.ImageIndex =
((int)(resources.GetObject("chkAutorun.ImageIndex")));
        this.chkAutorun.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("chkAutorun.ImeMode")));
        this.chkAutorun.Location =
((System.Drawing.Point)(resources.GetObject("chkAutorun.Location")));
        this.chkAutorun.Name = "chkAutorun";
        this.chkAutorun.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("chkAutorun.RightToLeft"))
);
        this.chkAutorun.Size =
((System.Drawing.Size)(resources.GetObject("chkAutorun.Size")));
        this.chkAutorun.TabIndex =
((int)(resources.GetObject("chkAutorun.TabIndex")));
        this.chkAutorun.Text = resources.GetString("chkAutorun.Text");
        this.chkAutorun.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("chkAutorun.TextAlign")));
        this.chkAutorun.Visible =
((bool)(resources.GetObject("chkAutorun.Visible")));
        //
        // lblAktIntervall
        //
        this.lblAktIntervall.AccessibleDescription =
((string)(resources.GetObject("lblAktIntervall.AccessibleDescription")));
```

```
        this.lblAktIntervall.AccessibleName =
((string)(resources.GetObject("lblAktIntervall.AccessibleName")));
        this.lblAktIntervall.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("lblAktIntervall.Anchor"))
);
        this.lblAktIntervall.AutoSize =
((bool)(resources.GetObject("lblAktIntervall.AutoSize")));
        this.lblAktIntervall.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("lblAktIntervall.Dock")));
        this.lblAktIntervall.Enabled =
((bool)(resources.GetObject("lblAktIntervall.Enabled")));
        this.lblAktIntervall.Font =
((System.Drawing.Font)(resources.GetObject("lblAktIntervall.Font")));
        this.lblAktIntervall.Image =
((System.Drawing.Image)(resources.GetObject("lblAktIntervall.Image")));
        this.lblAktIntervall.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblAktIntervall.ImageAlign"
)));
        this.lblAktIntervall.ImageIndex =
((int)(resources.GetObject("lblAktIntervall.ImageIndex")));
        this.lblAktIntervall.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("lblAktIntervall.ImeMode")));
        this.lblAktIntervall.Location =
((System.Drawing.Point)(resources.GetObject("lblAktIntervall.Location")));
        this.lblAktIntervall.Name = "lblAktIntervall";
        this.lblAktIntervall.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("lblAktIntervall.RightToLef
t")));
        this.lblAktIntervall.Size =
((System.Drawing.Size)(resources.GetObject("lblAktIntervall.Size")));
        this.lblAktIntervall.TabIndex =
((int)(resources.GetObject("lblAktIntervall.TabIndex")));
        this.lblAktIntervall.Text =
resources.GetString("lblAktIntervall.Text");
        this.lblAktIntervall.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblAktIntervall.TextAlign"
)));
        this.lblAktIntervall.Visible =
((bool)(resources.GetObject("lblAktIntervall.Visible")));
        //
        // lblMinutes
        //
        this.lblMinutes.AccessibleDescription =
((string)(resources.GetObject("lblMinutes.AccessibleDescription")));
        this.lblMinutes.AccessibleName =
((string)(resources.GetObject("lblMinutes.AccessibleName")));
        this.lblMinutes.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("lblMinutes.Anchor")));
        this.lblMinutes.AutoSize =
((bool)(resources.GetObject("lblMinutes.AutoSize")));
        this.lblMinutes.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("lblMinutes.Dock")));
        this.lblMinutes.Enabled =
((bool)(resources.GetObject("lblMinutes.Enabled")));
        this.lblMinutes.Font =
((System.Drawing.Font)(resources.GetObject("lblMinutes.Font")));
        this.lblMinutes.Image =
((System.Drawing.Image)(resources.GetObject("lblMinutes.Image")));
        this.lblMinutes.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblMinutes.ImageAlign")));
        this.lblMinutes.ImageIndex =
((int)(resources.GetObject("lblMinutes.ImageIndex")));
        this.lblMinutes.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("lblMinutes.ImeMode")));
        this.lblMinutes.Location =
((System.Drawing.Point)(resources.GetObject("lblMinutes.Location")));
        this.lblMinutes.Name = "lblMinutes";
        this.lblMinutes.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("lblMinutes.RightToLeft"))
);
;
```

