Compsee

Doc # AP3-APR-082001

Apex III

Technical Bulletin

Apex Programming Resource (APR)

The Apex Programming Resource (APR) contains a C Library (APR-C) suitable for inclusion with small, compact, medium, and large memory model Borland® C or Microsoft® C programs.

The C Library files along with the associated include file shown below can be found on the Compsee Product Support CD supplied with the unit.

Model Type	Microsoft® C Filename	Borland® C Filename	Include Filename
Small	***[a3mscs.lib]***	***[a3bcs.lib]***	***[a3lib.h]***
Compact	***[a3mscc.lib]***	***[a3bcc.lib]***	for all
Medium	***[a3mscm.lib]***	***[a3bcm.lib]***	model
Large	***[a3mscl.lib]***	***[a3bcl.lib]***	types

The C Library, separated into functional areas, contains the following functions:

Internal Speaker Functions

Beep

Syntax: void Beep(unsigned int frequency, unsigned int duration)
Input: *frequency* in hertz (Hz), *duration* in milliseconds.
Returns: None
Notes: Turns on the internal speaker at a given *frequency* for a given *duration*. If *frequency* is 0, *Beep()* doesn't try to make a sound. It just delays for the *duration*.

BeepError

Syntax: void BeepError(void)
Input: None
Returns: None
Notes: Issues a series of beeps through the internal speaker to indicate an error condition.

BeepOn

Syntax: void BeepOn(unsigned int hertz)
Input: Frequency of the sound in *hertz*.
Returns: None
Notes: Turns on the internal speaker at a given frequency (*hertz*). To turn the speaker off after this function, call the function *BeepOff()*.

BeepOff

Syntax: void BeepOff(void)
Input: None
Returns: None
Notes: Turns the speaker off after it has been turned on by a call to *BeepOn()*.

National Sales Office
2500 Port Malabar Blvd. NE
Palm Bay, FL 32905

800-628-3888 – V
321-724-4321 – V
321-723-2895 – F
sales@compsee.com

Corporate Headquarters
400 N. Main St.
Mt. Gilead, NC 27306

Screen (Display) Functions

GetTextSize

Syntax: int GetTextSize(void)
Input: None
Returns: TEXT_8X16, TEXT_8X8, TEXT_16X8, TEXT_16X16, on success (symbolic names are defined in ***[a3lib.h]***).
–1 if attempted in non-text mode.
Notes: Determines the current font size.

SetTextSize

Syntax: int SetTextSize(int size)
Input: *size* = TEXT_8X16, to change to 8 × 16 font
TEXT_8X8, to change to 8 × 8 font
TEXT_16X8, to change to 16 × 8 font
TEXT_16X16, to change to 16 × 16 font
(symbolic names are defined in ***[a3lib.h]***)
Returns: Previous setting.
Notes: Changes the displayed font size and sets the window size accordingly. *SetTextSize()* clears the screen.

GetCursorPosition

Syntax: void GetCursorPosition(struct textcoord *cursor)
Input: The *textcoord* structure, declared in ***[a3lib.h]***, contains the following elements:
short row;
short col;
Returns: None
Notes: Loads the structure with the current cursor coordinates (row, col). The text position given by the coordinates (1, 1) is defined as the upper-left corner of the text window.

SetCursorPosition

Syntax: void SetCursorPosition(short row, short column)
Input: Cursor coordinates.
Returns: None
Notes: Sets the current text position to the display point (*row*, *column*). The text position given by the coordinates (1, 1) is defined as the upper-left corner of the text window.

ScrnSave

Syntax: int ScrnSave(int *destin)
Input: Pointer to buffer.
Returns: 0 on error, non-zero on success.
Notes: Copies the viewable area of the text mode screen to memory.
The memory space required to save the screen is:
bytes = (h rows) × (w columns) × 2
where h and w are the number of rows and columns of the current screen mode.

ScrnRecall

Syntax: int ScrnRecall(int *source)
Input: Pointer to buffer.
Returns: 0 on error, non-zero on success.
Notes: Copies text from memory to the viewable area of the text mode screen.

National Sales Office
2500 Port Malabar Blvd. NE
Palm Bay, FL 32905

800-628-3888 – V
321-724-4321 – V
321-723-2895 – F
sales@compsee.com

Corporate Headquarters
400 N. Main St.
Mt. Gilead, NC 27306

Screen (Display) Functions - Continued

GetText

Syntax: int GetText(int left, int top, int right, int bottom, int *destin)
Input: Screen coordinates and pointer to buffer.
Returns: 1 on success, 0 on error.
Notes: Stores the contents of an onscreen rectangle defined by (*left, top*) and (*right, bottom*) into the area of memory **destin*.
GetText() reads the rectangle's contents into memory from left to right and top to bottom.
All coordinates are absolute coordinates, not window relative. The upper left corner is (1, 1).
Each position onscreen takes 2 bytes of memory. The space required for a rectangle w columns wide by h rows high is: bytes = (h rows) × (w columns) × 2.

