

Lehrgang: ccbasic

1. Anschlussbelegungen

1.1 Controller Platine

	+U Eingang	
	0V Eingang	
	+U Ausgang	
+5V		+5V
Port 16		Port 1
Port 15		Port 2
Port 14		Port 3
Port 13		Port 4
Port 12		Port 5
Port 11		Port 6
Port 10		Port 7
Port 9		Port 8
Beep		frei
AD 8		frei
frei		frei
frei		frei
AD 5		frei
AD 4		frei
AD 3		frei
AD 2		frei
AD 1		frei
U ref		frei
0V		0V

1.2 Anschlüsse des Autos

Schalter für Motoren	Analog - Digital Wandler				
oben Ein	LDR	Temperatur	Licht-hinten	Licht-vorn	für Ampel
unten Aus	AD 5	AD 4	AD 3	AD 2	AD 1

1.3 Ampelplatine

	rt	gn	
	P16	P15	
gn	ge	rt	rt P14 Rotausfall AD 1 stecken
P9	P10	P11	ge P13
			gn P12
			Anforderung AD 8

1.4 Solarplatine

Spannungsversorgung
+U (10V)
0V

Schalterstellungen
0 1 Port 16
0 1 Port 15
0 1 Port 14

Eingang 1 AD 8
Eingang 2 AD 7
Eingang 3 AD 4
Eingang 4 AD 1

0 1 Port 11

2. Einführende Beispiele

2.1 ' ccbasic\ubl\hallo1.bas
' Ausgabe von Text auf dem Miniterminal

```
print "Hallo!"  
pause 100  
print "Herr ";  
print "Ubl"
```

2.2 ' ccbasic\ubl\hallo2.bas
' Ausgabe von Text auf dem Miniterminal
' mit Endlosschleife

```
#loop  
  print "Hallo!"  
  pause 50  
  print "Herr ";  
  print "Ubl"  
  pause 200  
goto loop
```

2.3 ' ccbasic\ubl\ton1.bas
' Ausgabe eines Tones
' mit einer Endlosschleife

```
#loop  
  beep 625,12,1  
  pause 50  
goto loop
```

2.4 ' ccbasic\ubl\ad1.bas
' ad wandler

```
define wert ad[1] : define leds byteport[2]  
  
#loop  
  print wert : leds = wert : pause 50  
goto loop
```

2.5 ' ccbasic\ubl\led1.bas
' Leuchtdiode blinken lassen
' mit einer Endlosschleife

```
define led1 port[1]  
  
#loop  
  led1 = on  
  pause 100  
  led1 = off  
  pause 100  
goto loop
```

2.6 ' ccbasic\ubl\led2.bas
' Leuchtdioden schalten mit byteport
' mit einer Endlosschleife

```
define leds byteport[2]

#loop
  leds = &b10101010      ' d172
  pause 100
  leds = &b01010101      ' d85
  pause 100
goto loop
```

2.7 ' ccbasic\ubl\led3.bas
' Lauflicht

```
define leds byteport[2]
define t1 word

t1=40

#loop
  leds = &b10000001 : pause t1
  leds = &b01000010 : pause t1
  leds = &b00100100 : pause t1
  leds = &b00011000 : pause t1
  t1 = t1-2
  if t1 < 4 then t1 = 4
goto loop
```

2.8 ' ccbasic\ubl\einaus1.bas
'Test von Eingängen und Ausgängen

```
' leds
define led1 port[9] : define led2 port[10]
define led3 port[11] : define led4 port[12]

' taster
define taste1 port[5] : define taste2 port[6]
define taste3 port[7] : define taste4 port[8]

#loop
  led1=taste1 : led2=taste2
  led3=taste3 : led4=taste4
goto loop
```