```
        this.lblMinutes.Size =
((System.Drawing.Size)(resources.GetObject("lblMinutes.Size")));
        this.lblMinutes.TabIndex =
((int)(resources.GetObject("lblMinutes.TabIndex")));
        this.lblMinutes.Text = resources.GetString("lblMinutes.Text");
        this.lblMinutes.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblMinutes.TextAlign")));
        this.lblMinutes.Visible =
((bool)(resources.GetObject("lblMinutes.Visible")));
        //
        // txtInterval
        //
        this.txtInterval.AccessibleDescription =
((string)(resources.GetObject("txtInterval.AccessibleDescription")));
        this.txtInterval.AccessibleName =
((string)(resources.GetObject("txtInterval.AccessibleName")));
        this.txtInterval.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("txtInterval.Anchor")));
        this.txtInterval.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("txtInterval.Dock")));
        this.txtInterval.Enabled =
((bool)(resources.GetObject("txtInterval.Enabled")));
        this.txtInterval.Font =
((System.Drawing.Font)(resources.GetObject("txtInterval.Font")));
        this.txtInterval.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("txtInterval.ImeMode")));
        this.txtInterval.Location =
((System.Drawing.Point)(resources.GetObject("txtInterval.Location")));
        this.txtInterval.Minimum = new System.Decimal(new int[] {
            1,
            0,
            0,
            0});
        this.txtInterval.Name = "txtInterval";
        this.txtInterval.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("txtInterval.RightToLeft")));
        this.txtInterval.Size =
((System.Drawing.Size)(resources.GetObject("txtInterval.Size")));
        this.txtInterval.TabIndex =
((int)(resources.GetObject("txtInterval.TabIndex")));
        this.txtInterval.TextAlign =
((System.Windows.Forms.HorizontalAlignment)(resources.GetObject("txtInterval.TextAlign")));
        this.txtInterval.ThousandsSeparator =
((bool)(resources.GetObject("txtInterval.ThousandsSeparator")));
        this.txtInterval.UpDownAlign =
((System.Windows.Forms.LeftRightAlignment)(resources.GetObject("txtInterval.UpDownAlign")));
        this.txtInterval.Value = new System.Decimal(new int[] {
            1,
            0,
            0,
            0});
        this.txtInterval.Visible =
((bool)(resources.GetObject("txtInterval.Visible")));
        //
        // grpStatus
        //
        this.grpStatus.AccessibleDescription =
((string)(resources.GetObject("grpStatus.AccessibleDescription")));
```

```
        this.grpStatus.AccessibleName =
((string)(resources.GetObject("grpStatus.AccessibleName")));
        this.grpStatus.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("grpStatus.Anchor")));
        this.grpStatus.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("grpStatus.BackgroundImage")));
        this.grpStatus.Controls.AddRange(new
System.Windows.Forms.Control[] {
                                                    this.cmdAktual,
                                                    this.lstStatus});
        this.grpStatus.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("grpStatus.Dock")));
        this.grpStatus.Enabled =
((bool)(resources.GetObject("grpStatus.Enabled")));
        this.grpStatus.FlatStyle =
System.Windows.Forms.FlatStyle.System;
        this.grpStatus.Font =
((System.Drawing.Font)(resources.GetObject("grpStatus.Font")));
        this.grpStatus.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("grpStatus.ImeMode")));
        this.grpStatus.Location =
((System.Drawing.Point)(resources.GetObject("grpStatus.Location")));
        this.grpStatus.Name = "grpStatus";
        this.grpStatus.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("grpStatus.RightToLeft")));
        this.grpStatus.Size =
((System.Drawing.Size)(resources.GetObject("grpStatus.Size")));
        this.grpStatus.TabIndex =
((int)(resources.GetObject("grpStatus.TabIndex")));
        this.grpStatus.TabStop = false;
        this.grpStatus.Text = resources.GetString("grpStatus.Text");
        this.grpStatus.Visible =
((bool)(resources.GetObject("grpStatus.Visible")));
        //
        // cmdAktual
        //
        this.cmdAktual.AccessibleDescription =
((string)(resources.GetObject("cmdAktual.AccessibleDescription")));
        this.cmdAktual.AccessibleName =
((string)(resources.GetObject("cmdAktual.AccessibleName")));
        this.cmdAktual.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("cmdAktual.Anchor")));
        this.cmdAktual.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("cmdAktual.BackgroundImage")));
        this.cmdAktual.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("cmdAktual.Dock")));
        this.cmdAktual.Enabled =
((bool)(resources.GetObject("cmdAktual.Enabled")));
        this.cmdAktual.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("cmdAktual.FlatStyle")));
        this.cmdAktual.Font =
((System.Drawing.Font)(resources.GetObject("cmdAktual.Font")));
        this.cmdAktual.Image =
((System.Drawing.Image)(resources.GetObject("cmdAktual.Image")));
        this.cmdAktual.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("cmdAktual.ImageAlign")));
        this.cmdAktual.ImageIndex =
((int)(resources.GetObject("cmdAktual.ImageIndex")));
        this.cmdAktual.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("cmdAktual.ImeMode")));
        this.cmdAktual.Location =
((System.Drawing.Point)(resources.GetObject("cmdAktual.Location")));
        this.cmdAktual.Name = "cmdAktual";
        this.cmdAktual.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("cmdAktual.RightToLeft")));
        this.cmdAktual.Size =
((System.Drawing.Size)(resources.GetObject("cmdAktual.Size")));
```