PutText

Syntax: int PutText(int left, int top, int right, int bottom, int *source)
Input: Screen coordinates and pointer to buffer
Returns: 1 on success, 0 on error.
Notes: Writes the contents of the memory area **source* to the onscreen rectangle defined by (*left, top*) and (*right ,bottom*). *PutText()* is a text-mode function performing direct video output. All coordinates are absolute screen coordinates, not window relative. The upper left corner coordinate is (1, 1).

GetVideoConfig

Syntax: void GetVideoConfig(struct video_info *vc)
Input: Pointer to structure.
Returns: None
Notes: Retrieves information about the current video setup.
Active video mode, number of screen columns, active page, and offset to next page placed in supplied structure.
The *video_info* structure, declared in ***[a3lib.h]***, contains the following elements:

```
int mode;     // active video mode
int cols;     // number of screen columns
int page;     // active video page
int pageoff;  // offset to next video page
```

National Sales Office
2500 Port Malabar Blvd. NE
Palm Bay, FL 32905

800-628-3888 – V
321-724-4321 – V
321-723-2895 – F
sales@compsee.com

Corporate Headquarters
400 N. Main St.
Mt. Gilead, NC 27306

Power Management Functions

ApmGetPwrStatus

Syntax: int ApmGetPwrStatus(struct power *powerstatus)
Input: Pointer to structure.
Returns: 0 on success, non-zero on error.
Notes: Retrieves information about the current power status. Parameters returned in the *power* structure are as follows:

unsigned char acline	=	AC Line Status
	=	00 off-line
	=	01 on-line
	=	FFh unknown
unsigned char battery	=	Battery status
	=	00 high
	=	01 low
	=	02 critical
	=	03 charging
	=	FFh unknown
unsigned char charge	=	remaining Battery Pack capacity
	=	0 to 100 (percentage of full charge)
	=	FFh unknown
unsigned char flags	=	Battery flag
		bit 0 = high
		bit 1 = low
		bit 2 = critical
		bit 3 = charging
		bit 4 = battery not present
		bit 7 = no system battery
		= FFh unknown

PowerOff

Syntax: void PowerOff(void)
Input: None
Returns: None
Notes: Powers down the Apex III. Performs the equivalent of <FN> + <Power>. Upon subsequent depression of the <Power> key the Apex III will cold boot.

PowerLow

Syntax: void PowerLow(void)
Input: None
Returns: None
Notes: Places the Apex III into a low power state.

National Sales Office
2500 Port Malabar Blvd. NE
Palm Bay, FL 32905

800-628-3888 – V
321-724-4321 – V
321-723-2895 – F
sales@compsee.com

Corporate Headquarters
400 N. Main St.
Mt. Gilead, NC 27306

Data Input Functions

GetStringE

Syntax: int GetStringE(char far *s, int maxlen, int fieldlen, struct bcdparms *bcd, char options);

Input: *s* = a character buffer (array) for storing the data string and terminating NULL.
maxlen = size of buffer including terminating NULL.
fieldlen = the size of the data field. If *maxlen* is greater than *fieldlen*, the data will scroll within the display field.
bcd = pointer to a structure containing the registered Bar Code Driver parameters. See ***[a3lib.h]*** for the *bcdparms* structure declaration.
options = the OR combination of the following bits (defined in ***[a3lib.h]***).

STR_SCANNER	Enables data entry via the scanner
STR_KEYBRD	Enables data entry via the keyboard
STR_TERMSCN	Automatically terminates data entry following a scan.
STR_CAPS	Converts keyboard data to all caps.

Returns: On error, returns:
–1 if parameter error (*fieldlen* and *maxlen* must be >0).
–2 if Bar Code Driver not loaded.
–3 if not enough memory to perform function.
On success, returns the key that caused input to terminate. This return value is the same as the return value for *Getkey()* found in the Bar Code Driver library.

Notes: Gets a string from the keyboard or scanner and echoes it to the screen.
The string is echoed at the current cursor position using the current background and foreground colors.
With the keyboard enabled (via STR_KEYBRD), any extended key, a carriage return, or a linefeed terminates data entry. An <ESCape> nulls the string and returns. Regardless of the options, this function always accepts <ENT> as a terminator.
Data entry is terminated upon reaching *maxlen*.
A NULL byte is appended to *s* to mark the end of the string. This function should not be used when scanning symbols with an embedded NULL character.
The Bar Code Driver must be loaded and the application program must link in the Bar Code Driver library.
The calling program must register an external buffer with the Bar Code Driver using *BcApiRegisterBuffer()* prior to using *GetStringE()*.
This function always returns with the <SCAN> key disabled.

ShowField

Syntax: void ShowField(int len, short row, short col)

Input: *len* = number of spaces
row, col = The display point. The text position given by the coordinates (1, 1) is defined as the upper-left corner of the text window.

Returns: None

Notes: Prints *len* number of spaces using the current foreground and background colors at the specified cursor location. On completion, the cursor is positioned at the original location.