3. Ampelanlage

Rotausfall für rechte LED: Port 14 mit ad1 (unten) verbinden

```
' ccbasic\ubl\tampell.bas
' für Ampelplatine Rotausfall: Port 14 stecken mit ad1

define leds byteport[2]
define taste ad[8]
define ausfall ad[1]
define n byte

#loop
  leds=&b10100100 : gosub pk
  leds=&b10110100 : gosub pk
  leds=&b10001100 : gosub pl
  leds=&b10010100 : gosub pk
  leds=&b10100100 : gosub pk
  leds=&b10100110 : gosub pk
  leds=&b10100001 : gosub pl
  leds=&b10100010 : gosub pk
goto loop

#pk
  n=n+1
  if taste>50 then goto lied
  if ausfall>100 then gosub gelb_blinken
  pause 2
  if n<100 then goto pk
  n=0
return

#pl
  n=n+1
  if taste>50 then goto lied
  if ausfall>100 then gosub gelb_blinken
  pause 6
  if n<100 then goto pl
  n=0
return

#gelb_blinken
  leds=&b11010010 : pause 30
  leds=&b00000000 : pause 30
goto gelb_blinken

#lied
  n=0 : beep 625,12,1
  leds=&b10110110 : pause 200
  leds=&b10100100 : pause 200 : leds=&b01100100
  beep 625,12,1 : beep 550,12,1 : beep 500,12,1
  beep 475,12,1 : beep 425,24,1 : beep 425,24,1
  beep 375,12,1 : beep 375,12,1 : beep 375,12,1
  beep 375,12,1 : beep 425,24,24
  beep 375,12,1 : beep 375,12,1 : beep 375,12,1
  beep 375,12,1 : beep 425,24,24
  beep 475,12,1 : beep 475,12,1 : beep 475,12,1
  beep 475,12,1 : beep 500,24,1 : beep 500,24,1
  beep 425,12,1 : beep 425,12,1 : beep 425,12,1
  beep 425,12,1 : beep 625,24,1
  n=0
goto loop
```

4. Auto

```
4.1 ' ccbasic\ubl\adtest1.bas
' test der sensoren des autos
' vorn ist beim auto die schrittmotorplatine

define lichtli ad[2] : define lichtre ad[3]
define temp ad[4]    : define ldr ad[5]

#loop
  print lichtli,
  print lichtre,
  print temp,
  print ldr
  pause 100
goto loop
```

4.1.1 Messwerte der AD - Wandler

```
Reflexlichtschranken: weißes Papier 135; Tisch 150
                      schwarze Pappe 220
LDR:                  Tageslicht mit Raumbelichtung 34
                      Schatten durch Hand 140
Temperaturfühler:    20 C 135
                      Berührung durch Finger 108
```

```
4.2 ' ccbasic\ubl\ah2.bas
' auto für hessentag; auto parkt
' version fuer schueler

' definitionen
define m byteport[2] ' Schrittmotoren
define lichtli ad[2] ' Lichtschranke links
define lichtre ad[3] ' Lichtschranke rechts
define ldr ad[5]     ' LDR
define zahl byte    ' Schrittzahl
define s byte       ' vorgewaehlte Schrittzahl
define n byte       ' für Zeitverzoeigerung

' hauptprogramm

m=&b00000000

gosub warten_ldr

zahl=0

s = 20 : gosub rueck : pause 50

gosub reabfrage      : pause 50

s = 12 : gosub rueck

m=&b00000000
end

' unterprogramme

#reabfrage 'abfrage der rechten lichtschranke
  gosub evor
  if lichtre<150 then goto reabfrage
return
```

```

#evor      'einzelsschritte
          m = &b10000001 : gosub w : m = &b01000010 : gosub w
          m = &b00100100 : gosub w : m = &b00011000 : gosub w
return

#erueck    'einzelsschritte
          m = &b00011000 : gosub w : m = &b00100100 : gosub w
          m = &b01000010 : gosub w : m = &b10000001 : gosub w
return

#warten_ldr 'abfrage des fotowiderstandes
          if ldr < 50 then goto warten_ldr
          beep 625,12,1
return

#vor
          m = &b10000001 : gosub w : m = &b01000010 : gosub w
          m = &b00100100 : gosub w : m = &b00011000 : gosub w
          zahl = zahl + 1 : if zahl<s then goto vor
          zahl=0
return

#rueck
          m = &b00011000 : gosub w : m = &b00100100 : gosub w
          m = &b01000010 : gosub w : m = &b10000001 : gosub w
          zahl = zahl + 1 : if zahl<s then goto rueck
          zahl=0
return

#links
          m = &b10001000 : gosub w : m = &b01000100 : gosub w
          m = &b00100010 : gosub w : m = &b00010001 : gosub w
          zahl = zahl + 1 : if zahl<s then goto links
          zahl=0
return

#rechts
          m = &b00010001 : gosub w : m = &b00100010 : gosub w
          m = &b01000100 : gosub w : m = &b10001000 : gosub w
          zahl = zahl + 1 : if zahl<s then goto rechts
          zahl=0
return

#w          ' warten
          n=0 : n=1 : n=0 : n=1 : n=0 : n=1 : n=0 : n=1 : n=0
          n=1 : n=0 : n=1 : n=0 : n=1 : n=0 : n=1 : n=0 : n=1
          n=1 : n=0 : n=1 : n=0 : n=1 : n=0 : n=1 : n=0 : n=1
return