```
        this.cmdAktual.TabIndex =
((int)(resources.GetObject("cmdAktual.TabIndex")));
        this.cmdAktual.Text = resources.GetString("cmdAktual.Text");
        this.cmdAktual.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("cmdAktual.TextAlign")));
        this.cmdAktual.Visible =
((bool)(resources.GetObject("cmdAktual.Visible")));
        this.cmdAktual.Click += new
System.EventHandler(this.cmdAktual_Click);
        //
        // lstStatus
        //
        this.lstStatus.AccessibleDescription =
((string)(resources.GetObject("lstStatus.AccessibleDescription")));
        this.lstStatus.AccessibleName =
((string)(resources.GetObject("lstStatus.AccessibleName")));
        this.lstStatus.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("lstStatus.Anchor")));
        this.lstStatus.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("lstStatus.BackgroundImage")));
        this.lstStatus.ColumnWidth =
((int)(resources.GetObject("lstStatus.ColumnWidth")));
        this.lstStatus.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("lstStatus.Dock")));
        this.lstStatus.Enabled =
((bool)(resources.GetObject("lstStatus.Enabled")));
        this.lstStatus.Font =
((System.Drawing.Font)(resources.GetObject("lstStatus.Font")));
        this.lstStatus.HorizontalExtent =
((int)(resources.GetObject("lstStatus.HorizontalExtent")));
        this.lstStatus.HorizontalScrollbar =
((bool)(resources.GetObject("lstStatus.HorizontalScrollbar")));
        this.lstStatus.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("lstStatus.ImeMode")));
        this.lstStatus.IntegralHeight =
((bool)(resources.GetObject("lstStatus.IntegralHeight")));
        this.lstStatus.ItemHeight =
((int)(resources.GetObject("lstStatus.ItemHeight")));
        this.lstStatus.Location =
((System.Drawing.Point)(resources.GetObject("lstStatus.Location")));
        this.lstStatus.Name = "lstStatus";
        this.lstStatus.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("lstStatus.RightToLeft")));
        this.lstStatus.ScrollAlwaysVisible =
((bool)(resources.GetObject("lstStatus.ScrollAlwaysVisible")));
        this.lstStatus.Size =
((System.Drawing.Size)(resources.GetObject("lstStatus.Size")));
        this.lstStatus.TabIndex =
((int)(resources.GetObject("lstStatus.TabIndex")));
        this.lstStatus.Visible =
((bool)(resources.GetObject("lstStatus.Visible")));
        //
        // pnlHeader
        //
        this.pnlHeader.AccessibleDescription =
((string)(resources.GetObject("pnlHeader.AccessibleDescription")));
        this.pnlHeader.AccessibleName =
((string)(resources.GetObject("pnlHeader.AccessibleName")));
        this.pnlHeader.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("pnlHeader.Anchor")));
        this.pnlHeader.AutoScroll =
((bool)(resources.GetObject("pnlHeader.AutoScroll")));
        this.pnlHeader.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("pnlHeader.AutoScrollMargin")));
        this.pnlHeader.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("pnlHeader.AutoScrollMinSize")));
        this.pnlHeader.BackColor = System.Drawing.Color.White;
        this.pnlHeader.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("pnlHeader.BackgroundImage")));
```

```
        this.pnlHeader.Controls.AddRange(new
System.Windows.Forms.Control[] {

                                                    this.lblDescription,

                                                    this.lblTitel,

                                                    this.picLogo});

        this.pnlHeader.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("pnlHeader.Dock")));
        this.pnlHeader.Enabled =
((bool)(resources.GetObject("pnlHeader.Enabled")));
        this.pnlHeader.Font =
((System.Drawing.Font)(resources.GetObject("pnlHeader.Font")));
        this.pnlHeader.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("pnlHeader.ImeMode")));
        this.pnlHeader.Location =
((System.Drawing.Point)(resources.GetObject("pnlHeader.Location")));
        this.pnlHeader.Name = "pnlHeader";
        this.pnlHeader.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("pnlHeader.RightToLeft")));
        this.pnlHeader.Size =
((System.Drawing.Size)(resources.GetObject("pnlHeader.Size")));
        this.pnlHeader.TabIndex =
((int)(resources.GetObject("pnlHeader.TabIndex")));
        this.pnlHeader.Text = resources.GetString("pnlHeader.Text");
        this.pnlHeader.Visible =
((bool)(resources.GetObject("pnlHeader.Visible")));
        //
        // lblDescription
        //
        this.lblDescription.AccessibleDescription =
((string)(resources.GetObject("lblDescription.AccessibleDescription")));
        this.lblDescription.AccessibleName =
((string)(resources.GetObject("lblDescription.AccessibleName")));
        this.lblDescription.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("lblDescription.Anchor")));
;
        this.lblDescription.AutoSize =
((bool)(resources.GetObject("lblDescription.AutoSize")));
        this.lblDescription.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("lblDescription.Dock")));
        this.lblDescription.Enabled =
((bool)(resources.GetObject("lblDescription.Enabled")));
        this.lblDescription.Font =
((System.Drawing.Font)(resources.GetObject("lblDescription.Font")));
        this.lblDescription.Image =
((System.Drawing.Image)(resources.GetObject("lblDescription.Image")));
        this.lblDescription.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblDescription.ImageAlign")
));
        this.lblDescription.ImageIndex =
((int)(resources.GetObject("lblDescription.ImageIndex")));
        this.lblDescription.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("lblDescription.ImeMode")));
        this.lblDescription.Location =
((System.Drawing.Point)(resources.GetObject("lblDescription.Location")));
        this.lblDescription.Name = "lblDescription";
        this.lblDescription.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("lblDescription.RightToLeft")
));
        this.lblDescription.Size =
((System.Drawing.Size)(resources.GetObject("lblDescription.Size")));
        this.lblDescription.TabIndex =
((int)(resources.GetObject("lblDescription.TabIndex")));
        this.lblDescription.Text =
resources.GetString("lblDescription.Text");
        this.lblDescription.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblDescription.TextAlign")
));
};
```



