Texas Instruments CC-40 BASIC Quick Reference Card

A handy guide to the commands, statements, and functions of TI CC-40 BASIC for the Texas Instruments Compact Computer Model 40. For a complete description of these and other features, see the *Texas Instruments Compact Computer 40 User's Guide.*

Printed in U.S.A.

1052904-1

C: COMMAND **F:** FUNCTION **S:** STATEMENT
S*: STATEMENT USED ONLY IN PROGRAM

ABS(*numeric-expression*)
returns the absolute value of *numeric-expression*. **F**

ACCEPT [[AT(*column*)] [SIZE(*numeric-expression*)] [BEEP] [ERASE ALL] [VALIDATE(*data-type*, ...)] [NULL(*expression*)] ,] *variable*
suspends program execution until data is entered from the keyboard. Optionally, data is entered at *column* and validated and the following options executed. **S***

SIZE(*numeric-expression*): allows up to the absolute value of *numeric-expression* characters to be entered. If *numeric-expression* is positive, that many positions are blanked. If *numeric-expression* is negative, no positions are blanked. If SIZE appears, the cursor is left in the first position following the input field.

BEEP: sounds a short tone for each BEEP in the statement.

ERASE ALL: clears the entire display before accepting input.

VALIDATE *data-types*:

String-expression permits the characters contained in *string-expression*.

ALPHA permits all alphabetic characters.

UALPHA permits only uppercase alphabetic characters.

DIGIT permits 0 through 9.

NUMERIC permits 0 through 9, ".", "+", "–", and "E".

ALPHANUM permits all alphabetic characters and 0 through 9.

UALPHANUM permits only uppercase alphabetic characters and 0 through 9.

NULL(*expression*): provides a default value to be assigned to the variable.

ACS(*numeric-expression*)
returns the angle whose cosine is *numeric-expression* according to the current angle mode in effect. See DEG, GRAD, and RAD. **F**

CALL ADDMEM
appends the Random Access Memory (RAM) contained in an installed *Memory Expansion* cartridge to the useable resident memory. **C**

ASC(*string-expression*)
returns the ASCII code of the first character of *string-expression*. **F**

ASN(*numeric-expression*)
returns the angle whose sine is *numeric-expression* according to the current angle mode in effect. See DEG, GRAD, and RAD. **F**

ATN(*numeric-expression*)
returns the angle whose tangent is *numeric-expression* according to the current angle mode in effect. See DEG, GRAD, and RAD. **F**

ATTACH *sub-name1* [, *sub-name2* ...]
preserves the values of variables used in the listed subprogram(s) between calls to the subprogram(s). **S**

BREAK [*line-number-list*]
suspends program execution when encountered or optionally when lines in *line-number-list* are encountered. **S**

CALL *subprogram-name* [(*argument-list*)]
transfers control to the indicated subprogram. An optional *argument-list* can be passed. **S**

CALL CHAR(*character-code*, *pattern-identifier*)
defines the specified ASCII *character code(s)*, 0-6, using a 1 through 112 character hexadecimal coded string *pattern-identifier*. **S**

CHR$(*numeric-expression*)
returns the string character corresponding to the ASCII character code specified by *numeric-expression*. **F**

CALL CLEANUP
deletes unused variable names from the system. **C**

CLOSE *#file-number* [, DELETE]
terminates the association between a file and its current *file-number* and optionally deletes the file. **S**

CONTINUE [*line-number*]

resumes execution after a breakpoint occurs, optionally at the line specified by *line-number*. **C**

COS(*numeric-expression*)

returns the trigonometric cosine of *numeric-expression* according to the current angle mode in effect. See DEG, GRAD, and RAD. **F**

DATA *data-list*

stores numeric and string constant data in a program. **S**

CALL DEBUG

allows access to the assembly language debugger for testing assembly language subprograms. **S**

DEG

sets the units for angle calculations to degrees. **S**

DELETE *line-group* [, *line-group* ...]

removes lines specified in *line-group* from a program in memory. **S**

DELETE "*device.filename*"

removes the file specified in *filename* from *device*. **S**

DIM *array-name*(*integer1* [, *integer2*] [, *integer3*]) [,...]

specifies the dimensions of listed array(s) and reserves the necessary memory space. **S**

DISPLAY [[AT(*column*)] [BEEP] [ERASE ALL] [SIZE (*numeric-expression*)] [USING *line-number* *string-expression*] ,] *print-list*

displays the value(s) in *print-list*. Optionally, data is displayed at the position specified by *column* and the following options executed. **S**

BEEP: sounds a short tone for each BEEP in the statement.

ERASE ALL: clears the entire 80-column line.

