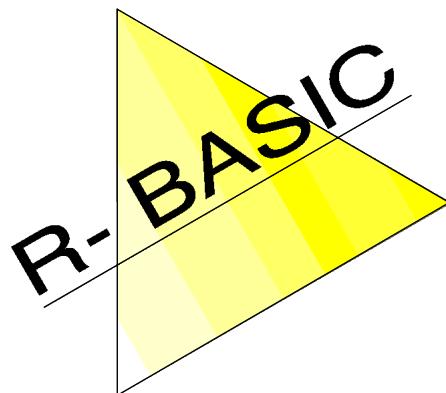


R-BASIC

Einfach unter PC/GEOS programmieren



R-BASIC Anhänge

Version 1.0

(Leerseite)

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Inhaltsverzeichnis

| | |
|--|----|
| A: ASCII -Steuercodes | 4 |
| B: Liste der Fehlercodes | 6 |
| C: R-BASIC-interne Struktur-Typen | 8 |
| D: Mitgelieferte Blockgrafik Zeichensätze | 18 |
| E: Numerische Werte der UI-Instance Variablen | 19 |
| F: Definitionsbereich der mathematischen Funktionen | 21 |
| G: Konstanten zur Musikwiedergabe | 24 |
| H: Tabellen und Übersichten zur KC-85 Kompatibilität | 27 |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

(Leerseite)

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Anhang A: ASCII-Steuercodes

ASCII-Codes im Bereich von Null bis 31 ist kein Buchstabe, sondern eine Sonderfunktion zugeordnet.

- Die Steuercodes werden mit PRINT CHR\$(code) ausgegeben. Sie funktionieren vollständig nur wenn der Screen ein BitmapContent-Objekt ist. Für alle anderen Fälle müssen Sie ausprobieren, ob es geht.
- Im GEOS-Font-Modus existieren keine festen Zeichen-Positionen, da die meisten Buchstaben verschieden breit sind. Die Steuerzeichen arbeiten daher mit einer "durchschnittlichen" Zeichenbreite. Ob die Funktion für Ihre Zwecke brauchbar ist, müssen Sie ausprobieren.
Grundsätzlich NICHT korrekt arbeiten Codes, die einzelne Zeichen löschen oder einfügen (ASC_CLEAR, ASC_BACKSPACE, ASC_DEL, ASC_INS).
- Um die Konstanten-Namen verwenden zu können, müssen Sie die KeyCode-Library einbinden, die Teil von R-BASIC ist:

```
Include "KeyCode"           ! Groß-/Kleinschreibung beachten
```

- Textobjekte (Memo, InputLine) unterstützen nur die Steuercodes ASC_TAB (Tabulator), ASC_ENTER (Zeilenumbruch) und 31 (Weicher Bindestrich, nur am Zeilenende sichtbar). Sie können mit Hilfe eines Backslash (\t für Tab, \r für Zeilenumbruch, \031 für weichen Bindestrich) in Texte eingefügt werden.
Alle anderen Codes werden als Punkt dargestellt. Der Code 26 führt zum Crash.

Tabelle: ASCII Steuercodes

InKey\$-Taste: Gedrückte Taste, damit InKey\$ diesen Code zurückgibt.

Bedeutung in Print: Funktion, die mit Print Chr\$(code) ausgeführt wird.

| ASCII-Code dez. hex. | Konstanten-Name (KeyCode-Library) | InKey\$ Taste | Bedeutung in Print |
|-------------------------|--------------------------------------|------------------|--|
| 01 (&H01) | ASC_CLEAR | - | Zeichen löschen, Cursor an nach links |
| 02 (&H02) | ASC_CLEAR_LINE | - | Zeile löschen, Cursor an Zeilenanfang |
| 03 (&H03) | - | - | - |
| 04 (&H04) | - | - | - |
| 05 (&H05) | - | - | - |
| 06 (&H06) | - | - | - |
| 07 (&H07) | ASC_BEEP | - | Signalton (Beep) |
| 08 (&H08) | ASC_BACKSPACE | Backsp. | Rückwärtsschritt. Zeichen auf Cursorposition löschen und Cursor nach links |
| 09 (&H09) | ASC_TAB | TAB | Anwahl der nächsten Tabulatorposition |
| 10 (&H0A) | ASC_DOWN | ↓ | Cursor 1 Zeile tiefer |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

| | | | |
|-----------|-----------------|--------|--|
| 11 (&H0B) | ASC_UP | ↑ | Cursor 1 Zeile höher |
| 12 (&H0C) | ASC_CLS | - | Bildschirm löschen |
| 13 (&H0D) | ASC_ENTER | Enter | Cursor an den Anfang der nächsten Zeile |
| 14 (&H0E) | ASC_LEFT | ← | Cursor 1 Zeichen nach links |
| 15 (&H0F) | ASC_RIGHT | → | Cursor 1 Zeichen nach rechts |
| 16 (&H10) | ASC_HOME | - | Cursor nach links oben |
| 17 (&H11) | ASC_PAGE_MODE | | PAGE-Mode einstellen |
| 18 (&H12) | ASC_PAGE_UP | Bild ↑ | |
| | ASC_SCROLL_MODE | | SCROLL-Mode einstellen |
| 19 (&H13) | ASC_PAGE_DOWN | Bild ↓ | |
| | ASC_LAYOUT_MODE | - | LAYOUT-Mode einstellen |
| 20 (&H14) | ASC_POS_END | Ende | - |
| 21 (&H15) | ASC_POS_1 | Pos 1 | Cursor an Zeilenanfang |
| 22 (&H16) | ASC_INS | Einfg | ein Zeichen an der Cursorposition einfügen |
| 23 (&H17) | ASC_DEL | Entf | ein Zeichen an der Cursorposition löschen |
| 24 (&H18) | - | - | - |
| 25 (&H19) | - | - | - |
| 26 (&H1A) | - | - | - |
| 27 (&H1B) | ASC_ESC | ESC | - |
| 28 (&H1C) | - | - | - |
| 29 (&H1D) | - | - | - |
| 30 (&H1E) | - | - | - |
| 31 (&H1F) | - | - | - |

SCROLL-Mode

Am Zeilenende wird eine neue Zeile eröffnet. Wenn das untere Ende des Text-Fensters erreicht wird, schiebt R-BASIC den Fensterinhalt um eine Zeile nach oben. Am unteren Fensterrand entsteht eine Leerzeile. Das funktioniert nur für BitmapContent Objekte.

PAGE-Mode

Am Zeilenende wird eine neue Zeile eröffnet. Wenn das untere Ende des Text-Fensters erreicht wird setzt R-BASIC den Cursor wieder nach links oben.

LAYOUT-Mode

Keinerlei Cursor-Restriktionen. Am Ende einer Zeile erfolgt kein Umbruch, der Text wird über das Fenster hinaus geschrieben. Negative Cursor-Koordinaten und Koordinaten außerhalb des Textfensters sind erlaubt.

Anhang B: Liste der Fehlercodes

Positive Werte sind Codes, die durch GEOS (bzw. DOS) erzeugt werden, negative Werte sind R-BASIC interne Codes. Siehe auch **fileError** und **ErrorText\$** (Handbuch "Spezielle Themen", Kapitel 6.2). Die GEOS- (bzw. DOS) Fehlercodes wurde 1:1 dem PC/GEOS-SDK entnommen. Es kann nicht ausgeschlossen werden, dass weitere Fehlercodes existieren.

| | |
|----------------------------------|-------|
| ERROR_CANCELED_BY_USER | (-1) |
| ERROR_RO_SKIPPED | (-2) |
| ERROR_CANNOT_COPY_FOLDERS | (-3) |
| ERROR_FILE_CANNOT_REPLACE_FOLDER | (-4) |
| ERROR_CANNOT_DELETE_FOLDERS | (-5) |
| ERROR_CANNOT_COPY_MOVE_LINK | (-6) |
| | |
| ERROR_FILE_NOT_OPEN | (-10) |
| ERROR_LINE_TO_LONG | (-11) |
| ERROR_FONT_SIZE_MISMATCH | (-12) |
| ERROR_INVALID_FONT_FILE | (-13) |
| ERROR_FONT_SIZE_MISMATCH | (-14) |
| ERROR_NO_PICTURE_LIST | (-15) |
| ERROR_CAPTION_TOO_BIG | (-16) |
| ERROR_INVALID_IMAGE | (-17) |
| ERROR_INVALID_WAV_FILE | (-18) |
| ERROR_WAV_FORMAT_NOT_SUPPORTED | (-19) |
| ERROR_WAV_OUTPUT_CANCELED | (-20) |
| ERROR_TEXT_TOO_LARGE | (-21) |
| ERROR_TEXT_TOO_SHORT | (-22) |
| | |
| ERROR_UNSUPPORTED_FUNCTION | 1 |
| ERROR_FILE_NOT_FOUND | 2 |
| ERROR_PATH_NOT_FOUND | 3 |
| ERROR_TOO_MANY_OPEN_FILES | 4 |
| ERROR_ACCESS_DENIED | 5 |
| ERROR_INSUFFICIENT_MEMORY | 8 |
| ERROR_INVALID_VOLUME | 15 |
| ERROR_IS_CURRENT_DIRECTORY | 16 |
| ERROR_DIFFERENT_DEVICE | 17 |
| ERROR_NO_MORE_FILES | 18 |
| ERROR_WRITE_PROTECTED | 19 |
| ERROR_UNKNOWN_VOLUME | 20 |
| ERROR_DRIVE_NOT_READY | 21 |
| ERROR_CRC_ERROR | 23 |
| ERROR_SEEK_ERROR | 25 |
| ERROR_UNKNOWN_MEDIA | 26 |
| ERROR_SECTOR_NOT_FOUND | 27 |
| ERROR_WRITEFAULT | 29 |
| ERROR_READFAULT | 30 |
| ERROR_GENERAL_FAILURE | 31 |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