```
        this.lblDescription.Visible =
((bool)(resources.GetObject("lblDescription.Visible")));
        //
        // lblTitel
        //
        this.lblTitel.AccessibleDescription =
((string)(resources.GetObject("lblTitel.AccessibleDescription")));
        this.lblTitel.AccessibleName =
((string)(resources.GetObject("lblTitel.AccessibleName")));
        this.lblTitel.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("lblTitel.Anchor")));
        this.lblTitel.AutoSize =
((bool)(resources.GetObject("lblTitel.AutoSize")));
        this.lblTitel.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("lblTitel.Dock")));
        this.lblTitel.Enabled =
((bool)(resources.GetObject("lblTitel.Enabled")));
        this.lblTitel.Font =
((System.Drawing.Font)(resources.GetObject("lblTitel.Font")));
        this.lblTitel.Image =
((System.Drawing.Image)(resources.GetObject("lblTitel.Image")));
        this.lblTitel.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblTitel.ImageAlign")));
        this.lblTitel.ImageIndex =
((int)(resources.GetObject("lblTitel.ImageIndex")));
        this.lblTitel.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("lblTitel.ImeMode")));
        this.lblTitel.Location =
((System.Drawing.Point)(resources.GetObject("lblTitel.Location")));
        this.lblTitel.Name = "lblTitel";
        this.lblTitel.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("lblTitel.RightToLeft")));
        this.lblTitel.Size =
((System.Drawing.Size)(resources.GetObject("lblTitel.Size")));
        this.lblTitel.TabIndex =
((int)(resources.GetObject("lblTitel.TabIndex")));
        this.lblTitel.Text = resources.GetString("lblTitel.Text");
        this.lblTitel.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblTitel.TextAlign")));
        this.lblTitel.Visible =
((bool)(resources.GetObject("lblTitel.Visible")));
        //
        // picLogo
        //
        this.picLogo.AccessibleDescription =
((string)(resources.GetObject("picLogo.AccessibleDescription")));
        this.picLogo.AccessibleName =
((string)(resources.GetObject("picLogo.AccessibleName")));
        this.picLogo.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("picLogo.Anchor")));
        this.picLogo.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("picLogo.BackgroundImage")));
        this.picLogo.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("picLogo.Dock")));
        this.picLogo.Enabled =
((bool)(resources.GetObject("picLogo.Enabled")));
        this.picLogo.Font =
((System.Drawing.Font)(resources.GetObject("picLogo.Font")));
        this.picLogo.Image =
((System.Drawing.Bitmap)(resources.GetObject("picLogo.Image")));
        this.picLogo.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("picLogo.ImeMode")));
        this.picLogo.Location =
((System.Drawing.Point)(resources.GetObject("picLogo.Location")));
        this.picLogo.Name = "picLogo";
        this.picLogo.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("picLogo.RightToLeft")));
        this.picLogo.Size =
((System.Drawing.Size)(resources.GetObject("picLogo.Size")));
```

```

        this.picLogo.SizeMode =
((System.Windows.Forms.PictureBoxSizeMode)(resources.GetObject("picLogo.SizeMode")))
);
        this.picLogo.TabIndex =
((int)(resources.GetObject("picLogo.TabIndex")));
        this.picLogo.TabStop = false;
        this.picLogo.Text = resources.GetString("picLogo.Text");
        this.picLogo.Visible =
((bool)(resources.GetObject("picLogo.Visible")));
        //
        // tabOptions
        //
        this.tabOptions.AccessibleDescription =
((string)(resources.GetObject("tabOptions.AccessibleDescription")));
        this.tabOptions.AccessibleName =
((string)(resources.GetObject("tabOptions.AccessibleName")));
        this.tabOptions.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("tabOptions.Anchor")));
        this.tabOptions.Appearance =
Crownwood.Magic.Controls.TabControl.VisualAppearance.MultiDocument;
        this.tabOptions.AutoScroll =
((bool)(resources.GetObject("tabOptions.AutoScroll")));
        this.tabOptions.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("tabOptions.AutoScrollMargin")));
        this.tabOptions.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("tabOptions.AutoScrollMinSize")));
        this.tabOptions.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("tabOptions.BackgroundImage")));
        this.tabOptions.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("tabOptions.Dock")));
        this.tabOptions.Enabled =
((bool)(resources.GetObject("tabOptions.Enabled")));
        this.tabOptions.Font =
((System.Drawing.Font)(resources.GetObject("tabOptions.Font")));
        this.tabOptions.HideTabsMode =
Crownwood.Magic.Controls.TabControl.HideTabsModes.ShowAlways;
        this.tabOptions.IDEPixelBorder = false;
        this.tabOptions.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("tabOptions.ImeMode")));
        this.tabOptions.Location =
((System.Drawing.Point)(resources.GetObject("tabOptions.Location")));
        this.tabOptions.Name = "tabOptions";
        this.tabOptions.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("tabOptions.RightToLeft")))
;
        this.tabOptions.SelectedIndex = 0;
        this.tabOptions.SelectedTab = this.tabpgOptions;
        this.tabOptions.ShowArrows = false;
        this.tabOptions.ShowClose = false;
        this.tabOptions.Size =
((System.Drawing.Size)(resources.GetObject("tabOptions.Size")));
        this.tabOptions.TabIndex =
((int)(resources.GetObject("tabOptions.TabIndex")));
        this.tabOptions.TabPages.AddRange(new
Crownwood.Magic.Controls.TabPage[] {

this.tabpgOptions,

this.tabpgState});
        this.tabOptions.Visible =
((bool)(resources.GetObject("tabOptions.Visible")));
        //
        // tabpgOptions
        //
        this.tabpgOptions.AccessibleDescription =
((string)(resources.GetObject("tabpgOptions.AccessibleDescription")));
        this.tabpgOptions.AccessibleName =
((string)(resources.GetObject("tabpgOptions.AccessibleName")));

```



