

; Ampelsteuerung

; =========================

JMP Start

; Datenbereich

DB 90 ; rot - rot

DB 98 ; rot - rot/orange

DB 84 ; rot - grün

DB 88 ; rot - grün/orange

DB 90 ; rot - rot

DB D0 ; rot/orange - rot

DB 30 ; grün - rot

DB 50 ; grün/orange - rot

; =========================

; Programmstart

Start:

MOV BL,2 ; Start des Datenbereichs in BL

Rep1:

MOV AL,[BL] ; Wert von Adresse BL in AL

OUT 01 ; Ausgabe von AL an Port 01

CMP AL,50 ; IF AL = 70 Datenbereich fertig

JZ Start ; THEN GOTO Start

INC BL ; BL = BL + 1

CMP AL,84 ; IF AL = 84 Ampel auf rot - grün

JZ SleepMittel ; THEN GOTO SleepMittel

CMP AL,30 ; IF AL = 30 Ampel auf grün - rot

JZ SleepLang ; THEN GOTO SleepLang

JMP Rep1 ; GOTO Rep1

; Vorbereitung für mittleres Warten

SleepMittel:

MOV CL,5 ; Wartezeit in CL

CALL 40 ; Aufruf Prozedur Sleep

JMP Rep1 ; GOTO Rep1

; Vorbereitung für langes Warten

SleepLang:

MOV CL,10 ; Wartezeit in CL

CALL 40 ; Aufruf Prozedur Sleep

JMP Rep1 ; GOTO Rep1

; -------------------------

; Start Prozedur Sleep

ORG 40 ; Festlegen des Prozedurstarts im Speicher

PUSHF ; CPU-Flags sichern

Sleep:

