### The LIBRARY Feature

Using the LIBRARY feature it is possible to add user written features to MMBasic and make them permanent and part of the language. For example, you might have written a series of subroutines and functions that perform sophisticated bit manipulation; these could be stored as a library and become part of MMBasic and perform the same as other built in functions that are already part of the language.

To install a library you need to write and test the routines as you would with any normal BASIC routines. When you are satisfied that they are working correctly you can use the LIBRARY SAVE command. This will save the routines in a hidden part of flash memory where they will be available to any BASIC program but they will not show when the LIST command is used and will not be deleted when a new program is loaded or NEW is used. However the saved subroutines and functions can be called from within the main program and can even be run at the command prompt (just like a built in function).

Some points to note:

- Library routines act exactly like normal BASIC code and can consist of any number of subroutines, functions and CFunctions. The only difference is that they are hidden and are not deleted when a new program is loaded.
- Library routines can create and access global variables and are subject to the same rules as the main program – for example, respecting OPTION EXPLICIT if it is set.
- When the routines are saved to the library space MMBasic will compress them by removing comments, extra spaces, blank lines and the hex codes in CFunctions. This makes the library very space efficient.
- You cannot use multiple saves to the library space. You must delete the library code (see below) before you can load a new library.

To delete the routines in the library space you use the LIBRARY DELETE command. This will clear the space and return the flash memory used by the library to the general pool used by normal programs. The only other way to delete a library is to re-flash the chip with MMBasic.

You can see what is in the library by using the LIBRARY LIST command which will list the contents of the library space. In addition the MEMORY command can be used to display the amount of flash memory used by the library.

The library can also include code that is not contained within a subroutine, function or CFunction. This code (if it exists) will be run automatically when the Micromite is first powered on or reset. This feature can be used to initialise hardware features or setup MMBasic in some way. For example, if the Micromite has a real time clock attached, the library could contain the following single line of code:

```
RTC GETTIME
```

This would cause the internal clock within MMBasic to be set to the current time on every power up or reset.

As another example, you could save the following into the library:

```
CFunction CPUSpeed
00000000 3c02bf81 8c45f000 8c43f000 3c02003d 24420900 7ca51400 70a23002
3c040393 34848700 7c6316c0 00c41021 00621007 3c03029f 24636300 10430005
00402021 00002821 00801021 03e00008 00a01821 3c0402dc 34846c00 00002821
00801021 03e00008 00a01821
End CFunction
```

This would have the effect of adding a new function (called CPUSpeed) to MMBasic. You could even run it at the command prompt:

```
> PRINT CPUSpeed()
 40000000
>
```