```

        this.tabpgOptions.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("tabpgOptions.Anchor")));
        this.tabpgOptions.AutoScroll =
((bool)(resources.GetObject("tabpgOptions.AutoScroll")));
        this.tabpgOptions.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("tabpgOptions.AutoScrollMargin")));
        this.tabpgOptions.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("tabpgOptions.AutoScrollMinSize")));
        this.tabpgOptions.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("tabpgOptions.BackgroundImage")));
        this.tabpgOptions.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.tabpgOptions.Controls.AddRange(new
System.Windows.Forms.Control[] {

                                                    this.grpCopyright,

                                                    this.grpOptions});
        this.tabpgOptions.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("tabpgOptions.Dock")));
        this.tabpgOptions.Enabled =
((bool)(resources.GetObject("tabpgOptions.Enabled")));
        this.tabpgOptions.Font =
((System.Drawing.Font)(resources.GetObject("tabpgOptions.Font")));
        this.tabpgOptions.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("tabpgOptions.ImeMode")));
        this.tabpgOptions.Location =
((System.Drawing.Point)(resources.GetObject("tabpgOptions.Location")));
        this.tabpgOptions.Name = "tabpgOptions";
        this.tabpgOptions.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("tabpgOptions.RightToLeft")
));
        this.tabpgOptions.Size =
((System.Drawing.Size)(resources.GetObject("tabpgOptions.Size")));
        this.tabpgOptions.TabIndex =
((int)(resources.GetObject("tabpgOptions.TabIndex")));
        this.tabpgOptions.Text =
resources.GetString("tabpgOptions.Text");
        this.tabpgOptions.Title =
resources.GetString("tabpgOptions.Title");
        this.tabpgOptions.Visible =
((bool)(resources.GetObject("tabpgOptions.Visible")));
        //
        // grpCopyright
        //
        this.grpCopyright.AccessibleDescription =
((string)(resources.GetObject("grpCopyright.AccessibleDescription")));
        this.grpCopyright.AccessibleName =
((string)(resources.GetObject("grpCopyright.AccessibleName")));
        this.grpCopyright.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("grpCopyright.Anchor")));
        this.grpCopyright.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("grpCopyright.BackgroundImage")));
        this.grpCopyright.Controls.AddRange(new
System.Windows.Forms.Control[] {

                                                    this.lblCopyright});
        this.grpCopyright.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("grpCopyright.Dock")));
        this.grpCopyright.Enabled =
((bool)(resources.GetObject("grpCopyright.Enabled")));
        this.grpCopyright.FlatStyle =
System.Windows.Forms.FlatStyle.System;
        this.grpCopyright.Font =
((System.Drawing.Font)(resources.GetObject("grpCopyright.Font")));
        this.grpCopyright.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("grpCopyright.ImeMode")));
        this.grpCopyright.Location =
((System.Drawing.Point)(resources.GetObject("grpCopyright.Location")));
        this.grpCopyright.Name = "grpCopyright";

```

```
        this.grpCopyright.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("grpCopyright.RightToLeft")
));
        this.grpCopyright.Size =
((System.Drawing.Size)(resources.GetObject("grpCopyright.Size")));
        this.grpCopyright.TabIndex =
((int)(resources.GetObject("grpCopyright.TabIndex")));
        this.grpCopyright.TabStop = false;
        this.grpCopyright.Text =
resources.GetString("grpCopyright.Text");
        this.grpCopyright.Visible =
((bool)(resources.GetObject("grpCopyright.Visible")));
        //
        // lblCopyright
        //
        this.lblCopyright.AccessibleDescription =
((string)(resources.GetObject("lblCopyright.AccessibleDescription")));
        this.lblCopyright.AccessibleName =
((string)(resources.GetObject("lblCopyright.AccessibleName")));
        this.lblCopyright.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("lblCopyright.Anchor")));
        this.lblCopyright.AutoSize =
((bool)(resources.GetObject("lblCopyright.AutoSize")));
        this.lblCopyright.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("lblCopyright.Dock")));
        this.lblCopyright.Enabled =
((bool)(resources.GetObject("lblCopyright.Enabled")));
        this.lblCopyright.FlatStyle =
System.Windows.Forms.FlatStyle.System;
        this.lblCopyright.Font =
((System.Drawing.Font)(resources.GetObject("lblCopyright.Font")));
        this.lblCopyright.Image =
((System.Drawing.Image)(resources.GetObject("lblCopyright.Image")));
        this.lblCopyright.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblCopyright.ImageAlign")));
;
        this.lblCopyright.ImageIndex =
((int)(resources.GetObject("lblCopyright.ImageIndex")));
        this.lblCopyright.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("lblCopyright.ImeMode")));
        this.lblCopyright.Location =
((System.Drawing.Point)(resources.GetObject("lblCopyright.Location")));
        this.lblCopyright.Name = "lblCopyright";
        this.lblCopyright.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("lblCopyright.RightToLeft")
));
        this.lblCopyright.Size =
((System.Drawing.Size)(resources.GetObject("lblCopyright.Size")));
        this.lblCopyright.TabIndex =
((int)(resources.GetObject("lblCopyright.TabIndex")));
        this.lblCopyright.Text =
resources.GetString("lblCopyright.Text");
        this.lblCopyright.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("lblCopyright.TextAlign")));
        this.lblCopyright.Visible =
((bool)(resources.GetObject("lblCopyright.Visible")));
        //
        // tabpgState
        //
        this.tabpgState.AccessibleDescription =
((string)(resources.GetObject("tabpgState.AccessibleDescription")));
        this.tabpgState.AccessibleName =
((string)(resources.GetObject("tabpgState.AccessibleName")));
        this.tabpgState.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("tabpgState.Anchor")));
        this.tabpgState.AutoScroll =
((bool)(resources.GetObject("tabpgState.AutoScroll")));
        this.tabpgState.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("tabpgState.AutoScrollMargin")));
```

