

MM Console+ User Guide

Introduction

MM Console+ is a MMBasic program designed to run on a Micromite Plus device and provide Console and related functionality when connected to any other Micromite device thus replacing the need to connect to a PC to act as the Console. In this document, the Micromite Plus device running the MM Console+ program will be referred to as the Console MM while the Micromite device whose console it is connected to will be referred to as the Target MM.

Specifically, it provides the following functions:

1. Basic Console Access - i.e. the ability to enter MMBasic commands at the MMBasic Prompt on the Target MM and to display output messages received from the Console of the Target MM.
2. Console Logging - all console messages can be logged to a log file stored on the SD Card of the Console MM. Logging can be enabled/disabled at any time and new log messages can be appended to an existing log file if desired.
3. Console Log File Viewing - the contents of an existing log file (must be a log file created by MM Console+) can be viewed and scrolled through as well as searching for occurrences of a specific word or text string in the log file.
4. Lite Editor - provides limited editing capability for MMBasic program (.bas) files stored on the SD Card of the Console MM. The editor is intended for small tweaks, corrections and similar maintenance type activities, it is **NOT** intended to support full-blown program development from scratch (refer to the Lite Editor section of this document for more details).
5. Program Crunch - provides the ability to "crunch" an existing MMBasic program file stored on the SD Card of the Console MM. The "crunched" file is saved with a different name so it does not overwrite the original source file. Statistics for the crunch process are displayed, including the % of file space saved, the file size of both original source and crunched files (in number of lines and in bytes), as well as the elapsed time taken to perform the crunch process.
6. Program Upload - a MMBasic program file stored on the SD Card of the Console MM can be uploaded to the Target MM. This uses the XMODEM RECEIVE method.
Optionally the file may be Crunched as part of the upload process, with the crunched version of the file also being saved, as well as the crunch statistics (as described above) being displayed.

Important Notes:

1. MM Console+ is **NOT** a VT100 Terminal Emulator and as such is incompatible with the MMBasic Built-In Editor function (the EDIT command is not supported and will not be allowed to be sent to the Target MM). However, the Lite Editor within MM Console+ does provide somewhat equivalent functionality with some additional flexibility and features.
2. MM Console+ **cannot** load the MMBasic firmware to a Target MM device - this still requires a PC with PICKit, Microbridge or 1455 interface module.

Hardware Requirements

- CPU** MM Console+ is designed to run on a Micromite Plus (MM+ / MX470) device. Either the 64 or 100 pin variants are acceptable. It will probably run on the more powerful devices (MMX, ARMmite F4, ARMMite H7) but none of these have been tested, and some source code modifications would be necessary.
It will **NOT** run on Micromite MkII (MM2 / MX170) or ARMMite L4 devices.
- KEYBOARD** An external (PS/2) keyboard is required.
- LCD** A screen with a minimum resolution of 800 x 480 is required.
Touch is not used by MM Console+, however the screen can support it and it can be configured without any adverse effects.
- SD CARD** Access to an SD Card is required.
This may be using the SD Card interface on the back of the LCD or a separate SD Card holder connected to the appropriate pins (e.g. CON10 of the E100 board).
- BUZZER** A Piezo Buzzer (as optionally fitted to the E100) is not required but is *HIGHLY recommended*. MM Console+ does make use of the beep function to provide audible feedback as to when keys that are not valid in the current scenario are pressed or when certain errors are detected.
- RTC** A RTC Module is not required but is *HIGHLY recommended*. It allows log file records and file access timestamps to use realistic date/times, and the correct current date/time will be displayed on the Status Bar of the MM Console+ program.
- COM1** A connection to COM Port 1 (Tx, Rx and Ground) on the Console MM is required.
Note: The connection between the Console MM (COM1) and the Target MM (Console) is a straight Serial connection - i.e. no UART/USB modules / Microbridge / 1455 interface modules are required or used. *The Target MM must also provide direct access to the Console Tx and Console Rx signals to be able to connect to the Console MM.*
A pair of HC-12 wireless devices could be used to provide a remote serial connection method.

Connecting the Console MM to a Target MM

To enable MM Console+ to act as the Console for a Target MM device the following physical connections need to be provided:-

<u>Console MM Device</u>		<u>Target MM Device</u>
COM1: Tx	<--->	Console Rx
COM1: Rx	<--->	Console Tx
Ground *	<--->	Ground *

(* if both devices share the same power source, the Ground connection is not actually required.)

MM Console+ Config Settings

There are a few configuration settings for MM Console+. These all have default settings that should allow the program to be used without adjustment, however they can be modified via the Configuration Settings screen accessed via F11 from the Basic Console Access mode (refer subsequent section). Any changes made to settings via this menu screen must be saved by using the 'Enter' key to exit the menu, or changes can be discarded by exiting using the 'Esc' key. The configurable settings are:-

1. Cursor Color. There is a choice of 4 colours for the cursor as displayed in data entry text boxes. The colour selection is via Function keys as specified on the Settings Menu screen.

F1 - Red
F2 - Yellow (*Default*)
F3 - Green
F4 - Magenta

2. Beep Volume. The Beep (more of a Chirp really) volume can be adjusted or even switched off. Selection is via Function keys as specified on the Settings Menu screen.

F5 - Off
F6 - Loud
F7 - Medium (*Default*)
F8 - Soft

3. Target MM Console Baud Rate. The COM1 Baud Rate on the Console MM must match the Console Baud Rate on the Target MM. The dialog box to enter the required rate is accessed from the Settings Menu via F9. The default Baud Rate is 38400.

The setting made via this menu option is for the **COM1 port on the Console MM** and changes do not take effect until the MM Console+ program is restarted.

4. Display Font Size. This allows selection of the Font Size for all text apart from text in data entry fields (i.e. entered via the keyboard) and the Status Bar on the bottom line of most screens. Selection is via Function keys as specified on the Settings Menu screen. Font changes do not take effect until the MM Console+ program is restarted.

F10 - Small - uses Font #1.
F11 - Large (*Default*) - uses Font #2.

The font used for data entry is a specially modified font (loaded as #16) providing space for the cursor indicator between characters (Insert mode) or as an underline below the characters (Replace mode). This font size is fixed and cannot be changed.

The Status Bar Panels always use Font #1 (small font) and cannot be changed.

Basic Console Access

The Basic Console Access is the default mode entered when the program is run. It allows MMBasic commands to be entered in immediate mode (at the Command Prompt), which are sent to the connected Target MM for execution. Any response to such commands, or other Console output messages generated by the Target MM will be received by the Console MM and displayed in the Console Output Display area on the screen (and optionally logged).

Figure 1 shows a sample of the Basic Console Access screen.

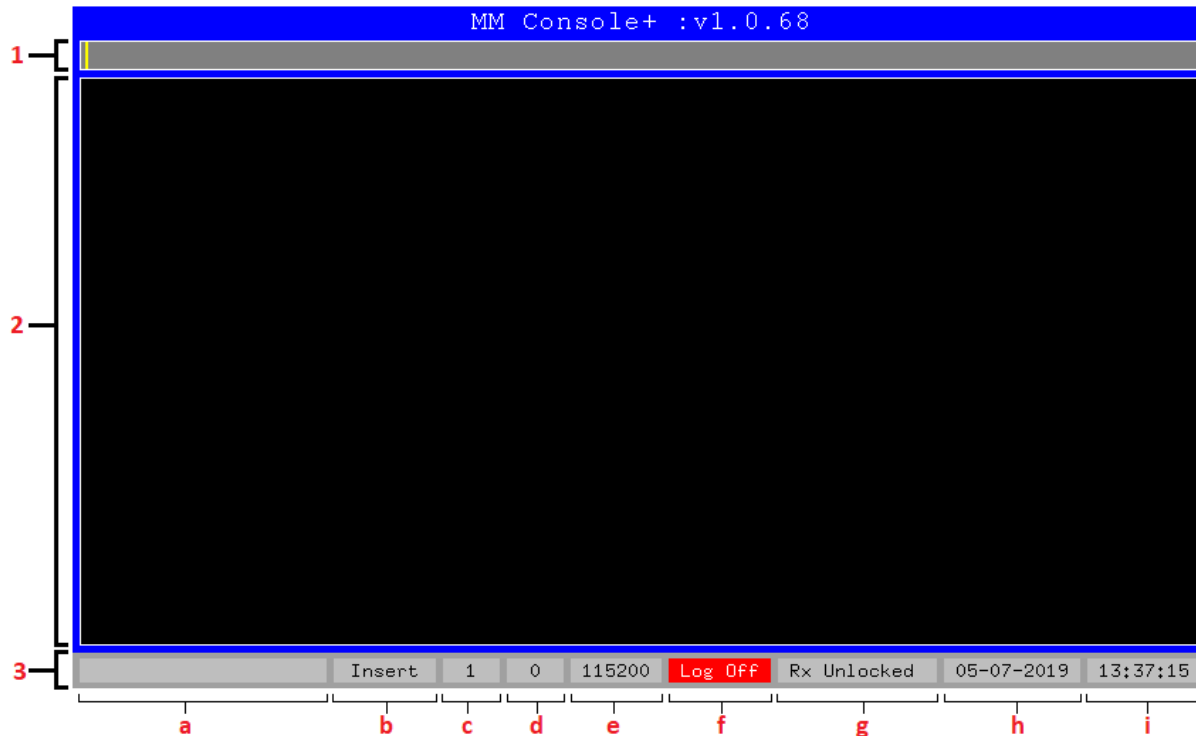


Figure 1.

The red numbers and letters down the left hand side and across the bottom of Figure 1 refer to the screen areas as described below:-

1. Command Line. Enter commands to be executed on the Target MM in this area.
Commands up to 255 characters long can be entered and the text will automatically scroll horizontally as the cursor moves.
2. Output Display Window Console output from the Target MM is displayed in this area.
Output messages up to 255 characters can be displayed, with the text being split and displayed on multiple lines if required. Lines that are continued onto the next line have a "..." continuation code displayed in the last 4 positions on the line. Up to 200 lines of output are retained and the navigation keys can be used to scroll backwards and forwards through the Display Window.
3. Status Bar The Status Bar displays some information panels at the bottom on the screen. The sub-items of the Status Bar display the following information.

Note: The text displayed on the Status Bar always uses Font #1.

- a. Notification Message Panel - displays certain messages regarding actions performed, etc. Most messages auto-clear from this panel in 15 seconds.
- b. Insert/Replace Mode - displays whether text input in the Command Line text box is using Insert Mode (default) or Replace Mode. The mode can be toggled via the *Ins* key on the keyboard. The style of the cursor (Vertical Bar or Underline) also indicates the current text input mode.
- c. Current Cursor Position within the Command Line string.
- d. Command Line text length.
- e. Baud Rate used to communicate with the Target MM device.
- f. Console Logging Status. (also uses Red/Green background).
- g. Rx Lock Status (refer description below).
- h. Current Console MM System Date.
- i. Current Console MM System Time.

Figure 2 shows a sample of the Basic Console Access screen in use as the Console for an ARMmite F4.

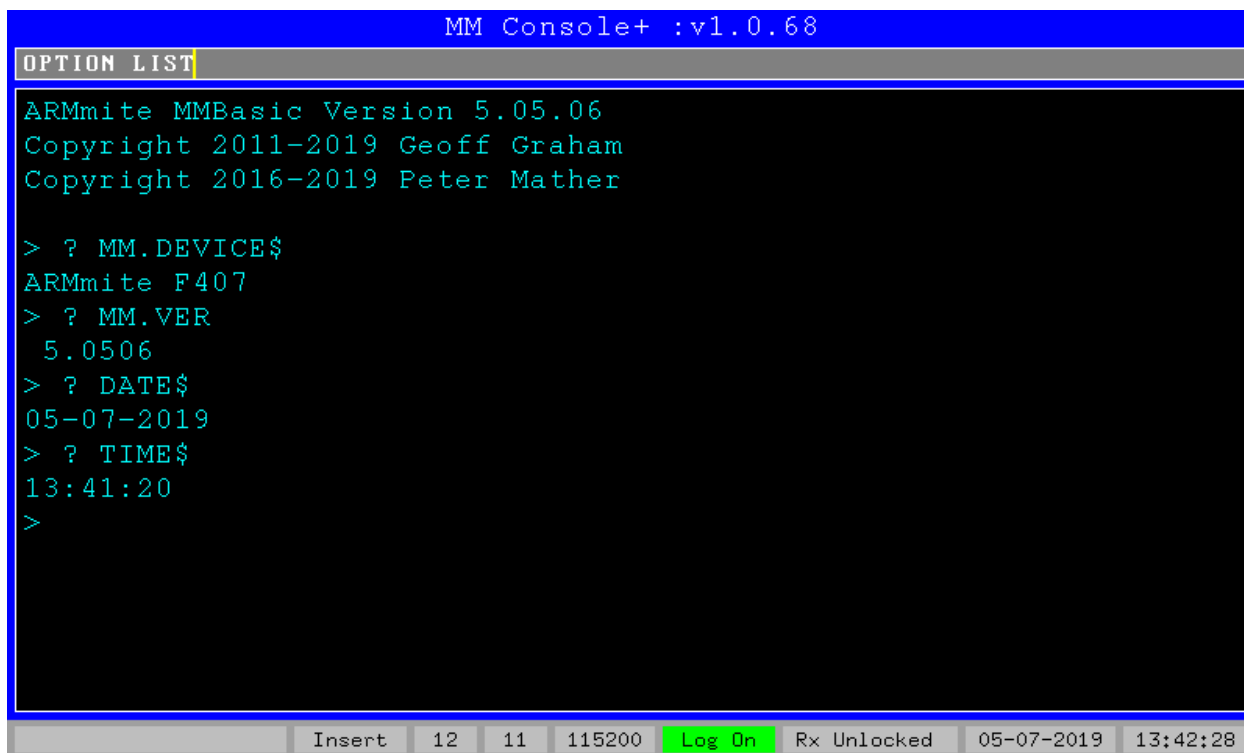


Figure 2.