```

4.3 Auto auf Kreuzung

```
' ccbasic\ubl\weg1s.bas version fuer schueler
' auto findet den richtigen weg

' definitionen
define m byteport[2]      ' Schrittmotoren
define lichtli ad[2]     ' Lichtschranke links
define lichtre ad[3]     ' Lichtschranke rechts
define ldr ad[5]         ' LDR
define zahl byte         ' Schrittzahl
define s byte            ' vorgewaehlte Schrittzahl
define n byte            ' fuer Zeitverzoegerung

' hauptprogramm
m=&b00000000
gosub warten_ldr
zahl=0
gosub reabfrage          : pause 50
gosub nach_rechts
m=&b00000000
end

' zusaetzliche unterprogramme

#nach_rechts
s=30 : gosub vor        : pause 50
s=16 : gosub rechts     : pause 50
s=20 : gosub vor        : pause 50
m=&b00000000
end
```

Die Lösung des Lehrers für die Aufgabenstellung 2

```
' ccbasic\ubl\weg1.bas
' auto findet den richtigen weg

' definitionen
define m byteport[2]      ' Schrittmotoren
define lichtli ad[2]      ' Lichtschranke links
define lichtre ad[3]      ' Lichtschranke rechts
define ldr ad[5]          ' LDR
define zahl byte          ' Schrittzahl
define s byte             ' vorgewaehlte Schrittzahl
define n byte             ' fuer Zeitverzoegerung

' hauptprogramm
m=&b00000000
gosub warten_ldr
zahl=0
gosub reabfrage           : pause 50
s=2 : gosub vor           : pause 50
if lichtli < 150 then gosub nach_links
if lichtli > 190 then gosub nach_rechts
m=&b00000000
end

' zusaetzliche unterprogramme

#nach_links
s=30 : gosub vor         : pause 50
s=16 : gosub links       : pause 50
s=20 : gosub vor         : pause 50
m=&b00000000
end

#liabfrage
gosub evor
if lichtli<150 then goto liabfrage
return
```