| | |
|---------------------------------------|-----|
| ERROR_SHARING_VIOLATION | 32 |
| ERROR_ALREADY_LOCKED | 33 |
| ERROR_SHARING_OVERFLOW | 36 |
| ERROR_SHORT_READ_WRITE | 128 |
| ERROR_INVALID_LONGNAME | 129 |
| ERROR_FILE_EXISTS | 130 |
| ERROR_DOS_EXEC_IN_PROGRESS | 131 |
| ERROR_FILE_IN_USE | 132 |
| ERROR_ARGS_TOO_LONG | 133 |
| ERROR_DISK_UNAVAILABLE | 134 |
| ERROR_DISK_STALE | 135 |
| ERROR_FILE_FORMAT_MISMATCH | 136 |
| ERROR_CANNOT_MAP_NAME | 137 |
| ERROR_DIRECTORY_NOT_EMPTY | 138 |
| ERROR_ATTR_NOT_SUPPORTED | 139 |
| ERROR_ATTR_NOT_FOUND | 140 |
| ERROR_ATTR_SIZE_MISMATCH | 141 |
| ERROR_ATTR_CANNOT_BE_SET | 142 |
| ERROR_CANNOT_MOVE_DIRECTORY | 143 |
| ERROR_PATH_TOO_LONG | 144 |
| ERROR_ARGS_INVALID | 145 |
| ERROR_CANNOT_FIND_COMMAND_INTERPRETER | 146 |
| ERROR_NO_TASK_DRIVER_LOADED | 147 |
| | |
| VM_FILE_EXISTS | 263 |
| VM_FILE_NOT_FOUND | 264 |
| VM_SHARING_DENIED | 265 |
| VM_OPEN_INVALID_VM_FILE | 266 |
| VM_CANNOT_CREATE | 267 |
| VM_TRUNCATE_FAILED | 268 |
| VM_WRITE_PROTECTED | 269 |
| VM_CANNOT_OPEN_SHARED_MULTIPLE | 270 |
| VM_FILE_FORMAT_MISMATCH | 271 |
| | |
| VM_UPDATE NOTHING_DIRTY | 272 |
| VM_UPDATE_INSUFFICIENT_DISK_SPACE | 273 |
| VM_UPDATE_BLOCK_WAS_LOCKED | 273 |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Anhang C: R-BASIC interne Struktur-Typen

DateAndTime

Siehe auch: Programmierhandbuch, Vol.2, Kapitel 2.11.2

Alle Zeitfunktionen, einschließlich der Funktionen zum Zugriff auf das Datei-Datum benutzen eine Struktur, die **DateAndTime** heißt.

```
STRUCT DateAndTime
    year      AS WORD      ' Jahr          (z.B. 2014)
    month     AS WORD      ' Monat         (1...12)
    day       AS WORD      ' Tag           (1 ... 31)
    hour      AS WORD      ' Stunde        (0 ... 23)
    minute    AS WORD      ' Minute        (0 ... 59)
    second    AS WORD      ' Sekunde       (0 ... 59)
END STRUCT
```

GeodeToken

Siehe auch: Spezielle Themen, Vol.2, Kapitel 9.2

Token und **Creator** einer Datei werden in einer Struktur gespeichert, die **GeodeToken** heißt und folgendermaßen definiert ist. Das Bild dazu befindet sich in der TokenDB-Datei.

```
STRUCT GeodeToken
    manufid   AS WORD      ' Manufacturer ID
                           '(Hersteller-Identifikation)
    tokenChars AS STRING(4)
END STRUCT
```

PrintFontStruct

Siehe auch: Spezielle Themen, Vol.2, Kapitel 2.7

Die Systemvariable **printFont** ist folgendermaßen definiert:

```
STRUCT PrintFontStruct
    type            as word
    fontID          as word
    fontSize        as word
    charWidth      as word
    lineHeight      as word
    style           as word      ' öffentlich
    base            as word
END STRCUT

DIM printFont AS PrintFontStruct
```

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Bedeutung der einzelnen Felder

| | |
|-------------------|---|
| type | Speichert den aktuell von PRINT verwendeten Font-Typ. Gültige Werte sind FT_FIXED (0, gesetzt von FontSetFixed), FT_GEOS (1, gesetzt von FontSetGeos) und FT_BLOCK (2, gesetzt von FontSetBlock). |
| fontID | Die GEOS-Font-ID für FT_FIXED und FT_GEOS. Bei ungültigen Werten wählt GEOS einen Ersatzfont, häufig die BISON-Schrift. |
| fontSize | Die Größe der Schrift. Bitmap-Schriften (z.B. FID_BISON) unterstützen nicht alle Größen. |
| charWidth | Breite eines Zeichens. Für type = FT_FIXED gilt: Wert in Pixeln Für type = FT_GEOS gilt: Wert in % von printFont.fontSize Für type = FT_BLOCK gilt: Wert in Pixeln |
| lineHeight | Zeilenabstand, in Pixeln |
| style | Text-Stil für FT_GEOS und FT_FIXED: Kombination von TS_xxx-Werten. Siehe unten. |
| base | Abstand der Text-Grundline vom oberen Rand der Zeile |

Tabelle: Text-Stile zur Benutzung mit **printFont.style**

F: Verfügbar im Fixed-Font Modus

G: Verfügbar im GEOS-Font-Modus

Im BlockFont Modus steht ausschließlich das Flag

TS_DONT_EXEC_CONTROLS (Wert: 32768 = &H8000) zur Verfügung, dass es erlaubt, ASCII-Codes unter 32 (Steuerzeichen) auszudrucken, statt sie "auszuführen".

| Text-Stil | Wert | Modus | Bedeutung |
|----------------|------|-------|--|
| TS_UNDERLINE | 1 | G, F | unterstrichene Schrift |
| TS_STRIKE_THRU | 2 | G | durchgestrichene Schrift |
| TS_SUBSCRIPT | 4 | G | tiefgestellte Schrift |
| TS_SUPERSCRIPT | 8 | G | hochgestellte Schrift |
| TS_ITALIC | 16 | G | kursive Schrift |
| TS_BOLD | 32 | G, F | fette Schrift |
| TS_OUTLINE | 64 | G | Wenn der Font sowohl Bitmap- und als auch Outline-Schrift enthält: Verwendung der Outline Schrift erzwingen |
| TS_CENTER | 256 | F | Buchstaben einzeln zentrieren (langsamer) ^(A) Standard bei einigen FID_-Werten im Fixed-Font-Modus |
| | 512 | G, F | Reserviert, intern verwendet Nicht benutzen! |
| TS_ERRORLINE | 1024 | F | Unterstrichen mit rot gepunkteter Linie ^(A) |

^(A) Wird von R-BASIC realisiert, keine System-Funktion

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

NumberFormatStruct

Siehe auch: Spezielle Themen, Vol.1, Kapitel 1.2

Die globale Variable `numberFormat` ist vom Typ `NumberFormatStruct`, der folgendermaßen definiert ist.

```
STRUCT NumberFormatStruct
    minDigits, maxDigits, digitMode      AS Integer
    highLimit, lowLimit                 AS Integer
    plusSign                           AS Integer
    exponentMode                      AS Integer
    preChars, addChars                 AS String(7)
    formatFlags                        AS Word
END STRUCT

DIM numberFormat AS NumberFormatStruct
```

| Feld | Funktion, Erlaubte Werte |
|------------------------|---|
| minDigits maxDigits | Einstellen der Stellenzahl 0 .. 15 |
| digitMode | Einstellen, welche Stellen für minDigits & maxDigits zählen DM_ALL_DIGITS, DM_FRAC_DIGITS, DM_VALID_DIGITS |
| highLimit lowLimit | Grenze für die Umschaltung in die 10er-Potenz Schreibweise Maximale Anzahl der Vorkommastellen (highLimit) oder der führenden Nullen (lowLimit) 0 .. 15 |
| plusSign | Positives Vorzeichen schreiben PS_NONE , PS_SPACE, PS_ALWAYS |
| exponentMode | Darstellung des Exponenten EXP_NORMAL, EXP_FORCE, EXP_SCI (Exponent in 3er- Schritten), EXP_FORCE + EXP_SCI |
| preChars addChars | vor- oder nachgestellte Zeichen Zeichenkette, maximal 7 Zeichen |
| formatFlags | Eine Kombination der folgenden Bits FF_PRINT_ADD_NO_SPACE Print hängt kein Leer- zeichen an FF_NO_EXP_LOW keine 10er-Potenz Schreibweise für Zahlen zwischen -1 und +1 FF_NO_EXP_HIGH keine 10er-Potenz Schreibweise für große Zahlen |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

GraphicDrawStruct

Siehe auch: Programmierhandbuch, Vol.3, Kapitel 2.8.4

Die **graphic** - Systemvariable ist vom Typ **GraphicDrawStruct**, der folgendermaßen definiert ist:

```
STRUCT GraphicDrawStruct
    mixMode          AS Word
    backColor        AS DWord
    lineColor        AS DWord
    lineDrawMask     AS Word
    lineWidth, lineStyle AS Word
    lineEnd, lineJoin AS Word
    areaColor        AS DWord
    areaDrawMask     AS Word
    textColor         AS DWord
    textDrawMask     AS Word
    drawFlags        AS Word
    reserve(6)       AS Word
        ! reserviert für zukünftige Erweiterungen
END STRUCT

DIM graphic AS GraphicDrawStruct
```

| Felder | Bedeutung |
|------------------------------------|--|
| areaColor, lineColor, textColor | Farbe für Flächen (areaColor), Linien (lineColor und Texte (textColor). Die Befehle COLOR in INK setzen all drei Farben. |
| backColor | Hintergrundfarbe für Texte. Wird von den Befehlen PAPER und COLOR belegt. Spezialwert 4096 (= BG_TRANSPARENT : Texte oder Blockgrafikzeichen werden mit transparentem Hintergrund ausgegeben). |
| mixMode | Schreibmodus für Flächen und Linien. Wirkt nicht auf Texte. Beschreibt, auf welche Weise neue Grafikausgaben mit bereits vorhandenen verknüpft werden. Siehe Tabelle unten. Standard MM_COPY , häufig verwendet: MM_XOR , MM_INVERT . |
| lineWidth | Liniendicke, in Pixeln |
| lineStyle | Linienstil, z.B. gestrichelt. Siehe Tabelle unten. |
| areaDrawMask | Füllmuster für Flächen. Die Fläche wird mit einem Muster hinterlegt, das von GEOS erzeugt wird. R-BASIC definiert einige Konstanten zur Arbeit mit Füllmustern. Siehe Tabelle unten. |
| lineDrawMask | Füllmuster für Linien. Details siehe areaDrawMask. |
| textDrawMask | Füllmuster für Text. Details siehe areaDrawMask. Selten verwendet, da Texte i.a. sehr klein sind. |
| lineEnd | Linienabschluss. Erlaubte Werte: 0: Normales Ende, 1: Halbrund, 2: Quadrat |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

| | |
|-----------|---|
| | Die Werte 1 und 2 verlängern die Linie etwas. |
| lineJoin | Verbindung zwischen Linien bei einer Figur (Rechteck). Erlaubte Werte: 0: Normal (eckig), 1: abgerundet, 2: abgeflacht. |
| drawFlags | aktuell nur GDF_SCALE_PSET |