Rx Lock Function

Normally, whenever data is received from the Target MM Console it is immediately displayed in the Output Display Window area on the screen. The text in this area will be automatically scrolled up so that the most recent message is visible in the Window. This can present a problem if you want to manually scroll the data in the Output Display Window back to view previous messages that are no longer in the visible portion of the window (up to 200 lines are held in a circular buffer, so you can scroll back to see messages no longer visible).

If a new message is received from the Target MM Console after you have scrolled back to view some previous message(s), the new message will cause the text in the window to auto reposition so that the new message is visible, thus you can no longer see the message(s) you had been viewing and you would need to scroll back again.

The Rx Lock function provides a way to temporarily stop new messages from being displayed - they are held in the COM1 buffer - so allowing you to be able to scroll the text in the window without it automatically repositioning again. Toggling the Rx Lock is achieved by using the F6 key. By default, the RX Lock is OFF and received data is processed immediately. When the Rx Lock is ON, the received data will tank in the COM1 buffer (up to 5K bytes). The Status Bar Panel (g) will display "Rx Lock nn%" in red font, where nn is the % of the 5K buffer that is currently used. The background colour of this panel will also change as the buffer fills up as an indicator that you could potentially lose data if you don't release the Rx Lock soon. Once the Rx Lock is released all tanked messages will be automatically processed and displayed in the Output Display Window.

Important Note:

The Rx Lock is automatically turned ON when any of the following function keys are used on the Basic Console Access screen to access other program functions: F7, F8, F9, F10 or F11.

It will also automatically be turned OFF again when the selected function is exited.

Key Assignments/Actions - Basic Console Access

Note: Keys with **Yellow** background operate on the Command Line text box.
 Keys with **Cyan** background operate on the Output Display Window.
 Keys without a background colour provide access to other functionality.

<u>Key</u>	<u>Function/Action</u>
AlphaNumerics	enter corresponding character value into the Command Line at the current cursor position.
BackSpace (BS)	remove the character preceding the current cursor position.
Enter	send the Command Line text terminated by a CR to the Target MM device.
Del	delete the character at the current cursor position.
Up Arrow	scroll Output Display Window up by 3 lines.

Down Arrow	scroll Output Display Window down by 3 lines.
Left Arrow	move the cursor left one character position.
Right Arrow	move the cursor right one character position.
Ins	toggle Insert / Replace mode.
Home	set current cursor position to Command Line character 1.
End	set current cursor position to last character on Command Line.
PgUp	scroll Output Display Window up by 1 page (+1 line).
PgDn	scroll Output Display Window down by 1 page (-1 line).
F1	send ^C (Ctrl-C or Break) to the Target MM device.
F2	send Set Date/Time to the Target MM device.
F3	clear the Command Line text box.
Shift F3	clear the Output Display Window (and circular buffer).
F4	recall previous Command Line text (up to the last 10 commands).
F5	enable/disable Console Logging.
F6	set/reset the Rx Lock flag.
F7	go to View Log File function.
F8	go to Edit Program File (Lite Editor) function
F9	go to Standalone Crunch Program File function.
F10	go to Upload Program File to Target MM function.
F11	go to MM Console+ Configuration Settings function.
F12	exit MM Console+ program.

Note: F7, F8, F9, F11 & F12 do not require a connection to a Target MM device.

Enable Console Logging (F5)

Use the F5 key from the Basic Console Access screen to enable or disable Console Logging.

When Console Logging is currently OFF, the dialog box to select or enter the required Log File Name will be displayed - see figure 3 for a sample of the Console Logging Setup dialog box. Note in this image that the Rx Lock has been turned on due to using function key F7 – see Status Bar Panel (g).

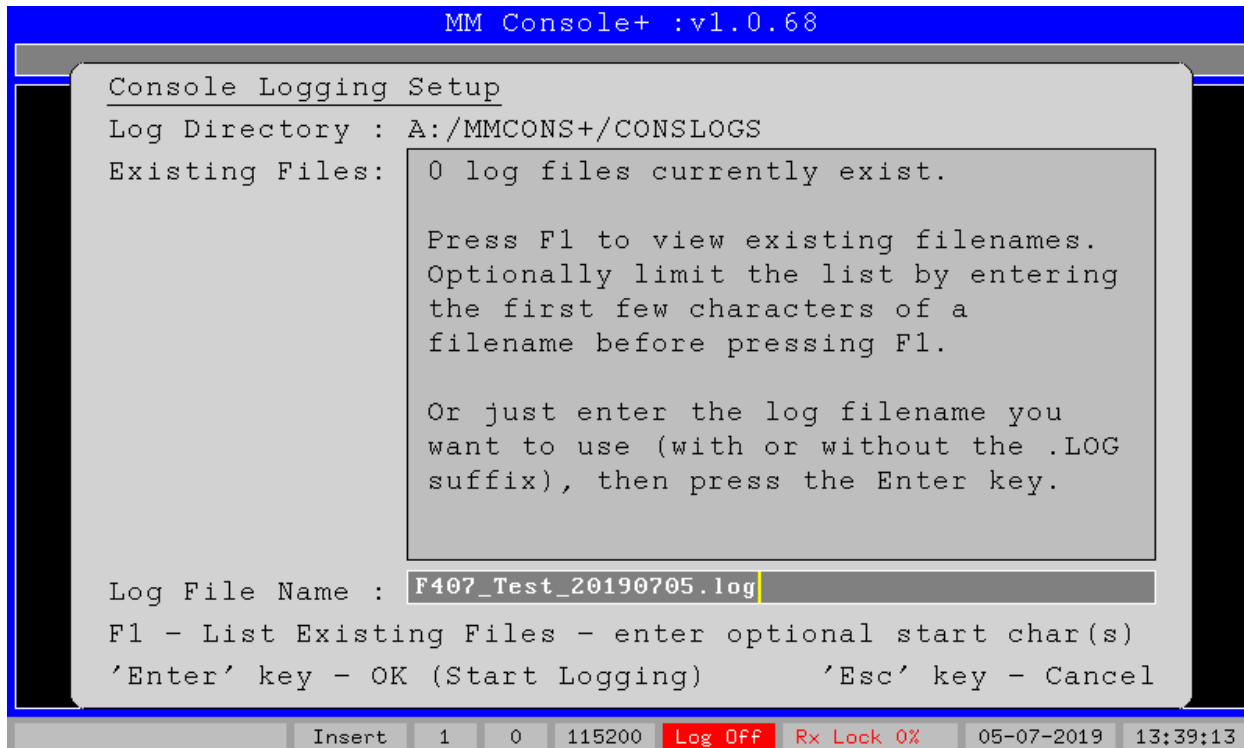


Figure 3.

You can either select an existing log file to replace or extend, or you can just enter the required log file name to be used.

Selecting an Existing Log File

To select from a list of existing Log Files, if you know the first few characters of the Log File Name you want to use, enter these in the Log File Name text box, then press F1. If you don't know the name or you want to see the list of all existing Log Files, leave the Log File Name text box blank and press F1.

In either case, pressing F1 will display a list of any existing Log Files (.log files located in the A:/MMCONS+/CONSLOGS folder) and if you had entered a partial file name only those names that match, will be displayed. Use the Down Arrow and/or Up Arrow keys to move the selection position in the list, including accessing that part of the list that may be on the next / previous page.

When the required file name is selected, press the 'Enter' key and the dialog box will be displayed to select whether to replace or extend (append to) the existing file. Once the selection has been made the Log File will be opened and Console Logging is now enabled.

Entering the Required Log File Name

Alternately, you can just enter the full name of the Log File you want to use in the Log File Name text box with or without the .log suffix, then press the 'Enter' key. If there IS an existing log file by that name, the dialog box to select whether to replace or extend (append to) the existing file or cancel to enter a different name will be displayed. Once the required selection has been made the Log File will be opened and Console Logging is now enabled.

If there is NO existing log file with the chosen name, it will be created and Console Logging is now enabled.

When Logging has been enabled, Status Bar Panel (f) will display "Log On" with a green background, and the message "Logging Enabled" will be displayed in the Notification Message Panel (a) for 15 seconds.

Key Assignments/Actions - Enable Console Logging Dialog Box

<u>Key</u>	<u>Function/Action</u>
Enter	proceed with enabling Console Logging using the File Name entered or selected.
Esc	return to Basic Console Access mode.
AlphaNumerics	enter file name characters (some keys disallowed as they are invalid in a file name).
BackSpace (BS)	backspace in the file name text box.
Del	delete the character at cursor from file name.
Left Arrow	move cursor left 1 character position.
Right Arrow	move cursor right 1 character position.
Ins	toggle Insert / Replace mode.
Home	set current cursor position to file name character 1.
End	set current cursor position to last character in the file name.
F1	display / re-display a list of existing log file names, optionally limited to matching the partial file name entered.
F3	clear the file name text box

Disable Console Logging (F5)

When Console Logging is currently ON, pressing the F5 key will display a message box advising that logging is enabled and allowing confirmation that it should be disabled (press the 'Enter' key) or leave it enabled (press the 'Esc' key).

When Logging has been disabled, Status Bar Panel (f) will display "Log Off" with a red background, and the message "Logging Disabled" will be displayed in the Notification Message Panel (a) for 15 seconds.

View Log File (F7)

The View Log File mode is entered via the F7 key from the Basic Console Access screen.

Note: Only log files created by the MM Console+ program can be viewed via this function. Additionally, it cannot be used to view the log file that is currently in use by the Console Logging function (if it is currently enabled).

It initially displays a dialog box where the file names for all existing Log Files (.log files in the A:/MMCONS+/CONSLOGS folder) are displayed. The Up / Down Arrow keys can be used to move the selected position within the list, including accessing subsequent or previous pages of the list. Once the required file has been selected (highlighted), press the 'Enter' key to proceed to the View Log File Mode screen.

Figure 4 shows a sample of the View Log File Selection dialog box.

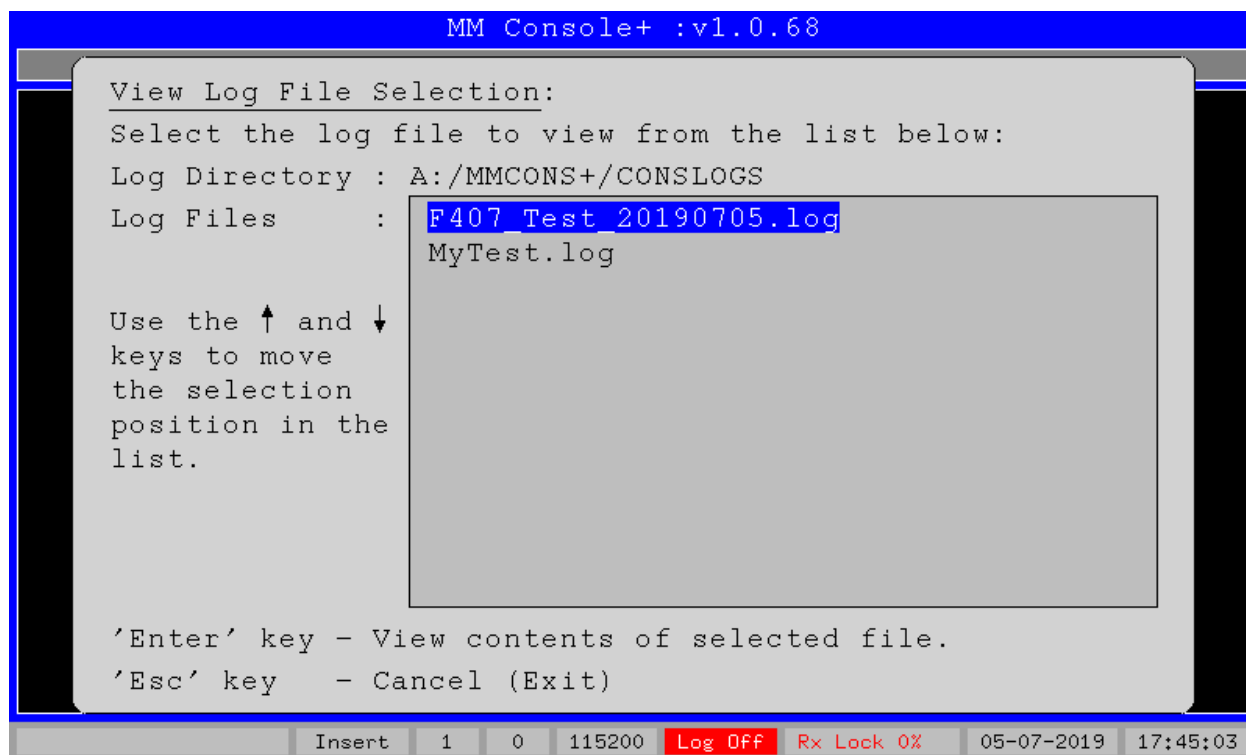


Figure 4.