SIZE(*numeric-expression*): limits the total number of characters displayed to the absolute value of *numeric-expression*. The specified field is always cleared prior to displaying data and the cursor is left in the first position following the display field.

USING: specifies the format. If *string-expression* is present, it defines the format. If *line-number* is present, it refers to the line number of an IMAGE statement. See IMAGE.

END

terminates program execution. **S**

EOF(*file-number***)**

returns the end-of-file condition of *file-number*. **F**

0: Not end-of-file

– 1: Logical end-of-file

CALL ERR(*error-code*, *error-type* [, *file-number*, *line-number*])

returns the *error-code* and *error-type* of the last uncleared error. Optionally, returns the *file-number* and *line-number* in which the error occurred. **S**

CALL EXEC(*execution-address* [, *argument-list*])

executes the assembly language program or subprogram located at *execution-address* and optionally passes an *argument-list*. **S**

EXP(*numeric-expression*)

returns the result of e^x, where x is *numeric-expression*. The value of e is 2.71828182846. **F**

FOR *control-variable* = *initial-value* **TO** *limit* **[STEP** *increment*]

repeats execution of statements between FOR and NEXT until *control-variable* exceeds *limit* (when *increment* is positive) or is less than *limit* (when *increment* is negative). STEP *increment* default is one. **S**

FORMAT *device*

initializes the current medium on *device*. **S**

FRE(*numeric-expression*)

returns information about the current use of memory.

Numeric-expression:

0: Memory not reserved for system operation.

1: Memory occupied by the current program.

2: Total free memory and temporarily reserved memory.

3: Largest block of free memory.

4: Total free memory.

5: Number of individual blocks of free memory space. **F**

CALL GETLANG(*numeric-variable*)

returns in *numeric-variable* the code of the international language used to display system messages and errors. **S**

CALL GETMEM(*numeric-expression*, *numeric-variable*)
reserves *numeric-expression* bytes of memory for storing data and assembly language programs and returns in *numeric-variable* the lowest address of the reserved memory. **S**

GOSUB *line-number*
transfers control to the subroutine that begins at *line-number.* **S***

GOTO *line-number*
transfers control to line specified by *line-number.* **S***

GRAD
sets the units for angle calculations to grads. **S**

IF *condition* **THEN** *action1* [**ELSE** *action2*]
performs *action1* if *condition* is true or performs *action2* if condition is false. If ELSE is omitted and *condition* is false, control is transferred to the next line. **S**

IMAGE *string-constant*
specifies the format in which data is PRINTed or DISPLAYed when USING is present. *String-constant* may be all or any of the following:

Letters, numbers, character not listed below: transferred directly.

#: replaced by the *print-list* values given in PRINT or DISPLAY.

^: replaced by the E and power numbers when there are 4 or 5 of these. **S**

CALL INDIC(*indicator-number* [, *indicator-state*])
turns the display indicator specified by *indicator-number* off if *indicator-state* is zero or on if *indicator-state* is nonzero or omitted. The user indicators are 1-6. **S**

INPUT [*input-prompt*;] *variable-list* [, *input-prompt; variable-list*] [...]
suspends program execution until data is entered from the keyboard. The optional *input-prompt* may indicate what data is expected. **S***

INPUT *#file-number* [, REC *numeric-expression*] , *variable-list*
assigns data from the indicated file to the variables in *variable-list*. Records are read sequentially unless REC appears. **S***

INT(*numeric-expression*)

returns the greatest integer less than or equal to *numeric-expression*. **F**

INTRND(*numeric-expression*)

returns an integer random number between 1 and the rounded value of *numeric-expression*. **F**

CALL IO(*device*, *command* [, *status-variable*])
(*string-variable* [, *status-variable*])

performs special control operations on peripheral devices. **S**

CALL KEY (*return-variable*, *status-variable*)

assigns the ASCII code of a key pressed from the keyboard to *return-variable*. Status information is returned in *status-variable*. **S**

Status-variable:

1 means a new key was pressed.
– 1 means the same key was pressed.
0 means no key was pressed.