GraphicDrawStruct - Linienstile

| Wert | Konstante | Bedeutung |
|------|------------|--------------------|
| 0 | LS_SOLID | durchgehend |
| 1 | LS_DASHED | gestrichelt |
| 2 | LS_DOTTED | gepunktet |
| 3 | LS_DASHDOT | Strich-Punkt |
| 4 | LS_DASDDOT | Strich-Doppelpunkt |

GraphicDrawStruct - Füllmuster (siehe auch nächste Seite)

| Wert | Konstante | Bedeutung |
|----------|------------|---|
| 0 - 24 | - | Von GEOS bereitgestellte Muster |
| 25 | DM_100 | "Normalzustand", 100% Deckung. |
| 26 - 88 | - | Unterschiedliche "Transparenzgrade". Größere Werte entsprechen höherer Transparenz. |
| 89 | DM_0 | Null % Deckung, vollständig transparent. |
| 90 - 127 | - | weitere Muster |
| 128 | DM_INVERSE | Wird zu einem der anderen Werte addiert. Das Muster wird invertiert. |

GraphicDrawStruct - Mix-Modes

| Wert | Konstante | Bedeutung |
|------|-----------|---|
| 0 | MM_CLEAR | Das Zeichnen einer Grafik löscht den überschriebenen Bereich. Sinnvoll beim Löschen von Bitmap-Masken (Transparenz setzen). |
| 1 | MM_COPY | Standardwert. Die neue Grafik überschreibt vorhandene. |
| 2 | MM_NOP | Die Grafikausgabe wird ignoriert. |
| 3 | MM_AND | Die Farben in dem überschriebenen Bereich werden logisch AND mit der Zeichenfarbe verknüpft. |
| 4 | MM_INVERT | Die Farben in dem überschriebenen Bereich werden logisch invertiert. Die Zeichenfarbe spielt |

R-BASIC Handbuch - Anhänge

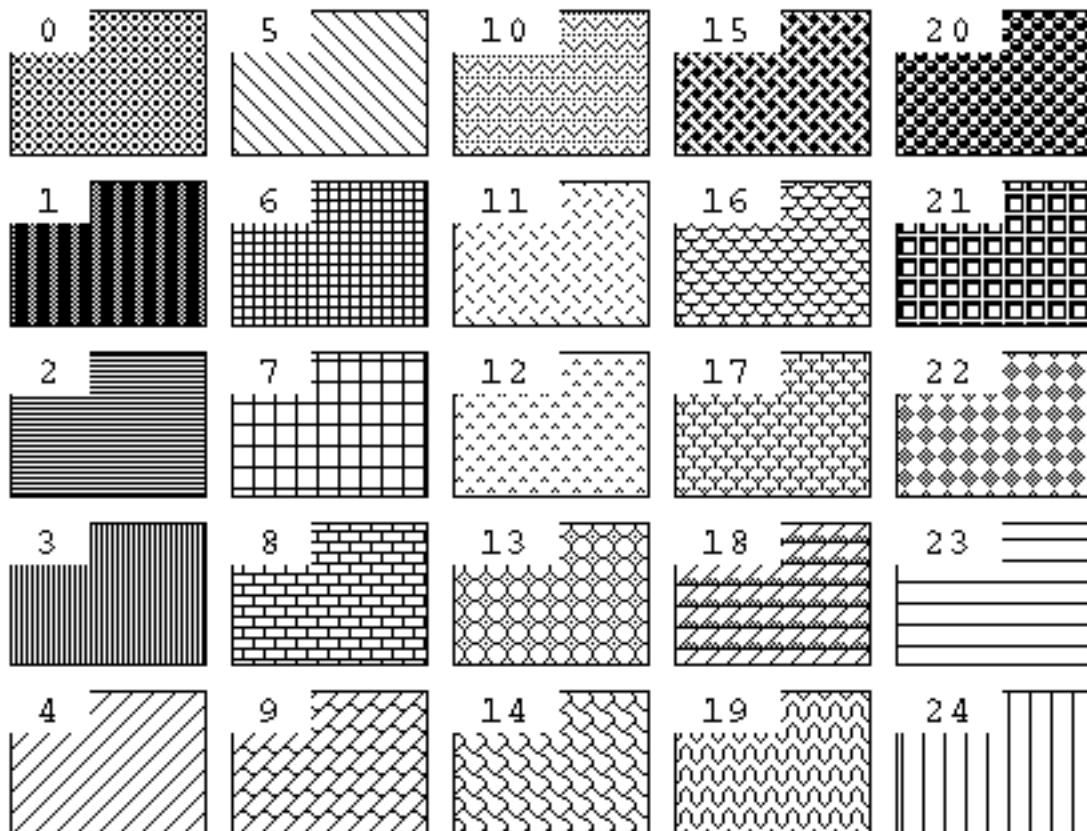
Einfach unter PC/GEOS programmieren

| | | |
|---|--------|---|
| | | keine Rolle. Zweimaliges Zeichnen stellt den Ausgangszustand wieder her. |
| 5 | MM_XOR | Die Farben in dem überschriebenen Bereich werden logisch XOR mit der Zeichenfarbe verknüpft. Zweimaliges Zeichnen stellt den Ausgangszustand wieder her. Schwarz (Farbwert Null) erzeugt keine Ausgabe. |
| 6 | MM_SET | Das Zeichnen einer Grafik setzt den überschriebenen Bereich auf schwarz. Sinnvoll beim Arbeiten mit Bitmap-Masken. |
| 7 | MM_OR | Die Farben in dem überschriebenen Bereich werden logisch OR mit der Zeichenfarbe verknüpft. |

GraphicDrawStruct - DrawFlags

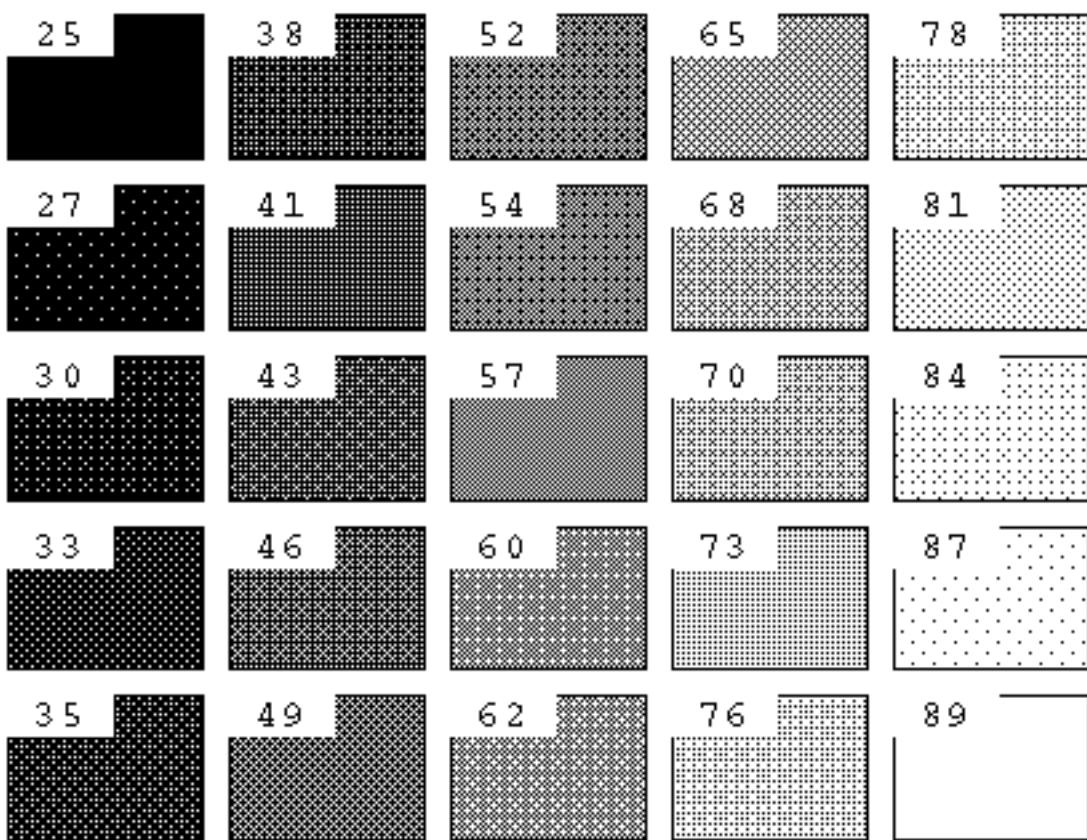
| Wert | Konstante | Bedeutung |
|------|----------------|---|
| 1 | GDF_SCALE_PSET | Die von PSet und PReset gesetzten Punkte werden als Flächen gezeichnet und so vergrößert, wenn der Screen skaliert ist. |

GraphicDrawStruct - Beispiele für Füllmuster



R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren



TransMatrix

Siehe auch: Objekt Handbuch, Vol.1, Kapitel 2.3.5

Die Struktur **TransMatrix** enthält die 6 variablen Elemente der Transformationsmatrix und ist folgendermaßen definiert:

```
STRUCT TransMatrix
    a1, b1, c1      AS REAL
    a2, b2, c2      AS REAL
END STRUCT
```

Jedes Mal, wenn GEOS einen einzelnen Punkt mit den Koordinaten (x; y) auf dem Bildschirm darstellt, wendet es die aktuell gültige Transformationsmatrix auf diesen Punkt an, um die Bildschirmkoordinaten (x'; y') zu erhalten:

$$\begin{aligned}x' &= a1 * x + b1 * y + c1 \\y' &= b1 * x + b2 * y + c2\end{aligned}$$

Die direkte Arbeit mit Transformationsmatrizen ist etwas für erfahrene und mathematisch versierte Programmierer. Verwenden Sie die Funktion **ScreenGetTransMatrix()** um die aktuelle Transformationsmatrix zu lesen und **ScreenSetTransMatrix** um eine Transformationsmatrix anzuwenden.