Figure 5 shows a sample of the View Log File Mode screen format.

```

MM Console+ :v1.0.68 - View Log File Mode
45 13:49:41 > ? TIME$
46 13:49:41 13:49:34
47
48      Logging disabled on 05-07-2019 at 13:50:06
49      =====
50      Logging commenced on 05-07-2019 at 17:39:32
51 17:39:44 ARMrte MMBasic Version 5.05.06
52 17:39:44 Copyright 2011-2019 Geoff Graham
53 17:39:44 Copyright 2016-2019 Peter Mather
54 17:39:44
55 17:39:52 > ? TIME$
56 17:39:52 17:39:43
57 17:40:13 > RUN
58 17:40:13 Starting Demol: v0.0.00 - 08-Jun-2019
59 17:40:28 Program terminated.
60 17:40:53 > CLS RGB(WHITE)
61 17:42:04 > CIRCLE MM.HRES\2,MM.VRES\2,75,3,1,RGB(BLACK),RGB(C
62
63      Logging closed on 05-07-2019 at 17:42:51 - program t
File: F407_Test_20190705.log      4      45      63      05-07-2019      17:45:51
a      b      c      d      e      f      g      h

```

Figure 5.

The red numbers and letters down the left hand side and across the bottom of Figure 5 refer to the screen areas as described below:-

1. Log File Contents The contents of the selected Log File is displayed in this area. Up to 28 lines (using Small display font) or 19 lines (using Large display font) are displayed in the window. The navigation keys (see below) can be used to scroll backwards and forwards through the file, as well as right and left to view portions of the line contents that may not fit within the confines of the display window.

Each line displayed includes:-

1. Line Sequence Number. This number should be sequentially incrementing though the file and primarily serves to give an indication as to the position of the displayed lines in relation to the whole file contents.
2. Timestamp. The timestamp of when the record was originally processed by the MM Console+ program.

Notes: It is NOT generated from the Time value on the Target MM device when the message was originally sent.

Additionally it MAY NOT reflect the true time that it was received by the MM Console device if the Rx Lock function was on at the time. In this case the message would have been held in the COM1 buffer for an indeterminate

period until the RX Lock was released and the message able to be processed.

3. Message Text. This is the line of message text as originally displayed by MM Console+ when this Log File was created. It may be one of a group of lines for a single message, if the message could not fit on a single display line (based on the display font setting at the time the Log File was created).

2. Status Bar The Status Bar displays some information panels at the bottom on the screen. The sub-items of the Status Bar display the following information.

Note: The text displayed on the Status Bar always uses Font #1.

- a. Log File Name Panel - displays the name of the currently displayed log file (may be truncated for long file names).
- b. Current horizontal scroll value - this is the offset of the first visible character on the lines displaying log file content.
- c. Line Sequence Number - for the log file line displayed as the topmost line on the current screen.
- d. Log File Length (Lines) - displays the total number of records (lines) in the currently displayed log file.
- e. Case Match Status - used during Log Search to display the status of the 'Match Case' search option.
- f. Search Results Available Notification - used to identify when there are Log Search results that can be returned to (via F2).
- g. Current Console MM System Date.
- h. Current Console MM System Time.

Key Assignments/Actions - View Log File

<u>Key</u>	<u>Function/Action</u>
Esc	return to Basic Console Access mode.
Up Arrow	scroll up 1 line (unless already at the start of the Log File).
Down Arrow	scroll down 1 line (unless already at the end of the Log File).
Left Arrow	scroll the start position for every line left by 1 character.
Right Arrow	scroll the start position for every line right by 1 character.
Home	set the horizontal scroll to display from character 1 on every line.
End	set the horizontal scroll so that the last character on the longest line <i>in the entire Log File</i> would be visible on the right-hand edge of the screen.

PgUp	scroll to display starting one screen prior to the current screen (unless already at the start of the file).
PgDn	scroll to display the next screen after the current screen (unless already at the end of the file).
F2	return to the previous Search Results (when Search Results are available - see Status Bar Panel (f)).
F7	initiate Log Search to search for occurrences of a specified word or string within the log file contents (see below for more details).

Log Search Function

Pressing F7 while in Log File View Mode will display the dialog box for initiating a Search of the log file contents. The dialog box allows entry of the Search String and settings for 2 search related options.

Figure 6 shows the Search Log Contents dialog box.

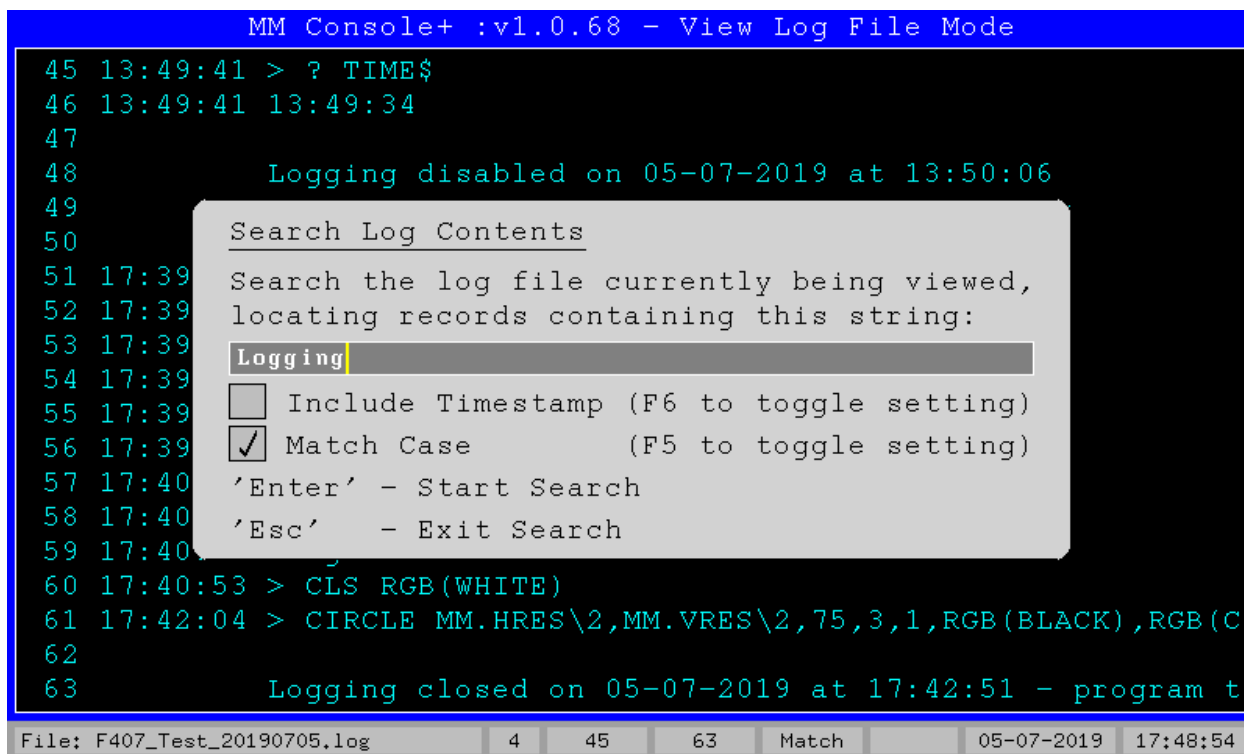


Figure 6.

1. Search String. This data entry text box allows the required target string to be entered
- up to 40 characters (using large display font)
or 33 characters (using small display font).

Note: The search matching uses the string exactly as entered (apart from case - see below). No wild-card characters or matching multiple but separate words are supported.

2. Include Timestamp When checked (use F6 to toggle setting), the search will also attempt to match the specified search string in the Timestamp portion of the log file content as well as the normal text portion.
3. Match Case When checked (use F5 to toggle setting), the search will only match on strings having the same case as the specified string.
When not checked, the search is not case sensitive.

Key Assignments/Actions - Log Search Dialog Box

<u>Key</u>	<u>Function/Action</u>
Enter	initiate Log Search using the parameters entered.
Esc	return to View Log File mode.
AlphaNumerics	enter search string characters.
BackSpace (BS)	backspace in the search string.
Del	delete character at cursor from search string.
Left Arrow	move cursor left 1 character position.
Right Arrow	move cursor right 1 character position.
Ins	toggle Insert / Replace mode.
Home	set current cursor position to search string character 1.
End	set current cursor position to last character in the search string.
F3	clear the search string data entry text box.
F5	toggle the 'Match Case' option setting.
F6	toggle the 'Incl Timestamp' option setting.

View Log Search Results

The View Log Search Results screen is displayed on a successful Log Search action.

All matches with the target search string in the log file contents (based on the search option settings) are displayed in a single list, which may not be able to be displayed in full on a single screen.

Figure 7 shows a sample of the View Log Search Results screen.

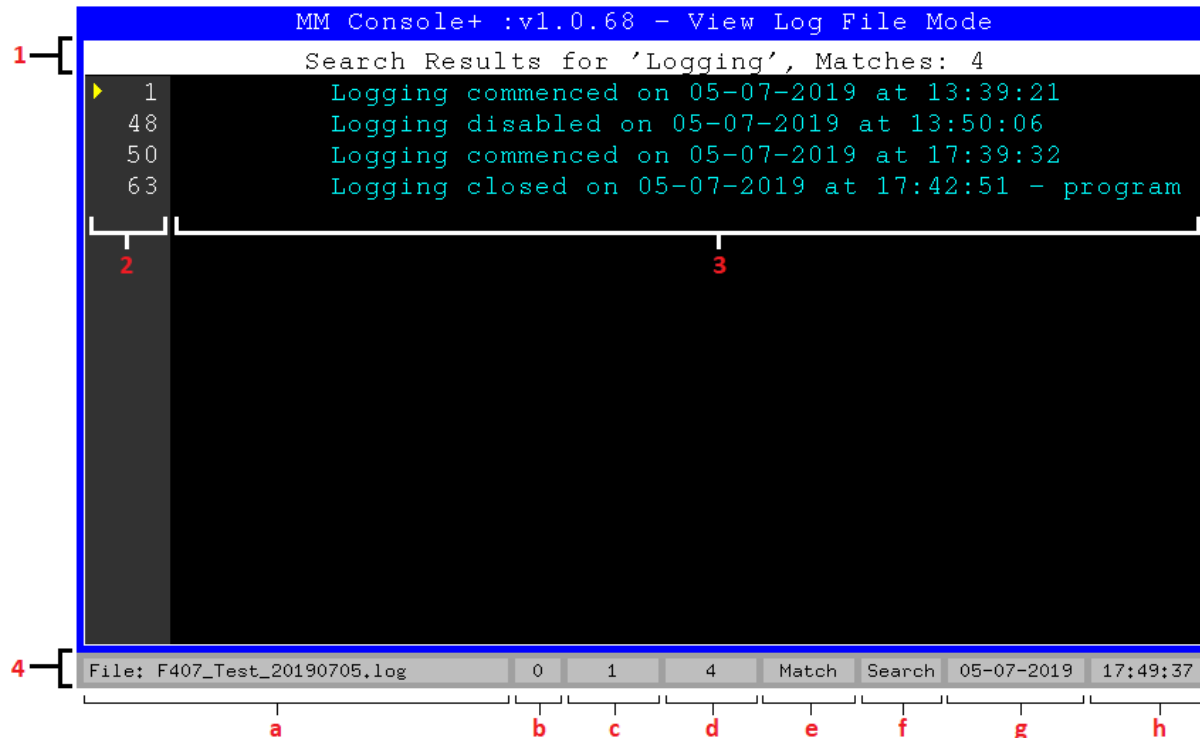


Figure 7.