DEC CL ; CL = CL - 1

JNZ Sleep ; Wiederholung cbis CL = 0

POPF ; CPU-Flags zurücksichern

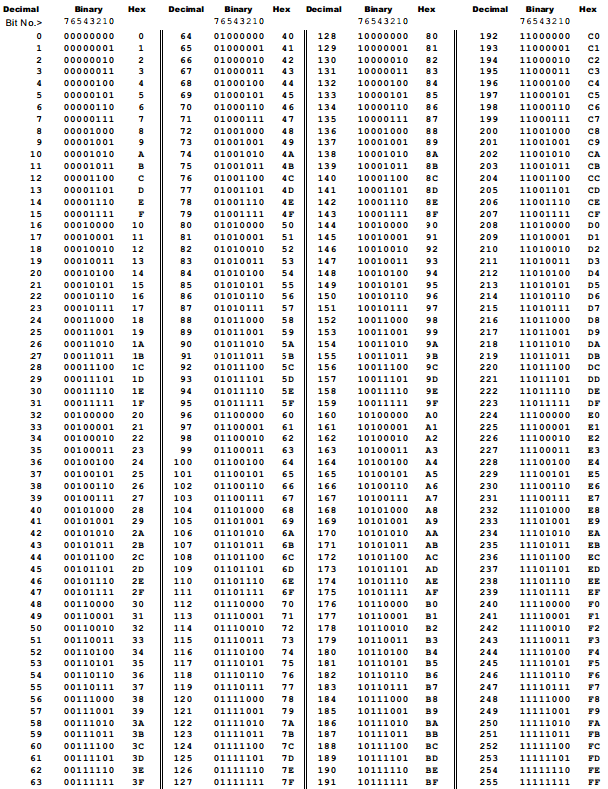
RET ; Zurück nach Aufrufort

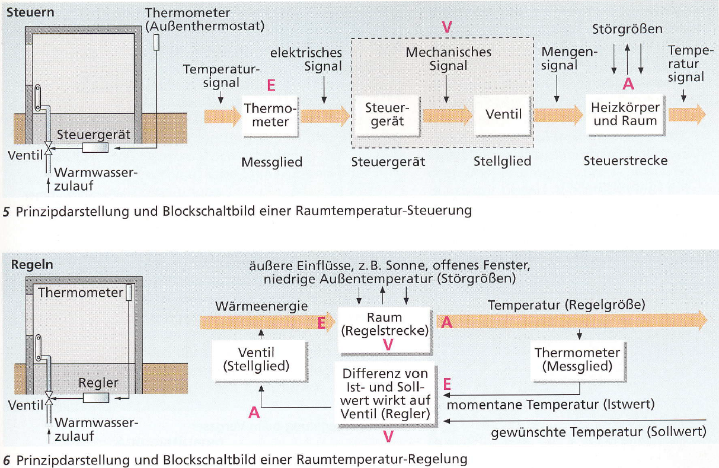
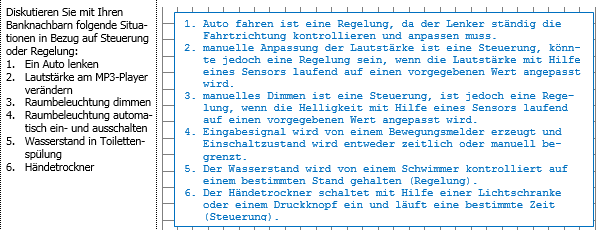
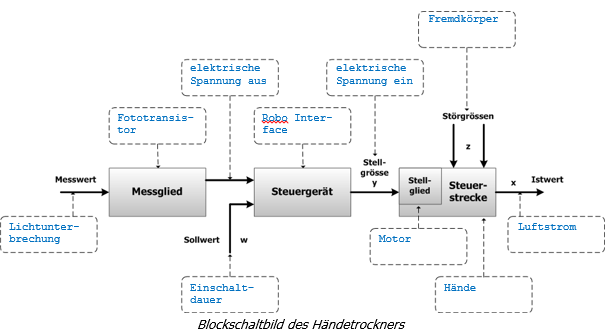