5. Messwerterfassung

5.1 vier Messwerte mit speichern im eeprom

```
'ccbasic\ubl cmessen_s.bas
'messung alle 5s mit mittelwertbildung
'für solarplatine

define wert byte      : define i byte      : define anzahl byte
define s1 port[16]    : define s2 port[15]
define s3 port[14]    : define s5 port[11]
define awert1 ad[1]   : define awert2 ad[4]
define awert3 ad[7]   : define awert4 ad[8]
define summe1 word    : define summe2 word
define summe3 word    : define summe4 word

'Programmbeginn
if s5 = on then gosub ausgeben

open# for write      'Datei ruecksetzen
close#

#loop1
  if s1 = on then anzahl = 12 else anzahl = 1 '12 -> 1 min; 1 -> 5 s
  if s3 = on then anzahl = 12*5             ' 5 min
  summe1=0 : summe2=0 : summe3=0 : summe4=0
  for i=1 to anzahl
    gosub messen
  next
  summe1=summe1/anzahl : summe2=summe2/anzahl
  summe3=summe3/anzahl : summe4=summe4/anzahl
  gosub speichern
goto loop1

#messen
  summe1=summe1+awert1 : summe2=summe2+awert2
  summe3=summe3+awert3 : summe4=summe4+awert4
  pause 250
return

#speichern
  open# for append
  print# summe1 : print# summe2
  print# summe3 : print# summe4
  close#
return

#ausgeben
  open# for read
  #naechster
  if eof then end
  input# wert : print wert,
  input# wert : print wert,
  input# wert : print wert,
input# wert : print wert
  pause 2
  goto naechster
```

```

'ccbasic\ubl cmessen_t.bas
'messung alle 5s mit mittelwertbildung
'für temperaturplatine

define wert byte      : define i byte      : define anzahl byte
define s1 port[16]   : define s2 port[15]
define s3 port[14]   : define s5 port[13]
define awert1 ad[1]  : define awert2 ad[2]
define awert3 ad[3]  : define awert4 ad[4]
define summe1 word   : define summe2 word
define summe3 word   : define summe4 word

'Programmbeginn
if s2 = on then gosub ausgeben

open# for write      'Datei ruecksetzen
close#

#loop1
  if s1 = on then anzahl = 12 else anzahl = 1 '12 -> 1 min; 1 -> 5 s
  if s3 = on then anzahl = 12*5             ' 5 min
  summe1=0 : summe2=0 : summe3=0 : summe4=0
  for i=1 to anzahl
    gosub messen
  next
  summe1=summe1/anzahl : summe2=summe2/anzahl
  summe3=summe3/anzahl : summe4=summe4/anzahl
  gosub speichern
goto loop1

#messen
  summe1=summe1+awert1 : summe2=summe2+awert2
  summe3=summe3+awert3 : summe4=summe4+awert4
  pause 250
return

#speichern
  open# for append
  print# summe1 : print# summe2
  print# summe3 : print# summe4
  close#
return

#ausgeben
  open# for read
  #naechster
  if eof then end
  input# wert : print wert,
  input# wert : print wert,
  input# wert : print wert,
input# wert : print wert
  pause 2
  goto naechster