The red numbers and letters down the left hand side and across the middle and bottom of Figure 7 refer to the screen areas as described below:-

- | | |
|--------------------|--|
| 1. Search Header | The Search Header displays the details of the Search String that was specified and the number of matches found. |
| 2. Line Sequence # | The darker gray column on the left displays both the Line Sequence Number for each line where a match was found and the 'selected line indicator' (right facing arrowhead symbol in the currently selected cursor colour). |
| 3. Line Contents | The black background area on the right displays the timestamp and line contents for each line where at least 1 match with the search string was found. |
| 4. Status Bar | The Status Bar uses the same format as the View Log File Status Bar (refer previous section). |

Key Assignments/Actions - View Search Results

<u>Key</u>	<u>Function/Action</u>
Enter	display the log file contents with the currently selected Search Results line positioned approximately 10 lines from the top of the page. The currently selected Search Results line is indicated by the 'selected line indicator' at the left edge of the screen.
Esc	return to the Log Search Dialog Box.
Up Arrow	move the 'selected line indicator' up by 1 line, invoking a PgUp action if used when already at the top line on the page.
Down Arrow	move the 'selected line indicator' down by 1 line, invoking a PgDn action if used when already at the last line on the page.
Left Arrow	scroll search results 1 character to the left (light gray area only), if not already starting from character 1 position.
Right Arrow	scroll search results 1 character to the right (light gray area only), if not already at the maximum scroll limit.
Home	set the horizontal scroll to display from character 1 on every line.
End	set the horizontal scroll so that the last character on the longest line <i>in the entire Log File</i> would be visible on the right-hand edge of the screen.
PgUp	scroll to display starting one screen prior to the current screen (unless already at the start of the Results List).
PgDn	scroll to display the next screen after the current screen (unless already at the end of the Results List).

Viewing the Log File for a selected Search Result line.

Figure 8 shows a sample of the View Log File Mode screen when accessed via a selected Search Results line (using the 'Enter' key – see above Key Assignment/Actions). The log file line for the selected Search Results entry is highlighted using black font on Navajo White background.

```

MM Console+ :v1.0.68 - View Log File Mode
38 13:48:35 320
39 13:48:45 > ? MM.VRES
40 13:48:45 240
41 13:49:19 > ? MM.FONTHEIGHT
42 13:49:19 13
43 13:49:33 > ? MM.FONTWIDTH
44 13:49:33 8
45 13:49:41 > ? TIME$
46 13:49:41 13:49:34
47
48 Logging disabled on 05-07-2019 at 13:50:06
49 =====
50 Logging commenced on 05-07-2019 at 17:39:32
51 17:39:44 ARmmite MMBasic Version 5.05.06
52 17:39:44 Copyright 2011-2019 Geoff Graham
53 17:39:44 Copyright 2016-2019 Peter Mather
54 17:39:44
55 17:39:52 > ? TIME$
56 17:39:52 17:39:43
File: F407_Test_20190705.log 0 38 63 Search 05-07-2019 17:51:36

```

Figure 8.

Note that since Status Bar Panel (f) is displaying "Search", using F2 on this screen will return directly to the previous Search Results list – see Figure 7 above.

Edit Program File - Lite Editor (F8)

The Lite Editor provides limited editing capability of MMBasic program (.bas) files stored on the SD Card of the Console MM and is only intended for minor changes, updates and corrections to existing program files, not for developing new programs from scratch.

The major limitations are:-

1. Original Program File Size. The Editor will only load an original program file of up to 3800 records (lines), and an error will be reported if the file size exceeds this value.
2. Record Size. The Editor will accept files with up to 120 bytes per record (+ CR/LF), and an error will be reported if this limit is exceeded by any record in the file.
3. Updates per Session Limit. The Editor only allows 290 "updates" per editing session. Refer to the section *Session Update Limits* below for more details on this topic.
4. Only basic "Find/Replace" functionality (No Find All / Replace All, No case sensitive search, No whole word search).
5. No "Undo/Redo" function.
6. No "Copy/Paste" function.
7. No "Tab/Shift Tab" function to assist with indent formatting.
8. No multi-line select for delete or formatting of code blocks.
9. No automatic capitalization of MMBasic keywords.
10. No colour coding of the various data types, keywords, etc.
11. No autosave. This means that if there is a program or system failure while there are unsaved changes pending, all unsaved changes will be lost, they cannot be recovered.
It is recommended that you use Save (F8) or Save As (F7) frequently and then open the Lite Editor again, selecting the file just saved to continue editing.

Despite these limitations the Lite Editor can still provide a useful editing capability especially since MM Console+ is not compatible with the MMBasic Built-In Editor. **Note:** Only program files already stored on the SD Card can be edited.

The Lite Editor (or Edit Program File) mode is entered via the F8 key from the Basic Console Access screen. It initially displays a dialog box where the file names for all existing Program Files (.bas files in the A:/MMCONS+/PROGS folder) are displayed. The Up / Down Arrow keys can be used to move the selected position within the list, including accessing subsequent or previous pages of the list. Once the required file has been selected (highlighted), press the 'Enter' key to proceed to the Lite Editor Mode screen.

Note: There is no provision for creating a "new" file for editing. You can only open existing .bas files located in the A:/MMCONS+/PROGS folder. If you want to be able to create small test programs you will need to provide an empty .bas file (or a small template file) and open that, then use Save As to save as a different name.

Figure 9 shows a sample of the Edit Program File Selection dialog box.

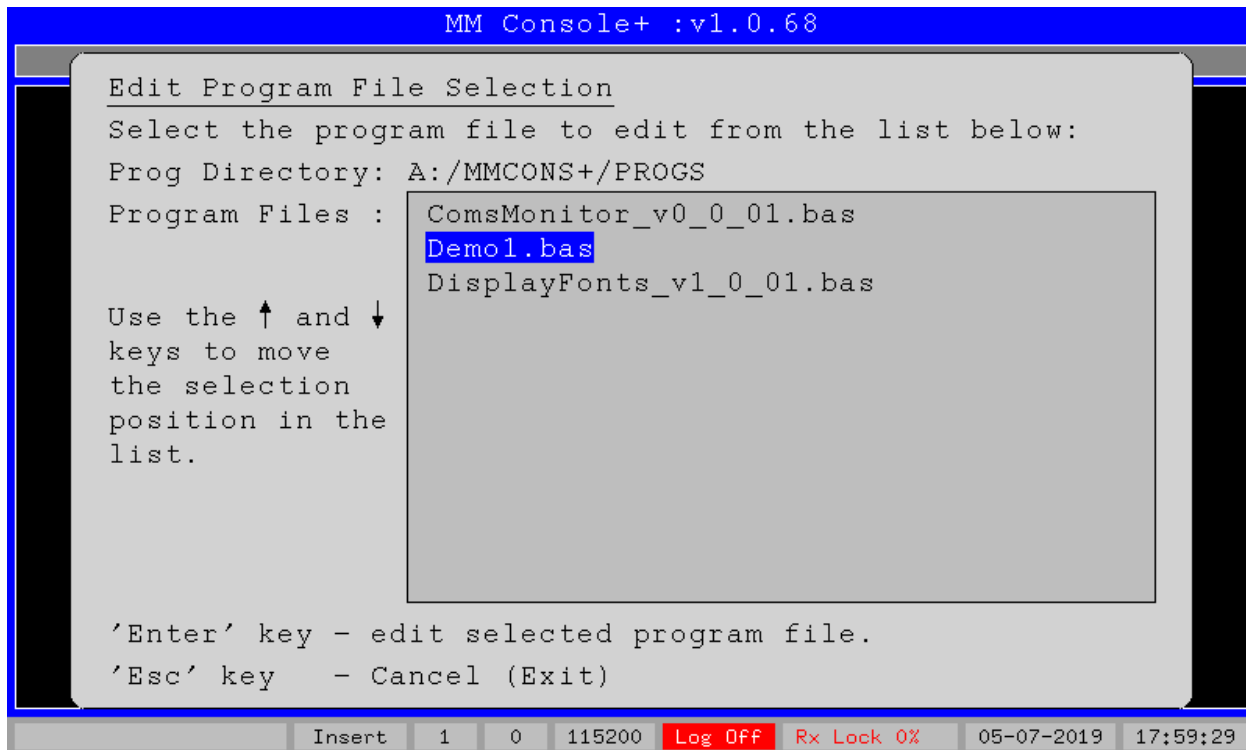


Figure 9.

Depending upon the size of the file selected, there can be a delay while the Editor prepares its internal tables, builds file pointers, etc., just be patient!! The “*LOADING*” notification message will flash in the upper left hand corner of the screen during this time as well as the File Length (Status Bar Panel (f)) updating periodically with the number of records processed so far.

Once the initialization process has completed, the first page of the source file will be displayed. Since the entire editable portion of the screen uses the fixed size data entry font (#16), the number of lines per page is 20 and number of visible characters per line is 60 regardless of the display font size selected for use by MM Console+.

Figure 10 shows a sample of the Lite Editor Mode screen.

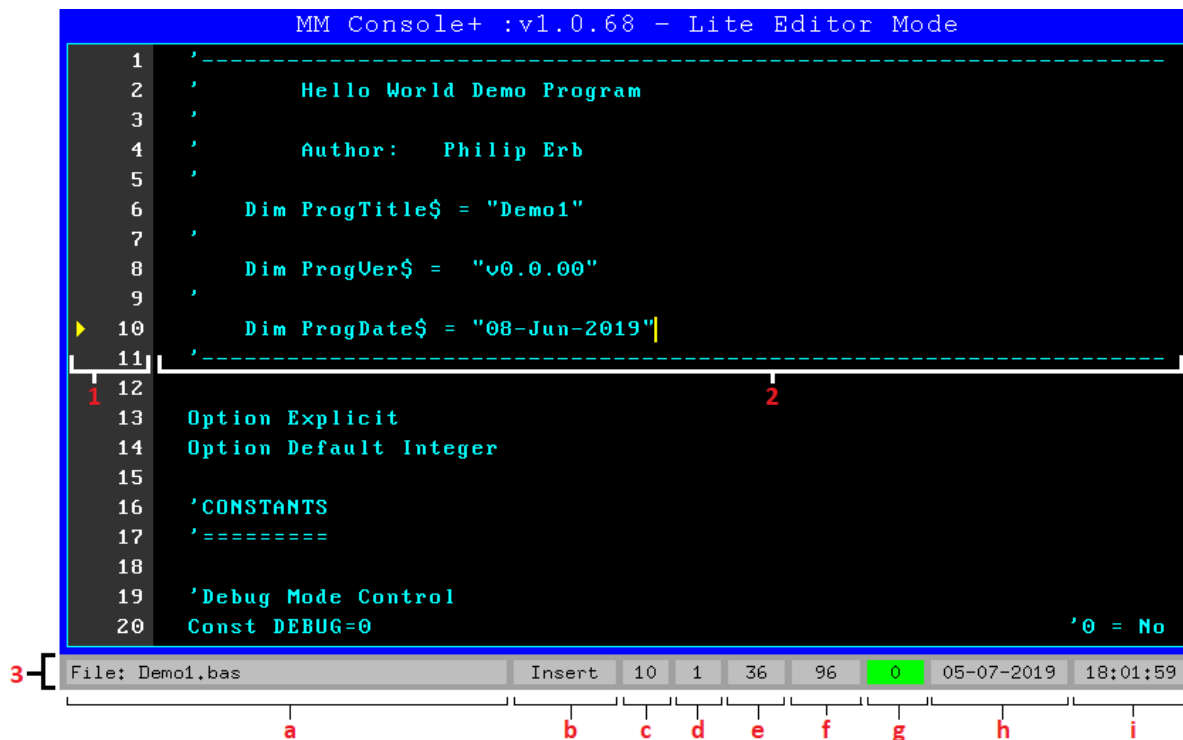


Figure 10.

The red numbers and letters on Figure 10 refer to the screen areas as described below:-

Edit Screen Format

1. Line Sequence # The dark gray column on the left displays both the Line Sequence Number for each line in the program and the 'selected line indicator' (right facing arrowhead symbol in the currently selected cursor colour).
The cursor will be positioned somewhere on the line identified by the 'selected line indicator'.
2. Line Contents The black background area on the right displays the line contents for each line. The content may be longer than can be displayed in the 60 characters available per line, in which case various navigation keys can be used to scroll the editable lines horizontally so that the previously hidden part of the content is visible.
Using navigation control keys, the cursor can be positioned to any character on any line within this area of the screen.
3. Status Bar The Status Bar displays some information panels at the bottom on the screen. The sub-items of the Status Bar display the following information.
Note: The text displayed on the Status Bar always uses Font #1.