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

PaletteEntry, FullPalette

Siehe auch: Objekt Handbuch, Vol.9, Kapitel 5.2.5.1

Bitmaps der Farbtiefe 8 Bit können nur 256 verschiedenen Farben darstellen. Welche der 16 Millionen möglichen Farben das sind wird über eine Tabelle bestimmt, die Palette heißt. Jeder Paletteneintrag enthält einen Farbwert für die Grundfarben rot, grün und blau. Die vollständige Palette besteht aus 256 solcher Einträge. Die Zählung beginnt dabei mit Null. Der Zugriff auf die Palette ist beim BitmapContent Objekt ausführlich beschrieben.

```
STRUCT PaletteEntry
    rt, gn, bl as BYTE
End Struct

STRUCT FullPalette
    item(255) as PaletteEntry
END Struct
```

AnyStruct

Siehe auch: Programmierhandbuch, Vol.1, Kapitel 2.2.8.8

In sehr seltenen Spezialfällen sind Routinen sinnvoll, denen eine Struktur beliebigen Typs übergeben werden soll oder die eine Struktur beliebigen Typs zurückgeben sollen. Dafür dient der Typ AnyStruct. Variablen oder Funktionen von diesem Typ sind zuweisungskompatibel zu jedem beliebigen Strukturtyp.

```
STRUCT AnyStruct
    any_struct_dummy_byte_array_ (3499) as BYTE
End struct
```

Hinweis: Eine Variable oder ein Parameter des Typs AnyStruct belegt 3500 Byte im Variablerspeicher.

GraphicInfo

Siehe auch: Objekt Handbuch, Vol.8, Kapitel 4.17.2

Die Routinen GetImageInfo, GetPictureInfo, GetGStringInfo und GetBitmapInfo sowie die Instancevariable imgInfo von Image-Objekten liefern Informationen über eine Grafik oder eine Grafikdatei in Form der folgenden Struktur:

```
STRUCT GraphicInfo
    sizeX as WORD
    sizeY as WORD
    bitsPerPixel as WORD
    numImages as WORD
End STRUCT
```

SizeX und sizeY enthalten dann die Abmessungen der Grafik in Pixeln. BitsPerPixel enthält die Farbtiefe. Für Graphic Strings wird bitsPerPixel auf Null

gesetzt, daran kann man Graphic Strings erkennen. NumImages enthält im Normalfall den Wert 1, im Fehlerfall den Wert Null. Für Grafikdateien (GetImageInfo-Routine) enthält numImages die Anzahl der Bilder in der Datei - oder ebenfalls Null, wenn die Datei keine Bilddatei ist.

ReleaseNumber

Siehe auch: Spezielle Themen, Vol.2, Kapitel 9.2

Die ReleaseNumber (Routinen FileGetRelease, FileSetRelease) enthält Informationen über die Version der Datei. Der R-BASIC Compiler setzt die ersten drei Felder der Releasenummer von Launcher und BIN-Datei entsprechend der Versionsnummer des Programms bzw. der Library (siehe Befehl Version\$).

```
STRUCT ReleaseNumber          ' Bedeutung der Felder:  
    rnMajor as WORD           ' große, meist inkompatible Neuerungen  
    rnMinor as WORD           ' kleinere, kompatible Neuerungen  
    rnChange AS WORD          ' interne Änderungen  
    rnEngineering as WORD     ' kleine interne Änderungen  
End STRUCT
```

R-BASIC verwendet die Struktur auf für die Routinen FileGetProtocol und FileSetProtocol. Dabei werden nur die ersten beiden Felder benutzt.

DocumentConfigStruct

Siehe auch: Objekt Handbuch, Vol.7, Kapitel 4.13.1

Das DocumentGuardian-Objekt speichert eine Struktur, in der alle zur Verwaltung von Dokumenten notwendigen Daten abgelegt sind.

```
STRUCT DocumentConfigStruct  
    noDocumentString$      As String(32)  
    templateFolder$        As String(32)  
    nameForNew$            As String(32)  
    fileType               As word  
    token                 As GeodeToken  
    creatorToken          As GeodeToken  
    matchMask$             As String(32)  
    matchFlags             As Word  
    reserved(4)            As Word  
End Struct
```

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

PointList

Siehe auch: Programmierhandbuch, Vol.3, Kapitel 2.8.3

Die Struktur enthält eine Liste von Punkten um Polygone, verbundene Linien und Splines zu zeichnen.

```
STRUCT PointList
    numPoints      as INTEGER
    xOffset, yOffset as INTEGER
    x(31)          as INTEGER
    y(31)          as INTEGER
End Struct
```

| Feld | Bedeutung, gültige Werte |
|------------------|--|
| numPoints | Anzahl der gültigen Koordinatenpaare in der Liste Erlaubte Werte: 2 ... 32 |
| xOffset, yOffset | Zusätzliches Offset für die Zeichenposition der Figur. Diese Werte werden zu jedem Koordinatenpaar addiert, bevor die Figur gezeichnet wird. |
| x(31), y(31) | Koordinatenpaare von bis zu 32 Punkten, aus denen die Figur gebildet wird. (x(0)/y(0) bis x(31)/y(31)) |

RectDWord

siehe auch: Objekthandbuch, Vol.5, Kapitel 4.9.2.4

Die Struktur enthält die Koordinaten eines Rechtecks.

```
Struct RectDWord
    x0, y0 as LongInt  ' linke obere Ecke
    x1, y1 as LongInt  ' rechte untere Ecke
End Struct
```

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Anhang D: Mitgelieferte Blockgrafik - Fonts

Die Dateien befinden sich im Ordner in USERDATA\R-BASIC\FONT. Verwenden Sie **FontSetBlock()** um den Blockgrafik-Modus zu aktivieren, **BlockLoad()** um die Dateien zu laden und bei Bedarf **BlockPoke()** oder **BlockREAD()** um einzelne Zeichen zu ändern.

Die IBM-Zeichensätze sind alle sehr gut lesbar, da sie speziell für den Einsatz als Blockgrafik-Zeichensätze entworfen wurden. Die Zeichen ab dem ASCII-Code 128 entsprechen dem DOS-Zeichensatz, d.h. es gibt Grafikzeichen z.B. für Rahmen und die Umlaute entsprechen nicht dem GEOS-Zeichensatz. Verwenden Sie **Convert\$()** mit dem Flag GEOS_TO_DOS um Umlaute zu auszugeben.

Das Programm-Beispiel "BlockFonts Lister" im Ordner "Beispiele\Text" zeigt die installierten Blockgrafik-Zeichensätze an.

| Dateiname | Breite | Höhe | Beschreibung |
|--------------|--------|------|---|
| SNS16x18.RBF | 16 | 18 | GEOS-Zeichensatz, URW Sans |
| MNO10x16.RBF | 10 | 16 | GEOS-Zeichensatz, URW Mono Entspricht dem R-BASIC Standard-Font |
| MINI8x8.RBF | 8 | 8 | GEOS-Zeichensatz, URW Mono unbearbeitet, daher schwer lesbar |
| IBM6x8.RBF | 6 | 8 | sehr kleine Schrift, IBM-Zeichensatz |
| IBM8x8.RBF | 8 | 8 | kleine Schrift, IBM-Zeichensatz |
| IBM8x8T.RBF | 8 | 8 | kleine Schrift, IBM-Zeichensatz dünnerne Zeichen als IBM8x8 |
| IBM6x14.RBF | 6 | 14 | IBM-Zeichensatz |
| IBM8x14.RBF | 8 | 14 | IBM-Zeichensatz |
| IBM8x14T.RBF | 8 | 14 | IBM-Zeichensatz dünnerne Zeichen als IBM8x14 |
| IBM8x14A.RBF | 8 | 14 | IBM-Zeichensatz Alternative Zeichen bei ASCII-Codes > 127 |
| IBM8x16.RBF | 8 | 16 | IBM-Zeichensatz Standardschrift auf DOS-Ebene |
| ALIEN16.RBF | 16 | 16 | Einige Grafikzeichen (Aliens, Mauern..) verwendet von einigen Beispielprogrammen |
| KC85_8.RBF | 8 | 8 | KC-85 Zeichensatz. Ab ASCII Code 128 |
| KC85_16.RBF | 16 | 16 | wiederholen sich die Zeichen. |
| KC85_24.RBF | 24 | 24 | |
| KC85Z_8.RBF | 8 | 8 | KC-85 Zeichensatz. Ab ASCII Code 128 |
| KC85Z_16.RBF | 16 | 16 | befinden sich Grafikzeichen. Tabelle dazu siehe Abschnitt H |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Anhang E: Numerische Werte der UI-Instance Variablen

Einige Instance-Variablen können mit der offiziellen R-BASIC Syntax nur im UI-Code gesetzt werden. Um diese Werte zur Laufzeit zu verändern verwendet man die Befehle **ObjAddHint** und **ObjRemoveHint** (siehe Objekt-Handbuch, Kapitel 3.3.8).