```
        this.tabpgState.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("tabpgState.AutoScrollMinSize")));
        this.tabpgState.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("tabpgState.BackgroundImage")));
        this.tabpgState.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.tabpgState.Controls.AddRange(new
System.Windows.Forms.Control[] {

                                                    this.grpStatus});
        this.tabpgState.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("tabpgState.Dock")));
        this.tabpgState.Enabled =
((bool)(resources.GetObject("tabpgState.Enabled")));
        this.tabpgState.Font =
((System.Drawing.Font)(resources.GetObject("tabpgState.Font")));
        this.tabpgState.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("tabpgState.ImeMode")));
        this.tabpgState.Location =
((System.Drawing.Point)(resources.GetObject("tabpgState.Location")));
        this.tabpgState.Name = "tabpgState";
        this.tabpgState.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("tabpgState.RightToLeft")));
;
        this.tabpgState.Selected = false;
        this.tabpgState.Size =
((System.Drawing.Size)(resources.GetObject("tabpgState.Size")));
        this.tabpgState.TabIndex =
((int)(resources.GetObject("tabpgState.TabIndex")));
        this.tabpgState.Text = resources.GetString("tabpgState.Text");
        this.tabpgState.Title = resources.GetString("tabpgState.Title");
        this.tabpgState.Visible =
((bool)(resources.GetObject("tabpgState.Visible")));
        //
        // frmMain
        //
        this.AccessibleDescription =
((string)(resources.GetObject("$this.AccessibleDescription")));
        this.AccessibleName =
((string)(resources.GetObject("$this.AccessibleName")));
        this.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("$this.Anchor")));
        this.AutoScaleBaseSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScaleBaseSize")));
        this.AutoScroll =
((bool)(resources.GetObject("$this.AutoScroll")));
        this.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMargin")));
        this.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMinSize")));
        this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
        this.ClientSize =
((System.Drawing.Size)(resources.GetObject("$this.ClientSize")));
        this.Controls.AddRange(new System.Windows.Forms.Control[] {

                                                    this.pnlHeader,

                                                    this.tabOptions,

                                                    this.cmdApply,

                                                    this.cmdCandle,

                                                    this.com});
        this.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("$this.Dock")));
        this.Enabled = ((bool)(resources.GetObject("$this.Enabled")));
        this.Font =
((System.Drawing.Font)(resources.GetObject("$this.Font")));
```

```

        this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedSingle;
        this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
        this.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("$this.ImeMode")));
        this.Location =
((System.Drawing.Point)(resources.GetObject("$this.Location")));
        this.MaximizeBox = false;
        this.MaximumSize =
((System.Drawing.Size)(resources.GetObject("$this.MaximumSize")));
        this.MinimizeBox = false;
        this.MinimumSize =
((System.Drawing.Size)(resources.GetObject("$this.MinimumSize")));
        this.Name = "frmMain";
        this.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("$this.RightToLeft")));
        this.ShowInTaskbar = false;
        this.StartPosition =
((System.Windows.Forms.FormStartPosition)(resources.GetObject("$this.StartPosition"
)));
        this.Text = resources.GetString("$this.Text");
        this.Visible = ((bool)(resources.GetObject("$this.Visible")));
        this.Closing += new
System.ComponentModel.CancelEventHandler(this.frmMain_Closing);
        this.VisibleChanged += new
System.EventHandler(this.frmMain_VisibleChanged);

        ((System.ComponentModel.ISupportInitialize)(this.com)).EndInit();
        this.grpOptions.ResumeLayout(false);

        ((System.ComponentModel.ISupportInitialize)(this.txtInterval)).EndInit();
        this.grpStatus.ResumeLayout(false);
        this.pnlHeader.ResumeLayout(false);
        this.tabpgOptions.ResumeLayout(false);
        this.grpCopyright.ResumeLayout(false);
        this.tabpgState.ResumeLayout(false);
        this.ResumeLayout(false);
    }
#endregion

/// <summary>
/// The main entry point for the application.
/// </summary>

[STAThread]
static void Main()
{
    if (!PrevInstance())
        Application.Run(new frmMain());
    else
        Application.Exit();
}