- a. Program File Name Panel - displays the name of the program file currently selected for editing (may be truncated for long file names).
- b. Insert / Replace Mode - displays whether text input is using Insert Mode (default) or Replace Mode. The mode can be toggled via the *Ins* key on the keyboard. The style of the cursor (Vertical Bar or Underline) also indicates the current text input mode.
- c. Current Line Number - displays the line number that currently has the cursor on it (relative to the screen, top line is 1, bottom line is 20).
- d. Horizontal Scroll Value - displays the offset of the leftmost character currently being displayed (first character of the file record displays as 1).
- e. Current Character Position - displays the position within the file record for the character at the current cursor position.
- f. File Length - displays the length of the currently loaded file in records (lines).
- g. Session Updates Count - displays the number of "updates" that have been made during the current editing session. The background color changes as the count approaches the session updates limit - refer to the section *Session Update Limits* below for more details on this topic.
- h. Current Console MM System Date.
- i. Current Console MM System Time.

Key Assignments/Actions - Lite Editor (F8)

<u>Key</u>	<u>Function/Action</u>
AlphaNumerics	enter corresponding character value at the current cursor position - see also Insert/Replace mode.
BackSpace (BS)	remove the character preceding the current cursor position. If the cursor is at the first character position on a line, the current line will be merged with the preceding line as long as the merged line length does not exceed the 120 character limit.
Enter	insert a new line at the current cursor position. If the cursor is not in character position 1, the current line data will be split with the text to the left of the cursor remaining on the current line, and any text to the right of the cursor being moved to the new, inserted line.
Esc	exit the Lite Editor, returning to Basic Console Access mode. If there are unsaved changes pending, a warning message box will be displayed allowing a choice of whether to discard the changes and exit, or cancel the exit so that the changes can be saved first.

Del	delete the character at the current cursor position. If the cursor is at the last character position on a line, the current line will be merged with the following line as long as the merged line length does not exceed the 120 character limit.
Up Arrow	move the cursor up by 1 line, if possible, it will stay in the same column. If the cursor is already on the top line of the screen, an auto PgUp will be invoked to scroll to the previous screen and reposition the cursor appropriately (unless already at Line 1).
Down Arrow	move the cursor down by 1 line, if possible, it will stay in the same column. If the cursor is already on the bottom line of the screen, an auto PgDn will be invoked to scroll to the next screen and reposition the cursor appropriately (unless already at the end of the file).
Left Arrow	move the cursor left one character position, including invoking horizontal scrolling if necessary (unless already at character position 1).
Right Arrow	move the cursor right one character position, including invoking horizontal scrolling if necessary (unless already at the last character on the line).
Ins	toggle Insert / Replace mode.
Home	set current cursor position to character 1 on the current line, including invoking horizontal scrolling if necessary.
End	set current cursor position to last character on the current line, including invoking horizontal scrolling if necessary.
PgUp	scroll to the previous page, so that the current top line becomes the bottom line on the previous page screen.
PgDn	scroll to the next page, so that the current bottom line becomes the top line on the next page screen.
F1	display from the first record in the file.
F2	display the final page so that the last record in the file is the bottom line on the final page screen.
F3	clear all text from the current line, but do not delete the line.
F5	delete the current line, including any data it contains.
F7	Save As - save the edited file as a different File Name.
F8	Save - save the edited file with the original File Name (overwrites the original file).
F10	Find/Replace - display the dialog box for entry of the 'Find' target text and optionally the 'Replace' text.
F11	Find Next - finds and displays the next occurrence of the specified 'Find' target text (if any found).
F12	Replace - replace the current occurrence of the specified 'Find' target text with the specified 'Replace' text.

Session Update Limits

As noted previously, the Lite Editor only allows 290 "updates" per editing session.

In this context "updates" refer to new records inserted and changes to existing records. Deletion of unchanged records from the loaded file do NOT count as "updates" and subsequent deletion of a previously inserted or modified record does not decrement or increment the "updates" count either.

Multiple modifications to the same record only count as a single "update", as does inserting a new record and then modifying it.

Splitting a record across two lines counts as two "updates" - one for the modification to the original record and one for inserting the new record for the second part of the original data.

Merging two lines into one counts as one "update".

The current count of "updates" pending (i.e. not yet saved), is displayed in Status Bar Panel (g). As the count approaches the limit (at 10 updates left), a warning message is displayed and the background to Status Bar Panel (g) will change from green to yellow.

Another warning will be displayed when 5 updates are left, and the Status Bar Panel background will change to orange.

If the count of updates reaches the limit, a warning that the file is now being treated as READ ONLY will be displayed, the Status Bar Panel background will change to red, and only non-updating keys will be valid. The non-updating keys are:- Up Arrow, Down Arrow, Left Arrow, Right Arrow, Ins, Home, End, PgUp, PgDn, F1, F2, F7, F8, F10, F11 and Esc.

Prior to reaching the update limit, it is recommended that the editing to that point be saved, either using F7 (Save As) or F8 (Save), however even after the file is considered to be READ ONLY, the edits can still be saved using either of these function keys.

After saving the file, editing can be continued by restarting the Editor and selecting the file just saved. As this is a new editing session the updates count will be reset and a further 290 updates are available.

Note: The Saved file can exceed the 3800 maximum records size, as that only applies to the file when initially selected for editing. In theory the saved file could be as large as 4090 records (3800 original records + 290 new records inserted). If the saved file grows to be greater than 3800 records it will no longer be possible to edit that file using the MM Console+ Lite Editor function.

Find / Replace Function

While in the Lite Editor F10 provides access to the dialog box to initiate a *Find* or *Find and Replace* function.

Figure 11 shows a sample of the Set Find Target / Replace With dialog box.

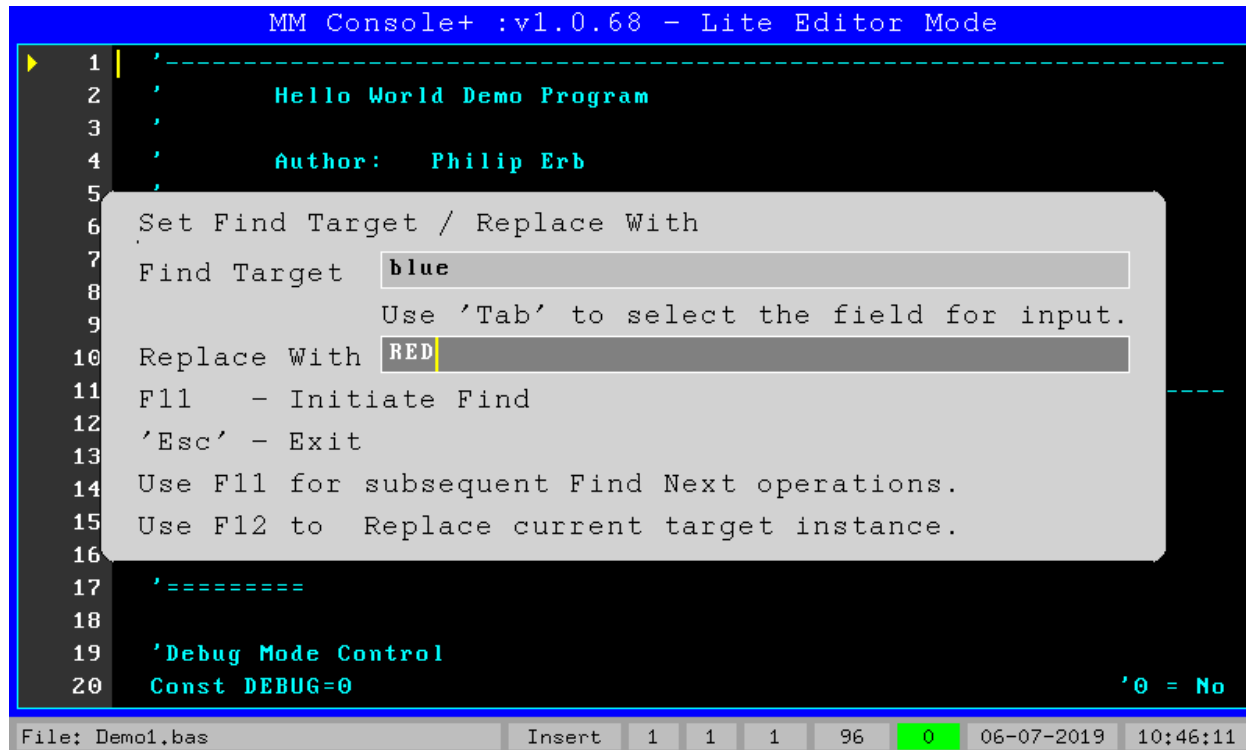


Figure 11.

Enter the Find Target string in the upper data entry textbox. If you also wish to use the Replace function, use the Tab key to move data entry to the lower data entry textbox and enter the required Replace With string – see example figure above.

Press the F11 key to initiate a search for the specified Find Target text. This search will commence from the current cursor position in the Lite Editor code window. While searching the “***SEARCHING***” notification message will flash in the upper left hand corner of the screen. Matching on the Find Target text is case insensitive and when a match is found, that portion of the code will be displayed in the Lite Editor window with the matched text highlighted – see Figure 12 below for an example.


```

MM Console+ :v1.0.68 - Lite Editor Mode
81      Case 4
82      Txt$="D"
83      End Select
84      End If
85
86      Text X+(i*32), Y, Txt$, LT, 4, 1, RGB(WHITE), RGB(BLUE) 'print curr
87
88      i=i+1                                     'ready
89      k=0                                     'reset interrupt flag (k)
90
91      If i>4 Then
92      'advance to second line or set flag to terminate
93      If j=0 Then j=1 Else j=-1
94      i=0
95      End If
96      End Sub

```

File: Demo1.bas Insert 6 1 55 96 0 06-07-2019 10:46:51

Figure 12.

If you had entered a Replace With string and you wish to replace this occurrence of the Find Target text, press F12. The highlighted text will be replaced with the previously specified Replace With text.

Note: If no Replace With text had been specified, pressing F12 will just sound a Beep and otherwise be ignored.

Press F11 to continue the search function (Find Next). If the search started at a point other than the first character of the first record of the code file, a message box may be displayed at some point indicating that the end of the file has been reached and searching can continue from the start of the file by pressing the F11 key. The search will terminate when it reaches the starting point – i.e. when the entire file has been searched, unless you choose to perform some other Lite Editor function before then.

Standalone Crunch Program File (F9)

“Crunching” refers to the process of removing all comments, blank lines and leading / trailing spaces on each line from the program file to reduce the size of the code that needs to be stored in the Target MM device’s Flash Memory. Comments and spaces, etc., are useful for human interpretation / understanding of the program code but are totally unnecessary for the MMBasic interpreter and if left in place only waste precious space in the Flash Memory area where the program code is stored.

The Standalone Crunch Program File function allows a selected program file to be crunched without being uploaded to the Target MM device (files can optionally be crunched as part of the upload process also). The original file remains unchanged and a crunched copy is automatically created and saved in the A:/MMCONS+/PROGS folder, so that either the original (un-crunched) file, or the crunched copy can be selected for a subsequent upload.

The Standalone Crunch Program File function is accessed via F9 from the Basic Console Access screen. It initially displays a dialog box where the file names for all existing Program Files (.bas files in the A:/MMCONS+/PROGS folder) are displayed. The Up / Down Arrow keys can be used to move the selected position within the list, including accessing subsequent or previous pages of the list. Once the required file has been selected (highlighted), press the 'Enter' key to proceed to the Crunch Program File mode dialog box.

Figure 13 shows a sample of the Standalone Crunch Program File Selection dialog box.