Die Tabelle enthält die für diese Funktionen nötigen Zahlenwerte.

| UI - Instance Variable bzw. Hint | numerischer Code | Datenwerte |
|------------------------------------|------------------|--------------------|
| GenericClass | | |
| DefaultFocus | 24624 | - |
| DefaultTarget | 24628 | - |
| MinimizeChildSpacing | 25068 | - |
| IncludeEndsInChildSpacing | 24728 | - |
| DivideWidthEqually | 24800 | - |
| DivideHeightEqually | 24804 | - |
| ExpandWidth | 24712 | - |
| ExpandHeight | 24708 | - |
| NoWiderThanChildren | 24760 | - |
| NoHigherThanChildren | 24756 | - |
| DrawInBox | 24704 | - |
| MakeToolBox | 24976 | - |
| MakeReplyBar | 24744 | - |
| NoSeparatorLine | 24608 | - |
| SizeWindowAsDesired | 24936 | - |
| NoSysMenu | 25172 | - |
| NoTitleBar | 25168 | - |
| WindowNotMovable | 25112 | - |
| PositionWindowAtMouse | 24942 | - |
| WindowPositionFromParent | 24908 | xPos, yPos as word |
| ExtendWindowToBottomRight | 24928 | - |
| ExtendWindowNearBottomRight | 24932 | - |
| WindowSizeFromParent | 24940 | xPos, yPos as word |
| WindowSizeFromScreen | 24944 | xPos, yPos as word |
| StaggerWindow | 24912 | - |
| CenterWindow | 24916 | - |
| TileWindow | 24920 | - |
| WindowNoConstraints | 25156 | - |
| MakeDelayedApply | 24824 | - |
| ApplyEvenIfNotEnabled | 25048 | - |
| ApplyEvenIfNotModified | 25044 | - |
| Primary | | |
| NoExpressMenu | 27144 | - |
| NoFileMenu | 27140 | - |
| PrimaryFullScreen | 27136 | - |
| PrimaryNoHelpButton | 27152 | - |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

| | | | |
|---------------------------------------|-------|---|--|
| Group | | | |
| CannotBeDefault | 26664 | – | |
| DontCenterTabbedChildren | 26724 | – | |
| Dialog | | | |
| NoFocus | 26688 | – | |
| MakeResizable | 26660 | – | |
| RadioButtonGroup | | | |
| DisplayCurrentSelection | 26652 | – | |
| ModifiedOnRedundantSelection | 26628 | – | |
| Number | | | |
| SliderShowMinMax | 26684 | – | |
| SliderNoDigitalDisplay | 26708 | – | |
| NavigateToNextFieldOnReturn | 26636 | – | |
| Button | | | |
| IsDestructive | 26664 | – | |
| BringsUpWindow | 26624 | – | |
| Memo und InputLine | | | |
| TextFrame | 26688 | – | |
| TextNoFrame | 26692 | – | |
| SelectableIfRO | 26632 | – | |
| Display | | | |
| MinimizedOnStartup | 26624 | – | |
| NotMinimizable | 26644 | – | |
| MaximizedOnStartup | 26632 | – | |
| NotMaximizable | 26648 | – | |
| NotResizable | 26664 | – | |
| NotRestorable | 26652 | – | |
| DisplayGroup | | | |
| NoFullScreenMode | 26644 | – | |
| NoOverlappingMode | 26648 | – | |
| FullSizedOnStartup | 26632 | – | |
| FullSizedOnStartup | 26636 | – | |
| TiledOnStartup | 26628 | – | |
| TileHorizontally | 26656 | – | |
| TileVertically | 26660 | – | |
| SizeIndependentlyOfDisplays | 26664 | – | |
| View | | | |
| DoNotWinScroll | 26648 | – | |
| ImmediateDragUpdates | 26652 | – | |
| DelayedDragUpdates | 26656 | – | |
| HideScrollersWhenNotScrollable | 26660 | – | |
| GenContent | | | |
| keepFocusVisible | 26624 | – | |

Nicht direkt vom PC/GEOS-SDK unterstützt und daher ohne Nummer

FileMenuChildren, DefaultScreen

Parameter:

xPos, yPos Codierte Prozent-Werte (prozentX, prozentY von 0 bis 100)
xPos = 32768 + (1023*prozentX/100) ' z.B. 50% = 33279
yPos = 32768 + (1023*prozentY/100) ' z.B. 25% = 33023

Anhang F: Definitionsbereich der mathematischen Funktionen

Einige der mathematischen Funktionen von R-BASIC haben einen eingeschränkten Definitionsbereich, d.h. sie sind für Argumente (x-Werte) außerhalb eines bestimmten Zahlenbereichs nicht anwendbar. Der Grund dafür kann mathematischer Natur (z.B. bei TAN(x)) sein, er kann aber auch im Zahlenbereich von GEOS (z.B. EXP(x)) begründet sein.

Die untenstehenden Tabellen zeigen den akzeptierten Definitionsbereich und das Ergebnis der Funktion, wenn das Argument x außerhalb diese Zahlenbereichs liegt.

| | |
|-----------|--|
| — | kann nicht auftreten |
| Fehler | Ein spezieller Fehler-Wert wird zurückgegeben |
| Überlauf | Ein spezieller Überlauf-Wert wird zurückgegeben |
| Unterlauf | Ein spezieller Unterlauf-Wert wird zurückgegeben |
| (***) | Verweis auf Anmerkung unter der Tabelle |

Rechnet man mit den Fehlerwerten (Fehler, Unterlauf und Überlauf) weiter, so bleibt der Fehlerwert erhalten. Gibt man einen Fehlerwert aus (Z.B. PRINT oder Str\$(x)), so wird der entsprechende Text ("Fehler", "Überlauf", "Unterlauf") ausgeben.

Beispiel:

| | |
|---------------|-------------------------------|
| PRINT SQR(-3) | ! Das Wort "Fehler" erscheint |
|---------------|-------------------------------|

Einfache mathematische Funktionen

| Funktion | Definitionsbereich | Ergebnis bei falschem x |
|----------|-------------------------------|-------------------------|
| ABS(x) | $-\infty \leq x \leq +\infty$ | — |
| SGN(x) | $-\infty \leq x \leq +\infty$ | — |
| INT(x) | $-\infty \leq x \leq +\infty$ | — |
| TRUNC(x) | $-\infty \leq x \leq +\infty$ | — |
| FRAC(x) | $-\infty \leq x \leq +\infty$ | — |
| ROUND(x) | $-\infty \leq x \leq +\infty$ | — |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Transzendenten Funktionen

| Funktion | Definitionsbereich | Ergebnis bei falschem x |
|----------|---------------------------------|---|
| SQR(x) | $0 \leq x \leq +\infty$ | $x < 0$: Fehler |
| EXP(x) | $-\infty \leq x \leq 11355,433$ | $x > 11355,433$: Überlauf |
| LN(x) | $0 < x \leq +\infty$ | $x = 0$: Unterlauf $x < 0$: Fehler |
| LOG(x) | $0 < x \leq +\infty$ | siehe LN |
| LG(x) | $0 < x \leq +\infty$ | siehe LN |

Trigonometrische Funktionen (*)

| Funktion | Definitionsbereich | Ergebnis bei falschem x |
|----------|---|--|
| SIN(x) | $-\infty \leq x \leq +\infty$ | — |
| COS(x) | $-\infty \leq x \leq +\infty$ | — |
| TAN(x) | $-\infty \leq x \leq +\infty$ außer ungeradzahlige Vielfache von $\pi/2$ | wenn $x = k \cdot \pi/2$ ($k = 1, 3, 5, \dots$) mathematisch: ∞ GEOS liefert aber eine "große" Zahl y; i. a. ist $ y > 1018$ |
| ASN(x) | $-1 \leq x \leq 1$ | $x < -1$ oder $x > 1$: Fehler |
| ACS(x) | $-1 \leq x \leq 1$ | $x < -1$ oder $x > 1$: Fehler |
| ATN(x) | $-\infty \leq x \leq +\infty$ | — |

- (*) Mathematisch sind die trigonometrischen Funktionen für beliebige x definiert (Ausnahme: Tangens). Für sehr große x (etwa ab 1017) können sich jedoch die numerischen Fehler hochschaukeln, so dass sich falsche oder sogar ungültige Werte (z.B. $\sin(x) > 1$) ergeben.

Hyperbolische Funktionen

| Funktion | Definitionsbereich | Ergebnis bei falschem x |
|----------|------------------------------------|---|
| SH(x) | $-11356,5 \dots +11356,5$ | $x < -11356,5$: Unterlauf $x > +11356,5$: Überlauf |
| COSH(x) | $-11356,5 \dots +11356,5$ | siehe SH(x) |
| TANH(x) | $-11356,5 \dots +11356,5$ | siehe SH(x) |
| ASNH(x) | $-\infty \leq x \leq +\infty$ (**) | s. unten |
| ACSH(x) | $1 \leq x \leq +\infty$ (***) | $x < 1$ Fehler |
| ATNH(x) | $-1 < x < 1$ | $x \leq -1$: Unterlauf $x \geq +1$: Überlauf |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

(**) Mathematisch ist ASH(x) für beliebige x definiert. Aus der internen Berechnung des Wertes ergeben sich folgende Einschränkungen:

$$\begin{array}{ll} x < -4.5 \cdot 10^9 & : \text{Unterlauf} \\ x > +1.09 \cdot 10^{2466} & : \text{Überlauf} \end{array}$$

(***) Aus der internen Berechnung des Wertes ergibt sich die Einschränkung:

$$x > +1.09 \cdot 10^{2466} : \text{Überlauf}$$

WWFixed-Funktionen

Für viele der oben genannten Funktionen gibt es entsprechende WWFixed-Funktionen. Mathematisch gelten hier die gleichen Bedingungen wie für die oben genannten Funktionen. Aufgrund der Tatsache, dass die WWFixed-Funktionen hochoptimierten Ganzzahl-Arithmetik-Code verwenden, gibt es zusätzliche Einschränkungen:

- Es erfolgt **keine Fehlerprüfung**. Statt der oben genannten Worte "Fehler", "Überlauf" und "Unterlauf" erhält man einfach ein fehlerhaftes Ergebnis.
- Argumente außerhalb des Definitionsbereichs werden nicht erkannt. Zum Beispiel führt die Division durch Null oder die Wurzel aus einer negativen Zahl nicht zu einem Fehler, sondern zu einem scheinbar zufälligen Wert.
- Überträge werden nicht erkannt. Ergibt sich bei einer Berechnung ein Wert größer als 32768 oder kleiner als -32768, so werden die höherwertigsten Bits abgeschnitten. Das führt meist zu einer Änderung des Vorzeichens der Zahl.
- FixAsn arbeitet nur für positive Argumente korrekt.

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Anhang G: Konstanten zur Musikwiedergabe

Die Konstanten sind in der Library "MusicValues" definiert.