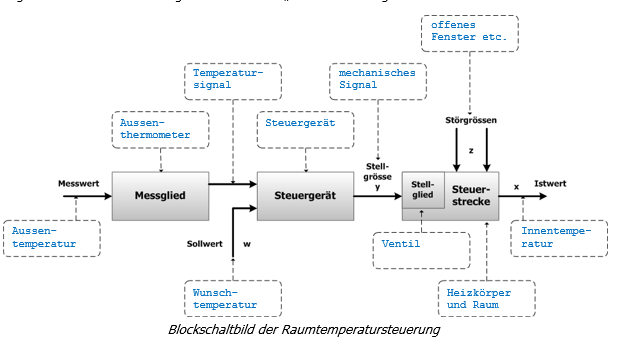
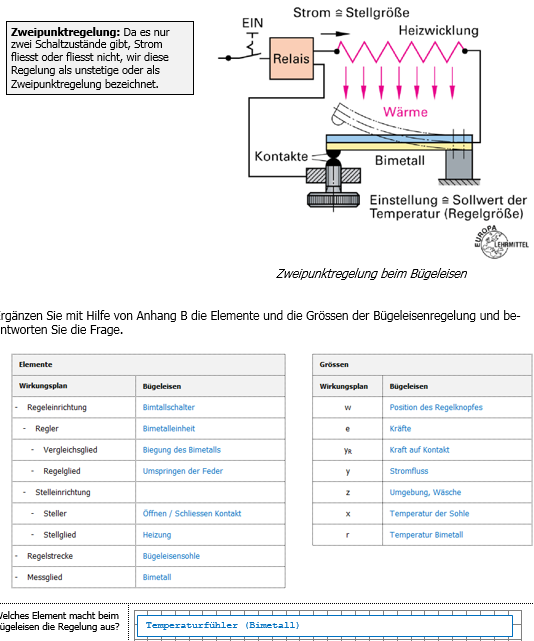
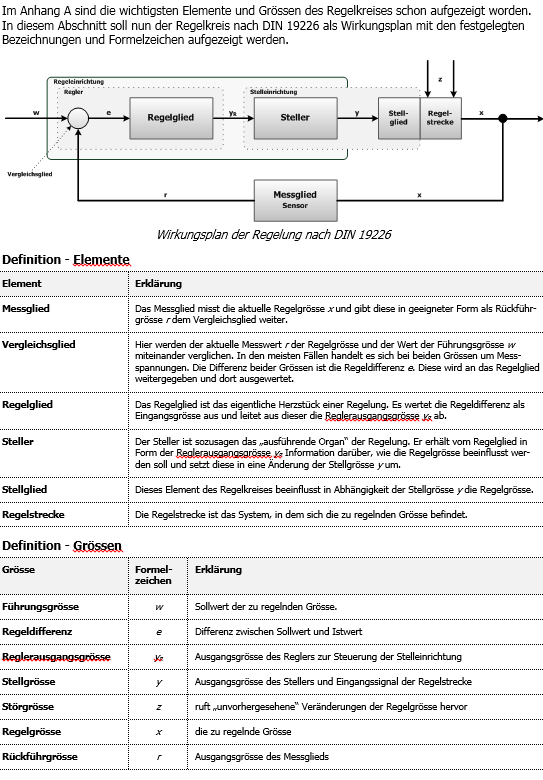
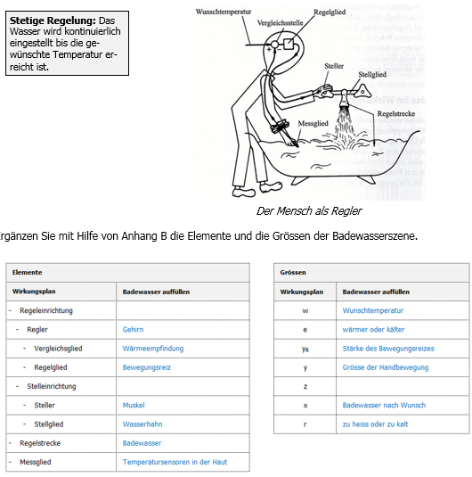
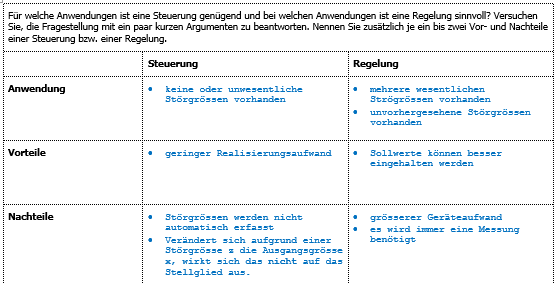
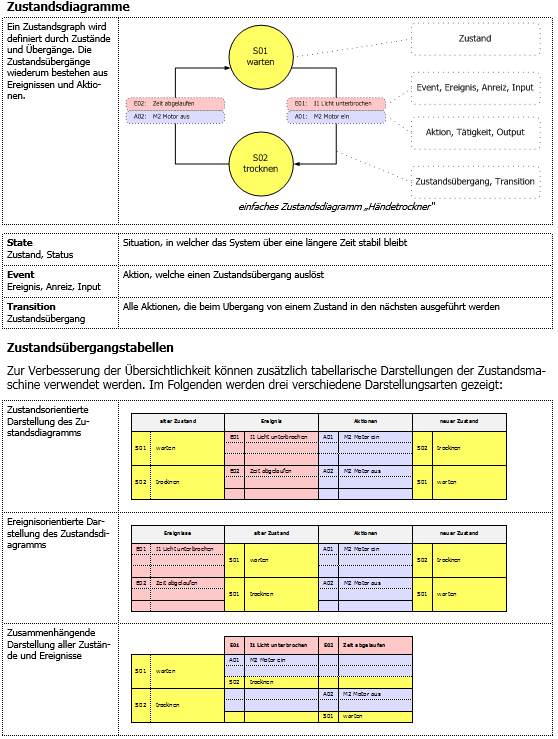
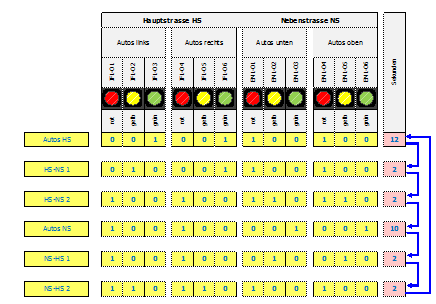
; Ende Prozedur Sleep