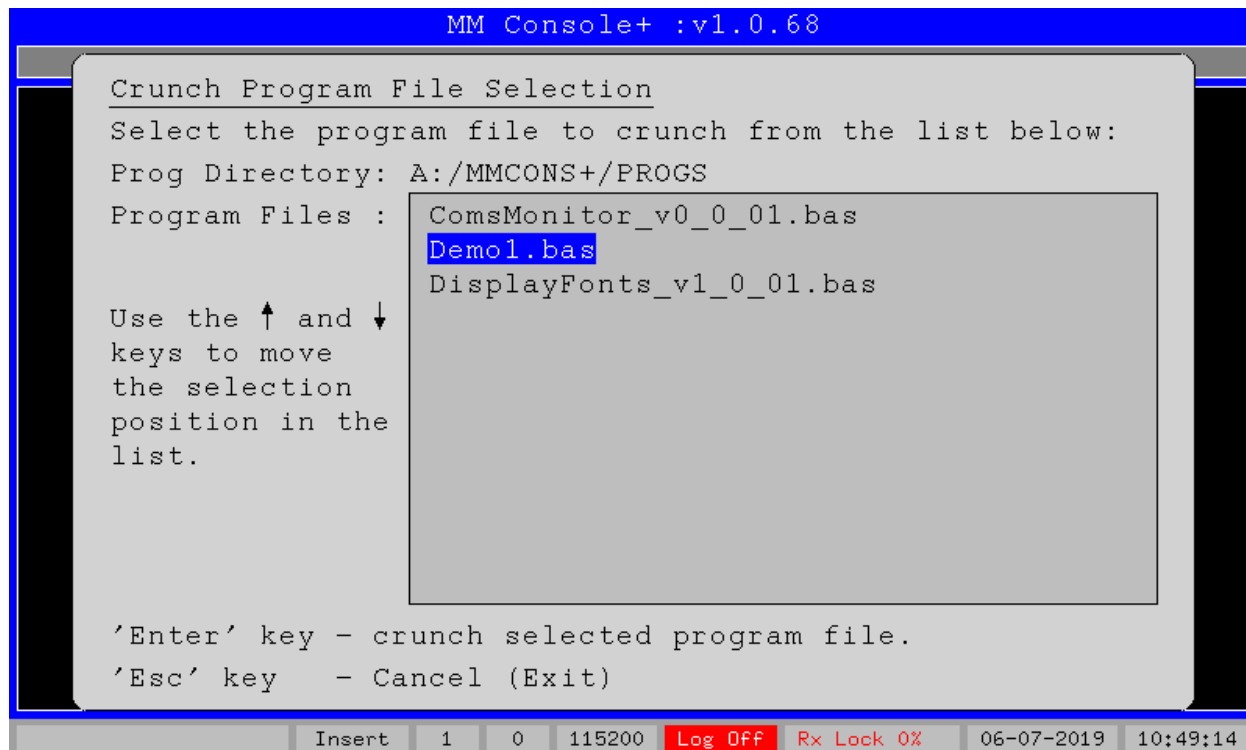


Figure 13.

The dialog box displays details for the selected file, including File Name (for confirming the correct file has been selected) and File Size (both Number of Lines and Total Bytes). It also displays preliminary details for the crunched file – mainly just the File Name that it will be saved as, which is the original File Name with ‘_Crunched’ added immediately prior to the ‘.bas’ file type suffix. Choice as to whether to Proceed with the

Crunch process ('Enter' key) or Cancel ('Esc' key) is provided – see Figure 14 below.

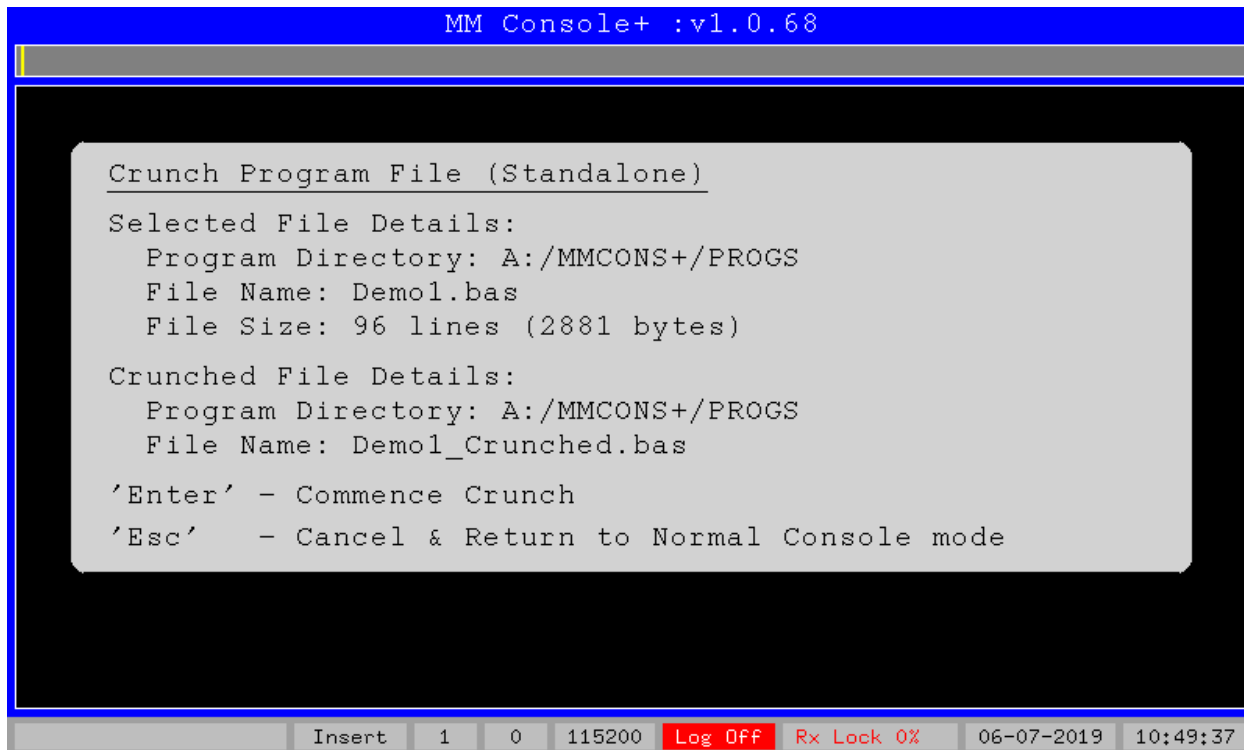
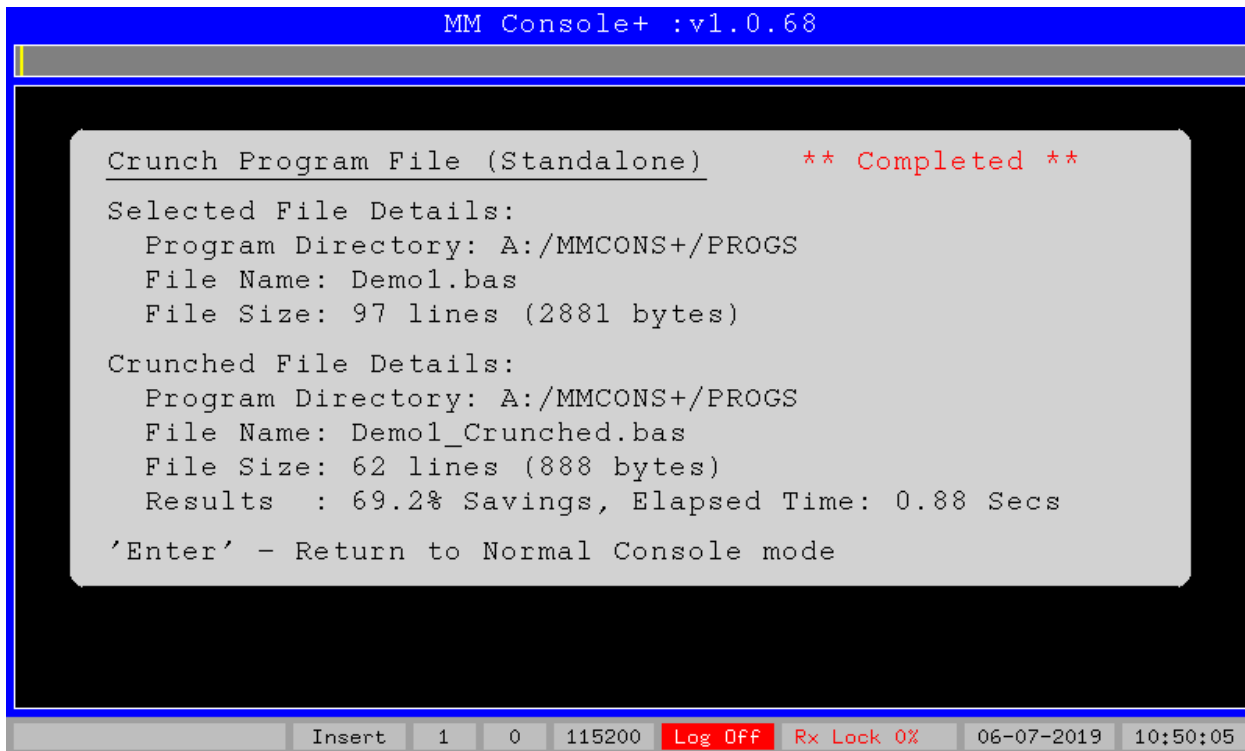


Figure 14.

When the 'Enter' key is pressed the Crunch process is started and a "Crunching File" message box is displayed until the Crunch is completed, when the dialog box will be updated to display additional crunched file details – File Size (both Lines and Bytes) for the Crunched File, the percentage of space saved (based on the Byte sizes) and the Elapsed Time that the crunch process took (in Seconds) – see Figure 15 below.



```
MM Console+ :v1.0.68

Crunch Program File (Standalone)      ** Completed **

Selected File Details:
  Program Directory: A:/MMCONS+/PROGS
  File Name: Demol.bas
  File Size: 97 lines (2881 bytes)

Crunched File Details:
  Program Directory: A:/MMCONS+/PROGS
  File Name: Demol_Crunched.bas
  File Size: 62 lines (888 bytes)
  Results : 69.2% Savings, Elapsed Time: 0.88 Secs

'Enter' - Return to Normal Console mode

Insert  1  0  115200  Log Off  Rx Lock 0%  06-07-2019  10:50:05
```

Figure 15.

There is no specific limit to the size of the file that can be crunched using the Standalone Crunch Program File function.

Press the 'Enter' key to return to the Basic Console Access mode.

Upload Program File (F10)

Uploading a Program File allows a code file stored on the SD Card of the MM Console device to be transferred to the Target MM device and saved in Flash Memory so that it can be run on that device. Optionally the selected program file being uploaded can also be crunched as part of the upload process.

The Upload Program File function is accessed via F10 from the Basic Console Access screen. It initially displays a dialog box where the file names for all existing Program Files (.bas files in the A:/MMCONS+/PROGS folder) are displayed, note that this list may contain both uncrunched and crunched program files. The Up / Down Arrow keys can be used to move the selected position within the list, including accessing subsequent or previous pages of the list.

Figure 16 shows a sample of the Upload Program File Selection dialog box.

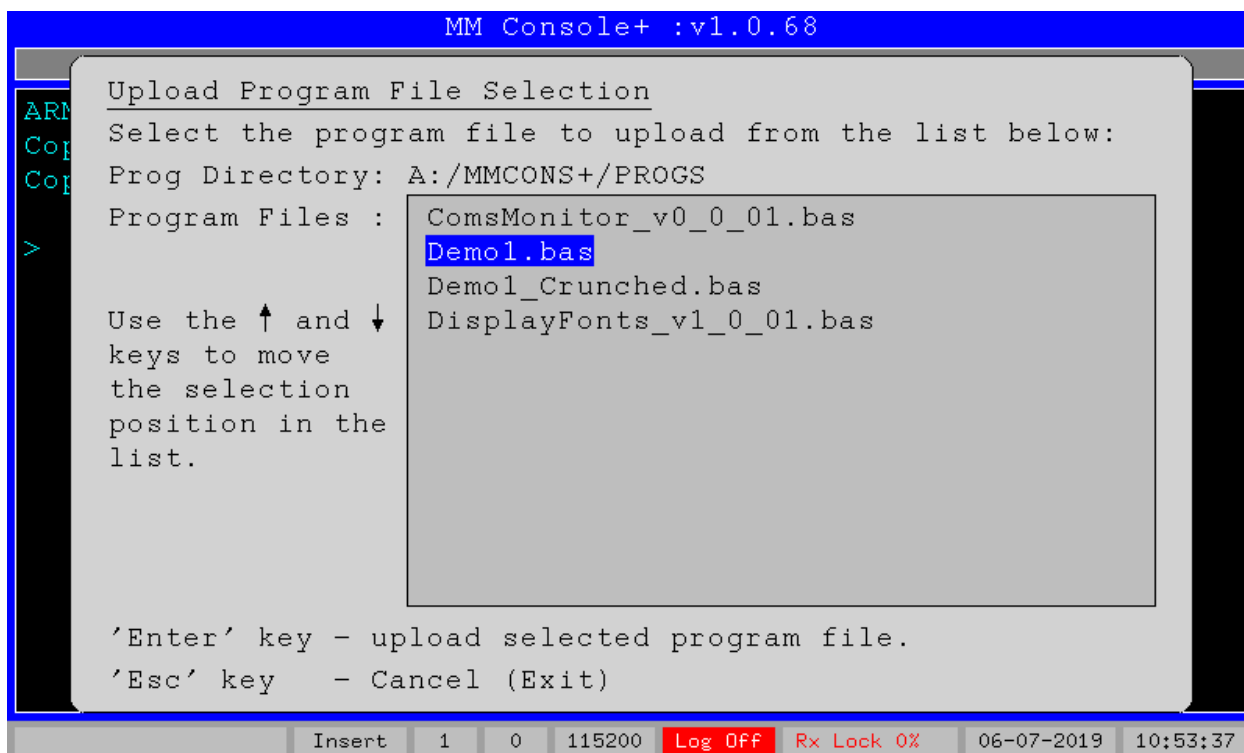


Figure 16.