KEY$

halts program execution until a single key is pressed. **F**

LEN (*string-expression*)

returns the number of characters in *string-expression*. **F**

[LET] *numeric-variable* [, *numeric-variable* ...] = *numeric-expression*

[LET] *string-variable* [, *string-variable* ...] = *string-expression*

assigns the value of an expression to the specified variable(s). **S**

LINPUT [*input-prompt*;] *string-variable*

suspends program execution until data is entered from the keyboard. The optional *input-prompt* may indicate what data is expected. **S***

LINPUT [#*file-number*, [REC *numeric-expression*,]] *string-variable*

assigns data from the indicated file to *string-variable*. Records are read sequentially unless REC appears. **S***

LIST [*line-group*]

sequentially displays all the program lines of the program in memory. Optionally, only the lines specified in *line-group* are displayed. **C**

LIST "*device.filename*" [, *line-group*]

sequentially lists all the program lines of the program in memory to the *device* specified. Optionally, only the lines specified in *line-group* are listed. **C**

LN(*numeric-expression*)

returns the natural logarithm of *numeric-expression*. **F**

CALL LOAD("*device.filename*")

loads assembly language subprograms from *filename* on the specified *device* into computer memory. **S**

LOG(*numeric-expression*)

returns the common logarithm of *numeric-expression*. **F**

NEW [ALL]

deletes the program and variables currently in memory and closes all open files. Optionally, the user-assigned strings, assembly language subprograms, and display indicators can be cleared, any expansion of memory cancelled, and the angle mode set to RAD. **C**

NEXT [*control-variable*]

See FOR statement. **S**

NUMBER [*initial-line*] [, *increment*]

generates sequenced line numbers starting at 100 in increments of 10. Optionally, you may specify the *initial-line* and/or *increment*. **C**

NUMERIC(*string-expression*)

returns:

– 1 if *string-expression* is a valid numeric constant.

0 if *string-expression* is not a valid numeric constant. **F**

OLD "*device.filename*"

loads the program in *filename* from *device* into memory. **C**

ON BREAK STOP
ON BREAK NEXT
ON BREAK ERROR

determines the action taken when a breakpoint occurs.

STOP: (default) halts execution of the program.

NEXT: causes breakpoints to be ignored.

ERROR: causes breakpoints to be treated as errors. **S**

ON ERROR STOP
ON ERROR *line-number*

determines the action taken when an error occurs during execution of a program.

STOP: (default) halts execution of the program.

Line-number: transfers control to the specified line when an error occurs. See RETURN. **S**

ON *numeric-expression* **GOSUB** *line-number1* [, *line-number2* ...]

transfers control to the subroutine with a beginning line number in the position corresponding to the value of *numeric-expression*. **S***

ON *numeric-expression* **GOTO** *line-number1* [, *line-number2* ...]

transfers control to the *line-number* in the position corresponding to the value of *numeric-expression*. **S***

ON WARNING PRINT
ON WARNING NEXT
ON WARNING ERROR

determines the action taken when a warning occurs.

PRINT: (default) prints a message and continues with the program.

NEXT: causes no message to be printed and the program to continue.

ERROR: causes warnings to be treated as errors. **S**

OPEN *#file-number*, "*device.filename*" [, *file-organization*]
[, *file-type*] [, *open-mode*] [, *record-length*]
enables a BASIC program to use the given *filename*. **S**

File-number: 0-255

Device.filename: peripheral device # and other device dependent information.

File-organization: RELATIVE or omitted for sequential files.

File-type: DISPLAY or INTERNAL.

Open-mode: UPDATE, INPUT, OUTPUT, or APPEND.

Record-length: VARIABLE followed by a numeric expression that specifies the maximum record length for the file. **S**

PAUSE [*numeric-expression*]
suspends program execution until the **[CLR]** or **[ENTER]** key is pressed or optionally for a specified number of seconds. **S**

PAUSE ALL
suspends program execution after each output line is sent to the display until the **[CLR]** or **[ENTER]** key is pressed. **S**

CALL PEEK(*address*, *numeric-variable1*
[, *numeric-variable2* ...])
returns values in *numeric-variable1*, *numeric-variable2*, etc. corresponding to the values in *address*, *address* + 1, etc. **S**

PI
returns the value of π as 3.14159265359. **F**

CALL POKE(*address*, *byte1* [, *byte2* ...])
writes the values of *byte1*, *byte2*, etc. in the memory location(s) specified by *address*, *address* + 1, etc. **S**

POS(*string1*, *string2*, *numeric-expression*)
returns the position of the first occurrence of *string2* in *string1*. Search begins at the position specified by *numeric-expression*. Returns zero if no match is found. **F**

PRINT [**USING** *line-number*
string-expression,] [*print-list*]

transfers optional *print-list* to the display. The optional USING specifies the format. If *string-expression* is present, it defines the format. If *line-number* is present, it refers to the line number of an IMAGE statement. See IMAGE. **S**

PRINT *#file-number* [, REC *numeric-expression*]
[, **USING** *line-number*
string-expression] [, *print-list*]

transfers *print-list* to the external file specified by *file-number*. REC directs *print-list* to the record specified in *numeric-expression*. The optional USING specifies the format. If *string-expression* is present, it defines the format. If *line-number* is present, it refers to the line number of an IMAGE statement. See IMAGE. **S**