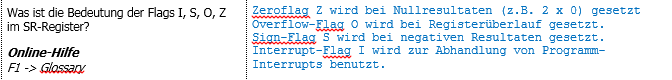
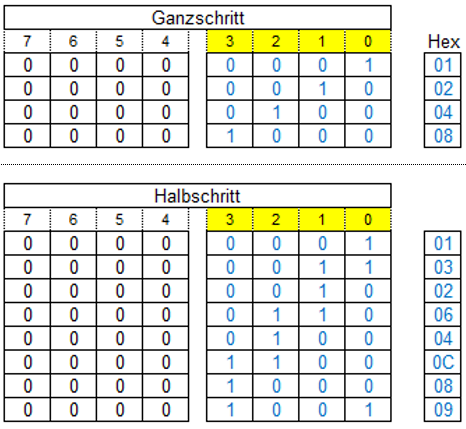
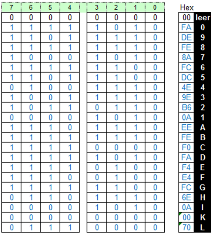
; -------------------------

END ; Programm Ende

; Programm Ende







; =========================

; Schrittmotor - Ganzschritte-Halbschritte

; =========================

JMP Start

; Datenbereich

DB 01

DB 02

DB 04

DB 08

DB 01

DB 03

DB 02

DB 06

DB 04

DB 0C

DB 08

DB 09

; =========================

; Programmstart

Start:

MOV BL,2 ; Start des Datenbereichs in BL

Rep1:

MOV AL,[BL] ; Wert von Adresse BL in AL

OUT 05 ; Ausgabe von AL an Port 01

CMP AL,09 ; IF AL = 70 Datenbereich fertig

JZ Start ; THEN GOTO Start

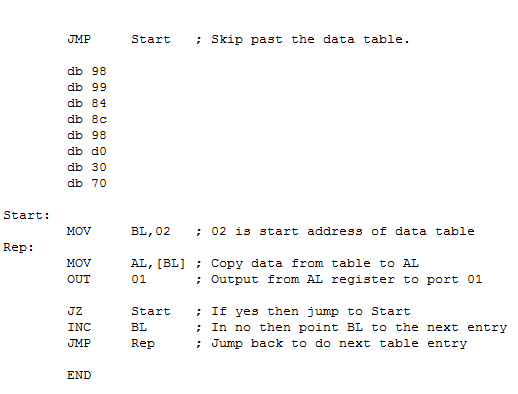
INC BL ; BL = BL + 1

JMP Rep1 ; GOTO Rep1

END

; Programm Ende

; =========================



; =========================

; Schrittmotor - Ganzschritte

; =========================

JMP Start

; Datenbereich

DB 01

DB 02

DB 04

DB 08

; =========================

; Programmstart

Start:

MOV BL,2 ; Start des Datenbereichs in BL

Rep1:

MOV AL,[BL] ; Wert von Adresse BL in AL

OUT 05 ; Ausgabe von AL an Port 01

CMP AL,08 ; IF AL = 70 Datenbereich fertig

JZ Start ; THEN GOTO Start

INC BL ; BL = BL + 1

JMP Rep1 ; GOTO Rep1

END

; Programm Ende

; =========================

; Schrittmotor - Seven Segment Cont-Down

; =========================

JMP Start

; Datenbereich

DB FA

DB DF

DB FF

DB 8B

DB FD

DB DD

DB 4F

DB 9F

DB B7

DB 0B

DB FB

Start:

MOV BL,02 ; 02 ist Startadresse des Datenbereichs

Rep1:

MOV AL,[BL] ; Kopiere Daten vom Datenbereich nach AL

OUT 02 ; Wert von AL nach Port 01 ausgeben

CMP AL,FB ; ist der letzte Datenbereich erreicht

JZ Blink ; wenn ja, springe zu Blink

INC BL ; wenn nein, setzte Zeiger in BL auf nächsten Datenbereich

JMP Rep1

Blink:

MOV CL,5

Rep2:

MOV AL,0

OUT 02

MOV AL,1

OUT 02

MOV AL,FA

OUT 02

MOV AL,FB

OUT 02

DEC CL ; Subtrahiere 1 von CL

JNZ Rep2 ; zurück zu Rep2 wenn CL nicht = 0

JMP Start ; sonst zurück zum Start

END

; =========================

; Schrittmotor - Halbschritte

; =========================

JMP Start

; Datenbereich

DB 01

DB 03

DB 02

DB 06

DB 04

DB 0C

DB 08

DB 09

; =========================

; Programmstart

Start:

MOV BL,2 ; Start des Datenbereichs in BL

Rep1:

MOV AL,[BL] ; Wert von Adresse BL in AL

OUT 05 ; Ausgabe von AL an Port 01

CMP AL,09 ; IF AL = 70 Datenbereich fertig

JZ Start ; THEN GOTO Start

INC BL ; BL = BL + 1

JMP Rep1 ; GOTO Rep1

END

; Programm Ende

; =========================

; ===== WORK OUT 2 PLUS 2 ======================================

CLO ; Close unwanted windows.

MOV AL,2 ; Copy a 2 into the AL register.

MOV BL,7F ; Copy a 2 into the BL register.

ADD AL,BL ; Add AL to BL. Answer goes into AL.

END ; Program ends

; ===== Program Ends ===========================================