Once the required file has been selected (highlighted), press the 'Enter' key to proceed to the Upload Program File options dialog box.

Figure 17 shows a sample of the Upload Program File options dialog box.

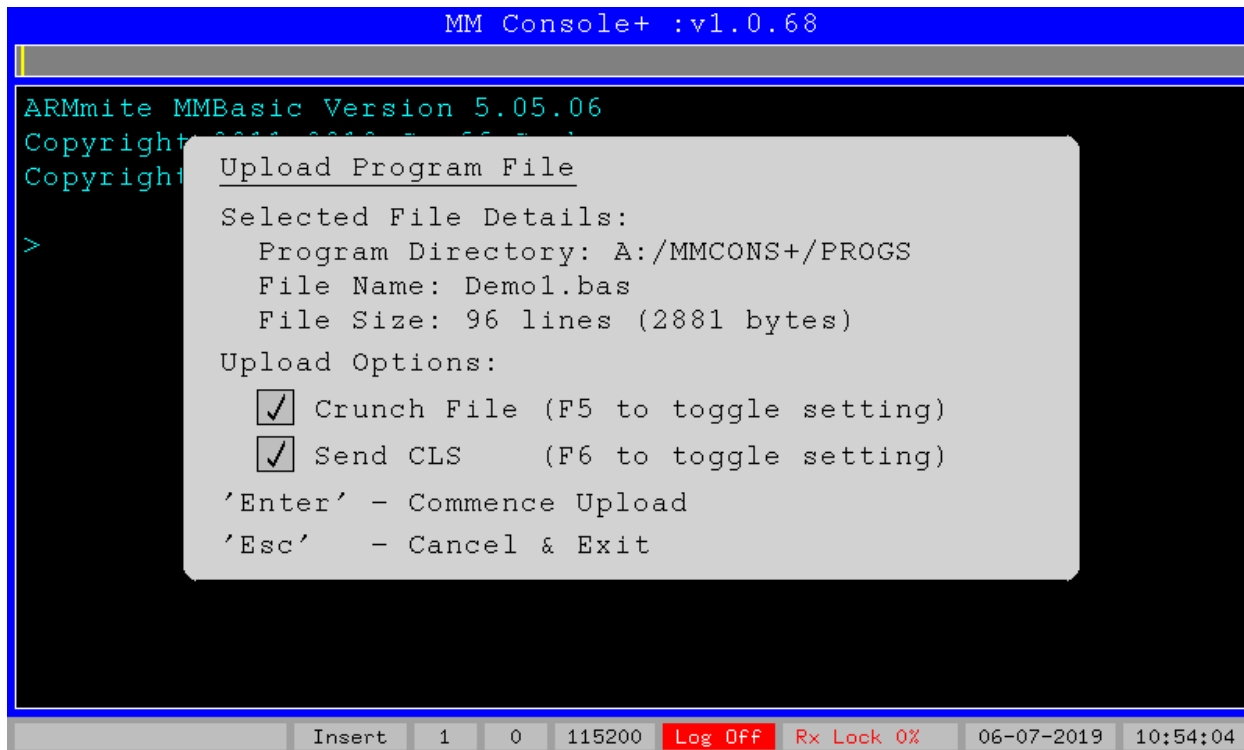


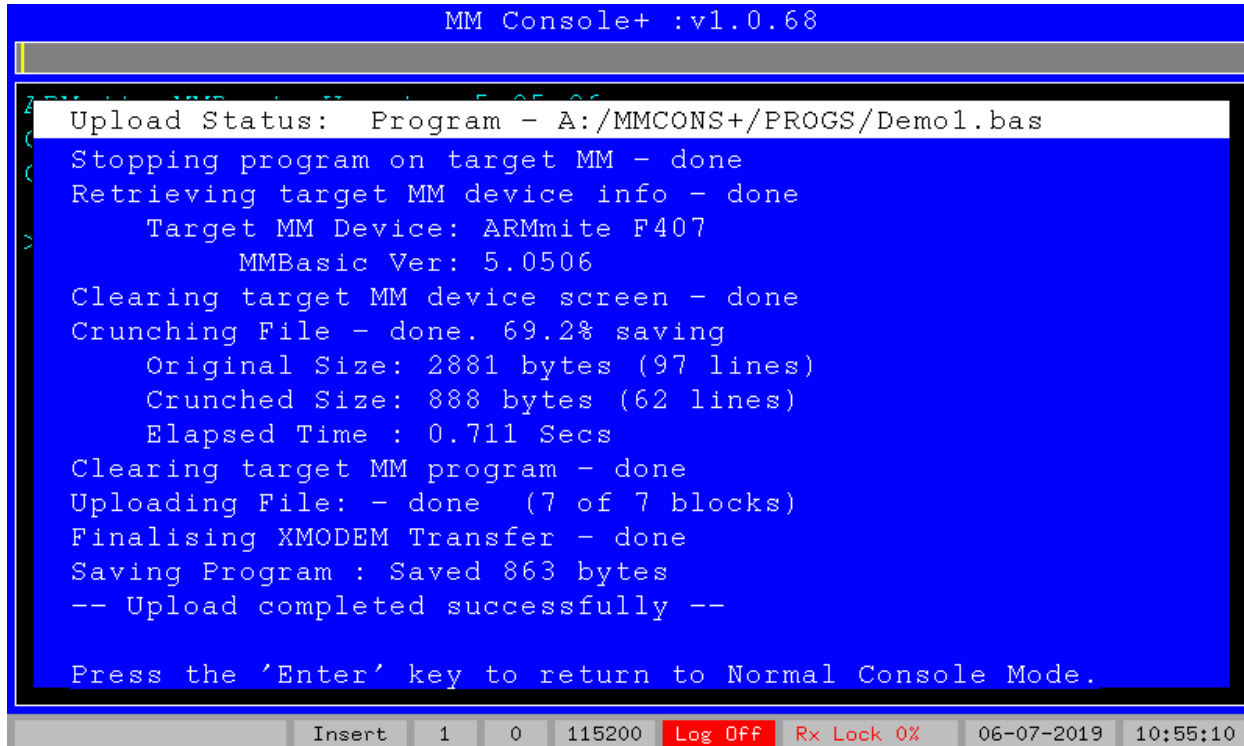
Figure 17.

The dialog box displays details for the selected file, including File Name (for confirming the correct file has been selected) and File Size (both Number of Lines and Total Bytes). It also displays the Upload Options that can be selected – Crunch File option and Send CLS option. These options can be toggled via the F5 / F6 Function Keys as noted in the dialog box.

When the Crunch File option is selected, a crunched version of the selected file will be created prior to the file upload (as described in the Standalone Crunch Program File section – see previous topic). The crunched file will be saved and will overwrite any file with the same name. The data in the crunched file will be used to upload to the Target MM device.

NOTE: There is really no point in choosing the Crunch File option if the file chosen for upload has already been crunched, however doing this should not cause any problems either.

Pressing the 'Enter' key will initiate the upload process and the Upload Status screen will be displayed as shown in Figure 18.



```
MM Console+ :v1.0.68
Upload Status:  Program - A:/MMCONS+/PROGS/Demo1.bas
Stopping program on target MM - done
Retrieving target MM device info - done
Target MM Device: ARMMite F407
MMBasic Ver: 5.0506
Clearing target MM device screen - done
Crunching File - done. 69.2% saving
Original Size: 2881 bytes (97 lines)
Crunched Size: 888 bytes (62 lines)
Elapsed Time : 0.711 Secs
Clearing target MM program - done
Uploading File: - done (7 of 7 blocks)
Finalising XMODEM Transfer - done
Saving Program : Saved 863 bytes
-- Upload completed successfully --

Press the 'Enter' key to return to Normal Console Mode.
```

Insert 1 0 115200 Log Off Rx Lock 0% 06-07-2019 10:55:10

Figure 18.

The status details will be displayed incrementally as the process runs. The above example shows a completed, successful upload with both Upload Options set. If any errors or failures are detected during the upload process, an appropriate error message will be shown at the point of failure and the update process will be terminated.

There is no specific limit to the size of the file that can be uploaded (and/or crunched), but obviously it must be of a size that is compatible with the Target MM device.

Figure 19 below shows the program uploaded to the ARMMite F4 in the example above being successfully run using the MM Console+ program as the F4 console.



```
MM Console+ :v1.0.68

ARMmite MMBasic Version 5.05.06
Copyright 2011-2019 Geoff Graham
Copyright 2016-2019 Peter Mather

> run
Starting Demol: v0.0.00 - 08-Jun-2019
Program terminated.
>
```

Insert 1 0 115200 Log Off Rx Unlocked 06-07-2019 10:55:56

Figure 19.

MM Console+ Program Installation and Setup

Pre-Instal Considerations

There are 2 items to consider before starting to install the MM Console+ program – the Data Input Font # and the Beeper Mode of Operation.

Data Input Font #

As noted earlier, MM Console+ uses a specially modified Font for all data input text fields. By default, this Font is assigned Font #16 and will be loaded into the Library area of the MM+ device running the MM Console+ program. If you already use a Font #16, or for any reason wish to use a different Font # you will need to edit the code files before loading them, as follows:-

- If you intend to load the pre-crunched files included in the release package use the following steps:
 - Open the MMConsole+Lib_Crunched.bas file using your PC based editor of choice.
 - Locate the line containing `"DefineFont #16"` – it should be on line 90.
 - Modify the Font Number to whatever value you wish to use instead of #16 and save the file.
 - Open the MMConsole+_Crunched.bas file using your PC based editor.
 - Locate the line containing `"Const FONT_IN%=16"` – it should be on line 9.
 - Modify the number assigned to this constant to match the number you used in the DefineFont command above, then save the file.

NOTE: In case you decide to use the full code files at some later time, you might want to also make the corresponding changes in the full files as well, see below for the steps.

- If you are going to use the full code files and either crunch them separately before loading or include the crunch as part of the load process (where available), use the following steps:
 - Open the MMConsole+Lib.bas file using your PC based editor of choice.
 - Locate the line containing `"DefineFont #16"` – it should be on line 260.
 - Modify the Font Number to whatever value you wish to use instead of #16 and save the file.
 - Open the MMConsole+.bas file using your PC based editor.
 - Locate the line containing `"Const FONT_IN%=16"` – it should be on line 31.
 - Modify the number assigned to this constant to match the number you used in the DefineFont command above, then save the file.

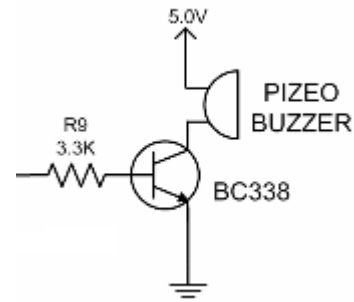
Beeper Mode of Operation

The Beeper can operate in direct mode – i.e. driven directly by any free Digital Output capable pin, or it can operate in GUI mode – i.e. using the GUI BEEP command. Depending on which mode you want to use and which pin you intend to use (for direct mode) you may need to modify the main code file to match.

If you use an E100 as the MM Console device and you have the optional Piezo Buzzer fitted (or intend to fit it),

then the Beeper uses pin 39, however you can still choose the mode of operation.

If you use a different MM+ device build you may not have a Beeper available, in which case I strongly suggest you built an outrigger beeper module so that you can benefit from the audible feedback that the beeper can provide. A suitable circuit is shown to the right and requires only 3 components and a 3 wire connection to the MM+ device (+5v, Gnd and Drive Signal). If you build such a module you can use any free Digital Output capable pin to drive the beeper. If you already have a Piezo Buzzer of some kind fitted to your MM+ device you will know which pin you use to drive it.



If you intend to use your MM+ device for other activities apart from solely the MM Console+ program you will probably want to have TOUCH configured although the MM Console+ program does not use it. If you configure TOUCH with the Click Pin option, that pin is not available to use with the Beeper direct drive option and you will need to use the GUI Mode option. If you configure TOUCH without the Click Pin option, then that pin is available and you can choose to use the direct drive option if you wish.

Even if you intend to dedicate this MM+ device to just the MM Console+ program, you may still choose to configure TOUCH with the Click Pin option so that you can use the GUI Mode anyway.