#region Eventlog Info schreiben
public void writeevent_info(string sSource, string sEvent)
{
    try
    {
        string sLog = "Application";
        if (!EventLog.SourceExists(sSource))
            EventLog.CreateEventSource(sSource, sLog);
        EventLog.WriteEntry(sSource, sEvent,
EventLogEntryType.Information, 1337);
    }
    catch{}
}
#endregion

```

```

#region Eventlog-Warning schreiben
public void writeevent_warning(string sSource, string sEvent)
{
    try
    {
        string sLog = "Application";
        if (!EventLog.SourceExists(sSource))
            EventLog.CreateEventSource(sSource, sLog);
        EventLog.WriteEntry(sSource, sEvent,
EventLogEntryType.Warning, 1337);
    }
    catch {}
}
#endregion

#region Com-Port Initialisierung
private void InitComPort()
{
    try
    {
        // Set the com port to be 1
        com.CommPort = 1;

        // This port is already open, close it to reset it.
        if (com.PortOpen) com.PortOpen = false;

        // Trigger the OnComm event whenever data is recieved
        com.RThreshold = 1;

        // Set the port to 9600 baud, no parity bit, 8 data bits,
1 stop bit (all standard)
        com.Settings = "9600,N,8,1";

        // Force the DTR line high, used sometimes to hang up
modems
        com.DTREnable = true;

        // No handshaking is used
        com.Handshaking = MSCommLib.HandshakeConstants.comNone;

        // Don't mess with binary streams
        com.InputMode =
MSCommLib.InputModeConstants.comInputModeBinary;

        // Read the entire waiting data when com.Input is used
        com.InputLen = 0;

        // Don't discard nulls, 0x00 is a useful byte
        com.NullDiscard = false;

        // Attach the event handler
        com.OnComm += new System.EventHandler(this.OnComm);

        // Open the com port
        com.PortOpen = true;
    }
    catch (System.Runtime.InteropServices.COMException)
    {
        Application.Exit();
    }
}
#endregion

#region Com-Port receive event
private void OnComm(object sender, EventArgs e) // MSCommLib OnComm
Event Handler
{
    // If data is waiting in the buffer, process it.
    if (com.InBufferCount > 0)
        ProcessComData((byte[]) com.Input);
}

```

```
    }

    #endregion

    #region Daten über Com-Port empfangen und verarbeiten
    private void ProcessComData(byte[] input)
    {
        string strInput;
        int intNumber;

        for (int intCounter = input.GetLowerBound(0); intCounter <=
input.GetUpperBound(0); intCounter++)
        {
            strInput = input[intCounter].ToString();
            intNumber =
ConvertBCD(Convert.ToString(input[intCounter], 2));

            if (input[intCounter] == 255)
            {
                //Zeit setzen und Status-Fenster aktualisieren.
                if (intStunde != 0 && intMinute != 0 && intSekunde
!= 0)
                {
                    SetTime(ChangeFormat(intStunde.ToString()) +
":" + ChangeFormat(intMinute.ToString()) + ":" +
ChangeFormat(intSekunde.ToString()));

                    //Status schreiben
                    Object objInput = new Object();
                    objInput = "Zeit: " +
ChangeFormat(intStunde.ToString()) + ":" + ChangeFormat(intMinute.ToString()) + ":"
+ ChangeFormat(intSekunde.ToString()) + " wurde erfolgreich aktualisiert";
                    lstStatus.Items.Insert(0, objInput);
                    this.intStatusCounter = 99;
                }

                //Datum aktualisieren
                if (intTag != 0 && intMonat != 0 && intJahr != 0)
                {
                    SetDate(ChangeFormat(intTag.ToString()) + "-"
+ ChangeFormat(intMonat.ToString()) + "-" + ChangeFormat(intJahr.ToString()));
                }

                //Com-Port schliessen
                com.PortOpen = false;
            }

            switch (this.intStatusCounter)
            {
                case 1:
                    intSekunde = intNumber;
                    intStatusCounter++;
                    break;
                case 2:
                    intMinute = intNumber;
                    intStatusCounter++;
                    break;
                case 3:
                    intStunde = intNumber;
                    intStatusCounter++;
                    break;
                case 4:
                    intTag = intNumber;
                    intStatusCounter++;
                    break;
                case 5:
                    intMonat = intNumber;
                    intStatusCounter++;
                    break;
            }
        }
    }
}

```

```
        case 6:
            intJahr = 2000 + intNumber;
            intStatusCounter = 1;
            break;
        }
    }
}
#endregion

#region Daten über Com-Port senden
private bool SendData(string strdata)
{
    if (com.PortOpen)
    {
        com.Output = strdata;
        return true;
    }
    else
        return false;
}
#endregion

#region Funktion um die System-Zeit zu setzen
private void SetTime(string strTime)
{
    Process setTimeProcess = new Process();
    setTimeProcess.StartInfo.WindowStyle =
System.Diagnostics.ProcessWindowStyle.Hidden;
    setTimeProcess.StartInfo.Arguments = "/c time " + strTime;
    setTimeProcess.Start();
}
#endregion