```

5.2 Temperaturmessung mit Tabelle und LCD Anzeige

```
' mtemp1.BAS
' temperaturmessung mit tabelle und lcd-anzeige

define lcd_buf byte : define lcd_param byte
define output word : define value word
define temp word

define eingang1 ad[2] : define eingang2 ad[3]
define eingang3 ad[4]

define lcd_port byteport[2] : define lcd_rs port[14]
define lcd_rw port[13] : define lcd_e port[15]

define T_ &H54 : define LEERZ &H20 : define NULL &H30
define MINUS &H2D : define GLEICH &H3D
define Punkt 46

' Hauptprogramm
gosub LCD_INIT
#loop
  looktab temptab, eingang1, temp
  gosub ausgabe1
  looktab temptab, eingang2, temp
  gosub ausgabe2
  looktab temptab, eingang3, temp
  gosub ausgabe3
  pause 10
goto loop
end

#ausgabe1
  value = temp
  lcd_param = 1 : gosub LCD_GOTOLINE
  lcd_param = T_ : gosub LCD_WRITECHAR
  lcd_param = 49 : gosub LCD_WRITECHAR
  gosub WRITEVALUE
  lcd_param = LEERZ : gosub LCD_WRITECHAR
return

#ausgabe2
  value = temp
  lcd_param = T_ : gosub LCD_WRITECHAR
  lcd_param = 50 : gosub LCD_WRITECHAR
  gosub WRITEVALUE
return

#ausgabe3
  value = temp
  lcd_param = 2 : gosub LCD_GOTOLINE
  lcd_param = T_ : gosub LCD_WRITECHAR
  lcd_param = 51 : gosub LCD_WRITECHAR
  gosub WRITEVALUE
return

#WRITEVALUE
  output = abs(value)
  ' negatives Vorzeichen oder Blank
  if value < 0 then lcd_param = MINUS else lcd_param = LEERZ
  gosub LCD_WRITECHAR

  ' Hunderter-Dezimalstelle oder Leerzeichen
  if abs(value)>=100 then lcd_param=NULL+output/100 else lcd_param=LEERZ
  gosub LCD_WRITECHAR
  output = output mod 100
```

```

' Zehner-Dezimalstelle oder Leerzeichen
if abs(value)>=10 then lcd_param=NULL+output/10 else lcd_param=LEERZ
gosub LCD_WRITECHAR

' Dezimalpunkt
lcd_param = Punkt          : gosub LCD_WRITECHAR

' einer Dezimalstelle oder Leerzeichen
lcd_param = NULL + output mod 10
gosub LCD_WRITECHAR
return

' LCD_Interface
'( muss in jedes Programm mit LCD-Ausgabe eingefuegt werden)

#LCD_INIT

' alle ports 0
lcd_port = OFF

' 8-Bit-Modus aktivieren
lcd_param=&H38 : gosub LCD_WRITECMD

' mit 8-Bit-Command in 4-Bit-Modus umschalten
lcd_port=&B00000010
tog lcd_e
tog lcd_e

' ab jetzt 4-Bit-Modus
lcd_param = &H28 : gosub LCD_WRITECMD
lcd_param = &H0C : gosub LCD_WRITECMD

' Display loeschen
#LCD_CLS
lcd_param = &H02 : gosub LCD_WRITECMD
lcd_param = &H01 : gosub LCD_WRITECMD
return

' Zeilenwechsel
#LCD_GOTOLINE
if lcd_param = 1 then lcd_param = &H80
if lcd_param = 2 then lcd_param = &HC0
goto LCD_WRITECMD

' LCD-Kommando
#LCD_WRITECMD
lcd_buf = OFF
goto LCD_WRITE

' Zeichenausgabe
#LCD_WRITECHAR
lcd_buf = &B00100000

' Kommando oder Zeichen an Display senden
#LCD_WRITE
lcd_port = lcd_buf or (lcd_param shr 4) ' Hi-Nibble
tog lcd_e
tog lcd_e
lcd_port = lcd_buf or (lcd_param and &H0F) ' Lo-Nibble
tog lcd_e
tog lcd_e
return
end
table temptab
600 600 600 600 600 600 600 600 600 600 600 600 600 600 600 600
600 600 600 600 600 600 600 600 600 600 600 600 600 600 600 600

```

```
600 600 600 600 600 600 600 600 600 600 600 600 600 600 600 600
600 600 600 600 600 600 600 600 600 600 600 600 600 593 587 581 557
568 562 556 550 543 537 531 525 518 512 506 500 494 489 484 478
473 468 463 457 452 447 442 436 431 426 421 415 410 405 400 395
390 386 381 377 372 368 363 359 354 350 345 340 336 331 327 322
318 313 309 304 300 295 291 287 283 279 275 270 266 262 258 254
250 245 241 237 233 229 225 220 216 212 208 204 200 195 101 187
183 179 175 170 166 162 158 154 150 145 141 137 133 129 125 120
116 112 108 104 100 95 91 86 82 78 73 69 65 60 56 52
47 43 39 34 30 26 21 17 13 8 4 0 -6 -11 -16 -22
-27 -32 -37 -43 -48 -53 -58 -64
-69 -74 -79 -85 -90 -95 -100 -107
-113 -119 -125 -132 -138 -144 -150 -157
-163 -169 -175 -182 -188 -194 -200 -200
-200 -200 -200 -200 -200 -200 -200 -200
-200 -200 -200 -200 -200 -200 -200 -200
-200 -200 -200 -200 -200 -200 -200 -200
```