Tabelle: Notenfrequenzen

Die Routinen erwarten ganze Zahlen. Deswegen sind die Frequenzen der Konstanten gerundet. Beachten Sie, dass die Note _B identisch mit _H ist. Der Nachsatz _SH bedeutet: einen halben Ton erhöht (sharpened)

| Note | Frequenz Grundton | Konstante und zugehöriger Wert | | | |
|------|----------------------|--------------------------------|----------------|---------------------|------------------|
| c' | C4 | 261,626 | LOW_C = 262 | MIDDLE_C = 523 | HIGH_C = 1047 |
| cis' | C#4 | 277,183 | LOW_C_SH = 277 | MIDDLE_C_SH = 554 | HIGH_C_SH = 1109 |
| d' | D4 | 293,665 | LOW_D = 294 | MIDDLE_D = 587 | HIGH_D = 1175 |
| dis' | D#4 | 311,127 | LOW_D_SH = 311 | MIDDLE_D_SH = 622 | HIGH_D_SH = 1245 |
| e' | E4 | 329,628 | LOW_E = 330 | MIDDLE_E = 659 | HIGH_E = 1319 |
| f' | F4 | 349,228 | LOW_F = 349 | MIDDLE_F = 698 | HIGH_F = 1397 |
| fis' | F#4 | 369,994 | LOW_F_SH = 370 | MIDDLE_F_SH = 740 | HIGH_F_SH = 1480 |
| g' | G4 | 391,995 | LOW_G = 392 | MIDDLE_G = 784 | HIGH_G = 1568 |
| gis' | G#4 | 415,305 | LOW_G_SH = 415 | MIDDLE_G_SH = 831 | HIGH_G_SH = 1661 |
| a' | A4 | 440,000 | LOW_A = 440 | MIDDLE_A = 880 | HIGH_A = 1760 |
| ais' | A#4 | 466,164 | LOW_A_SH = 466 | MIDDLE_A_SH = 932 | HIGH_A_SH = 1865 |
| h' | B4 | 493,883 | LOW_H = 494 | MIDDLE_H = 988 | HIGH_H = 1976 |
| | | | LOW_B = LOW_H | MIDDLE_B = MIDDLE_H | HIGH_B = HIGH_H |
| c'' | C5 | 523,251 | MIDDLE_C = 523 | HIGH_C = 1047 | X_HIGH_C = 2093 |

Tabelle: Musikinstrumente

Fett markierte Werte sind als Konstanten in R-BASIC (ohne das Einbinden der Library MusicValues) definiert.

| Instrument | Wert | Instrument | Wert |
|----------------------------|-----------|-----------------|------|
| IP_GRAND_PIANO | 0 | IP_SOPRANO_SAX | 64 |
| IP_BRIGHT_PIANO | 1 | IP_ALTO_SAX | 65 |
| IP ELECTRIC_GRAND_PIANO | 2 | IP_TENOR_SAX | 66 |
| IP_HONKY_TONK_PIANO | 3 | IP_BARITONE_SAX | 67 |
| IP ELECTRIC_PIANO_1 | 4 | IP_OBOE | 68 |
| IP_ELECTRIC_PIANO_2 | 5 | IP_ENGLISH_HORN | 69 |
| IP_HARPSICORD | 6 | IP_BASSOON | 70 |
| IP_CLAVICORD | 7 | IP_CLARINET | 71 |
| IP_CELESTA | 8 | IP_PICCOLO | 72 |
| IP_GLOCKENSPIEL | 9 | IP_FLUTE | 73 |
| IP_MUSIC_BOX | 10 | IP_RECORDER | 74 |
| IP_VIBraphone | 11 | IP_PAN_FLUTE | 75 |
| IP MARIMBA | 12 | IP_BLOWN_BOTTLE | 76 |
| IP_XYLOPHONE | 13 | IP_SHAKUHACHI | 77 |
| IP_TUBULAR_BELLS | 14 | IP_WHISTLE | 78 |
| IP_DULCIMER | 15 | IP_OCARINA | 79 |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

| | | | |
|---------------------------|-----------|------------------------|------------|
| IP_DRAWBAR_ORGAN | 16 | IP_LEAD_SQUARE | 80 |
| IP_PERCUSSIVE_ORGAN | 17 | IP_LEAD_SAWTOOTH | 81 |
| IP_ROCK_ORGAN | 18 | IP_LEAD_CALLIOPE | 82 |
| IP_CHURCH_ORGAN | 19 | IP_LEAD_CHIFF | 83 |
| IP_REED_ORGAN | 20 | IP_LEAD_CHARANG | 84 |
| IP_ACCORDIAN | 21 | IP_LEAD_VOICE | 85 |
| IP_HARMONICA | 22 | IP_LEAD_FIFTHS | 86 |
| IP_TANGO_ACCORDION | 23 | IP_LEAD_BASS_LEAD | 87 |
| IP_ACOUSTIC_NYLON_GUITAR | 24 | IP_PAD_NEW AGE | 88 |
| IP_ACOUSTIC_STEEL_GUITAR | 25 | IP_PAD_WARM | 89 |
| IP_ELECTRIC_JAZZ_GUITAR | 26 | IP_PAD_POLYSYNTH | 90 |
| IP_ELECTRIC_CLEAN_GUITAR | 27 | IP_PAD_CHOIR | 91 |
| IP_ELECTRIC_MUTED_GUITAR | 28 | IP_PAD_BOWED | 92 |
| IP_OVERDRIVEN_GUITAR | 29 | IP_PAD_METALLIC | 93 |
| IP_DISTORTION_GUITAR | 30 | IP_PAD_HALO | 94 |
| IP_GUITAR_HARMONICS | 31 | IP_PAD_SWEEP | 95 |
| IP_ACOUSTIC_BASS | 32 | IP_FX_RAIN | 96 |
| IP_ELECTRIC_FINGERED_BASS | 33 | IP_FX_SOUNDTRACK | 97 |
| IP_ELECTRIC_PICKED_BASS | 34 | IP_FX_CRYSTAL | 98 |
| IP_FRETLESS_BASS | 35 | IP_FX_ATMOSPHERE | 99 |
| IP_SLAP_BASS_1 | 36 | IP_FX_BRIGHTNESS | 100 |
| IP_SLAP_BASS_2 | 37 | IP_FX_GOBLINS | 101 |
| IP_SYNTH_BASS_1 | 38 | IP_FX_ECHOES | 102 |
| IP_SYNTH_BASS_2 | 39 | IP_FX SCI FI | 103 |
| IP_VIOLIN | 40 | IP_SITAR | 104 |
| IP_VIOLA | 41 | IP_BANJO | 105 |
| IP_CELLO | 42 | IP_SHAMISEN | 106 |
| IP_CONTRABASS | 43 | IP_KOTO | 107 |
| IP_TREMELO_STRINGS | 44 | IP_KALIMBA | 108 |
| IP_PIZZICATO_STRINGS | 45 | IP_BAG_PIPE | 109 |
| IP_ORCHESTRAL_HARP | 46 | IP_FIDDLE | 110 |
| IP_TIMPANI | 47 | IP_SHANAI | 111 |
| IP_STRING_ENSEMBLE_1 | 48 | IP_TINKLE_BELL | 112 |
| IP_STRING_ENSEMBLE_2 | 49 | IP_AGOGO | 113 |
| IP_SYNTH_STRINGS_1 | 50 | IP_STEEL_DRUMS | 114 |
| IP_SYNTH_STRINGS_2 | 51 | IP_WOODBLOCK | 115 |
| IP_CHOIR_AAHS | 52 | IP_TAIKO_DRUM | 116 |
| IP_VOICE_OOHS | 53 | IP_MELODIC_TOM | 117 |
| IP_SYNTH_VOICE | 54 | IP_SYNTH_DRUM | 118 |
| IP_ORCHESTRA_HIT | 55 | IP_REVERSE_CYMBAL | 119 |
| IP_TRUMPET | 56 | IP_GUITAR_FRET_NOISE | 120 |
| IP_TROMBONE | 57 | IP_BREATH_NOISE | 121 |
| IP_TUBA | 58 | IP_SEASHORE | 122 |
| IP_MUTED_TRUMPET | 59 | IP_BIRD_TWEET | 123 |
| IP_FRENCH_HORN | 60 | IP_TELEPHONE_RING | 124 |
| IP_BRASS_SECTION | 61 | IP_HELICOPTER | 125 |
| IP_SYNTH_BRASS_1 | 62 | IP_APPLAUSE | 126 |
| IP_SYNTH_BRASS_2 | 63 | IP_GUNSHOT | 127 |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Tabelle: Percussion-Instrumente

Damit Percussioninstrumente gut klingen sollten sie laut Dokumentation mit einer vorgegebenen Frequenz gespielt werden. Fett markierte Werte sind als Konstanten in R-BASIC (ohne das Einbinden der Library MusicValues) definiert.

Tipp 1: Verwenden Sie ruhig auch andere Frequenzen (Noten).