- If you intend to load the pre-crunched main program file included in the release package use the following steps:
 - Using Direct Drive Mode on an E100 with Piezo Buzzer fitted.

Note: This requires that either TOUCH is NOT configured, or is configured WITHOUT the Click Pin

 - Open the MMConsole+_Crunched.bas file using your PC based editor of choice.
 - Locate the line containing "CONST PIN_BEEP%=0" – it should be on line 6.
 - Modify the number assigned to this constant to 39, then save the file.
 - Using Direct Drive Mode on a non-E100.

Note: This requires that either TOUCH is NOT configured, or is configured WITHOUT the Click Pin

 - Open the MMConsole+_Crunched.bas file using your PC based editor of choice.
 - Locate the line containing "CONST PIN_BEEP%=0" – it should be on line 6.
 - Modify the number assigned to this constant to the pin number that you have chosen as the Drive Pin for your Piezo Buzzer module, then save the file.
 - Using GUI Mode on any MM+.

Note: This requires that TOUCH is configured WITH the appropriate Click Pin option.

 - No change to the code file is required.

NOTE: In case you decide to use the full code file at some later time, you might want to also make the corresponding changes in the full file as well, see below for the steps.

- If you are going to use the full code file and either crunch it separately before loading or include the crunch as part of the load process (where available), use the following steps:
 - Using Direct Drive Mode on an E100 with Piezo Buzzer fitted.

Note: This requires that either TOUCH is NOT configured, or is configured WITHOUT the Click Pin

 - Open the MMConsole+.bas file using your PC based editor of choice.
 - Locate the line containing `"CONST PIN_BEEP%=0"` – it should be on line 23, **COMMENT OUT** this line.
 - Locate the line containing `"CONST PIN_BEEP%=39"` – it should be on line 24, **UN-COMMENT** this line.
 - Save the file.
 - Using Direct Drive Mode on a non-E100.

Note: This requires that either TOUCH is NOT configured, or is configured WITHOUT the Click Pin

 - Open the MMConsole+.bas file using your PC based editor of choice.
 - Locate the line containing `"CONST PIN_BEEP%=0"` – it should be on line 23, **COMMENT OUT** this line.
 - Locate the line containing `"CONST PIN_BEEP%=39"` – it should be on line 24, **UN-COMMENT** this line and modify the value assigned to match the pin number that you have chosen as the Drive Pin for your Piezo Buzzer module.
 - Save the file.
 - Using GUI Mode on any MM+.

Note: This requires that TOUCH is configured WITH the appropriate Click Pin option.

 - No change to the code file is required.

SD Card Setup

The MM Console+ program requires access to a SD Card with the following directory structure setup (it can have additional folders in the root directory if you wish, MM Console+ will ignore them).

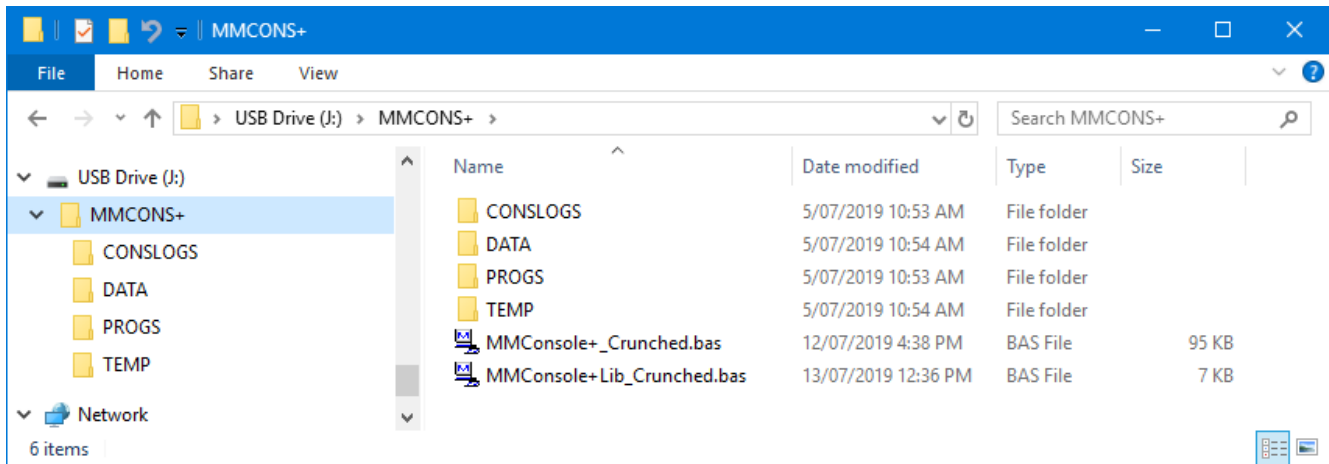


Figure 20 – SD Card Directory Structure.

Setup this directory structure on a suitable SD Card before attempting to install the program.

While not required, it is recommended that the pre-crunched versions of both the main MM Console+ code file and the associated Library file are added to the MMCONS+/ directory as shown in Figure 20. This allows loading / re-loading these files directly from the SD Card if required and is much faster than uploading from a Terminal Emulator running on your PC.

You MUST add the 'Strings.dat' file from the release package to the MMCONS+/DATA directory – see Figure 21.

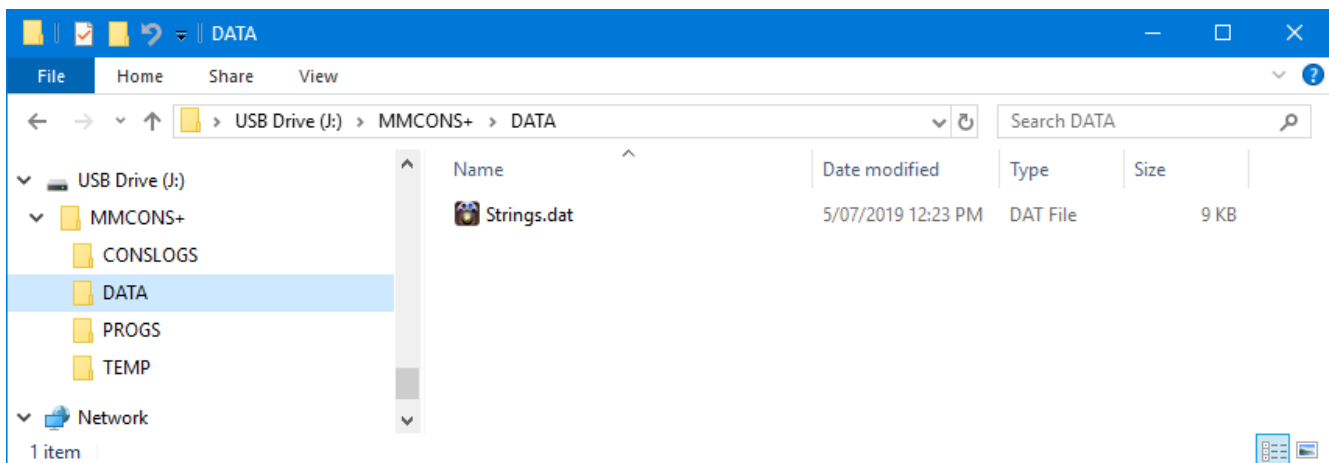


Figure 21 – Strings.dat file added to the MMCONS+/DATA directory.

You can add any code files that you wish to be able to edit, crunch and/or upload to a Target MM device to the MMCONS+/PROGS directory.

Any Console Log files that you create while running MM Console+ will be located in the MMCONS+/CONSLOGS directory. These files can be viewed by the View Log File function of the MM Console+ program or any text file editor on a PC.

The MMCONS+/TEMP directory is for temporary files used by the Lite Editor function. These files are normally deleted on exiting the Lite Editor function, however files may be left behind if the MM Console+ program crashes, the MM+ device hangs, crashes or loses power unexpectedly. Any files in the MMCONS+/TEMP directory when an Editor session is NOT active can safely be deleted – they are of no use.

LCDPANEL CONSOLE Mode Setup (Optional)

While not required, it is highly recommended that you configure the MM+ device running the MM Console+ program for LCDPANEL CONSOLE mode. With this option setup, whenever the MM+ device is not running a program, it acts as it's own Console. This means that you can use the LCDPanel and external keyboard to change OPTION settings, load other programs (if you have the required .bas files stored on the SD Card), re-initiating the MM Console+ program (or whatever program is currently loaded) if it has been terminated, etc., all WITHOUT needing a connection to a terminal emulator running on a PC. Another step in being able to do MM stuff un-tethered from a PC !!

The following OPTION setting is recommended, as it will allow LCDPANEL CONSOLE mode to “blend in” with the MM Console+ program, however you can use any settings that suit you or leave LCDPANEL CONSOLE mode disabled.

```
OPTION LCDPANEL CONSOLE 2, RGB(WHITE), RGB(BLUE)
```

Loading the Code Files

The code is split into 2 parts – 1 part loads to the Library area and the second part loads into the normal Program Flash memory area. This is partly because the combined file is too large to fit into the available RAM memory space in which it is temporarily buffered prior to saving to Flash memory. The two part load process provides a workaround for this problem.

IMPORTANT NOTES

The full version of the main MM Console+ code file is too large to fit into Flash memory (by over 3 times !!!). You MUST either load / upload the pre-crunched version of the file supplied in the release package OR you must crunch the full file either before loading or as part of the load / upload process.

Also note that there are only a few K bytes of Flash memory space not used, this being reserved for future patches that may be necessary to fix bugs. If you make any of your own modifications to the main program code it may no longer be able to run on a MM+ device and you may need to consider an MMX device.

Program Load Options

There are 3 possible ways to load the MM Console+ program code into the MM+ device:-

- Upload from a terminal emulator running on a PC connected to the MM+ device console. You can use either AUTOSAVE or XMODEM RECEIVE depending upon your favourite upload method.
- Use the RUN “filename” command. This requires that the MM Console+ code files be available on the SD Card and the SD Card accessible to the MM+ device. The MM+ needs to be either using the LCDPANEL CONSOLE option or have its console connected to a terminal emulator on a PC.
- Use the LOAD “filename” command (with/without the ‘R’ option). Again this requires that the MM Console+ code files be available on the SD Card and the SD Card accessible to the MM+ device. The MM+ needs to be either using the LCDPANEL CONSOLE option or have its console connected to a terminal emulator on a PC.

Regardless of the method used to load / upload the files you need to:

1. Load / upload the MMConsole+Lib.bas file first (choose the pre-crunched file, create your own crunched file or use an option that performs the crunch as part of the upload process).
2. When the load / upload has completed, do a LIBRARY SAVE to save this code to the MM Library.
Note: Make sure the total size used by the Library after the save does not exceed 8KB, otherwise there will be insufficient Flash memory space left to load the main MM Console+ program !!
You may need to delete pre-existing Library code if it is taking up too much space.
3. Load / upload the main MMConsole+.bas file (choose the pre-crunched file, create your own crunched file or use an option that performs the crunch as part of the upload process).
4. Ensure the SD Card containing the directory structure described previously AND has the ‘Strings.dat’ file available in the MMCONS+/DATA directory, is accessible and remains accessible while the MM Console+ program is running.

Function Key Actions on MM Console+ Main Screens

Basic Console Access Screen

Note: Keys with **Yellow** background operate on the Command Line text box.
Keys with **Cyan** background operate on the Output Display Window.

F1	send ^C (Ctrl-C or Break) to the Target MM device.	F6	set/reset the Rx Lock flag.
F2	send Set Date/Time to the Target MM device.	F7	go to View Log File function.
F3	clear the Command Line text box.	F8	go to Edit Program File (Lite Editor) function
Shift F3	clear the Output Display Window (and circular buffer).	F9	go to Standalone Crunch Program File function.
F4	recall previous Command Line text (up to the last 10 commands).	F10	go to Upload Program File to Target MM function.
F5	enable/disable Console Logging.	F11	go to MM Console+ Configuration Settings function.
		F12	exit MM Console+ program.

View Log File Mode Screen

F2	return to the previous Search Results (when Search Results are available).	F7	initiate Log Search to search for occurrences of a specified word or string within the log file contents.
-----------	--	-----------	---

Lite Editor Mode Screen

F1	display from the first record in the file.	F8	Save - save the edited file with the original File Name (overwrites the original file).
F2	display the final page so that the last record in the file is the bottom line on the final page screen.	F10	Find/Replace - display the dialog box for entry of the 'Find' target text and optionally the 'Replace' text.
F3	clear all text from the current line, but do not delete the line.	F11	Find Next - finds and displays the next occurrence of the specified 'Find' target text (if any found).
F5	delete the current line, including any data it contains.	F12	Replace - replace the current occurrence of the specified 'Find' target text with the specified 'Replace' text.
F7	Save As - save the edited file as a different File Name.		