RAD

sets the units for angle calculations to radians. **S**

RANDOMIZE [*numeric-expression*]

resets the random number generator to an unpredictable sequence. With optional *numeric-expression*, the sequence is repeatable. **S**

READ *variable-list*

assigns numeric and string constants from DATA statements to *variable-list*. **S***

RELEASE *sub-name1* [, *sub-name2* ...]

releases the specified subprogram(s), and thus releases the memory space that was reserved for the subprogram variables between subprogram calls. See ATTACH. **S**

CALL RELMEM(*numeric-expression*)

releases memory previously reserved by the GETMEM subprogram, starting with the address given in *numeric-expression*. **S**

REM [*character-string*]

indicates internal program documentation with no effect on program execution. **S**

RENUMBER [*initial-line*] [, *increment*]

renumbers lines starting at 100 in increments of 10. Optionally, you may specify the *initial-line* and/or *increment*. **C**

RESTORE [*line-number*]

indicates that the next READ operation will take data from the first DATA statement in the program or, optionally, from the first DATA statement after *line-number*. **S**

RESTORE [#*file-number* [, REC *numeric-expression*]]

resets the file pointer to the beginning of the file or, optionally, to *numeric-expression*. **S**

RETURN

transfers control from a subroutine to the statement following the corresponding GOSUB or ON GOSUB statement. **S***

RETURN [NEXT]
RETURN [*line-number*]

controls program action after an error has occurred when an ON ERROR statement has been executed.

RETURN: returns control to the statement where the error occurred and executes it again.

RETURN *line-number*: transfers control to the given line.

RETURN NEXT: transfers control to the statement after the one in which the error occurred. **S***

RND

returns a pseudo-random number greater than or equal to zero and less than one. **F**

RPT$(*string-expression*, *numeric-expression*)

returns a string that is *numeric-expression* repetitions of *string-expression* linked together. **F**

RUN [*line-number*]

RUN ["*program-name*"]

RUN ["*device.filename*"]

starts execution of a program at the lowest program statement of the program currently in memory. Optionally, the program in memory starts executing at *line-number*, *program-name* is loaded from a *Solid State Software*™ cartridge and executed, or the program in *filename* is loaded from *device* and executed. **S**

SAVE "*device.filename*" [, PROTECTED]

copies the BASIC program in memory to the given *filename* on the specified *device*. Optionally, the copied program in *filename* cannot be listed, edited, or saved. **C**

SEG$(*string-expression*, *position*, *length*)

returns a substring of *string-expression* beginning at *position* and extending for *length* characters. **F**

CALL SETLANG(*numeric-expression*)

selects the language in which system messages and errors are displayed. **S**

Numeric-expression: 0 – English
1 – German

SGN(*numeric-expression*)

returns: 1 if *numeric-expression* is positive.
0 if *numeric-expression* is zero.
– 1 if *numeric-expression* is negative. **F**

SIN(*numeric-expression*)

returns the trigonometric sine of *numeric-expression* according to the current angle mode in effect. See DEG, GRAD, and RAD. **F**

SQR(*numeric-expression*)

returns the positive square root of *numeric-expression*. **F**

STOP

terminates program execution. **S**

STR$(*numeric-expression*)

returns the string representation of the value of *numeric-expression*. **F**

SUB *subprogram-name* [(*parameter-list*)]

indicates the beginning of *subprogram-name* with optional *parameter-list*. **S***

SUBEND

indicates the end of a subprogram and transfers control from a subprogram to the statement following the CALL statement. **S***

SUBEXIT

transfers control from a subprogram to the statement following the CALL statement. **S***

TAB(*numeric-expression*)

controls column position of the output from a PRINT or DISPLAY statement. **F**

TAN(*numeric-expression*)

returns the trigonometric tangent of *numeric-expression* according to the current angle mode in effect. See DEG, GRAD, and RAD. **F**

UNBREAK [*line-list*]

removes all breakpoints or optionally those in *line-list*. **S**

USING *line-number*

USING *string-expression*

formats the *print-list* of a PRINT or DISPLAY statement. If *string-expression* is present, it defines the format. If *line-number* is present, it refers to the line number of an IMAGE statement. See IMAGE. **S**

VAL(*string-expression*)

returns the numerical value of *string-expression*. **F**

VERIFY "*device.filename*" [, PROTECTED]

checks that data was saved on an external device or loaded into memory correctly. *Device.filename* specifies the device and identifies the file. PROTECTED must be specified if the program is a protected program. **C**

CALL VERSION(*numeric-variable*)

returns a value indicating the version of BASIC that is being used. CC-40 BASIC returns a value of 10. **S**