| Percussion-Instrument | Wert | Zugehörige Frequenz | Wert |
|------------------------------|------------|-----------------------|------|
| IP_ACOUSTIC_BASS_DRUM | 128 | FR_ACOUSTIC_BASS_DRUM | 247 |
| IP_BASS_DRUM_1 | 129 | FR_BASS_DRUM_1 | 131 |
| IP_SIDE_STICK | 130 | FR_SIDE_STICK | 262 |
| IP_ACOUSTIC_SNARE | 131 | FR_ACOUSTIC_SNARE | 233 |
| IP_HAND_CLAP | 132 | FR_HAND_CLAP | 392 |
| IP_ELECTRIC_SNARE | 133 | FR_ELECTRIC_SNARE | 233 |
| IP_LOW_FLOOR_TOM | 134 | FR_LOW_FLOOR_TOM | 98 |
| IP_CLOSED_HI_HAT | 135 | FR_CLOSED_HI_HAT | 1047 |
| IP_HIGH_FLOOR_TOM | 136 | FR_HIGH_FLOOR_TOM | 110 |
| IP_PEDAL_HI_HAT | 137 | FR_PEDAL_HI_HAT | 156 |
| IP_LOW_TOM | 138 | FR_LOW_TOM | 87 |
| IP_OPEN_HI_HAT | 139 | FR_OPEN_HI_HAT | 523 |
| IP_LOW_MID_TOM | 140 | FR_LOW_MID_TOM | 87 |
| IP_HI_MID_TOM | 141 | FR_HI_MID_TOM | 123 |
| IP_CRASH_CYMBAL_1 | 142 | FR_CRASH_CYMBAL_1 | 523 |
| IP_HIGH_TOM | 143 | FR_HIGH_TOM | 87 |
| IP_RIDE_CYMBAL_1 | 144 | FR_RIDE_CYMBAL_1 | 1047 |
| IP_CHINESE_CYMBAL | 145 | FR_CHINESE_CYMBAL | 247 |
| IP_RIDE_BELL | 146 | FR_RIDE_BELL | 330 |
| IP_TAMBOURINE | 147 | FR_TAMBOURINE | 1047 |
| IP_SPLASH_CYMBAL | 148 | FR_SPLASH_CYMBAL | 208 |
| IP_COWBELL | 149 | FR_COWBELL | 233 |
| IP_CRASH_CYMBAL_2 | 150 | FR_CRASH_CYMBAL_2 | 392 |
| IP_VIBRASLAP | 151 | FR_VIBRASLAP | 104 |
| IP_RIDE_CYMBAL_2 | 152 | FR_RIDE_CYMBAL_2 | 1047 |
| IP_HI_BONGO | 153 | FR_HI_BONGO | 131 |
| IP_LOW_BONGO | 154 | FR_LOW_BONGO | 131 |
| IP_MUTE_HI_CONGA | 155 | FR_MUTE_HI_CONGA | 156 |
| IP_OPEN_HI_CONGA | 156 | FR_OPEN_HI_CONGA | 262 |
| IP_LOW_CONGA | 157 | FR_LOW_CONGA | 131 |
| IP_HI_TIMBALE | 158 | FR_HI_TIMBALE | 131 |
| IP_LOW_TIMBALE | 159 | FR_LOW_TIMBALE | 131 |
| IP_HIGH_AGOGO | 160 | FR_HIGH_AGOGO | 1047 |
| IP_LOW_AGOGO | 161 | FR_LOW_AGOGO | 415 |
| IP_CABASA | 162 | FR_CABASA | 1047 |
| IP_MARACAS | 163 | FR_MARACAS | 1047 |
| IP_SHORT_WHISTLE | 164 | FR_SHORT_WHISTLE | 1047 |
| IP_LONG_WHISTLE | 165 | FR_LONG_WHISTLE | 1047 |
| IP_SHORT_GUIRO | 166 | FR_SHORT_GUIRO | 92 |
| IP_LONG_GUIRO | 167 | FR_LONG_GUIRO | 87 |
| IP_CLAVES | 168 | FR_CLAVES | 104 |
| IP_HI_WOOD_BLOCK | 169 | FR_HI_WOOD_BLOCK | 104 |
| IP_LOW_WOOD_BLOCK | 170 | FR_LOW_WOOD_BLOCK | 104 |
| IP_MUTE_CUICA | 171 | FR_MUTE_CUICA | 392 |
| IP_OPEN_CUICA | 172 | FR_OPEN_CUICA | 392 |
| IP_MUTE_TRIANGLE | 173 | FR_MUTE_TRIANGLE | 698 |
| IP_OPEN_TRIANGLE | 174 | FR_OPEN_TRIANGLE | 698 |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Anhang H: Tabellen und Übersichten zur KC-85 Kompatibilität

Aufbau des KC-85 Bildschirms:

Bildschirmgröße: 320 x 256 Pixel, entsprechend 32 Zeilen zu je 40 Zeichen bei einer Zeichengröße von 8x8 Pixeln. Für je 8 nebeneinander liegende Pixel wird in einem Byte im Pixel-RAM gespeichert, ob die Vordergrund- oder die Hintergrundfarbe angezeigt werden soll. Für je 4 übereinanderliegende Bytes des Pixel-RAM wird im Color-RAM die Vordergrund- und die Hintergrundfarbe abgelegt.

R-BASIC kann einen größeren (640 x 512 bzw. 960 x 768 Pixel) Bildschirm mit entsprechend größerem Zeichensatz verwenden, ohne die Kompatibilität aufzugeben.

Farbcodes

Die Farbcodes der 16 Standardfarben unterscheiden sich in R-BASIC deutlich von den im KC-85 BASIC verwendeten Werten. Die Unterschiede sind in der folgenden Tabelle dargestellt. Um blinkende Zeichen zu realisieren verwendet R-BASIC eine Bitmap mit Palette und die in der Tabelle angegebenen Farbwerte um die KC-85 Farben darzustellen.

| Farocode (Farbe unter GEOS) | Farbe auf dem KC | GEOS Color-Index im KC Mode | |
|--------------------------------|------------------|-----------------------------|-------------|
| | | Vordergrund | Hintergrund |
| 0 (Schwarz) | Schwarz | 0 | 40 |
| 1 (Blau) | Blau | 87 | 44 |
| 2 (Grün) | Rot | 226 | 184 |
| 3 (Türkis) | Purpur | 237 | 188 |
| 4 (Rot) | Grün | 107 | 84 |
| 5 (Lila) | Türkis | 111 | 68 |
| 6 (Braun) | Gelb | 251 | 208 |
| 7 (Hellgrau) | Weiß / Grau ** | 15 | 26 |
| 8 (Dunkelblau) | Schwarz | 16 | - |
| 9 (Hellblau) | Violett | 153 | - |
| 10 (Hellgrün) | Orange | 238 | - |
| 11 (Helltürkis) | Purpurrot | 229 | - |
| 12 (Hellrot) | Grünblau | 145 | - |
| 13 (Helllila) | Blaugrün | 93 | - |
| 14 (Gelb) | Gelbgrün | 179 | - |
| 15 (Weiß) | Weiß | 31 | - |

** Als Vordergrundfarbe: Weiß, als Hintergrundfarbe: Grau

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Aufbau eines Color-Byte im Color-RAM



v: Vordergrundfarbe
b: Hintergrundfarbe

30: Verschiebung um 30° auf dem Farbkreis
blink: Vordergrundfarbe blinkend

Der Wert berechnet sich zu

$$\text{colorbyte} = 128 * \text{blink} + 8 * v + h$$

mit blink = 0 (blitzen aus) oder blink = 1 (blitzen an)
v: Vordergrundfarbe 0 .. 15
h: Hintergrundfarbe 0 .. 7

Steuertasten und Steuercodes

Die Codes einiger Steuertasten unterscheiden sich auf dem KC-85 von denen in R-BASIC. Das Bit 12 (&h1000) in den kc85Features bewirkt, dass Inkey\$ den zugehörigen KC-Code liefert.

Die ESC-Taste ersetzt **bei laufendem BASIC Programm** die STOP-Taste des KC-85. Hier ist die Zweitbelegung (Code 27) unter R-BASIC nicht verfügbar. Der Code der BRK-Taste des KC-85 (Code 03) ist über die Kombination Strg-Ende erreichbar. Diese Taste bricht auf dem KC-85 ein BASIC Programm ab, unter R-BASIC jedoch nicht. Dazu gibt es in R-BASIC die Tastenkombination Strg-B.

Achtung! Die Zweitbelegung der KC-85-Steuertasten ist unter R-BASIC mit der Kombination **Strg+Taste** erreichbar, **nicht** über Shift+Taste!

| Taste PC | Code in R-BASIC | Taste KC-85 | Code auf dem KC-85 Erstbelegung | Code auf dem KC-85 Zweitbelegung |
|-----------|-----------------|-------------|------------------------------------|-------------------------------------|
| ↑ | 11 | ↑ | 11 (0Bhex) | 17 (11hex) |
| ↓ | 10 | ↓ | 10 (0Ahex) | 18 (12hex) |
| → | 15 | → | 09 (0Fhex) | 24 (18hex) |
| ← | 14 | ← | 08 (0Ehex) | 25 (19hex) |
| Einfg | 22 | INS | 26 (1Ahex) | 20 (14hex) |
| Entf | 23 | DEL | 31 (1Fhex) | 02 (02hex) |
| Pos1 | 21 | HOM | 16 (10hex) | 12 (0Chex) |
| Ende | 20 | – / BRK | 18 (12hex) | 03 (03hex) |
| Bild ↑ | 17 | | 17 (11hex) | 17 (11hex) |
| Bild ↓ | 18 | | 18 (12hex) | 18 (12hex) |
| ESC | 27 | STOP | 19 (13hex) | – |
| Backspace | 8 | CLR | 01 (01hex) | 01 (01hex) |

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Zeichensatz

Die BlockGrafik Zeichensätze für den KC (KCxxxx.RBF) enthalten für die Codes unter 32 (Steuerzeichen) die in der Tabelle angegebenen Symbole. Um sie auszugeben muss man das Bit TS_DONT_EXEC_CONTROLS in Feld "style" der Systemvariablen "printFont" setzen. Der folgende Code gibt die Pfeile aus:

```
printfont.style = TS_DONT_EXEC_CONTROLS  
    ' Zeichen unter 32 drucken und nicht "ausführen"  
print "\011\010\008\009"  
printfont.style = 0
```

| Code Dezimal | Hex | Zeichen |
|-----------------|-----|---------|-----------------|-----|---------|-----------------|-----|---------|-----------------|-----|---------|
| 0 | 0 | — | 8 | 8 | ↔ | 16 | 10 | ↑ | 24 | 18 | → |
| 1 | 1 | ↶ | 9 | 9 | → | 17 | 11 | ↓ | 25 | 19 | ← |
| 2 | 2 | ↖ | 10 | A | ↙ | 18 | 12 | ⤒ | 26 | 1A | ⤓ |
| 3 | 3 | ⤔ | 11 | B | ⤓ | 19 | 13 | ⤕ | 27 | 1B | ⤖ |
| 4 | 4 | ⤖ | 12 | C | ⤔ | 20 | 14 | ⤗ | 28 | 1C | ⤘ |
| 5 | 5 | ⤙ | 13 | D | ⤖ | 21 | 15 | ⤙ | 29 | 1D | ⤙ |
| 6 | 6 | ⤚ | 14 | E | ⤙ | 22 | 16 | ⤚ | 30 | 1E | ⤚ |
| 7 | 7 | ⤛ | 15 | F | ⤚ | 23 | 17 | ⤛ | 31 | 1F | ⤛ |

Im Bereich von 32 (20hex) bis 127 (1Fhex) stimmen die Zeichencodes von KC und ASCII bzw. GEOS weitgehend überein. Die folgende Tabelle enthält einen Überblick über die Zeichen des KC-85.

| Bereich | Hex | Zeichen |
|-----------|---------|---|
| 32 - 90 | 20 - 5A | ASCII-Zeichen: Sonderzeichen, Ziffern, Großbuchstaben |
| 91 - 96 | 5B - 60 | Teilweise Abweichungen zu ASCII 91: Vollzeichen 92: (Strich, senkrecht) 93: Negationszeichen 94: ^ (Exponent) 95: _ Unterstrich 96: © (Copyright) |
| 97 - 122 | 61 - 7A | ASCII-Zeichen: Kleinbuchstaben |
| 123 - 127 | 7B - 7F | Deutsche Umlaute (nur Kleinbuchstaben) 123: ä 124: ö 125: ü 126: ß 127: Quadrat-Zeichen |