tabend

5.3 Energie auf das Schuldach; mit LCD Anzeige

```
' msolc3.BAS
' P pro m^2 und P des Schuldaches

define output word : define value word

define lcd_buf byte : define lcd_param byte
define awert ad[1]
define lcd_port byteport[2] : define lcd_rs port[14]
define lcd_rw port[13] : define lcd_e port[15]

define D_ 68 : define P_ 80 : define W_ 87 : define LEERZ 32
define NULL 48 : define GLEICH 61 : define m 109 : define hoch 94
define zwei 50 : define a 97 : define c 99 : define h 104
define k 107

' Hauptprogramm
gosub LCD_INIT
#loop
    gosub ausgabe1
    gosub ausgabe2
    pause 2
goto loop
end

#ausgabe1
value = (awert*105)/20
lcd_param = 1 : gosub LCD_GOTOLINE
lcd_param = m : gosub LCD_WRITECHAR
lcd_param = hoch : gosub LCD_WRITECHAR
lcd_param = zwei : gosub LCD_WRITECHAR
lcd_param = LEERZ : gosub LCD_WRITECHAR
lcd_param = LEERZ : gosub LCD_WRITECHAR
lcd_param = P_ : gosub LCD_WRITECHAR
lcd_param = GLEICH : gosub LCD_WRITECHAR
gosub WRITEVALUE
lcd_param = LEERZ : gosub LCD_WRITECHAR
lcd_param = LEERZ : gosub LCD_WRITECHAR
lcd_param = W_ : gosub LCD_WRITECHAR
return

#ausgabe2
value = ((awert*105)/20)*27)/10
lcd_param = 2 : gosub LCD_GOTOLINE
lcd_param = D_ : gosub LCD_WRITECHAR
lcd_param = a : gosub LCD_WRITECHAR
lcd_param = c : gosub LCD_WRITECHAR
lcd_param = h : gosub LCD_WRITECHAR
lcd_param = LEERZ : gosub LCD_WRITECHAR
lcd_param = P_ : gosub LCD_WRITECHAR
lcd_param = GLEICH : gosub LCD_WRITECHAR
gosub WRITEVALUE
lcd_param = LEERZ : gosub LCD_WRITECHAR
lcd_param = k : gosub LCD_WRITECHAR
lcd_param = W_ : gosub LCD_WRITECHAR
return

#WRITEVALUE
output = abs(value)

' Tausender-Dezimalstelle oder Leerzeichen
if abs(value)>=1000 then lcd_param=NULL+output/1000 else lcd_param=LEERZ
gosub LCD_WRITECHAR
output = output mod 1000
```

```
' Hunderter-Dezimalstelle oder Leerzeichen
if abs(value)>=100 then lcd_param=NULL+output/100 else lcd_param=LEERZ
gosub LCD_WRITECHAR
output = output mod 100

' Zehner-Dezimalstelle oder Leerzeichen
if abs(value)>=10 then lcd_param=NULL+output/10 else lcd_param=LEERZ
gosub LCD_WRITECHAR

' einer Dezimalstelle oder Leerzeichen
lcd_param = NULL + output mod 10
gosub LCD_WRITECHAR
return
```

LCD Interface siehe 5.2

6. Messwerte von ccbasic in einer excel-Tabelle eichen

- Messwerte von Hyperterminal in Word-Pad kopieren
- excel Datei öffnen; alle Dateien; Word-Pad Datei mit excel öffnen; Messwerte werden in Spalten angezeigt
- in eichen1 Spalten F bis L in Zwischenspeicher kopieren
- Spalten F bis L aus dem Zwischenspeicher in die Datei mit den Messwerten einfügen
- Erklärung der Funktion sverweis

=sverweis(A1;\$F\$1:\$G\$248;2)

A1 erster Messwert

\$F\$1:\$G\$248 Tabelle in der die Zuordnung Zahl --- Temperatur steht

2 zweite Spalte der Zuordnungstabelle