GetchLP

Syntax: int GetchLP(void)

Input: None

Returns: The character read from the keyboard.

Notes: Gets a single character from the keyboard without echoing to the screen. Places the Apex III in a low power state while waiting on input. This function is a low power version of the standard function *getch()*.

GetcheLP

Syntax: int GetcheLP(void)

Input: None

Returns: The character read from the keyboard.

Notes: Gets a single character from the keyboard and echoes it to the current text window. Places the Apex III in a low power state while waiting on input. This function is a low power version of the standard funciton *getche()*.

National Sales Office
2500 Port Malabar Blvd. NE
Palm Bay, FL 32905

800-628-3888 – V
321-724-4321 – V
321-723-2895 – F
sales@compsee.com

Corporate Headquarters
400 N. Main St.
Mt. Gilead, NC 27306

Post Scan Processing Functions

ToEan13

Syntax: int ToEan13(char far *string, unsigned far *decodedCount, int far *codetype)
Input: Pointers to NULL terminated UPC-A string, *decodedCount*, *codetype*.
Returns: 0 on success, non-zero on failure.
Notes: Converts a UPC-A string to an EAN-13 string by adding the leading zero, changing the decoded count, and updating the code type. No overflow checking is performed.

ToIsbn

Syntax: int ToIsbn(char far *string, unsigned far *decodedCount, int far *codetype)
Input: Pointers to NULL terminated EAN-13 string, *decodedCount*, *codetype*.
Returns: 0 on success, non-zero on failure.
Notes: Converts a Bookland EAN-13 string to an ISBN string. Updates the decoded count.

ToUpca

Syntax: int ToUpca(char far *string, unsigned far *decodedCount, int far *codetype)
Input: Pointers to NULL terminated UPC-E string, decoded count, code type.
Returns: 0 on success, non-zero on failure.
Notes: Converts a UPC-E string in the form NXXXXXXC, where N is the number system character, X is a data character, and C is the checksum character, to a UPC-A string in the form NXXXXXXXXXXC by expanding per the UCC rules, changing the decoded count, and updating the code type. No overflow checking is performed.

Serial Port Functions

GetComPortMode

Syntax: unsigned char GetComPortMode(void)
Input: None
Returns: 0 = RS-232, 1 = Infrared. The symbolic names RS232 and IR are defined in ***[a3lib.h]***.
Notes: Retrieves the active serial communication mode, infrared (IR) or RS-232.

SetComPortMode

Syntax: unsigned char SetComPortMode(unsigned char mode)
Input: 0 for RS-232, 1 for Infrared.
Returns: Previous setting. The symbolic names RS232 and IR are defined in ***[a3lib.h]***.
Notes: Enables either the infrared (IR) port or the RS-232 port for serial communications.

National Sales Office
2500 Port Malabar Blvd. NE
Palm Bay, FL 32905

800-628-3888 – V
321-724-4321 – V
321-723-2895 – F
sales@compsee.com

Corporate Headquarters
400 N. Main St.
Mt. Gilead, NC 27306

Miscellaneous Functions

Backlight

Syntax: char Backlight(char state)
Input: 0 = Off, non-zero = On
Returns: Previous setting.
Notes: Turns the Apex III PDT backlight on or off. This is an Apex III PDT specific function. The symbolic names BKLTOFF and BKLTON are defined in ***[a3lib.h]***.

BootCold

Syntax: void BootCold(void)
Input: None
Returns: None
Notes: Cold boots the Apex III. The Apex III will perform the memory tests when cold booted.

BootWarm

Syntax: void BootWarm(void)
Input: None
Returns: None
Notes: Warm boots the Apex III. The Apex III will NOT perform the memory tests when warm booted.

CapsLock

Syntax: char CapsLock(char state)
Input: 0 = Off, non-zero = On
Returns: Previous setting.
Notes: Sets the Caps Lock ON or OFF.

GetDiskSerialNo

Syntax: int GetDiskSerialNo(struct diskinfo_sn *userdisk)
Input: Pointer to structure.
Returns: 0 if no error and places disk drive serial number information in structure.
Extended error number on error.
Notes: Retrieves Drive C disk serial number info level, disk serial number, volume label, and file system type.
The *diskinfo_sn* structure declared in the include file ***[a3lib.h]*** contains the following elements:

```
unsigned int infolevel;                    // info level
long int serialno;      // disk serial number (binary)
char volumelabel[12];                      // volume label or "NO NAME"
                        // if none present
                        // (NULL terminated)
char systemtype[9];  // file system type-string
                        // (NULL terminated)
```

Wait

Syntax: void Wait(unsigned int milliseconds)
Input: Number of milliseconds to delay.
Returns: None
Notes: Delays the current program from execution for the number of specified milliseconds. Analogous to the Borland C function *delay()*.

Examples

To assist developers in using the available APR functions discussed within this section, examples of each of the APR functions are located on the Compsee Product Support CD. The fully commented examples are located within the SAMPLES sub-folder of the APR directory (folder).