Bei den Zeichensätzen KC85_8.RBF, KC85_16.RBF und KC85_24.RBF wiederholen sich ab Code 128 die Zeichen. Die Zeichensätze KC85Z_8.RBF,

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

KC85Z_16.RBF und KC85Z_24.RBF enthalten ab Code 128 Grafikzeichen, die vom Z9001 übernommen wurden und die im folgenden Bild dargestellt sind:

| Aufbau der Tabelle: | | Code dezimal | - | Code hexadezimal | - | Grafikzeichen | | |
|---------------------|----|--------------|-----|------------------|----|---------------|-----|----|
| 128 | 80 | | 144 | 90 | | 160 | A0 | |
| 129 | 81 | | 145 | 91 | | 161 | A1 | |
| 130 | 82 | | 146 | 92 | | 162 | A2 | |
| 131 | 83 | | 147 | 93 | | 163 | A3 | |
| 132 | 84 | | 148 | 94 | | 164 | A4 | |
| 133 | 85 | | 149 | 95 | | 165 | A5 | |
| 134 | 86 | | 150 | 96 | | 166 | A6 | |
| 135 | 87 | | 151 | 97 | | 167 | A7 | |
| 136 | 88 | | 152 | 98 | | 168 | A8 | |
| 137 | 89 | | 153 | 99 | | 169 | A9 | |
| 138 | 8A | | 154 | 9A | | 170 | AA | |
| 139 | 8B | | 155 | 9B | | 171 | AB | |
| 140 | 8C | | 156 | 9C | | 172 | AC | |
| 141 | 8D | | 157 | 9D | | 173 | AD | |
| 142 | 8E | | 158 | 9E | | 174 | AE | |
| 143 | 8F | | 159 | 9F | | 175 | AF | |
| | | | | 176 | B0 | | 192 | C0 |
| | | | | 177 | B1 | | 193 | C1 |
| | | | | 178 | B2 | | 194 | C2 |
| | | | | 179 | B3 | | 195 | C3 |
| | | | | 180 | B4 | | 196 | C4 |
| | | | | 181 | B5 | | 197 | C5 |
| | | | | 182 | B6 | | 198 | C6 |
| | | | | 183 | B7 | | 199 | C7 |
| | | | | | *) | | | |
| | | | | 200 | C8 | | 213 | D5 |
| | | | | 201 | C9 | | 214 | D6 |
| | | | | 202 | CA | | 215 | D7 |
| | | | | 203 | CB | | 216 | D8 |
| | | | | 204 | CC | | 217 | D9 |
| | | | | 205 | CD | | 218 | DA |
| | | | | 206 | CE | | 219 | DB |
| | | | | 207 | CF | | 220 | DC |
| | | | | | | | 221 | DD |
| | | | | | | | 222 | DE |
| | | | | | | | 223 | DF |
| | | | | | | | 224 | E0 |
| | | | | | | | 225 | E1 |
| | | | | | | | 226 | E2 |
| | | | | | | | 227 | E3 |
| | | | | | | | 228 | E4 |
| | | | | | | | 229 | E5 |
| | | | | | | | 230 | E6 |
| | | | | | | | 231 | E7 |
| | | | | | | | 232 | E8 |
| | | | | | | | 233 | E9 |
| | | | | | | | 234 | EA |
| | | | | | | | 235 | EB |
| | | | | | | | 236 | EC |
| | | | | | | | 237 | ED |
| | | | | | | | 238 | EE |
| | | | | | | | 239 | EF |
| | | | | | | | 240 | F0 |
| | | | | | | | 241 | F1 |
| | | | | | | | 242 | F2 |
| | | | | | | | 243 | F3 |
| | | | | | | | 244 | F4 |
| | | | | | | | 245 | F5 |
| | | | | | | | 246 | F6 |
| | | | | | | | 247 | F7 |
| | | | | | | | 248 | F8 |
| | | | | | | | 249 | F9 |
| | | | | | | | 250 | FA |
| | | | | | | | 251 | FB |
| | | | | | | | 252 | FC |
| | | | | | | | 253 | FD |
| | | | | | | | 254 | FE |
| | | | | | | | 255 | FF |

Sonderzeichen

Auf dem KC-85 unterscheiden sich die Codes von Umlauten und Sonderzeichen von den auf dem PC üblichen ASCII Codes. Das Bit 11 in kc85Features bestimmt, ob die ASCII-Codes der Umlaute und Sonderzeichen auf der Tastatur für INPUT und INKEY\$ zu den entsprechenden KC-85 Codes konvertiert werden. Die folgenden Sonderzeichen des KC-85-Zeichensatzes sind **bei laufendem BASIC Programm** über spezielle Tastenkombinationen erreichbar.

| Zeichen | Code | Taste | Zeichen | Code | Taste |
|---------|------|--------------|---------|------|-----------|
| – | 93 | AltGr + – | | 92 | AltGr + < |
| █ | 91 | AltGr + Leer | @ | 64 | AltGr + q |
| □ | 127 | AltGr + ß | © | 96 | Strg + g |

Aufbau und Arbeitszellen des IRM (KC 85/3)

Physisch liegt der IRM ab Adresse 8000 (hex). Der Adressparameter von VPEEK und VPOKE beginnt jedoch bei Null. Deswegen muss man jeweils 8000 (hex) abziehen. Die drei folgenden Beispiele greifen jeweils auf die gleiche Speicherstelle zu:

```
VPOKE &hB7AA - &h8000, 0
VPOKE &h37AA, 0
VPOKE 14250, 0
```

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

| Adresse hex | VPOKE VPEEK | Inhalt |
|-------------|-------------|---|
| 8000 | 0 | 10240 Byte Pixelspeicher Für die Zeichenspalten 0 bis 31 ab Adresse 0 Für die Zeichenspalten 32 bis 39 ab Adresse 8192 |
| A800 | 10240 | 2560 Byte Colorspeicher Für die Zeichenspalten 0 bis 31 ab Adresse 10240 Für die Zeichenspalten 32 bis 39 ab Adresse 12288 |
| B200 | 12800 | 1280 Byte ASCII-Speicher, entsprechend 32 x 40 Zeichen. Das Bit 16 in kc85Features bestimmt, ob bei Textausgaben mit Print und Input hier die ASCII-Codes der ausgegebenen Zeichen abgelegt werden, damit sie mit VPEEK und VGet\$ gelesen werden können. |
| B700 | 14080 | Kassettenpuffer. Unbenutzt unter R-BASIC. |
| B780 | 14208 | Arbeitsspeicher des KC-Betriebssystems. Folgende Arbeitszellen werden von R-BASIC unterstützt, wenn das entsprechende Bit in den kc85Features gesetzt ist: |
| B7A2 | 14242 | Steuerbyte für Bildschirmausgabe. Nur Bit 3 wird unterstützt, indem in printFont.style das Bit TS_DONT_EXEC_CONTROLS angepasst wird: Steuercodes ausführen (Bit 3 = 0), Steuercodes als Zeichen ausgeben (Bit 3 = 1) |
| B7A6 | 14246 | Adresse (2 Byte) des Zeichengenerators(*) für die Zeichen 20hex bis 5Fhex |
| B7A8 | 14248 | Adresse (2 Byte) des Zeichengenerators(*) für die Zeichen 00hex bis 1Fhex und 60hex bis 7Fhex. |
| B7AA | 14250 | Adresse (2 Byte) des Zeichengenerators(*) für die Zeichen A0hex bis DFhex |
| B7AC | 14252 | Adresse (2 Byte) des Zeichengenerators(*) für die Zeichen 80hex bis 9Fhex und E0hex bis FFhex |
| B800 | 14336 | Modulsteuerwortspeicher. Unbenutzt unter R-BASIC. |
| B900 | 14592 | Funktionstastenspeicher. Unbenutzt unter R-BASIC. |
| B99C | 14748 | Fenstervektorspeicher. Unbenutzt unter R-BASIC. |
| BA00 | 14848 | Frei für Anwender. |

(*) Der Zeichengenerator ist der Speicher, der die Pixel-Information für die Darstellung der entsprechenden ASCII-Zeichen auf dem Bildschirm enthält. Unter R-BASIC entspricht das dem Block-Font Modus.

Per Default liegt der Zeichengenerator beim KC im ROM (Adresse > C000hex). Wird der Inhalt der Arbeitszelle durch einen VPOKE Befehl auf einen Wert unter C000hex geändert und ist das entsprechende Bit in den kc85Features gesetzt, so gleicht R-BASIC den Inhalt seines Blockgrafik Zeichengenerators mit dem Inhalt des durch die Arbeitszelle referenzierten Bereichs ab. Das bewirkt, dass Änderungen des KC-Zeichensatzes mit POKE, DOKE oder VPOKE auch unter R-BASIC wirksam sind.

R-BASIC Handbuch - Anhänge

Einfach unter PC/GEOS programmieren

Folgende Arbeitszellen im IRM werden zwar von R-BASIC nicht unterstützt, ihre Funktion könnte jedoch durch entsprechende Programmierung, z.B. mit dem WINDOW Befehl, "nachempfunden" werden, wenn das zu portierende Programm diese Zellen belegt oder verwendet. Die anderen Arbeitszellen sind Maschinencode spezifisch. Für eine komplette Liste der Arbeitszellen sehen Sie bitte in die Originalliteratur.

| Adresse hex | VPOKE VPEEK | Inhalt |
|-------------|----------------|--|
| B79B | 14235 | Nummer des aktuellen Bildschirmfensters aus dem Fensterverktorspeicher |
| B79C | 14236 | Fensteranfang (low-Byte: Spalte, high-Byte: Zeile) |
| B79E | 14238 | Fenstergröße (low-Byte: Spalten, high-Byte: Zeilen) |
| B7A0 | 14240 | relative Cursor-Position im Fenster (low-Byte: Spalte, high-Byte: Zeile) |
| B7A2 | 14242 | Steuerbyte für Bildschirmausgabe Bit 0: Schreiben Zeichen ein/aus Bit 1: schreiben Farbe ein/aus (Bit 3: "Steuercode ausführen" wird unterstützt) |
| B7A3 | 14243 | Farbbyte für Bildschirmausgabe. |