#region Funktion um das System-Datum zu setzen
private void SetDate(string strDate)
{
    Process setDateProcess = new Process();
    setDateProcess.StartInfo.WindowStyle =
System.Diagnostics.ProcessWindowStyle.Hidden;
    setDateProcess.StartInfo.Arguments = "/c date " + strDate;
    setDateProcess.Start();
}
#endregion

private void cmdNotify_Quit_Click(object sender, System.EventArgs e)
{
    if (com.PortOpen)
        com.PortOpen = false;
    if (tmrRequest.Enabled)
        tmrRequest.Enabled = false;
    Application.Exit();
}

#region Neuer Request an Uhr senden
public void newRequest(object sender, System.EventArgs e)
{
    SendNewRequest();
    this.intStatusCounter = 1;
}
#endregion

private void menuItem2_Click(object sender, System.EventArgs e)
{
    //Switch für die Pause (Context-Menü des Notify-Icons)
    switch (mnuPause.Checked)
    {
        case false:
            tmrRequest.Enabled = false;
            mnuPause.Checked = true;
    }
}
```

```

        break;
    case true:
        tmrRequest.Enabled = true;
        mnuPause.Checked = false;
        SendNewRequest();
        this.intStatusCounter = 1;
        break;
    }
}

private void cmdCancle_Click(object sender, System.EventArgs e)
{
    this.Visible = false;
}

private void mnuOptions_Click(object sender, System.EventArgs e)
{
    this.visibility = true;
    this.Visible = true;
}

private void frmMain_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
{
    //e.Cancel = true;
    //this.Visible = false;
}

public void SendNewRequest()
{
    //Com-port öffnen falls geschlossen.
    if (!com.PortOpen)
        com.PortOpen = true;

    //Request an die Uhr senden
    if(this.intStatusCounter != 99) //Status schreiben
    {
        Object objInput = new Object();
        objInput = "letzte Synchronisation konnte nicht
erfolgreich durchgeführt werden!";
        lstStatus.Items.Insert(0,objInput);
        writeevent_warning("SyncTime","letzte Synchronisation
konnte nicht erfolgreich durchgeführt werden!");
    }

    this.intStatusCounter = 1;
    byte[] bytRequest = new byte[1];
    bytRequest[0] = 0;
    SendData(Encoding.ASCII.GetString(bytRequest));
}

static int ConvertBCD(string strBCD)
{
    //BCD Code in Binäre-Elemente zerlegen
    string retString;
    while(strBCD.Length < 8)
    {
        strBCD = "0" + strBCD;
    }

    retString = BinToDec(strBCD.Substring(0,4)).ToString();
    retString += BinToDec(strBCD.Substring(4,4)).ToString();
    return Convert.ToInt16(retString);
}

static int BinToDec(string strBin)
{
    //Konvertieren von Binär nach Dezimal
    int intDec = 0;

```



```

        int intMultip = 1;
        for (int intCounter = 0; intCounter <= 3;intCounter ++)
        {
            if (strBin.Substring(3 - intCounter,1) == "1")
                intDec += intMultip;

            intMultip *= 2;
        }

        return intDec;
    }

    static string ChangeFormat(string strZahl)
    {
        //Anpassung der Zahlen, die nur 1-Stellig sind
        if (strZahl.Length == 1)
            return "0" + strZahl;
        else
            return strZahl;
    }

    private void cmdAktual_Click(object sender, System.EventArgs e)
    {
        //neuer Request für die manuelle Synchronisation
        SendNewRequest();
    }

    private void cmdApply_Click(object sender, System.EventArgs e)
    {
        //Einstellungen speichern
        XmlTextWriter objXMLWriter = new
        XmlTextWriter(Application.StartupPath +
        "\\settings.xml", System.Text.Encoding.UTF8);
        objXMLWriter.WriteStartDocument();
        objXMLWriter.WriteStartElement("SyncTime");
        objXMLWriter.WriteStartElement("Settings");

        objXMLWriter.WriteElementString("SyncIntervall",txtInterval.Value.ToString())
;

        objXMLWriter.WriteElementString("AutoRun",chkAutorun.Checked.ToString());
        objXMLWriter.WriteEndElement();
        objXMLWriter.WriteEndElement();
        objXMLWriter.Flush();
        objXMLWriter.Close();

        //Autostart-Einträge in der Registry durchführen
        if (chkAutorun.Checked)
        {
            //Autostart aktivieren
            RegistryKey key =
            Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
            ", true);

            key.SetValue("SyncTime",
            Application.ExecutablePath.ToString());
            chkAutorun.Checked = true;
        }
        else
        {
            try
            {
                //Autostart deaktivieren
                RegistryKey key =
                Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
                ", true);

                key.DeleteValue("SyncTime");
                chkAutorun.Checked = false;
            }
            catch{}
        }
    }
}

```

```
        tmrRequest.Interval = ((int)txtInterval.Value * 60000);
        this.Visible = false;
    }

    private void notIcon_DoubleClick(object sender, System.EventArgs e)
    {
        this.visibility = true;
        this.Visible = true;
    }

    static bool PrevInstance()
    {
        if
(System.Diagnostics.Process.GetProcessesByName(System.Diagnostics.Process.GetCurrent
tProcess().ToString()).Length > 0)
            return true;
        else
            return false;
    }

    private void frmMain_VisibleChanged(object sender, System.EventArgs e)
    {
        if (!this.visibility)
            this.Visible = false;
    }
}
}
```

## **Anhang G: Arbeitsjournal**

## **Anhang H: Selbsteinschätzung**

## **Anhang I: Beurteilung durch Lehrer**